



**Universidad de Valladolid**

**ESCUELA DE INGENIERÍA INFORMÁTICA (SG)**

**Grado en Ingeniería Informática de Servicios y  
Aplicaciones**

**LECTOR DE PRESUPUESTOS DE  
LAS UNIVERSIDADES DE  
CASTILLA Y LEÓN**

---

**Alumno: Juan José Palacios de la Flor**

**Tutor: Juan José Álvarez Sánchez**



## Contenido

1.-MOTIVACIONES Y OBJETIVOS .....	7
1.1.- MOTIVACIONES.....	8
1.2.-ÁMBITO.....	10
1.3.-OBJETIVOS .....	11
2.-DESCRIPCIÓN TÉCNICA .....	13
2.1.- ESTRUCTURA.....	14
2.2.- LENGUAJE DE PROGRAMACION .....	16
2.3.- HERRAMIENTAS UTILIZADAS.....	18
3.-GESTION DEL PROYECTO.....	21
3.1.-CICLO DE VIDA Y ORGANIZACIÓN DEL PROYECTO .....	21
3.2.-PLANIFICACIÓN Y PRESUPUESTO .....	24
3.2.1.-PLANIFICACIÓN MEDIANTE PUNTOS DE FUNCION .....	24
3.2.2.-COCOMO .....	28
3.3.-PRESUPUESTO Y PLANIFICACIÓN INICIAL .....	30
3.3.1.-RECURSOS HARDWARE .....	30
3.3.2.-RECURSOS SOFTWARE .....	31
3.3.3.-RECURSOS HUMANOS.....	31
3.3.4.-PRESUPUESTO TOTAL.....	32
3.3.5.-PLANIFICACIÓN INICIAL .....	33
3.4.-SEGUIMIENTO Y CORRECCIONES .....	35
3.4.1.-CORRECCIÓN DE PLANIFICACIÓN Y PRESUPUESTO.....	35
3.4.2.-CORRECCIÓN DE PRESUPUESTO.....	37
4.-ANALISIS .....	39
4.1.-OBJETIVO DEL ANALISIS .....	39
4.2.- ALCANCE DEL ANALISIS .....	39
4.3.- DEFINICION DEL SISTEMA.....	40
4.4.- RESTRICCIONES GENERALES .....	40
4.5 IDENTIFICACION DE LOS USUARIOS .....	41
4.6.-REQUISITOS .....	42
4.6.1.-ESPECIFICACION DE LOS CASOS DE USO .....	42
4.6.2.- ESPECIFICACION DE LOS CASOS DE USO DETALLADO .....	43
4.6.3.-DEFINICIÓN DE LOS REQUISITOS DEL SISTEMA .....	45

4.6.4.-IDENTIFICACIÓN DE LOS REQUISITOS .....	46
5.- DISEÑO .....	55
5.1.- DISEÑO DE LA ARQUITECTURA DE SOPORTE .....	55
5.1.1-ALCANCE .....	55
5.1.2- DISEÑO DE LA ARQUITECTURA DE SOPORTE .....	55
5.2.-DIAGRAMAS DE SECUENCIA.....	57
5.3.- DISEÑO DE CLASES .....	64
5.3.1 DIAGRAMA DE CLASES .....	64
5.3.2.- CLASES, ATRIBUTOS Y METODOS ESTRUCTURALES.....	66
5.5.3.- DESCRIPCION DE CLASES DE CAPA INTERMEDIA (SERVLETS).....	73
5.5.4.- DESCRIPCION DE PÁGINAS DE LA CAPA CLIENTE .....	76
5.6.- DISEÑO FÍSICO DE DATOS .....	78
5.6.1.- DIAGRAMA ENTIDAD- RELACIÓN (E-R) .....	78
5.6.2.- PASO DEL MODELO E-R AL MODELO RELACIONAL.....	80
5.6.3.- MODELO RELACIONAL .....	81
5.6.4.- DICCIONARIO DE DATOS. ....	82
5.7.- INTERFAZ DE USUARIO .....	84
5.7.1.- ACCESO AL SISTEMA .....	84
5.7.2.- PÁGINA PRINCIPAL .....	84
5.7.3.- LEER PRESUPUESTO EN PDF .....	85
5.7.4.- VER PRESUPUESTO LEÍDO/ PROCESADO .....	86
9.7.5.- EXPLICACIÓN DE INTEGRIDAD DE DATOS.....	86
5.7.6.- COMPROBACIÓN DE LA INTEGRIDAD DE LOS DATOS .....	87
5.7.7.- SELECCIÓN DE PRESUPUESTOS A COMPARAR.....	88
5.7.8.- COMPARACIÓN DE PRESUPUESTOS.....	89
5.7.9.- AYUDA AL USUARIO DE LA APLICACIÓN .....	91
5.7.10.- CONFIGURACIÓN DE LA BASE DE DATOS .....	92
5.7.11.-CONFIGURACIÓN DE LOS PARÁMETROS DE LECTURA .....	93
5.7.11.- OTROS.....	94
5.8 DISEÑO LA LECTURA DEL BOLETIN OFICIAL.....	95
5.8.1 FLUJO DE LECTURA DEL PDF .....	95
5.2.3 RELACIONES DE DATOS .....	97
5.8.4 ESTRUCTURA DEL PRESUPUESTO .....	98
5.8.5 PERSISTENCIA DE DATOS .....	100
6.-PLAN DE VALIDACIÓN Y VERIFICACIÓN DEL SOFTWARE .....	103
6.1.-TAREAS Y ELEMENTOS DE PRUEBA .....	103
6.2.- ENTORNO Y FORMATO DE PRUEBAS .....	104
6.3.- PRUEBAS .....	105

7.- CONCLUSIONES Y POSIBLES AMPLIACIONES .....	115
7.1.- CONCLUSIONES .....	115
7.2.-POSIBLES AMPLIACIONES .....	115
8.- BIBLIOGRAFÍA .....	119
9.-ANEXOS .....	121
9.1.-ANEXO CONTROL DE VERSIONES .....	121
9.2 INDICES DE DIAGRAMAS FIGURAS E IMAGENES .....	123
9.2.1.-DIAGRAMAS .....	123
9.2.2.-IMAGENES.....	123
9.2.3.-TABLAS.....	124



# 1.-MOTIVACIONES Y OBJETIVOS

En el presente documento se explica todo lo relativo al TFG “Lector de Presupuestos de Universidad en PDF” desde aspectos técnicos, funcionales, manuales de usuario, etc.

Comenzamos con una breve explicación del trabajo:

El TFG es una aplicación Web capaz de procesar la información que contiene los boletines oficiales publicados por la Junta de Castilla y León que hacen referencia a los presupuestos de las universidades públicas. Esta información procesada puede ser almacenada en una base de datos para su posterior visualización, comparación, descarga....

Me he decantado por la realización de este TFG, ya que me ha parecido bastante interesante, y aunque aplicaciones Web con bases de datos hemos visto y desarrollado muchas a lo largo de la carrera. La innovación consiste en la lectura y procesamiento de los boletines oficiales, y el sistema que asegura la integridad de los datos.

## 1.1.- MOTIVACIONES

Hace no mucho tiempo la consulta de la edición en papel del BOE comportaba un sacrificio de tiempo y esfuerzo notable (el día 31 de diciembre de 2008 se publica el último BOE impreso).

Por entonces los que no trabajaban para la Administración acudían como peregrinos a los Gobiernos Civiles, Ayuntamientos o Bibliotecas públicas para solicitar la consulta a los Boletines, examinarlos cuidadosamente y fotocopiar las partes que interesaban. Normalmente había que esperar colas y no siempre se localizaba el Boletín apetecido, pues bien se facilitaba el lote desordenado o bien incompleto, ya que algún ciudadano impaciente que optaba por arrancar la página apetecida.

Y si se quería un texto legal publicado había que pagar por el mismo y tirarlo a la papelería para comprar el nuevo si el legislador tenía a bien cambiar la norma precedente.



Ahora el BOE ha cambiado hacia una piel moderna y útil. Lo gestiona la Agencia Estatal del BOE y su servicio va más allá con una interesantísima y gratuita oferta. Y es que el artículo 14 del Real Decreto 181/2008,<sup>4</sup> de 8 de febrero, de ordenación del diario oficial *Boletín Oficial del Estado*, establece en su párrafo 1 que:

*Los ciudadanos tendrán acceso libre y gratuito a la edición electrónica del Boletín Oficial del Estado. Dicho acceso comprenderá la posibilidad de búsqueda y consulta del contenido del diario, así como la posibilidad de archivo e impresión, tanto del diario completo como de cada una de las disposiciones, actos o anuncios que lo componen*

Y es que en la actualidad tenemos a nuestra disposición, a golpe de clic todos los boletines oficiales publicados por la administración (estatal o autonómica), en formato pdf. Con las ventajas que nos proporciona la informática y en particular este formato: como la rápida búsqueda, el fácil almacenamiento, Mantenimiento del aspecto y la integridad de los documentos originales.

Pero aun con todas estas ventajas que se han enumerado la verdad es que la lectura de un documento del boletín oficial no es fácil de comprender para un usuario que carezca de conocimientos de derecho o burocráticos.



Esto me hace llegar a la conclusión de que existe una necesidad, que no está cubierta, de extraer la información que se consideren más interesantes del boletín oficial y servirlos en un formato mucho más simple, concreto, que facilite la comprensión de este a los usuarios.

El contenido de los boletines oficiales abarca un grupo amplísimo de campos y temas de donde los que se puede sacar información como: Leyes, reglamentos tratados, concursos, oposiciones, notificaciones, subvenciones, edictos.... Al ser tan variado es necesario. La especificación de algún campo para extraer esta información.

El trabajo que se expone en el siguiente documento se centra en los presupuestos de los boletines oficiales en concreto en los presupuestos de las universidades de Castilla y León.

BOCYL



BOLETÍN OFICIAL DE CASTILLA Y LEÓN



VNIVERSIDAD  
D SALAMANCA

## **1.2.-ÁMBITO**

Como ya se ha explicado el TFG es una aplicación Web subida en un servidor a la que accederán los usuarios desde cualquier navegador Web.

El ámbito de uso es amplio, ya que la aplicación puede ser para uso personal, uso educativo (en la rama de educación económica). Y gracias al sistema de garantía de la integridad de los datos, se podría dar un uso profesional de la información obtenida, en ramas de periodismo y/o otras tesis.

### **1.3.-OBJETIVOS**

El objetivo principal es desarrollar una aplicación capaz de leer la información contenida en los documentos pdf que publica el estado acerca de los presupuestos de las distintas universidades. Procesar esta información almacenarla y ponerla a disposición de los usuarios.

El sistema entonces tendrá dos partes bien diferenciadas, una que se encarga de la elección y extracción de datos del pdf para su almacenamiento; y otra que utilizará los datos almacenados para generar una exposición de estos.

El reto que plantea este trabajo no es tanto la complejidad funcional de la aplicación, sino la posibilidad de componer un trabajo completo con varios módulos y obtener como resultado una aplicación original que puede sentar las bases a varias ampliaciones en este campo.



## 2.-DESCRIPCIÓN TÉCNICA

La arquitectura inicial planteada se basa en la existencia de tres módulos bien diferenciados: Un modulo de lectura y almacenamiento de datos, un modulo de seguridad e integridad de datos y un modulo de exposición de datos. Tal y como se detalla en la FIGURA. A continuación se detalla la finalidad de cada uno de estos módulos.

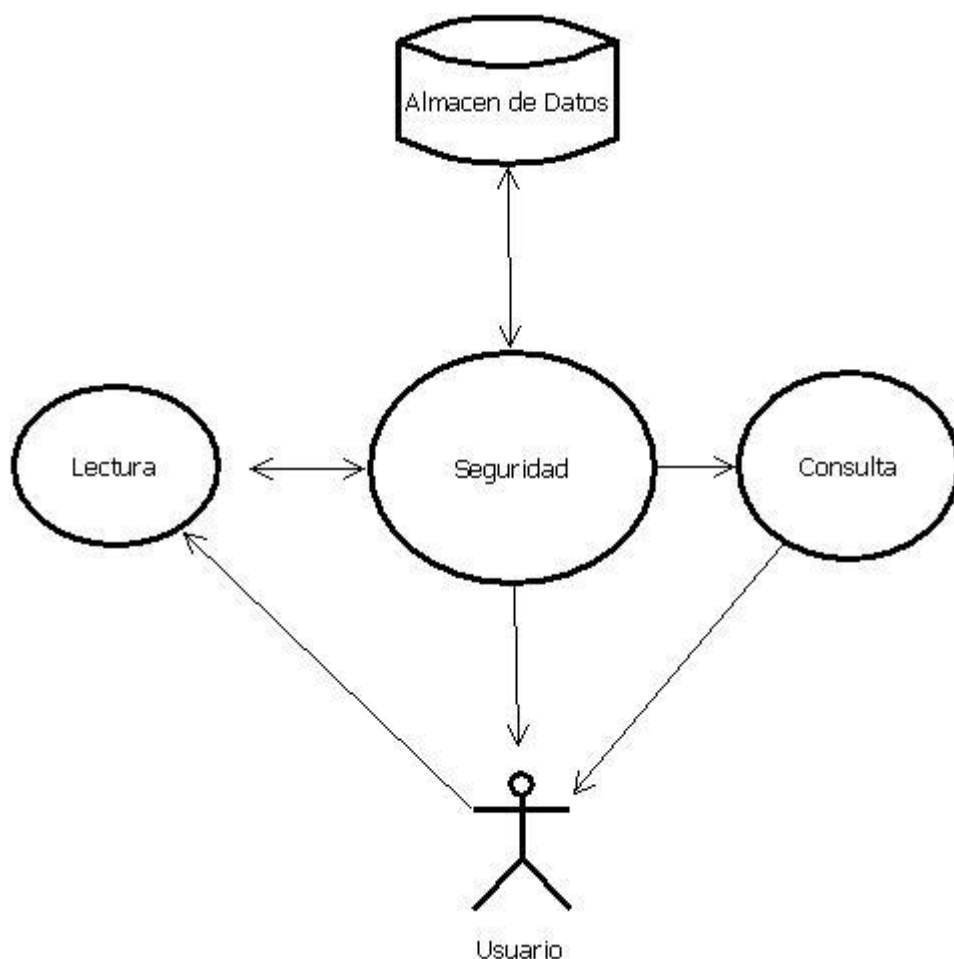


DIAGRAMA Funcionamiento General

El modulo de lectura Obtiene los datos contenidos en los boletines oficiales y almacenarlos en una estructura determinada y tiene como objetivo automatizar en lo posible la obtención del pdf y la lectura de este.

El modulo de seguridad e integridad de datos tiene como objetivo Garantizar mediante un protocolo de seguridad, la integridad del volcado de datos y la integridad de la base de datos.

Modulo de Consulta de Datos, este modulo permite consultar los datos almacenados, que nos los muestra en un entorno sencillo y amable.

## 2.1.- ESTRUCTURA

El punto más complicado del proyecto consiste en la lectura de los boletines oficiales por tanto, las decisiones sobre que metodología, herramientas se han basado en la disponibilidad facilidad y compatibilidad con la lectura de documentos en formato de pdf que es cómo podemos obtener estos boletines de los sitios oficiales.

Se decidió por usar la librería PDFBox que es una librería Java de código abierto que permite trabajar con documentos en formato pdf. Estas librerías permiten acciones relativas a la creación, manipulación y extracción del contenido de documentos en formato pdf.



Se ha elegido esta herramienta por su eficacia sencillez e facilidad de integración con una aplicación.

En cuanto a la arquitectura, se ha seguido una arquitectura **cliente-servidor** propia de las aplicaciones Web, en la que el cliente realiza peticiones a un servidor y éste genera respuestas a esas peticiones.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no tiene porqué ejecutarse únicamente sobre una sola máquina ni es necesariamente un único programa.

La mayoría de los servicios de Internet son tipo de cliente-servidor. La acción de visitar un sitio Web requiere una arquitectura cliente-servidor, ya que, el servidor Web sirve las páginas Web al navegador (al cliente).

El desarrollo de este tipo de programas se basa en una arquitectura de tres capas, donde el núcleo de la aplicación reside en una capa intermedia a la que los usuarios acceder remotamente mediante clientes “ligeros”, encargados de realizar simplemente tareas de captura y presentación de datos.

### Tres CAPAS

Las aplicaciones informáticas basadas en el modelo de tres capas, se caracterizan porque los distintos actores y elementos que la componen se encuentran distribuidos en tres bloques o capas, habitualmente conectadas entre sí a través de una red. Estas capas son:

- Capa cliente
- Capa intermedia
- Capa de datos

### **CAPA CLIENTE**

Se conoce como capa cliente a aquella con la que interactúa el usuario de la aplicación. Es por ello que sus funciones están limitadas básicamente a:

- Captura de datos de usuario y envío de estos a la capa intermedia.
- Presentación de resultados generados por la aplicación.

La función cliente es realizada por un navegador Web, el cual se comunica con la capa intermedia utilizando el **protocolo HTTP**. Los navegadores utilizan las llamadas **páginas Web para interactuar con el usuario**. Estas son enviadas por la capa intermedia al navegador cuando necesita solicitar datos al usuario o presentarle información. Las páginas Web se crean utilizando el lenguaje HTML y el estándar de hoja de estilo CSS, Además se ha usado en esta capa el lenguaje de programación javascript, generándose en muchos casos de forma dinámica por la capa intermedia cuando tiene que presentar al usuario información extraída de la base de datos.

### **LA CAPA INTERMEDIA**

La capa intermedia en una arquitectura de tres capas contiene el núcleo de la aplicación, incluyendo toda la lógica de negocio de la misma. La aplicación se encuentra instalada en una máquina independiente, conocida como servidor, a la que acceden los clientes a través de la red.

Podríamos resumir las funciones de esta capa intermedia de en los siguientes puntos:

- Captura de los datos enviados por el usuario desde la capa cliente.
- Procesamiento de información y, en general, implementación de la lógica de negocio, incluyendo el acceso a los datos.
- Generación de respuestas y envío de las mismas al cliente.

#### **Interacción cliente/capa intermedia**

Capa intermedia en una aplicación Web, ésta puede dividirse a su vez en dos subcapas:

##### **Subcapa Web.**

Esta capa se encarga de la interacción con el cliente, realizando la captura de los datos enviados por éste.

Cuando un cliente quiere solicitar un recurso del servidor (ejecución de una aplicación, página de información, etc.), envía una petición HTTP. Ésta está formada por una cabecera en la que se incluyen una serie de datos, como la dirección del recurso solicitado (URL) además de información relativa a las capacidades y características del navegador cliente.

Una vez procesada la petición, el servidor envía una respuesta HTTP al cliente. En el caso de nuestra aplicación la Capa web se forma con los distintos recursos JSP, Servlets y HTML.

### **Subcapa de negocio.**

Como se desprende de su nombre, esta capa implementa la lógica de negocio de la aplicación, accediendo a las bases de datos.

### **CAPA DE DATOS**

La capa de datos en una arquitectura de tres capas tiene como misión el almacenamiento permanente de los datos utilizados por la aplicación y la gestión de la seguridad de los mismos.

El acceso a los datos desde la aplicación se realizará a través del API JDBC

En un primer momento el sistema de base de datos es Javadb que en realidad es sistema compatible da la bbdd Apache derby, un gestor de bbdd relacional escrito en Java que tiene un tamaño de apenas 2 Mb y es de código abierto. Pero he decidido realizar la adaptación a otras bases de datos, dando la posibilidad de configurar la bbdd a utilizar en la misma aplicación.

De esta manera es posible guardar los datos en otro tipo de base de datos como por ejemplo MySQL, haciendo de la aplicación más versátil, multi-base de datos siempre con el lenguaje SQL.

## **2.2.- LENGUAJE DE PROGRAMACION**

El lenguaje de programación escogido es **Java** porque es un lenguaje que tiene las siguientes características:

**Java** 6 es un lenguaje de programación, de propósito, general orientado a objetos. Además de un lenguaje de programación, java nos proporciona todo un conjunto de especificaciones, tecnologías y librerías de clases, mediante los cuales se pueden crear diferentes tipos de programas informáticos capaces de ser ejecutados en una amplia variedad de sistemas operativos.



Java es un lenguaje simple, orientado a objetos, distribuido, Interpretado, robusto, seguro, de arquitectura neutral, portable, multi-hilo, dinámico y de alto desempeño.



Dentro de Java hemos utilizado las siguientes tecnologías:

### Servlets 3.0

Entre las muchas particularidades que distinguen a la Web de los restantes medios de comunicación, está la capacidad de interacción. Interacción que inicialmente era bastante reducida y se conseguía a través de código HTML y algunos elementos embebidos de JavaScript. Los Servlet son la respuesta de la tecnología en Java a la programación de la Interfaz de Compuerta Común (CGI). Son programas que se ejecutan en el servidor, realizando la función de una capa intermedia entre una petición proveniente de un navegador Web u otro cliente HTTP, y las aplicaciones del servidor, pudiendo utilizar toda la paquetería y potencialidades del lenguaje. Su función principal es proveer páginas web dinámicas y personalizadas, utilizando para este objetivo accesos a bases de datos, flujos de trabajo y otros recursos. Para poder utilizar los Servlets dentro de nuestras aplicaciones web, debemos, por norma general, complementar éste con un contenedor de Servlets, que no es más que un servidor que tenga soporte para el API Servlet. En este caso Se ha decidido por Tomcat.



### JSP 2.3:

Las Java Server Pages, o páginas JSP como se les llama habitualmente, son un tipo de componente que forma parte de la capa intermedia de una aplicación Web J2EE, concretamente, dentro de la subcapa WEB. Aunque se trate de un componente que, aparentemente, realiza las mismas funciones que un servlet, esto es, capturar datos del cliente y generar dinámicamente respuestas, la tecnología JSP no representa ningún elemento redundante. Más al contrario, ambas tecnologías se complementan perfectamente.



En una página JSP se combinan bloques de texto HTML (o cualquier otro lenguaje de marcado) con código Java de servidor. Estos archivos forman parte de la aplicación Web.

Esto significa que el código Java es ejecutado en el servidor en el momento en que la página es solicitada por el cliente, enviándose como respuesta al cliente la página resultante de combinar el bloque HTML fijo que aparece en la página JSP con la parte generada de forma dinámica.

Librerías Java utilizadas:

- PDFBox 1.8.7: como hemos comentado en el punto anterior se ha usado PDFBOX. Es una librería de código abierto que permite trabajar con documentos en formato PDF. Estas librerías permiten acciones relativas a la creación, manipulación y extracción del contenido en formato pdf.
- jFreeChart 1.0.19 Esta es una librería gratuita para el lenguaje de programación, que permite la creación de gráficos de forma muy sencilla. jFreeChart utiliza también la librería jcommon 1.0.23.



### HTML 4.01

Hyper Text Markup Language es el lenguaje que se emplea para escribir páginas web. Es un lenguaje de Hipertexto, es decir, un lenguaje que permite escribir texto de una forma estructurada, y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento. Estos documentos pueden ser visualizados en los navegadores que se

encargan de interpretar el código HTML y mostrar a los usuarios las páginas web resultantes.

### **CSS 2.1**

Hojas de estilo style sheets son hojas que contienen una colección de reglas que afectan a la apariencia del documento. Estas reglas se refieren al modo en que aparecerá un documento en pantalla cuando un usuario lo cargue en el navegador, controlando colores, márgenes, tamaños, fuentes....



### **JAVAScript. 1.7**

JavaScript no es un lenguaje de programación propiamente dicho, sino un lenguaje de *scripts*. Hay otros lenguajes de script. Con JavaScript daremos respuesta a eventos de usuario, como entrada de datos por ejemplo. Para hacer este tipo de verificaciones no es necesario enviar ningún tipo de información al servidor. Se procesa todo en el lado del cliente, en el navegador web.

### **Google Charts. 40**

Es una API en javascript que nos permite visualizar datos en la Web, La galería nos provee de varios tipos de gráficos diferentes. Gráficos, de barras de Área de tarta....

### **TOMCAT**



Es un servidor de aplicaciones de software libre, que implementa las tecnologías definidas en la plataforma JavaEE y permite ejecutar aplicaciones definidas que siguen esta especificación.

## **2.3.- HERRAMIENTAS UTILIZADAS**

Para el desarrollo del TFG se han utilizado las siguientes herramientas:

#### **Hardware:**

- Un PC con las características mínimas actuales e una impresora.

#### **Software:**

- Windows 7: Sistema operativo de 64 bits sobre el que se ha basado el desarrollo del TFG.
- NetBeans 8.0.2: Es un IDE (Entorno Integrado de Desarrollo) que permite desarrollar aplicaciones, contiene varias librerías y herramientas que facilitan la programación.
- Mercurial 3.5.1: Es un sistema de control de versiones multiplataforma para desarrolladores de software.



