



**Universidad de Valladolid**

**E.T.S Ingeniería Informática**

Trabajo Fin de Grado

Grado en Ingeniería Informática [T.I.]

# **Desarrollo de una aplicación Android de realidad aumentada como ayuda a la enseñanza de física**

Autor:  
D. José Martín Gago





**Universidad de Valladolid**

**E.T.S Ingeniería Informática**

Trabajo Fin de Grado

Grado en Ingeniería Informática [T.I]

# **Desarrollo de una aplicación Android de realidad aumentada como ayuda a la enseñanza de física**

Autor:

D. José Martín Gago

Tutores:

Manuel Ángel González Delgado

César Llamas Bello



# Índice

|       |   |    |
|-------|---|----|
| 1.    | Introducción .....                                | 1  |
| 1.1   | Objetivos.....                                    | 2  |
| 1.2   | Alcance .....                                     | 3  |
| 1.3   | Realidad Aumentada.....                           | 4  |
| 1.3.1 | La realidad aumentada en la actualidad .....      | 4  |
| 1.3.2 | Aplicaciones de realidad aumentada.....           | 7  |
| 1.3.3 | Frameworks de realidad aumentada.....             | 12 |
| 1.4   | Vuforia.....                                      | 15 |
| 1.5   | Android.....                                      | 18 |
| 2.    | Planificación.....                                | 20 |
| 2.1   | Propósitos, alcance y objetivos.....              | 21 |
| 2.2   | Supuestos y restricciones.....                    | 22 |
| 2.3   | Plan de control del proyecto .....                | 23 |
| 2.3.1 | Gestión de requisitos.....                        | 23 |
| 2.3.2 | Control de la planificación temporal.....         | 23 |
| 2.3.3 | Control de recursos .....                         | 23 |
| 2.3.4 | Control de costes.....                            | 23 |
| 2.4   | Plan de riesgos .....                             | 24 |
| 2.5   | Planificación temporal estimada del Proyecto..... | 28 |
| 2.6   | Planificación temporal real del Proyecto.....     | 32 |
| 3.    | Análisis.....                                     | 36 |
| 3.1   | Propósito del sistema .....                       | 37 |
| 3.2   | Planteamiento del problema y objetivos.....       | 38 |
| 3.3   | Requisitos .....                                  | 40 |
| 3.3.1 | Requisitos de usuario .....                       | 40 |
| 3.3.2 | Requisitos funcionales .....                      | 40 |
| 3.3.3 | Requisitos no funcionales .....                   | 41 |
| 3.3.4 | Requisitos de la información .....                | 42 |
| 3.4   | Casos de uso .....                                | 43 |

|  |    |
|--|----|
| 3.4.1 Enfocar instrumento del laboratorio.....     | 44 |
| 3.4.2 Pulsar botón virtual .....                   | 45 |
| 3.4.3 Descargar base de datos .....                | 46 |
| 3.4.4 Ver información de la aplicación .....       | 47 |
| 3.4.5 Añadir un botón virtual .....                | 48 |
| 3.4.6 Editar un botón virtual.....                 | 49 |
| 3.4.7 Eliminar botón virtual .....                 | 50 |
| 3.4.8 Establecer tamaño original de la imagen..... | 51 |
| 3.4.9 Modificar tamaño de una imagen.....          | 52 |
| 3.4.10 Crear base de datos.....                    | 53 |
| 3.4.11 Seleccionar elemento.....                   | 55 |
| 3.4.12 Modificar descripción de una imagen .....   | 56 |
| 3.5 Matriz de trazabilidad .....                   | 58 |
| 3.6 Modelo del dominio.....                        | 60 |
| 3.7 Diagramas de secuencia.....                    | 61 |
| 3.7.1 Enfocar instrumento de laboratorio.....      | 62 |
| 3.7.2 Pulsar botón.....                            | 63 |
| 3.7.3 Descargar base de datos .....                | 64 |
| 3.7.4 Crear base de datos.....                     | 65 |
| 3.7.5 Añadir botón.....                            | 66 |
| 3.7.6 Editar botón .....                           | 67 |
| 3.7.7 Borrar botón .....                           | 68 |
| 3.7.8 Seleccionar y establecer tamaño .....        | 69 |
| 3.7.9 Modificar tamaño original.....               | 70 |
| 3.7.10 Ver información de la aplicación .....      | 71 |
| 3.7.11 Modificar descriptor de la imagen.....      | 72 |
| 4.    Diseño.....                                  | 73 |
| 4.1 Arquitectura del sistema .....                 | 74 |
| 4.1.1 Decisiones tomadas .....                     | 74 |
| 4.1.2 Visión global .....                          | 76 |
| 4.1.3 Diseño de la arquitectura.....               | 78 |
| 4.2 Casos de uso de diseño .....                   | 80 |
| 4.2.1 Enfocar instrumento del laboratorio.....     | 80 |
| 4.2.2 Pulsar botón virtual .....                   | 81 |
| 4.2.3 Descargar base de datos .....                | 82 |

|   |     |
|---|-----|
| 4.2.4 Ver información de la aplicación .....        | 83  |
| 4.2.5 Añadir un botón virtual .....                 | 84  |
| 4.2.6 Editar un botón virtual .....                 | 85  |
| 4.2.7 Eliminar botón virtual .....                  | 87  |
| 4.2.8 Establecer tamaño original de la imagen ..... | 88  |
| 4.2.9 Modificar tamaño de una imagen.....           | 89  |
| 4.2.10 Crear base de datos .....                    | 90  |
| 4.2.11 Seleccionar elemento .....                   | 92  |
| 4.2.9 Modificar descripción de una imagen.....      | 93  |
| 4.3 Diagramas de secuencia.....                     | 95  |
| 4.3.1 Enfocar instrumento del laboratorio .....     | 96  |
| 4.3.2 Pulsar botón .....                            | 97  |
| 4.3.3 Descargar base de datos .....                 | 98  |
| 4.3.4 Crear base de datos .....                     | 99  |
| 4.3.5 Opciones botones .....                        | 100 |
| 4.3.6 Añadir botón .....                            | 101 |
| 4.3.7 Editar botón.....                             | 102 |
| 4.3.8 Borrar botón .....                            | 103 |
| 4.3.9 Seleccionar y establecer tamaño .....         | 104 |
| 4.3.10 Modificar tamaño original .....              | 105 |
| 4.3.11 Ver información de la aplicación.....        | 106 |
| 4.3.12 Modificar descripción .....                  | 107 |
| 4.4 Diagrama de clases de diseño .....              | 108 |
| 4.4.1 Diagrama de clases de dispositivo .....       | 109 |
| 4.4.2 Diagrama de clases de servidor.....           | 110 |
| 5. Implementación.....                              | 111 |
| 5.1 Aplicación Android .....                        | 112 |
| 5.2 Programa Python .....                           | 114 |
| 5.3 Herramientas utilizadas .....                   | 116 |
| 6. Pruebas .....                                    | 117 |
| 6.1 Pruebas de caja negra .....                     | 118 |
| 6.1.1 Programa Python.....                          | 118 |
| 6.1.2 Programa Android.....                         | 123 |
| 6.2 Pruebas de caja blanca.....                     | 124 |

|  |     |
|--|-----|
| 6.2.1 Programa Python .....                | 125 |
| 6.2.2 Aplicación Android .....             | 131 |
| 7. Conclusiones y trabajos futuros .....   | 134 |
| 7.1 Conclusiones .....                     | 134 |
| 7.2 Trabajos futuros .....                 | 135 |
| 8. Bibliografía.....                       | 136 |
| 9. Anexos.....                             | 137 |
| 9.1 Contenido del CD .....                 | 137 |
| 9.2 Glosario de términos .....             | 138 |
| 9.3 Integración de Vuforia en Eclipse..... | 140 |
| 9.4 Manual de usuario administrador.....   | 141 |
| 9.4.1 Target Manager .....                 | 142 |
| 9.4.2 Gestor de Botones .....              | 145 |
| 9.5 Manual usuario aplicación móvil.....   | 152 |



# 1. Introducción

En este documento se presentarán las etapas de desarrollo de la aplicación móvil de realidad aumentada cuya finalidad es la ayuda en los laboratorios de los alumnos de la asignatura de física y además se creará un programa que permita al responsable de añadir más elementos de laboratorio a la aplicación realizar esta tarea de manera más sencilla.

## 1.1 Objetivos

El objetivo es la realización de una aplicación móvil que permita una mayor familiarización a los alumnos con los elementos del laboratorio y de esta manera la curva de aprendizaje no tenga una pendiente tan pronunciada.

Debido a que los experimentos llevados a cabo en el laboratorio de física no se han visto antes por los alumnos en sus centros de educación se encuentran ante un gran desafío, por lo cual entendiendo mejor como utilizar los elementos del laboratorio tendrán menos problemas para llevar a cabo las prácticas en el tiempo establecido.

Esta parte de la asignatura de física cuenta actualmente un veinte por ciento de la nota, lo que puede traducirse como un punto de inflexión para el estado anímico del alumno antes de realizar el examen final.

Además será necesario que el profesor pueda añadir nuevos elementos de laboratorio a la aplicación de la manera más sencilla posible, ya que cuenta con un número muy elevado de alumnos y debe llevar a cabo un gran número de evaluaciones, por lo que se tratará de restarle el menor tiempo posible.

## 1.2 Alcance

La aplicación será utilizada por los alumnos de la asignatura de física de la Universidad de Valladolid, se procurará realizar un diseño que permita una gran escalabilidad.

Al igual que en el resto de aplicaciones móviles el profesor pondrá a disposición de los alumnos la APK en la página de la asignatura del campus virtual de la Universidad de Valladolid para su descarga e instalación.

### 1.3 Realidad Aumentada

Podemos describir la realidad aumentada a grandes rasgos como la superposición de contenido en 2D y 3D en imágenes reales, utilizando un dispositivo tecnológico.

Según Ronald Azuma (1997) la realidad aumentada:

1. Combina elementos reales y virtuales.
2. Es interactiva en tiempo real.
3. Está registrada en 3D.

Los elementos que forman la realidad aumentada son:

1. Un dispositivo electrónico que capture imágenes (webcam o dispositivo móvil).
2. Un software que transforma los datos reales en realidad aumentada.
3. Marcadores, elementos donde se reproducirán los modelos 2D o 3D.
4. Pantalla, donde se mostrarán los elementos de realidad aumentada junto con los reales.

Podemos realizar una separación en niveles según el grado de realidad aumentada, según Prendes Espinosa, los denominados *niveles de la realidad aumentada* se pueden definir como los diferentes grados de complejidad que tienen las aplicaciones según las tecnologías que implementan. A mayor nivel de realidad aumentada, más ricas y avanzadas serán las funcionalidades de la aplicación.

Lens Fitzgerald propone una clasificación en cuatro niveles (de 0 a 3):

- Nivel 0, las aplicaciones enlazan con el mundo físico utilizando códigos de barras o códigos QR.
- Nivel 1, las aplicaciones utilizan marcadores (imágenes en blanco y negro con un dibujo en su interior) para el reconocimiento de objetos 2D y 3D.
- Nivel 2, las aplicaciones sustituyen los marcadores por el uso de GPS, la brújula y demás sensores del dispositivo, permitiendo de esta manera determinar la localización del usuario y mostrando *puntos de interés* (elementos de realidad aumentada) sobre una imagen real.
- Nivel 3, aquí encontraríamos dispositivos como Google Glass o lentes de contacto de alta tecnología entre otros. Estos son capaces de lograr la inmersión del usuario.

#### 1.3.1 La realidad aumentada en la actualidad

Actualmente la realidad aumentada está comenzando a florecer y podemos encontrarla en lugares de lo más diversos:

## Trabajo Fin de Grado

---

- Videojuegos, los cuales utilizan el entorno para generar nuestro mundo de juego. Además encontramos en el desarrollo a grandes compañías como Valve que está trabajando en un proyecto de *computación vestible* que ha descrito como la *visión de Terminator* por el desarrollador Michael Abrash.



**Figura 1:** Videojuego Invizimals

- Militares, el Departamento de Defensa de Estados Unidos ya cuenta con lentes de contacto de realidad aumentada, producidos por la empresa Innovega. Estos no muestran texto frente a los ojos, sino que tienen un filtro que permiten a la persona que los lleva enfocar mejor a larga y corta distancia, proveen un campo de visión más amplio y permiten mostrar una imagen adicional a un costado. Lo cual también podría llegar a utilizarse en medicina.
- Ventanas aumentadas, Toyota ha presentado un tipo de ventanas con realidad aumentada cuya funcionalidad es entretener a los pasajeros de los asientos traseros y evitar de esta manera que distraigan al conductor. Esto podría llegar a utilizarse en tiendas con escaparates de realidad aumentada.



**Figura 2:** Ventanas de realidad aumentada de Toyota

- Parabrisas, las empresas del automóvil están comenzando a innovar en sus modelos haciendo pruebas con la realidad aumentada, también encontramos coches con parabrisas de

## Trabajo Fin de Grado

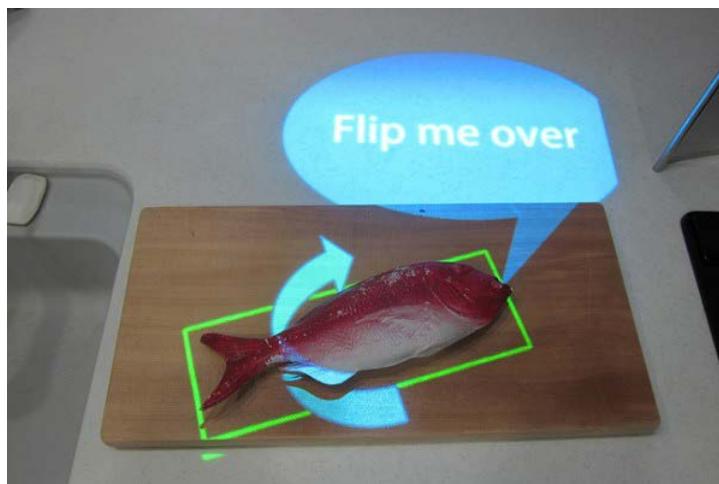
---

realidad aumentada, aunque están evaluado si estos llegan a distraer al conductor o si es beneficioso. En el CES 2012 Mercedes Benz mostró su prototipo que permite interactuar con gestos con el panel.



**Figura 3:** Parabrisas con realidad aumentada

- Cocina, la realidad aumentada ha llegado incluso a estas labores del día a día de toda persona, encontramos la propuesta de Intel que presentó una idea de realidad aumentada que reconoce ingredientes, sugiere recetas o para hacer la lista de la compra cuando el usuario note que le falta algo.



**Figura 4:** Aplicación de Intel para la cocina

- Gafas, las Google Glass ya fueron presentadas hace tiempo, y aunque de momento solo siguen en venta para desarrolladores y aún no hay versión comercial sí que han llegado noticias de acuerdos con marcas para incluir esta tecnología en gafas de diseño como las de la marca Ray Ban.

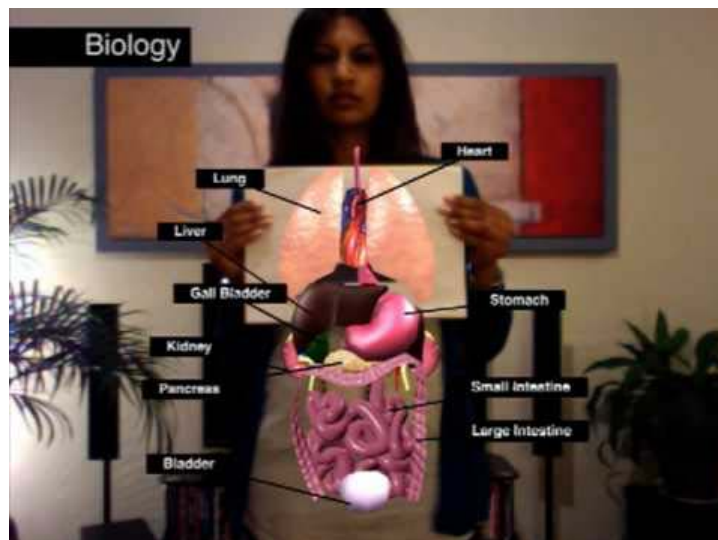


**Figura 5:** Gafas Ray Ban con Google Glass

## 1.3.2 Aplicaciones de realidad aumentada

También podemos encontrar aplicaciones de realidad aumentada como:

- LearnAR: Es una herramienta de aprendizaje interactiva, consiste en diez programas de estudios para maestros y estudiantes. Contiene actividades de matemáticas, ciencias, anatomía, física, geometría, educación física e idiomas.



**Figura 6:** LearnAR en el temario de biología

- Google Sky Map: Aplicación destinada a quien le interesa observar el espacio en el estudio de astronomía. El programa identifica estrellas, constelaciones, planetas y cuerpos celestes enfocando la cámara del móvil al cielo.



**Figura 7:** Google Sky Map escaneando el cielo

- WordLens: Aplicación que traduce texto directamente de una imagen. Solo es necesario tomar una fotografía a un texto desconocido y se obtiene una traducción.



**Figura 8:** WordLens traduciendo un texto

- Wikitude World Browser: Aplicación que sirve para conseguir información utilizando su posición geográfica.



**Figura 9:** Wikitude World Browser utilizando el GPS



## Trabajo Fin de Grado

- Goggles: Aplicación que automatiza la búsqueda en Internet de objetos reales, usando su código de barras (si se trata de productos) o su ubicación (puntos geográficos).

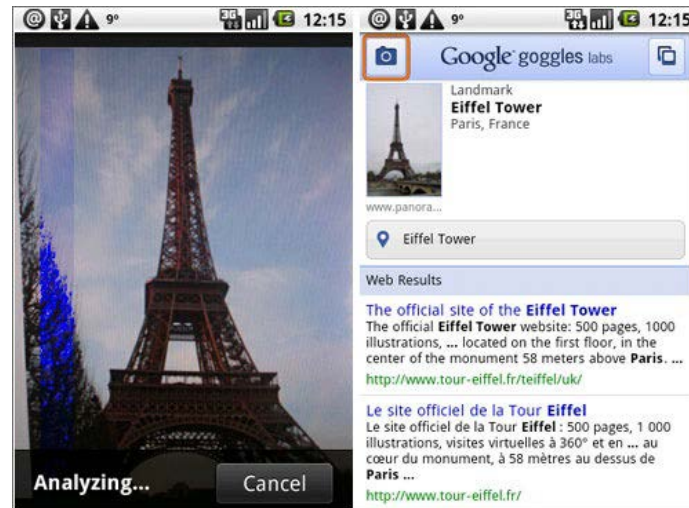


Figura 10: Goggles reconociendo la torre Eiffel

- TAT Augmented ID: Aplicación que reconoce la cara de una persona y muestra los servicios en los que está presente (e-mail, Twitter, Facebook...).

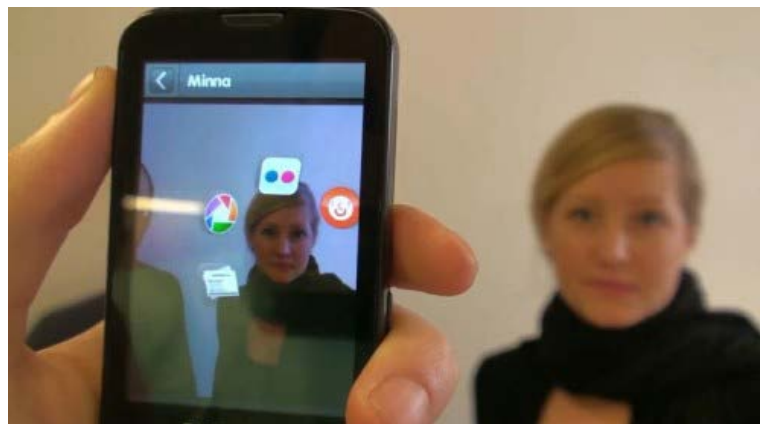


Figura 11: TAT Augmented ID encontrando los perfiles de una persona

- Nokia - Point & Find: Con esta aplicación si tomamos una imagen del poster de una película se obtiene información sobre ella, o con el código de barras de una prenda de vestir devuelve los precios en otros negocios asociados. También se pueden etiquetar objetos y lugares de una ciudad y compartirlos con otros usuarios de la aplicación.



Figura 12: Nokia - Point & Find reconociendo una película

- Junaio: Aplicación que permite depurar y reemplazar la publicidad por otros contenidos.



Figura 13: Junaio sustituyendo publicidad por un famoso cuadro

- TwittARound: Aplicación que nos permite observar todos los tweets que se están publicando en tiempo real cerca de donde se encuentra el dispositivo en el cual se hace la consulta.



Figura 14: TwittARound mostrando tweets

- Lookator: Aplicación que permite encontrar la mejor señal Wifi cerca de la ubicación del dispositivo viendo la información por pantalla.



Figura 15: Lookator mostrando las redes Wifi disponibles

- Yelp Monocle: Red social para buscar información sobre los locales comerciales cercanos al usuario.

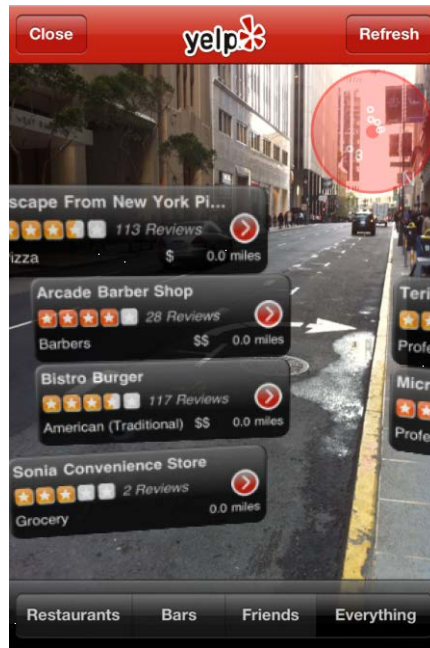


Figura 16: Yelp Monocle mostrando información sobre los locales cercanos

### 1.3.3 Frameworks de realidad aumentada

Para poder realizar la aplicación debemos encontrar el framework para realidad aumentada que mejor se ajuste para el desarrollo, por lo que es necesario llevar a cabo un pequeño estudio de los diferentes frameworks disponibles:

- Wikitude, desarrollado por la compañía austriaca Wikitude GmbH y fue publicado en octubre de 2008 como software gratuito. Posee reconocimiento de imágenes, modelado 3D y combinar esto con el GPS. En un principio llevaba a cabo el reconocimiento utilizando la localización mediante GPS o Wifi, pero desde 2012 puede reconocer las imágenes gracias a la tecnología de reconocimiento de imágenes con la que cuenta.

Tiene un plugin de Unity3D para poder integrarlo con Wikitude y las aplicaciones se desarrollan con JavaScript.

- Vuforia, desarrollado por la compañía estadounidense Qualcomm, reconoce imágenes planas, objetos 3d simples (cajas, cilindros) en tiempo real. Por lo que permite posicionar los objetos virtuales con respecto al mundo real.

Podemos desarrollar las aplicaciones en C++, Java, Objective-C y .Net por lo que es muy versátil en este aspecto. Posee un plugin para Unity3D que permite desarrollar desde su interfaz y que permite exportar fácilmente tanto a Android como a IOS.

- ARToolKit, desarrollado por Hirokazu Kato en 1999 fue publicado por el HIT Lab (Human Interface Technology Laboratory) de la Universidad de Washington. Se mantiene como proyecto de código abierto con licencias comerciales (desarrollo en Java, C y C++).

## Trabajo Fin de Grado

---

- ARTag, alternativa a ARToolKit, usa un procesamiento de imagen y símbolos más complejo con mayor fiabilidad y resistencia a la luz. Tiene soporte de código abierto de Goblin XNA.
- ARToolKitPlus, versión extendida de ARToolKit, es recomendada solo para usuarios experimentados y desarrolladores de realidad aumentada.
- Studierstube Tracker, sucesor de ARToolKitPlus que detecta marcadores 2D. Diseñado para soportar tanto dispositivos móviles como ordenadores. No lee imágenes de una cámara, no renderiza nada, no sigue elementos naturales y no depende o da soporte a ninguna unidad de hardware.
- NyARToolKit, reconoce marcadores del tipo ARToolKit, la mayor parte de la documentación está en japonés, lo que dificulta mucho la búsqueda de información de cualquier tipo.
- Layar, desarrollado por la compañía alemana del mismo nombre. Se trata de un navegador móvil que permite a los usuarios encontrar elementos basándose en la tecnología de realidad aumentada. Utiliza un sistema de capas que funcionan sobre el navegador, permite el uso de objetos 3D, animaciones para objetos 2D y 3D y uso de localización.
- DroidAr, permite el uso de la localización y marcadores basados en realidad aumentada. El prototipo de esta herramienta de desarrollo se creó en 2010 y aún tiene soporte. Proporciona el SDK de forma gratuita con una licencia de código abierto para proyectos no comerciales.
- Metaio, desarrollado por la compañía alemana Metaio GmbH (también responsable de la aplicación Junaio) que el 28 de mayo de 2015 fue adquirida por Apple. Los productos y suscripciones estarán disponibles hasta el 15 de diciembre de 2015.

En un principio se estableció como meta un reconocimiento utilizando un servidor externo, y debido a las limitaciones del reconocimiento en la nube de Vuforia (solo 1000 reconocimientos al mes), se descartó el uso de Vuforia.

Me centré en Wikitude, pero su limitación de 200 usuarios máximo para la aplicación en su versión gratuita para estudiantes lo hacía inviable por lo que también fue descartado.

Los sucesores de ARToolKit tenían la limitación del uso solo de marcadores por lo que no resultaban ser una gran ayuda en el laboratorio (ya que la experiencia del usuario consistiría tan solo en descargar un PDF al dispositivo gracias al marcador).

## Trabajo Fin de Grado

---

Finalmente tras un cambio de perspectiva con Vuforia consideré la opción de utilizar el reconcomiendo desde el dispositivo móvil utilizando una base de datos local, permitiendo al usuario actualizar la base de datos de elementos a reconocer desde un servidor y un programa en este servidor que permita al profesor añadir botones virtuales a los diferentes elementos. Esta opción de Vuforia permite tener un máximo de 200 targets (cantidad que no va a superarse en el contexto del laboratorio de física) en la base de datos pero permite un número de reconocimientos ilimitado.

## 1.4 Vuforia

Podemos ver en la siguiente imagen como funciona el desarrollo de aplicaciones con el SDK de Vuforia. La plataforma consiste en el motor de reconocimiento de Vuforia (Vuforia Engine), el sistema de gestión de marcadores e imágenes (Target Management System en el Target Manager) y opcionalmente el uso de la base de datos en la nube (Cloud Target Database).

### Platform Anatomy

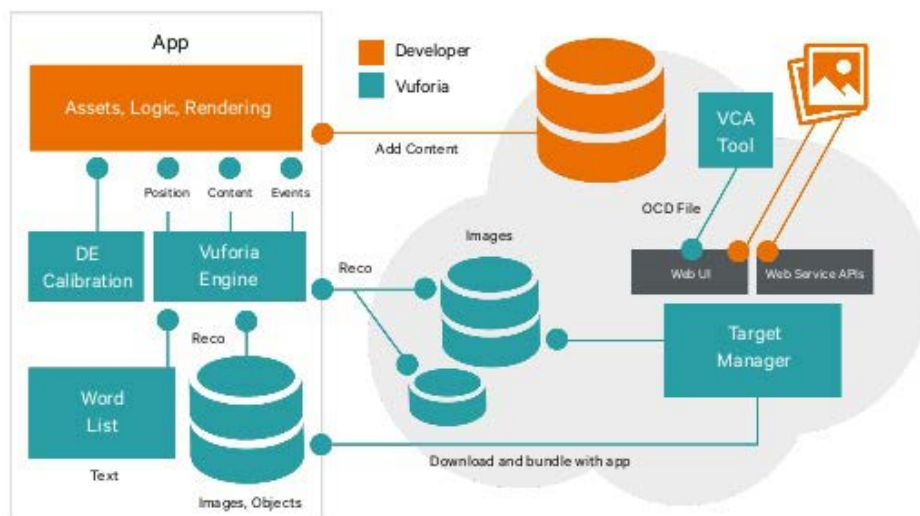


Figura 17: Anatomía de la plataforma

Vuforia también nos permite reconocer imágenes, objetos, entornos, texto y marcadores. Tiene una gran escalabilidad debido a la posibilidad de aumentar la base de datos de targets tanto de reconocimiento en la nube como en local. Permite la reproducción de contenidos, la interacción con las imágenes, tanto reales como virtuales mediante botones virtuales y la creación de efectos. Da soporte a las herramientas de desarrollo como Eclipse, Unity, Xcode y recientemente Android Studio.

En la siguiente imagen podemos ver la forma de interacción de los distintos elementos del SDK. La cámara del dispositivo móvil realiza la captura y se envían estas imágenes al Módulo Conversor que las transforma a una representación soportada por OpenGL ES y se comprime a una menor resolución para su manejo. Tras esto, es transmitida al Módulo Tracker donde se encuentran los algoritmos de reconocimiento de imagen que detectan y rastrean los objetos del mundo real que se han indicado anteriormente. Este módulo devuelve qué marcador o imagen se ha encontrado y se está rastreando, y pasa esta información al módulo que se encarga de añadir el contenido adicional que se debe mostrar sobre el objeto reconocido, renderizando directamente sobre la cámara para crear la realidad aumentada.

## Vuforia SDK

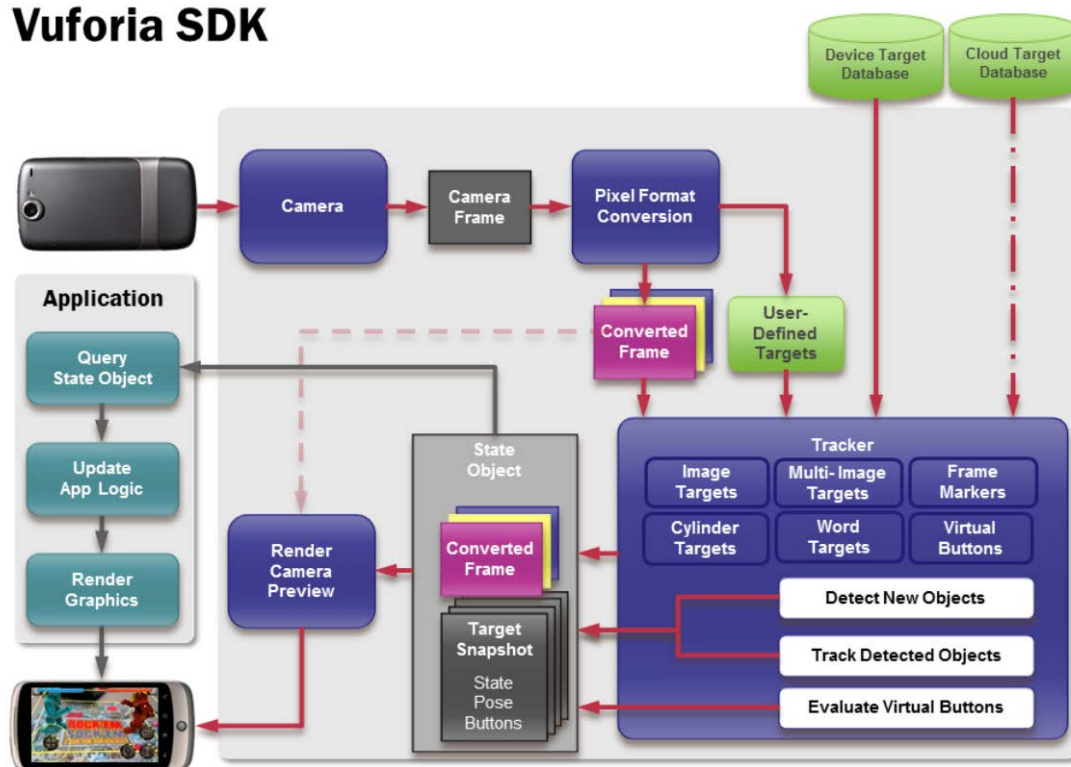


Figura 18: SDK de Vuforia

Podemos distinguir en una aplicación de Vuforia los siguientes componentes:

- Cámara, encargada de capturar las imágenes.
- Base de datos del dispositivo, se crea en el Target Manager y es donde se almacenan la colección de targets que reconocerá el Tracker, es almacenada en el dispositivo de manera local.
- Targets, utilizados por el rastreador (Tracker) para reconocer un objeto del mundo real, podemos distinguir:
  1. Image Targets: fotos, páginas de libros cubiertas de estos, posters...
  2. User Defined Targets: el usuario crea los targets durante el uso de la aplicación, no son incluidos por el desarrollador en la base de datos durante la creación de la aplicación.
  3. Cylinder Targets: imágenes en superficies cilíndricas.
  4. Frame Markers: marcadores.
- Tracker, se encarga de analizar la imagen tomada por la cámara y detectar los objetos del mundo real que coinciden con los almacenados en la base de datos.

Tal y como informa Vuforia en su página para desarrolladores, un botón ideal tiene los siguientes atributos:



## Trabajo Fin de Grado

---

1. Tamaño, el botón tendrá un tamaño de al menos el 10% del tamaño de la imagen.
2. Forma, botones fácilmente identificables con el resto de la imagen.
3. Textura o contraste, evitar colocar botones en áreas con poco contraste.
4. Bordes alrededor del objetivo del botón para no perder el seguimiento del mismo cuando el usuario lo pulse.

## 1.5 Android

La practica será realizada para dispositivos con sistema operativo Android, ya que es el más común.

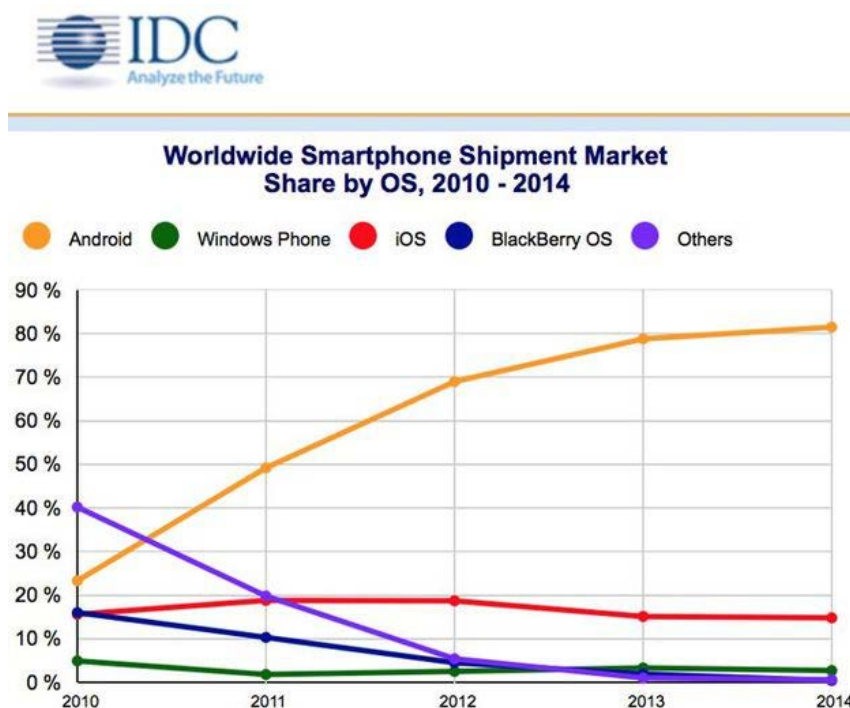


Figura 19: Cuota de mercado del 2010 al 2014

Inicialmente fue desarrollado por Android Inc., empresa respaldada económicamente por Google y después compró en 2005. Fue diseñado en un principio para dispositivos móviles (teléfonos inteligentes o tabletas) con pantalla táctil.

Android fue presentado en 2007 y el primer teléfono móvil con sistema operativo Android fue el HTC Dream que se vendió en 2008.

Android está formado por una arquitectura en capas (4 capas):

- El núcleo Linux, formado por el sistema operativo Linux, proporciona los servicios de seguridad, manejo de memoria, multiproceso, la pila de protocolos y el soporte de drivers para dispositivos.
- Runtime de Android, basado en la máquina virtual de Java, Google tuvo que crear una máquina virtual (la máquina virtual Dalvik) que pudiera responder a las limitaciones de poca memoria, procesador limitado...
- Delega al núcleo Linux algunas funciones como threading y el manejo de memoria a bajo nivel.

# Trabajo Fin de Grado

- A partir de Android 5.0 se reemplaza Dalvik por ART.
- Librerías nativas, incluye un conjunto de librerías en C/C++.
- Entorno de aplicación, capa diseñada para la reutilización de componentes, las aplicaciones publican sus capacidades y otras pueden hacer uso de ellas (según las restricciones de seguridad).
- Aplicaciones, capa formada por el conjunto de aplicaciones instaladas.

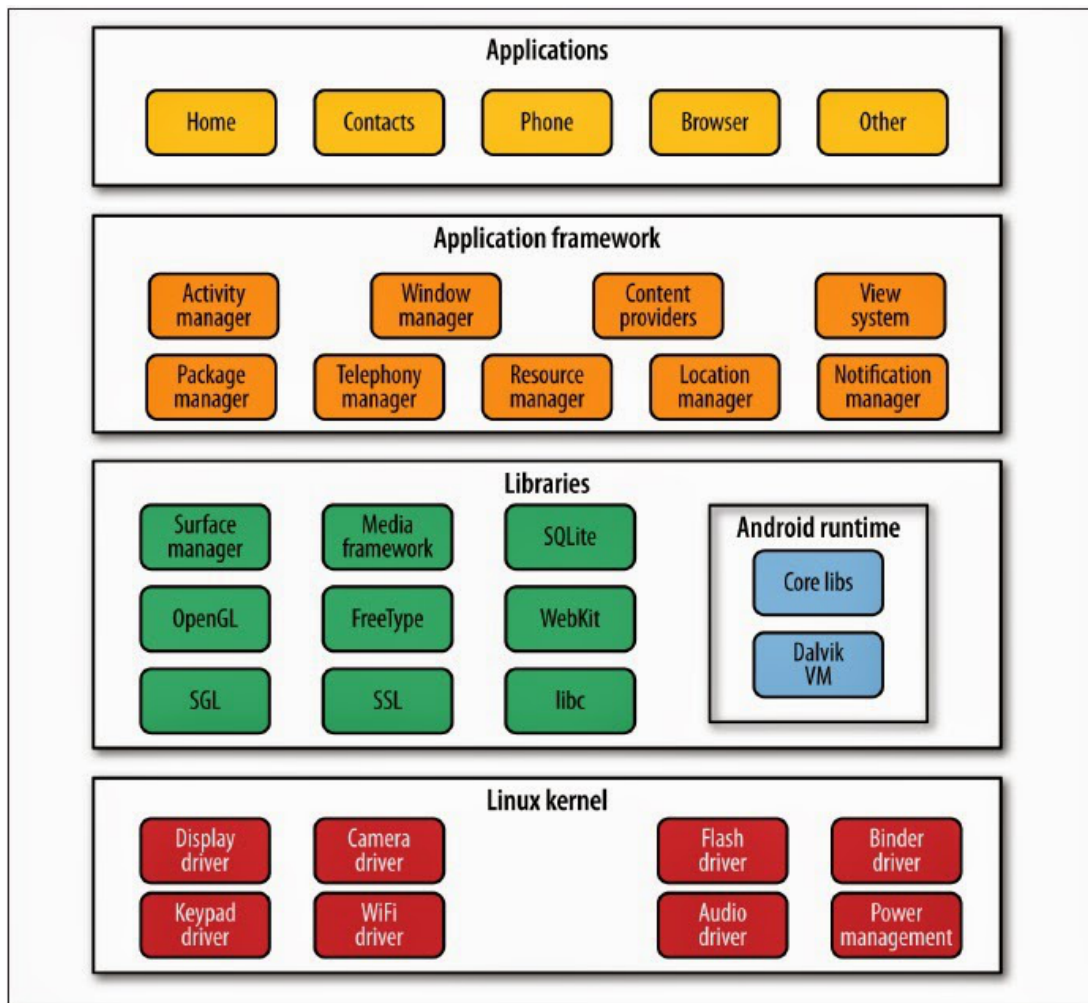


Figura 20: Arquitectura Android

## **2. Planificación**

Se llevará a cabo una estrategia de desarrollo, con unas metas y objetivos para el desarrollo satisfactorio del proyecto. Mediante una buena planificación conseguiremos elevar el nivel de éxito a la hora de acabar el proyecto en las fechas establecidas.

Una mala planificación puede llevar al fracaso cualquier proyecto, por lo que debemos tener cuidado y enfrentarnos a los riesgos que aparezcan de la manera más eficiente para acabar con éxito el proyecto.

## 2.1 Propósitos, alcance y objetivos

El principal objetivo de este proyecto es la realización de una aplicación móvil que permita a los alumnos de física interactuar con los elementos de laboratorio utilizando su dispositivo con Android para obtener información sobre los elementos de laboratorio y de esta manera ayudar a los mismos durante la realización de las prácticas de física.

Junto con ello, se alcanzaran los objetivos descritos a continuación:

- Se creará un programa que facilite la creación de bases de datos para que el profesor o administrador del servidor donde se encontraran estas bases de datos, pueda generar los archivos que necesita la aplicación móvil sin necesidad de editar manualmente los archivos.
- El servidor contendrá un conjunto de bases de datos disponibles para poder cambiar de versión de la aplicación.
- El programa que crea las nuevas bases de datos se asemejará en lo mayor posible al programa que se utiliza ya en el laboratorio de física para generar gráficas.
- Se permitirá el servicio en diferentes idiomas (en este caso se dará servicio solo en inglés y español)
- La aplicación contará con una base de datos por defecto para que no sea necesario descargar ninguna base de datos para funcionar con unos mínimos.
- La aplicación mostrará la base de datos que se está usando si el usuario lo requiere.

