

# Sistema de Presupuestos Web

Memoria de Proyecto Final de Máster

**Máster Universitario en Desarrollo de sitios y aplicaciones web**

**Autora: María Sarmiento Calvo**

Consultor: Carlos Caballero González

Profesor: César Pablo Córcoles Briongos

06/01/2020



Esta obra está sujeta a una licencia de Reconocimiento-No Comercial-Sin Obra Derivada

[3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

*A Carlos, por despejarme el camino.*

*A Lorenzo y Vera, por esperarme infinitamente sin condiciones.*

*A Carlos Caballero, por la comprensión y el apoyo.*

## Abstract

El proyecto se basa en un sistema centralizado que permite la elaboración de presupuestos online por parte de los clientes de una empresa.

La aplicación facilitará a la empresa la administración de la información, es esto, el alta, baja, modificación de los datos de usuarios registrados y de los productos, así como las distintas tarifas asignadas a un cliente. También se almacenará un listado con los presupuestos elaborados por los clientes. Por su parte, el cliente accederá al sistema para iniciar el proceso de realización de presupuestos, pudiendo buscar el producto deseado e introduciendo las características del mismo requeridas. Al finalizar el presupuesto. Este presupuesto quedará almacenado en el sistema para consultas posteriores.

La aplicación será intuitiva, fácil de usar y visualmente atractiva.

**Palabras clave:** administración, gestión, empresa, presupuesto, navegador

## **Abstract (english version)**

The project is based on a centralized system that allows the preparation of online budgets by customers of a company.

The application will make it easier for the company to manage the information, that is, create, read, update and delete the data of registered users and products, as well as the different rates assigned to a customer. A list with the budgets prepared by customers will also be stored. For its part, the client will access the system to start the process of making budgets, being able to search for the desired product and entering the required characteristics. This budget will be stored in the system for later consultations.

The application will be intuitive, easy to use and visually attractive.

Keywords: administration, management, business, budget, browser

# Índice

1. Introducción.....	10
1.1. Contexto y justificación .....	11
1.2. Objetivos del trabajo .....	11
1.3. Enfoque y método seguido .....	11
1.4. Planificación del Trabajo .....	12
2. Definición del problema.....	13
3. Objetivos .....	15
3.1. Principal.....	15
3.2. Secundarios.....	15
4. Antecedentes .....	16
5. Metodología.....	17
6. Recursos .....	18
7. Arquitectura del sistema.....	19
7.1. Descripción general .....	19
7.2. Descripción de la información.....	19
7.2.1. Desarrollo de los subsistemas de la aplicación .....	20
7.2.2. Supuestos semánticos.....	21
7.2.3. Análisis de los tipos de entidad.....	22
7.2.4. Análisis de los tipos de interrelación.....	32
7.2.5. Diagrama Entidad-Interrelación (E-R).....	37
7.3. Especificación y modelación de la función.....	38
7.3.1. Diagramas de Flujo de Datos.....	38
7.4. Especificación de requisitos de la interfaz .....	54
7.4.1. Ergonomía de la interfaz.....	55
7.4.2. Ubicación de la aplicación en la pantalla .....	55
7.4.3. Información a mostrar, formato y situación de la misma en la pantalla .....	55
7.4.4. Validación de los datos de entrada .....	56
8. Especificación del sistema .....	57
8.1. Diseño de datos .....	57
8.1.1. Modelo relacional .....	57
8.1.2. Normalización del modelo.....	59
8.1.3. Definición sintáctica de las tablas .....	61
8.2. Diseño arquitectónico.....	65
8.3. Diseño de la interfaz .....	67
8.3.1. Descripción de la interfaz.....	67
8.3.2. Prototipo.....	68

8.3.3. Diseño final .....	74
9. Pruebas .....	79
9.1. Casos de prueba .....	79
10. Futuras mejoras .....	81
11. Conclusiones.....	82
Anexo 1. Librerías externas utilizadas .....	83
Anexo 2. Bibliografía .....	84

# Figuras y tablas

## Índice de figuras

Figura 1: Tipo de entidad <i>Usuario</i> .....	24
Figura 2: Tipo de entidad <i>Tarifa</i> .....	24
Figura 3: Tipo de entidad <i>Artículo</i> .....	24
Figura 4: Tipo de entidad <i>Presupuesto</i> .....	30
Figura 4: Tipo de entidad <i>Detalle</i> .....	32
Figura 5: Tipo de interrelación <i>Usuario-Tarifa</i> .....	33
Figura 6: Tipo de interrelación <i>Usuario-Presupuesto</i> .....	34
Figura 7: Tipo de interrelación <i>Artículo-Detalle</i> .....	35
Figura 7: Tipo de interrelación <i>Usuario-Detalle</i> .....	36
Figura 8: Diagrama E-R de la base de datos .....	37
Figura 9: Nivel de Abstracción 0: <i>Diagrama de contexto</i> .....	39
Figura 10: Nivel de abstracción 1: <i>Diagrama de subsistemas</i> .....	40
Figura 11: Nivel de abstracción 2: Refinamiento <i>Procesamiento Datos Usuario</i> .....	42
Figura 12: Nivel de abstracción 2: Refinamiento <i>Procesamiento Datos Artículo</i> .....	44
Figura 13: Nivel de abstracción 2: Refinamiento <i>Procesamiento Datos Tarifa</i> .....	45
Figura 14: Nivel de abstracción 2: Refinamiento <i>Consultas Presupuestos</i> .....	46
Figura 15: Nivel de abstracción 2: Refinamiento <i>Procesamiento Datos Presupuesto</i> .....	47
Figura 16: Nivel de abstracción 3: Refinamiento <i>Crear Usuario</i> .....	47
Figura 17: Nivel de abstracción 3: Refinamiento <i>Eliminar Usuario</i> .....	48
Figura 18: Nivel de abstracción 3: Refinamiento <i>Modificar Usuario</i> .....	49
Figura 19: Nivel de abstracción 3: Refinamiento <i>Crear Artículo</i> .....	49
Figura 20: Nivel de abstracción 3: Refinamiento <i>Eliminar Artículo</i> .....	50
Figura 21: Nivel de abstracción 3: Refinamiento <i>Modificar Artículo</i> .....	51
Figura 22: Nivel de abstracción 3: Refinamiento <i>Crear Tarifa</i> .....	51
Figura 23: Nivel de abstracción 3: Refinamiento <i>Eliminar Tarifa</i> .....	52
Figura 24: Nivel de abstracción 3: Refinamiento <i>Modificar Tarifa</i> .....	53
Figura 25: Nivel de abstracción 3: Refinamiento <i>Crear Presupuesto</i> .....	53
Figura 26: Nivel de abstracción 3: Refinamiento <i>Eliminar Artículo Presupuesto</i> .....	54
Figura 27: Nivel de abstracción 3: Refinamiento <i>Modificar Presupuesto</i> .....	54
Figura 28: Diagrama de Descomposición general .....	65
Figura 29: Diagrama de Descomposición de ADMINISTRAR USUARIOS .....	66
Figura 30: Esquema de interfaz .....	67
Figura 31: Prototipo de la página de acceso (Login) .....	68
Figura 32: Prototipo de la página inicial (parte Administración) .....	69
Figura 33: Prototipo de la página Listado (parte Administración) .....	70
Figura 34: Prototipo de la página Añadir (parte Administración) .....	70
Figura 35: Prototipo de la página Editar (parte Administración) .....	71
Figura 36: Prototipo de la página Consultas de Presupuestos (parte Administración) .....	72
Figura 37: Prototipo de la página Detalle de Presupuestos (parte Administración) .....	72
Figura 38: Prototipo de la página inicial (parte Creación de Presupuestos) .....	73