- Wampserver 1.6.1.33 Utilizado para la aplicación Web. Es un entorno de desarrollo Web de Windows. Permite crear aplicaciones Web con Apache, PHP y una base de datos MySQL. Utiliza Windows como sistema operativo, MySQL como gestor de base de datos. Se ha usado para la integración de la base de Datos en el desarrollo.
- DIA 0.97.2: Es un programa de propósito general para la creación de diagramas
- STARUML 2.5.1.0: Es otro programa para la creación de diagramas UML.
- Microsoft Office 2007: Herramienta propia del sistema operativo Windows utilizada para el desarrollo de la documentación de este TFG.



## **3.-GESTION DEL PROYECTO**

### **3.1.-CICLO DE VIDA Y ORGANIZACIÓN DEL PROYECTO**

En primer lugar se definirá este concepto para seguidamente explicar que actividades se realizan en cada una de las fases que establece el modelo elegido para finalmente justificar la elección del ciclo de vida.

Se entiende por ciclo de vida software el conjunto de fases, procesos y actividades requeridas para ofertar, desarrollar, probar, integrar, explotar y mantener un producto software indicando como funciones principales la de por un lado determinar el orden de las fases y procesos y por otro el de establecer los criterios de transición para pasar de una fase a otra.

Para el presente proyecto el ciclo de vida elegido ha sido el modelo “Modelo Iterativo”, que es el que más se adapta a las necesidades de desarrollo en este caso, ya que el propio producto a lo largo de su desarrollo puede considerarse una sucesión de versiones que progresan hasta llegar al estado deseado. En cada versión las especificaciones se van resolviendo paulatinamente, completando las del versión anterior y con mayores funcionalidades del producto. Estas iteraciones se repetirán hasta obtener un producto completo.

En cada iteración se completan varias fases, que se explican brevemente a continuación:

- **Análisis:** En esta fase se determinan (o amplían) los requisitos software del sistema, especificando a un alto nivel la arquitectura de la solución que se propone para dichos requisitos.
- **Diseño:** En esta segunda fase, se diseñan tanto los interfaces de usuario de la aplicación, como la arquitectura a un nivel más bajo de especificación, detallando los procesos del sistema.
- **Implementación:** En esta etapa se codifican los interfaces de usuario, se codifican los procesos, y se documenta el manual de usuario.
- **Pruebas:** En esta última fase, se definen las pruebas a realizar por el prototipo en cuestión y se llevan a cabo.

Este ciclo se ha considerado el más adecuado debido a los siguientes motivos:

- Permite evaluar distintas posibilidades de desarrollo, ya que en las versiones se puede fácilmente añadir o eliminar funcionalidades.
- Permite al tutor o cliente realizar un seguimiento periódico del proyecto para verificar si se han comprendido bien los requisitos y si se está desarrollando la aplicación de acuerdo con éstos.
- Experimenta sobre los aspectos del sistema que representan mayor complejidad.

- El usuario reacciona mucho mejor ante el prototipo, sobre el que puede experimentar, que no sobre una especificación escrita.
- Se incrementa la calidad del producto final, ya que el prototipo permite trabajar.

A la hora de llevar a cabo cualquier proyecto, se hace necesaria la tarea de realizar una planificación, detallando el tiempo estimado necesario para cumplir las distintas actividades y tareas que forman parte del proyecto, así como los recursos asignados a cada una de ellas. Por ello, se ha desarrollado una planificación de trabajo en función del ciclo de vida seleccionado anteriormente, ciclo de vida por modelo iterativo.

En lo que se refiere a los recursos humanos que intervienen en el desarrollo del proyecto, a continuación se muestran las personas y roles que participan en el proyecto. Las funciones de las que se encarga cada uno de ellos son:

- Jefe de Proyecto: Se encarga de la gestión del proyecto, su organización, planificación y supervisión a lo largo de todo el desarrollo del mismo.
- Analista: Se encarga de obtener y redactar los requisitos de usuario, además de modelar los procesos y tareas a codificar.
- Diseñador: Su tarea es el diseño de las interfaces, la arquitectura del sistema y el plan de verificación y validación.
- Programador: Se encarga de la codificación del sistema, así como de llevar a cabo las pruebas necesarias sobre el mismo.

El RBS (Resource Breakdown Structure) es una representación jerárquica de los recursos, tanto humanos como materiales, necesarios para la planificación de un proyecto. Esta técnica de organización de proyectos software tiene por objeto representar la organización humana del proyecto, así como la estructura de recursos tecnológicos y materiales.

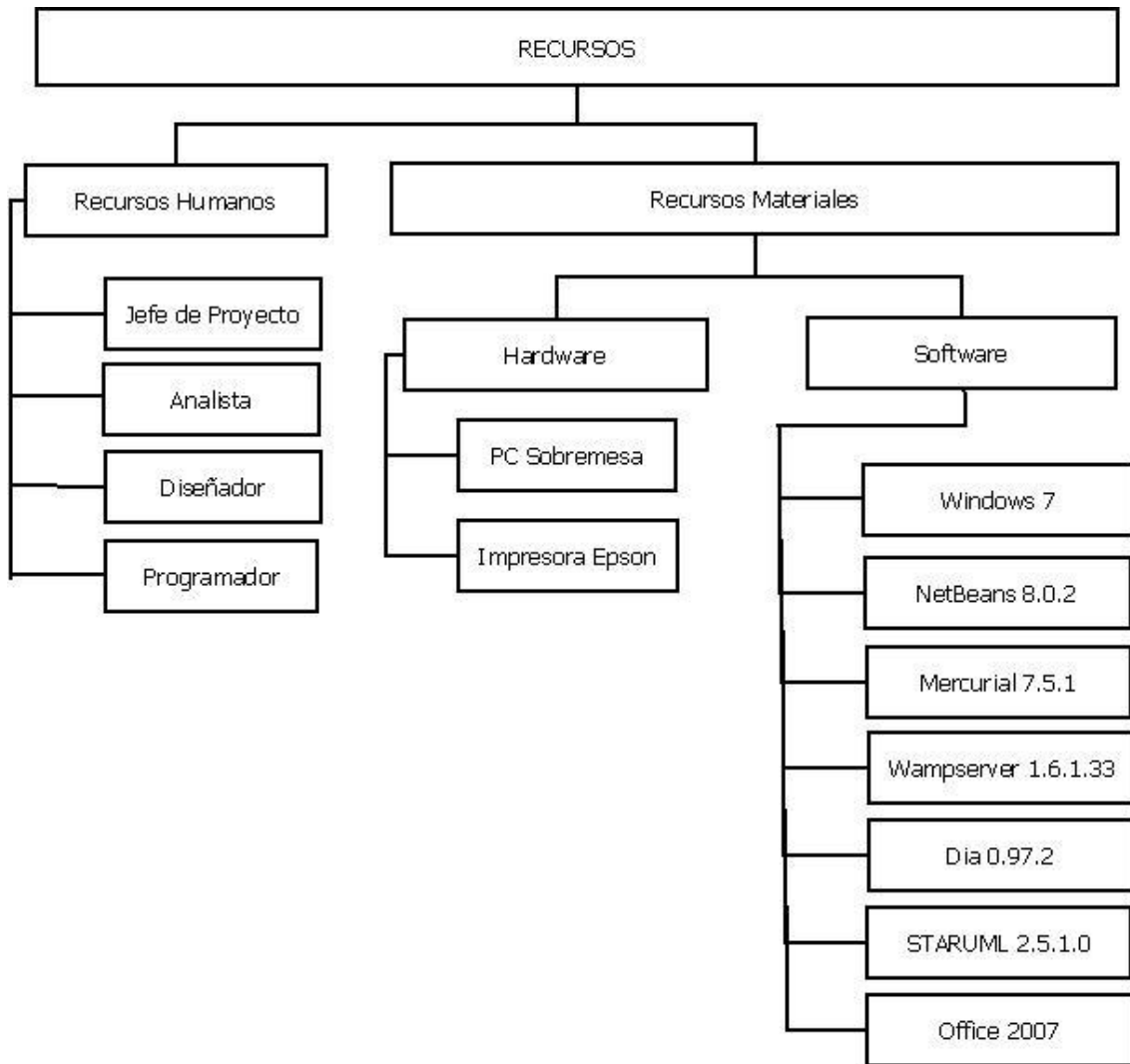


DIAGRAMA RBS

## 3.2.-PLANIFICACIÓN Y PRESUPUESTO

La estimación del coste y el esfuerzo son muy complejos, no es una ciencia exacta, es una predicción que se basa en un modelo probabilista. Es complejo porque son demasiados los elementos variables que pueden afectar al coste del software y el esfuerzo para desarrollarlo. Aun así la estimación del coste del proyecto software proporciona una aproximación con un grado de riesgo aceptable.

Dividimos en dos apartados la estimación del coste del proyecto, por un lado la estimación del esfuerzo y por otro la estimación de los recursos.

Cabe destacar que en el apartado de estimación del esfuerzo tendremos dos tipos de cálculo del esfuerzo, una con el modelo COCOMO y otro desde un enfoque diferente, teniendo en cuenta el tamaño de componentes de la aplicación desde la experiencia, lo que se conoce como estimación por analogía con otros proyectos ya realizados sobre el mismo dominio de aplicación.

El presupuesto se ha realizado utilizando la estimación por puntos de función (PF) y mediante COCOMO. El análisis del punto de función es utilizado como alternativa a la medida del tamaño de una aplicación informática en líneas de código, independiente también de la tecnología empleada. Los puntos de función se determinan mediante la estimación del número de entradas, salidas, archivos de datos, consultas e interfaces.

Las métricas usadas para estimar el tamaño del proyecto software deben ser razonablemente fáciles de usar en etapas tempranas del proyecto y fácilmente medibles una vez terminado éste para que las comparaciones entre estas situaciones inicial y final nos provean de retroalimentación para hacer estimaciones más precisas en futuros proyectos.

### 3.2.1.-PLANIFICACIÓN MEDIANTE PUNTOS DE FUNCION

La estimación por puntos de función (PF) es la siguiente:

Los valores de los dominios de información y su complejidad se definen de la siguiente manera:

- ❖ Número de entradas de usuario, se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación:
  - URL: Complejidad simple.
  - DNI usuario: complejidad Simple.
  - Información y datos de conexión con bbdd: Complejidad Simple
  - Información y parámetros de lectura del PDF: Complejidad Simple

Entradas de usuario: 4 (4 simples)

- ❖ Número de salidas de usuario, se cuenta cada salida que proporciona al usuario información orientada a la aplicación, informes, pantallas, mensajes de error...



- Pantalla principal de la aplicación: Complejidad Simple.
- Pantalla de lectura de la URL: Complejidad Simple.
- Pantalla de muestra del presupuesto procesado: Complejidad Media.
- Pantalla de explicación de integridad: Complejidad Simple
- Pantalla interactiva de integridad: Complejidad media
- Pantalla de Ayuda: Complejidad simple
- Listado de Presupuestos: Complejidad Simple.
- Pantalla de configuración de bbdd: Complejidad Simple
- Pantalla de configuración de parámetros: Complejidad Simple.
- Pantalla de comparación de presupuestos: Complejidad Media.

Salidas de Usuario 10 (3 de complejidad Media y 7 simple).

- ❖ Número de consultas de usuario, se cuenta cada entrada interactiva que genera alguna respuesta software inmediata en forma de salida interactiva
  - Despliegue detallado de presupuesto procesado: Complejidad simple.
  - Presupuestos escogidos para su comparación. Complejidad simple
  - Despliegue de detalles en la comparación de presupuestos Complejidad Simple.
  - Información para obtener el código HASH.
  - Información para obtener la página aleatoria.

Consultas de usuario 5 (5 simple)

- ❖ Número de ficheros internos: Se cuenta cada archivo maestro lógico, esto es, un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente.
  - Base de Datos: Complejidad alta
  - Fichero de configuración de bbdd: Complejidad simple
  - Fichero de configuración de parámetros de lectura: Complejidad Simple.

Ficheros internos 3 (2 de complejidad simple y uno de alta complejidad)

- ❖ Número de ficheros externos: Se cuentan todas las interfaces legibles por la máquina, que se utilizan para transmitir información a otro sistema.
  - Presupuesto en formato PDF: Complejidad alta.
  - Datos exportados en formato csv. Complejidad alta.
  - Página aleatoria en formato PDF. Complejidad Alta.
  - Fichero de texto con todo el pdf transformado. Complejidad simple
  - Fichero de texto con solo la página aleatoria (obtenida desde el PDF). Complejidad simple
  - Fichero de texto con solo la página aleatoria (obtenida desde el documento de texto completo). Complejidad simple
  - Fichero de texto con líneas aleatorias obtenidas del documento de texto. Complejidad simple
  - Fichero de texto con líneas aleatorias obtenidas a partir de la estructura del presupuesto procesado. Complejidad simple
  - Fichero de texto con datos aleatorios obtenidos a partir de la bbdd Complejidad simple

- Fichero de texto creado a partir de los datos del presupuesto procesado.  
Complejidad simple

Ficheros externos 10 (3 de alta complejidad y 7 simple)

Una vez identificadas y clasificadas las funciones vamos a proceder a calcular sus puntos de función sin ajustar PFNA:

TIPO DE FUNCION	COMPLEJIDAD	TOTAL X COMPLEJIDAD	TOTAL TIPO	TOTAL	SUMA
Ficheros internos	Simple	X7	2	14	29
	Media	X10	0	0	
	Alta	X15	1	15	
Ficheros externos	Simple	X5	7	35	69
	Media	X7	0	0	
	Alta	X10	3	30	
Entradas usuario	Simple	X3	4	12	12
	Media	X4	0	0	
	Alta	X6	0	0	
Salidas usuario	Simple	X4	7	28	43
	Media	X5	3	15	
	Alta	X7	0	0	
Consultas usuario	Simple	X3	5	15	15
	Media	X4	0	0	
	Alta	X6	0	0	
<b>TOTAL PUNTOS DE FUNCION NO AJUSTADOS</b>					<b>168</b>

Tabla: Correspondencia PFNA

Una vez obtenidos los PFNA, deben ser ajustados mediante un Factor de Ajuste (FA). El cálculo del factor de ajuste está basado en 14 características generales de los sistemas que miden la funcionalidad general y complejidad de la aplicación. A cada característica se le atribuye un peso de 0 a 5 que indica el grado de complejidad que tiene cada característica.

Grado	Descripción
0	No está presente o su complejidad no es tenida en cuenta
1	Complejidad mínima
2	Complejidad moderada
3	Complejidad promedio
4	Complejidad significativa
5	Complejidad fuerte

Tabla Factor de Ajuste

Calculo del grado de cada característica para obtener el Factor de ajuste.

Factores de Complejidad: FC	Nivel
Comunicación de datos	2
Rendimiento	0
Frecuencia de transacciones	2
Interface del usuario final	3
Procesos complejos	0
Facilidad de mantenimiento	0
Instalación en múltiples lugares	1
Procesamiento distribuido	2
Gran carga de trabajo	0
Entrada on-line de datos	4
Actualizaciones on-line	0
Uso de otros sistemas. Reusabilidad	0
Facilidad de operación	3
Facilidad de cambios	1
<b>TOTAL</b>	<b>18</b>

Tabla Factores de complejidad

Calculo del Factor de Ajuste F.A.

$$F.A = (0,01 \times FC) + 0,65$$

$$F.A = (0,01 \times 18) + 0,65 = 0,83$$

Calculo de los Puntos de Función PF

$$PF = PFNA \times FA$$

$$PF = 168 \times 0,83 = 139,44$$

Debido a que la aplicación se desarrollará en lenguaje JAVA y sabemos que para cada punto de función equivale a 53 líneas de código (LOC) obtenido de la tabla propuesta por Caper T. Jones:

Líneas de código =  $139,44 \times 53 = 7390,32 \rightarrow 7,39032$  KLOC (Kilo líneas de código)

### 3.2.2.-COCOMO

Cocomo es el acrónimo de Modelo Constructivo de Costos (CONstructive COSt MOdel) es un modelo matemático utilizado para la estimación de costos software.

Proyecto Software	a	e	c	d
<b>Orgánico</b>	3,2	1,05	2,5	0,38
<b>Semi-acoplado</b>	3,0	1,12	2,5	0,35
<b>Empotrado</b>	2,8	1,20	2,5	0,32

TABLA COCOMO MODELO

Nuestro proyecto se clasifica como modo Orgánico, porque es un proyecto desarrollado en un entorno estable con pocas presiones de plazos de tiempo y relativamente pequeño.

Así pues las fórmulas serán las siguientes:

- **E = Esfuerzo = a KLDC<sup>e</sup> \* FAE** (persona x mes)
- **T = Tiempo de duración del desarrollo = c Esfuerzo<sup>d</sup>** (meses)
- **P= Personal = E/T** (personas)

$$\text{Esfuerzo nominal} = 3,2 \times 7,39032^{1,05} = 26,14 \text{ personas} - \text{mes}$$

Para realizar el modelo necesitamos también el factor de complejidad total, que se obtiene mediante la multiplicación de los valores evaluados en 15 factores específicos de coste que se observan en la siguiente tabla.

Atributos	Valor					
	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra
<b>Atributos software</b>						
Fiabilidad	0,75	0,88	1	1,15	1,40	
Tamaño base de datos		0,94	1	1,08	1,16	
Complejidad	0,70	0,85	1	1,15	1,30	1,65
<b>Atributos hardware</b>						
Restricciones de tiempo de ejecución			1	1,11	1,30	1,66
Restricciones de memoria virtual			1	1,06	1,21	1,56
Volatilidad de la maquina virtual		0,87	1	1,15	1,30	
Tiempo de respuesta		0,87	1	1,07	1,15	
<b>Atributos Personal</b>						
Capacidad de análisis	1,46	1,19	1	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1	0,91	0,82	
Calidad de los programadores	1,42	1,17	1	0,86	0,70	
Experiencia en la maquina virtual	1,21	1,10	1	0,90		
Experiencia en el lenguaje	1,14	1,07	1	0,95		
<b>Atributos Proyecto</b>						
Técnicas actualizadas de programación	1,24	1,10	1	0,91	0,82	
Utilización de software	1,24	1,10	1	0,91	0,83	
Restricciones de tiempo desarrollo	1,22	1,08	1	1,04	1,10	

TABLA COCOMO FACTORES ESPECIFICOS DE COSTE

Para aplicar el COCOMO intermedio he aplicado los siguientes factores de coste:

- Un tamaño de bbdd pequeño. Factor 0,94
- Muy alta fiabilidad (los datos deben ser fiables) Factor 1,40
- Bajo tiempo de respuesta Factor 0,87
- Considero tener una buena capacidad de análisis. Factor 0,86
- Considero buena la calidad de programación. Factor 0,86
- Poca experiencia en estas aplicaciones. Factor 1,13
- Poca experiencia con la maquina virtual. Factor 1,10
- Se usarán técnicas de programación modernas, plantillas y demás por lo que aplicamos un factor de coste 0,91
- El nivel en cuanto a restricciones de tiempo será bajo Factor 1,08

Así, para calcular el factor de ajuste del esfuerzo (FAE) se multiplican los valores:

$$FAE = \prod_{k=1}^n \text{factor } k = 0,94 * 1,40 * 0,87 * 0,86 * 0,86 * 1,13 * 1,10 * 0,91 * 1,08 = 1,034$$

$$\text{Esfuerzo} = 26,14 * 1,034 = \mathbf{27,03 \text{ personas- mes}}$$

$$\text{Tiempo} = 2,5 * 27,03^{0,38} = \mathbf{8,75 \text{ meses}}$$

Productividad= LDC/Esfuerzo = 7390,32/27,03 = 273,41 Líneas de código/persona-mes

$$\mathbf{P = E/T = 27,03 / 8,75 = 3,48 \text{ personas}}$$

Según estos datos se necesitarían un mínimo de 4 personas para realizar el proyecto en menos de 9 meses. Dado que es un proyecto individual el tiempo estimado de realización del proyecto es de 27,03 meses.

Puesto que un mes tiene 160 horas de trabajo como media la duración del proyecto es de **4325 horas**.

### 3.3.-PRESUPUESTO Y PLANIFICACIÓN INICIAL

Para desarrollar el proyecto se necesitan medios Hardware y Software, cuyo coste proporcional al uso se debe ver reflejado en el presupuesto. De igual manera se incluye el coste de los recursos humanos utilizados.

#### 3.3.1.-RECURSOS HARDWARE

RECURSO	COSTE (€)	USO(%)	COSTE FINAL (€)
Ordenador Personal	800	25	200
Conexión a Internet	250	20	50
Impresora	70	15	10,5
<b>TOTAL</b>			<b>260,5</b>

TABLA presupuestos recursos HW

### 3.3.2.-RECURSOS SOFTWARE

RECURSO	COSTE (€)	USO (%)	COSTE FINAL (€)
Windows 7	40	15	6
WampServer	0	-	0
IDE NetBeans	0	-	0
Mozilla Firefox	0	-	0
Microsoft Office	69	15	10,35
Acrobat Reader	0	-	0
GanttProject	0	-	0
StarUML	0	-	0
Mercuria	0	-	0
DIA	0	-	0
Tomcat	0	-	0
Gelastic LunaCloud	0	-	0
<b>TOTAL</b>			<b>16,35</b>

TABLA PRESUPUESTO RECURSO SW

### 3.3.3.-RECURSOS HUMANOS

A partir de las estimaciones de esfuerzo y tiempo calculados en el apartado anterior, calcularemos los recursos humanos que se utilizarán. Cabe destacar que el proyecto lo realizará 1 persona que alternará sus labores como programador, analista, diseñador.

Se estima la duración de cada tarea del proyecto en horas tomando como referencia la estimación anteriormente realizada con el modelo COCOMO. El resultado no se transforma en meses porque el tiempo dedicado cada mes o semana viene siendo muy irregular. Dicha estimación daba como resultado un total de:

TAREA	%	DURACION(HORAS)
Requisitos del sistema	5	216,25
Análisis del sistema	10	432,5
Diseño del sistema	15	648,75
Implementación	40	1730
Pruebas	15	648,75
Documentación	5	216,25
Sobrecarga	10	432,5
<b>TOTAL</b>		<b>4325</b>

TABLA Duración prevista de tareas

Sueldos para los integrantes del proyecto:

	Tiempo (horas)	Sueldo (€/hora)	Coste (€)
<b>Analista/Diseñador</b>	1297,5	15	19.462,5
<b>Programador</b>	2162,5	10	2.162,5
<b>Probador</b>	648,75	10	6.487,5
<b>Documentalista</b>	216,25	10	2.162,5
<b>TOTAL</b>			49.737,5

TABLA Coste Previsto de Tareas

### 3.3.4.-PRESUPUESTO TOTAL

El Presupuesto total estará formado por la suma de los costes de los recursos humanos y los materiales:

Recurso	Coste
<b>Recursos Humanos</b>	49.737,5
<b>Recursos Software</b>	16,35
<b>Recursos Hardware</b>	260,5
<b>TOTAL</b>	50.014,35

TABLA Coste total previsto



### **3.3.5.-PLANIFICACIÓN INICIAL**

Se estima (Según COCOMO) que la duración total del proyecto es de 4325 horas comenzando su desarrollo el día 10 de octubre de 2014 estimando que se trabajaría 8 horas por día laborable de la semana. La fecha de finalización se estima que sería 540 días. El 20 de junio de 2016, cuando el sistema se encuentre finalizado y documentado para su entrega al cliente.

Afortunadamente, por experiencia sabemos que el modelo COCOMO suele estar muy sobreestimado, por lo que los tiempos y los costes reales serán menores a los estimados.