### 2.2 Supuestos y restricciones

A continuación se listan las restricciones del sistema, tanto funcionales como técnicas, las cuales durante la realización del proyecto podrán ir aumentando:

- La aplicación necesitará conexión a internet para descargar las bases de datos nuevas.
- La aplicación tan solo guardará de manera local los archivos referidos a la última base de datos descargada.
- El programa gestor de botones necesitará como entrada para generar la nueva base de datos los dos archivos que genera el Target Manager de Vuforia.
- El programa que crea las nuevas bases de datos se ejecutará desde el servidor con permisos de administrador.
- La conexión del administrador con el servidor se llevará a cabo mediante SSH.
- No se comprobarán las cuatro propiedades que especifica Vuforia para un botón ideal, se permitirá su creación y dependerá del administrador la decisión de eliminarlo o mantenerlo, ya que aunque no cumpla a la perfección las propiedades puede funcionar de igual manera.
- La estructura de directorios necesaria para el gestor de botones debe existir para ejecutarlo.

## **2.3 Plan de control del proyecto**

### **2.3.1 Gestión de requisitos**

Una vez se establezcan los requisitos de la aplicación y del programa, lo cual se llevará a cabo en la siguiente fase, cualquier cambio en los requisitos se traducirá en una realimentación de proyecto volviendo atrás en el desarrollo para comprobar cuál es su efecto tanto en planificación, análisis, desarrollo, implementación y pruebas.

### **2.3.2 Control de la planificación temporal**

Mediante la definición de las distintas fases del proyecto y su establecimiento de límites temporales tendremos un control de la planificación.

### **2.3.3 Control de recursos**

Debido a que tan solo contamos con un miembro para los recursos humanos, este estará sobrecargado ya que no será posible delegar tareas.

### **2.3.4 Control de costes**

Debido a que nos encontramos ante un proyecto estudiantil, no tendremos costes con la realización del proyecto (además todas las aplicaciones que se utilizaran no tendrán cargo para el desarrollador), pero tal y como se estipula en la guía docente, la duración de este proyecto será de 300 horas, por lo cual solo se necesitaría establecer un salario por cada hora trabajada para calcular el valor aproximado del coste del proyecto.

## 2.4 Plan de riesgos

Se presentan los riesgos que pueden afectar al proyecto por probabilidad de más baja a más alta y dentro de este por exposición.

Debido a que la presentación del proyecto de final de grado se realiza en julio, los retrasos a la hora de llevarlo a cabo no serán importantes mientras no se supere mayo.

La exposición de los riesgo ha sido calculada utilizando el ISO/IEC 27005:2008 (definido en el glosario).

|  |  |  |                          |
|--|--|--|--------------------------|
| <b>Número:</b> 01  | <b>Nombre del riesgo:</b> Cambio del análisis  |  | <b>Fecha:</b> 01-10-2014 |
| <b>Descripción del riesgo</b>  |  |  |                          |
| Tras haber formulado el análisis de la aplicación, es necesario cambiar los requisitos de la aplicación y debemos revisar todo el trabajo realizado hasta ese momento.                     |  |  |                          |
| <b>Causas</b>  |  |  |                          |
| <ol style="list-style-type: none"> <li>1. Mala interpretación de la solución.</li> <li>2. Cambio de tecnología de desarrollo.</li> <li>3. Modificación de las especificaciones.</li> </ol> |  |  |                          |
| <b>Consecuencias</b>   |  |  |                          |
| Retraso de la aplicación debiendo revisar el trabajo realizado hasta el momento para comprobar si es válido  |  |  |                          |
| <b>Plan de acción</b>  |  |  |                          |
| <b>Estrategia:</b> Protección del riesgo / Aceptación del riesgo   | <b>Descripción:</b> Al tener un modelo de trabajo en el cual estamos en varias fases a la vez la revisión es constante. Si es necesario corregir algo nos encontraremos ante un retraso. |  |                          |
| <b>Impacto:</b> Alto   | <b>Probabilidad:</b> Muy baja  |  | <b>Exposición:</b> 3     |

**Tabla 1:** Riesgos



## Trabajo Fin de Grado

|   |  |                          |
|---|--|--------------------------|
| <b>Número:</b> 02   | <b>Nombre del riesgo:</b> Indisposición del alumno   | <b>Fecha:</b> 01-10-2014 |
| <b>Descripción del riesgo</b>   |  |                          |
| El alumno se encuentra indispuerto y no puede realizar el trabajo que tiene asignado durante ese periodo en la planificación. |  |                          |
| <b>Causas</b>   |  |                          |
| 4. Enfermedad.<br>5. Accidente.   |  |                          |
| <b>Consecuencias</b>  |  |                          |
| Se producirá un retraso en el desarrollo del proyecto.  |  |                          |
| <b>Plan de acción</b>   |  |                          |
| <b>Estrategia:</b> Reserva del riesgo / Aceptación del riesgo   | <b>Descripción:</b> Las únicas solución posible debido a que no hay más personas desarrollando la aplicación serán: <ul style="list-style-type: none"> <li>• Aumento de las horas de trabajo diarias.</li> <li>• Asumir el retraso en el desarrollo de la aplicación.</li> </ul> |                          |
| <b>Impacto:</b> Alto  | <b>Probabilidad:</b> Muy baja  | <b>Exposición:</b> 3     |

Tabla 2: Riesgos

|   |  |                          |
|---|--|--------------------------|
| <b>Número:</b> 03   | <b>Nombre del riesgo:</b> Fallo de los equipos de desarrollo   | <b>Fecha:</b> 01-10-2014 |
| <b>Descripción del riesgo</b>   |  |                          |
| Los instrumentos de desarrollo sufren errores o se estropean por lo que no son utilizables. |  |                          |
| <b>Causas</b>   |  |                          |
| 6. Fallo hardware.<br>7. Fallo software.  |  |                          |
| <b>Consecuencias</b>  |  |                          |
| Será necesario utilizar el equipo portátil disponible o de el simulador que provee Eclipse. |  |                          |
| <b>Plan de acción</b>   |  |                          |
| <b>Estrategia:</b> Protección del riesgo  | <b>Descripción:</b> Solucionar los problemas de los equipos principales mientras se utilizan los de apoyo. |                          |
| <b>Impacto:</b> Medio   | <b>Probabilidad:</b> Baja  | <b>Exposición:</b> 3     |

Tabla 3: Riesgos

## Trabajo Fin de Grado

|  |   |                          |
|--|---|--------------------------|
| <b>Número:</b> 04  | <b>Nombre del riesgo:</b> Estudio de asignaturas.   | <b>Fecha:</b> 01-10-2014 |
| <b>Descripción del riesgo</b>  |   |                          |
| El alumno necesita centrarse en otra asignatura debido a algún control o evento. |   |                          |
| <b>Causas</b>  |   |                          |
| 8. Realización de exámenes.  |   |                          |
| <b>Consecuencias</b>   |   |                          |
| La finalización del proyecto será más tarde de lo esperado.                      |   |                          |
| <b>Plan de acción</b>  |   |                          |
| <b>Estrategia:</b> Aceptación del riesgo   | <b>Descripción:</b> Retomar el proyecto cuando las otras tareas no requieran tanto tiempo del estudiante. |                          |
| <b>Impacto:</b> Medio  | <b>Probabilidad:</b> Medio  | <b>Exposición:</b> 4     |

**Tabla 4:** Riesgos

|  |  |                          |
|--|--|--------------------------|
| <b>Número:</b> 05  | <b>Nombre del riesgo:</b> Estimaciones imprecisas.   | <b>Fecha:</b> 01-10-2014 |
| <b>Descripción del riesgo</b>  |  |                          |
| La mala planificación del proyecto provoca que las estimaciones no se vean cumplidas.  |  |                          |
| <b>Causas</b>  |  |                          |
| <ul style="list-style-type: none"> <li>• Demasiada carga de trabajo.</li> <li>• Sobrecarga de los recursos.</li> <li>• Imposibilidad de redistribuir recursos ante ocurrencia de riesgos.</li> </ul> |  |                          |
| <b>Consecuencias</b>   |  |                          |
| La finalización del proyecto será más tarde de lo esperado.  |  |                          |
| <b>Plan de acción</b>  |  |                          |
| <b>Estrategia:</b> Protección del riesgo   | <b>Descripción:</b> Llevar a cabo un seguimiento de la planificación para poder reaccionar a tiempo. |                          |
| <b>Impacto:</b> Alto   | <b>Probabilidad:</b> Medio   | <b>Exposición:</b> 5     |

**Tabla 5:** Riesgos

## Trabajo Fin de Grado

|   |   |                          |
|---|---|--------------------------|
| <b>Número:</b> 06   | <b>Nombre del riesgo:</b> Poca destreza con las herramientas.   | <b>Fecha:</b> 01-10-2014 |
| <b>Descripción del riesgo</b>                                     |   |                          |
| El desarrollo utilizando Vuforia es más complejo de los esperado. |   |                          |
| <b>Causas</b>   |   |                          |
| Ninguna familiarización con el entorno de desarrollo.             |   |                          |
| <b>Consecuencias</b>  |   |                          |
| Es necesario invertir más tiempo en el proyecto.                  |   |                          |
| <b>Plan de acción</b>   |   |                          |
| <b>Estrategia:</b> Aceptación del riesgo                          | <b>Descripción:</b> Retomar el proyecto cuando las otras tareas no requieran tanto tiempo del estudiante. |                          |
| <b>Impacto:</b> Medio   | <b>Probabilidad:</b> Alta   | <b>Exposición:</b> 5     |

Tabla 6: Riesgos

|  |   |                          |
|--|---|--------------------------|
| <b>Número:</b> 07  | <b>Nombre del riesgo:</b> Practicas de empresa.   | <b>Fecha:</b> 01-10-2014 |
| <b>Descripción del riesgo</b>  |   |                          |
| El estudiante comenzará la asignatura de prácticas de empresa, por lo que el tiempo para realizar el proyecto se verá muy reducido y probablemente interrumpido. |   |                          |
| <b>Causas</b>  |   |                          |
| 1. Este año se cursa la asignatura prácticas de empresa.   |   |                          |
| <b>Consecuencias</b>   |   |                          |
| La finalización del proyecto será más tarde de lo esperado.  |   |                          |
| <b>Plan de acción</b>  |   |                          |
| <b>Estrategia:</b> Aceptación / Protección del riesgo  | <b>Descripción:</b> El proyecto se verá interrumpido si el alumno no puede continuar con el proyecto compaginándolo con las prácticas de empresa y la asistencia a clase. |                          |
| <b>Impacto:</b> Alta   | <b>Probabilidad:</b> Muy alta   | <b>Exposición:</b> 7     |

Tabla 7: Riesgos

## 2.5 Planificación temporal estimada del Proyecto

A continuación se presenta la planificación que se ha llevado a cabo tras el análisis de riesgos, objetivos y alcance del proyecto.

Teniendo en cuenta que se trata de una asignatura de doce créditos y en la guía docente se indica que se deben dedicar trescientas horas en total para completar con éxito la asignatura, la planificación temporal está estimada en función a estos datos.

Para ello se divide el proyecto en 6 fases de desarrollo como puede verse a continuación.

| Nombre de tarea         | Duración       | Comienzo            | Fin                 |
|-------------------------|----------------|---------------------|---------------------|
| ▲ <b>Proyecto</b>       | <b>74 días</b> | <b>lun 29/09/14</b> | <b>jue 08/01/15</b> |
| ▷ <b>Inicio</b>         | <b>17 días</b> | <b>lun 29/09/14</b> | <b>mar 21/10/14</b> |
| ▷ <b>Análisis</b>       | <b>16 días</b> | <b>lun 13/10/14</b> | <b>lun 03/11/14</b> |
| ▷ <b>Diseño</b>         | <b>23 días</b> | <b>lun 27/10/14</b> | <b>mié 26/11/14</b> |
| ▷ <b>Implementación</b> | <b>25 días</b> | <b>lun 24/11/14</b> | <b>vie 26/12/14</b> |
| ▷ <b>Pruebas</b>        | <b>53 días</b> | <b>jue 23/10/14</b> | <b>lun 05/01/15</b> |
| ▷ <b>Documentación</b>  | <b>3 días</b>  | <b>mar 06/01/15</b> | <b>jue 08/01/15</b> |

**Figura 21:** Planificación global

Como puede verse se han estimado 74 días de duración, empleando una media de cuatro horas y media al día obtenemos 333 horas, lo que es superior a lo especificado por la guía docente, por lo cual debería finalizarse el proyecto dentro de los márgenes de tiempo. Y se tendrá un gran margen de tiempo para poder presentarse a la primera convocatoria en julio.

La primera fase consiste a grandes rasgos, en la realización de la introducción y planificación de este documento. Puede verse más detalladamente las tareas en la siguiente imagen.

## Trabajo Fin de Grado

| Nombre de tarea                      | Duración       | Comienzo            | Fin                 | Predecesoras |
|--------------------------------------|----------------|---------------------|---------------------|--------------|
| <b>Proyecto</b>                      | <b>74 días</b> | <b>lun 29/09/14</b> | <b>jue 08/01/15</b> |              |
| <b>Inicio</b>                        | <b>17 días</b> | <b>lun 29/09/14</b> | <b>mar 21/10/14</b> |              |
| Asignación de proyecto               | 1 día          | lun 29/09/14        | lun 29/09/14        |              |
| Redacción de la documentación        | 16 días        | mar 30/09/14        | mar 21/10/14        | 3            |
| Determinar alcance                   | 1 día          | mar 30/09/14        | mar 30/09/14        | 3            |
| Planificación                        | 3 días         | mié 01/10/14        | vie 03/10/14        | 3            |
| Definir recursos                     | 1 día          | jue 02/10/14        | jue 02/10/14        | 5            |
| Estimar la duración                  | 2 días         | lun 06/10/14        | mar 07/10/14        | 7            |
| Plan de control de proyecto          | 3 días         | lun 06/10/14        | mié 08/10/14        | 6            |
| Identificación y análisis de riesgos | 5 días         | mié 01/10/14        | mar 07/10/14        | 5            |

**Figura 22:** Planificación inicio

La segunda fase será análisis, en la cual se especificarán los requisitos, casos de uso, realización del modelo del dominio y de los diagramas de secuencia.

| Nombre de tarea                           | Duración       | Comienzo            | Fin                 | Predecesoras |
|---|----------------|---------------------|---------------------|--------------|
| <b>Análisis</b>                           | <b>16 días</b> | <b>lun 13/10/14</b> | <b>lun 03/11/14</b> |              |
| Redacción de la documentación de análisis | 16 días        | lun 13/10/14        | lun 03/11/14        |              |
| <b>Análisis de requisitos</b>             | <b>5 días</b>  | <b>lun 13/10/14</b> | <b>vie 17/10/14</b> |              |
| Requisitos de usuario                     | 1 día          | lun 13/10/14        | lun 13/10/14        | 10           |
| Requisitos funcionales                    | 4 días         | mar 14/10/14        | vie 17/10/14        | 14           |
| Requisitos no funcionales                 | 1 día          | mar 14/10/14        | mar 14/10/14        | 14           |
| Requisitos de información                 | 1 día          | mié 15/10/14        | mié 15/10/14        | 14           |
| <b>Análisis de casos de uso</b>           | <b>6 días</b>  | <b>lun 20/10/14</b> | <b>lun 27/10/14</b> |              |
| Identificación y realización de casos de  | 1 día          | lun 20/10/14        | lun 20/10/14        | 13           |
| Matriz de trazabilidad                    | 1 día          | mar 21/10/14        | mar 21/10/14        | 19           |
| Modelo del dominio                        | 1 día          | mié 22/10/14        | mié 22/10/14        | 19           |
| Diagramas de secuencia                    | 3 días         | jue 23/10/14        | lun 27/10/14        | 21           |

**Figura 23:** Planificación análisis

## Trabajo Fin de Grado

La tercera fase es el diseño del sistema, en la cual se tomarán decisiones para la implementación, se especificará la arquitectura del proyecto, se detallarán más los casos de uso, se realizará el modelo del dominio de diseño y los diagramas de secuencia correspondientes.

| Nombre de tarea                         | Duración       | Comienzo            | Fin                 | Predecesoras |
|---|----------------|---------------------|---------------------|--------------|
| <b>▸ Diseño</b>                         | <b>23 días</b> | <b>lun 27/10/14</b> | <b>mié 26/11/14</b> |              |
| Redacción de la documentación de diseño | 23 días        | lun 27/10/14        | mié 26/11/14        |              |
| <b>▸ Descripción de la arquitectura</b> | <b>11 días</b> | <b>lun 27/10/14</b> | <b>lun 10/11/14</b> |              |
| Decisiones de diseño                    | 2 días         | lun 27/10/14        | mar 28/10/14        |              |
| Identificación de la tecnología         | 4 días         | mar 28/10/14        | vie 31/10/14        | 18           |
| Identificación de interfaces            | 2 días         | lun 03/11/14        | mar 04/11/14        | 27           |
| Realización de diagramas                | 4 días         | mié 05/11/14        | lun 10/11/14        | 28           |
| <b>▸ Diseño de la arquitectura</b>      | <b>3 días</b>  | <b>lun 10/11/14</b> | <b>mié 12/11/14</b> |              |
| Descomposicion                          | 2 días         | mar 11/11/14        | mié 12/11/14        | 25           |
| Vista lógica del sistema                | 1 día          | lun 10/11/14        | lun 10/11/14        | 27           |
| Casos de uso de diseño                  | 1 día          | jue 13/11/14        | jue 13/11/14        | 30           |
| Diagramas de secuencia de diseño        | 6 días         | vie 14/11/14        | vie 21/11/14        | 33           |
| Modelo del dominio de diseño            | 3 días         | lun 24/11/14        | mié 26/11/14        | 34           |

**Figura 24:** Planificación diseño

En la cuarta fase se llevará a cabo la implementación, se desarrollará y documentará el código y se comenzarán a realizar pruebas durante el desarrollo.

| Nombre de tarea              | Duración       | Comienzo            | Fin                 | Predecesoras |
|------------------------------|----------------|---------------------|---------------------|--------------|
| <b>▸ Implementacion</b>      | <b>25 días</b> | <b>lun 24/11/14</b> | <b>vie 26/12/14</b> |              |
| Desarrollo del código        | 20 días        | lun 24/11/14        | vie 19/12/14        | 34           |
| Pruebas de los programadores | 18 días        | lun 01/12/14        | mié 24/12/14        | 37FC-75%     |
| Integracion de los módulos   | 2 días         | jue 25/12/14        | vie 26/12/14        | 38           |

**Figura 25:** Planificación implementación

## Trabajo Fin de Grado

En la quinta fase se llevarán a cabo las pruebas de caja blanca y de caja negra. Durante el desarrollo del código se comenzará a realizar el plan de pruebas.

| Nombre de tarea                | Duración       | Comienzo            | Fin                 | Predecesoras |
|--------------------------------|----------------|---------------------|---------------------|--------------|
| <b>Pruebas</b>                 | <b>53 días</b> | <b>jue 23/10/14</b> | <b>lun 05/01/15</b> |              |
| <b>Caja blanca</b>             | <b>53 días</b> | <b>jue 23/10/14</b> | <b>lun 05/01/15</b> |              |
| Plan de pruebas de caja blanca | 3 días         | jue 23/10/14        | lun 27/10/14        | 21           |
| Pruebas de caja blanca         | 3 días         | lun 29/12/14        | mié 31/12/14        | 42;39        |
| Identificar anomalías          | 1 día          | jue 01/01/15        | jue 01/01/15        | 43           |
| Modificar código               | 2 días         | vie 02/01/15        | lun 05/01/15        | 44           |
| <b>Caja negra</b>              | <b>53 días</b> | <b>jue 23/10/14</b> | <b>lun 05/01/15</b> |              |
| Plan de pruebas de caja negra  | 3 días         | jue 23/10/14        | lun 27/10/14        | 21           |
| Pruebas de caja negra          | 3 días         | lun 29/12/14        | mié 31/12/14        | 47;39        |
| Identificar anomalías          | 1 día          | jue 01/01/15        | jue 01/01/15        | 48           |
| Modificar código               | 2 días         | vie 02/01/15        | lun 05/01/15        | 49           |

**Figura 26:** Planificación pruebas

En la última fase del proyecto consistirá en una revisión de la documentación y el desarrollo de los manuales.

| Nombre de tarea                 | Duración      | Comienzo            | Fin                 | Predecesoras |
|---------------------------------|---------------|---------------------|---------------------|--------------|
| <b>Documentación</b>            | <b>3 días</b> | <b>mar 06/01/15</b> | <b>jue 08/01/15</b> |              |
| Documentación del desarrollo    | 1 día         | mar 06/01/15        | mar 06/01/15        | 40           |
| Integración de la documentación | 1 día         | mié 07/01/15        | mié 07/01/15        | 52           |
| Revisión de la documentación    | 1 día         | jue 08/01/15        | jue 08/01/15        | 53           |
| Manuales                        | 1 día         | mar 06/01/15        | mar 06/01/15        | 40           |

**Figura 27:** Planificación documentación

### 2.6 Planificación temporal real del Proyecto

Durante el proceso de realización del proyecto una gran cantidad de los riesgos se produjeron, incluso los que tenían pocas probabilidades de ocurrir. Esto provocó muchos retrasos en el proyecto y por poco no se superó la convocatoria de julio debiendo presentar el trabajo en septiembre.

Los riesgos que se produjeron, en orden de aparición, fueron los siguientes:

1. Practicas de empresa. Se sabía que este riesgo era muy posible que apareciera, pero el problema fue la imposibilidad de poder compaginar la asignatura de prácticas con las clases y el desarrollo del proyecto.
2. Estudio de asignaturas. A las dos semanas de retomar el proyecto fue necesario realizar otro parón para el estudio de inglés al presentarse a los exámenes de Cambridge.
3. Estudio de asignaturas e indisposición del alumno. Tras realizar los exámenes de inglés y retomar el desarrollo, fue necesario parar de nuevo por la proximidad de unos exámenes y por enfermedad. Por lo que nos encontramos ante un nuevo parón en este caso de dos semanas.
4. Estudios de asignaturas. A inicios de mayo fue necesario parar el proyecto para el estudio de la asignatura de Evaluación de Sistemas Informáticos.

Por todo esto podemos estar seguros que el riesgo que más se ha repetido ha sido unas estimaciones imprecisas, y que pese a tener un gran margen con respecto a la convocatoria de julio todo esto casi hace que la realización del proyecto para la convocatoria de julio casi fracase.

El uso de lenguajes de programación conocidos (Python) ha acelerado el tiempo de desarrollo pensado para las aplicaciones con ellos, así como la reutilización de módulos de otras prácticas realizadas a lo largo de estos años, gracias a esto se ha podido emplear más tiempo al desarrollo con aquellos lenguajes de programación que eran nuevos hasta ahora (Vuforia).

A continuación podemos ver como ha quedado la planificación real del proyecto con los retrasos provocados.



## Trabajo Fin de Grado

| Nombre de tarea  | Duración | Comienzo     | Fin          |
|------------------|----------|--------------|--------------|
| ▲ Proyecto       | 197 días | lun 29/09/14 | mar 30/06/15 |
| ▷ Inicio         | 114 días | lun 29/09/14 | jue 05/03/15 |
| ▷ Analisis       | 39 días  | lun 19/01/15 | jue 12/03/15 |
| ▷ Diseño         | 66 días  | lun 19/01/15 | lun 20/04/15 |
| ▷ Implementacion | 45 días  | jue 16/04/15 | mié 17/06/15 |
| ▷ Pruebas        | 106 días | jue 29/01/15 | jue 25/06/15 |
| ▷ Documentación  | 3 días   | vie 26/06/15 | mar 30/06/15 |

**Figura 28:** Planificación real proyecto

Como puede verse en la siguiente imagen, la fase de inicio superó la planificación en 97 días.

| Nombre de tarea                      | Duración | Comienzo     | Fin          | Predecesoras |
|--------------------------------------|----------|--------------|--------------|--------------|
| ▲ Proyecto                           | 197 días | lun 29/09/14 | mar 30/06/15 |              |
| ▲ Inicio                             | 114 días | lun 29/09/14 | jue 05/03/15 |              |
| Asiganción de proyecto               | 1 día    | lun 29/09/14 | lun 29/09/14 |              |
| Redacción de la documentación        | 16 días  | mar 30/09/14 | jue 05/03/15 | 3            |
| Determinar alcance                   | 1 día    | mar 30/09/14 | mar 30/09/14 | 3            |
| Planificación                        | 3 días   | mié 01/10/14 | vie 03/10/14 | 3            |
| Definir recursos                     | 1 día    | jue 02/10/14 | jue 02/10/14 | 5            |
| Estimar la duración                  | 2 días   | lun 06/10/14 | lun 19/01/15 | 7            |
| Plan de control de proyecto          | 3 días   | lun 06/10/14 | mar 20/01/15 | 6            |
| Identificación y análisis de riesgos | 5 días   | mié 01/10/14 | lun 19/01/15 | 5            |

**Figura 29:** Planificación real inicio

Como puede verse en la siguiente imagen, la fase de análisis superó la planificación en 23 días.

## Trabajo Fin de Grado

| Nombre de tarea                           | Duración       | Comienzo            | Fin                 | Predecesoras |
|---|----------------|---------------------|---------------------|--------------|
| <b>▲ Análisis</b>                         | <b>39 días</b> | <b>lun 19/01/15</b> | <b>jue 12/03/15</b> |              |
| Redacción de la documentación de análisis | 16 días        | lun 19/01/15        | jue 12/03/15        |              |
| <b>▲ Análisis de requisitos</b>           | <b>5 días</b>  | <b>mar 20/01/15</b> | <b>lun 26/01/15</b> |              |
| Requisitos de usuario                     | 1 día          | mar 20/01/15        | mar 20/01/15        | 10           |
| Requisitos funcionales                    | 4 días         | mié 21/01/15        | lun 26/01/15        | 14           |
| Requisitos no                             | 1 día          | mié 21/01/15        | mié 21/01/15        | 14           |
| Requisitos de                             | 1 día          | mié 21/01/15        | mié 21/01/15        | 14           |
| <b>▲ Análisis de casos de uso</b>         | <b>28 días</b> | <b>mar 27/01/15</b> | <b>jue 05/03/15</b> |              |
| Identificación y realización de casos de  | 1 día          | mar 27/01/15        | mar 27/01/15        | 13           |
| Matriz de trazabilidad                    | 1 día          | mié 28/01/15        | mié 28/01/15        | 19           |
| Modelo del dominio                        | 1 día          | mié 28/01/15        | mié 28/01/15        | 19           |
| Diagramas de secuencia                    | 3 días         | jue 29/01/15        | jue 05/03/15        | 21           |

**Figura 30:** Planificación real análisis

Como puede verse en la siguiente imagen, la fase de diseño superó la planificación en 43 días.

| Nombre de tarea                         | Duración       | Comienzo            | Fin                 | Predecesoras |
|---|----------------|---------------------|---------------------|--------------|
| <b>▲ Diseño</b>                         | <b>66 días</b> | <b>lun 19/01/15</b> | <b>lun 20/04/15</b> |              |
| Redacción de la documentación de diseño | 23 días        | lun 19/01/15        | lun 06/04/15        |              |
| ▷ Descripción de la arquitectura        | 44 días        | lun 19/01/15        | jue 19/03/15        |              |
| ▷ Diseño de la arquitectura             | 18 días        | jue 12/03/15        | lun 06/04/15        |              |
| Casos de uso de diseño                  | 1 día          | mar 07/04/15        | mar 07/04/15        | 30           |
| Diagramas de secuencia de diseño        | 6 días         | mié 08/04/15        | mié 15/04/15        | 33           |
| Modelo del dominio de diseño            | 3 días         | jue 16/04/15        | lun 20/04/15        | 34           |

**Figura 31:** Planificación real diseño

Como puede verse en la siguiente imagen, la fase de implementación superó la planificación en 20 días.

## Trabajo Fin de Grado

| Nombre de tarea              | Duración       | Comienzo            | Fin                 | Predecesoras |
|------------------------------|----------------|---------------------|---------------------|--------------|
| ▸ <b>Implementación</b>      | <b>45 días</b> | <b>jue 16/04/15</b> | <b>mié 17/06/15</b> |              |
| Desarrollo del código        | 20 días        | jue 16/04/15        | mié 10/06/15        | 34           |
| Pruebas de los programadores | 18 días        | jue 23/04/15        | lun 15/06/15        | 37FC-75%     |
| Integración de los módulos   | 2 días         | mar 16/06/15        | mié 17/06/15        | 38           |

**Figura 32:** Planificación real implementación

Como puede verse en la siguiente imagen, la fase de pruebas superó la planificación en 53 días.

| Nombre de tarea        | Duración        | Comienzo            | Fin                 | Predecesoras |
|------------------------|-----------------|---------------------|---------------------|--------------|
| ▸ <b>Pruebas</b>       | <b>106 días</b> | <b>jue 29/01/15</b> | <b>jue 25/06/15</b> |              |
| ▸ <b>Caja blanca</b>   | <b>106 días</b> | <b>jue 29/01/15</b> | <b>jue 25/06/15</b> |              |
| Plan de pruebas de     | 3 días          | jue 29/01/15        | jue 05/03/15        | 21           |
| Pruebas de caja blanca | 3 días          | jue 18/06/15        | lun 22/06/15        | 42;39        |
| Identificar anomalías  | 1 día           | mar 23/06/15        | mar 23/06/15        | 43           |
| Modificar código       | 2 días          | mié 24/06/15        | jue 25/06/15        | 44           |
| ▸ <b>Caja negra</b>    | <b>106 días</b> | <b>jue 29/01/15</b> | <b>jue 25/06/15</b> |              |
| Plan de pruebas de     | 3 días          | jue 29/01/15        | jue 05/03/15        | 21           |
| Pruebas de caja negra  | 3 días          | jue 18/06/15        | lun 22/06/15        | 47;39        |
| Identificar anomalías  | 1 día           | mar 23/06/15        | mar 23/06/15        | 48           |
| Modificar código       | 2 días          | mié 24/06/15        | jue 25/06/15        | 49           |

**Figura 33:** Planificación real pruebas

Como puede verse en la siguiente imagen, la fase de documentación no superó la planificación.

| Nombre de tarea              | Duración      | Comienzo            | Fin                 | Predecesoras |
|------------------------------|---------------|---------------------|---------------------|--------------|
| ▸ <b>Documentación</b>       | <b>3 días</b> | <b>vie 26/06/15</b> | <b>mar 30/06/15</b> |              |
| Documentación del desarrollo | 1 día         | vie 26/06/15        | vie 26/06/15        | 40           |
| Integración de la            | 1 día         | lun 29/06/15        | lun 29/06/15        | 52           |
| Revisión de la documentación | 1 día         | mar 30/06/15        | mar 30/06/15        | 53           |
| Manuales                     | 1 día         | vie 26/06/15        | vie 26/06/15        | 40           |

**Figura 34:** Planificación real documentación

### **3. Análisis**

En este capítulo se presentará la elaboración del análisis del proyecto a desarrollar.

La fase de análisis es clave a la hora de realizar el proyecto en las fechas establecidas, por lo que se volcará una gran intensidad en su realización para evitar tener que realimentar el proyecto debido a un mal análisis.

### **3.1 Propósito del sistema**

La aplicación que se llevará a cabo tiene como principal propósito ser una ayuda para los alumnos durante los laboratorios de la asignatura de física, por ello también entra en el ámbito de la aplicación proporcionar un sistema sencillo para gestionar las bases de datos de la aplicación. Se detallan a continuación los objetivos que se persiguen con su elaboración y los requisitos que deberá cumplir.

### 3.2 Planteamiento del problema y objetivos

Durante la elaboración de las prácticas de laboratorio los estudiantes no cuentan con mucho tiempo para su desarrollo, por lo cual un buen conocimiento del funcionamiento de las máquinas de laboratorio puede ser determinante.

Para proporcionar el mayor apoyo a los mismos se llevará a cabo la elaboración de una aplicación que reconozca los aparatos del laboratorio, dibuje botones virtuales sobre los reales e indique al alumno la funcionalidad de estos.

Para que este proyecto pueda ser retomado en un futuro y ampliadas sus funcionalidades se desarrollará buscando la máxima flexibilidad y escalabilidad para los intereses de los stakeholders presentados a continuación:

| Nombre        | Descripción                                 | Responsabilidades   |
|---------------|---|---|
| Usuario final | Cualquier alumno de la asignatura de física | <ul style="list-style-type: none"> <li>• Consultar información sobre los elemento del laboratorio</li> </ul>  |
| Administrador | Encargado de gestionar las bases de datos   | <ul style="list-style-type: none"> <li>• Mantener la disponibilidad de la base de datos</li> <li>• Encargado del mantenimiento y actualización de la base de datos</li> <li>• Realizar cambios en la base de datos</li> </ul> |

**Tabla 8:** Usuarios y responsabilidades

La aplicación tendrá que permitir que los stakeholders lleven a cabo sus responsabilidades. También es necesario llevar a cabo un análisis de las principales necesidades y características, lo que nos ayudará en la resolución de problemas, las cuales se presentan en la siguiente tabla.

| Necesidad                      | Prioridad | Características  |
|--------------------------------|-----------|--|
| Listar bases de datos          | Alta      | <ul style="list-style-type: none"> <li>• Proporcionar un listado con todas las bases de datos disponibles ordenadas por su fecha</li> </ul>                  |
| Descargar base de datos        | Alta      | <ul style="list-style-type: none"> <li>• Descargar una nueva base de datos que sustituirá a la local disponible en el dispositivo</li> </ul>                 |
| Crear/Editar una base de datos | Alta      | <ul style="list-style-type: none"> <li>• Crear una nueva base de datos desde cero o editar una ya existente generando una nueva base de datos con</li> </ul> |

# Trabajo Fin de Grado

---

|  |  |                    |
|--|--|--------------------|
|  |  | los nuevos valores |
|--|--|--------------------|

**Tabla 9:** Necesidades y características

## 3.3 Requisitos

### 3.3.1 Requisitos de usuario

- RU01.- El usuario administrador podrá crear nuevos botones virtuales.
- RU02.- El usuario administrador podrá cancelar la creación de nuevos botones virtuales.
- RU03.- El usuario administrador podrá editar los botones virtuales.
- RU04.- El usuario administrador podrá cancelar la edición de los botones virtuales.
- RU05.- El usuario administrador podrá eliminar los botones virtuales.
- RU06.- El usuario administrador establecerá el tamaño original de las imágenes a reconocer.
- RU07.- El usuario administrador podrá modificar el tamaño original de las imágenes a reconocer.
- RU08.- El usuario administrador podrá salir sin guardar la base de datos.
- RU09.- El usuario administrador podrá crear bases de datos para distintos idiomas.
- RU010.- El usuario final podrá descargar de entre una lista una base de datos nueva.
- RU011.- El usuario final podrá cancelar la descarga de una base de datos nueva.
- RU012.- El usuario final podrá pulsar los botones virtuales y obtener información.
- RU013.- El usuario final podrá consultar información sobre la aplicación.

### 3.3.2 Requisitos funcionales

- RF01.- El sistema deberá reconocer los elementos del laboratorio.
- RF02.- El sistema permitirá al usuario final descargar diferentes bases de datos.
- RF03.- El sistema permitirá al usuario final cancelar la descarga de una base de datos.
- RF04.- El sistema proveerá un listado de actualizaciones.
- RF05.- El sistema permitirá añadir botones virtuales a los elementos a reconocer.
- RF06.- El sistema permitirá borrar botones virtuales a los elementos a reconocer.
- RF07.- El sistema permitirá editar botones virtuales a los elementos a reconocer.
- RF08.- El sistema mostrará los valores antiguos del botón que se esté editando.
- RF09.- El sistema mostrará al usuario final una animación mientras descarga las actualizaciones.
- RF010.- El sistema permitirá establecer el tamaño original de la imagen a reconocer.
- RF011.- El sistema permitirá modificar el tamaño original de la imagen a reconocer.
- RF012.- El sistema recalculará los botones virtuales si se modifica el tamaño original.
- RF013.- El sistema recuperará los valores de las imágenes a reconocer anteriores al crear una nueva base de datos si estas se encuentran presentes en el nuevo archivo.
- RF014.- El sistema validará los nuevos botones para asegurarse de que son válidos.
- RF015.- El sistema permitirá al usuario final reconocer los elementos base sin tener que descargar archivos.
- RF016.- El sistema permitirá al usuario administrador guardar la nueva base de datos.



RF017.- El sistema permitirá al usuario administrador salir sin guardar ningún cambio.

RF018.- El sistema separará los diferentes archivos por carpetas.

RF019.- El sistema permitirá crear bases de datos en distintos idiomas.

RF020.- El sistema solo mostrará al usuario final las bases de datos correspondientes al idioma que use en su dispositivo.

RF021.- El sistema mostrará información sobre la aplicación al usuario final.

RF022.- El sistema permitirá al usuario administrador modificar el descriptor de los elementos a reconocer.

### **3.3.3 Requisitos no funcionales**

RNF01.- El servidor contará con un servidor apache 2.4.7.

RNF02.- El servidor contará con PHP 5.5.9.

RNF03.- El sistema funcionará con cualquier dispositivo con Android 4.0 (90,4% de los dispositivos).

RNF04.- El sistema deberá estar conectado a Internet para funcionar.

RNF05.- El sistema utilizará el framework de Qualcomm, Vuforia.

#### **Usabilidad**

RNF06.- Un usuario con conocimientos medios sobre las tecnologías podrá utilizar todas las funcionalidades de las que dispone el sistema.

RNF07.- Los mensajes de error de la aplicación deben poder ser fácilmente identificados por el usuario.

#### **Fiabilidad**

RNF08.- El servidor web estará disponible 24/7.

#### **Rendimiento**

RNF09.- El servidor responderá a las peticiones en un máximo de 3 segundos.

#### **Mantenibilidad**

RNF010.- El sistema estará completamente documentado contando con manuales de usuario.

RNF011.- El sistema será lo más simple y claro posible para un mejor mantenimiento futuro.

#### **Seguridad**

RNF011.- Los archivos serán accesibles en modo lectura desde el exterior por cualquier usuario ya que no contienen datos sensibles.

RNF012.- Solo el usuario administrador podrá modificar las bases de datos.

#### **Verificación de datos**

RNF013.- El sistema deberá validar la información introducida por el usuario administrado en cualquiera de las opciones del programa gestor de botones.

### **3.3.4 Requisitos de la información**

RI01.- El sistema guardará por separado los archivos referidos a la base de datos Vuforia y los archivos utilizados por la aplicación móvil.

RI02.- El sistema guardará información acerca de los elementos a reconocer, concretamente, el identificador de la imagen, el descriptor de la imagen, el tamaño de la imagen en la base de datos Vuforia, el tamaño original de la imagen, el identificador de los botones, las coordenadas de los botones transformadas, el estado de enabled para cada botón y el mensaje de los botones.

RI03.- El sistema guardará información sobre la versión actual de la base de datos.

## 3.4 Casos de uso

A continuación podemos ver los diagramas de casos de uso de análisis tanto para el usuario final como el administrador.