## Sistema de Presupuestos Web

Figura 39: Prototipo de la página Crear Presupuesto (parte Creación de Presupuestos) .....	74
Figura 40: Página de acceso (Login) .....	74
Figura 41: Página inicial (parte Administración) .....	75
Figura 42: Página Listado (parte Administración) .....	75
Figura 43: Página Añadir Tarifa (parte Administración) .....	76
Figura 43: Página Editar Tarifa (parte Administración) .....	76
Figura 44: Página Consultas de Presupuestos (parte Administración) .....	77
Figura 45: Prototipo de la página Detalle de Presupuestos (parte Administración) .....	77
Figura 46: Página Detalle de Presupuestos (parte Creación de Presupuestos) .....	78
Figura 47: Página Crear Presupuesto (parte Creación de Presupuestos) .....	78

# 1. Introducción

Hoy día, la mayor parte de las relaciones comerciales se hace a través de Internet. El acceso generalizado a Internet hace posible que una persona pueda acceder a servicios de cualquier empresa desde su domicilio o desde cualquier lugar donde se encuentre, a través de sus dispositivos móviles y con un tiempo de respuesta inmediato.

Un problema común que se da en pequeñas y medianas empresas (PYMES) es que sus clientes deben contactar con la empresa para pedir un presupuesto de sus productos, con el coste de tiempo que eso supone desde la petición del presupuesto hasta el envío del mismo al cliente. El presupuesto es el cómputo anticipado del coste de un servicio ofrecido por la empresa, y es un documento muy solicitado en determinadas actividades comerciales, como es el caso de la actividad de la empresa para la que se desarrolla este proyecto.

Los clientes de estas empresas son, por un lado, los que acuden directamente a las mismas; y, por el otro, un número considerable de otros comercios que ofrecen a sus clientes el servicio de esta empresa como si lo ofrecieran ellas mismas. Dichas tiendas se ven en la necesidad de elaborar con bastante frecuencia, para sus respectivos clientes, presupuestos de este servicio que ofrecen a través de otra empresa.

El presupuesto es un documento que, dependiendo de la actividad presupuestada, puede contener gran cantidad de datos que pueden cambiar con el tiempo con relativa frecuencia, y cálculos complejos. Elaborarlos requiere tiempo y un fluido flujo de información entre empresa y clientes, y su distribución en papel es un inconveniente. Por ello, Internet se nos presenta como la tecnología a través de la cual podemos afrontar eficientemente este problema, y así poder actualizar los datos.

Además, otro inconveniente con el que se encuentran muchas PYMES es que no disponen de un software específico a sus necesidades y a su actividad comercial, por lo que se ven obligadas a usar un software genérico que no cubre sus necesidades en su totalidad.

En este proyecto se afronta la realización de un sistema centralizado y gestionado completamente a través de Internet para la elaboración de presupuestos para el tipo de empresa anterior, para otras empresas que venden productos de la primera, e incluso para otros clientes que soliciten estar registrados en el sistema.

Aunque la aplicación está pensada para todo tipo de empresas que necesiten realizar presupuestos vía Internet, se ha usado como referencia concreta la elaboración de presupuestos de diversos productos de una empresa de cuadros.

## 1.1. Contexto y justificación

El sistema de elaboración de presupuestos vía web pretende servir como una herramienta de ayuda en la realización de presupuestos y gestión de los artículos y usuarios para un cierto tipo de empresas que requieren una continua comunicación con otras empresas o con sus clientes.

En la mayoría de estas empresas, la realización de los presupuestos para los clientes está informatizada, pero sólo es accesible al personal de la misma.

Los clientes que requieran un presupuesto hacen su petición por teléfono o email a la empresa. Éstas toman nota de los artículos solicitados, sus medidas y todos los datos necesarios para, posteriormente, hacer el presupuesto y entregárselo al cliente. Para dar este servicio lo más rápidamente posible, se debe restar tiempo de la actividad comercial frente al público para dedicárselo a tal propósito. El problema es que las peticiones presupuestarias son muy frecuentes y resta un tiempo considerable que se puede emplear en otras tareas.

Por tanto, el propósito de este proyecto se centra en facilitar, reducir y agilizar, en todo lo posible, los procesos de petición de un presupuesto y de su posterior realización.

## 1.2. Objetivos del trabajo

El objetivo del proyecto es el desarrollo e implantación de un Sistema software que, implantado en un servidor, permita a los clientes de un negocio a elaborar sus propios presupuestos.

Para ello, el sistema permitirá el acceso a dos tipos de usuario: administrador y cliente/usuario mediante un login y contraseña. El administrador se encargará del mantenimiento del sistema, es decir, de dar de alta/baja a los usuarios, artículos y demás datos necesarios. El usuario/cliente podrá realizar presupuestos eligiendo los artículos que desea mediante una búsqueda de los mismos. Por su parte, el sistema deberá almacenar toda la información referente a los datos de acceso al sistema de los usuarios y a los datos de los productos y presupuestos realizados.

## 1.3. Enfoque y método seguido

El enfoque del proyecto es profesionalizador y se seguirá una metodología SCRUM para el desarrollo del mismo. Esto consistirá en desarrollar incrementalmente los requisitos del proyecto en periodos de tiempo fijos, dando prioridad a los que tengan más valor para el equipo consultor-autor. El equipo se sincronizará periódicamente y se realizarán las adaptaciones necesarias. Tras cada iteración se muestra al consultor el resultado real obtenido para que se tomen las decisiones necesarias en relación a lo observado. Las fases del proyecto no serán secuenciales, sino que se solaparán a lo largo del desarrollo del proyecto.

## 1.4. Planificación del Trabajo

La primera etapa en todo desarrollo de un proyecto es la definición de requisitos del mismo. A ella se llegará con las distintas conversaciones o acuerdos con el consultor.

En una primera conversación se realizará una descripción de la información que se requiera y de los objetivos buscados. Después de este primer contacto, y lo largo de todo el desarrollo del proyecto, se mantendrán sucesivas conversaciones para clarificar ideas y para comentar y evaluar el trabajo realizado hasta entonces con el consultor del proyecto.

Una vez definidos, concretados y clarificados todos los requisitos, se comenzará con el desarrollo del proyecto, propiamente dicho.

Las fases que hay que completar en el desarrollo del proyecto son las siguientes:

- **Análisis del problema:** en esta fase se identificarán las funciones que deberá realizar el software, el rendimiento esperado y las interfaces necesarias.
- **Diseño:** fase en la que se determinan las características formales del software (estructura de los datos, arquitectura del sistema, interfaz y procedimientos), lo que permitirá la realización del mismo en la siguiente etapa.
- **Codificación:** se implementará el sistema software con las herramientas y lenguajes de programación escogidos.
- **Pruebas:** en esta fase se verificará y validará el software generado. De esta forma, se procederá a la búsqueda de posibles errores y a su posterior solución.
- **Documentación:** esta fase se realizará a lo largo de todo el desarrollo del proyecto y cuyo objetivo final será la obtención de la memoria del proyecto.

Los dos primeros puntos se corresponden a la entrega de la PEC2. Los dos últimos puntos engloban todo el grueso del proyecto y corresponde a la entrega de la PEC3.

