



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería Mecánica

Desarrollo de software específico para el diseño y cálculo de zapatas continuas

Autor: ALARIO BERCIANOS, Juan Miguel

Tutor: FOCES MEDIAVILLA, Antonio

Departamento: Construcciones Arquitectónicas, Ingeniería del
Terreno y Mecánica de Medios Continuos y Teoría de Estructuras

Valladolid, julio de 2016

Resumen

El uso de dispositivos móviles, tales como tablets y smartphones, es algo común en la actualidad. Esto, unido a la creciente potencia de los mismos, hace de las aplicaciones propias para tales dispositivos herramientas cada vez más útiles y potentes. Por esto el desarrollo de las mismas se está convirtiendo en un sector en auge. En este trabajo se ha desarrollado una aplicación para dispositivos Android, tablets en concreto, que sirve para el cálculo de zapatas corridas, permitiendo así tener una herramienta de carácter portable para realizar comprobaciones rápidas sobre el terreno.

Palabras clave: aplicación, app, Android, cimentaciones, zapata corrida.

Abstract

The use of mobile devices, such as tablets and smartphones, is commonplace today. This, coupled with the increasing power thereof, makes for such applications themselves increasingly useful and powerful tools devices. Therefore, the development of them is becoming a booming sector. In this work is to develop an application for Android, tablet devices in particular, serve to calculate footings, allowing to have a character portable tool for quick checks on the spot.

Keywords : application, app, Android , concrete foundations, footing .

Índice

1-INTRODUCCIÓN:	15
1.1-JUSTIFICACIÓN:.....	15
1.2-OBJETIVOS	16
1.3-ORGANIZACIÓN DEL DOCUMENTO	16
2-ZAPATAS:	17
2.1-TIPOS Y DESCRIPCIONES:	17
2.2- ZAPATAS CORRIDAS:.....	20
2.2.1-MODOS DE FALLO:.....	21
2.2.2-DISTRIBUCIÓN DE PRESIONES SOBRE EL TERRENOS:	23
2.2.3-CÁLCULO DE LAS DIMESIONES DE LA ZAPATA:	27
2.2.4-OTRAS COMPROBACIONES:.....	32
2.3-CÁLCULO DE ARMADURAS:	36
2.3.1-CÁLCULO DE LA ARMADURA PRINCIPAL:	36
2.3.1.1-ARMADURAS RÍGIDAS:	37
2.3.1.2-ARMADURAS FLEXIBLE:	38
2.3.1.3-ANCLAJE DE LA ARMADURA PRINCIPAL:	43
2.3.3-CÁLCULO DE LA ARMADURA ESPERA:	44
3-ANDROID:	47
3.1-GENERALIDADES.....	47
3.2-PROGRAMACIÓN:	48
3.2.1-ENTORNO DE DESARROLLO:.....	48
3.2.2-FUNDAMENTOS DE PROGRAMACIÓN EN ANDROID:.....	50
3.2.2.1-ACTIVITIES:.....	51
3.2.2.2-FRAGMENTS:	53
3.2.2.3-INTERACCIÓN ACTIVITY-FRAGMENT:	55
3.2.2.4-INTERACCIÓN ENTRE ACTIVIDADES.....	57
4-PROGRAMA:	59
4.1-FUNCIONALIDADES Y FUNCIONAMIENTO DE LA APLICACIÓN:.....	59
4.2-CLASES JAVA:	65
4.2.1-CLASES PARA LA INTERFÁZ ANDROID:.....	66
4.2.2-OTRAS CLASES:	68
4.3-RECURSOS:.....	69
4.4-CÁLCULOS:	70
5-EJEMPLOS DE APLICACIONES:	73

CONCLUSIONES:.....	85
BIBLIOGRAFÍA:	87

Lista de figuras

Figura 1.1.1-Evolución de los distintos SO móviles. Fuente: http://www.idc.com/prodserv/smartphone-os-market-share.jsp	15
Figura 2.1-Tipos de cimentaciones. Fuente: "Cimentaciones"	17
Figura 2.1.1-Tipos de zapatas según estructura a soportar. Fuente: "Cimentaciones"	18
Figura 2.1.2-Tipos de zapatas según la disposición del paramento vertical. Fuente: "Cimentaciones"	19
Figura 2.1.3-Tipos de zapatas según la posición relativa la zapata del pilar/muro. Fuente: "Cimentaciones"	19
Figura 2.1.3-Tipos de zapatas según el vuelo. Fuente: "Cimentaciones"	20
Figura 2.2.1—Dimensiones principales de una zapata corrida. Fuente: "Cimentaciones"	21
Figura 2.2.1.1—Principales modos de fallo en zapatas. Fuente: "Cimentaciones"	22
Figura 2.2.2.1—Leyes de presiones en función del tipo de zapata y suelo. Fuente: "Cimentaciones"	23
Figura 2.2.2.2—Resultantes F y M2 de las cargas aplicadas sobre la zapata y reacción lineal del terreno cuando se cumple la condición de rigidez relativa. Fuente: "Cimentaciones"	25
Figura 2.2.2.3—Resultantes F y M2 de las cargas aplicadas sobre la zapata y reacción triangular del terreno cuando se cumple la condición de rigidez relativa y N se aplica fuera del núcleo central de la zapata. Fuente: "Cimentaciones"	26
Figura 2.2.2.4—Distribución rectangular y constante de tensiones aplicada sobre el ancho equivalente de la zapata. Fuente: "Cimentaciones"	26
Figura 2.2.3.2.1—Esfuerzo cortante y momento flector que actúa sobre cada sección de la zapata. Fuente: "Cimentaciones"	31
Figura 2.2.4.1—Cargas aplicadas sobre una zapata. Fuente: "Cimentaciones"	34
Figura 2.2.4.2— Ac , <i>eficaz</i> de acuerdo a la norma EHE.....	36
Figura 2.3.1.1.1—Cargas y reacciones aplicadas sobre una zapata para el método discretizado de bielas y tirantes. Fuente: "Cálculo de estructuras de hormigón armado"	37
Figura 2.3.1.2.1—Dominio 1. Fuente: "Cálculo de estructuras de hormigón armado"	39
Figura 2.3.1.2.2—Dominio 2. Fuente: "Cálculo de estructuras de hormigón armado"	40
Figura 2.3.1.2.3—Dominio 3. Fuente: "Cálculo de estructuras de hormigón armado"	40
Figura 2.3.1.2.4—Dominio 4. Fuente: "Cálculo de estructuras de hormigón armado"	40
Figura 2.3.1.2.5—Dominio 4.a. Fuente: "Cálculo de estructuras de hormigón armado"	41
Figura 2.3.1.2.6—Dominio 5. Fuente: "Cálculo de estructuras de hormigón armado"	41
Figura 2.3.1.3.1—Tipos de anclaje de la armadura. Fuente: "Cálculo de estructuras de hormigón armado"	43
Figura 2.3.3.1—Dimensiones de la armadura en espera.	45
Figura 3.1- Capas del sistema operativo Android. Fuente: wikipedia	48
Figura 3.2.1.2-Sugerencias de código	49

Figura 3.2.1.1-Renderizado en tiempo real de código XML para diseño de layouts	49
Figura 3.2.1.3-Simulador de dispositivo virtual	50
Figura 3.2.2.1.1-Ciclo de vida de una Activity. Fuente: developer.android.com/reference/android/app/Activity.html	52
Figura 3.2.2.1.2-Apilamiento de actividades. Fuente: developer.android.com/reference/android/app/Activity.html	53
Figura 3.2.2.2.1-Uso de fragments en diferentes pantallas. Fuente: https://developer.android.com/guide/components/fragments.html	53
Figura 3.2.2.2.2-Ciclo de vida de un Fragment. Fuente: https://developer.android.com/guide/components/fragments.html	54
Figura 4.1.1-Pantalla de inicio de la aplicación. Panel izquierdo deslizable.....	59
Figura 4.1.2-Pantalla para introducir nuevos datos. Panel derecho deslizable.....	60
Figura 4.1.3-Cargar dato guardado. Sobre impresión en pantalla y cargado en editables.	61
Figura 4.1.4-Guardado de nuevos datos. Aparecen disponibles en la lista deslizable.....	61
Figura 4.1.6-Pantalla inicial de comprobación dimensionado. Introducción y selección de dimensiones.....	62
Figura 4.1.5-Pantalla inicial de dimensionado óptimo. Sobre impresión de los datos cargados..	62
Figura 4.1.7-Pantalla inicial de armado en rotura. Introducción y selección de datos.	63
Figura 4.1.8-Pantalla con los resultados numéricos del módulo de Armado en rotura.....	64
Figura 4.2.1-Clases Java de la aplicación.	65
Figura 4.3.1-Clases Java de la aplicación.	69
Figura 4.4.1-Clases Java de la aplicación.	71
Figura 5.1.2-Valor de los esfuerzos nominales	73
Figura 5.1.1-Todos los datos de ejemplo que la aplicación trae cargados.....	73
Figura 5.1.3-Valor de los datos del hormigón.....	74
Figura 5.1.4-Valor de los datos del acero	74
Figura 5.1.5-Valor de los datos del terreno	75
Figura 5.1.6-Valor de los datos de la geometría.....	75
Figura 5.1.7-Pantalla donde se muestran los resultados del dimensionado óptimo	76
Figura 5.1.8-Pantalla donde se muestran las dimensiones de resultados	77
Figura 5.1.9-Pantalla donde se muestran las dimensiones de resultados	78
Figura 5.1.10-Pantalla donde se muestra el anclaje de la armadura	79
Figura 5.1.11-anclaje en gancho y con gancho prolongado	79
Figura 5.1.12-dimensiones a comprobar para el primer ejemplo.....	79
Figura 5.1.13-dimensiones a comprobar para el primer ejemplo.....	80
Figura 5.1.14-dimensiones a comprobar para el segundo ejemplo	80
Figura 5.1.15-solución del segundo ejemplo.....	80

Figura 5.1.16-Croquis de la zapata con muro excéntrico	81
Figura 5.1.17-Sobre impresión de los datos guardados, de armadura a la derecha y de materiales a la izquierda.....	82
Figura 5.1.18-Croquis con los datos de la armadura.	82
Figura 5.1.18-Datos del ejemplo 1.....	83
Figura 5.1.19-Resultados del ejemplo 1 de armado en rotura.....	83
Figura 5.1.20-Datos del ejemplo 2.....	84
Figura 5.1.20- Resultados del ejemplo 2 de armado en rotura.	84

Lista de Tablas

<i>Tabla 2.2.2.1—Valores experimentales del coeficiente de balasto de placas cuadradas de 750 mm y 300 mm de ancho. Fuente: “Cimentaciones”</i>	24
<i>Tabla 2.2.4.1—Coeficientes de seguridad parciales</i>	33
<i>Tabla 2.2.4.2—Aberturas características de fisura</i>	35

Nomenclatura

A1	Área de la armadura de tracción
A2	Área de la armadura de compresión
a2	Ancho de la zapata
<i>Ac,eficaz</i>	Área de hormigón de la zona de recubrimiento, en donde las barras a tracción influyen de forma efectiva en la abertura de las fisuras.
As	Sección total de las armaduras situadas en el área <i>Ac,eficaz</i>
b2	Unidad de longitud de cálculo de la zapata
c	Recubrimiento de las armaduras traccionadas
cd	valor de cálculo minorado de la cohesión, $coh/2$.
coh	valor de cálculo de la cohesión sin minorar
d	Canto útil de la zapata
Ec	Módulo de elasticidad medio del hormigón
eN	Excentricidad del muro respecto a la zapata
Es	Módulo de deformación longitudinal del acero.
fck	Resistencia minorada del hormigón a compresión
fck	Resistencia característica o de proyecto del hormigón a compresión
fyd	límite elástico del acero minorado para las armaduras pasivas
fyk	límite elástico del acero sin minorar para las armaduras pasivas
h	Canto total de la zapata
ht	Profundidad a la que se entierra la zapata
I	Momento de inercia de la zapata
k1	Coeficiente que representa la influencia del diagrama de tracciones en la sección Coeficiente de valor 1,0 para los casos de carga instantánea no repetida y 0,5 para los restantes.
k2	
Kc	Coeficiente o módulo de balasto del terreno
l	Longitud total de la zapata
lb	Longitud básica de anclaje
l _{disponible}	Longitud de anclaje disponible
r1	Recubrimiento geométrico lateral
r2	Recubrimiento geométrico inferior
s	Distancia entre barras longitudinales (armadura de reparto)
sm	Separación media de fisuras
Uc	Cuantía de Compresión
Us	Cuantía de tracción
v	Vuelo de la zapata
Vcu	Cortante resistido por la sección de hormigón
Vd	Cortante que actúa sobre la sección de calculo
<i>wk</i>	abertura característica de fisuración
δ	Canto útil reducido
δ_2	Recubrimiento mecánico reducido de la armadura menos traccionada
μd_1	Momento reducido con influencia del axil
ξ	Profundidad de la línea neutra reducida

u_d	Axil reducido
γ_h	Peso específico del hormigón
γ_t	Peso específico del terreno
ω_1	Cuantía geométrica de la armadura de tracción
ω_2	Cuantía geométrica de la armadura de compresión
w_{max}	abertura máxima de fisuración
β	Coeficiente que relaciona la abertura media de fisura con el valor característico
γ_s	Coeficiente de seguridad de resistencia del acero
γ_c	Coeficiente de seguridad de resistencia del hormigón
ε_1	Deformación en la armadura de tracción
ε_2	Deformación en la armadura de compresión
ε_c	Deformación del hormigón
ε_m	Alargamiento medio de las armaduras, teniendo en cuenta la colaboración del hormigón entre fisuras
ε_y	Deformación en el límite elástico
ρ_1	Cuantía geométrica de la armadura de tracción
σ_t	Distribución de tensiones del terreno
σ_s	Tensión de servicio de la armadura pasiva en la hipótesis de sección fisurada.
σ_{sr}	Tensión de la armadura en la sección fisurada en el instante en que se fisura el hormigón.

1-INTRODUCCIÓN:

1.1-JUSTIFICACIÓN:

La estandarización y amplia expansión de dispositivos como los Smartphones y Tablets, con una potencia de cálculo cada vez más importante y unas dimensiones muy reducidas, hacen de las herramientas de cálculo portables, como son las aplicaciones para dichos dispositivos, unas herramientas muy valiosas para que un ingeniero pueda complementar su trabajo de campo.

El poder acceder en cualquier parte a una herramienta de cálculo rápido para realizar comprobaciones sobre el terreno puede ser determinante a la hora de ahorrar posibles problemas en la ejecución de proyectos, pudiendo esto traducirse en ahorro de costes y tiempos, algo de mucha importancia en la tarea de un ingeniero.

Para llevar esto a cabo, se ha elegido como sistema operativo sobre el que montar la aplicación Android, debido principalmente a que este es el sistema operativo móvil más extendido hasta la fecha, copando una cuota de mercado mucho mayor que sus dos principales competidores, iOS y WindowsPhone, juntos, como podemos ver en la figura 1.1.1.

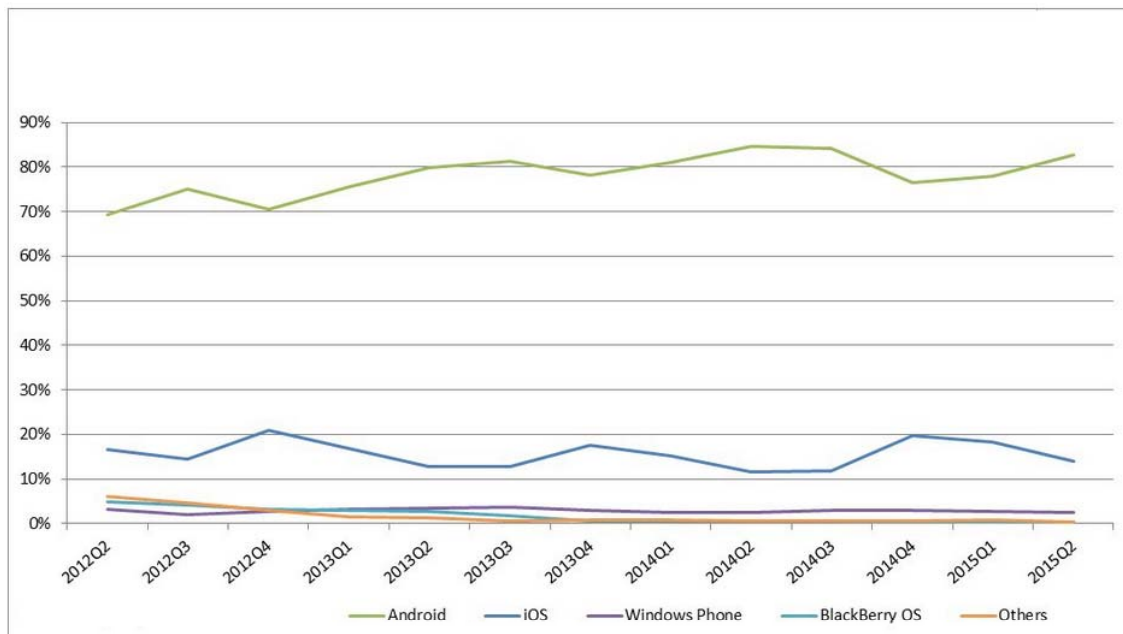


Figura 1.1.1-Evolución de los distintos SO móviles. Fuente: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>

La adaptación de ésta misma aplicación a los distintos es también posible y puede ser objeto de posteriores líneas de trabajo, pero en el trabajo presente se ha decidido optar por hacerlo solamente para Android per ser esto suficientemente extenso.

Además de esto, se ha elegido dentro de Android el formato Tablet, en concreto para pantallas con resolución a par de 720p, por considerarse su tamaño más apto para una aplicación de este tipo y considerarse también que el tamaño de pantalla de un

Smartphone convencional (entre 4,5" y 5,5") resultaría un espacio de trabajo engorroso donde no se pueden ver todos los resultados fácilmente.

1.2-OBJETIVOS

Con lo descrito en el apartado anterior en mente, en lo referente al propio trabajo, el objetivo es el de realizar una aplicación de carácter portable para el cálculo y comprobación de zapatas continuas o corridas, permitiendo así al usuario de la misma tener disponible una herramienta fuera de su espacio habitual de cálculo y realizar cálculos y comprobaciones puntuales.

Por tanto, el objetivo del trabajo es llegar a tener una aplicación que nos permita, a través de unos datos de entrada, obtener el dimensionado óptimo de una zapata continua o corrida de acuerdo a lo que norma vigente determina. Para esto, vamos a programar diferentes algoritmos en java que nos permita llegar a estas soluciones, y mediante Android construir una interfaz gráfica que nos permita verlos, interactuar con el programa y montar la aplicación sobre dispositivos que corran con este sistema operativo.

Por otro lado, como objetivo secundario busca completar la formación reglada en el plan 455 para el Graduado en Ingeniería Mecánica con conocimientos que no han sido impartidos en la misma, como son:

- Cálculo de estructuras de hormigón.
- Programación en java y Android.

1.3-ORGANIZACIÓN DEL DOCUMENTO

El trabajo consta de cuatro puntos a desarrollar, además de este primero de introducción y las conclusiones y bibliografía que cierran el mismo.

A lo largo del punto 2 referente a zapatas, se van a describir aspectos generales sobre los distintos tipos de cimentaciones existentes, para después concretar de manera más extensa el caso concreto que va a resolver nuestra aplicación, el de la zapata corrida o continua. Sobre éste se tratarán además todos los cálculos necesarios para la misma, desde los referentes a su dimensionamiento hasta los cálculos sobre la armadura a disponer. Todos ellos de acuerdo a la normativa vigente.

En el siguiente apartado, punto 3, se tratarán aspectos referentes a Android, tanto aspectos generales sobre el propio sistema operativo, como aspectos referentes a la programación de los que se hace uso en el desarrollo de la aplicación.

En cuanto al punto 4, en él se tratará la propia aplicación, describiendo su funcionamiento y las posibilidades que ofrece, así como aspectos sobre su desarrollo y funcionamiento interno.

El último de los puntos a desarrollar mostrará una serie de ejemplos realizados con la aplicación, aprovechando estos mismos para explicar los elementos que aparecen por pantalla.

2-ZAPATAS:

A lo largo de este apartado se va a hablar sobre las zapatas. Se va a describir en qué consisten estas, que tipos existen y todos los cálculos necesarios para llevar a cabo los dimensionamientos que la aplicación objeto del trabajo realiza.

Antes de comenzar a hablar sobre los diferentes tipos de zapatas que existen, se puede definir lo que es una zapata de la siguiente forma: *cimentación superficial (menos de 3 metros de profundidad) en una zona aislada de la estructura de uso habitual cuando el terreno tiene una resistencia media/alta en relación con las cargas que deben transmitirse y su superficie es suficientemente homogénea, para evitar así asentamientos diferenciales.*

En resumen, los tipos de cimentaciones existentes son los siguientes:

- **Cimentaciones superficiales:** como ya se ha dicho, con profundidades menores a tres metros. Además de las **zapatas**, también son cimentaciones superficiales **vigas** (figura 2.1-a), **emparrillados**, **losas** y cualquier combinación de ellas.
- **Cimentaciones semiprofundas:** entre tres y seis metros de profundidad. Los **pozos de cimentación** (figura 2.1-b) son un ejemplo de las cimentaciones semiprofundas.
- **Cimentaciones profundas:** para cimentaciones de más de cinco o seis metros de profundidad. Este tipo de cimentaciones se materializan a base de **pilotes** (figura 2.1-c).

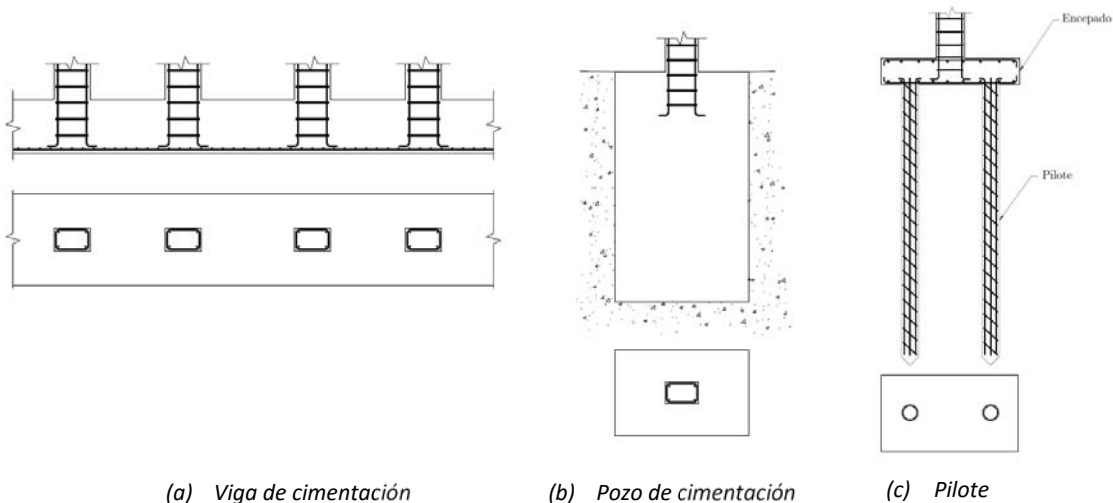


Figura 2.1-Tipos de cimentaciones. Fuente: "Cálculo de Cimentaciones"

Dentro de las zapatas también existen diferentes tipos, que serán descritos en el apartado siguiente.

2.1-TIPOS Y DESCRIPCIONES:

Genéricamente, las zapatas suelen ser de base poligonal, siendo lo más habitual que sean rectangulares, aunque en algunos casos particulares pueden ser de base circular.

Los diferentes tipos de zapatas pueden clasificarse atendiendo a diferentes criterios:

- **En función de lo que soporten:** si es uno o varios pilares, un muro, etc., que se representan en la figura 2.1.1:
 - **Aisladas:** sólo soportan un pilar (figura 2.1.1-a).
 - **Combinadas:** soportan dos o tres pilares (figura 2.1.1-b). Estas se emplean generalmente en medianerías para evitar la carga excéntrica sobre la zapata de borde o para pilares próximos entres sí, sirviendo en general para aumentar la superficie de carga, disminuyendo la presión sobre el terreno y reduciendo asentamientos diferenciales.
 - **Continuas o corridas:** soportan un muro (figura 2.1.1-c). Estas serán las que resuelva nuestra aplicación.

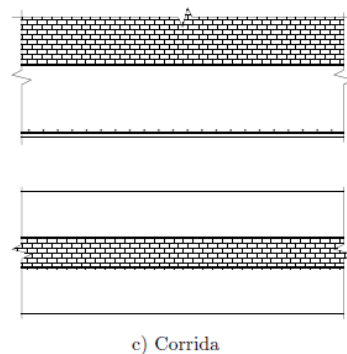
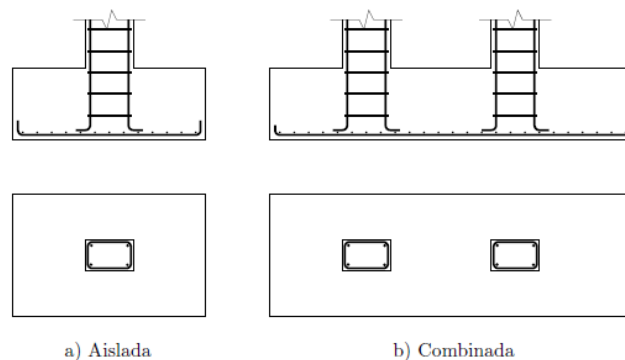


Figura 2.1.1-Tipos de zapatas según estructura a soportar. Fuente: "Cimentaciones"

- **En función de la disposición del paramento vertical:** se representan en la figura 2.1.2:
 - **Recta:** Paramentos verticales rectos y sin escalones. Es la más sencilla en cuanto a ejecución y es la que plantea la aplicación (figura 2.1.2-a).
 - **Escalonadas:** los paramentos verticales son rectos, presentando uno o varios escalones. Solución para el caso de una dimensión de planta de la zapata elevada, sirviendo para reducir la cantidad de hormigón usada aligerando así el cimientamiento (figura 2.1.2-b).
 - **Piramidales:** el paramento presenta, de abajo a arriba, un paramento inicialmente recto que luego se inclina hacia adentro con un cierto ángulo. Presentan la mayor complejidad de ejecución de los 3 tipos (figura 2.1.2-c).

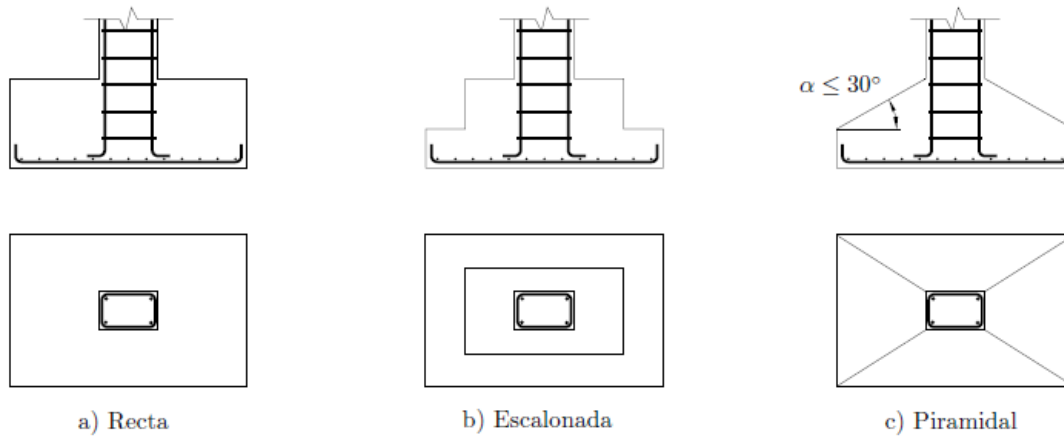


Figura 2.1.2-Tipos de zapatas según la disposición del paramento vertical. Fuente: "Cálculo de Cimentaciones"

- Según la posición relativa a la zapata del pilar/muro: se representan en la figura 2.1.3:
 - **Centrada:** eje del pilar o plano del muro pasan por el centro de gravedad de la zapata: Son las habituales de uso (figura 2.1.3-a).
 - **Excéntrica:** por exclusión, todas las que no son centradas. Interesa su uso cuando el momento transmitido a la zapata es importante. Entre las de este tipo destacan las de esquina (figura 2.1.3-b) y las de medianería (figura 2.1.3-c).

Respecto a este criterio, la aplicación es capaz de resolver los dos casos, aunque sin llegar al caso de excentricidad extremo de la zapata de esquina, por cómo hemos resuelto el anclaje de la zapata, ya que ese no es el caso a resolver por la aplicación.

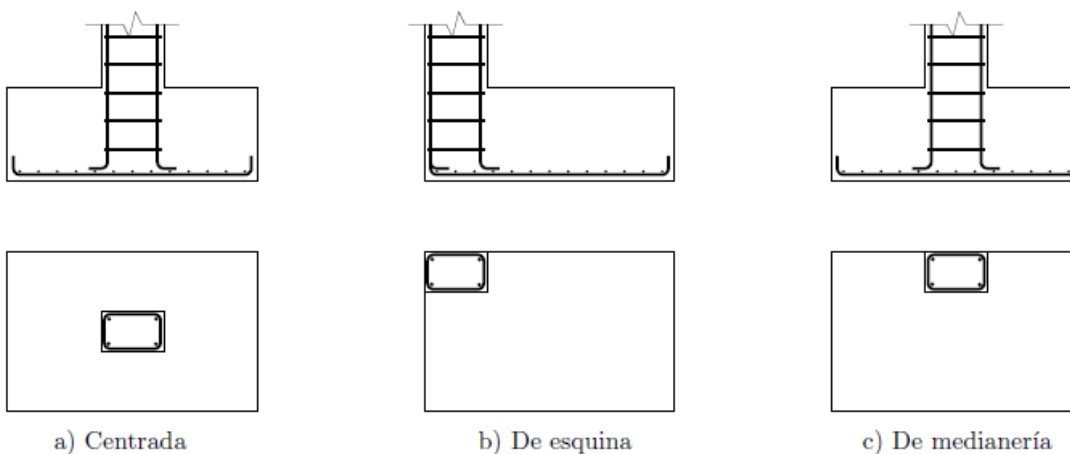


Figura 2.1.3-Tipos de zapatas según la posición relativa la zapata del pilar/muro. Fuente: "Cálculo Cimentaciones"

Además de estas tres clasificaciones, la Norma EHE hace otra clasificación para el armado de las zapatas y de los encepados (figura 2.1.4):

- **Rígidos:** elementos de gran rigidez, por lo que la distribución de presiones en el canto no es lineal a nivel de sección. Para la resolución en este caso se plantea el método de bielas

y tirantes, que se verá más adelante en este mismo apartado y con el que obtendremos la cuantía de la armadura a disponer. En este caso se supone que la zapata se comporta como sólido rígido sometido a los esfuerzos transmitidos por el pilar y a la distribución de tensiones resultantes en el terreno o a las reacciones en los pilotes.

Entre este grupo se encuentran:

- Encepados con un vuelo v en la dirección principal en la que este es máximo menor o igual a $2h$, entendiéndose que el vuelo es la distancia existente entre el borde del pilar y el eje del pilote más cercano al extremo del encepado y que h es el canto total del mismo.
 - Zapatas con un vuelo v en la dirección principal en la que este es máximo menor o igual a $2h$, entendiéndose que el vuelo es la distancia existente entre el borde del pilar y el extremo de la zapata y que h es el canto total de la misma medido en el paramento del soporte.
 - Pozos de cimentación.
 - Los elementos masivos de cimentación, como contrapesos y muros de gravedad, entre otros.
- **Flexibles:** en este caso, al contrario que en el anterior, la cimentación posee capacidad para deformarse, por lo que se supone lineal la distribución de tensiones en el canto a nivel de sección. Para armar en este caso se aplica la teoría de flexión y se puede admitir la hipótesis de sección deformada plana. Las reacciones en los pilotes y la respuesta del terreno dependen en gran medida de la flexibilidad relativa cemento-terreno, por lo que se deberá considerar un modelo de interacción suelo cemento adecuado. Esto también se verá en más profundidad a lo largo de este capítulo.

Son cimentaciones flexibles los siguientes elementos:

- Encepados cuyo vuelo v en la dirección principal de mayor vuelo es mayor que $2h$.
- Zapatas cuyo vuelo v en la dirección principal de mayor vuelo es mayor que $2h$.
- Losas de cimentación.

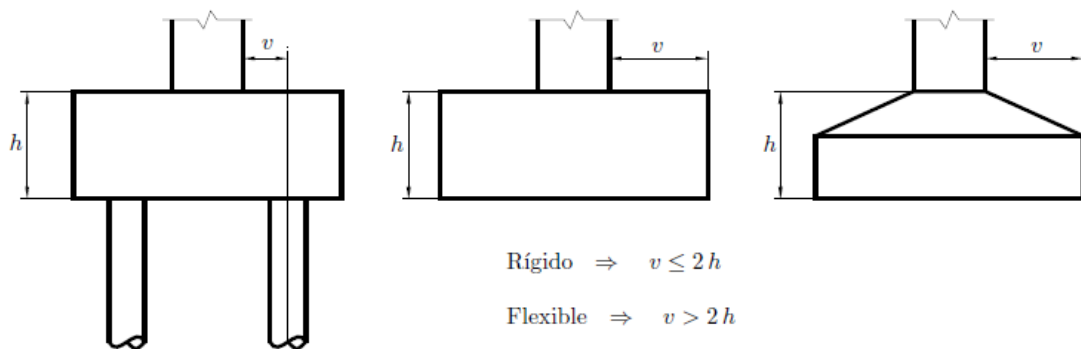


Figura 2.1.3-Tipos de zapatas según el vuelo. Fuente: "Cálculo Cimentaciones"

2.2- ZAPATAS CORRIDAS:

La aplicación desarrollada tiene como objetivo, como ya se ha venido comentando, dimensionar y comprobar una zapata corrida, así que una vez se han hecho las introducciones

pertinentes respecto a los diferentes tipos de cimentaciones y de zapatas, se va tratar más profundamente la zapata corrida.

Las zapatas corridas o continuas son, como ya se ha venido diciendo, cimentaciones superficiales que soportan un muro. La transmisión de cargas desde el muro a la zapata se realiza a través de una distribución sobre una franja estrecha (ancho del muro) y larga (longitud del muro). Como el ancho del muro es pequeño en comparación con la dimensión de la zapata en la misma dirección, puede suponerse que esta última recibe la carga en forma de unas distribuciones lineales de fuerzas y de momentos en la dirección longitudinal del muro que soporta.

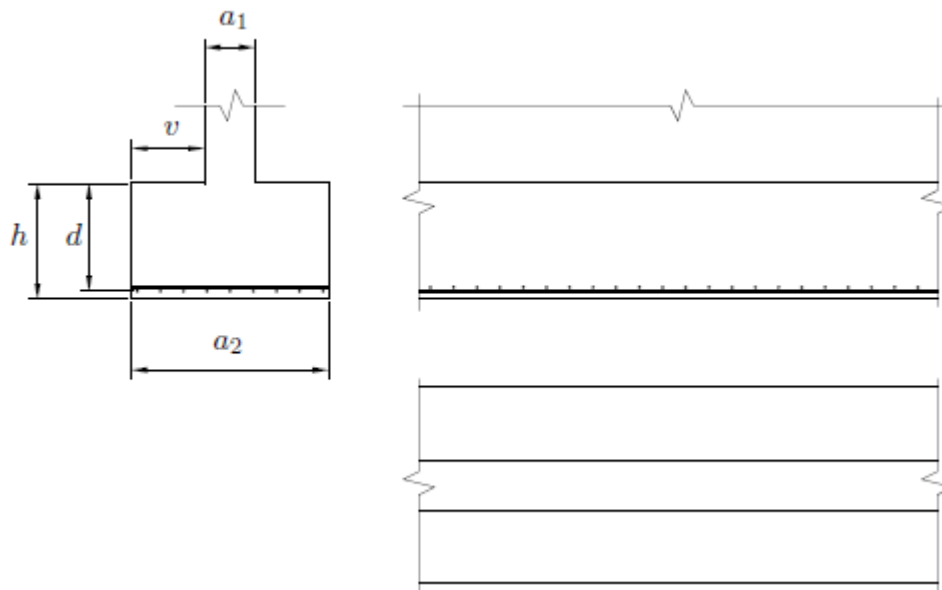


Figura 2.2.1—Dimensiones principales de una zapata corrida. Fuente: “Cálculo de Cimentaciones”

Por lo anterior, las armaduras principales en este tipo de zapatas son perpendiculares al muro y las secundarias son las que llevan la dirección longitudinal del mismo, salvo casos excepcionales en los que haya voladizos en esta última dirección. Así pues, para el análisis de zapatas corridas se considerará un elemento cuya sección transversal es la de la zapata a analizar y con una profundidad (dimensión en la dirección longitudinal del muro) igual a la unidad. Es decir, las variables que se manejen en el cálculo estarán dadas por unidad de longitud de muro.

En este capítulo se va a ver diferentes aspectos de este tipo de cimentaciones, comenzando por los modos de fallo que se pueden producir en este tipo de zapatas y terminando por los diferentes cálculos necesarios para dimensionarla, concretando posteriormente como cómo lo hace la aplicación.

2.2.1-MODOS DE FALLO:

Los principales modos de fallo en zapatas corridas, comunes a cualquier tipo de zapatas, son los siguientes, todos ellos representados en la figura 2.2.1.1:

- **Fallo a flexión con rotura frágil** (figura 2.2.1.1-a): Este tipo de fallo produce una rotura frágil de la pieza (sin fisuración previa de aviso) y puede presentarse cuando se disponen unas cuantías a tracción (U_s) y a compresión (U_c) tales que:

$$U_s < 0,04 \cdot U_c$$

Este caso no es de interés puesto que este tipo de armado no es el que realiza la aplicación, puesto que no dispone en ningún caso de armadura a compresión.

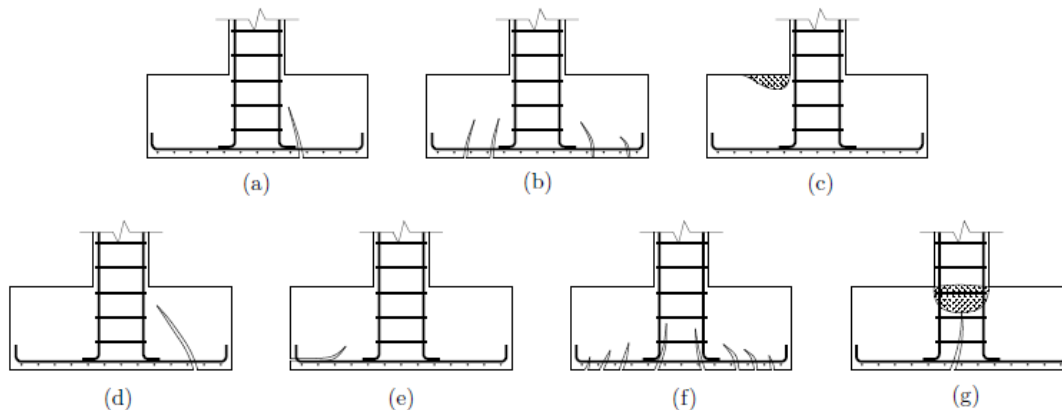


Figura 2.2.1.1—Principales modos de fallo en zapatas. Fuente: "Cálculo de Cimentaciones"

- **Fallo a flexión por agotamiento de la armadura de tracción** (figura 2.2.1.1b): se produce cuando se dispone una armadura de flexión a tracción insuficiente. Se precede por una fisuración considerable (rotura dúctil) en la planta de la zapata, lo que hace que al ser esta una zona no visible de la misma no sirve de señal de aviso observable.
- **Fallo a flexión por agotamiento del hormigón comprimido** (figura 2.2.1.1.c): aparecen ligeras fisuras en la parte comprimida próxima al paramento del muro y paralelas a la dirección de la armadura. Se produce para una cuantía de la armadura de tracción muy grande, por lo que el acero trabaja a tensiones muy bajas (está infrutilizado desde el punto de vista resistente). Este modo de fallo es posible teóricamente según la norma EHE, aunque es muy raro que aparezca puesto que el exceso de cuantía de acero no es habitual dado su carácter antieconómico.
- **Fallo por cortante** (figura 2.2.1.1.d): aquí las fisuras que se producen están localizadas en la base de la zapata e inclinadas aproximadamente 45° respecto de dicha base. Si no se dispone armadura de cortante (que es lo más habitual y como resuelve la aplicación el problema), se produce cuando el canto de la zapata es insuficiente para transmitir las cargas a las que está sometida. En el caso de que se disponga armadura de cortante, este fallo se da cuando la cuantía de la misma también es insuficiente. Más adelante se verá cómo se utiliza este mecanismo de fallo para el cálculo del canto de la zapata, evitando así el uso de la armadura de cortante.
- **Fallo por anclaje de las armaduras** (figura 2.2.1.1.e): se produce cuando la longitud de anclaje de las armaduras dispuestas es menor de la estrictamente necesaria, haciendo que las tensiones de adherencia superan a la que resiste el hormigón. De suceder esto, aparecen fisuras en los extremos de las armaduras y progresan hacia adentro paralelas a las mismas.
- **Fallo por fisuración excesiva** (figura 2.2.1.1.f): cuando la fisuración es excesiva se puede producir corrosión en las armaduras, causando así una disminución de la sección resistente de la misma y, en última instancia, a un fallo de la pieza a flexión por

agotamiento. En el caso de zapatas debe tenerse especial cuidado en el análisis del mencionado estado límite cuando esta trabaja a flexión, dado que frecuentemente están en ambientes húmedos y la fisuración ni es observable ni puede ser reparada. Más adelante en este capítulo veremos cómo calcular la fisuración de acuerdo a la norma EHE.

- **Fallo por aplastamiento del hormigón en la interfase con el muro** (figura 2.2.1.1.g): este se produce por tracciones horizontales excesivas en zapatas muy rígidas debido a una excesiva compresión del muro sobre la zapata. Dado que la resistencia del hormigón que compone la zapata es inferior a la del muro, éste se hinca en la zapata. No es un tipo de fallo habitual en la práctica.

2.2.2-DISTRIBUCIÓN DE PRESIONES SOBRE EL TERRENOS:

Otro aspecto a tener en cuenta para el cálculo del dimensionado tanto de una zapata corrida como de cualquier tipo de cimentación es el cómo se distribuye la presión sobre el terreno.

Esta distribución de tensiones es muy difícil de calcular con exactitud, puesto que dependen de multitud de variables. Entre ellas destacan la rigidez de la zapata y las características de tensión-deformación del suelo.

Es evidente que, de forma general, el terreno no responde de una manera uniforme, ya que donde haya mayor carga se producirá un mayor asiento de la cimentación y la reacción del terreno será mayor. La distribución de presiones que éste ejerce sobre el cimiento, que es la que nos interesa conocer puesto que determinará si se sobrepasa o no la tensión admisible como se verá más adelante en este mismo capítulo, depende de varios factores como la rigidez del cimiento y sobre todo del tipo de suelo en el que estemos.

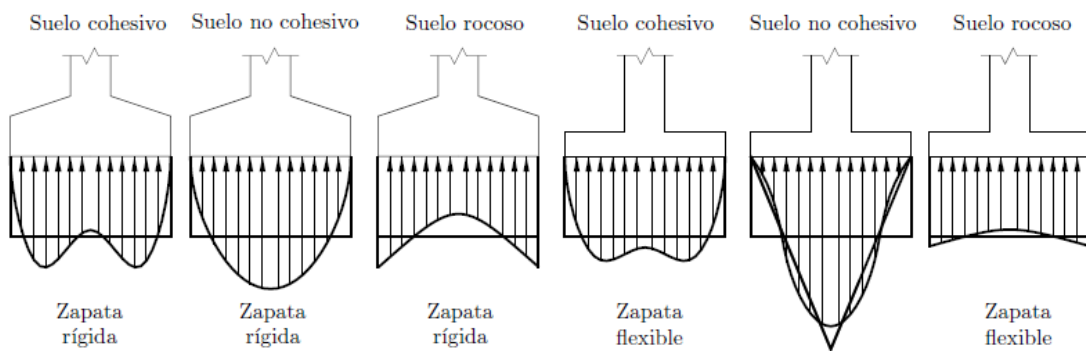


Figura 2.2.2.1—Leyes de presiones en función del tipo de zapata y suelo. Fuente: "Cálculo de Cimentaciones"

Las distintas leyes de presiones pueden verse representadas en función del tipo de zapata y del suelo, a modo orientativo y de manera muy simplificada, en la figura 2.2.2.1 a trazo continuo.

Como vemos estas leyes de presiones se ven afectadas por el tipo de zapata, siendo para el caso flexible mayor la presión en el centro y reduciéndose en los extremos mientras que en el caso de la zapata rígida ocurre lo contrario, esta se reduce en el centro expensas de aumentar en los extremos.

Con respecto al terreno también vemos variación, ya que la zapata puede fluir lateralmente cuando la cohesión del terreno es despreciable, con lo que la presión en los extremos reduce y

aumenta en el centro, y según esta aumenta se va invirtiendo este comportamiento, reduciendo la presión en la zona central y aumentando en los extremos.

En la práctica las distribuciones de presiones reales se asimilan a distribuciones uniformes o lineales, adoptando el caso más desfavorable en cada caso, y para que esto sea posible debe cumplirse la condición de **rigidez relativa suelo-cimiento**, que es suficiente para la mayoría de los casos. Esto se realiza comprobando el vuelo máximo de la zapata de la siguiente forma:

$$v \leq 0,88 \left(\frac{4 E I}{K_c b} \right)^{\frac{1}{4}} \quad (2.1)$$

- Con: E = módulo de elasticidad medio del hormigón de la zapata (E_c)
 I = momento de inercia de la zapata respecto de un eje paralelo a la dirección longitudinal del muro
 K_c = coeficiente o módulo de balasto del terreno
 b = profundidad de la zapata (b_2 , habitualmente la unidad)

Respecto al coeficiente de balasto, suele calcular con valores hallados experimentalmente para unas dimensiones determinadas, siendo el más habitual el K_{300} , referente a una placa cuadrada de 300 mm de lado, que es el que utiliza la aplicación. Algunos de estos valores podemos verlo en la Tabla 2.2.2.1.

Tipo de terreno	$K_{750}^{(1)}$ (N/mm^3)	$K_{300}^{(2)}$ (N/mm^3)
Arcillas blandas	$K_{750} \leq 0,018$	$K_{300} \leq 0,04$
Arcillas compactas	$0,018 \leq K_{750} \leq 0,04$	$0,04 \leq K_{300} \leq 0,09$
Arcillas poco densas	$0,01 \leq K_{750} \leq 0,02$	$0,02 \leq K_{300} \leq 0,05$
Arenas de compacidad media	$0,02 \leq K_{750} \leq 0,04$	$0,05 \leq K_{300} \leq 0,09$
Arenas densas	$0,04 \leq K_{750} \leq 0,08$	$0,09 \leq K_{300} \leq 0,18$
Rocas y gravas compactas	$0,08 \leq K_{750}$	$0,18 \leq K_{300}$

¹ Obtenidos con placa de carga circular de 750 mm de diámetro
² Obtenidos con placa de carga cuadrada de 300 mm de lado

Tabla 2.2.2.1—Valores experimentales del coeficiente de balasto de placas cuadradas de 750 mm y 300 mm de ancho. Fuente: "Cálculo de Cimentaciones"

Con respecto a esta tabla 2.2.2.1, comentar que los coeficientes están expresados en N/mm^3 , mientras que en la aplicación deben ser introducidos en MN/m^3 , con lo que habría que multiplicarlos por mil antes de introducirlos.

Para terminar con este breve inciso sobre la comprobación de rigidez relativa necesaria para admitir que la distribución es lineal, el cálculo del coeficiente K_c a partir de K_{300} , es el siguiente:

$$K_c = \frac{2}{3} \left[\frac{0,3}{a_2} K_{300} \left(1 + \frac{a_2}{2l} \right) \right] \quad (2.2)$$

- Con: a_2 = ancho de la zapata
 l = longitud total de la zapata

Una vez realizada se cumple la comprobación de rigidez relativa suelo-cimiento, se puede calcular la reacción del terreno a partir de la distribución de presiones lineal. Para esto se considera la aplicación de una fuerza vertical de compresión F y un momento M_2 , que serán la resultante de todas las cargas que actúan sobre la zapata y se aplican sobre el centro de gravedad de la misma. Generalmente no se consideran la resultante horizontal porque se ve contrarrestada por los empujes laterales, lo que se confirmará mediante la **comprobación de deslizamiento**. La aplicación de estas fuerzas y su distribución de presiones se muestra en la figura 2.2.2.2.

Calculando por equilibrio de fuerzas y momentos la reacción del terreno obtenemos la

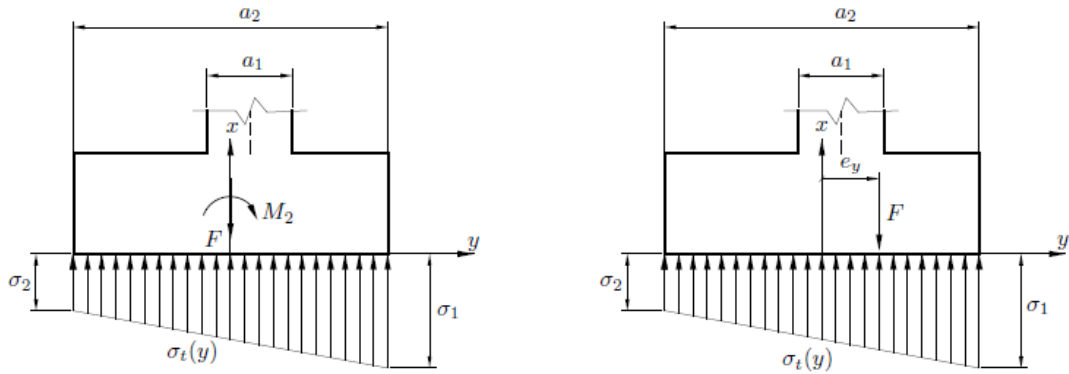


Figura 2.2.2.2—Resultantes F y M_2 de las cargas aplicadas sobre la zapata y reacción lineal del terreno cuando se cumple la condición de rigidez relativa. Fuente: "Cimentaciones"

siguiente expresión para la presión por unidad de longitud de muro:

$$\sigma_t(y) = \frac{F}{a_2} + \frac{M_2}{\frac{1}{12}a_2^3} y = \frac{F}{a_2} \left(1 + \frac{12 e_y}{a_2^2} y \right) \quad (2.3)$$

donde la resultante vertical N se aplica con excentricidad e_y con respecto del centro de gravedad de la base de la zapata, por lo que $M_2 = F e_y$.

La ecuación (2.3) puede arrojar valores negativos, lo que no puede suceder en la realidad porque el terreno no trabaja a tracción, con lo que sólo será válida si la resultante N se aplica sobre el núcleo central de la base de la zapata, lo cual matemáticamente se cumple si:

$$|e_y| \leq \frac{a_2}{6} \quad (2.4)$$

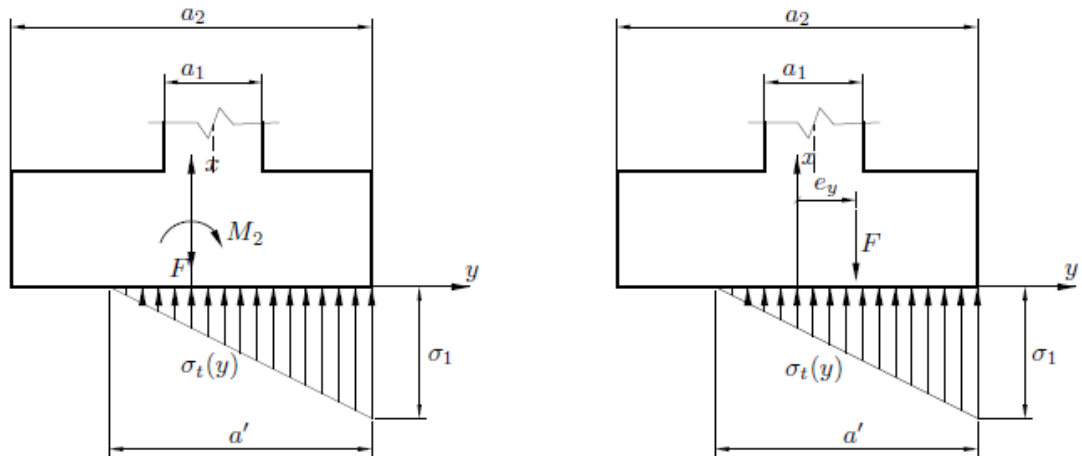


Figura 2.2.2.3—Resultantes F y M_2 de las cargas aplicadas sobre la zapata y reacción triangular del terreno cuando se cumple la condición de rigidez relativa y N se aplica fuera del núcleo central de la zapata. Fuente: "Cálculo de Cimentaciones"

De no satisfacerse la condición (2.4) se pasa a considerar una distribución triangular como la mostrada en la figura 2.2.2.3, a partir de la cual, planteando de nuevo equilibrio de momentos y fuerzas, se obtienen:

$$\sigma_t(y) = \begin{cases} \sigma_1 \left(1 + \frac{1}{a'} \left(y - \frac{a_2}{2} \right) \right) = \frac{8}{9} \frac{a_2 - 3e_y + y}{(a_2 - 2e_y)^2} F & \text{si } \frac{-a_2}{2} - a' \leq y \leq \frac{a_2}{2} \\ 0 & \text{si } \frac{-a_2}{2} \leq y \leq \frac{-a_2}{2} - a' \end{cases} \quad (2.5)$$

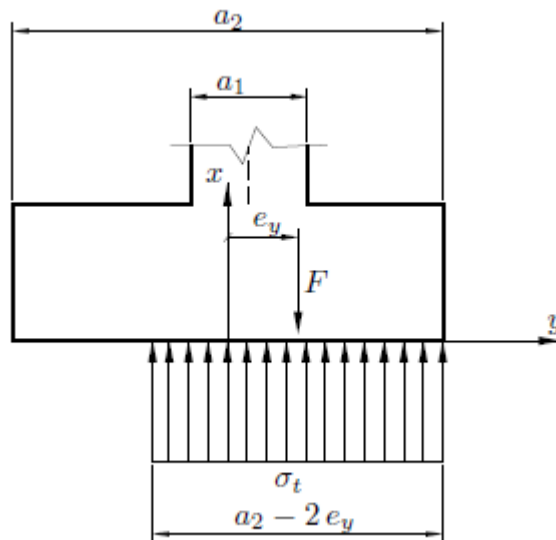


Figura 2.2.2.4—Distribución rectangular y constante de tensiones aplicada sobre el ancho equivalente de la zapata. Fuente: "Cálculo de Cimentaciones"

La tendencia que siguen los nuevos métodos de comprobación de tensiones en el terreno bajo los cimientos (por ejemplo, el Eurocódigo EC-7) es sustituir la distribución triangular de la Figura 2.2.2.3 por uno rectangular donde la tensión es constante y se aplica en el ancho equivalente, tal como muestra la figura 2.2.2.4. Tal distribución constante es:

$$\sigma_t = \frac{F}{a_2 - 2 e_y} \quad (2.6)$$

pero para poder aplicar esto debe cumplirse:

$$|e_y| \leq \frac{a_2}{3} \quad (2.7)$$

En el caso de no poder asumir una distribución de tensiones como la mostrada en la figura 2.2.2.2, se produce que el extremo donde se aplica tensión está más comprimido que el opuesto, donde no se aplica tensión alguna, levantándose éste. Este efecto se evita aumentando la dimensión a_2 si es posible o desplazar la posición relativa del muro para que la excentricidad de la fuerza resultante sea nula. En el caso de la aplicación se aplica el primer criterio, dimensionando con este el ancho de la zapata, como veremos a continuación.

En cualquier caso, dado que al aumentar la excentricidad e_y aumenta en gran medida la tensión máxima σ_1 , siempre debemos satisfacer la condición dada por la ecuación (2.7).

2.2.3-CÁLCULO DE LAS DIMENSIONES DE LA ZAPATA:

Las dimensiones que debemos determinar para el cálculo de la zapata son el ancho a_2 y el canto h , ya que el cálculo se realizará por unidad de longitud de la zapata y la dimensión del muro será un dato de entrada.

Haciendo un resumen de lo comentado previamente, el dimensionado de la zapata debe satisfacer:

- **La distribución de presiones bajo la zapata debe ser lo más uniforme posible:** Como ya se ha comentado debemos buscar que la distribución sea uniforme, pudiendo admitir también una distribución lineal si la excentricidad e_y es pequeña. La distribución preferible la mostrada en la figura 2.2.2.2.
- **Las presiones máximas sobre el terreno deben ser admisibles:** el valor máximo de la distribución debe ser inferior a la que el terreno admite para evitar el hundimiento de la zapata.
- **Los asientos totales máximos deben ser admisibles:** esto significa que deben requerirse limitaciones sobre los valores máximos del asiento de la zapata, lo que normalmente se traduce en una limitación sobre las presiones máximas. Esto es así para evitar que la zapata gire, con lo que de nuevo lo que se busca es una distribución lo más uniforme posible.
- **Los asientos diferenciales máximos deben ser admisibles:** es conveniente que todas las zapatas de la estructura asienten una misma cantidad, evitando que las distorsiones angulares entre muros y/o pilares contiguos sean mínimas. En general esto se consigue haciendo que la presión máxima bajo todas las zapatas sea aproximadamente la misma.
- **No se debe producir el vuelco de las zapatas:** se debe cumplir la condición de comprobación de vuelco en la zapata, que se comentará más adelante.

- **No debe producirse el deslizamiento de la zapata:** se debe cumplir la condición de deslizamiento de vuelco en la zapata, que se comentará más adelante.
- **No debe ser necesario disponer armaduras ni de cortante ni de punzonamiento:** esto se hace por motivos económicos. Para ello se debe cumplir la condición sobre el cortante que también veremos a continuación, que además utilizaremos para dimensionar el canto de la zapata, puesto que la forma de evitar la armadura de cortante es aumentar dicho canto. Respecto al punzonamiento no es una comprobación de interés para la zapata continua, ya que sólo se produce de existir huecos muy próximos en el muro, y no es el caso a tratar por la aplicación.

El proceso de cálculo de las dimensiones será pues un proceso iterativo, ya que a la hora de aplicar por primera vez alguna de las condiciones para hallar alguna dimensión pueden ser desconocidas otras dimensiones, o alguna magnitud que dependa de ésta, con lo que es preciso hacer inicialmente alguna estimación de las mismas.

2.2.3.1-CÁLCULO DEL ANCHO DE LA ZAPATA:

El ancho a_2 de la zapata se deduce principalmente de las seis primeras consideraciones descritas anteriormente. Para ello se utilizan los esfuerzos resultantes F y M_2 en la base de la capta, deducidos a partir de las siguientes cargas, **todas ellas con sus valores característicos** (sin mayorar):

- Combinación más desfavorable de cargas transmitidas por muro que descansa sobre la zapata.
- Peso propio de la zapata.
- Peso del terreno que gravita directamente sobre la zapata.
- Empuje de tierras sobre la zapata.
- Empuje hidrostático sobre la zapata en terrenos anegados.
- Cualquier otra carga, ya sea gravitatoria o de cualquier otra índole, que esté aplicada directamente sobre la zapata.

Una vez calculadas F y M_2 según estas cargas y con las distribuciones de presiones descritas en el apartado anterior, se establecen la condición de seguridad y el ancho de la zapata que resulta. Estas son, para cada uno de los casos descritos:

- **Distribución constante** (figura 2.2.2.2 con $M_2=0$, $e_y=0$ y $\sigma_1 = \sigma_2$):

$$\left. \begin{array}{l} \sigma_t = \frac{F}{a_2} \\ (\sigma_t(y))_{max} \leq \sigma_{t,adm} \end{array} \right\} \Rightarrow \frac{F}{a_2} \leq \sigma_{t,adm} \Rightarrow a_2 \geq \frac{F}{\sigma_{t,adm}} \quad (2.8)$$

con $\sigma_{t,adm}$ la presión máxima admisible por el terreno.

- **Distribución trapezoidal** (figura 2.2.2.2):

$$\left. \begin{array}{l} \sigma_t(y) = \frac{F}{a_2} \left(1 + \frac{12 e_y}{a_2^2} y \right) \\ (\sigma_t(y))_{max} = \sigma_t(a_2/2) \leq \frac{5}{4} \sigma_{t,adm} \end{array} \right\} \Rightarrow \frac{F}{a_2} \left(1 + \frac{6 e_y}{a_2} \right) \leq \frac{5}{4} \sigma_{t,adm} \Rightarrow a_2 \geq \frac{2}{5} \frac{F}{\sigma_{t,adm}} \left(1 \pm \sqrt{1 + 30 e_y \frac{\sigma_{t,adm}}{F}} \right) \quad (2.9)$$

En este caso se permite sobrepasar la tensión máxima admisible en un 25%. No obstante, debe verificarse también que la presión en el punto medio de la distribución no sobrepase la máxima admisible por el terreno, para que el asiento total máximo sea admisible. Esto es:

$$\sigma_t(0) \leq \sigma_{t,adm} \Rightarrow \frac{F}{a_2} \leq \sigma_{t,adm} \Rightarrow a_2 \geq \frac{F}{\sigma_{t,adm}} \quad (2.10)$$

con lo que de las dos soluciones aportadas por la ecuación (2.9) la válida será:

$$a_2 \geq \frac{5}{4} \left(1 + \sqrt{1 + 30 e_y \frac{\sigma_{t,adm}}{F}} \right) \quad (2.11)$$

Recordando que esto debe cumplir también la condición dada por (2.4), que limita el ancho tal que:

$$a_2 \geq 6 |e_y| \quad (2.12)$$

➤ **Distribución triangular** (figura 2.2.2.3):

$$\left. \begin{aligned} \sigma_t(y) &= \frac{8}{9} \frac{a_2 - 3e_y + y}{(a_2 - 2e_y)^2} F \\ (\sigma_t(y))_{max} = \sigma_1 &\leq \frac{5}{4} \sigma_{t,adm} \end{aligned} \right\} \Rightarrow \frac{4}{3} \frac{F}{a_2 - 2e_y} \leq \frac{5}{4} \sigma_{t,adm} \Rightarrow a_2 \geq \frac{16}{15} \frac{F}{\sigma_{t,adm}} + 2e_y \quad (2.13)$$

Vemos que de nuevo se permite sobrepasar la tensión máxima admisible un 25%, pero debe verificar que no la sobrepase en el centro, con lo que:

$$(\sigma_t(y))_{media} = \frac{F}{a'} \leq \sigma_{t,adm} \Rightarrow \frac{F}{\frac{3}{2}(a_2 - 2e_y)} \leq \sigma_{t,adm} \Rightarrow a_2 \geq \frac{2}{3} \frac{F}{\sigma_{t,adm}} + 2e_y \quad (2.14)$$

Además, para que aparezca este tipo de distribución la resultante F debe ser aplicada fuera del núcleo central de la zapata, por lo que debe verificarse lo siguiente:

$$3|e_y| \leq a_2 \leq 6|e_y| \quad (2.15)$$

➤ **Distribución rectangular** (figura 2.2.24):

$$\left. \begin{aligned} \sigma_t &= \frac{F}{a_2 - 2e_y} \\ \sigma_t &\leq \sigma_{t,adm} \end{aligned} \right\} \Rightarrow \frac{F}{a_2 - 2e_y} \leq \sigma_{t,adm} \Rightarrow a_2 \geq \frac{F}{\sigma_{t,adm}} + 2e_y \quad (2.16)$$

Debe también verificar:

$$a_2 \geq 3|e_y| \quad (2.17)$$

Todas las expresiones mostradas en este apartado tienen como incógnita, a parte evidentemente del ancho a_2 de la zapata, el canto h del que dependen algunos de sus términos,

como el peso propio de la zapata que debe ser incluido en la resultante de las fuerzas, por lo que se supondrá un valor h en el proceso iterativo de cálculo con el que se calculará el ancho, y con estos valores se realizarán todas las comprobaciones necesarias.

2.2.3.2-CÁLCULO DEL CANTO TOTAL:

Con el ancho a_2 calculado, ya sólo nos falta el canto h . Para este se aplicará la condición de que no sea necesaria armadura de cortantes. Para ello debemos conocer cuál es el valor del esfuerzo cortante en la sección crítica. Para su cálculo se supondrá que la zapata es un elemento apoyado en el muro y sometido a la distribución de tensiones σ_{td} que equilibra las resultantes F_d y M_{d2} del siguiente sistema de cargas, en este caso con su valor de cálculo, es decir, mayoradas con su correspondiente coeficiente de seguridad:

- Combinación más desfavorable de cargas transmitidas por muro que descansa sobre la zapata.
- Empuje de tierras sobre la zapata.
- Empuje hidrostático sobre la zapata en terrenos anegados.
- Cualquier otra carga, ya sea gravitatoria o de cualquier otra índole, que esté aplicada directamente sobre la zapata y cuyo efecto no sea el simple aplastamiento de la misma. Como norma general, todas aquéllas cargas que no estén uniformemente distribuidas sobre toda la parte superior de la zapata.

Con esto calculamos F_d y M_{d2} y conocemos también a_2 puesto que se ha calculado previamente, por lo que nos falta determinar la distribución de tensiones sobre el terreno que, admitiendo que esta es triangular o trapezoidal, se realiza como se indicó en el apartado anterior, con lo que queda:

$$\sigma_{td}(y) = \begin{cases} \frac{F_d}{a_2} \left(1 + \frac{12 e_{yd}}{a_2^2} y \right) & \text{si } |e_y| \leq \frac{a_2}{6} \\ \frac{8}{9} \frac{a_2 - 3 e_{yd} + y}{(a_2 - 2 e_{yd})^2} F_d & \text{si } \frac{a_2}{3} \geq |e_y| \geq \frac{a_2}{6} \end{cases} \quad (2.18)$$

siendo $e_{yd} = M_{d2}/F_d$.

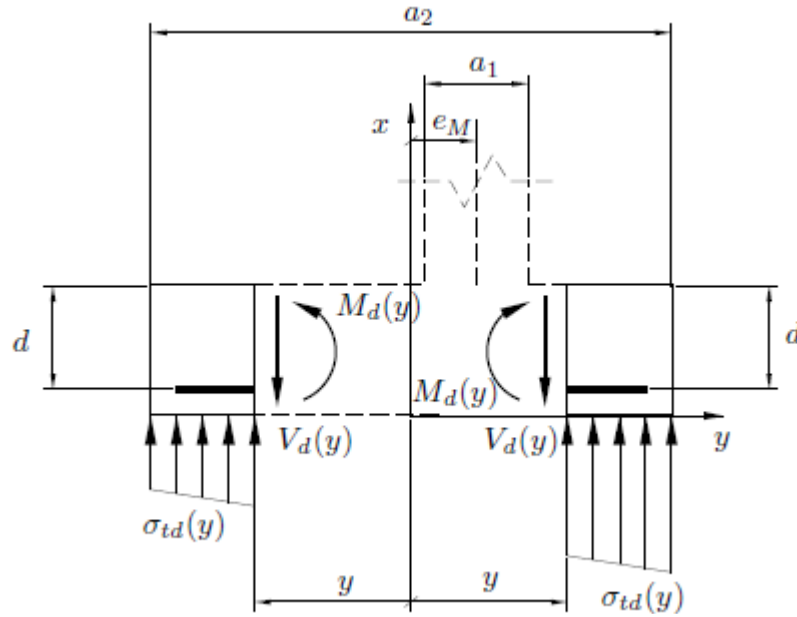


Figura 2.2.3.2.1—Esfuerzo cortante y momento flector que actúa sobre cada sección de la zapata. Fuente: "Cálculo de Cimentaciones"

El esfuerzo cortante que actúa sobre cada sección de la zapata (figura 2.2.3.1) es:

$$V_d(y) = \begin{cases} \int_y^{a_2/2} \sigma_{td}(y') dy' & \text{si } e_M \leq y \leq \frac{a_2}{2} \\ \int_y^{-a_2/2} \sigma_{td}(y') dy' & \text{si } -\frac{a_2}{2} \leq y \leq e_M \end{cases} \quad (2.19)$$

siendo e_M la excentricidad del muro respecto del centro de gravedad de la zapata.

Como se busca dimensionar la zapata para que no necesite ni armadura de cortante ni armadura de compresión, se debe satisfacer, de acuerdo a la norma EHE, la siguiente expresión:

$$V_d \leq V_{cu} = \frac{0,18}{\gamma_c} \xi (100 \rho_1 f_{ck})^{1/3} b_0 d \quad (2.20)$$

- Con:
- V_d = cortante que actúa sobre la sección.
 - V_{cu} = cortante resistido por la sección.
 - f_{ck} = resistencia característica del hormigón, en MPa.
 - d = canto útil de la sección, en mm.
 - ρ_1 = cuantía geométrica de la armadura de tracción.
 - b_0 = ancho de la sección, en mm (b_2).
 - γ_c = coeficiente de seguridad de la resistencia de proyecto del hormigón.
- $$\xi = \left(1 + \sqrt{\frac{200}{d}} \right)$$

Este cortante tiene además un valor mínimo, que se corresponde, de nuevo de acuerdo a la norma EHE, con la siguiente expresión:

$$V_{cu} = \frac{0,075}{\gamma_c} \xi^{3/2} f_{ck}^{1/2} b_0 d \quad (2.21)$$

Respecto a la ecuación (2.20), la cuantía mecánica ρ_1 tendrá un valor no mayor que 0,02 para aceros B-400 y no mayor que 0,025 para aceros B-500.

La sección de referencia que se utilizará para calcular V_d es la dada por la norma EHE, que denomina como sección S_2 , y que es una *sección recta, paralela al paramento del muro y tal que dista del mismo una distancia d (canto útil)*. Esta sección está presente a los dos lados de la zapata, por lo que se deberá tomarse aquella en la que el esfuerzo cortante sea mayor.

Con estas ecuaciones obtenemos el canto mínimo necesario de la zapata sin armadura de cortante ni compresión, pero este método presenta el inconveniente de que la expresión (2.20) depende de la cuantía ρ_1 . Está no puede ser calculada puesto que para ello necesitamos el canto, así que se aproxima por lo siguiente:

$$\rho_1 = \frac{A_s}{d} = \frac{\frac{M_d}{0,9 f_{yd} d}}{d} = \frac{M_d}{0,9 f_{yd} d^2} \quad (2.22)$$

siendo M_d el momento flector de cálculo para las armaduras de tracción, que será el que se produzca en la sección de referencia S_1 , que también define la norma EHE, y que se define como una sección plana, paralela a la cara del soporte o del muro que está situada detrás de dicha cara a una distancia igual a $0,15 a_1$, siendo a_1 como ya se ha visto el ancho del muro. Igual que en el caso anterior, de las dos secciones posibles se escogerá aquella en la que se produzca un momento mayor.

Con respecto al valor de $0,15 a_1$ mencionado en el párrafo anterior, éste se corresponde a un muro de hormigón, y puede ser sustituido por lo siguiente:

- $0,25 a_1$, cuando se trate de muros de ladrillo o mampostería.
- La mitad de la distancia entre la cara del soporte y el borde de la placa de acero, cuando se trate de soportes metálicos sobre placas de reparto de acero.

Con esto se hace un primer cálculo, que deberá repetirse cuando se calcule la armadura, para comprobar que se cumple estas condiciones.