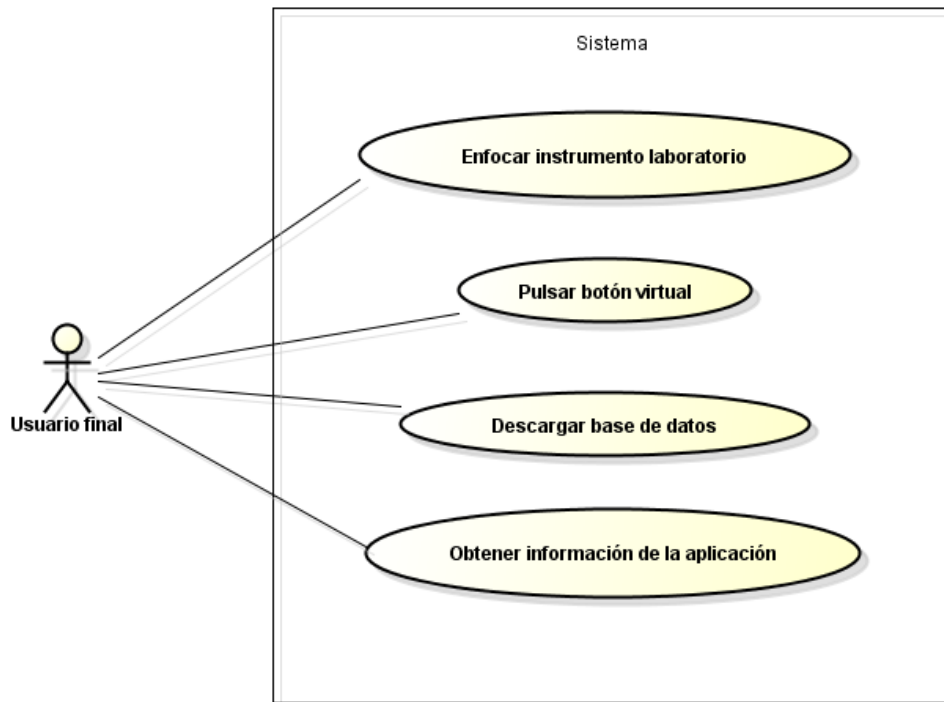


Figura 35: Diagrama casos de uso usuario final

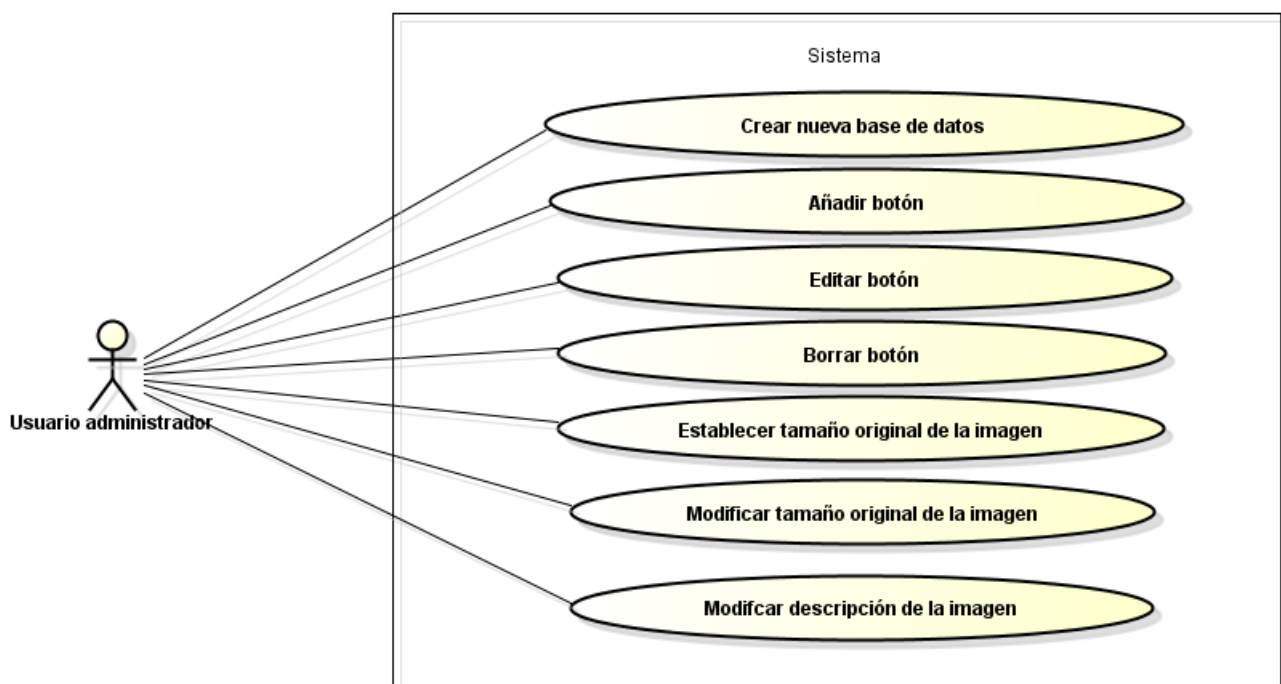


Figura 36: Diagrama casos de uso usuario administrador

## 3.4.1 Enfocar instrumento del laboratorio

### Descripción:

Un usuario final desea conocer si hay información en la aplicación sobre un elemento del laboratorio.

### Actores

- Usuario final.

### Precondiciones

No hay precondiciones.

### Flujo normal de eventos

1. El caso de uso comienza cuando el usuario inicia el reconocimiento de imágenes.
2. El sistema activa la cámara y comienza a capturar imágenes.
3. El usuario dirige la cámara del dispositivo hacia el elemento del laboratorio.
4. El sistema compara las imágenes captadas con la cámara con las de la base de datos.
5. El sistema muestra el nombre de la imagen reconocida.
6. El sistema dibuja los botones virtuales referidos a ese elemento.
7. El caso de uso termina.

### Flujo alternativo

#### *Elemento no encontrado*

Si en el paso 4 el sistema no encuentra ninguna concordancia con la base de datos.

1. El sistema continuará comparando hasta encontrar algo que concuerde con los elementos de la base de datos.

#### *Cancelación*

En cualquier paso del caso de uso el usuario cancela el reconocimiento de imágenes.

1. El caso de uso termina.

#### *No hay botones asociados a la imagen*

Si en el paso 4 el sistema encuentra una concordancia pero no tiene asociada ningún botón virtual.

1. El sistema muestra la descripción(si existe) de la imagen reconocida.
2. El caso de uso termina.

### Escenarios clave

Un usuario final de la aplicación, Carlos Bedit, se encuentra realizando una práctica de la asignatura de física y debido a su desconocimiento del elemento de laboratorio que tiene delante

decide utilizar la aplicación para comprobar qué es y para qué sirven los botones. Enfoca con su móvil el elemento de laboratorio y la aplicación le indica que es la fuente de alimentación y dibuja unos botones virtuales.

### Postcondiciones

#### *Finalización satisfactoria*

El usuario obtiene información sobre el elemento del laboratorio.

#### *Finalización por cancelación*

El usuario vuelve atrás.

#### *Finalización con errores*

El usuario no obtiene ninguna información sobre el elemento del laboratorio.

### 3.4.2 Pulsar botón virtual

#### Descripción:

Un usuario final desea conocer la función de cierto botón de un elemento del laboratorio.

#### Actores

- Usuario final.

#### Precondiciones

- El usuario ha enfocado el elemento del laboratorio.
- El elemento del laboratorio se encuentra en la base de datos del dispositivo.
- El sistema ha reconocido el elemento.
- El elemento del laboratorio tiene algún botón virtual.

#### Flujo normal de eventos

1. El caso de uso comienza cuando el usuario pulsa el botón virtual.
2. El sistema accede a la descripción de la funcionalidad de ese botón.
3. El sistema muestra la información al usuario.
4. El caso de uso termina.

#### Escenarios clave

Uno de los usuario finales, Víctor Valdecantos, ha enfocado un elemento de laboratorio para conocer sus funcionalidades, la aplicación lo ha reconocido y ha dibujado unos botones virtuales, Víctor pulsa uno de esos botones y la aplicación le devuelve la descripción de la utilidad de ese botón.

## Postcondiciones

### *Finalización satisfactoria*

El usuario obtiene información sobre el botón de un elemento del laboratorio.

## 3.4.3 Descargar base de datos

### **Descripción:**

El usuario final desea descargar una nueva base de datos para de esta manera reconocer más elementos del laboratorio..

### **Actores**

- Usuario final.

### **Precondiciones**

- Base de datos nueva disponible para descargar

### **Flujo normal de eventos**

1. El caso de uso comienza cuando el usuario va a actualizar.
2. El sistema comprueba la conexión a internet.
3. El sistema conecta con el servidor y descarga la lista de bases de datos.
4. El sistema le muestra la lista de las bases de datos disponibles ordenadas por fecha.
5. El usuario selecciona una base de datos.
6. El sistema descarga esa base de datos.
7. El sistema sobrescribe los archivos antiguos con los nuevos valores.
8. El sistema vuelve atrás.
9. El caso de uso termina.

### **Flujo alternativo**

#### *Cancelación de descarga*

Si en cualquier paso el usuario cancela la descarga de la base de datos.

1. El sistema cancela la descarga.
2. El sistema vuelve atrás.
3. El caso de uso termina.

#### *Activar internet*

Si en el paso 2 el sistema detecta que no hay conexión a internet.

1. El sistema abre la configuración de internet.

2. El usuario activa internet.
3. El caso de uso continua en el paso 3.

## **Escenarios clave**

Uno de los usuarios finales, Javier Fernández, va a visualizar las actualizaciones disponibles, tras comprobar que puede descargar una base de datos más nueva, selecciona esa base de datos y comienza la descarga

## **Postcondiciones**

### *Finalización satisfactoria*

Los archivos correspondientes a la nueva base de datos para reconocer los elementos del laboratorio se han guardado de manera local en el dispositivo.

### *Finalización por cancelación*

El usuario final continua con la misma base de datos con el mismo número de imágenes a reconocer.

## **3.4.4 Ver información de la aplicación**

### **Descripción:**

Un usuario final desea conocer información referida a la aplicación.

### **Actores**

- Usuario final.

### **Precondiciones**

No hay precondiciones.

### **Flujo normal de eventos**

1. El caso de uso comienza cuando el usuario va a la información de la aplicación.
2. El sistema muestra la información referida a la aplicación al usuario.
3. El caso de uso termina.

## **Escenarios clave**

Un usuario final de la aplicación, Juan Mieres, tiene en breves que realizar una práctica de física, antes de acudir a la práctica quiere saber qué versión de base de datos tiene en el dispositivo. Va a información y ve la versión y otros datos.

## **Postcondiciones**

## *Finalización satisfactoria*

El usuario obtiene información sobre la aplicación.

### **3.4.5 Añadir un botón virtual**

#### **Descripción:**

Un usuario administrador desea añadir un botón virtual a un elemento del laboratorio.

#### **Actores**

- Usuario administrador.

#### **Precondiciones**

- Se ha seleccionado un elemento.
- Se ha establecido el tamaño original de la imagen.
- Se ha seleccionado el elemento al cual se añadirá información.

#### **Flujo normal de eventos**

1. El caso de uso comienza cuando el usuario selecciona añadir botón.
2. El sistema pide los datos referidos al nuevo botón al usuario.
3. El usuario introduce los datos del nuevo botón.
4. El sistema valida los datos.
5. El sistema almacena los datos referidos al nuevo botón.
6. El caso de uso termina.

#### **Flujo alternativo**

##### *Datos inválidos*

Si en el paso 4 el sistema determina que los valores no son válidos.

1. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 2.

##### *Cancelación*

En cualquiera de los pasos el usuario indica que quiere cancelar el caso de uso.

1. El caso de uso finaliza

#### **Escenarios clave**

Un usuario administrador, Manuel González, se encuentra añadiendo información sobre un elemento del laboratorio, decide crear un botón virtual para explicar la funcionalidad de un botón o un grupo de botones reales. Introduce su nombre, las coordenadas, el estado de enabled y el mensaje que se mostrará.



## Postcondiciones

### *Finalización satisfactoria*

Los cambios quedan reflejados en el sistema.

### *Finalización por cancelación*

Los cambios no se llevan a cabo y la información del sistema no se ve alterada.

## 3.4.6 Editar un botón virtual

### Descripción:

Un usuario administrador desea editar un botón virtual de un elemento del laboratorio.

### Actores

- Usuario administrador.

### Precondiciones

- Se ha establecido el tamaño original de la imagen.
- Se ha seleccionado el elemento al cual se modificará la información.
- El elemento seleccionado posee al menos un botón virtual.

### Flujo normal de eventos

1. El caso de uso comienza cuando el usuario selecciona editar botón.
2. El sistema pide al usuario que indique el botón que quiere editar.
3. El usuario selecciona entre los botones existentes el que desea editar.
4. El sistema pide los datos referidos al botón al usuario.
5. El usuario introduce los nuevos datos del botón.
6. El sistema valida los datos.
7. El sistema almacena los nuevos datos referidos al botón.
8. El caso de uso termina.

### Flujo alternativo

#### *Datos inválidos*

Si en el paso 6 el sistema determina que los valores no son válidos.

1. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 4.

#### *Cancelación*

En cualquiera de los pasos el usuario indica que quiere cancelar el caso de uso.

1. El caso de uso finaliza.

## **Escenarios clave**

Un usuario administrador, Manuel González, se encuentra modificando la información sobre los botones de un elementos del laboratorio, decide alterar un botón virtual. Introduce su nuevo nombre, las nuevas coordenadas, el nuevo estado de enabled y el nuevo mensaje que se mostrará.

## **Postcondiciones**

### *Finalización satisfactoria*

Los cambios quedan reflejados en el sistema.

### *Finalización por cancelación*

Los cambios no se llevan a cabo y la información del sistema no se ve alterada.

## **3.4.7 Eliminar botón virtual**

### **Descripción:**

Un usuario administrador desea eliminar un botón virtual de un elemento del laboratorio.

### **Actores**

- Usuario administrador.

### **Precondiciones**

- Se ha establecido el tamaño original de la imagen.
- Se ha seleccionado el elemento al cual se modificará la información.
- El elemento seleccionado posee al menos un botón virtual.

### **Flujo normal de eventos**

1. El caso de uso comienza cuando el usuario selecciona eliminar botón.
2. El sistema pide al usuario que indique el botón que quiere eliminar.
3. El usuario selecciona entre los botones existentes el que desea eliminar.
4. El sistema valida el botón.
5. El sistema elimina el botón.
6. El caso de uso termina.

### **Flujo alternativo**

#### *Datos inválidos*

Si en el paso 4 el sistema determina que los valores no son válidos.

1. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 2.

### ***Cancelación***

En cualquiera de los pasos el usuario indica que quiere cancelar el caso de uso.

1. El caso de uso finaliza.

### **Escenarios clave**

Un usuario administrador, Manuel González, se encuentra modificando la información sobre los botones de un elementos del laboratorio, decide eliminar un botón virtual. Selecciona el botón a eliminar y este es borrado de la información del sistema.

### **Postcondiciones**

#### ***Finalización satisfactoria***

Los cambios quedan reflejados en el sistema.

#### ***Finalización por cancelación***

Los cambios no se llevan a cabo y la información del sistema no se ve alterada.

## **3.4.8 Establecer tamaño original de la imagen**

### **Descripción:**

Un usuario administrador desea añadir información a un elemento de la base de datos por primera vez, por lo cual tendrá que indicar el tamaño original de la imagen.

### **Actores**

- Usuario administrador.

### **Precondiciones**

- Primera vez que se selecciona el elemento de la base de datos.
- No tiene asociado ningún botón virtual.
- No se ha definido su tamaño original.

### **Flujo normal de eventos**

1. El caso de uso comienza cuando el usuario selecciona un elemento de la base de datos.
2. El sistema comprueba y verifica que falta el valor de tamaño original.
3. El sistema solicita al usuario el tamaño original de la imagen.
4. El usuario introduce el valor del tamaño original.
5. El sistema valida los datos.
6. El sistema almacena el valor.

7. El caso de uso termina.

## **Flujo alternativo**

### *Datos inválidos*

Si en el paso 5 el sistema determina que los valores no son válidos.

1. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 3.

### *Cancelación*

En cualquiera de los pasos el usuario indica que quiere cancelar el caso de uso.

1. El caso de uso termina

## **Escenarios clave**

Un usuario administrador, Manuel González, quiere modificar un elemento de la base de datos. Debido a que es la primera vez que edita ese elemento, el sistema le solicita el tamaño original de la imagen, Manuel indica el tamaño original, el sistema almacena el valor y Manuel puede continuar con la edición del elemento.

## **Postcondiciones**

### *Finalización satisfactoria*

El valor queda almacenado en el sistema.

### *Finalización por cancelación*

La información del sistema no se ve alterada.

## **3.4.9 Modificar tamaño de una imagen**

### **Descripción:**

Un usuario administrador desea modificar el tamaño original de un Target.

### **Actores**

- Usuario administrador.

### **Precondiciones**

No hay precondiciones.

### **Flujo normal de eventos**

1. El caso de uso comienza cuando el usuario selecciona modificar el tamaño original de un elemento.

2. El sistema pide al usuario el nuevo tamaño original.
3. El usuario introduce el nuevo tamaño original.
4. El sistema verifica los datos.
5. El sistema comprueba si el elemento posee algún botón, en caso afirmativo el sistema recalcula los botones con respecto al nuevo tamaño original.
6. El sistema almacena los datos.
7. El caso de uso termina.

### Flujo alternativo

#### *Datos inválidos*

Si en el paso 4 el sistema determina que los valores no son válidos.

1. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 2.

#### *Cancelación*

En cualquiera de los pasos el usuario indica que quiere cancelar el caso de uso.

1. El caso de uso finaliza

### Escenarios clave

Un usuario administrador, Manuel González, desea modificar el tamaño original de un elemento debido a que se equivocó a la hora de establecer este tamaño por primera vez.

### Postcondiciones

#### *Finalización satisfactoria*

Los cambios quedan reflejados en el sistema.

#### *Finalización por cancelación*

Los cambios no se llevan a cabo.

## 3.4.10 Crear base de datos

### Descripción:

El usuario administrador desea crear una base de datos con elementos del laboratorio a reconocer.

### Actores

- Usuario administrador.

### Precondiciones

- Se dispone de los dos archivos generados por el Target Manager de Vuforia.
- Se ha establecido la información correspondiente a los elementos a reconocer.

### **Flujo normal de eventos**

1. El caso de uso comienza cuando el usuario inicia el gestor de botones pasándole los argumentos e indicando el idioma de la base de datos antigua a cargar.
2. El sistema valida el idioma y los argumentos.
3. El sistema busca y carga la base de datos anterior y cruza los datos con los nuevos archivos.
4. El usuario realiza los casos de uso de añadir/editar/borrar botones y/o modificar el tamaño original. Y cuando ha terminado de establecer los parámetros de la nueva base de datos indica que desea guardar.
5. El sistema solicita al usuario el idioma asociado a la base de datos.
6. El usuario introduce el idioma asociado.
7. El sistema valida el idioma asociado.
8. El sistema crea los nuevos archivos.
9. El caso de uso termina.

### **Flujo alternativo**

#### ***Cancelación***

En el paso 4, el usuario decide no guardar los datos introducidos y salir sin guardar.

1. El caso de uso termina.

#### ***Datos inválidos***

En el paso 2 el sistema comprueba que el idioma asociado no es válido.

1. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 1.

#### ***Datos inválidos 2***

En el paso 7 el sistema comprueba que el idioma asociado no es válido.

1. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 5.

#### ***Base de datos inexistente***

En el paso 3 el sistema comprueba que no hay ninguna base de datos antigua.

1. El sistema utilizará solo los datos de entrada de los argumentos para crear la nueva base de datos.
2. El caso de uso continua en el paso 4.

#### ***Base de datos inexistente y argumentos vacío***

En el paso 3 el sistema comprueba que no hay ninguna base de datos antigua y no se han recibido ficheros por argumento.

1. El caso de uso termina con error.

### ***Base de datos inexistente***

En el paso 3 el sistema comprueba que no hay ninguna base de datos antigua.

1. El sistema utilizará solo los datos de entrada de los argumentos para crear la nueva base de datos.

### ***Argumentos vacios***

En el paso 1 el usuario no pasa ningún argumento.

1. El sistema carga la base de datos antigua con el idioma seleccionado y no la cruza con nada.
2. El caso de uso continua en el paso 4.

### **Escenarios clave**

Manuel González tras haber añadido la información sobre los elementos de laboratorio y sus botones desea crear una nueva base de datos para ser descargada por los alumnos. Selecciona guardar y se crean los archivos requeridos para descargar por la aplicación.

### **Postcondiciones**

#### ***Finalización satisfactoria***

Los nuevos archivos se encuentran disponibles para descargar en su directorio correspondiente.

#### ***Finalización por cancelación***

No se crea ningún archivo.

#### ***Finalización con errores***

No se crea ningún archivo.

## **3.4.11 Seleccionar elemento**

### **Descripción:**

Un usuario administrador desea añadir información a un elemento de la base de datos, por lo cual procede a seleccionar el elemento.

### **Actores**

- Usuario administrador.

## Precondiciones

- La base de datos dispone de elementos.

## Flujo normal de eventos

1. El caso de uso comienza cuando el usuario selecciona un elemento de la base de datos.
2. El sistema valida la entrada.
3. El sistema muestra las opciones que puede realizar sobre ese elemento.
4. El caso de uso termina.

## Flujo alternativo

### *Datos inválidos*

Si en el paso 2 el sistema determina que los valores no son válidos.

1. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 1.

## Escenarios clave

Un usuario administrador, Manuel González, quiere modificar un elemento de la base de datos. Manuel selecciona el elemento que quiere modificar y se lo introduce por pantalla para proceder a su edición.

## Postcondiciones

### *Finalización satisfactoria*

Se ha seleccionado el elemento con éxito y se puede proceder a su edición.

## 3.4.12 Modificar descripción de una imagen

### Descripción:

Un usuario administrador desea modificar la descripción de un Target.

### Actores

- Usuario administrador.

### Precondiciones

No hay precondiciones.

### Flujo normal de eventos

1. El caso de uso comienza cuando el usuario selecciona modificar la descripción de un elemento.



2. El sistema pide al usuario la nueva descripción.
3. El usuario introduce la nueva tamaño descripción.
4. El sistema verifica los datos.
5. El sistema almacena los datos.
6. El caso de uso termina.

## **Flujo alternativo**

### ***Datos inválidos***

Si en el paso 4 el sistema determina que los valores no son válidos.

1. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 2.

### ***Cancelación***

En cualquiera de los pasos el usuario indica que quiere cancelar el caso de uso.

1. El caso de uso finaliza

## **Escenarios clave**

Un usuario administrador, Manuel González, desea modificar la descripción de un elemento debido a que se equivocó a la hora de establecer esta descripción anteriormente. Selecciona el elemento del laboratorio a modificar la descripción e introduce el nuevo valor. La nueva descripción queda almacenada.

## **Postcondiciones**

### ***Finalización satisfactoria***

Los cambios quedan reflejados en el sistema.

### ***Finalización por cancelación***

Los cambios no se llevan a cabo.

### 3.5 Matriz de trazabilidad

Se presenta a continuación la matriz de trazabilidad, la cual relaciona los casos de uso identificados con los requisitos funciones.

#### **Requisitos funcionales**

- RF01.- El sistema deberá reconocer los elementos del laboratorio.
- RF02.- El sistema permitirá al usuario final descargar diferentes bases de datos.
- RF03.- El sistema permitirá al usuario final cancelar la descarga de una base de datos.
- RF04.- El sistema proveerá un listado de actualizaciones.
- RF05.- El sistema permitirá añadir botones virtuales a los elementos a reconocer.
- RF06.- El sistema permitirá borrar botones virtuales a los elementos a reconocer.
- RF07.- El sistema permitirá editar botones virtuales a los elementos a reconocer.
- RF08.- El sistema mostrará los valores antiguos del botón que se esté editando.
- RF09.- El sistema mostrará al usuario final una animación mientras descarga las actualizaciones.
- RF010.- El sistema permitirá establecer el tamaño original de la imagen a reconocer.
- RF011.- El sistema permitirá modificar el tamaño original de la imagen a reconocer.
- RF012.- El sistema recalculará los botones virtuales si se modifica el tamaño original.
- RF013.- El sistema recuperará los valores de las imágenes a reconocer anteriores al crear una nueva base de datos si estas se encuentran presentes en el nuevo archivo.
- RF014.- El sistema validará los nuevos botones para asegurarse de que son válidos.
- RF015.- El sistema permitirá al usuario final reconocer los elementos base sin tener que descargar archivos.
- RF016.- El sistema permitirá al usuario administrador guardar la nueva base de datos.
- RF017.- El sistema permitirá al usuario administrador salir sin guardar ningún cambio.
- RF018.- El sistema separará los diferentes archivos por carpetas.
- RF019.- El sistema permitirá crear bases de datos en distintos idiomas.
- RF020.- El sistema solo mostrará al usuario final las bases de datos correspondientes al idioma que use en su dispositivo.
- RF021.- El sistema mostrará información sobre la aplicación al usuario final.
- RF022.- El sistema permitirá al usuario administrador modificar el descriptor de los elementos a reconocer.

#### **Casos de uso**

- CU01.- Enfocar instrumento del laboratorio.
- CU02.- Pulsar un botón virtual.
- CU03.- Descargar una base de datos.
- CU04.- Ver información de la aplicación.
- CU05.- Añadir un botón virtual.

## Trabajo Fin de Grado

---

CU06.- Editar un botón virtual.

CU07.- Eliminar un botón virtual.

CU08.- Establecer tamaño de la imagen virtual.

CU09.- Modificar tamaño de la imagen virtual.

CU010.- Crear base de datos.

CU011.- Seleccionar elemento.

CU012.- Modificar descripción de una imagen.

### Matriz

Se busca plasmar en la matriz de trazabilidad si todos los requisitos están cubiertos y dónde.

|       | CU01 | CU02 | CU03 | CU04 | CU05 | CU06 | CU07 | CU08 | CU09 | CU010 | CU011 | CU012 |
|-------|------|------|------|------|------|------|------|------|------|-------|-------|-------|
| RF01  | X    | X    |      |      |      |      |      |      |      |       |       |       |
| RF02  |      |      | X    |      |      |      |      |      |      |       |       |       |
| RF03  |      |      | X    |      |      |      |      |      |      |       |       |       |
| RF04  |      |      | X    |      |      |      |      |      |      |       |       |       |
| RF05  |      |      |      |      | X    |      |      |      |      |       | X     |       |
| RF06  |      |      |      |      |      |      | X    |      |      |       | X     |       |
| RF07  |      |      |      |      |      | X    |      |      |      |       | X     |       |
| RF08  |      |      |      |      |      | X    |      |      |      |       | X     |       |
| RF09  |      |      | X    |      |      |      |      |      |      |       |       |       |
| RF010 |      |      |      |      |      |      |      | X    |      |       | X     |       |
| RF011 |      |      |      |      |      |      |      |      | X    |       | X     |       |
| RF012 |      |      |      |      |      |      |      |      | X    |       |       |       |
| RF013 |      |      |      |      |      | X    | X    |      | X    | X     |       |       |
| RF014 |      |      |      |      | X    | X    |      |      |      |       | X     |       |
| RF015 | X    | X    |      |      |      |      |      |      |      |       |       |       |
| RF016 |      |      |      |      |      |      |      |      |      | X     |       |       |
| RF017 |      |      |      |      |      |      |      |      |      | X     |       |       |
| RF018 |      |      |      |      |      |      |      |      |      | X     |       |       |
| RF019 |      |      |      |      |      |      |      |      |      | X     |       |       |
| RF020 | X    | X    | X    |      |      |      |      |      |      |       |       |       |
| RF021 |      |      |      | X    |      |      |      |      |      |       |       |       |
| RF022 |      |      |      |      |      |      |      |      |      |       | X     | X     |

**Tabla 10:** Matriz de trazabilidad

### 3.6 Modelo del dominio

Este es el diagrama de clases correspondiente al modelo del dominio de análisis. En el describiremos las diferentes entidades, atributos, papeles, relaciones y las restricciones del dominio del problema.

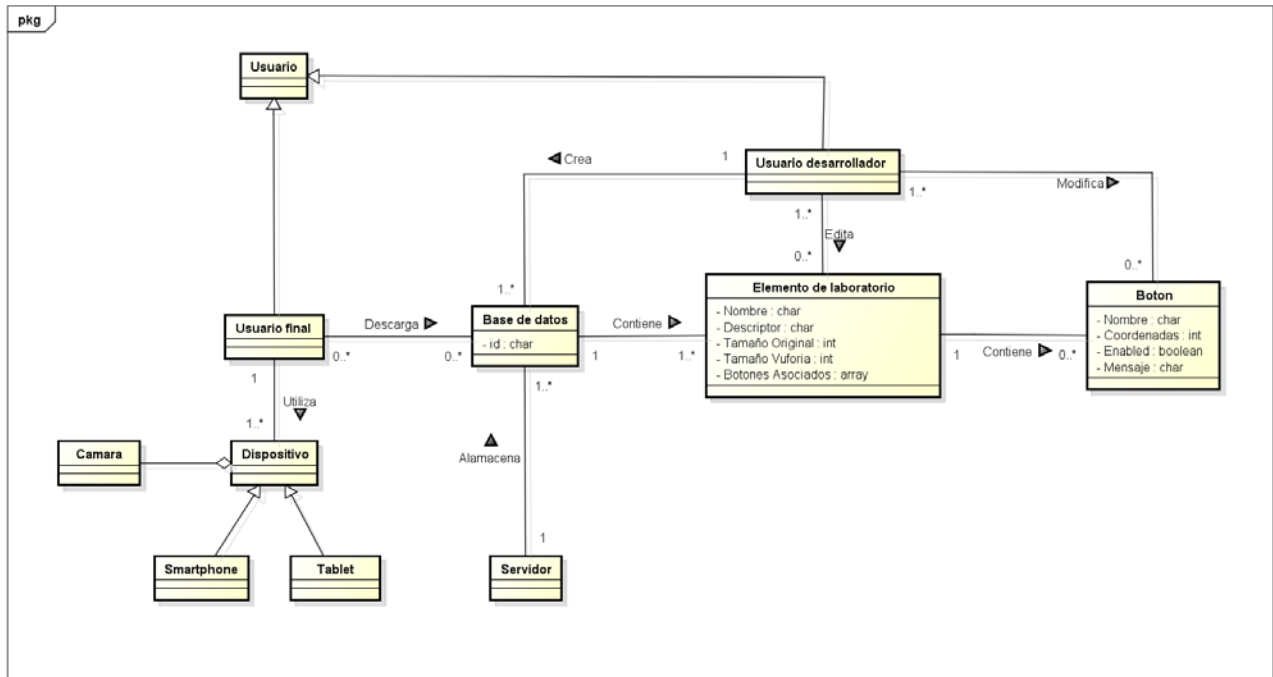


Figura 37: Modelo del dominio

### **3.7 Diagramas de secuencia**

A continuación se mostrarán los diagramas de secuencia de los casos de uso. En los cuales se modelarán las interacciones entre los objetos según UML.

Estos diagramas nos ayudarán a determinar los objetos necesarios para la implementación de los distintos escenarios.