## 2. Definición del problema

A continuación se describirá detalladamente todos los aspectos del problema a solucionar, de manera que se defina desde un punto de vista más técnico.

### Funcionamiento

El proyecto a desarrollar debe tener un funcionamiento muy intuitivo, de manera que dé sentido al desarrollo de la aplicación.

Por ello, la solución software estará formada por dos partes claramente diferenciadas.

Por un lado, se distingue la sección de administración del sistema. En ella, el administrador podrá gestionar todos los datos que se manejan en el sistema. Es decir, introducir, modificar, eliminar y visualizar todos los datos relativos a los usuarios y a los artículos; así como visualizar y realizar consultas sobre los presupuestos elaborados por los clientes.

Por otro lado, se encuentra la zona de los clientes. Esta sección dará la posibilidad a cada cliente a elaborar su propio presupuesto a medida. El cliente irá introduciendo uno a uno los datos de los artículos que desea añadir al presupuesto. A la vez se mostrará por pantalla todos los detalles de los artículos que se van presupuestando.

### Entorno

#### **Entorno de usuario**

El tipo de usuario al que va dirigido esta herramienta es un usuario con una profesión relacionada con el comercio, con conocimientos básicos o medios sobre manejo del ordenador y familiarizado con programas informáticos. Es por ello que, probablemente, el uso de esta herramienta se haga en el lugar del trabajo bajo las condiciones cotidianas de cada uno. Por tanto, el proyecto está orientado a ser visualmente atractivo para no ser rechazado, y ser lo más útil posible. Para cumplir tal cometido, se utilizarán las hojas de estilo CSS para darle al sistema un entorno gráfico agradable.

#### **Entorno hardware**

Como se ha dicho anteriormente, la herramienta será comúnmente utilizada en el puesto de trabajo (aunque también se puede usar con un dispositivo móvil con acceso a Internet), por lo que debe poder ejecutarse bajo cualquier tipo de ordenador con prestaciones hardware normales y con un rendimiento que resulte adecuado para que pueda llevarse a cabo la ejecución de ésta en condiciones correctas. Por ello, se

ha pensado en una herramienta que no consuma demasiados recursos, cuyo coste computacional sea bajo.

## **Entorno software**

Debido a todo lo anteriormente citado, el entorno software no será muy exigente para que la interacción con el usuario tenga el resultado de éxito que se pretende. Ya que la aplicación se desarrollará bajo los lenguajes de programación HTML5 (HTML5 Introduction - W3Schools, 2020), CSS3 (CSS Tutorial - W3Schools, 2020), jQuery (jQuery, 2020), CodeIgniter (CodeIgniter, 2020) y MySQL (MySQL, 2020), para su utilización tan sólo será necesario tener instalado un navegador web (Internet Explorer, Firefox, Chrome, ...).

## **Estandarización**

La aplicación cumplirá los estándares para diseño Web recomendados por la organización Word Wide Web Consortium (W3C) (World Wide Web Consortium (W3C), 2020).

Por ello, cada uno de los lenguajes de programación utilizados en el desarrollo del proyecto seguirán su propio estándar o reglas a la hora de la creación de código.

## **Calidad y fiabilidad**

Se hará todo lo posible para que la herramienta sea lo más cómoda de manejar y que no tenga errores de funcionamiento para que el usuario no desconfíe de ella y deje de utilizarla.

Para ello, el proyecto deberá cumplir las siguientes características de calidad y fiabilidad:

- **Correctitud**: deberá estar libre de errores tras realizar las pruebas oportunas que lo verifiquen.
- **Facilidad de uso**: la aplicación se deberá desarrollar para que el tipo de usuario al que está destinada no tenga ningún problema en su uso y manejo.
- **Eficiencia**: se procurará que las tareas que realizará la herramienta, se hagan de forma óptima y en el menor tiempo posible.
- **Confiabilidad**: se deberá desarrollar un sistema de control de errores para que no se produzca ninguna situación inesperada.
- **Integridad**: se deberá garantizar la seguridad de los datos introducidos por los usuarios.
- **Adaptabilidad**: el software se podrá ejecutar en un ordenador que no posea características especiales.

## 3. Objetivos

### 3.1. Principal

Del objetivo general del proyecto, que es desarrollar e implementar un sistema software que permita elaborar presupuestos del servicio de enmarcado de cuadros a través de Internet, se pueden concretar los siguientes objetivos:

### 3.2. Secundarios

- **Objetivo 1. Control de acceso:** el sistema será capaz de administrar dos tipos diferentes de usuarios (administrador y cliente/usuario) mediante login y contraseña. Habrá un único administrador encargado del mantenimiento del sistema y dar de alta/baja a los usuarios y artículos. El administrador registrará a los clientes y les proporcionará sus respectivos login y contraseña para el acceso al sistema.
- **Objetivo 2. Gestión de usuarios y artículos:** el sistema deberá almacenar en una base de datos toda la información referente a los datos de acceso al sistema de todos los usuarios, y los datos de todos los artículos ofrecidos en este servicio por la empresa y que se utilizarán para elaborar presupuestos.
- **Objetivo 3. Acceso web y facilidad de uso:** el sistema se alojará en un servidor web y podrá ser accedido vía Internet desde cualquier navegador. En todo el desarrollo del proyecto se tendrá siempre en cuenta que, dada la diversidad de usuarios y de conocimientos de dichos usuarios sobre el uso de aplicaciones informáticas, la aplicación deberá poseer una gran facilidad de uso.

## 4. Antecedentes

En la actualidad se puede encontrar diversas aplicaciones que abordan el tema de la realización de presupuestos. Los más conocidos y comúnmente utilizados se centran en áreas profesionales como la construcción y la contabilidad comercial. Algunos ejemplos de aplicaciones son:

- **Presto 2019** (RIB Spain, 2020): es un software de presupuestos para edificación, obras civiles e instalaciones (referencia).
- **Eneboo** (anteriormente AbanQ) (Eneboo Open Source ERP, 2020): se trata de un software open source orientado a la administración, gestión comercial, finanzas, etc. Su aplicación abarca desde la gestión financiera y comercial en empresas hasta la adaptación a procesos complejos de producción (referencia).
- **Facturaplus** (Sage FacturaPlus, 2020): es uno de los software de gestión comercial más utilizados.

Todas estas aplicaciones son muy completas y útiles, pero a la vez son muy amplias y extensas, y no se centran exclusivamente en la función presupuestaria. Además son aplicaciones que se ejecutan en modo local.

Por estas razones es por lo que se pretende desarrollar este proyecto, para dar una solución específica y centrada en la elaboración de presupuestos y la gestión de los artículos de una determinada empresa.

## 5. Metodología

Para que el proyecto presente un desarrollo adecuado y una correcta elaboración, se necesita un programa de tareas a realizar. El seguimiento de este esquema no es estricto, ya que en cualquier momento se puede volver a una fase anterior para realizar mejoras en ella.

La primera etapa en todo desarrollo de un proyecto es la definición de requisitos del mismo. A ella se plantearán todos los requisitos del sistema que se pretende desarrollar. Se realizará una descripción detallada de la información que se requiera y de los objetivos buscados.

Una vez definidos, concretados y clarificados todos los requisitos, se comenzará con el desarrollo del proyecto, propiamente dicho.