## DIAGRAMA DE GANTT INICIAL

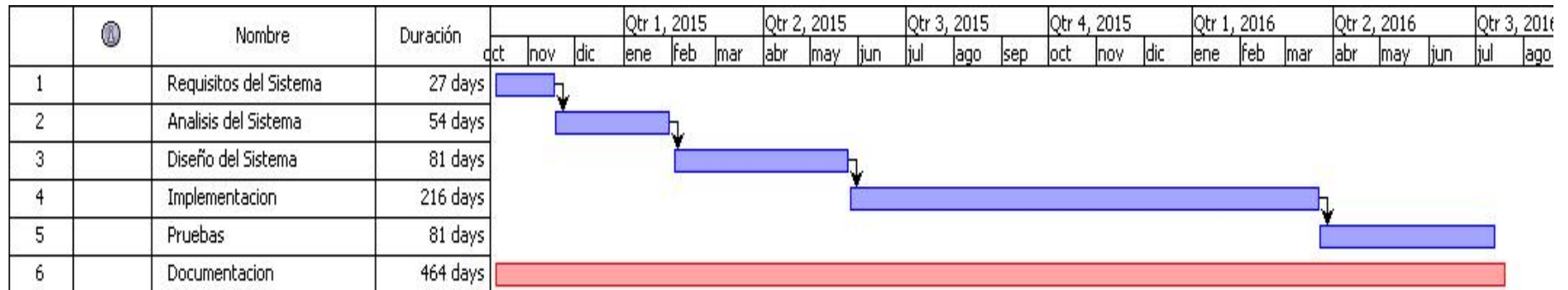


Diagrama de Gantt Inicial

## **3.4.-SEGUIMIENTO Y CORRECCIONES**

En este apartado vamos a analizar el seguimiento real del plazo del proyecto. Esto se ha realizado con la ayuda del programa de gestión openProj.

### **3.4.1.-CORRECCIÓN DE PLANIFICACIÓN Y PRESUPUESTO**

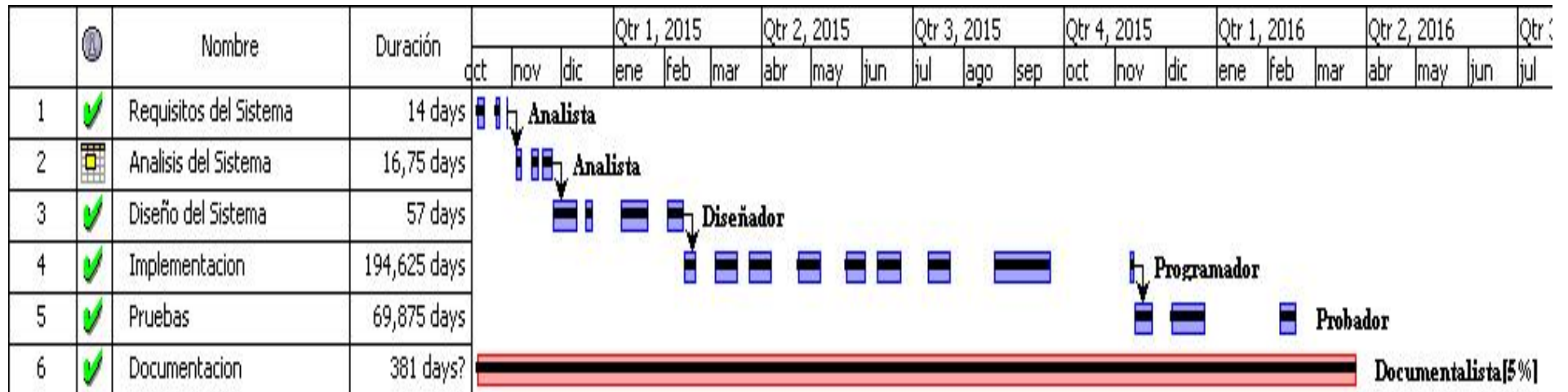
Como hemos señalado en el apartado anterior los resultados distan en mucho de la estimación realizada con COCOMO.

El mayor problema ha sido que debido a circunstancias de disponibilidad de tiempo, la realización del proyecto ha sido mediante periodos cortos y discontinuos, lo que hacía más laborioso la vuelta al trabajo al tener que volver a ponerse al día para retomar la labor en el punto donde se había quedado la última vez.

Diagrama de Gantt Final.

Con la fecha de inicio el día 10 de octubre de 2014 y la fecha de finalización el 28 de marzo de 2016.

## DIAGRAMA DE GANT CORREGIDO



### 3.4.2.-CORRECCIÓN DE PRESUPUESTO.

TAREA	%	DURACION(HORAS)
Requisitos del sistema	4,8	76
Análisis del sistema	7,7	122
Diseño del sistema	19	300
Implementación	49,5	780
Pruebas	15,2	240
Documentación	3,8	60
<b>TOTAL</b>		<b>1578</b>

TABLA Duración corregida de tareas

	Tiempo (horas)	Sueldo (€/hora)	Coste (€)
Analista/Diseñador	498	15	7.470,00
Programador	780	10	7.800,00
Probador	240	10	2.400,00
Documentalista	60	10	600,00
<b>TOTAL</b>			<b>18.270,00</b>

TABLA Coste corregido de Recursos humanos

Recurso	Coste
Recursos Humanos	18.270
Recursos Software	16,35
Recursos Hardware	260,5
<b>TOTAL</b>	<b>18.546,85</b>

TABLA coste total corregido



## **4.-ANALISIS**

### **4.1.-OBJETIVO DEL ANALISIS**

Al realizar el Análisis del Sistema se pretende obtener una colección completa y detallada de los requisitos del sistema, tomando como punto de partida los requisitos identificados en el Estudio de Viabilidad del Sistema.

El documento generado en esta fase será la base para el desarrollo de la fase de Diseño del Sistema, en el que se especificará el diseño completo del sistema de información.

### **4.2.- ALCANCE DEL ANALISIS**

Esta fase tiene como objetivo obtener una especificación detallada del sistema que se va a diseñar. Mediante su producto, el Documento de Análisis del Sistema, se pretende captar cuales son las necesidades que tiene el cliente.

En primer lugar, se definirá cual es el alcance del sistema que se desea desarrollar, así como el entorno tecnológico asociado al proyecto. Además, se identificarán los diferentes participantes que aparecen a lo largo de la vida del proyecto así como los usuarios finales.

A continuación, se definirán cuales son los requisitos software que debe cumplir el sistema a desarrollar, tomando como punto de partida los casos de uso y los requisitos de usuario.

Por último, se definirán las interfaces de usuario que se utilizarán, dejando como última tarea las comprobaciones de calidad sobre los distintos modelos y requisitos software que se han generado durante la fase de análisis.

### **4.3.- DEFINICION DEL SISTEMA**

En este apartado se determina el alcance del sistema a desarrollar, para que permita satisfacer las necesidades planteadas.

El sistema a desarrollar consiste en una aplicación Web que permita procesar los documentos PDF en los que se publican los presupuestos de las universidades.

Como pieza principal en el desarrollo de este TFG se pretende desarrollar una aplicación Web que cumpla con las siguientes características:

- Leer Presupuestos de universidades en PDF.
- Guardar datos de estos presupuestos.
- Comparar Presupuestos.
- Exportar Datos.

Asimismo de forma complementaria se debe desarrollar un sistema que permita garantizar y/o comprobar la integridad de los datos procesados de los presupuestos leídos y/o guardados.

### **4.4.- RESTRICCIONES GENERALES**

A continuación se detallan las restricciones que deberá cumplir el sistema a diseñar:

En un primer lugar la aplicación funcionara en cualquier S.O. solo será necesario un navegador para poder utilizarla.

La interfaz, de la aplicación Web, con la que interaccionará el usuario final, debe proporcionar todas las funcionalidades descritas en los requisitos software del sistema, de una forma amigable e intuitiva, evitando posibles ambigüedades que puedan ocasionar confusión al usuario.

El idioma usado en el desarrollo del sistema será el castellano.



## 4.5 IDENTIFICACION DE LOS USUARIOS

En este apartado se van a detallar los usuarios que participan en el proceso de análisis del sistema, así como los usuarios que lo validarán y aceptarán finalmente.

En primer lugar, en el proceso de análisis van a participar:

- El desarrollador: La persona encargada de llevar a cabo el desarrollo del presente proyecto
- Cliente o tutor: En este caso tutor, que espera un correcto desarrollo del producto final en términos de calidad tiempo y coste Además, son parte importante a la hora de la obtención de los requisitos del sistema a diseñar, con el objetivo de obtener un producto que cumpla con las necesidades que le han llevado a solicitar el sistema.

En cuanto a los usuarios finales del sistema, éstos serán los que se describen a continuación:

- Usuarios: Son aquellas personas que harán uso del sistema desarrollado. Se encargarán tanto de la configuración del sistema como de su puesta en producción.

## 4.6.-REQUISITOS

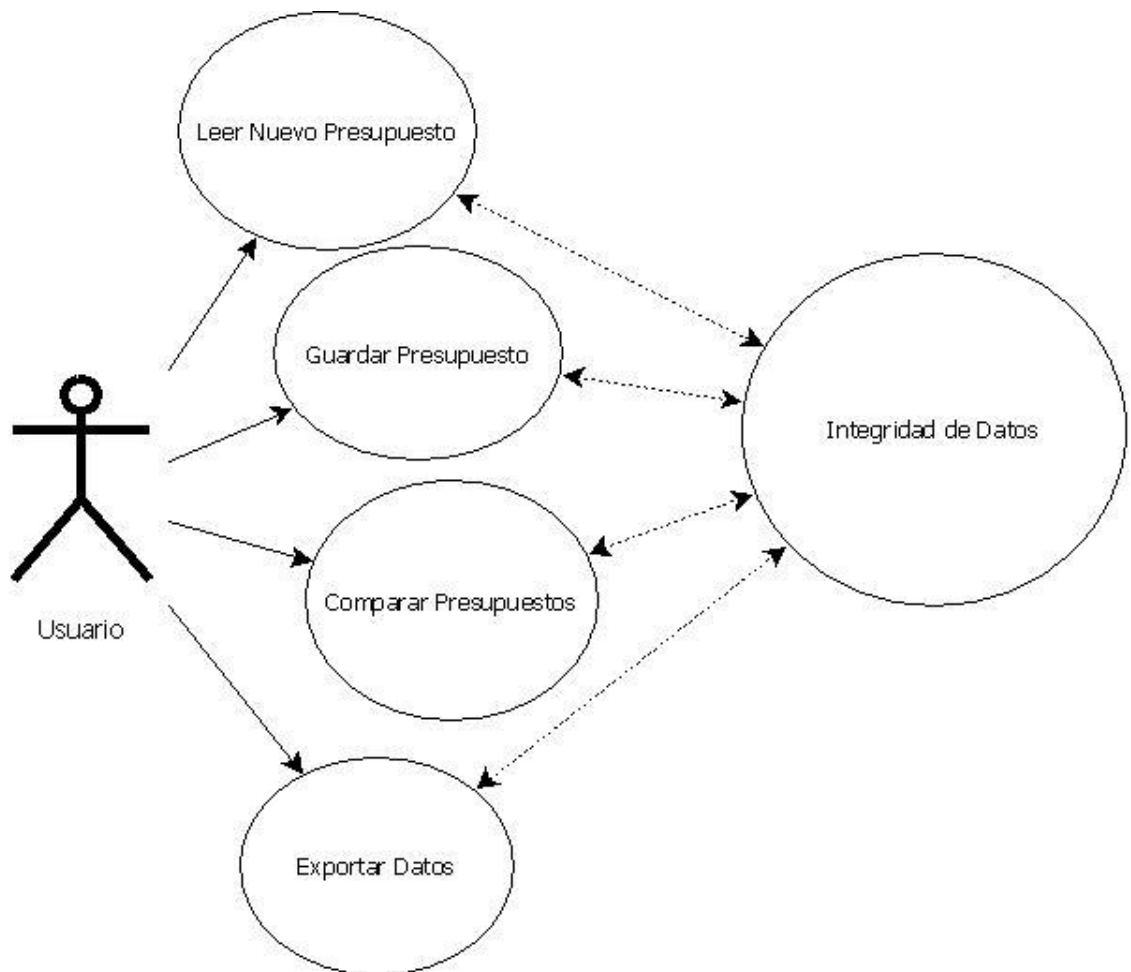
Se realiza la extracción de requisitos con el fin de cumplir con las funcionalidades debe proporcionar la aplicación.

Esta extracción de requisitos se realiza también con el objetivo de orientar al cliente, permitiendo al equipo de desarrollo obtener una lista de requisitos detallada, completa y sin ambigüedades. Estos requisitos proporcionarán una visión general de la aplicación, sin ahondar en aspectos técnicos, estableciendo las principales funcionalidades y restricciones, sirviendo de base a posteriores procesos del ciclo de vida.

Previamente a la redacción de los requisitos han sido identificados los casos de uso con el fin de lograr una mejor definición.

### 4.6.1.-ESPECIFICACION DE LOS CASOS DE USO

A continuación se muestran los diagramas de casos de uso para cada uno de los módulos del sistema que se van a desarrollar en este proyecto:



#### 4.6.2.- ESPECIFICACION DE LOS CASOS DE USO DETALLADO

En este apartado se realiza la especificación de los casos de uso. Cada caso de uso estará especificado por los siguientes atributos:

- **Identificador:** Identifica al caso de uso de forma única. Debe seguir el formato: *CU-XX*, siendo *XX* un valor numérico único para cada caso de uso.
- **Nombre:** Breve especificación textual del caso de uso.
- **Actores:** Tipo de usuario del sistema que inicia el caso de uso.
- **Objetivo:** Finalidad del caso de uso.
- **Precondiciones:** Estado previo que se debe cumplir para poder realizar una operación.
- **Post-condiciones:** Estado en el que queda el sistema tras realizar una operación.
- **Escenario básico:** Especifica la manera en la que interactúa un actor con el sistema y cuál es la respuesta que el sistema le ofrece.

Identificador	CU-01
Nombre	Leer nuevo Presupuesto
Actor	Usuario
Objetivo	Procesar la información de un presupuesto en pdf que se obtiene de la URL.
Precondiciones	Se debe tener conexión a internet. La URL debe apuntar a un fichero en formato PDF con unas características compatibles con el sistema.
Post- Condiciones	Información procesada.
Escenario Básico	El usuario introduce en el formulario la URL del presupuesto y pulsa el botón enviar. La aplicación muestra la información procesada en una página web.

TABLA Caso de uso CU-01

Identificador	CU-02
Nombre	Guardar Presupuesto
Actor	Usuario
Objetivo	Almacenar la información en la base de datos.
Precondiciones	La aplicación debe haber procesado un presupuesto previamente. El presupuesto no debe estar almacenado en la base de datos. Debe haber una base de datos activa, con permisos, configurada en la aplicación.
Post- Condiciones	La información se almacena en la base de datos.
Escenario Básico	La aplicación muestra la información procesada de un presupuesto El usuario selecciona guardar. La aplicación vuelve a mostrar la información

TABLA Caso de uso CU-02

Identificador	CU-03
Nombre	Comprobar Integridad
Actor	Usuario
Objetivo	Mostrar a los usuarios, documentos y claves donde pueden comprobar la integridad de los datos procesados
Precondiciones	Se debe haber procesado y/o guardado un presupuesto.
Post- Condiciones	Se generan diferentes archivos para la consulta del usuario, cuya coincidencia garantizan la integridad de los datos
Escenario Básico	El usuario accede a la página de integridad. La aplicación muestra una página explicativa con diferentes enlaces. La aplicación crea diferentes archivos que el usuario puede ver o descargar para comprobar la integridad.

TABLA Caso de uso CU-03

Identificador	CU-04
Nombre	Comparar Presupuestos
Actor	Usuario
Objetivo	La aplicación muestra los datos almacenados en la base de datos, ordenada y dinámicamente para la comparación de estos.
Precondiciones	Debe haber presupuesto/s almacenado/s en la base de datos
Post-Condicion	La aplicación muestra una página Web con distintas opciones para comparar
Escenario Básico	La aplicación muestra una lista con todos los presupuestos. El usuario selección los presupuestos que quiere comparar La aplicación muestra una página dinámica con la comparación. El usuario puede interactuar con esta página.

TABLA Caso de uso CU-04

Identificador	CU-05
Nombre	Exportar Datos
Actor	Usuario
Objetivo	Poder descargar en un archivo CSV los datos obtenidos en una comparación de presupuestos. Se almacena en BBDD un archivo de seguridad
Precondiciones	Se debe haber visualizado una comparación de presupuestos
Post-Condicion	Se genera un archivo .csv que es posible descargarlo en el PC cliente.
Escenario Básico	El usuario visualiza una comparación de presupuestos. El usuario pulsa Exportar. La aplicación pide el DNI del usuario. El usuario introduce el DNI La aplicación almacena en una bbdd un HASH con el DNI y con el CSV. El usuario ve o descarga el archivo.

TABLA Caso de uso CU-05

### 4.6.3.-DEFINICIÓN DE LOS REQUISITOS DEL SISTEMA

En este apartado se realiza una extracción de requisitos del sistema con el fin de presentar las principales funcionalidades deseadas por el cliente, sirviendo de base a posteriores fases del ciclo de vida del proyecto. Los requisitos identificados proporcionarán al cliente una visión general de la aplicación, de forma completa y sin ambigüedades.

#### 4.6.4.-IDENTIFICACIÓN DE LOS REQUISITOS

En esta tarea se realiza la obtención detallada de requisitos de usuario mediante sesiones de trabajo con el cliente. La lista de requisitos obtenida no es definitiva y podrá ser modificada durante el proyecto, añadiendo requisitos no contemplados, o bien eliminando o modificando requisitos incorrectos, de forma que se terminen cubriendo todas las funcionalidades expuestas por el cliente.

La definición de requisitos recoge lo que quiere el cliente y lo que necesita, englobando los requisitos obtenidos del usuario en dos grandes categorías:

- **Requisitos de capacidad:** Representan lo que necesitan los usuarios para resolver un problema o lograr un objetivo.
- **Requisitos de restricción:** Son las restricciones impuestas por los usuarios sobre cómo se debe resolver el problema o cómo se debe alcanzar el objetivo.

Cada requisito de usuario debe incluir en su definición una serie de atributos, los cuales proporcionen toda la información necesaria para su seguimiento posterior y su clasificación. Estos atributos se describen a continuación:

- ❖ **Identificador:** Cada requisito de usuario debe estar identificado de forma única. Este identificador tendrá el siguiente formato: *RU-C-nn* o *RU-R-nn*, donde:
  - RU: Indica que se trata de un requisito de usuario.
  - C: Indica que se trata de un requisito de usuario de capacidad.
  - R: Indica que se trata de un requisito de usuario de restricción.
  - nn: Tomará valores numéricos dentro del rango 00-99.
- ❖ **Prioridad:** Se asignará una prioridad a cada requisito con el fin de poder realizar una planificación correcta durante fases posteriores. Su clasificación puede tomar los valores: alta, media y baja.
- ❖ **Necesidad:** Los requisitos clasificados como esenciales para el usuario no pueden ser eliminados, mientras que los demás requisitos estarán siempre sujetos a modificación en el caso de que exista una causa que lo justifique. Por lo tanto, se clasificarán descendientemente de acuerdo a su necesidad de la siguiente forma: esencial, deseable y opcional.
- ❖ **Claridad:** Identifica la falta o existencia de ambigüedad de un requisito, esto es, si puede ser interpretado de varias formas dependiendo del contexto. Los requisitos serán clasificados de acuerdo a su claridad de forma descendente de la siguiente forma: alta, media y baja.
- ❖ **Fuente:** Identifica el origen del requisito, que puede estar en el usuario, una fuente externa como un documento, o el propio equipo de desarrollo del proyecto.

- ❖ **Estabilidad:** Algunos requisitos pueden no estar sujetos a cambios durante el proyecto debido a su naturaleza, mientras que otros pueden estar sujetos a determinados cambios por el desarrollo de la etapa de diseño o los requisitos software. Los requisitos poco estables deberán ser observados con mayor cuidado durante el desarrollo del proyecto al ser susceptibles a cambios. Cada requisito se clasificará en: estable o inestable.
- ❖ **Verificabilidad:** Indica si el cumplimiento de un requisito en el sistema puede ser susceptible de comprobación, esto es, si se puede verificar que el requisito se ha incorporado en el diseño y que en el sistema se puede verificar su cumplimiento. La verificabilidad de cada requisito se clasificará en: alta, media y baja.

Identificador	RU-C-01		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Media
Descripción	La información almacenada en la base de datos no se puede eliminar		

TABLA Requisito RU-C-01

Identificador	RU-C-02		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Descripción	La aplicación debe obtener el PDF a través de internet.		

TABLA Requisito RU-C-02

Identificador	RU-C-03		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Descripción	La aplicación selecciona y clasifica la información del presupuesto de la universidad.		

TABLA Requisito RU-C-03

Identificador	RU-C-04		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Descripción	La aplicación procesa la información generando una estructura interna.		

TABLA Requisito RU-C-04

Identificador	RU-C-05		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Descripción	La aplicación almacena la información en una base de datos.		

TABLA Requisito RU-C-05

Identificador	RU-C-06		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Descripción	La aplicación puede recuperar la información de la base de datos y procesarla.		

TABLA Requisito RU-C-06

Identificador	RU-C-07		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Descripción	La aplicación genera unos archivos de seguimiento de la lectura del PDF para poder demostrar que la información del pdf no ha sido modificada en el procesamiento de esta.		

TABLA Requisito RU-C-07

Identificador	RU-C-08		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Media
Descripción	Después de leer y procesar un presupuesto la aplicación mostrará los datos procesados		

TABLA Requisito RU-C-08



Identificador	RU-C-09		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Claridad	media	Verificabilidad	Alta
Descripción	La aplicación mostrará una página con la lista de todos los presupuestos almacenados en base de datos.		

TABLA Requisito RU-C-09

Identificador	RU-C-10		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Descripción	Después de leer y procesar un presupuesto, el sistema dará la opción a los usuarios de guardarlo en la bbdd.		

TABLA Requisito RU-C-10

Identificador	RU-C-11		
Prioridad	Media	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Claridad	media	Verificabilidad	Media
Descripción	La aplicación permitirá escoger más de un presupuesto almacenado en la base de datos, para su comparación.		

TABLA Requisito RU-C-11

Identificador	RU-C-12		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Claridad	Alta	Verificabilidad	Alta
Descripción	La aplicación mostrará una página con los datos de los presupuestos escogidos para su comparación.		

TABLA Requisito RU-C-12

Identificador	RU-C-13		
Prioridad	Media	Fuente	Cliente
Necesidad	Deseable	Estabilidad	Estable
Claridad	media	Verificabilidad	Media
Descripción	La aplicación mostrará una página con ayuda para el usuario.		

TABLA Requisito RU-C-13

Identificador	RU-R-01		
Prioridad	Media	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Claridad	media	Verificabilidad	Media
Descripción	Los archivos que se generaran para la comprobación de la integridad de datos debe estar en formato legible ".txt" o "pdf"		

TABLA Requisito RU-R-01

Identificador	RU-R-02		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Claridad	media	Verificabilidad	media
Descripción	Disponer de una interfaz sencilla e intuitiva que facilite el uso de la aplicación.		

TABLA Requisito RU-R-02

Identificador	RU-R-03		
Prioridad	Baja	Fuente	Cliente
Necesidad	Deseable	Estabilidad	Estable
Claridad	media	Verificabilidad	baja
Descripción	Se mostrara un mensaje de "espera" cuando la aplicación esté trabajando en la lectura, procesamiento.... Es decir, cuando el usuario deba esperar cierto tiempo.		

TABLA Requisito RU-R-03

Identificador	RU-R-04		
Prioridad	Media	Fuente	Cliente
Necesidad	deseable	Estabilidad	Estable
Claridad	alta	Verificabilidad	media
Descripción	La página de visualización del presupuesto leído será interactiva para el usuario.		

TABLA Requisito RU-R-04

Identificador	RU-R-05		
Prioridad	Media	Fuente	Cliente
Necesidad	Deseable	Estabilidad	Estable
Claridad	alta	Verificabilidad	Media
Descripción	Mostrar gráficos de tipo tarta con los datos del presupuesto procesado.		

TABLA Requisito RU-R-05

Identificador	RU-R-06		
Prioridad	Media	Fuente	Cliente
Necesidad	Deseable	Estabilidad	Estable
Claridad	alta	Verificabilidad	Media
Descripción	La pantalla de comparación de presupuestos debe de ser interactiva.		

TABLA Requisito RU-R-06

Identificador	RU-R-07		
Prioridad	Media	Fuente	Cliente
Necesidad	Deseable	Estabilidad	Estable
Claridad	media	Verificabilidad	Media
Descripción	La pantalla de comparación de presupuestos debe mostrar la información de los presupuestos en gráficos de barras.		

TABLA Requisito RU-R-07

Lector de Presupuestos de las Universidades de Castilla y León.  
Autor: Juan José Palacios de La Flor

---

Identificador	RU-R-07		
Prioridad	Alta	Fuente	Cliente
Necesidad	Esencial	Estabilidad	Estable
Claridad	alta	Verificabilidad	Alta
Descripción	La aplicación mostrará una pantalla donde se explique todo el proceso de lectura y almacenamiento en bdd, así como las distintas medidas de seguridad que nos garantizan la integridad de los datos.		