### 3.7.1 Enfocar instrumento de laboratorio

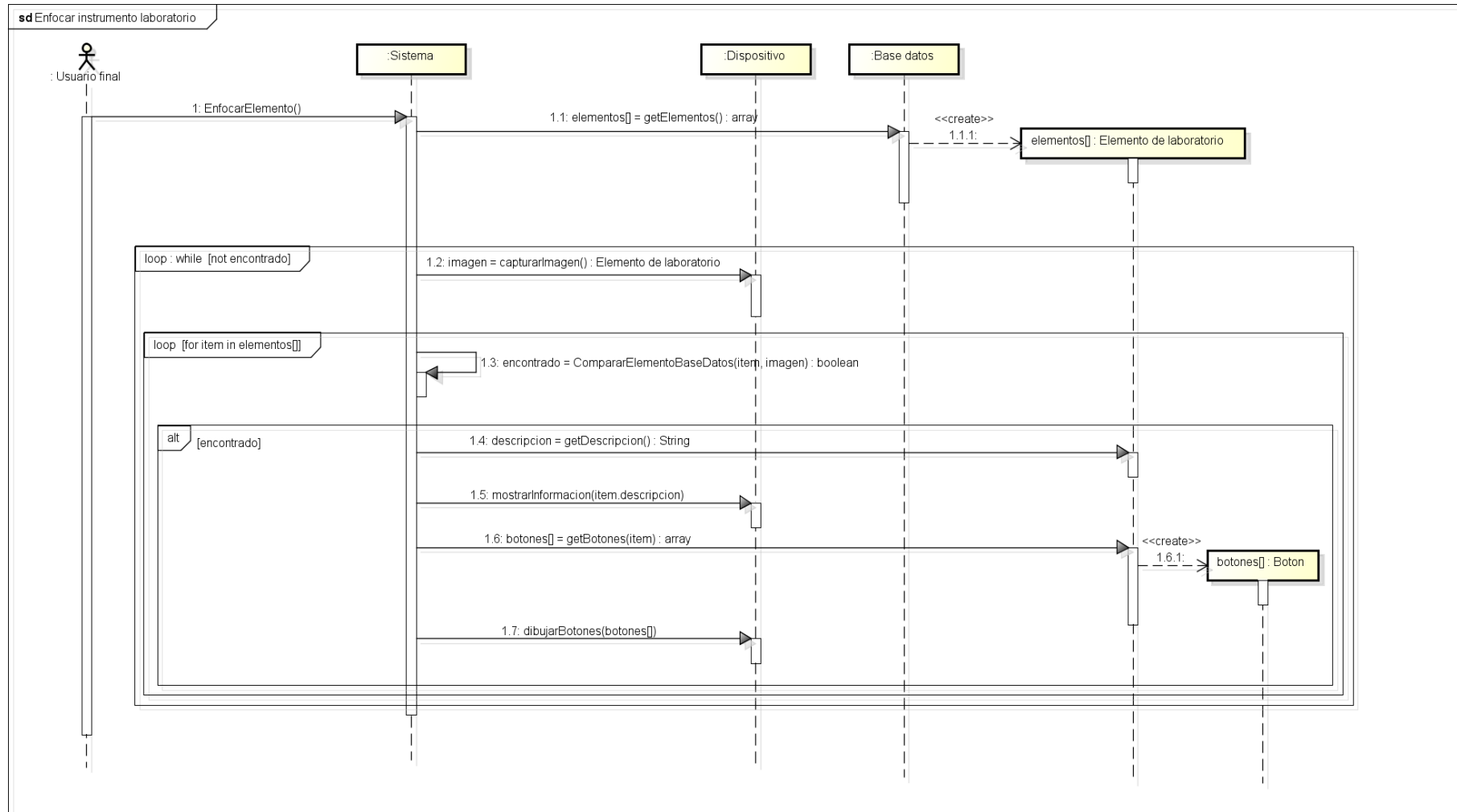


Figura 38: diagrama de secuencia

### 3.7.2 Pulsar botón

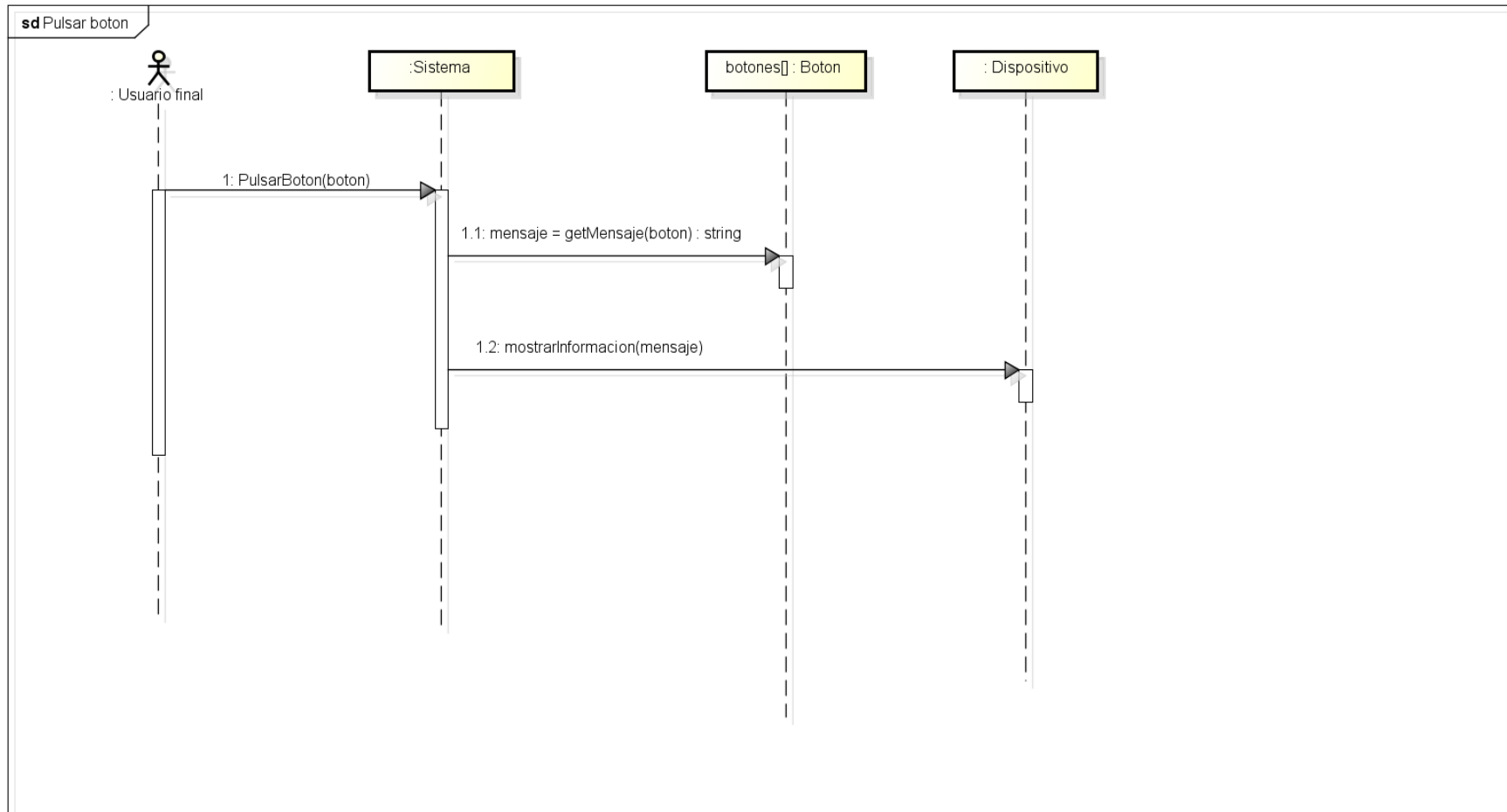


Figura 39: diagrama de secuencia

### 3.7.3 Descargar base de datos

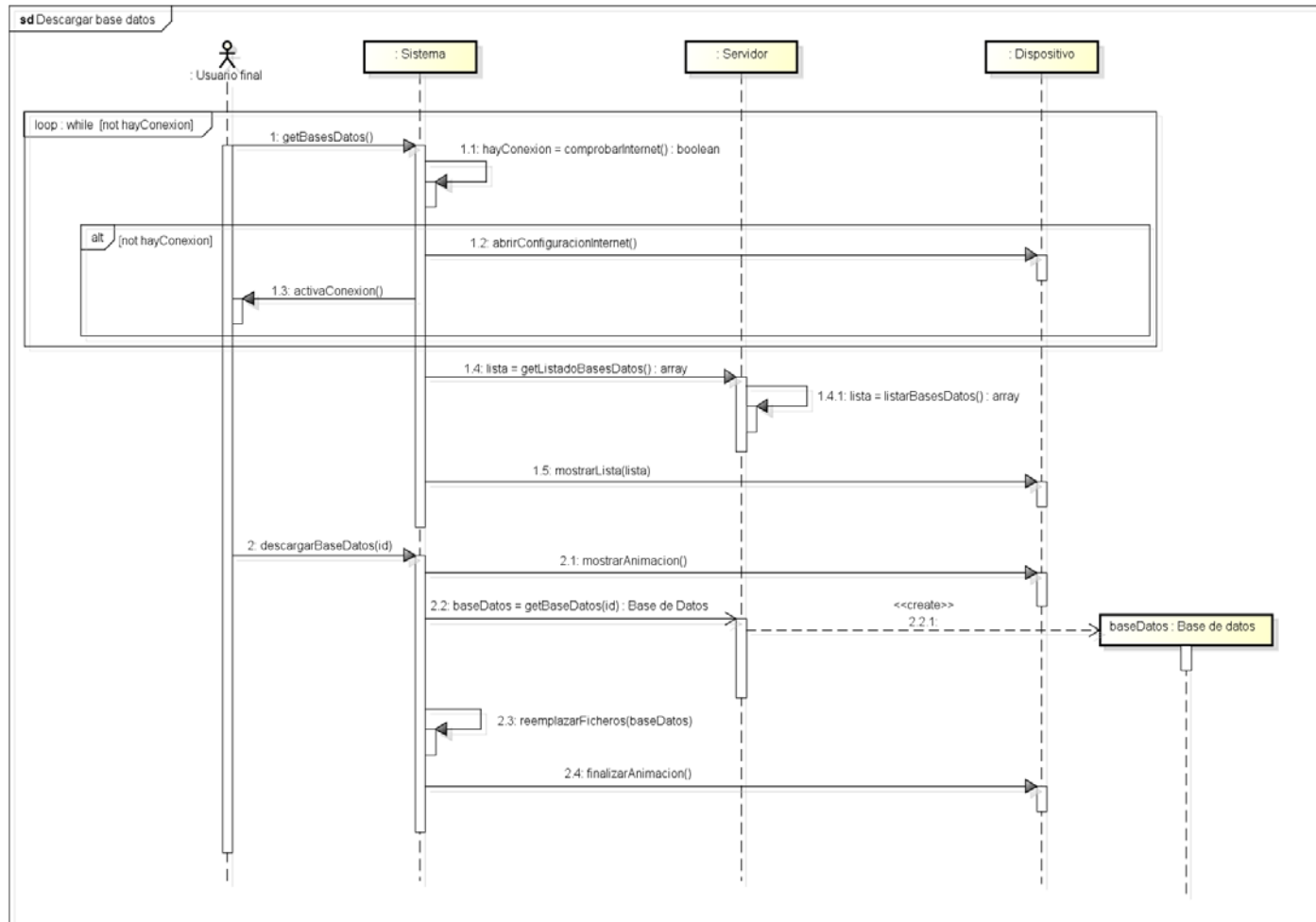


Figura 40: diagrama de secuencia



### 3.7.4 Crear base de datos

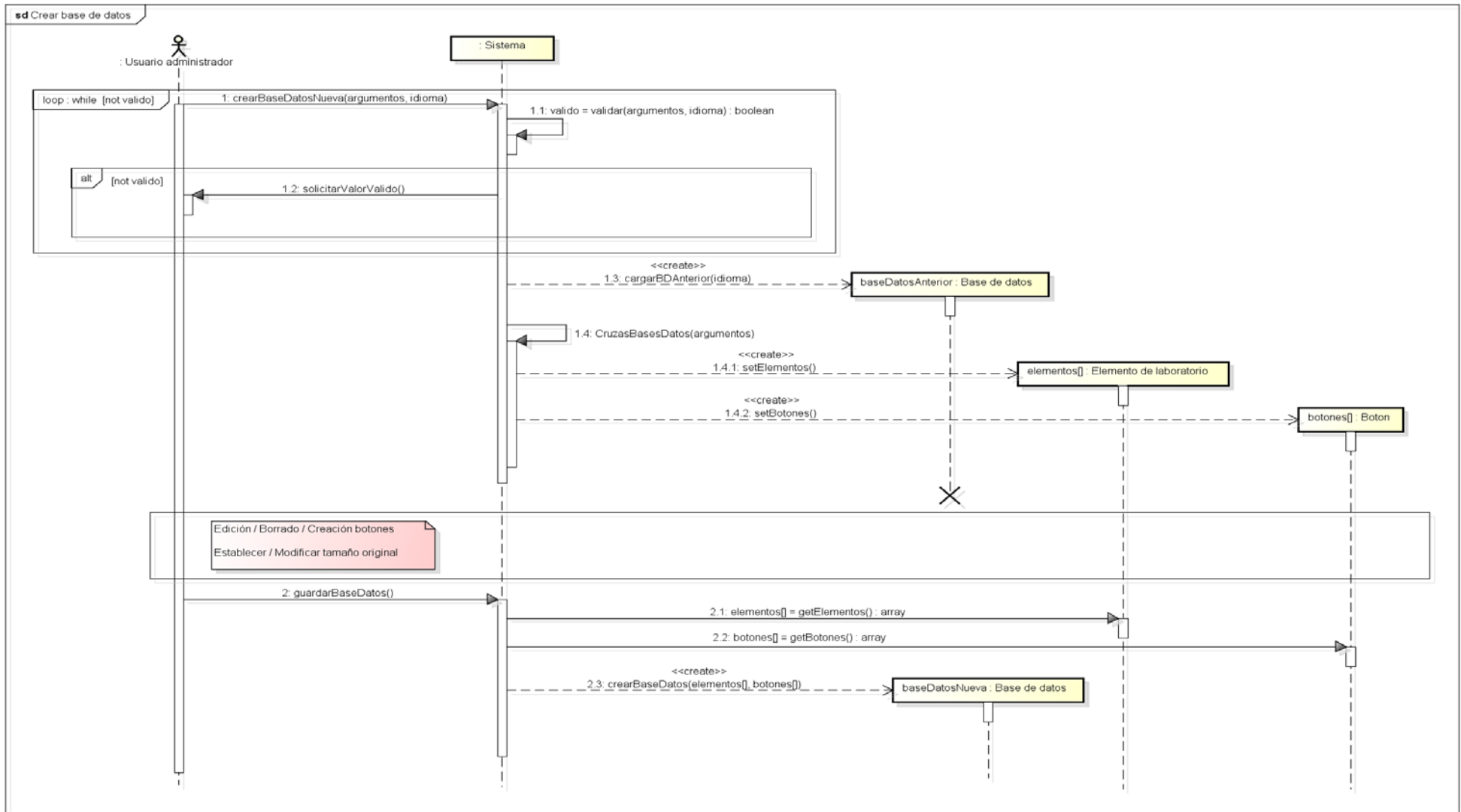


Figura 41: diagrama de secuencia

### 3.7.5 Añadir botón

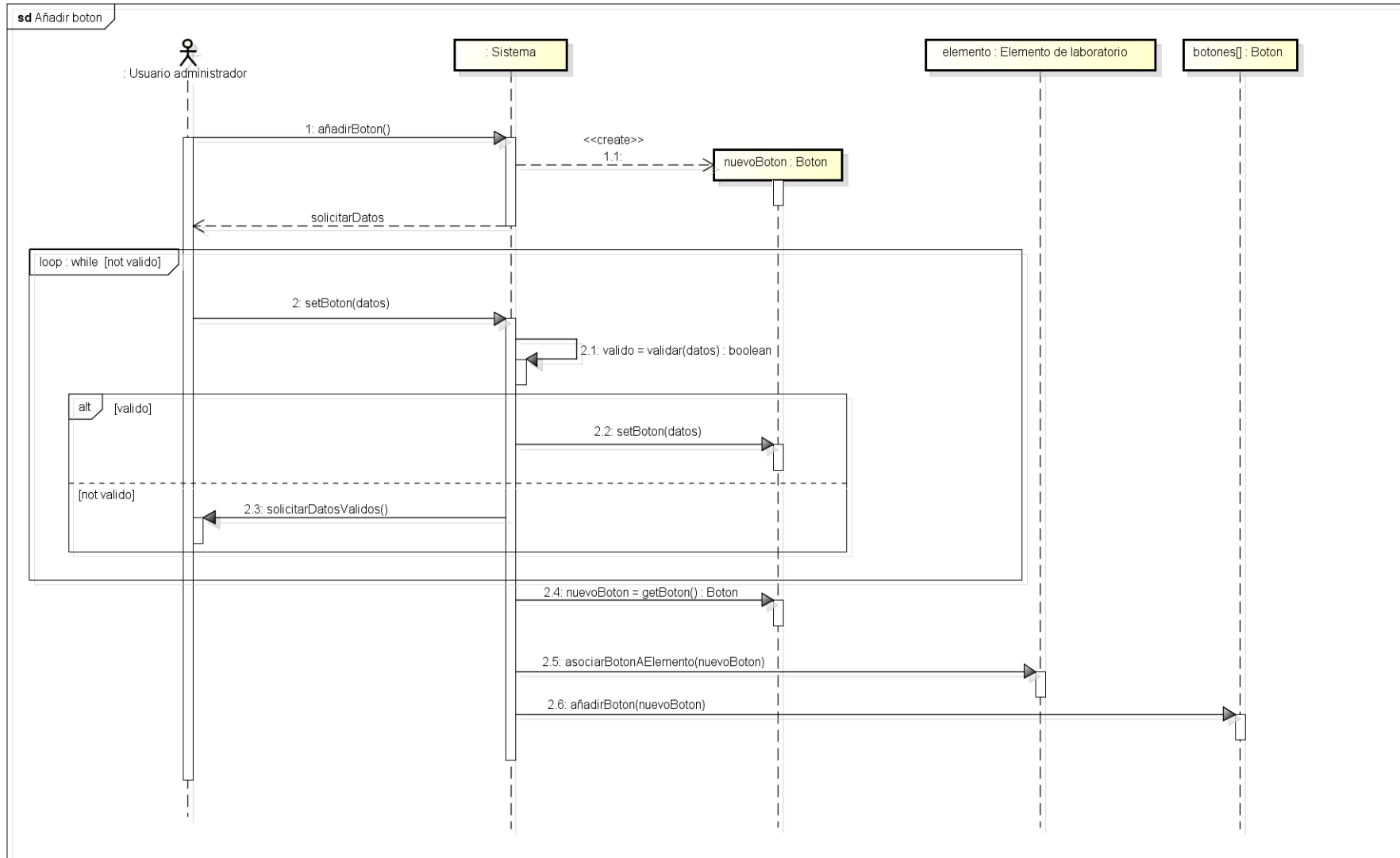


Figura 42: diagrama de secuencia

3.7.6 Editar botón

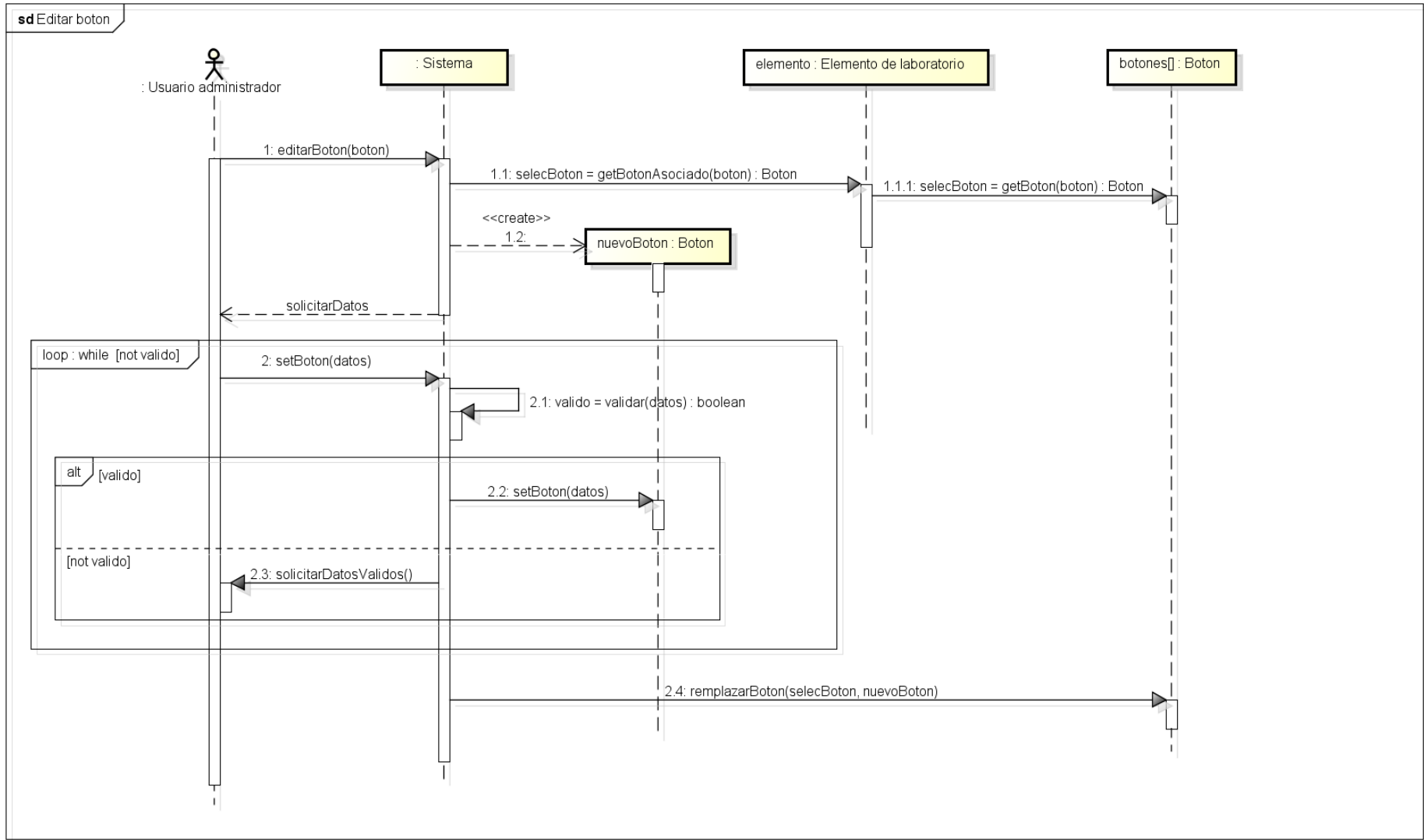


Figura 43: diagrama de secuencia

## 3.7.7 Borrar botón

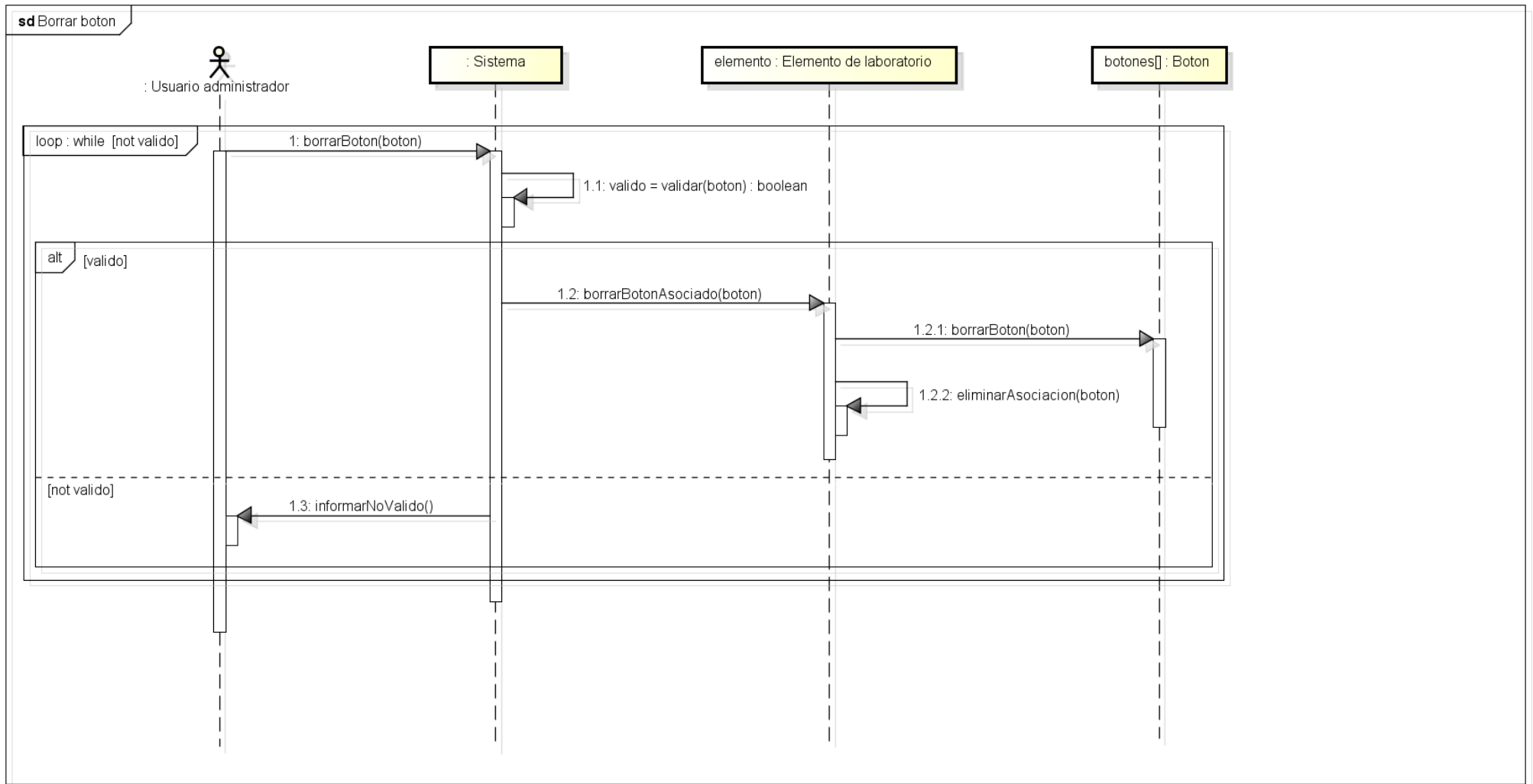


Figura 44: diagrama de secuencia

## 3.7.8 Seleccionar y establecer tamaño

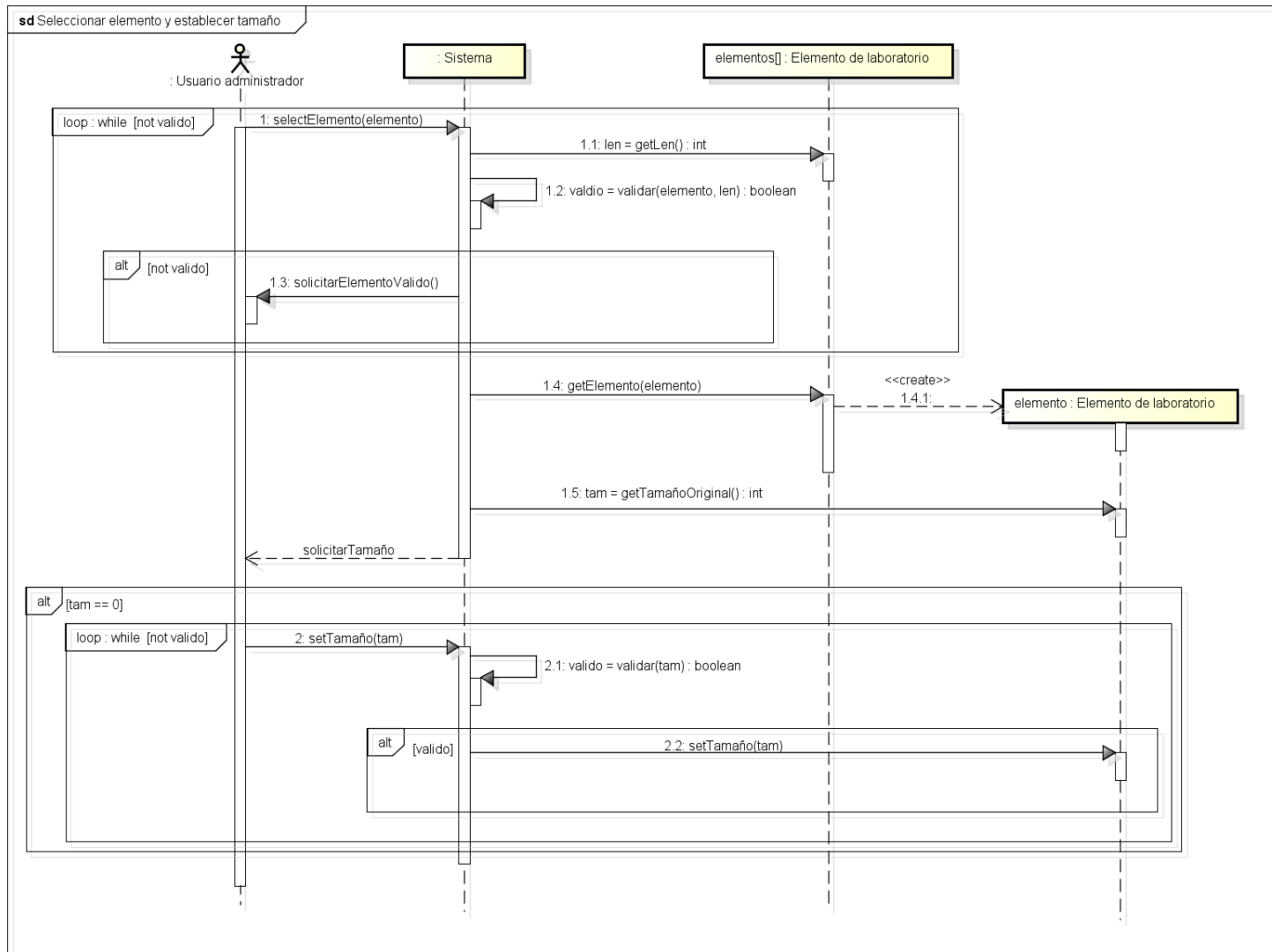


Figura 45: diagrama de secuencia

### 3.7.9 Modificar tamaño original

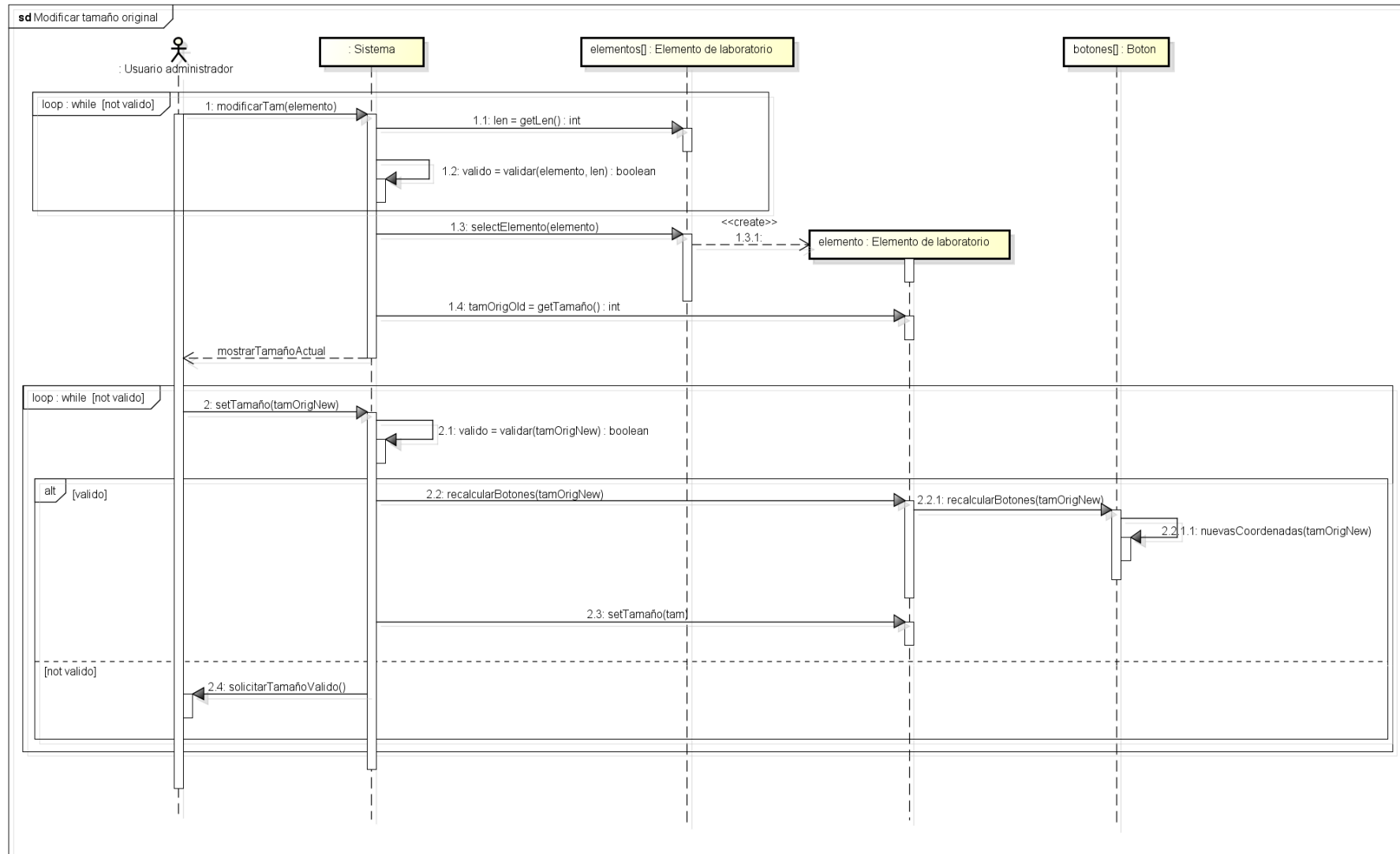


Figura 46: diagrama de secuencia

3.7.10 Ver información de la aplicación

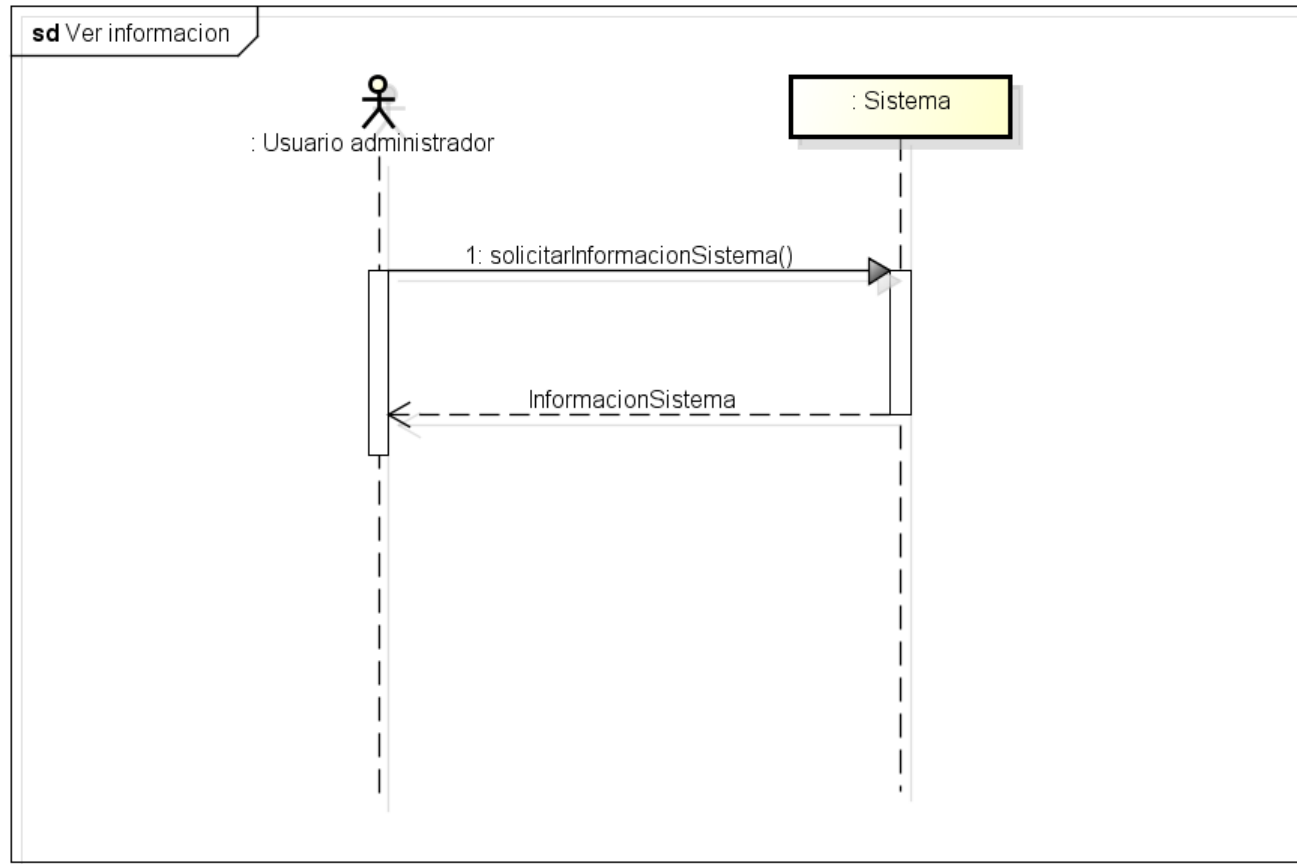


Figura 47: diagrama de secuencia

### 3.7.11 Modificar descriptor de la imagen

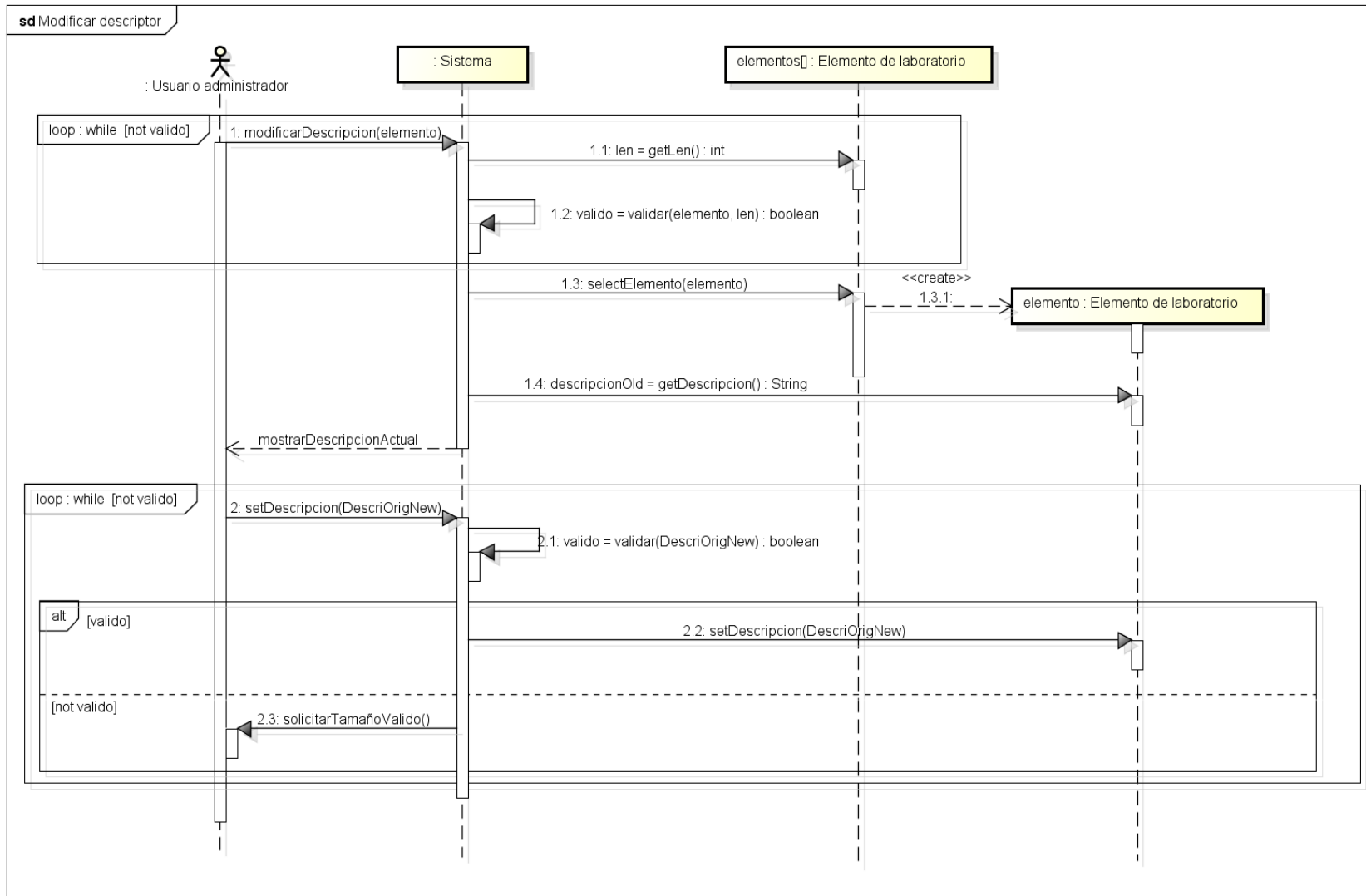


Figura 48: diagrama de secuencia



### **4. Diseño**

Se presentará a lo largo del siguiente capítulo las decisiones tomadas en cuanto al diseño que seguirá el proyecto, así como la arquitectura del sistema.

Esta fase de diseño es el centro del núcleo técnico de la ingeniería del software. Se llevará a cabo un diseño de datos, un diseño de la arquitectura y un diseño de los componentes del sistema.

### 4.1 Arquitectura del sistema

#### 4.1.1 Decisiones tomadas

El objetivo principal es desarrollar una aplicación móvil que soporte las necesidades que han sido presentadas en los capítulos anteriores y un programa que facilite la gestión de los Targets y sus botones virtuales. Para este servicio buscaremos la mayor flexibilidad, robustez y escalabilidad posible durante su construcción y que tenga la mayor sencillez posible para el usuario adaptándose a sistemas que el usuario ya conozca o que debería conocer.

El primer paso es determinar los elementos sobre los que dicho servicio debe correr, es decir ¿qué lenguajes de programación serán compatibles? ¿qué tipo de servidor se debe utilizar? ¿mediante qué servicio realizar las transmisiones de datos?, etc.

La aplicación móvil se desarrollará utilizando Java y el SDK de Vuforia, este último no ha sido visto hasta el desarrollo del proyecto, pero la gran cantidad de documentación facilitará su uso.

La aplicación móvil tendrá gráficamente el estilo por defecto debido a que se están desarrollando estilos para las aplicaciones de fin de grado y sea más fácil aplicarlos cuando estén disponibles al no tener que quitarlos antiguos y poner los nuevos.

En la aplicación móvil la pantalla de reconocimiento de botones estará bloqueada en horizontal, ya que tras pruebas se ha comprobado que se enfoca y se puede visualizar los elementos del laboratorio con mayor comodidad y la información referida a ellos. La pantalla para la descarga de bases de datos será bloqueada en horizontal, ya que al estar mostrando una lista de actualizaciones con unos nombres relativamente cortos, el formato horizontal es el idóneo, además evitamos problemas al girar la pantalla y rehacer el activity perdiendo las referencias a los objetos que se utilizaran después de acabar la descarga en paralelo.

A la hora de almacenar las referencias a la base de datos disponible en el dispositivo para la aplicación móvil, se utilizará `sharedPreferences`, un método que aunque lleva `deprecated` desde la versión 3 de Android no ha sido retirado y por lo tanto nada hace pensar a corto plazo que vaya a ser retirado de la API. Además ya ha sido utilizado anteriormente por lo que la tarea se realizará antes al ser un método conocido y nos podremos concentrar en aquellos desarrollos en los que no se tiene tanto dominio.

El número de botones virtuales será cambiante. Se permitirá al usuario seleccionar la base de datos que quiere utilizar para el reconocimiento de elementos de laboratorio teniendo un listado de bases de datos disponibles para descargar.

## Trabajo Fin de Grado

---

Las bases de datos que se muestren al usuario corresponderán con las disponibles para el idioma que tiene configurado en el dispositivo.

La base de datos de Vuforia generada por el Target Manager está formada por un archivo XML y otro DAT, se ha decidido crear un nuevo fichero que contendrá las descripciones, mensajes de los botones, identificadores, etc., para mantener la estructura de estos archivos que utiliza Vuforia y evitar que algunos de los cambios realizados entren en conflicto con la estructura del documento que valida Vuforia (o que el archivo que valida este XML de Vuforia cambie en un futuro y nuestros datos añadidos provoquen que se generen errores).

Los archivos referidos a una misma base de datos se diferenciarán unos de otros utilizando un identificador común (mismo nombre) y manteniéndose agrupados según el tipo de archivo que se trate en un directorio concreto.

En un principio se estimó que la mejor opción era crear un servidor ftp para realizar las conexiones y descargas de archivos al dispositivo, pero tras hablar con los técnicos del laboratorio estos indicaron que podría haber problemas debido a la redirección de puertos y aconsejaron utilizar un servidor web para la labor.

Como en prácticas anteriores se ha utilizado ya la pila LAMP (Sistema operativo Linux, servidor web Apache, base de datos MySQL y lenguaje de programación PHP) el servidor se adaptará a estas características para obtener un funcionamiento óptimo, solo que en este caso no se utilizará MySQL debido a tener las bases de datos generadas por Vuforia. PHP solo servirá para listar los directorios donde se encuentran los archivos a descargar.

El programa principal del servidor web que permitirá al usuario administrador gestionar los Targets y los botones virtuales de los mismos se desarrollará en Python, ya que ha sido un lenguaje de programación muy utilizado durante estos años y este conocimiento del lenguaje de programación facilitará la tarea en este sentido.

Los elementos a reconocer solo se pueden eliminar cuando se genera los elementos de la base de datos con el Target Manager, el programa en Python será solo para gestión de botones virtuales.

Ninguno de los archivos accesibles desde el exterior al servidor es sensible, por ello no es necesario establecer una conexión segura, por lo cual nos centraremos en los atributos de calidad mencionados anteriormente.

Tanto la parte del servidor como la de la aplicación móvil se realizará empleando el patrón de arquitectura modelo-vista-controlador. Además en la aplicación móvil se llevará a cabo el patrón adaptador a la hora de mostrar al usuario la lista de bases de datos disponibles.

En el programa en Python no es posible separar el modelo vista controlador en distintos ficheros debido a la necesidad de utilizar variables globales y que Python carece de *include*, tan solo es posible realizar *import*. Por lo que mantendremos el principio del modelo vista controlador y tan solo la vista mostrará información al usuario (salvo errores por los que se deba finalizar la ejecución, que el controlador informará de los sucedido y terminará el programa).

## 4.1.2 Visión global

### Estructura de directorios en el servidor

Como se indico anteriormente llevaremos a cabo una separación de los archivos según el tipo de archivo del que se trate utilizando la estructura de directorios siguiente:

1. Directorio *datVuforia*, contendrá los archivos DAT generados por el Target Manager.
2. Directorio *xmlVuforia*, contendrá los archivos XML generados por el Target Manager y editados por el gestor de botones.
3. Directorio *xmlInformacion*, contendrá los archivos XML generador por el gestor de botones y que utilizará la aplicación móvil.

Como se indicó en las restricciones, estos directorios deben existir para que el programa de gestión de botones funcione.

### Despliegue

En el siguiente diagrama de despliegue se puede observar como el cliente se comunica con el servidor con la pequeña aplicación PHP que le indicará las bases de datos disponibles para que pueda descargarlas.

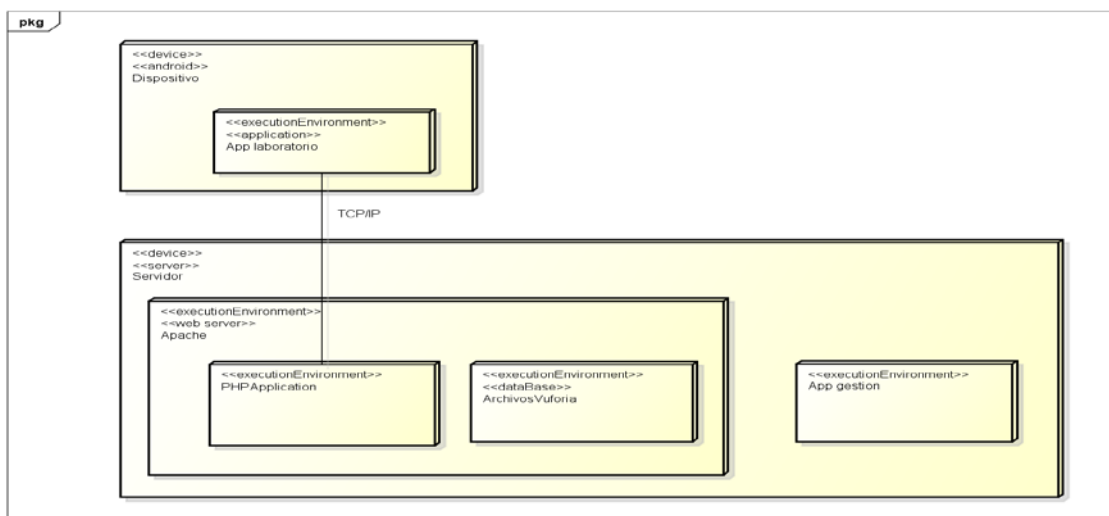


Figura 49: Diagrama de despliegue

## Componentes

Como se puede ver en el diagrama de componentes siguiente, podemos comprender mejor la arquitectura y funcionamiento del sistema, se muestra como la aplicación de gestión depende de los archivos de bases de datos anteriores para generar las nuevas.

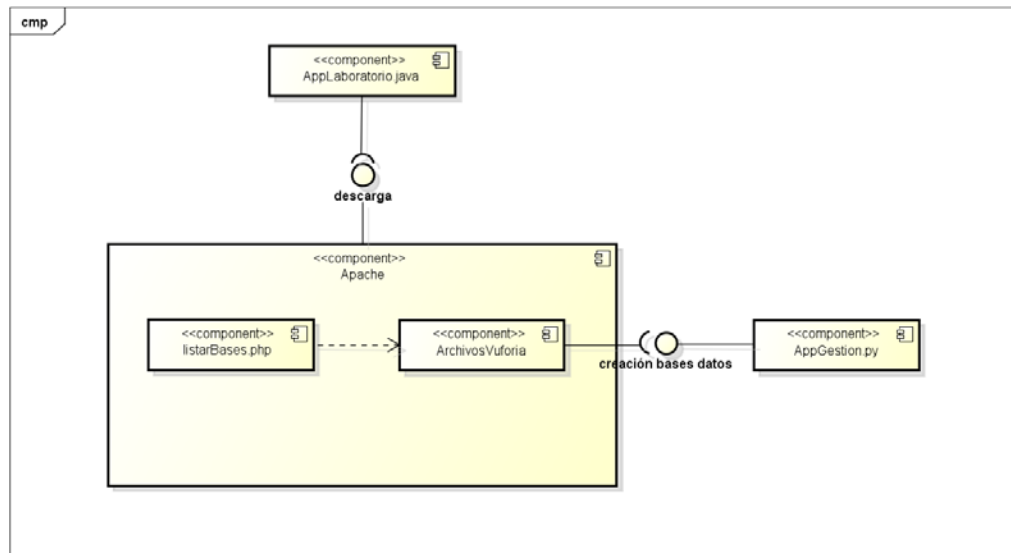


Figura 50: Diagrama de componentes

## Modelo vista controlador y adaptador

Para la comprensión de los diagramas de secuencia posteriores se mostrarán en las siguientes imágenes el diagrama de secuencia del patrón MVC y del patrón adaptador.

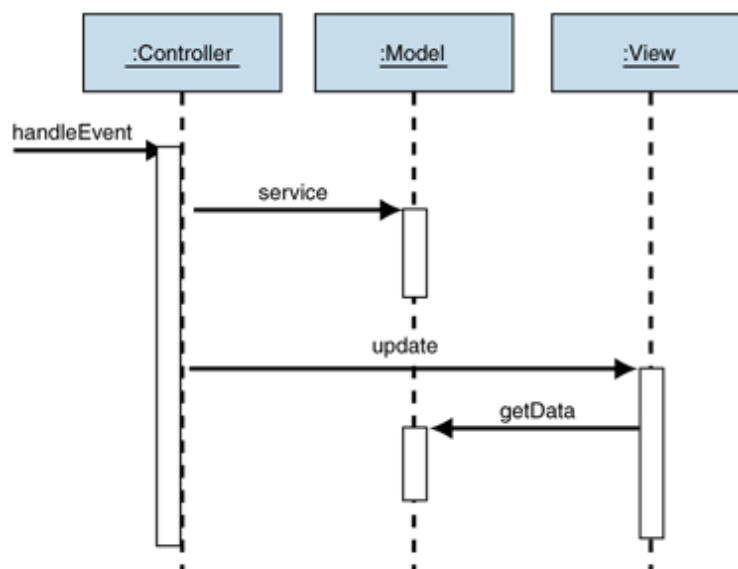


Figura 51: Diagrama modelo vista controlador

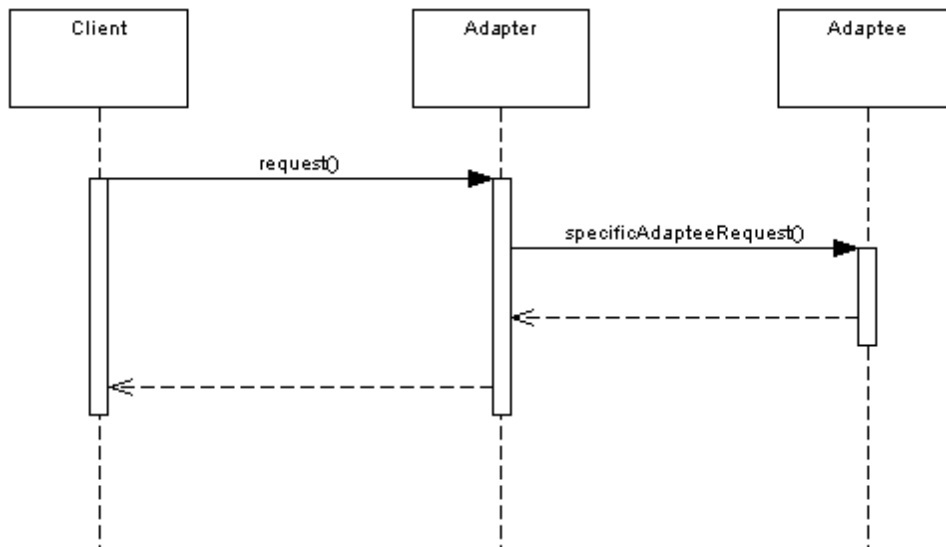


Figura 52: Diagrama adaptador

## 4.1.3 Diseño de la arquitectura

### Descomposición en subsistemas

Tenemos dos subsistemas bien definidos en el proyecto, el subsistema de la aplicación Android, el cual contendrá toda la lógica para reconocer elementos y diseñado en Java y el subsistema del servidor web, cuyo principal propósito es dar soporte a la aplicación Android, también se encuentra la parte del programa de gestión de botones que ayudará al administrador a crear nuevas bases de datos.