Las fases que hay que completar en el desarrollo del proyecto son las siguientes:

- *Análisis de los requisitos que se han obtenido en las primeras etapas del proyecto.*
- *Análisis:* Identificación de las funciones que deberá realizar el software, y de los distintos subsistemas que lo compondrán. Estudio del rendimiento esperado y de las interfaces necesarias.
- *Diseño:* fase en la que se determinan las características formales del software (estructura de los datos, arquitectura del sistema, interfaz y procedimientos), lo que permitirá la realización del mismo en la siguiente etapa.
- *Codificación:* se implementará el sistema software con las herramientas y lenguajes de programación escogidos.
- *Pruebas:* en esta fase se verificará y validará el software generado. De esta forma, se procederá a la búsqueda de posibles errores y a su posterior solución.
- *Documentación:* esta fase se realizará a lo largo de todo el desarrollo del proyecto y cuyo objetivo final será la obtención de la memoria del proyecto.

## 6. Recursos

A continuación se describen los diferentes recursos tecnológicos utilizados.

### Recursos software

El entorno de desarrollo utilizado estará constituido por las siguientes herramientas:

- Sistema operativo macOS Mojave, en su versión 10.14.6: soporte para la programación de la herramienta, la ejecución y consiguiente uso de los programas que se citan a continuación.
- Servidor web Apache( incluido en el entorno MAMP): que implementa el protocolo HTTP.
- Lenguajes de programación front-end: HTML5, CSS3, Bootstrap y jQuery.
- CodeIgniter como entorno de desarrollo basado en PHP.
- Gestor de bases de datos MySQL.
- PhpMyAdmin: administración de MySQL a través de páginas web.
- Navegador Google Chrome.
- Microsoft Word: procesador de texto para generar toda la documentación escrita del proyecto.
- Visual Studio Code: editor de texto usado para escribir el código fuente.
- Dia: editor de diagramas.
- Illustrator: creación de imágenes, diagramas y logotipo.
- Balsamiq Mockups: creación prototipos.

### Recursos hardware

El equipo hardware utilizado para llevar a cabo el desarrollo de la aplicación es un iMac con las siguientes características:

- Procesador Intel Core i5 3,4 GHz.
- Memoria RAM: 16 GB 1600 MHz DDR3.
- Tarjeta gráfica: NVIDIA GeForce GTX 780M 4 GB.
- Disco duro: 1 TB Fusion Drive.

## 7. Arquitectura del sistema

Se puede definir la arquitectura del sistema mediante los siguientes ítems:

- **Descripción general del problema.** En este apartado se realizará una definición del problema, especificando cuáles son los objetivos generales del mismo, considerándose para ello toda la información vista anteriormente.
- **Descripción de la información.** En este apartado se realizará una descripción detallada del problema a resolver, analizando cuáles son los datos o información a tratar y los flujos de datos presentes en el sistema.
- **Descripción funcional.** En este apartado se realizará una descripción de las diferentes funciones a implementar para conseguir manipular los datos en función de los requisitos establecidos.

### 7.1. Descripción general

El objetivo principal de este proyecto es la creación de una aplicación web que, facilitando su acceso a través de cualquier navegador, posibilite a los clientes de un tipo de empresas a elaborar su propio presupuesto de manera fácil, rápida y cómoda.

Para ello, es necesaria la presencia de una base de datos que contenga la información personal de los usuarios que realizarán los presupuestos y los datos relacionados con los artículos de la empresa los cuales van a ser presupuestados. De esta forma, la información aquí almacenada dará lugar a la aplicación web del presente proyecto. En ella existirán dos interfaces que proporcionarán una serie de utilidades según el tipo de usuario: cliente o administrador.

### 7.2. Descripción de la información

En este apartado se describe el dominio de la información que será manejado por el sistema, describiendo cada uno de ítems de datos individuales que son aceptados por el sistema (información de entrada), los cambios a los que son sometidos por el procesamiento y los que el sistema devuelve como salida.

Dicha técnica representa gráficamente y tabularmente los objetos (entidades) y las relaciones entre ellos (interrelaciones) que forman parte de la estructura del sistema.

### 7.2.1. Desarrollo de los subsistemas de la aplicación

A continuación se explica en más detalle cada uno de los dos subsistemas que componen la aplicación.

En la página principal aparecerá el control de acceso para los usuarios. Éstos deberán introducir correctamente su nombre y contraseña para poder iniciar sesión en la aplicación. Esto conlleva a que, según el tipo de usuario que se valide, tendrá diferentes privilegios de acceso a la información y, por lo tanto, a su manejo.

**Subsistema gestor de administración:** En él, el administrador deberá introducir toda la información requerida de los usuarios y de los artículos disponibles de la empresa en la base de datos. De esta forma, le será más fácil llevar a cabo las tareas propias de esta actividad: gestión de los usuarios, gestión de los artículos, gestión de las tarifas y consulta de los presupuestos realizados por los clientes. En la gestión de los datos, el administrador podrá introducirlos, modificarlos, eliminarlos y consultarlos de la base de datos en el momento que desee. En la consulta de los presupuestos el administrador, en base a unos criterios de búsqueda por fecha, podrá visualizar todos aquellos presupuestos ya realizados por los clientes y que están almacenados en la base de datos. El administrador proporcionará a cada cliente su nombre de usuario y contraseña.

**Subsistema creador de presupuestos:** en él, los usuarios previamente registrados por el administrador en el sistema podrán elaborar paso a paso un presupuesto adaptado a sus necesidades. En un primer lugar, deberá introducir la referencia del artículo a presupuestar (todas las referencias de los artículos las tienen los clientes, ya que son proporcionadas por la empresa). El sistema buscará los datos del artículo pertenecientes a dicha referencia, tales como la descripción y el precio de venta al público o PVP, y los mostrará en pantalla. Una vez aquí, el usuario introducirá las unidades que desea de dicho artículo. El sistema calculará el importe del artículo, el subtotal y el importe total del presupuesto, aplicándole el descuento correspondiente a dicho cliente, cuya valor decidirá el administrador. El usuario también tendrá la opción de modificar los datos introducidos en los artículos presupuestados y borrar aquellos artículos que ya no precise. Una vez terminado el presupuesto, el usuario saldrá del sistema.

## 7.2.2. Supuestos semánticos

Como se dijo anteriormente, es necesario realizar una base de datos que mantenga toda la información necesaria para el correcto funcionamiento de la aplicación. En dicha base de datos se almacenará toda la información relativa a los usuarios, las tarifas, los artículos y los presupuestos.

Para el diseño de la base de datos se consideran los siguientes supuestos semánticos:

SUPUESTO 1: Todo usuario es identificado a través de su login o nombre de usuario. Además, de cada usuario se almacenará su clave de acceso a la aplicación, su nombre y su tarifa correspondiente.

SUPUESTO 2: Cada usuario tiene asociada una única tarifa, la cual le proporciona un descuento sobre el PVP de los artículos. Más de un usuario puede tener asignada la misma tarifa.

SUPUESTO 3: Las tarifas se identifican por un número. A éste le corresponde un porcentaje, que será el descuento a aplicar.

SUPUESTO 4: Un usuario puede realizar cuantos presupuestos quiera, pero un determinado presupuesto sólo puede pertenecer a un único usuario.

SUPUESTO 5: Un presupuesto posee un único número de presupuesto que lo identifica. Además de él, interesará mantener cierta información de los artículos presupuestados, tal como la referencia, unidades, PVP, importe, subtotal, total y el login del usuario que ha realizado el presupuesto.