2.2.4-OTRAS COMPROBACIONES:

A lo largo de este capítulo se han desarrollado algunas de las comprobaciones necesarias que debe cumplir la zapata para ajustarse a la norma, como son la comprobación de la **rigidez relativa suelo-cimiento**, dada por la expresión (2.1) y que nos permite asimilar la distribución de tensiones real a una uniforme o lineal, la comprobación de **hundimiento**, desarrollada en el punto 2.2.3.1 y a través de la cual dimensionamos el ancho de la zapata, y la comprobación de **cortante**, desarrollada en el punto 2.2.3.2 y que nos da el canto mínimo para el que no se necesita armadura ni de cortante ni de compresión.

A parte de estas, para que la zapata calculada cumpla con la norma deben realizarse más comprobaciones, en concreto las de **deslizamiento, vuelco y fisuración**.

Antes de pasar a describir estas comprobaciones, comentar que para todas ellas se utilizan lo que se define en el CTE como coeficientes de seguridad parciales, mostrados en la tabla 2.2.4.1. Con esto, los coeficientes resultado de las comprobaciones tendrán que ser mayor o igual a la unidad para ser válidos.

Situación de dimensionado	Tipo	Materiales		Acciones	
		γ_R	γ_M	γ_E	γ_F
Persistente o transitoria	Hundimiento	3,0 ⁽¹⁾	1,0	1,0	1,0
	Deslizamiento	1,5 ⁽²⁾	1,0	1,0	1,0
	Vuelco ⁽²⁾				
	Acciones estabilizadoras	1,0	1,0	0,9 ⁽³⁾	1,0
	Acciones desestabilizadoras	1,0	1,0	1,8	1,0
	Estabilidad global	1,0	1,8	1,0	1,0
	Capacidad estructural	- ⁽⁴⁾	- ⁽⁴⁾	1,6 ⁽⁵⁾	1,0
	Pilotes				
	Arrancamiento	3,5	1,0	1,0	1,0
	Rotura horizontal	3,5	1,0	1,0	1,0
	Pantallas				
	Estabilidad fondo excavación	1,0	2,5 ⁽⁶⁾	1,0	1,0
	Sifonamiento	1,0	2,0	1,0	1,0
	Rotación o traslación				
	Equilibrio límite	1	1,0	0,6 ⁽⁷⁾	1,0
	Modelo de Winkler	1	1,0	0,6 ⁽⁷⁾	1,0
Elementos finitos	1,0	1,5	1,0	1,0	
Extraordinaria	Hundimiento	2,0 ⁽⁸⁾	1,0	1,0	1,0
	Deslizamiento	1,1 ⁽²⁾	1,0	1,0	1,0
	Vuelco ⁽²⁾				
	Acciones estabilizadoras	1,0	1,0	0,9	1,0
	Acciones desestabilizadoras	1,0	1,0	1,2	1,0
	Estabilidad global	1,0	1,2	1,0	1,0
	Capacidad estructural	- ⁽⁴⁾	- ⁽⁴⁾	1,0	1,0
	Pilotes				
	Arrancamiento	2,3	1,0	1,0	1,0
	Rotura horizontal	2,3	1,0	1,0	1,0
	Pantallas				
	Rotación o traslación				
	Equilibrio límite	-	-	-	-
	Modelo de Winkler	1,0	1,0	0,8	1,0
	Elementos finitos	1,0	1,2	1,0	1,0

Tabla 2.2.4.1— Coeficientes de seguridad parciales.

Con esto dicho, las comprobaciones que faltan por explicar:

- Vuelco:** el fallo por vuelco se produce cuando las cargas cuya línea de acción es paralela a la planta del cimienta son importantes y actúan a una distancia apreciable de la o cuando el momento flector es importante. Este fallo puede producir el giro de la pieza alrededor de una de las aristas de su planta. Al suceder esto, hay una reorganización de presiones sobre el terreno, aumentando en la zona que se hunde y disminuyendo o anulándose en la zona que se levanta.
 Para evitar esto, la condición que debe satisfacerse es que los momentos estabilizantes sean mayores o iguales a los momentos desestabilizantes, cada uno con su respectivo coeficiente parcial de seguridad (tabla 2.2.4.1).

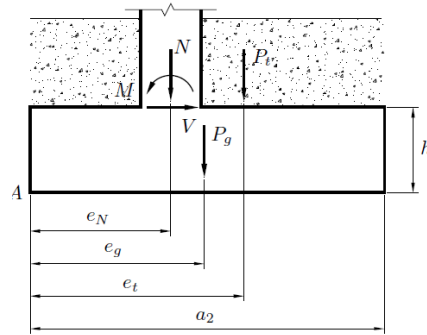


Figura 2.2.4.1—Cargas aplicadas sobre una zapata. Fuente: “Cálculo de Cimentaciones”

Con el ejemplo de la figura 2.2.4.1, vemos todas las cargas que actúan sobre una zapata en el sentido en que realmente actúan, con N , M y V son los esfuerzos nominales de cálculo en la cara superior de la cimentación en la hipótesis de carga más desfavorable, P_g es el peso propio de la zapata, P_t es la suma del peso del terreno que gravita directamente sobre la cimentación más cualquier otro tipo de cargas aplicadas directamente sobre el elemento (no transmitidas a través del pilar). P_g y P_t siempre tendrán un efecto estabilizador, mientras que el momento M y el cortante V podrán ser acciones estabilizadoras o desestabilizadoras. Así si $M > V h$, la zapata tenderá a girar respecto del extremo A, con lo que la condición de seguridad será:

$$N e_N + P_g e_g + P_t e_t + V h \geq M \quad (2.23)$$

mientras que si $M < V h$ sucederá lo contrario, teniendo así:

$$N(a_2 - e_N) + P_g(a_2 - e_g) + P_t(a_2 - e_t) + M \geq V h \quad (2.24)$$

Recordar que, como ya se ha dicho, se deben aplicar los coeficientes parciales de seguridad indicados en la tabla 2.2.4.1, tanto para los momentos estabilizadores como para los desestabilizadores.

- Deslizamiento:** este fallo se produce cuando las cargas cuya línea de acción es paralela a su base superan a las fuerzas de rozamiento existente en la interface cimiento–suelo. Esta comprobación es especialmente crítica cuando los esfuerzos cortantes existentes en la base del pilar o muro que soporta son considerables y la zapata no está arriostrada lateralmente. Como fuerza estabilizante se considerará siempre la de rozamiento entre la base de la zapata y el terreno o la cohesión con éste, no teniendo en cuenta los empujes pasivos del terreno sobre sus paramentos verticales a menos que esté garantizada su actuación permanente y el plano superior de la zapata esté situado, al menos, un metro por debajo de la superficie libre. En este caso, volviendo al ejemplo de la figura 2.2.4.1 y para una zapata de dimensiones $a_2 \times b_2$, nos quedan las expresiones:

$$(N + P_g + P_t) \tan \delta \geq V \text{ en suelos sin cohesión} \quad (2.25)$$

$$a_2 b_2 c_d \geq V \text{ en suelos cohesivos} \quad (2.25)$$

siendo δ el ángulo de rozamiento entre el terreno y la zapata (suele adoptarse $\delta = 2 \varphi / 3$, donde φ es el ángulo de rozamiento interno del terreno), c_d el valor de cálculo minorado de

la cohesión, que se suele hacer igual a la mitad del valor teórico de la misma, y con el coeficiente de seguridad parcial que corresponda de la tabla 2.2.4.1.

- **Fisuración:** este fallo solo será considerado si la zapata en cuestión trabaja a flexión. La fisuración consiste en la aparición de microfisuras de unas pocas décimas de milímetro en elementos de hormigón armado, pudiendo esto traducirse en corrosión sobre la armadura y por tanto en una pérdida de su capacidad resistente.

Esta comprobación se realiza según lo dispuesto en la norma EHE para el caso de fisuración por tracción. Según esta, la comprobación general del Estado Límite de Fisuración por tracción consiste en satisfacer la siguiente expresión:

$$w_k \leq w_{max} \quad (2.26)$$

donde:

w_k Abertura característica de fisura.

w_{max} Abertura máxima de fisura definida en la tabla 2.2.4.2.

El cálculo de la abertura característica de fisura viene dado por:

$$w_k = \beta s_m \varepsilon_m \quad (2.27)$$

donde:

β Coeficiente que relaciona la abertura media de fisura con el valor característico y vale 1,3 para fisuración producida por acciones indirectas solamente y 1,7 para el resto de los casos.

s_m Separación media de fisuras, expresada en *mm*.

$$s_m = 2c + 0,2s + 0,4k_1 \frac{\emptyset A_{c,eficaz}}{A_s} \quad (2.28)$$

Clase de exposición, según artículo 8º	$w_{m\acute{a}x}$ [mm]	
	Hormigón armado (para la combinación cuasipermanente de acciones)	Hormigón pretensado (para la combinación frecuente de acciones)
I	0,4	0,2
IIa, IIb, H	0,3	0,2 ⁽¹⁾
IIIa, IIIb, IV, F, Qa ⁽²⁾	0,2	Descompresión
IIIc, Qb ⁽²⁾ , Qc ⁽²⁾	0,1	

Tabla 2.2.4.2—Aberturas características de fisura.

ε_m Alargamiento medio de las armaduras, teniendo en cuenta la colaboración del hormigón entre fisuras.

$$\varepsilon_m = \frac{\sigma_s}{E_s} \left[1 - k_2 \left(\frac{\sigma_{sr}}{\sigma_s} \right)^2 \right] \geq 0,4 \frac{\sigma_s}{E_s} \quad (2.29)$$

c Recubrimiento de las armaduras traccionadas.

s Distancia entre barras longitudinales. Si $s > 15 \emptyset$ se tomará $s = 15 \emptyset$.

k_1 Coeficiente que representa la influencia del diagrama de tracciones en la sección, de valor

$$k_1 = \frac{\varepsilon_1 + \varepsilon_2}{8 \varepsilon_1} \quad (2.29)$$

donde ε_1 y ε_2 son las deformaciones máxima y mínima calculadas en sección fisurada, en los límites de la zona traccionada. En nuestro caso $\varepsilon_2 = 0$.

\emptyset Diámetro de la barra traccionada más gruesa o diámetro equivalente en el caso de grupo de barras.

$A_{c,eficaz}$ Área de hormigón de la zona de recubrimiento, definida en la figura 2.2.4.2, en donde las barras a tracción influyen de forma efectiva en la abertura de las fisuras.

A_s Sección total de las armaduras situadas en el área $A_{c,eficaz}$.

σ_s Tensión de servicio de la armadura pasiva en la hipótesis de sección fisurada.

E_s Módulo de deformación longitudinal del acero.

k_2 Coeficiente de valor 1,0 para los casos de carga instantánea no repetida y 0,5 para los restantes.

σ_{ST} Tensión de la armadura en la sección fisurada en el instante en que se fisura el hormigón.

2.3-CÁLCULO DE ARMADURAS:

Para terminar con este apartado falta por explicar el cálculo de la armadura a disponer en la zapata. Como ya se ha venido comentando en la zapata no se dispondrá ni de armadura a cortante ni de armadura de compresión, por lo que las armaduras a calcular serán la armadura de tracción, que será la principal, la de reparto y la armadura en espera.

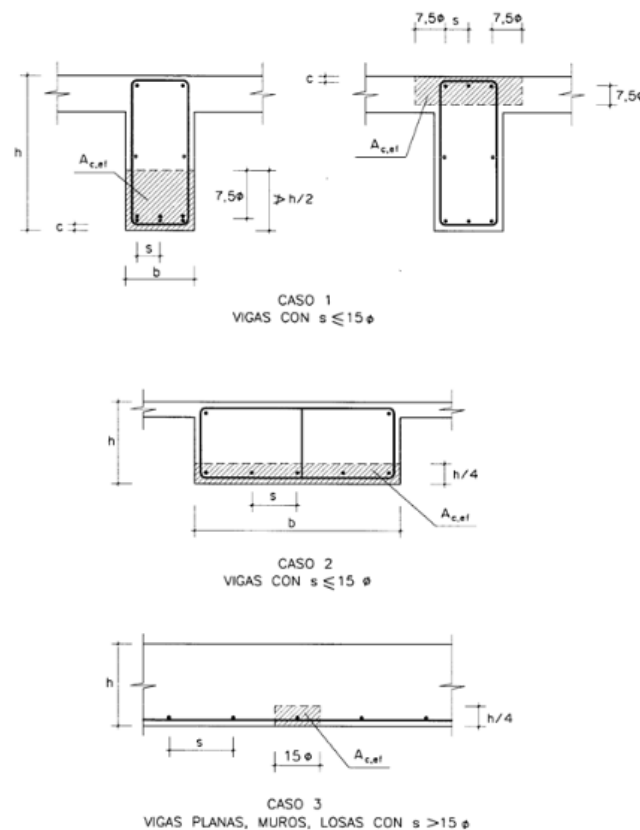


Figura 2.2.4.2— $A_{c,eficaz}$ de acuerdo a la norma EHE

2.3.1-CÁLCULO DE LA ARMADURA PRINCIPAL:

Como ya se ha visto, la zapata puede ser considerada, en función de sus dimensiones, flexible o rígida, y en función de esto se la armadura a disponer se calcula de una u otra forma: mediante

el método de bielas y tirantes en el caso rígido y aplicando la teoría general de flexión en el caso flexible.

2.3.1.1-ARMADURAS RÍGIDAS:

En este caso se utilizará el método de bielas y tirantes, en concreto el **método discretizado de bielas y tirantes** que se define en la norma EHE.

Según este método, las cargas N_d y M_d ($e_d = M_d/N_d$) actuantes en la base del muro se sustituyen por dos cargas equivalentes N_{d1} y N_{d2} aplicadas a una distancia $a_1/4$ del paramento del muro por su parte interior y a una profundidad $0,15 d$ por el interior de la zapata medida desde la base del muro. Por otro lado, la respuesta del terreno (distribución σ_{td}) se sustituye por otras dos fuerzas R_{d1} y R_{d2} tales que la primera es estáticamente equivalente al trapecio de presiones que actúa entre $-a_2/2$ y 0 , mientras que la segunda equivale a la distribución de presiones aplicada entre 0 y $a_2/2$. De esta forma, se puede considerar la celosía equivalente formada por las bielas extremas C_1 y C_2 y el trozo de armadura comprendida entre ambas. Asimismo, existen otras dos bielas intermedias (menos cargadas que las mencionadas anteriormente) y un tirante superior que conecta todas las bielas. Todo esto lo podemos ver en la figura 2.3.1.1.1.

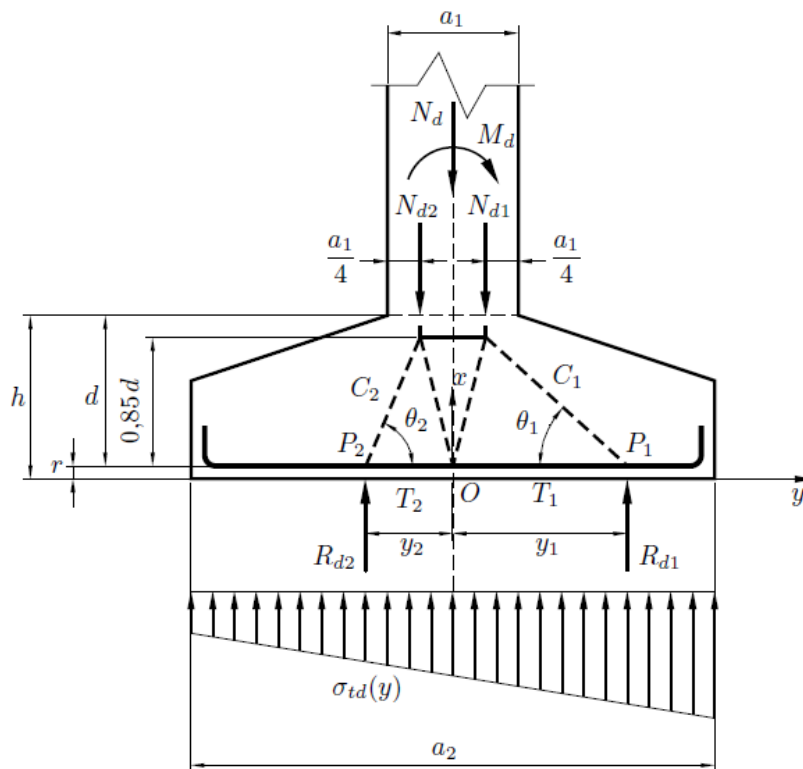


Figura 2.3.1.1.1—Cargas y reacciones aplicadas sobre una zapata para el método discretizado de bielas y tirantes. Fuente: “Cálculo de estructuras de hormigón armado”

Por equilibrio de fuerzas y momentos obtenemos:

$$\begin{cases} N_{d1} = \frac{N_d}{2} \left(1 + 4 \frac{e_d}{a_1} \right) \\ N_{d2} = \frac{N_d}{2} \left(1 - 4 \frac{e_d}{a_1} \right) \end{cases} \quad (2.30)$$

Integrando la distribución de tensiones, en este caso trapezoidal, obtenemos las reacciones R_{d1} y R_{d2} :

$$R_{d1} = \int_0^{a_2/2} \sigma_{td}(y) dy = \frac{N_d}{2} \left(1 + 3 \frac{e_d}{a_2} \right) \quad (2.31)$$

$$R_{d2} = \int_{-a_2/2}^0 \sigma_{td}(y) dy = \frac{N_d}{2} \left(1 - 3 \frac{e_d}{a_2} \right) \quad (2.32)$$

$$y_1 R_{d1} = \int_0^{a_2/2} \sigma_{td}(y) y dy \Rightarrow y_1 = \frac{1}{R_{d1}} \int_0^{a_2/2} \sigma_{td}(y) y dy = \frac{a_2 + 4 e_d}{a_2 + 3 e_d} \frac{a_2}{4} \quad (2.33)$$

$$y_2 R_{d2} = \int_{-a_2/2}^0 \sigma_{td}(y) y dy \Rightarrow y_2 = \frac{1}{R_{d2}} \int_{-a_2/2}^0 \sigma_{td}(y) y dy = \frac{a_2 - 4 e_d}{a_2 - 3 e_d} \frac{a_2}{4} \quad (2.33)$$

Para T_1 y T_2 , tracciones en la armadura, a uno y otro lado del punto O:

$$\tan \theta_1 = \frac{R_{d1}}{T_1} = \frac{0,85 d}{y_1 - \frac{a_1}{4}} \Rightarrow T_1 = \frac{R_{d1}}{0,85 d} \left(y_1 - \frac{a_1}{4} \right) \quad (2.34)$$

$$\tan \theta_2 = \frac{R_{d2}}{T_2} = \frac{0,85 d}{y_2 - \frac{a_1}{4}} \Rightarrow T_2 = \frac{R_{d2}}{0,85 d} \left(y_2 - \frac{a_1}{4} \right) \quad (2.35)$$

Con esto, según la disposición de la figura, dado que $R_{d1} \geq R_{d2}$ y $y_1 \geq y_2$, se tendrá que $T_1 \geq T_2$. Si, como es lógico, se dispone la misma armadura en toda la base de la zapata, el área A_1 de la misma será:

$$A_1 = \frac{T_1}{f_{yd}} = \frac{R_{d1}}{0,85 f_{yd} d} \left(y_1 - \frac{a_1}{4} \right) \quad (2.35)$$

De tratarse de una zapata con un muro excéntrico, el proceso de cálculo será el mismo, modificando los límites de integración de la distribución de tensiones, con lo que para el área se tomará la tensión mayor de las dos calculadas.

2.3.1.2-ARMADURAS FLEXIBLE:

En este caso el dimensionamiento de la zapata se realiza con el estado límite de agotamiento resistente, que también se define en la norma EHE. Para su aplicación, se parte de las siguientes hipótesis:

1. Sobre la deformación de la sección y las ecuaciones de compatibilidad:

- Se suponen pequeñas deformaciones.
- Se dan las condiciones de adherencia perfecta, de modo que las deformaciones normales (alargamientos y contracciones) en las armaduras son iguales a las que se producen en el hormigón que las envuelve.
- Se admite la hipótesis de Navier-Bernoulli, por la que la sección permanece plana tras la deformación, por lo que la ley de deformaciones normales varía linealmente a lo largo de la misma.

2. Sobre la ley de comportamiento del hormigón:

- La pendiente inicial de las curvas tensión-deformación del hormigón (módulo de elasticidad tangente inicial, E_{0j}) se incrementa a medida que aumenta la resistencia a compresión.
- La porción ascendente de las curvas se asemeja a una parábola con vértice en el punto de tensión máxima.

- La deformación ε_0 asociada al valor máximo de la tensión se incrementa con la resistencia del hormigón.
- La pendiente del brazo descendiente es más suave que la del ascendente en hormigones de resistencia moderada. Ésta pendiente descendente se incrementa con la resistencia del hormigón.
- La deformación máxima alcanzada ε_c decrece con la resistencia del hormigón.

3. Sobre la ley de comportamiento del hormigón:

- Se adopta el diagrama birrectilíneo como relación de comportamiento del acero, cuyo tramo lineal tiene pendiente igual al módulo de elasticidad del acero, $E_s = 200000 \text{ MPa}$.
- La deformación del acero en tracción se limita al 10 por 1000, y en compresión se considera la misma que el hormigón, 0.0035.
- El punto del límite elástico, tanto para tracción como para compresión, viene definido por $\varepsilon_y = \frac{f_{yd}}{E_s}$.

4. Sobre las ecuaciones de equilibrio entre fuerzas y esfuerzos:

Se admite la hipótesis de pequeños desplazamientos, por lo que el equilibrio entre las acciones externas y los esfuerzos internos se plantea en la configuración indeformada de la pieza.

5. Sobre las ecuaciones de equivalencia estática entre tensiones y esfuerzos:

Se desprecia la contribución de las tensiones que se originan en el hormigón traccionado, que en condiciones de rotura está fisurado, y es el acero el que pasa a soportar las tensiones de tracción.

A partir de aquí, se definen los llamados dominios de deformación, que asocian la forma de trabajo de la sección a los esfuerzos y deformaciones sobre la misma, recorriendo de manera continua desde la tracción simple hasta la compresión simple. Estos dominios se definen de la siguiente forma:

➤ Dominio 1:

Toda la sección trabaja a tracción, con una ley de deformaciones que pivota entrono al punto de alargamiento más traccionado $\varepsilon_0 = 0.01$ (punto A de la figura 2.3.1.2.1). Comprende pues un intervalo de la línea neutra tal que: $-\infty \leq x \leq 0$.

La rotura en este dominio se produce por exceso de alargamiento en la armadura de tracción, la cual siempre se encuentra en fluencia. Mientras tanto, la armadura de compresión estará en fluencia o no dependiendo de la profundidad Reducida de la línea neutra.

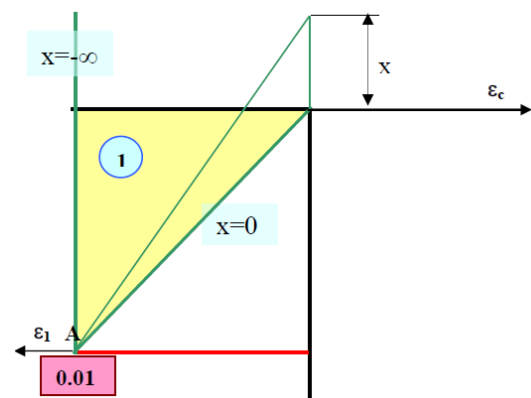


Figura 2.3.1.2.1—Dominio 1. Fuente: "Cálculo de estructuras de hormigón armado"

➤ **Dominio 2:**

A partir de aquí se trabajará a flexión. En este se las leyes de deformación siguen pivotando sobre el mismo punto de la armadura de tracción hasta la que hace que la deformación en la fibra más comprimida del hormigón sea la límite, $\epsilon_c = 0.0035$, comprendiendo un intervalo de la línea neutra tal que: $0 < x < \frac{7}{27} d$.

En este la rotura se produce igual que en el caso anterior, por el alargamiento de la armadura de tracción, que trabaja a fluencia. La armadura de compresión puede trabajar a tracción o compresión, dependiendo de la profundidad de la línea neutra, y hacerlo en fluencia o no dependiendo también de la misma.

➤ **Dominio 3:**

En este caso se pivota respecto del punto B, por lo que la fibra más comprimida del hormigón siempre se encuentra en la deformación límite y se limita inferiormente por la recta $x = \frac{7}{27} d$, con lo que el intervalo de la línea neutra queda tal que: $\frac{7}{27} d \leq x \leq \frac{0.0035}{0.0035 + \epsilon_y} d$.

En este caso las dos armaduras trabajan a fluencia, con lo que resulta ser el más óptimo.

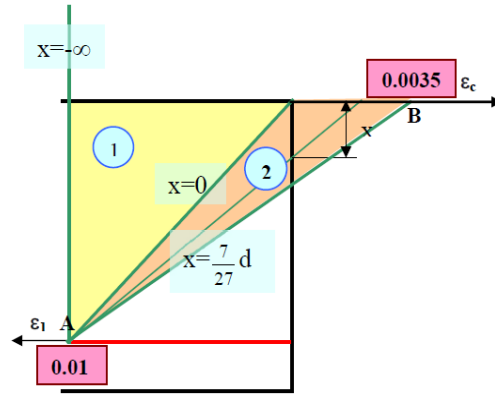


Figura 2.3.1.2.2—Dominio 2. Fuente: “Cálculo de estructuras de hormigón armado”

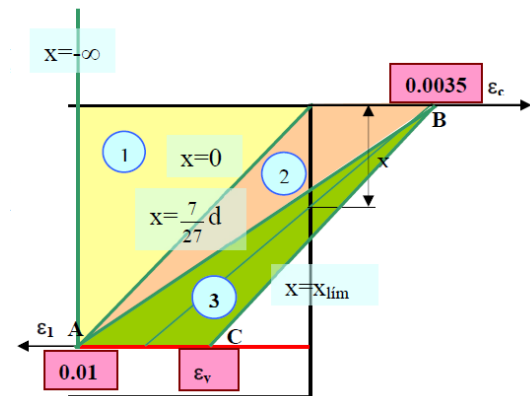


Figura 2.3.1.2.3—Dominio 3. Fuente: “Cálculo de estructuras de hormigón armado”

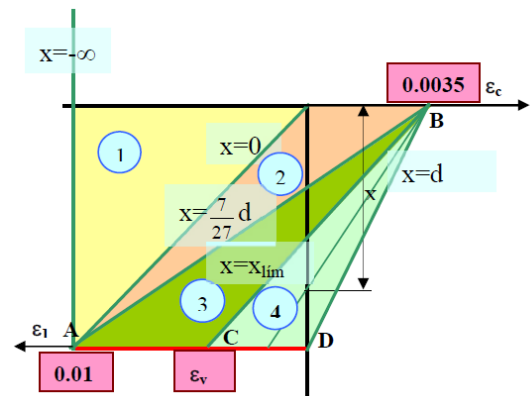


Figura 2.3.1.2.4—Dominio 4. Fuente: “Cálculo de estructuras de hormigón armado”

➤ **Dominio 4:**

Se sigue pivotando respecto al punto B, desde el límite inferior del caso anterior hasta $x = d$, quedando así un intervalo de la línea neutra tal que: $\frac{0.0035}{0.0035 + \epsilon_y} d \leq x \leq d$.

En este caso la armadura de tracción deja de estar en fluencia, mientras que la de compresión continúa estándolo.

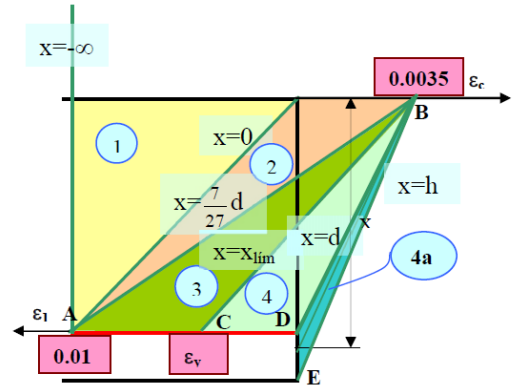


Figura 2.3.1.2.5—Dominio 4.a. Fuente: “Cálculo de estructuras de hormigón armado”

➤ **Dominio 4.a:**

Se sigue pivotando en el punto B hasta que toda la sección trabaja a compresión, quedando así un intervalo de la línea neutra tal que: $d < x < h$.

La armadura de tracción trabaja ahora a compresión por debajo del límite de fluencia, mientras que la de tracción sigue trabajando a fluencia.

➤ **Dominio 5:**

Toda la sección de la armadura trabajará a compresión. Ahora se pivotará en torno al punto $x_G = \frac{3}{7}h$, y la profundidad de la línea neutra abracará el intervalo $h \leq x \leq \infty$.

En este dominio las dos armaduras trabajan a compresión, y mientras que la armadura de compresión siempre trabaja a fluencia, la de tracción lo hará dependiendo de cuál sea la profundidad de la línea neutra.

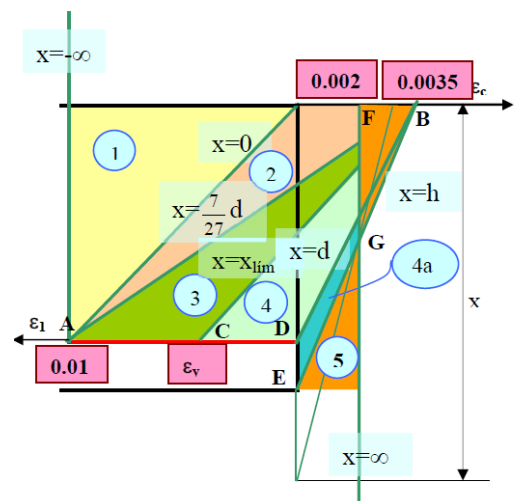


Figura 2.3.1.2.6—Dominio 5. Fuente: “Cálculo de estructuras de hormigón armado”

Con esto, el dimensionamiento de la armadura según este método se realiza, para los dominios de flexión y compresión, de la siguiente forma:

Ecuaciones adimensionales de los dominios 2, 3 y 4:

$$v_d = -\omega_1 \cdot \frac{\sigma_1(\xi)}{f_{yd}} + \omega_2 \cdot \frac{\sigma_2(\xi)}{f_{yd}} + \xi \cdot \psi(\xi) \quad (2.36)$$

$$\mu_{d1} = \omega_2 \cdot \frac{\sigma_2(\xi)}{f_{yd}} \cdot (1 - \delta_2) + \xi \cdot \psi(\xi) \cdot (1 - \xi \cdot \lambda(\xi)) \quad (2.37)$$

Siendo μ_{d1} y v_d los esfuerzos reducidos, ξ la profundidad de la línea neutra reducida y ω_1 y ω_2 las cuantías mecánicas de las armaduras.

La solución óptima se alcanzará en el dominio 3, cuando las dos armaduras trabajan a fluencia, y además cuando la contribución del hormigón μ_{u1}^c sea máxima y las cuantías mínimas. Calculando estos valores óptimos obtenemos:

$$\xi_{\text{opt}} = \frac{119}{198} (1 + \delta_2) \quad \xi_{\text{opt}} \leq \frac{0.0035}{0.0035 + \varepsilon_y} \quad (2.38)$$

$$\mu_{u1, \text{opt}}^c = \xi_{\text{opt}} \cdot \Psi \cdot (1 - \xi_{\text{opt}} \cdot \lambda) \quad (2.39)$$

con δ_2 el recubrimiento mecánico reducido de la armadura menos traccionada.

Como vemos ξ_{opt} está muy próxima al límite del dominio 3 con el 4. Vemos también que si $\mu_{d1} \leq \mu_{u1, \text{opt}}^c$ el hormigón será capaz de resistir todo el momento, por lo que no hace falta armadura de compresión. Por lo tanto, si se cumple la condición anteriormente citada se dimensionará sin armadura de compresión, en el dominio que corresponda. Para ello se calcula ξ a partir de las expresiones (2.36) y (2.37) y con ella ya podemos calcular las cuantías.

Si por el contrario nos encontramos con que $\mu_{d1} > \mu_{u1, \text{opt}}^c$, primero tratará de armarse en el dominio 3 con las dos armaduras, con las siguientes ecuaciones del dominio:

$$v_d = -\omega_1 + \omega_2 + \xi_{\text{opt}} \cdot \Psi \quad \mu_{d1} = \omega_2 \cdot (1 - \delta_2) + \xi_{\text{opt}} \cdot \Psi \cdot (1 - \xi_{\text{opt}} \cdot \lambda) \quad (2.40)$$

y para ello debe cumplirse que $v_d \geq \xi_{\text{opt}} \cdot \Psi + (1 - \delta_2)^{-1} \cdot (\mu_{d1} - \mu_{u1, \text{opt}}^c)$.

De no cumplirse la condición del axil se buscará armar en el dominio 3, 4 o 4.a sin armadura de tracción, para los que se utilizan las fórmulas:

$$v_d = \omega_2 + \xi \cdot \Psi \quad \mu_{d1} = \omega_2 \cdot (1 - \delta_2) + \xi \cdot \Psi \cdot (1 - \xi \cdot \lambda) \quad (2.41)$$

y en el dominio 5 también sin armadura de tracción:

$$v_d = \omega_2 + \delta \cdot \Psi(\xi) \quad \mu_{d1} = \omega_2 \cdot (1 - \delta_2) + \delta \cdot \Psi(\xi) \cdot (1 - \delta \cdot \lambda(\xi)) \quad (2.42)$$

Por último, si se estuviera en el caso de flexión simple ($\xi = \infty$), se dispondrían ambas armaduras de nuevo, y se calcularía su profundidad de línea neutra reducida con las siguientes fórmulas:

$$v_d = \omega_1 + \omega_2 + 0.85 \cdot \delta \quad \mu_{d1} = \omega_2 \cdot (1 - \delta_2) + 0.85 \cdot \delta \cdot \left(1 - \frac{\delta}{2}\right) \quad (2.42)$$

Con esto queda descrito resumidamente cómo se dimensiona la armadura en el caso de flexión. Respecto a la zapata corrida, al calcularse con las consideraciones que se han ido mencionando previamente, se hará siempre en los dominios donde no aparezca armadura de compresión.

2.3.1.3-ANCLAJE DE LA ARMADURA PRINCIPAL:

En ambos casos el método de anclaje es similar, lo único que cambia es la longitud de anclaje disponible:

- **Longitud de anclaje disponible para zapata rígida:** será la menor de las distancias desde los puntos de aplicación de R_{d1} y R_{d2} hasta el extremo pertinente, excluyendo de esta el recubrimiento de la armadura. En el caso representado por la figura 2.3.1.1.1 es $\frac{a_2}{2} - y_1 - r_1$.

- **Longitud de anclaje disponible para zapata flexible:** según lo dispuesto en la norma EHE, esta longitud será la que se corresponda al más desfavorable de los siguientes casos:

- Desde una sección S_2 situada a un canto útil de la sección de referencia S_I , que ya se definió en el apartado 2.2.3.2.
- La armadura se anclará a partir de la sección S_3 , situada a $0,5 h$ del extremo de la zapata

Con esta longitud disponible se comprueba cómo debe anclarse la armadura, sabiendo que l_b es la longitud básica de anclaje, obtenida de la norma EHE en función del diámetro del redondo y el tipo de hormigón:

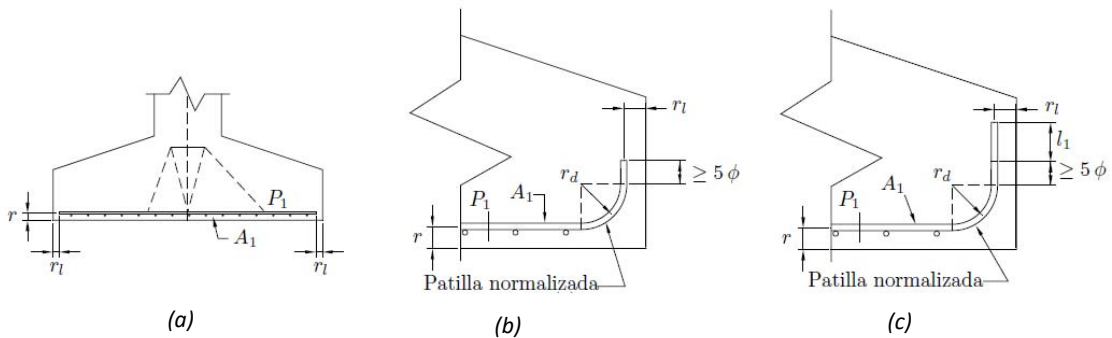


Figura 2.3.1.3.1—Tipos de anclaje de la armadura. Fuente: "Cálculo de estructuras de hormigón armado"

1. $l_b \frac{A_1}{A_{1,real}} \leq l_{disponible}$ (figura 2.3.1.3.1-a): en este caso, hay longitud suficiente para el anclaje por adherencia en prolongación recta.
2. $l_b \frac{A_1}{A_{1,real}} > l_{disponible} \geq 0,7 l_b \frac{A_1}{A_{1,real}}$ (figura 2.3.1.3.1-b): en este caso, no hay longitud suficiente para el anclaje por adherencia en prolongación recta, pero sí la hay para hacerlo mediante una patilla o gancho normalizado.
3. $0,7 l_b \frac{A_1}{A_{1,real}} > l_{disponible}$ (figura 2.3.1.3.1-c): En este caso, tampoco hay longitud suficiente para el anclaje mediante una patilla o gancho normalizado, por lo que es necesario prolongar la armadura verticalmente más allá del gancho una distancia l_1 dada por:

$$l_1 \geq l_b \frac{A_1}{A_{1,real}} - \frac{l_{disponible}}{0,7} \quad (2.43)$$

2.3.2-CÁLCULO DE LA ARMADURA DE REPARTO:

Sobre esta armadura no se ha realizado un cálculo propio, simplemente se estima como el 30% de la armadura principal, estimación suficiente y válida por estar del lado de la seguridad. No podrá ser menor a la definida por la cuantía mínima de la zapata (tabla 2.3.2.1), en este caso la mitad de ella por tratarse de una losa.

Tipo de elemento estructural		Tipo de acero	
		Aceros con $f_y = 400\text{N/mm}^2$	Aceros con $f_y = 500\text{N/mm}^2$
Pilares		4,0	4,0
Losas ⁽¹⁾		2,0	1,8
Forjados unidireccionales	Nervios ⁽²⁾	4,0	3,0
	Armadura de reparto perpendicular a los nervios ⁽³⁾	1,4	1,1
	Armadura de reparto paralela a los nervios ⁽³⁾	0,7	0,6
Vigas ⁽⁴⁾		3,3	2,8
Muros ⁽⁵⁾	Armadura horizontal	4,0	3,2
	Armadura vertical	1,2	0,9

Tabla 2.3.2.1—Cantías geométricas mínimas, en tanto por 1000, referidas a la sección total de hormigón., Seaún norma EHE

2.3.3-CÁLCULO DE LA ARMADURA ESPERA:

Esta armadura será la que realice la unión muro-zapata, de ser el muro de hormigón. Las dimensiones a tratar son las mostradas en la figura 2.3.3.1, siendo \varnothing_0 el diámetro de la armadura del muro un dato conocido.

- L_1 : longitud de solape, según la norma EHE.
- $L_2 : \frac{2}{3} l_b$, siendo l_b la longitud básica de anclaje para la armadura con diámetro \varnothing_0 .
- $L_3 : \frac{1}{3} l_b$, siendo su valor como mínimo 300 mm y además teniendo que estar unida como mínimo a dos barras de la armadura de reparto.

Estas longitudes son así para el caso de un solo redondo, de tratarse de redondos empalmados se realizarán las modificaciones que la norma EHE determina.

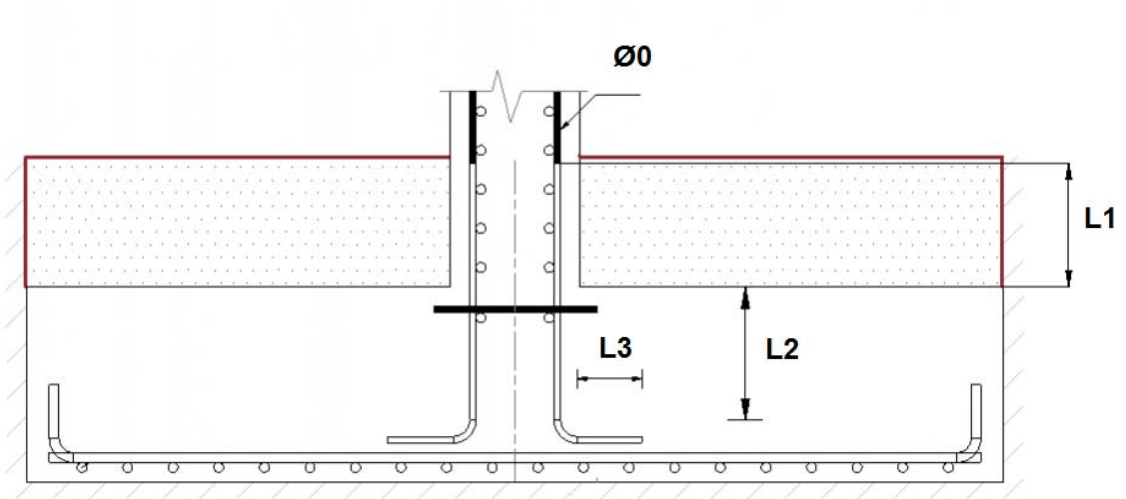


Figura 2.3.3.1—Dimensiones de la armadura en espera.

3-ANDROID:

3.1-GENERALIDADES.

El objetivo de este apartado es el de tener una idea general de lo que es Android hoy en día, lo que abarca y cuáles son sus funciones, ahondando así en la breve justificación que se hizo en el primer apartado sobre el por qué se eligió este sistema operativo para desarrollar la aplicación.

El origen de Android, en líneas generales, se puede resumir con lo siguiente:

- Android inc. se funda en octubre 2003.
- Google compra Android inc. agosto de 2005.
- En noviembre de 2007 se libera el código bajo licencia Apache al crearse la Open Handset Alliance. Se considera el cumpleaños de Android.
- En 2008 se crean los primeros chips compatibles con Android OS por Qualcomm y Texas Instruments y con esto se lanza el primer teléfono Android, el HTC Dream.
- Se empiezan a denominar con nombres de postres a las versiones de Android a partir de 2009.

Con respecto al sistema operativo y su nomenclatura podemos destacar que existen 3 maneras de nombrarlo:

- La comercial, con el nombre de un postre.
- La de los fabricantes, con la versión y subversión. Por ejemplo 4.4.
- La de desarrollador con el nivel de API.

En el caso de nuestra aplicación, esta se ha desarrollado para la versión de Android 4.0.3 IceCreamSandwich API15, siendo elegida por ser suficientemente avanzada para el desarrollo de la aplicación y por abarcar un gran número de dispositivos, ya que aparte de funcionar en dispositivos que montan dicha versión también funciona en los de versiones superiores.

En cuanto a la arquitectura del sistema operativo, comentar a grandes rasgos lo siguiente:

- Aplicaciones: sobre la capa más superficial del SO se pueden ejecutar aplicaciones en java.
- Framework de las aplicaciones: esta capa nos da acceso al API para rehusar componentes o modificarlos.
- Bibliotecas en C/C++: a través de esta capa se usan bibliotecas en estos lenguajes, a través del Framework.
- Runtime en Android: capa que almacena las bibliotecas del lenguaje y la instancia en la máquina virtual Dalvik.
- Núcleo Linux: capa de abstracción del hardware y servicios de seguridad, gestión de memoria, de procesos, pila de red, modelo de los controladores, tec.

Los puntos anteriormente descritos podemos verlos resumidos en la siguiente figura 3.1, donde vemos cada una de estas capas en su correspondiente nivel.

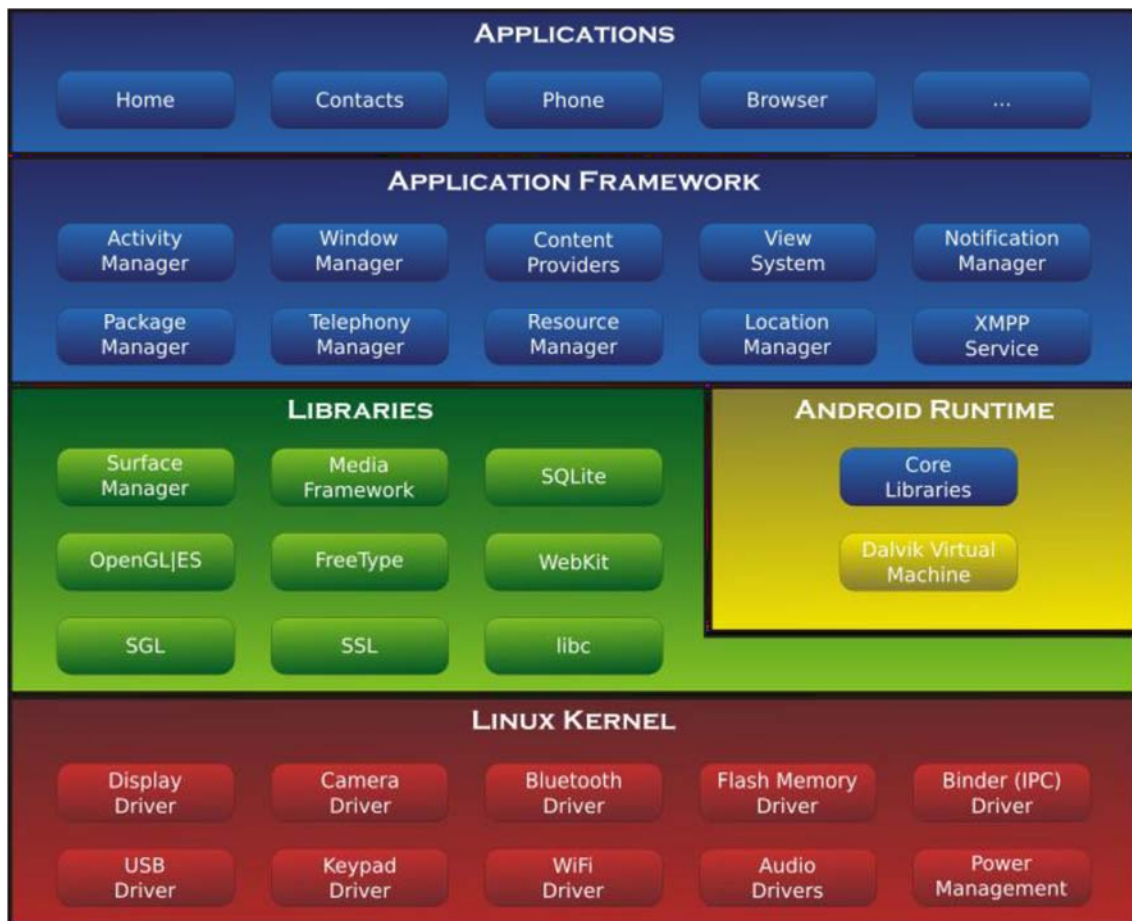


Figura 3.1- Capas del sistema operativo Android. Fuente: Wikipedia

3.2-PROGRAMACIÓN:

En este apartado se va a tratar brevemente y de forma genérica los aspectos básicos de la programación en Android que han sido utilizados para el desarrollo de la aplicación. En un próximo capítulo se tratará de forma específica la implementación realizada de cada uno de ellos.

Antes de empezar a comentar los aspectos propios de la programación, se considera importante introducir el entorno de trabajo elegido, puesto que es al fin y al cabo lo que ha permitido programar la aplicación.

3.2.1-ENTORNO DE DESARROLLO:

El entorno que se ha usado para desarrollar la aplicación ha sido Android Studio, que es el entorno de desarrollo integrado (IDE) oficial para la plataforma Android y para el desarrollo de aplicaciones para la misma.

Las características de este IDE se pueden resumir en los siguientes aspectos:

- Renderización en tiempo real. Lee y ejecuta código XML en tiempo real para agilizar el desarrollo de elementos gráficos.
- Consola de desarrollador: consejos de optimización, ayudas, estadísticas...
- Soporte para construcción basada en Gradle.

- Refactorización en tiempo real.
- Plantillas precargadas con modelos de diseño comunes de Android y otros componentes.
- Soporte para programar aplicaciones para Android Wear.

En resumen, es una IDE como pueden ser NetBeans o Eclipse, pero enfocada enteramente al desarrollo de programas para Android y programar en los lenguajes que éste admite (esencialmente Java y C/C++), y que además nos permite ejecutar nuestras aplicaciones a través de un emulador virtual o de un dispositivo real a través de USB.

Algunos ejemplos de las funcionalidades que nos ofrece esta IDE:

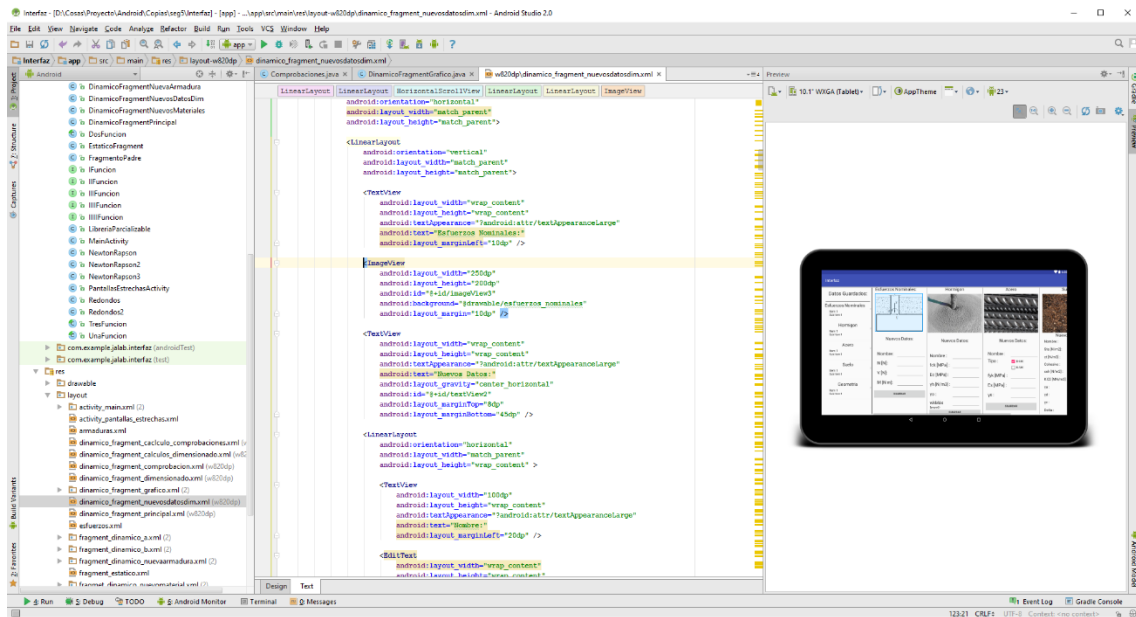


Figura 3.2.1.1-Renderizado en tiempo real de código XML para diseño de layouts



Figura 3.2.1.2-Sugerencias de código

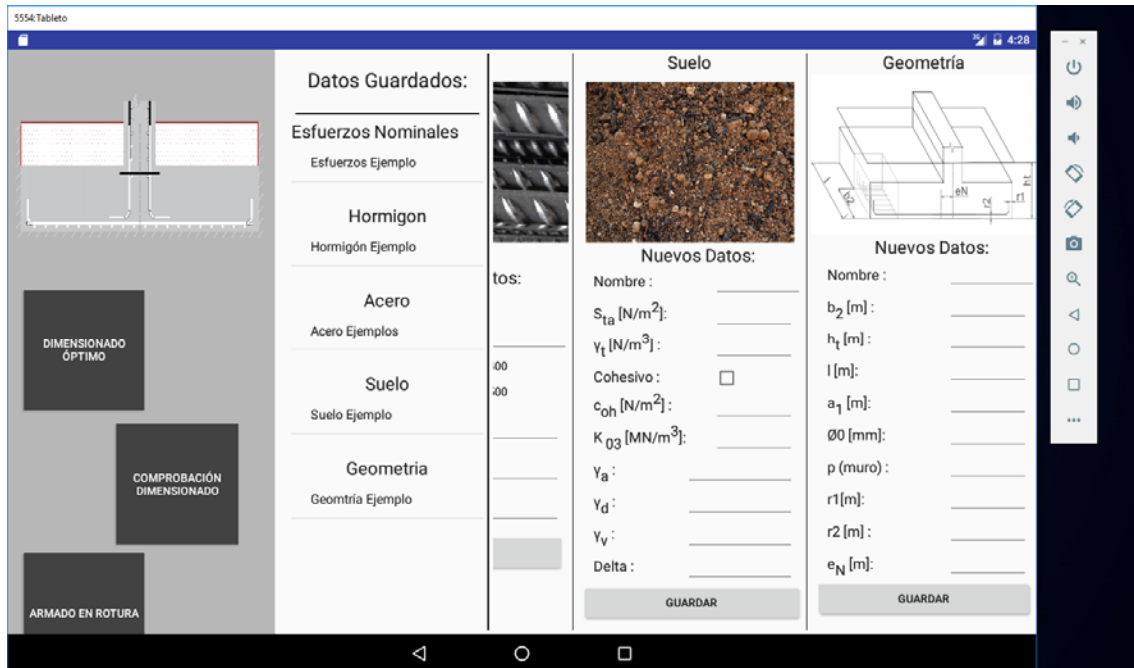


Figura 3.2.1.3-Simulador de dispositivo virtual

3.2.2-FUNDAMENTOS DE PROGRAMACIÓN EN ANDROID:

Ahora toca hablar sobre algunos fundamentos básicos sobre la programación en Android.

Primero destacar que, como ya se ha venido introduciendo, para programar en Android usamos más de un lenguaje. Principalmente podemos distinguirlos de la siguiente manera:

- Lenguaje para programar dinámicamente: en nuestro caso (y lo más habitual) Java. Con el programamos la parte de la lógica. En definitiva, como nos deja entrever su nombre, la parte dinámica del programa.
- Lenguaje para programar estáticamente: para esto se utiliza XML, con el que se programa básicamente los elementos visuales y recursos, los de carácter estático, puesto que también se pueden programar mediante Java si no lo son.

Con esto dicho, toca hablar de la estructura de programación propia de Android, empezando por sus componentes básicos:

- Activity: Representa una pantalla con interfaz de usuario. Se tratará a continuación.
- Service: Se ejecuta en segundo plano. Sirve para realizar aplicaciones que necesiten ejecutarse de continuo o realizar trabajo que provenga de diferentes procesos.
- Content provider: Permite a una App acceder a datos de otra App.
- Broadcast Reciver: Responde a mensajes difundidos a todo el sistema.

Todos estos componentes influyen en el funcionamiento de nuestro dispositivo, por lo que deben de ser declarados sus permisos en el AndroidManifest, que es un archivo XML donde se declaran qué componentes puede ejecutar la aplicación.

Respecto a los componentes mencionados anteriormente cabe destacar que, debido a que el desarrollo de la aplicación objeto de este trabajo no requiere de interacción con otras, nuestra aplicación solamente hace uso de Activities. Sabiendo esto, conviene profundizar en los

componentes utilizados en la aplicación, para comprender su funcionamiento. Estos son básicamente dos: Activities y Fragmentes.

3.2.2.1-ACTIVITIES:

Una Activity es algo enfocado a lo que el usuario puede hacer en el dispositivo (una “Actividad”), algo así como un hilo de ejecución del programa asociado a una pantalla o interfaz gráfica. Además, estas actividades tienen que ser ejecutadas individualmente, ya que no podemos tener varias actividades ejecutándose al mismo tiempo.

Si cambiamos la actividad ejecutada, antes de iniciar la nueva, finaliza la anterior “muriendo” o “pausándose” según el tipo de transición y el momento en el ciclo de vida de la actividad, que veremos a continuación. Así pues, no podemos acceder a información dinámica de una actividad en otra. Si necesitamos continuidad de ejecución entre actividades, que estas compartan información construida dinámicamente en su código, necesitamos que estas se comuniquen enviándose la información pertinente, por ejemplo, enviando esta información mediante objetos de la clase Intent, que nos permiten enviar información de diferentes tipos, desde simples cadenas de caracteres, *Strings*, a objetos más complejos, en formato parcelado para que el núcleo Linux de Android pueda gestionarlo en su memoria.

Las etapas de ejecución de las actividades vienen definidas por su *Ciclo de Vida*, que podemos ver en la figura 3.2.2.1.1. Como vemos en el diagrama, al iniciar la actividad se van ejecutando una serie de métodos en función del estado de la aplicación. Desde *onCreate()* cuando la iniciamos, hasta *onDestroy()* para “destruir” la actividad.

Cuando construimos una clase con la intención de que esta sea una actividad hacemos que herede de Activity, con lo que tenemos disponibles todos los métodos del ciclo de vida para ejecutar código durante el estado que nos interesa. Al hacer esto podemos sobrescribir los métodos que heredamos de Activity y ejecutar código en el momento en que la Activity se encuentre en ese punto de su ciclo de vida. A continuación, un ejemplo de lo mismo:

En el ejemplo anterior vemos como se asocia un layout, *activity_main*, en el momento en que la actividad se crea, pudiendo además en este momento ejecutar más código.

Ejemplo de cómo ejecutar código en un estado de la actividad

```
public class miActividad extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //Nuestro código a ejecutar en este momento
    }
}
```

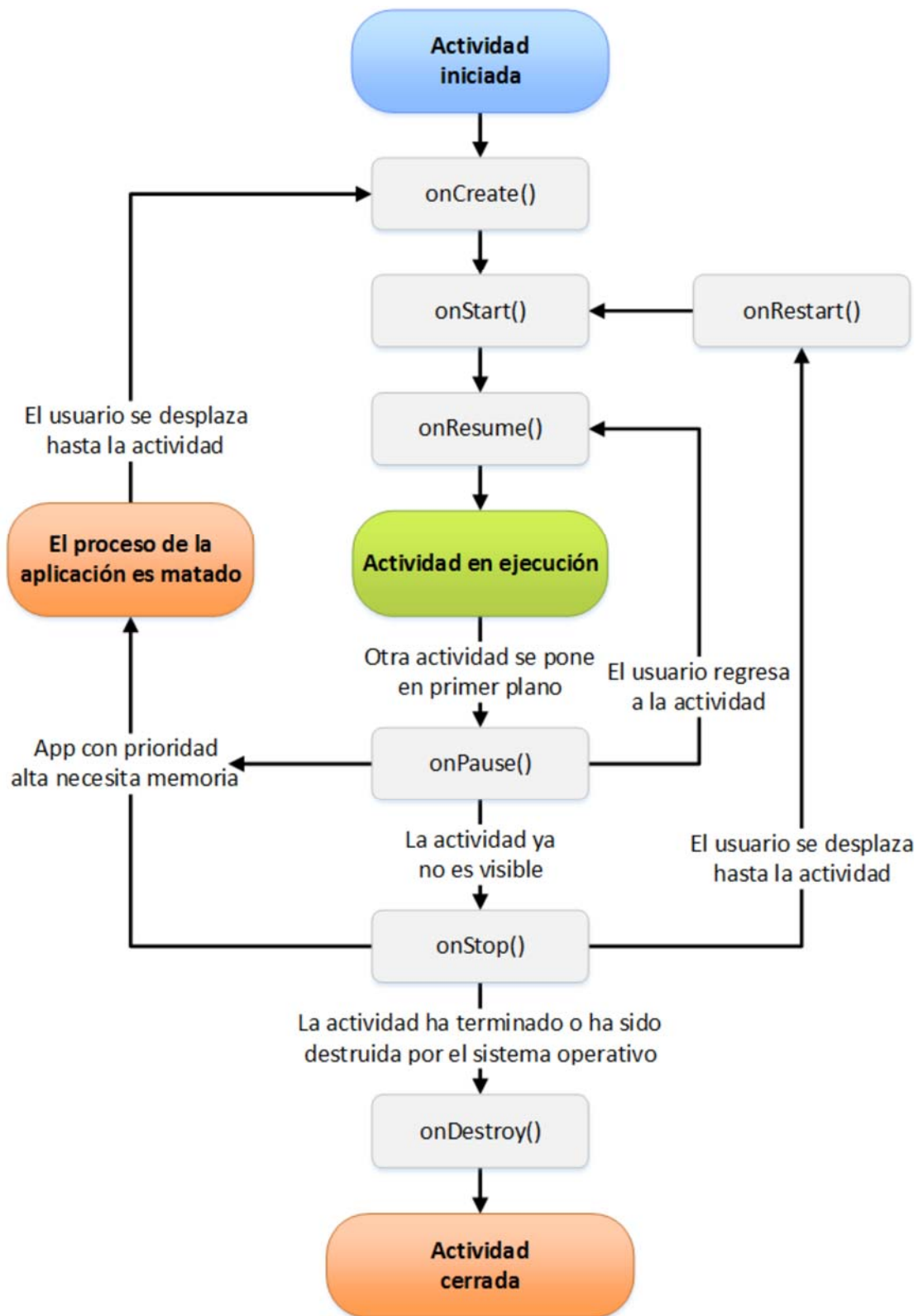


Figura 3.2.2.1.1-Ciclo de vida de una Activity. Fuente: developer.android.com/reference/android/app/Activity.html

Como ya se ha dicho no se pueden ejecutar varias actividades a la vez, por lo que cuando ejecutamos una nueva actividad, lo que hace el sistema operativo es construir una pila de actividades, con lo que la Activity que está en la cima de la pila es la que se está ejecutando y

mostrando al usuario. En el momento en que la Activity sobre la cima se desapile (deje de existir; por ejemplo, al pulsar el botón de “atrás”), la que está por debajo será la que ahora ocupe la cima de la pila y por tanto la que estará al frente del usuario. Este funcionamiento lo podemos ver reflejado en la figura 3.2.2.1.2.

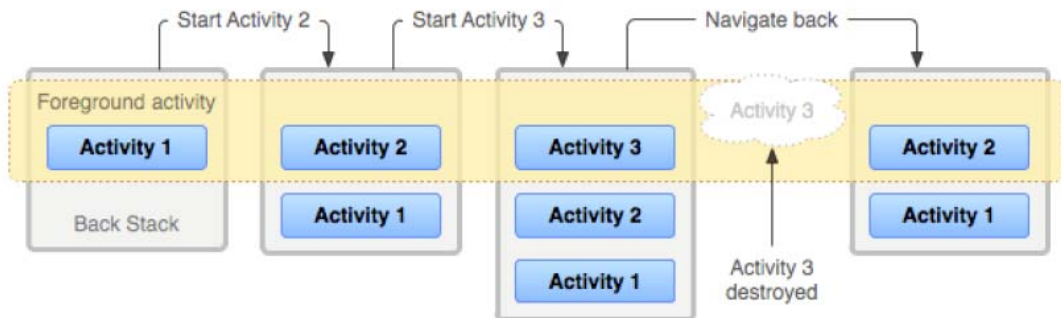


Figura 3.2.2.1.2-Apilamiento de actividades. Fuente: developer.android.com/reference/android/app/Activity.html

Como ya se ha comentado no se pueden ejecutar varias actividades a la vez, pero tampoco se puede dividir una en otras Activities más pequeñas contenidas en una misma Activity. De esto se encargan los Fragments, que veremos a continuación.

3.2.2.2-FRAGMENTS:

Un Fragment es un comportamiento (no visible) o una porción de la interfaz gráfica (visible). Es decir, es un módulo que puede unirse a otros. El principal motivo de que apareciesen los fragments es que una misma aplicación pudiera ser multi-pantalla. Por este motivo, su mayor virtud es que nos permite la reutilización de código.

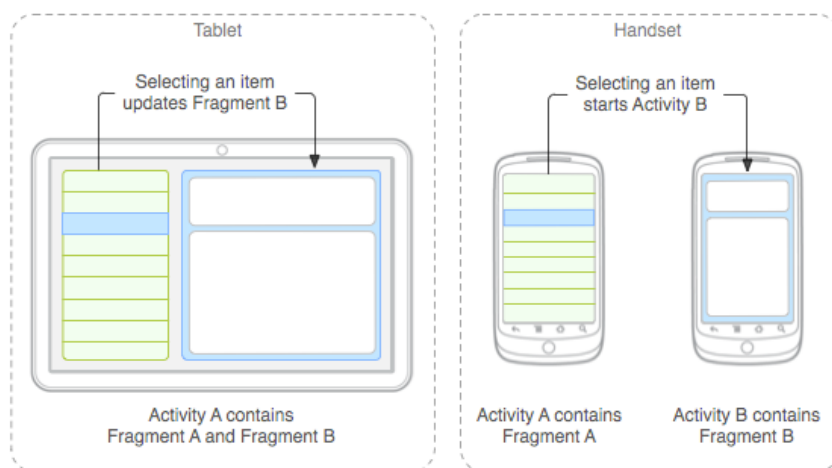


Figura 3.2.2.2.1-Usos de fragments en diferentes pantallas. Fuente: <https://developer.android.com/guide/components/fragments.html>

Las principales ventajas que nos ofrece el uso de fragments son 2:

- Modularidad: se pueden poner donde queramos.
- Reusabilidad: se pueden utilizar tantas veces como se necesiten.