Ambos subsistemas se comunican utilizando la interfaz de conexión que se establece entre ambos como puede verse a continuación.

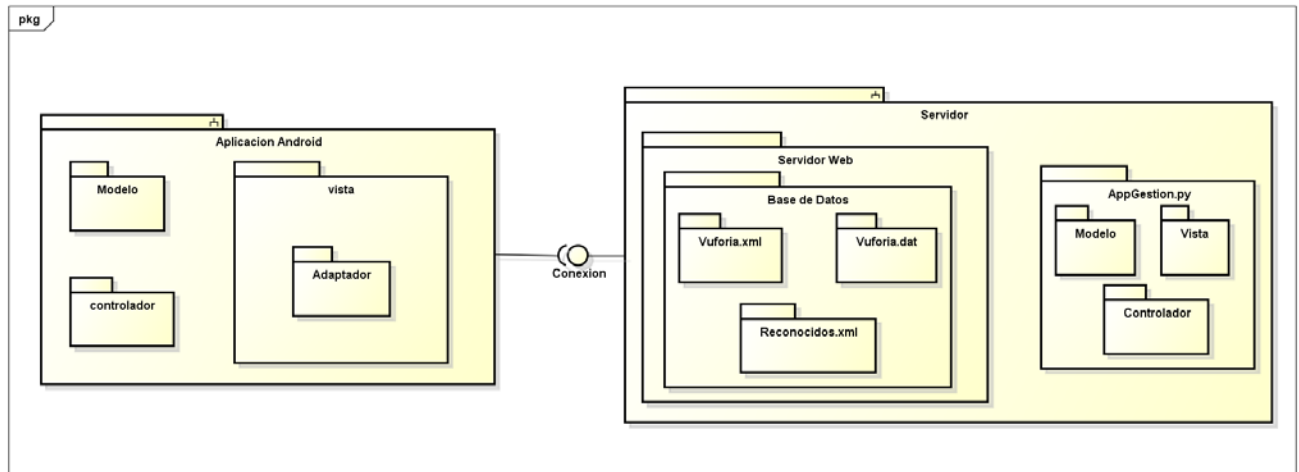


Figura 53: Diagrama de paquetes

## Vista lógica

A continuación presentamos la vista lógica del sistema, donde se muestran los componentes principales del diseño y las relaciones entre los mismos.

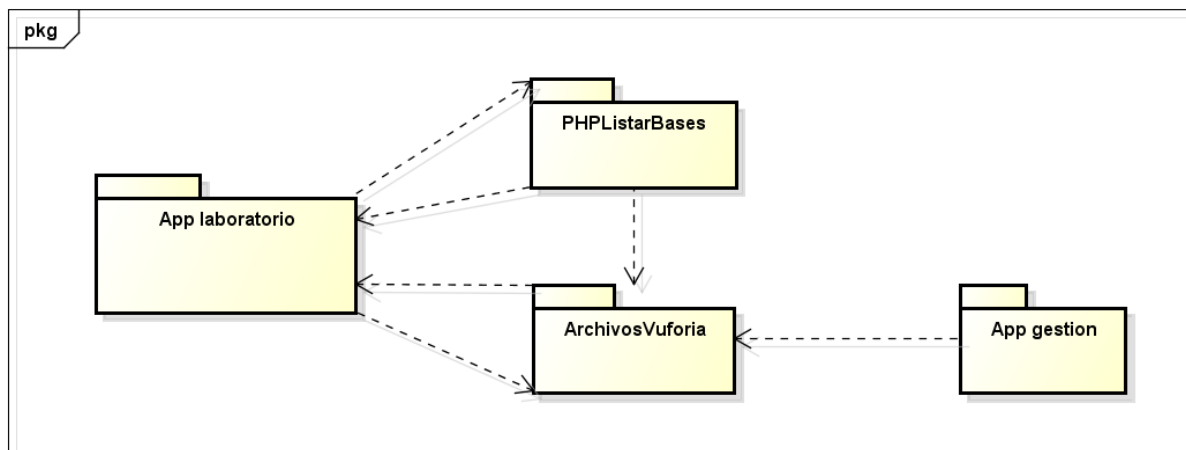


Figura 54: Vista lógica

## 4.2 Casos de uso de diseño

### 4.2.1 Enfocar instrumento del laboratorio

#### Descripción:

Un usuario final desea conocer si hay información en la aplicación sobre un elemento del laboratorio.

#### Actores

- Usuario final.

#### Precondiciones

No hay precondiciones.

#### Flujo normal de eventos

1. El caso de uso comienza cuando el usuario inicia la aplicación y pulsa el botón pulsando el botón “Reconocimiento de imágenes”.
2. El sistema activa la cámara y comienza a capturar imágenes.
3. El usuario dirige la cámara del dispositivo hacia el elemento del laboratorio.
4. El sistema compara las imágenes captadas con la cámara con las de la base de datos.
5. El sistema muestra el nombre de la imagen reconocida en el *textview* situado en la parte superior.
6. El sistema dibuja los botones virtuales referidos a ese elemento.
7. El caso de uso termina.

#### Flujo alternativo

##### *Elemento no encontrado*

Si en el paso 4 el sistema no encuentra ninguna concordancia con la base de datos.

1. El sistema continuará comparando hasta encontrar algo que concuerde con los elementos de la base de datos.

##### *Cancelación*

En cualquier paso del caso de uso el usuario pulsa el botón de volver atrás.

1. El caso de uso termina.

##### *Error, no hay botones asociados a la imagen*

Si en el paso 4 el sistema encuentra una concordancia pero no tiene asociada ningún botón virtual.

1. El sistema muestra la descripción(si existe) de la imagen reconocida.



2. El caso de uso termina.

## Escenarios clave

Un usuario final de la aplicación, Carlos Benedit, se encuentra realizando una práctica de la asignatura de física y debido a su desconocimiento del elemento de laboratorio que tiene delante decide utilizar la aplicación para comprobar qué es y para qué sirven los botones. Enfoca con su móvil el elemento de laboratorio y la aplicación le indica que es la fuente de alimentación y dibuja unos botones virtuales.

## Postcondiciones

### *Finalización satisfactoria*

El usuario obtiene información sobre el elemento del laboratorio.

### *Finalización por cancelación*

El usuario vuelve a la pantalla inicial de la aplicación.

### *Finalización con errores*

El usuario no obtiene ninguna información sobre el elemento del laboratorio.

## 4.2.2 Pulsar botón virtual

### Descripción:

Un usuario final desea conocer la función de cierto botón de un elemento del laboratorio.

### Actores

- Usuario final.

### Precondiciones

- El usuario ha enfocado el elemento del laboratorio.
- El elemento del laboratorio se encuentra en la base de datos del dispositivo.
- El sistema ha reconocido el elemento.
- El elemento del laboratorio tiene algún botón virtual.

### Flujo normal de eventos

1. El caso de uso comienza cuando el usuario pulsa alguno de los botones virtuales que ha dibujado la aplicación.
2. El sistema accede a la descripción de la funcionalidad de ese botón.
3. El sistema muestra la información al usuario en el textview situado en la parte superior..
4. El caso de uso termina.

## Escenarios clave

Uno de los usuarios finales, Víctor Valdecantos, ha enfocado un elemento de laboratorio para conocer sus funcionalidades, la aplicación lo ha reconocido y ha dibujado unos botones virtuales, Víctor pulsa uno de esos botones y la aplicación le devuelve la descripción de la utilidad de ese botón.

## Postcondiciones

### *Finalización satisfactoria*

El usuario obtiene información sobre el botón de un elemento del laboratorio.

## 4.2.3 Descargar base de datos

### Descripción:

El usuario final desea descargar una nueva base de datos para de esta manera reconocer más elementos del laboratorio..

### Actores

- Usuario final.

### Precondiciones

- Base de datos nueva disponible para descargar

### Flujo normal de eventos

1. El caso de uso comienza cuando el usuario pulsa el botón de “Actualizar base de datos”.
2. El sistema comprueba que hay conexión a internet.
3. El sistema conecta con el archivo PHP del servidor que lista un directorio y descarga esa lista de bases de datos disponibles.
4. El sistema le muestra la lista de las bases de datos disponibles ordenadas por fecha.
5. El usuario selecciona una base de datos pulsando sobre ella (listitem).
6. El sistema conecta con el servidor y descarga los tres archivos correspondientes a esa base de datos.
7. El sistema sobrescribe los tres archivos antiguos con los nuevos.
8. El sistema vuelve a la pantalla inicial
9. El caso de uso termina.

### Flujo alternativo

#### *Cancelación de descarga*

Si en cualquier paso el usuario cancela la descarga de la base de datos.

1. El sistema cancela la descarga.
2. El sistema vuelve a la pantalla inicial.
3. El caso de uso termina.

### *Activar internet*

Si en el paso 2 el sistema detecta que no hay conexión a internet.

1. El sistema abre la configuración de internet.
2. El usuario activa internet.
3. El caso de uso continua en el paso 3.

### **Escenarios clave**

Uno de los usuarios finales, Javier Fernández, va a visualizar las actualizaciones disponibles, tras comprobar que puede descargar una base de datos más nueva, selecciona esa base de datos y comienza la descarga. Los archivos se actualizan y puede reconocer los elementos que hay en la nueva base de datos.

### **Postcondiciones**

#### *Finalización satisfactoria*

Los archivos correspondientes a la nueva base de datos para reconocer los elementos del laboratorio se han guardado de manera local en el dispositivo.

#### *Finalización por cancelación*

El usuario final continua con la misma base de datos con el mismo número de imágenes a reconocer.

## **4.2.4 Ver información de la aplicación**

### **Descripción:**

Un usuario final desea conocer información referida a la aplicación.

### **Actores**

- Usuario final.

### **Precondiciones**

No hay precondiciones.

### **Flujo normal de eventos**

1. El caso de uso comienza cuando el usuario pulsa "acerca de" en la pantalla inicial.
2. El sistema muestra la información referida a la base de datos actual, la universidad para la que ha sido diseñada y el nombre del desarrollador.
3. El caso de uso termina.

### Escenarios clave

Un usuario final de la aplicación, Juan Mieres, tiene en breves que realizar una práctica de física, antes de acudir a la práctica quiere saber qué versión de base de datos tiene en el dispositivo. Va a información y ve la versión y otros datos.

### Postcondiciones

#### *Finalización satisfactoria*

El usuario obtiene información sobre la aplicación.

#### *Finalización por cancelación*

El usuario vuelve atrás.

## 4.2.5 Añadir un botón virtual

### Descripción:

Un usuario administrador desea añadir un botón virtual a un elemento del laboratorio.

### Actores

- Usuario administrador.

### Precondiciones

- Se ha seleccionado un elemento.
- Se ha establecido el tamaño original de la imagen.
- Se ha seleccionado el elemento al cual se añadirá información.

### Flujo normal de eventos

1. El caso de uso comienza cuando el usuario indica que desea añadir botón introduciendo por el terminal el carácter "A" o "a".
2. El sistema pide los datos(nombre, coordenadas, enabled y mensaje) referidos al nuevo botón al usuario.
3. El usuario introduce los datos(nombre, coordenadas, enabled y mensaje) del nuevo botón.
4. El sistema valida los datos, comprobando que ningún valor es nulo, que el nuevo botón no colisiona con los antiguos y que enabled es "true" o "false".

5. El sistema almacena los datos referidos al nuevo botón en el array que contiene todos los registros de la base de datos.
6. El caso de uso termina.

### Flujo alternativo

#### *Datos inválidos*

Si en el paso 4 el sistema determina que los valores no son válidos.

1. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 2.

#### *Cancelación*

En cualquiera de los pasos el usuario indica que quiere cancelar el caso de uso.

1. El caso de uso finaliza

### Escenarios clave

Un usuario administrador, Manuel González, se encuentra añadiendo información sobre un elemento del laboratorio, decide crear un botón virtual para explicar la funcionalidad de un botón o un grupo de botones reales. Introduce su nombre, las coordenadas, el estado de enabled y el mensaje que se mostrará.

### Postcondiciones

#### *Finalización satisfactoria*

Los cambios quedan reflejados en el sistema almacenados en el array que contiene los valores de la base de datos.

#### *Finalización por cancelación*

Los cambios no se llevan a cabo y la información del sistema no se ve alterada.

## 4.2.6 Editar un botón virtual

### Descripción:

Un usuario administrador desea editar un botón virtual de un elemento del laboratorio.

### Actores

- Usuario administrador.

### Precondiciones

- Se ha establecido el tamaño original de la imagen.
- Se ha seleccionado el elemento al cual se modificará la información.

- El elemento seleccionado posee al menos un botón virtual.

### **Flujo normal de eventos**

- El caso de uso comienza cuando el usuario selecciona editar botón introduciendo en el terminal el carácter “E” o “e”.
- El sistema pide al usuario por pantalla que indique el botón que quiere editar.
- El usuario selecciona entre los botones existentes el que desea editar introduciendo su identificador por el terminal.
- El sistema pide los datos(nombre, coordenadas, enabled y mensaje) referidos al botón al usuario.
- El usuario introduce los nuevos datos(nombre, coordenadas, enabled y mensaje) del botón.
- El sistema valida los datos, comprobando que las nuevas coordenadas no colisiona con los otros botones, que enabled es “true” o “false” y en caso de un valor nulo de entrada recupera el valor antiguo para ese campo concreto.
- El sistema sobrescribe los datos referidos al botón en el array que contiene todos los registros de la base de datos..
- El caso de uso termina.

### **Flujo alternativo**

#### *Datos inválidos*

Si en el paso 6 el sistema determina que los valores no son válidos.

1. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 4.

#### *Cancelación*

En cualquiera de los pasos el usuario indica que quiere cancelar el caso de uso.

1. El caso de uso finaliza.

### **Escenarios clave**

Un usuario administrador, Manuel González, se encuentra modificando la información sobre los botones de un elementos del laboratorio, decide alterar un botón virtual. Introduce su nuevo nombre, las nuevas coordenadas, el nuevo estado de enabled y el nuevo mensaje que se mostrará.

### **Postcondiciones**

#### *Finalización satisfactoria*

Los cambios quedan reflejados en el sistema, almacenados en el array que contiene los valores de la base de datos.

#### *Finalización por cancelación*

Los cambios no se llevan a cabo y la información del sistema no se ve alterada.

## 4.2.7 Eliminar botón virtual

### Descripción:

Un usuario administrador desea eliminar un botón virtual de un elemento del laboratorio.

### Actores

- Usuario administrador.

### Precondiciones

- Se ha establecido el tamaño original de la imagen.
- Se ha seleccionado el elemento al cual se modificará la información.
- El elemento seleccionado posee al menos un botón virtual.

### Flujo normal de eventos

1. El caso de uso comienza cuando el usuario selecciona borrar botón introduciendo en el terminal el carácter “B” o “b”.
2. El sistema pide al usuario por pantalla que indique el botón que quiere eliminar.
3. El usuario selecciona entre los botones existentes el que desea eliminar introduciendo su identificador por el terminal.
4. El sistema valida el botón.
5. El sistema elimina el botón del array que contiene los valores de la base de datos.
6. El caso de uso termina.

### Flujo alternativo

#### *Datos inválidos*

Si en el paso 4 el sistema determina que los valores no son válidos.

1. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 2.

### Escenarios clave

Un usuario administrador, Manuel González, se encuentra modificando la información sobre los botones de un elementos del laboratorio, decide eliminar un botón virtual. Selecciona el botón a eliminar y este es borrado de la información del sistema.

### Postcondiciones

#### *Finalización satisfactoria*

Los cambios quedan reflejados en el sistema, eliminando en el array que contiene los valores de la base de datos los datos sobre el botón que ha seleccionado el usuario.

## 4.2.8 Establecer tamaño original de la imagen

### Descripción:

Un usuario administrador desea añadir información a un elemento de la base de datos por primera vez, por lo cual tendrá que indicar el tamaño original de la imagen.

### Actores

- Usuario administrador.

### Precondiciones

- Primera vez que se selecciona el elemento de la base de datos.
- No tiene asociado ningún botón virtual.
- No se ha definido su tamaño original.

### Flujo normal de eventos

1. El caso de uso comienza cuando el usuario selecciona un elemento de la base de datos introduciendo su identificador en el terminal.
2. El sistema comprueba el array que contiene los valores de la base de datos y verifica que falta el valor de tamaño original para ese elemento seleccionado.
3. El sistema solicita por pantalla al usuario el tamaño original de la imagen.
4. El usuario introduce por el terminal el valor del tamaño original.
5. El sistema valida los datos, para evitar que se establezca un valor nulo.
6. El sistema almacena el valor en el array que contiene los valores de la base de datos.
7. El caso de uso termina.

### Flujo alternativo

#### *Datos inválidos*

Si en el paso 5 el sistema determina que los valores no son válidos.

1. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 3.

#### *Cancelación*

En cualquiera de los pasos el usuario indica que quiere cancelar el caso de uso.

1. El caso de uso termina

### Escenarios clave

Un usuario administrador, Manuel González, quiere modificar un elemento de la base de datos. Debido a que es la primera vez que edita ese elemento, el sistema le solicita el tamaño original de la imagen, Manuel indica el tamaño original, el sistema almacena el valor y Manuel puede continuar con la edición del elemento.



## Postcondiciones

### *Finalización satisfactoria*

El valor queda almacenado en el sistema, almacenados en el array que contiene los valores de la base de datos.

### *Finalización por cancelación*

La información del sistema no se ve alterada.

## 4.2.9 Modificar tamaño de una imagen

### Descripción:

Un usuario administrador desea modificar el tamaño original de una Target.

### Actores

- Usuario administrador.

### Precondiciones

No hay precondiciones.

### Flujo normal de eventos

1. El caso de uso comienza cuando el usuario selecciona modificar tamaño introduciendo en el terminal el carácter “M” o “m”.
2. El sistema pide al usuario por pantalla que introduzca el elemento que desea modificar.
3. El usuario introduce por el terminal el elemento a modificar.
4. El sistema pide al usuario por pantalla el nuevo tamaño original.
5. El usuario introduce en el terminal el nuevo tamaño original.
6. El sistema valida los datos, para asegurarse que el nuevo tamaño no es menor que las coordenadas de algún botón.
7. El sistema comprueba si el elemento posee algún botón, en caso afirmativo el sistema recalcula los botones con respecto al nuevo tamaño original.
8. El sistema almacena los datos en el array que contiene los valores de la base de datos..
9. El caso de uso termina.

### Flujo alternativo

#### *Datos inválidos*

Si en el paso 6 el sistema determina que los valores no son válidos.

1. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 4.

#### *Cancelación*

En cualquiera de los pasos el usuario indica que quiere cancelar el caso de uso.

1. El caso de uso finaliza

### **Escenarios clave**

Un usuario administrador, Manuel González, desea modificar el tamaño original de un elemento debido a que se equivocó a la hora de establecer este tamaño por primera vez.

### **Postcondiciones**

#### *Finalización satisfactoria*

Los cambios quedan reflejados en el sistema, almacenados en el array que contiene los valores de la base de datos.

#### *Finalización por cancelación*

Los cambios no se llevan a cabo y la información del sistema no se ve alterada.

## **4.2.10 Crear base de datos**

### **Descripción:**

El usuario administrador desea crear una base de datos con elementos del laboratorio a reconocer.

### **Actores**

- Usuario administrador.

### **Precondiciones**

- Se dispone de los dos archivos generados por el Target Manager de Vuforia.
- Se han establecido la información correspondiente a los elementos a reconocer.

### **Flujo normal de eventos**

1. El caso de uso comienza cuando el usuario inicia el gestor de botones pasándole los argumentos.
2. El sistema valida los argumentos.
3. El sistema solicita al usuario el idioma de la última base de datos a cargar.
4. El usuario introduce el idioma (ES español o EN inglés).
5. El sistema valida el idioma.
6. El sistema carga la base de datos anterior y cruza los datos con los nuevos archivos.
7. El usuario realiza los casos de uso de añadir/editar/borrar botones y/o modificar el tamaño original. Y cuando ha terminado de establecer los parámetros de la nueva base de datos indica que desea guardar introduciendo en el terminal el carácter "G" o "g".

8. El sistema valida el carácter de entrada.
9. El sistema solicita al usuario el idioma asociado a la base de datos (español o inglés).
10. El usuario introduce el idioma asociado.
11. El sistema valida el idioma asociado.
12. El sistema crea los nuevos archivos.
13. El caso de uso termina.

## **Flujo alternativo**

### ***Cancelación***

En el paso 7, el usuario decide no guardar los datos introducidos y salir sin guardar.

1. El caso de uso termina.

### ***Datos inválidos 1***

En el paso 2 el sistema comprueba que los argumentos no son válidos.

1. El sistema indicará al usuario que el valor introducido no es válido.
2. El caso de uso termina.

### ***Datos inválidos 2***

En el paso 4 el sistema comprueba que el idioma asociado no es válido.

1. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 3.

### ***Datos inválidos 3***

En el paso 11 el sistema comprueba que el idioma asociado no es válido.

1. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 9.

### ***Base de datos inexistente***

En el paso 6 el sistema comprueba que no hay ninguna base de datos antigua.

1. El sistema utilizará solo los datos de entrada de los argumentos para crear la nueva base de datos.
2. El caso de uso continua en el paso 7.

### ***Base de datos inexistente y argumento vacío***

En el paso 6 el sistema comprueba que no hay ninguna base de datos antigua y no se han recibido ficheros por argumento.

1. El caso de uso termina con error.

### ***Argumentos vacíos***

En el paso 1 el usuario no pasa ningún argumento.

1. El sistema carga la base de datos antigua con el idioma seleccionado y no la cruza con nada
2. El caso de uso continua en el paso 7.

### **Escenarios clave**

Manuel González tras haber añadido la información sobre los elementos de laboratorio y sus botones desea crear una nueva base de datos para ser descargada por los alumnos. Selecciona guardar y se crean los archivos requeridos para descargar por la aplicación.

### **Postcondiciones**

#### *Finalización satisfactoria*

Los nuevos archivos se encuentran disponibles para descargar en su directorio correspondiente.

#### *Finalización por cancelación*

No se crea ningún archivo.

#### *Finalización con errores*

No se crea ningún archivo.

## **4.2.11 Seleccionar elemento**

### **Descripción:**

Un usuario administrador desea añadir información a un elemento de la base de datos, por lo cual procede a seleccionar el elemento.

### **Actores**

- Usuario administrador.

### **Precondiciones**

No hay precondiciones.

### **Flujo normal de eventos**

1. El caso de uso comienza cuando el usuario selecciona un elemento de la base de datos introduciendo su identificador por el terminal.
2. El sistema valida la entrada, comprobando que el valor está dentro de rango.
3. El sistema muestra las opciones que dispone el usuario sobre la edición de los botones de ese elemento.
4. El caso de uso termina.

## Flujo alternativo

### *Datos inválidos*

Si en el paso 2 el sistema determina que los valores no son válidos.

7. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 1.

## Escenarios clave

Un usuario administrador, Manuel González, quiere modificar un elemento de la base de datos. Manuel selecciona el elemento que quiere modificar y se lo introduce por pantalla para proceder a su edición.

## Postcondiciones

### *Finalización satisfactoria*

Se ha seleccionado el elemento con éxito y se puede proceder a su edición.

## 4.2.9 Modificar descripción de una imagen

### Descripción:

Un usuario administrador desea modificar la descripción de una Target.

### Actores

- Usuario administrador.

### Precondiciones

No hay precondiciones.

### Flujo normal de eventos

1. El caso de uso comienza cuando el usuario selecciona modificar descripción introduciendo en el terminal el carácter “D” o “d”.
2. El sistema pide al usuario por pantalla que introduzca el elemento que desea modificar.
3. El usuario introduce por el terminal el elemento a modificar.
4. El sistema pide al usuario por pantalla la nueva descripción.
5. El usuario introduce en el terminal la nueva descripción.
6. El sistema valida los datos.
7. El sistema almacena los datos en el array que contiene los valores de la base de datos..
8. El caso de uso termina.

## Flujo alternativo

## ***Datos inválidos***

Si en el paso 6 el sistema determina que los valores no son válidos.

2. El sistema indicará al usuario que el valor introducido no es válido, se vuelve al paso 4.

## ***Cancelación***

En cualquiera de los pasos el usuario indica que quiere cancelar el caso de uso.

2. El caso de uso finaliza

## **Escenarios clave**

Un usuario administrador, Manuel González, desea modificar la descripción de un elemento debido a que se equivocó a la hora de establecer esta descripción anteriormente. Selecciona el elemento del laboratorio a modificar la descripción e introduce el nuevo valor. La nueva descripción queda almacenada.

## **Postcondiciones**

### ***Finalización satisfactoria***

Los cambios quedan reflejados en el sistema, almacenados en el array que contiene los valores de la base de datos.

### ***Finalización por cancelación***

Los cambios no se llevan a cabo y la información del sistema no se ve alterada.

### **4.3 Diagramas de secuencia**

A continuación se presentan los diagramas de secuencia de diseño.

### 4.3.1 Enfocar instrumento del laboratorio

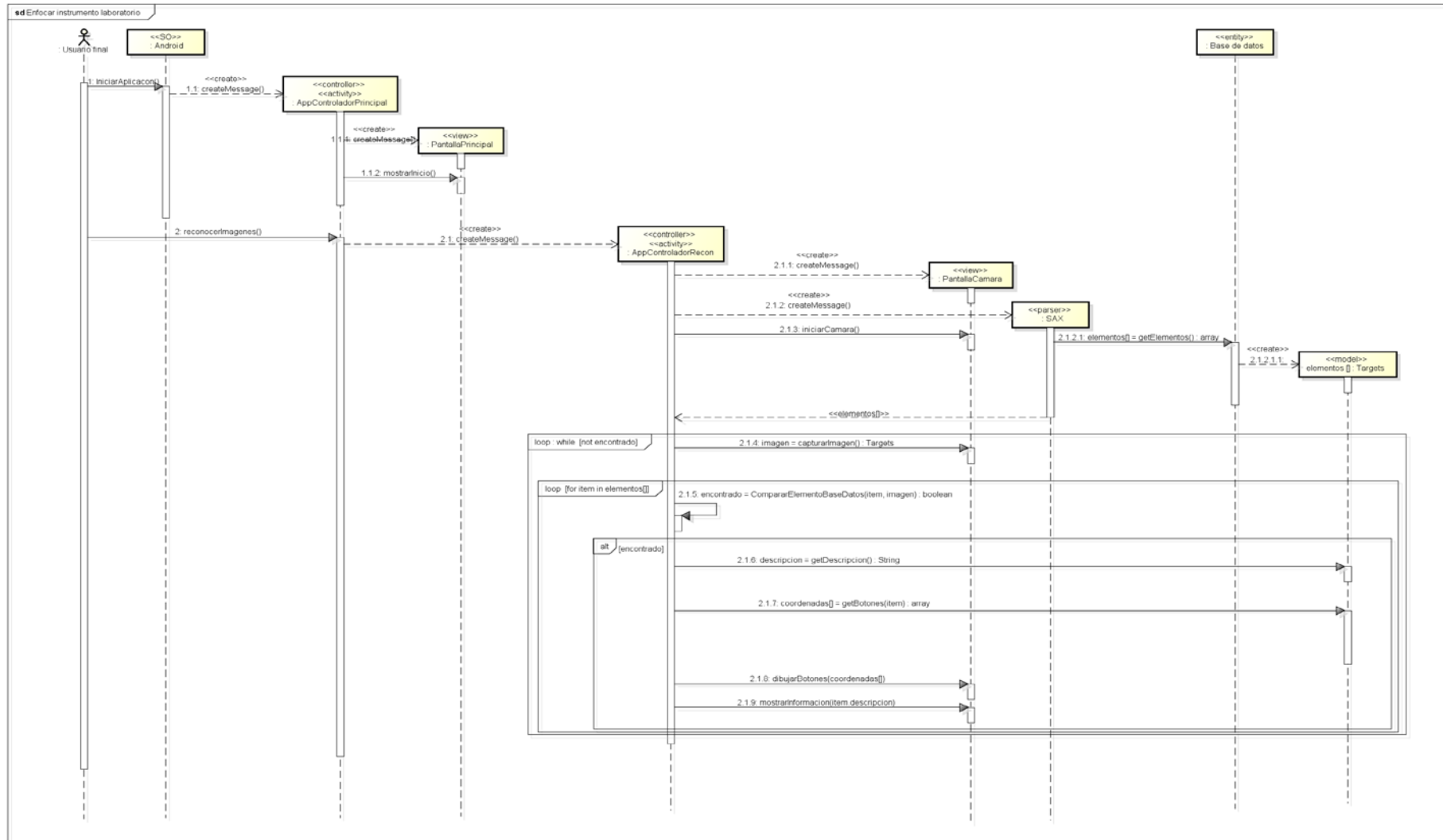


Figura 55: Diagrama de secuencia



### 4.3.2 Pulsar botón

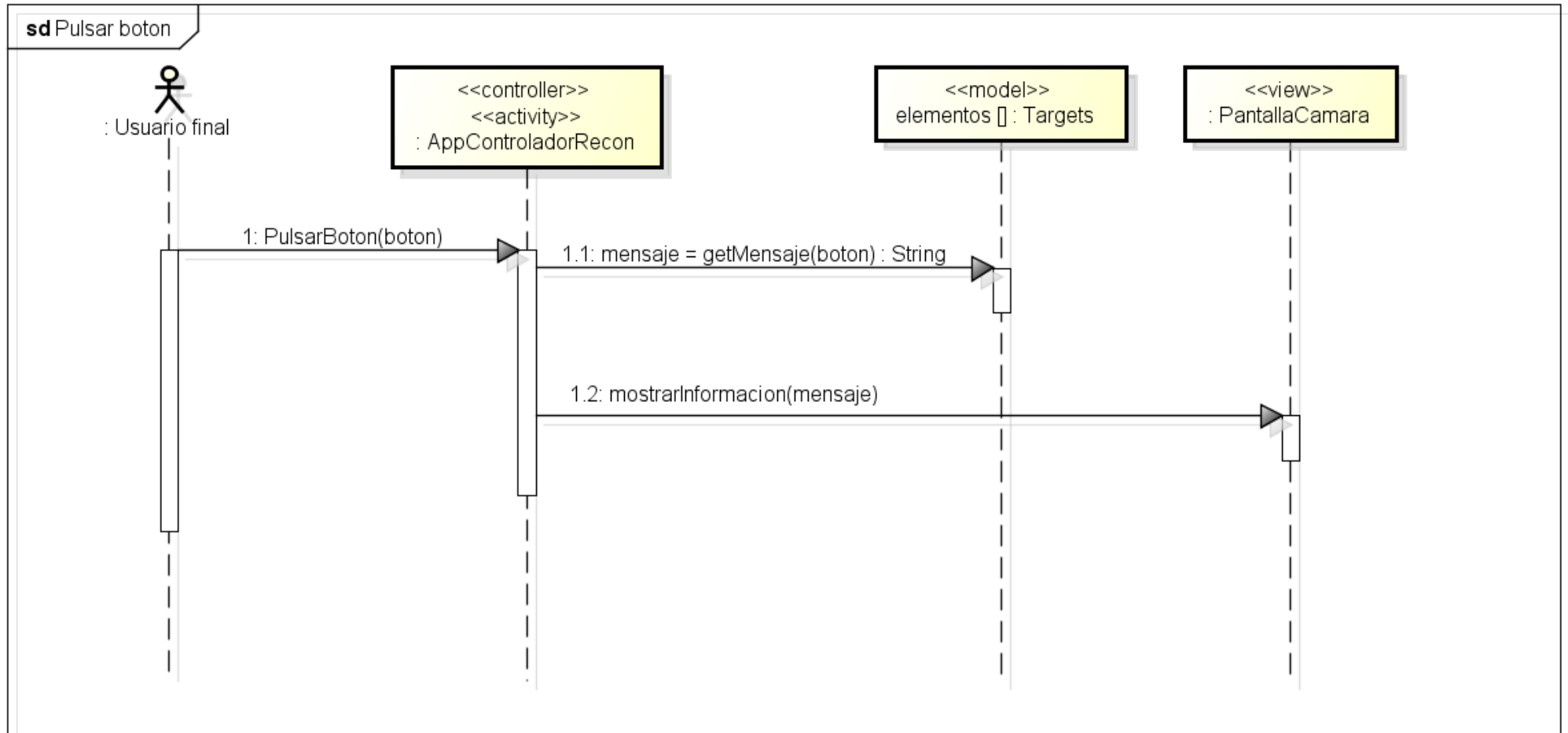


Figura 56: Diagrama de secuencia

## 4.3.3 Descargar base de datos

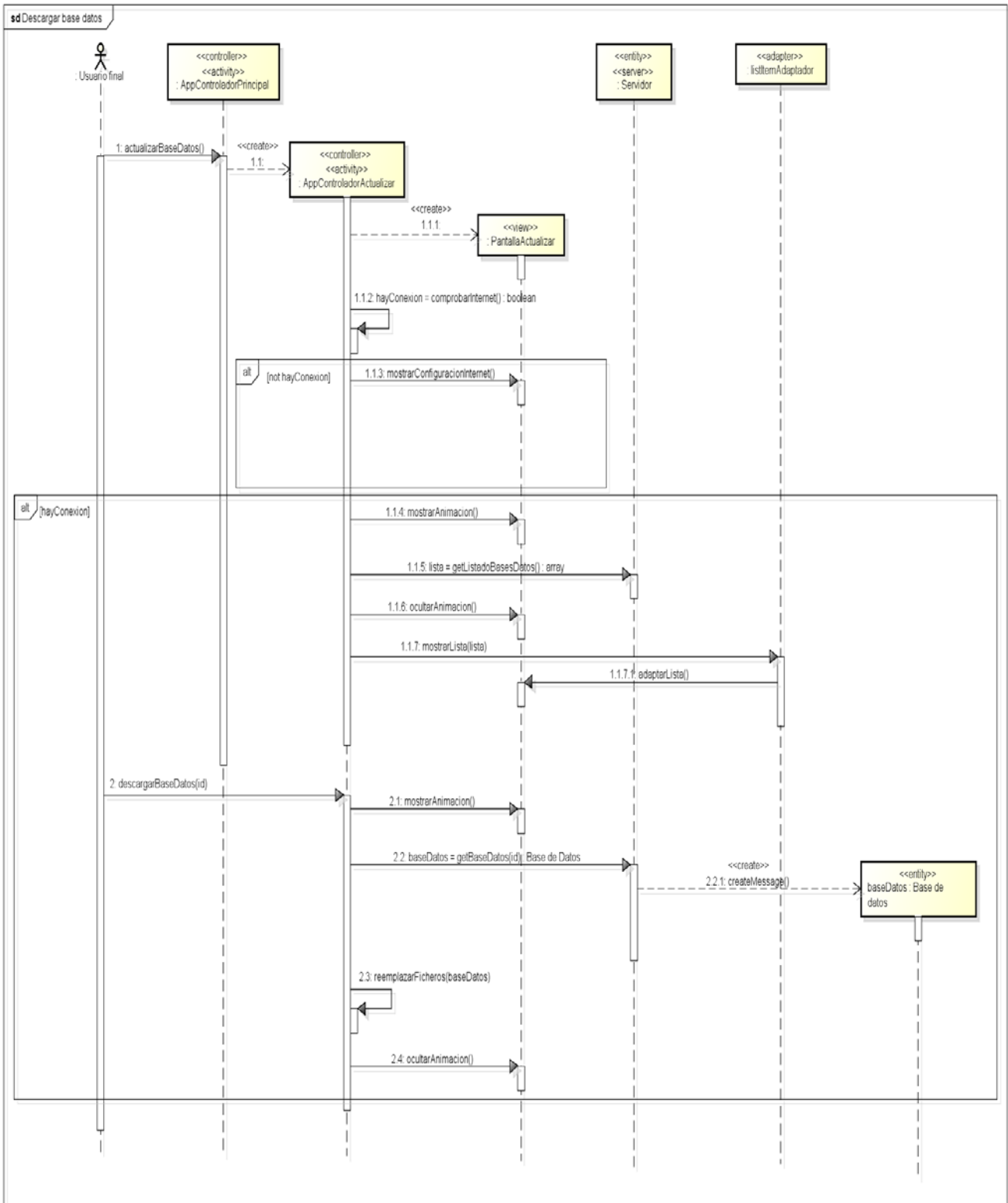


Figura 57: Diagrama de secuencia

## 4.3.4 Crear base de datos

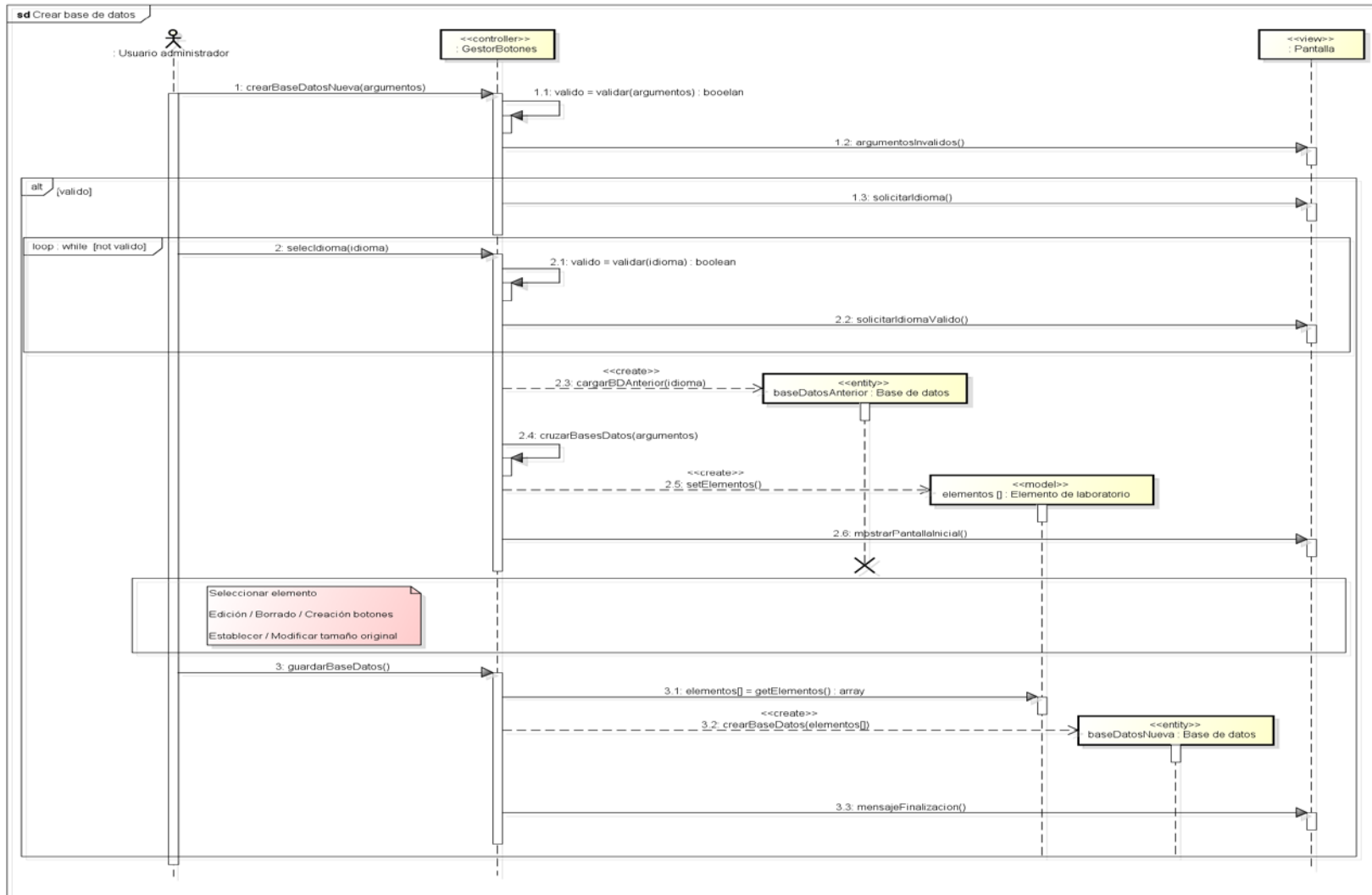


Figura 58: Diagrama de secuencia

## 4.3.5 Opciones botones

Antes de editar, añadir o borrar un botón el usuario debe seleccionar la operación que desea llevar a cabo, la cual se ilustra en este diagrama de secuencia