TABLA Requisito RU-R-07





## **5.- DISEÑO**

### **5.1.- DISEÑO DE LA ARQUITECTURA DE SOPORTE**

El objetivo principal de este documento es presentar el diseño del sistema realizado de forma detallada. También se estudiará la tecnología que será de utilidad para llevar a cabo esta actividad.

Se realizará una especificación detallada de los componentes en los que se dividirá el sistema, con el fin de cubrir todas las decisiones de diseño correspondientes a la fase de construcción. De esta manera, sólo quedarán posibles decisiones a tomar relacionadas con el lenguaje de programación seleccionado para la codificación del sistema.

El Documento de Diseño del Sistema es de vital importancia en el desarrollo de un proyecto software, ya que marca las pautas para las posteriores fases de construcción y de implantación final del sistema, siendo un documento básico para los programadores encargados de la implementación.

#### **5.1.1-ALCANCE**

El presente documento presenta una especificación detallada de los componentes en los que se dividirá el sistema, permitiendo cubrir todas las decisiones de diseño que se verán reflejadas en la fase construcción.

Por otro lado, se especificará el entorno tecnológico necesario para que el sistema pueda entrar en ejecución. Contendrá además la planificación de capacidades, los requisitos de administración, el control de accesos, la seguridad y la operación.

#### **5.1.2- DISEÑO DE LA ARQUITECTURA DE SOPORTE**

En este apartado se presentan los elementos más significativos del sistema: los componentes de la arquitectura.

Como se introdujo en el apartado 5 (Descripción Técnica). El sistema sigue una arquitectura cliente-servidor en 3 niveles (o 3 capas).

Componentes de esta arquitectura:

- Servidor: ROL que desempeña un equipo ofreciendo un conjunto de servicios a los clientes, tales como manejo de archivos, páginas web, control de acceso....
- Cliente: ROL que desempeña un equipo demandando servicios de los servidores, pero también puede hacer ciertas tareas de procesamiento local, como desplegar páginas Web, mostrar ventanas....

Los tres niveles corresponderían a:

- Presentación (capa 1): Software que permite presentar de forma adecuada los resultados de una aplicación mediante las páginas Web. Lo que en el caso de nuestra aplicación serán la páginas HTML y jsp.
- Aplicación o lógica de negocio (capa 2): Software que entrega un resultado útil para el usuario en nuestro caso los servlets y las clases .java.
- Administración de datos (capa3): El manejo de los datos en una Base de Datos, que sirven a las aplicaciones de la lógica de negocio. En el caso de nuestro TFG esta capa se desarrolla en un servidor de BBDD MySQL.

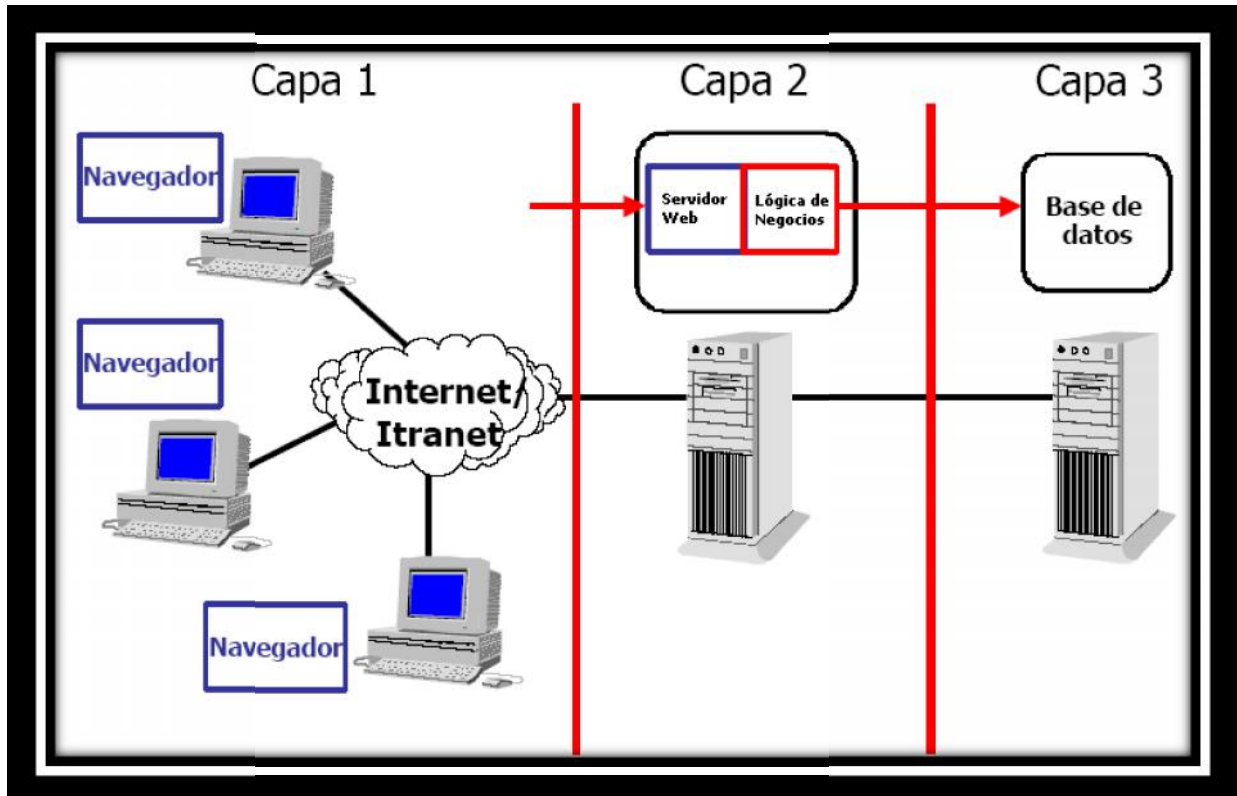


Diagrama ARQUITECTURA CLIENTE SERVIDOR 3 capas



## 5.2.-DIAGRAMAS DE SECUENCIA

El diagrama de secuencia es un tipo de diagrama usado para modelar interacción entre objetos en un sistema según UML.

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. El diagrama de secuencia contiene detalles de implementación, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

Típicamente se examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si se dispone de la descripción de cada caso de uso como una secuencia de varios pasos, entonces se puede atender a esos pasos para descubrir qué objetos son necesarios para seguir los pasos. Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como flechas horizontales.

A continuación se muestran los diagramas de secuencia para los casos de uso vistos anteriormente.

### Lectura de Nuevo Presupuesto.

Collaboration1::Interaction1::SequenceDiagram1

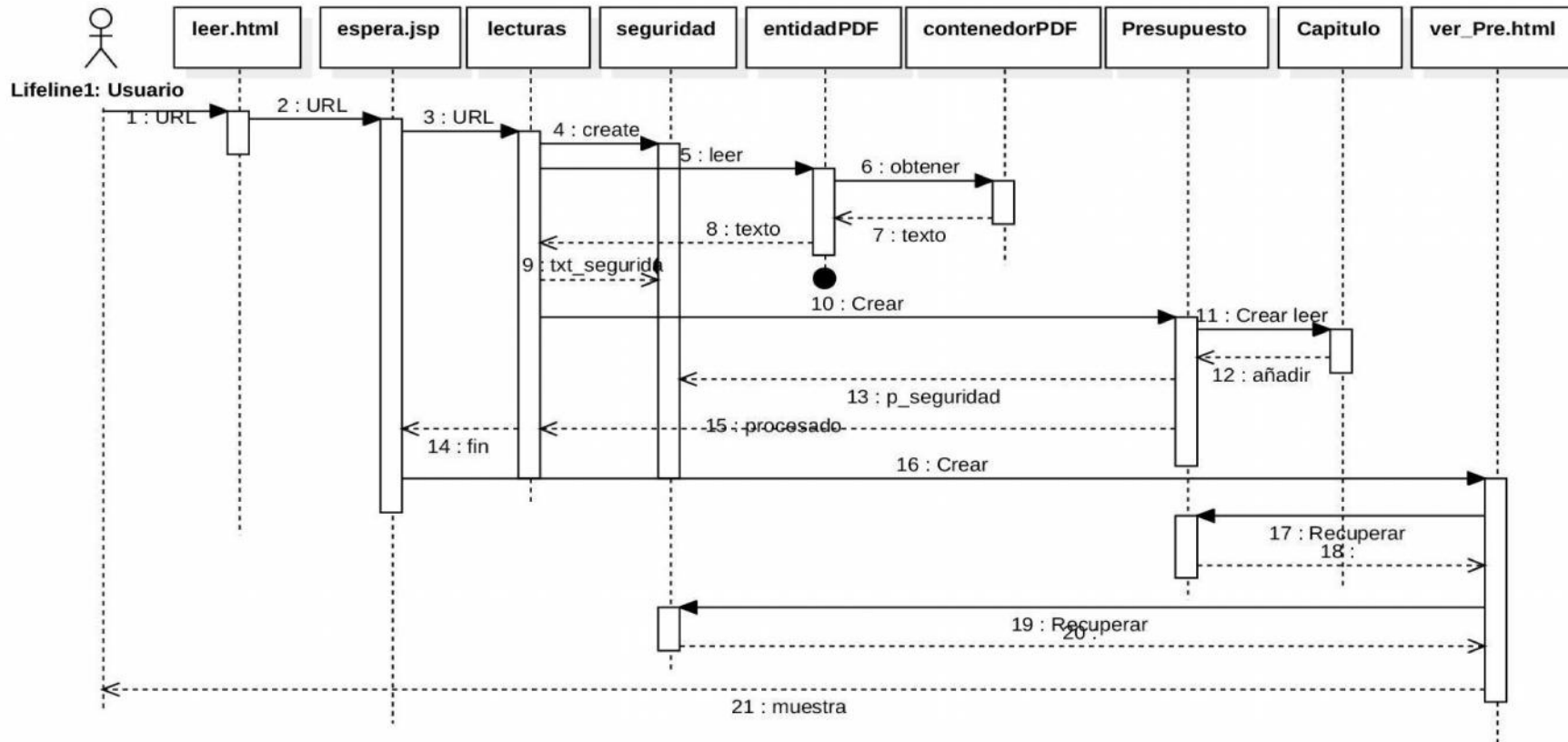


DIAGRAMA SEGUENCIA LECTURA NUEVO PRESUPUESTO

### Guardar Presupuesto.

Collaboration2::Interaction1::SequenceDiagram1

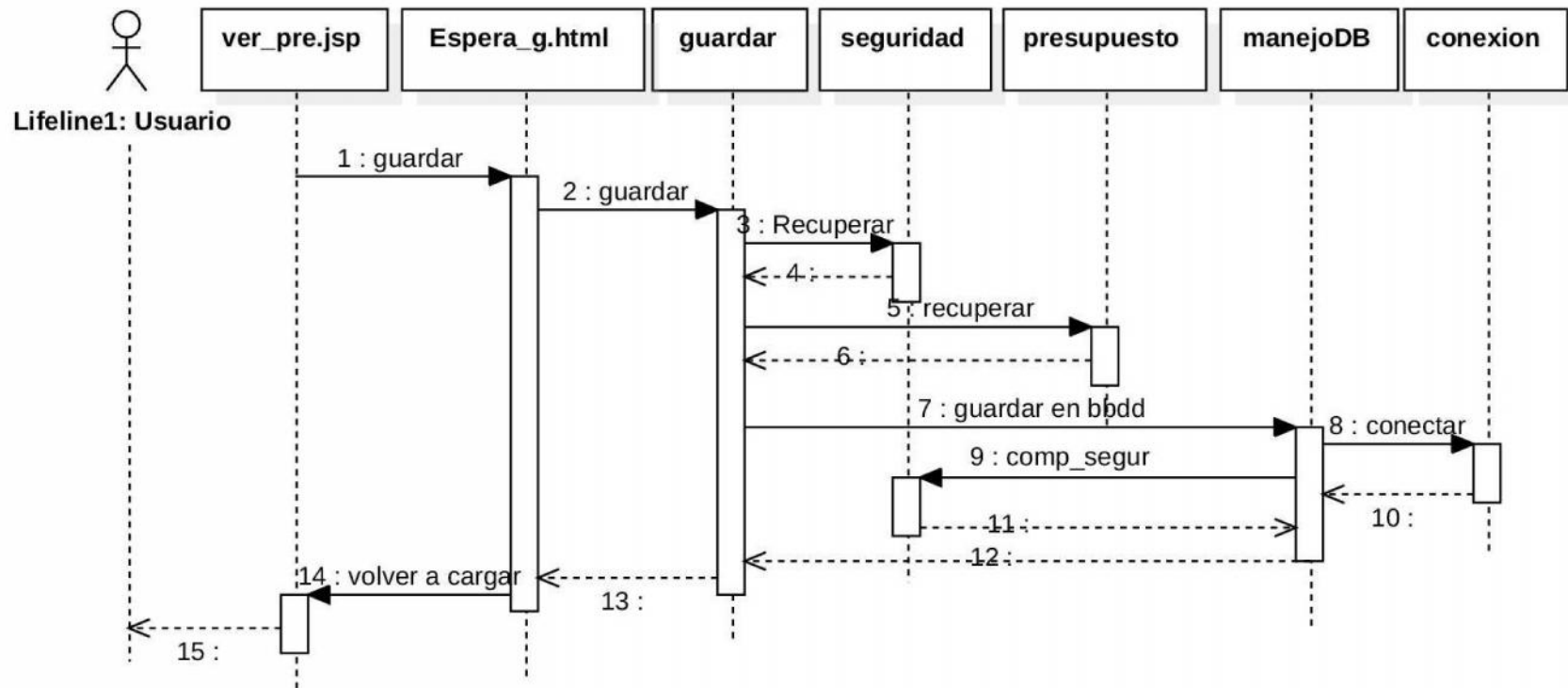


DIAGRAMA Secuencia guardar nuevo presupuesto

### Comprobar Integridad.

Collaboration3::Interaction1::SequenceDiagram1

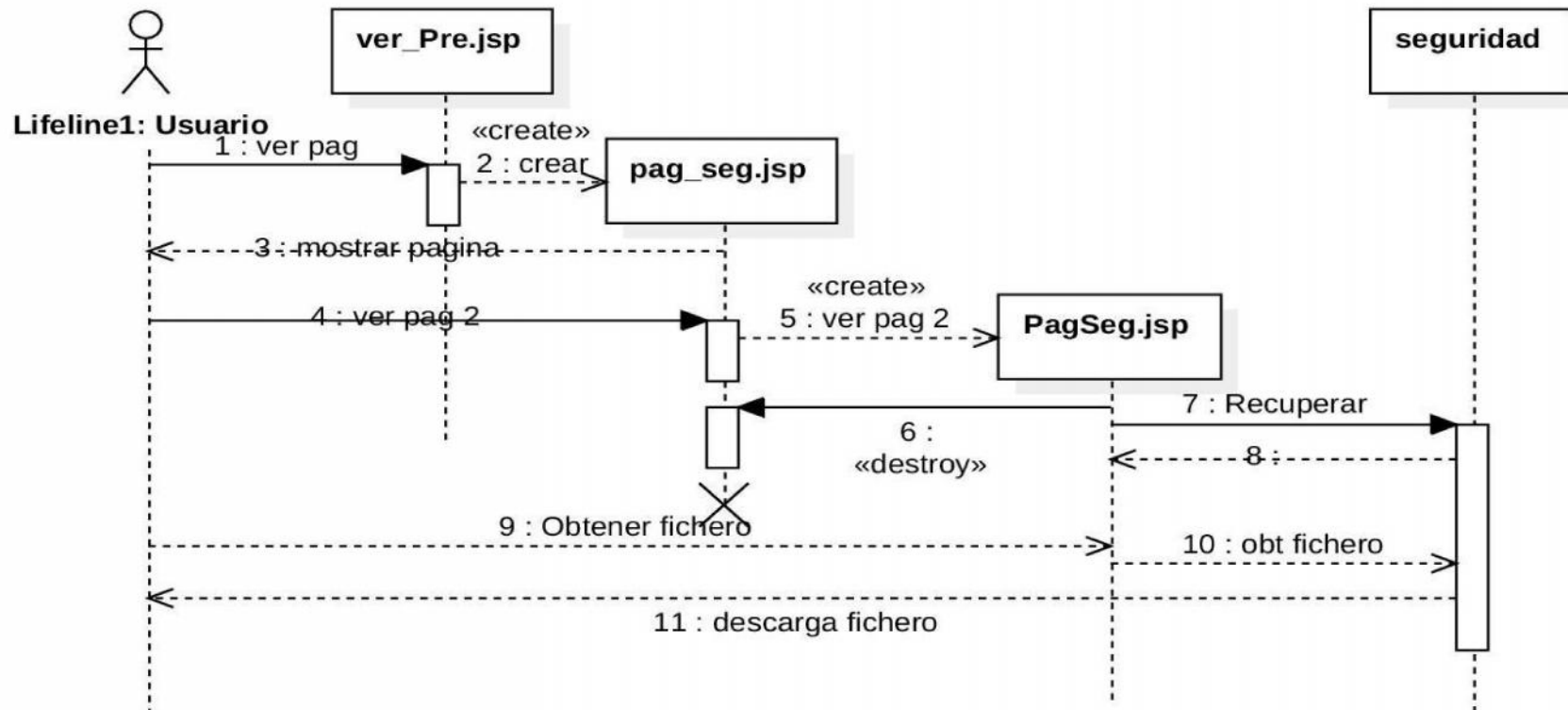


DIAGRAMA Secuencia Comprobar integridad

### Comparar Presupuestos: Selección de presupuestos para comparar

Collaboration4::Interaction1::SequenceDiagram1

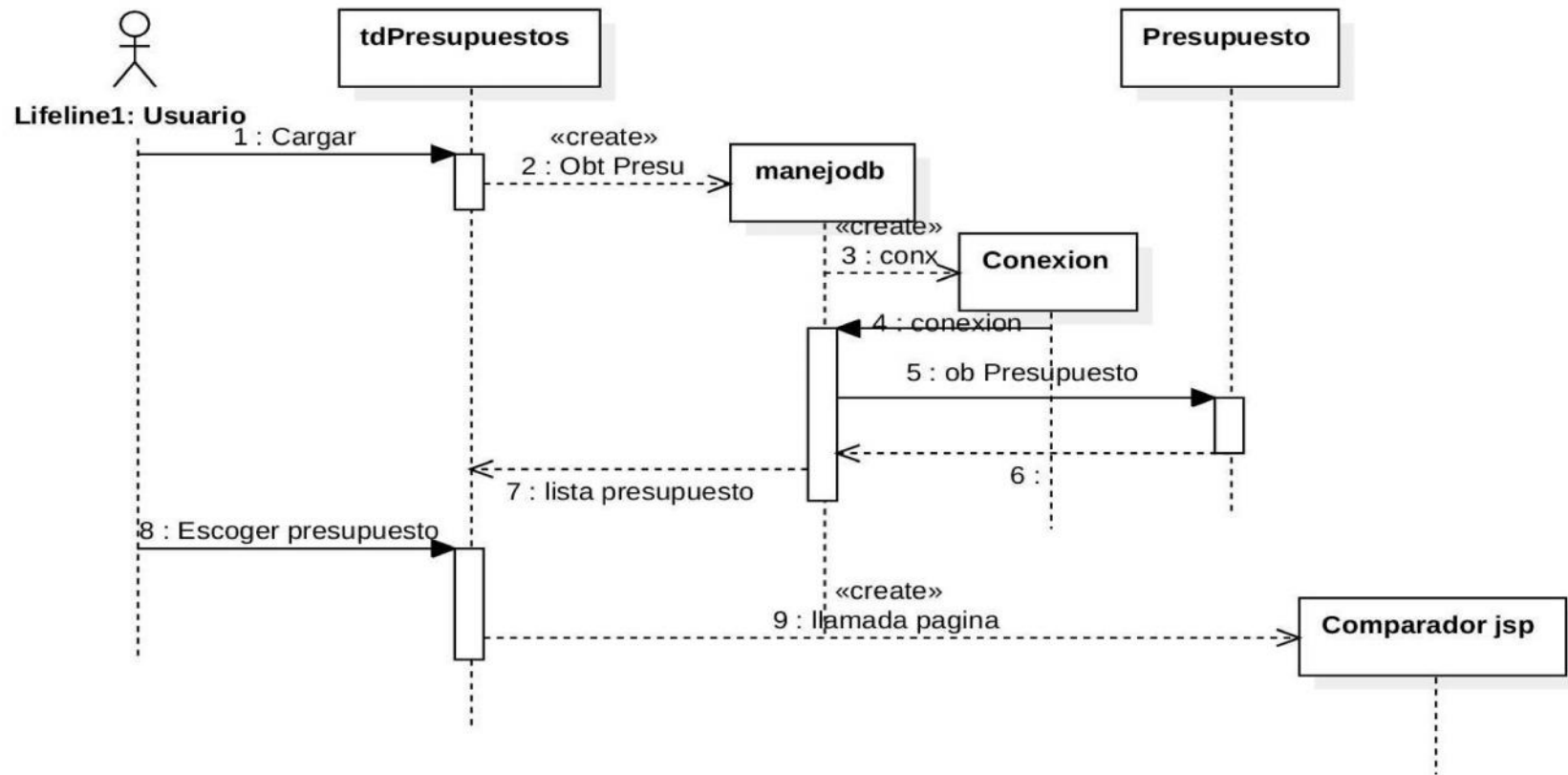


DIAGRAMA Secuencia Selección Presupuestos a Comparar

Comparar Presupuestos: Comparar detalles de los presupuestos.