Como podemos ver en la figura el ciclo de vida del Fragment es muy similar al de la Activity, y es fácil entender que su comportamiento es Análogo. Un ejemplo de esto:

Ejemplo de cómo ejecutar código en un estado de un fragment

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                        Bundle savedInstanceState) {
    View rootView=inflater.inflate(R.layout.dinamico_fragment_dimensionado,
                                container, false);

    //Código a ejecutar al crear el Fragment.
    return rootView;
}
```

En este ejemplo vemos lo mismo que en el de la actividad, como asociar un layout al fragment al ser este creado. Vemos que hay diferencias con respecto a como lo hace la actividad, pero su relación con respecto al ciclo de vida es la misma.

3.2.2.3-INTERACCIÓN ACTIVITY-FRAGMENT:

Ya se han explicado lo que son los dos elementos principales de nuestra aplicación y de cualquier aplicación estándar en Android, y ahora se va a explicar cómo interactúan estos dos elementos.

Lo primero a destacar es que un fragment siempre se va a montar sobre una actividad. Por lo que el mismo fragment tiene que ser creado e instanciado en su actividad contenedora, y además necesita recibir la información que necesite para llevar a cabo sus tareas, siendo la manera más sencilla de hacerlo a través de la propia instancia. A continuación, un ejemplo extraído de la propia aplicación objeto de este trabajo:

Activity donde se crea e instancia el Fragment y se le envía información

```
public class MainActivity extends Activity implements FragmentoPadre.FragmentoPadre{
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //Más código.
        DinamicoFragmentNuevaArmadura fragmentElegido=DinamicoFragmentNuevaArmadura
                                                    .newInstance(dArmadura);

        getFragmentManager().beginTransaction()
            .replace(R.id.frameLayout$activity_main$contenedorFragmentsDinamicos,
                fragmentElegido)
            .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
            .commit();
    }
}
```

Fragment creado que recibe la información

```
public class DinamicoFragmentNuevaArmadura extends FragmentoPadre {
    public static DinamicoFragmentNuevaArmadura newInstance(DatosArmadura[]
        dArmaduraGuardados){
        DinamicoFragmentNuevaArmadura f=new DinamicoFragmentNuevaArmadura();
        dArmaduras=dArmaduraGuardados;
    return f;
}
```

En los extractos de código mostrados arriba vemos como instanciamos en la actividad el Fragment, haciéndolo a través de una clase estática definida en la clase del fragmento, por la cual la actividad le envía información y el fragment la recibe, guarda y le devuelve a la actividad un objeto de la propia clase del fragmento, pero con la información ya cargada, que en este caso es un objeto creado previamente.

En este ejemplo también vemos que el fragmento hereda de Fragment (en verdad hereda de FragmentePadre, que no es más que una clase que hereda de Fragment que se ha creado para resolver la aplicación y que se desarrollará más adelante cuando se hable sobre el programa), mientras que la actividad extiende de una interfaz definida en el fragment. Esto se trata de la otra forma de comunicación entre fragment y actividad, que es la de enviar información del fragment a la actividad. Vemos esto sobre el código:

Fragment que envía la información

```
public class DinamicoFragmentNuevaArmadura extends FragmentoPadre {
    public FragmentoPadre mListener;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View rootView =
            inflater.inflate(R.layout.fragment_dinamico_nuevaarmadura,
                container, false);
        Button atrasNuevaArmadura=(Button)
            rootView.findViewById(R.id.atrasNuevaArmadura);
        atrasNuevaArmadura.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                mListener.onClickCambiarFragmentDinamico
                    (EstaticoFragment.FLAG_FRAGMENT_A);
            }
        });
        return rootView;
    }
    @Override
    public void onAttach(Activity activity) {
        super.onAttach(activity);
        try{
            mListener = (FragmentoPadre) activity;
        }catch (ClassCastException e){
            throw new ClassCastException(activity.toString());
        }
    }
}
```

Fragment donde se define la interfaz

```
public class FragmentoPadre extends Fragment{

    public interface FragmentoPadre{
        public void onClickCambiarFragmentDinamico(int flagFragment);
    }
}
```


Activity que recibe la información

```
public class MainActivity extends Activity implements
    FragmentoPadre.FragmentoPadre{

    @Override
    public void onClickCambiarFragmentDinamico(int flagFragment) {
        //Código a ejecutar que hace uso de flagfragment.
    }
}
```

Con esto vemos como se crea una interfaz en el fragmento, en este caso en el `FragmentoPadre`, y se le definen una serie de métodos abstractos, que serán los que nos permitan enviar información entre otras cosas. Esta interfaz deberá ser implementada por la actividad, en donde deberemos sobrescribir sus métodos abstractos con el código que queramos que se ejecute.

En cuanto al fragmento, vemos que en el tenemos que crear un objeto interfaz, al que en el `onAttach()`, que es la parte del ciclo de vida del fragment en el que este se asocia a la actividad, instanciamos este objeto interfaz con la actividad sobre la que está montada, haciendo un `cast` a la misma para darle el formato de la interfaz. Esto se puede hacer porque la actividad implementa la interfaz.

3.2.2.4-INTERACCIÓN ENTRE ACTIVIDADES

Para terminar este apartado sobre programación en Android lo único de interés que nos queda por ver es como intercambiar información entre actividades.

Como ya se ha venido comentando al iniciar una actividad nueva la anterior deja de estar disponible, por lo que si una actividad continua el proceso de otra, o su proceso necesita información procedente de la anterior necesitamos que esta se lo envíe. Para esto se utilizan objetos de tipo `Intent`, que nos permiten trasladar es información de una clase a otra.

Podemos ver su funcionamiento en los siguientes fragmentos de código:

Activity que envía la información

```
public class MainActivity extends Activity{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Intent intent = new
            Intent(getApplicationContext(), PantallasEstrechActivity.class);
        intent.putExtra("dArmadura", dArmadura);
        startActivity(intent);
    }
}
```

Activity que recibe la información

```
public class PantallasEstrechActivity extends Activity{

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pantallas_estrech);

        Intent recibido=getIntent();
        dArmadura=recivido.getExtras().getParcelable("datos");

    }
}
```

Con esto se ve cómo se envía y recibe información entre activities. Si lo que enviamos son elementos simples como cadenas de caracteres o números (int, double, etc.) esto se puede hacer directamente, mientras que, si lo que se envía son elementos más complejos como objetos, necesitamos darles un formato que el sistema operativo puede gestionar en su memoria. Esto se puede hacer mediante las clases Serilizable o Parcelable. En nuestro caso Parcelable, porque es una clase específica de Android, mientras que Serializable es genérica de java.

Esto en otro breve ejemplo:

Clase para un objeto parcelable

```
public class DatosArmadura implements Parcelable {

    public String nombre = " ";

    public DatosArmadura(){
        //Constructor vacío, necesario para parcializar el objeto.
    }
    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(this.nombre);
    }
    protected DatosArmadura(Parcel in) {
        this.nombre = in.readString();
    }
    public static final Parcelable.Creator<DatosArmadura> CREATOR = new
        Parcelable.Creator<DatosArmadura>() {

        @Override
        public DatosArmadura createFromParcel(Parcel source) {
            return new DatosArmadura(source);
        }
        @Override
        public DatosArmadura[] newArray(int size) {
            return new DatosArmadura[size];
        }
    };
}
```

Y con esto queda explicado brevemente los elementos propios de programación en Android utilizados para la aplicación.

4-PROGRAMA:

En este apartado se van a comentar aspectos del programa y de su funcionamiento.

4.1-FUNCIONALIDADES Y FUNCIONAMIENTO DE LA APLICACIÓN:

Antes de comentar aspectos internos del programa como son las clases y los recursos, se va a explicar cuáles son las funcionalidades del programa y cómo funciona este.

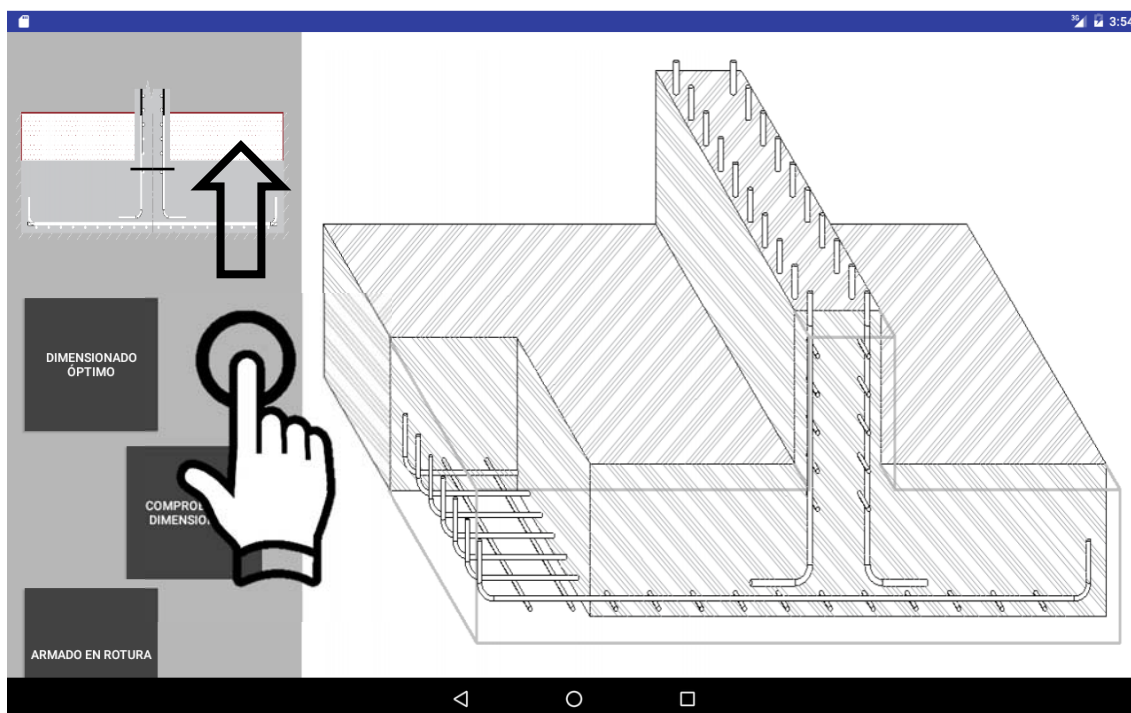


Figura 4.1.1-Pantalla de inicio de la aplicación. Panel izquierdo deslizable.

En primer lugar, al iniciar la aplicación podemos distinguir en la pantalla de inicio dos partes: a la izquierda un panel con botones para navegar entre los distintos módulos y a la derecha un panel mayor, en este caso con una imagen de presentación, pero que será donde se vaya mostrando la información que nos va ofreciendo la aplicación. Esto lo podemos ver en la figura 4.1.1.

A través de este menú lateral podemos acceder a los tres módulos de la aplicación:

- **DIMENSIONADO ÓPTIMO:** en este módulo obtenemos un dimensionado para la zapata que cumple todas las condiciones definidas, así como el valor de los coeficientes de seguridad que justifican esta elección.
- **COMPROBACIÓN DIMENSIONADO:** en este módulo lo que hacemos es, a partir de unas dimensiones de zapata dada, realizar los cálculos de las comprobaciones y cálculo de armaduras necesarias y longitudes de anclaje.

- **ARMADO EN ROTURA:** este módulo nos permite resolver, de manera más pormenorizada, el problema para el caso de una zapata que trabaje a flexión. A diferencia de los anteriores este no busca dimensionar la zapata, sino solamente el acero a disponer óptimo de la misma, con las dimensiones conocidas previamente.

A parte de esto, conviene comentar algunos aspectos sobre el funcionamiento de la aplicación y de sus módulos:

- **Nuevos datos:** en la pantalla para introducir nuevos datos de entrada del problema, común para los módulos de dimensionado y comprobación, aparte de introducir los datos tenemos un par de funcionalidades interesantes.

Al entrar en esta pantalla vemos que se divide en dos zonas, a la izquierda listas donde están guardados los diferentes grupos de datos y a la derecha un panel deslizante horizontalmente donde se encuentran los editables para introducir nuevos datos. Esto lo podemos ver en la figura 4.1.2.

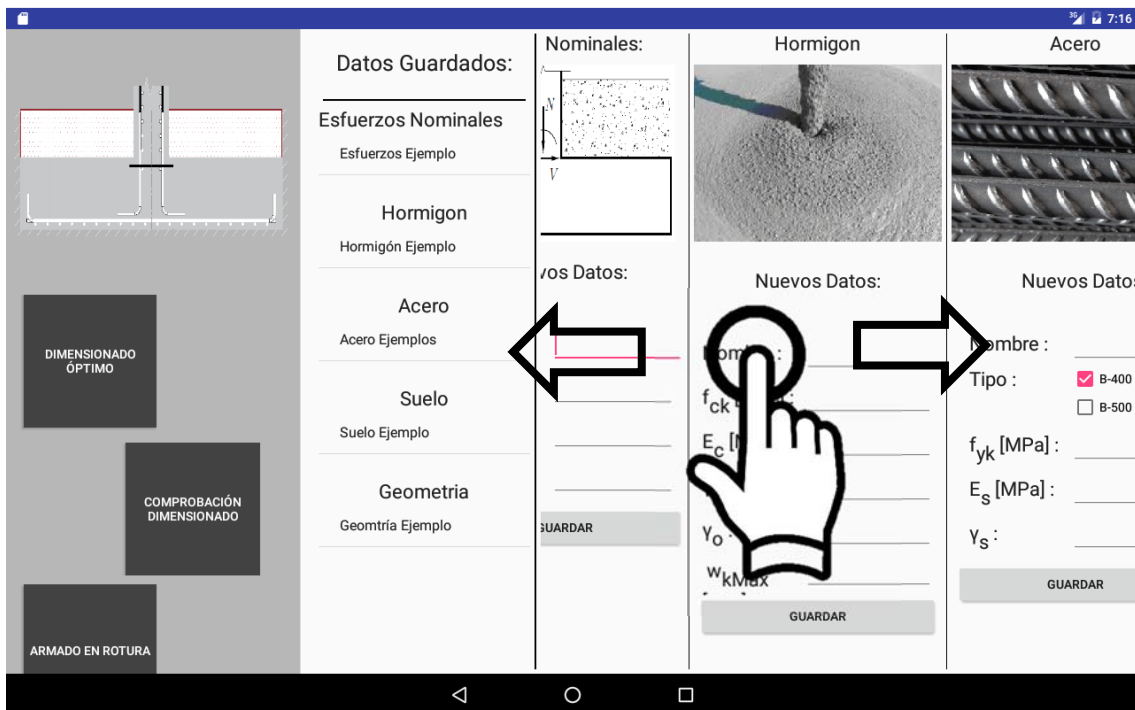


Figura 4.1.2-Pantalla para introducir nuevos datos. Panel derecho deslizable.

Además de esto, en el panel de la izquierda, el que contiene a las listas de objetos guardados, podemos seleccionar cuál de los grupos de datos guardados queremos cargar para los cálculos. Al seleccionar uno los datos cargados serán sobrepresionados en pantalla y además se cargarán en los editables para introducir nuevos datos, pudiendo así reutilizar datos ya cargados e introducir otros nuevos a partir de unos ya existentes, modificando solo lo que resulte necesario. Una vez guardado el nuevo grupo de datos nos aparece disponible en su lista pertinente, en el espacio izquierdo de la pantalla, el que es fijo. Esto lo podemos ver en las figuras 4.1.3 y 4.1.4.

- Comprobación dimensionado:** análogo al anterior, con la única diferencia en que la primera pantalla nos permite además guardar y cargar datos sobre las dimensiones principales (ancho y canto) de la zapata a comprobar. Podemos ver esto en la figura 4.1.6



Figura 4.1.5-Pantalla inicial de dimensionado óptimo. Sobre impresión de los datos cargados.

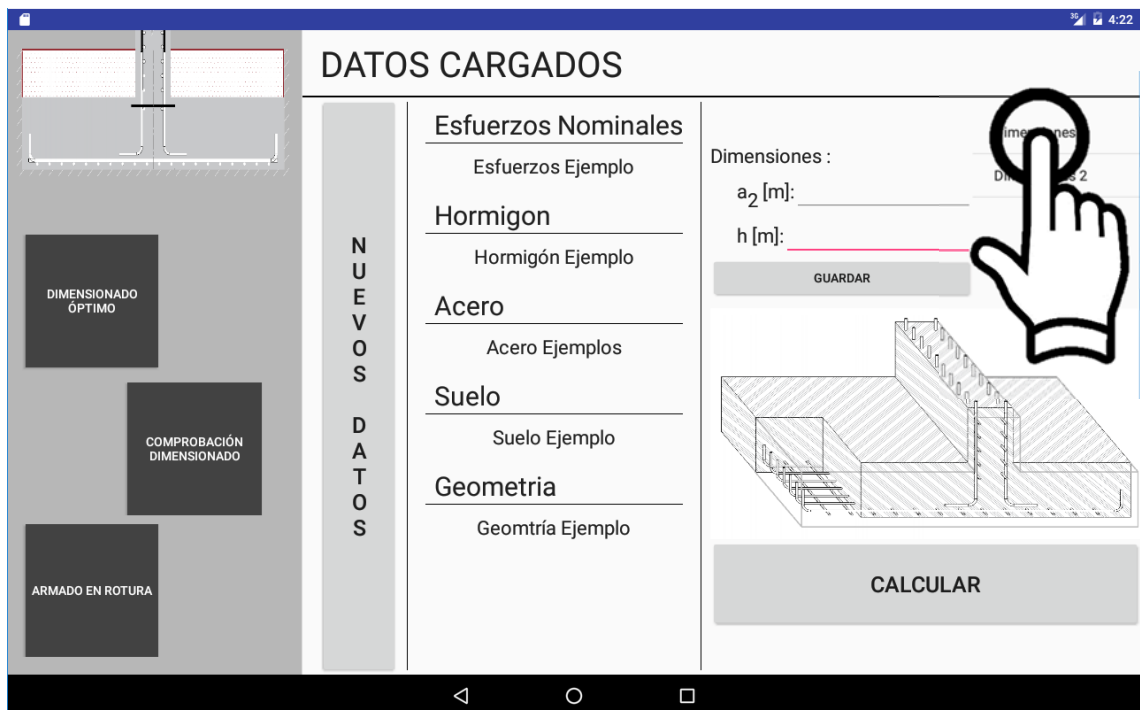


Figura 4.1.6-Pantalla inicial de comprobación dimensionado. Introducción y selección de dimensiones.

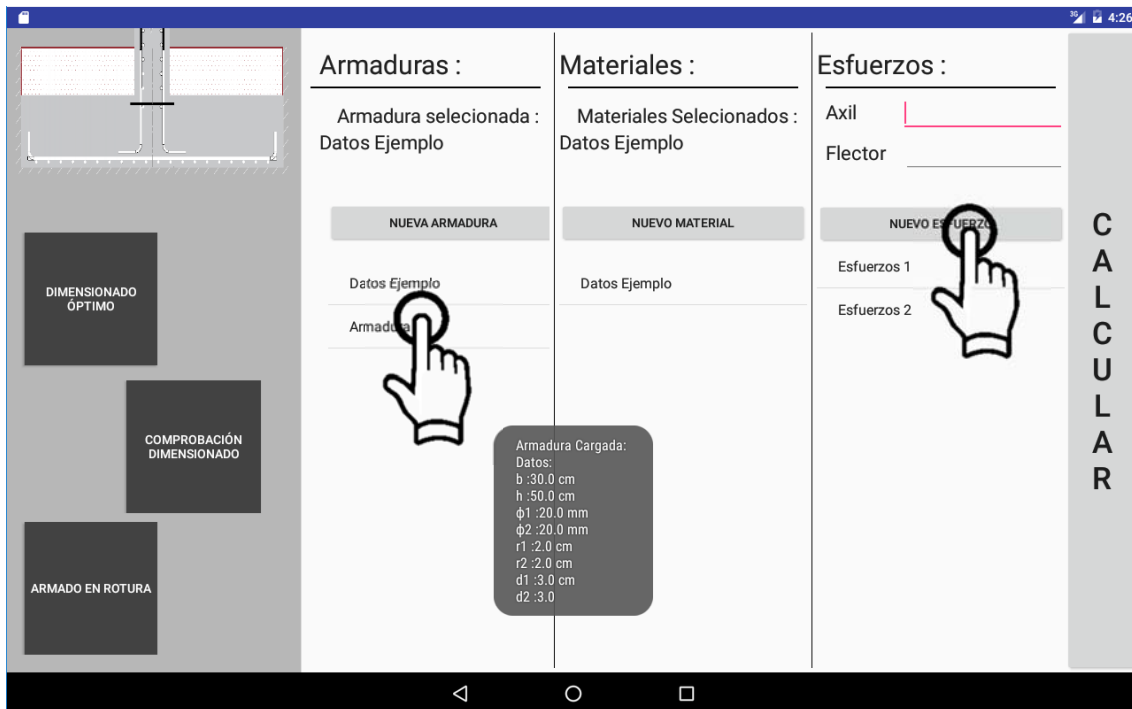


Figura 4.1.7-Pantalla inicial de armado en rotura. Introducción y selección de datos.

- **Armado en rotura:** en este último módulo tenemos varias cosas que comentar. La primera pantalla es de entrada y gestión de datos. Tenemos disponibles los datos cargados de armadura, materiales y esfuerzos. También tenemos acceso a las pantallas para introducir nuevos datos de armadura y materiales, así como a la que muestra los resultados. Por último tenemos también la opción de introducir nuevos esfuerzos. Todo esto lo podemos ver en la figura 4.1.7.

Para terminar con este módulo y con la descripción de las funcionalidades del programa, tenemos las pantallas que nos muestran los resultados de los cálculos, una numéricamente y otra mediante el diagrama de interacción de esfuerzos reducidos. En la primera tenemos una pantalla muy simple donde nos muestra un listado de los resultados disponibles, figura 4.1.8, en donde podemos seleccionar el resultado a mostrar pulsando sobre él. En la segunda tenemos el gráfico que muestra el diagrama de interacción, en donde podemos mostrar los resultados de nuestros cálculos y además los límites de los dominios en el que trabaja la armadura resultado, haciendo esto cuando pulsamos sobre el punto correspondiente en el listado de la izquierda, como muestra la figura 4.1.9.

Con quedan vistas las funcionalidades y funcionamiento de la aplicación. En apartados posteriores de la memoria se realizarán ejemplos con la aplicación, aprovechando estos para explicar más en profundidad todos los elementos que aparecen en pantalla.

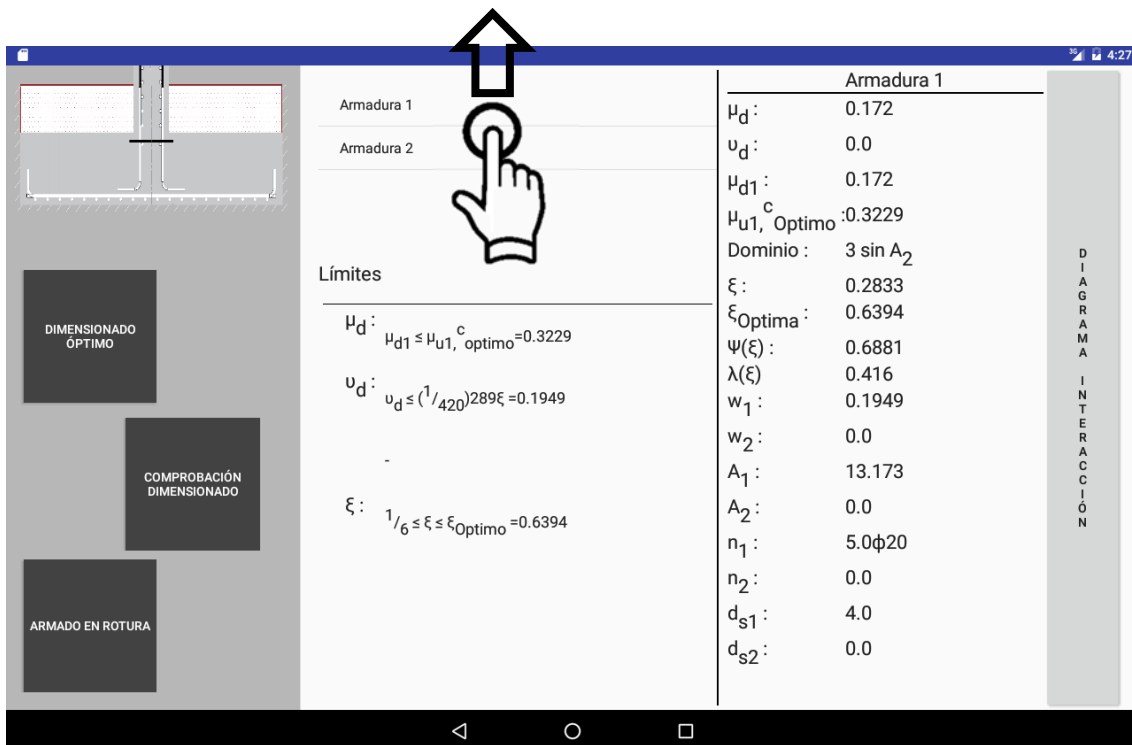


Figura 4.1.8-Pantalla con los resultados numéricos del módulo de Armado en rotura.

4.2-CLASES JAVA:

Las clases java utilizadas en el programa son las siguientes:





































-  Armado
-  Armado2
-  ArmadoNoSimetrico
-  ArmadoNoSimetrico2
-  Comprobaciones
-  ComprobacionesConDimensiones
-  CuatroFuncion
-  DatosAcero
-  DatosArmadura
-  DatosDimensiones
-  DatosEsfuerzos
-  DatosEsfuerzosNominales
-  DatosGeometria
-  DatosHormigon
-  DatosMateriales
-  DatosSuelo
-  DinamicoFragmentA
-  DinamicoFragmentB
-  DinamicoFragmentCalculoComprobaciones
-  DinamicofragmentCalculoDimensionado
-  DinamicoFragmentComprobaciones
-  DinamicoFragmentDatosDim
-  DinamicoFragmentGrafico
-  DinamicoFragmentNuevaArmadura
-  DinamicoFragmentNuevosDatosDim
-  DinamicoFragmentNuevosMateriales
-  DinamicoFragmentPrincipal
-  EstaticoFragment
-  FragmentoPadre
-  IFuncion
-  LibreriaParcializable
-  MainActivity
-  NewtonRapson
-  Redondos
-  Redondos2
-  UnaFuncion

Figura 4.2.1-Clases Java de la aplicación.

La finalidad de este apartado es la de comentar las clases utilizadas en el desarrollo del programa y lo que hace cada una. El código de las mismas será expuesto en el anexo sobre el código.

Respecto a las clases mostradas en la figura 4.1.1 podemos dividir las en dos tipos: clases para el control de la interfaz Android y el resto.

4.2.1-CLASES PARA LA INTERFAZ ANDROID:

Estas clases son las que nos permiten gestionar la interfaz de nuestro programa y los eventos a realizar al interactuar con ellas.

Las clases comprendidas en este apartado son las siguientes:

- **MainActivity:** se trata de la actividad principal del programa. Al arrancar la aplicación esta será la actividad que se empieza a ejecutar.
 - *Hereda de Activity.*
 - *Implementa FragmentoPadre.FragmentPadre.*

- **FragmentoPadre:** es una clase que agrupa todos los métodos abstractos que necesitan los diferentes fragmentos para enviar información a la actividad principal, creando la interfaz que los contenga a todos, FragmentPadre. El objetivo de esta clase es simplificar el uso de todos estos métodos en los diferentes fragmentos.
 - *Hereda de Fragment.*

- **EstaticoFragment:** clase que muestra el fragmento estático del programa, que es el situado a la izquierda de la pantalla, que no cambia en ningún momento. Su función es la de permitirnos navegar por las diferentes pantallas de la aplicación de una manera sencilla.
 - *Hereda de FragmentoPadre.*

- **DinamicoFragmentPrincipal:** los DinamicoFragment son los fragmentos dinámicos, los que sí que van cambiando según nos movemos dentro de la aplicación. Este en concreto es el que nos lleva a la pantalla de inicio de la aplicación, que no tiene más funcionalidad que la de presentar la aplicación.
 - *Hereda de Fragment.*

- **DinamicoFragmentA:** se trata de la primera pantalla del módulo de armado en rotura, donde podemos gestionar datos guardados de armadura y materiales e introducir otros nuevos de esfuerzos.
 - *Hereda de FragmentoPadre.*

- **DinamicoFragmentB:** este nos muestra los resultados del módulo de armado en rotura.
 - *Hereda de FragmentoPadre.*

- **DinamicoFragmentNuevaArmadura:** todavía en el módulo de armado en rotura, esta clase nos permite introducir por pantalla una nueva geometría de la armadura.
 - *Hereda de FragmentoPadre.*

- **DinamicoFragmentNuevosMateriales:** análoga a la anterior, pero para introducir datos de materiales.
 - *Hereda de FragmentoPadre.*

- **DinamicoFragmentDatosDim:** primera clase del módulo de dimensionamiento óptimo. En ella podemos ver los datos que están cargados y desplazarnos a otras pantallas.
 - *Hereda de FragmentoPadre.*

- **DinamicoFragmentComprobaciones:** análogo a la anterior, pero para el módulo de comprobación de dimensionado. En esta además de lo mismo que en la anterior también podemos introducir unas dimensiones de una zapata para calcular el área de acero a disponer y comprobar si cumple con las condiciones de hundimiento, cortante, etc.
 - *Hereda de FragmentoPadre.*

- **DinamicoFragmentNuevosDatosDim:** a través de esta clase podemos introducir los datos de entrada tanto del módulo de dimensionado óptimo como para el de comprobación de dimensionado.
 - *Hereda de FragmentoPadre.*

- **DinamicoFragmentCalculoDimensionado:** nos muestra los resultados del módulo de dimensionado óptimo.
 - *Hereda de FragmentoPadre.*

- **DinamicoFramentCalculoComprabaciones:** análogo al anterior, pero para el módulo de comprobación de dimensiones.
 - *Hereda de FragmentoPadre.*

Con esto ya hemos visto de una manera muy sintetizada las clases java que nos permiten gestionar el programa sobre Android y someramente cuál es su objetivo. Más adelante se explicará de manera esquemática cómo funcionan estas clases y como interaccionan entre sí.

4.2.2-OTRAS CLASES:

Aquí se engloban todas las otras clases, tanto las que sirven para realizar los cálculos pertinentes como las que sirven para construir objetos donde almacenar datos. Entre ellas hay algunas que están duplicadas. Estas son esencialmente las mismas clases, pero con diferentes argumentos de entrada y de salida. Esto se podría haber resuelto añadiendo constructores y métodos, pero por simplicidad se ha optado por duplicarlas con los mínimos cambios necesarios.

Las clases comprendidas en este apartado son las siguientes:

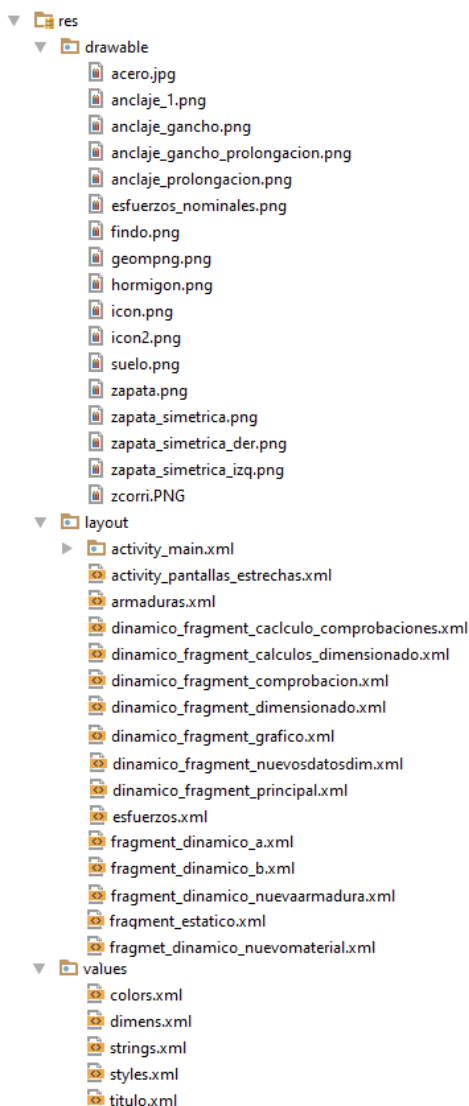
- **Armado/Armado2:** estas clases comprenden todos los métodos necesarios para calcular el armado óptimo de una zapata flexible, como por ejemplo la profundidad de la línea neutra óptima o los límites de los diferentes dominios.
- **ArmadoNoSimetrico/ArmadoNoSimetrico2:** en estas se construyen los objetos que serán las armaduras en el caso flexible, pudiendo acceder si al área de acero a disponer.
 - *Heredan de Armado/Armado2.*
- **Redondos/Redondos2:** con estas clases se realizan los cálculos necesarios para dimensionar los redondos, tanto en el caso flexible como en el rígido como también en para la armadura de reparto, una vez se tiene el área de acero a disponer.
- **Comprobaciones:** esta clase efectúa las comprobaciones pertinentes y nos devuelve los datos de las dimensiones óptimas y de las comprobaciones realizadas.
- **ComprobacionesConDimensiones:** análoga a la anterior, pero con las dimensiones como datos de entrada. La función de esta clase es solamente la de realizar las comprobaciones.
- **NewtonRaphson:** esta clase nos permite resolver ecuaciones no lineales mediante el método de Newton-Raphson.
- **IFuncion:** en esta clase definimos una interfaz para poder construir funciones en cualquier otra clase.
- **Una Función:** clase Abstracta que implementa IFuncion, para poder usar esta.
 - *Implementa IFuncion.*
- **Clases para almacenar los datos de los materiales:** estas clases son las siguientes:
 - DatosAcero.
 - DatosArmadura.
 - DatosDimensiones.
 - DatosEsfuerzos.
 - DatosEsfuerzosNominales.

- DatosGeometría.
- DatosHormigon.
- DatosMateriales.
- DatosSuelo.

Por sus nombres podemos saber fácilmente que datos almacena cada clase.

4.3-RECURSOS:

Los recursos son archivos donde se almacenan otras partes del programa. Principalmente son archivos XML, pero también hay imágenes y estilos.



Respecto a los archivos XML, estos contienen información en dicho formato de

Fragmento de archivo XML con Strings.

```

<resources>
  <string name="app_name">"Interfaz"</string>

  <string name="fi1">φ<sub>1</sub> :</string>
  <string name="boton_1">Dimensionado
    Óptimo</string>
  <string name="boton_2">Comprobación
    Dimensionado</string>
  <string name="boton_3">Armado en
    rotura</string>

  <string name="b">b :</string>
  <string name="h">h :</string>
  <string name="fi2">φ<sub>2</sub>
    [mm] :</string>
  <string name="r1">r<sub>1</sub> :</string>
</resources>

```

diferentes elementos del programa, como los layout, cadenas de caracteres almacenadas en un mismo archivo para poder acceder a ellas desde cualquier archivo, arrayws con cadenas de caracteres y otros elementos utilizados por el programa a nivel interno. Podemos ver un listado de estos elementos en la figura 4.3.1.

Podemos ver a continuación un ejemplo de cómo están escritos estos archivos XML.

Figura 4.3.1-Clases Java de la aplicación.

4.4-CÁLCULOS:

Para terminar este apartado sobre el funcionamiento del programa se van a describir cómo funcionan las clases que resuelven los problemas planteados por la teoría. Para ver esto sin la necesidad de extendernos viendo el código directamente, se va a describir su funcionamiento a través de diagramas de flujo.

Las clases que realizan los cálculos necesarios para obtener los resultados de nuestra aplicación son las siguientes: **comprobaciones/comprobacionesConDimensiones**, **armadoNoSimetrico/armadoNoSimetrico2** y **redondos/redondos2**.

De las clases mencionadas anteriormente ya se ha comentado cuál es su objetivo en el apartado donde se describen las clases del programa. Ahora se va a ver cómo resuelven su cometido.

Mientras que las clases de comprobaciones son clases extensas con muchos métodos, las otras son más breves, la de **armadoNoSimetrico** porque hereda todos sus métodos de **armado** y por tanto en ella solo se hacen los cálculos necesarios, y la clase de **redondos** solamente dimensiona los redondos necesarios.

Por lo anteriormente dicho vamos explicar estas clases de la siguiente forma: para las clases **comprobaciones/comprobacionesConDimensiones** se mostrará un diagrama de flujo para ver como interaccionan todos sus métodos y para **redondos/redondos2** y **armadoNoSimetrico/armadoNoSimetrico2** veremos en resumen de cómo funcionan.

1. **armadoNoSimetrico/armadoNoSimetrico2:** a partir de los datos de entrada del problema calcula los esfuerzos reducidos, y en función de en qué dominio se encuentren estos realiza unos cálculos u otros, según lo visto en la teoría del apartado 2.4.2, obteniendo así las áreas de acero a disponer, que necesitamos para cualquiera de los tres módulos de la aplicación.

Como ya se comentó antes **armadoNoSimetrico/armadoNoSimetrico2** son esencialmente las mismas clases, pero con distintos argumentos de entrada, resuelto sí por simplicidad.

2. **redondos/redondos2:** estas clases nos calculan los redondos a partir de un área dada. Parte generalmente del diámetro de redondo más pequeño que permite la norma, 12mm para la armadura principal y 10mm. El algoritmo calcula el número de redondos y la distancia entre ellos, y si la distancia es menor que 2cm, el valor del tamaño del diámetro o 1.25 veces el tamaño máximo del árido.
3. **comprobaciones/comprobacionesConDimensiones:** esta clase comprende todos los cálculos para dimensionar la zapata y realizar las comprobaciones pertinentes. Hace uso además de las dos anteriores. Las dos son básicamente las mismas con la única diferencia que la primera construye pares de soluciones canto-ancho y la segunda parte de un ancho-canto dado.

Estas se pueden explicar a través del siguiente diagrama de flujo de la figura 4.4.1.

COMPROBACIONES

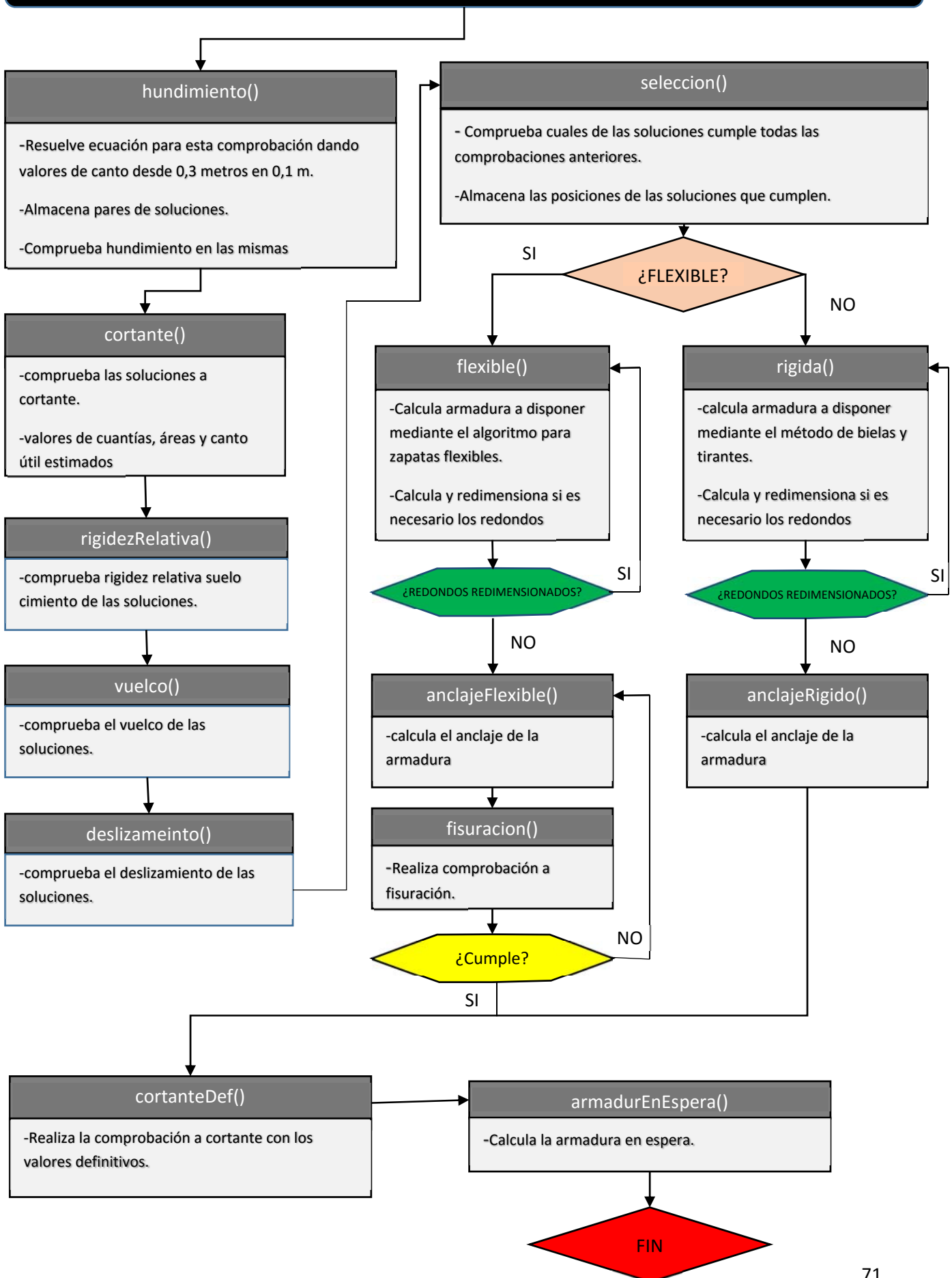


Figura 4.4.1-Clases Java de la aplicación.

5-EJEMPLOS DE APLICACIÓN:

En este apartado se van a resolver algunos ejercicios de ejemplo, para así terminar de ver qué es lo que se puede con la aplicación y aprovechar los mismos para explicar qué son cada uno de los elementos que se muestran en cada una de las pantallas.

Para comenzar vamos a ver el ejemplo que viene cargado como datos de ejemplo. Podemos ver estos datos en la figura 5.1.1.

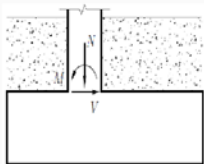
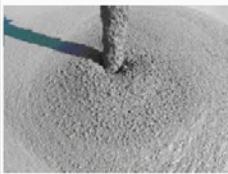


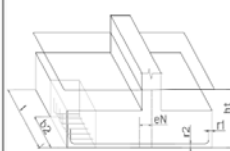
Esfuerzos Nominales:	Hormigon	Acero	Suelo	Geometría
 <p>Nuevos Datos:</p> <p>Nombre: <u>Esfuerzos</u></p> <p>N [N]: <u>400000.0</u></p> <p>V [N]: <u>10000.0</u></p> <p>M [N m]: <u>30000.0</u></p> <p>GUARDAR</p>	 <p>Nuevos Datos:</p> <p>Nombre: <u>Hormigón Ejemplo</u></p> <p>f_{ck} [MPa]: <u>25.0</u></p> <p>E_c [MPa]: <u>20000.0</u></p> <p>γ_h [N/m³]: <u>25000.0</u></p> <p>γ_c: <u>1.5</u></p> <p>$w_{kMáx}$ [mm]: <u>0.4</u></p> <p>GUARDAR</p>	 <p>Nuevos Datos:</p> <p>Nombre: <u>Acero Ejemplos</u></p> <p>Tipo: <input checked="" type="checkbox"/> B-400 <input type="checkbox"/> B-500</p> <p>f_{yk} [MPa]: <u>400.0</u></p> <p>E_s [MPa]: <u>200000.0</u></p> <p>γ_s: <u>1.15</u></p> <p>GUARDAR</p>	 <p>Nuevos Datos:</p> <p>Nombre: <u>Suelo Ejemplo</u></p> <p>S_{ta} [N/m²]: <u>200000.0</u></p> <p>γ_t [N/m³]: <u>20000.0</u></p> <p>Cohesivo: <input checked="" type="checkbox"/></p> <p>c_{oh} [N/m²]: <u>87500.0</u></p> <p>K_{03} [MN/m³]: <u>45.0</u></p> <p>Delta: <u>0.0</u></p> <p>GUARDAR</p>	 <p>Nuevos Datos:</p> <p>Nombre: <u>Geometría</u></p> <p>b_2 [m]: <u>1.0</u></p> <p>h_1 [m]: <u>2.0</u></p> <p>l [m]: <u>10.0</u></p> <p>a_1 [m]: <u>0.25</u></p> <p>$\emptyset 0$ [mm]: <u>20.0</u></p> <p>p (muro): <u>0.35</u></p> <p>$r1$ [m]: <u>0.07</u></p> <p>$r2$ [m]: <u>0.03</u></p> <p>e_N [m]: <u>0.0</u></p> <p>GUARDAR</p>

Figura 5.1.1-Todos los datos de ejemplo que la aplicación trae cargados.

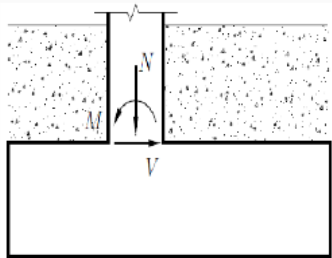
Vamos a ver y explicar cada uno de los grupos de datos:

- **Esfuerzos nominales:** se tratan de los esfuerzos sin minorar que transmite el muro a la zapata. La propia figura que acompaña a los datos nos muestra cual es cual. Esto y sus valores podemos verlo en la figura 5.2.

Recopilando el valor de estos datos y significado de estos datos:

- **N= 400000** Valor del axil en Newtons
- **V=10000** Valor del cortante en Newtons
- **M=30000** Valor del flector en Newtons

Esfuerzos Nominales:



Nuevos Datos:

Nombre: Esfuerzos

N [N]: 400000.0

V [N]: 10000.0

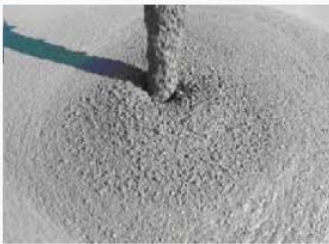
M [N m]: 30000.0

GUARDAR

Figura 5.1.2-Valor de los esfuerzos nominales

- **Hormigon:** en este apartado tenemos los datos concernientes al hormigon, que podemos ver en la figura 5.1.3. Estos son:
 - f_{ck} : resistencia de proyecto el hormigón a compresión, en MPa.
Su valor en el ejemplo es de 25 MPa.
 - E_c : módulo de elasticidad medio del hormigón, en MPa.
Su valor en el ejemplo es de 20000 MPa.
 - γ_h : peso específico de del hormigon utilizado, en N/m³.
Su valor en el ejemplo es de 25 kN/ m³.
 - γ_c : coeficiente de minoración o seguridad de la resistencia de proyecto el hormigón a compresión.
Su valor en el ejemplo es de 1.5.
 - w_{kMax} : abertura máxima de fisuración según la norma, dad en mm.
Su valor en el ejemplo es de 0.4.

Hormigon



Nuevos Datos:

Nombre : Hormigón Ejemplo

f_{ck} [MPa] : 25.0

E_c [MPa] : 20000.0

γ_h [N/m³] : 25000.0

γ_c : 1.5

$w_{kMáx}$ [mm] : 0.4

GUARDAR

Figura 5.1.3-Valor de los datos del hormigón

- **Acero:** en este apartado tenemos los datos concernientes al acero, que podemos ver en la figura 5.1.4. Estos son:
 - **Tipo de acero:** tenemos disponibles los dos tipos de hacer habituales para este tipo de usos, el B-400/B-400-S y el B-500/B-500-S.
El seleccionado en el ejemplo es el B-400.
 - f_{yk} : límite elástico del acero en proyecto para las armaduras pasivas, en MPa.
Este dato es redundante con el anterior, puesto que el tipo de acero ya viene asociado a un valor de f_{ck} , pero se ha decidido poder modificarlo para tener más posibilidades.
Su valor en el ejemplo es de 400 MPa.
 - γ_s : coeficiente de minoración o seguridad del límite elástico del acero.
Su valor en el ejemplo es de 1.15.
 - E_s : módulo de elasticidad del acero, en MPa.
Su valor en el ejemplo es de 200000 MPa.

Acero



Nuevos Datos:

Nombre : Acero Ejemplos

Tipo : B-400
 B-500

f_{yk} [MPa] : 400.0

E_s [MPa] : 200000.0

γ_s : 1.15

GUARDAR

Figura 5.1.4-Valor de los datos del acero

- **Suelo:** en este apartado tenemos los datos concernientes al suelo, que podemos ver en la figura 5.1.5. Estos son:
 - **St_a:** presión máxima admisible o de hundimiento del terreno, en N/m², minorado con su coeficiente parcial correspondiente (tabla 2.2.4.1) como se comenta en el punto 2. Su valor en el ejemplo es de 200 kN/ m².
 - **γ_t:** peso específico de del terreno, en N/m³. Su valor en el ejemplo es de 20 kN/ m³.
 - **Cohesivo:** determina si el suelo es cohesivo o no. En el caso del ejemplo sí lo es.
 - **C_{oh}:** valor de cálculo de la cohesión del terreno sin minorar, en N/ m². Su valor en el ejemplo es de 87500 N/ m².
 - **K₀₃:** coeficiente de balasto to para una placa de dimensión 30x30 cm, en MN/ m³. Su valor en el ejemplo es de 45 MN/ m³.
 - **Delta:** ángulo de rozamiento entre el suelo y la zapata.

Suelo



Nuevos Datos:

Nombre :

S_{ta} [N/m²]: 200000.0

γ_t [N/m³]: 20000.0

Cohesivo :

c_{oh} [N/m²]: 87500.0

K₀₃ [MN/m³]: 45.0

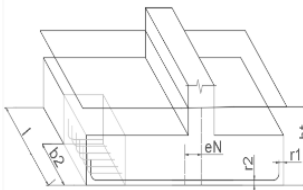
Delta : 0.0

GUARDAR

Figura 5.1.5-Valor de los datos del terreno

- **Geometría:** en este apartado tenemos los datos concernientes a la geometría exterior de la zapata, que podemos ver en la figura 5.1.6. Estos son:
 - **b₂:** unidad de largo por la que se realizan los calculos, en m. Recomendable usar 1 m. Su valor en el ejemplo es de 1m.
 - **h_t:** distancia entre la superficie del suelo y la superior de la zapata. Su valor en el ejemplo es de 2 m.
 - **l:** longitud total del muro. Su valor en el ejemplo es de 10 m.
 - **a₁:** ancho del muro, en m. Su valor en el ejemplo es de 0.25 m
 - **∅₀:** diámetro de la armadura del muro, en mm. Su valor en el ejemplo es de 20 mm.
 - **p:** valor que depende del material de construcción del muro, 0.35 si es de hormigón y 0,25 si es de ladrillo. En nuestro ejemplo será de hormigón.
 - **r₁:** recubrimiento geométrico en los laterales, en m. Su valor en el ejemplo es de 0.07.
 - **r₂:** recubrimiento geométrico en la base, en m. Su valor en el ejemplo es de 0.03

Geometría



Nuevos Datos:

Nombre : Geometria

b₂ [m]: 1.0

h_t [m]: 2.0

l [m]: 10.0

a₁ [m]: 0.25

∅₀ [mm]: 20.0

p (muro) : 0.35

r₁[m]: 0.07

r₂ [m]: 0.03

e_N [m]: 0.0

GUARDAR

Figura 5.1.6-Valor de los datos de la geometría

- e_N : excentricidad del muro, en m. Será positiva si este se encuentra a la derecha del centro de la zapata y negativa si lo está a la izquierda.
Su valor en el ejemplo es de 0.

Con estos datos cargados vamos a ver qué resultados vamos obteniendo, primero en el módulo de dimensionado óptimo:

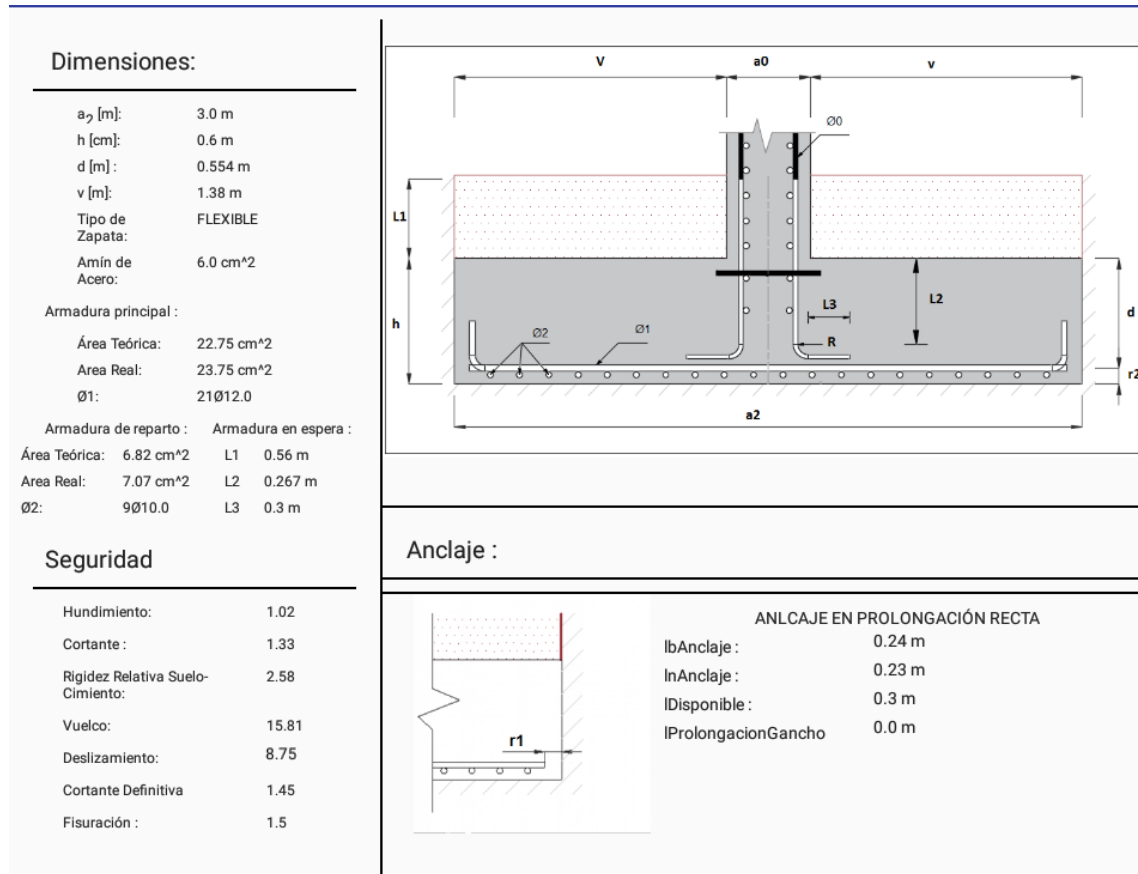


Figura 5.1.7-Pantalla donde se muestran los resultados del dimensionado óptimo

En esta pantalla de resultados, la mostrada en la figura 5.1.7, podemos distinguir cuatro espacios perfectamente diferenciados: dimensiones, seguridad, un croquis de la zapata con las cotas relevantes y un espacio para el anclaje.

En primer lugar, en el espacio para **dimensiones**, nos encontramos los siguientes elementos, reflejados en la figura 5.1.8.:

- **a_2 [m]**: ancho total de la zapata, en metros y múltiplos de 0,25 m.
Vemos que para estos datos el resultado es un ancho de 3 m.
- **h [m]**: canto de la zapata, en metros y múltiplos de 0.1.
Vemos que para estos datos el resultado es 0.6 m.
- **d [m]**: canto útil de la zapata, en metros.
Vemos que para estos datos el resultado es 0.554 m.
- **v [m]**: vuelo de la zapata, en metros. En caso de ser descentrada será el vuelo máximo, como se verá en otro ejemplo más adelante.
Su valor es de 1.38 m, que será el mismo a los dos lados por tratarse de un problema centrado.

- **Tipo de zapata:** nos dirá si es FLEXIBLE o RÍGIDA, para saber así que cálculos ha realizado.
- **Amín de Acero:** área mínima de acero a disponer, dada por la cuantía mínima del acero seleccionado.
- **Armadura principal:** nos muestra los datos de la armadura principal que ha calculado, que son tres:
 - **Área teórica:** es el área necesaria calculada, en cm^2 .
 - **Área real:** será mayor que la teórica, porque será la que se introduzca realmente con los redondos.
 - **Ø1:** redondos de esta armadura, con formato nº de redondos+ Ø + diámetro de los redondos.
- **Armadura de reparto:** datos de la armadura de reparto, que son los mismos que la principal:
 - **Área teórica:** es el área necesaria calculada, en cm^2 .
 - **Área real:** será mayor que la teórica, porque será la que se introduzca realmente con los redondos.
 - **Ø2:** redondos de esta armadura, con formato nº de redondos+ Ø + diámetro de los redondos.
- **Armadura en espera:** nos muestra las longitudes necesarias para la unión zapata muro, correspondientes a un solo diámetro de la armadura del muro.

Estas longitudes son, como hemos dicho, las necesarias. De ser estas superior a las admisibles por las dimensiones de la zapata, como en el caso de la longitud L2 que puede no caber en el ancho calculado, siempre se pueden unir varios redondos, por lo que se ha resuelto no recalcular en caso necesario y dejar esto a elección del usuario.

Dimensiones:			
a ₂ [m]:	3.0 m		
h [m]:	0.6 m		
d [m]:	0.554 m		
v [m]:	1.38 m		
Tipo de Zapata:	FLEXIBLE		
Amín de Acero:	6.0 cm^2		
Armadura principal :			
Área Teórica:	22.75 cm^2		
Area Real:	23.75 cm^2		
Ø1:	21Ø12.0		
Armadura de reparto :	Armadura en espera :		
Área Teórica:	6.82 cm^2	L1	0.56 m
Area Real:	7.07 cm^2	L2	0.267 m
Ø2:	9Ø10.0	L3	0.3 m

Figura 5.1.8-Pantalla donde se muestran las dimensiones de resultados

La mayoría de estos datos se pueden ver representados en el croquis que se sitúa a su derecha, como se ve en la figura 5.1.7.

Antes de continuar, comentar que los resultados de áreas y redondos vienen dados por unidad de longitud b₂, por eso es recomendable meter esta como un metro, por simplicidad.

Otro de los espacios de resultados mostrados por pantalla es el de **seguridad**, en el que podemos ver los coeficientes de seguridad calculados para verificar que la zapata esta dimensionada de acuerdo a la norma, representado en la figura 5.1.19 Estos coeficientes se han calculado de acuerdo a lo descrito en el apartado referente a los cálculos, que por utilizar el criterio de coeficientes parciales descrito en el CTE, estos serán suficientes si son mayores o iguales a la unidad. Destaca la comprobación a cortante definitiva, que se realiza con el área real

que se va a disponer, puesto que la anterior se ha hecho con estimaciones, y también la fisuración, que solo se realiza y muestra de ser la zapata de tipo flexible.

Seguridad	
Hundimiento:	1.02
Cortante :	1.33
Rígidez Relativa Suelo-Cimiento:	2.58
Vuelco:	15.81
Deslizamiento:	8.75
Cortante Definitiva	1.45
Fisuración :	1.5

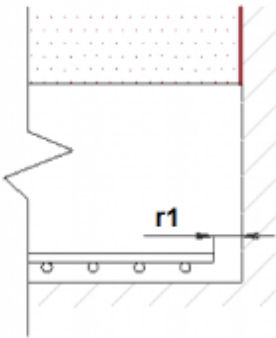
Figura 5.1.9-Pantalla donde se muestran las dimensiones de resultados

Para terminar con esta pantalla y este ejemplo, está la parte para el anclaje, que nos muestra un esquema de este y los siguientes datos:

- **Tipo de anclaje:** que pueden ser los vistos en el apartado correspondiente sobre su teoría.
- **lbAnclaje:** se trata de la longitud básica de anclaje, según lo dispuesto en la norma.
- **InAnclaje:** longitud neta de anclaje.
- **IDisponible:** longitud disponible para el anclaje.
- **IProlongaciónGancho:** longitud de la prolongación del gancho, en caso de ser necesaria.

Podemos ver como estos datos se muestran en pantalla en la figura 5.1.10. También

Anclaje :



ANLCAJE EN PROLONGACIÓN RECTA

lbAnclaje :	0.24 m
InAnclaje :	0.23 m
IDisponible :	0.3 m
IProlongacionGancho	0.0 m

Figura 5.1.10-Pantalla donde se muestra el anclaje de la armadura

vemos como se mostrarían por pantalla los diferentes tipos de anclaje en la figura 5.1.11, puesto que en los ejemplos realizados siempre es el de prolongación recta.

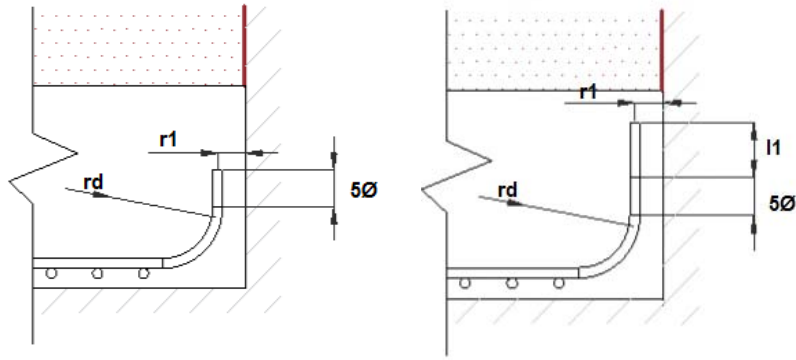


Figura 5.1.11-anclaje en gancho y con gancho prolongado

Con esto vemos los datos que arroja el programa cuando utilizamos la aplicación para realizar el dimensionado óptimo, ahora vamos a ver cómo usar el módulo de comprobación de dimensionado.

Para esto vamos a partir de los mismos datos y de los resultados anteriores, para ver soluciones próximas a la anterior y comprobar qué es lo que nos muestra la aplicación.

Lo primero es introducir unas dimensiones a comprobar:

Dimensiones :

a_2 [m]: 3

h [m]: 0.7

GUARDAR

Figura 5.1.12-dimensiones a comprobar para el primer ejemplo

Como vemos en la figura 5.1.12 los datos para la primera comprobación son $a_2=3$ metros y $h=0.7$ metros. Los resultados son mostrados en la figura 5.1.13.

Como vemos la pantalla que muestra los resultados es la misma para los dos módulos.

Analizando los nuevos resultados, vemos que al haber añadido 10 centímetros al canto tenemos una zapata rígida, con un área de acero a disponer para la armadura principal menor, y que apenas cumple las condiciones de seguridad.

Además, en este caso al tratarse de una zapata rígida no hace la comprobación de fisuración.

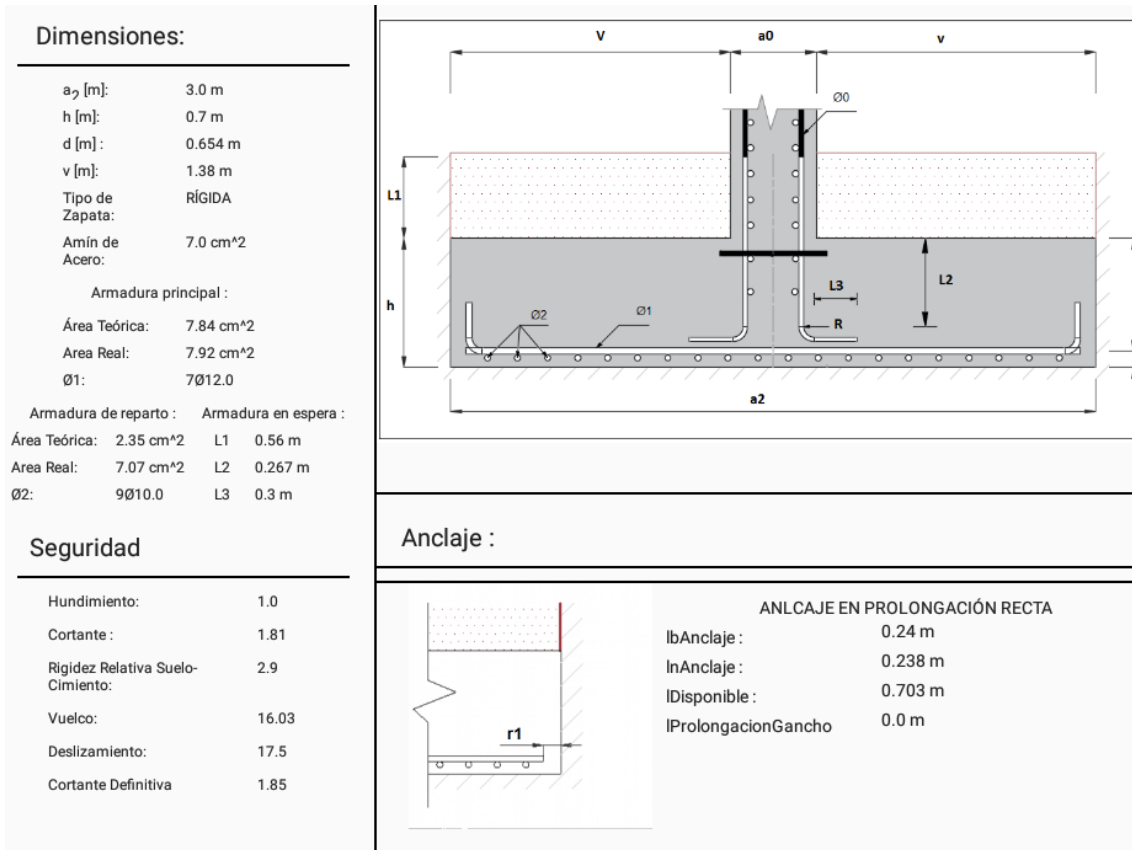


Figura 5.1.13-dimensiones a comprobar para el primer ejemplo

Otro ejemplo interesante es reduciendo el canto y ver qué ocurre, así que ahora vamos a introducir los siguientes datos, que vemos en la figura 5.1.14, que son $a_2=3$ metros y $h=0.5$ metros.

Dimensiones :	
a_2 [m]:	3
h [m]:	0.5
<input type="button" value="GUARDAR"/>	

Figura 5.1.14-dimensiones a comprobar para el segundo ejemplo

Si ahora vemos los resultados que obtenemos con estos datos, que aparecen en la figura 5.1.15, vemos que al reducir el canto hemos aumentado el área de acero a disponer, y que además ahora no cumple todos los criterios de seguridad, por lo que no es una solución válida.

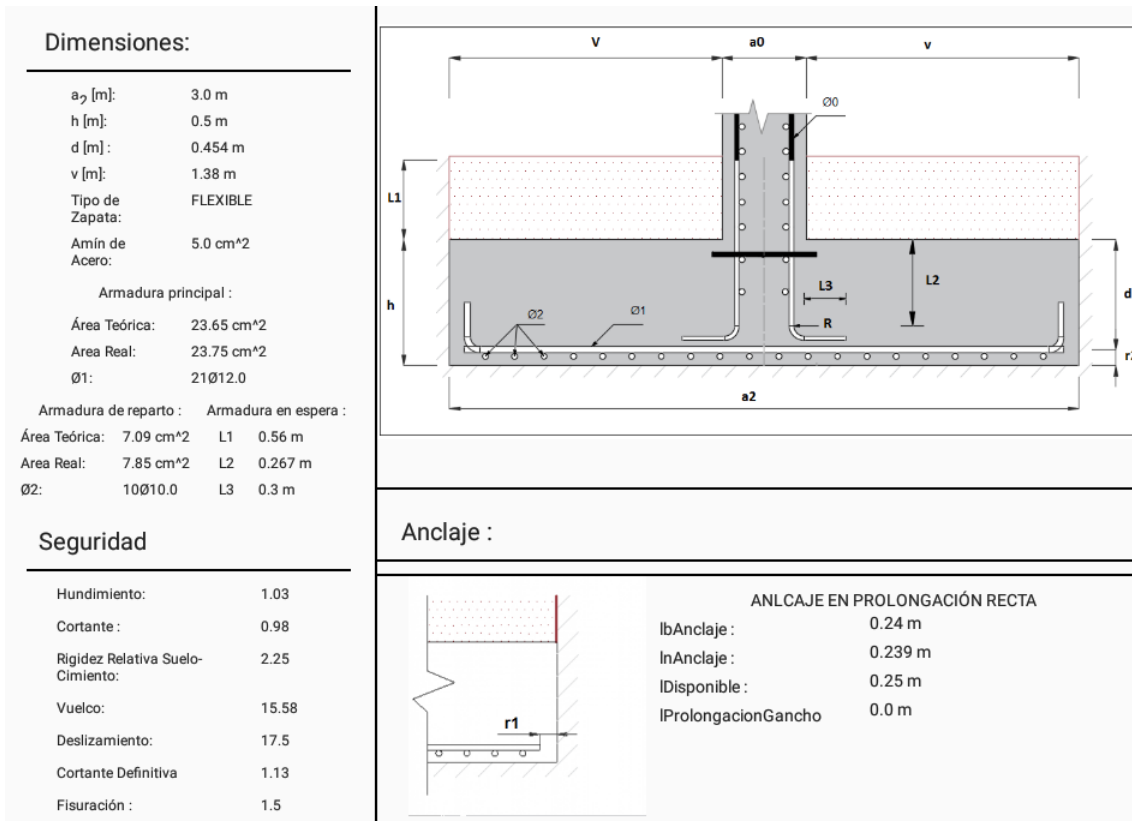


Figura 5.1.15-solución del segundo ejemplo

Con estos 3 ejemplos vemos qué se puede hacer con los módulos de dimensionado óptimo y comprobación de dimensiones, que se completa viendo cómo se modifica el croquis de la zapata si introducimos excentricidad, representados en la figura 5.1.16.

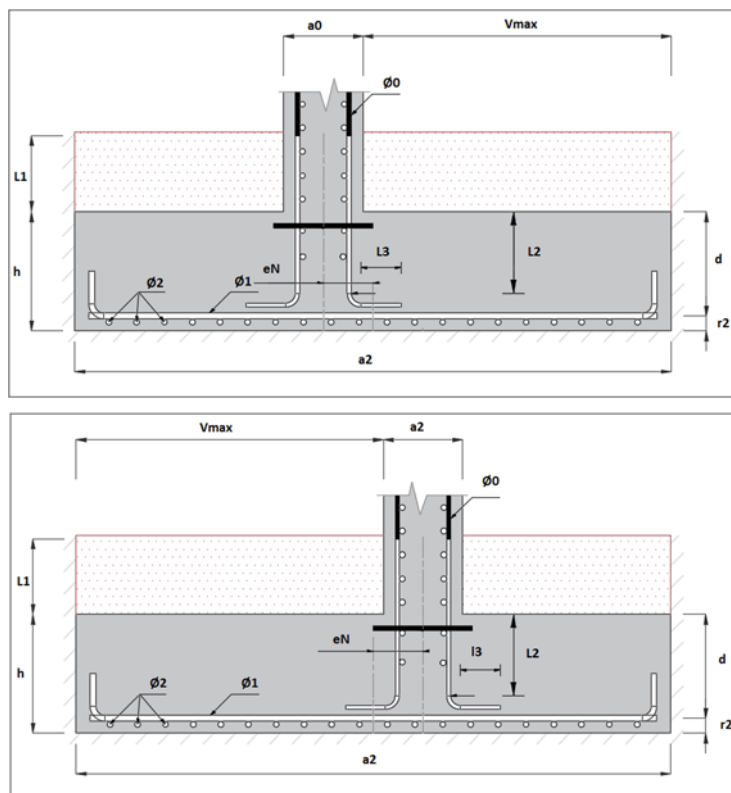


Figura 5.1.16-Croquis de la zapata con muro excéntrico

Con los datos preparados solamente falta introducir los valores de los esfuerzos a calcular:

- **Ejemplo 1 armado en rotura:**

Esfuerzos :	
Axil	0
Flector	190000

Nd=0 N.
Md=190000N

Vamos a la pantalla de resultados para estos esfuerzos.

Figura 5.1.18-Datos del ejemplo 1

Armadura 1	Armadura 1
	μ_d : 0.172
	ν_d : 0.0
	μ_{d1} : 0.172
	$\mu_{u1, c}^{Optimo}$: 0.3229
	Dominio : 3 sin A ₂
	ξ : 0.2833
	ξ_{Optima} : 0.6394
	$\Psi(\xi)$: 0.6881
	$\lambda(\xi)$: 0.416
	w_1 : 0.1949
	w_2 : 0.0
	A ₁ : 13.173
	A ₂ : 0.0
	n ₁ : 5.0φ20
	n ₂ : 0.0
	d _{s1} : 4.0
	d _{s2} : 0.0
Límites	
μ_d :	$\mu_{d1} \leq \mu_{u1, c}^{Optimo} = 0.3229$
ν_d :	$\nu_d \leq (1/420)^{289} \xi = 0.1949$
-	
ξ :	$1/6 \leq \xi \leq \xi_{Optimo} = 0.6394$

Figura 5.1.19-Resultados del ejemplo 1 de armado en rotura.

Viendo la pantalla de resultados distinguimos 3 partes: arriba a la izquierda, donde tenemos una lista deslizable con las distintas soluciones, como ya se comentó en el apartado 4, abajo a la derecha donde aparecen los límites de los esfuerzos reducidos y de la profundidad de la línea neutra correspondientes a lo visto en la teoría sobre estos cálculos y finalmente a la derecha un listado con una serie de datos.

Estos datos que aparecen a la derecha son los siguientes:

- μ_d : momento reducido.
- ν_d : axil reducido.
- μ_{d1} : momento reducido respecto a la armadura de tracción.
- $\mu_{u1, c}^{Optimo}$: momento reducido óptimo que el hormigón es capaz de resistir, que es el del límite del dominio 3.
- ξ : profundidad de la línea neutra reducida.

- ξ_{Optima} : profundidad de la línea neutra reducida óptima.
- $\Psi(\xi)$ y $\lambda(\xi)$: parámetros adimensionales.
- ω_1 y ω_2 : cuantías mecánicas de la armadura de tracción y de la de compresión respectivamente.
- A_1 y A_2 : áreas de acero a disponer para las armaduras de tracción y compresión.
- n_1 y n_2 : número y diámetro de los redondos de cada una de las armaduras, en el mismo formato que los casos anteriores.
- d_{s1} y d_{s2} : distancia entre los redondos de las armaduras.

Sobre los resultados vemos que se arma en el dominio 3 sin armadura de compresión, con cinco redondos de 20 milímetros de diámetro.

➤ **Ejemplo 2 armado en rotura:**

Para terminar este apartado se va a ver otro ejemplo, que es el siguiente:

Esfuerzos :		Nd=1750000 N.
Axil	1750000	Md=80000N
Flector	80000	Vamos a la pantalla de resultados para estos esfuerzos.

Figura 5.1.20-Datos del ejemplo 2

	Armadura 1	Armadura 2
	μ_d :	0.0724
	ν_d :	0.7445
	μ_{d1} :	0.4209
	$\mu_{u1, \text{Optimo}}^c$:	0.3229
	Dominio :	4.a sin A_1 y con A_2
	ξ :	1.0618
	ξ_{Optima} :	0.6394
	$\Psi(\xi)$:	0.6881
	$\lambda(\xi)$:	0.416
	w_1 :	0.0
	w_2 :	0.0139
	A_1 :	0.0
	A_2 :	0.9398
	n_1 :	0.0
	n_2 :	2.0 ϕ 20
	d_{s1} :	0.0
	d_{s2} :	22.0
Límites		
μ_d :	$\mu_{d1} \geq \mu_{u1, \text{Optimo}}^c (\xi^{4-4.a}) = 0.3319$	
ν_d :	$\nu_d \geq \xi^{4-4.a} \cdot \psi + (1-\delta_2)^{-1} \cdot (\mu_{d1} - \mu_{u1, \text{Optimo}}^c (\xi^{4-4.a})) = 0.70$	
	$\nu_d \leq \nu_u^c (\xi^{4.a-5}) + (1-\delta_2)^{-1} \cdot (\mu_{d1} - \mu_{u1, \text{Optimo}}^c (\xi^{4.a-5})) = 0$	
ξ :	$1 \leq \xi \leq \delta = 1.0638$	

Figura 5.1.20- Resultados del ejemplo 2 de armado en rotura.

En este caso vemos que lo arma en el dominio 4.a sin la armadura de tracción.

CONCLUSIONES:

Recordando los objetivos marcados en el apartado 1.2., el objetivo del trabajo era el de llegar a tener una aplicación que nos permita, a través de unos datos de entrada, obtener el dimensionado óptimo de una zapata continua o corrida de acuerdo lo que la norma vigente determina.

Con respecto a esto, el trabajo desarrollado ha conseguido lo previsto, el tener una aplicación portable que permite lo que se definió al comienzo del mismo, el dimensionado de la zapata continua.

Además, se ha conseguido, reutilizando buena parte del código necesario para el dimensionamiento, ofrecer también la opción de realizar las comprobaciones pertinentes a una zapata ya dimensionada, determinando si esta es o no apta según los criterios que se definen en las diferentes normas.

También se ha conseguido la función extra de dimensionar el caso de una cimentación flexible, arrojando datos más extensos de la misma ya que el caso de la zapata corrida se restringe al uso exclusivo de la armadura de tracción, y este módulo nos permite el dimensionado en los dominios con ambas armaduras.

Para concluir este apartado, los objetivos secundarios de aprendizaje han sido también cumplidos, aportando este trabajo al alumno los conocimientos complementarios esperados sobre programación Java y Android y sobre el cálculo de estructuras de hormigón, así como también otros como el manejo y consulta de normativas técnicas.

BIBLIOGRAFÍA:

- Instrucción de Hormigón Estructural [EHE] (2008)
- Código Técnico de la Edificación [CTE] (2006)
- Foces, A. (2005), Apuntes de Cálculos de Cimentaciones, Valladolid, Escuela de Ingenierías Industriales.
- Foces, A. (2005), Apuntes de Cálculo de estructuras de hormigón armado en rotura, Valladolid, Escuela de Ingenierías Industriales.
- Calaver, J. (2004). Cálculo de estructuras de cimentación, 4ta Edición, España: INFOPRINT S.A.
- Ribas, J. (2014). Desarrollo De Aplicaciones Para Android - Edición 2015, España: ANAYA
- Invarato, R. (2015). Android 100%. Recuperado de: <http://jarroba.com/libro-android-100-gratis/>

ANEXO: CÓDIGO.

Índice

- Armado.class	5
-ArmamadoNoSimetrico.class:.....	7
-ArmadoNoSimetrico2.class.....	12
-Comprobaciones.class	17
-ComprobacionesConDimensiones.class	35
-DatosAcero.class	52
-DatosArmadura.class	53
-DatosDimensiones.class.....	55
-DatosEsfuerzos.class	56
-DatosEsfuerzosNominales.class.....	57
- DatosGeometria.class	58
-DatosHormigon.class	60
-DatosMateriales.class	61
-DatosSuelo.class.....	62
-DinamicoFragmentA.class.....	64
-DinamicoFragmentB.class.....	68
-DinamicoFragmentCalculoComprobaciones.class	73
-DinamicofragmentCalculoDimensionado.class	78
-DinamicoFragmentComprobaciones.class.....	83
-DinamicoFragmentDatosDim.class	88
-DinamicoFragmentGrafico.class	91
-DinamicoFragmentNuevaArmadura.class	95
-DinamicoFragmentNuevosDatosDim.class	97
-DinamicoFragmentNuevosMateriales.class.....	107
-DinamicoFragmentPrincipal.class	108
-EstaticoFragment.class	109
-FragmentoPadre.class.....	111
-IFuncion.class	111
-MainActivity.class	112
-NewtonRapson.class	116
-Redondos.class.....	117
-Redondos2.class.....	119
-UnaFuncion.class	120

-strings.XML:.....	121
-activity-main.XML:	124
-dinamico_fragment_calculo_comprobaciones.XML:	124
-dinamico_fragment_calculos_dimensionado.XML:	136
-dinamico_fragment_comprobacion.XML:	147
-dinamico_fragment_dimensionado.XML:	151
-dinamico_fragment_grafico.XML:	154
-dinamico_fragment_nuevosdatosdim.XML:.....	155
-dinamico_fragment_principal.XML:	170
-dinamico_fragment_a.XML:	171
-dinamico_fragment_b.XML:	174
-dinamico_fragment_nuevaarmadura.XML:.....	181
- fragment_estatico.XML:.....	185
-fragmen_dinamico_nuevomaterial.XML:	187

-Armado.class