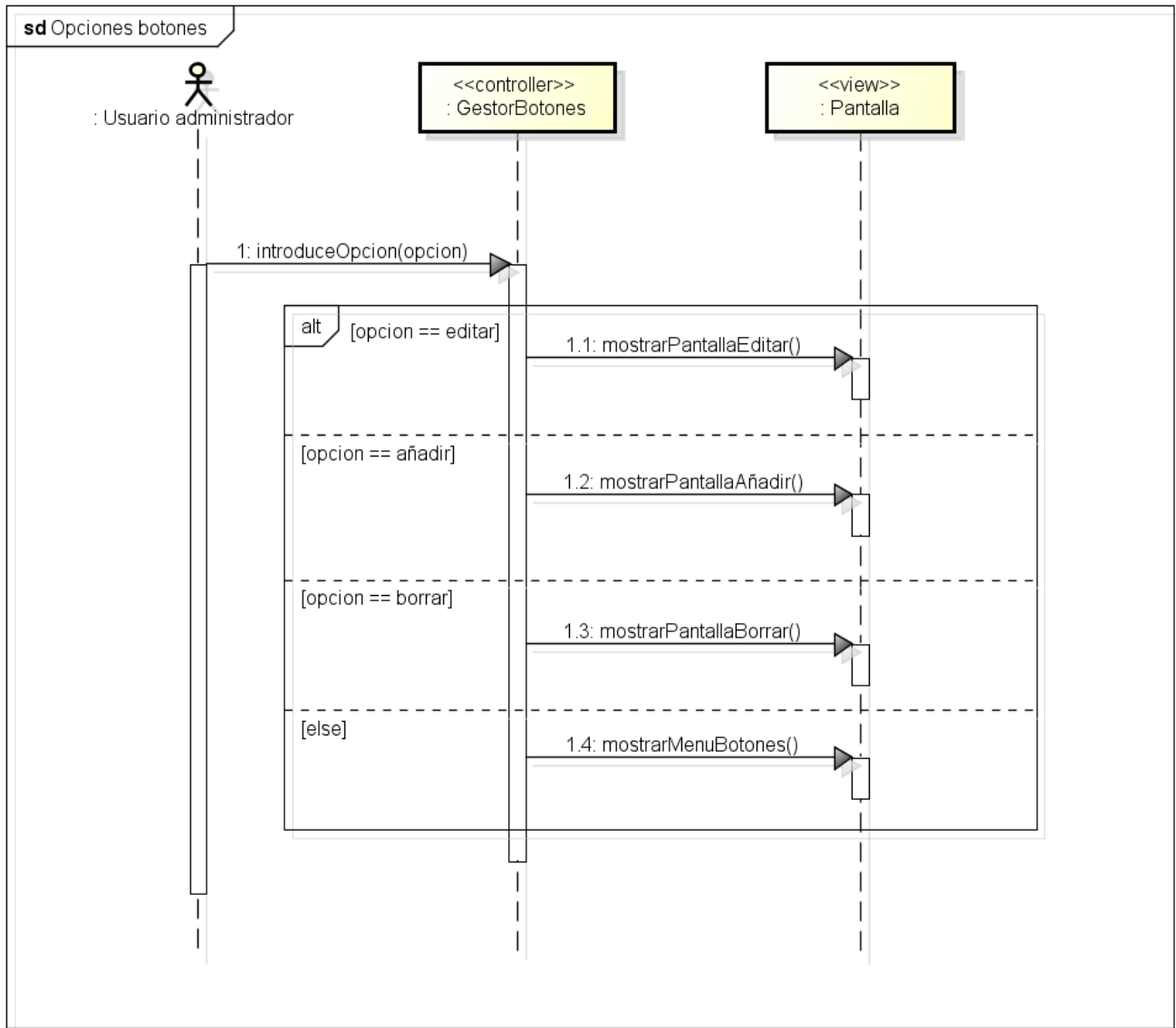


Figura 59: Diagrama de secuencia

## 4.3.6 Añadir botón

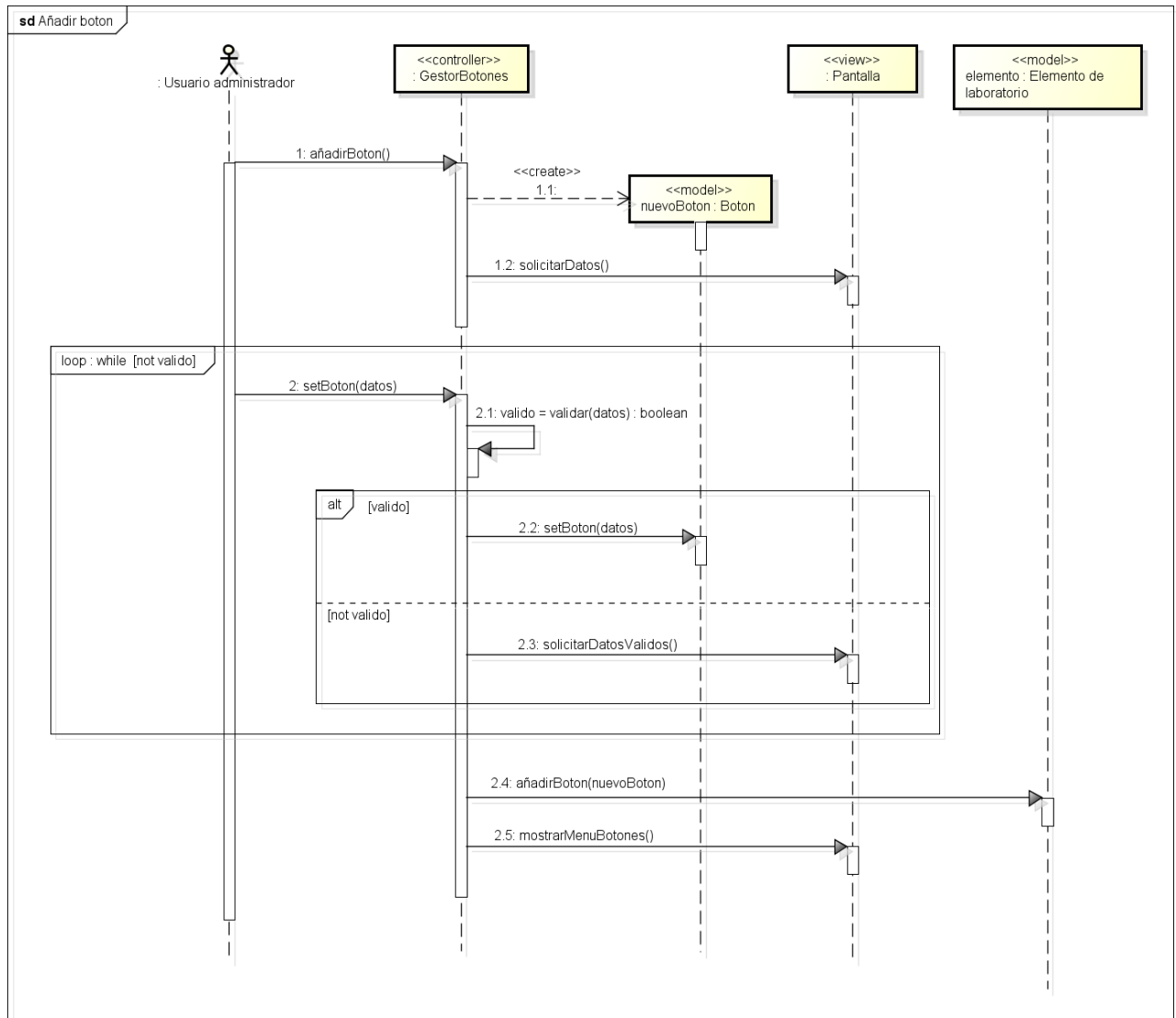
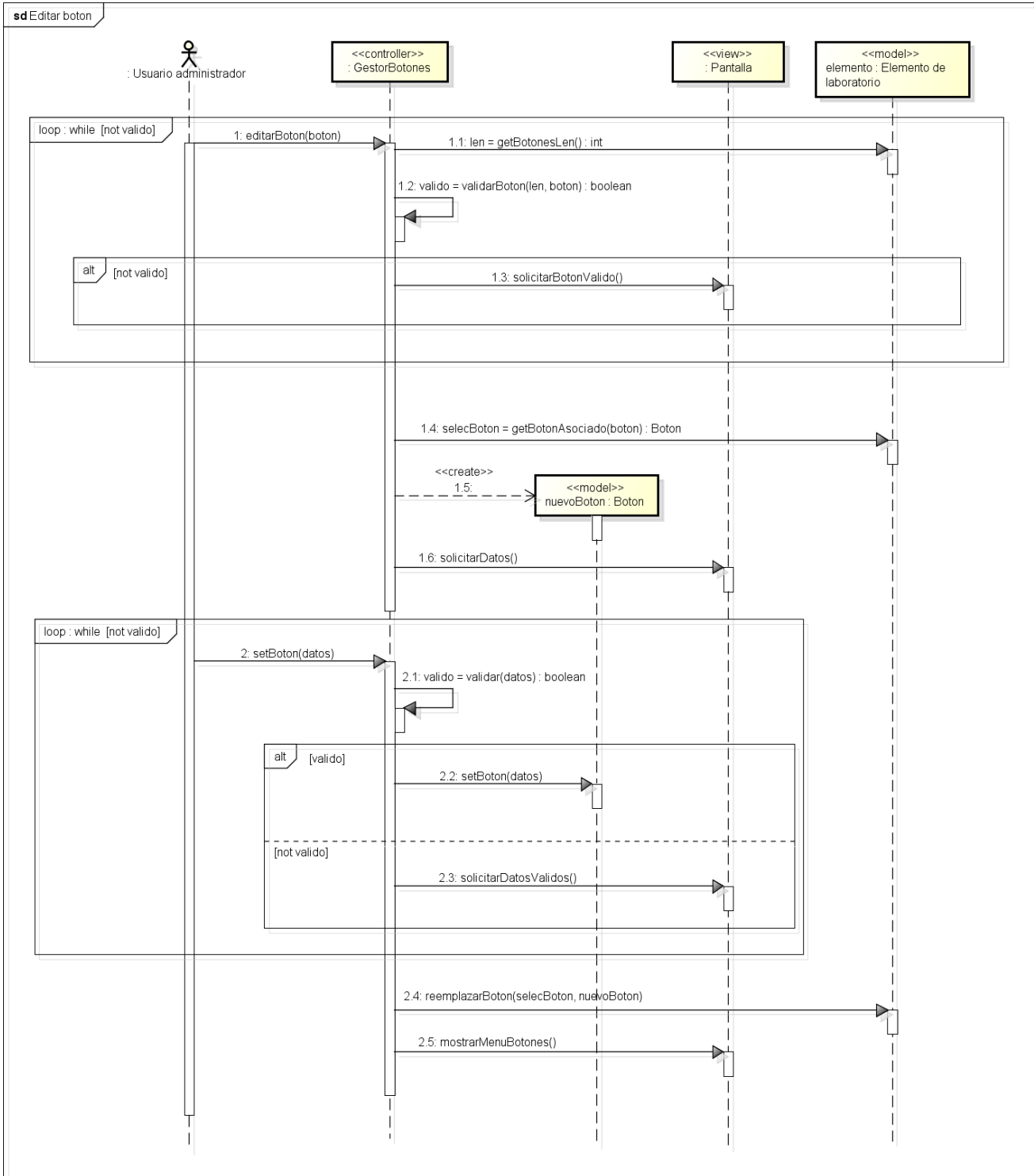


Figura 60: Diagrama de secuencia

## 4.3.7 Editar botón



**Figura 61:** Diagrama de secuencia

## 4.3.8 Borrar botón

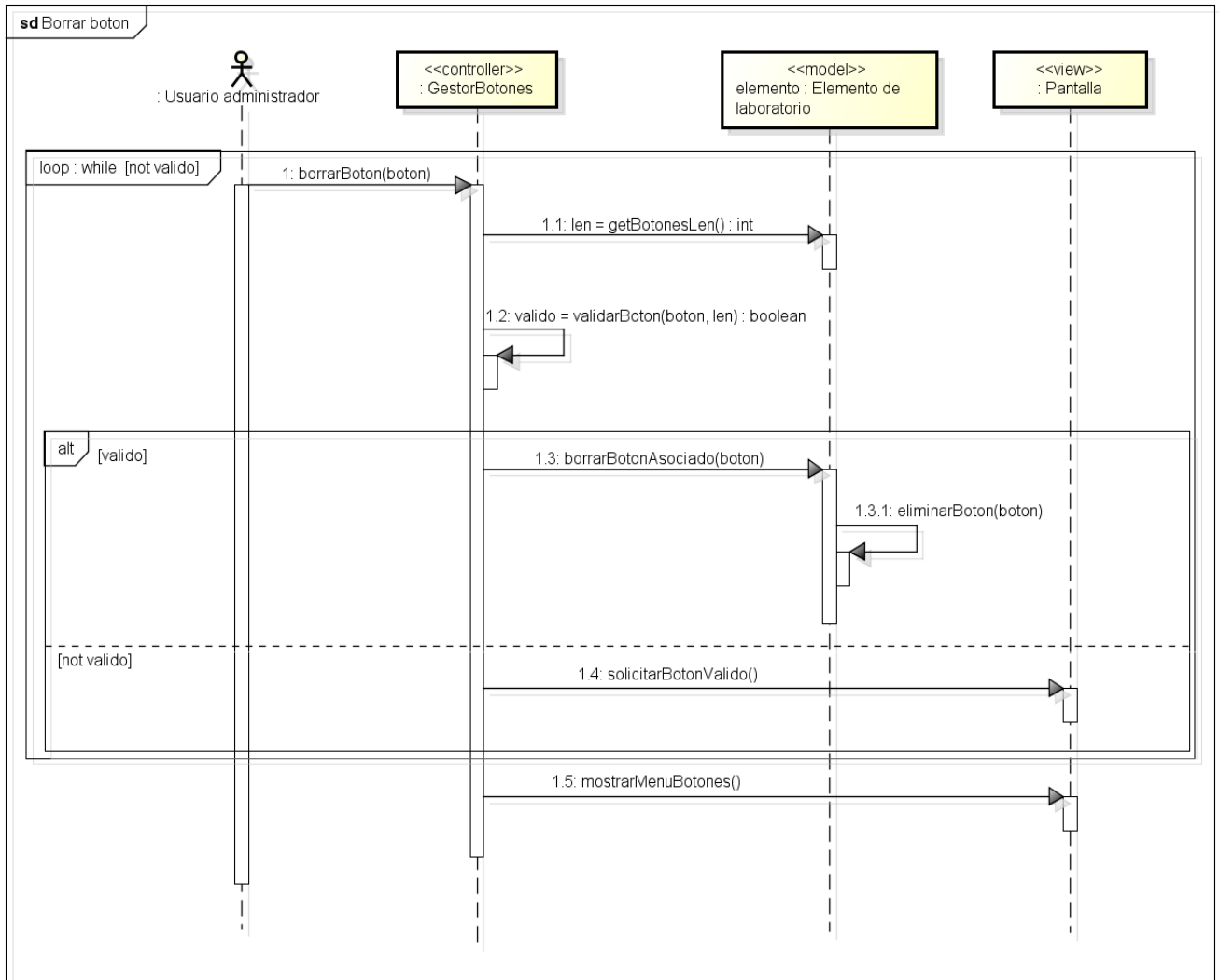


Figura 62: Diagrama de secuencia

## 4.3.9 Seleccionar y establecer tamaño

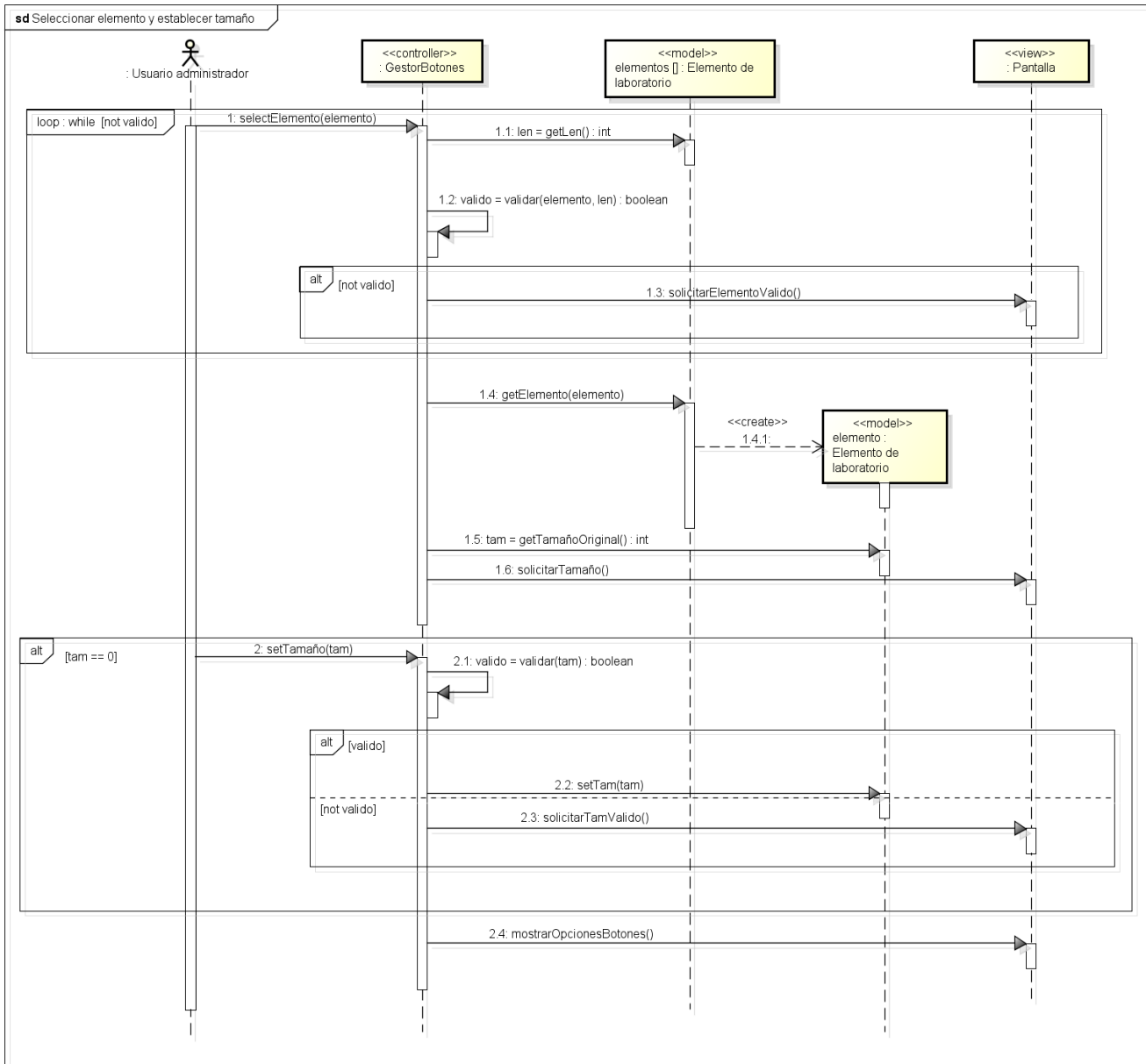


Figura 63: Diagrama de secuencia



## 4.3.10 Modificar tamaño original

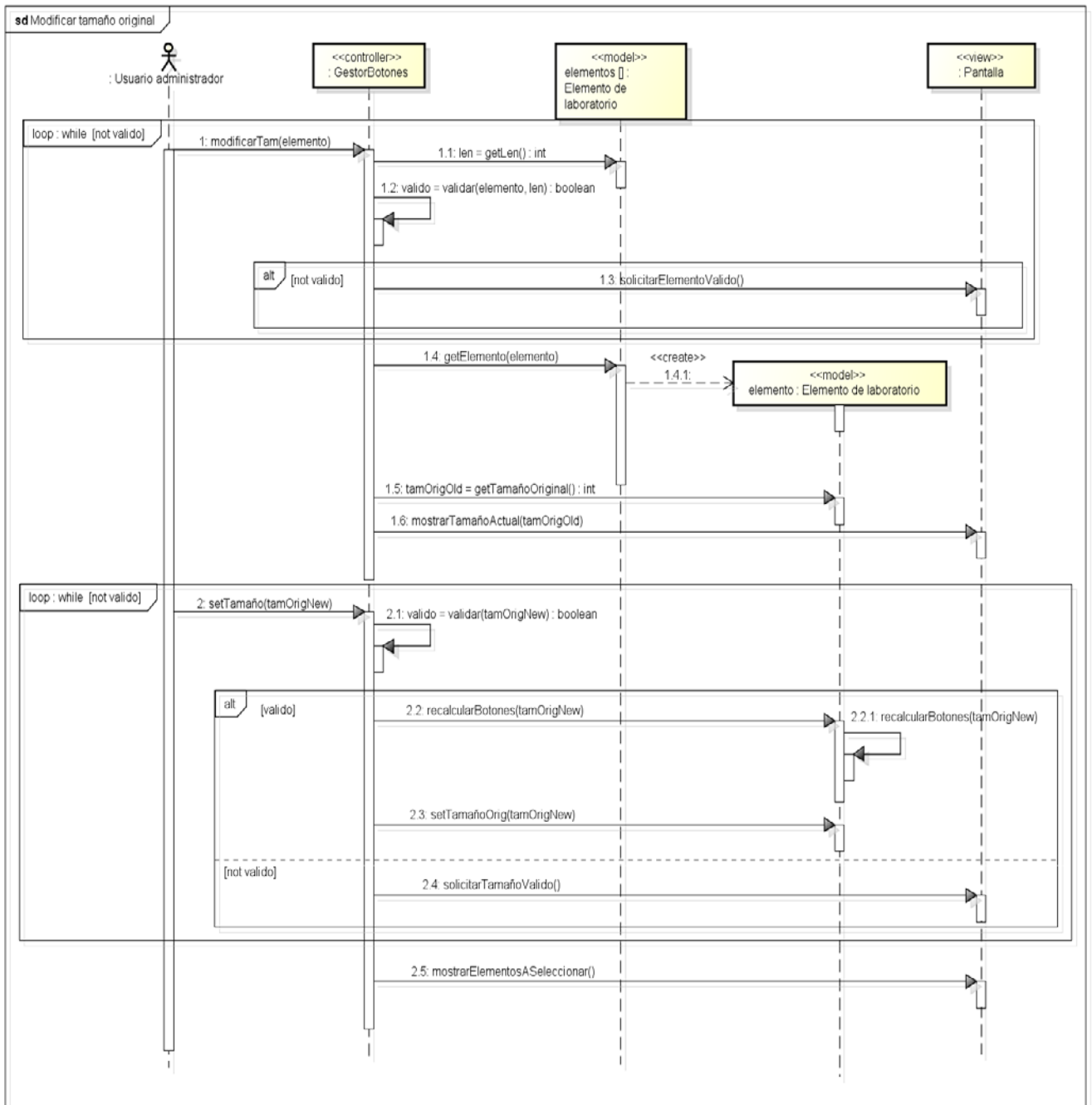


Figura 64: Diagrama de secuencia

### 4.3.11 Ver información de la aplicación

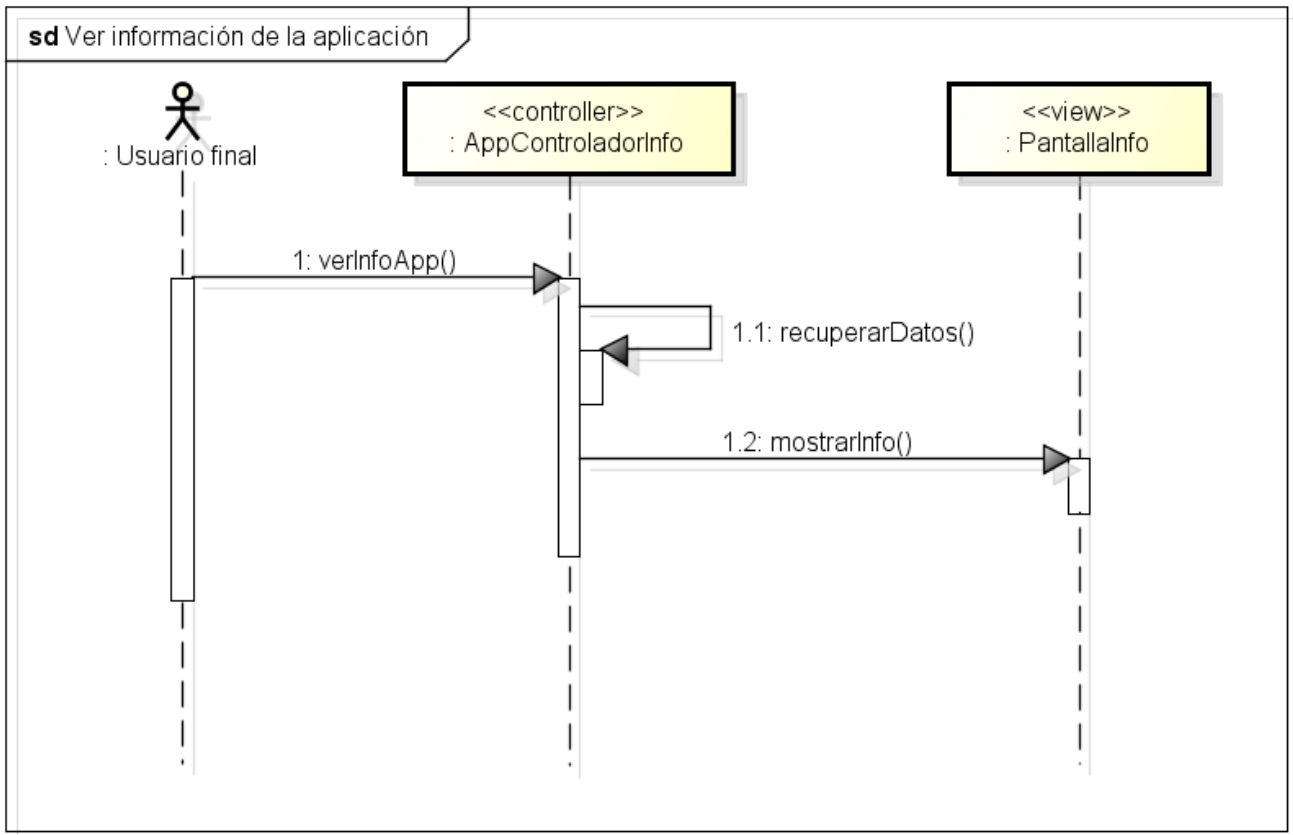


Figura 65: Diagrama de secuencia

## 4.3.12 Modificar descripción

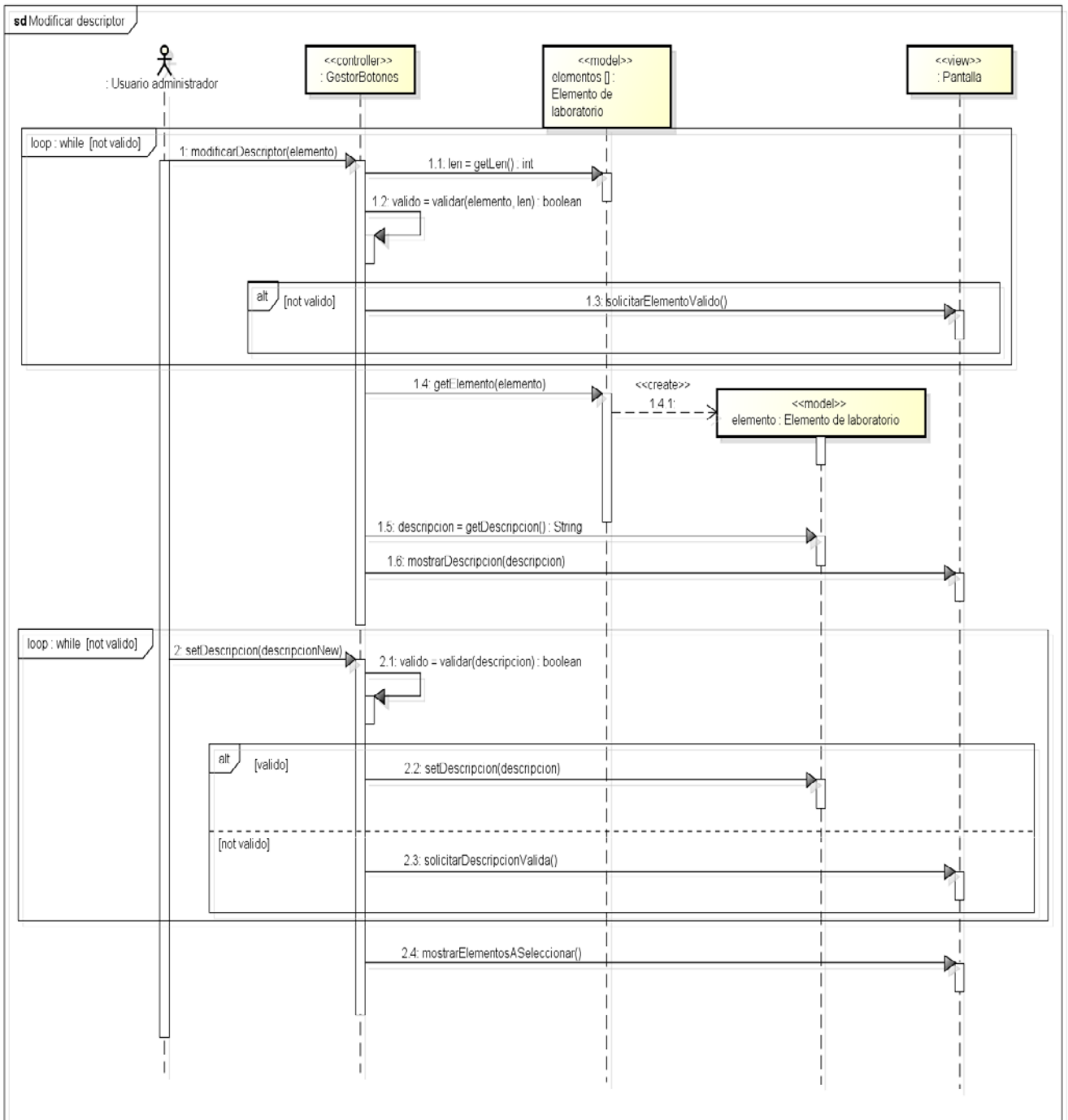


Figura 66: Diagrama de secuencia

### **4.4 Diagrama de clases de diseño**

Como se puede ver a continuación se ha creado un diagrama de clases para cada subsistema para simplificarlo a la hora de mostrarlo y ya que los elementos del modelo de cada subsistema son ligeramente diferentes en estructura aunque compartan un gran número de atributos.

4.4.1 Diagrama de clases de dispositivo

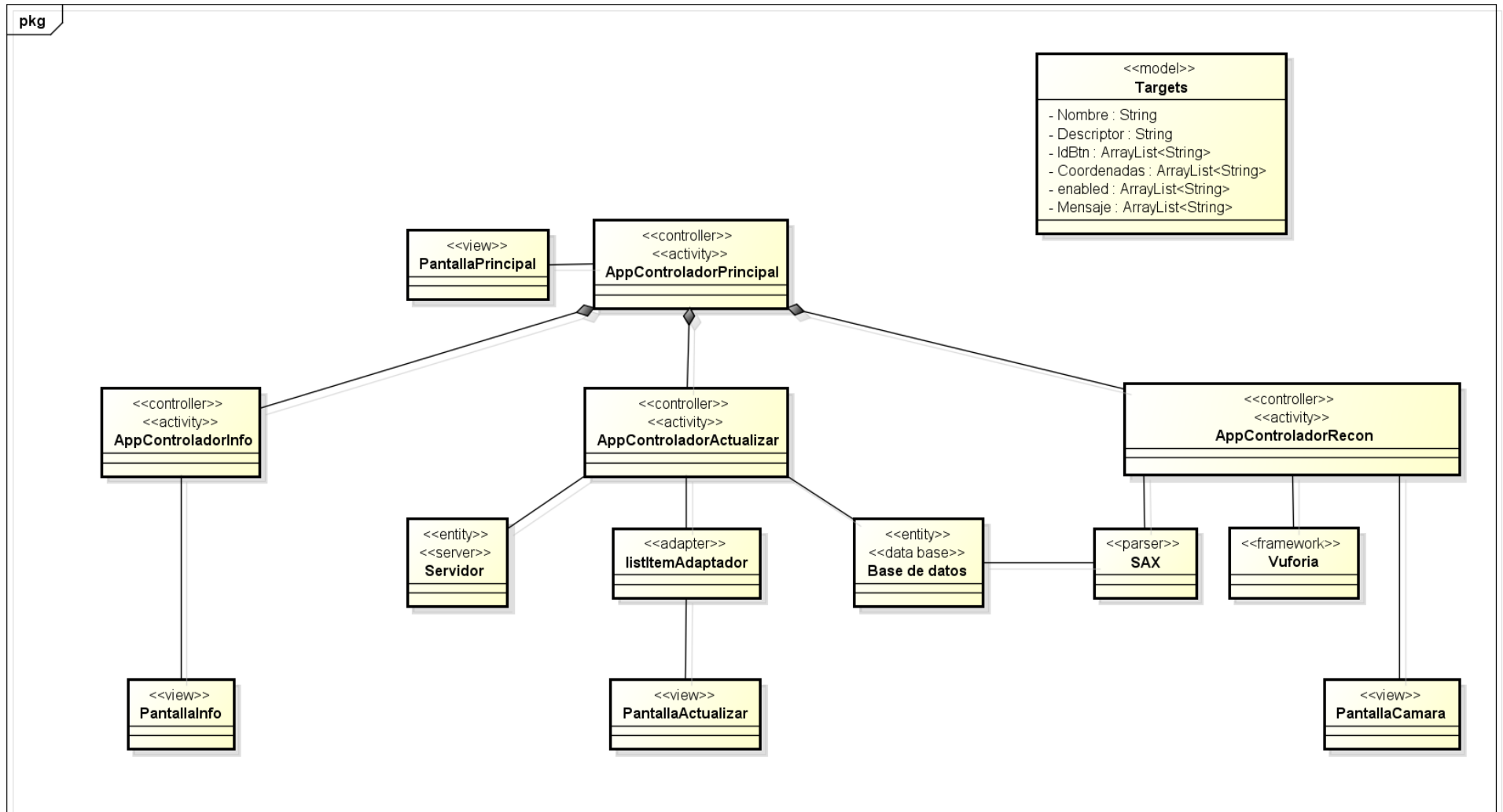


Figura 67: Diagrama de clases

## 4.4.2 Diagrama de clases de servidor

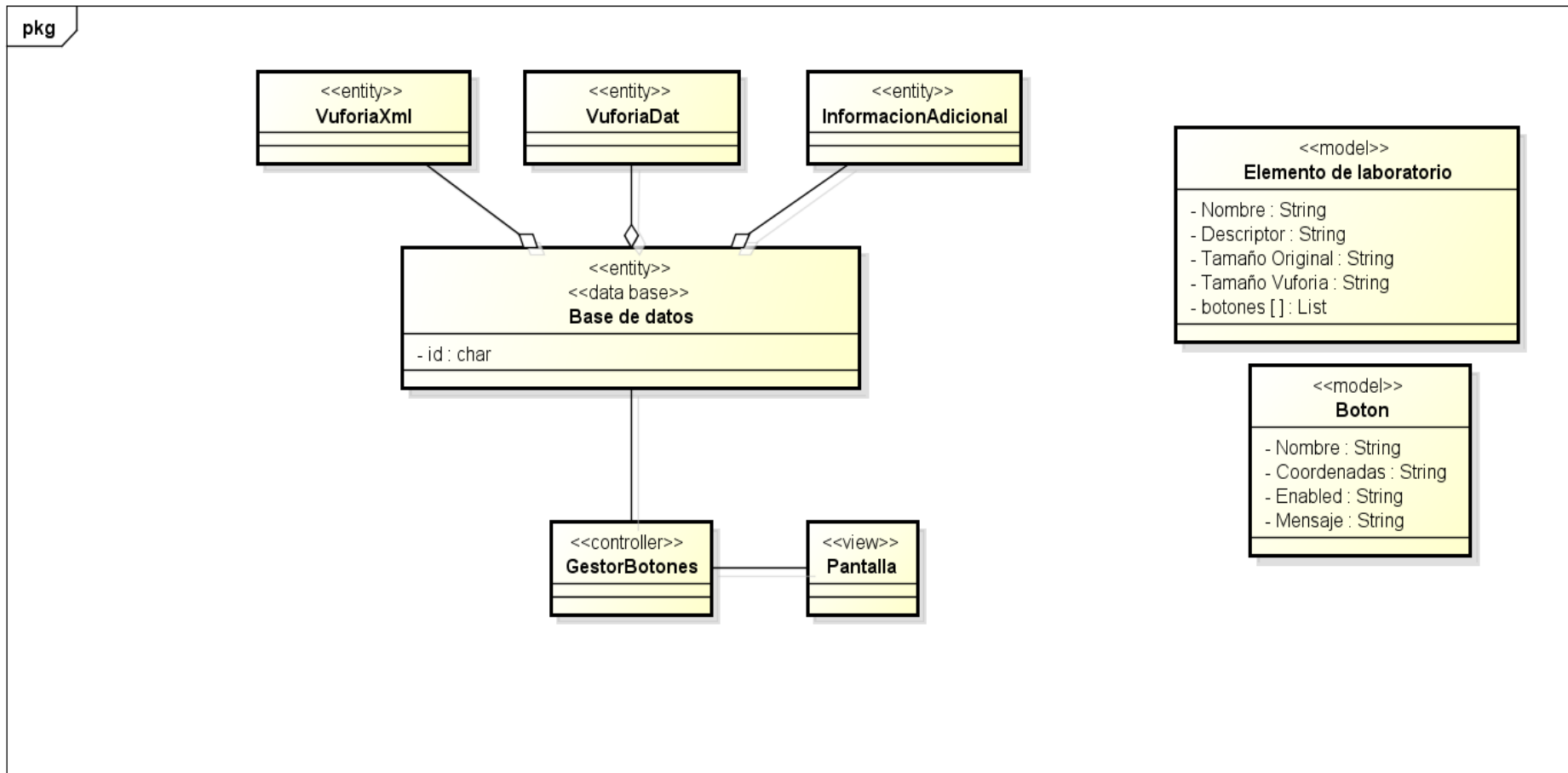


Figura 68: Diagrama de clases

# 5. Implementación

En este capítulo se detallarán algunos de los puntos más relevantes durante la implementación del proyecto, nos centraremos en aspectos más técnicos relativos al Framework de Vuforia que son más complejos, ya que la aplicación está completamente documentada (Javadoc la aplicación Android y Pydoc el programa en Python) no se detallará mucho sobre el código.

## 5.1 Aplicación Android

A la hora de dibujar los botones de un objeto reconocido la primera operación es la comprobación de que Vuforia ha reconocido algún elemento, y tras esto se carga el objeto reconocido como se ve a continuación:

```
// Se comprueba si ha sido encontrado algún elemento por Vuforia
if (state.getNumTrackableResults() > 0)
{
    // Cargamos el objeto reconocido en una clase base para todos los objetos de resultado
    // y su matriz de posición
    TrackableResult trackableResult = state.getTrackableResult(0);
    float[] modelViewMatrix = Tool.convertPose2GLMatrix(
        trackableResult.getPose().getData());
}
```

Figura 69: Código aplicación Android

Tras haber encontrado un elemento, buscamos su homólogo en el ArrayList donde se ha cargado el fichero con las descripciones de los elementos de laboratorio y de sus botones. Y tras encontrarlo lo almacenamos en una variable que será sobre la que se trabajará (se hace de esta manera para tener un código más simple, ya que sería posible guardar solo el índice sobre el ArrayList y trabajar sobre él).

```
//Si hay algun trackable buscamos cual hemos encontrado de los contenidos en el arraylist
//y lo guardamos en target para mayor comodidad a la hora de operar
for(int i=0;i<targetsSize;i++) {
    if(trackableResult.getTrackable().getName().equals(targets.get(i).getTarget())){
        target = targets.get(i);
    }
}
```

Figura 70: Código aplicación Android

Se calcula el índice del botón de la iteración actual para, en caso de haber sido pulsado tener el índice referido al botón.

```
// Recorremos los nombres de los botones asociados al Target
// para encontrar el boton referente a la iteración actual
for (int j = 0; j < target.getIdBtn().size(); ++j)
{
    //Si hay alguna coincidencia se almacena el indice y se termina el bucle
    if (button.getName().compareTo(
        target.getIdBtn().get(j)) == 0)
    {
        buttonIndex = j;
        break;
    }
}
```

Figura 71: Código aplicación Android



Si el botón de la iteración actual se ha pulsado se accede al textview del layout para escribir el mensaje correspondiente a ese botón pulsado utilizando el ArrayList y el índice del botón calculado anteriormente. Además cambiando la variable pulsado a *true* evitamos que intente escribir después nombre del elemento de laboratorio reconocido.

```
// Si el boton ha sido pulsado se muestra el mensaje asociado
if (buttonResult.isPressed())
{
    //Cargamos el mensaje asociado al boton utilizando el indice
    mensaje = target.getMensaje().get(buttonIndex);
    mActivity.runOnUiThread(new Runnable() {

        @Override
        public void run() {
            //Carga del mensaje en el textview
            textView1.setText(mensaje);
        }
    });

    //Evitamos que se escriba el target si se ha pulsado algún boton.
    pulsado = true;
}
```

Figura 72: Código aplicación Android

A la hora de cargar las coordenadas para dibujar los botones se lleva a cabo de manera dinámica para no tener una limitación de botones mínima o máxima por cada elemento de laboratorio.

```
// Recorremos el target encontrado añadiendo los botones virtuales a la variable
// de la clase Rectangle
Rectangle vbRectangle[] = new Rectangle[target.getIdBtn().size()];
int cont = 0;
for(String coor : target.getCoordenadas()){
    vbRectangle[cont] = new Rectangle(Float.parseFloat(coor.split(" ")[0]),
        Float.parseFloat(coor.split(" ")[1]),
        Float.parseFloat(coor.split(" ")[2]),
        Float.parseFloat(coor.split(" ")[3]));
    cont++;
}
```

Figura 73: Código aplicación Android

## 5.2 Programa Python

La siguiente función se encarga de convertir las coordenadas en píxeles que introduce el usuario a las coordenadas que utilizará Vuforia durante el reconocimiento de imágenes y que serán las almacenadas en los ficheros XML.

```
def getCoordenadasTransformadas(self, cadena, sizeAncho, sizeAlto, sizeAnchoOrig, sizeAltoOrig):
    """ Funcion que recibe las coordenadas en pixels, el alto y ancho de Vuforia y el ancho y
        alto original. Devuelve las coordenadas transformadas """
    #Coordenadas en Pixels
    coord = cadena.split()

    #Normalizacion
    u1 = float(coord[0]) / sizeAnchoOrig
    v1 = float(coord[1]) / sizeAltoOrig
    u2 = float(coord[2]) / sizeAnchoOrig
    v2 = float(coord[3]) / sizeAltoOrig

    #Giro imagen verticalmente
    v1G = 1 - v1
    v2G = 1 - v2

    #Reescalar [-1, 1]
    ru1 = -1 + 2 * u1
    rv1 = -1 + 2 * v1G
    ru2 = -1 + 2 * u2
    rv2 = -1 + 2 * v2G

    #Coordenadas en 3D
    x1 = (ru1 * sizeAncho) / 2
    y1 = (rv1 * sizeAlto) / 2
    x2 = (ru2 * sizeAncho) / 2
    y2 = (rv2 * sizeAlto) / 2

    #Coordenadas transformadas
    coordTrans = str(x1) + " " + str(y1) + " " + str(x2) + " " + str(y2)

    return coordTrans
```

**Figura 74:** Código programa Python

Como puede verse a continuación, se lleva a cabo el cálculo de las coordenadas en píxeles a través de las coordenadas transformadas según la fórmula que utiliza Vuforia para dibujar luego los botones sobre los elementos que reconoce.