SUPUESTO 6: Un presupuesto está compuesto de uno o varios artículos, y un artículo puede formar parte de muchos presupuestos o de ninguno.

SUPUESTO 7: De un artículo interesa saber su referencia, por la cual se le identifica unívocamente, el tipo de artículo que es, su descripción, su PVP y su desperdicio.

SUPUESTO 8. El administrador podrá ver un resumen de todos los presupuestos realizados y guardados en el sistema, pero interesa ver un detalle de cada uno de ellos con información como el número del presupuesto, la referencia, descripción, unidades, pvp e importe de los artículos presupuestados, así como el login del usuario que ha realizado el presupuesto.

### 7.2.3. Análisis de los tipos de entidad

Como parte del análisis de la información a tratar y como paso previo al diseño de la base de datos que formará parte del programa, se considera la existencia de los siguientes tipos de entidad extraídos de los supuestos semánticos anteriormente expuestos. Para la descripción de cada uno de ellos se explicarán los puntos:

- *Descripción*: indicará el objeto del mundo real al cual representa el tipo de entidad y la definición que éste tiene en el mundo real.
- *Atributos heredados*: indicarán aquellos atributos que no son propios del tipo de entidad, sino que son heredados de otra/-s entidad/-es por el tipo de relación que las une.
- *Lista de atributos*: se mostrará el número de atributos que posee el tipo de entidad, así como cada atributos perteneciente a él. De ellos se especificará su dominio, multiplicidad, restricciones y comentarios.
- *Identificador principal*: indicará el/los atributo/-s con el que se reconocerá al tipo de entidad unívocamente.
- *Diagrama*: representación gráfica del tipo de entidad mediante el diagrama Entidad-Interrelación.
- *Ejemplo*: se mostrará un ejemplo de un posible caso real de ocurrencia del tipo de entidad.

#### 7.2.3.1. Tipo de entidad *Usuario*

**Descripción**: representa al objeto del mundo real “usuario”, y que se puede definir como “persona que utiliza habitualmente el sistema informático”. Para este tipo de entidad se van a considerar los atributos *login*, *clave* y *nombre* (SUPUESTO 1).

**Atributos heredados**: ninguno.

**Lista de atributos**: la entidad tiene tres atributos propios, los cuales son

Atributo *login*: representa el nombre de usuario necesario para el control de acceso al sistema e identifica de forma unívoca a un usuario. Este atributo será considerado el identificador principal del tipo de entidad.

Dominio: tomará valores alfanuméricos.

Multiplicidad: no tiene, ya que sólo existirá un “login” por cada uno de los usuarios que se introduzcan en la base de datos.

Restricciones: la longitud máxima de la cadena alfanumérica es de 8 caracteres.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

Atributo *clave*: representa la contraseña que el usuario debe introducir para el control de acceso al sistema, junto con login.

Dominio: tomará valores alfanuméricos.

Multiplicidad: no tiene, ya que sólo existirá una “clave” por cada uno de los usuarios que se introduzcan en la base de datos.

Restricciones: la longitud máxima de la cadena alfanumérica es de 10 caracteres.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

Atributo *nombre*: representa el nombre del usuario.

Dominio: tomará cualquier combinación de valores alfanuméricos válidos.

Multiplicidad: no tiene, ya que sólo existirá una “clave” por cada uno de los usuarios que se introduzcan en la base de datos.

Restricciones: la longitud máxima de la cadena alfanumérica es de 10 caracteres.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

Atributo *nivel*: representa el tipo de usuario.

Dominio: tomará valores numéricos.

Multiplicidad: no tiene, ya que sólo existirá un “nivel” por cada uno de los usuarios que se introduzcan en la base de datos.

Restricciones: se introducirá un número de 1 dígito como máximo.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

**Identificador principal:** *login*

**Diagrama:**

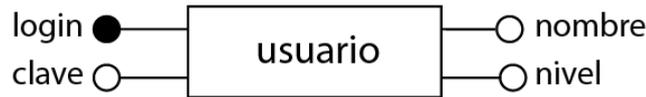


Figura 1: Tipo de entidad *Usuario*

**Ejemplo:** una ocurrencia del tipo de entidad *Usuario* podría ser:

**login:** uMariaMolina

**clave:** mariaMolina56

**nombre:** Interiores Maria Molina

**nivel:** 2

### 7.2.3.2. Tipo de entidad *Tarifa*

**Descripción:** representa al objeto del mundo real “tarifa”, y que se puede definir como “escala que modifica el precio de venta al público de los artículos”. Para este tipo de entidad se van a considerar los atributos *idtarifa* y *valor* (SUPUESTOS 2 y 3).

**Atributos heredados:** ninguno.

**Lista de atributos:** la entidad tiene dos atributos propios, los cuales son

Atributo *idtarifa*: representa el nombre de la tarifa que se asigna a un usuario y la identifica de manera unívoca. Este atributo será considerado el identificador principal del tipo de entidad.

Dominio: tomará valores numéricos.

Multiplicidad: no tiene, ya que sólo existirá una “tarifa” por cada uno de los usuarios que se introduzcan en la base de datos.

Restricciones: se introducirá un número de 2 dígitos como máximo.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

Atributo *valor*: representa el porcentaje que se descontará del PVP de cada artículo.

Dominio: tomará valores numéricos.

Multiplicidad: no tiene, ya que sólo existirá una “clave” por cada uno de los usuarios que se introduzcan en la base de datos.

Restricciones: el valor que tomará este atributo será de tipo decimal y menor que la unidad.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

**Identificador principal:** *idtarifa*

**Diagrama:**

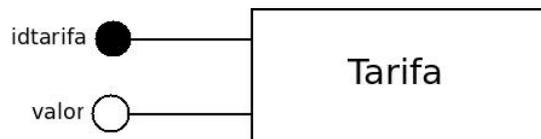


Figura 2: Tipo de entidad *Tarifa*

**Ejemplo:** una ocurrencia del tipo de entidad *Tarifa* podría ser:

***idtarifa:*** 2

***valor:*** 0.30

### 7.2.3.3. Tipo de entidad *Articulo*

**Descripción:** representa al objeto del mundo real “artículo”, y que se puede definir como “objeto perteneciente a un negocio con el cual se comercia y cuyo objetivo es ser utilizado en los presupuestos”. Para este tipo de entidad se van a considerar los atributos *referencia*, *tipo*, *descripción*, *pvp* y *desperdicio* (SUPUESTOS 6 y 7).

**Atributos heredados:** ninguno.

**Lista de atributos:** la entidad tiene cinco atributos propios, los cuales son

Atributo *referencia*: representa el código del artículo y lo identifica de forma unívoca. Este atributo será considerado el identificador principal del tipo de entidad.

Dominio: tomará cualquier combinación de valores alfanuméricos válidos.

Multiplicidad: no tiene, ya que sólo existirá una “referencia” por cada uno de los artículos que se introduzcan en la base de datos.

Restricciones: la longitud máxima de la cadena alfanumérica es de 10 caracteres.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

Atributo *tipo*: representa el tipo de artículo que tiene el comercio. Según la forma por la que se presupuestan, existen tres tipos de artículos vendidos: por unidades, por metros lineales y por metros cuadrados.

Dominio: tomará valores alfanuméricos.

Multiplicidad: no tiene, ya que sólo existirá un “tipo” por cada uno de los artículos que se introduzcan en la base de datos.