```
public class Armado {
    public Armado () {

    }

    public static double psi=0, lmb=0;
    //valores límite:
    public static double limNu=0;

    int i=0;//contador

    //Método para escribir solo 4 decimales:
    public static double red(double n, int i){
        int p=(int) Math.pow(10,i);
        return Math rint(n*p)/p;
    }
    public static double[] resolver(int a, int b, double c){
        double r= Math.sqrt(b*b-4*a*c);
        double s1= (double) ((-1)*b+r)/(2*a);
        double s2= (double) ((-1)*b-r)/(2*a);
        double [] s= {red(s1,4), red(s2,4)};
        return s;
    }
    public static double[] resolver(double mul, double nu, double
        delta2){
        double a=-0.6881*0.416;
        double bb=delta2*0.6881;
        double c=nu*(1-delta2)-mul;
        double r= Math.sqrt(bb*bb-4*a*c);
        double s1= (double) ((-1)*bb+r)/(2*a);
        double s2= (double) ((-1)*bb-r)/(2*a);
        double [] s= {red(s1,4), red(s2,4)};
        return s;
    }
    public static double getPlOptimaMaxima(double ey){
        double plOptimaMaxima=0.0035/(0.0035+ey);//Pln optima máxima
        return plOptimaMaxima;
    }
    public static double getPlOptima(double delta2, double ey){
        double plOptima=0.60101*(1+delta2);//Pln optima máxima
        if(getPlOptimaMaxima(ey)<plOptima){
            plOptima=getPlOptimaMaxima(ey);
        }
        return plOptima;
    }
    public static double getMuOptimo(double plOptima, double ey){
        //Ventan.plOptima.setText(String.valueOf(red(plOptima,4)));
        return getMu(plOptima, 3);
        //return plOptima*getPsi()* (1-plOptima*getLmb());//El óptimo siempre es el valor
frontera
    }

    public static double getPsi(double pln, int i){
        if(i==1){
            psi=1.416666667*((3*pln-8*pln*pln)/((1-pln)*(1-pln)));
        }
        if(i==2){
            psi=0.056666667*((16*pln-1)/pln);
        }
        else{
            psi=0.6881;
        }
        return psi;
    }
    public static double getPsi(double pln, double delta){
        psi=0.0425*(1029*pln*pln-882*pln*delta+125*delta*delta)/((3*delta-
7*pln)*(3*delta-7*pln));
        return psi;
    }
    public static double getPsi(){
        psi=0.6881;
        return psi;
    }
}
```

```

public static double getLmb(double pln, int i){
    if(i==1){
        lmb=((4-9*pln)/(12-32*pln));
    }
    if(i==2){
        lmb=(1-22*pln+171*pln*pln)*(20*pln*(16*pln-1));
    }
    else{
        lmb=0.416;
    }
    return lmb;
}
public static double getLmb(double pln, double delta){
    lmb=0.2142857*(2041*pln*pln-2058*pln*delta+185*delta*delta)/(1029*pln*pln-
882*pln*delta+125*delta*delta);
    return lmb;
}
public static double getLmb(){
    lmb=0.416;
    return lmb;
}
public static double getNu(double plnLim, int i){
    return plnLim*getPsi(plnLim, i);
}
public static double getNu(double plnLim){
    return plnLim*getPsi(plnLim, 0);
}
/*Lo mismo para el momento reducido del hormigon*/
public static double getMu(double plnLim, int i){
    return getNu(plnLim, i)*(1-plnLim*getLmb(plnLim,i));
}
}

public static double getLimiteNu(double mul,double delta2, double
delta,double ey, int i){
    if (i==4){//NO SE MUY BIEN, ;PREGUNTAR!
        limNu=getNu(getPLOptima(delta2,ey))+((mul-
getMu(getPLOptima(delta2,ey), i))/(1-delta2));
    }
    if (i==5){
        limNu=getNu(getPLOptimaMaxima(ey))+((mul-
getMu(getPLOptimaMaxima(ey), i))/(1-delta2));
    }
    if (i==6){
        limNu=getNu(1)+((mul-getMu(1, i))/(1-delta2));//valor de
pln de esta frontera es 1
    }
    if (i==7){
        limNu=getNu(delta)+((mul-getMu(delta, i))/(1-delta2));
    }
    if (i==8){
        limNu=0.85*delta+((mul-(0.85*delta*(1-delta/2)))/(1-
delta2));
    }
    return limNu;
}
}
public static double getLimiteNu(double pln, int i){
    if(i==1){
        limNu=pln*pln*(1.416667*((3-8*pln)/((1-pln)*(1-pln))));
    }
    if (i==2){
        limNu=((272*pln-17)/300);
    }
    if (i==3){
        limNu=((289*pln)/420);
    }
    return limNu;
}
}
public static double getSigma1(double ey, double fyd, double pln, double delta){
    return 0.002*fyd*(pln-1)/((pln-(3*delta)/7)*ey);
}
}
}

```

-ArmamadoNoSimetrico.class:

```
public class ArmadoNoSimetrico extends Armado {

    private static String nombre;

    private static int contadorDominio;

    private static double delta;
    private static double delta2;
    private static double mu;
    private static double nu;
    private static double mu1;
    private static double w1;
    private static double w2;
    private static double a1;
    private static double a2;
    private static double pln;
    private static double plnOptima;
    private static double plnOptimaMaxima;
    private static double psi;
    private static double lmb;
    private static double muOptimo;
    //armadura={ey,delta,delta2,mu,nu,mu1,contadorDominio,pln,w1,w2,A1,A2}
    public ArmadoNoSimetrico() {

    }

    public static void setNombre(int cont){
        nombre="Armadura "+String.valueOf(cont);
    }

    public static ArmadoNoSimetrico CalculoArmadura(DatosArmadura
        dArma, DatosMateriales dMat, DatosEsfuerzos dEsf) {

        ArmadoNoSimetrico arm=new ArmadoNoSimetrico();

        nombre="Armadura ";

        pln = 0; //Profundidad línea neutra.

        contadorDominio = 0;//arcador para delimitar dominio

        w1 = 0; w2 = 0;//Cuantías mecánicas de las armaduras
        double[] s;//Vector para almacenar soluciones de ecuación
            grado<1

        //dArma={b, h, dia1, dia2, r1, r2, d1, d2, dx};
        double b = dArma.getB();
        double h = dArma.getH();
        double dia1 = dArma.getDial();
        double dia2 = dArma.getDia2();
        double r1 = dArma.getR1();
        double r2 = dArma.getR2();
        double d1 = dArma.getD1();
        double d2 = dArma.getD2();
        double dx = dArma.getDx();

        //dMat={fyd, fcd, es, ey};
        double fyd = dMat.getFyd();
        double fcd = dMat.getFcd();
        double es = dMat.getEs();
        double ey = dMat.getEy();

        //dEsf={md,nd};
        double md = dEsf.getMd();
        double nd = dEsf.getNd();

        //Calculo cantos reducidos:
        delta = h / dx;
        delta2 = d2 / dx;
        //Calculo esfuerzos reducidos:
        mu = md / (b * 1e-2 * dx * 1e-2 * dx * 1e-2 * fcd * 1e6);
```

```

nu = nd / (b * 1e-2 * dx * 1e-2 * fcd * 1e6);
mu1 = mu + nu * (1 - delta / 2);
//Calculo momento reducido resistido por el hormigón en la
    situación óptima:
muOptimo = getMuOptimo(getPlOptima(delta2, ey), ey);

//Calculo los valores límite del axil reducido que necesito:
double nuLim4 = getLimiteNu(mu1, delta2, delta, ey, 4);

double nuLim5 = getLimiteNu(mu1, delta2, delta, ey, 5);

double nuLim6 = getLimiteNu(mu1, delta2, delta, ey, 6);

double nulin7 = getLimiteNu(mu1, delta2, delta, ey, 7);

double nuLim8 = getLimiteNu(mu1, delta2, delta, ey, 8); //es el
                                                    máximo

//Con esto ya puedo delimitar el dominio
if (mu1 <= muOptimo) {
    w2 = 0;
    if (mu1 <= 0.08854) { //dominio 2.a sin A1
        contadorDominio = 1;
        /*Como en este dominio mu1 me viene definido por un
            polinomio mayor que 2, lo resuelvo numéricamente por
            Newton Raphson*/
        IFuncion f = new UnaFuncion() {
            @Override
            public double eval(double x) {
                return 1.0625 * (((4 - 12 * x + 3 * x * x) * x * x) / ((1 - x) *
(1 - x)))
                    - mu1;
            }
        };
        IFuncion df = new UnaFuncion() {
            @Override
            public double eval(double x) {
                return 1.0625 * ((6 * x - 12) * x * x) / ((1 - x) * (1 - x))
                    + 2.125 * ((3 * x * x - 12 * x + 4) * x) / ((1 - x) * (1
- x))
                    + 2.125 * ((3 * x * x - 12 * x + 4)) / ((1 - x) * (1 -
x) * (1 - x));
            }
        };
        NewtonRapson nr = new NewtonRapson();
        pln = nr.raiz(f, df, 0.0833333, 1e-4, 1000);
        if (pln < 0.16666666667 && pln >= 0) {
            w1 = w2 + pln * getPsi(pln, contadorDominio) - nu;
        } else {
        }
    }
}
if (mu1 > 0.08854 && mu1 <= 0.1592) { //Dominio 2.b sin A2
    contadorDominio = 2;

    s = resolver(57, -114, (117.6470588 * mu1 + 7));
    if (s[0] < 0.259259 && s[0] >= 0.16666667) {
        pln = s[0];
        w1 = w2 + pln * getPsi(pln, contadorDominio) - nu;
    }
    if (s[1] < 0.259259 && s[1] >= 0.16666667) {
        pln = s[1];
        w1 = w2 + pln * getPsi(pln, contadorDominio) - nu;
    } else { //masl asunto
    }
}
if (mu1 > 0.1592) { //Dominio 3 sin A2

    //este dominio:
    contadorDominio = 3;
    s = resolver(99, -238, (345.8823529 * mu1));
    if (s[0] <= getPlOptima(delta2, ey) && s[0] >= 0.259259) {
        pln = s[0];
        w1 = w2 + pln * getPsi(pln, contadorDominio) - nu;
    }
    if (s[1] <= getPlOptima(delta2, ey) && s[1] >= 0.259259) {

```



```

        pln = s[1];
        w1 = w2 + pln * getPsi(pln, contadorDominio) - nu;
    } else {
    }
}
}
}
if (mul > muOptimo) {
    if (nu <= nuLim4) { //dominio 3 con A2
        contadorDominio = 4;
        pln = getP1Optima(delta2, ey);
        w2 = ((mul - muOptimo) / (1 - delta2));
        w1 = w2 + pln * getPsi() - nu;
    }
    if (nu > nuLim4) { //dominios 3, 4 y 4.a sin A1 y con A2
        w1 = 0;
        if (nu <= nuLim5) { //3 sin A1
            contadorDominio = 5;
            s = resolver(mul, nu, delta2);
            if (s[0] <= getP1Optima(delta2, ey) && s[0] >= 0.259259) {
                pln = s[0];
            }
            if (s[1] <= getP1Optima(delta2, ey) && s[1] >= 0.259259) {
                pln = s[1];
            } else {
            }
        }
        if (nu > nuLim5 && nu <= nuLim6) { //4 sin A1
            contadorDominio = 6;
            s = resolver(mul, nu, delta2);
            if (s[0] > getP1Optima(delta2, ey) && s[0] <= 1) {
                pln = s[0];
            }
            if (s[1] > getP1Optima(delta2, ey) && s[1] <= 1) {
                pln = s[1];
            } else {
            }
        }
        if (nu > nuLim6 && nu <= nulum7) { //4.a sin A1
            contadorDominio = 7;
            s = resolver(mul, nu, delta2);
            if (s[0] > 1 && s[0] <= delta) {
                pln = s[0];
            }
            if (s[1] > 1 && s[1] <= delta) {
                pln = s[1];
            } else {
            }
        }
        if (nu > nulum7 && nu < 0.85 * delta + ((mul -
            0.85 * delta * (1 - delta / 2)) / (1 - delta2)))
        { //5 sin A1, To chungo, lo hago luego
            contadorDominio = 8;
            IIIIIFuncion f = new CincoFuncion() {
                @Override
                public double eval(double x) {
                    return ((nu-17*(delta/400))*(125*delta*delta-
882*delta*x+1029*x*x)/
                    (3*delta-7*x)*(3*delta-7*x))* (1-
delta2)+17*(delta/400)*
                    (125*delta*delta-882*delta*x+1029*x*x)* (1-
3*(delta/14))*
                    (185*delta*delta-
2058*delta*x+2041*x*x)/(125*delta*delta-
882*delta*x+1029*x*x))/(3*delta-7*x)*(3*delta-7*x))-
mul;
                }
            };
            IIIIIFuncion df = new CincoFuncion() {
                @Override
                public double eval(double x) {
                    return -(17)*((784*delta2-672*delta-
405*x)*delta*delta*delta)/
                    (350*(3*delta-7*x)*(3*delta-7*x)*(3*delta-7*x));
                }
            };

```

```

    }
    };
    NewtonRapson2 nr = new NewtonRapson2();
    pln = nr.raiz(f, df, delta, 1e-4, 1000);
}
w2 = nu - pln * getPsi();
}
}
//Dominio 5 con A1 y A2. Compresión simple:
if (mu1 >= 0.85 * delta * (1 - delta / 2) && nu >= 0.85 * delta + ((mu1 -
    0.85 * delta * (1 - delta / 2)) / (1 - delta2))) {
    contadorDominio = 9;
    w2 = (mu1 - 0.85 * delta * (1 - delta / 2)) / (1 - delta2);
    w1 = nu - w2 - 0.85 * delta;
}
//Dimensionamiento en tracción:
if (mu1 < 0) {
    pln = (ey - 0.01 * delta2) / (ey - 0.01);
    w2 = Math.abs(mu1) / (1 - delta2);
    w1 = w2 - nu;
}

//Con esto calculo las áreas:
a1=((w1 * dx * b * fcd) / fyd);
a2=((w2 * dx * b * fcd) / fyd);
plnOptima=(getPLOptima(delta2, ey));
psi=(getPsi(pln, contadorDominio));
lmb=(getLmb(pln, contadorDominio));
plnOptimaMaxima=(getPLOptimaMaxima(ey));

return arm;
}

public double getDelta() {
    return delta;
}

public int getContadorDominio() {
    return contadorDominio;
}

public double getPsi2() {
    return psi;
}

public double getLmb2() {
    return lmb;
}

public double getMuOptimo2() {
    return muOptimo;
}

public double getPlnOptima2() {
    return plnOptima;
}

public double getPlnOptimaMaxima() {
    return plnOptimaMaxima;
}

public double getDelta2() {
    return delta2;
}

public double getMu() {
    return mu;
}

public double getNu() {
    return nu;
}

public double getMu1() {
    return mu1;
}
}

```

```
public double getW1() {  
    return w1;  
}  
  
public double getW2() {  
    return w2;  
}  
  
public double getA1() {  
    return a1;  
}  
  
public double getA2() {  
    return a2;  
}  
  
public double getPln() {  
    return pln;  
}  
  
public String getNombre() {  
    return nombre;  
}  
}
```

-ArmadoNoSimetrico2.class

```
public class ArmadoNoSimetrico2 extends Armado2 {

    private String nombre;

    private int contadorDominio;

    private double delta;
    private double delta2;
    private double mu;
    private double nu;
    private double mu1;
    private double w1;
    private double w2;
    private double a1;
    private double a2;
    private double d;
    private double pln;
    private double plnOptima;
    private double plnOptimaMaxima;
    private double psi;
    private double lmb;
    private double muOptimo;
    private double dia1=0;

    //armadura={ey,delta,delta2,mu,nu,mu1,contadorDominio,pln,w1,w2,A1,A2}
    public ArmadoNoSimetrico2() {

    }

    public void setNombre(int cont){
        nombre="Armadura "+String.valueOf(cont);
    }

    public static ArmadoNoSimetrico2 CalculoArmadura(double b2, double canto,
                                                    double fyk, double fck,
                                                    double Gs, double Go,
                                                    double E, double Md, double
Nd, double r, double dia22) {

        ArmadoNoSimetrico2 arm=new ArmadoNoSimetrico2();

        //int pos=dEsf.getNombre().indexOf(" ");

        String Nombre="Armadura ";//+dEsf.getNombre().substring(pos+1);

        double pln = 0; //Profundidad línea neutra.

        int contadorDominio = 0;//arcador para delimitar dominio

        double w1 = 0, w2 = 0;//Cuantías mecánicas de las armaduras
        double[] s;//Vector para almacenar soluciones de ecuación grado<1

        //dArma={b, h, dia1, dia2, r1, r2, d1, d2, dx};
        double b = b2;
        double h = canto;
        double dia1 = 12;
        double dia2 = 12;
        double r1 = r*100;
        double r2 = 2;
        double d1 = 3;
        double d2 = 3;
        double dx = h-r1-dia1*0.5*0.1-dia22*0.1;

        //dMat={fyd, fcd, es, ey};
        double fyd = fyk/Gs;
        double fcd = fck/Go;
        double es = E;
        double ey = fyd/es;;
    }
}
```

```

//dEsf={md,nd};
double md = Md;
double nd = Nd;

//Calculo cantos reducidos:
final double delta = h / dx;
final double delta2 = d2 / dx;
//Calculo esfuerzos reducidos:
double mu = md / (b * 1e-2 * dx * 1e-2 * dx * 1e-2 * fcd * 1e6);
final double nu = nd / (b * 1e-2 * dx * 1e-2 * fcd * 1e6);
final double mul = mu + nu * (1 - delta / 2);
//Calculo momento reducido resistido por el hormigón en la itución óptima:
double muOptimo = getMuOptimo(getPlOptima(delta2, ey), ey);

//Calculo los valores límite del axil reducido que necesito:
double nuLim4 = getLimiteNu(mul, delta2, delta, ey, 4);

double nuLim5 = getLimiteNu(mul, delta2, delta, ey, 5);

double nuLim6 = getLimiteNu(mul, delta2, delta, ey, 6);

double nulin7 = getLimiteNu(mul, delta2, delta, ey, 7);

double nuLim8 = getLimiteNu(mul, delta2, delta, ey, 8); //es el máximo

//Con esto ya puedo delimitar el dominio
if (mul <= muOptimo) {
    w2 = 0;
    if (mul <= 0.08854) { //dominio 2.a sin A1
        contadorDominio = 1;

        IIIIIIFuncion f = new CincoFuncion() {
            @Override
            public double eval(double x) {
                return 1.0625 * (((4 - 12 * x + 3 * x * x) * x * x) / ((1 - x) *
(1 - x)))
                    - mul;
            }
        };
        IIIIIIFuncion df = new CincoFuncion() {
            @Override
            public double eval(double x) {
                return 1.0625 * ((6 * x - 12) * x * x) / ((1 - x) * (1 - x))
                    + 2.125 * ((3 * x * x - 12 * x + 4) * x) / ((1 - x) * (1
- x))
                    + 2.125 * ((3 * x * x - 12 * x + 4)) / ((1 - x) * (1 -
x) * (1 - x));
            }
        };
        NewtonRapson2 nr = new NewtonRapson2();
        pln = nr.raiz(f, df, 0.0833333, 1e-4, 1000);
        if (pln < 0.16666666667 && pln >= 0) {
            w1 = w2 + pln * getPsi(pln, contadorDominio) - nu;
        } else {
            //AlertBox.display("Error", "NewtonRapson 1");
        }
    }
    if (mul > 0.08854 && mul <= 0.1592) { //Dominio 2.b sin A2

        contadorDominio = 2;

        s = resolver(57, -114, (117.6470588 * mul + 7));
        if (s[0] < 0.259259 && s[0] >= 0.16666667) {
            pln = s[0];
            w1 = w2 + pln * getPsi(pln, contadorDominio) - nu;
        }
        if (s[1] < 0.259259 && s[1] >= 0.16666667) {
            pln = s[1];
            w1 = w2 + pln * getPsi(pln, contadorDominio) - nu;
        } else { //masl asunto
            //AlertBox.display("Error", "Ecuación 1");
        }
    }
    if (mul > 0.1592) { //Dominio 3 sin A2
        //Resuelvo la ecuacion de segundo grado con los términos de
        //este dominio:
        contadorDominio = 3;
    }
}

```

```

s = resolver(99, -238, (345.8823529 * mul));
if (s[0] <= getPlOptima(delta2, ey) && s[0] >= 0.259259) {
    pln = s[0];
    w1 = w2 + pln * getPsi(pln, contadorDominio) - nu;
}
if (s[1] <= getPlOptima(delta2, ey) && s[1] >= 0.259259) {
    pln = s[1];
    w1 = w2 + pln * getPsi(pln, contadorDominio) - nu;
} else {
}
}
}
if (mul > muOptimo) {
    if (nu <= nuLim4) { //dominio 3 con A2
        contadorDominio = 4;
        pln = getPlOptima(delta2, ey);
        w2 = ((mul - muOptimo) / (1 - delta2));
        w1 = w2 + pln * getPsi() - nu;
    }
    if (nu > nuLim4) { //dominios 3, 4 y 4.a sin A1 y con A2
        w1 = 0;
        if (nu <= nuLim5) { //3 sin A1
            contadorDominio = 5;
            s = resolver(mul, nu, delta2);
            if (s[0] <= getPlOptima(delta2, ey) && s[0] >= 0.259259) {
                pln = s[0];
            }
            if (s[1] <= getPlOptima(delta2, ey) && s[1] >= 0.259259) {
                pln = s[1];
            } else {
            }
        }
        if (nu > nuLim5 && nu <= nuLim6) { //4 sin A1
            contadorDominio = 6;
            s = resolver(mul, nu, delta2);
            if (s[0] > getPlOptima(delta2, ey) && s[0] <= 1) {
                pln = s[0];
            }
            if (s[1] > getPlOptima(delta2, ey) && s[1] <= 1) {
                pln = s[1];
            } else {
                //AlertBox.display("Error", "Ecuacion 4");
            }
        }
        if (nu > nuLim6 && nu <= nulim7) { //4.a sin A1
            contadorDominio = 7;
            s = resolver(mul, nu, delta2);
            if (s[0] > 1 && s[0] <= delta) {
                pln = s[0];
            }
            if (s[1] > 1 && s[1] <= delta) {
                pln = s[1];
            } else {
                //AlertBox.display("Error", "Ecuación 5");
            }
        }
    }
    if (nu > nulim7 && nu < 0.85 * delta + ((mul -
        0.85 * delta * (1 - delta / 2)) / (1 - delta2)))
    { //5 sin A1
        contadorDominio = 8;
        IIIIFuncion f = new CincoFuncion() {
            @Override
            public double eval(double x) {
                return ((nu-17*(delta/400)*(125*delta*delta-
882*delta*x+1029*x*x)/
                    (3*delta-7*x)*(3*delta-7*x))* (1-
delta2)+17*(delta/400)*
                    (125*delta*delta-882*delta*x+1029*x*x))* (1-
3*(delta/14)*
                    (185*delta*delta-
2058*delta*x+2041*x*x)/(125*delta*delta-
                    882*delta*x+1029*x*x))/(3*delta-7*x)*(3*delta-7*x))-
mul;
            }
        }
}
}

```

```

    };
    IIIIIIFuncion df = new CincoFuncion() {
        @Override
        public double eval(double x) {
            return -(17)*((784*delta2-672*delta-
405*x)*delta*delta*delta)/
                                (350*(3*delta-7*x)*(3*delta-7*x)*(3*delta-7*x));
        }
    };
    NewtonRapson2 nr = new NewtonRapson2();
    pln = nr.raiz(f, df, delta, 1e-4, 1000);
    //if(pln<delta) error
}
w2 = nu - pln * getPsi();
}
}
//Dominio 5 con A1 y A2. Compresión simple:
if (mul >= 0.85 * delta * (1 - delta / 2) && nu >= 0.85 * delta + ((mul -
0.85 * delta * (1 - delta / 2)) / (1 - delta2))) {
    contadorDominio = 9;
    w2 = (mul - 0.85 * delta * (1 - delta / 2)) / (1 - delta2);
    w1 = nu - w2 - 0.85 * delta;
}
//Dimensionamiento en tracción:
if (mul < 0) {
    pln = (ey - 0.01 * delta2) / (ey - 0.01);
    w2 = Math.abs(mul) / (1 - delta2);
    w1 = w2 - nu;
}

//Con esto calculo las áreas:
arm.a1=((w1 * dx * b * fcd) / fyd);
arm.a2=((w2 * dx * b * fcd) / fyd);
arm.plnOptima=(getPlOptima(delta2, ey));
arm.psi=(getPsi(pln, contadorDominio));
arm.lmb=(getLmb(pln, contadorDominio));
arm.plnOptimaMaxima=(getPlOptimaMaxima(ey));
arm.d=dx;
return arm;
}

public double getDelta() {
    return delta;
}

public double getD() {
    return d;
}

public int getContadorDominio() {
    return contadorDominio;
}

public double getPsi2() {
    return psi;
}

public double getLmb2() {
    return lmb;
}

public double getMuOptimo2() {
    return muOptimo;
}

public double getPlnOptima2() {
    return plnOptima;
}

public double getPlnOptimaMaxima() {
    return plnOptimaMaxima;
}

public double getDelta2() {
    return delta2;
}
}

```

```
public double getMu() {  
    return mu;  
}  
  
public double getNu() {  
    return nu;  
}  
  
public double getMu1() {  
    return mu1;  
}  
  
public double getW1() {  
    return w1;  
}  
  
public double getW2() {  
    return w2;  
}  
  
public double getA1() {  
    return a1;  
}  
  
public double getA2() {  
    return a2;  
}  
  
public double getPln() {  
    return pln;  
}  
  
public String getNombre() {  
    return nombre;  
}  
}
```

-Comprobaciones.class

```
public class Comprobaciones {

    //Variables:
    private double
        /*Esfuerzos Nominales*/
        N=0, //Axil [N]
        V=0, //Cortante [N]
        M=0, //Flector [N*m]

    /*Acero*/
    fyk=0, //límite elástico de proyecto de las armaduras pasivas [Pa]
    Es=0, //módulo de elasticidad del acero[Pa]
    Gs=0, //coef reductor

    /*Hormigon*/
    fck=0, //resist. de proyecto del hormigon a compresion[Pa]
    Gh=0, //peso específico aprox [N/m^3]
    Go=0, //coeficiente de minoracion
    Ec=0, //módulo de elasticidad medio del hormigon de la zapata[Pa]
    wkMax=0,

    /*Geometria*/
    r=0, //recubrimiento geometrico [m]
        r1=0,
        dia0=0,
        p=0, //tipo de muro
        en=0, //excentricidad del muro[m]
        b2=0, //ancho de calculo[m]
        a1=0, //espesor del muro[m]
        l=0, //longitud del muro[m]
        ht=0, //profundidad de la zapata[m]

    /*Suelo*/
    Sta=0, //presión máxima admisible(presion hundimiento del terreno*ancho
    seccion)[N/m]=[Pa*m]
        Gt=0, //peso terreno/m^3 [N/m^3]
        K03=0, //coeficiente balastro 30x30
        coh=0, //valor de claculo de la cohesión del terreno SIN MINORAR [N/m^2]=[Pa]
        Gd= 0, //coef seguridad fente a deslizamiento,
        Gv=0, //coef seguridad frentea vuelco
        del=0, //angulo de rozamiento entre esl suelo y la zapata
        Ga=0; //coeficiente seguridad cortante

    private boolean B400=true;
    private boolean B500=false;

    private boolean cohesivo=true;
    private double[] D=new double[20];

    private ArmadoNoSimetrico2[] armaduras=new ArmadoNoSimetrico2[20];
    private Redondos2[] redondos=new Redondos2[20];

    private double[] ancho=new double[20];
    private double[] canto=new double[20];
    private double[] e=new double[20];

    private double[] seguridadH=new double[20];
    private String[] estadoHundimiento=new String[20];

    private double[] cortante=new double[20];
    private double[] momentoDeCalculo=new double[20];

    private double[] cortanteH=new double[20];
    private double[] seguridadC=new double[20];
    private String[] estadoCortante=new String[20];

    private double[] vuelo=new double[20];
    private double[] vueloMax=new double[20];
    private double[] seguridadR=new double[20];
    private String[] estadoRigidez=new String[20];

    private double[] Mdesf=new double[20];
    private double[] Mfav=new double[20];
    private double[] seguridadV=new double[20];
    private String[] estadoVuelco=new String[20];
}
```

```

private double[] Vdesf=new double[20];
private double[] Vfav=new double[20];
private double[] seguridadD=new double[20];
private String[] estadoDeslizamiento=new String[20];

private String[] estadoCortanteDefinitivo=new String[20];

private double[] cortanteDefinitivo=new double[20];
private double[] cortanteHDefinitivo=new double[20];
private double[] seguridadCDefinitiva=new double[20];

private double T1=0,T2=0;

private double[] seguridadW=new double[20];
private double[] seguridadWW=new double[20];

private String[] tipoArmadura=new String[20];
private String estdoCortanteDefinitivo;
private String[] estdoAnclaje=new String[20];;
//private static String tipoArmadura;
private int cont=1;

private double[] diametro=new double[20];
private double[] diametroReparto=new double[20];
private final double diametros[]={12, 14, 16, 20, 25, 32, 40};
private final double diametros2[]={10,12, 14, 16, 20, 25, 32, 40};
private double[] lbAnclaje=new double[20];
private double[] lnAnclaje=new double[20];
private double[] longitudDisponible=new double[20];
private double[] l1=new double[20];
private double[] pln=new double[20];
private double[] dsl=new double[20];
private double[] dslW=new double[20];
private double anchoDef=0;
private double cantoDef=0;

private double[] areaReal=new double[20];
private double[] areaRealW=new double[20];
private double[] areaRealReparto=new double[20];
private double[] areaTeorica=new double[20];
private double[] areaTeoricaReparto=new double[20];
private double[] areaMinima=new double[20];
private double[] x1=new double[20];
private double[] x2=new double[20];
private int[] nRedondos=new int[20];
private int[] nRedondosW=new int[20];
private int[] nRedondosReparto=new int[20];

private double VdM=0;
private double MdM=0;

//Vectores longitudes basicas de anclaje: [m]
//B400:
private double[] lb400H25={0.24,0.28,0.32,0.48,0.75,1.230,1.92};
private double[] lb400H30={0.24,0.28,0.32,0.40,0.63,1.020,1.6};
private double[] lb400H35={0.24,0.28,0.32,0.40,0.56,0.92,1.44};
private double[] lb400H40={0.24,0.28,0.32,0.40,0.50,0.82,1.28};
private double[] lb400H45={0.24,0.28,0.32,0.40,0.50,0.72,1.12};
private double[] lb400H50={0.24,0.28,0.32,0.40,0.50,0.72,1.12};
//B500:
private double[] lb500H25={0.3,0.35,0.4,0.6,0.938,1.536,2.4};
private double[] lb500H30={0.3,0.35,0.4,0.52,0.813,1.331,2.08};
private double[] lb500H35={0.3,0.35,0.4,0.5,0.75,1.229,1.92};
private double[] lb500H40={0.3,0.35,0.4,0.5,0.688,1.126,1.76};
private double[] lb500H45={0.3,0.35,0.4,0.5,0.625,1.024,1.6};
private double[] lb500H50={0.3,0.35,0.4,0.5,0.625,1.024,1.6};

private int[] marcadorValidos=new int[20];
private int contMarcVald=0;

private double L1es=0,L2es=0,L3es=0;

private int posicion=0;
private int posicionD=0;
private int posicionD1=0;

```

```

    private double St=0;

    public static Comprobaciones calculo(double n,double v,double m, boolean
b400,boolean b500,
                                         double Fyk,double es,double gs,double
Fck,double gh,double go,
                                         double ec,double R, double P,double En,double
B2,double A1,
                                         double L, double Ht,double sta,double gt,double
k03,double Coh,
                                         double gd,double gv,double Del,double
ga,boolean cohe, double r1, double dia0,double wkMax){
    Comprobaciones f=new Comprobaciones();
    f.setN(n);
    f.setV(v);
    f.setM(m);
    f.setB400(b400);
    f.setB500(b500);
    f.setFyk(Fyk);
    f.setEs(es);
    f.setGs(gs);
    f.setFck(Fck);
    f.setGh(gh);
    f.setGo(go);
    f.setEc(ec);
    f.setR(R);
    f.setP(P);
    f.setEn(En);
    f.setB2(B2);
    f.setA1(A1);
    f.setL(L);
    f.setHt(Ht);
    f.setSta(sta);
    f.setGt(gt);
    f.setK03(k03);
    f.setCoh(Coh);
    f.setGd(gd);
    f.setGv(gv);
    f.setDel(Del);
    f.setGa(ga);
    f.setCohesivo(cohe);
    f.setR1(r1);
    f.setDia0(dia0);
    f.setWkMax(wkMax);
    f.hundimiento();
    for(int i=0;i<=19;i++){
        f.cortante(f.ancho[i],f.canto[i],i);
        f.rigidezRelativa(f.ancho[i],f.canto[i],i);
        f.vuelco(f.ancho[i],f.canto[i],i);
        f.deslizamiento(f.ancho[i],f.canto[i],i);
        f.seleccion(i);
        /*if(vuelco[marcadorValidos[i]]>2*canto[marcadorValidos[i]])
flexible(ancho[marcadorValidos[i]],
canto[marcadorValidos[i]],momentoDeCalculo[marcadorValidos[i]],marcadorValidos[i]);
        */if(f.vuelco[f.marcadorValidos[i]]<=2*f.canto[f.marcadorValidos[i]])
f.rigido(f.ancho[f.marcadorValidos[i]],
f.canto[f.marcadorValidos[i]],f.marcadorValidos[i]);
    }
    for(int j=0; j<f.contMarcVald; j++){
        if(f.vuelco[f.marcadorValidos[j]]>2*f.canto[f.marcadorValidos[j]]) {
            f.flexible(f.ancho[f.marcadorValidos[j]],
f.canto[f.marcadorValidos[j]],f.momentoDeCalculo[f.marcadorValidos[j]],f.marcadorValidos
[j]);
        }
        f.anclajeFlexible(f.areaTeorica[f.marcadorValidos[j]],f.areaReal[f.marcadorValidos[j]],f
.canto[f.marcadorValidos[j]],f.ancho[f.marcadorValidos[j]],f.diametro[f.marcadorValidos[
j]],f.marcadorValidos[j]);
        f.fisuracion(f.marcadorValidos[j]);
    }
    if(f.vuelco[f.marcadorValidos[j]]<=2*f.canto[f.marcadorValidos[j]]) {
        f.rigido(f.ancho[f.marcadorValidos[j]],
f.canto[f.marcadorValidos[j]],f.marcadorValidos[j]);
    }
}

```

```

f.anclajeRigido(f. areaTeorica[f.marcadorValidos[j]], f. areaReal[f.marcadorValidos[j]], f. c
anto[f.marcadorValidos[j]], f. ancho[f.marcadorValidos[j]], f. diametro[f.marcadorValidos[j]
], f. x1[f.marcadorValidos[j]], f. x2[f.marcadorValidos[j]], f.marcadorValidos[j]);
    }

f.cortanteDef(f. areaReal[f.marcadorValidos[j]]/10000, f. ancho[f.marcadorValidos[j]], f. can
to[f.marcadorValidos[j]], f. diametro[f.marcadorValidos[j]]*0.001, f. momentoDeCalculo[f.mar
cadorValidos[j]],
        f.marcadorValidos[j]);
    }
f.armaduraEspera();
return f;
}

public void hundimiento(){
    /*Con todo en funcion de los datos conocidos igualo la reaccion del terreno y la
presion máxima admisible*/
    //Resolucion por newton rapshon de la ecuación
    //Ecuaciones sacadas de Maple==> archivo comprobaciones
    /*
    * Con este construyo:
    * ancho[];
    * canto[];
    * seguridadH[];
    * estadoHundimineto[];
    * e[];
    */
    CuatroFuncion f=new CuatroFuncion() {
        @Override
        public double eval(double a2,double h) {
            return
            (Gh*Gh*a2*a2*b2*b2*h*h+2*Gh*Gt*a2*a2*b2*b2*h*ht+Gt*Gt*a2*a2*b2*b2*ht*ht
            -Gh*Sta*a2*a2*b2*b2*h-Gt*Sta*a2*a2*b2*ht+2*Gh*N*a2*b2*h+
            2*Gt*N*a2*b2*ht-N*Sta*a2+2*N*Sta*en+2*Sta*V*h+2*M*Sta+N*N)
            / (Gh*a2*a2*b2*h+Gt*a2*a2*b2*ht+N*a2-2*N*en-2*V*h-2*M);
        }
    };
    CuatroFuncion df=new CuatroFuncion() {
        @Override
        public double eval(double a2,double h) {
            return -(Gh*Gh*N*a2*a2*b2*b2*h*h+4*Gh*Gh*N*a2*b2*b2*en*h*h+
            4*Gh*Gh*V*a2*b2*b2*h*h+2*Gh*Gt*N*a2*a2*b2*b2*h*ht+
            8*Gh*Gt*N*a2*b2*b2*en*h*ht+8*Gh*Gt*V*a2*b2*b2*h*h*ht+
            Gt*Gt*N*a2*a2*b2*b2*ht*ht+4*Gt*Gt*N*a2*b2*b2*en*ht*ht+4*Gt*Gt*V*
            a2*b2*b2*h*ht*ht+4*Gh*Gh*M*a2*b2*b2*h*h+8*Gh*Gt*M*a2*b2*b2*h*
            ht+4*Gt*Gt*M*a2*b2*b2*ht*ht+2*Gh*N*N*a2*b2*h+4*Gh*N*N*b2*en*h
            +4*Gh*N*V*b2*h*ht+2*Gt*N*N*a2*b2*ht+4*Gt*N*N*b2*en*ht+
            4*Gt*N*V*b2*h*ht+4*Gh*M*N*b2*h+4*Gt*M*N*b2*ht+N*N*N)/ (
            (Gh*a2*a2*b2*h+Gt*a2*a2*b2*ht+N*a2-2*N*en-2*V*h-2*M) *
            (Gh*a2*a2*b2*h+Gt*a2*a2*b2*ht+N*a2-2*N*en-2*V*h-2*M));
        }
    };
    NewtonRapson3 rn=new NewtonRapson3();
    //hecho esto cosntruyo el vector con soluciones
    double Pg=0, Pt=0, F=0, Mt=0;
    for(double j=0.3; cont<=20; j=j+0.1) {
        //canto[cont-1]=h y j=h, ancho[cont-1]=a2:
        canto[cont-1]=j;//h ;guardo
        ancho[cont-1]=Math.ceil(rn.raiz(f, df, 1, 1e-5,
1000000000, j)/0.25)*0.25;//guardo el ancho obtenido de la raiz como multiplo de 0.25(el
inmediatamente superior)
        St=(Gh*ancho[cont-1]*b2*j+Gt*ancho[cont-1]*b2*ht+N)/(ancho[cont-1]-
(2*(V*j+M))/(Gh*ancho[cont-1]*b2*j+Gt*ancho[cont-1]*b2*ht+N));//calculo la respuesta del
terreno St=(N+Pg+Pt)/ancho equivalente
        seguridadH[cont-1]=Sta/St;//coeficiente de seguridad
        Pg=Gh*ancho[cont-1]*canto[cont-1]*b2;
        Pt=Gt*ancho[cont-1]*ht*b2;
        F=N+Pg+Pt;//Fuerzas resultantes
        Mt=M+V*canto[cont-1]+en*N;//Momento resultante
        e[cont-1]=(Mt/F);//vector Exentricidades
        estadoHundimiento[cont-1]="CORRECTO";
        if(seguridadH[cont-1]<1) estadoHundimiento[cont-
1]="HUNDIMIENTO";//Comprobacion si cumple la condicion de hundimiento
        if(ancho[cont-1]<3*e[cont-1]) estadoHundimiento[cont-1]="NO

```

```

APLICABLE;//Comprobacion si cumple la condicion para que sea aplicable la distribucion
de tensiones constante
    cont++;
}
}

public void cortante(double a2, double h, int i){
    /*
    *
    * Saco:
    * estadoCortante[];
    * cortante[]
    * momentoDeCalculo[];
    * cortanteH[];
    * seguridadC[];
    *
    * */
    //Canto útil:
    double d=h-0.1;
    //Fuerzas y momento Resultatne:
    double Fcd=N*Ga;
    double Mcd=(M+V*h+en*N)*Ga;
    double ed=Mcd/Fcd;
    //seccion comprobacion cortante
    double s2=en+0.5*a1+d;//a la derecha del muro
    double s22=en-0.5*a1-d;//a la izquierda del muro
    double s2Max=s2;//sección más alejada del centro. Para comprobar si esta se
sale o no de la zapata
    if (Math.abs(s2Max)<Math.abs(s22)) s2Max=s22;//escojo la más alejada
    //sección comprobacion momento
    double s1=en+p*a1;//a la derecha del muro
    double s11=en-p*a1;//a la izquierda del muro
    double s1Max=s1;//sección más alejada del centro. Para comprobar si esta se sale
o no de la zapata
    if (Math.abs(s1Max)<Math.abs(s11)) s1Max=s11;//escojo la más alejada
    //Variables
    double xi=0, Vdm=0, Vd1=0, Vd2=0, Mdm=0, Md1=0, Md2=0, fcv=0, As=0, rho=0, Vu2=0, Vu2Min=0;
    //System.out.println("a2="+a2+" h"+h+" d="+d+" Fcd="+Fcd+" Mcd="+Mcd+"
ed="+ed+" s2="+s2+" a2/2="+a2/2+" 6*ed/a2="+6*ed/a2);
    if (Math.abs(s2Max)>a2/2 || 3*ed/a2>1) estadoCortante[i]="NO APLICABLE";//si la
sección máxima sesale de la zapata o no se cumple condicion para distribución triangular
    if (Math.abs(s2Max)<=a2/2 && 6*ed/a2<=1){//calculos con distribución
    //Cortantes de cálculo aproximado: integrales con Maple archivo => comp1
    if (Math.abs(s2)<a2/2) Vd1=(0.5)*Fcd*(a2*a2*a2+3*a2*a2*ed-2*a2*a2*s2-
12*ed*s2*s2)/(a2*a2*a2*b2);//Sección de la derecha
    if (Math.abs(s22)<a2/2) Vd2=(0.5)*Fcd*(a2*a2*a2-
3*a2*a2*ed+2*a2*a2*s22+12*ed*s22*s2)/(a2*a2*a2*b2);//sección de la izquierda
    //seleciono el valor máximo, que será el que comruebe
    Vdm=Vd1;
    if (Vdm<Vd2) Vdm=Vd2;
    VdM=Vdm;
    //lo mismo con los momentos: integrales con Maple archivo => comp1
    if (Math.abs(s1)<a2/2) Md1=(0.125)*Fcd*(a2*a2*a2*a2+4*a2*a2*a2*ed-
4*a2*a2*a2*s1-12*a2*a2*ed*s1+4*a2*a2*s1*s1+16*ed*s1*s1*s1)/(a2*a2*a2*b2);//Sección de la
derecha
    if (Math.abs(s11)<a2/2) Md2=(0.125)*Fcd*(a2*a2*a2*a2+-
4*a2*a2*a2*ed+4*a2*a2*a2*s11-
12*a2*a2*ed*s11+4*a2*a2*s11*s11+16*ed*s11*s11*s11)/(a2*a2*a2*b2);//sección de la
izquierda
    //seleciono el valor máximo, que será el que comruebe
    Mdm=Md1;
    if (Mdm<Md2) Mdm=Md2;
    MdM=Mdm;
    xi=1+Math.sqrt(200/(d*1000));//xi aproximada
    if (xi>2) xi=2;
    fcv=fck*1e-6;//paso fcv a MPa
    As=(Mdm*Gs)/(0.9*d*fyk);//Area en la sección de momento máximo aproximada
    rho=As/(b2*d);
    if (B400) if (rho>0.02) rho=0.02;
    if (B500) if (rho>0.025) rho=0.025;
    //Valores de resistencia Aproximada del hormigón a cortante:Maple archivo=>
comprobaciones2
    Vu2=8.354859901*1e5*xi*Math.pow(rho*fcv,1.0/3.0)*b2*d/Go;
    Vu2Min=75000.*Math.pow(xi,1.5)*Math.sqrt(fcv)*b2*d/Go;
    if (Vu2<Vu2Min) Vu2=Vu2Min;
    //Comprobación cortante
    if (Vu2<Vdm) estadoCortante[i]="ROTURA";

```

```

        if (Vu2 >= Vdm) estadoCortante[i] = "CORRECTO";
    }
    //lo mismo para distribución triangular
    if (Math.abs(s2Max) <= a2/2 && 6*ed/a2 < 1 && 3*ed/a2 <= 1) {
        //Ecuacione Maple archivo => comp2
        if (Math.abs(s2) < a2/2) Vd1 = Fcd * (5*a2*a2-4*a2*ed-8*a2*s2+8*ed*s2-
4*s2*a2) / (9*b2*(a2-2*ed)*(a2-2*ed));
        if (Math.abs(s22) < a2/2) Vd2 = Fcd * (3*a2*a2-4*a2*ed+8*a2*s22-
8*ed*s22+4*s22*a2) / (9*b2*(a2-2*ed)*(a2-2*ed));
        Vdm = Vd1;
        if (Vdm < Vd2) Vdm = Vd2;
        VdM = Vdm;
        if (Math.abs(s1) < a2/2) Md1 = Fcd * (4*a2*a2*a2-3*a2*a2*ed-
15*a2*a2*s1+12*a2*ed*s1+12*a2*s1*s1-12*ed*s1*s1+4*s1*s1*s1) / (27*b2*(a2-2*ed)*(a2-2*ed));
        if (Math.abs(s11) < a2/2) Md2 = Fcd * (2*a2*a2*a2-3*a2*a2*ed+9*a2*a2*s11-
12*a2*ed*s11+12*a2*s1*s1-12*ed*s11*s11+4*s1*s1*s11) / (27*b2*(a2-2*ed)*(a2-2*ed));
        Mdm = Md1;
        if (Mdm < Md2) Mdm = Md2;
        Mdm = Mdm;
        xi = 1 + Math.sqrt(200 / (d*1000));
        if (xi > 2) xi = 2;
        fcv = fck*1e-6;
        As = (Mdm*Gs) / (0.9*d*fyk);
        rho = As / (b2*d);
        if (B400) if (rho > 0.02) rho = 0.02;
        if (B500) if (rho > 0.025) rho = 0.025;
        Vu2 = 8.354859901*1e5*xi*Math.pow(rho*fcv, 1.0/3.0)*b2*d/Go;
        Vu2Min = 75000.*Math.pow(xi, 1.5)*Math.sqrt(fcv)*b2*d/Go;
        if (Vu2 < Vu2Min) Vu2 = Vu2Min;
        if (Vu2 < Vdm) estadoCortante[i] = "ROTURA";
        if (Vu2 >= Vdm) estadoCortante[i] = "CORRECTO";
    }
    //Por ultimo almaceno los datos calculados

    if (estadoCortante[i].equals("NO
APLICABLE") || estadoCortante[i].equals("ROTURA")) {
        cortante[i] = Vdm;
        momentoDeCalculo[i] = Mdm;
        cortanteH[i] = Vu2;
        seguridadC[i] = Vu2/Vdm;
    }
    if (estadoCortante[i].equals("CORRECTO")) {
        cortante[i] = Vdm;
        momentoDeCalculo[i] = Mdm;
        cortanteH[i] = Vu2;
        seguridadC[i] = Vu2/Vdm;
    }
}

public void rigidezRelativa(double a2, double h, int i) {
    /*
    * Saco:
    * estadoRigidez[];
    * vuelo[];
    * vueloMax[];
    * seguridadR[];
    * */
    double v = 0;
    double v1 = (a2+2*en-a1)/2; //Vuelo del lado derecho
    double v2 = (a2-2*en-a1)/2; //vuelo del lado izquierdo
    v = v1; //Vuelo mayor para comprobacion
    if (v < v2) v = v2;
    double Ic = (b2*h*h*h)/12; //Momento de inercia
    double Kb = 2*(K03*(0.3/a2)*(1+a2/(2*1)))/3; //coef de balasto
    //Comprobación
    double ue = Math.pow((4*Ec*Ic)/(b2*Kb), 0.25);
    double vMax = 0.25*Math.PI*ue; //Vuelo máximo permitido
    if (v > vMax) {
        estadoRigidez[i] = "INCORRECTOR";
        vuelo[i] = v;
        vueloMax[i] = vMax;
        seguridadR[i] = vMax/v;
        if (vuelo[i] > 2*canto[i]) tipoArmadura[i] = "FLEXIBLE";
        if (vuelo[i] <= 2*canto[i]) tipoArmadura[i] = "RÍGIDA";
    }
}

```

```

        if (v<=vMax) {
            estadoRigidez[i]="CORRECTO";
            vuelo[i]=v;
            vueloMax[i]=vMax;
            seguridadR[i]=vMax/v;
            if (vuelo[i]>2*canto[i]) tipoArmadura[i]="FLEXIBLE";
            if (vuelo[i]<=2*canto[i]) tipoArmadura[i]="RÍGIDA";
        }
    }

    public void vuelco(double a2, double h, int i){
        /*
        * estadoVuelco[];
        * Mdesf[];
        * Mfav[];
        * seguridad[];
        * */
        double Pg=Gh*a2*h*b2;
        double Pt=Gt*a2*ht*b2;
        double et=a2/2;//excentricidad zapatay terreno
        //defino momentos estabilizantes y momentos desestabilizantes
        double Mdes=Gv*M;
        double Mest=((N+Pg+Pt)*et+N*en+V*h);
        if (V*h>M) {
            Mdes=V*h*Gv;
            Mest=(Pg+Pt)*et+M+N*(a2/2-en);
        }
        //Compruebo
        if (Mdes>Mest) {
            estadoVuelco[i]="INCORRECTO";
            Mdesf[i]=0;
            Mfav[i]=0;
            seguridadV[i]=0;
        }
        if (Mdes<=Mest) {
            estadoVuelco[i]="CORRECTO";
            Mdesf[i]=Mdes;
            Mfav[i]=Mest;
            seguridadV[i]=Mest/Mdes;
        }
    }

    public void deslizamiento(double a2, double h, int i){
        /*
        * estadoDeslizamiento[];
        * Vdesf[];
        * Vfav[];
        * seguridadD[];
        * */
        double Pg=Gh*a2*h*b2;
        double Pt=Gt*a2*ht*b2;
        double Vdes=Gd*V;//Fuerza destabilizante
        double Vest=0;
        //Defino la Fuerza estabilizante, en función del tipo de suelo
        Vest=a2*b2*coh/1.5;
        if (!cohesivo) Vest=(N+Pg+Pt)*Math.tan(del);
        if (Vdes>Vest) {
            estadoDeslizamiento[i]="INCORRECTO";
            Vdesf[i]=0;
            Vfav[i]=0;
            seguridadD[i]=0;
        }
        if (Vdes<=Vest) {
            estadoDeslizamiento[i]="CORRECTO";
            Vdesf[i]=Vdes;
            Vfav[i]=Vest;
            seguridadD[i]=Vest/Vdes;
        }
    }

    public void seleccion(int i){

        if(estadoHundimiento[i].equals("CORRECTO")&&estadoCortante[i].equals("CORRECTO")&&
        estadoRigidez[i].equals("CORRECTO")&&estadoVuelco[i].equals("CORRECTO")&&
        estadoDeslizamiento[i].equals("CORRECTO")){

```

```

        marcadorValidos[contMarcVald]=i;
        contMarcVald++;
    }
}

public void flexible(double a2, double h, double M, int i){
    /*Armaduras[i]
    redondos[]

    * */
    ArmadoNoSimetrico2 armadura;
    double dial=diametros[0];
    double dia2=diametros2[0];
    diametro[i] = 1;
    diametroReparto[i] = 1;
    Redondos2 redondo=new Redondos2();
    Redondos2 redondo2=new Redondos2();
    int j=0;
    int k=0;
    while(dial!=diametro[i]&&dia2!=diametroReparto[i]) {
        dial=diametros[j];
        dia2=diametros2[k];
        armadura = ArmadoNoSimetrico2.CalculoArmadura(b2 * 100, h * 100, fyk /
1000000, fck / 1000000, Gs, Go, Es, M, 0, r, dia2);
        areaTeorica[i] = armadura.getA1();
        areaTeoricaReparto[i] = areaTeorica[i] * 0.3;
        areaMinima[i] = 0.0018 * b2 * h*10000*0.5;
        if (B400) areaMinima[i] = 0.002 * b2 * h*10000*0.5;
        if (areaTeoricaReparto[i] >= areaMinima[i]){
            redondo2.calculoRedondos(b2 * 100, areaTeoricaReparto[i], 0, dia2);
            diametroReparto[i] = diametros2[redondo2.getContDial()];
            nRedondosReparto[i] = redondo2.getN1();
        }
        if (areaTeoricaReparto[i] < areaMinima[i]) {
            redondo2.calculoRedondos(b2 * 100, areaMinima[i], 0, dia2);
            diametroReparto[i] = diametros2[redondo2.getContDial()];
            nRedondosReparto[i] = redondo2.getN1();
        }
        if (areaTeorica[i] >= areaMinima[i]){
            redondo.calculoRedondos(b2 * 100, areaTeorica[i], 0, dial);
            diametro[i] = diametros[redondo.getContDial()-1];
            nRedondos[i] = redondo.getN1();
        }
        if (areaTeorica[i] < areaMinima[i]) {
            redondo.calculoRedondos(b2 * 100, areaMinima[i], 0, dial);
            diametro[i] = diametros[redondo.getContDial()-1];
            nRedondos[i] = redondo.getN1();
        }
        armaduras[i] = armadura;
        redondos[i] = redondo;
        if(dial!=diametro[i])j++;

        if(dia2!=diametroReparto[i])k++;

        areaReal[i] = (nRedondos[i] * Math.PI * diametro[i] * diametro[i] * 0.01 /
4);
        areaRealReparto[i] = (nRedondosReparto[i] * Math.PI * diametroReparto[i] *
diametroReparto[i] * 0.01 / 4);
        posicionD=redondo.getContDial()-1;
        D[i]=armadura.getD()*0.01;
        pln[i]=armadura.getPln();
        ds1[i]=redondo.getDs1();
    }
}

public void rigido(double a2, double h, int i){

    //Calculo zapata rígida:
    Redondos2 redondo=new Redondos2();
    Redondos2 redondo2=new Redondos2();

    double dial=diametros[0];
    double dia2=diametros2[0];
    diametro[i] = 1;
    diametroReparto[i] = 1;

```



```

int j=0;
int k=0;

double Nd=N;
double ey=M/N;
while (dia1!=diametro[i]&&dia2!=diametroReparto[i]) {
    dia1=diametros[j];
    dia2=diametros2[k];

    double d = h - dial * 0.001 * 0.5 - dia2 * 0.001-r;

    double Rd1 = (0.5) * Nd * (a2 * a2 * a2 - 2 * a2 * a2 * en + 3 * a2 * a2 *
ey - 12 * en * en * ey) / (a2 * a2 * a2);
    double Rd2 = (0.5) * Nd * (a2 * a2 * a2 + 2 * a2 * a2 * en - 3 * a2 * a2 *
ey + 12 * en * en * ey) / (a2 * a2 * a2);
    double X1 = (0.25) * (a2 * a2 * a2 + 2 * a2 * a2 * en + 4 * a2 * a2 * ey + 8
* a2 * en * ey + 16 * en * en * ey) / (a2 * a2 + 3 * a2 * ey + 6 * en * ey);
    double X2 = (0.25) * (a2 * a2 * a2 - 2 * a2 * a2 * en - 4 * a2 * a2 * ey + 8
* a2 * en * ey - 16 * en * en * ey) / (a2 * a2 - 3 * a2 * ey + 6 * en * ey);
    double T1 = Rd1 * (X1 - a1 / 4) / (0.85 * d);
    double T2 = Rd2 * (X2 - a1 / 4) / (0.85 * d);
    x1 [i]= X1;
    x2 [i]= X2;
    double As = 0;
    if (T1 > T2) As = T1 * Gs / (fyk);
    if (T2 > T1) As = T2 * Gs / (fyk);

    areaTeorica [i]= As * 10000;
    areaTeoricaReparto [i]= areaTeorica [i]* 0.3;
    areaMinima [i]= 0.0018 * b2 * h * 10000*0.5;
    if (B400) areaMinima [i]= 0.002 * b2 * h * 10000*0.5;
    if (areaTeoricaReparto [i]>= areaMinima[i]) {
        redondo2.calculoRedondos(b2 * 100, areaTeoricaReparto[i], 0, dia2);
        diametroReparto[i] = diametros2[redondo2.getContDial() ];
        nRedondosReparto[i] = redondo2.getN1();
    }
    if (areaTeoricaReparto[i] < areaMinima[i]) {
        redondo2.calculoRedondos(b2 * 100, areaMinima[i], 0, dia2);
        diametroReparto[i] = diametros2[redondo2.getContDial() ];
        nRedondosReparto[i] = redondo2.getN1();
    }
    if (areaTeorica[i] >= areaMinima[i]) {
        redondo.calculoRedondos(b2 * 100, areaTeorica[i], 0, dia1);
        diametro[i] = diametros[redondo.getContDial() - 1];
        nRedondos[i] = redondo.getN1();
    }
    if (areaTeorica[i] < areaMinima[i]) {
        redondo.calculoRedondos(b2 * 100, areaMinima[i], 0, dia1);
        diametro[i] = diametros[redondo.getContDial() - 1];
        nRedondos[i] = redondo.getN1();
    }
    redondos[i] = redondo;
    areaReal[i] = nRedondos[i] * Math.PI * diametro[i]* diametro[i] * 0.01 / 4;
    areaRealReparto[i] = nRedondosReparto[i] * Math.PI * diametroReparto[i] *
diametroReparto[i] * 0.01 / 4;
    if (dial != diametro[i]) j++;
    if (dia2 != diametroReparto[i]) k++;
    D[i]=d;
    posicionD=redondo.getContDial()-1;
}
}

public void cortanteDef(double A, double a2, double h,double dia, double Mmax, int
i){
    //Comprobacion a cortante con Iso datos calculados:
    //Canto útil:
    double d=h-r-dia/2;
    //Fuerzas y momento Resultatne:
    double Fcd=N*Ga;
    double Mcd=(M+V*h+en*N)*Ga;
    double ed=Mcd/Fcd;
    //seccion comprobacion cortante
    double s2=en+0.5*a1+d;//a la derecha del muro
    double s22=en-0.5*a1-d;//a la izquierda del muro
    double s2Max=s2;//sección más alejada del centro. Para comprobar si esta se
sale o no de la zapata

```

```

if(Math.abs(s2Max)<Math.abs(s22))s2Max=s22;//escojo la más alejada
//sección comprobación momento
double s1=en+p*a1;//a la derecha del muro
double s11=en-p*a1;//a la izquierda del muro
double s1Max=s1;//sección más alejada del centro. Para comprobar si esta se sale
o no de la zapata
if(Math.abs(s1Max)<Math.abs(s11))s1Max=s11;//escojo la más alejada
//Variables
double xi=0,Vdm=0,Vd1=0,Vd2=0,Mdm=0,Md1=0,Md2=0,fcv=0,As=0,rho=0,Vu2=0,Vu2Min=0;
//System.out.println("a2="+a2+" h"+h+" d="+d+" Fcd="+Fcd+" Mcd="+Mcd+"
ed="+ed+" s2="+s2+" a2/2="+a2/2+" 6*ed/a2="+6*ed/a2);
if(Math.abs(s2Max)>a2/2 || 3*ed/a2>1) {
    estadoCortanteDefinitivo[i]="NO APLICABLE";
    System.out.println("d "+d+" a2 "+a2+" ed "+3*ed/a2);
} //si la sección máxima sesale de la zapata o no se cumple condicion para
distribución triangular
if(Math.abs(s2Max)<=a2/2 && 6*ed/a2<=1){ //calculos con distribución
//Cortantes de cálculo aproximado: integrales con Maple archivo => comp1
if(Math.abs(s2)<a2/2)Vd1=(0.5)*Fcd*(a2*a2*a2+3*a2*a2*ed-2*a2*a2*s2-
12*ed*s2*s2)/(a2*a2*a2*b2); //Sección de la derecha
if(Math.abs(s22)<a2/2)Vd2=(0.5)*Fcd*(a2*a2*a2-
3*a2*a2*ed+2*a2*a2*s22+12*ed*s22*s22)/(a2*a2*a2*b2); //sección de la izquierda
//selecciono el valor máximo, que será el que comrruebe
Vdm=Vd1;
if(Vdm<Vd2)Vdm=Vd2;
VdM=Vdm;
System.out.println("Vd1 :"+Vd1+" Vd2: "+Vd2+" Vdm: "+Vdm);
xi=1+Math.sqrt(200/(d*1000)); //xi aproximada
if(xi>2)xi=2;
fcv=fck*1e-6; //paso fcv a MPa
rho=A/(b2*d);
if(B400) if(rho>0.02)rho=0.02;
if(B500) if(rho>0.025)rho=0.025;
//Valores de resistencia Aproximada del hormigón a cortante:Maple archivo=>
comprobaciones2
Vu2=8.354859901*1e5*xi*Math.pow(rho*fcv,1.0/3.0)*b2*d/Go;
Vu2Min=75000.*Math.pow(xi,1.5)*Math.sqrt(fcv)*b2*d/Go;
if(Vu2<Vu2Min)Vu2=Vu2Min;
//Comprobación cortante
if(Vu2<Vdm) estadoCortanteDefinitivo[i]="ROTURA";
if(Vu2>=Vdm) estadoCortanteDefinitivo[i]="CORRECTO";
}
//lo mismo para distribución triangular
if(Math.abs(s2Max)<=a2/2 && 6*ed/a2>1 && 3*ed/a2<=1) {
//Ecuacione Maple archivo => comp2
if(Math.abs(s2)<a2/2)Vd1=Fcd*(5*a2*a2-4*a2*ed-8*a2*s2+8*ed*s2-
4*s2*a2)/(9*b2*(a2-2*ed)*(a2-2*ed));
if(Math.abs(s22)<a2/2)Vd2=Fcd*(3*a2*a2-4*a2*ed+8*a2*s22-
8*ed*s22+4*s22*a2)/(9*b2*(a2-2*ed)*(a2-2*ed));
Vdm=Vd1;
if(Vdm<Vd2)Vdm=Vd2;
VdM=Vdm;
xi=1+Math.sqrt(200/(d*1000));
if(xi>2)xi=2;
fcv=fck*1e-6;
rho=A/(b2*d);
if(B400) if(rho>0.02)rho=0.02;
if(B500) if(rho>0.025)rho=0.025;
System.out.println(A+" t "+rho);
Vu2=8.354859901*1e5*xi*Math.pow(rho*fcv,1.0/3.0)*b2*d/Go;
Vu2Min=75000.*Math.pow(xi,1.5)*Math.sqrt(fcv)*b2*d/Go;
if(Vu2<Vu2Min)Vu2=Vu2Min;
//System.out.println("s1="+s1+" Vd="+Vd+" Md="+Md+" xi="+xi+"
fcv="+fcv+" As="+As+" rho="+rho+" Vu2="+Vu2+" Vu2Min="+Vu2Min);
if(Vu2<Vdm) estadoCortanteDefinitivo[i]="ROTURA";
if(Vu2>=Vdm) estadoCortanteDefinitivo[i]="CORRECTO";
}

if(estadoCortanteDefinitivo[i].equals("NO
APLICABLE") || estadoCortanteDefinitivo.equals("ROTURA")) {
    cortanteDefinitivo[i]=Vdm;
    //momentoDeCalculo[i]=Mdm;
    cortanteHDefinitivo[i]=Vu2;
    seguridadCDefinitiva[i]=Vu2/Vdm;
}
if(estadoCortanteDefinitivo[i].equals("CORRECTO")) {
    cortanteDefinitivo[i]=Vdm;
}

```

```

        //momentoDeCalculo[i]=Mdm;
        cortanteHDefinitivo[i]=Vu2;
        seguridadCDefinitiva[i]=Vu2/Vdm;
    }
}

public void anclajeFlexible(double areaTeorica, double areaReal, double h, double
a2,double dia,int i){
    double lf1=0,lf2=0,lf3=0;
    double d=h-r1-dia/2;
    lf1=0.5*a2-en-p*a1-d;
    lf2=0.5*a1+en-0.35*a1-d;
    lf3=0.5*h;
    //Hallo la longitud de anclaje básica, en funcion de fck y de el diámetro del
redondo:
    if (B400) {
        if (fck/1000000<=25) lbAnclaje[i]=lb400H25[posicionD];
        if (fck/1000000<=30) lbAnclaje[i]=lb400H30[posicionD];
        if (fck/1000000<=35) lbAnclaje[i]=lb400H35[posicionD];
        if (fck/1000000<=40) lbAnclaje[i]=lb400H40[posicionD];
        if (fck/1000000<=45) lbAnclaje[i]=lb400H45[posicionD];
        if (fck/1000000<=50) lbAnclaje[i]=lb400H50[posicionD];
    }
    if (B500) {
        if (fck/1000000<=25) lbAnclaje[i]=lb500H25[posicionD];
        if (fck/1000000<=30) lbAnclaje[i]=lb500H30[posicionD];
        if (fck/1000000<=35) lbAnclaje[i]=lb500H35[posicionD];
        if (fck/1000000<=40) lbAnclaje[i]=lb500H40[posicionD];
        if (fck/1000000<=45) lbAnclaje[i]=lb500H45[posicionD];
        if (fck/1000000<=50) lbAnclaje[i]=lb500H50[posicionD];
    }
    lnAnclaje[i]=lbAnclaje[i]*(areaTeorica/areaReal);
    //Flexible:
    //Escojo la longitud menor, y lo hago comprobando que ningunade las secciones se
encuentra fuera de la zapata;
    longitudDisponible[i]=lf1;
    if (lf2<lf1) {
        if (lf2>0.5*a2-r) longitudDisponible[i]=lf2;
        if (lf3<lf2) longitudDisponible[i]=lf3;
    }
    if (lf3<lf1) {
        if (a2*0.5-en-lf3>0.5*a1 && a2*0.5-en-lf3>0.5*a1 ) longitudDisponible[i]=lf3;
        if (lf2<lf3) if (lf2>0.5*a2-r) longitudDisponible[i]=lf2;
    }
    if (longitudDisponible[i]<=0) estdoAnclaje[i]="NO SE PUEDE ANCLAR";
    if (longitudDisponible[i]>=lnAnclaje[i]) estdoAnclaje[i]="ANLCAJE EN PROLONGACIÓN
RECTA";
    if (longitudDisponble[i]<lnAnclaje[i] &&
longitudDisponible[i]>=0.7*lnAnclaje[i]) estdoAnclaje[i]="ANLCAJE CON GANCHO O PATILLA";
    if (longitudDisponble[i]<0.7*lnAnclaje[i] ) {
        estdoAnclaje[i]="ANLCAJE CON GANCHO O PATILLA PROLONGADA";
        l1[i]=lnAnclaje[i]-longitudDisponible[i]/0.7;
    }

    System.out.println(areaTeorica+" "+areaReal+" "+lnAnclaje+"
"+longitudDisponible+ " "+estdoAnclaje);//[m]
}

public void anclajeRigido(double areaTeorica, double areaReal, double h, double
a2,double dia,double x1, double x2,int i){
    double lr1=0,lr2=0;
    lr1=0.5*a2-en-x1-r;
    lr2=0.5*a2+en-x2-r;
    if (B400) {
        if (fck/1000000<=25) lbAnclaje[i]=lb400H25[posicionD];
        if (fck/1000000<=30) lbAnclaje[i]=lb400H30[posicionD];
        if (fck/1000000<=35) lbAnclaje[i]=lb400H35[posicionD];
        if (fck/1000000<=40) lbAnclaje[i]=lb400H40[posicionD];
        if (fck/1000000<=45) lbAnclaje[i]=lb400H45[posicionD];
        if (fck/1000000<=50) lbAnclaje[i]=lb400H50[posicionD];
    }
    if (B500) {
        if (fck/1000000<=25) lbAnclaje[i]=lb500H25[posicionD];
        if (fck/1000000<=30) lbAnclaje[i]=lb500H30[posicionD];
        if (fck/1000000<=35) lbAnclaje[i]=lb500H35[posicionD];
        if (fck/1000000<=40) lbAnclaje[i]=lb500H40[posicionD];
        if (fck/1000000<=45) lbAnclaje[i]=lb500H45[posicionD];
    }
}

```