Durante el proceso de edición de botones se le muestran al usuario las coordenadas en píxeles, para que pueda compararlos con los valores introducidos por si ha cometido algún error. Aunque después en los archivos XML se almacenan las coordenadas transformadas.

```
def getCoordenadasPixels(self, cadena, sizeAncho, sizeAlto, sizeAnchoOrig, sizeAltoOrig):
    """ Funcion que recibe las coordenadas transformadas, el alto y ancho de Vuforia y
        el ancho y alto original. Devuelve las coordenadas en pixels """

    #Coordenadas transformadas en 3D
    coordTrans = cadena.split()

    #Inverso calculo coordenadas en 3D
    ru1 = float(coordTrans[0]) * 2 / sizeAncho
    rv1 = float(coordTrans[1]) * 2 / sizeAlto
    ru2 = float(coordTrans[2]) * 2 / sizeAncho
    rv2 = float(coordTrans[3]) * 2 / sizeAlto

    #Inverso del reescalado [-1, 1]
    u1 = (ru1 + 1) / 2
    v1G = (rv1 + 1) / 2
    u2 = (ru2 + 1) / 2
    v2G = (rv2 + 1) / 2

    #Inverso del giro de la imagen verticalmente
    v1 = 1 - v1G
    v2 = 1 - v2G

    #Inverso de la normalizacion
    Px = int(round(u1 * sizeAnchoOrig, 0))
    Py = int(round(v1 * sizeAltoOrig, 0))
    Qx = int(round(u2 * sizeAnchoOrig, 0))
    Qy = int(round(v2 * sizeAltoOrig, 0))

    #Coordenadas en Pixels
    coord = str(Px) + " " + str(Py) + " " + str(Qx) + " " + str(Qy)

    return coord
```

**Figura 75:** Código programa Python

## 5.3 Herramientas utilizadas

Para la realización del proyecto se han empleado las herramientas que se muestran en la siguiente tabla:

| Herramienta | Versión      |
|-------------|--------------|
| Linux       | Ubuntu 14.04 |
| Apache      | 2.4.7        |
| PHP         | 5.5.9        |
| Vuforia     | SDK 4-0-103  |
| Eclipse     | 4.2.1        |
| Python      | 2.7.6        |
| Android     | ADT 23.0.2   |

**Tabla 10:** Herramientas utilizadas.

### **6. Pruebas**

A continuación se presenta la batería de pruebas realizada. Una buena batería de pruebas sirve para depurar de manera correcta un programa y evitar que aparezcan errores durante su utilización por parte de los usuarios.

## 6.1 Pruebas de caja negra

\*No se considerará la falta de datos como una de las pruebas. La salida en cualquier caso que falten datos debe ser un aviso al usuario. Salvo en editar botón, que el identificará la falta de datos por el no cambio de los anteriores y pasará a pedir el siguiente valor.

### 6.1.1 Programa Python

#### *Añadiendo botones*

##### Particiones en casos de equivalencia

| Condición  | Casos válidos   | Casos no válidos  |
|--|---|---|
| Introducción de nombre /<br>identificador de botón | El nombre/identificador de botón es único. (1)  | El nombre/identificador está en uso (4)   |
| Introducción de coordenadas                        | Las coordenadas del botón corresponden a 4 valores numéricos sin ningún conflicto con los botones existentes. (2) | <ul style="list-style-type: none"> <li>• No se trata de 4 elementos numéricos. (5.1)</li> <li>• Las coordenadas no corresponden a puntos opuestos. (5.2)</li> <li>• Las coordenadas se encuentran dentro de otro botón. (5.3)</li> <li>• Las coordenadas contienen otro botón en su interior. (5.4)</li> <li>• Las coordenadas coinciden con los vértices de otro botón. (5.5)</li> <li>• Las coordenadas se encuentran en la arista de otro botón. (5.6)</li> <li>• Las coordenadas no se encuentran dentro del tamaño original de la imagen. (5.7)</li> </ul> |
| Introducción de enabled                            | El valor de enabled se encuentra en [true, false]. (3)  | <ul style="list-style-type: none"> <li>• El valor introducido por el usuario no corresponde</li> </ul>  |

## Trabajo Fin de Grado

|  |  |                           |
|--|--|---------------------------|
|  |  | con “true” o “false”. (6) |
|--|--|---------------------------|

**Tabla 11:** Casos de equivalencia.

### Casos de prueba

| Caso de prueba   | Casos cubiertos | Salida esperada |
|--|-----------------|-----------------|
| El usuario introduce un nombre/identificador de botón que ya se utiliza en otro elemento de laboratorio. | 1               | OK              |
| El usuario introduce un nombre/identificador de botón que ya se utiliza en ese elemento de laboratorio.  | 4               | ERROR           |
| El usuario introduce como coordenadas 4 valores numéricos que no generan conflictos.                     | 2               | OK              |
| El usuario introduce como coordenadas "15 16 34 ASD"   | 5.1             | ERROR           |
| El usuario introduce como coordenadas "15 34 15 75"  | 5.2             | ERROR           |
| El usuario introduce unas coordenadas que se encuentran en el interior de otro botón.                    | 5.3             | ERROR           |
| El usuario introduce unas coordenadas muy grandes que contienen a otro botón en su interior.             | 5.4             | ERROR           |
| El usuario introduce unas coordenadas con un vértice igual al de otro botón.                             | 5.5             | ERROR           |
| El usuario introduce unas coordenadas con un vértice situado en la arista de otro botón.                 | 5.6             | ERROR           |
| El usuario introduce unas  | 5.7             | ERROR           |

## Trabajo Fin de Grado

|   |   |       |
|---|---|-------|
| coordenadas situadas fuera del tamaño original d la imagen. |   |       |
| El usuario introduce "FALSE" como valor para enabled.       | 3 | OK    |
| El usuario introduce "verdadero" como valor para enabled.   | 6 | ERROR |

**Tabla 12:** Casos de prueba.

### *Editando botones*

Mismas pruebas que añadiendo botones con la diferencia que se permite volver a introducir los mismos datos que ya había para ese botón.

### *Borrado de botones*

#### Particiones en casos de equivalencia

| Condición                   | Casos válidos   | Casos no válidos        |
|-----------------------------|---|-------------------------|
| Selección de botón a borrar | El botón se encuentra dentro del rango del numero de botones. (1) | El botón no existe. (2) |

**Tabla 13:** Casos de equivalencia.

#### Casos de prueba

| Caso de prueba  | Casos cubiertos | Salida esperada |
|---|-----------------|-----------------|
| El usuario selecciona un botón existente en el elemento del laboratorio.        | 1               | OK              |
| El usuario entra a un Target sin botones e introduce un identificador de botón. | 2               | ERROR           |

**Tabla 14:** Casos de prueba.



## Trabajo Fin de Grado

### *Modificar tamaño*

#### Particiones en casos de equivalencia

| Condición                      | Casos válidos  | Casos no válidos  |
|--------------------------------|--|---|
| Seleccionar Target a modificar | El Target existe. (1)  | El Target no existe. (3)  |
| Nuevo tamaño.                  | <ul style="list-style-type: none"><li>El nuevo tamaño es mayor que las coordenadas de los botones existentes. (2.1)</li><li>Los datos introducidos corresponden a dos valores numéricos. (2.2)</li></ul> | <ul style="list-style-type: none"><li>El nuevo tamaño es menor que los botones existentes. (4.1)</li><li>Los datos no corresponden a dos valores numéricos. (4.2)</li></ul> |

**Tabla 15:** Casos de equivalencia.

#### Casos de prueba

| Caso de prueba  | Casos cubiertos | Salida esperada |
|---|-----------------|-----------------|
| El usuario selecciona un Target existente para modificar  | 1               | OK              |
| El usuario selecciona un Target inexistente para modificar  | 3               | ERROR           |
| El usuario introduce como nuevo tamaño "250 300" y los botones del Target se encuentran dentro de las coordenadas | 2.1 y 2.2       | OK              |
| El usuario introduce como nuevo tamaño "250 300" teniendo un botón fuera de ese tamaño.                           | 4.1             | ERROR           |
| El usuario introduce como nuevo tamaño "250 350 5F"   | 4.2             | ERROR           |

**Tabla 16:** Casos de prueba.

### *Establecer tamaño*

Similar a modificar tamaño, solo que no se recalculan las posiciones de los botones.

### *Modificar descripción*

Similar a modificar tamaño, solo que no se recalculan las posiciones de los botones.

## Trabajo Fin de Grado

### *Seleccionar Target*

La prueba se cubre con la primera condición de modificar tamaño de los casos de equivalencia.

### *Crear base de datos*

Particiones en casos de equivalencia

| Condición                                 | Casos válidos   | Casos no válidos  |
|---|---|---|
| El usuario introduce argumentos.          | El usuario introduce dos argumentos y los ficheros existen. (1) | <ul style="list-style-type: none"><li>• El usuario no introduce 2 argumentos. (5.1)</li><li>• Los ficheros de entrada no existen. (5.2)</li></ul>                             |
| El usuario no introduce argumentos.       | Existe una base de datos antigua. (2)                           | No existe ninguna base de datos antigua. (6)  |
| El usuario selecciona idioma de carga.    | Hay una base de datos antigua con el idioma seleccionado. (3)   | <ul style="list-style-type: none"><li>• No hay ninguna base de datos antigua con el idioma seleccionado. (7.1)</li><li>• El idioma seleccionado no es válido. (7.2)</li></ul> |
| El usuario selecciona idioma para guardar | El usuario elige como idioma español o inglés. (4)              | El usuario no escoge ni el español ni el inglés. (8)  |

**Tabla 17:** Casos de equivalencia.

### Casos de prueba

| Caso de prueba  | Casos cubiertos | Salida esperada |
|---|-----------------|-----------------|
| El usuario introduce como argumentos un fichero XML y otro DAT existentes     | 1               | OK              |
| El usuario no introduce ningún argumento y se carga la base de datos anterior | 2               | OK              |
| El usuario selecciona como idioma de carga español                            | 3               | OK              |
| El usuario selecciona como  | 4               | OK              |

|   |     |       |
|---|-----|-------|
| idioma para guardar el inglés.  |     |       |
| El usuario introduce tres argumentos por terminal.  | 5.1 | ERROR |
| El usuario introduce como argumentos un fichero XML y otro DAT que no existen.                    | 5.2 | ERROR |
| El usuario no introduce argumentos y no hay bases de datos anteriores.                            | 6   | ERROR |
| El usuario no introduce argumentos y no hay bases de datos anteriores con el idioma seleccionado. | 7.1 | ERROR |
| El usuario selecciona como idioma de carga el ruso.   | 7.2 | ERROR |
| El usuario selecciona como idioma de carga el francés.  | 8   | ERROR |

**Tabla 18:** Casos de prueba.

### 6.1.2 Programa Android

El usuario en ningún momento debe introducir datos a la aplicación, tan solo se mueve entre las pantallas disponibles obteniendo información. No hay casos en las pruebas de caja negra en la cual el usuario deba rellenar algún formulario y la aplicación deba controlar la entrada de datos, por lo cual si hay algún error lo veremos con las pruebas de caja blanca controlando el flujo de la aplicación.

### **6.2 Pruebas de caja blanca**

Las pruebas de caja blanca son necesarias para comprobar el correcto funcionamiento de la aplicación en el ámbito interno. Para poder realizarlas es necesario tener una visión del flujo de la aplicación para comprobar, en este caso, que no hay fallos a la hora de navegar por los menús de la aplicación y generar los botones.