Restricciones: los únicos valores que puede tener un tipo son “unidad”, “m” y “m2”.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

Atributo *descripcion*: representa la descripción del artículo.

Dominio: tomará cualquier combinación de valores alfanuméricos válidos.

Multiplicidad: no tiene, ya que sólo existirá una “descripción” por cada uno de los artículos que se introduzcan en la base de datos.

Restricciones: la longitud máxima de la cadena alfanumérica es de 50 caracteres.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

Atributo *pvp*: representa el precio de venta al público que tendrá el artículo.

Dominio: tomará valores numéricos.

Multiplicidad: no tiene, ya que sólo existirá un “pvp” por cada uno de los artículos que se introduzcan en la base de datos.

Restricciones: el valor que tomará será un número real.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

Atributo *desperdicio*: representa el valor que se desperdicia a la hora de medir ciertos tipos de artículos.

Dominio: tomará valores numéricos.

Multiplicidad: no tiene, ya que sólo existirá un “desperdicio” por cada uno de los artículos que se introduzcan en la base de datos.

Restricciones: el valor que tomará será un número real.

Comentarios: Aunque este atributo es obligatorio, podrá tener valores nulos, ya que los artículos que se miden por unidades o por metros lineales no tienen desperdicio.

**Identificador principal:** *referencia*

**Diagrama:**

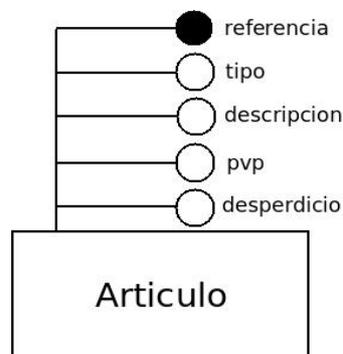


Figura 3: Tipo de entidad *Artículo*

**Ejemplo:** una ocurrencia del tipo de entidad *Articulo* podría ser:

**referencia:** M789

**tipo:** m

**descripcion:** Moldura Fina Negra

**pvp:** 10.5

**desperdicio:** 0.15

#### 7.2.3.4. Tipo de entidad *Presupuesto*

**Descripción:** representa al objeto del mundo real “presupuesto”, y que se puede definir como “cómputo anticipado de gastos o ingresos, o de unos y otros en un negocio”. Para este tipo de entidad se van a considerar los atributos *idpresupuesto*, *referencia*, *unidades*, *pvp*, *subtotal*, *fecha* y *login* (SUPUESTO 5y 6).

**Atributos heredados:** ninguno

**Lista de atributos:** la entidad tiene nueve atributos propios, los cuales son

Atributo *idpresupuesto*: representa el número de presupuesto. Este atributo será considerado el identificador principal del tipo de entidad.

Dominio: tomará valores numéricos.

Multiplicidad: no tiene, ya que sólo existirá un “*idpresupuesto*” por cada uno de los presupuestos que se introduzcan en la base de datos.

Restricciones: los valores que podrá tomar son números naturales y consecutivos.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

Atributo *subtotal*: representa el sumatorio de importes de todos los artículos del presupuesto.

Dominio: tomará valores numéricos.

Multiplicidad: no tiene, ya que sólo existirá un “*subtotal*” por cada uno de los artículos que se introduzcan en la base de datos.

Restricciones: el valor que tomará será un número real.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

Atributo *total*: representa la cuantía total del presupuesto.

Dominio: tomará valores numéricos.

Multiplicidad: no tiene, ya que sólo existirá un “total” por cada uno de los presupuestos que se introduzcan en la base de datos.

Restricciones: el valor que tomará será un número real.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

Atributo *fecha*: representa el día, el mes, el año y la hora a los que es realizado un presupuesto.

Dominio: tomará como valor cualquier fecha válida.

Multiplicidad: no tiene, ya que sólo existirá una “fecha” por cada uno de los presupuestos que se introduzcan en la base de datos.

Restricciones: el valor que tomará será la fecha actual del sistema.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

Atributo *tarifa*: representa el valor del descuento que tiene un usuario y es aplicable a los presupuestos que éste realice.

Dominio: tomará valores numéricos.

Multiplicidad: no tiene, ya que sólo existirá una “tarifa” por cada presupuesto que se introduzca en la base de datos.

Restricciones: el valor que tomará será un número real.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

**Identificador principal:** *idpresupuesto*

**Diagrama:**



Figura 4: Tipo de entidad *Presupuesto*

**Ejemplo:** una ocurrencia del tipo de entidad *Presupuesto* podría ser:

***idpresupuesto:*** 15

***subtotal:*** 348.50

***tarifa:*** 0.15

***total:*** 344.20

***fecha:*** 2019-10-10

### 7.2.3.5. Tipo de entidad *Detalle*

**Descripción:** representa al objeto del mundo real “detalle”, y que se puede definir como “información pormenorizada de un presupuesto realizado”. Para este tipo de entidad se van a considerar los atributos *id*, *unidades*, *pvp* e *importe* (8).

**Atributos heredados:** ninguno

**Lista de atributos:** la entidad tiene nueve atributos propios, los cuales son

Atributo *id*: representa el número del registro detalle. Este atributo será considerado el identificador principal del tipo de entidad.

Dominio: tomará valores numéricos.

Multiplicidad: no tiene, ya que sólo existirá un “id” por cada uno de los presupuestos que se introduzcan en la base de datos.

Restricciones: los valores que podrá tomar son números naturales y consecutivos.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

Atributo *unidades*: representa la cantidad de un mismo artículo que se ha presupuestado.

Dominio: tomará valores numéricos.

Multiplicidad: no tiene, ya que sólo existirá una “unidad” por cada uno de los artículos que se introduzcan en la base de datos.

Restricciones: los valores que podrá tomar son números naturales.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

Atributo *pvp*: representa el precio de venta al público que tendrá el artículo presupuestado.

Dominio: tomará valores numéricos.

Multiplicidad: no tiene, ya que sólo existirá un “pvp” por cada uno de los artículos que se introduzcan en la base de datos.

Restricciones: el valor que tomará será un número real.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

Atributo *importe*: representa el precio del artículo calculado según las unidades y el PVP del mismo.

Dominio: tomará valores numéricos.

Multiplicidad: no tiene, ya que sólo existirá un “importe” por cada uno de los artículos que se introduzcan en la base de datos.

Restricciones: el valor que tomará será un número real.

Comentarios: este atributo no podrá tener valores nulos y es obligatorio, es decir, siempre será requerido.

**Identificador principal:** *id*

**Diagrama:**

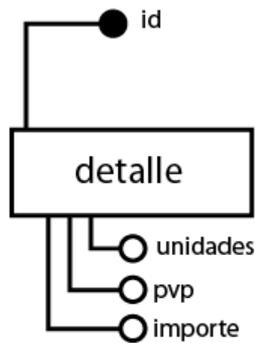


Figura 4: Tipo de entidad *Detalle*

**Ejemplo:** una ocurrencia del tipo de entidad *Detalle* podría ser:

***id:*** 15

***unidades:*** 2

***pvp:*** 9.00

***importe:*** 18.00

## 7.2.4. Análisis de los tipos de interrelación

Los tipos de entidad antes mencionados se encuentran relacionados de la manera que a continuación se expone. Para la descripción de cada uno de ellos se explicarán los puntos:

- **Descripción:** indicará los tipos de entidad que intervienen en la interrelación y a qué se debe dicha relación.
- **Características:** mostrará el nombre, tipo de correspondencia, cardinalidades y la relación con la que participa cada tipo de entidad en la interrelación.
- **Diagrama:** representación gráfica del tipo de interrelación mediante el diagrama Entidad-Interrelación.