```

        if(fck/1000000<=50) lbAnclaje[i]=lb500H50[posicionD];
    }
    lnAnclaje[i]=lbAnclaje[i]*(areaTeorica/areaReal);
    longitudDisponible[i]=lr1;
    if(lr2<lr1) longitudDisponible[i]=lr2;
    if(longitudDisponible[i]<=0) estdoAnclaje[i]="NO SE PUEDE ANCLAR";
    if(longitudDisponible[i]>=lnAnclaje[i]) estdoAnclaje[i]="ANLCAJE EN PROLONGACIÓN
RECTA";
    if(longitudDisponible[i]<lnAnclaje[i] &&
longitudDisponible[i]>=0.7*lnAnclaje[i]) estdoAnclaje[i]="ANLCAJE CON GANCHO O PATILLA";
    if(longitudDisponible[i]<0.7*lnAnclaje[i]){
        estdoAnclaje[i]="ANLCAJE CON GANCHO O PATILLA PROLONGADA";
        ll[i]=lnAnclaje[i]-longitudDisponible[i]/0.7;
    }
}

public void armaduraEspera(){
    int t=0;
    double lb=0;
    while(dia0!=diametros[t]){
        t++;
        posicionD1=t;
    }
    if(B400){
        if(fck/1000000<=25) lb=lb400H25[posicionD1];
        if(fck/1000000<=30) lb=lb400H30[posicionD1];
        if(fck/1000000<=35) lb=lb400H35[posicionD1];
        if(fck/1000000<=40) lb=lb400H40[posicionD1];
        if(fck/1000000<=45) lb=lb400H45[posicionD1];
        if(fck/1000000<=50) lb=lb400H50[posicionD1];
    }
    if(B500){
        if(fck/1000000<=25) lb=lb500H25[posicionD1];
        if(fck/1000000<=30) lb=lb500H30[posicionD1];
        if(fck/1000000<=35) lb=lb500H35[posicionD1];
        if(fck/1000000<=40) lb=lb500H40[posicionD1];
        if(fck/1000000<=45) lb=lb500H45[posicionD1];
        if(fck/1000000<=50) lb=lb500H50[posicionD1];
    }
    L1es=1.4*lb;
    L2es=2*lb/3;
    L3es=lb/3;
    if(L3es<0.3) L3es=0.3;
}

public void fisuracion(int i){
    double sigmasr=0, Esm=0, Acef=0, As=0, Sm=0, wk=0;
    int naux=0, t=0;
    sigmasr=((1-pln[i])/((canto[i]/D[i])-pln[i]))*0.3*Math.pow(fck,0.6666667);
    Esm=(fyk/Es)*(1-0.5*(sigmasr/fyk)*(sigmasr/fyk));
    if(ds1[i]<=1.5*diametro[i]){
        Acef=1000*(canto[i]*100)/4;
        naux=nRedondos[i];
        if(b2!=1) naux=(int)(nRedondos[i]/b2)+1;
        As=naux*diametro[i]*diametro[i]*Math.PI*0.25;
    }
    if(ds1[i]>1.5*diametro[i]){
        Acef=15*diametro[i]*canto[i]*0.25*100;
        As=Math.PI*diametro[i]*diametro[i]*0.25;
    }
    Sm=2*(canto[i]-D[i])*1000+0.2*ds1[i]*10+0.4*0.125*(diametro[i]*Acef)/As;
    wk=1.3*Sm*Esm;
    seguridadW[i]=wkMax/wk;
    if(seguridadW[i]<1){
        try {
            nRedondosW[i] = nRedondos[i];
            ds1W[i] = ds1[i];
            while (seguridadW[i] < 1) {
                nRedondosW[i]++;
                ds1W[i] = (b2 - 2 * r * 100 - nRedondosW[i] * diametro[i]) /
(nRedondosW[i] - 1);
            }
            if (ds1[i] <= 1.5 * diametro[i]) {
                Acef = 1000 * (canto[i] * 100) / 4;
                naux = nRedondosW[i];
                if (b2 != 1) naux = (int) (nRedondosW[i] / b2) + 1;
                As = naux * diametro[i] * diametro[i] * Math.PI * 0.25;
            }
            if (ds1[i] > 1.5 * diametro[i]) {

```

```

        Acef = 15 * diametro[i] * canto[i] * 0.25 * 100;
        As = Math.PI * diametro[i] * diametro[i] * 0.25;
    }
    Sm = 2 * (canto[i] - D[i]) * 1000 + 0.2 * ds1W[i] * 10 + 0.4 * 0.125
* (diametro[i] * Acef) / As;
    wk = 1.3 * Sm * Esm;
    seguridadWW[i] = wkMax / wk;
    t++;
}
    areaRealW[i] = nRedondos[i] * diametro[i] * diametro[i] * 0.01 * Math.PI
* 0.25;

    if(ds1W[i]<2||ds1W[i]<diametro[i]){
        nRedondosW[i]=0;
        ds1[i]=0;
        areaRealW[i]=0;
        seguridadWW[i]=0;
    }
} catch(Exception ex){
    nRedondosW[i]=0;
    ds1W[i]=0;
    areaRealW[i]=0;
    seguridadWW[i]=0;
}
}
}

public double getL1es() {
    return L1es;
}

public double getL2es() {
    return L2es;
}

public double getL3es() {
    return L3es;
}

public double[] getVuelo() {
    return vuelo;
}

public double getA1() {
    return a1;
}

public void setA1(double a1) {
    this.a1 = a1;
}

public boolean isB400() {
    return B400;
}

public void setB400(boolean b400) {
    B400 = b400;
}

public boolean isB500() {
    return B500;
}

public void setB500(boolean b500) {
    B500 = b500;
}

public double getB2() {
    return b2;
}

public void setB2(double b2) {
    this.b2 = b2;
}

public double getCoh() {
    return coh;
}
}

```

```

public void setCoh(double coh) {
    this.coh = coh;
}

public boolean isCohesivo() {
    return cohesivo;
}

public void setCohesivo(boolean cohesivo) {
    this.cohesivo = cohesivo;
}

public double getDel() {
    return del;
}

public void setDel(double del) {
    this.del = del;
}

public double getFck() {
    return fck;
}

public void setFck(double fck) {
    this.fck = fck;
}

public double getFyk() {
    return fyk;
}

public void setFyk(double fyk) {
    this.fyk = fyk;
}

public double getGa() {
    return Ga;
}

public void setGa(double ga) {
    Ga = ga;
}

public double getGd() {
    return Gd;
}

public void setGd(double gd) {
    Gd = gd;
}

public double getGh() {
    return Gh;
}

public void setGh(double gh) {
    Gh = gh;
}

public double getEs() {
    return Es;
}

public void setEs(double es) {
    Es = es;
}

public double getEc() {
    return Ec;
}

public void setEc(double ec) {
    Ec = ec;
}

public double getGo() {

```

```

    return Go;
}

public void setGo(double go) {
    Go = go;
}

public double getGs() {
    return Gs;
}

public void setGs(double gs) {
    Gs = gs;
}

public double getGt() {
    return Gt;
}

public void setGt(double gt) {
    Gt = gt;
}

public double getGv() {
    return Gv;
}

public void setGv(double gv) {
    Gv = gv;
}

public double getHt() {
    return ht;
}

public void setHt(double ht) {
    this.ht = ht;
}

public double getK03() {
    return K03;
}

public void setK03(double k03) {
    K03 = k03;
}

public double getM() {
    return M;
}

public void setM(double m) {
    M = m;
}

public double getN() {
    return N;
}

public void setN(double n) {
    N = n;
}

public double getP() {
    return p;
}

public void setP(double p) {
    this.p = p;
}

public double getV() {
    return V;
}

public void setV(double v) {
    V = v;
}

```

```

}

public double getSta() {
    return Sta;
}

public void setSta(double sta) {
    Sta = sta;
}

public void setR(double r) {
    this.r = r;
}

public void setEn(double en) {
    this.en = en;
}

public void setL(double l) {
    this.l = l;
}

public double[] getAncho() {
    return ancho;
}

public double[] getCanto() {
    return canto;
}

public int getContMarcVald() {
    return contMarcVald;
}

public int[] getMarcadorValidos() {
    return marcadorValidos;
}

public double[] getAreaTeorica() {
    return areaTeorica;
}

public double[] getAreaReal() {
    return areaReal;
}

public int[] getnRedondos() {
    return nRedondos;
}

public double[] getDiametro() {
    return diametro;
}

public String[] getTipoArmadura() {
    return tipoArmadura;
}

public double[] getSeguridadC() {
    return seguridadC;
}

public double[] getSeguridadCDefinitiva() {
    return seguridadCDefinitiva;
}

public double[] getSeguridadH() {
    return seguridadH;
}

public double[] getSeguridadD() {
    return seguridadD;
}

public double[] getSeguridadR() {
    return seguridadR;
}

```



```

public double[] getSeguridadV() {
    return seguridadV;
}

public String[] getEstdoAnclaje() {
    return estdoAnclaje;
}

public double[] getLbAnclaje() {
    return lbAnclaje;
}

public double[] getLnAnclaje() {
    return lnAnclaje;
}

public double[] getLongitudDisponible() {
    return longitudDisponible;
}

public double[] getL1() {
    return l1;
}

public double getR1() {
    return r1;
}

public void setR1(double r1) {
    this.r1 = r1;
}

public double getDia0() {
    return dia0;
}

public void setDia0(double dia0) {
    this.dia0 = dia0;
}

public double getEn() {
    return en;
}

public double[] getD() {
    return D;
}

public double[] getAreaRealReparto() {
    return areaRealReparto;
}

public double[] getAreaTeoricaReparto() {
    return areaTeoricaReparto;
}

public int[] getnRedondosReparto() {
    return nRedondosReparto;
}

public double[] getDiametroReparto() {
    return diametroReparto;
}

public void setWkMax(double wkMax) {
    this.wkMax = wkMax;
}

public double[] getAreaRealW() {
    return areaRealW;
}

public double[] getSeguridadW() {
    return seguridadW;
}

```

```
public double[] getSeguridadWW() {  
    return seguridadWW;  
}  
  
public int[] getnRedondosW() {  
    return nRedondosW;  
}  
}
```

-ComprobacionesConDimensiones.class

```
public class ComprobacionesConDimensiones {

    //Variables:
    private double
        /*Esfuerzos Nominales*/
        N=0, //Axil [N]
        V=0, //Cortante [N]
        M=0, //Flector [N*m]
    /*Acero*/
        fyk=0, //límite elástico de proyecto de las armaduras pasivas [Pa]
        Es=0, //módulo de elasticidad del acero[Pa]
        Gs=0, //coef reductor
    /*Hormigon*/
        fck=0, //resist. de proyecto del hormigon a compresion[Pa]
        Gh=0, //densidad aprox [N/m^3]
        Go=0, //coeficiente de minoracion
        Ec=0, //módulo de elasticidad medio del hormigon de la zapata[Pa]
        wkMax=0.4,
    /*Geometria*/
        r=0, //recubrimiento geometrico [m]
        r1=0,
        dia0=0,
        p=0, //tipo de muro
        en=0, //excentricidad del muro[m]
        b2=0, //ancho de calculo[m]
        a1=0, //espesor del muro[m]
        l=0, //longitud del muro[m]
        ht=0, //profundidad de la zapata[m]
    /*Suelo*/
        Sta=0, //presión máxima admisible(presion hundimiento del terreno*ancho
seccion)[N/m^2]
        Gt=0, //peso terreno/m^3 [N/m^3]
        K03=0, //coeficiente balastro 30x30
        coh=0, //valor de claculo de la cohesión del terreno SIN MINORAR [N/m^2]=[Pa]
        Gd= 0, //coef seguridad fente a deslizamiento,
        Gv=0, //coef seguridad frentea vuelco
        del=0, //angulo de rozamiento entre esl suelo y la zapata
        Ga=0; //coeficiente seguridad cortante

    private boolean B400=true;
    private boolean B500=false;

    private boolean cohesivo=true;

    private ArmadoNoSimetrico2 armaduras;
    private Redondos2 redondos;

    private double anch=0;
    private double ancho=0;
    private double canto=0;
    private double e=0;

    private double seguridadH=0;
    private String estadoHundimiento=" ";

    private double cortante=0;
    private double momentoDeCalculo=0;

    private double cortanteH=0;
    private double seguridadC=0;
    private String estadoCortante=" ";

    private double vuelo=0;
    private double vueloMax=0;
    private double seguridadR=0;
    private String estadoRigidez=" ";

    private double Mdesf=0;
    private double Mfav=0;
    private double seguridadV=0;
    private String estadoVuelco=" ";

    private double Vdesf=0;
```

```

private double Vfav=0;
private double seguridadD=0;
private String estadoDeslizamiento=" ";

private String estadoCortanteDefinitivo="";

private double cortanteDefinitivo=0;
private double cortanteHDefinitivo=0;
private double seguridadCDefinitiva=0;

private double seguridadW=0,seguridadWW=0;

private String tipoArmadura=" ";
private String estdoCortanteDefinitivo;
private String estdoAnclaje=" ";
//private static String tipoArmadura;
private int cont=1;

private double diametro=0;
private double diametroReparto=0;
private final double diametros[]={12, 14, 16, 20, 25, 32, 40};
private final double diametros2[]={10, 12, 14, 16, 20, 25, 32, 40};
private double lbAnclaje=0;
private double lnAnclaje=0;
private double longitudDisponible=0;
private double l1=0;
private double anchoDef=0;
private double cantoDef=0;

private double areaReal=0;
private double areaRealw=0;
private double areaRealReparto=0;
private double areaTeorica=0;
private double areaTeoricaReparto=0;
private double areaMinima=0;
private double x1=0;
private double x2=0;
private int nRedondos=0;
private int nRedondosw=0;
private int nRedondosReparto=0;
private double D=0;

private double VdM=0;
private double MdM=0;

//Vectores longitudes basicas de anclaje: [m]
//B400:
private double[] lb400H25={0.24,0.28,0.32,0.48,0.75,1.230,1.92};
private double[] lb400H30={0.24,0.28,0.32,0.40,0.63,1.020,1.6};
private double[] lb400H35={0.24,0.28,0.32,0.40,0.56,0.92,1.44};
private double[] lb400H40={0.24,0.28,0.32,0.40,0.50,0.82,1.28};
private double[] lb400H45={0.24,0.28,0.32,0.40,0.50,0.72,1.12};
private double[] lb400H50={0.24,0.28,0.32,0.40,0.50,0.72,1.12};
//B500:
private double[] lb500H25={0.3,0.35,0.4,0.6,0.938,1.536,2.4};
private double[] lb500H30={0.3,0.35,0.4,0.52,0.813,1.331,2.08};
private double[] lb500H35={0.3,0.35,0.4,0.5,0.75,1.229,1.92};
private double[] lb500H40={0.3,0.35,0.4,0.5,0.688,1.126,1.76};
private double[] lb500H45={0.3,0.35,0.4,0.5,0.625,1.024,1.6};
private double[] lb500H50={0.3,0.35,0.4,0.5,0.625,1.024,1.6};

private int[] marcadorValidos=new int[20];
private int contMarcVald=0;

private int posicion=0;
private int posicionD=0;
private int posicionD1=0;

private double L1es=0,L2es=0,L3es=0;
private double pln=0,ds1=0,ds1W=0;

private double T1=0,T2=0;

private double St=0;

```

```

    public static ComprobacionesConDimensiones calculo(double n,double v,double m,
boolean b400,boolean b500,
double Fyk,double es,double gs,double
Fck,double gh,double go,
double ec,double R, double P,double En,double
B2,double A1,
double L, double Ht,double sta,double gt,double
k03,double Coh,
double gd,double gv,double Del,double
ga,boolean cohe,double ancho,
double canto,double r1, double
dia0,double wkMax){
    ComprobacionesConDimensiones f=new ComprobacionesConDimensiones();
    f.setN(n);
    f.setV(v);
    f.setM(m);
    f.setB400(b400);
    f.setB500(b500);
    f.setFyk(Fyk);
    f.setEs(es);
    f.setGs(gs);
    f.setFck(Fck);
    f.setGh(gh);
    f.setGo(go);
    f.setEc(ec);
    f.setR(R);
    f.setP(P);
    f.setEn(En);
    f.setB2(B2);
    f.setA1(A1);
    f.setL(L);
    f.setHt(Ht);
    f.setSta(sta);
    f.setGt(gt);
    f.setK03(k03);
    f.setCoh(Coh);
    f.setGd(gd);
    f.setGv(gv);
    f.setDel(Del);
    f.setGa(ga);
    f.setCohesivo(cohe);
    f.canto=canto;
    f.anch=ancho;
    f.setR1(r1);
    f.setDia0(dia0);
    f.setWkMax(wkMax);
    f.hundimiento();
    f.cortante(f.anch, canto);
    f.rigidezRelativa(f.anch, canto);
    f.vuelco(f.anch, canto);
    f.deslizamiento(f.anch, canto);
    if(f.vuelo>2*f.canto) {
        f.flexible(f.anch, f.canto,f.momentoDeCalculo);
        f.anclajeFlexible(f.areaTeorica,f.areaReal,f.canto,f.anch,f.diametro);
        f.fisuracion();
    }
    if(f.vuelo<=2*f.canto) {
        f.rigido(f.anch, f.canto);
    }
    f.anclajeRigido(f.areaTeorica,f.areaReal,f.canto,f.anch,f.diametro,f.x1,f.x2);
}

f.cortanteDef(f.areaReal/10000,f.anch,f.canto,f.diametro*0.001,f.momentoDeCalculo);
f.armaduraEspera();
return f;
}

public void hundimiento(){
    //hecho estpo cosntruyo el vector con soluciones
    double Pg=0,Pt=0,F=0,Mt=0;

    //canto[cont-1]=h y j=h, ancho[cont-1]=a2:
    //h ;guardo

```

```

        ancho=Math.ceil(ancho/0.25)*0.25;//guardo el ancho obtenido de la raiz como
multiplo de 0.25(el inmediatamente superior)
        St=(Gh*ancho*b2*canto+Gt*ancho*b2*ht+N)/(ancho-
(2*(V*canto+M))/(Gh*ancho*b2*canto+Gt*ancho*b2*ht+N));//calculo la respuesta del terreno
St=(N+Pg+Pt)/ancho equivalente
        seguridadH=Sta/St;//coeficiente de seguridad
        Pg=Gh*ancho*canto*b2;
        Pt=Gt*ancho*ht*b2;
        F=N+Pg+Pt;//Fuerzas resultantes
        Mt=M+V*canto+en*N;//Momento resultante
        e=(Mt/F);//vector Exentricidades
        estadoHundimiento="CORRECTO";
        if(seguridadH<1)estadoHundimiento="HUNDIMIENTO";//Comprobacion si cumple la
condicion de hundimiento
        if(ancho<3*e)estadoHundimiento="NO APLICABLE";//Comprobacion si cumple la
condicion para que sea aplicable la distribucion de tensiones constante
    }

    public void cortante(double a2, double h){
        /*
        *
        * Saco:
        * estadoCortante[];
        * cortante[]
        * momentoDeCalculo[];
        * cortanteH[];
        * seguridadC[];
        *
        * */
        //Canto útil:
        double d=h-0.1;
        //Fuerzas y momento Resultatne:
        double Fcd=N*Ga;
        double Mcd=(M+V*h+en*N)*Ga;
        double ed=Mcd/Fcd;
        //seccion comprobacion cortante
        double s2=en+0.5*a1+d;//a la derecha del muro
        double s22=en-0.5*a1-d;//a la izquierda del muro
        double s2Max=s2;//sección más alejada del centro. Para comprobar si esta se
sale o no de la zapata
        if(Math.abs(s2Max)<Math.abs(s22))s2Max=s22;//escojo la más alejada
        //sección comprobacion momento
        double s1=en+p*a1;//a la derecha del muro
        double s11=en-p*a1;//a la izquierda del muro
        double s1Max=s1;//sección más alejada del centro. Para comprobar si esta se sale
o no de la zapata
        if(Math.abs(s1Max)<Math.abs(s11))s1Max=s11;//escojo la más alejada
        //Variables
        double xi=0,Vdm=0,Vd1=0,Vd2=0,Mdm=0,Md1=0,Md2=0,fcv=0,As=0,rho=0,Vu2=0,Vu2Min=0;
        //System.out.println("a2="+a2+" h"+h+" d="+d+" Fcd="+Fcd+" Mcd="+Mcd+"
ed="+ed+" s2="+s2+" a2/2="+a2/2+" 6*ed/a2="+6*ed/a2);
        if(Math.abs(s2Max)>a2/2 || 3*ed/a2>1) estadoCortante="NO APLICABLE";//si la
sección máxima sesale de la zapata o no se cumple condicion para distribución triangular
        if(Math.abs(s2Max)<=a2/2 && 6*ed/a2<=1){//calculos con distribución
        //Cortantes de cálculo aproximado: integrales con Maple archivo => comp1
        if(Math.abs(s2)<a2/2)Vd1=(0.5)*Fcd*(a2*a2*a2+3*a2*a2*ed-2*a2*a2*s2-
12*ed*s2*s2)/(a2*a2*a2*b2);//Sección de la derecha
        if(Math.abs(s22)<a2/2)Vd2=(0.5)*Fcd*(a2*a2*a2-
3*a2*a2*ed+2*a2*a2*s22+12*ed*s22*s22)/(a2*a2*a2*b2);//sección de la izquierda
        //selecciono el valor máximo, que será el que comrruebe
        Vdm=Vd1;
        if(Vdm<Vd2)Vdm=Vd2;
        VdM=Vdm;
        //lo mismo con los momentos: integrales con Maple archivo => comp1
        if(Math.abs(s1)<a2/2)Md1=(0.125)*Fcd*(a2*a2*a2*a2+4*a2*a2*a2*ed-
4*a2*a2*a2*s1-12*a2*a2*ed*s1+4*a2*a2*s1*s1+16*ed*s1*s1*s1)/(a2*a2*a2*b2);//Sección de la
derecha
        if(Math.abs(s11)<a2/2)Md2=(0.125)*Fcd*(a2*a2*a2*a2+-
4*a2*a2*a2*ed+4*a2*a2*a2*s11-
12*a2*a2*ed*s11+4*a2*a2*s11*s11+16*ed*s11*s11*s11)/(a2*a2*a2*b2);//sección de la
izquierda
        //selecciono el valor máximo, que será el que comrruebe
        Mdm=Md1;
        if(Mdm<Md2)Mdm=Md2;
        MdM=Mdm;
        xi=1+Math.sqrt(200/(d*1000));//xi aproximada
    }

```

```

        if(xi>2)xi=2;
        fcv=fck*1e-6;//paso fcv a MPa
        As=(Mdm*Gs)/(0.9*d*fyk);//Area en la sección de momento máximo aproximada
        rho=As/(b2*d);
        if(B400)if(rho>0.02)rho=0.02;
        if(B500)if(rho>0.025)rho=0.025;
        //Valores de resistencia Aproximada del hormigón a cortante:Maple archivo=>
comprobaciones2
        Vu2=8.354859901*1e5*xi*Math.pow(rho*fcv,1.0/3.0)*b2*d/Go;
        Vu2Min=75000.*Math.pow(xi,1.5)*Math.sqrt(fcv)*b2*d/Go;
        if(Vu2<Vu2Min)Vu2=Vu2Min;
        //Comprobación cortante
        if(Vu2<Vdm)estadoCortante="ROTURA";
        if(Vu2>=Vdm)estadoCortante="CORRECTO";
    }
    //lo mismo para distribución triangular
    if(Math.abs(s2Max)<=a2/2 && 6*ed/a2<1 && 3*ed/a2<=1){
        //Ecuacione Maple archivo => comp2
        if(Math.abs(s2)<a2/2)Vd1=Fcd*(5*a2*a2-4*a2*ed-8*a2*s2+8*ed*s2-
4*s2*a2)/(9*b2*(a2-2*ed)*(a2-2*ed));
        if(Math.abs(s22)<a2/2)Vd2=Fcd*(3*a2*a2-4*a2*ed+8*a2*s22-
8*ed*s22+4*s22*a2)/(9*b2*(a2-2*ed)*(a2-2*ed));
        Vdm=Vd1;
        if(Vdm<Vd2)Vdm=Vd2;
        VdM=Vdm;
        if(Math.abs(s1)<a2/2)Md1=Fcd*(4*a2*a2*a2-3*a2*a2*ed-
15*a2*a2*s1+12*a2*ed*s1+12*a2*s1*s1-12*ed*s1*s1+4*s1*s1*s1)/(27*b2*(a2-2*ed)*(a2-2*ed));
        if(Math.abs(s11)<a2/2)Md2=Fcd*(2*a2*a2*a2-3*a2*a2*ed+9*a2*a2*s11-
12*a2*ed*s11+12*a2*s1*s1-12*ed*s11*s11+4*s1*s1*s11)/(27*b2*(a2-2*ed)*(a2-2*ed));
        Mdm=Md1;
        if(Mdm<Md2)Mdm=Md2;
        Mdm=Mdm;
        xi=1+Math.sqrt(200/(d*1000));
        if(xi>2)xi=2;
        fcv=fck*1e-6;
        As=(Mdm*Gs)/(0.9*d*fyk);
        rho=As/(b2*d);
        if(B400)if(rho>0.02)rho=0.02;
        if(B500)if(rho>0.025)rho=0.025;
        Vu2=8.354859901*1e5*xi*Math.pow(rho*fcv,1.0/3.0)*b2*d/Go;
        Vu2Min=75000.*Math.pow(xi,1.5)*Math.sqrt(fcv)*b2*d/Go;
        if(Vu2<Vu2Min)Vu2=Vu2Min;
        if(Vu2<Vdm)estadoCortante="ROTURA";
        if(Vu2>=Vdm)estadoCortante="CORRECTO";
    }
//Por ultimo almaceno los datos calculados

    if(estadoCortante.equals("NO APLICABLE")||estadoCortante.equals("ROTURA")){
        cortante=Vdm;
        momentoDeCalculo=Mdm;
        cortanteH=Vu2;
        seguridadC=Vu2/Vdm;
    }
    if(estadoCortante.equals("CORRECTO")){
        cortante=Vdm;
        momentoDeCalculo=Mdm;
        cortanteH=Vu2;
        seguridadC=Vu2/Vdm;
    }
}

public void rigidezRelativa(double a2, double h){
    /*
    * Saco:
    * estadoRigidez[];
    * vuelo[];
    * vueloMax[];
    * seguridadR[];
    * */
    double v=0;
    double v1=(a2+2*en-a1)/2;//Vuelo del lado derecho
    double v2=(a2-2*en-a1)/2;//vuelo del lado izquierdo
    v=v1;//Vuelo mayor para comprobacion
    if(v<v2)v=v2;
    double Ic=(b2*h*h*h)/12;//Momento de inercia

```

```

double Kb=2*(K03*(0.3/a2)*(1+a2/(2*1)))/3;//coef de balasto
//Comprobación
double ue=Math.pow((4*Ec*Ic)/(b2*Kb),0.25);
double vMax=0.25*Math.PI*ue;//Vuelomáximo permitido
if(v>vMax){
    estadoRigidez="INCORRECTOR";
    vuelo=v;
    vueloMax=vMax;
    seguridadR=vMax/v;
    if(vuelo>2*canto) tipoArmadura="FLEXIBLE";
    if(vuelo<=2*canto) tipoArmadura="RÍGIDA";
}
if(v<=vMax){
    estadoRigidez="CORRECTO";
    vuelo=v;
    vueloMax=vMax;
    seguridadR=vMax/v;
    if(vuelo>2*canto) tipoArmadura="FLEXIBLE";
    if(vuelo<=2*canto) tipoArmadura="RÍGIDA";
}
}

public void vuelco(double a2, double h){
    /*
    * estadoVuelco[];
    * Mdesf[];
    * Mfav[];
    * seguridad[];
    */
    double Pg=Gh*a2*h*b2;
    double Pt=Gt*a2*ht*b2;
    double et=a2/2;//excentricidad zapatay terreno
    //defino momentos estabilizantes y momentos desestabilizantes
    double Mdes=Gv*M;
    double Mest=((N+Pg+Pt)*et+N*en+V*h);
    if(V*h>M){
        Mdes=V*h*Gv;
        Mest=(Pg+Pt)*et+M+N*(a2/2-en);
    }
    //Compruebo
    if(Mdes>Mest){
        estadoVuelco="INCORRECTO";
        Mdesf=Mdes;
        Mfav=Mest;
        seguridadV=Mest/Mdes;
    }
    if(Mdes<=Mest){
        estadoVuelco="CORRECTO";
        Mdesf=Mdes;
        Mfav=Mest;
        seguridadV=Mest/Mdes;
    }
}

public void deslizamiento(double a2, double h){
    /*
    * estadoDeslizamiento[];
    * Vdesf[];
    * Vfav[];
    * seguridadD[];
    */
    double Pg=Gh*a2*h*b2;
    double Pt=Gt*a2*ht*b2;
    double Vdes=Gd*V;//Fuerza destabilizante
    double Vest=0;
    //Defino la Fuerza estabilizante, en función del tipo de suelo
    Vest=a2*b2*coh/1.5;
    if(!cohesivo)Vest=(N+Pg+Pt)*Math.tan(del);
    if(Vdes>Vest){
        estadoDeslizamiento="INCORRECTO";
        Vdesf=Vdes;
        Vfav=Vest;
        seguridadD=Vest/Vdes;
    }
    if(Vdes<=Vest){
        estadoDeslizamiento="CORRECTO";
        Vdesf=Vdes;
    }
}

```



```

        Vfav=Vest;
        seguridadD=Vest/Vdes;
    }
}

public void flexible(double a2, double h, double M){

    ArmadoNoSimetrico2 armadura;
    double dial=diametros[0];
    double dia2=diametros2[0];
    diametro = 1;
    diametroReparto = 1;
    Redondos2 redondo=new Redondos2();
    Redondos2 redondo2=new Redondos2();
    int j=0;
    int k=0;
    while (dial!=diametro&&dia2!=diametroReparto) {
        dial=diametros[j];
        dia2=diametros2[k];
        armadura = ArmadoNoSimetrico2.CalculoArmadura(b2 * 100, h * 100, fyk /
1000000, fck / 1000000, Gs, Go, Es, M, 0, r, dia2);
        areaTeorica = armadura.getA1();
        areaTeoricaReparto = areaTeorica * 0.3;
        areaMinima = 0.0018 * b2 * h*10000*0.5;
        if (B400) areaMinima = 0.002 * b2 * h*10000*0.5;
        if (areaTeoricaReparto >= areaMinima){
            redondo2.calculoRedondos(b2 * 100, areaTeoricaReparto, 0, dia2);
            diametroReparto = diametros2[redondo2.getContDial()];
            nRedondosReparto = redondo2.getN1();
        }
        if (areaTeoricaReparto < areaMinima) {
            redondo2.calculoRedondos(b2 * 100, areaMinima, 0, dia2);
            diametroReparto = diametros2[redondo2.getContDial()];
            nRedondosReparto = redondo2.getN1();
        }
        if (areaTeorica >= areaMinima){
            redondo.calculoRedondos(b2 * 100, areaTeorica, 0, dial);
            diametro = diametros[redondo.getContDial()-1];
            nRedondos = redondo.getN1();
        }
        if (areaTeorica < areaMinima) {
            redondo.calculoRedondos(b2 * 100, areaMinima, 0, dial);
            diametro = diametros[redondo.getContDial()-1];
            nRedondos = redondo.getN1();
        }
        armaduras = armadura;
        redondos = redondo;
        if(dial!=diametro)j++;

        if(dia2!=diametroReparto)k++;
        //[cm]
        areaReal = (nRedondos * Math.PI * diametro * diametro * 0.01 / 4);
        areaRealReparto = (nRedondosReparto * Math.PI * diametroReparto *
diametroReparto * 0.01 / 4);

        posicionD=redondo.getContDial()-1;

        D=armadura.getD()*0.01;
        pln=armadura.getPln();
        dsl=redondo.getDsl();
    }

}

public void flexible(double a2, double h, double M, double dial){

    ArmadoNoSimetrico2 armadura;
    //double dial=diametros[0];
    double dia2=diametros2[0];
    diametro = 1;
    diametroReparto = 1;
    Redondos2 redondo=new Redondos2();
    Redondos2 redondo2=new Redondos2();
    int j=0;
    int k=0;
    while (dial!=diametros[j])j++;
    while (dial!=diametro&&dia2!=diametroReparto) {

```

```

        dia1=diametros[j];
        dia2=diametros2[k];
        armadura = ArmadoNoSimetrico2.CalculoArmadura(b2 * 100, h * 100, fyk /
1000000, fck / 1000000, Gs, Go, Es, M, 0, r, dia2);
        areaTeorica = armadura.getA1();
        areaTeoricaReparto = areaTeorica * 0.3;
        areaMinima = 0.0018 * b2 * h*10000*0.5;
        if (B400) areaMinima = 0.002 * b2 * h*10000*0.5;
        if (areaTeoricaReparto >= areaMinima){
            redondo2.calculoRedondos(b2 * 100, areaTeoricaReparto, 0, dia2);
            diametroReparto = diametros2[redondo2.getContDial()];
            nRedondosReparto = redondo2.getN1();
        }
        if (areaTeoricaReparto < areaMinima) {
            redondo2.calculoRedondos(b2 * 100, areaMinima, 0, dia2);
            diametroReparto = diametros2[redondo2.getContDial()];
            nRedondosReparto = redondo2.getN1();
        }
        if (areaTeorica >= areaMinima){
            redondo.calculoRedondos(b2 * 100, areaTeorica, 0, dia1);
            diametro = diametros[redondo.getContDial()-1];
            nRedondos = redondo.getN1();
        }
        if (areaTeorica < areaMinima) {
            redondo.calculoRedondos(b2 * 100, areaMinima, 0, dia1);
            diametro = diametros[redondo.getContDial()-1];
            nRedondos = redondo.getN1();
        }
        armaduras = armadura;
        redondos = redondo;
        if(dia1!=diametro)j++;

        if(dia2!=diametroReparto)k++;
        //[cm]
        areaReal = (nRedondos * Math.PI * diametro * diametro * 0.01 / 4);
        areaRealReparto = (nRedondosReparto * Math.PI * diametroReparto *
diametroReparto * 0.01 / 4);

        posicionD=redondo.getContDial()-1;

        D=armadura.getD()*0.01;
        pln=armadura.getPln();
        dsl=redondo.getDsl();
    }
}

public void rigido(double a2, double h){

    Redondos2 redondo=new Redondos2();
    Redondos2 redondo2=new Redondos2();

    double dia1=diametros[0];
    double dia2=diametros2[0];
    diametro = 1;
    diametroReparto = 1;

    int j=0;
    int k=0;

    double Nd=N;
    double ey=M/N;
    while(dia1!=diametro&&dia2!=diametroReparto) {
        dia1=diametros[j];
        dia2=diametros2[k];

        double d = h - dia1 * 0.001 * 0.5 - dia2 * 0.001-r;

        double Rd1 = (0.5) * Nd * (a2 * a2 * a2 - 2 * a2 * a2 * en + 3 * a2 * a2 *
ey - 12 * en * en * ey) / (a2 * a2 * a2);
        double Rd2 = (0.5) * Nd * (a2 * a2 * a2 + 2 * a2 * a2 * en - 3 * a2 * a2 *
ey + 12 * en * en * ey) / (a2 * a2 * a2);
        double X1 = (0.25) * (a2 * a2 * a2 + 2 * a2 * a2 * en + 4 * a2 * a2 * ey + 8
* a2 * en * ey + 16 * en * en * ey) / (a2 * a2 + 3 * a2 * ey + 6 * en * ey);
        double X2 = (0.25) * (a2 * a2 * a2 - 2 * a2 * a2 * en - 4 * a2 * a2 * ey + 8

```

```

* a2 * en * ey - 16 * en * en * ey) / (a2 * a2 - 3 * a2 * ey + 6 * en * ey);
double T1 = Rd1 * (X1 - a1 / 4) / (0.85 * d);
double T2 = Rd2 * (X2 - a1 / 4) / (0.85 * d);
x1 = X1;
x2 = X2;
double As = 0;
if (T1 > T2) As = T1 * Gs / (fyk);
if (T2 > T1) As = T2 * Gs / (fyk);

areaTeorica = As * 10000;
areaTeoricaReparto = areaTeorica * 0.3;
areaMinima = 0.0018 * b2 * h * 10000*0.5;
if (B400) areaMinima = 0.002 * b2 * h * 10000*0.5;
if (areaTeoricaReparto >= areaMinima) {
    redondo2.calculoRedondos(b2 * 100, areaTeoricaReparto, 0, dia2);
    diametroReparto = diametros2[redondo2.getContDial()];
    nRedondosReparto = redondo2.getN1();
}
if (areaTeoricaReparto < areaMinima) {
    redondo2.calculoRedondos(b2 * 100, areaMinima, 0, dia2);
    diametroReparto = diametros2[redondo2.getContDial()];
    nRedondosReparto = redondo2.getN1();
}
if (areaTeorica >= areaMinima) {
    redondo.calculoRedondos(b2 * 100, areaTeorica, 0, dial);
    diametro = diametros[redondo.getContDial() - 1];
    nRedondos = redondo.getN1();
}
if (areaTeorica < areaMinima) {
    redondo.calculoRedondos(b2 * 100, areaMinima, 0, dial);
    diametro = diametros[redondo.getContDial() - 1];
    nRedondos = redondo.getN1();
}
redondos = redondo;
areaReal = nRedondos * Math.PI * diametro * diametro * 0.01 / 4;
areaRealReparto = nRedondosReparto * Math.PI * diametroReparto *
diametroReparto * 0.01 / 4;
if (dial != diametro) j++;
if (dia2 != diametroReparto) k++;
D=d;
posicionD=redondo.getContDial()-1;
}

}

public void cortanteDef(double A, double a2, double h, double dia, double Mmax) {
    //Comprobacion a cortante con lso datos calculados:
    //Canto útil:
    double d=h-r-dia/2;
    //Fuerzas y momento Resultatne:
    double Fcd=N*Ga;
    double Mcd=(M+V*h+en*N)*Ga;
    double ed=Mcd/Fcd;
    //seccion comprobacion cortante
    double s2=en+0.5*a1+d;//a la derecha del muro
    double s22=en-0.5*a1-d;//a la izquierda del muro
    double s2Max=s2;//sección más alejada del centro. Para comprobar si esta se
sale o no de la zapata
    if(Math.abs(s2Max)<Math.abs(s22)) s2Max=s22;//escojo la más alejada
    //sección comprobacion momento
    double s1=en+p*a1;//a la derecha del muro
    double s11=en-p*a1;//a la izquierda del muro
    double s1Max=s1;//sección más alejada del centro. Para comprobar si esta se sale
o no de la zapata
    if(Math.abs(s1Max)<Math.abs(s11)) s1Max=s11;//escojo la más alejada
    //Variables
    double xi=0, Vdm=0, Vd1=0, Vd2=0, Mdm=0, Md1=0, Md2=0, fcv=0, As=0, rho=0, Vu2=0, Vu2Min=0;
    //System.out.println("a2="+a2+" h"+h+" d="+d+" Fcd="+Fcd+" Mcd="+Mcd+"
ed="+ed+" s2="+s2+" a2/2="+a2/2+" 6*ed/a2="+6*ed/a2);
    if(Math.abs(s2Max)>a2/2 || 3*ed/a2>1) {
        estadoCortanteDefinitivo="NO APLICABLE";
        System.out.println("d "+d+" a2 "+a2+" ed "+3*ed/a2);
    }//si la sección máxima sesale de la zapata o no se cumple condicion para
distribución triangular
    if(Math.abs(s2Max)<=a2/2 && 6*ed/a2<=1){//calculos con distribución
    //Cortantes de cálculo aproximado: integrales con Maple archivo => comp1
    if(Math.abs(s2)<a2/2)Vd1=(0.5)*Fcd*(a2*a2*a2+3*a2*a2*ed-2*a2*a2*s2-

```

```

12*ed*s2*s2)/(a2*a2*a2*b2); //Sección de la derecha
    if(Math.abs(s22)<a2/2)Vd2=(0.5)*Fcd*(a2*a2*a2-
3*a2*a2*ed+2*a2*a2*s22+12*ed*s22*s22)/(a2*a2*a2*b2); //sección de la izquierda
    //selecciono el valor máximo, que será el que comrruebe
    Vdm=Vd1;
    if(Vdm<Vd2)Vdm=Vd2;
    VdM=Vdm;
    xi=1+Math.sqrt(200/(d*1000)); //xi aproximada
    if(xi>2)xi=2;
    fcv=fck*1e-6; //paso fcv a MPa
    rho=A/(b2*d);
    if(B400)if(rho>0.02)rho=0.02;
    if(B500)if(rho>0.025)rho=0.025;
    //Valores de resistencia Aproximada del hormigón a cortante:Maple archivo=>
comprobaciones2
    Vu2=8.354859901*1e5*xi*Math.pow(rho*fcv,1.0/3.0)*b2*d/Go;
    Vu2Min=75000.*Math.pow(xi,1.5)*Math.sqrt(fcv)*b2*d/Go;
    if(Vu2<Vu2Min)Vu2=Vu2Min;
    //Comprobación cortante
    if(Vu2<Vdm)estadoCortanteDefinitivo="ROTURA";
    if(Vu2>=Vdm)estadoCortanteDefinitivo="CORRECTO";
}
//lo mismo para distribución triangular
if(Math.abs(s2Max)<=a2/2 && 6*ed/a2>1 && 3*ed/a2<=1){
    //Ecuacione Maple archivo => comp2
    if(Math.abs(s2)<a2/2)Vd1=Fcd*(5*a2*a2-4*a2*ed-8*a2*s2+8*ed*s2-
4*s2*a2)/(9*b2*(a2-2*ed)*(a2-2*ed));
    if(Math.abs(s22)<a2/2)Vd2=Fcd*(3*a2*a2-4*a2*ed+8*a2*s22-
8*ed*s22+4*s22*a2)/(9*b2*(a2-2*ed)*(a2-2*ed));
    Vdm=Vd1;
    if(Vdm<Vd2)Vdm=Vd2;
    VdM=Vdm;
    xi=1+Math.sqrt(200/(d*1000));
    if(xi>2)xi=2;
    fcv=fck*1e-6;
    rho=A/(b2*d);
    if(B400)if(rho>0.02)rho=0.02;
    if(B500)if(rho>0.025)rho=0.025;
    Vu2=8.354859901*1e5*xi*Math.pow(rho*fcv,1.0/3.0)*b2*d/Go;
    Vu2Min=75000.*Math.pow(xi,1.5)*Math.sqrt(fcv)*b2*d/Go;
    if(Vu2<Vu2Min)Vu2=Vu2Min;
    //System.out.println("s1="+s1+" Vd="+Vd+" Md="+Md+" xi="+xi+"
fcv="+fcv+" As="+As+" rho="+rho+" Vu2="+Vu2+" Vu2Min="+Vu2Min);
    if(Vu2<Vdm)estadoCortanteDefinitivo="ROTURA";
    if(Vu2>=Vdm)estadoCortanteDefinitivo="CORRECTO";
}
}

if(estadoCortanteDefinitivo.equals("NO
APLICABLE")||estadoCortanteDefinitivo.equals("ROTURA")){
    cortanteDefinitivo=Vdm;
    //momentoDeCalculo[i]=Mdm;
    cortanteHDefinitivo=Vu2;
    seguridadCDefinitiva=Vu2/Vdm;
}
if(estadoCortanteDefinitivo.equals("CORRECTO")){
    cortanteDefinitivo=Vdm;
    //momentoDeCalculo[i]=Mdm;
    cortanteHDefinitivo=Vu2;
    seguridadCDefinitiva=Vu2/Vdm;
}
}

public void anclajeFlexible(double areaTeorica, double areaReal, double h, double
a2,double dia){
    double lf1=0,lf2=0,lf3=0;
    double d=h-r-dia/2;
    lf1=0.5*a2-en*p*a1-d;
    lf2=0.5*a1+en-0.35*a1-d;
    lf3=0.5*h;
    //Hallo la longitud de anclaje básica, en funcion de fck y de el diámetro del
redondo:
    if(B400){
        if(fck/1000000<=25)lbAnclaje=lb400H25[posicionD];
        if(fck/1000000<=30)lbAnclaje=lb400H30[posicionD];
        if(fck/1000000<=35)lbAnclaje=lb400H35[posicionD];
        if(fck/1000000<=40)lbAnclaje=lb400H40[posicionD];
    }
}

```

```

        if(fck/1000000<=45) lbAnclaje=lb400H45[posicionD];
        if(fck/1000000<=50) lbAnclaje=lb400H50[posicionD];
    }
    if(B500){
        if(fck/1000000<=25) lbAnclaje=lb500H25[posicionD];
        if(fck/1000000<=30) lbAnclaje=lb500H30[posicionD];
        if(fck/1000000<=35) lbAnclaje=lb500H35[posicionD];
        if(fck/1000000<=40) lbAnclaje=lb500H40[posicionD];
        if(fck/1000000<=45) lbAnclaje=lb500H45[posicionD];
        if(fck/1000000<=50) lbAnclaje=lb500H50[posicionD];
    }
    lnAnclaje=lbAnclaje*(areaTeorica/areaReal);
    //Flexible:
    //Escojo la longitud menor, y lo hago comprobando que ningunade las secciones se
    encuentra fuera de la zapata;
    longitudDisponible=lf1;
    if(lf2<lf1){
        if(lf2>0.5*a2-r) longitudDisponible=lf2;
        if(lf3<lf2) longitudDisponible=lf3;
    }
    if(lf3<lf1){
        if(a2*0.5-en-lf3>0.5*a1 && a2*0.5-en-lf3>0.5*a1 ) longitudDisponible=lf3;
        if(lf2<lf3) if(lf2>0.5*a2-r) longitudDisponible=lf2;
    }
    if(longitudDisponible<=0) estdoAnclaje="NO SE PUEDE ANCLAR";
    if(longitudDisponible>=lnAnclaje) estdoAnclaje="ANLCAJE EN PROLONGACIÓN RECTA";
    if(longitudDisponible<lnAnclaje &&
longitudDisponible>=0.7*lnAnclaje) estdoAnclaje="ANLCAJE CON GANCHO O PATILLA";
    if(longitudDisponible<0.7*lnAnclaje ){
        estdoAnclaje="ANLCAJE CON GANCHO O PATILLA PROLONGADA";
        l1=lnAnclaje-longitudDisponible/0.7;
    }
}

public void anclajeRigido(double areaTeorica, double areaReal, double h, double
a2,double dia,double x1, double x2){
    double lr1=0,lr2=0;
    lr1=0.5*a2-en-x1-r;
    lr2=0.5*a2+en-x2-r;
    if(B400){
        if(fck/1000000<=25) lbAnclaje=lb400H25[posicionD];
        if(fck/1000000<=30) lbAnclaje=lb400H30[posicionD];
        if(fck/1000000<=35) lbAnclaje=lb400H35[posicionD];
        if(fck/1000000<=40) lbAnclaje=lb400H40[posicionD];
        if(fck/1000000<=45) lbAnclaje=lb400H45[posicionD];
        if(fck/1000000<=50) lbAnclaje=lb400H50[posicionD];
    }
    if(B500){
        if(fck/1000000<=25) lbAnclaje=lb500H25[posicionD];
        if(fck/1000000<=30) lbAnclaje=lb500H30[posicionD];
        if(fck/1000000<=35) lbAnclaje=lb500H35[posicionD];
        if(fck/1000000<=40) lbAnclaje=lb500H40[posicionD];
        if(fck/1000000<=45) lbAnclaje=lb500H45[posicionD];
        if(fck/1000000<=50) lbAnclaje=lb500H50[posicionD];
    }
    lnAnclaje=lbAnclaje*(areaTeorica/areaReal);
    longitudDisponible=lr1;
    if(lr2<lr1) longitudDisponible=lr2;
    if(longitudDisponible<=0) estdoAnclaje="NO SE PUEDE ANCLAR";
    if(longitudDisponible>=lnAnclaje) estdoAnclaje="ANLCAJE EN PROLONGACIÓN RECTA";
    if(longitudDisponible<lnAnclaje &&
longitudDisponible>=0.7*lnAnclaje) estdoAnclaje="ANLCAJE CON GANCHO O PATILLA";
    if(longitudDisponible<0.7*lnAnclaje){
        estdoAnclaje="ANLCAJE CON GANCHO O PATILLA PROLONGADA";
        l1=lnAnclaje-longitudDisponible/0.7;
    }
}

public void armaduraEspera(){
    int t=0;
    double lb=0;
    while(dia0!=diametros[t]){
        t++;
        posicionD1=t;
    }
    if(B400){
        if(fck/1000000<=25) lb=lb400H25[posicionD1];

```

```

        if(fck/1000000<=30) lb=lb400H30[posicionD1];
        if(fck/1000000<=35) lb=lb400H35[posicionD1];
        if(fck/1000000<=40) lb=lb400H40[posicionD1];
        if(fck/1000000<=45) lb=lb400H45[posicionD1];
        if(fck/1000000<=50) lb=lb400H50[posicionD1];
    }
    if(B500){
        if(fck/1000000<=25) lb=lb500H25[posicionD1];
        if(fck/1000000<=30) lb=lb500H30[posicionD1];
        if(fck/1000000<=35) lb=lb500H35[posicionD1];
        if(fck/1000000<=40) lb=lb500H40[posicionD1];
        if(fck/1000000<=45) lb=lb500H45[posicionD1];
        if(fck/1000000<=50) lb=lb500H50[posicionD1];
    }
    L1es=1.4*lb;
    L2es=2*lb/3;
    L3es=lb/3;
    if (L3es<0.3)L3es=0.3;
}

public void fisuracion(){
    double sigmasr=0,Esm=0,Acef=0,As=0,Sm=0,wk=0;
    int naux=0,t=0;
    sigmasr=((1-pln)/((canto/D)-pln))*0.3*Math.pow(fck,0.6666667);
    Esm=(fyk/Es)*(1-0.5*(sigmasr/fyk)*(sigmasr/fyk));
    if(ds1<=1.5*diametro){
        Acef=1000*(canto*100)/4;
        naux=nRedondos;
        if(b2!=1)naux=(int)(nRedondos/b2)+1;
        As=naux*diametro*diametro*Math.PI*0.25;
    }
    if(ds1>1.5*diametro){
        Acef=15*diametro*canto*0.25*100;
        As=Math.PI*diametro*diametro*0.25;
    }
    Sm=2*(canto-D)*1000+0.2*ds1*10+0.4*0.125*(diametro*Acef)/As;
    wk=1.3*Sm*Esm;
    seguridadW=wkMax/wk;
    if(seguridadW<1){
        try {
            nRedondosw = nRedondos;
            ds1W = ds1;
            while (seguridadW < 1) {
                nRedondosw++;
                ds1W = (b2 - 2 * r * 100 - nRedondosw * diametro) / (nRedondosw -
1);

                if (ds1 <= 1.5 * diametro) {
                    Acef = 1000 * (canto * 100) / 4;
                    naux = nRedondosw;
                    if (b2 != 1) naux = (int) (nRedondosw / b2) + 1;
                    As = naux * diametro * diametro * Math.PI * 0.25;
                }
                if (ds1 > 1.5 * diametro) {
                    Acef = 15 * diametro * canto * 0.25 * 100;
                    As = Math.PI * diametro * diametro * 0.25;
                }
                Sm = 2 * (canto - D) * 1000 + 0.2 * ds1W * 10 + 0.4 * 0.125 *
(diametro * Acef) / As;
                wk = 1.3 * Sm * Esm;
                seguridadWW = wkMax / wk;
                t++;
            }
            areaRealw = nRedondos * diametro * diametro * 0.01 * Math.PI * 0.25;
            if(ds1W<2||ds1W<diametro){
                nRedondosw=0;
                ds1W=0;
                areaRealw=0;
                seguridadWW=0;
            }
        } catch (Exception ex){
            nRedondosw=0;
            ds1W=0;
            areaRealw=0;
            seguridadWW=0;
        }
    }
}
}
}

```

```

public int getnRedondosw() {
    return nRedondosw;
}

public double getSeguridadWW() {
    return seguridadWW;
}

public double getAreaRealw() {
    return areaRealw;
}

public double getDslW() {
    return dslW;
}

public double getSeguridadW() {
    return seguridadW;
}

public double getL1es() {
    return L1es;
}

public double getL2es() {
    return L2es;
}

public double getL3es() {
    return L3es;
}

public double getA1() {
    return a1;
}

public void setA1(double a1) {
    this.a1 = a1;
}

public boolean isB400() {
    return B400;
}

public void setB400(boolean b400) {
    B400 = b400;
}

public boolean isB500() {
    return B500;
}

public void setB500(boolean b500) {
    B500 = b500;
}

public double getB2() {
    return b2;
}

public void setB2(double b2) {
    this.b2 = b2;
}

public double getCoh() {
    return coh;
}

public void setCoh(double coh) {
    this.coh = coh;
}

public boolean isCohesivo() {
    return cohesivo;
}

```

```

public void setCohesivo(boolean cohesivo) {
    this.cohesivo = cohesivo;
}

public double getDel() {
    return del;
}

public void setDel(double del) {
    this.del = del;
}

public double getFck() {
    return fck;
}

public void setFck(double fck) {
    this.fck = fck;
}

public double getFyk() {
    return fyk;
}

public void setFyk(double fyk) {
    this.fyk = fyk;
}

public double getGa() {
    return Ga;
}

public void setGa(double ga) {
    Ga = ga;
}

public double getGd() {
    return Gd;
}

public void setGd(double gd) {
    Gd = gd;
}

public double getGh() {
    return Gh;
}

public void setGh(double gh) {
    Gh = gh;
}

public double getEs() {
    return Es;
}

public void setEs(double es) {
    Es = es;
}

public double getEc() {
    return Ec;
}

public void setEc(double ec) {
    Ec = ec;
}

public double getGo() {
    return Go;
}

public void setGo(double go) {
    Go = go;
}

public double getGs() {

```



```

        return Gs;
    }

    public void setGs(double gs) {
        Gs = gs;
    }

    public double getGt() {
        return Gt;
    }

    public void setGt(double gt) {
        Gt = gt;
    }

    public double getGv() {
        return Gv;
    }

    public void setGv(double gv) {
        Gv = gv;
    }

    public double getHt() {
        return ht;
    }

    public void setHt(double ht) {
        this.ht = ht;
    }

    public double getK03() {
        return K03;
    }

    public void setK03(double k03) {
        K03 = k03;
    }

    public double getM() {
        return M;
    }

    public void setM(double m) {
        M = m;
    }

    public double getN() {
        return N;
    }

    public void setN(double n) {
        N = n;
    }

    public double getP() {
        return p;
    }

    public void setP(double p) {
        this.p = p;
    }

    public double getV() {
        return V;
    }

    public void setV(double v) {
        V = v;
    }

    public double getSta() {
        return Sta;
    }

    public void setSta(double sta) {
        Sta = sta;
    }

```

```

}

public void setR(double r) {
    this.r = r;
}

public void setEn(double en) {
    this.en = en;
}

public void setL(double l) {
    this.l = l;
}

public double getAncho() {
    return ancho;
}

public double getCanto() {
    return canto;
}

public int getContMarcVald() {
    return contMarcVald;
}

public double getAreaTeorica() {
    return areaTeorica;
}

public double getAreaReal() {
    return areaReal;
}

public int getnRedondos() {
    return nRedondos;
}

public double getDiametro() {
    return diametro;
}

public String getTipoArmadura() {
    return tipoArmadura;
}

public double getSeguridadC() {
    return seguridadC;
}

public double getSeguridadCDefinitiva() {
    return seguridadCDefinitiva;
}

public double getSeguridadH() {
    return seguridadH;
}

public double getSeguridadD() {
    return seguridadD;
}

public double getSeguridadR() {
    return seguridadR;
}

public double getSeguridadV() {
    return seguridadV;
}

public String getEstdoAnclaje() {
    return estdoAnclaje;
}

public double getLbAnclaje() {

```

```

        return lbAnclaje;
    }

    public double getLnAnclaje() {
        return lnAnclaje;
    }

    public double getLongitudDisponible() {
        return longitudDisponible;
    }

    public double getL1() {
        return l1;
    }

    public void setR1(double r1) {
        this.r1 = r1;
    }

    public void setDia0(double dia0) {
        this.dia0 = dia0;
    }

    public double getVuelo() {
        return vuelo;
    }

    public double getAreaRealReparto() {
        return areaRealReparto;
    }

    public double getAreaTeoricaReparto() {
        return areaTeoricaReparto;
    }

    public double getD() {
        return D;
    }

    public double getDiametroReparto() {
        return diametroReparto;
    }

    public int getnRedondosReparto() {
        return nRedondosReparto;
    }

    public double getEn() {
        return en;
    }

    public void setWkMax(double wkMax) {
        this.wkMax = wkMax;
    }
}

```

-DatosAcero.class

```
public class DatosAcero {

    private String nombre=" ";
    private double fyk=0,Es=0,Gs=0;
    private boolean B400=false,B500=false;
    /*fyk=4000000000, //límite elástico de proyecto de las armaduras pasivas
    Ec=200000000000.0, //módulo de elasticidad
    Gs=1.15, //coef reductor*/

    public static DatosAcero datosEjemplo(){
        DatosAcero f=new DatosAcero();
        f.setNombre("Acero Ejemplos");
        f.setFyk(4000000000);
        f.setEs(200000000000.0);
        f.setGs(1.15);
        f.setB400(true);
        f.setB500(false);
        return f;
    }

    public boolean isB400() {
        return B400;
    }

    public void setB400(boolean b400) {
        B400 = b400;
    }

    public boolean isB500() {
        return B500;
    }

    public void setB500(boolean b500) {
        B500 = b500;
    }

    public double getEs() {
        return Es;
    }

    public void setEs(double ec) {
        Es = ec;
    }

    public double getFyk() {
        return fyk;
    }

    public void setFyk(double fyk) {
        this.fyk = fyk;
    }

    public double getGs() {
        return Gs;
    }

    public void setGs(double gs) {
        Gs = gs;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
}
```

-DatosArmadura.class

```
public class DatosArmadura implements Parcelable {

    public String nombre = " ";
    public double b=0, h=0, dial=0, dia2=0, r1=0, r2=0, d1=0, d2=0;

    public static DatosArmadura datosEjemplo(){

        DatosArmadura f=new DatosArmadura();
        f.b=30;
        f.h=50;
        f.dial=20;
        f.dia2=20;
        f.r1=2;
        f.r2=2;
        f.d1=3;
        f.d2=3;
        f.nombre="Datos Ejemplo";

        return f;
    }

    public DatosArmadura(){

    }

    public String getNombre() {
        return nombre;
    }

    public double getB() {
        return b;
    }

    public double getH() {
        return h;
    }

    public double getDial() {
        return dial;
    }

    public double getDia2() {
        return dia2;
    }

    public double getR1() {
        return r1;
    }

    public double getR2() {
        return r2;
    }

    public double getD1() {
        return d1;
    }

    public double getD2() {
        return d2;
    }

    public double getDx(){
        return h-r1-dial/20;
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
```

```

public void writeToParcel(Parcel dest, int flags) {
    dest.writeString(this.nombre);
    dest.writeDouble(this.b);
    dest.writeDouble(this.h);
    dest.writeDouble(this.dia1);
    dest.writeDouble(this.dia2);
    dest.writeDouble(this.r1);
    dest.writeDouble(this.r2);
    dest.writeDouble(this.d1);
    dest.writeDouble(this.d2);
}

protected DatosArmadura(Parcel in) {
    this.nombre = in.readString();
    this.b = in.readDouble();
    this.h = in.readDouble();
    this.dia1 = in.readDouble();
    this.dia2 = in.readDouble();
    this.r1 = in.readDouble();
    this.r2 = in.readDouble();
    this.d1 = in.readDouble();
    this.d2 = in.readDouble();
}

public static final Parcelable.Creator<DatosArmadura> CREATOR = new
Parcelable.Creator<DatosArmadura>() {
    @Override
    public DatosArmadura createFromParcel(Parcel source) {
        return new DatosArmadura(source);
    }

    @Override
    public DatosArmadura[] newArray(int size) {
        return new DatosArmadura[size];
    }
};
}

```

-DatosDimensiones.class

```
public class DatosDimensiones {  
  
    private String nombre=" ";  
    private double a2=0,h=0;  
  
    public double getA2() {  
        return a2;  
    }  
  
    public void setA2(double a2) {  
        this.a2 = a2;  
    }  
  
    public double getH() {  
        return h;  
    }  
  
    public void setH(double h) {  
        this.h = h;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
}
```

-DatosEsfuerzos.class

```
public class DatosEsfuerzos implements Parcelable {

    public String nombre = " ";
    public double nd=0, md=0;

    public String getNombre() {
        return nombre;
    }

    public Double getNd() {
        return nd;
    }

    public Double getMd() {
        return md;
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(this.nombre);
        dest.writeDouble(this.nd);
        dest.writeDouble(this.md);
    }

    public DatosEsfuerzos() {
    }

    protected DatosEsfuerzos(Parcel in) {
        this.nombre = in.readString();
        this.nd = in.readDouble();
        this.md = in.readDouble();
    }

    public static final Parcelable.Creator<DatosEsfuerzos> CREATOR = new
    Parcelable.Creator<DatosEsfuerzos>() {
        @Override
        public DatosEsfuerzos createFromParcel(Parcel source) {
            return new DatosEsfuerzos(source);
        }

        @Override
        public DatosEsfuerzos[] newArray(int size) {
            return new DatosEsfuerzos[size];
        }
    };
}
```

-DatosEsfuerzosNominales.class

```
public class DatosEsfuerzosNominales {

    private double N=0,M=0,V=0;
    private String nombre=" ";

    public static DatosEsfuerzosNominales datosEjemplo(){
        DatosEsfuerzosNominales f=new DatosEsfuerzosNominales();
        f.setNombre("Esfuerzos Ejemplo");
        /*N=400000,//Axil
           V=10000,//Cortante
           M=30000,//Flector*/
        f.setN(400000);
        f.setV(10000);
        f.setM(30000);
        return f;
    }

    public double getM() {
        return M;
    }

    public void setM(double m) {
        M = m;
    }

    public double getN() {
        return N;
    }

    public void setN(double n) {
        N = n;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public double getV() {
        return V;
    }

    public void setV(double v) {
        V = v;
    }
}
```

-DatosGeometria.class

```
public class DatosGeometria {

    private String nombre=" ";
    private double r=0,p=0,en=0,b2=0,a1=0,l=0,ht=0,dia0=0,r1=0;

    /*r=0.07,//recubrimiento geometrico
       p=0.35,//tipo de muro
       en=0,//excentricidad del muro
       b2=1,//ancho de calculo
       a1=0.25,//espesor del muro
       l=10,//longitud del muro
       ht=2,//profundidad de la zapata*/

    public static DatosGeometria datosEjemplo(){
        DatosGeometria f=new DatosGeometria();
        f.setNombre("Geometría Ejemplo");
        f.setR(0.03);
        f.setP(0.35);
        f.setB2(1);
        f.setA1(0.25);
        f.setL(10);
        f.setHt(2);
        f.setEn(0);
        f.setDia0(20);
        f.setR1(0.07);
        return f;
    }

    public double getDia0() {
        return dia0;
    }

    public void setDia0(double dia0) {
        this.dia0 = dia0;
    }

    public double getR1() {
        return r1;
    }

    public void setR1(double r2) {
        this.r1 = r2;
    }

    public double getA1() {
        return a1;
    }

    public void setA1(double a1) {
        this.a1 = a1;
    }

    public double getB2() {
        return b2;
    }

    public void setB2(double b2) {
        this.b2 = b2;
    }

    public double getEn() {
        return en;
    }

    public void setEn(double en) {
        this.en = en;
    }

    public double getHt() {
        return ht;
    }
}
```

```
public void setHt(double ht) {
    this.ht = ht;
}

public double getL() {
    return l;
}

public void setL(double l) {
    this.l = l;
}

public String getNombre() {
    return nombre;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public double getP() {
    return p;
}

public void setP(double p) {
    this.p = p;
}

public double getR() {
    return r;
}

public void setR(double r) {
    this.r = r;
}
}
```

-DatosHormigon.class

```
public class DatosHormigon {

    private String nombre=" ";
    private double fck=0,Go=0,Gh=0,Ec=0,wkMax=0;
    /*fck=25000000,Gh=25000,Go=1.5,*/

    public static DatosHormigon datosEjemplo(){
        DatosHormigon f=new DatosHormigon();
        f.setNombre("Hormigón Ejemplo");
        f.setFck(25000000);
        f.setGh(25000);
        f.setGo(1.5);
        f.setEc(20000000000.0);
        f.setWkMax(0.4);
        return f;
    }

    public double getWkMax() {
        return wkMax;
    }

    public void setWkMax(double wkMax) {
        this.wkMax = wkMax;
    }

    public double getEc() {
        return Ec;
    }

    public void setEc(double e) {
        Ec = e;
    }

    public double getFck() {
        return fck;
    }

    public void setFck(double fck) {
        this.fck = fck;
    }

    public double getGh() {
        return Gh;
    }

    public void setGh(double gh) {
        Gh = gh;
    }

    public double getGo() {
        return Go;
    }

    public void setGo(double go) {
        Go = go;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
}
```

-DatosMateriales.class

```
public class DatosMateriales implements Parcelable {

    public String nombre = " ";
    public double fcd=0,fyd=0,es=0;

    public static DatosMateriales datosEjemplo(){

        DatosMateriales f=new DatosMateriales();
        f.fcd=16.67;
        f.fyd=347.83;
        f.es=200000;
        f.nombre="Datos Ejemplo";

        return f;
    }

    public String getNombre() {
        return nombre;
    }

    public Double getFcd() {
        return fcd;
    }

    public Double getFyd() {
        return fyd;
    }

    public Double getEs() {
        return es;
    }

    public Double getEy(){
        return fyd/es;
    }

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(this.nombre);
        dest.writeDouble(this.fcd);
        dest.writeDouble(this.fyd);
        dest.writeDouble(this.es);
    }

    public DatosMateriales() {
    }

    protected DatosMateriales(Parcel in) {
        this.nombre = in.readString();
        this.fcd = in.readDouble();
        this.fyd = in.readDouble();
        this.es = in.readDouble();
    }

    public static final Parcelable.Creator<DatosMateriales> CREATOR = new
Parcelable.Creator<DatosMateriales>() {
        @Override
        public DatosMateriales createFromParcel(Parcel source) {
            return new DatosMateriales(source);
        }

        @Override
        public DatosMateriales[] newArray(int size) {
            return new DatosMateriales[size];
        }
    }
}
```

```
    }  
};  
}
```

-DatosSuelo.class

```
public class DatosSuelo {  
  
    /*Sta=200000,//presion hundimiento del terreno/ ancho seccion  
       Gt=20000,//densidad terreno  
       K03=45000000,//coeficiente balastro 30x30  
       coh=87500,//coeficiente de cohesión del terreno  
       Gd= 1,//coef seguridad fente a deslizamiento,  
       Gv=1.8,//coef seguridad frente a vuelco  
       del=0,//angulo de rozamiento entre esl suelo y la zapata  
       Ga=1.6;//coeficiente seguridad cortante*/  
  
    private String nombre=" ";  
    private double Sta=0, Gt=20000,K03=45000000,coh=87500,Gd=1,Gv=1.8,del=1,Ga=1.6;  
    private boolean cohe=true;  
  
    public static DatosSuelo datosEjemplo(){  
        DatosSuelo f=new DatosSuelo();  
        f.setNombre("Suelo Ejemplo");  
        f.setSta(200000);  
        f.setGt(20000);  
        f.setCoh(87500);  
        f.setGd(1);  
        f.setGv(1.8);  
        f.setDel(0);  
        f.setGa(1.6);  
        f.setCohe(true);  
        return f;  
    }  
  
    public boolean isCohe() {  
        return cohe;  
    }  
  
    public void setCohe(boolean cohe) {  
        this.cohe = cohe;  
    }  
  
    public boolean getCohe(){return cohe;}  
    public double getCoh() {  
        return coh;  
    }  
  
    public void setCoh(double coh) {  
        this.coh = coh;  
    }  
  
    public double getDel() {  
        return del;  
    }  
  
    public void setDel(double del) {  
        this.del = del;  
    }  
  
    public double getGa() {  
        return Ga;  
    }  
  
    public void setGa(double ga) {  
        Ga = ga;  
    }  
  
    public double getGd() {  
        return Gd;  
    }  
}
```

```
public void setGd(double gd) {
    Gd = gd;
}

public double getGt() {
    return Gt;
}

public void setGt(double gt) {
    Gt = gt;
}

public double getGv() {
    return Gv;
}

public void setGv(double gv) {
    Gv = gv;
}

public double getK03() {
    return K03;
}

public void setK03(double k03) {
    K03 = k03;
}

public String getNombre() {
    return nombre;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public double getSta() {
    return Sta;
}

public void setSta(double sta) {
    Sta = sta;
}
}
```

-DinamicoFragmentA.class

```
public class DinamicoFragmentA extends FragmentoPadre {

    public final static int FLAG_FRAGMENT_C = 3;
    public final static int FLAG_FRAGMENT_D = 4;

    private int cont=0;

    private static DatosArmadura[] dArmaduras=new DatosArmadura[50];
    private static DatosMateriales[] dMateriales=new DatosMateriales[50];
    private static DatosEsfuerzos[] dEsfuerzos=new DatosEsfuerzos[50];

    private int marcMat=0;
    private int marcEsf=0;
    private int marcArma=0;

    private ListView datosArmaduraLW;
    private ListView datosMateraillesLW;
    private ListView datosEsfuerzosLW;

    private ArrayAdapter datsArmaLWAdppter=null;
    private ArrayAdapter datsMatLWAdppter=null;
    private ArrayAdapter datsEsfLWAdppter=null;

    private int contItmsArma=0;
    private int contItmsMat=0;
    private int contItmsEsf=0;

    private static String armaduraSeleccionada;
    private static String materialSeleccionado;

    private TextView armaduraCargada;
    private TextView materialCargado;
    private TextView esfuerzoComp;

    private EditText nud;
    private EditText mud;

    private FragmentoPadre mListener;

    public DinamicoFragmentA() {}

    public static DinamicoFragmentA newInstance(DatosArmadura[] dArmaduraGuardadas,
                                                DatosMateriales[] dMaterialesGuardados,
                                                DatosEsfuerzos[] dEsfuerzosGuardados,
                                                String armaduraSeleccionadaGuardada,
                                                String materialSeleccionadoGuardado){