## 6.2.1 Programa Python

### Añadir / Editar botones

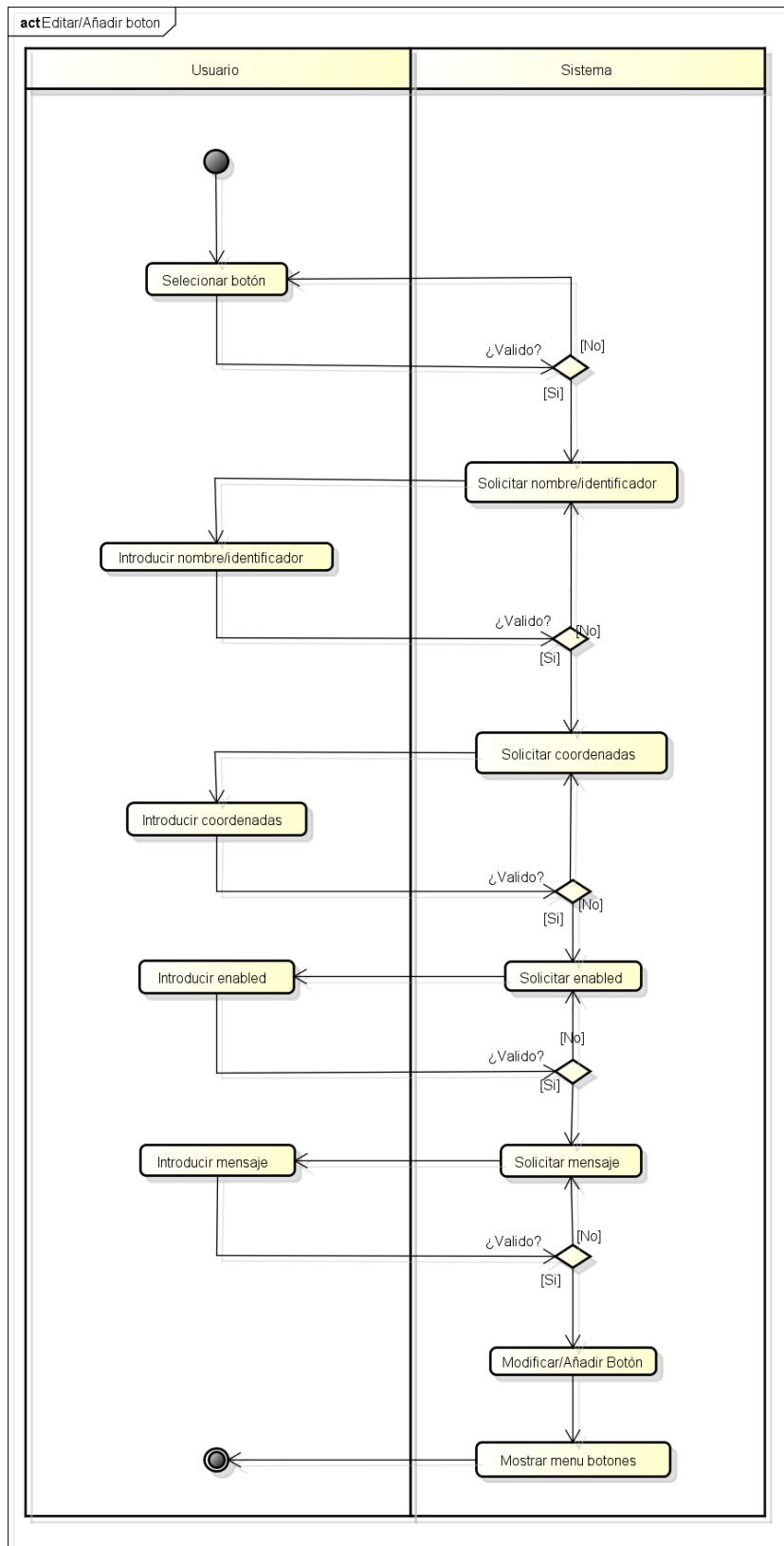


Figura 76: Diagrama de actividad

## Borrar botón

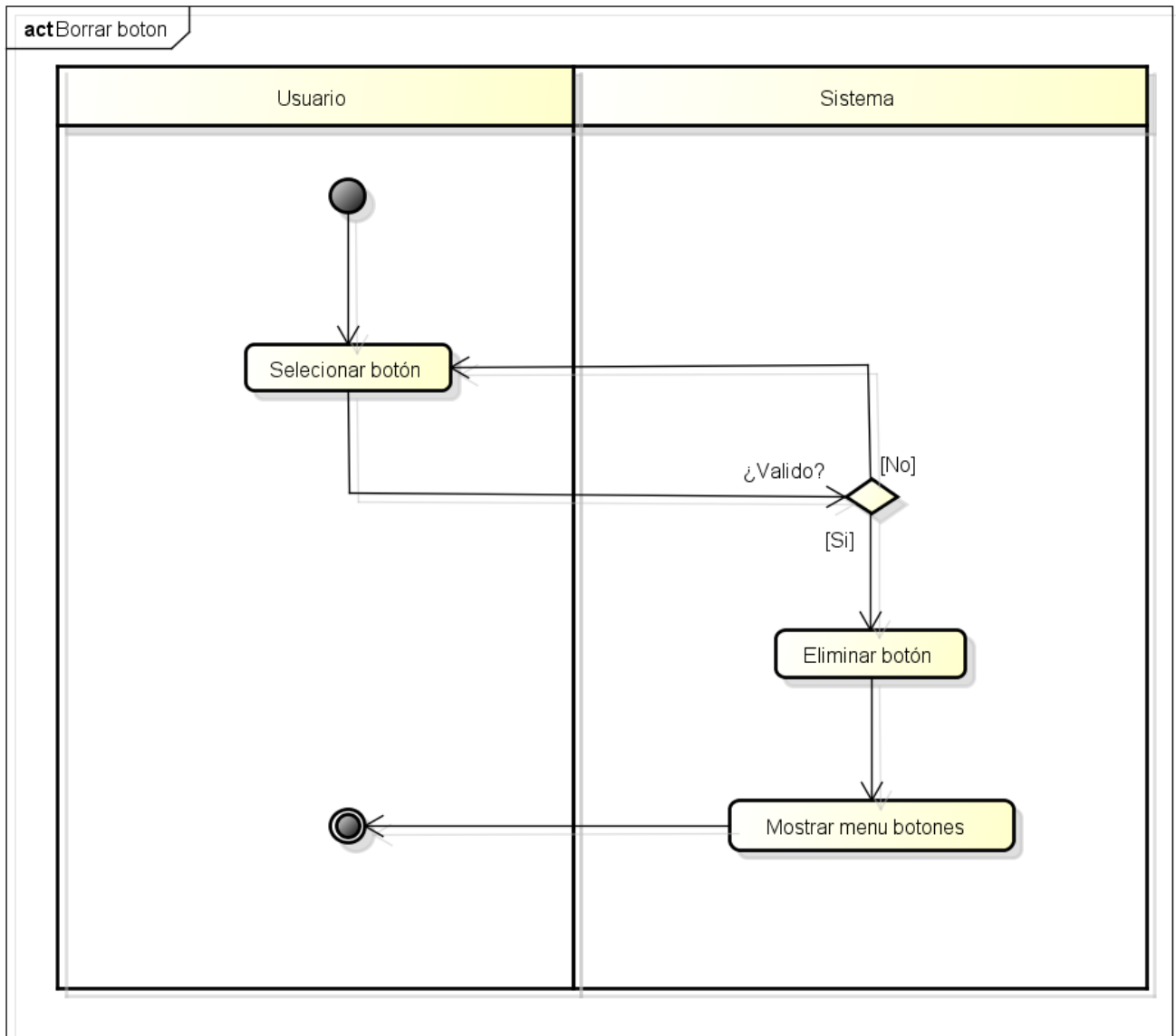


Figura 77: Diagrama de actividad

## Seleccionar Target y establecer tamaño

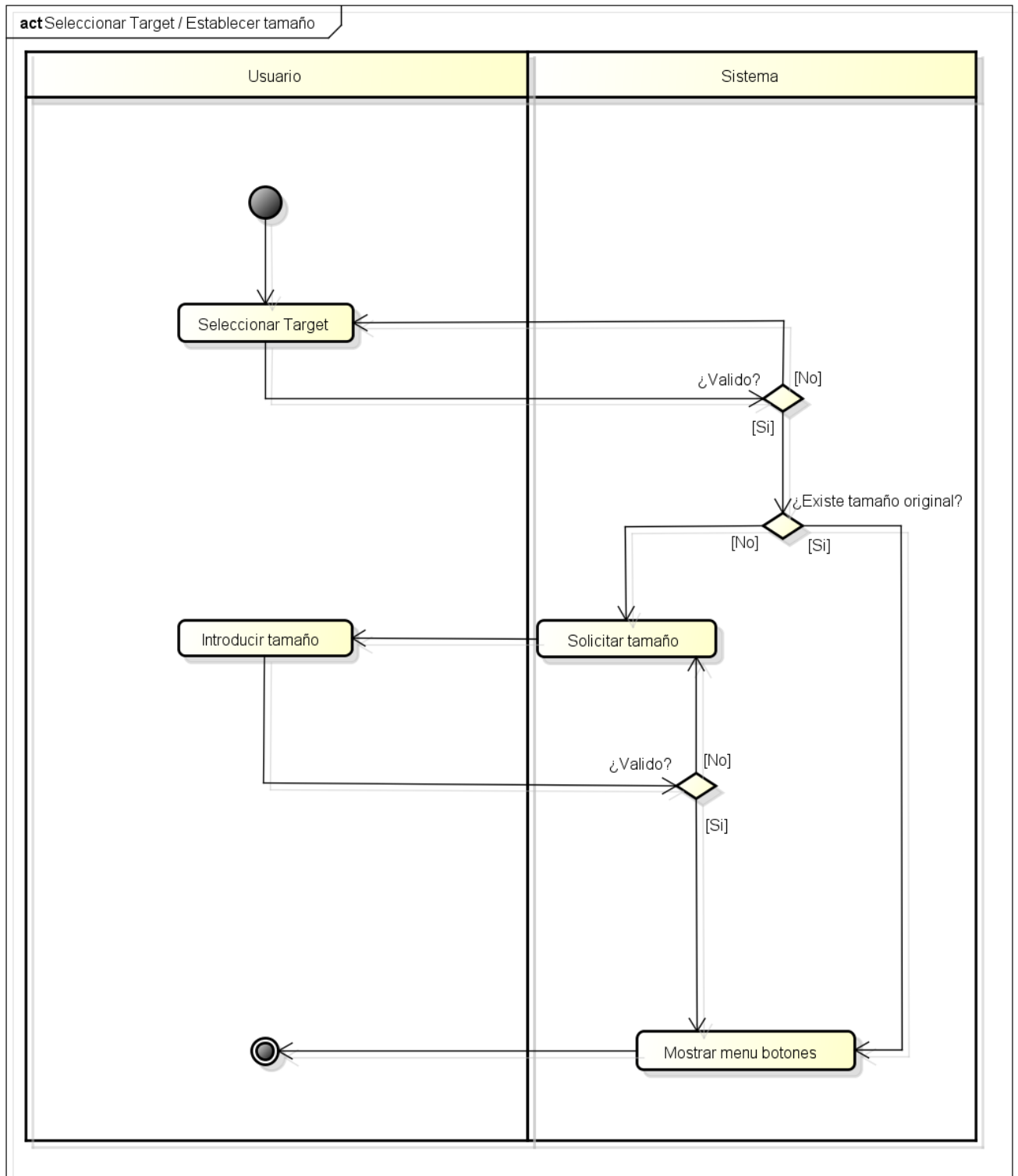


Figura 78: Diagrama de actividad

## Modificar tamaño

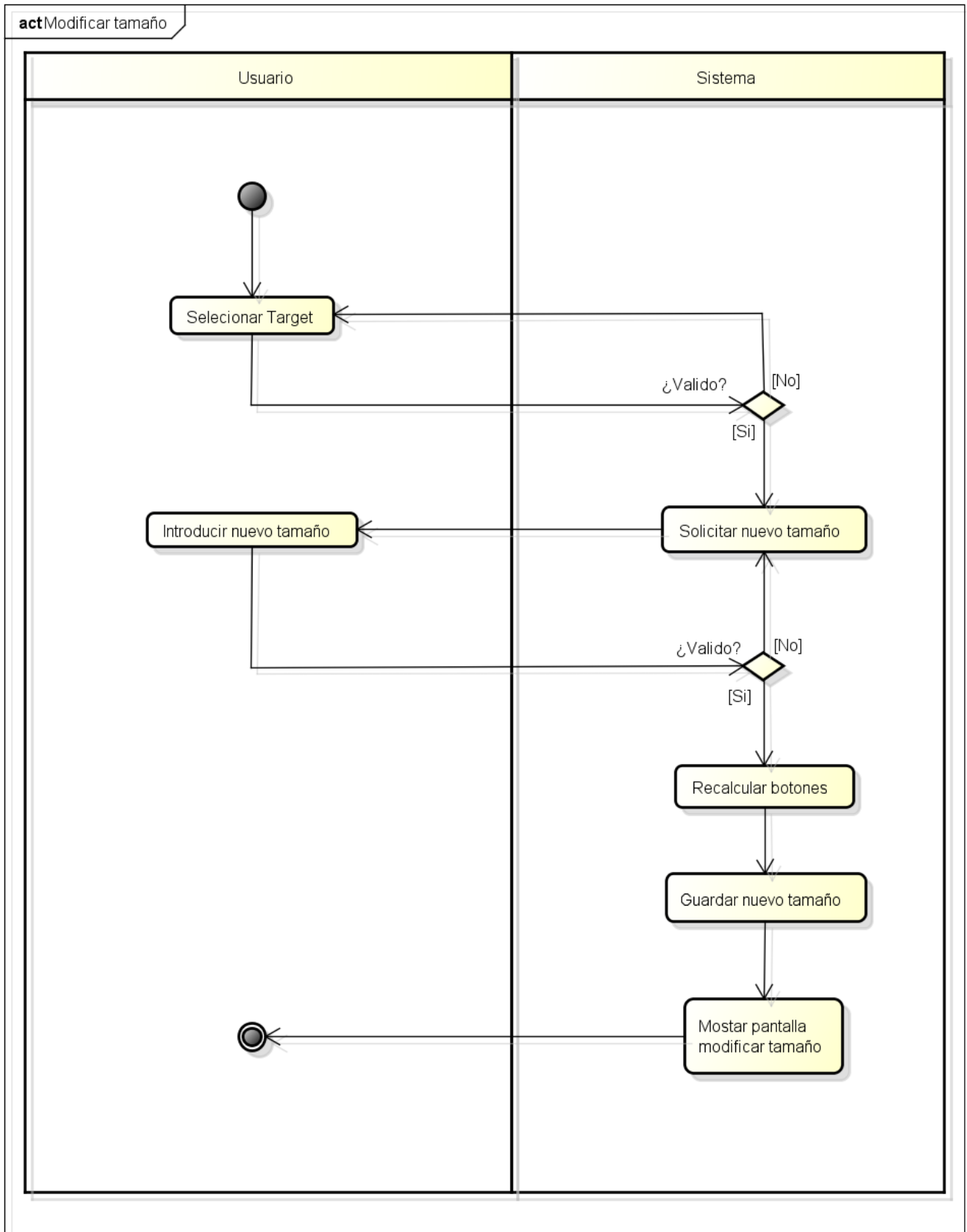


Figura 79: Diagrama de actividad



## Crear base de datos

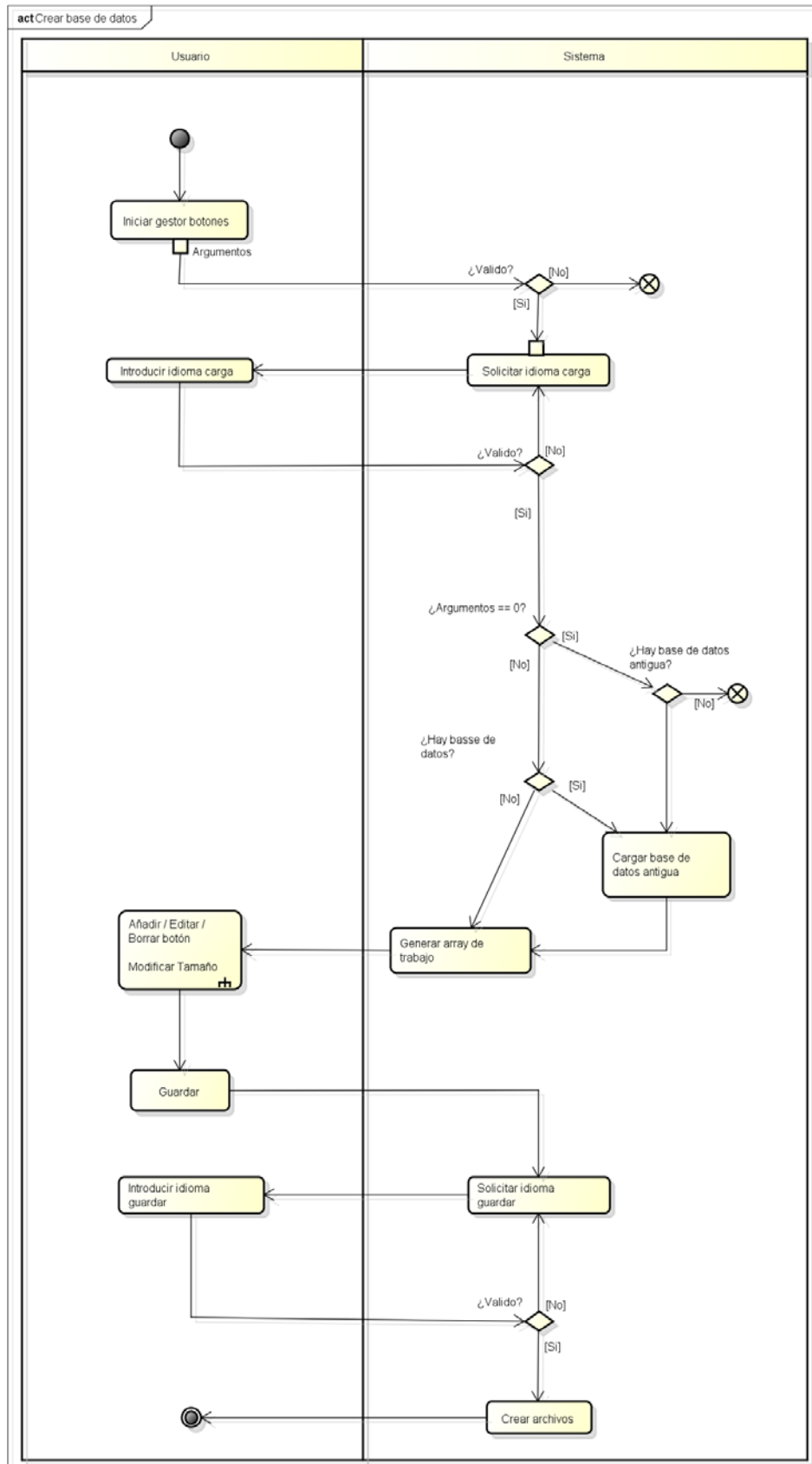


Figura 80: Diagrama de actividad

## Crear base de datos

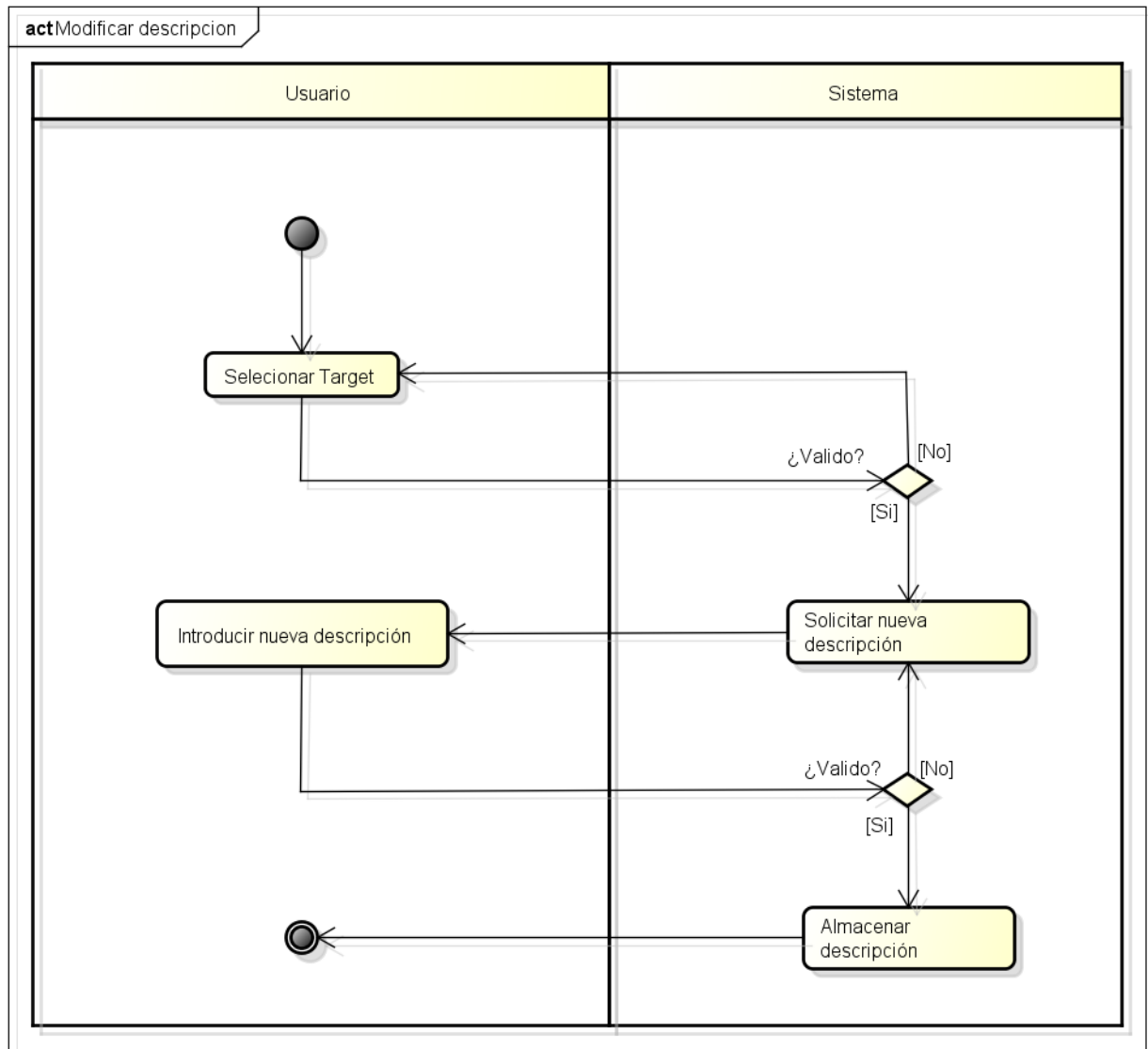


Figura 81: Diagrama de actividad

## 6.2.2 Aplicación Android

### Reconocer instrumento de laboratorio

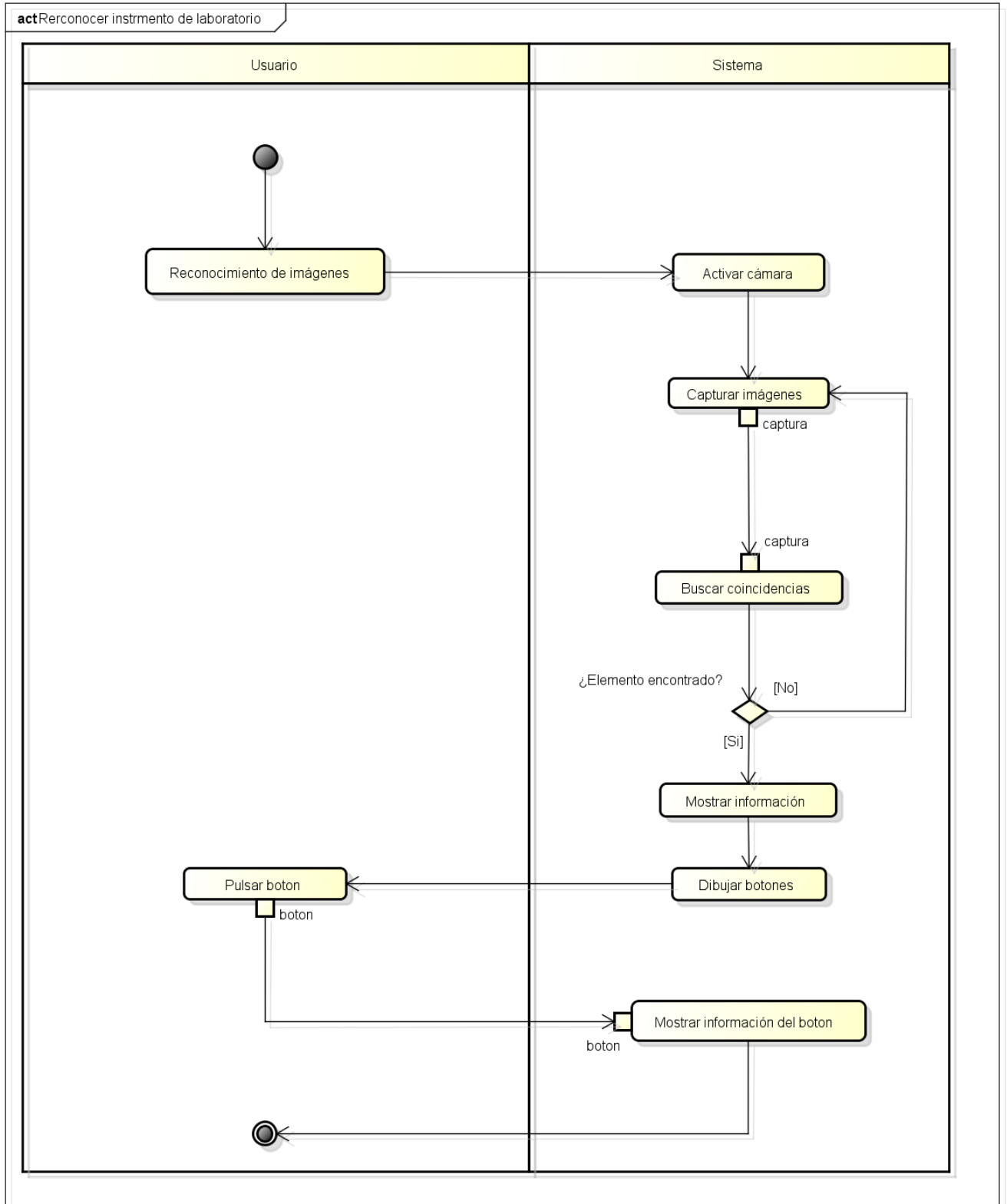


Figura 82: Diagrama de actividad

## Actualizar base de datos

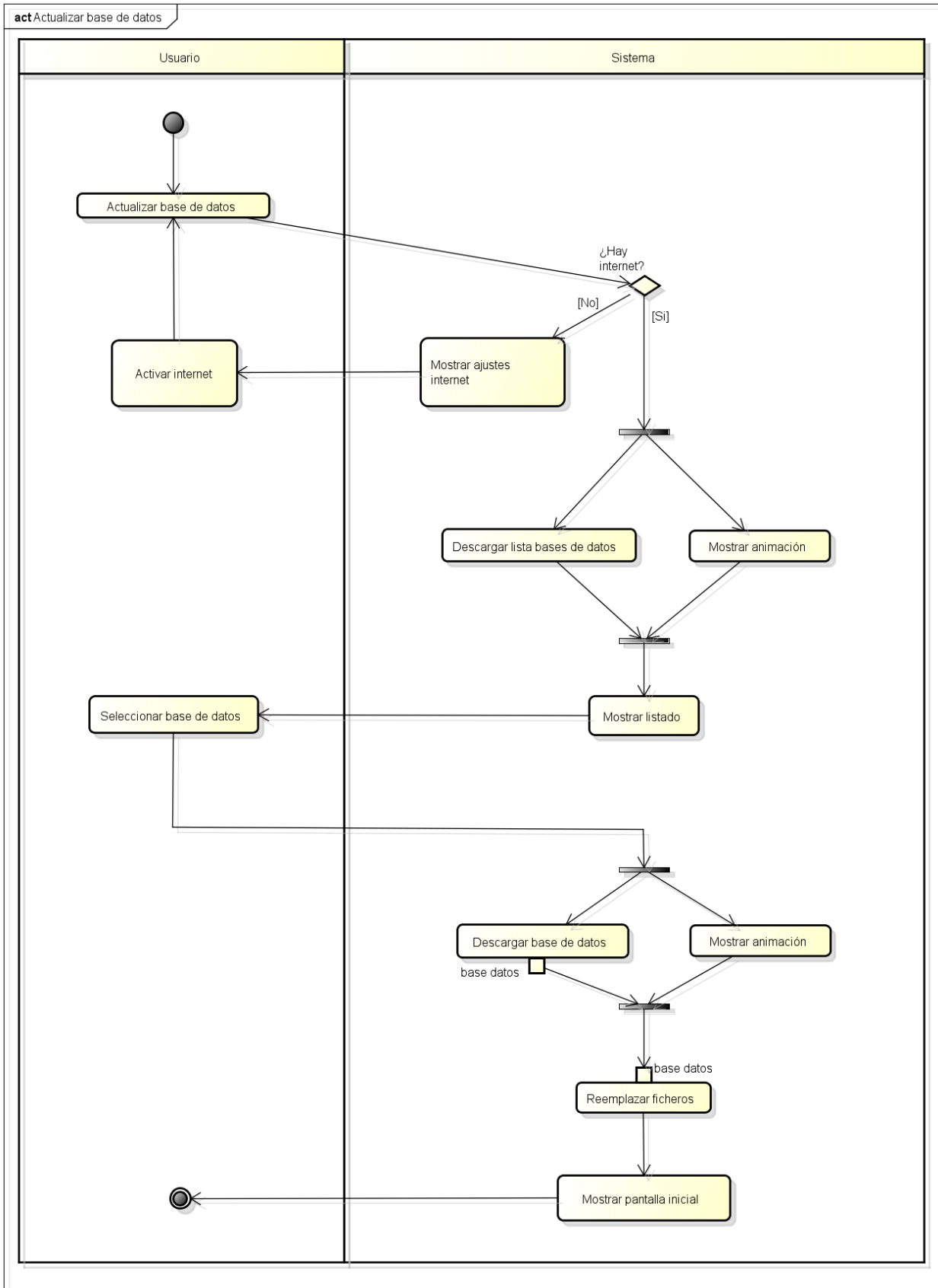
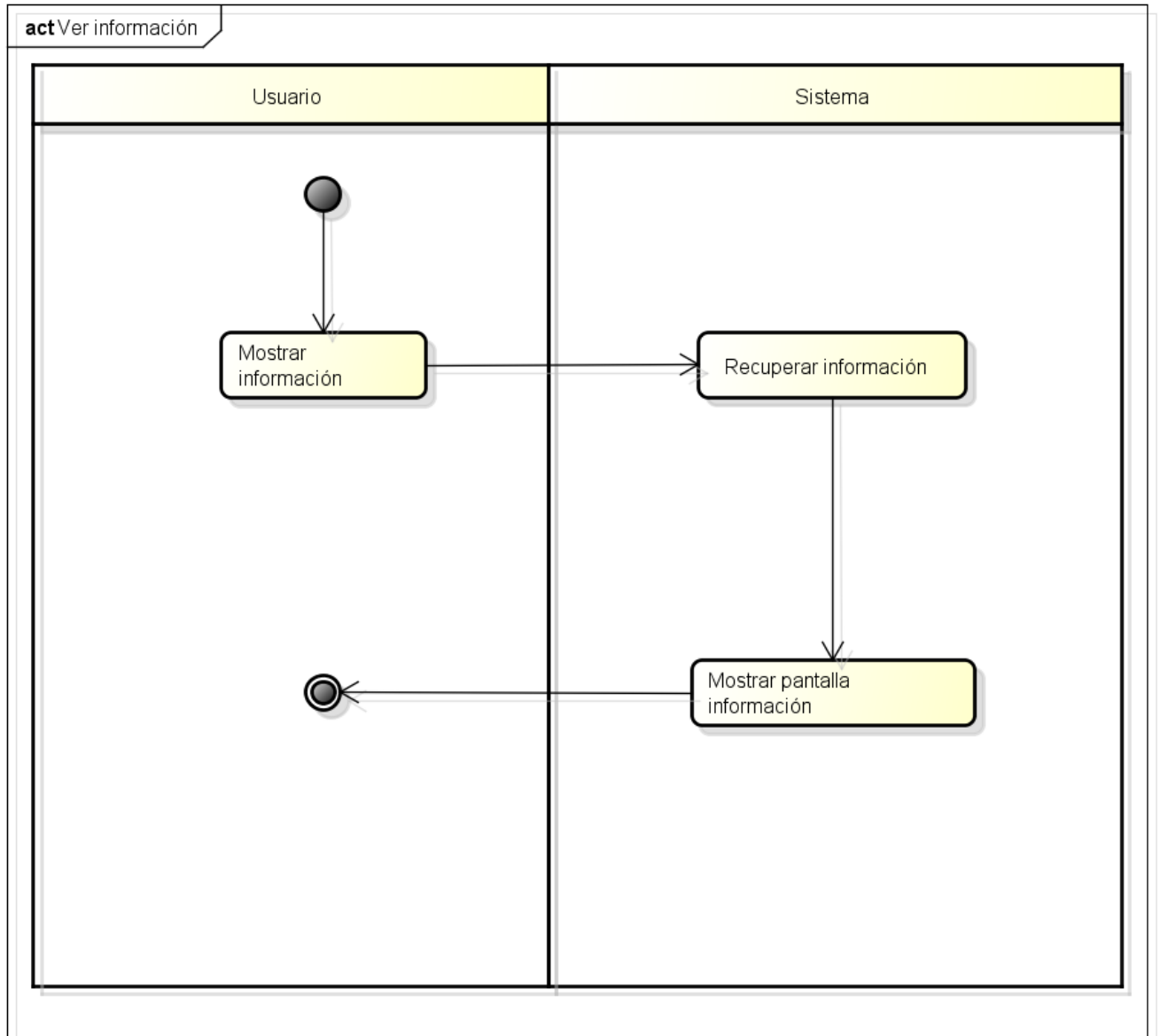


Figura 83: Diagrama de actividad

## Ver información



**Figura 84:** Diagrama de actividad

# 7. Conclusiones y trabajos futuros

## 7.1 Conclusiones

Durante el desarrollo del proyecto se han afrontado muchos retos, lo que en un principio parecía un trabajo con una gran cantidad de tiempo, poco a poco se convirtió en un desarrollo estresante plagado de riesgos que aparecían sin cesar. Todo esto ha sido una experiencia angustiosa en algunos momentos pero enriquecedora para el futuro, ya que trabajar bajo gran presión no será algo nuevo en el tan esperado futuro laboral.

En un principio el desarrollo con Vuforia se hizo tedioso, ya que no se comprendían las funciones y lógica de este framework. Se trataba de realizar ingeniería inversa en la sección "How to" de la página para desarrolladores, pero sin mucho acierto, hasta que encontré la API de Vuforia y comencé a saber manejarme con los métodos. Esto demuestra la importancia de una buena documentación.

El desarrollo con Vuforia ha servido para tocar la punta del iceberg de esta nueva tecnología que parece que es ahora cuando va a tener su gran auge y servirá para poder afrontar futuros proyectos de realidad aumentada pensando de una manera distinta, pues estas aplicaciones no interactúan con el usuario de la misma manera que una aplicación tradicional.

Este desarrollo en solitario también ha servido para comprender la gran importancia del trabajo en equipo, ya que al llevar a cabo el proyecto en solitario los parones debidos a la aparición de riesgos eran inevitables. Así como la falta de otra visión crítica sobre las decisiones de análisis y diseño, pues es bien sabido que *cuatro ojos ven más que dos*.

Aunque en un principio el propósito de este proyecto era la realización de una aplicación móvil solamente, ha resultado muy satisfactorio el desarrollo del programa que ayuda a gestionar los botones, ya que de esta manera se ha aumentado mucho la usabilidad de la aplicación al proponer un método sencillo de aumentar el número de elementos a desarrollar y sin este programa la aplicación daría la sensación de estar incompleta.

Además, realizar el programa de gestión de botones en un lenguaje de programación que se conocía bastante bien resultó todo un acierto, de haber utilizado otro lenguaje distinto habría sido imposible la presentación del proyecto en julio, ya que los desarrollos en Python fueron muy eficaces y esto permitió centrarse más en el código Java de la aplicación Android.

### 7.2 Trabajos futuros

Desde que se pensó la necesidad de un programa que ayudara a gestionar los botones me centré en desarrollar algo similar al programa ya existente en el laboratorio de física para generar rectas, pensando que de esta manera la curva de aprendizaje del usuario que tuviera que trabajar con el gestor de botones tendría una pendiente muy reducida.

Fue esta obsesión por intentar facilitar el trabajo al usuario final lo que me cegó y no me permitió ver hasta comienzos de junio, cuando ya era demasiado tarde para cambiar todas las especificaciones del proyecto y todo el código ya realizado, la posibilidad de llevar a cabo un desarrollo en PHP que permitiera al usuario desarrollador subir los archivos sin necesidad de entrar mediante sesión SSH al servidor y llevar toda la gestión.

Por ello en un futuro podría trasladarse toda la lógica del programa en Python a una página PHP que realice todas las validaciones y cálculos necesarios para crear las nuevas bases de datos. De esta manera resultaría en una interfaz más amigable con el usuario desarrollador.

Otro posible trabajo futuro sería la creación por modelados en 3D de elementos de laboratorio en realidad aumentada, permitiendo a los usuarios finales (alumnos) interactuar con estos elementos de realidad aumentada sin necesidad de encender siquiera los instrumentos. En este caso la lógica de la aplicación sería muy complicada y no sería posible que una sola persona se embarcara en este proyecto ya que al contrario de la propuesta anterior sería necesarios unos conocimientos en física muy elevados así como en modelado 3D.

## 8. Bibliografía

### Bibliografía

[1] Proyecto fin de carrera: Aplicación de realidad aumentada Android, por Nuria González Delgado. Universidad de Valladolid.

[2] "El gran libro de Android". Jesús Tomás Gironés. Editorial Marcombo.

### Webgrafía

[3] <http://developer.vuforia.com>

[4] <https://developer.vuforia.com/resources/api/main>

[5] <http://developer.android.com/reference/packages.html>

[6] <http://docs.oracle.com/javase/7/docs/api/>

[7] <http://stackoverflow.com/>

[8] <http://www.sgoliver.net/blog/curso-de-programacion-android/>

[9] <https://docs.python.org/2/library/xml.etree.elementtree.html>

[10] <http://www.americalearningmedia.com/component/content/article/69-tester/264-13-aplicaciones-de-realidad-aumentada>

[11] <http://www.mkyong.com/java/how-to-convert-inputstream-to-file-in-java/>



## **9. Anexos**

### **9.1 Contenido del DVD**

El CD está dividido en 4 directorios que son los siguientes:

1. AppMovil, que contiene la APK y una copia del código fuente realizado de la aplicación móvil.
2. Documentación, en el cual está incluido este documento.
3. ProgramaGestor, que contiene el código Python del programa gestor de botones para la aplicación móvil.

### 9.2 Glosario de términos

APK: Archivo ejecutable para instalar aplicaciones en Android. Variante del formato JAR de Java (misma funcionalidad).

2D: 2 dimensiones.

3D: 3 dimensiones.

Código QR: Matriz en dos dimensiones formada por cuadrados negros sobre fondo blanco.

Wifi: Conexión a internet inalámbrica.

SDK: Kit de desarrollo software. Conjunto de herramientas para crear aplicaciones para un sistema concreto.

Framework: Estructura de soporte conceptual y tecnológica de soporte definido.

GPS: "Global Positioning System", es un sistema de navegación por satélite.

Plugin: Modulo software o hardware que aumenta las funcionalidades de otro (característica o servicio).

Unity3D: Herramienta para desarrollo en el modelado en 3D.

Target: Imagen objetivo que es detectado por Vuforia.

Target Manager: Aplicación web que genera las bases de datos (de dispositivo o en la nube) usadas por Vuforia.

Tracking: Rastreo o seguimiento de las imágenes por parte de Vuforia.

Threading: Uso de distintos hilos de procesamiento.

SSH: Protocolo y programa que lo implementa, sirve para acceder a máquinas a través de la red.

Stakeholders: Parte interesada.

CU: Casos de uso.

RU: Requisitos de usuario.

RF: Requisitos funcionales.

RNF: Requisitos no funcionales.

RI: Requisitos de la información.

ISO/IEC 27005:2008: Estándar utilizado para identificar la exposición al riesgo.

|                 |           | Likelihood of Incident Scenario |     |        |      |           |
|-----------------|-----------|---------------------------------|-----|--------|------|-----------|
|                 |           | Very Low                        | Low | Medium | High | Very High |
| Business Impact | Very Low  | 0                               | 1   | 2      | 3    | 4         |
|                 | Low       | 1                               | 2   | 3      | 4    | 5         |
|                 | Medium    | 2                               | 3   | 4      | 5    | 6         |
|                 | High      | 3                               | 4   | 5      | 6    | 7         |
|                 | Very High | 4                               | 5   | 6      | 7    | 8         |

**Tabla 19:** Escala de riesgos

TextView: Elemento gráfico de Android que sirve para mostrar un mensaje.

Listitem: Elemento gráfico de Android que además de mostrar un mensaje es clickeable.

Array: Vector de elementos de otro tipo.

True: Valor booleano (cierto).

False: Valor booleano (falso).

Javadoc: Utilidad de Oracle que permite generar documentación en formato HTML.

Pydoc: Herramienta incluida con Python que permite generar documentación en formato HTML:

HTML: Lenguaje elaborado para el desarrollo de páginas web.

ArrayList: Clase similar a Array, con la diferencia que ArrayList permite almacenar un número de elementos dinámico.

Layout: Elemento gráfico contenedor de otros elementos gráficos de Android.

XML: Lenguaje de marcas que permite la organización y etiquetado de documentos.

DAT: Extensión de ficheros de tipo binario.

API: Conjunto de rutinas, funciones y procedimientos que ofrece una biblioteca.

ADT: Plugin para Eclipse que proporciona un conjunto de herramientas que se integran en el IDE de Eclipse.

IDE: Entorno de desarrollo integrado.

Parser: Interprete de palabras.

Include: Se incluyen las clases dentro del programa y cualquier función puede hacer uso de esas clases.

Import: Se importan las clases dentro del programa y solo las funciones de ese programa que realiza la importación pueden hacer uso de esas clases.

Activity: Componente de la aplicación que provee una pantalla para los usuarios para que interactúen con ella.

SharedPreferences: Interfaz para acceder y modificar los datos de preferencias.

### 9.3 Integración de Vuforia en Eclipse

A la hora de integrar Vuforia en Eclipse podemos encontrar ciertas dificultades. Algunos de los errores comunes que encontraremos serán:

1. Archivo libVuforia.so no se añade en libs, por lo cual tendremos que añadirlo de manera manual creando antes la carpeta armeabi-v7a de la cual colgará.
2. La carpeta externa del SDK de Vuforia no se ha añadido, por lo cual en propiedades del proyecto vamos a *Java Build Path* en *libraries* y ahí haciendo click sobre *add external class folder* añadimos la carpeta /Vuforia SDK/build/java.
3. La variable QCAR\_SDK\_ROOT no se ha añadido, en el mismo lugar que antes pero esta vez haciendo click sobre *add variable* introducimos:  
"QCAR\_SDK\_ROOT/build/java/vuforia/Vuforia.jar"
4. La variable QCAR\_SDK\_ROOT se ha añadido pero no es correcta, tan solo debemos editarla e introducir la ruta anterior.
5. La variable QCAR\_SDK\_ROOT no se exporta, en *Java Build Path*, vamos a *order and export* y ahí la seleccionamos para que se exporte también.

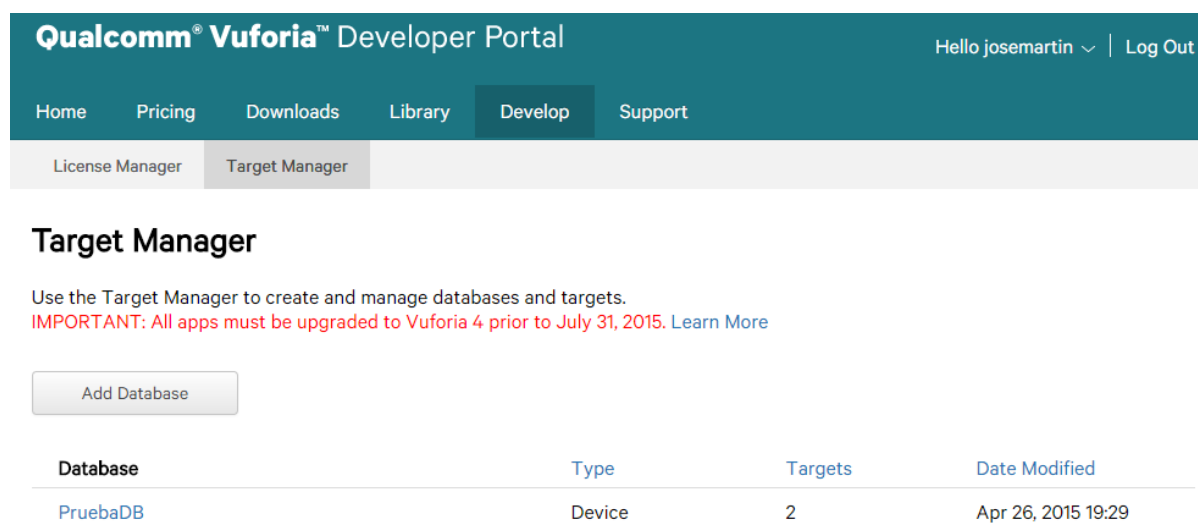
### **9.4 Manual de usuario administrador**

A continuación se explicará como un usuario administrador gestiona la creación y edición de nuevas bases de datos. Todo este proceso se explicará desde el principio. Por lo que en primer lugar se explicará el funcionamiento del Target Manager y después, el gestor de botones.

## 9.4.1 Target Manager

### Crear base de datos

El primer paso es dirigirse a la página de Vuforia, registrarse y entrar al Target Manager. Ahí al pulsar sobre *Add Database* crearemos una nueva base de datos para nuestra aplicación. Debido a que el proyecto está desarrollado sobre la versión 4 como se indicó en herramientas utilizadas, no es necesario modificar nada para que siga funcionando a partir del día 31 de julio como indica la advertencia.



The screenshot shows the Qualcomm Vuforia Developer Portal. The top navigation bar includes 'Home', 'Pricing', 'Downloads', 'Library', 'Develop', and 'Support'. Below this, there are tabs for 'License Manager' and 'Target Manager'. The 'Target Manager' section is active, displaying a message: 'Use the Target Manager to create and manage databases and targets. IMPORTANT: All apps must be upgraded to Vuforia 4 prior to July 31, 2015. Learn More'. There is an 'Add Database' button. Below the button is a table with the following data:

| Database | Type   | Targets | Date Modified      |
|----------|--------|---------|--------------------|
| PruebaDB | Device | 2       | Apr 26, 2015 19:29 |

Figura 85: Target Manager

Tras pulsar *Add Database*, se abrirá la siguiente pantalla en la cual debemos indicar el nombre de la base de datos. Y le damos a *create*.



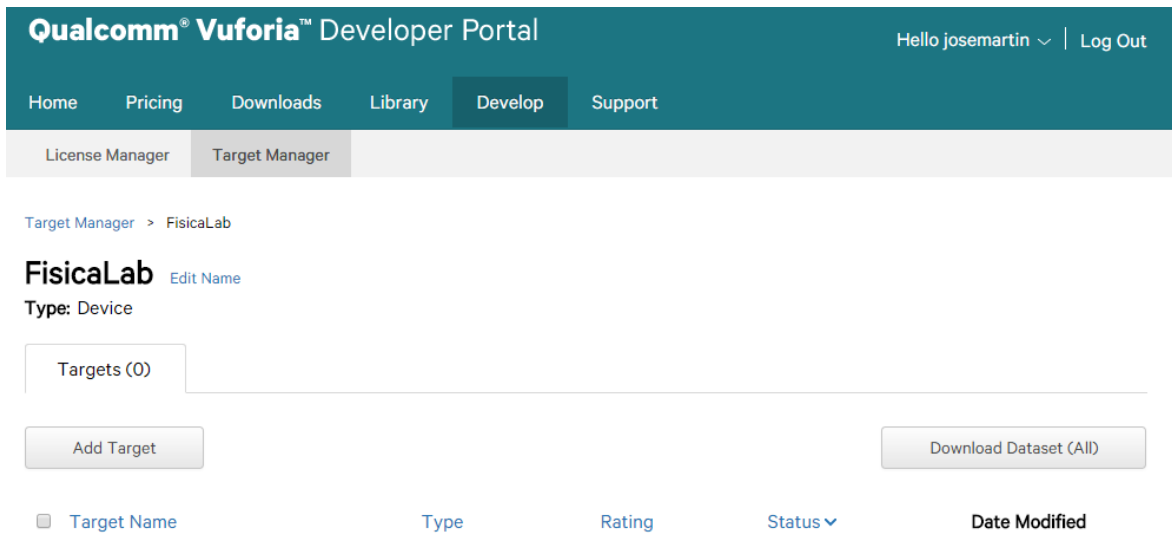
The 'Create Database' dialog box has a title bar and the following fields:

- Name:** A text input field containing 'FisicaLab|'.
- Type:** Two radio button options: 'Device' (selected) and 'Cloud'.
- At the bottom right, there are two buttons: 'Cancel' and 'Create'.

Figura 86: Estableciendo nombre de la base de datos nueva

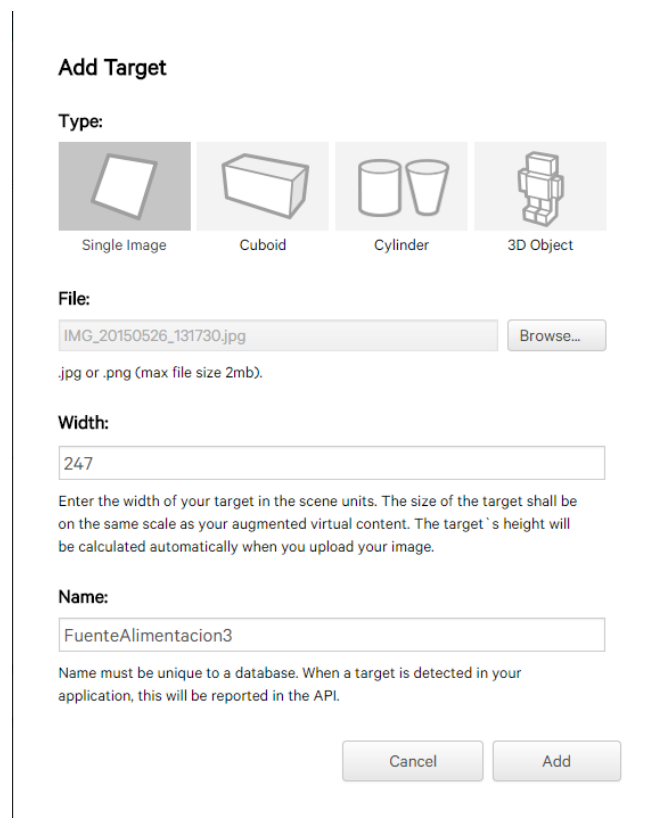
A continuación, procederemos a añadir elementos a reconocer a la base de datos clickeando en *Add Target*. Con lo que se nos abrirá una nueva ventana emergente.

# Trabajo Fin de Grado



**Figura 87:** Pantalla inicial de la base de datos

Establecemos la imagen como *Simple Image*, seleccionamos la imagen, como anchura establecemos 247 que es lo que recomienda Vuforia e indicamos el nombre, el cual debe ser único.



**Figura 88:** Añadiendo Target

Tras esto, la imagen se subirá y se añadirá a la base de datos, su estado será *Processing* (procesando), y en cuanto el Target Manager la analice establecerá el Rating de la imagen.

The screenshot shows the Qualcomm Vuforia Developer Portal interface. At the top, there is a navigation bar with the following items: Home, Pricing, Downloads, Library, Develop (highlighted), and Support. Below this, there are sub-navigation options: License Manager and Target Manager (highlighted). The main content area shows the breadcrumb 'Target Manager > FisicaLab'. The title 'FisicaLab' is displayed with an 'Edit Name' link. Below the title, it says 'Type: Device'. There is a search box labeled 'Targets (1)'. Two buttons are visible: 'Add Target' and 'Download Dataset (All)'. A table lists the targets with the following columns: Target Name, Type, Rating, Status, and Date Modified. One target is listed: 'FuenteAlimentacion' with a rating of 5 stars and a status of 'Processing', last modified on Jul 01, 2015 18:46.

| <input type="checkbox"/> | Target Name        | Type         | Rating | Status     | Date Modified      |
|--------------------------|--------------------|--------------|--------|------------|--------------------|
| <input type="checkbox"/> | FuenteAlimentacion | Single Image | ★★★★★  | Processing | Jul 01, 2015 18:46 |

**Figura 89:** Añadiendo Target

Cuando tengamos todos los elementos cargados en la base de datos, tan solo debemos clicar en *Download Dataset (All)* con lo que descargaremos la base de datos y podremos pasar al gestor de botones.



## 9.4.2 Gestor de Botones

### Iniciar el gestor de botones

Es importante que a la hora de ejecutar el gestor de botones se haga con permisos de administrador, pues va a generar ficheros en un directorio el cual la escritura está restringida. Por ello el comando para ejecutar el gestor de botones es como se ve en la siguiente imagen.

```
sudo python gestorBotones.py FisicaLab.xml Fisicalab.dat
```

**Figura 90:** Comando ejecución del gestor de botones

Con ello, nos lanzará la pantalla para cargar las bases de datos antiguas en la que seleccionaremos si queremos cargar la última base de datos en español o en inglés tal y como se indica en el menú.

```
Editor de botones virtuales para la aplicación móvil de Física
Selección de idioma
Indique si desea cargar la última base de datos disponible con valores en español (ES) o en Inglés (EN)

Idioma: es
```

**Figura 91:** Seleccionando idioma de carga

Tras esto, cargará (si existe) la última base de datos, y nos mostrará el menú principal. En el cual podremos seleccionar un Target, modificar el tamaño original de un Target, modificar su descriptor, guardar y salir o salir sin guardar.

```
Editor de botones virtuales para la aplicación móvil de Física
Lista de Targets:
Target: 1
    nombre: Polimetro
    tamaño original:
    descriptor:
    numero de botones: 0
Target: 2
    nombre: FuenteAlimentacion2
    tamaño original:
    descriptor:
    numero de botones: 0
Target: 3
    nombre: FuenteAlimentacion
    tamaño original:
    descriptor:
    numero de botones: 0
Seleccionar Target (id) - Modificar tamaño (M) - Modificar descriptor (D) - Guardar y salir (G) - Salir sin guardar (S)
Opcion: █
```

**Figura 92:** Menú principal

### Selecciona un Target

Seleccionamos el primer Target introduciendo su identificador numérico (1) y pulsando la tecla de retorno, con lo cual al no tener establecido un tamaño original nos pedirá que indiquemos este tamaño original. Por lo que necesitaremos saber el tamaño original de la imagen que subimos al Target Manager de Vuforia.

```
Editor de botones virtuales para la aplicación móvil de Física
Es necesario establecer el tamaño original del Target para continuar
Target: 1
Introduzca el ancho y alto original: - Cancelar y volver (C)
Establecer tamaño:
```

**Figura 93:** Establecer tamaño original

Tras introducir un tamaño válido se nos mostrará el menú principal con las opciones posible para realizar sobre el Target, como editar un botón existente, borrar un botón existente, añadir un nuevo botón, volver a la pantalla anterior, guardar y salir o salir sin guardar.

```
Editor de botones virtuales para la aplicación móvil de Física
Lista de Botones:
Editar boton (E) - Borrar boton (B) - Añadir boton (A) - Volver atras (V) - Guardar y salir (G) - Salir sin guardar (S)
Opcion:
```

**Figura 94:** Menú botones del Target

## Añadir un botón

Introduciendo el carácter a en el menú anterior, accederemos a la pantalla de añadir un nuevo botón.

```
Editor de botones virtuales para la aplicación móvil de Física
Añadiendo boton: 1
nombre:
coordenadas (Pixels):
enabled:
mensaje:
Introduzca el nuevo nombre: - Cancelar y volver (C)
Valores boton:
```

**Figura 95:** Pantalla añadir botón

En primer lugar se nos solicitará el nombre del botón (debe ser único).

```
Editor de botones virtuales para la aplicación móvil de Física
Añadiendo boton: 1
nombre:
coordenadas (Pixels):
enabled:
mensaje:
Introduzca el nuevo nombre: - Cancelar y volver (C)
Valores boton: CC
```

**Figura 96:** Estableciendo nombre de botón

# Trabajo Fin de Grado

Tras introducir el nombre, la pantalla se irá actualizando e irá mostrando al usuario los valores introducidos para que el usuario pueda comprobar en todo momento si se ha equivocado y quiere cancelar el proceso. El siguiente valor que deberá introducir el usuario son las coordenadas del botón como puede verse a continuación.

```
Editor de botones virtuales para la aplicación móvil de Física

Añadiendo boton: 1
nombre: CC
coordenadas (Pixels):
enabled:
mensaje:

Introduzca las nuevas coordenadas Px, Py, Qx, Qy(en pixels, deben corresponder a 2 vértices opuestos): - Cancelar y volver (C)
Valores boton: 560 450 780 580
```

**Figura 97:** Estableciendo coordenadas de botón

El siguiente valor que se solicita es *enabled* que solo puede ser true o false.

```
Editor de botones virtuales para la aplicación móvil de Física

Añadiendo boton: 1
nombre: CC
coordenadas (Pixels): 560 450 780 580
enabled:
mensaje:

Introduzca el nuevo valor de enabled(true o false): - Cancelar y volver (C)
Valores boton: true
```

**Figura 98:** Estableciendo valor boolean de botón

El último valor a introducir será el mensaje a mostrar en la aplicación.

```
Editor de botones virtuales para la aplicación móvil de Física

Añadiendo boton: 1
nombre: CC
coordenadas (Pixels): 560 450 780 580
enabled: true
mensaje:

Introduzca el nuevo mensaje del boton: - Cancelar y volver (C)
Valores boton: Medir corriente continua
```

**Figura 99:** Estableciendo mensaje de botón

Con lo que finalizaremos el proceso y el botón se añadirá al Target y el programa nos devolverá a la pantalla del menú de botones donde podremos ver el botón añadido. Como podemos ver en la siguiente imagen.

```
Editor de botones virtuales para la aplicación móvil de Física
Lista de Botones:
  Boton: 1
    nombre: CC
    coordenadas (Pixels): 560 450 780 580
    enabled: true
    mensaje: Medir corriente continua

Editar boton (E) - Borrar boton (B) - Añadir boton (A) - Volver atras (V) - Guardar y salir (G) - Salir sin guardar (S)
Opcion: █
```

Figura 100: Menú botones del Target

## Editar botón

Seleccionando editar en el menú de botones se abrirá la pantalla de editar botón, en la cual debemos seleccionar el botón que deseamos editar mediante su identificador numérico.

```
Editor de botones virtuales para la aplicación móvil de Física
Seleccione un boton a editar:
  Boton: 1
    nombre: CC
  Boton: 2
    nombre: CA
  Boton: 3
    nombre: Conex

Boton (id) - Volver atras (V)
Boton: █
```

Figura 101: Menú edición botones

Se nos mostrará la pantalla del botón que se está editando y sus valores antiguos. El usuario puede optar por volver a introducir los valores de nuevo simplemente pulsando la tecla de retorno enviando un carácter vacío, el gestor de botones recuperará el valor antiguo para los nuevos valores del botón.

```
Editor de botones virtuales para la aplicación móvil de Física

  Editando boton: 3
    nombre: Conex
    coordenadas (Pixels): 300 1120 820 1270
    enabled: true
    mensaje: Conexiones

  Nuevos valores de boton: 3
    nombre:
    coordenadas (Pixels):
    enabled:
    mensaje:

Introduzca el nuevo nombre:   -   Cancelar y volver (C)
Nuevo valor boton:
```

Figura 102: Editando botón

En el momento que el usuario llegue al valor que desea cambiar tan solo tiene que escribirlo y pulsar la tecla de retorno.

```
Editor de botones virtuales para la aplicación móvil de Física

Editando boton: 3
  nombre: Conex
  coordenadas (Pixels): 300 1120 820 1270
  enabled: true
  mensaje: Conexiones

Nuevos valores de boton: 3
  nombre: Conex
  coordenadas (Pixels): 300 1120 820 1270
  enabled: true
  mensaje:

Introduzca el nuevo mensaje del boton:      -   Cancelar y volver (C)
Nuevo valor boton:  Conexiones para medir corriente, tension y resistencias
```

**Figura 103:** Editando botón

Se devolverá al usuario a la pantalla de menú de botones donde podrá ver los cambios.

### **Borrar botón**

En la pantalla de menú de botones, el usuario selecciona borrar botón, con lo que se muestra la pantalla de borrado de botón, en la cual el usuario selecciona el botón a borrar mediante su identificador numérico de botón.

```
Editor de botones virtuales para la aplicación móvil de Física

Seleccione un boton a borrar:
  Boton: 1
    nombre: CC
  Boton: 2
    nombre: CA
  Boton: 3
    nombre: Conex
  Boton: 4
    nombre: NoVale

Boton (id) - Volver atras (V)
Boton: 4
```

**Figura 104:** Borrar botón

El gestor de botones eliminará el botón del Target y se devolverá a la pantalla de menú de botones.

### **Modificar descriptor de Target**

Si en la pantalla inicial en lugar de seleccionar un Target para modificar sus elementos, el usuario selecciona modificar descriptor, se muestra la pantalla de modificar descriptors, en la cual aparecen los Targets con sus descriptors para modificar. Y las distintas opciones disponibles como seleccionar un Target, volver atrás, guardar y salir o salir sin guardar.

## Trabajo Fin de Grado

---

El usuario selecciona el Target a modificar mediante su identificador numérico.

```
Editor de botones virtuales para la aplicación móvil de Física
Modificar descriptor de los Targets:
  Target: 1
    nombre: Polimetro
    descriptor:
  Target: 2
    nombre: FuenteAlimentacion2
    descriptor:
  Target: 3
    nombre: FuenteAlimentacion
    descriptor:

Seleccionar Target (id) - Volver atras (V) - Guardar y salir (G) - Salir sin guardar (S)
Opcion: 1
```

105: Pantalla descriptores

Por lo que se muestra la pantalla en la cual el usuario establecerá el nuevo valor del descriptor. Se muestra también el valor antiguo del descriptor.

```
Editor de botones virtuales para la aplicación móvil de Física
Modificando descriptor del Target: 1
  Descriptor del Target:
Intoduzca el descriptor del Target: - Volver atras (V)
Nuevo descriptor: Polimetro
```

106: Establecer descriptor

Y se devuelve al usuario a la pantalla de seleccionar Targets para modificar su descriptor, en la cual podrá ver los cambios realizados.

```
Editor de botones virtuales para la aplicación móvil de Física
Modificar descriptor de los Targets:
  Target: 1
    nombre: Polimetro
    descriptor: Polimetro
  Target: 2
    nombre: FuenteAlimentacion2
    descriptor:
  Target: 3
    nombre: FuenteAlimentacion
    descriptor:

Seleccionar Target (id) - Volver atras (V) - Guardar y salir (G) - Salir sin guardar (S)
Opcion:
```

107: Pantalla descriptores

### Modificar tamaño original

## Trabajo Fin de Grado

---

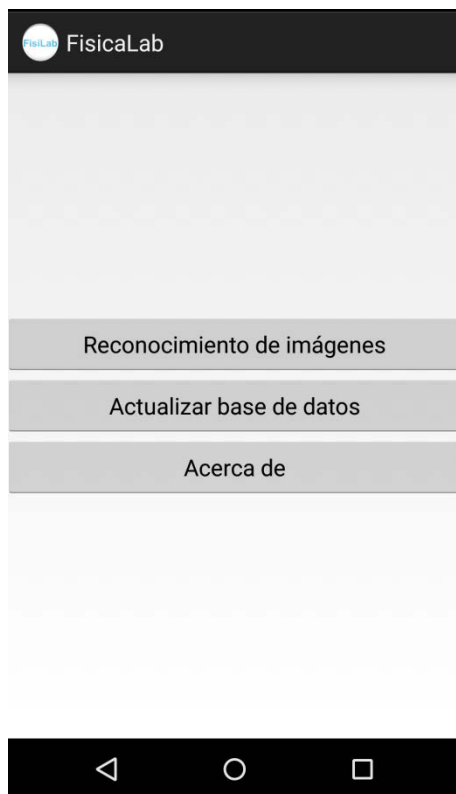
Este caso es muy similar a modificar el descriptor de un Target, la mecánica a la hora de cambiar el valor es la misma, por lo que no se mostrará en imágenes para no aumentar el tamaño del documento en exceso.

### 9.5 Manual usuario aplicación móvil.

A continuación se explicará como un usuario de la aplicación móvil puede utilizar la misma para en el laboratorio de física reconocer los elementos, pulsar los botones virtuales, descargar actualizaciones y ver información referida a la aplicación.

#### Reconocer elementos

Para ello tan solo debemos iniciar la aplicación y nos aparecerá la pantalla inicial como se puede ver, y pulsar el botón "reconocimiento de imágenes".



**108:** Pantalla principal aplicación móvil

Tras pulsar "Reconocimiento de imágenes" la cámara comenzara a buscar y mostrar por pantalla lo que reconozca dibujando los botones virtuales.



# Trabajo Fin de Grado



**109: Reconocimiento**

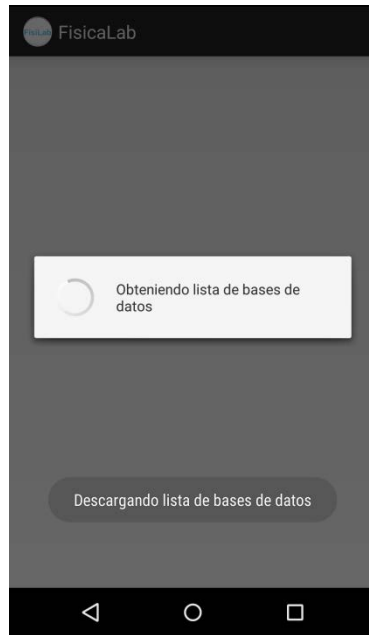
Si se pulsa uno de los botones se mostrará la información referida a este en la base de datos.



**110: Pulsación de botón**

## Descargar base de datos

En la pantalla inicial se pulsa el botón "Actualizar base de datos", mientras se descarga la lista de bases de datos se muestra una animación, y después se muestra el listado completo de bases de datos.

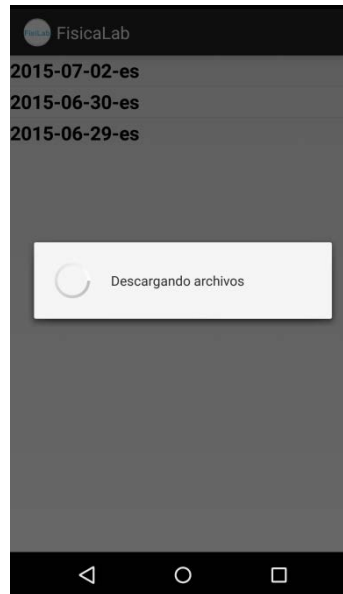


**111:** Animación descargando lista de actualizaciones



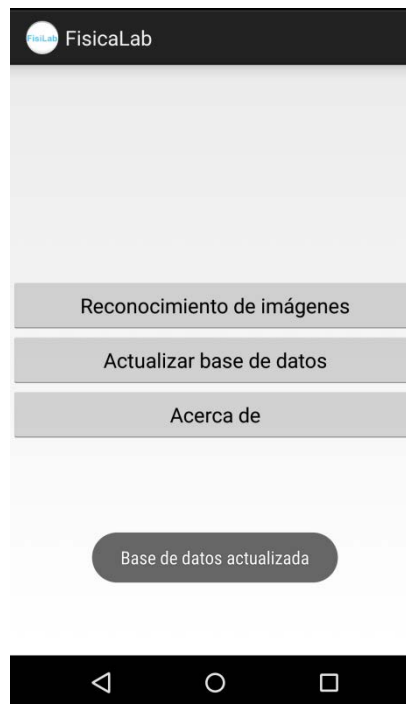
**112:** Lista de actualizaciones

Si se pulsa sobre alguna, se muestra una animación mientras se descarga esa base de datos y se sustituyen los ficheros en el dispositivo.



113: Animación descargando base de datos

Se devuelve a la pantalla inicial tras terminar.



114: Pantalla principal con información

## Ver información

En la pantalla inicial se pulsa el botón "Acerca de", se abre una nueva pantalla en la cual se muestran datos sobre su finalidad, la base de datos en el dispositivo y el desarrollador de la aplicación.



**115:** Pantalla "Acerca de"