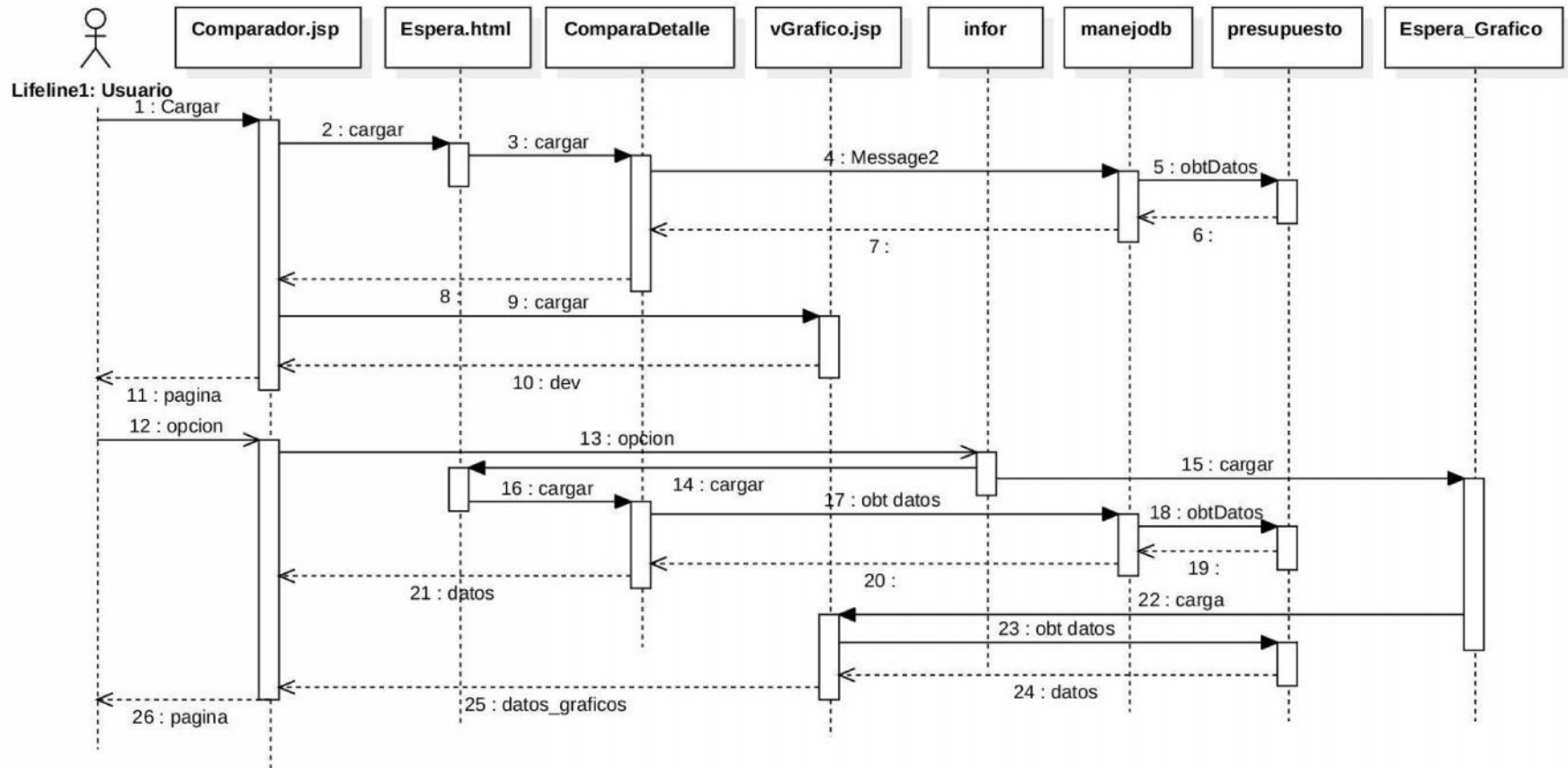


DIAGRAMA Secuencia Comparar detalle de presupuestos

### Exportar Datos

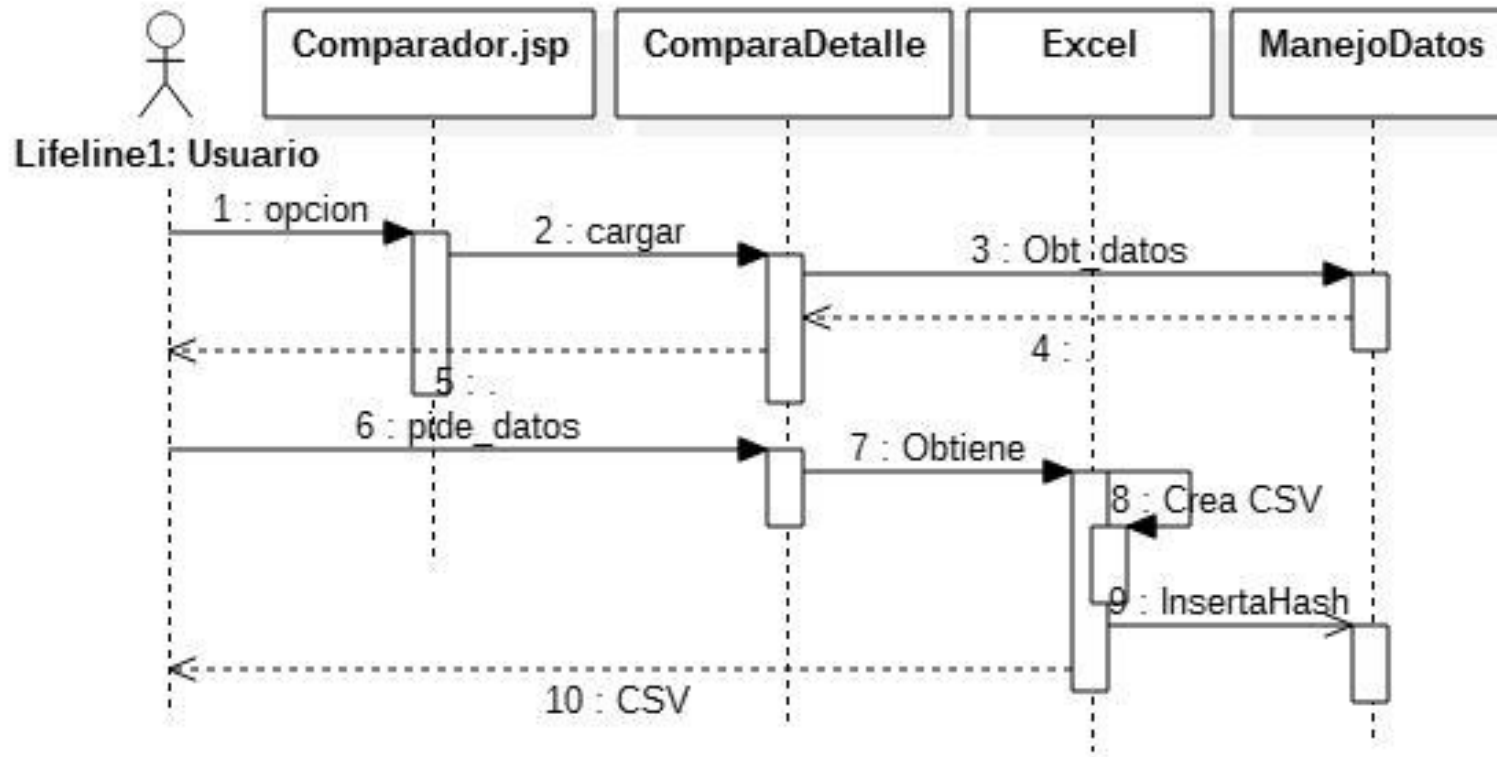


DIAGRAMA Secuencia Exportar Datos

## **5.3.- DISEÑO DE CLASES**

En esta sección se desarrolla el modelo de clases del sistema que se está diseñando.

Los modelos aquí presentados servirán de guía al programador, por lo que los identificadores, relaciones, atributos y funciones son orientativos, dejando al programador libertad para programar bajo su propio criterio.

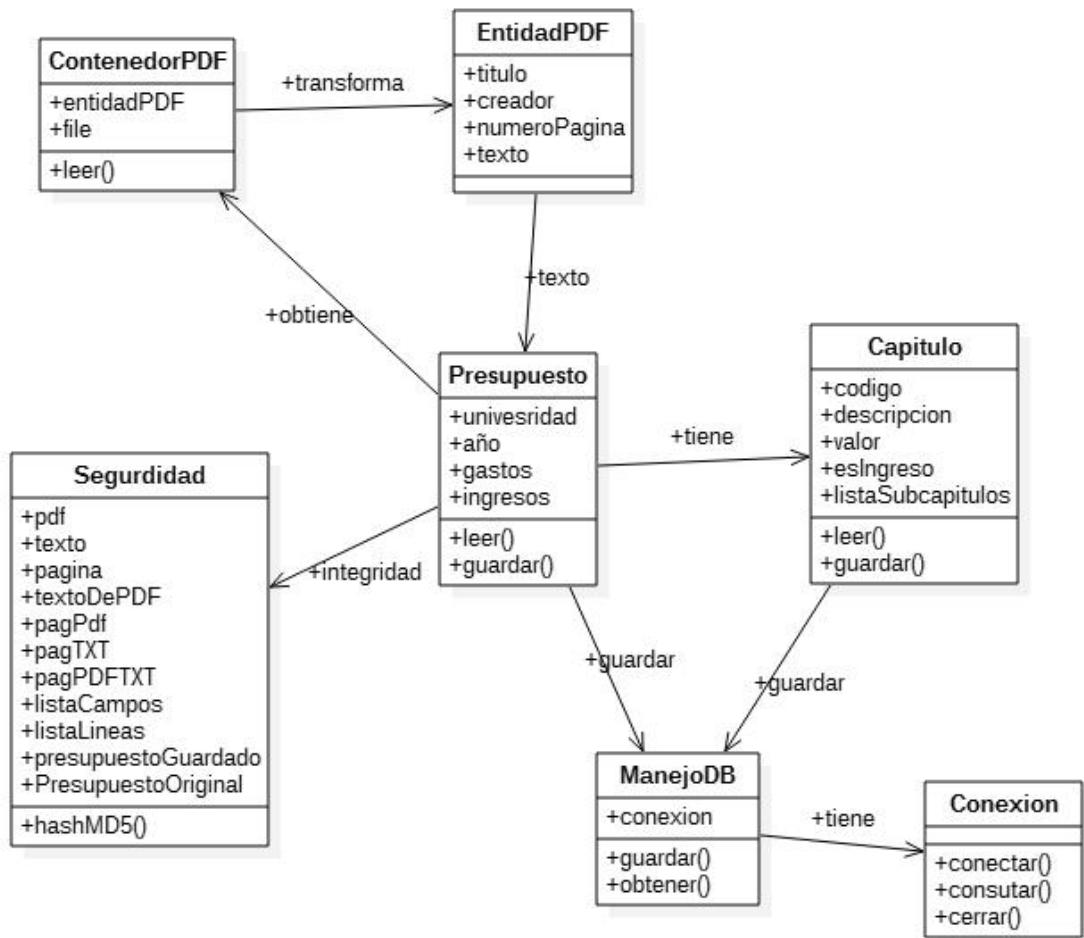
### **5.3.1 DIAGRAMA DE CLASES**

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, orientadas a objetos.

El diagrama de clases incluye mucha más información como la relación entre un objeto y otro, la herencia de propiedades de otro objeto, conjuntos de operaciones/propiedades que son implementadas para una interfaz gráfica.

Presenta las clases del sistema con sus relaciones estructurales y de herencia.





**DIAGRAMA de Clases.**

### 5.3.2.- CLASES, ATRIBUTOS Y METODOS ESTRUCTURALES

A continuación se describen las clases de diseño identificadas, incluyendo el detalle de sus atributos y métodos.

CLASE		Presupuesto
Descripción	Clase que representa un presupuesto procesado.	
<b>ATRIBUTOS</b>		
Identificador	universidad	
Descripción	Texto que indica la universidad relacionada con el presupuesto	
Identificador	año	
Descripción	Texto que indica el año relacionado con el presupuesto	
Identificador	Id	
Descripción	Identificador del presupuesto interno al programa	
Identificador	Ingresos	
Descripción	Lista de Capítulos que se corresponden con los ingresos reflejados en el presupuesto	
Identificador	Gastos	
Descripción	Lista de Capítulos que se corresponden con los gastos reflejados en el presupuesto	
<b>FUNCIONES Y PROCEDIMIENTOS</b>		
Signatura	Leer(Texto)	
Descripción	A partir de un texto pasado como parámetro este procedimiento extrae la información y la procesa completando los campos del presupuesto.	
Signatura	Guardar()	
Descripción	Guarda en la estructura de base de datos los atributos formateados adecuadamente.	
Signatura	PresupuestoTotal()	
Descripción	Función que devuelve la cantidad total del presupuesto.	
Signatura	M_Gasto()	
Descripción	Función que devuelve una cadena de texto con los gastos	
Signatura	M_Ingreso()	
Descripción	Función que devuelve una cadena de texto con los gastos	

TABLA Clase Presupuesto

CLASE		Capítulo
Descripción	Clase que representa a un capítulo o terna del presupuesto.	
<b>ATRIBUTOS</b>		
Identificador	id	
Descripción	Identificador del capítulo interno en el programa	
Identificador	Código	
Descripción	Cadena de texto que representa el código capítulo	
Identificador	Descripción	
Descripción	Cadena de texto que representa	
Identificador	Valor	
Descripción	Numero que representa la cantidad del capítulo	
Identificador	Lista	
Descripción	Lista de subcapítulos que dependen de este capítulo	
Identificador	ingreso	
Descripción	Valor booleano que representa si el capítulo representa ingreso o gasto	
<b>FUNCIONES Y PROCEDIMIENTOS</b>		
Signatura	Guardar(idPres, idpadre, esingreso)	
Descripción	Guarda en la estructura de base de datos los atributos del capítulo formateados y relacionados adecuadamente.	
Signatura	LeerLinea(texto)	
Descripción	Extrae de la línea de texto pasada como parámetro los valores determinados para completar los campos del capítulo	
Signatura	filaCSV()	
Descripción	Devuelve una fila para añadirla a un archivo CSV	
Signatura	Buscar(código)	
Descripción	Devuelve la lista de subcapítulos que dependen del capítulo cuyo código se ha pasado como parámetro.	

TABLA Clase Capítulo

CLASE		Conexion
Descripción		Clase para poder establecer una conexión a la base de datos
<b>ATRIBUTOS</b>		
Identificador		s
Descripción		Statement El objeto que se utiliza para ejecutar una sentencia de SQL y devolver los resultados que produce.
Identificador		c
Descripción		Connection Una conexión representa una sesión con una base de datos específica
<b>FUNCIONES Y PROCEDIMIENTOS</b>		
Signatura		Conectar()
Descripción		Abre la conexión con la bbdd.
Signatura		Cerrar()
Descripción		Cierra la conexión con bbdd.
Signatura		Consulta(texto)
Descripción		Lanza la consulta recibida como parámetro contra la bbdd.
Signatura		Crear()
Descripción		Se lanza una consulta que crea la estructura de tablas en la base de datos. Para inicializar.

TABLA Clase Conexión

CLASE		ManejoDB
Descripción	Esta clase es donde se crean y se ejecutan las diferentes consultas a la base de datos.	
<b>ATRIBUTOS</b>		
Identificador	C	
Descripción	Atributo de la clase conexión anteriormente definida.	
<b>FUNCIONES Y PROCEDIMIENTOS</b>		
Signatura	ExistePresupuesto(universidad, anno)	
Descripción	Función que comprueba si existe un presupuesto en la bbdd con los datos recibidos como parámetros	
Signatura	ObtPresupuesto(id)	
Descripción	Crea un presupuesto a partir de los datos que hay en bbdd cuyo identificador es recibido como parámetro	
Signatura	insertaCapitulo(código, descripción, valor, idpadre)	
Descripción	Inserta en la bbdd un capitulo con los datos recibidos como parámetros y devuelve el identificador del insertado	
Signatura	InsertarDatosExp(DNI, datos)	
Descripción	Inserta en la tabla los datos que se almacenan cada vez que un usuario los exporta.	
Signatura	Obt_Lista_Presupuestos()	
Descripción	Función que nos devuelve una lista con todos los presupuestos almacenados en la base de datos	
Signatura	Obt_Lista_Presupuestos_Nombre()	
Descripción	Función que nos devuelve una lista pero solo con los títulos de los presupuestos.	
Signatura	Obt_capitulo(id)	
Descripción	Función que recupera de bbdd el capitulo cuyo identificador es recibido como parámetro.	
Signatura	InsertaOrganismo(nombreUniversidad)	
Descripción	Inserta en la tabla Organismo (o universidades) un nuevo registro con el parámetro recibido.	
Signatura	InsertaPresupuesto(ano, universidad)	
Descripción	Inserta en la tabla de bbdd un nuevo registro de un presupuesto con los parámetros recibidos.	

TABLA Clase ManejoDB

CLASE	Seguridad
Descripción	Esta clase es la encargada de gestionar los archivos que nos garantizan la integridad de los datos de los documentos procesados.
<b>ATRIBUTOS</b>	
Identificador	Ruta
Descripción	URL que nos indica donde está el pdf, procesado.
Identificador	CamposConcatenados
Descripción	Cadena de texto que representa unos campos procesados escogidos de manera aleatoria
Identificador	LineasConcatenadas
Descripción	Cadena de texto de líneas del documento original escogidas de manera aleatoria
Identificador	PresAleatoriaOriginal
Descripción	Texto formado por valores procesados escogidos aleatoriamente
Identificador	PresAleatoriaGuardada
Descripción	Texto formado por valores extraídos de la bbdd escogidos de manera aleatoria
Identificador	TXTCompleto
Descripción	Texto extraído del archivo PDF
Identificador	TextoPDF
Descripción	Texto extraído de una página aleatoria del documento PDF
Identificador	DocumentoPDF
Descripción	Página en formato PDF extraída del documento original
Identificador	listaCampos
Descripción	Lista aleatoria de campos procesados
Identificador	listaLineas
Descripción	Lista aleatoria de líneas de texto del documento
Identificador	miniTXT
Descripción	Texto que corresponde a una página aleatoria extraída del texto transformado
Identificador	Página guardada
Descripción	Numero de la página que se selecciona aleatoriamente.
<b>FUNCIONES Y PROCEDIMIENTOS</b>	
Signatura	GuardadoSeguro(Presupuesto)
Descripción	Guarda en BBDD el presupuesto recibido como parámetro almacenando en la clase los valores que garantizaran la integridad de los datos originales
Signatura	CrearFicheroTXT(parámetro)
Descripción	Crea localmente un fichero de texto que según el parámetro recibido tendría un nombre y un contenido determinado.
Signatura	CrearPDFMini()

Descripción	Crea localmente un fichero en formato PDF con la página aleatoria
Signatura	HASH(parámetro)
Descripción	Función que devuelve una cadena de texto que se extrae de la función hash del documento que se le pasaría como parámetro.
Signatura	PáginaGuardada()
Descripción	Devuelve un numero entero que representa a la página en PDF que se guarda de manera aleatoria.

TABLA Clase Seguridad

CLASE	EntidadPDF
Descripción	Clase donde se almacena toda la información y meta-informacion de un PDF
<b>ATRIBUTOS</b>	
Identificador	Título
Descripción	Cadena de texto
Identificador	Autor
Descripción	Cadena de texto
Identificador	Numeropáginas
Descripción	Numero entero
Identificador	Creador
Descripción	Cadena de texto
Identificador	Contenido
Descripción	Cadena de texto
Identificador	Docu
Descripción	PDDocument que representa todo el contenido
Identificador	Tema
Descripción	Cadena de texto
<b>FUNCIONES Y PROCEDIMIENTOS</b>	
Signatura	
Descripción	

TABLA Clase EntidadPDF

CLASE		contenedorPDF
Descripción	Clase donde se extrae el documento en PDF para su procesamiento	
<b>ATRIBUTOS</b>		
Identificador	File	
Descripción	Clase FileInputStream donde obtenemos los bytes de entrada	
Identificador	document	
Descripción	Clase PDDocument que representa en memoria al documentoPDF	
<b>FUNCIONES Y PROCEDIMIENTOS</b>		
Signatura	convertirModelo(URL, seguridad)	
Descripción	Descarga el pdf de la URL pasada como parámetro, y almacena los datos correspondientes en la clase seguridad	
Signatura	GetDocument()	
Descripción	Devuelve el Documento	

TABLA Clase Contenedor PDF



### 5.5.3.- DESCRIPCION DE CLASES DE CAPA INTERMEDIA (SERVLETS)

Además de las clases estructurales descritas anteriormente se han diseñado clases Servlet para realizar acciones que se ejecutan en el servidor, realizando la función de una capa intermedia entre una petición proveniente de un navegador Web.

Su función principal es proveer páginas web dinámicas y personalizadas, utilizando para este objetivo accesos a bases de datos, flujos de trabajo y otros recursos.

La interfaz ServletRequest permite al servlet acceder a la información pasada por el cliente como, los nombres de parámetros, el protocolo, los nombres de los host remotos que hacen la solicitud y el servidor que la recibe. Esta interfaz permite a los servlets el acceso a métodos que permiten manejar la presentación de la respuesta como salida en el navegador, a través de los cuales consiguen los datos desde el cliente que usa protocolos como HTTP POST.

A continuación se describirán los Servlets y la función de cada uno:

NOMBRE	ComparaDetalle
Descripción	<p>Recupera como parámetros una lista de presupuestos y una cadena de texto. Y muestra en la pantalla de un navegador Web debidamente formateado, los detalles de cada presupuesto.</p> <p>La cadena es del formato: "lnn" donde l es una letra ("i" o "g") que hacen referencia a ingresos o gastos.</p> <p>Y "n" son números que corresponden al subcampo del presupuesto.</p> <p>Según la cadena introducida se mostrarán unos detalles u otros: Si la cadena está vacía se muestran el total,</p>

TABLA Clase Servlet ComparaDetalle

NOMBRE	Descarga
Descripción	Esta clase genera una página que crea una descarga de un archivo que recuperamos como parámetro.

TABLA Clase Servlet Descarga

NOMBRE	GuardaConfigBD
Descripción	En esta clase genera una página que recupera los datos que se han enviado de un formulario. Estos datos son los parámetros para conectarse a una base de datos. Si logra hacer la conexión, se almacenan estos datos en un archivo de configuración llamado config.txt

TABLA Clase Servlet GuardaConfig

NOMBRE	<b>guardaParametros</b>
Descripción	Este servlet recuperaría los datos introducidos en un formulario y los guardaría en un fichero llamado patametros.txt. Estos datos son las constantes con las que el proceso de lectura utiliza para procesar el documento.

TABLA Clase Servlet guardaParametros

NOMBRE	<b>Guardar</b>
Descripción	Servlet que recupera como parámetros un presupuesto y una instancia de seguridad, y guarda el presupuesto en bbdd.

TABLA Clase Servlet Guardar

NOMBRE	<b>Infor</b>
Descripción	Servlet que recupera un lista de presupuestos y una cadena de texto recuperada como parámetro, para generar dinámicamente un nuevo formulario según el texto de la cadena recuperada, además llama a otros servlets en distintos iframes.  Esta página se muestra dentro de un div en la página de comparar presupuestos.

TABLA Clase Servlet Infor

NOMBRE	<b>Lecturas</b>
Descripción	Este servlet genera una página que recupera el parámetro URL de un formulario, y lanza el proceso de lectura del presupuesto con la URL introducida. Y después redirige a la página que muestra el presupuesto leído.

TABLA Clase Servlet Lecturas

NOMBRE	<b>Xls</b>
Descripción	Este servlet crea un archivo csv con los datos recuperados como parámetros de csv, además recupera el parámetro DNI, guardando estos parámetros en la bbdd debidamente modificados (guarda el HASH de estos) y llama a la descarga de el archivo csv.

TABLA Clase Servlet XLS

Otras Clases de esta capa que no son Servlet:

Nombre	Grafico
Descripción	Es una clase que contiene 4 funciones, cada función nos devuelve un grafico de barras o circular, en forma de la clase jFreechar. La diferencia entre las distintas funciones es la cantidad de parámetros de entrada, ya que cada parámetro correspondería a un presupuesto.

TABLA Clase Auxiliar Grafico

#### 5.5.4.- DESCRIPCION DE PÁGINAS DE LA CAPA CLIENTE

A continuación describiremos el diseño de la estructura de las páginas por las que presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato).

Estas son páginas en formato HTML, xhtml o jsp, junto a javascript y css.

Nombre	Inicio
Descripción	Es la página de inicio de la aplicación Web donde se describe brevemente la función de esta.

TABLA Página Inicio

Nombre	leer
Descripción	Es la página nos muestra un formulario en el que el usuario introducirá la URL del presupuesto que desea procesar.

TABLA Página leer

Nombre	verPresupuesto
Descripción	Esta página muestra al usuario el presupuesto los datos del presupuesto procesado, mediante javascript se mostrará un gráfico (con google api chart ) y se mostraran y ocultaran secciones de esta página lo que dará un aspecto dinámico a esta.

TABLA Página verPresupuesto

Nombre	ayuda
Descripción	Página en la que se describe cada función y aspecto de la aplicación web para ayudar al usuario.

TABLA Página ayuda

Nombre	Integridad_Explicacion
Descripción	Es la página que explica a los usuarios el sistema que garantiza la integridad de los datos procesados y guardados en la aplicación.

TABLA Página Integridad\_Explicacion

Nombre	páginaSeguridad
Descripción	En esta página se muestra de manera grafica el proceso de lectura de un presupuesto así como el proceso que garantiza la integridad de los datos procesados, pudiendo, de forma interactiva y gracias a javascript, comprobar dinámicamente este sistema, descargando archivos y mostrando los códigos hash md5 de estos datos.

TABLA Página páginaSeguridad

Nombre	TodosPresupuestos
Descripción	Esta página nos muestra un formulario donde están listados todos los presupuestos almacenados en la base de datos, que nos permite seleccionarlos y mandarlos a otra página para compararlos

TABLA Página TodosPresupuestos

Nombre	Comparacion
Descripción	Esta página muestra al usuario la comparación de los presupuestos escogidos anteriormente. Es una página con la que el usuario puede interactuar y esta respondería dinámicamente gracias a la tecnología AJAX.

TABLA Página Comparacion

Nombre	ConfiguracionBD
Descripción	Página web que nos ofrece un formulario para poder configurar la base de datos que usará la aplicación Web

TABLA Página ConfiguracionBD

Nombre	ParametrosLectura
Descripción	Página web que nos ofrece un formulario para poder configurar las constantes por las cuales se va a regir el proceso de lectura del pdf.

TABLA Página ParametrosLectura

Además de las páginas descritas se crean páginas de puente que muestra un mensaje de espera en el tiempo que la aplicación realiza los distintos procesos. Estas páginas se nombraran como espera.html, espera2.html.....

## **5.6.- DISEÑO FÍSICO DE DATOS**

### **5.6.1.- DIAGRAMA ENTIDAD- RELACIÓN (E-R)**

El modelo Entidad-Relación (E-R) representa un determinado dominio utilizando entidades y relaciones entre ellas.

Entidades:

- Cada entidad se caracteriza por el valor de sus atributos.
- Un atributo (clave) identifica unívocamente a cada entidad.

Relaciones:

- Una relación describe una asociación entre varias entidades.
- Esta asociación determina el número de entidades (cardinalidad) relacionadas en la misma.