        DinamicoFragmentA f=new DinamicoFragmentA();

        dArmaduras=dArmaduraGuardadas;
        dMateriales=dMaterialesGuardados;
        dEsfuerzos=dEsfuerzosGuardados;

        armaduraSeleccionada=armaduraSeleccionadaGuardada;
        materialSeleccionado=materialSeleccionadoGuardado;

        return f;
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_dinamico_a, container,
false);

        armaduraCargada=(TextView) rootView.findViewById(R.id.armaduraCargada);
        materialCargado=(TextView) rootView.findViewById(R.id.materialCargado);
        esfuerzoComp=(TextView) rootView.findViewById(R.id.esfuerzoComp);

        armaduraCargada.setText(armaduraSeleccionada);
    }
}
```



```

materialCargado.setText(materialSeleccionado);

nud=(EditText) rootView.findViewById(R.id.nud);
mud=(EditText) rootView.findViewById(R.id.mud);

datosArmaduraLW= (ListView) rootView.findViewById(R.id.datosArmaduraLW);
for(int i=0; dArmaduras[i]!=null; i++){
    contItmsArma++;
}
putItemsLW(dArmaduras, contItmsArma);
datosArmaduraLW.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
        armaduraSeleccionada=String.valueOf(parent.getItemAtPosition(position));
        armaduraCargada.setText(armaduraSeleccionada);
        mListener.pasaArmaduraActual(armaduraSeleccionada);
        marcArma=0;

while(!dArmaduras[marcArma].getNombre().equals(armaduraSeleccionada) && marcArma<contItmsAr
ma) marcArma++;
        Toast.makeText(getActivity(), "Armadura Cargada:" + "\nDatos:\nb :"+
            +String.valueOf(dArmaduras[marcArma].getB())+" cm\nh :"+
            String.valueOf(dArmaduras[marcArma].getH())+" cm\nφ1 :"+
            String.valueOf(dArmaduras[marcArma].getDial())+" mm\nφ2
:"+
            String.valueOf(dArmaduras[marcArma].getDia2())+" mm\nr1
:"+
            String.valueOf(dArmaduras[marcArma].getR1())+" cm\nr2
:"+
            String.valueOf(dArmaduras[marcArma].getR2())+" cm\nr1
:"+
            String.valueOf(dArmaduras[marcArma].getD1())+" cm\nr2
:"+
            String.valueOf(dArmaduras[marcArma].getD2())
            , Toast.LENGTH_SHORT).show();
    }
});

datosMaterialesLW= (ListView) rootView.findViewById(R.id.datosMaterialesLW);
for(int i=0; dMateriales[i]!=null; i++){
    contItmsMat++;
}
putItemsLW(dMateriales, contItmsMat);
datosMaterialesLW.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
        materialSeleccionado=String.valueOf(parent.getItemAtPosition(position));
        materialCargado.setText(materialSeleccionado);
        mListener.pasaMaterialActual(materialSeleccionado);
        marcMat=0;

while(!dMateriales[marcMat].getNombre().equals(materialSeleccionado) && marcMat<contItmsMat
) marcMat++;
        Toast.makeText(getActivity(), "Materiales Cargados:" + "\nDatos:\nfyd :"+
            +String.valueOf(dMateriales[marcMat].getFyd())+"
Mpa\nFcd :"+
            String.valueOf(dMateriales[marcMat].getFcd())+" Mpa\nEs
:"+
            String.valueOf(dMateriales[marcMat].getEs())+" Mpa"
            , Toast.LENGTH_SHORT).show();
    }
});

datosEsfuerzosLW= (ListView) rootView.findViewById(R.id.datosEsfuerzosLW);
if (dEsfuerzos[0]!=null) {
    for(int i=0; dEsfuerzos[i]!=null; i++){
        contItmsEsf++;
    }
    putItemsLW(dEsfuerzos, contItmsEsf);
}
datosEsfuerzosLW.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
        marcEsf=0;

```

```

while(!dEsfuerzos[marcEsf].getNombre().equals(String.valueOf(parent.getItemAtPosition(position)))&&marcEsf<contItmsEsf)marcEsf++;
    Toast.makeText(getActivity(), "Materiales Cargados:"+" \nDatos:\nN :"+
        +String.valueOf(dEsfuerzos[marcEsf].getNd())+" N\nM :"+
        +String.valueOf(dEsfuerzos[marcEsf].getMd())+" Nm\n"
        , Toast.LENGTH_SHORT).show();
    }
});

Button irCalculos=(Button) rootView.findViewById(R.id.irCalculos);
irCalculos.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

mListener.onClickCambiarFragmentDinamico(EstaticoFragment.FLAG_FRAGMENT_B);
    }
});

Button nuevaArmadura=(Button) rootView.findViewById(R.id.nuevaArmadura);
nuevaArmadura.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        mListener.onClickCambiarFragmentDinamico(FLAG_FRAGMENT_C);
    }
});

Button nuevoMaterial=(Button) rootView.findViewById(R.id.nuevoMaterial);
nuevoMaterial.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        mListener.onClickCambiarFragmentDinamico(FLAG_FRAGMENT_D);
    }
});

Button guardarEsfuerzo=(Button) rootView.findViewById(R.id.guardarEsfuerzo);
guardarEsfuerzo.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        try {
            for( int j=0;dEsfuerzos[contItmsEsf]!=null; j++){
                contItmsEsf=j;
            }
            dEsfuerzos[contItmsEsf]=new DatosEsfuerzos();
            dEsfuerzos[contItmsEsf].nombre="Esfuerzos
"+String.valueOf(contItmsEsf+1);
            dEsfuerzos[contItmsEsf].nd=Double.parseDouble("0" +
nud.getText().toString());
            dEsfuerzos[contItmsEsf].md=Double.parseDouble("0" +
mud.getText().toString());
            nud.setText("");
            mud.setText("");
            contItmsEsf++;
            putItemsLW(dEsfuerzos, contItmsEsf);
            guardando(v);
        }catch(NumberFormatException e){
            esfuerzoComp.setText("Error al introducir los datos");
            dEsfuerzos[contItmsEsf]=null;
        }
    }
});

if(getActivity() instanceof PantallasEstrechActivity){
    Button vovler=(Button) rootView.findViewById(R.id.volverA);
    vovler.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            mListener.volver();
        }
    });
}

if(getActivity() instanceof PantallasEstrechActivity){
    Button volverA=(Button) rootView.findViewById(R.id.volverA);
    volverA.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

```

```

        //mListener.onClickCambiarFragmentDinamico();
    }
    });
}

return rootView;
}

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);
    try{
        mListener = (FragmentPadre) activity;
    }catch (ClassCastException e){
        throw new ClassCastException(activity.toString());
    }
}

public void guardando(View v){mListener.pasaDatos(dEsfuerzos);}

public void putItemsLW(DatosArmadura[] items, int i){

    String[] meme=new String[i];
    for(int v=0; v<i; v++){
        meme[v]=items[v].getNombre();
    }

    datsArmaLWAdapter= new ArrayAdapter(getActivity(),
android.R.layout.simple_list_item_1, meme);
    datosArmaduraLW.setAdapter(datsArmaLWAdapter);
}

public void putItemsLW(DatosMateriales[] items, int i){

    String[] meme=new String[i];
    for(int v=0; v<i; v++){
        meme[v]=items[v].getNombre();
    }

    datsMatLWAdapter= new ArrayAdapter(getActivity(),
android.R.layout.simple_list_item_1, meme);
    datosMateraillesLW.setAdapter(datsMatLWAdapter);
}

public void putItemsLW(DatosEsfuerzos[] items, int i){

    String[] meme=new String[i];
    for(int v=0; v<i; v++){
        meme[v]=items[v].getNombre();
    }

    datsEsfLWAdapter= new ArrayAdapter(getActivity(),
android.R.layout.simple_list_item_1, meme);
    datosEsfuerzosLW.setAdapter(datsEsfLWAdapter);
}
}
}

```

-DinamicoFragmentB.class

```
public class DinamicoFragmentB extends FragmentoPadre {

    public final static int FLAG_FRAGMENT_E = 5;

    private static TextView textTry;

    private static DatosArmadura armaduraActual=new DatosArmadura();
    private static DatosMateriales materialActual=new DatosMateriales();
    private static DatosEsfuerzos[] dEsfuerzos=new DatosEsfuerzos[50];
    private ArmadoNoSimetrico armado=new ArmadoNoSimetrico();
    private Redondos redondos=new Redondos();

    private TextView arma;
    private TextView mud;
    private TextView nud;
    private TextView mudl;
    private TextView muOptimo;
    private TextView dominio;
    private TextView pln;
    private TextView plnOptimo;
    private TextView psi;
    private TextView lambda;
    private TextView w1;
    private TextView w2;
    private TextView a1;
    private TextView a2;
    private TextView n1;
    private TextView n2;
    private TextView ds1;
    private TextView ds2;

    private TextView muLim;
    private TextView nuLim1;
    private TextView nuLim2;
    private TextView plnLim;

    private int contItmsArma=0;
    private int contArma=0;
    private int marcadorArmadura=0;

    private ListView armadurasLW;

    private ArrayAdapter datsArmaLWAdppter=null;

    private String armaduraSeleccionada=" ";
    private String dominios[]=new String[9];
    private String diametros[]=new String[10];
    private String limitesMu[]=new String[10];
    private String limitesNu[]=new String[20];
    private String limitesPln[]=new String[10];

    private LibreriaParcializable datos;

    public DinamicoFragmentB() {}

    private FragmentoPadre mListener;

    public static DinamicoFragmentB newInstance(DatosArmadura armaduraActualPasada,
                                                DatosMateriales materialActualPasado,
                                                DatosEsfuerzos[] dEsfuerzosPasados)
    {
        DinamicoFragmentB f=new DinamicoFragmentB();

        armaduraActual=armaduraActualPasada;
        materialActual=materialActualPasado;
        dEsfuerzos=dEsfuerzosPasados;

        return f;
    }
}
```

```

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
    View rootView = inflater.inflate(R.layout.fragment_dinamico_b, container,
false);

    arma=(TextView) rootView.findViewById(R.id.arma);
    mud=(TextView) rootView.findViewById(R.id.mud);
    nud=(TextView) rootView.findViewById(R.id.nud);
    mud1=(TextView) rootView.findViewById(R.id.mud1);
    muOptimo=(TextView) rootView.findViewById(R.id.muOptimo);
    dominio=(TextView) rootView.findViewById(R.id.dominio);
    pln=(TextView) rootView.findViewById(R.id.pln);
    plnOptimo=(TextView) rootView.findViewById(R.id.plnOptima);
    psi=(TextView) rootView.findViewById(R.id.psi);
    lambda=(TextView) rootView.findViewById(R.id.lambda);
    w1=(TextView) rootView.findViewById(R.id.w1);
    w2=(TextView) rootView.findViewById(R.id.w2);
    a1=(TextView) rootView.findViewById(R.id.a1);
    a2=(TextView) rootView.findViewById(R.id.a2);
    n1=(TextView) rootView.findViewById(R.id.n1);
    n2=(TextView) rootView.findViewById(R.id.n2);
    ds1=(TextView) rootView.findViewById(R.id.ds1);
    ds2=(TextView) rootView.findViewById(R.id.ds2);

    muLim=(TextView) rootView.findViewById(R.id.muLim);
    nuLim1=(TextView) rootView.findViewById(R.id.nuLim1);
    nuLim2=(TextView) rootView.findViewById(R.id.nuLim2);
    plnLim=(TextView) rootView.findViewById(R.id.plnLim1);

    dominios=getResources().getStringArray(R.array.DOMINIOS);
    /*{0="No se ha encontrado el dominio",
1="2.a sin A2",
2="2.b sin A2",
3="3 sin A2",
4="3 con A1 y A2",
5="3 sin A1 y con A2",
6="4 sin A1 y con A2",
7="4.a sin A1 y con A2",
8="5 sin A1 y con A2",
9="5 con A1 y A2" }*/
    diametros=getResources().getStringArray(R.array.DIAMETROS_NORMALIZADOS);
    limitesMu=getResources().getStringArray(R.array.LIMITES_MU);
    limitesNu=getResources().getStringArray(R.array.LIMITES_NU);
    limitesNu=getResources().getStringArray(R.array.LIMITES_NU);
    limitesPln=getResources().getStringArray(R.array.LIMITES_PLN);

    armadurasLW=(ListView) rootView.findViewById(R.id.armadurasLW);
    while (dEsfuerzos[contItmsArma]!=null) {
        contItmsArma++;
    }
    putItemsLW(contItmsArma);

    if (dEsfuerzos[0]!=null) {

        armaduraSeleccionada="Armadura 1";
        armado = ArmadoNoSimetrico.CalculoArmadura(armaduraActual, materialActual,
dEsfuerzos[0]);
        redondos=new Redondos();
        redondos.calculoRedondos(armaduraActual, armado.getA1(), armado.getA2());
        imprimirResultados(armado, redondos, materialActual.getEy());

    }

    Button irGrafico=(Button) rootView.findViewById(R.id.irGrafico);
    irGrafico.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

mListener.onClickCambiarFragmentDinamico(DinamicoFragmentB.FLAG_FRAGMENT_E);
        }
    });

    armadurasLW.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {

```

```

                armaduraSeleccionada=String.valueOf(parent.getItemAtPosition(position));
                int
pos=Integer.parseInt(armaduraSeleccionada.substring(armaduraSeleccionada.indexOf(" ")+1));
                armado= ArmadoNoSimetrico.CalculoArmadura(armaduraActual,
materialActual, dEsfuerzos[pos-1]);
                redondos=new Redondos();
                redondos.calculoRedondos(armaduraActual, armado.getA1(),
armado.getA2());
                armado.setNombre(pos);
                imprimirResultados(armado, redondos, materialActual.getEy());
        }
    });

    if(getActivity() instanceof PantallasEstrechActivity) {
        Button volver = (Button) rootView.findViewById(R.id.volverB);
        volver.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

mListener.onClickCambiarFragmentDinamico(EstaticoFragment.FLAG_FRAGMENT_A);
            }
        });
    }

    return rootView;
}

public void putItemsLW( int i){

    String[] meme=new String[i];
    for(int v=0; v<i; v++){
        meme[v]="Armadura "+String.valueOf(v+1);
    }

    datsArmaLWAdapter= new ArrayAdapter(getActivity(),
android.R.layout.simple_list_item_1, meme);
    armadurasLW.setAdapter(datsArmaLWAdapter);
}

//Fuencion para redondear resultados
public static double red(double n, int i){
    int p=(int) Math.pow(10,i);
    return Math rint(n*p)/p;
}

public void imprimirResultados(ArmadoNoSimetrico armado,Redondos redondos, double
ey){
    int contador=armado.getContadorDominio();
    double delta2=armado.getDelta2();
    //Valores armadura:
    arma.setText(armaduraSeleccionada);
    mud.setText(String.valueOf(red(armado.getMu(),4)));
    nud.setText(String.valueOf(red(armado.getNu(),4)));
    mud1.setText(String.valueOf(red(armado.getMul(),4)));
    muOptimo.setText(String.valueOf(red(armado.getMuOptimo2(),4)));
    dominio.setText(Html.fromHtml(dominios[armado.getContadorDominio()]));
    pln.setText(String.valueOf(red(armado.getPln(),4)));
    plnOptimo.setText(String.valueOf(red(armado.getPlnOptima2(),4)));
    psi.setText(String.valueOf(red(armado.getPsi2(),4)));
    lambda.setText(String.valueOf(red(armado.getLmb2(),4)));
    w1.setText(String.valueOf(red(armado.getW1(),4)));
    w2.setText(String.valueOf(red(armado.getW2(),4)));
    a1.setText(String.valueOf(red(armado.getA1(),4)));
    a2.setText(String.valueOf(red(armado.getA2(),4)));
    if(armado.getW1()==0){
        n1.setText("0.0");
        ds1.setText("0.0");
    }
    if(armado.getW1()!=0){
n1.setText(String.valueOf(red(redondos.getN1(),4))+diametros[redondos.getContDial()]);
        ds1.setText(String.valueOf(red(redondos.getDs1(),4)));
    }
    if(armado.getW2()==0){
        n2.setText("0.0");
        ds2.setText("0.0");
    }
}

```

```

    }
    if(armado.getW2() !=0) {
n2.setText (String.valueOf (red (redondos.getN2 (), 4))+diametros [redondos.getContDia2 ()]);
        ds2.setText (String.valueOf (red (redondos.getDs2 (), 4)));
    }

    //Límites; armadura={ey,delta,delta2,mu,nu,mu1,contadorDominio,pln,w1,w2,A1,A2}
    if(contador==0 || contador==8){
        muLim.setText ("");
        nuLim1.setText ("");
        nuLim2.setText ("");
        plnLim.setText ("");
    }
    if(contador==1){
        muLim.setText (Html.fromHtml (limitesMu [contador]));

nuLim1.setText (Html.fromHtml (limitesNu [contador]+String.valueOf (red (armado.getLimiteNu (a
rmado.getPln (), contador), 4))));
        nuLim2.setText (Html.fromHtml (limitesNu [contador+10]));
        plnLim.setText (Html.fromHtml (limitesPln [contador]));
    }
    if(contador==2){
        muLim.setText (Html.fromHtml (limitesMu [contador]));

nuLim1.setText (Html.fromHtml (limitesNu [contador]+String.valueOf (red (armado.getLimiteNu (a
rmado.getPln (), contador), 4))));
        nuLim2.setText (Html.fromHtml (limitesNu [contador+10]));
        plnLim.setText (Html.fromHtml (limitesPln [contador]));
    }
    if(contador==3){
        muLim.setText (Html.fromHtml (limitesMu [contador]+
            String.valueOf (red (armado.
                getMuOptimo (armado.getP1Optima (delta2,ey), ey), 4))));
        nuLim1.setText (Html.fromHtml (limitesNu [contador]+
            String.valueOf (red (armado.getLimiteNu (armado.getPln (), contador), 4))));
        nuLim2.setText (Html.fromHtml (limitesNu [contador+10]));
        plnLim.setText (Html.fromHtml (limitesPln [contador]+
            String.valueOf (red (armado.
                getP1Optima (armado.getDelta2 (), materialActual.getEy (), 4))));
    }
    if(contador==4){
        muLim.setText (Html.fromHtml (limitesMu [contador]+
            String.valueOf (red (armado.
                getMuOptimo (armado.getP1Optima (delta2,ey), ey), 4))));
        nuLim1.setText (Html.fromHtml (limitesNu [contador]+String.valueOf (red (armado.getLimiteNu (a
rmado.getPln (), contador), 4))));
        nuLim2.setText (Html.fromHtml (limitesNu [contador+10]));
        plnLim.setText (Html.fromHtml (limitesPln [contador]+
            String.valueOf (red (armado.
                getP1Optima (armado.getDelta2 (), materialActual.getEy (), 4))));
    }
    if(contador==5){
        muLim.setText (Html.fromHtml (limitesMu [contador]+
            String.valueOf (red (armado.
                getMuOptimo (armado.getP1Optima (delta2,ey), ey), 4))));
        nuLim1.setText (Html.fromHtml (limitesNu [contador]+
            String.valueOf (red (armado.getLimiteNu (armado.getMul (),
                delta2, armado.getDelta (), ey, contador), 4))));
        nuLim2.setText (Html.fromHtml (limitesNu [contador+10]));
        plnLim.setText (Html.fromHtml (limitesPln [contador]+
            String.valueOf (red (armado.
                getP1Optima (armado.getDelta2 (), materialActual.getEy (), 4))));
    }
    if(contador==6){
        muLim.setText (Html.fromHtml (limitesMu [contador]+
            String.valueOf (red (armado.getMu (armado.getP1OptimaMaxima (ey), contador), 4))));
        nuLim1.setText (Html.fromHtml (limitesNu [contador]+
            String.valueOf (red (armado.getLimiteNu (armado.getMul (),
                delta2, armado.getDelta (), ey, contador-1), 4))));
        nuLim2.setText (Html.fromHtml (limitesNu [contador+10]+

```

```

        String.valueOf(red(armado.getLimiteNu(armado.getMul(),
            delta2, armado.getDelta(), ey, contador),4)));
        plnLim.setText(Html.fromHtml(limitesPln[contador]));
    }
    if(contador==7){
        muLim.setText(Html.fromHtml(limitesMu[contador]+
String.valueOf(red(armado.getMu(armado.getPLOptimaMaxima(ey), contador),4)));
        nuLim1.setText(Html.fromHtml(limitesNu[contador]+
            String.valueOf(red(armado.getLimiteNu(armado.getMul(),
                delta2, armado.getDelta(), ey, contador-1),4)));
        nuLim2.setText(Html.fromHtml(limitesNu[contador+10]+
            String.valueOf(red(armado.getLimiteNu(armado.getMul(),
                delta2, armado.getDelta(), ey, contador),4)));
        plnLim.setText(Html.fromHtml(limitesPln[contador]+String.valueOf(
            red(armado.getDelta(),4)));
    }
    if(contador==9){
        muLim.setText(Html.fromHtml(limitesMu[contador]+
            String.valueOf(red(0.85*armado.getDelta()*(1-
armado.getDelta()/2),4)));
        nuLim1.setText(Html.fromHtml(limitesNu[contador]+
            String.valueOf(red(0.85*armado.getDelta()+((armado.getMul()-
            0.85*armado.getDelta()*(1-armado.getDelta()/2))/(1-
delta2)),4)));
        nuLim2.setText(Html.fromHtml(limitesNu[contador+10]+String.valueOf(red(armado.getDelta()
,4)));
        plnLim.setText(Html.fromHtml(limitesPln[contador]+String.valueOf(
            red(armado.getDelta(),4)));
    }
}

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);
    try{
        mListener = (FragmentPadre) activity;
    }catch (ClassCastException e){
        throw new ClassCastException(activity.toString());
    }
}
}
}

```

-DinamicoFragmentCalculoComprobaciones.class

```
public class DinamicoFragmentCalculoComprobaciones extends FragmentoPadre{

    private DatosEsfuerzosNominales[] dEsfuerzos=new DatosEsfuerzosNominales[50];
    private String esfuerzoActual="Esfuerzos Ejemplos";
    private DatosHormigon[] dHormigon=new DatosHormigon[50];
    private String hormigonSeleccionado=" ";
    private DatosAcero[] dAcero=new DatosAcero[50];
    private String aceSeleccionado=" ";
    private DatosSuelo[] dSuelo=new DatosSuelo[50];
    private String suelSeleccionado=" ";
    private DatosGeometria[] dGeometria=new DatosGeometria[50];
    private String geomSeleccionado=" ";
    private DatosDimensiones[] dDimensiones=new DatosDimensiones[50];
    private String dimeSeleccionado=" ";

    private int contEsf=0;
    private int marcEsf=0;
    private int contHorm=0;
    private int marcHorm=0;
    private int contAce=0;
    private int marcAce=0;
    private int contSuel=0;
    private int marcSuel=0;
    private int contGeom=0;
    private int marcGeom=0;
    private int contDim=0;
    private int marcDim=0;

    private TextView ancho;
    private TextView canto;
    private TextView d;
    private TextView At;
    private TextView At2;
    private TextView Ar;
    private TextView Ar2;
    private TextView Am;
    private TextView NR;
    private TextView NR2;
    private TextView Vuel;
    private TextView V;
    //private TextView dia;
    private TextView hund;
    private TextView cortante;
    private TextView rigidez;
    private TextView vuelco;
    private TextView deslizamiento;
    private TextView cortanteDef;
    private TextView tipo;
    private TextView lb;
    private TextView ln;
    private TextView ld;
    private TextView ll;
    private TextView Lles;
    private TextView L2es;
    private TextView L3es;
    private TextView tipoAnclaje;
    private TextView fisuracion;
    private TextView seguridadw;
    private ImageView zapata;
    private ImageView anclaje;

    private double anchos=0;
    private double cantos=0;
    private double areaTeorica=0;
    private double areaTeoricaReparto=0;
    private double areaReal=0;
    private double areaRealReparto=0;
    private int nRedondos=0;
    private int nRedondosReparto=0;
    private double diametro=0;
    private double diametroReparto=0;
    private double vuelo=0;
```

```

private double D=0;

private double lbas=0;
private double lnet=0;
private double ldsip=0;
private double L1=0;
private double lles=0;
private double l2es=0;
private double l3es=0;
private String estAnclaje=" ";

private double seguridadH=0;
private double seguridadC=0;
private double seguridadR=0;
private double seguridadV=0;
private double seguridadD=0;
private double seguridadCDefinitiva=0;

private String tipoArmadur=" ";

public static DinamicoFragmentCalculoComprobaciones
newInstance(DatosEsfuerzosNominales[] dEsf, String esf, DatosHormigon[] dHorm,
String horm,
DatosAcero[] dAce,String ace, DatosSuelo[] dSue,String sue,
DatosGeometria[]
dGeom,String geo,DatosDimensiones[] dDim, String dim){
    DinamicoFragmentCalculoComprobaciones f=new
DinamicoFragmentCalculoComprobaciones();
    f.dEsfuerzos=dEsf;
    f.dHormigon=dHorm;
    f.dAcero=dAce;
    f.dSuelo=dSue;
    f.dGeometria=dGeom;
    f.dDimensiones=dDim;
    f.esfuerzoActual=esf;
    f.hormigonSeleccionado=horm;
    f.aceSeleccionado=ace;
    f.suelSeleccionado=sue;
    f.geomSeleccionado=geo;
    f.dimeSeleccionado=dim;
    return f;
}

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
    View
    rootView=inflater.inflate(R.layout.dinamico_fragment_caclculo_comprobaciones,container,
false);

    ancho=(TextView) rootView.findViewById(R.id.ancho);
    canto=(TextView) rootView.findViewById(R.id.canto);
    At=(TextView) rootView.findViewById(R.id.At);
    Ar=(TextView) rootView.findViewById(R.id.Ar);
    NR=(TextView) rootView.findViewById(R.id.NR);
    //dia=(TextView) rootView.findViewById(R.id.dia);
    hund=(TextView) rootView.findViewById(R.id.hund);
    cortante=(TextView) rootView.findViewById(R.id.cortante);
    rigidez=(TextView) rootView.findViewById(R.id.rigidez);
    vuelco=(TextView) rootView.findViewById(R.id.vuelco);
    deslizamiento=(TextView) rootView.findViewById(R.id.deslizamiento);
    cortanteDef=(TextView) rootView.findViewById(R.id.cortanteDef);
    tipo=(TextView) rootView.findViewById(R.id.tipo);
    NR=(TextView) rootView.findViewById(R.id.NR);
    //dia=(TextView) rootView.findViewById(R.id.dia);
    lb=(TextView) rootView.findViewById(R.id.lb);
    ld=(TextView) rootView.findViewById(R.id.ld);
    ln=(TextView) rootView.findViewById(R.id.ln);
    ll=(TextView) rootView.findViewById(R.id.ll);
    Lles=(TextView) rootView.findViewById(R.id.Lles);
    L2es=(TextView) rootView.findViewById(R.id.L2es);
    L3es=(TextView) rootView.findViewById(R.id.L3es);

```

```

        tipoAnclaje=(TextView) rootView.findViewById(R.id.tipoAnclaje);
        anclaje=(ImageView) rootView.findViewById(R.id.anclaje);
        fisuracion=(TextView) rootView.findViewById(R.id.fisuracion);
        seguridadw=(TextView) rootView.findViewById(R.id.segF);
        while (dEsfuerzos[contEsf] != null) {
            contEsf++;
        }

while (dEsfuerzos[marcEsf].getNombre().equals(esfuerzoActual) == false && marcEsf < contEsf)
marcEsf++;
    while (dHormigon[contHorm] != null) {
        contHorm++;
    }

while (dHormigon[marcHorm].getNombre().equals(hormigonSeleccionado) == false && marcHorm < contHorm)
marcHorm++;
    while (dAcero[contAce] != null) {
        contAce++;
    }

while (dAcero[marcAce].getNombre().equals(aceSeleccionado) == false && marcAce < contEsf)
marcAce++;
    while (dSuelo[contSuel] != null) {
        contSuel++;
    }

while (dSuelo[marcSuel].getNombre().equals(suelSeleccionado) == false && marcSuel < contSuel)
marcSuel++;
    while (dGeometria[contGeom] != null) {
        contGeom++;
    }

while (dGeometria[marcGeom].getNombre().equals(geomSeleccionado) == false && marcGeom < contGeom)
marcGeom++;
    if (dDimensiones[0] != null) {
        while (dDimensiones[contDim] != null) {
            contDim++;
        }
    }

while (dDimensiones[marcDim].getNombre().equals(dimeSeleccionado) == false && marcDim < contDim)
marcDim++;
    try {
        ComprobacionesConDimensiones comp =
ComprobacionesConDimensiones.calculo(dEsfuerzos[marcEsf].getN(),
dEsfuerzos[marcEsf].getV(), dEsfuerzos[marcEsf].getM(), dAcero[marcAce].isB400(),
dAcero[marcAce].isB500(),
        dAcero[marcAce].getFyk(), dAcero[marcAce].getEs(),
dAcero[marcAce].getGs(), dHormigon[marcHorm].getFck(), dHormigon[marcHorm].getGh(),
dHormigon[marcHorm].getGo(), dHormigon[marcHorm].getEc(),
        dGeometria[marcGeom].getR(), dGeometria[marcGeom].getP(),
dGeometria[marcGeom].getEn(), dGeometria[marcGeom].getB2(),
dGeometria[marcGeom].getA1(), dGeometria[marcGeom].getL(), dGeometria[marcGeom].getHt(),
        dSuelo[marcSuel].getSta(), dSuelo[marcSuel].getGt(),
dSuelo[marcSuel].getK03(), dSuelo[marcSuel].getCoh(), dSuelo[marcSuel].getGd(),
dSuelo[marcSuel].getGv(), dSuelo[marcSuel].getDel(), dSuelo[marcSuel].getGa(),
        dSuelo[marcSuel].isCoh(), dDimensiones[marcDim].getA2(),
dDimensiones[marcDim].getH(), dGeometria[marcGeom].getR1(), dGeometria[marcGeom].getDia0(),
dHormigon[marcHorm].getWkMax());
        ancho=(TextView) rootView.findViewById(R.id.ancho);
        canto=(TextView) rootView.findViewById(R.id.canto);
        d=(TextView) rootView.findViewById(R.id.d);
        At=(TextView) rootView.findViewById(R.id.At);
        At2=(TextView) rootView.findViewById(R.id.At2);
        Ar=(TextView) rootView.findViewById(R.id.Ar);
        Ar2=(TextView) rootView.findViewById(R.id.Ar2);
        NR=(TextView) rootView.findViewById(R.id.NR);
        NR2=(TextView) rootView.findViewById(R.id.NR2);
        Am=(TextView) rootView.findViewById(R.id.Am);
        Vuel=(TextView) rootView.findViewById(R.id.Vuel);
        V=(TextView) rootView.findViewById(R.id.V);
        //dia=(TextView) rootView.findViewById(R.id.dia);
        hund=(TextView) rootView.findViewById(R.id.hund);
        cortante=(TextView) rootView.findViewById(R.id.cortante);
        rigidez=(TextView) rootView.findViewById(R.id.rigidez);
        vuelco=(TextView) rootView.findViewById(R.id.vuelco);
        deslizamiento=(TextView) rootView.findViewById(R.id.deslizamiento);
        cortanteDef=(TextView) rootView.findViewById(R.id.cortanteDef);
    }

```

```

tipo=(TextView) rootView.findViewById(R.id.tipo);
NR=(TextView) rootView.findViewById(R.id.NR);
//dia=(TextView) rootView.findViewById(R.id.dia);
lb=(TextView) rootView.findViewById(R.id.lb);
ld=(TextView) rootView.findViewById(R.id.ld);
ln=(TextView) rootView.findViewById(R.id.ln);
l1=(TextView) rootView.findViewById(R.id.l1);
tipoAnclaje=(TextView) rootView.findViewById(R.id.tipoAnclaje);
zapata=(ImageView) rootView.findViewById(R.id.zapata);

anchos=comp.getAncho();
cantos=comp.getCanto();

areaTeorica=comp.getAreaTeorica();
areaTeoricaReparto=comp.getAreaTeoricaReparto();
areaRealReparto=comp.getAreaRealReparto();
areaReal=comp.getAreaReal();
tipoArmadur=comp.getTipoArmadura();
nRedondos=comp.getnRedondos();
nRedondosReparto=comp.getnRedondosReparto();
diametro=comp.getDiametro();
diametroReparto=comp.getDiametroReparto();
seguridadH=comp.getSeguridadH();
seguridadC=comp.getSeguridadC();
seguridadR=comp.getSeguridadR();
seguridadV=comp.getSeguridadV();
seguridadD=comp.getSeguridadD();
seguridadCDefinitiva=comp.getSeguridadCDefinitiva();
lnet=comp.getLnAnclaje();
lbas=comp.getLbAnclaje();
ldsip=comp.getLongitudDisponible();
L1=comp.getL1();
estAnclaje=comp.getEstdoAnclaje();
vuelo=comp.getVuelo();
D=comp.getD();
l1es=comp.getL1es();
l2es=comp.getL2es();
l3es=comp.getL3es();

ancho.setText(String.valueOf(anchos) + " m");
canto.setText(String.valueOf(cantos) + " m");
d.setText(String.valueOf(Math.ceil(D/0.001)*0.001)+" m");
At.setText(String.valueOf(red(areaTeorica, 2) + " cm^2"));
At2.setText(String.valueOf(red(areaTeoricaReparto, 2)+ " cm^2"));
Ar.setText(String.valueOf(red(areaReal, 2) + " cm^2"));
Ar2.setText(String.valueOf(red(areaRealReparto, 2)+" cm^2"));

if(comp.isB400())Am.setText(String.valueOf(red(cantos*100*comp.getB2()*100*0.002*0.5, 2)+
" cm^2"));

if(comp.isB500())Am.setText(String.valueOf(red(cantos*100*comp.getB2()*100*0.0018*0.5, 2)+
" cm^2"));

tipo.setText(tipoArmadur);
//dia.setText(String.valueOf(diametro) + " mm");
NR.setText(String.valueOf(nRedondos)+"Ø"+String.valueOf(diametro));

NR2.setText(String.valueOf(nRedondosReparto)+"Ø"+String.valueOf(diametroReparto));
hund.setText(String.valueOf(red(Math.abs(seguridadH), 2)));
cortante.setText(String.valueOf(red(Math.abs(seguridadC), 2)));
rigidez.setText(String.valueOf(red(Math.abs(seguridadR), 2)));
vuelco.setText(String.valueOf(red(Math.abs(seguridadV), 2)));
deslizamiento.setText(String.valueOf(red(Math.abs(seguridadD), 2)));
cortanteDef.setText(String.valueOf(red(Math.abs(seguridadCDefinitiva),
2)));

lb.setText(String.valueOf(red(lbas, 3) + " m");
ln.setText(String.valueOf(red(lnet, 3) + " m");
ld.setText(String.valueOf(red(ldsip, 3) + " m");
l1.setText(String.valueOf(red(L1, 3) + " m");
tipoAnclaje.setText(estAnclaje);
if(estAnclaje.equals("ANLCAJE EN PROLONGACIÓN RECTA"))
anclaje.setImageResource(R.drawable.anclaje_prolongacion);
if(estAnclaje.equals("ANLCAJE CON GANCHO O PATILLA"))
anclaje.setImageResource(R.drawable.anclaje_gancho);
if(estAnclaje.equals("ANLCAJE CON GANCHO O PATILLA PROLONGADA"))
anclaje.setImageResource(R.drawable.anclaje_gancho_prolongacion);
zapata.setImageResource(R.drawable.zapata_simetrica);

```

```

        if(comp.getEn()>0){
            zapata.setImageResource(R.drawable.zapata_simetrica_der);
            Vuel.setText("Vmax [m]:");
        }
        if(comp.getEn()<0){
            zapata.setImageResource(R.drawable.zapata_simetrica_izq);
            Vuel.setText("Vmax [m]:");
        }
        V.setText(String.valueOf(red(Math.abs(vuelo),2)+" m");
        L1es.setText(String.valueOf(red(l1es,3)+" m");
        L2es.setText(String.valueOf(red(l2es,3)+" m");
        L3es.setText(String.valueOf(red(l3es,3)+" m");
        if(tipoArmadur.equals("FLEXIBLE")){
            fisuracion.setText("Fisuración :");
            seguridadw.setText(String.valueOf(red(comp.getSeguridadW(),2)));
            if(comp.getSeguridadWW()!=0){
                seguridadw.setText(String.valueOf(comp.getSeguridadW()));
                Ar.setText(String.valueOf(comp.getSeguridadW()));
            }
        }
        NR.setText(String.valueOf(comp.getnRedondosw()+"Ø"+String.valueOf(diametro));
    }
} catch (Exception ex){
    Toast.makeText(getActivity(), " No se ha encontrado solución válida"
        , Toast.LENGTH_SHORT).show();
}
}

return rootView;
}

//Fuencion para redondear resultados
public static double red(double n, int i){
    int p=(int) Math.pow(10,i);
    return Math rint(n*p)/p;
}
}

```

-DinamicofragmentCalculoDimensionado.class

```
public class DinamicofragmentCalculoDimensionado extends FragmentoPadre{

    private static DatosEsfuerzosNominales[] dEsfuerzos=new DatosEsfuerzosNominales[50];
    private static String esfuerzoActual="Esfuerzos Ejemplos";
    private static DatosHormigon[] dHormigon=new DatosHormigon[50];
    private static String hormigonSeleccionado=" ";
    private static DatosAcero[] dAcero=new DatosAcero[50];
    private static String aceSeleccionado=" ";
    private static DatosSuelo[] dSuelo=new DatosSuelo[50];
    private static String suelSeleccionado=" ";
    private static DatosGeometria[] dGeometria=new DatosGeometria[50];
    private static String geomSeleccionado=" ";

    private TextView ancho;
    private TextView canto;
    private TextView d;
    private TextView At;
    private TextView At2;
    private TextView Ar;
    private TextView Ar2;
    private TextView Am;
    private TextView NR;
    private TextView NR2;
    private TextView Vuel;
    private TextView V;
    //private TextView dia;
    private TextView hund;
    private TextView cortante;
    private TextView rigidez;
    private TextView vuelco;
    private TextView deslizamiento;
    private TextView cortanteDef;
    private TextView tipo;
    private TextView lb;
    private TextView ln;
    private TextView ld;
    private TextView ll;
    private TextView tipoAnclaje;
    private TextView lles;
    private TextView L2es;
    private TextView L3es;
    private TextView fisuracion;
    private TextView seguridadw;
    private ImageView zapata;
    private ImageView anclaje;

    private int contEsf=0;
    private int marcEsf=0;
    private int contHorm=0;
    private int marcHorm=0;
    private int contAce=0;
    private int marcAce=0;
    private int contSuel=0;
    private int marcSuel=0;
    private int contGeom=0;
    private int marcGeom=0;

    private double[] anchos=new double[20];
    private double[] cantos=new double[20];
    private double[] areaTeorica=new double[20];
    private double[] D=new double[20];
    private double[] areaReal=new double[20];
    private int[] nRedondos=new int[20];
    private double[] diametro=new double[20];
    private double[] areaTeoricaReparto=new double[20];
    private double[] areaRealReparto=new double[20];
    private int[] nRedondosReparto=new int[20];
    private double[] diametroReparto=new double[20];
    private double[] vuelo=new double[20];

    private double[] lbas=new double[20];
    private double[] lnet=new double[20];
```

```

private double[] ldsip=new double[20];
private double[] l1=new double[20];
private String[] estAnclaje=new String[20];

private double[] seguridadH=new double[20];
private double[] seguridadC=new double[20];
private double[] seguridadR=new double[20];
private double[] seguridadV=new double[20];
private double[] seguridadD=new double[20];
private double[] seguridadCDefinitiva=new double[20];

private int[] marcadorValidos=new int[20];
private int contMarcVald=0;
private int mostrado=0;
private double l1es=0,l2es=0,l3es=0;

private String[] tipoArmadur=new String[20];

public static DinamicofragmentCalculoDimensionado
newInstance(DatosEsfuerzosNominales[] dEsf, String esf, DatosHormigon[] dHorm,
String horm,
DatosAcero[] dAce,String ace, DatosSuelo[] dSue,String sue,
DatosGeometria[]
dGeom,String geo){
    DinamicofragmentCalculoDimensionado f=new DinamicofragmentCalculoDimensionado();
    dEsfuerzos=dEsf;
    dHormigon=dHorm;
    dAcero=dAce;
    dSuelo=dSue;
    dGeometria=dGeom;
    esfuerzoActual=esf;
    hormigonSeleccionado=horm;
    aceSeleccionado=ace;
    suelSeleccionado=sue;
    geomSeleccionado=geo;

    return f;
}

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
    View
    rootView=inflater.inflate(R.layout.dinamico_fragment_calculos_dimensionado,container,false);

    ancho=(TextView) rootView.findViewById(R.id.ancho);
    canto=(TextView) rootView.findViewById(R.id.canto);
    d=(TextView) rootView.findViewById(R.id.d);
    At=(TextView) rootView.findViewById(R.id.At);
    At2=(TextView) rootView.findViewById(R.id.At2);
    Ar=(TextView) rootView.findViewById(R.id.Ar);
    Ar2=(TextView) rootView.findViewById(R.id.Ar2);
    NR=(TextView) rootView.findViewById(R.id.NR);
    NR2=(TextView) rootView.findViewById(R.id.NR2);
    Am=(TextView) rootView.findViewById(R.id.Am);
    Vuel=(TextView) rootView.findViewById(R.id.Vuel);
    V=(TextView) rootView.findViewById(R.id.V);
    //dia=(TextView) rootView.findViewById(R.id.dia);
    hund=(TextView) rootView.findViewById(R.id.hund);
    cortante=(TextView) rootView.findViewById(R.id.cortante);
    rigidez=(TextView) rootView.findViewById(R.id.rigidez);
    vuelco=(TextView) rootView.findViewById(R.id.vuelco);
    deslizamiento=(TextView) rootView.findViewById(R.id.deslizamiento);
    cortanteDef=(TextView) rootView.findViewById(R.id.cortanteDef);
    tipo=(TextView) rootView.findViewById(R.id.tipo);
    NR=(TextView) rootView.findViewById(R.id.NR);
    //dia=(TextView) rootView.findViewById(R.id.dia);
    lb=(TextView) rootView.findViewById(R.id.lb);
    ld=(TextView) rootView.findViewById(R.id.ld);
    ln=(TextView) rootView.findViewById(R.id.ln);
    ll=(TextView) rootView.findViewById(R.id.ll);

```

```

    tipoAnclaje=(TextView) rootView.findViewById(R.id.tipoAnclaje);
    zapata=(ImageView) rootView.findViewById(R.id.zapata);
    anclaje=(ImageView) rootView.findViewById(R.id.anclaje);
    L1es=(TextView) rootView.findViewById(R.id.L1es);
    L2es=(TextView) rootView.findViewById(R.id.L2es);
    L3es=(TextView) rootView.findViewById(R.id.L3es);
    fisuracion=(TextView) rootView.findViewById(R.id.fisuracion);
    seguridadw=(TextView) rootView.findViewById(R.id.segF);
    while(dEsfuerzos[contEsf]!=null){
        contEsf++;
    }

while(dEsfuerzos[marcEsf].getNombre().equals(esfuerzoActual)==false&&marcEsf<contEsf)
marcEsf++;
    while(dHormigon[contHorm]!=null){
        contHorm++;
    }

while(dHormigon[marcHorm].getNombre().equals(hormigonSeleccionado)==false&&marcHorm<contHorm)
marcHorm++;
    while(dAcero[contAce]!=null){
        contAce++;
    }

while(dAcero[marcAce].getNombre().equals(aceSeleccionado)==false&&marcAce<contEsf)
marcAce++;
    while(dSuelo[contSuel]!=null){
        contSuel++;
    }

while(dSuelo[marcSuel].getNombre().equals(suelSeleccionado)==false&&marcSuel<contSuel)
marcSuel++;
    while(dGeometria[contGeom]!=null){
        contGeom++;
    }

while(dGeometria[marcGeom].getNombre().equals(geomSeleccionado)==false&&marcGeom<contGeom)
) marcGeom++;
    try {
        Comprobaciones comp = Comprobaciones.calculo(dEsfuerzos[marcEsf].getN(),
dEsfuerzos[marcEsf].getV(), dEsfuerzos[marcEsf].getM(), dAcero[marcAce].isB400(),
dAcero[marcAce].isB500(),
        dAcero[marcAce].getFyk(), dAcero[marcAce].getEs(),
dAcero[marcAce].getGs(), dHormigon[marcHorm].getFck(), dHormigon[marcHorm].getGh(),
dHormigon[marcHorm].getGo(), dHormigon[marcHorm].getEc(),
        dGeometria[marcGeom].getR(), dGeometria[marcGeom].getP(),
dGeometria[marcGeom].getEn(), dGeometria[marcGeom].getB2(),
dGeometria[marcGeom].getAl(), dGeometria[marcGeom].getL(), dGeometria[marcGeom].getHt(),
        dSuelo[marcSuel].getSta(), dSuelo[marcSuel].getGt(),
dSuelo[marcSuel].getK03(), dSuelo[marcSuel].getCoh(), dSuelo[marcSuel].getGd(),
dSuelo[marcSuel].getGv(), dSuelo[marcSuel].getDel(), dSuelo[marcSuel].getGa(),

dSuelo[marcSuel].isCoh(), dGeometria[marcGeom].getRl(), dGeometria[marcGeom].getDia0(), dHormigon[marcHorm].getWkMax());
        anchos=comp.getAncho();
        cantos=comp.getCanto();
        marcadorValidos=comp.getMarcadorValidos();
        contMarcVald=comp.getContMarcVald();
        areaTeorica=comp.getAreaTeorica();
        areaTeoricaReparto=comp.getAreaTeoricaReparto();
        areaRealReparto=comp.getAreaRealReparto();
        areaReal=comp.getAreaReal();
        tipoArmatur=comp.getTipoArmadura();
        nRedondos=comp.getnRedondos();
        nRedondosReparto=comp.getnRedondosReparto();
        diametro=comp.getDiametro();
        diametroReparto=comp.getDiametroReparto();
        seguridadH=comp.getSeguridadH();
        seguridadC=comp.getSeguridadC();
        seguridadR=comp.getSeguridadR();
        seguridadV=comp.getSeguridadV();
        seguridadD=comp.getSeguridadD();
        seguridadCDefinitiva=comp.getSeguridadCDefinitiva();
        lnet=comp.getLnAnclaje();
        lbas=comp.getLbAnclaje();
        ldsip=comp.getLongitudDisponible();
        Ll=comp.getLl();
    }

```



```

        estAnclaje=comp.getEstdoAnclaje();
        vuelo=comp.getVuelo();
        D=comp.getD();
        l1es=comp.getL1es();
        l2es=comp.getL2es();
        l3es=comp.getL3es();

        ancho.setText(String.valueOf(anchos[marcadorValidos[mostrado]])+" m");

        canto.setText(String.valueOf(red(Math.ceil(cantos[marcadorValidos[mostrado]]/0.1)*0.1,1)
)+" m");

        d.setText(String.valueOf(Math.ceil(D[marcadorValidos[mostrado]]/0.001)*0.001+" m"));
        At.setText(String.valueOf(red(areaTeorica[marcadorValidos[mostrado]],2)+"
cm^2"));

        At2.setText(String.valueOf(red(areaTeoricaReparto[marcadorValidos[mostrado]],2)+"
cm^2"));
        Ar.setText(String.valueOf(red(areaReal[marcadorValidos[mostrado]],2)+"
cm^2"));

        Ar2.setText(String.valueOf(red(areaRealReparto[marcadorValidos[mostrado]],2)+" cm^2"));

        if(comp.isB400())Am.setText(String.valueOf(red(cantos[marcadorValidos[mostrado]]*100*com
p.getB2()*100*0.002*0.5,2)+" cm^2"));

        if(comp.isB500())Am.setText(String.valueOf(red(cantos[marcadorValidos[mostrado]]*100*com
p.getB2()*100*0.0018*0.5,2)+" cm^2"));
        tipo.setText(tipoArmadur[marcadorValidos[mostrado]]);
        //dia.setText(String.valueOf(diametro[marcadorValidos[mostrado]]+" mm");

        NR.setText(String.valueOf(nRedondos[marcadorValidos[mostrado]])+"Ø"+String.valueOf(diame
tro[marcadorValidos[mostrado]]));

        NR2.setText(String.valueOf(nRedondosReparto[marcadorValidos[mostrado]])+"Ø"+String.value
Of(diametroReparto[marcadorValidos[mostrado]]));

        hund.setText(String.valueOf(red(Math.abs(seguridadH[marcadorValidos[mostrado]]),2));

        cortante.setText(String.valueOf(red(Math.abs(seguridadC[marcadorValidos[mostrado]]),2));
;

        rigidez.setText(String.valueOf(red(Math.abs(seguridadR[marcadorValidos[mostrado]]),2));

        vuelco.setText(String.valueOf(red(Math.abs(seguridadV[marcadorValidos[mostrado]]),2));

        deslizamiento.setText(String.valueOf(red(Math.abs(seguridadD[marcadorValidos[mostrado]]
,2)));

        cortanteDef.setText(String.valueOf(red(Math.abs(seguridadCDefinitiva[marcadorValidos[mos
trado]],2)));
        lb.setText(String.valueOf(red(lbas[marcadorValidos[mostrado]],3)+" m");
        ln.setText(String.valueOf(red(lnet[marcadorValidos[mostrado]],3)+" m");
        ld.setText(String.valueOf(red(ldsip[marcadorValidos[mostrado]],3)+" m");
        l1.setText(String.valueOf(red(L1[marcadorValidos[mostrado]],3)+" m");
        l1es.setText(String.valueOf(red(l1es,3)+" m");
        l2es.setText(String.valueOf(red(l2es,3)+" m");
        l3es.setText(String.valueOf(red(l3es,3)+" m");
        tipoAnclaje.setText(estAnclaje[marcadorValidos[mostrado]]);
        if(estAnclaje[marcadorValidos[mostrado]].equals("ANLCAJE EN PROLONGACIÓN
RECTA")) anclaje.setImageResource(R.drawable.anclaje_prolongacion);
        if(estAnclaje[marcadorValidos[mostrado]].equals("ANLCAJE CON GANCHO O
PATILLA")) anclaje.setImageResource(R.drawable.anclaje_gancho);
        if(estAnclaje[marcadorValidos[mostrado]].equals("ANLCAJE CON GANCHO O
PATILLA PROLONGADA")) anclaje.setImageResource(R.drawable.anclaje_gancho_prolongacion);
        zapata.setImageResource(R.drawable.zapata_simetrica);
        if(comp.getEn()>0){
            zapata.setImageResource(R.drawable.zapata_simetrica_der);
            Vucl.setText("Vmax [m]:");
        }
        if(comp.getEn()<0){
            zapata.setImageResource(R.drawable.zapata_simetrica_izq);
            Vucl.setText("Vmax [m]:");
        }
        V.setText(String.valueOf(red(Math.abs(vuelo[marcadorValidos[mostrado]]),2)+" m");

```

```

        if(tipoArmadur[marcadorValidos[mostrado]].equals("FLEXIBLE")){
            fisuracion.setText("Fisuración :");
        }

        seguridadw.setText(String.valueOf(red(comp.getSeguridadW()[marcadorValidos[mostrado]],2)
    ));
            if(comp.getSeguridadWW()[marcadorValidos[mostrado]]!=0){
                seguridadw.setText(String.valueOf(comp.getSeguridadW()[marcadorValidos[mostrado]]));
            }

        Ar.setText(String.valueOf(comp.getSeguridadW()[marcadorValidos[mostrado]]));

        NR.setText(String.valueOf(comp.getnRedondosW()[marcadorValidos[mostrado]]+"Ø"+String.va
lueOf(diametro[marcadorValidos[mostrado]]));
    }
}

} catch (Exception ex) {
    Toast.makeText(getActivity(), " No se ha encontrado solución válida"
        , Toast.LENGTH_SHORT).show();
}

return rootView;
}

//Fuencion para redondear resultados
public static double red(double n, int i){
    int p=(int) Math.pow(10,i);
    return Math rint (n*p)/p;
}
}

```

-DinamicoFragmentComprobaciones.class

```
public class DinamicoFragmentComprobaciones extends FragmentoPadre{

    private FragmentoPadre mListener;

    private DatosEsfuerzosNominales[] dEsfuerzos=new DatosEsfuerzosNominales[50];
    private String esfuerzoSeleccionado=" ";
    private DatosHormigon[] dHormigon=new DatosHormigon[50];
    private String hormigonSeleccionado=" ";
    private DatosAcero[] dAcero=new DatosAcero[50];
    private String aceroSeleccionado=" ";
    private DatosSuelo[] dSuelo=new DatosSuelo[50];
    private String sueloSeleccionado=" ";
    private DatosGeometria[] dGeometria=new DatosGeometria[50];
    private String geomSeleccionado=" ";
    private DatosDimensiones[] dDimensiones=new DatosDimensiones[50];
    private String dimSeleccionado=" ";

    private int marcEsf=0;
    private int conEsf=0;
    private int marcHorm=0;
    private int conHorm=0;
    private int marcAce=0;
    private int conAce=0;
    private int marcSuel=0;
    private int conSuel=0;
    private int marcGeom=0;
    private int conGeom=0;
    private int marcDim=0;
    private int conDim=0;

    private String acero="";

    private TextView esfN;
    private TextView horm;
    private TextView ace;
    private TextView suel;
    private TextView geom;

    private EditText a2;
    private EditText h;

    private ListView dimensionesLW;
    private ArrayAdapter dimensionesLWAdapter=null;

    public static DinamicoFragmentComprobaciones newInstance (DatosEsfuerzosNominales[]
dEsf, String esfuerzoGuardado,
                                DatosHormigon[] dHorm,String
hormGuard,
                                DatosAcero[] dAce,String
aceroGuard,
                                DatosSuelo[] dSue, String
sueloGuard,
                                DatosGeometria[] dGeom, String
geomGuard,
                                DatosDimensiones[] dDim,String
dimGuard){
        DinamicoFragmentComprobaciones f=new DinamicoFragmentComprobaciones ();
        f.dEsfuerzos=dEsf;
        f.esfuerzoSeleccionado=esfuerzoGuardado;
        while (f.dEsfuerzos[f.conEsf]!=null) f.conEsf++;
        while (!f.dEsfuerzos[f.marcEsf].getNombre().equals(f.esfuerzoSeleccionado) &&
f.marcEsf<f.conEsf) f.marcEsf++;
        f.dHormigon=dHorm;
        f.hormigonSeleccionado=hormGuard;
        while (f.dHormigon[f.conHorm]!=null) f.conHorm++;
        while (!f.dHormigon[f.marcHorm].getNombre().equals(f.hormigonSeleccionado) &&
f.marcHorm<f.conHorm) f.marcHorm++;
        f.dAcero=dAce;
        f.aceroSeleccionado=aceroGuard;
        while (f.dAcero[f.conAce]!=null) f.conAce++;
        while (!f.dAcero[f.marcAce].getNombre().equals(f.aceroSeleccionado) &&
f.marcAce<f.conAce) f.marcAce++;
```

```

        f.dSuelo=dSue;
        f.sueloSeleccionado=sueloGuard;
        while (f.dSuelo[f.conSuel] != null) f.conSuel++;
        while (!f.dSuelo[f.marcSuel].getNombre().equals(f.sueloSeleccionado) &&
f.marcSuel < f.conSuel) f.marcSuel++;
        f.dGeometria=dGeom;
        f.geomSeleccionado=geomGuard;
        while (f.dGeometria[f.conGeom] != null) f.conGeom++;
        while (!f.dGeometria[f.marcGeom].getNombre().equals(f.geomSeleccionado) &&
f.marcGeom < f.conGeom) f.marcGeom++;
        f.dDimensiones=dDim;
        f.dimSeleccionado=dimGuard;
        if (f.dDimensiones[0] != null) {
            while (f.dDimensiones[f.conDim] != null) f.conDim++;
            while (!f.dDimensiones[f.marcDim].getNombre().equals(f.dimSeleccionado) &&
f.marcDim < f.conDim)
                f.marcDim++;
        }
        return f;
    }

    @Override
    public void onAttach(Activity activity) {
        super.onAttach(activity);
        try{
            mListener = (FragmentPadre) activity;
        } catch (ClassCastException e){
            throw new ClassCastException(activity.toString());
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {

        View
        rootView=inflater.inflate(R.layout.dinamico_fragment_comprobacion, container, false);

        esfN=(TextView) rootView.findViewById(R.id.esfN);
        if (dEsfuerzos[0] != null) {
            esfN.setText(esfuerzoSeleccionado);
            esfN.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    Toast.makeText(getActivity(), "Esfuerzo Cargado:" + "\nDatos:\nN : "
                        +String.valueOf(dEsfuerzos[marcEsf].getN())+" N\nV
:"+
                        String.valueOf(dEsfuerzos[marcEsf].getV())+" N\nM
:"+
                        String.valueOf(dEsfuerzos[marcEsf].getM())+" N*m"
                        , Toast.LENGTH_SHORT).show();
                }
            });
        }

        horm=(TextView) rootView.findViewById(R.id.horm);
        if (dHormigon[0] != null) {
            horm.setText(hormigonSeleccionado);
            horm.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    Toast.makeText(getActivity(), "Hromigon Cargado:\n"+ String.valueOf(
                        dHormigon[marcHorm].getNombre())+"\nDatos:\nGh : "
                        +String.valueOf(dHormigon[marcHorm].getGh())+"
N/m^3\nEc : "+
String.valueOf(dHormigon[marcHorm].getEc()/1000000)+" Mpa\nfck : "+
String.valueOf(dHormigon[marcHorm].getFck()/1000000)+" Mpa\nGo : "+
                        String.valueOf(dHormigon[marcHorm].getGo())+" "
                        , Toast.LENGTH_SHORT).show();
                }
            });
        }

        ace=(TextView) rootView.findViewById(R.id.ace);
        if (dAcero[0] != null) {
            ace.setText(aceroSeleccionado);

```

```

ace.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(dAcero[marcAce].isB400()) acero="B-400";
        if(dAcero[marcAce].isB500()) acero="B-500";
        Toast.makeText(getActivity(), "Acero Cargado:"+"\nDatos:\nTipo de
acero :"+
                                acero+" \nfyk :"+
                                +String.valueOf(dAcero[marcAce].getFyk()/1000000)+"
MPa\nE :"+
                                String.valueOf(dAcero[marcAce].getEs()/1000000)+"
MPa\nGs :"+
                                String.valueOf(dAcero[marcAce].getGs())+" "
                                , Toast.LENGTH_SHORT).show();
    }
});

suel=(TextView) rootView.findViewById(R.id.suel);
if(dSuelo[0]!=null){
    suel.setText(sueloSeleccionado);
    suel.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText(getActivity(), "Suelo Cargado:"+" \nDatos:\nSta :"+
                                +String.valueOf(dSuelo[marcAce].getSta())+" N\nGt"+
                                String.valueOf(dSuelo[marcAce].getGt())+" N/m^3\ncoh
:"+
                                String.valueOf(dSuelo[marcAce].getCoh())+"\nGd :"+
                                String.valueOf(dSuelo[marcAce].getGd())+"\nK03 :"+
                                String.valueOf(dSuelo[marcAce].getK03()/1000000)+"
MN/m^3\nDelta :"+
                                String.valueOf(dSuelo[marcAce].getDel())+"\nGa :"+
                                String.valueOf(dSuelo[marcAce].getGa())+"\nGv :"+
                                String.valueOf(dSuelo[marcAce].getGv())+" "
                                , Toast.LENGTH_SHORT).show();
        }
    });
}

geom=(TextView) rootView.findViewById(R.id.geom);
if(dGeometria[0]!=null){
    geom.setText(geomSeleccionado);
    geom.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText(getActivity(), "Geometría Cargado:"+" \nDatos:\nr1 :"+
                                +String.valueOf(dGeometria[marcGeom].getR1())+"
m\nr2 :"+
                                String.valueOf(dGeometria[marcGeom].getR1())+" m\nØ0
:"+
                                String.valueOf(dGeometria[marcGeom].getDia0())+"
mm\nen :"+
                                String.valueOf(dGeometria[marcGeom].getEn())+" m\nb2
:"+
                                String.valueOf(dGeometria[marcGeom].getB2())+" m\na1
:"+
                                String.valueOf(dGeometria[marcGeom].getA1())+" m\nl
:"+
                                String.valueOf(dGeometria[marcGeom].getL())+" m\nht
:"+
                                String.valueOf(dGeometria[marcGeom].getHt())+"
m\np(muro) :"+
                                String.valueOf(dGeometria[marcGeom].getP())
                                , Toast.LENGTH_SHORT).show();
        }
    });
}

a2=(EditText) rootView.findViewById(R.id.a2);
h=(EditText) rootView.findViewById(R.id.h);

Button irNuevosDatos=(Button) rootView.findViewById(R.id.irNuevosDatos);
irNuevosDatos.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        mListener.onClickCambiarFragmentDinamico(MainActivity.FLAG_FRAGMENT_G);
    }
});

```

```

    });

    Button guardarDimensiones=(Button)
    rootView.findViewById(R.id.guardarDimensiones);
    guardarDimensiones.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            try {
                conDim=0;
                while(dDimensiones[conDim]!=null){
                    conDim++;
                }
                if(dDimensiones[conDim]==null){
                    dDimensiones[conDim] = new DatosDimensiones();
                    dDimensiones[conDim].setNombre("Dimensiones
"+String.valueOf(conDim+1));
                    dDimensiones[conDim].setA2(Double.parseDouble("0" +
a2.getText().toString().replaceAll(" ", "")));
                    dDimensiones[conDim].setH(Double.parseDouble("0" +
h.getText().toString().replaceAll(" ", "")));
                    a2.setText("");
                    h.setText("");
                    if(conDim==0) {
                        dimSeleccionado=dDimensiones[conDim].getNombre();
                        mListener.pasaDimensionesActual(dimSeleccionado);
                    }
                    mListener.pasaDatos(dDimensiones);
                    putItemsLW(dDimensiones, conDim+1);
                }
            } catch (NumberFormatException e) {
                Context context = getActivity();
                CharSequence text = "Fallo al introducir\n los datos";
                int duration = Toast.LENGTH_SHORT;

                Toast toast = Toast.makeText(context, text, duration);
                toast.show();
            }
        }
    });

    dimensionesLW= (ListView) rootView.findViewById(R.id.dimensionesLW);
    dimensionesLW.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
            dimSeleccionado=String.valueOf(parent.getItemAtPosition(position));
            marcDim=0;
            conDim=0;
            while(dDimensiones[conDim]!=null){
                conDim++;
            }

            while(dDimensiones[marcDim].getNombre().equals(dimSeleccionado)==false&&marcDim<conDim)
            marcDim++;

            mListener.pasaDimensionesActual(dimSeleccionado);
            Toast.makeText(getActivity(), "Dimensiones Cargadas:"+"\nDatos:\na2 : "
: "+"
                +String.valueOf(dDimensiones[marcDim].getA2())+" m\nh
                String.valueOf(dDimensiones[marcDim].getH())+" m"
                , Toast.LENGTH_SHORT).show();
        }
    });
    putItemsLW(dDimensiones, conDim);

    Button irCalculosDim=(Button) rootView.findViewById(R.id.irCalculosDim);
    irCalculosDim.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            mListener.onClickCambiarFragmentDinamico(MainActivity.FLAG_FRAGMENT_K);
        }
    });

    return rootView;
}

```

```
public void putItemsLW(DatosDimensiones[] items, int i){  
    String[] meme=new String[i];  
    for(int v=0; v<i; v++){  
        meme[v]=items[v].getNombre();  
    }  
  
    dimensionesLWAdapter= new ArrayAdapter(getActivity(),  
android.R.layout.simple_list_item_1, meme);  
    dimensionesLW.setAdapter(dimensionesLWAdapter);  
}  
}
```

-DinamicoFragmentDatosDim.class

```
public class DinamicoFragmentDatosDim extends FragmentoPadre{

    private FragmentoPadre mListener;

    private DatosEsfuerzosNominales[] dEsfuerzos=new DatosEsfuerzosNominales[50];
    private String esfuerzoSeleccionado=" ";
    private DatosHormigon[] dHormigon=new DatosHormigon[50];
    private String hormigonSeleccionado=" ";
    private DatosAcero[] dAcero=new DatosAcero[50];
    private String aceroSeleccionado=" ";
    private DatosSuelo[] dSuelo=new DatosSuelo[50];
    private String sueloSeleccionado=" ";
    private DatosGeometria[] dGeometria=new DatosGeometria[50];
    private String geomSeleccionado=" ";

    private static String acero="";

    private int marcEsf=0;
    private int conEsf=0;
    private int marcHorm=0;
    private int conHorm=0;
    private int marcAce=0;
    private int conAce=0;
    private int marcSuel=0;
    private int conSuel=0;
    private int marcGeom=0;
    private int conGeom=0;

    private TextView esfN;
    private TextView horm;
    private TextView ace;
    private TextView suel;
    private TextView geom;

    public static DinamicoFragmentDatosDim newInstance(DatosEsfuerzosNominales[] dEsf,
String esfuerzoGuardado,
                                                    DatosHormigon[] dHorm,String
hormGuardado,
                                                    DatosAcero[] dAce,String
aceroGuardado,
                                                    DatosSuelo[] dSue, String
sueloGuardado,
                                                    DatosGeometria[] dGeom, String
geomGuardado){
        DinamicoFragmentDatosDim f=new DinamicoFragmentDatosDim();
        f.dEsfuerzos=dEsf;
        f.esfuerzoSeleccionado=esfuerzoGuardado;
        while (f.dEsfuerzos[f.conEsf]!=null) f.conEsf++;
        while (!f.dEsfuerzos[f.marcEsf].getNombre().equals(f.esfuerzoSeleccionado) &&
f.marcEsf<f.conEsf) f.marcEsf++;
        f.dHormigon=dHorm;
        f.hormigonSeleccionado=hormGuardado;
        while (f.dHormigon[f.conHorm]!=null) f.conHorm++;
        while (!f.dHormigon[f.marcHorm].getNombre().equals(f.hormigonSeleccionado) &&
f.marcHorm<f.conHorm) f.marcHorm++;
        f.dAcero=dAce;
        f.aceroSeleccionado=aceroGuardado;
        while (f.dAcero[f.conAce]!=null) f.conAce++;
        while (!f.dAcero[f.marcAce].getNombre().equals(f.aceroSeleccionado) &&
f.marcAce<f.conAce) f.marcAce++;
        f.dSuelo=dSue;
        f.sueloSeleccionado=sueloGuardado;
        while (f.dSuelo[f.conSuel]!=null) f.conSuel++;
        while (!f.dSuelo[f.marcSuel].getNombre().equals(f.sueloSeleccionado) &&
f.marcSuel<f.conSuel) f.marcSuel++;
        f.dGeometria=dGeom;
        f.geomSeleccionado=geomGuardado;
        while (f.dGeometria[f.conGeom]!=null) f.conGeom++;
        while (!f.dGeometria[f.marcGeom].getNombre().equals(f.geomSeleccionado) &&
f.marcGeom<f.conGeom) f.marcGeom++;
        return f;
    }
}
```



```

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);
    try{
        mListener = (FragmentPadre) activity;
    }catch (ClassCastException e){
        throw new ClassCastException(activity.toString());
    }
}
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

    View
    rootView=inflater.inflate(R.layout.dinamico_fragment_dimensionado,container,false);

    esfN=(TextView) rootView.findViewById(R.id.esfN);
    if(dEsfuerzos[0]!=null){
        esfN.setText(esfuerzoSeleccionado);
        esfN.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(getActivity(), "Esfuerzo Cargado:"+" \nDatos:\nN :"+
                    String.valueOf(dEsfuerzos[marcEsf].getN())+" N\nV
:"+
                    String.valueOf(dEsfuerzos[marcEsf].getV())+" N\nM
:"+
                    String.valueOf(dEsfuerzos[marcEsf].getM())+" N*m"
                    , Toast.LENGTH_SHORT).show();
            }
        });
    }

    horm=(TextView)rootView.findViewById(R.id.horm);
    if(dHormigon[0]!=null){
        horm.setText(hormigonSeleccionado);
        horm.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(getActivity(), "Hromigon Cargado:\n"+ String.valueOf(
                    dHormigon[marcHorm].getNombre())+" \nDatos:\nGh :"+
                    String.valueOf(dHormigon[marcHorm].getGh())+"
N/m^3\nEc :"+
                    String.valueOf(dHormigon[marcHorm].getEc()/1000000)+" Mpa\nfck :"+
                    String.valueOf(dHormigon[marcHorm].getFck()/1000000)+" Mpa\nGo :"+
                    String.valueOf(dHormigon[marcHorm].getGo())+" "
                    , Toast.LENGTH_SHORT).show();
            }
        });
    }

    ace=(TextView)rootView.findViewById(R.id.ace);
    if(dAcero[0]!=null){
        ace.setText(aceroSeleccionado);
        ace.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(dAcero[marcAce].isB400())acero="B-400";
                if(dAcero[marcAce].isB500())acero="B-500";
                Toast.makeText(getActivity(), "Acero Cargado:"+" \nDatos:\nTipo de
acero :"+
                    acero+" \nfyk :"+
                    String.valueOf(dAcero[marcAce].getFyk()/1000000)+"
MPa\nE :"+
                    String.valueOf(dAcero[marcAce].getEs()/1000000)+"
MPa\nGs :"+
                    String.valueOf(dAcero[marcAce].getGs())+" "
                    , Toast.LENGTH_SHORT).show();
            }
        });
    }

    suel=(TextView)rootView.findViewById(R.id.suel);
    if(dSuelo[0]!=null){

```

```

        suel.setText(sueloSeleccionado);
        suel.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(getActivity(), "Suelo Cargado:" + "\nDatos:\nSta :"+
                    +String.valueOf(dSuelo[marcSuel].getSta())+" N\nGt"+
                    String.valueOf(dSuelo[marcSuel].getGt())+"
N/m^3\ncoh :"+
                    String.valueOf(dSuelo[marcSuel].getCoh())+"\nGd :"+
                    String.valueOf(dSuelo[marcSuel].getGd())+" \nK03 :"+
                    String.valueOf(dSuelo[marcSuel].getK03()/1000000)+"
MN/m^3\nDelta :"+
                    String.valueOf(dSuelo[marcSuel].getDel())+"\nGa :"+
                    String.valueOf(dSuelo[marcSuel].getGa())+"\nGv :"+
                    String.valueOf(dSuelo[marcSuel].getGv())+"",
                    Toast.LENGTH_SHORT).show();
            }
        });
    }

    geom=(TextView) rootView.findViewById(R.id.geom);
    if(dGeometria[0]!=null){
        geom.setText(geomSeleccionado);
        geom.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(getActivity(), "Geometría Cargado:" + "\nDatos:\nr1 :"+
                    +String.valueOf(dGeometria[marcGeom].getR1())+"
m\nr2 :"+
                    String.valueOf(dGeometria[marcGeom].getR1())+" m\nØ0
:"+
                    String.valueOf(dGeometria[marcGeom].getDia0())+"
mm\nen :"+
                    String.valueOf(dGeometria[marcGeom].getEn())+" m\nb2
:"+
                    String.valueOf(dGeometria[marcGeom].getB2())+" m\na1
:"+
                    String.valueOf(dGeometria[marcGeom].getA1())+" m\nl
:"+
                    String.valueOf(dGeometria[marcGeom].getL())+" m\nht
:"+
                    String.valueOf(dGeometria[marcGeom].getHt())+"
m\np(muro) :"+
                    String.valueOf(dGeometria[marcGeom].getP())
                    , Toast.LENGTH_SHORT).show();
            }
        });
    }

    Button irNuevosDatos=(Button) rootView.findViewById(R.id.irNuevosDatos);
    irNuevosDatos.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            mListener.onClickCambiarFragmentDinamico(MainActivity.FLAG_FRAGMENT_G);
        }
    });
    Button irCalculosDim=(Button) rootView.findViewById(R.id.irCalculosDim);
    irCalculosDim.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            mListener.onClickCambiarFragmentDinamico(MainActivity.FLAG_FRAGMENT_J);
        }
    });
    return rootView;
}
}

```

-DinamicoFragmentGrafico.class