### 7.2.4.1. Tipo de interrelación *Usuario-Tarifa*

**Descripción:** este tipo de interrelación relaciona los tipos de entidad *Usuario* y *Tarifa*, indicando la tarifa que tiene asociada un usuario.

**Características:**

Nombre del tipo de interrelación: Usuario-Tarifa (U-T)

Tipo de correspondencia: 1:N

Cardinalidades: Se considera que un usuario posee una única tarifa que modificará el precio de los artículos, por lo que el tipo de entidad *Usuario* participa en la interrelación con cardinalidad (1, 1). Por otro lado, una tarifa puede estar asociada a un usuario, a varios o a ninguno. Con esto, se puede decir que el tipo de entidad *Tarifa* participa con cardinalidad (0/1, n) (SUPUESTO 2).

Relación de los tipos de entidad: Usuario-Tarifa => posee; Tarifa-Usuario => es\_poseida\_por

**Diagrama:**

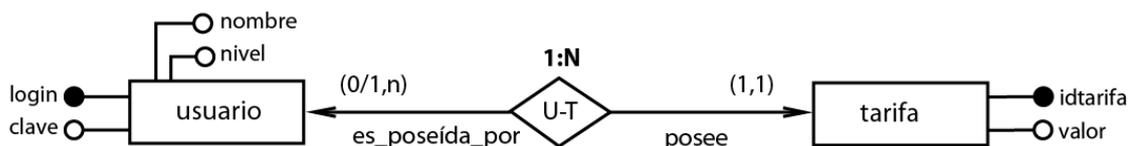


Figura 5: Tipo de interrelación *Usuario-Tarifa*

### 7.2.4.2. Tipo de interrelación *Usuario-Presupuesto*

**Descripción:** este tipo de interrelación relaciona los tipos de entidad *Usuario* y *Presupuesto*, y representa al usuario que realiza un presupuesto.

**Características:**

Nombre del tipo de interrelación: Usuario-Presupuesto (U-P)

Tipo de correspondencia: 1:N

Cardinalidades: Se considera que un usuario puede realizar varios presupuestos, por lo que el tipo de entidad *Usuario* participa en la interrelación con cardinalidad (0/1, n). Por otro lado, un determinado presupuesto sólo puede haber sido realizado por un único usuario, participando así el tipo de entidad *Presupuesto* con cardinalidad (1, 1) (SUPUESTO 4).

Relación de los tipos de entidad: Usuario-Presupuesto => realiza; Presupuesto-Usuario => es\_realizado\_por

**Diagrama:**

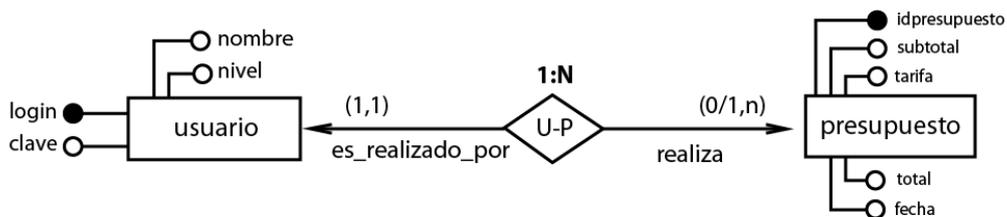


Figura 6: Tipo de interrelación *Usuario-Presupuesto*

### 7.2.4.3. Tipo de interrelación *Articulo-Detalle*

**Descripción:** este tipo de interrelación relaciona los tipos de entidad *Detalle* y *Articulo*, y representa los artículos que componen el detalle de un determinado presupuesto.

**Características:**

Nombre del tipo de interrelación: Detalle-Articulo (D-A)

Tipo de correspondencia: N:1

Cardinalidades: Se considera que el detalle de un presupuesto tiene que estar compuesto por un artículo, por lo que el tipo de entidad *Articulo* participa en la interrelación con cardinalidad (1, 1). Por otro lado, un artículo puede estar presente o no en un determinado detalle de presupuesto. Con esto, se puede decir que el tipo de entidad *Detalle* participa con cardinalidad (0/1, n) (SUPUESTO 8).

Relación de los tipos de entidad: Detalle-Articulo => esta\_formado\_por; Articulo-Detalle => forma\_parte\_de

**Diagrama:**

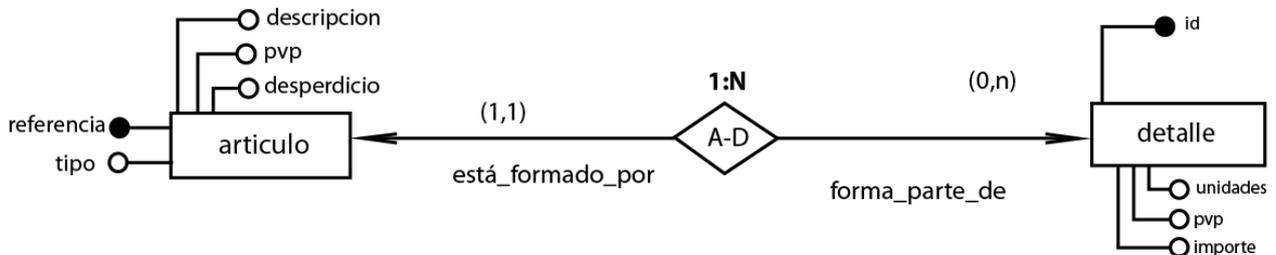


Figura 7: Tipo de interrelación *Articulo-Detalle*

#### 7.2.4.4. Tipo de interrelación *Usuario-Detalle*

**Descripción:** este tipo de interrelación relaciona los tipos de entidad *Detalle* y *Usuario*, y representa los usuarios que forman parte del detalle de un determinado presupuesto.

**Características:**

Nombre del tipo de interrelación: Usuario-Detalle (U-D)

Tipo de correspondencia: N:1

Cardinalidades: Se considera que el detalle de un presupuesto puede pertenecer a un usuario, por lo que el tipo de entidad *Detalle* participa en la interrelación con cardinalidad (1, 1). Por otro lado, un usuario puede estar presente o no en un determinado detalle de presupuesto. Con esto, se puede decir que el tipo de entidad *Usuario* participa con cardinalidad (0, n) (SUPUESTO 8).