A continuación se muestra el modelo E/R de la base de datos:

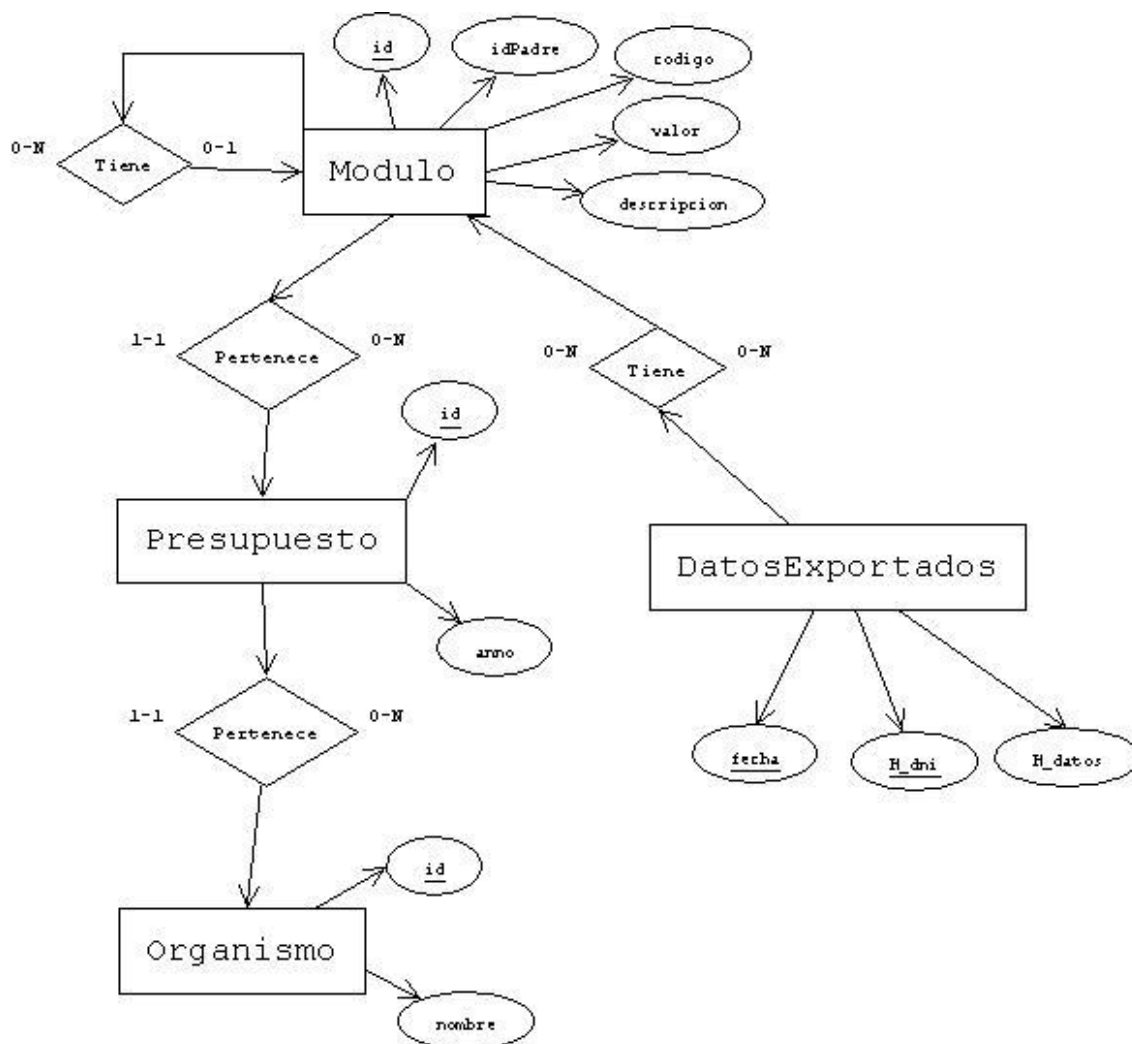


DIAGRAMA Entidad-Relación

La Entidad DatosExportados es en la que se almacena un registro de los datos que se han exportado los usuarios, para tener una garantía de integridad de estos. En ella se almacenan los siguientes datos:

- Un identificador interno (un numero)
- fecha y hora en la cual el usuario ha hecho la exportación.
- Una cadena de texto que representa al DNI del usuario codificado con el algoritmo hash MD5.
- Una cadena de texto que representa los datos que se ha exportado el usuario en el formato csv y que esta codificado con el algoritmo hash MD5.

La Entidad Organismo representa a los organismos, en nuestro caso las universidades, que expiden los presupuestos. Que tienen los siguientes datos:

- Un identificador interno (un numero)
- Una cadena de texto con el nombre de la universidad.

La Entidad Presupuesto almacenaría los datos del presupuesto procesado y almacenado. En el que están los datos:

- Un identificador interno (un numero)
- Una numero que representa al año

Entre la entidad Presupuesto y la entidad Organismo existe una relación, de modo que un presupuesto tiene que pertenecer obligatoriamente a un organismo, por lo que la relación será de 1:N.

La entidad que hemos nombrado como Modulo representa a un capitulo, articulo, concepto o subconcepto perteneciente a un presupuesto. Esto es una terna de datos compuesta por un código, una descripción y una cantidad. Esta entidad tendría los siguientes datos:

- Un identificador interno (un número).
- Una cadena de texto que representa a el código
- Una cadena de texto de descripción
- Un numero que representa el valor
- Un número que representa al identificador del presupuesto al que pertenece.
- Un valor booleano que indica si el modulo pertenece a el bloque de gastos o de ingresos.
- Un identificador de un modulo que indica si este modulo es un submodulo que pertenece a otro (es hijo).

Entre la entidad Presupuesto y la entidad Modulo existe una relación en la cual un modulo pertenece a un presupuesto (solamente a uno). Por lo tanto la relación existente es de tipo 1:N.

Existe además una relación entre dos entidades Modulo. Un modulo puede pertenecer o no a otro modulo (pero solamente a uno) es decir ser un submodulo. Por lo que también un modulo puede tener o no tener un modulo hijo. De modo que la relación es 1:N.

### **5.6.2.- PASO DEL MODELO E-R AL MODELO RELACIONAL**

Para pasar a tablas todos los datos sin dejarnos nada y que las tablas tengan sentido por si solas tenemos que seguir unos pasos:

- Toda entidad se transforma en una tabla todo atributo se transforma en una columna dentro de la tabla a la que pertenece. El identificador de la entidad se convierte en la clave primaria de la tabla.
- En las relaciones 1:N la clave primaria de la entidad con cardinalidad 1 pasa a la tabla de la entidad cuya cardinalidad es N.
- Toda relación N:M se convierte en una tabla que tendrá como clave primaria las dos claves primarias de las entidades que se asocian.



### 5.6.3.- MODELO RELACIONAL

El modelo relacional representa los datos y sus relaciones utilizando el concepto de tabla:

- Cada columna almacena el valor de un determinado atributo.
- Cada fila (tupla) comprende los valores de los atributos de una determinada entidad o relación.

El modelo relacional organiza la BD en registros:

- Cada tabla contiene registros de un tipo particular.
- Cada tipo de registro tiene un número determinado de campos o atributos.

El modelo relacional está en un nivel de abstracción inferior al del modelo E-R:

- En general, el modelo E-R se realiza primero y luego se transforma a relacional.
- 

A continuación, se muestra el modelo Relacional de la base de datos de este TFG, tras haber aplicado las transformaciones descritas anteriormente.



DIAGRAMA Modelo Relacional

#### 5.6.4.- DICCIONARIO DE DATOS.

##### Entidad Organismo

Descripción: Se refiere a la universidad que expide los presupuestos.

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<b>ID</b>	int(11)			No	<i>Ninguna</i>	AUTO_INCREMENT
<b>NOMBRE</b>	varchar(60)	latin1_swedish_ci		Sí	<i>NULL</i>	

##### Entidad Presupuesto

Descripción: Presupuesto procesado y almacenado.

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<b>ID</b>	int(11)			No	<i>Ninguna</i>	AUTO_INCREMENT
<b>ANNO</b>	int(11)			Sí	<i>NULL</i>	
<b>IDORGANISMO</b>	int(11)			Sí	<i>NULL</i>	

##### Entidad Modulo:

Descripción: Se refiere a los capítulo, artículos, conceptos y subconceptos que pertenecen a un presupuesto.

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<b>ID</b>	int(11)			No	<i>Ninguna</i>	AUTO_INCREMENT
<b>CODIGO</b>	varchar(30)	latin1_swedish_ci		Sí	<i>NULL</i>	
<b>DESCRIPCION</b>	varchar(150)	latin1_swedish_ci		Sí	<i>NULL</i>	
<b>VALOR</b>	int(11)			Sí	<i>NULL</i>	
<b>IDPADRE</b>	int(11)			Sí	<i>NULL</i>	
<b>IDPRESUPUESTO</b>	int(11)			Sí	<i>NULL</i>	
<b>INGRESOS</b>	smallint(6)			Sí	<i>NULL</i>	

##### Entidad DatosExportados

Descripción: registros de seguimiento de los datos exportados por los usuarios

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<b>ID</b>	int(11)			No	<i>Ninguna</i>	AUTO_INCREMENT
<b>FECHA</b>	datetime			Sí	<i>NULL</i>	
<b>H_DNI</b>	varchar(33)	latin1_swedish_ci		Sí	<i>NULL</i>	
<b>H_CSV</b>	varchar(33)	latin1_swedish_ci		Sí	<i>NULL</i>	

### Entidad ModuloDatos

Descripción: Tabla que relaciona los Datos exportados con los módulos.

Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
IDMODULO	int(11)			Si	NULL	
IDDATOS	int(11)			Si	NULL	

## 5.7.- INTERFAZ DE USUARIO

En este punto se pretende dar una explicación general del comportamiento de la aplicación y como se accede a él, de forma que sin necesidad de conocer todo el sistema sepa que funciones encontrara en cada uno de los mantenimientos de la aplicación.

### 5.7.1.- ACCESO AL SISTEMA

Para entrar en la aplicación basta con introducir la dirección de la página de inicio en la barra de direcciones del explorador de Internet.

### 5.7.2.- PÁGINA PRINCIPAL

Esta es la pantalla principal de la aplicación, es nuestra área de trabajo donde se cargarán todas las secciones.



IMAGEN Página inicio

En la pantalla principal se pueden diferenciar 3 partes:

- Una cabecera con el título y el subtítulo.
- Una barra desde donde se puede navegar a todos los formularios de la aplicación.
- Área general: Donde aparecen las pantallas que seleccione el usuario.

### 5.7.3.- LEER PRESUPUESTO EN PDF

Para leer un presupuesto el usuario debe introducir la URL donde se encuentra el presupuesto en PDF.



The screenshot shows a web interface titled "Leer nuevo PDF" in a blue serif font. Below the title is a blue circular icon with a white question mark. Underneath the icon, the text "Introducir la URL donde descargar el pdf" is displayed. A long, empty text input field follows. Below the input field are two buttons: "Leer" and "Limpiar". At the bottom of the interface, an example URL is provided: "Por ejemplo: <http://bocyl.jcyl.es/boletines/2013/06/12/pdf/BOCYL-D-12062013-11.pdf>".

IMAGEN Página Leer Presupuesto

#### 5.7.4.- VER PRESUPUESTO LEÍDO/ PROCESADO

En esta pantalla se mostrarán los detalles del presupuesto que se acaba de leer y procesar. En ella se podrán desplegar haciendo clic sobre el elemento, los sub-campos que pertenecen a este elemento además mostrando una grafica tipo tarta con estos detalles.



IMAGEN Página Ver Presupuesto Procesado

#### 9.7.5.- EXPLICACIÓN DE INTEGRIDAD DE DATOS.

Dentro de esta página podremos leer el método por el cual se procesan los presupuestos. Se explica detalladamente la metodología por la que se garantiza que los datos han sido manipulados en este proceso.

### 5.7.6.- COMPROBACIÓN DE LA INTEGRIDAD DE LOS DATOS

En este apartado de la aplicación Web podemos ver un esquema interactivo del flujo de lectura, procesado y almacenado de información de un presupuesto. Colocando el puntero del ratón sobre los distintos campos del esquema nos aparecerá una breve descripción del proceso. Y pulsando sobre los iconos de MD5 nos aparecerá el código hash correspondiente al archivo que se obtiene. Pulsando sobre el icono de cada archivo podríamos descargar o abrir este en el PC del cliente.

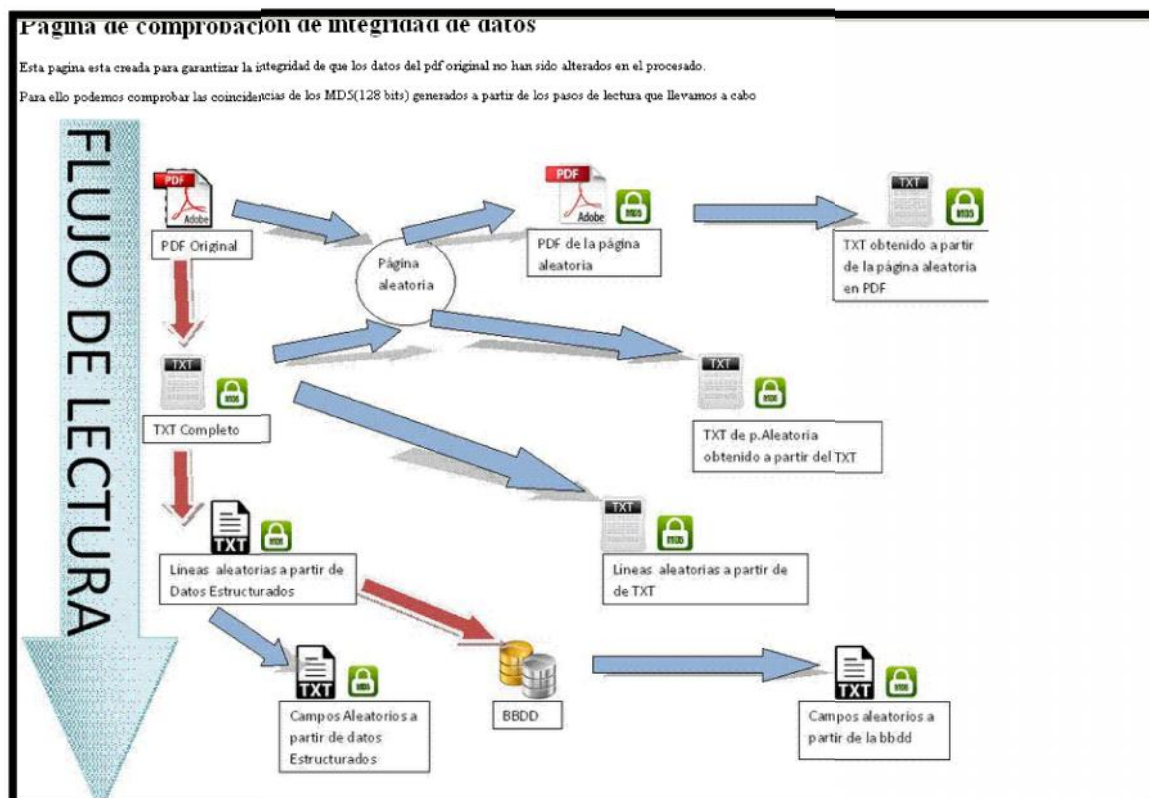


IMAGEN Página interactiva de comprobación de integridad

### 5.7.7.- SELECCIÓN DE PRESUPUESTOS A COMPARAR

Dentro de esta página aparece un formulario en el que podremos escoger los presupuestos que queramos comparar.

Los presupuestos se mostrarán agrupados por organismo emisor y ordenados por fecha.



The screenshot shows a web interface titled "Listado de Presupuestos Almacenados". It features a blue question mark icon and the instruction "Seleccionar el/los presupuestos a mostrar". Below this, there are three columns of checkboxes, each headed by a university name: LEÓN, SALAMANCA, and VALLADOLID. Each column lists the years from 2010 to 2015. At the bottom left, there is a "Comparar" button.

LEÓN	SALAMANCA	VALLADOLID
<input type="checkbox"/> LEÓN-2010	<input type="checkbox"/> SALAMANCA-2010	<input type="checkbox"/> VALLADOLID-2010
<input type="checkbox"/> LEÓN-2011	<input type="checkbox"/> SALAMANCA-2011	<input type="checkbox"/> VALLADOLID-2011
<input type="checkbox"/> LEÓN-2012	<input type="checkbox"/> SALAMANCA-2012	<input type="checkbox"/> VALLADOLID-2012
<input type="checkbox"/> LEÓN-2013	<input type="checkbox"/> SALAMANCA-2013	<input type="checkbox"/> VALLADOLID-2013
<input type="checkbox"/> LEÓN-2014	<input type="checkbox"/> SALAMANCA-2014	<input type="checkbox"/> VALLADOLID-2014
<input type="checkbox"/> LEÓN-2015	<input type="checkbox"/> SALAMANCA-2015	<input type="checkbox"/> VALLADOLID-2015

IMAGEN Página Selección de presupuestos



### 5.7.8.- COMPARACIÓN DE PRESUPUESTOS

En esta página se harán las comparaciones de los presupuestos.

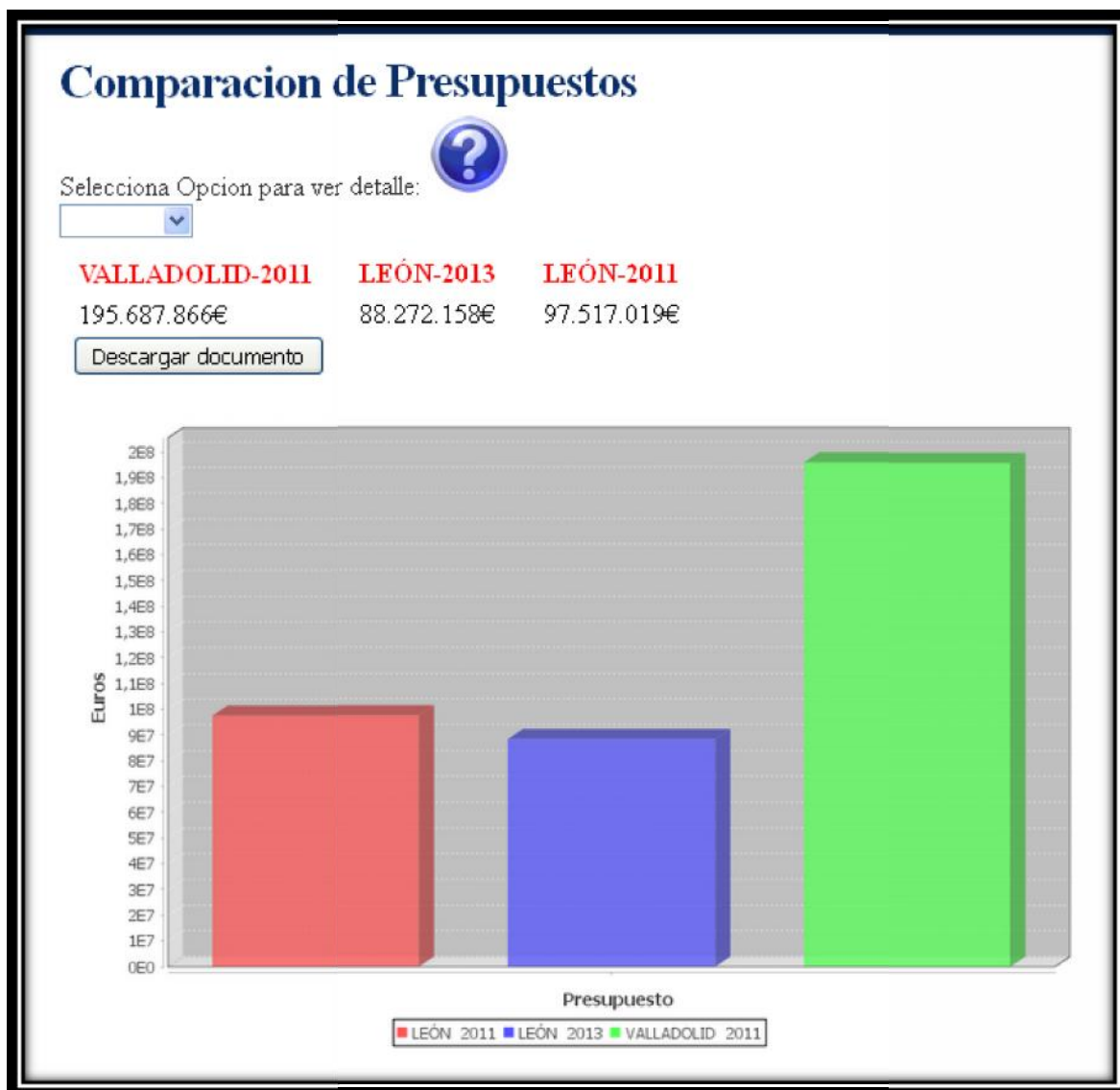


IMAGEN Página comparación de presupuestos

Esta página usara la tecnología Ajax por lo que será dinámica es decir que el usuario podrá interactuar con ella y la página responderá a las acciones de este.

Tiene 3 partes bien diferenciadas:

El formulario donde aparecerá una o varias listas en las que los usuarios escogerán los datos que quieren que la aplicación les muestre en esta página.

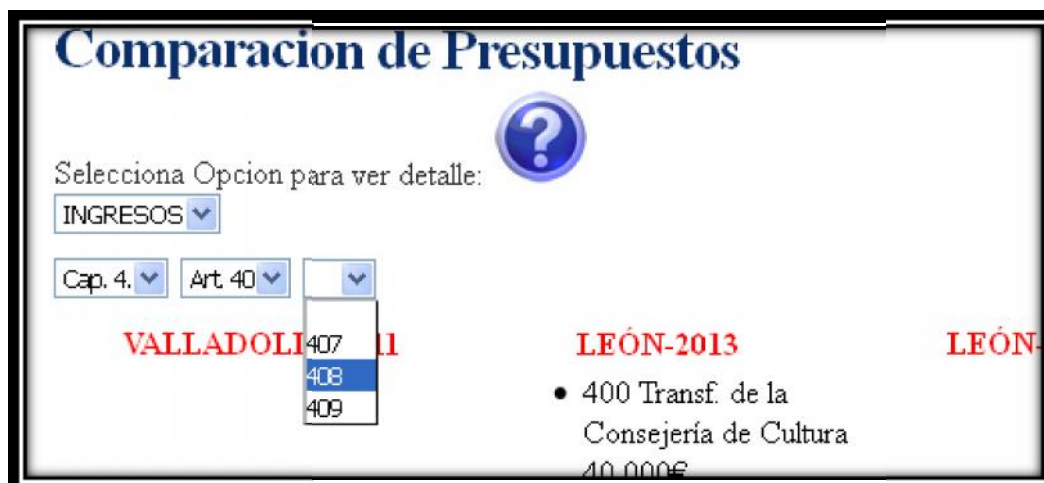


IMAGEN Página formulario interactivo

La parte donde la aplicación mostrará los datos obtenidos en la base de datos y además aparecería el botón donde descargar los datos.

VALLADOLID-2011	LEÓN-2013	LEÓN-2011
• Art. 31 Precios públicos 23.115.000€	• Art. 31 Precios públicos 18.888.113€	• Art. 31 Precios públicos 14.002.000€
• Art. 32 Ingresos por prestación de otros servicios 5.845.000€	• Art. 32 Otros ingresos procedentes de prestación de 3.522.500€	• Art. 32 Otros ingresos procedentes de prestación de 3.385.306€
• Art. 33 Venta de bienes 113.000€	• Art. 33 Venta de bienes 77.000€	• Art. 33 Venta de bienes 175.000€
• Art. 38 Reintegros de operaciones corrientes 350.000€	• Art. 38 Reintegros de operaciones corrientes 50.000€	• Art. 38 Reintegros de operaciones corrientes 43.000€
• Art. 39 Otros Ingresos 1.505.000€	• Art. 39 Otros Ingresos 40.000€	• Art. 39 Otros Ingresos 120.000€

Descargar documento

IMAGEN Página comparación detallada

La ultima parte de la página donde aparecerá una grafica de tipo barras o de tipo tarta. Que representara los datos obtenidos.