```
public class DinamicoFragmentGrafico extends FragmentoPadre {

    private XAxis xAxis;
    private YAxis yAxis;

    private ArrayList<Entry> recta = new ArrayList<Entry>();
    private ArrayList<Entry> curva = new ArrayList<Entry>();
    private ArrayList<Entry> curva2 = new ArrayList<Entry>();
    private ArrayList<Entry> limiteI ;
    private ArrayList<Entry> limiteS ;
    private ArrayList<String> xVals = new ArrayList<String>();

    private ArrayList<ILineDataSet> dataSets = new ArrayList<ILineDataSet>();

    private LineChart chart;

    private LineDataSet setPuntos;
    private LineDataSet setCompLimitS;
    private LineDataSet setCompLimitI;
    private LineData data;

    private boolean comp=false;
    private boolean comp2=false;

    private boolean limitI=false;
    private boolean limitS=false;

    private static DatosArmadura dArmadura=new DatosArmadura();
    private static DatosMateriales dMateriales=new DatosMateriales();
    private static DatosEsfuerzos[] dEsfuerzos=new DatosEsfuerzos[50];

    private final double x0=0;
    private final double x05=0.05;
    private final double x1=0.08854;
    private final double x2=0.159;
    private double x3=0.3229;
    private double x4=0.3319;
    private final double x5=0.40185;
    private double x6=0.408;
    private double x7=0.42327;
    private final double y0=0;
    private final double y05=0.05;
    private final double y1=0.0944;
    private final double y2=0.17839;
    private double y3=0.43255;
    private double y4=0.45961;
    private final double y5=0.6881;
    private double y6=0.73202;
    private double y7=0.9042;

    private ListView limitesLW;

    private ArrayAdapter datsLWAdpater=null;

    private double yMax=1.3, xMax=0.8, mulMax=0, nuMax=0;

    private int contPuntos=0;

    private FragmentPadre mListener;

    private TresFuncion curvaHormigonX1=new TresFuncion() {
        @Override
        public double eval(double pln, double delta) {
            return (17*(4-12*pln+3*pln*pln)*pln*pln)/(16*(1-pln)*(1-pln));
        }
    };
    private TresFuncion curvaHormigonX2=new TresFuncion() {
        @Override
        public double eval(double pln, double delta) {
            return (((-1)*(17)*(7-114*pln+57*pln*pln))/2000);
        }
    };
};
```

```

private TresFuncion curvaHormigonX3=new TresFuncion() {
    @Override
    public double eval(double pln, double delta) {
        return ((17*pln*(238-99*pln))/5880);
    }
};
private TresFuncion curvaHormigonX4=new TresFuncion() {
    @Override
    public double eval(double pln, double delta) {
        return ((17)* (147*delta*(2-delta)+(128*delta*delta*delta*(6*delta-7))
        /((7*pln-3*delta)*(7*pln-3*delta)))/5880);
    }
};
private TresFuncion curvaHormigonY1=new TresFuncion(){
    @Override
    public double eval(double pln, double delta) {
        return (17 * (3 - 8 * pln) * pln * pln) / (12 * (1 - pln) * (1 - pln));
    }
};
private TresFuncion curvaHormigonY2=new TresFuncion(){
    @Override
    public double eval(double pln, double delta) {
        return ((272 * pln - 17) / 300);
    }
};
private TresFuncion curvaHormigonY3=new TresFuncion(){
    @Override
    public double eval(double pln, double delta) {
        return (289 * pln) / 420;
    }
};
private TresFuncion curvaHormigonY4=new TresFuncion(){
    @Override
    public double eval(double pln, double delta) {
        return ((17 * (delta / 420)) * ((1029 * pln * pln - 882 * pln * delta + 125
* delta * delta)
        / ((7 * pln - 3 * delta) * (7 * pln - 3 * delta) ));
    }
};
private UnaFuncion interpolacionLagrange=new UnaFuncion() {
    @Override
    public double eval(double x) {
        return (((x-x05)*(x-x1)*(x-x2)*(x-x3)*(x-x4)*(x-x5)*(x-x6)*(x-x7))/((x0-
x05)*(x0-x1)*(x0-x2)*(x0-x3)*(x0-x4)*(x0-x5)*(x0-x6)*(x0-x7)))*y0
        + (((x-x0)*(x-x1)*(x-x2)*(x-x3)*(x-x4)*(x-x5)*(x-x6)*(x-x7))/((x05-
x0)*(x05-x1)*(x05-x2)*(x05-x3)*(x05-x4)*(x05-x5)*(x05-x6)*(x05-x7)))*y05
        + (((x-x0)*(x-x05)*(x-x2)*(x-x3)*(x-x4)*(x-x5)*(x-x6)*(x-x7))/((x1-
x0)*(x1-x05)*(x1-x2)*(x1-x4)*(x1-x5)*(x1-x6)*(x1-x7)))*y1
        + (((x-x0)*(x-x05)*(x-x1)*(x-x3)*(x-x4)*(x-x5)*(x-x6)*(x-x7))/((x2-
x0)*(x2-x05)*(x2-x1)*(x2-x3)*(x2-x4)*(x2-x5)*(x2-x6)*(x2-x7)))*y2
        + (((x-x0)*(x-x05)*(x-x1)*(x-x2)*(x-x4)*(x-x5)*(x-x6)*(x-x7))/((x3-
x0)*(x3-x05)*(x3-x1)*(x3-x4)*(x3-x5)*(x3-x6)*(x3-x7)))*y3
        + (((x-x0)*(x-x05)*(x-x1)*(x-x2)*(x-x3)*(x-x5)*(x-x6)*(x-x7))/((x4-
x0)*(x4-x05)*(x4-x1)*(x4-x2)*(x4-x3)*(x4-x5)*(x4-x6)*(x4-x7)))*y4
        + (((x-x0)*(x-x05)*(x-x1)*(x-x2)*(x-x3)*(x-x4)*(x-x6)*(x-x7))/((x5-
x0)*(x5-x05)*(x5-x1)*(x5-x3)*(x5-x4)*(x5-x6)*(x5-x7)))*y5
        + (((x-x0)*(x-x05)*(x-x1)*(x-x2)*(x-x3)*(x-x4)*(x-x5)*(x-x7))/((x6-
x0)*(x6-x05)*(x6-x1)*(x6-x2)*(x6-x3)*(x6-x4)*(x6-x5)*(x6-x7)))*y6
        + (((x-x0)*(x-x05)*(x-x1)*(x-x2)*(x-x3)*(x-x4)*(x-x5)*(x-x6))/((x7-
x0)*(x7-x05)*(x7-x1)*(x7-x2)*(x7-x3)*(x7-x4)*(x7-x5)*(x7-x6)))*y7;
    }
};
private UnaFuncion rectaG= new UnaFuncion() {
    @Override
    public double eval(double x) {
        if(dEsfuerzos[0]!=null) return (1/(1
        -(armado(dMateriales,dArmadura,dEsfuerzos[0]).getDelta()/2))*x;
        else return 47*x/22;
    }
};
private UnaFuncion rectaGInversa= new UnaFuncion() {
    @Override
    public double eval(double y) {
        if(dEsfuerzos[0]!=null) return ((1
        -(armado(dMateriales,dArmadura,dEsfuerzos[0]).getDelta()/2))/1)*y;
        else return 22*y/47;
    }
};

```

```

};
private DosFuncion fLim=new DosFuncion() {
    @Override
    public double eval(double x, double x0, double y0) {
        return ((x-x0)/(1-
armado(dMateriales,dArmadura,dEsfuerzos[0]).getDelta2()))+y0;
    }
};

public static DinamicoFragmentGrafico newInstance(DatosArmadura
dArma,DatosMateriales dMat, DatosEsfuerzos[] dEsf ){

    DinamicoFragmentGrafico f =new DinamicoFragmentGrafico();

    dArmadura=dArma;
    dMateriales=dMat;
    dEsfuerzos=dEsf;

    return f;
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
    View rootView = inflater.inflate(R.layout.dinamico_fragment_grafico, container,
false);

    //Instancia del gráfico
    chart = (LineChart) rootView.findViewById(R.id.chart);

    Button vovlerGrafico=(Button) rootView.findViewById(R.id.volverCalculos);
    vovlerGrafico.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

mListener.onClickCambiarFragmentDinamico(EstaticoFragment.FLAG_FRAGMENT_B);
        }
    });

    limitesLW=(ListView) rootView.findViewById(R.id.limitesLW);

    Button mostrarPuntos=(Button) rootView.findViewById(R.id.mostrarPuntos);

    //Eje x:
    xAxis = chart.getXAxis();
    xAxis.setPosition(XAxis.XAxisPosition.BOTTOM);
    xAxis.setDrawGridLines(false);
    //Eje y:
    yAxis = chart.getAxisLeft();
    yAxis.setDrawGridLines(false);
    yAxis.setLabelCount(1, true);
    yAxis.setDrawLabels(true);
    yAxis.setDrawTopYLabelEntry(false);
    yAxis.setAxisMaxValue((float) yMax);

    for(int i=0;dEsfuerzos[i]!=null;i++){
        contPuntos++;
    }
    putItemsLW(contPuntos);

    if(dEsfuerzos[0]!=null){
        //Si detecta armadura reconfiguro límites del grafico si es necesario
        mulMax=armado(dMateriales,dArmadura,dEsfuerzos[0]).getMul();
        nuMax=armado(dMateriales,dArmadura,dEsfuerzos[0]).getNu();
        cargar();
        for(int i=0; dEsfuerzos[i]!=null ;i++) {
            if (armado(dMateriales,dArmadura,dEsfuerzos[i]).getMul() > mulMax)
mulMax=
                armado(dMateriales,dArmadura,dEsfuerzos[i]).getMul();
            if (armado(dMateriales,dArmadura,dEsfuerzos[i]).getNu() > nuMax) nuMax=
                armado(dMateriales,dArmadura,dEsfuerzos[i]).getNu();
            if (nuMax >= yMax - 0.2) {
                yMax = nuMax + 0.2;
                yAxis.setAxisMaxValue((float) yMax);
            }
            if (mulMax >= xMax - 0.2) {
                xMax = mulMax + 0.2;

```

```

        xAxis.setAxisMaxValue((float) xMax);
    }
}

mostrarPuntos.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        int j=0;
        if (comp==false&&dEsfuerzos[0]!=null) {
            imprimirPuntos(contPuntos);
            j=1;
        }
        if (comp==true&&dEsfuerzos[0]!=null) {
            borrarDatos(setPuntos);
            if (setCompLimits!=null) borrarDatos(setCompLimits);
            if (setCompLimitI!=null) borrarDatos(setCompLimitI);
            j=2;
        }
        if (j==1) comp=true;
        if (j==2) comp=false;
    }
});

limitesLW.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
        if (comp) {
            if (setCompLimits!=null) borrarDatos(setCompLimits);
            if (setCompLimitI!=null) borrarDatos(setCompLimitI);
            String punto =
String.valueOf(parent.getItemAtPosition(position));
            int pos = Integer.parseInt(punto.substring(punto.indexOf(" ") +
1));

            double limUlt = rectaG.eval(
                armado(dMateriales, dArmadura, dEsfuerzos[pos -
1]).getMul());

            pintarLimites(limUlt, pos - 1);
        }
    }
});
imprimirGrafico();
}

imprimirGrafico();

if (getActivity() instanceof PantallasEstrechActivity) {
    Button vovler=(Button) rootView.findViewById(R.id.volverCalculos);
    vovler.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
mListener.onClickCambiarFragmentDinamico(EstaticoFragment.FLAG_FRAGMENT_A);
        }
    });
}

return rootView;
}

public void cargar() {
    //Los puntos dependen solo de geometria-materiales, con lo que paratodos los
    resultado son
    //los mismos, así que me basta con calcular los de cero.
    x3=0.31016+0.2067*armado(dMateriales,dArmadura,dEsfuerzos[0]).getDelta2()-
    0.10339*armado(dMateriales,dArmadura,dEsfuerzos[0]).

```

-DinamicoFragmentNuevaArmadura.class

```
public class DinamicoFragmentNuevaArmadura extends FragmentoPadre {

    private EditText nombreArmadura;
    private EditText b;
    private EditText h;
    private EditText dial;
    private EditText dia2;
    private EditText r1;
    private EditText r2;
    private EditText d1;
    private EditText d2;

    private TextView cosa;

    private static DatosArmadura[] dArmaduras =new DatosArmadura[50];

    private static int contItmsArma=0;

    /*Interfaz antes de agrupar
    public interface GuardadoArmadura{
        public void pasaArmadura(DatosArmadura[] dArmadura);
        public void onClickCambiarFragment2(int flagFragmet);
    }*/

    public FragmentoPadre mListener;

    public static DinamicoFragmentNuevaArmadura newInstance (DatosArmadura[]
dArmaduraGuardados){
        DinamicoFragmentNuevaArmadura f=new DinamicoFragmentNuevaArmadura();

        dArmaduras=dArmaduraGuardados;
        for(int i=0; dArmaduras[i]!=null; i++){
            contItmsArma=i;
        }
        contItmsArma++;

        return f;
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_dinamico_nuevaarmadura,
container, false);

        nombreArmadura=(EditText) rootView.findViewById(R.id.nombreArmadura);
        b=(EditText) rootView.findViewById(R.id.b);
        h=(EditText) rootView.findViewById(R.id.h);
        dial=(EditText) rootView.findViewById(R.id.dial);
        dia2=(EditText) rootView.findViewById(R.id.dia2);
        r1=(EditText) rootView.findViewById(R.id.r1);
        r2=(EditText) rootView.findViewById(R.id.b);
        d1=(EditText) rootView.findViewById(R.id.b);
        d2=(EditText) rootView.findViewById(R.id.b);

        cosa=(TextView) rootView.findViewById(R.id.cosa);

        Button nuevaArmadura= (Button) rootView.findViewById(R.id.nuevaArmadura);
        nuevaArmadura.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {

                try {
                    dArmaduras[contItmsArma] = new DatosArmadura();
                    dArmaduras[contItmsArma].nombre =
nombreArmadura.getText().toString();

```

```

        dArmaduras[contItmsArma].b = Double.parseDouble("0" +
b.getText().toString());
        dArmaduras[contItmsArma].h = Double.parseDouble("0" +
h.getText().toString());
        dArmaduras[contItmsArma].dia1 = Double.parseDouble("0" +
dia1.getText().toString());
        dArmaduras[contItmsArma].dia2 = Double.parseDouble("0" +
dia2.getText().toString());
        dArmaduras[contItmsArma].r1 = Double.parseDouble("0" +
r1.getText().toString());
        dArmaduras[contItmsArma].r2 = Double.parseDouble("0" +
r2.getText().toString());
        dArmaduras[contItmsArma].d1 = Double.parseDouble("0" +
d1.getText().toString());
        dArmaduras[contItmsArma].d2 = Double.parseDouble("0" +
d2.getText().toString());
        guardando(v);
    } catch (NumberFormatException e) {
        cosa.setText("Error al introducir los datos");
    }
    }
});

Button atrasNuevaArmadura=(Button)
rootView.findViewById(R.id.atrasNuevaArmadura);
atrasNuevaArmadura.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

mListener.onClickCambiarFragmentDinamico(EstaticoFragment.FLAG_FRAGMENT_A);
    }
});

return rootView;
}

public void guardando(View v){

    mListener.pasaDatos(dArmaduras);
    mListener.onClickCambiarFragmentDinamico(EstaticoFragment.FLAG_FRAGMENT_A);
}

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);
    try{
        mListener = (FragmentPadre) activity;
    } catch (ClassCastException e){
        throw new ClassCastException(activity.toString());
    }
}
}
}

```

-DinamicoFragmentNuevosDatosDim.class

```
public class DinamicoFragmentNuevosDatosDim extends FragmentoPadre {

    private FragmentoPadre mListener;

    private EditText N;
    private EditText V;
    private EditText M;
    private EditText nombreEsf;

    private EditText fck;
    private EditText Go;
    private EditText Gh;
    private EditText Ec;
    private EditText wkMax;
    private EditText nombreHorm;

    private EditText fyk;
    private EditText Es;
    private EditText Gs;
    private EditText nombreAce;

    private EditText Sta;
    private EditText Gt;
    private EditText Gd;
    private EditText Gv;
    private EditText Ga;
    private EditText K03;
    private EditText delta;
    private EditText coh;
    private EditText nombreSuel;

    private EditText r;
    private EditText r1;
    private EditText dia0;
    private EditText p;
    private EditText en;
    private EditText b2;
    private EditText a1;
    private EditText l;
    private EditText ht;
    private EditText nombreGeom;

    private static String acero=" ";

    private CheckBox cohe;
    private CheckBox checkB400;
    private CheckBox checkB500;
    private boolean cohesion=false;
    private String[] estadoSuelo={"Cohesivo","No cohesivo"};
    private int estSuelo=0;

    private int contEsf=0;
    private int marcEsf=0;

    private int contAce=0;
    private int marcAce=0;

    private int contHorm=0;
    private int marcHorm=0;

    private int contSuel=0;
    private int marcSuel=0;

    private int contGeom=0;
    private int marcGeom=0;

    private ListView esfNominaleLW;
    private ArrayAdapter esfNominaleLWAdapter=null;

    private ListView hormigonLW;
    private ArrayAdapter hormigonLWAdapter=null;
}
```

```

private ListView aceroLW;
private ArrayAdapter aceroLWAdapter=null;

private ListView sueloLW;
private ArrayAdapter sueloLWAdapter=null;

private ListView geometriaLW;
private ArrayAdapter geometriaLWAdapter=null;

private static DatosEsfuerzosNominales[] dEsfuerzos=new DatosEsfuerzosNominales[50];
private static String esfuerzoActual="Esfuerzos Ejemplos";
private static DatosHormigon[] dHormigon=new DatosHormigon[50];
private static String hormigonSeleccionado=" ";
private static DatosAcero[] dAcero=new DatosAcero[50];
private static String aceSeleccionado=" ";
private static DatosSuelo[] dSuelo=new DatosSuelo[50];
private static String suelSeleccionado=" ";
private static DatosGeometria[] dGeometria=new DatosGeometria[50];
private static String geomSeleccionado=" ";

public static DinamicoFragmentNuevosDatosDim newInstance(DatosEsfuerzosNominales[]
dEsf,String esfGuardado,
DatosHormigon[]
dHorm,String hormGuard,
DatosAcero[] dAce, String
aceGuard,
DatosSuelo[] dSuel,String
suelGuard,
DatosGeometria[]
dGeom,String geomGuard){
    DinamicoFragmentNuevosDatosDim f=new DinamicoFragmentNuevosDatosDim();
    dEsfuerzos=dEsf;
    esfuerzoActual=esfGuardado;
    dHormigon=dHorm;
    hormigonSeleccionado=hormGuard;
    dAcero=dAce;
    aceSeleccionado=aceGuard;
    dSuelo=dSuel;
    suelSeleccionado=suelGuard;
    dGeometria=dGeom;
    geomSeleccionado=geomGuard;
    return f;
}

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);
    try{
        mListener = (FragmentPadre) activity;
    }catch (ClassCastException e){
        throw new ClassCastException(activity.toString());
    }
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
    View
    rootView=inflater.inflate(R.layout.dinamico_fragment_nuevosdatosdim,container,false);
    //Esfuerzos Nominales:
    N=(EditText) rootView.findViewById(R.id.N);
    V=(EditText) rootView.findViewById(R.id.V);
    M=(EditText) rootView.findViewById(R.id.M);
    nombreEsf=(EditText) rootView.findViewById(R.id.nombreEsf);

    while(dEsfuerzos[contEsf]!=null){
        contEsf++;
    }

    esfNominaleLW= (ListView) rootView.findViewById(R.id.esfNominaleLW);
    esfNominaleLW.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
            esfuerzoActual=String.valueOf(parent.getItemAtPosition(position));
            marcEsf=0;

```

```

while (dEsfuerzos[marcEsf].getNombre().equals(esfuerzoActual) == false && marcEsf < contEsf)
marcEsf++;
mListener.pasaEsfuerzoActual(esfuerzoActual);
Toast.makeText(getActivity(), "Esfuerzo Cargado:\n"+
String.valueOf(
    dEsfuerzos[marcEsf].getNombre()+"\nDatos:\nN :"+
    +String.valueOf(dEsfuerzos[marcEsf].getN())+" N\nV :"+
    String.valueOf(dEsfuerzos[marcEsf].getV())+" N\nM :"+
    String.valueOf(dEsfuerzos[marcEsf].getM())+" N*m"
    , Toast.LENGTH_SHORT).show();
N.setText(String.valueOf(dEsfuerzos[marcEsf].getN()));
M.setText(String.valueOf(dEsfuerzos[marcEsf].getM()));
V.setText(String.valueOf(dEsfuerzos[marcEsf].getV()));
nombreEsf.setText(dEsfuerzos[marcEsf].getNombre());
});
putItemsLW(dEsfuerzos, contEsf);

Button guardarEsfNominales=(Button)
rootView.findViewById(R.id.guardarEsfNominales);
guardarEsfNominales.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
try {
contEsf=0;
while (dEsfuerzos[contEsf] != null) {
contEsf++;
}
if (dEsfuerzos[contEsf] == null) {
//dEsfuerzos[contEsf] = new DatosEsfuerzosNominales();
DatosEsfuerzosNominales esf = new DatosEsfuerzosNominales();

/*if (nombreEsf.getText().toString() != null){
dEsfuerzos[contEsf].setNombre(nombreEsf.getText().toString());
}*/
if (nombreEsf.getText().toString().isEmpty()) {
esf.setNombre("Sin Nombre "+String.valueOf(contEsf+1));
}
else{
esf.setNombre(nombreEsf.getText().toString());
}
String nom=esf.getNombre();
boolean rep=false;
for (int i=0; i<contEsf; i++) {
if (nom.equals(dEsfuerzos[i].getNombre())) rep=true;
}
if (rep) esf.setNombre(nom+"|");
String Nstring=N.getText().toString().replaceAll(" ", "");
if (Nstring.indexOf("-") != -1) esf.setN(Double.parseDouble("0"+
+Nstring.replaceAll("-", "") * (-1)));
if (Nstring.indexOf("-") == -1) esf.setN(Double.parseDouble("0"+
+Nstring.replaceAll("-", "")));
String Vstring=V.getText().toString().replaceAll(" ", "");
if (Vstring.indexOf("-") != -1) esf.setV(Double.parseDouble("0"+
+Vstring.replaceAll("-", "") * (-1)));
if (Vstring.indexOf("-") == -1) esf.setV(Double.parseDouble("0"+
+Vstring.replaceAll("-", "")));
String Mstring=M.getText().toString().replaceAll(" ", "");
if (Mstring.indexOf("-") != -1) esf.setM(Double.parseDouble("0"+
+Mstring.replaceAll("-", "") * (-1)));
if (Mstring.indexOf("-") == -1) esf.setM(Double.parseDouble("0"+
+Mstring.replaceAll("-", "")));

N.setText("");
V.setText("");
M.setText("");
nombreEsf.setText("");
dEsfuerzos[contEsf]=esf;
mListener.pasaDatos(dEsfuerzos);
putItemsLW(dEsfuerzos, contEsf+1);
}

} catch (NumberFormatException e) {
Context context = getActivity();
CharSequence text = "Fallo al introducir\n los datos";
int duration = Toast.LENGTH_SHORT;

```

```

        Toast toast = Toast.makeText(context, text, duration);
        toast.show();
    }
}
});
//Hormigon

fck=(EditText) rootView.findViewById(R.id.fck);
Gh=(EditText) rootView.findViewById(R.id.Gh);
Go=(EditText) rootView.findViewById(R.id.Go);
Ec=(EditText) rootView.findViewById(R.id.Ec);
wkMax=(EditText) rootView.findViewById(R.id.wkMax);
nombreHorm=(EditText) rootView.findViewById(R.id.nomHorm);

while (dHormigon[contHorm]!=null) {
    contHorm++;
}

hormigonLW= (ListView) rootView.findViewById(R.id.hormigonLW);
hormigonLW.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
        hormigonSeleccionado=String.valueOf(parent.getItemAtPosition(position));
        marcHorm=0;

while (dHormigon[marcHorm].getNombre().equals(hormigonSeleccionado)==false&&marcHorm<contH
orm) marcHorm++;
    mListener.pasaHormigonActual(hormigonSeleccionado);
    Toast.makeText(getActivity(), "Hromigon Cargado:\n"+ String.valueOf(
        dHormigon[marcHorm].getNombre()+"\nDatos:\nGh :"+
        +String.valueOf(dHormigon[marcHorm].getGh())+" N/m^3\nEc
:"+
        String.valueOf(dHormigon[marcHorm].getEc()/1000000)+"
Mpa\nfck :"+
        String.valueOf(dHormigon[marcHorm].getFck()/1000000)+"
Mpa\nGo :"+
        String.valueOf(dHormigon[marcHorm].getGo())+"\nwkMax= "+
        String.valueOf((dHormigon[marcHorm].getWkMax()))+" mm"
        , Toast.LENGTH_SHORT).show();
    fck.setText(String.valueOf(dHormigon[marcHorm].getFck()/1000000));
    Go.setText(String.valueOf(dHormigon[marcHorm].getGo()));
    Gh.setText(String.valueOf(dHormigon[marcHorm].getGh()));
    Ec.setText(String.valueOf(dHormigon[marcHorm].getEc()/1000000));
    wkMax.setText(String.valueOf(dHormigon[marcHorm].getWkMax()));
    nombreHorm.setText(dHormigon[marcHorm].getNombre());
}
});
putItemsLW(dHormigon, contHorm);

Button guardarHormigon=(Button) rootView.findViewById(R.id.guardarHormi);
guardarHormigon.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            contHorm=0;
            while (dHormigon[contHorm]!=null) {
                contHorm++;
            }
            if (dHormigon[contHorm]==null) {
                dHormigon[contHorm] = new DatosHormigon();
                if (nombreHorm.getText().toString()!=null) {
dHormigon[contHorm].setNombre(nombreHorm.getText().toString());
                }
                if (nombreHorm.getText().toString().equals("")) {
                    dHormigon[contHorm].setNombre("Sin Nombre
"+String.valueOf(contHorm+1));
                }
                dHormigon[contHorm].setFck(Double.parseDouble("0" +
fck.getText().toString().replaceAll(" ", ""))*1000000);
                dHormigon[contHorm].setGo(Double.parseDouble("0" +
Go.getText().toString().replaceAll(" ", "")));
                dHormigon[contHorm].setGh(Double.parseDouble("0" +
Gh.getText().toString().replaceAll(" ", "")));
                dHormigon[contHorm].setEc(Double.parseDouble("0" +

```

```

Ec.getText().toString().replaceAll(" ", ""))*1000000);
    dHormigon[contHorm].setWkMax(Double.parseDouble("0" +
wkMax.getText().toString().replaceAll(" ", "")));
    fck.setText("");
    Go.setText("");
    Gh.setText("");
    Ec.setText("");
    wkMax.setText("");
    nombreHorm.setText("");
    mListener.pasaDatos(dHormigon);
    putItemsLW(dHormigon, contHorm+1);
}

} catch (NumberFormatException e) {
    Context context = getActivity();
    CharSequence text = "Fallo al introducir\n los datos";
    int duration = Toast.LENGTH_SHORT;

    Toast toast = Toast.makeText(context, text, duration);
    toast.show();
}
});
//Acero
fyk=(EditText) rootView.findViewById(R.id.fyk);
Es=(EditText) rootView.findViewById(R.id.Es);
Gs=(EditText) rootView.findViewById(R.id.Gs);
checkB400=(CheckBox) rootView.findViewById(R.id.checkB400);
checkB500=(CheckBox) rootView.findViewById(R.id.checkB500);
nombreAce=(EditText) rootView.findViewById(R.id.nomAce);

checkB400.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        checkB500.setChecked(false);
    }
});
checkB500.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        checkB400.setChecked(false);
    }
});
while (dAcero[contAce] != null) {
    contAce++;
}

aceroLW= (ListView) rootView.findViewById(R.id.aceroLW);
aceroLW.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
        aceSeleccionado=String.valueOf(parent.getItemAtPosition(position));
        marcAce=0;

while (dAcero[marcAce].getNombre().equals(aceSeleccionado) == false && marcAce < contEsf)
marcAce++;

mListener.pasaAceroActual(aceSeleccionado);
if (dAcero[marcAce].isB400()) acero="B-400";
if (dAcero[marcAce].isB500()) acero="B-500";
Toast.makeText(getActivity(), "Acero Cargado:"+"\nDatos:\nTipo de acero
:"+
        acero+" \nfyk :"+
        +String.valueOf(dAcero[marcAce].getFyk()/1000000)+"
MPa\nE :"+
        String.valueOf(dAcero[marcAce].getEs()/1000000)+"
MPa\nGs :"+
        String.valueOf(dAcero[marcAce].getGs())+" "
, Toast.LENGTH_SHORT).show();
fyk.setText(String.valueOf(dAcero[marcAce].getFyk()/1000000));
Es.setText(String.valueOf(dAcero[marcAce].getEs()/1000000));
Gs.setText(String.valueOf(dAcero[marcAce].getGs()));
checkB400.setChecked(dAcero[marcAce].isB400());
checkB500.setChecked(dAcero[marcAce].isB500());
nombreAce.setText(dAcero[marcAce].getNombre());
}
});
}
});

```

```

putItemsLW(dAcero, contAce);

Button guardarAcero=(Button) rootView.findViewById(R.id.guardarAcero);
guardarAcero.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            contAce=0;
            while(dAcero[contAce]!=null){
                contAce++;
            }
            if(dAcero[contAce]==null){
                dAcero[contAce] = new DatosAcero();
                if (nombreAce.getText().toString()!=null){
                    dAcero[contAce].setNombre(nombreAce.getText().toString());
                }
                if (nombreAce.getText().toString().equals("")){
                    dAcero[contAce].setNombre("Sin Nombre
"+String.valueOf(contEsf+1));
                }
                dAcero[contAce].setFyk(Double.parseDouble("0" +
fyk.getText().toString().replaceAll(" ", ""))*1000000);
                dAcero[contAce].setEs(Double.parseDouble("0" +
Es.getText().toString().replaceAll(" ", ""))*1000000);
                dAcero[contAce].setGs(Double.parseDouble("0" +
Gs.getText().toString().replaceAll(" ", "")));
                dAcero[contAce].setB400(checkB400.isChecked());
                dAcero[contAce].setB500(checkB500.isChecked());
                fyk.setText("");
                Gs.setText("");
                Es.setText("");
                nombreAce.setText("");
                mListener.pasaDatos(dAcero);
                putItemsLW(dAcero, contAce+1);
            }
        } catch (NumberFormatException e){
            Context context = getActivity();
            CharSequence text = "Fallo al introducir\n los datos";
            int duration = Toast.LENGTH_SHORT;

            Toast toast = Toast.makeText(context, text, duration);
            toast.show();
        }
    }
});
//Suelo
Sta=(EditText) rootView.findViewById(R.id.Sta);
Gt=(EditText) rootView.findViewById(R.id.Gt);
Gd=(EditText) rootView.findViewById(R.id.Gd);
Ga=(EditText) rootView.findViewById(R.id.Ga);
Gv=(EditText) rootView.findViewById(R.id.Gv);
delta=(EditText) rootView.findViewById(R.id.del);
K03=(EditText) rootView.findViewById(R.id.K03);
cohe=(EditText) rootView.findViewById(R.id.cohe);
cohe=(CheckBox) rootView.findViewById(R.id.cohe);
nombreSuel=(EditText) rootView.findViewById(R.id.nomSuel);

while (dSuelo[contSuel]!=null){
    contSuel++;
}

sueloLW= (ListView) rootView.findViewById(R.id.sueloLW);
sueloLW.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
        suelSeleccionado=String.valueOf(parent.getItemAtPosition(position));
        marcSuel=0;

while (dSuelo[marcSuel].getNombre().equals(suelSeleccionado)==false&&marcSuel<contSuel)
marcSuel++;

        mListener.pasaSueloActual(suelSeleccionado);
        if(!dSuelo[marcSuel].getCohe()) estSuelo=1;
        if(dSuelo[marcSuel].getCohe()) estSuelo=0;
        Toast.makeText(getActivity(), "Suelo Cargado:" + "\nDatos:\nSta : "
+String.valueOf(dSuelo[marcSuel].getSta())+" N/m^2\nGt"+

```

```

        String.valueOf(dSuelo[marcSuel].getGt())+" N/m^3\nTipo
de suelo: :"+
        estadoSuelo[estSuelo]+" \ncoh :"+
        String.valueOf(dSuelo[marcSuel].getCoh())+"\nGd :"+
        String.valueOf(dSuelo[marcSuel].getGd())+"\nK03 :"+
        String.valueOf(dSuelo[marcSuel].getK03()/1000000)+"
MN/m^3\nDelta :"+
        String.valueOf(dSuelo[marcSuel].getDel())+"\nGa :"+
        String.valueOf(dSuelo[marcSuel].getGa())+"\nGv :"+
        String.valueOf(dSuelo[marcSuel].getGv())+"
        , Toast.LENGTH_SHORT).show();
        Sta.setText(String.valueOf(dSuelo[marcSuel].getSta()));
        Gt.setText(String.valueOf(dSuelo[marcSuel].getGt()));
        Gd.setText(String.valueOf(dSuelo[marcSuel].getGd()));
        Ga.setText(String.valueOf(dSuelo[marcSuel].getGa()));
        Gv.setText(String.valueOf(dSuelo[marcSuel].getGv()));
        delta.setText(String.valueOf(dSuelo[marcSuel].getDel()));
        K03.setText(String.valueOf(dSuelo[marcSuel].getK03()/1000000));
        coh.setText(String.valueOf(dSuelo[marcSuel].getCoh()));
        cohe.setChecked(dSuelo[marcSuel].isCohe());
        nombreSuel.setText(dSuelo[marcSuel].getNombre());
    }
});
putItemsLW(dSuelo, contSuel);

Button guardarSuelo=(Button) rootView.findViewById(R.id.guardarSuelo);
guardarSuelo.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            contSuel=0;
            while(dSuelo[contSuel]!=null){
                contSuel++;
            }
            if(dSuelo[contSuel]==null){
                dSuelo[contSuel] = new DatosSuelo();
                if (nombreSuel.getText().toString()!=null){
                    dSuelo[contSuel].setNombre(nombreSuel.getText().toString());
                }
                if (nombreSuel.getText().toString().equals("")){
                    dSuelo[contSuel].setNombre("Sin Nombre
"+String.valueOf(contSuel+1));
                }
                dSuelo[contSuel].setSta(Double.parseDouble("0" +
Sta.getText().toString().replaceAll(" ", "")));
                dSuelo[contSuel].setGt(Double.parseDouble("0" +
Gt.getText().toString().replaceAll(" ", "")));
                dSuelo[contSuel].setCoh(Double.parseDouble("0" +
coh.getText().toString().replaceAll(" ", "")));
                dSuelo[contSuel].setGd(Double.parseDouble("0" +
Gd.getText().toString().replaceAll(" ", "")));
                dSuelo[contSuel].setGv(Double.parseDouble("0" +
Gv.getText().toString().replaceAll(" ", "")));
                dSuelo[contSuel].setGa(Double.parseDouble("0" +
Ga.getText().toString().replaceAll(" ", "")));
                dSuelo[contSuel].setDel(Double.parseDouble("0" +
delta.getText().toString().replaceAll(" ", "")));
                dSuelo[contSuel].setK03(Double.parseDouble("0" +
K03.getText().toString().replaceAll(" ", ""))*1000000);
                if(cohe.isChecked()) cohesion=true;
                dSuelo[contSuel].setCohe(cohesion);
                Sta.setText("");
                Gt.setText("");
                Gd.setText("");
                Ga.setText("");
                Gv.setText("");
                delta.setText("");
                K03.setText("");
                coh.setText("");
                nombreSuel.setText("");
                mListener.pasaDatos(dSuelo);
                putItemsLW(dSuelo, contSuel+1);
            }
        } catch (NumberFormatException e){
            Context context = getActivity();
            CharSequence text = "Fallo al introducir\n los datos";

```

```

        int duration = Toast.LENGTH_SHORT;

        Toast toast = Toast.makeText(context, text, duration);
        toast.show();
    }
});
//Geometría
r=(EditText) rootView.findViewById(R.id.r);
r1=(EditText) rootView.findViewById(R.id.r1);
dia0=(EditText) rootView.findViewById(R.id.dia0);
p=(EditText) rootView.findViewById(R.id.p);
en=(EditText) rootView.findViewById(R.id.en);
b2=(EditText) rootView.findViewById(R.id.b2);
a1=(EditText) rootView.findViewById(R.id.a1);
l=(EditText) rootView.findViewById(R.id.l);
ht=(EditText) rootView.findViewById(R.id.ht);
nombreGeom=(EditText) rootView.findViewById(R.id.nomGeom);

while (dGeometria[contGeom] != null) {
    contGeom++;
}

geometriaLW= (ListView) rootView.findViewById(R.id.geometriaLW);
geometriaLW.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
        geomSeleccionado=String.valueOf(parent.getItemAtPosition(position));
        marcGeom=0;

        while (dGeometria[marcGeom].getNombre().equals(geomSeleccionado) == false && marcGeom < contGeom)
        ) marcGeom++;

        mListener.pasaGeomActual(geomSeleccionado);
        Toast.makeText(getActivity(), "Geometría Cargado:" + "\nDatos:\nr1 :"+
        +String.valueOf(dGeometria[marcGeom].getR1())+" m\nr2
:"+
        +String.valueOf(dGeometria[marcGeom].getR1())+" m\nr0 :"+
        +String.valueOf(dGeometria[marcGeom].getDia0())+" mm\nen
:"+
        +String.valueOf(dGeometria[marcGeom].getEn())+" m\nb2 :"+
        +String.valueOf(dGeometria[marcGeom].getB2())+" m\na1 :"+
        +String.valueOf(dGeometria[marcGeom].getA1())+" m\nl :"+
        +String.valueOf(dGeometria[marcGeom].getL())+" m\nht :"+
        +String.valueOf(dGeometria[marcGeom].getHt())+"
m\np (muro) :"+
        +String.valueOf(dGeometria[marcGeom].getP())
        , Toast.LENGTH_SHORT).show();
        r.setText(String.valueOf(dGeometria[marcGeom].getR()));
        r1.setText(String.valueOf(dGeometria[marcGeom].getR1()));
        dia0.setText(String.valueOf(dGeometria[marcGeom].getDia0()));
        p.setText(String.valueOf(dGeometria[marcGeom].getP()));
        en.setText(String.valueOf(dGeometria[marcGeom].getEn()));
        b2.setText(String.valueOf(dGeometria[marcGeom].getB2()));
        a1.setText(String.valueOf(dGeometria[marcGeom].getA1()));
        l.setText(String.valueOf(dGeometria[marcGeom].getL()));
        ht.setText(String.valueOf(dGeometria[marcGeom].getHt()));
        nombreGeom.setText(dGeometria[marcGeom].getNombre());
    }
});
putItemsLW(dGeometria, contGeom);

Button gaurdaGeometria=(Button) rootView.findViewById(R.id.guardarGeom);
gaurdaGeometria.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            contGeom=0;
            while (dGeometria[contGeom] != null) {
                contGeom++;
            }
            if (dGeometria[contGeom] == null) {
                dGeometria[contGeom] = new DatosGeometria();
                if (nombreGeom.getText().toString() != null) {
                    dGeometria[contGeom].setNombre(nombreGeom.getText().toString());
                }
            }
        }
    }
});

```



```

        if (nombreGeom.getText().toString().equals("")){
            dGeometria[contGeom].setNombre("Sin Nombre
"+String.valueOf(contSuel+1));
        }
        dGeometria[contGeom].setR(Double.parseDouble("0" +
r.getText().toString().replaceAll(" ", "")));
        dGeometria[contGeom].setP(Double.parseDouble("0" +
p.getText().toString().replaceAll(" ", "")));
        String enString=en.getText().toString().replaceAll(" ", "");
        if(enString.indexOf("-")!=-1)
dGeometria[contGeom].setEn(Double.parseDouble("0" + enString.replaceAll("-", ""))*(-1));
        if(enString.indexOf("-")==-
1)dGeometria[contGeom].setEn(Double.parseDouble("0" + enString));
        dGeometria[contGeom].setB2(Double.parseDouble("0" +
b2.getText().toString().replaceAll(" ", "")));
        dGeometria[contGeom].setR1(Double.parseDouble("0" +
r1.getText().toString().replaceAll(" ", "")));
        dGeometria[contGeom].setDia0(Double.parseDouble("0" +
dia0.getText().toString().replaceAll(" ", "")));
        dGeometria[contGeom].setA1(Double.parseDouble("0" +
a1.getText().toString().replaceAll(" ", "")));
        dGeometria[contGeom].setL(Double.parseDouble("0" +
l.getText().toString().replaceAll(" ", "")));
        dGeometria[contGeom].setHt(Double.parseDouble("0" +
ht.getText().toString().replaceAll(" ", "")));
        r.setText("");
        p.setText("");
        en.setText("");
        dia0.setText("");
        r1.setText("");
        b2.setText("");
        a1.setText("");
        l.setText("");
        ht.setText("");
        nombreGeom.setText("");
        mListener.pasaDatos(dGeometria);
        putItemsLW(dGeometria, contGeom+1);
    }

} catch (NumberFormatException e){
    Context context = getActivity();
    CharSequence text = "Fallo al introducir\n los datos";
    int duration = Toast.LENGTH_SHORT;

    Toast toast = Toast.makeText(context, text, duration);
    toast.show();
}
});

return rootView;
}

public void putItemsLW(DatosEsfuerzosNominales[] items, int i){

    String[] meme=new String[i];
    for(int v=0; v<i; v++){
        meme[v]=items[v].getNombre();
    }

    esfNominaleLWAdapter= new ArrayAdapter(getActivity(),
android.R.layout.simple_list_item_1, meme);
    esfNominaleLW.setAdapter(esfNominaleLWAdapter);
}

public void putItemsLW(DatosHormigon[] items, int i){

    String[] meme=new String[i];
    for(int v=0; v<i; v++){
        meme[v]=items[v].getNombre();
    }

    hormigonLWAdapter= new ArrayAdapter(getActivity(),
android.R.layout.simple_list_item_1, meme);
    hormigonLW.setAdapter(hormigonLWAdapter);
}
}

```

```

public void putItemsLW(DatosAcero[] items, int i){
    String[] meme=new String[i];
    for(int v=0; v<i; v++){
        meme[v]=items[v].getNombre();
    }

    aceroLWAdapter= new ArrayAdapter(getActivity(),
android.R.layout.simple_list_item_1, meme);
    aceroLW.setAdapter(aceroLWAdapter);
}

public void putItemsLW(DatosSuelo[] items, int i){
    String[] meme=new String[i];
    for(int v=0; v<i; v++){
        meme[v]=items[v].getNombre();
    }

    sueloLWAdapter= new ArrayAdapter(getActivity(),
android.R.layout.simple_list_item_1, meme);
    sueloLW.setAdapter(sueloLWAdapter);
}

public void putItemsLW(DatosGeometria[] items, int i){
    String[] meme=new String[i];
    for(int v=0; v<i; v++){
        meme[v]=items[v].getNombre();
    }

    geometriaLWAdapter= new ArrayAdapter(getActivity(),
android.R.layout.simple_list_item_1, meme);
    geometriaLW.setAdapter(geometriaLWAdapter);
}
}

```

-DinamicoFragmentNuevosMateriales.class

```
public class DinamicoFragmentNuevosMateriales extends FragmentoPadre {

    private static DatosMateriales[] dMateriales =new DatosMateriales[50];

    private static int contItmsMat=0;

    private EditText nombreMaterial;
    private EditText fcd;
    private EditText fyd;
    private EditText es;

    private TextView cosal;

    public FragmentoPadre mListener;

    public static DinamicoFragmentNuevosMateriales newInstance(DatosMateriales[]
dMaterialesGuardados){
        DinamicoFragmentNuevosMateriales f=new DinamicoFragmentNuevosMateriales();

        dMateriales=dMaterialesGuardados;
        for(int i=0; dMateriales[i]!=null; i++){
            contItmsMat=i;
        }
        contItmsMat++;

        return f;
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragmet_dinamico_nuevomaterial,
container, false);

        nombreMaterial=(EditText) rootView.findViewById(R.id.nombreMateriales);
        fcd=(EditText) rootView.findViewById(R.id.fcd);
        fyd=(EditText) rootView.findViewById(R.id.fyd);
        es=(EditText) rootView.findViewById(R.id.es);

        cosal=(TextView) rootView.findViewById(R.id.cosal);

        Button guardarMateriales= (Button)
rootView.findViewById(R.id.guardarMateriales);
        guardarMateriales.setOnClickListener(new View.OnClickListener(){

            @Override
            public void onClick(View v) {

                try {
                    dMateriales[contItmsMat] = new DatosMateriales();
                    dMateriales[contItmsMat].nombre =
nombreMaterial.getText().toString();
                    dMateriales[contItmsMat].fcd = Double.parseDouble("0" +
fcd.getText().toString());
                    dMateriales[contItmsMat].fyd = Double.parseDouble("0" +
fyd.getText().toString());
                    dMateriales[contItmsMat].es = Double.parseDouble("0" +
es.getText().toString());
                    guardando(v);
                } catch (NumberFormatException e) {
                    cosal.setText("Error al introducir los datos");
                }
            }
        });

        Button atrasNuevoMaterial=(Button)
rootView.findViewById(R.id.atrasNuevoMaterial);
        atrasNuevoMaterial.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

mListener.onClickCambiarFragmentDinamico(EstaticoFragment.FLAG_FRAGMENT_A);
```

```

        }
    });
    return rootView;
}

public void guardando(View v){

    mListener.pasaDatos(dMateriales);
    mListener.onClickCambiarFragmentDinamico(EstaticoFragment.FLAG_FRAGMENT_A);
}

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);
    try{
        mListener = (FragmentPadre) activity;
    }catch (ClassCastException e){
        throw new ClassCastException(activity.toString());
    }
}
}
}

```

-DinamicoFragmentPrincipal.class

```

public class DinamicoFragmentPrincipal extends FragmentoPadre {

    @Override
    public void onAttach(Activity activity) {
        super.onAttach(activity);
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        View
rootView=inflater.inflate(R.layout.dinamico_fragment_principal,container,false);

        return rootView;
    }
}

```

-EstaticoFragment.class

```
public class EstaticoFragment extends FragmentoPadre {

    public final static int FLAG_FRAGMENT_A = 1;
    public final static int FLAG_FRAGMENT_B = 2;

    private static FragmentoPadre mListenerVacio = new FragmentoPadre() {
        @Override
        public void onClickCambiarFragmentDinamico(int flagFragment) {}

        @Override
        public void volver() {

        }

        @Override
        public void pasaArmaduraActual(String armaduraSeleccionada) {

        }

        @Override
        public void pasaMaterialActual(String materialSeleccionado) {

        }

        @Override
        public void pasaEsfuerzoActual(String esfuerzoSeleccionado) {

        }

        @Override
        public void pasaHormigonActual(String hormigonSeleccionado) {

        }

        @Override
        public void pasaAceroActual(String aceroSeleccionado) {

        }

        @Override
        public void pasaSueloActual(String sueloSeleccionado) {

        }

        @Override
        public void pasaGeomActual(String geomSeleccionado) {

        }

        @Override
        public void pasaDimensionesActual(String dimSeleccionado) {

        }

        @Override
        public void pasaDatos(DatosArmadura[] dArmadura) {

        }

        @Override
        public void pasaDatos(DatosMateriales[] dMateriales) {

        }

        @Override
        public void pasaDatos(DatosEsfuerzos[] dEsfuerzos) {

        }

        @Override
        public void pasaDatos(DatosEsfuerzosNominales[] dEsfuerzos) {
```

```

    }

    @Override
    public void pasaDatos(DatosHormigon[] dHormigon) {

    }

    @Override
    public void pasaDatos(DatosAcero[] dAcero) {

    }

    @Override
    public void pasaDatos(DatosSuelo[] dSuelo) {

    }

    @Override
    public void pasaDatos(DatosGeometria[] dGeom) {

    }

    @Override
    public void pasaDatos(DatosDimensiones[] dDim) {

    }
};

private FragmentPadre mListener = mListnerVacio;

public EstaticoFragment() {
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {

    View rootView = inflater.inflate(R.layout.fragment_estatico, container, false);

    Button botonA = (Button)
rootView.findViewById(R.id.button$fragment_estatico$cargarFragmentDinamicoA);
    botonA.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            mListener.onClickCambiarFragmentDinamico(MainActivity.FLAG_FRAGMENT_F);
        }
    });

    Button botonB = (Button)
rootView.findViewById(R.id.button$fragment_estatico$cargarFragmentDinamicoB);
    botonB.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            mListener.onClickCambiarFragmentDinamico(MainActivity.FLAG_FRAGMENT_H);
        }
    });

    Button
botonC=(Button)rootView.findViewById(R.id.button$fragment_estatico$cargarFragmentDinamic
oGrafico);
    botonC.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            mListener.onClickCambiarFragmentDinamico(FLAG_FRAGMENT_A);
        }
    });
    return rootView;
}

@Override
public void onAttach(Activity activity) {
    super.onAttach(activity);
    try {
        mListener = (FragmentPadre) activity;
    } catch (ClassCastException e) {
        throw new IllegalStateException("La clase " + activity.toString() + " debe

```

```

implementar de la interfaz " + FragmentPadre.class.getName() + " del Fragment al que
quiere escuchar");
    }
}
@Override
public void onDetach() {
    super.onDetach();
    mListener = mListenerVacio;
}
}

```

-FragmentoPadre.class

```

public class FragmentoPadre extends Fragment{

    public interface FragmentPadre{

        public void onClickCambiarFragmentDinamico(int flagFragment);

        public void volver();

        public void pasaArmaduraActual(String armaduraSeleccionada);
        public void pasaMaterialActual(String materialSeleccionado);
        public void pasaEsfuerzoActual(String esfuerzoSeleccionado);
        public void pasaHormigonActual(String hormigonSeleccionado);
        public void pasaAceroActual(String aceroSeleccionado);
        public void pasaSueloActual(String sueloSeleccionado);
        public void pasaGeomActual(String geomSeleccionado);
        public void pasaDimensionesActual(String dimSeleccionado);

        public void pasaDatos(DatosArmadura[] dArmadura);
        public void pasaDatos(DatosMateriales[] dMateriales);
        public void pasaDatos(DatosEsfuerzos[] dEsfuerzos);
        public void pasaDatos(DatosEsfuerzosNominales[] dEsfuerzos);
        public void pasaDatos(DatosHormigon[] dHormigon);
        public void pasaDatos(DatosAcero[] dAcero);
        public void pasaDatos(DatosSuelo[] dSuelo);
        public void pasaDatos(DatosGeometria[] dGeom);
        public void pasaDatos(DatosDimensiones[] dDim);

    }
}

```

-IFuncion.class

```

public interface IFuncion {
    //Método evaluador. Sera sobrescrito.
    double eval(double x); //Siendo x la incógnita, que es pln
}

```

-MainActivity.class

```
public class MainActivity extends Activity implements FragmentoPadre.FragmentPadre{

    private boolean mDosFragmentos = false;

    private DatosArmadura[] dArmadura= new DatosArmadura[50];
    private DatosMateriales[] dMateriales=new DatosMateriales[50];
    private DatosEsfuerzos[] dEsfuerzos=new DatosEsfuerzos[50];
    private DatosEsfuerzosNominales[] dEsfuerzosNominales=new
DatosEsfuerzosNominales[50];
    private DatosHormigon[] dHormigon=new DatosHormigon[50];
    private DatosAcero[] dAcero=new DatosAcero[50];
    private DatosSuelo[] dSuelo=new DatosSuelo[50];
    private DatosGeometria[]dGeometria=new DatosGeometria[50];
    private DatosDimensiones[]dDimensiones=new DatosDimensiones[50];

    public final static int FLAG_FRAGMENT_F=6;
    public final static int FLAG_FRAGMENT_G=7;
    public final static int FLAG_FRAGMENT_H=8;
    public final static int FLAG_FRAGMENT_I=9;
    public final static int FLAG_FRAGMENT_J=10;
    public final static int FLAG_FRAGMENT_K=11;

    private LibreriaParcializable datos=new LibreriaParcializable();

    private String armaduraSeleccionada="Datos Ejemplo";
    private String materialSeleccionado="Datos Ejemplo";
    private static String esfuerzoSeleccionado="Esfuerzos Ejemplo";
    private static String hormigonSeleccionado="Hormigón Ejemplo";
    private static String aceroSeleccionado="Acero Ejemplos";
    private static String sueloSeleccionado="Suelo Ejemplo";
    private static String geometriaSeleccionada="Geometría Ejemplo";
    private static String dimSeleccionado=" ";

    private int flagFragmentn=0;

    private int contArma=0;
    private int contMat=0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Intent intent = getIntent();
        if(getIntent().getExtras() != null) {
            datos=intent.getExtras().getParcelable("datos");
            Bundle extras = getIntent().getExtras();
            int flagFragment =
extras.getInt(PantallasEstrechasActivity.KEY_FLAG_FRAGMENT_ENVIADO);
            if(flagFragment>5){
                flagFragment=flagFragment-5;
                this.onClickCambiarFragmentDinamico(flagFragment);
            }
        }
        View frameLayoutContenedor = findViewById(R.id
            .frameLayout$activity_main$contenedorFragmentsDinamicos);
        mDosFragmentos = frameLayoutContenedor != null && frameLayoutContenedor
            .getVisibility() == View.VISIBLE;
        dArmadura[0]=DatosArmadura.datosEjemplo();
        dMateriales[0]=DatosMateriales.datosEjemplo();
        dEsfuerzosNominales[0]=DatosEsfuerzosNominales.datosEjemplo();
        dHormigon[0]=DatosHormigon.datosEjemplo();
        dAcero[0]=DatosAcero.datosEjemplo();
        dSuelo[0]=DatosSuelo.datosEjemplo();
        dGeometria[0]=DatosGeometria.datosEjemplo();
        datos.setdArm(dArmadura);
        datos.setdMat(dMateriales);
    }
}
```



```

        if (mDosFragmentos) {
            this.onClickCambiarFragmentDinamico(FLAG_FRAGMENT_I);
        }
    }

    @Override
    public void onClickCambiarFragmentDinamico(int flagFragment) {

        if (mDosFragmentos) {

            contArma=0;
            contMat=0;

            while(dArmadura[contArma].getNombre().equals(armaduraSeleccionada)==false) {
                contArma++;
            }
            while(dMateriales[contMat].getNombre().equals(materialSeleccionado)==false) {
                contMat++;
            }

            switch (flagFragment) {
                case EstaticoFragment.FLAG_FRAGMENT_A: {
                    DinamicoFragmentA
                    fragmentElegido=DinamicoFragmentA.newInstance(dArmadura,
                        dMateriales,dEsfuerzos,
                        armaduraSeleccionada,materialSeleccionado);
                    getFragmentManager().beginTransaction()

                    .replace(R.id.frameLayout$activity_main$contenedorFragmentsDinamicos,
                        fragmentElegido)
                        .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
                        .commit();

                    break;
                }
                case EstaticoFragment.FLAG_FRAGMENT_B: {
                    DinamicoFragmentB fragmentElegido = DinamicoFragmentB.newInstance(
                        dArmadura[contArma], dMateriales[contMat], dEsfuerzos);
                    getFragmentManager().beginTransaction()

                    .replace(R.id.frameLayout$activity_main$contenedorFragmentsDinamicos,
                        fragmentElegido)
                        .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
                        .commit();

                    break;
                }
                case DinamicoFragmentA.FLAG_FRAGMENT_C: {
                    DinamicoFragmentNuevaArmadura
                    fragmentElegido=DinamicoFragmentNuevaArmadura
                        .newInstance(dArmadura);
                    getFragmentManager().beginTransaction()

                    .replace(R.id.frameLayout$activity_main$contenedorFragmentsDinamicos,
                        fragmentElegido)
                        .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
                        .commit();

                    break;
                }
                case DinamicoFragmentA.FLAG_FRAGMENT_D: {
                    DinamicoFragmentNuevosMateriales
                    fragmentElegido=DinamicoFragmentNuevosMateriales
                        .newInstance(dMateriales);
                    getFragmentManager().beginTransaction()

                    .replace(R.id.frameLayout$activity_main$contenedorFragmentsDinamicos,
                        fragmentElegido)
                        .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
                        .commit();

                    break;
                }
                case DinamicoFragmentB.FLAG_FRAGMENT_E: {
                    DinamicoFragmentGrafico
                    fragmentElegido=DinamicoFragmentGrafico.newInstance(dArmadura[contArma],
                        dMateriales[contMat], dEsfuerzos);
                    getFragmentManager().beginTransaction()

```

```

.replace(R.id.frameLayout$activity_main$contenedorFragmentsDinamicos,
        fragmentElegido)
        .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
        .commit();
        break;
    }
    case FLAG_FRAGMENT_F: {
        DinamicoFragmentDatosDim
        fragmentElegido=DinamicoFragmentDatosDim.newInstance(
            dEsfuerzosNominales,
            esfuerzoSeleccionado, dHormigon, hormigonSeleccionado,
            dAcero, aceroSeleccionado, dSuelo, sueloSeleccionado, dGeometria, geometriaSeleccionada);
        getFragmentManager().beginTransaction()

.replace(R.id.frameLayout$activity_main$contenedorFragmentsDinamicos,
        fragmentElegido)
        .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
        .commit();
        break;
    }
    case FLAG_FRAGMENT_G: {
        DinamicoFragmentNuevosDatosDim
        fragmentElegido=DinamicoFragmentNuevosDatosDim.newInstance(
            dEsfuerzosNominales,
            esfuerzoSeleccionado, dHormigon, hormigonSeleccionado,
            dAcero, aceroSeleccionado, dSuelo, sueloSeleccionado, dGeometria, geometriaSeleccionada);
        getFragmentManager().beginTransaction()

.replace(R.id.frameLayout$activity_main$contenedorFragmentsDinamicos,
        fragmentElegido)
        .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
        .commit();
        break;
    }
    case FLAG_FRAGMENT_H: {
        DinamicoFragmentComprobaciones
        fragmentElegido=DinamicoFragmentComprobaciones.newInstance(
            dEsfuerzosNominales,
            esfuerzoSeleccionado, dHormigon, hormigonSeleccionado,
            dAcero, aceroSeleccionado, dSuelo, sueloSeleccionado, dGeometria, geometriaSeleccionada,
            dDimensiones, dimSeleccionado);
        getFragmentManager().beginTransaction()

.replace(R.id.frameLayout$activity_main$contenedorFragmentsDinamicos,
        fragmentElegido)
        .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
        .commit();
        break;
    }
    case FLAG_FRAGMENT_I: {
        DinamicoFragmentPrincipal fragmentElegido=new
        DinamicoFragmentPrincipal();
        getFragmentManager().beginTransaction()

.replace(R.id.frameLayout$activity_main$contenedorFragmentsDinamicos,
        fragmentElegido)
        .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
        .commit();
        break;
    }
    case FLAG_FRAGMENT_J: {
        DinamicofragmentCalculoDimensionado
        fragmentElegido=DinamicoFragmentCalculoDimensionado.newInstance(dEsfuerzosNominales,
            esfuerzoSeleccionado, dHormigon, hormigonSeleccionado,
            dAcero, aceroSeleccionado, dSuelo, sueloSeleccionado, dGeometria, geometriaSeleccionada);
        getFragmentManager().beginTransaction()

.replace(R.id.frameLayout$activity_main$contenedorFragmentsDinamicos,
        fragmentElegido)
        .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
        .commit();
        break;
    }
}

```

```

        case FLAG_FRAGMENT_K: {
            DinamicoFragmentCalculoComprobaciones
            fragmentElegido=DinamicoFragmentCalculoComprobaciones.newInstance(dEsfuerzosNominales,
            esfuerzoSeleccionado,dHormigon,hormigonSeleccionado,

dAcero,aceroSeleccionado,dSuelo,sueloSeleccionado,dGeometria,geometriaSeleccionada,dDimensi
ones,dimSeleccionado);
            getFragmentManager().beginTransaction()

.replace(R.id.frameLayout$activity_main$contenedorFragmentsDinamicos,
            fragmentElegido)
            .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE)
            .commit();

            break;
        }

    } else {
        Intent intent = new Intent();
        intent.setClass(this, PantallasEstrechActivity.class);
        intent.putExtra(PantallasEstrechActivity.KEY_FLAG_FRAGMENT_ENVIADO,
flagFragment);
        intent.putExtra("datos", datos);
        startActivity(intent);
    }
}

@Override
public void volver() {

}

@Override
public void pasaArmaduraActual(String armaduraSeleccionada) {
    this.armaduraSeleccionada=armaduraSeleccionada;
}

@Override
public void pasaMaterialActual(String materialSeleccionado) {
    this.materialSeleccionado=materialSeleccionado;
}

@Override
public void pasaEsfuerzoActual(String esfuerzoSeleccionado) {
    this.esfuerzoSeleccionado=esfuerzoSeleccionado;
}

@Override
public void pasaHormigonActual(String hormigonSeleccionado) {
    this.hormigonSeleccionado=hormigonSeleccionado;
}

@Override
public void pasaAceroActual(String aceroSeleccionado) {
    this.aceroSeleccionado=aceroSeleccionado;
}

@Override
public void pasaSueloActual(String sueloSeleccionado) {
    this.sueloSeleccionado=sueloSeleccionado;
}

@Override
public void pasaGeomActual(String geomSeleccionado) {
    this.geometriaSeleccionada=geomSeleccionado;
}

@Override
public void pasaDimensionesActual(String dimSeleccionado) {
    this.dimSeleccionado=dimSeleccionado;
}

@Override
public void pasaDatos(DatosArmadura[] dArmadura) {

```

```

        this.dArmadura=dArmadura;
    }

    @Override
    public void pasaDatos(DatosMateriales[] dMateriales) {
        this.dMateriales=dMateriales;
    }

    @Override
    public void pasaDatos(DatosEsfuerzos[] dEsfuerzos) {
        this.dEsfuerzos=dEsfuerzos;
    }

    @Override
    public void pasaDatos(DatosEsfuerzosNominales[] dEsfuerzos) {
        this.dEsfuerzosNominales=dEsfuerzos;
    }

    @Override
    public void pasaDatos(DatosHormigon[] dHormigon) {
        this.dHormigon=dHormigon;
    }

    @Override
    public void pasaDatos(DatosAcero[] dAcero) {
        this.dAcero=dAcero;
    }

    @Override
    public void pasaDatos(DatosSuelo[] dSuelo) {
        this.dSuelo=dSuelo;
    }

    @Override
    public void pasaDatos(DatosGeometria[] dGeom) {
        this.dGeometria=dGeom;
    }

    @Override
    public void pasaDatos(DatosDimensiones[] dDim) {
        this.dDimensiones=dDim;
    }
}

```

-NewtonRapson.class

```

public class NewtonRapson {
    /*
     * Método para obtener la raiz, que es pln.
     * Recive como argumentos:
     * f=>la funcion, que sera mu en sus distintas formas.
     * df=> derivada de f.
     * x=>variable a resolver, que es pln
     * Sobre esta x: como antes de inicial la resloución del método ya he
     * identificado el dominio, y por tanto el intervalo donde estará la
     * raiz, voy a enviar el valor medio como primer nodo.
     * e=>error de la interpolación
     * n=> n° de interpolaciones
     */
    public double raiz(IFuncion f, IFuncion df, double x, double e, int n){
        double r=Double.NaN;
        int k=0;
        while (Math.abs(f.eval(x))>e && k<n){
            x=x-f.eval(x)/df.eval(x);
            k++;
        }
        if (k<n) r=x;
        return r;
    }
}

```

-Redondos.class

```
public class Redondos {

    private static int n1;
    private static int n2;
    //comrpovadores de cumplimiento de la norma
    private static int v1;
    private static int v2;
    //identificadores del diámetro;
    private static int contDia1;
    private static int contDia2;
    //distancias separación redondos:
    private static double ds1;
    private static double ds2;

    public void Redondos() {
        n1=n2=v1=v2=contDia1=contDia2=0;
        ds1=ds2=0;
    }

    public static void calculoRedondos(DatosArmadura dArma, double A1, double A2) {

        //resultados={ A1, A2, n1, n2, ds1, ds2, v1, v2, contDia1, contDia2};
        //Vector con los valores de los diámetros normalizados:
        final double diametros[] = {6, 8, 10, 12, 14, 16, 20, 25, 32, 40};

        double b = dArma.getB();
        double r1 = dArma.getR1();
        double r2 = dArma.getR2();
        double dial = dArma.getDial();
        double dia2 = dArma.getDia2();

        if (A1 != 0) { //para evitar el error si no hay A1
            n1 = (int) ((int) ((4 * A1) / (Math.PI * dial * dial * 1e-2)) + 1);
            if (n1 < 2) { //comprobación del número de redondos según la norma
                n1 = 2;
            }
            ds1 = (double) (b - 2 * r1 - n1 * dial * 0.1) / (n1 - 1);
            //identifico posición del diámetro en el vector:
            contDia1=0;
            while (diametros[contDia1] != dial && contDia1 < 10) {
                contDia1++;
            }
            //Criterio de validación según la norma: Recalculo en caso de no cumplirlo
            while (ds1 < 2 && ds1 != 0) { //RECORDAR: falta condición del tamaño del
                árido
                    v1++; //utilizo este contador para ver si recalcula
                    contDia1++;
                    n1 = (int) ((int) ((4 * A1) / (Math.PI * diametros[contDia1] *
diametros[contDia1] * 1e-2)) + 1);
                    if (n1 < 2) {
                        n1 = 2;
                    }
                    ds1 = (double) (b - 2 * r1 - n1 * diametros[contDia1] * 0.1) / (n1 - 1);
                }
            }
        }
        if (A2 != 0) { //lo mismo para A2
            n2 = (int) ((int) ((4 * A2) / (Math.PI * dia2 * dia2 * 1e-2)) + 1);
            if (n2 < 2) {
                n2 = 2;
            }
            ds2 = (double) (b - 2 * r2 - n2 * dia2 * 0.1) / (n2 - 1);
            contDia2=0;
            while (diametros[contDia2] != dia2) {
                contDia2++;
            }
            while (ds2 < 2 && ds2 != 0) { //RECORDAR: falta condición del tamaño del
```

árido

```
        v2++;
        contDia2++;
        n2 = (int) ((int) ((4 * A2) / (Math.PI * diametros[contDia2] *
diametros[contDia2] * 1e-2)) + 1);
        if (n2 < 2) {
            n2 = 2;
        }
        ds2 = (double) (b - 2 * r2 - n2 * diametros[contDia2] * 0.1) / (n2 - 1);
    }
}

public int getN1() {
    return n1;
}

public int getN2() {
    return n2;
}

public double getDs1() {
    return ds1;
}

public double getDs2() {
    return ds2;
}

public int getV1() {
    return v1;
}

public int getV2() {
    return v2;
}

public int getContDia1() {
    return contDia1;
}

public int getContDia2() {
    return contDia2;
}
}
```

-Redondos2.class

```
public class Redondos2 {

    private int n1;
    private int n2;
    //comrpovadores de cumplimiento de la norma
    private int v1;
    private int v2;
    //identificadores del diámetro;
    private int contDial;
    private int contDia2;
    //distancias separación redondos:
    private double ds1;
    private double ds2;

    public void Redondos() {
        n1=n2=v1=v2=contDial=contDia2=0;
        ds1=ds2=0;
    }

    public void calculoRedondos(double b2, double A1, double A2, double dial) {

        //Redondos2 f=new Redondos2();

        //resultados={ A1, A2, n1, n2, ds1, ds2, v1, v2, contDial1, contDia2};
        //Vector con los valores de los diámetros normalizados:
        final double diametros[] = { 10, 12, 14, 16, 20, 25, 32, 40};

        double b = b2;
        double r1 = 2;
        double r2 = 2;
        double dial = dial;
        double dia2 = 12;

        if (A1 != 0) {//para evitar el error si no hay A1
            n1 = (int) ((int) ((4 * A1) / (Math.PI * dial * dial * 1e-2)) + 1);
            if (n1 < 2) {//comprobación del número de redondos segun la norma
                n1 = 2;
            }
            ds1 = (double) (b - 2 * r1 - n1 * dial * 0.1) / (n1 - 1);
            //identifico posición del diámetro en el vector:
            contDial=0;
            while (diametros[contDial] != dial && contDial < 10) {
                contDial++;
            }
            //Criterio de validación segun la norma: Recalculo en caso de no cumplirlo
            while (ds1 < 2 && ds1 != 0) {//RECORDAR: falta condición del tamaño del
                árido
                v1++;//utilizo este contador para ver si recalcula
                contDial++;
                n1 = (int) ((int) ((4 * A1) / (Math.PI * diametros[contDial] *
diametros[contDial] * 1e-2)) + 1);
                if (n1 < 2) {
                    n1 = 2;
                }
                ds1 = (double) (b - 2 * r1 - n1 * diametros[contDial] * 0.1) / (n1 - 1);
            }
        }
        if (A2 != 0) {//lo mismo para A2
            n2 = (int) ((int) ((4 * A2) / (Math.PI * dia2 * dia2 * 1e-2)) + 1);
            if (n2 < 2) {
                n2 = 2;
            }
            ds2 = (double) (b - 2 * r2 - n2 * dia2 * 0.1) / (n2 - 1);
            contDia2=0;
            while (diametros[contDia2] != dia2) {
                contDia2++;
            }
            while (ds2 < 2 && ds2 != 0) {//RECORDAR: falta condición del tamaño del
                árido
                v2++;
                contDia2++;
                n2 = (int) ((int) ((4 * A2) / (Math.PI * diametros[contDia2] *

```

```

diametros[contDia2] * 1e-2)) + 1);
        if (n2 < 2) {
            n2 = 2;
        }
        ds2 = (double) (b - 2 * r2 - n2 * diametros[contDia2] * 0.1) / (n2 - 1);
    }

    }
    //return f;
}

public int getN1() {
    return n1;
}

public int getN2() {
    return n2;
}

public double getDs1() {
    return ds1;
}

public double getDs2() {
    return ds2;
}

public int getV1() {
    return v1;
}

public int getV2() {
    return v2;
}

public int getContDia1() {
    return contDia1;
}

public int getContDia2() {
    return contDia2;
}
}

```

-UnaFuncion.class

```

public abstract class UnaFuncion implements IFuncion{

    @Override
    public abstract double eval(double x);

}

```

-strings.XML:

```
<resources>

  <string name="app_name">"Interfaz"</string>

  <string name="fil"> $\phi$ <sub>1</sub> </string>
  <string name="boton_1">Dimensionado Óptimo</string>
  <string name="boton_2">Comprobación Dimensionado</string>
  <string name="boton_3">Armado en rotura</string>