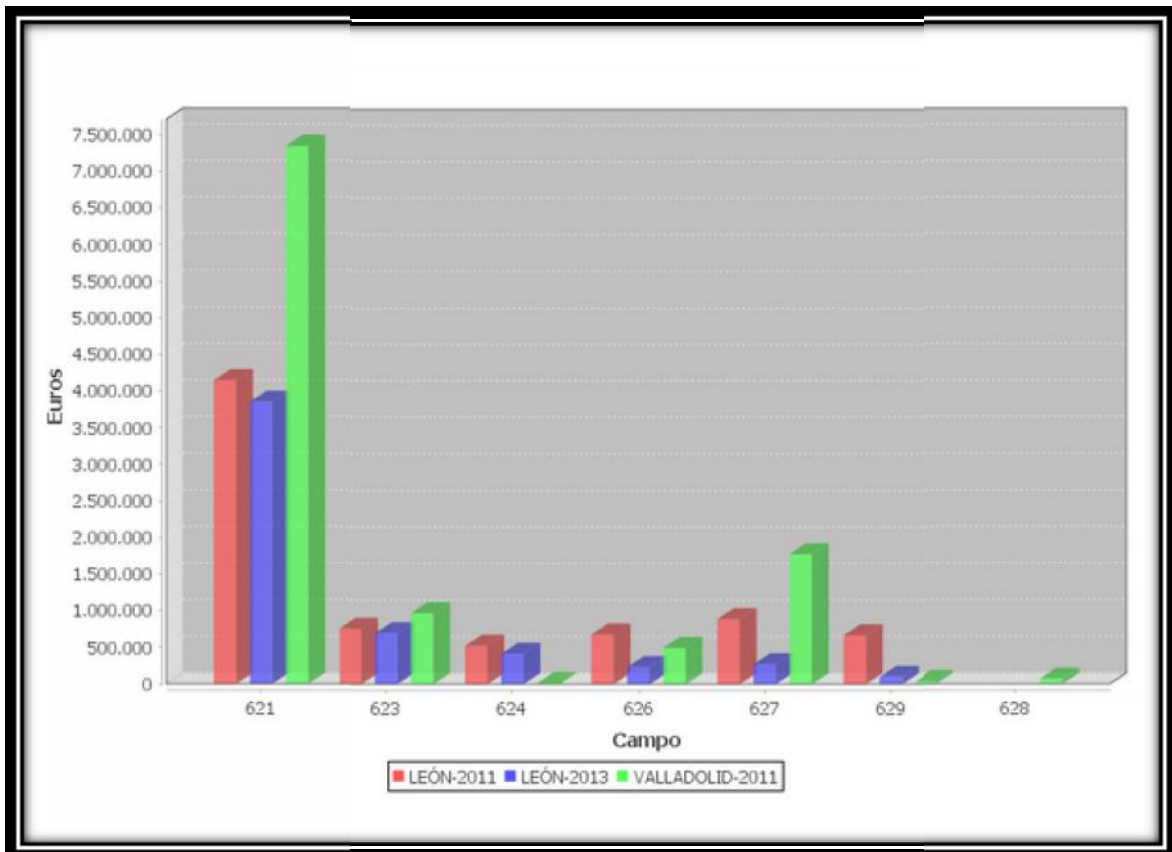



IMAGEN Página comparación grafica detallada

### 5.7.9.- AYUDA AL USUARIO DE LA APLICACIÓN

En todas las páginas de la aplicación Web existen enlaces en el icono  y pulsando sobre el nos aparecerá la página de ayuda en otra ventana o pestaña.

### 5.7.10.- CONFIGURACIÓN DE LA BASE DE DATOS

Esta página nos muestra un formulario en el que se podrá configurar la base de datos que usará la aplicación Web, introduciendo los parámetros y pulsando enviar.



The image shows a web form titled "Configuración de la Base de Datos" with a blue question mark icon. The form contains several input fields and a submit button. The fields are labeled as follows:

Label	Value
Clase de Conexion	com.mysql.jdbc.Driver
servidor	jdbc:mysql://localhost
usuario	root
password	
bbdd	PROY


At the bottom left of the form is a button labeled "Enviar consulta".

IMAGEN Página Formulario Configuración de la BBDD

### 5.7.11.-CONFIGURACIÓN DE LOS PARÁMETROS DE LECTURA

Esta página nos muestra un formulario en el que se podrá configurar la base de datos que usará la aplicación Web, introduciendo los parámetros y pulsando enviar.

## Parametros de lectura del pdf



En esta pagina se podra configurar ciertos parametros de lectura de PDF.

Seleccione el parametro que desea modificar

Cadena previa al organismo al que pertenece el Presupuesto

Cadena previa al organismo al que pertenece el Presupuesto

**GUARDAR PARÁMETROS**

IMAGEN Página formulario parámetros de lectura

### 5.7.11.- OTROS

Para los periodos en los que la aplicación web está “trabajando”, consultando a la base de datos, descargando presupuestos, etc. La aplicación muestra una página con un mensaje de espera.



IMAGEN Página de Espera

## 5.8 DISEÑO LA LECTURA DEL BOLETIN OFICIAL.

### 5.8.1 FLUJO DE LECTURA DEL PDF

Se considera la parte más importante y delicada del proyecto por lo que a continuación describiremos detalladamente el proceso de Lectura.

En un primer momento para la lectura del boletín oficial se consideró un documento de referencia. Es una resolución que emitió el BOE el 29/1/2014 *de la Dirección General de Presupuestos, por la que se establecen los códigos que definen la clasificación económica*. En él se dictan normas para la elaboración de los Presupuestos Generales del Estado para 2014, regula, en su artículo 6, la estructura de los Presupuestos Generales del Estado.

Pero al comparar los códigos y los datos con los presupuestos que nos interesan leer, si que sirve para una idea general pero en cuestión específica difiere en bastantes cosas por lo que al final se ha optado por generar otro modelo sin seguir específicamente esta resolución.

Comparativa de la resolución y el modelo de los datos que se quieren leer.

#### 1. Convertir el PDF en texto legible:

Creando una clase auxiliar llamada EntidadPDF y con la herramienta PDFBox se consigue a través del método "parse" obtener el texto contenido en el pdf, a partir de la URL donde se aloja dicho pdf en formato texto plano, que después se procesará para obtener la información que necesitamos.

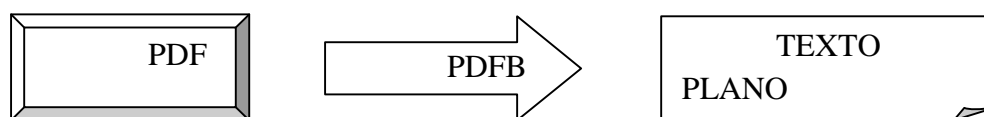


DIAGRAMA Flujo de Lectura Paso 1

#### 2. Separar los ingresos y los gastos:

A través del estudio de los formatos de distintos presupuestos, se ha logrado distinguir en el texto plano donde empiezan y terminan los ingresos y los gastos; que seguidamente se separan para tratarlos según su condición.

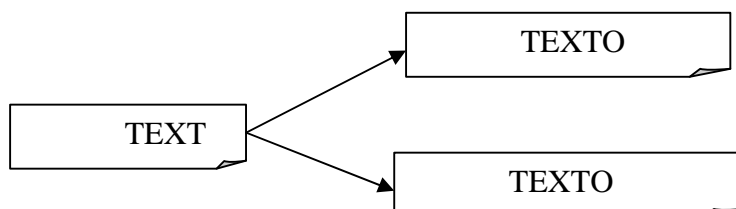


DIAGRAMA Flujo de Lectura Paso 2

**3. La obtención de los Módulos:**

Tras el estudio de los presupuestos se ha llegado a la conclusión de que la estructura de los presupuestos se compone de módulos o trinomios. Estos módulos se componen de 3 características un código una descripción y un valor.

El siguiente paso es obtener estos módulos leyendo el texto plano tanto de ingresos como de gastos. Obteniendo una lista simple con todos estos elementos.



DIAGRAMA Flujo de Lectura Paso 3

**4. Estructurar el presupuesto:**

A partir de tener ya la lista de estos módulos que componen el presupuesto el siguiente paso es organizarlos en una estructura que equivalga al presupuesto ya que estos módulos son de varios tipos: capítulos, artículos, conceptos y subconceptos y tienen la jerarquía que un capítulo contiene varios artículos, los artículos se componen de conceptos y estos conceptos pueden estar compuestos de subconceptos a su vez. Lo que viene a ser una estructura anidada.



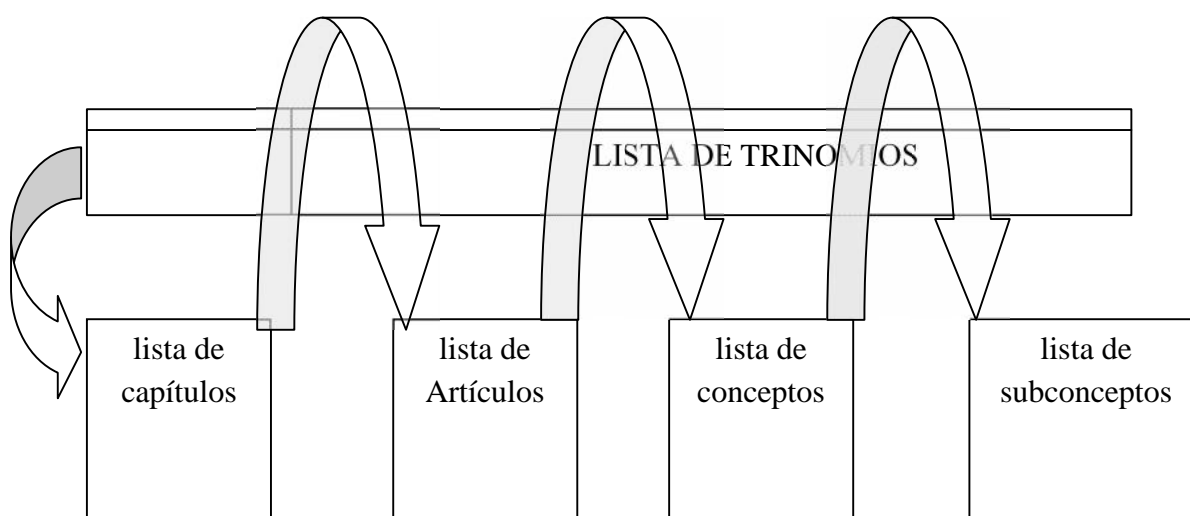


DIAGRAMA Flujo de Lectura Paso 4

### 5.2.3 RELACIONES DE DATOS

Los códigos de los módulos, también tienen un patrón que facilita la posterior estructuración y se organizan de la siguiente forma: Los capítulos tienen un código numérico que va del 1 al 9, que haremos referencia a él como "C". Los códigos de los artículos tienen el siguiente formato: "CA" donde C es el código a el capítulo que pertenece y A es un número comprendido entre el 0 y 9. Los Conceptos siguen un código con el formato "AAN" donde AA son los dígitos correspondientes al artículo que corresponde el concepto y N es un valor numérico comprendido entre el 0 y el 9. El código que corresponde a los subconceptos sigue el siguiente patrón NNN.SS donde NNN es el código del concepto al que pertenece el subconcepto y SS es el dígito del 00 a 99 correspondiente al subconcepto concreto.

Por lo que a partir de la lista de los módulos o trinomios obtenemos los capítulos, para cada capítulo conseguimos los artículos correspondientes. Como cada artículo tiene distintos conceptos se seleccionan y se anidan en la estructura a cada artículo e igualmente se hará la misma operación con los conceptos y subconceptos.

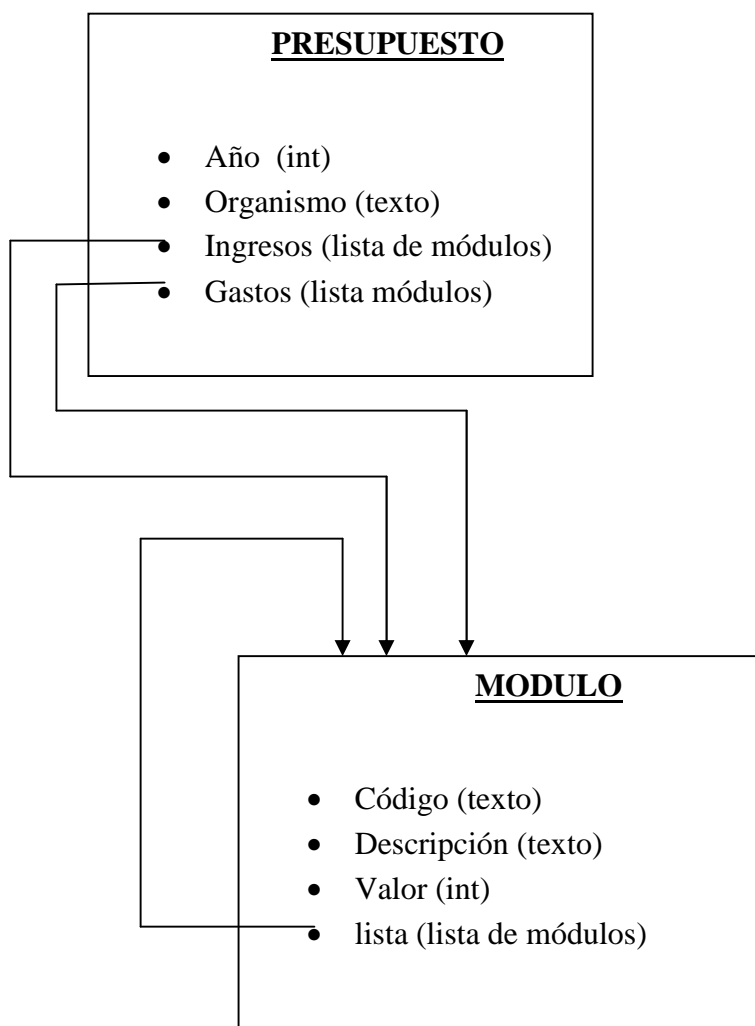


DIAGRAMA Relaciones de Datos Estructurales

#### 5.8.4 ESTRUCTURA DEL PRESUPUESTO

Mediante el estudio de diversos presupuestos universitarios hemos conseguido homogeneizar una estructura que representa el presupuesto.

Cada presupuesto tiene un único organismo que hace referencia al propietario del presupuesto, es decir a quien hace referencia este.

Otro dato a tener en cuenta es la fecha del presupuesto es decir a que periodo vigente corresponden los datos económicos obtenidos en el boletín oficial son anuales por lo que el dato que leemos y almacenamos no es otro que el año.

Cada presupuesto tiene una Lista de ingresos y una lista de gastos estas dos listas son estructuralmente hablando iguales; son listas de capítulos. Cada capítulo de estas

listas tiene un código, una descripción, un valor, y una lista de artículos. El valor es igual a la suma de todos los valores de la lista de artículos.

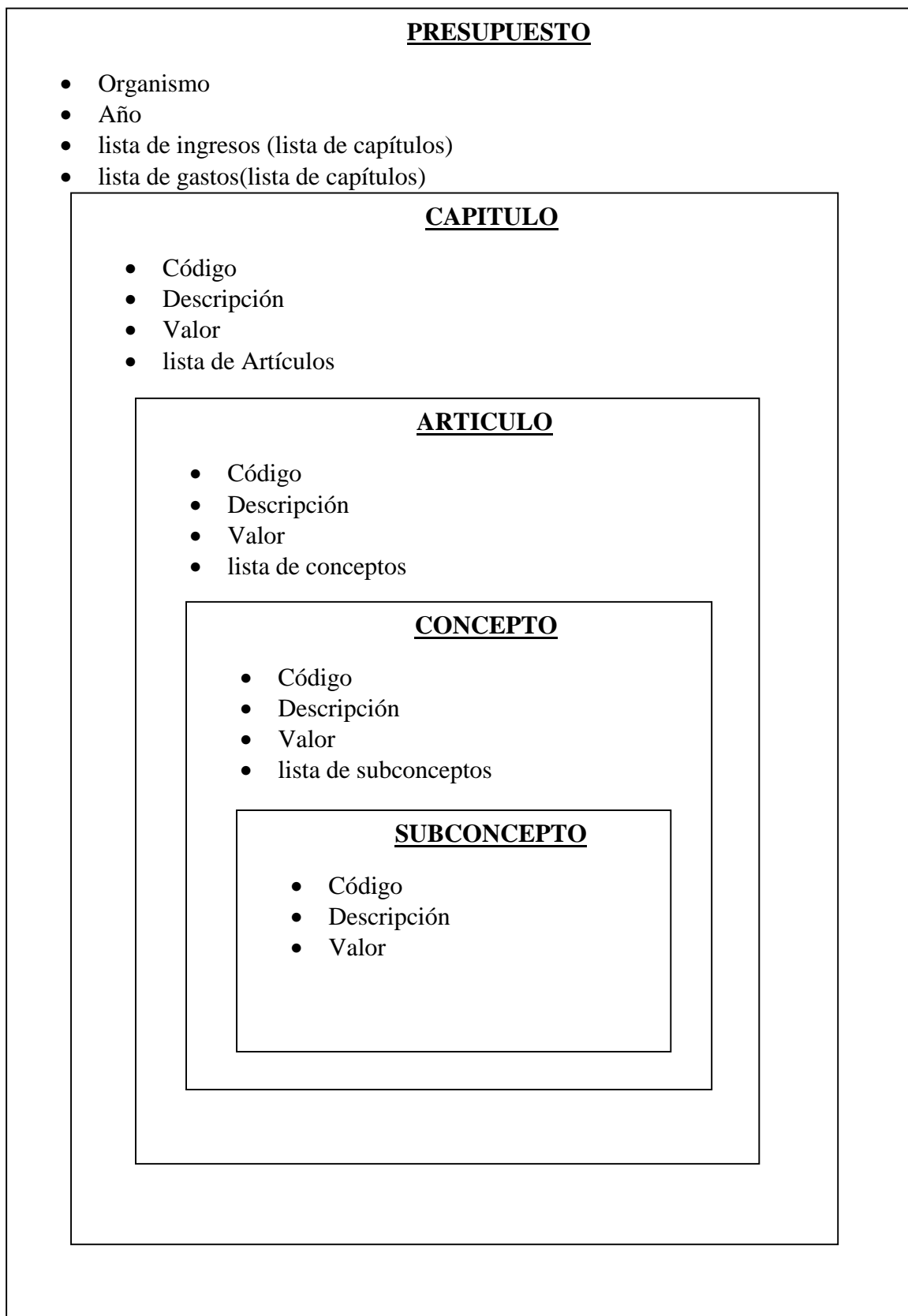


DIAGRAMA Estructura de un Presupuesto

### **5.8.5 PERSISTENCIA DE DATOS**

El programa se ha diseñado para que los presupuestos leídos puedan ser almacenados en una base de datos (en principio de manera local) para su posterior uso.

Se ha decidido permitir configurar la aplicación para que sea posible usar varios tipos de bases de datos

Contiene todas las funciones y protecciones requeridas en una base de datos.

Es muy fácil de integrar con una aplicación de java ya que es 100% java.

El diagrama de entidad relación es muy sencillo.





## **6.-PLAN DE VALIDACIÓN Y VERIFICACIÓN DEL SOFTWARE**

El objetivo de este documento es presentar la definición de un plan de pruebas que permitan comprobar que el sistema funciona correctamente, cumpliendo con las especificaciones dadas por el cliente en su solicitud.

La fase de pruebas es una etapa más del ciclo de vida de un programa. Sin embargo, en esta etapa se ha invertido una gran cantidad de tiempo y esfuerzo para poder subsanar los errores y pequeños detalles que surgieron en dicha etapa. Por esta razón se ha convertido en una fase clave en cuanto al funcionamiento de la aplicación se refiere.

El plan de pruebas propuesto detalla todas las características a comprobar del sistema, junto con los procedimientos a llevar a cabo para realizar dichas comprobaciones. Cada una de las pruebas a realizar debe llevar asociada una descripción completa, los pasos a seguir para su realización y los criterios para la aceptación de la prueba. Asimismo se especifica también el entorno necesario para la realización de las pruebas, y el proceso de realización de informes de las pruebas realizadas.

### **6.1.-TAREAS Y ELEMENTOS DE PRUEBA**

Aquí estableceremos que elementos del sistema van a someterse a prueba. Todas las funcionalidades del sistema deben ser sometidas a pruebas, pero es necesario identificar cada una de las partes involucradas en el proceso de verificación y validación. De esta forma se puede controlar de forma más exhaustiva las pruebas de carácter que se han de realizar, tanto de elementos particulares, como generales.

De esta forma se identifican los siguientes elementos que serán sometidos a pruebas:

- Modulo de lectura del PDF: Como componente principal de la aplicación se realizaran pruebas para comprobar el correcto funcionamiento de lectura y procesamiento de la información.

- Modulo de almacenamiento y recuperación de datos en la bbdd: Se harán pruebas para comprobar que se almacenan correctamente la información en la base de datos. Y que la aplicación puede recuperar la información de manera correcta.
- Modulo de aseguración de la integridad de la información: Se debe probar que se generan correctamente los documentos que nos permitirán comprobar la integridad de datos. También se probará que estos documentos son recuperables (descargables) por el usuario.
- Modulo de comparación de información almacenada en la base de datos: Se probará que la aplicación nos muestra correctamente los presupuestos almacenados así la información relativa a estos presupuestos.
- Modulo de exportación de la información: Se realizarán pruebas para comprobar el correcto funcionamiento de la exportación de datos, tanto la propia exportación como la creación del registro de datos exportados.

Desarrollar un sistema de calidad que cumpla con el resultado esperado por el cliente requiere definir un conjunto de pruebas lo más completo posible, de forma que cada una de las funcionalidades del sistema se compruebe correctamente.

Por lo tanto, las tareas definidas para desarrollar las pruebas de este proyecto son las siguientes:

- **Definición de la Prueba:** Se debe realizar una descripción acerca de la prueba, especificando cual es el objetivo de la misma, qué elementos del sistema serán sometidos a prueba y las características que debe tener el sistema para que la prueba pueda ser ejecutada. Además se detallará cada uno de los pasos a realizar para la correcta ejecución de la prueba.
- **Ejecución de la prueba de aceptación:** Observando el comportamiento del sistema y los resultados obtenidos se podrá determinar si la prueba es superada con éxito o no.
- **Registro de la prueba:** Deberá generarse una tabla con la información acerca de la prueba, los resultados y los posibles fallos encontrados.

## 6.2.- ENTORNO Y FORMATO DE PRUEBAS

Para que las pruebas definidas puedan ser ejecutadas de manera satisfactoria y produzcan los mismos resultados que en el entorno en el que el sistema real debe ser implantado, es necesario definir las características que debe tener el sistema donde se van a realizar las pruebas.



En el caso de este proyecto, el entorno de pruebas tendrá las mismas características que el entorno de implantación del sistema, por lo que en este apartado no es necesario definir nuevas necesidades para el entorno.

#### CRITERIO DE ACEPTACIÓN O RECHAZO DE UNA PRUEBA

Al realizar la ejecución de cada uno de los casos de prueba definidos en el plan, es necesario determinar si la prueba se considera válida porque el sistema funciona como debería, o si presenta algún tipo de fallo en alguna de las funcionalidades requeridas.

El criterio establecido en este proyecto para considerar una prueba como *Aceptada*, considera necesario que dicha prueba se ejecute según lo definido, es decir, el sistema tendrá que responder de la manera exacta que detalle el caso de prueba correspondiente. En el caso de que el sistema produzca algún tipo de fallo o proporcione un resultado diferente al esperado, la prueba será *Rechazada*.

#### FORMATO DE LAS PRUEBAS

A continuación se describe el formato de tabla utilizado para la especificación de las pruebas y los distintos campos que contiene:

- **Identificador:** Nombra de manera única a la prueba siguiendo el formato PR-XXX, donde XXX tomará valores numéricos dentro del rango 000-999.
- **Descripción:** Especifica el objetivo de la prueba.
- **Precondiciones:** Describe el estado en el que se debe encontrar el sistema de forma previa a la realización de la prueba.
- **Pasos a seguir:** Establece la secuencia de pasos que debe realizar el responsable de la prueba sobre el sistema.

### 6.3.- PRUEBAS

#### PRUEBAS CAJA BLANCA

Las pruebas de caja blanca están dirigidas a las funciones internas de un módulo de la aplicación. Este tipo de prueba se ha llevado a cabo en primer lugar, sobre un módulo concreto, para luego realizar las pruebas de caja negra sobre varios subsistemas.

Estas pruebas se han ido realizando a la que vez que se iba desarrollando el código, de forma que cada módulo ha sido probado de forma independiente al resto.

A medida que se ha ido probando el código se ha intentado recorrer todos los caminos posibles. Como por ejemplo la comprobación de bucles o revisando el uso de las variables.

### PRUEBAS CAJA NEGRA

Las pruebas de caja negra se centran en el estudio de las entradas y salidas sin tener en cuenta su funcionamiento interno, comprobando que los requisitos se han cumplido.

No son una alternativa a las pruebas de caja blanca, sino que se han realizado las dos por separado y de forma complementaria, para detectar diferentes tipos de errores.