  <string name="b">b </string>
  <string name="h">h </string>
  <string name="fi2"> $\phi$ <sub>2</sub> [mm] </string>
  <string name="r1">r<sub>1</sub> </string>
  <string name="r2">r<sub>2</sub> </string>
  <string name="d1">d<sub>1</sub> </string>
  <string name="d2">d<sub>2</sub> </string>
  <string name="datos_guardados">Armaduras </string>
  <string name="nuevos_datos">Nuevos datos</string>
  <string name="guardar">Guardar</string>
  <string name="cargar_armadura">Cargar</string>
  <string name="nombre">Nombre </string>
  <string name="datos_amateriales">Datos amateriales </string>
  <string name="fcd">F<sub>cd</sub> </string>
  <string name="fyd">F<sub>y</sub>d </string>
  <string name="datos_nueva_armadura">Datos Nueva Armadura</string>
  <string name="nueva_armadura">Nueva Armadura</string>
  <string name="armadura_seleccionada">Armadura seleccionada </string>
  <string name="datos_nuevos_materiales">Datos Nuevos Materiales </string>
  <string name="materiales">Materiales </string>
  <string name="materiales_seleccionados">Materiales Seleccionados </string>
  <string name="nuevo_material">Nuevo Material</string>
  <string name="l_mites">Límites</string>
  <string name="nud"> $\nu$ <sub>d</sub> </string>
  <string name="mud"> $\mu$ <sub>d</sub> </string>
  <string name="epsilon"> $\xi$  </string>
  <string name="mud1"> $\mu$ <sub>d1</sub> </string>
  <string name="muoptimo"> $\mu$ <sub>ul</sub>,</sub><sup>c</sup><sub>Optimo</sub> </string>
  <string name="dominio">Dominio </string>
  <string name="plnoptima"> $\xi$ <sub>Optima</sub> </string>
  <string name="psi"> $\Psi(\xi)$  </string>
  <string name="lamabda"> $\lambda(\xi)$ </string>
  <string name="w1">w<sub>1</sub> </string>
  <string name="w2">w<sub>2</sub> </string>
  <string name="a1">A<sub>1</sub> </string>
  <string name="a2">A<sub>2</sub> </string>
  <string name="n1">n<sub>1</sub> </string>
  <string name="n2">n<sub>2</sub> </string>
  <string name="ds1">d<sub>s1</sub> </string>
  <string name="ds2">d<sub>s2</sub> </string>
  <string name="gr_fico">Gráfico</string>
  <string name="volver">Volver</string>
  <string name="mostrar_puntos">Mostrar puntos</string>
  <string name="fck">f<sub>ck</sub> [MPa] </string>
  <string name="go"> $\gamma$ <sub>o</sub> </string>
  <string name="gs"> $\gamma$ <sub>s</sub> </string>
  <string name="fyk">f<sub>y</sub>k [MPa] </string>
  <string name="ec">E<sub>c</sub> [MPa] </string>
  <string name="es">E<sub>s</sub> [MPa] </string>
  <string name="e">E<sub>s</sub> [MPa] </string>
  <string name="gt"> $\gamma$ <sub>t</sub> [N/m<sup>3</sup>] </string>
  <string name="coh">c<sub>oh</sub> [N/m<sup>2</sup>] </string>
  <string name="k03">K <sub>03</sub> [MN/m<sup>3</sup>] </string>
  <string name="b2">b<sub>2</sub> [m] </string>
  <string name="ht">h<sub>t</sub> [m] </string>
  <string name="sta">S<sub>ta</sub> [N/m<sup>2</sup>] </string>
  <string name="ga"> $\gamma$ <sub>a</sub> </string>
  <string name="gd"> $\gamma$ <sub>d</sub> </string>
  <string name="gv"> $\gamma$ <sub>v</sub> </string>
  <string name="delta"> $\delta$  </string>
  <string name="aa1">a<sub>1</sub> [m] </string>
  <string name="l">l [m] </string>
  <string name="r">r [m] </string>
```

```

<string name="p">p (muro) :</string>
<string name="en">e<sub>N</sub> [m]:</string>
<string name="aa2">a<sub>2</sub> [m]:</string>
<string name="hh">h [m]:</string>
<string name="wkMax">w<sub>kMáx</sub> [mm]:</string>
<string-array name="DOMINIOS">
  <item>No se ha encontrado el dominio</item>
  <item>2.a sin A<sub>2</sub></item>
  <item>2.b sin A<sub>2</sub></item>
  <item>3 sin A<sub>2</sub></item>
  <item>3 con A<sub>1</sub> y A<sub>2</sub></item>
  <item>3 sin A<sub>1</sub></item>
  <item>4 sin A<sub>1</sub> y con A<sub>2</sub></item>
  <item>4.a sin A<sub>1</sub> y con A<sub>2</sub></item>
  <item>5 sin A<sub>1</sub> y con A<sub>2</sub></item>
  <item>5 con A<sub>1</sub> y con A<sub>2</sub></item>
</string-array>
<string-array name="DIAMETROS_NORMAILIZADOS">
  <item>φ6</item>
  <item>φ8</item>
  <item>φ10</item>
  <item>φ12</item>
  <item>φ14</item>
  <item>φ16</item>
  <item>φ20</item>
  <item>φ25</item>
  <item>φ32</item>
  <item>φ40</item>
</string-array>
<string-array name="LIMITES_MU">
  <item></item>
  <item> $\mu_{d1} \leq \mu_{u1} \supseteq 2.a-2.b \supseteq 0.08854$ </item>
  <item> $\nu_{d1} \leq (\supseteq 1 \supseteq / \supseteq 300 \supseteq) (272\xi-17) = 0.1592$ </item>
  <item> $\mu_{d1} \leq \mu_{u1} \supseteq c \supseteq \supseteq \text{optimo} \supseteq$ </item><!--
String.valueOf(red(getMuOptimo(getP1Optima(delta2,ey),ey),4))-->
  <item> $\mu_{d1} \geq \mu_{u1} \supseteq c \supseteq \supseteq \text{optimo} \supseteq$ </item><!--
String.valueOf(red(getMuOptimo(getP1Optima(delta2,ey),ey),4))-->
  <item> $\mu_{d1} \geq \mu_{u1} \supseteq c \supseteq \supseteq \text{optimo} \supseteq$ </item><!--
String.valueOf(red(getMuOptimo(getP1Optima(delta2,ey),ey),4))-->
  <item> $\mu_{d1} \geq \mu_{u1} \supseteq c \supseteq (\xi \supseteq 3-4 \supseteq) =$ </item><!--
String.valueOf(red(getMu(getP1OptimaMaxima(ey),contadorDominio),4))-->
  <item> $\mu_{d1} \geq \mu_{u1} \supseteq c \supseteq (\xi \supseteq 4-4.a \supseteq) =$ </item><!--
String.valueOf(red(getMu(getP1OptimaMaxima(ey),contadorDominio),4))-->
  <item></item>
  <item> $\mu_{d1} \geq 0.85 \cdot \delta \cdot (1 - \delta \supseteq / \supseteq 2 \supseteq) =$ </item><!-- String.valueOf(red(0.85*delta*(1-delta/2),4))-->
</string-array>
<string-array name="LIMITES_NU">
  <!-- Voy a ponerlos limite primero en la posicion natural del contador y límite
segundo
en contador +10-->
  <item></item>
  <item> $\nu_{d1} \leq (\supseteq 17 \supseteq / \supseteq 12 \supseteq) \xi \supseteq 2 \supseteq (3-8\xi) / (1-\xi) \supseteq 2 \supseteq =$ </item><!--
String.valueOf(red(getLimiteNu(pln,contadorDominio),4))-->
  <item> $\nu_{d1} \leq (\supseteq 1 \supseteq / \supseteq 300 \supseteq) (272\xi-17) =$ </item><!--
String.valueOf(red(getLimiteNu(pln,contadorDominio),4))-->
  <item> $\nu_{d1} \leq (\supseteq 1 \supseteq / \supseteq 420 \supseteq) 289\xi =$ </item><!--
String.valueOf(red(getLimiteNu(pln,contadorDominio),4))-->
  <item> $\nu_{d1} \leq \xi \supseteq \text{Optimo} \supseteq \cdot \Psi + (1 - \delta \supseteq / \supseteq 2 \supseteq) \supseteq -1 \supseteq \cdot (\mu_{d1} \supseteq - \mu_{u1} \supseteq c \supseteq \supseteq \text{optimo} \supseteq) =$ </item><!--
String.valueOf(red(getLimiteNu(mul,delta2,delta1,ey,contadorDominio),4))-->
  <item> $\nu_{d1} \geq \xi \supseteq \text{Optimo} \supseteq \cdot \Psi + (1 - \delta \supseteq / \supseteq 2 \supseteq) \supseteq -1 \supseteq \cdot (\mu_{d1} \supseteq - \mu_{u1} \supseteq c \supseteq \supseteq \text{optimo} \supseteq) =$ </item><!--
String.valueOf(red(getLimiteNu(mul,delta2,delta,ey,contadorDominio-1),4))-->
  <item> $\nu_{d1} \geq \xi \supseteq 3-4 \supseteq \cdot \Psi + (1 -$ </item>

```

```


$$\delta^{2\epsilon} \sup_{\mu} (\mu^{d_1} - \mu^{u_1} \sup_c (\xi^{3-4\epsilon})) =$$

String.valueOf(red(getLimiteNu(mu1, delta2, delta, ey, contadorDominio-1),4));-->

$$\delta^{2\epsilon} \sup_{\mu} (\mu^{d_1} - \mu^{u_1} \sup_c (\xi^{4-4.a\epsilon})) =$$

String.valueOf(red(getLimiteNu(mu1, delta2, delta, ey, contadorDominio-1),4));-->

$$1 \sup_{\mu} (\mu^{d_1} - 0.85 \cdot \delta \cdot (1 - \delta^{2\epsilon}) \sup_{\mu} (\xi^{4-4.a\epsilon})) =$$

String.valueOf(red(0.85*delta+(mu1-0.85*delta*(1-delta/2))/(1-delta2),4));-->

</item>
</item>
</item>
</item>
</item>
</item>

$$v^{d\epsilon} \sup_{\mu} (\xi^{3-4\epsilon} + (1 - \delta^{2\epsilon}) \sup_{\mu} (\mu^{d_1} - \mu^{u_1} \sup_c (\xi^{3-4\epsilon}))) =$$

String.valueOf(red(getLimiteNu(mu1, delta2, delta, ey, contadorDominio),4));-->

$$v^{d\epsilon} \sup_{\mu} (\xi^{4-4.a\epsilon} + (1 - \delta^{2\epsilon}) \sup_{\mu} (\mu^{d_1} - \mu^{u_1} \sup_c (\xi^{4-4.a\epsilon}))) =$$

String.valueOf(red(getLimiteNu(mu1, delta2, delta, ey, contadorDominio),4));-->

$$v^{d\epsilon} \sup_{\mu} (\xi^{4.a-5\epsilon} + (1 - \delta^{2\epsilon}) \sup_{\mu} (\mu^{d_1} - \mu^{u_1} \sup_c (\xi^{4.a-5\epsilon}))) =$$

String.valueOf(red(getLimiteNu(mu1, delta2, delta, ey, contadorDominio),4));-->
</item>
</item>
<!-- String.valueOf(red(delta,4));-->
</string-array>
<string-array name="LIMITES_PLN">
<item>
<math display="block">\sup_{\xi} \sup_{\mu} \xi \sup_{\mu} \xi^6 \leq \xi \sup_{\mu} \xi^6 \sup_{\mu} \xi^7 \leq \xi \sup_{\mu} \xi^6 \sup_{\mu} \xi \leq \xi \sup_{\mu} \xi^6 \sup_{\mu} \xi \leq \xi \sup_{\mu} \xi^6 \sup_{\mu} \xi \leq 1
<!-- +String.valueOf(red(getP1Optima(arma[2], arma[0]),4));-->
<!-- +String.valueOf(red(getP1Optima(arma[2], arma[0]),4));-->
<!-- +String.valueOf(red(getP1Optima(arma[2], arma[0]),4));-->
<!-- +String.valueOf(red(arma[1],4));-->
</item>
<math display="block">\xi \geq \delta =
<!-- " +String.valueOf(red(arma[1],4));-->
</string-array>
</resources>

```

-activity-main.XML:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/parent"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:baselineAligned="false"
    android:divider="?android:attr/dividerHorizontal"
    android:orientation="horizontal"
    android:showDividers="middle">
    <fragment
        android:id="@+id/fragment$activity_main$fragmentEstatico"
        android:name="com.example.jalab.interfaz.EstaticoFragment"
        android:layout_width="332dp"
        android:layout_height="match_parent"
        tools:layout="@layout/fragment_estatico" />
    <FrameLayout
        android:id="@+id/frameLayout$activity_main$contenedorFragmentsDinamicos"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1" />
</LinearLayout>
```

-dinamico_fragment_calculo_comprobaciones.XML:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="wrap_content"
        android:layout_height="match_parent">

        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:text=" Dimensiones: "
                android:layout_marginLeft="30dp"
                android:layout_marginTop="30dp"
                android:textSize="20dp" />

        </LinearLayout>

        <View
            android:layout_width="fill_parent"
            android:layout_height="2dp"
            android:background="@android:color/black"
            android:layout_marginTop="10dp"
            android:layout_marginLeft="20dp" />

        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal">

            <TextView
                android:layout_width="80dp"
```

```

        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/aa2"
        android:layout_marginTop="10dp"
        android:textSize="12dp" />

<TextView
    android:layout_width="75dp"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="10dp"
    android:id="@+id/ancho"
    android:textSize="12dp" />

</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/h"
        android:layout_marginTop="5dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="5dp"
        android:id="@+id/canto"
        android:textSize="12dp" />

</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="d [m] :"
        android:layout_marginTop="5dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="5dp"
        android:id="@+id/d"
        android:textSize="12dp" />

</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="v [m] :"

```

```

        android:layout_marginTop="5dp"
        android:textSize="12dp"
        android:id="@+id/Vuel" />

<TextView
    android:layout_width="75dp"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="5dp"
    android:id="@+id/V"
    android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Tipo de Zapata:"
        android:layout_marginTop="5dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="5dp"
        android:id="@+id/tipo"
        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Amin de Acero:"
        android:layout_marginTop="5dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="5dp"
        android:id="@+id/Am"
        android:textSize="12dp" />
</LinearLayout>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Armadura principal : "
    android:textSize="12dp"
    android:layout_marginLeft="80dp"
    android:layout_marginTop="10dp" />

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

```

```

<TextView
    android:layout_width="80dp"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Àrea Teórica:"
    android:layout_marginTop="10dp"
    android:textSize="12dp" />

<TextView
    android:layout_width="75dp"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="10dp"
    android:id="@+id/At"
    android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

<TextView
    android:layout_width="80dp"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Àrea Real:"
    android:layout_marginTop="5dp"
    android:textSize="12dp" />

<TextView
    android:layout_width="75dp"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="5dp"
    android:id="@+id/Ar"
    android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

<TextView
    android:layout_width="80dp"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Ø1:"
    android:layout_marginTop="5dp"
    android:textSize="12dp" />

<TextView
    android:layout_width="75dp"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="5dp"
    android:id="@+id/NR"
    android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

<LinearLayout
    android:orientation="vertical"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

<TextView

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Armadura de reparto : "
        android:textSize="12dp"
        android:layout_marginLeft="30dp"
        android:layout_marginTop="10dp" />

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginLeft="10dp">

    <TextView
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Área Teórica:"
        android:layout_marginTop="5dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="5dp"
        android:layout_marginTop="5dp"
        android:id="@+id/At2"
        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginLeft="10dp">

    <TextView
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Area Real:"
        android:layout_marginTop="5dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="5dp"
        android:layout_marginTop="5dp"
        android:id="@+id/Ar2"
        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginLeft="10dp">

    <TextView
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Ø2:"
        android:layout_marginTop="5dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"

```



```

        android:layout_marginLeft="5dp"
        android:layout_marginTop="5dp"
        android:id="@+id/NR2"
        android:textSize="12dp" />
    </LinearLayout>
</LinearLayout>

<LinearLayout
    android:orientation="vertical"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Armadura en espera :"
        android:textSize="12dp"
        android:layout_marginLeft="0dp"
        android:layout_marginTop="10dp" />

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal" >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="L1"
            android:layout_marginTop="5dp"
            android:textSize="12dp"
            android:layout_marginLeft="10dp" />

        <TextView
            android:layout_width="75dp"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:layout_marginLeft="20dp"
            android:layout_marginTop="5dp"
            android:id="@+id/L1es"
            android:textSize="12dp" />
    </LinearLayout>

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal" >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="L2"
            android:layout_marginTop="5dp"
            android:textSize="12dp"
            android:layout_marginLeft="10dp" />

        <TextView
            android:layout_width="75dp"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:layout_marginLeft="20dp"
            android:layout_marginTop="5dp"
            android:id="@+id/L2es"
            android:textSize="12dp" />
    </LinearLayout>

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal" >

```

```

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="L3"
            android:layout_marginTop="5dp"
            android:textSize="12dp"
            android:layout_marginLeft="10dp" />

        <TextView
            android:layout_width="75dp"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:layout_marginLeft="20dp"
            android:layout_marginTop="5dp"
            android:id="@+id/L3es"
            android:textSize="12dp" />
    </LinearLayout>
</LinearLayout>
</LinearLayout>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Seguridad"
    android:layout_marginLeft="30dp"
    android:layout_marginTop="20dp"
    android:textSize="20dp" />
<View
    android:layout_width="fill_parent"
    android:layout_height="2dp"
    android:background="@android:color/black"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="20dp" />

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Hundimiento: "
        android:layout_marginLeft="45dp"
        android:layout_marginTop="10dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="10dp"
        android:id="@+id/hund"
        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Cortante :"
        android:layout_marginLeft="45dp"
        android:layout_marginTop="10dp"
        android:textSize="12dp" />

    <TextView

```

```

        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="10dp"
        android:id="@+id/cortante"
        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Rigidez Relativa Suelo-Cimiento:"
        android:layout_marginLeft="45dp"
        android:layout_marginTop="10dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="10dp"
        android:id="@+id/rigidez"
        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Vuelco:"
        android:layout_marginLeft="45dp"
        android:layout_marginTop="10dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="10dp"
        android:id="@+id/vuelco"
        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Deslizamiento:"
        android:layout_marginLeft="45dp"
        android:layout_marginTop="10dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"

```

```

        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="10dp"
        android:id="@+id/deslizamiento"
        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Cortante Definitiva"
        android:layout_marginLeft="45dp"
        android:layout_marginTop="10dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="10dp"
        android:id="@+id/cortanteDef"
        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="45dp"
        android:layout_marginTop="10dp"
        android:textSize="12dp"
        android:id="@+id/fisuracion" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="10dp"
        android:id="@+id/segF"
        android:textSize="12dp" />
</LinearLayout>

</LinearLayout>
<View
    android:layout_width="2dp"
    android:layout_height="fill_parent"
    android:background="@android:color/black"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="20dp" />

<LinearLayout
    android:orientation="vertical"
    android:layout_width="wrap_content"
    android:layout_height="match_parent">

    <ImageView
        android:layout_width="650dp"
        android:layout_height="406dp"
        android:id="@+id/zampata" />

    <View
        android:layout_width="fill_parent"

```

```

        android:layout_height="2dp"
        android:background="@android:color/black"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="0dp" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Anclaje :"
    android:layout_marginTop="20dp"
    android:textSize="20dp"
    android:layout_marginLeft="20dp" />
<View
    android:layout_width="fill_parent"
    android:layout_height="2dp"
    android:background="@android:color/black"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="0dp" />

<View
    android:layout_width="fill_parent"
    android:layout_height="2dp"
    android:background="@android:color/black"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="0dp" />

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:id="@+id/anclaje"
        android:layout_marginLeft="25dp" />

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content">

            <TextView
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:id="@+id/tipoAnclaje"
                android:gravity="center_horizontal"
                android:layout_marginTop="10dp"
                android:textSize="14dp" />

            <View
                android:layout_width="fill_parent"
                android:layout_height="1dp"
                android:background="@android:color/black"
                android:layout_marginTop="10dp"
                android:layout_marginLeft="10dp"
                android:layout_marginRight="10dp"
                android:layout_gravity="left" />

        </LinearLayout>

    </LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="175dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="lbAnclaje :"

```

```

        android:layout_marginLeft="10dp"
        android:layout_marginTop="5dp"
        android:textSize="14dp" />

<TextView
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:id="@+id/lb"
    android:textSize="14dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="175dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="lnAnclaje :"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="5dp"
        android:textSize="14dp" />

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/ln"
        android:textSize="14dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="175dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="lDisponible :"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="5dp"
        android:textSize="14dp" />

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/ld"
        android:textSize="14dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="175dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="lProlongacionGancho"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="5dp"
        android:textSize="14dp" />

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/l1"
        android:textSize="14dp" />

```

```
        </LinearLayout>
    </LinearLayout>
</LinearLayout>
<View
    android:layout_width="fill_parent"
    android:layout_height="2dp"
    android:background="@android:color/black"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="0dp" />
</LinearLayout>
<View
    android:layout_width="2dp"
    android:layout_height="fill_parent"
    android:background="@android:color/black"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="0dp" />
</LinearLayout>
```

-dinamico_fragment_calculos_dimensionado.XML:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="wrap_content"
        android:layout_height="match_parent">

        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:text=" Dimensiones: "
                android:layout_marginLeft="30dp"
                android:layout_marginTop="30dp"
                android:textSize="20dp" />

        </LinearLayout>

        <View
            android:layout_width="fill_parent"
            android:layout_height="2dp"
            android:background="@android:color/black"
            android:layout_marginTop="10dp"
            android:layout_marginLeft="20dp" />

        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal">

            <TextView
                android:layout_width="80dp"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:text="@string/aa2"
                android:layout_marginTop="10dp"
                android:textSize="12dp" />

            <TextView
                android:layout_width="75dp"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:layout_marginLeft="20dp"
                android:layout_marginTop="10dp"
                android:id="@+id/ancho"
                android:textSize="12dp" />

        </LinearLayout>

        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal" >

            <TextView
                android:layout_width="80dp"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:text="@string/h"
                android:layout_marginTop="5dp"
                android:textSize="12dp" />

            <TextView
```



```

        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="5dp"
        android:id="@+id/canto"
        android:textSize="12dp" />
</LinearLayout>
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >
    <TextView
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="d [m] :"
        android:layout_marginTop="5dp"
        android:textSize="12dp" />
    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="5dp"
        android:id="@+id/d"
        android:textSize="12dp" />
</LinearLayout>
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >
    <TextView
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="v [m] :"
        android:layout_marginTop="5dp"
        android:textSize="12dp"
        android:id="@+id/Vuel" />
    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="5dp"
        android:id="@+id/v"
        android:textSize="12dp" />
</LinearLayout>
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >
    <TextView
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Tipo de Zapata:"
        android:layout_marginTop="5dp"
        android:textSize="12dp" />
    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"

```

```

        android:layout_marginLeft="20dp"
        android:layout_marginTop="5dp"
        android:id="@+id/tipo"
        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Amin de Acero:"
        android:layout_marginTop="5dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="5dp"
        android:id="@+id/Am"
        android:textSize="12dp" />
</LinearLayout>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Armadura principal :"
    android:textSize="12dp"
    android:layout_marginLeft="30dp"
    android:layout_marginTop="10dp" />

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Área Teórica:"
        android:layout_marginTop="10dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="10dp"
        android:id="@+id/At"
        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Area Real:"
        android:layout_marginTop="5dp"
        android:textSize="12dp" />

```

```

<TextView
    android:layout_width="75dp"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="5dp"
    android:id="@+id/Ar"
    android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Ø1:"
        android:layout_marginTop="5dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="5dp"
        android:id="@+id/NR"
        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="Armadura de reparto :"
            android:textSize="12dp"
            android:layout_marginLeft="30dp"
            android:layout_marginTop="10dp" />

        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal" >

            <TextView
                android:layout_width="80dp"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:text="Área Teórica:"
                android:layout_marginTop="5dp"
                android:textSize="12dp"
                android:layout_marginLeft="10dp" />

            <TextView
                android:layout_width="75dp"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:layout_marginLeft="5dp"
                android:layout_marginTop="5dp"
                android:id="@+id/At2"
                android:textSize="12dp" />
        </LinearLayout>
    </LinearLayout>

```

```

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Area Real:"
        android:layout_marginTop="5dp"
        android:textSize="12dp"
        android:layout_marginLeft="10dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="5dp"
        android:layout_marginTop="5dp"
        android:id="@+id/Ar2"
        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Ø2:"
        android:layout_marginTop="5dp"
        android:textSize="12dp"
        android:layout_marginLeft="10dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="5dp"
        android:layout_marginTop="5dp"
        android:id="@+id/NR2"
        android:textSize="12dp" />
</LinearLayout>
</LinearLayout>

<LinearLayout
    android:orientation="vertical"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Armadura en espera :"
        android:textSize="12dp"
        android:layout_marginLeft="0dp"
        android:layout_marginTop="10dp"
        android:id="@+id/textView" />

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal" >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"

```

```

        android:text="L1"
        android:layout_marginTop="5dp"
        android:textSize="12dp"
        android:layout_marginLeft="10dp" />

<TextView
    android:layout_width="75dp"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="5dp"
    android:id="@+id/L1es"
    android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="L2"
    android:layout_marginTop="5dp"
    android:textSize="12dp"
    android:layout_marginLeft="10dp" />

<TextView
    android:layout_width="75dp"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="5dp"
    android:id="@+id/L2es"
    android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="L3"
    android:layout_marginTop="5dp"
    android:textSize="12dp"
    android:layout_marginLeft="10dp" />

<TextView
    android:layout_width="75dp"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="5dp"
    android:id="@+id/L3es"
    android:textSize="12dp" />
</LinearLayout>
</LinearLayout>
</LinearLayout>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Seguridad"
    android:layout_marginLeft="30dp"
    android:layout_marginTop="20dp"
    android:textSize="20dp" />
<View
    android:layout_width="fill_parent"

```

```

        android:layout_height="2dp"
        android:background="@android:color/black"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="20dp" />

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Hundimiento: "
        android:layout_marginLeft="45dp"
        android:layout_marginTop="10dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="10dp"
        android:id="@+id/hund"
        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Cortante : "
        android:layout_marginLeft="45dp"
        android:layout_marginTop="10dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="10dp"
        android:id="@+id/cortante"
        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Rigidez Relativa Suelo-Cimiento:"
        android:layout_marginLeft="45dp"
        android:layout_marginTop="10dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="10dp"
        android:id="@+id/rigidez"

```

```

        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Vuelco:"
        android:layout_marginLeft="45dp"
        android:layout_marginTop="10dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="10dp"
        android:id="@+id/vuelco"
        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Deslizamiento:"
        android:layout_marginLeft="45dp"
        android:layout_marginTop="10dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="10dp"
        android:id="@+id/deslizamiento"
        android:textSize="12dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Cortante Definitiva"
        android:layout_marginLeft="45dp"
        android:layout_marginTop="10dp"
        android:textSize="12dp" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="10dp"
        android:id="@+id/cortanteDef"
        android:textSize="12dp" />
</LinearLayout>

```

```

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="45dp"
        android:layout_marginTop="10dp"
        android:textSize="12dp"
        android:id="@+id/fisuracion" />

    <TextView
        android:layout_width="75dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="10dp"
        android:id="@+id/segF"
        android:textSize="12dp" />
</LinearLayout>

</LinearLayout>
<View
    android:layout_width="2dp"
    android:layout_height="fill_parent"
    android:background="@android:color/black"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="20dp" />

<LinearLayout
    android:orientation="vertical"
    android:layout_width="wrap_content"
    android:layout_height="match_parent">

    <ImageView
        android:layout_width="650dp"
        android:layout_height="406dp"
        android:id="@+id/zampata" />

    <View
        android:layout_width="fill_parent"
        android:layout_height="2dp"
        android:background="@android:color/black"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="0dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Anclaje : "
        android:layout_marginTop="20dp"
        android:textSize="20dp"
        android:layout_marginLeft="20dp" />

    <View
        android:layout_width="fill_parent"
        android:layout_height="2dp"
        android:background="@android:color/black"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="0dp" />

    <View
        android:layout_width="fill_parent"
        android:layout_height="2dp"
        android:background="@android:color/black"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="0dp" />

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

```



```

<ImageView
    android:layout_width="200dp"
    android:layout_height="200dp"
    android:id="@+id/anclaje"
    android:layout_marginLeft="25dp" />

<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:id="@+id/tipoAnclaje"
            android:gravity="center_horizontal"
            android:layout_marginTop="10dp"
            android:textSize="14dp" />

        <View
            android:layout_width="fill_parent"
            android:layout_height="1dp"
            android:background="@android:color/black"
            android:layout_marginTop="10dp"
            android:layout_marginLeft="10dp"
            android:layout_marginRight="10dp"
            android:layout_gravity="left" />

    </LinearLayout>

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="175dp"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="lbAnclaje :"
            android:layout_marginLeft="10dp"
            android:layout_marginTop="5dp"
            android:textSize="14dp" />

        <TextView
            android:layout_width="150dp"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:id="@+id/lb"
            android:textSize="14dp" />

    </LinearLayout>

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="175dp"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="lnAnclaje :"
            android:layout_marginLeft="10dp"
            android:layout_marginTop="5dp"
            android:textSize="14dp" />

        <TextView
            android:layout_width="150dp"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:id="@+id/ln"

```

```

        android:textSize="14dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="175dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="lDisponible :"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="5dp"
        android:textSize="14dp" />

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/l1d"
        android:textSize="14dp" />

</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="175dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="lProlongacionGancho"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="5dp"
        android:textSize="14dp" />

    <TextView
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/l11"
        android:textSize="14dp" />

</LinearLayout>

</LinearLayout>

</LinearLayout>

<View
    android:layout_width="fill_parent"
    android:layout_height="2dp"
    android:background="@android:color/black"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="0dp" />

</LinearLayout>
<View
    android:layout_width="2dp"
    android:layout_height="fill_parent"
    android:background="@android:color/black"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="0dp" />

</LinearLayout>

```

-dinamico_fragment_comprobacion.XML:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="fill_parent"
        android:layout_gravity="center_horizontal">

        <LinearLayout
            android:orientation="vertical"
            android:layout_width="wrap_content"
            android:layout_height="match_parent">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:text="DATOS CARGADOS"
                android:layout_marginLeft="20dp"
                android:layout_marginTop="10dp"
                android:layout_marginRight="10dp"
                android:layout_marginBottom="10dp"
                android:textSize="40dp" />

            <View
                android:layout_width="fill_parent"
                android:layout_height="2dp"
                android:background="@android:color/black"/>

            <LinearLayout
                android:orientation="horizontal"
                android:layout_width="wrap_content"
                android:layout_height="match_parent">

                <Button
                    android:layout_width="wrap_content"
                    android:layout_height="fill_parent"
                    android:text="@string/nuevosDatos"
                    android:id="@+id/irNuevosDatos"
                    android:layout_marginLeft="20dp"
                    android:textSize="25dp"
                    android:layout_marginRight="10dp" />

                <View
                    android:layout_height="fill_parent"
                    android:layout_width="1dp"
                    android:background="@android:color/black"
                    android:layout_marginTop="5dp"
                    android:layout_marginBottom="5dp" />

                <LinearLayout
                    android:orientation="vertical"
                    android:layout_width="wrap_content"
                    android:layout_height="match_parent"
                    android:layout_marginLeft="20dp"
                    android:layout_marginRight="20dp">

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:textAppearance="?android:attr/textAppearanceLarge"
                        android:text="Esfuerzos Nominales"
                        android:layout_marginLeft="10dp"
                        android:layout_marginTop="10dp"
                        android:textSize="30dp" />

                    <View
                        android:layout_width="fill_parent"
                        android:layout_height="1dp"
                        android:background="@android:color/black"/>

                    <TextView
                        android:layout_width="fill_parent"
```

```

        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/esfN"
        android:text="No se han cargado datos"
        android:layout_gravity="center_vertical"
        android:gravity="center_vertical|center_horizontal"
        android:layout_marginTop="10dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_marginLeft="10dp"
    android:text="Hormigon"
    android:layout_marginTop="20dp"
    android:textSize="30dp" />

<View
    android:layout_width="fill_parent"
    android:layout_height="1dp"
    android:background="@android:color/black"/>

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="No se han cargado datos"
    android:gravity="center_vertical|center_horizontal"
    android:id="@+id/horm"
    android:layout_marginTop="10dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Acero"
    android:layout_marginLeft="10dp"
    android:layout_marginTop="20dp"
    android:textSize="30dp" />

<View
    android:layout_width="fill_parent"
    android:layout_height="1dp"
    android:background="@android:color/black"/>

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="No se han cargado datos"
    android:id="@+id/ace"
    android:gravity="center_vertical|center_horizontal"
    android:layout_marginTop="10dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Suelo"
    android:layout_marginLeft="10dp"
    android:layout_marginTop="20dp"
    android:textSize="30dp" />

<View
    android:layout_width="fill_parent"
    android:layout_height="1dp"
    android:background="@android:color/black"/>

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="No se han cargado datos"
    android:id="@+id/suel"
    android:gravity="center_vertical|center_horizontal"
    android:layout_marginTop="10dp" />

<TextView
    android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Geometria"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="20dp"
        android:textSize="30dp" />
<View
    android:layout_width="fill_parent"
    android:layout_height="1dp"
    android:background="@android:color/black" />

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="No se han cargado datos"
    android:id="@+id/geom"
    android:gravity="center_vertical|center_horizontal"
    android:layout_marginTop="10dp" />

</LinearLayout>
<View
    android:layout_height="fill_parent"
    android:layout_width="1dp"
    android:background="@android:color/black"
    android:layout_marginTop="5dp"
    android:layout_marginBottom="5dp" />

<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="10dp">

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <LinearLayout
            android:orientation="vertical"
            android:layout_width="298dp"
            android:layout_height="wrap_content">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

                android:textAppearance="?android:attr/textAppearanceLarge"
                android:text="Dimensiones :"
                android:layout_marginTop="50dp" />

            <LinearLayout
                android:orientation="horizontal"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_gravity="center_horizontal">

                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="50dp"

                    android:textAppearance="?android:attr/textAppearanceLarge"
                    android:text="@string/aa2"
                    android:layout_marginLeft="30dp"
                    android:gravity="center_vertical" />

                <EditText
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:inputType="numberDecimal"
                    android:ems="10"
                    android:id="@+id/a2"
                    android:layout_weight="1" />

            </LinearLayout>

        </LinearLayout>

    </LinearLayout>

```

```

        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="50dp"

android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="@string/hh"
            android:layout_marginLeft="30dp"
            android:gravity="center_vertical" />

        <EditText
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:inputType="numberDecimal"
            android:ems="10"
            android:id="@+id/h"
            android:layout_weight="1" />
    </LinearLayout>

    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Guardar"
        android:id="@+id/guardarDimensiones" />
</LinearLayout>

<ListView
    android:layout_width="wrap_content"
    android:layout_height="fill_parent"
    android:id="@+id/dimensionesLW"
    android:layout_marginTop="15dp" />
</LinearLayout>

<ImageView
    android:layout_width="496dp"
    android:layout_height="261dp"
    android:id="@+id/imageView"
    android:background="@drawable/zapata"
    android:layout_marginTop="10dp" />

<Button
    android:layout_width="fill_parent"
    android:layout_height="100dp"
    android:text="Calcular"
    android:id="@+id/irCalculosDim"
    android:layout_gravity="center_horizontal"
    android:textSize="25dp" />

</LinearLayout>

</LinearLayout>

</LinearLayout>

</LinearLayout>

```

-dinamico_fragment_dimensionado.XML:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="fill_parent"
        android:layout_gravity="center_horizontal">

        <LinearLayout
            android:orientation="vertical"
            android:layout_width="wrap_content"
            android:layout_height="match_parent">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:text="DATOS CARGADOS"
                android:layout_marginLeft="20dp"
                android:layout_marginTop="10dp"
                android:layout_marginRight="10dp"
                android:layout_marginBottom="10dp"
                android:textSize="40dp" />

            <View
                android:layout_width="fill_parent"
                android:layout_height="2dp"
                android:background="@android:color/black"/>

            <LinearLayout
                android:orientation="horizontal"
                android:layout_width="wrap_content"
                android:layout_height="match_parent">

                <Button
                    android:layout_width="wrap_content"
                    android:layout_height="fill_parent"
                    android:text="@string/nuevosDatos"
                    android:id="@+id/irNuevosDatos"
                    android:layout_marginLeft="20dp"
                    android:textSize="25dp"
                    android:layout_marginRight="10dp" />

                <View
                    android:layout_height="fill_parent"
                    android:layout_width="1dp"
                    android:background="@android:color/black"
                    android:layout_marginTop="5dp"
                    android:layout_marginBottom="5dp" />

                <LinearLayout
                    android:orientation="vertical"
                    android:layout_width="wrap_content"
                    android:layout_height="match_parent"
                    android:layout_marginLeft="20dp"
                    android:layout_marginRight="20dp">

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:textAppearance="?android:attr/textAppearanceLarge"
                        android:text="Esfuerzos Nominales"
                        android:layout_marginLeft="10dp"
                        android:layout_marginTop="10dp"
                        android:textSize="30dp" />

                    <View
                        android:layout_width="fill_parent"
                        android:layout_height="1dp"
                        android:background="@android:color/black"/>

                    <TextView
                        android:layout_width="fill_parent"
```

```

        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/esfN"
        android:text="No se han cargado datos"
        android:layout_gravity="center_vertical"
        android:gravity="center_vertical|center_horizontal"
        android:layout_marginTop="10dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_marginLeft="10dp"
    android:text="Hormigon"
    android:layout_marginTop="20dp"
    android:textSize="30dp" />

<View
    android:layout_width="fill_parent"
    android:layout_height="1dp"
    android:background="@android:color/black"/>

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="No se han cargado datos"
    android:gravity="center_vertical|center_horizontal"
    android:id="@+id/horm"
    android:layout_marginTop="10dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Acero"
    android:layout_marginLeft="10dp"
    android:layout_marginTop="20dp"
    android:textSize="30dp" />

<View
    android:layout_width="fill_parent"
    android:layout_height="1dp"
    android:background="@android:color/black"/>

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="No se han cargado datos"
    android:id="@+id/ace"
    android:gravity="center_vertical|center_horizontal"
    android:layout_marginTop="10dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Suelo"
    android:layout_marginLeft="10dp"
    android:layout_marginTop="20dp"
    android:textSize="30dp" />

<View
    android:layout_width="fill_parent"
    android:layout_height="1dp"
    android:background="@android:color/black"/>

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="No se han cargado datos"
    android:id="@+id/suel"
    android:gravity="center_vertical|center_horizontal"
    android:layout_marginTop="10dp" />

<TextView
    android:layout_width="wrap_content"

```



```

        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Geometria"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="20dp"
        android:textSize="30dp" />
<View
    android:layout_width="fill_parent"
    android:layout_height="1dp"
    android:background="@android:color/black"/>

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="No se han cargado datos"
    android:id="@+id/geom"
    android:gravity="center_vertical|center_horizontal"
    android:layout_marginTop="10dp" />

</LinearLayout>
<View
    android:layout_height="fill_parent"
    android:layout_width="1dp"
    android:background="@android:color/black"
    android:layout_marginTop="5dp"
    android:layout_marginBottom="5dp" />

<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="10dp">

    <ImageView
        android:layout_width="496dp"
        android:layout_height="261dp"
        android:id="@+id/imageView"
        android:background="@drawable/zapata" />

    <Button
        android:layout_width="fill_parent"
        android:layout_height="100dp"
        android:text="Calcular"
        android:id="@+id/irCalculosDim"
        android:layout_gravity="center_horizontal"
        android:textSize="25dp" />
</LinearLayout>

</LinearLayout>

</LinearLayout>
</LinearLayout>
</LinearLayout>

```

-dinamico_fragment_grafico.XML:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="Datos Guardados"
            android:layout_marginLeft="20dp"
            android:layout_marginTop="20dp" />

    </LinearLayout>

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:weightSum="1">

        <LinearLayout
            android:orientation="vertical"
            android:layout_width="247dp"
            android:layout_height="match_parent">
            <View
                android:layout_width="fill_parent"
                android:layout_height="2dp"
                android:background="@android:color/black"/>
            <Button
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:text="@string/mostrar_puntos"
                android:id="@+id/mostrarPuntos"
                android:layout_gravity="center_horizontal"
                android:layout_margin="5dp" />

            <ListView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:id="@+id/limitesLW"
                android:layout_gravity="center_horizontal"
                android:layout_marginLeft="10dp" />
        </LinearLayout>
        <View
            android:layout_width="2dp"
            android:layout_height="fill_parent"
            android:background="@android:color/black"/>
        <com.github.mikephil.charting.charts.LineChart
            android:id="@+id/chart"
            android:layout_width="578dp"
            android:layout_height="match_parent" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="fill_parent"
            android:text="@string/volver"
            android:id="@+id/volverCalculos"
            android:layout_gravity="center_vertical" />

    </LinearLayout>

</LinearLayout>
```

-dinamico_fragment_nuevosdatosdim.XML:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:orientation="vertical"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_marginLeft="20dp">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:text="Datos Guardados:"
                android:layout_margin="20dp"
                android:textSize="25dp" />

            <View
                android:layout_width="fill_parent"
                android:layout_height="2dp"
                android:background="@android:color/black"
                android:layout_margin="5dp" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:text="Esfuerzos Nominales" />

            <ListView
                android:layout_width="match_parent"
                android:layout_height="75dp"
                android:id="@+id/esfNominaleLW" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:text="Hormigon"
                android:layout_gravity="center_horizontal" />

            <ListView
                android:layout_width="match_parent"
                android:layout_height="75dp"
                android:id="@+id/hormigonLW"
                android:layout_gravity="center_horizontal" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:text="Acero"
                android:layout_gravity="center_horizontal" />

            <ListView
                android:layout_width="match_parent"
                android:layout_height="75dp"
                android:id="@+id/aceroLW"
                android:layout_gravity="center_horizontal" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
```

```

        android:text="Suelo"
        android:layout_gravity="center_horizontal" />

<ListView
    android:layout_width="match_parent"
    android:layout_height="75dp"
    android:id="@+id/sueloLW"
    android:layout_gravity="center_horizontal" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Geometria"
    android:layout_gravity="center_horizontal" />

<ListView
    android:layout_width="match_parent"
    android:layout_height="75dp"
    android:id="@+id/geometriaLW"
    android:layout_gravity="center_horizontal" />

</LinearLayout>

<View
    android:layout_width="2dp"
    android:layout_height="fill_parent"
    android:background="@android:color/black"
    android:layout_margin="5dp" />

<HorizontalScrollView
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:id="@+id/horizontalScrollView" >

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="Esfuerzos Nominales:"
            android:layout_marginLeft="10dp" />

        <ImageView
            android:layout_width="250dp"
            android:layout_height="200dp"
            android:id="@+id/imageView3"
            android:background="@drawable/esfuerzos_nominales"
            android:layout_margin="10dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="Nuevos Datos:"
            android:layout_gravity="center_horizontal"
            android:id="@+id/textView2"
            android:layout_marginTop="8dp"
            android:layout_marginBottom="45dp" />

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"

```

```

        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Nombre: "
        android:layout_marginLeft="20dp" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/nombreEsf"
    android:layout_weight="1" />

</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="100dp"
        android:layout_height="50dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="N [N]: "
        android:layout_marginLeft="20dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberSigned"
        android:ems="10"
        android:id="@+id/N"
        android:layout_weight="1"
        android:layout_marginRight="10dp" />

</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="100dp"
        android:layout_height="50dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="V [N]: "
        android:layout_marginLeft="20dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberSigned"
        android:ems="10"
        android:id="@+id/V"
        android:layout_weight="1"
        android:layout_marginRight="10dp" />

</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="100dp"
        android:layout_height="50dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="M [N m]: "
        android:layout_marginLeft="20dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberSigned"
        android:ems="10"
        android:id="@+id/M"

```

```

        android:layout_weight="1"
        android:layout_marginRight="10dp" />
</LinearLayout>

<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Guardar"
    android:id="@+id/guardarEsfNominales"
    android:layout_marginTop="10dp" />

</LinearLayout>

<View
    android:layout_width="1dp"
    android:layout_height="fill_parent"
    android:background="@android:color/black"
    android:layout_margin="5dp" />

<LinearLayout
    android:orientation="vertical"
    android:layout_width="280dp"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Hormigon"
        android:layout_gravity="center_horizontal" />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="200dp"
        android:id="@+id/imageView2"
        android:layout_gravity="center_horizontal"
        android:background="@drawable/hormigon"
        android:layout_marginTop="10dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Nuevos Datos:"
        android:id="@+id/textView4"
        android:layout_gravity="center_horizontal"
        android:layout_marginBottom="45dp"
        android:layout_marginTop="28dp" />

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:layout_width="115dp"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="@string/nombre"
            android:layout_marginLeft="10dp" />

        <EditText
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/nomHorm"
            android:layout_weight="1" />

    </LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="115dp"
        android:layout_height="50dp"

```

```

        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/fck"
        android:layout_marginLeft="10dp"
        android:gravity="center_vertical" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:ems="10"
    android:id="@+id/fck"
    android:layout_weight="1"
    android:layout_marginRight="10dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="115dp"
        android:layout_height="50dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/ec"
        android:layout_marginLeft="10dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/Ec"
        android:layout_weight="1"
        android:layout_marginRight="10dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="115dp"
        android:layout_height="50dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/gh"
        android:layout_marginLeft="10dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/Gh"
        android:layout_weight="1"
        android:layout_marginRight="10dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="115dp"
        android:layout_height="50dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/go"
        android:layout_marginLeft="10dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

```

```

        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/Go"
        android:layout_weight="1"
        android:layout_marginRight="10dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="115dp"
        android:layout_height="50dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/wkMax"
        android:layout_marginLeft="10dp"
        android:gravity="center_vertical"
        android:textSize="16dp" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/wkMax"
        android:layout_weight="1"
        android:layout_marginRight="10dp" />
</LinearLayout>

<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/guardar"
    android:id="@+id/guardarHormi"
    android:layout_marginRight="5dp"
    android:layout_marginLeft="5dp" />

</LinearLayout>

<View
    android:layout_width="1dp"
    android:layout_height="fill_parent"
    android:background="@android:color/black"
    android:layout_margin="5dp" />

<LinearLayout
    android:orientation="vertical"
    android:layout_width="280dp"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Acero"
        android:layout_gravity="center_horizontal" />

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:id="@+id/imageView4"
        android:layout_gravity="center_horizontal"
        android:background="@drawable/acero"
        android:layout_marginTop="10dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Nuevos Datos:"
        android:layout_marginLeft="10dp"
        android:id="@+id/textView3"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="28dp"
        android:layout_marginBottom="35dp" />

```



```

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="115dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/nombre"
        android:layout_marginLeft="20dp" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/nomAce"
        android:layout_weight="1" />

</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="115dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Tipo :"
        android:layout_marginLeft="20dp" />

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <CheckBox
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="B-400"
            android:id="@+id/checkB400"
            android:checked="true" />

        <CheckBox
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="B-500"
            android:id="@+id/checkB500"
            android:checked="false" />

    </LinearLayout>

</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="115dp"
        android:layout_height="50dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/fyk"
        android:layout_marginLeft="20dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/fyk"
        android:layout_weight="1"
        android:layout_marginRight="10dp" />

</LinearLayout>

```

```

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="115dp"
        android:layout_height="50dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/e"
        android:layout_marginLeft="20dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/Es"
        android:layout_weight="1"
        android:layout_marginRight="10dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="115dp"
        android:layout_height="50dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/gS"
        android:layout_marginLeft="20dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/Gs"
        android:layout_weight="1"
        android:layout_marginRight="10dp" />
</LinearLayout>

<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/guardar"
    android:id="@+id/guardarAcero"
    android:layout_marginLeft="5dp"
    android:layout_marginRight="5dp"
    android:layout_marginTop="10dp" />
</LinearLayout>

<View
    android:layout_width="1dp"
    android:layout_height="fill_parent"
    android:background="@android:color/black"
    android:layout_margin="5dp" />

<LinearLayout
    android:orientation="vertical"
    android:layout_width="280dp"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Suelo"
        android:layout_gravity="center_horizontal" />

    <ImageView
        android:layout_width="wrap_content"

```

```

        android:layout_height="200dp"
        android:id="@+id/imageView5"
        android:layout_gravity="center_horizontal"
        android:background="@drawable/suelo"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginTop="10dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Nuevos Datos:"
    android:layout_marginLeft="10dp"
    android:layout_gravity="center_horizontal" />

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/nombre"
        android:layout_marginLeft="20dp"
        android:id="@+id/textView7"
        android:textSize="18dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:id="@+id/nomSuel"
        android:layout_weight="1"
        android:textSize="14dp"
        android:gravity="bottom" />

</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/sta"
        android:layout_marginLeft="20dp"
        android:textSize="18dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/Sta"
        android:layout_weight="1"
        android:layout_marginRight="10dp"
        android:textSize="14dp"
        android:gravity="bottom" />

</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/gt"

```

```

        android:layout_marginLeft="20dp"
        android:gravity="center_vertical"
        android:textSize="18dp" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="40dp"
    android:inputType="numberDecimal"
    android:ems="10"
    android:id="@+id/Gt"
    android:layout_weight="1"
    android:layout_marginRight="10dp"
    android:textSize="14dp"
    android:gravity="bottom" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="150dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Cohesivo :"
        android:layout_marginLeft="20dp"
        android:gravity="center_vertical"
        android:textSize="18dp" />

    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/cohe" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/coh"
        android:layout_marginLeft="20dp"
        android:gravity="center_vertical"
        android:textSize="18dp" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/coh"
        android:layout_weight="1"
        android:layout_marginRight="10dp"
        android:textSize="14dp"
        android:gravity="bottom" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/k03"
        android:layout_marginLeft="20dp"
        android:gravity="center_vertical"
        android:textSize="18dp" />

    <EditText

```

```

        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/K03"
        android:layout_weight="1"
        android:layout_marginRight="10dp"
        android:textSize="14dp"
        android:gravity="bottom" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="115dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/ga"
        android:layout_marginLeft="20dp"
        android:textSize="18dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/Ga"
        android:layout_weight="1"
        android:layout_marginRight="10dp"
        android:textSize="14dp"
        android:gravity="bottom" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="115dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/gd"
        android:layout_marginLeft="20dp"
        android:textSize="18dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/Gd"
        android:layout_weight="1"
        android:layout_marginRight="10dp"
        android:textSize="14dp"
        android:gravity="bottom" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="115dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/gv"
        android:layout_marginLeft="20dp"
        android:textSize="18dp"
        android:gravity="center_vertical" />

```

```

        <EditText
            android:layout_width="wrap_content"
            android:layout_height="40dp"
            android:inputType="numberDecimal"
            android:ems="10"
            android:id="@+id/Gv"
            android:layout_weight="1"
            android:layout_marginRight="10dp"
            android:textSize="14dp"
            android:gravity="bottom" />
    </LinearLayout>

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:layout_width="115dp"
            android:layout_height="35dp"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="Delta :"
            android:layout_marginLeft="20dp"
            android:textSize="18dp"
            android:gravity="center_vertical" />

        <EditText
            android:layout_width="wrap_content"
            android:layout_height="40dp"
            android:inputType="numberDecimal"
            android:ems="10"
            android:id="@+id/del"
            android:layout_weight="1"
            android:layout_marginRight="10dp"
            android:textSize="14dp"
            android:gravity="bottom" />
    </LinearLayout>

    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/guardar"
        android:id="@+id/guardarSuelo"
        android:layout_marginLeft="5dp"
        android:layout_marginBottom="5dp" />

</LinearLayout>

<View
    android:layout_width="1dp"
    android:layout_height="fill_parent"
    android:background="@android:color/black"
    android:layout_margin="5dp" />

<LinearLayout
    android:orientation="vertical"
    android:layout_width="280dp"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Geometria"
        android:layout_gravity="center_horizontal"
        android:singleLine="true" />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="200dp"
        android:id="@+id/imageView6"
        android:background="@drawable/geompng" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"

```

```

        android:text="Nuevos Datos:"
        android:layout_marginLeft="10dp"
        android:layout_gravity="center_horizontal" />

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/nombre"
        android:layout_marginLeft="20dp"
        android:singleLine="true"
        android:textSize="18dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="fill_parent"
        android:layout_height="40dp"
        android:id="@+id/nomGeom"
        android:gravity="bottom"
        android:textSize="16dp" />

</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/b2"
        android:layout_marginLeft="20dp"
        android:textSize="18dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/b2"
        android:layout_weight="1"
        android:layout_marginRight="10dp"
        android:linksClickable="false"
        android:gravity="center_vertical|bottom"
        android:textSize="16dp" />

</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/ht"
        android:layout_marginLeft="20dp"
        android:textIsSelectable="true"
        android:textSize="18dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/ht"
        android:layout_weight="1"

```

```

        android:layout_marginRight="10dp"
        android:gravity="center_vertical|bottom"
        android:textSize="16dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/l1"
        android:layout_marginLeft="20dp"
        android:textSize="18dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/l1"
        android:layout_weight="1"
        android:layout_marginRight="10dp"
        android:gravity="center_vertical|bottom"
        android:textSize="16dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/aa1"
        android:layout_marginLeft="20dp"
        android:singleLine="true"
        android:textSize="18dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/aa1"
        android:layout_weight="1"
        android:layout_marginRight="10dp"
        android:gravity="center_vertical|bottom"
        android:textSize="16dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/mm1"
        android:layout_marginLeft="20dp"
        android:textSize="18dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:inputType="numberDecimal"
        android:ems="10"

```



```

        android:id="@+id/dia0"
        android:layout_weight="1"
        android:layout_marginRight="10dp"
        android:gravity="center_vertical|bottom"
        android:textSize="16dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/p"
        android:layout_marginLeft="20dp"
        android:singleLine="true"
        android:textSize="18dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/p"
        android:layout_weight="1"
        android:layout_marginRight="10dp"
        android:gravity="center_vertical|bottom"
        android:textSize="16dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="r1 [m] : "
        android:layout_marginLeft="20dp"
        android:singleLine="true"
        android:textSize="18dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/r1"
        android:layout_weight="1"
        android:layout_marginRight="10dp"
        android:gravity="center_vertical|bottom"
        android:textSize="16dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="r2 [m] : "
        android:layout_marginLeft="20dp"
        android:singleLine="true"
        android:textSize="18dp"
        android:gravity="center_vertical" />

    <EditText

```

```

        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:inputType="numberDecimal"
        android:ems="10"
        android:id="@+id/r"
        android:layout_weight="1"
        android:layout_marginRight="10dp"
        android:gravity="center_vertical|bottom"
        android:textSize="16dp" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <TextView
        android:layout_width="150dp"
        android:layout_height="35dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/en"
        android:layout_marginLeft="20dp"
        android:inputType="numberSigned"
        android:singleLine="true"
        android:textSize="18dp"
        android:gravity="center_vertical" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="40dp"

        android:ems="10"
        android:id="@+id/en"
        android:layout_weight="1"
        android:layout_marginRight="10dp"
        android:gravity="center_vertical|bottom"
        android:textSize="16dp" />
</LinearLayout>

<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/guardar"
    android:id="@+id/guardarGeom"
    android:layout_marginLeft="5dp"
    android:layout_marginRight="5dp" />

</LinearLayout>

</LinearLayout>
</HorizontalScrollView>

</LinearLayout>
</LinearLayout>

```

-dinamico_fragment_principal.XML:

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/zapata">

</LinearLayout>

```

-dinamico_fragment_a.XML:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    >

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:weightSum="1">

        <LinearLayout
            android:orientation="vertical"
            android:layout_width="280dp"
            android:layout_height="fill_parent"
            android:weightSum="1">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:text="@string/datos_guardados"
                android:layout_marginTop="20dp"
                android:layout_marginLeft="20dp"
                android:textSize="30dp" />

            <View
                android:layout_width="fill_parent"
                android:layout_height="2dp"
                android:background="@android:color/black"
                android:layout_marginLeft="10dp"
                android:layout_marginBottom="5dp"
                android:layout_marginRight="10dp"
                android:layout_marginTop="5dp" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:text="@string/armadura_seleccionada"
                android:layout_marginTop="10dp"
                android:layout_marginLeft="40dp" />

            <TextView
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:id="@+id/armaduraCargada"
                android:layout_marginLeft="20dp" />

            <Button
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:text="@string/nueva_armadura"
                android:id="@+id/nuevaArmadura"
                android:layout_gravity="center_horizontal"
                android:layout_marginTop="53dp"
                android:layout_marginLeft="20dp" />

            <ListView
                android:layout_width="match_parent"
                android:layout_height="367dp"
                android:id="@+id/datosArmaduraLW"
                android:layout_gravity="center_horizontal"
                android:layout_marginTop="20dp"
                android:layout_marginLeft="20dp" />

        </LinearLayout>

    <View
        android:layout_height="fill_parent"
        android:layout_width="1dp"
    >
```

```

        android:background="@android:color/black"
        android:layout_marginTop="5dp"
        android:layout_marginBottom="5dp"
        android:layout_marginLeft="5dp"
        android:layout_marginRight="5dp" />
<LinearLayout
    android:orientation="vertical"
    android:layout_width="280dp"
    android:layout_height="fill_parent"
    android:weightSum="1">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/materiales"
        android:layout_marginTop="20dp"
        android:textSize="30dp" />

    <View
        android:layout_width="fill_parent"
        android:layout_height="2dp"
        android:background="@android:color/black"
        android:layout_marginLeft="10dp"
        android:layout_marginBottom="5dp"
        android:layout_marginRight="10dp"
        android:layout_marginTop="5dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/materiales_seleccionados"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="20dp" />

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/materialCargado"
        android:layout_gravity="center_horizontal" />

    <Button
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/nuevo_material"
        android:id="@+id/nuevoMaterial"
        android:layout_marginTop="53dp" />

    <ListView
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:id="@+id/datosMateraillesLW"
        android:layout_marginTop="20dp" />

</LinearLayout>
<View
    android:layout_height="fill_parent"
    android:layout_width="1dp"
    android:background="@android:color/black"
    android:layout_marginTop="5dp"
    android:layout_marginBottom="5dp"
    android:layout_marginLeft="5dp"
    android:layout_marginRight="5dp" />
<LinearLayout
    android:orientation="vertical"
    android:layout_width="280dp"
    android:layout_height="match_parent"
    android:weightSum="1">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Esfuerzos : "
        android:layout_marginTop="20dp"
        android:textSize="30dp" />

```

```

<View
    android:layout_width="fill_parent"
    android:layout_height="2dp"
    android:background="@android:color/black"
    android:layout_marginLeft="10dp"
    android:layout_marginBottom="5dp"
    android:layout_marginRight="10dp"
    android:layout_marginTop="5dp" />

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Axil"
        android:layout_marginLeft="10dp" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:id="@+id/nud"
        android:layout_weight="1" />
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Flector"
        android:layout_marginLeft="10dp" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:id="@+id/mud"
        android:layout_weight="1" />
</LinearLayout>

<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:id="@+id/esfuerzoComp"
    android:layout_gravity="center"
    android:gravity="center_vertical|center_horizontal" />

<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Nuevo Esfuerzo"
    android:id="@+id/guardarEsfuerzo"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="2dp" />

<ListView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/datosEsfuerzosLW"
    android:layout_gravity="center_horizontal" />
</LinearLayout>

<Button
    android:layout_width="wrap_content"
    android:layout_height="fill_parent"

```

```

        android:text="@string/calcular"
        android:id="@+id/irCalculos"
        android:layout_gravity="center_vertical"
        android:textSize="35dp" />
    </LinearLayout>
</LinearLayout>

```

-dinamico_fragment_b.XML:

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:weightSum="1">

        <LinearLayout
            android:orientation="vertical"
            android:layout_width="450dp"
            android:layout_height="match_parent"
            android:weightSum="1"
            android:layout_marginLeft="20dp">

            <ListView
                android:layout_width="match_parent"
                android:layout_height="199dp"
                android:id="@+id/armadurasLW"
                android:layout_marginTop="20dp" />

            <TableLayout
                android:layout_width="match_parent"
                android:layout_height="fill_parent"
                android:layout_weight="1">

                <TableRow
                    android:layout_width="40dp"
                    android:layout_height="40dp">

                    <TextView
                        android:textAppearance="?android:attr/textAppearanceLarge"
                        android:text="@string/l_mites"
                        android:layout_column="0"
                        android:layout_width="75dp"
                        android:layout_height="50dp" />
                </TableRow>

                <View
                    android:layout_width="fill_parent"
                    android:layout_height="1dp"
                    android:background="@android:color/black"
                    android:layout_marginLeft="5dp"
                    android:layout_marginRight="5dp" />

                <TableRow
                    android:layout_width="match_parent"
                    android:layout_height="match_parent">

                    <TextView
                        android:textAppearance="?android:attr/textAppearanceLarge"
                        android:text="@string/mud"
                        android:layout_column="0"
                        android:layout_height="fill_parent"
                        android:layout_marginLeft="30dp" />

                    <TextView
                        android:layout_width="350dp"
                        android:layout_height="70dp"

```

```

        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/muLim"
        android:layout_column="1"
        android:textSize="18dp"
        android:gravity="center_vertical" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/nud"
        android:layout_column="0"
        android:layout_height="fill_parent"
        android:layout_marginLeft="30dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="70dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/nuLim1"
        android:layout_column="1"
        android:textSize="18dp"
        android:gravity="center_vertical" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="70dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/nuLim2"
        android:layout_column="1"
        android:textSize="18dp"
        android:gravity="center_vertical" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/epsilon"
        android:layout_column="0"
        android:layout_height="fill_parent"
        android:layout_marginLeft="30dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="70dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/plnLim1"
        android:layout_column="1"
        android:textSize="18dp"
        android:gravity="center_vertical" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/plnLim2"
        android:layout_column="1"
        android:layout_width="50dp" />
</TableRow>
</TableLayout>
</LinearLayout>
<View
    android:layout_width="2dp"

```

```

        android:layout_height="fill_parent"
        android:background="@android:color/black"
        android:layout_marginTop="5dp"
        android:layout_marginBottom="5dp"
        android:layout_marginRight="10dp" />
<TableLayout
    android:layout_width="309dp"
    android:layout_height="match_parent"
    android:layout_weight="0.71">

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:layout_width="250dp"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:id="@+id/arma"
            android:layout_column="1" />
    </TableRow>
    <View
        android:layout_width="fill_parent"
        android:layout_height="2dp"
        android:background="@android:color/black"/>
    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="@string/mud"
            android:layout_column="0"
            android:layout_height="40dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:id="@+id/mud"
            android:layout_column="1" />
    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="@string/nud"
            android:layout_column="0"
            android:layout_height="40dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:id="@+id/nud"
            android:layout_column="1" />
    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="@string/mud1"
            android:layout_column="0"
            android:layout_height="40dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:layout_column="1" />
    </TableRow>

```



```

        android:id="@+id/mud1"
        android:layout_column="1" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/muoptimo"
        android:layout_column="0"
        android:layout_height="40dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/muOptimo"
        android:layout_column="1" />

</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/dominio"
        android:layout_column="0" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/dominio"
        android:layout_column="1" />

</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/epsilon"
        android:layout_column="0" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/pln"
        android:layout_column="1" />

</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/plnoptima"
        android:layout_column="0"
        android:layout_height="40dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/plnOptima"
        android:layout_column="1" />

```

```

</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/psi"
        android:layout_column="0" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/psi"
        android:layout_column="1" />

</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/lamabda"
        android:layout_column="0" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/lambda"
        android:layout_column="1" />

</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/w1"
        android:layout_column="0"
        android:layout_height="40dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/w1"
        android:layout_column="1" />

</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/w2"
        android:layout_column="0"
        android:layout_height="40dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:id="@+id/w2"
        android:layout_column="1" />

</TableRow>

<TableRow

```

```

        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="@string/a1"
            android:layout_column="0"
            android:layout_height="40dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:id="@+id/a1"
            android:layout_column="1" />

    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="@string/a2"
            android:layout_column="0"
            android:layout_height="40dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:id="@+id/a2"
            android:layout_column="1" />

    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="@string/n1"
            android:layout_column="0"
            android:layout_height="40dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:id="@+id/n1"
            android:layout_column="1" />

    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="@string/n2"
            android:layout_column="0"
            android:layout_height="40dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:id="@+id/n2"
            android:layout_column="1" />

    </TableRow>

    <TableRow
        android:layout_width="match_parent"

```

```

        android:layout_height="match_parent">

        <TextView
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="@string/ds1"
            android:layout_column="0"
            android:layout_height="40dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:id="@+id/ds1"
            android:layout_column="1" />

    </TableRow>

    <TableRow
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="@string/ds2"
            android:layout_column="0"
            android:layout_height="40dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:id="@+id/ds2"
            android:layout_column="1" />

    </TableRow>
</TableLayout>

<Button
    android:layout_width="wrap_content"
    android:layout_height="fill_parent"
    android:text="@string/grafico"
    android:layout_gravity="center_vertical"
    android:id="@+id/irGrafico" />
</LinearLayout>
</LinearLayout>

```

-dinamico_fragment_nuevaarmadura.XML:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:weightSum="1">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="201dp"
        android:layout_height="match_parent">

        <TextView
            android:layout_width="140dp"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="@string/datos_nueva_armadura"
            android:layout_marginLeft="20dp"
            android:layout_marginTop="50dp" />

        <Button
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="@string/guardar"
            android:id="@+id/nuevaArmadura" />

        <Button
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Volver"
            android:id="@+id/atrasNuevaArmadura"
            android:layout_gravity="center_horizontal" />
    </LinearLayout>

    <View
        android:layout_height="fill_parent"
        android:layout_width="2dp"
        android:background="@android:color/black"
        android:layout_margin="5dp" />

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TableLayout
            android:layout_width="285dp"
            android:layout_height="match_parent">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textAppearance="?android:attr/textAppearanceLarge"
                android:text="@string/nuevos_datos"
                android:layout_marginTop="20dp"
                android:id="@+id/cosa"
                android:textSize="40dp" />

            <View
                android:layout_width="fill_parent"
                android:layout_height="2dp"
                android:background="@android:color/black"
                android:layout_margin="10dp" />

            <TableLayout
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="20dp"
                android:layout_weight="0.38" >

                <TableRow
                    android:layout_width="match_parent"
                    android:layout_height="match_parent" >

                    <TextView
                        android:textAppearance="?android:attr/textAppearanceLarge"
```

```

        android:text="@string/nombre"
        android:layout_column="0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/nombreArmadura"
    android:singleLine="true"
    android:layout_column="1" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/b"
        android:layout_column="0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <EditText
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:id="@+id/b"
        android:singleLine="true"
        android:layout_column="1" />

</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/h"
        android:layout_column="0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:id="@+id/h"
        android:singleLine="true"
        android:layout_column="1" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/fil"
        android:layout_column="0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/dial"
        android:singleLine="true"
        android:layout_column="1" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

```

```

<TextView
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="@string/fi2"
    android:layout_column="0"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="numberDecimal"
    android:id="@+id/dia2"
    android:singleLine="true"
    android:layout_column="1" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/r1"
        android:layout_column="0" />

    <EditText
        android:id="@+id/r1"
        android:layout_column="1"
        android:singleLine="true"
        android:inputType="numberDecimal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/r2"
        android:layout_column="0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:id="@+id/r2"
        android:layout_column="1"
        android:singleLine="true" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/d1"
        android:layout_column="0" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:id="@+id/d1"
        android:layout_column="1"
        android:singleLine="true" />
</TableRow>

<TableRow

```

```
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <TextView
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="@string/d2"
            android:layout_column="0"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <EditText
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:inputType="numberDecimal"
            android:id="@+id/d2"
            android:layout_column="1"
            android:singleLine="true" />
    </TableRow>
</TableLayout>

</TableLayout>

<ImageView
    android:layout_width="336dp"
    android:layout_height="423dp"
    android:id="@+id/imageView8"
    android:background="@drawable/armadura"
    android:layout_marginTop="50dp" />
</LinearLayout>

</LinearLayout>
```

-fragment_estatico.XML:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/design_fab_shadow_start_color"
    android:gravity="center"
    android:orientation="vertical" >

    <ScrollView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/scrollView5" >

        <LinearLayout
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <ImageView
                android:layout_width="wrap_content"
                android:layout_height="200dp"
                android:id="@+id/imageView7"
                android:layout_gravity="center_horizontal"
                android:background="@drawable/icon2"
                android:layout_marginTop="50dp"
                android:layout_marginBottom="50dp" />

            <LinearLayout
                android:orientation="horizontal"
                android:layout_width="match_parent"
                android:layout_height="157dp">

                <Button
                    android:id="@+id/button$fragment_estatico$cargarFragmentDinamicoA"
                    android:layout_width="150dp"
                    android:layout_height="150dp"
                    android:text="@string/boton_1"
                    android:layout_marginTop="0dp"
                    android:layout_marginLeft="20dp"
                    android:layout_marginRight="20dp"
                    android:layout_marginBottom="0dp"
                    android:background="@color/material_grey_800"
                    android:textColor="@color/abc_primary_text_material_dark" />
            </LinearLayout>

            <LinearLayout
                android:orientation="horizontal"
                android:layout_width="match_parent"
                android:layout_height="150dp"
                android:weightSum="1"
                android:layout_marginTop="10dp">

                <Space
                    android:layout_width="wrap_content"
                    android:layout_height="match_parent"
                    android:layout_gravity="center_vertical"
                    android:layout_weight="0.81" />

                <Button
                    android:id="@+id/button$fragment_estatico$cargarFragmentDinamicoB"
                    android:layout_width="150dp"
                    android:layout_height="150dp"
                    android:text="@string/boton_2"
                    android:layout_marginBottom="0dp"
                    android:layout_marginRight="20dp"
                    android:layout_marginLeft="20dp"
                    android:layout_weight="0.02"
                    android:layout_gravity="right"
                    android:background="@color/material_grey_800"
                    android:textColor="@color/abc_primary_text_material_dark" />
            </LinearLayout>
        </LinearLayout>
    </ScrollView>
</LinearLayout>
```

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="20dp">

    <Button
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:text="@string/boton_3"

        android:id="@+id/boton$fragment_estatico$cargarFragmentDinamicoGrafico"
        android:layout_marginLeft="20dp"
        android:background="@color/material_grey_800"
        android:textColor="@color/abc_primary_text_material_dark" />
    </LinearLayout>
</LinearLayout>
</ScrollView>
</LinearLayout>
```

-fragmen_dinamico_nuevomaterial.XML:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="201dp"
        android:layout_height="match_parent" >

        <TextView
            android:layout_width="140dp"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="@string/datos_nuevos_materiales"
            android:layout_marginLeft="20dp"
            android:layout_marginTop="50dp" />

        <Button
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="@string/guardar"
            android:id="@+id/guardarMateriales" />

        <Button
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Volver"
            android:id="@+id/atrasNuevoMaterial"
            android:layout_gravity="center_horizontal" />
    </LinearLayout>

    <View
        android:layout_height="fill_parent"
        android:layout_width="2dp"
        android:background="@android:color/black"
        android:layout_margin="5dp" />

    <TableLayout
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="0.39" >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:text="@string/nuevos_datos"
            android:layout_marginTop="20dp"
            android:id="@+id/cosa1"
            android:textSize="40dp" />

        <View
            android:layout_height="1dp"
            android:layout_width="fill_parent"
            android:background="@android:color/black"
            android:layout_margin="10dp" />

        <TableLayout
            android:layout_width="499dp"
            android:layout_height="wrap_content"
            android:layout_marginLeft="20dp"
            android:layout_weight="0.38" >

            <TableRow
                android:layout_width="match_parent"
                android:layout_height="match_parent" >

                <TextView
                    android:textAppearance="?android:attr/textAppearanceLarge"
                    android:text="@string/nombre"
                    android:layout_column="0"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content" />

                <EditText
                    android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:id="@+id/nombreMateriales"
        android:singleLine="true"
        android:layout_column="1" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="fcd :"
        android:layout_column="0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <EditText
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:id="@+id/fcd"
        android:singleLine="true"
        android:layout_column="1" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="fyd :"
        android:layout_column="0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:id="@+id/fyd"
        android:singleLine="true"
        android:layout_column="1" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Es :"
        android:layout_column="0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="numberDecimal"
        android:id="@+id/es"
        android:singleLine="true"
        android:layout_column="1" />
</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

</TableRow>

```

```
<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

</TableRow>

<TableRow
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

</TableRow>
</TableLayout>
</TableLayout>
</LinearLayout>
```