A continuación se muestran las pruebas realizadas:

<b>Identificador</b>	<b>PR-001</b>
<b>Descripción</b>	Arrancar la aplicación correctamente
<b>Precondiciones</b>	Ninguna
<b>Pasos a seguir</b>	Escribir la URL en la barra de direcciones del navegador. Comprobar que la aplicación Web arranca

TABLA Prueba PR-001

<b>Identificador</b>	<b>PR-002</b>
<b>Descripción</b>	Comprobación de enlaces.
<b>Precondiciones</b>	Ninguna
<b>Pasos a seguir</b>	Hacer clic en los distintos enlaces que se cargan en la página. Comprobar que todos los enlaces redirigen de manera correcta a la página/sección correspondiente.

TABLA Prueba PR-002

<b>Identificador</b>	<b>PR-003</b>
<b>Descripción</b>	Lectura de una URL correcta.
<b>Precondiciones</b>	Ninguna (página de leer presupuestos)
<b>Pasos a seguir</b>	Introducir en el formulario la url que enlaza con un presupuesto de una universidad de Castilla y león publicado en el BOCYL. Pulsar en el botón “leer” Comprobar que se ha cargado correctamente la página donde nos muestra el presupuesto leído.

TABLA Prueba PR-003

<b>Identificador</b>	<b>PR-004</b>
<b>Descripción</b>	Lectura de una URL falsa
<b>Precondiciones</b>	Ninguna (página de leer presupuestos)
<b>Pasos a seguir</b>	Introducir en el formulario una cadena de texto que no corresponda a ninguna URL Pulsar el botón “leer” Comprobar aparece una página que indica que la URL es incorrecta.

TABLA Prueba PR-004

<b>Identificador</b>	<b>PR-005</b>
<b>Descripción</b>	Lectura de una URL incorrecta
<b>Precondiciones</b>	Ninguna (página de leer presupuestos)
<b>Pasos a seguir</b>	Introducir en el formulario una URL que no apunte a ninguna dirección verdadera. Pulsar el botón “leer” Comprobar aparece una página que indica que la URL es incorrecta.

TABLA Prueba PR-005

<b>Identificador</b>	<b>PR-006</b>
<b>Descripción</b>	Lectura de una URL incorrecta
<b>Precondiciones</b>	Ninguna (página de leer presupuestos)
<b>Pasos a seguir</b>	Introducir en el formulario una URL que a una página que no contenga ningún archivo en formato pdf. Pulsar el botón “leer” Comprobar aparece una página que indica que la URL es incorrecta.

TABLA Prueba PR-006

<b>Identificador</b>	<b>PR-007</b>
<b>Descripción</b>	Lectura de una URL incorrecta
<b>Precondiciones</b>	Ninguna (página de leer presupuestos)
<b>Pasos a seguir</b>	<p>Introducir en el formulario una URL que no apunte a archivo en formato pdf que no contenga ningún presupuesto valido</p> <p>Pulsar el botón “leer”</p> <p>Comprobar aparece una página que indica que la URL es incorrecta.</p>

TABLA Prueba PR-007

<b>Identificador</b>	<b>PR-008</b>
<b>Descripción</b>	Visualización del presupuesto leído
<b>Precondiciones</b>	Un presupuesto en formato pdf ha sido leído y procesado por la aplicación.
<b>Pasos a seguir</b>	<p>Se comprueba que la cantidad total del presupuesto es la que indica en el pdf.</p> <p>Se comprueba que el año y la universidad que nos muestra la aplicación son los correctos que sacados del pdf.</p> <p>Pulsar en “GASTOS”</p> <p>Comprobar que se despliega un submenú con los capítulos que componen los gastos del presupuesto, y comprobar que estos corresponden al pdf leído, que no falta ninguno</p> <p>Comprobar que aparece un grafico que representa los distintos capítulos de gastos.</p> <p>Hacer clic en un capitulo desplegado anteriormente.</p> <p>Comprobar que aparecen los artículos dependientes del capítulo sin faltar ninguno.</p> <p>Comprobar que aparece el grafico con estos artículos.</p> <p>Hacer la misma operación para los conceptos y los subconceptos, de los gastos y también de los “ingresos”</p>

TABLA Prueba PR-008

<b>Identificador</b>	<b>PR-009</b>
<b>Descripción</b>	Grabación del presupuesto en la base de datos
<b>Precondiciones</b>	Un presupuesto ha sido leído y procesado por la aplicación, y este presupuesto no existe en la base de datos. Tener una base de datos correctamente configurada en nuestra aplicación.
<b>Pasos a seguir</b>	Hacer clic en el enlace que aparece con el texto “Guardar presupuesto” Comprobar que no aparece ningún error. Comprobar en la base de datos que se han introducido el nuevo presupuesto y los datos de este en las tablas que corresponden a cada campo.

TABLA Prueba PR-009

<b>Identificador</b>	<b>PR-010</b>
<b>Descripción</b>	Grabar presupuesto en base de datos incorrecta
<b>Precondiciones</b>	Un presupuesto ha sido leído y procesado por la aplicación, y este presupuesto no existe en la base de datos. No tener una base de datos correctamente configurada en nuestra aplicación.
<b>Pasos a seguir</b>	Hacer clic en el enlace que aparece con el texto “Guardar Presupuesto” Comprobar que aparece en la pantalla el mensaje que dice que no hay base de datos correctamente configurada.

TABLA Prueba PR-010

<b>Identificador</b>	<b>PR-011</b>
<b>Descripción</b>	Ver la pantalla de explicación de la garantía de integridad de información.
<b>Precondiciones</b>	Un presupuesto ha sido leído y procesado por la aplicación
<b>Pasos a seguir</b>	Hacer clic en el enlace que aparece con el texto “Garantía de integridad de información”. Comprobar que aparece la página donde se explica el sistema. de garantía de integración de datos.

TABLA Prueba PR-011

<b>Identificador</b>	<b>PR-012</b>
<b>Descripción</b>	Ver pantalla interactiva de explicación del flujo de lectura y procesamiento del presupuesto.
<b>Precondiciones</b>	Un presupuesto ha sido leído y procesado y se ha accedido a la página de explicación.
<b>Pasos a seguir</b>	Hacer clic desde la página de explicación en el enlace que lleva a la página interactiva. Comprobar que se abre la página con la imagen correctamente. Comprobar que tras situar el ratón encima de las flechas de flujo aparece una descripción de la acción

TABLA Prueba PR-012

<b>Identificador</b>	<b>PR-013</b>
<b>Descripción</b>	Comprobación de las HASH
<b>Precondiciones</b>	Un presupuesto leído y procesado y la página de seguridad interactiva abierta
<b>Pasos a seguir</b>	Hacer clic en los distintos iconos de MD5. Comprobar que en cada icono aparece un mensaje con el código Hash MD5 que correspondería a el archivo de donde se ha sacado el código. Comprobar que los códigos hash MD5 coinciden con sus pares correspondientes.

TABLA Prueba PR-013

<b>Identificador</b>	<b>PR-014</b>
<b>Descripción</b>	Comprobación archivos descargables
<b>Precondiciones</b>	Un presupuesto leído y procesado y la página de seguridad interactiva abierta
<b>Pasos a seguir</b>	Hacer clic en los iconos de TXT y de PDF que aparecen en la imagen. Comprobar que se lanza una ventana en el navegador en la que podemos descargar el archivo en el PC cliente. En la mayoría de navegadores da la opción de descargar o abrir con un programa determinado. Comprobar que el texto y la composición de estos archivos es correcto.

TABLA Prueba PR-014

<b>Identificador</b>	<b>PR-015</b>
----------------------	---------------

<b>Descripción</b>	Comprobación de los listados de presupuestos
<b>Precondiciones</b>	Ninguna(página de comparación)
<b>Pasos a seguir</b>	Ir a la página de comparación de presupuestos. Comprobar que se listan de manera ordenada por organismo emisor y fecha todos los presupuestos almacenados en la base de datos.

TABLA Prueba PR-015

<b>Identificador</b>	<b>PR-016</b>
<b>Descripción</b>	Comprobación de selección de presupuestos
<b>Precondiciones</b>	Página de comparación
<b>Pasos a seguir</b>	Click en los distintos checkbox que se adjuntan junto los presupuestos listados. Comprobar que se marcan los checkbox. Comprobar que solo deja marcar como mucho 4 presupuestos, ya que es el límite de comparación y que alerta con un mensaje si marcas más de 4.

TABLA Prueba PR-016

<b>Identificador</b>	<b>PR-017</b>
<b>Descripción</b>	Comparación de un solo presupuesto
<b>Precondiciones</b>	Haber escogido un presupuesto en la página de Selección de presupuestos y haber seleccionado solamente uno
<b>Pasos a seguir</b>	Comprobar que el total es el correcto Seleccionar una opción de la lista desplegable. Comprobar que aparece un grafico de tarta, aparecen los datos desplegados, y un nuevo desplegable con la información que corresponde a el presupuesto. Además de la información listada

TABLA Prueba PR-017

<b>Identificador</b>	<b>PR-018</b>
<b>Descripción</b>	Comparacion de dos presupuestos
<b>Precondiciones</b>	Haber marcado en la página de selección de presupuestos dos de los checkbox disponibles.
<b>Pasos a seguir</b>	Comprobar que el total es el correcto a cada presupuesto Seleccionar una opción de la lista desplegable. Comprobar que aparece un grafico de barras , con la información correctamente comparada, aparecen los datos desplegados, y un nuevo desplegable con la información que corresponde a el presupuesto.

TABLA Prueba PR-018

<b>Identificador</b>	<b>PR-019</b>
----------------------	---------------

<b>Descripción</b>	Comparacion de tres presupuestos
<b>Precondiciones</b>	Haber marcado en la página de selección de presupuestos tres de los checkbox disponibles.
<b>Pasos a seguir</b>	<p>Comprobar que el total es el correcto a cada presupuesto</p> <p>Seleccionar una opción de la lista desplegable.</p> <p>Comprobar que aparece un grafico de barras , con la información correctamente comparada, aparecen los datos desplegados, y un nuevo desplegable con la información que corresponde a el presupuesto.</p>

TABLA Prueba PR-019

<b>Identificador</b>	<b>PR-020</b>
<b>Descripción</b>	Comparacion de cuatro presupuestos
<b>Precondiciones</b>	Haber marcado en la página de selección de presupuestos cuatro de los checkbox disponibles.
<b>Pasos a seguir</b>	<p>Comprobar que el total es el correcto a cada presupuesto</p> <p>Seleccionar una opción de la lista desplegable.</p> <p>Comprobar que aparece un grafico de barras , con la información correctamente comparada, aparecen los datos desplegados, y un nuevo desplegable con la información que corresponde a el presupuesto.</p>

TABLA Prueba PR-020

<b>Identificador</b>	<b>PR-021</b>
<b>Descripción</b>	Descargar información a un archivo csv
<b>Precondiciones</b>	Haber hecho una comparación de presupuestos y teniendo esta en pantalla.
<b>Pasos a seguir</b>	<p>Hacer clic en el botón donde pone “Descargar documento”</p> <p>Introducir un DNI o NIE correcto en el formulario emergente y pulsar aceptar.</p> <p>Seleccionar en la ventana emergente si quieres descargar el archivo o abrirlo con un programa directamente.</p> <p>Comprobar que los datos descargados en el CRV son exactamente los que corresponden a los presupuestos.</p> <p>Comprobar que se ha almacenado en la base de datos correctamente los siguientes datos:</p> <p>La fecha y hora de la descarga</p> <p>Código HASH MD5 formada con el DNI o NIE introducido</p> <p>Código HASH MD5 generado a partir del propi CSV.</p>

TABLA Prueba PR-021

<b>Identificador</b>	<b>PR-022</b>
----------------------	---------------



<b>Descripción</b>	Configurar Base de datos
<b>Precondiciones</b>	Acceder a la pantalla de configuración de bbdd
<b>Pasos a seguir</b>	Rellenar el formulario con la correspondiente información de una base de datos en línea. Hacer clic en el botón “Enviar” Comprobar que ha conectado, aparece un mensaje en pantalla.

TABLA Prueba PR-022

<b>Identificador</b>	<b>PR-023</b>
<b>Descripción</b>	Configurar Base de Datos incorrectamente
<b>Precondiciones</b>	Acceder a la pantalla de configuración de bbdd
<b>Pasos a seguir</b>	Rellenar el formulario con la correspondiente información errónea de una base de datos. Hacer clic en el botón “Enviar” Comprobar que aparece un mensaje en pantalla con el error.

TABLA Prueba PR-023

<b>Identificador</b>	<b>PR-024</b>
<b>Descripción</b>	Configurar Parámetros de lectura del PDF
<b>Precondiciones</b>	Acceder a la pantalla de configuración de parámetros
<b>Pasos a seguir</b>	Rellenar el formulario con los datos de los parámetros. Hacer clic en el botón “Enviar” Comprobar que en el archivo Parametros.txt se han almacenado correctamente los parámetros introducidos.

TABLA Prueba PR-024



## **7.- CONCLUSIONES Y POSIBLES**

### **AMPLIACIONES**

#### **7.1.- CONCLUSIONES**

Con la realización de este proyecto se pretendía alcanzar una serie de objetivos marcados al inicio del mismo, tarea que se ha conseguido satisfactoriamente.

En primer lugar, se ha creado un sistema que proporciona la funcionalidad deseada.

Se ha seguido la planificación establecida al principio del proyecto, salvo en ligeras modificaciones, cumpliendo los plazos de entrega establecidos al comienzo del proyecto.

Se han puesto en práctica los conocimientos adquiridos a lo largo de la carrera, sobre todo en cuanto al uso de metodologías, programación orientada a objetos y gestión de proyectos software.

Se puede concluir finalmente, que se ha logrado alcanzar el objetivo de desarrollar un proyecto software completo, adquiriendo nuevos conocimientos acerca de los roles a desarrollar en este tipo de proyectos, y comprobando la dificultad existente en cuanto a esfuerzo, dedicación y coste de desarrollar un proyecto de forma correcta.

Personalmente me ha gustado trabajar en este proyecto, por la experiencia de crear algo nuevo y original y ofrecer facilidades a futuros usuarios. Además aprender sistemas burocráticos referentes a los boletines oficiales que desconocía.

#### **7.2.- POSIBLES AMPLIACIONES**

Algunas de las posibles ampliaciones que se podrían realizar en un futuro se han ido comentando a lo largo del documento, pero se resumen en las siguientes:

- En un principio se pensó en desarrollar una aplicación que fuera capaz de leer cualquier presupuesto publicado en el BOE debido a *Resolución de 20 de enero de 2014, de la Dirección General de Presupuestos, por la que se establecen los códigos que definen la clasificación económica*. Que establece unas normas básicas que se siguen para unificar los

boletines oficiales. Pero debido a que los presupuestos anteriores no siguen estas normas se ha decidido especializar la lectura a solo los presupuestos de las universidades. Pero aun así el formato de un presupuesto u otro era bastante distinto por lo que nos hemos centrado en la lectura de los presupuestos de las universidades de Castilla y León. Por lo que la primera posible ampliación sería que se pudiera leer presupuestos de todas las universidades españolas. Y otra ampliación ya, sería que la aplicación fuera capaz de procesar cualquier presupuesto. Estas ampliaciones no se verían demasiado complicadas si todos los presupuestos siguieran fielmente las normas de la Resolución del 20/01/2014.

- También es sabido que la junta de Castilla y León algunas veces publica boletines oficiales de corrección de presupuestos, una posible ampliación del programa sería que fuera capaz de leer estos documentos, procesar la información y actualizar los datos de los presupuestos que corrigiera.
- Otra posible ampliación pudiera ser un control de cuentas de usuarios. Es decir que cada usuario pudiera registrarse, y tener su propia cuenta con sus propios presupuestos procesados, pudiendo trabajar con ellos, procesar nuevos, eliminar o modificar presupuestos ya procesados etc.





## 8.- BIBLIOGRAFÍA

- Titulo: Programador certificado Java2:  
Autores: Antonio Martin Sierra  
Editorial: AlfaOmega Grupo 2008.
- Titulo: Programación en Java:  
Autores: Jesús Sánchez Allende / Gabriel Huescas Fernández/ Baltasar Fernández Majón/Pilar moreno García.  
Editorial: MC GRAW HILL 2ªedición.
- Titulo: [Murach's Java servlets and JSP : training & reference /](#)  
Autores: Joel [Fresno, CA]  
Editorial: Mike Murach & Associates, cop. 2008
- Titulo: [JavaScript / David Sawyer](#) McFarland  
Autor: David Sawyer Madrid  
Editorial: Anaya Multimedia, 2009.

### ENLACES

- HTML
  - <http://lineadecodigo.com>
- PDFBOX:
  - <https://pdfbox.apache.org>.
  - <http://www.adictosaltrabajo.com>
- JAVA
  - <https://unpocodejava.wordpress.com/>
  - <http://www.lawebdelprogramador.com>
  - <http://geekalt42.net>
  - <http://www.printmyfolders.com>.
  - <http://www.Syncrom.com>
  - <http://www.aprendeaprogramar.es>
  - <http://java.foroactivo.com>
  - <http://leanpub.com>
  - <http://aulafacil.com>
- Boletines oficiales
  - <http://contencioso.es/>
  - <http://bocyl.jcyl.es>





## 9.-ANEXOS

### 9.1.-ANEXO CONTROL DE VERSIONES

El Objetivo de este control de versiones es documentar la gestión de los cambios que se realizan, en el producto de este proyecto, explicando los bugs corregidos y las nuevas funcionalidades añadidas sobre este producto.

#### Versión 1.0

Esta es la primera versión del software creado, Este software ya tiene implementadas correctamente todas las funcionalidades de la aplicación cumpliendo con los requisitos iniciales. Ésta versión es la primera versión estable sin bugs críticos y que cumple con todos los requisitos y funciones diseñados

#### Versión 1.1

Estos son los cambios que realizan en esta versión:

C-1: Se modifica la página web del control de integridad de datos, En ella se explicara de forma detallada el proceso de lectura y el sistema de seguridad en formato de texto. Las páginas modificadas son Pag\_seg.jsp y PagSeg.jsp

C-2: Se modifica la página de inicio para que muestre una introducción básica a lo que hace la aplicación. La página modificada es index.jsp

C-3 Se modifica la página de lectura del pdf para que muestre un mensaje de espera para la lectura del pdf. La página modificada es leer.html

C3.1 Se crean páginas de espera para que muestren mensaje de espera para todos los procesos que pueden tardar mucho tiempo.

C-4 Se modifica la página de ver el PDF para que muestre otro mensaje en los casos de que se intente explorar y ver el grafico de un nodo que no tenga subnodos "hijos" la página modificada es: verPre.jsp

C-5 Se añade una página de ayuda a la aplicación.

~~C-6 Se modifica la aplicación para que no se pierda la página que nos muestra la información del presupuesto pdf recién procesado, al navegar por otros menús.~~

### Versión 1.2.

F-1 Se añade una nueva funcionalidad. Esta consiste en la opción de poder exportar datos en formato csv para poder visualizarlos en una hoja de datos de Excel.

C-1 Modificación de la página ComparaDetalle para que a la vez que se actualiza dinámicamente genere un fichero csv y muestre un enlace o botón para poder descargarlo.

C-2 Establecer un sistema para garantizar la integridad de que los datos exportados corresponden a los datos originales. Para ello almacenamos en la base de datos la fecha el código hash obtenido del DNI introducido del usuario y el código hash obtenido del archivo csv.

Para esto es necesario crear una nueva tabla en la base de datos. Que almacene 3 campos fecha y hora, hash del DNI y hash del csv. También es necesario crear una pantalla para que el usuario introduzca el DNI.

C-3 cambiar la apariencia de la página tbPresupuestos.jsp para que muestre los presupuestos ordenados en una tabla por universidad y por año. También se optimiza para que la carga de los presupuestos sea más rápida.

### Versión 2.0

C.1 Cambiamos el servidor de aplicaciones Glashfish por un servidor Tomcat, porque se ajusta más a las necesidades del proyecto.

C2. Cambiamos la apariencia de todo el sitio Web, usamos otra hoja de estilos de apariencia más profesional.

### Versión 2.0.1

C.1.Se Crea un proceso de validación de DNI-NIE. Para la funcionalidad de exportar datos el Sistema comprueba que se ha introducido un DNI valido, y muestra alerta de error en el caso de meter.

## 9.2 INDICES DE DIAGRAMAS FIGURAS E IMAGENES

### 9.2.1.-DIAGRAMAS

.Secuencia Comparar detalle de presupuestos .....	62
.Secuencia Comprobar integridad .....	60
.Secuencia Exportar Datos.....	63
.Secuencia Guardar.....	59
.Secuencia Selección Presupuestos a Comparar.....	61
Arquitectura Cliente Servidor.....	56
Diagrama de Casos de Uso.....	42
Diagrama De Clases.....	65
Entidad-Relacion .....	79
Estructura de un Presupuesto.....	100
Flujo de Lectura Paso 1 .....	95
Flujo de Lectura Paso 2 .....	96
Flujo de Lectura Paso 3 .....	96
Flujo de Lectura Paso 4 .....	97
Funcionamiento General.....	13
Modelo Relaccional .....	81
RBS.....	23
Relaciones de Datos Estructurales .....	98
Secuencia lectura presupuesto .....	58

### 9.2.2.-IMAGENES

Página comparación de presupuestos .....	89
Página comparación detallada .....	90
Página comparación grafica detallada .....	91
Página de Espera.....	94
Página Formulario Configuracion de la BBDD .....	92
Página formulario interactivo .....	90
Página formulario parámetros de lectura.....	93
Página inicio .....	84
Página interactiva de comprobación de integridad.....	87
Página Leer Presupuesto.....	85
Página Selección de presupuestos .....	88
Página Ver Presupuesto Procesado.....	86

**9.2.3.-TABLAS**

.coste total corregido.....	37
.Coste total previsto .....	32
Caso de uso CU-01 .....	43
Caso de uso CU-02 .....	44
Caso de uso CU-03 .....	44
Caso de uso CU-04 .....	45
Clase Auxiliar Grafico.....	75
Clase Capitulo .....	68
Clase Contenedor PDF .....	72
Clase EntidadPDF .....	71
Clase ManejoDB .....	69
Clase Presupuesto .....	67
Clase Seguridad .....	70, 71
Clase Servlet ComparaDetalle.....	73
Clase Servlet Descarga .....	73
Clase Servlet guardaParametros .....	74
Clase Servlet Guardar .....	74
Clase Servlet Lecturas .....	74
Clase Servlet XLS .....	74
Cocomo correspondencia PFNA .....	26
Cocomo Factor de Ajuste .....	27
Cocomo Factores de complejidad.....	27
Cocomo Factores Especificos de Coste.....	29
Cocomo Modelo .....	28
Coste corregido de Recursos humanos.....	37
Coste previsto de Tareas.....	32
Duracion corregida de tareas .....	37
duracion prevista de tareas .....	31
Página ayuda.....	76
Página Comparacion.....	77
Página Inicio .....	76
Página Integridad_Explicacion.....	76
Página leer .....	76
Página páginaSeguridad .....	77
Página ParametrosLectura .....	77
Página TodosPresupuestos .....	77
Página verPresupuesto.....	76
Presupuesto Recursos HW.....	30
Presupuesto Recursos SW .....	31
Prueba PR-001 .....	106
Prueba PR-002.....	106
Prueba PR-003 .....	107
Prueba PR-004.....	107
Prueba PR-005.....	107
Prueba PR-006.....	107
Prueba PR-007.....	108
Prueba PR-008.....	108
Prueba PR-009.....	109

Prueba PR-010.....	109
Prueba PR-011.....	109
Prueba PR-012.....	110
Prueba PR-013.....	110
Prueba PR-014.....	110
Prueba PR-015.....	111
Prueba PR-016.....	111
Prueba PR-017.....	111
Prueba PR-018.....	111
Prueba PR-019.....	112
Prueba PR-020.....	112
Prueba PR-021.....	112
Prueba PR-022.....	113
Prueba PR-023.....	113
Prueba PR-024.....	113
Requisito	
RU-R-05.....	51
Requisito RU-C-01.....	47
Requisito RU-C-02.....	47
Requisito RU-C-03.....	47
Requisito RU-C-04.....	48
Requisito RU-C-06.....	48
Requisito RU-C-07.....	48
Requisito RU-C-11.....	49
Requisito RU-C-12.....	49
Requisito RU-C-13.....	50
Requisito RU-R-02.....	50
Requisito RU-R-03.....	50
Requisito RU-R-04.....	51
Requisito RU-R-06.....	51
Requisito RU-R-07.....	52