Relación de los tipos de entidad: Detalle-Usuario => esta\_formado\_por; Usuario-Detalle => forma\_parte\_de

Diagrama:

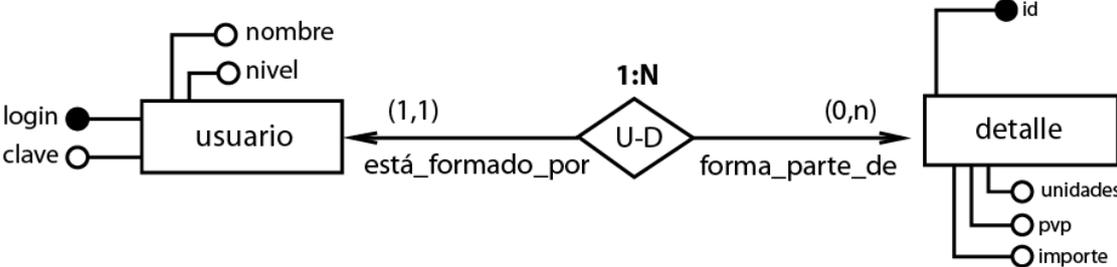


Figura 7: Tipo de interrelación *Usuario-Detalle*

### 7.2.5. Diagrama Entidad-Interrelación (E-R)

Una vez estudiados los diferentes tipos de entidades que componen el sistema y las interrelaciones que existen entre ellos, se obtiene el siguiente diagrama que representa la base de datos a diseñar:

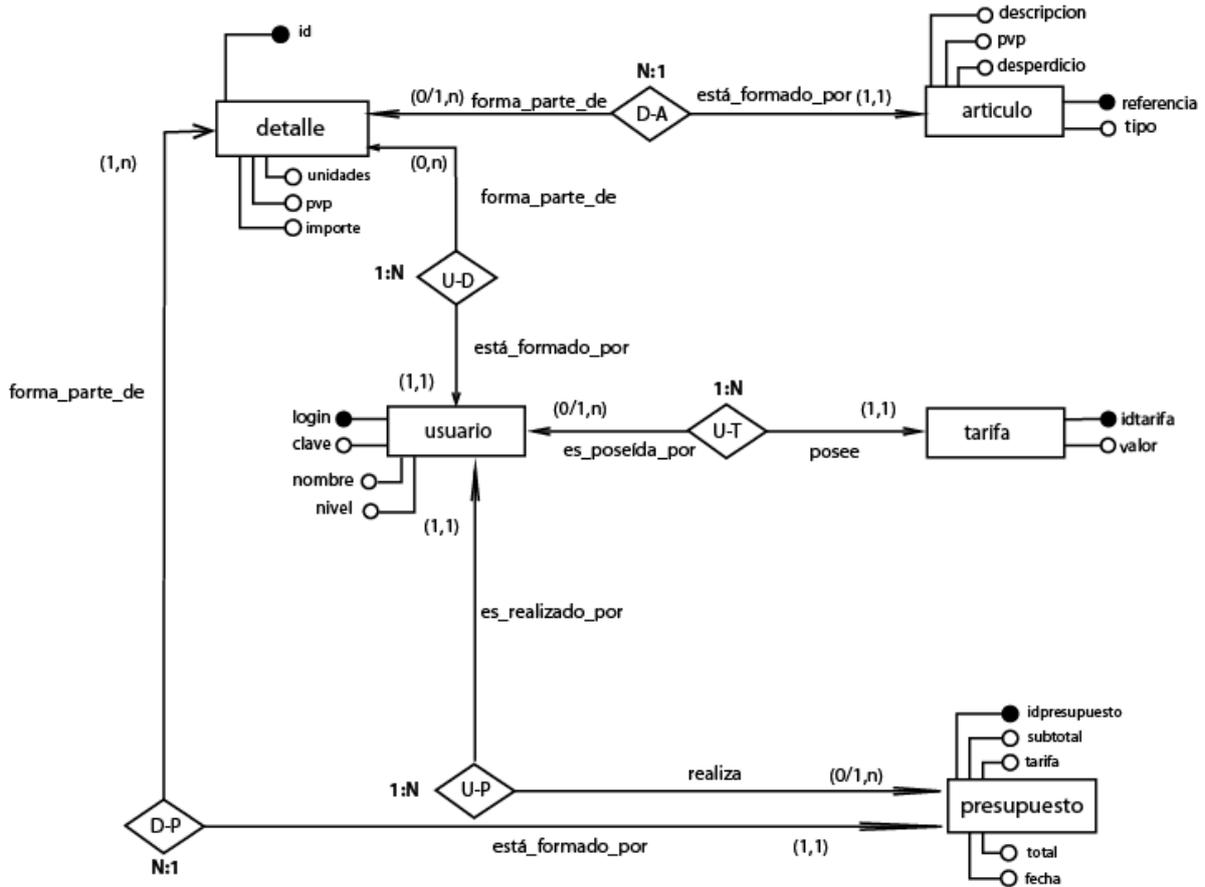


Figura 8: Diagrama E-R de la base de datos

### 7.3. Especificación y modelación de la función

En este punto se describe la funcionalidad que debe tener la aplicación a desarrollar para satisfacer los objetivos, requisitos y restricciones expuestos anteriormente y permitir que la manipulación de la información tratada por la misma se produzca de manera correcta. Esta descripción se realizará analizando por separado cada una de las tareas individuales principales que debe proporcionar la aplicación, estudiando en cada una de ellas la secuencia de transformaciones de la información desde su entrada en el sistema hasta su salida.

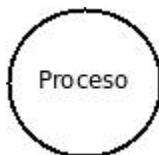
Para llevar a cabo esta tarea se va a hacer uso de dos técnicas: los Diagramas de Flujo de Datos (DFD), usada para el modelado del procesamiento de la información en el sistema, y el Diccionario de Datos (DD), que contiene la definición textual de todos los objetos requeridos o producidos por el software.

#### 7.3.1. Diagramas de Flujo de Datos

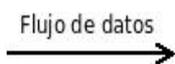
Un Diagrama de Flujo de Datos es una representación gráfica en forma de red que refleja el flujo de la información y las transformaciones que se aplican sobre ella al moverse desde la entrada hasta la salida del sistema. Los DFDs permiten una descomposición funcional del sistema en distintos niveles de abstracción. Los niveles superiores definen las funciones del sistema de forma general, mientras que los niveles inferiores lo hacen de manera más detallada, de forma que no puedan descomponerse más debido a su sencillez.

La metodología seguida para la representación de los DFDs será la expuesta por Yourdon-DeMarco, por ser la más extendida.

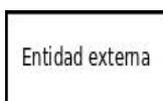
Los componentes básicos utilizados por esta metodología son los siguientes:



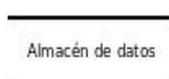
Representa un proceso, es decir, una función que tiene que realizar el sistema y que transforma los flujos de datos de entrada en uno o varios flujos de datos de salida.



Representa un flujo de datos, es decir, los datos que circulan entre las funciones y cuyo sentido viene indicado por la dirección de la flecha.



Representa una entidad externa, es decir, un generador o consumidor de información del sistema y que no pertenece al mismo.



Representa un almacén de datos, es decir, los datos del sistema que se encuentran almacenados, independientemente del dispositivo utilizado.

### 7.3.1.1. Nivel de abstracción 0: Diagrama de contexto

En este nivel se delimita la frontera del sistema con el mundo exterior y define los flujos de entrada y salida del sistema con el entorno.

En la figura 9 se representa el Sistema de Elaboración de Presupuestos como un único proceso en el que recibirá datos de entrada por parte de la entidad externa *Teclado y Ratón del usuario* y mostrará la salida de esos datos por la entidad *Pantalla*.

La entidad externa *Teclado y ratón del usuario* permitirá al usuario moverse por el sistema e introducir información en el mismo.

La entidad externa *Pantalla* representa el monitor como dispositivo que mostrará los datos que introduzca el usuario, los distintos mensajes del sistema, los datos consultados y los resultados de las operaciones realizadas.



Figura 9: Nivel de Abstracción 0: *Diagrama de contexto*

### 7.3.1.2. Nivel de abstracción 1: Diagrama de subsistemas

A continuación se muestran los distintos procesos que compondrán el sistema general representado en el modelo de contexto. También se definen cada uno de los elementos participantes en el diagrama.

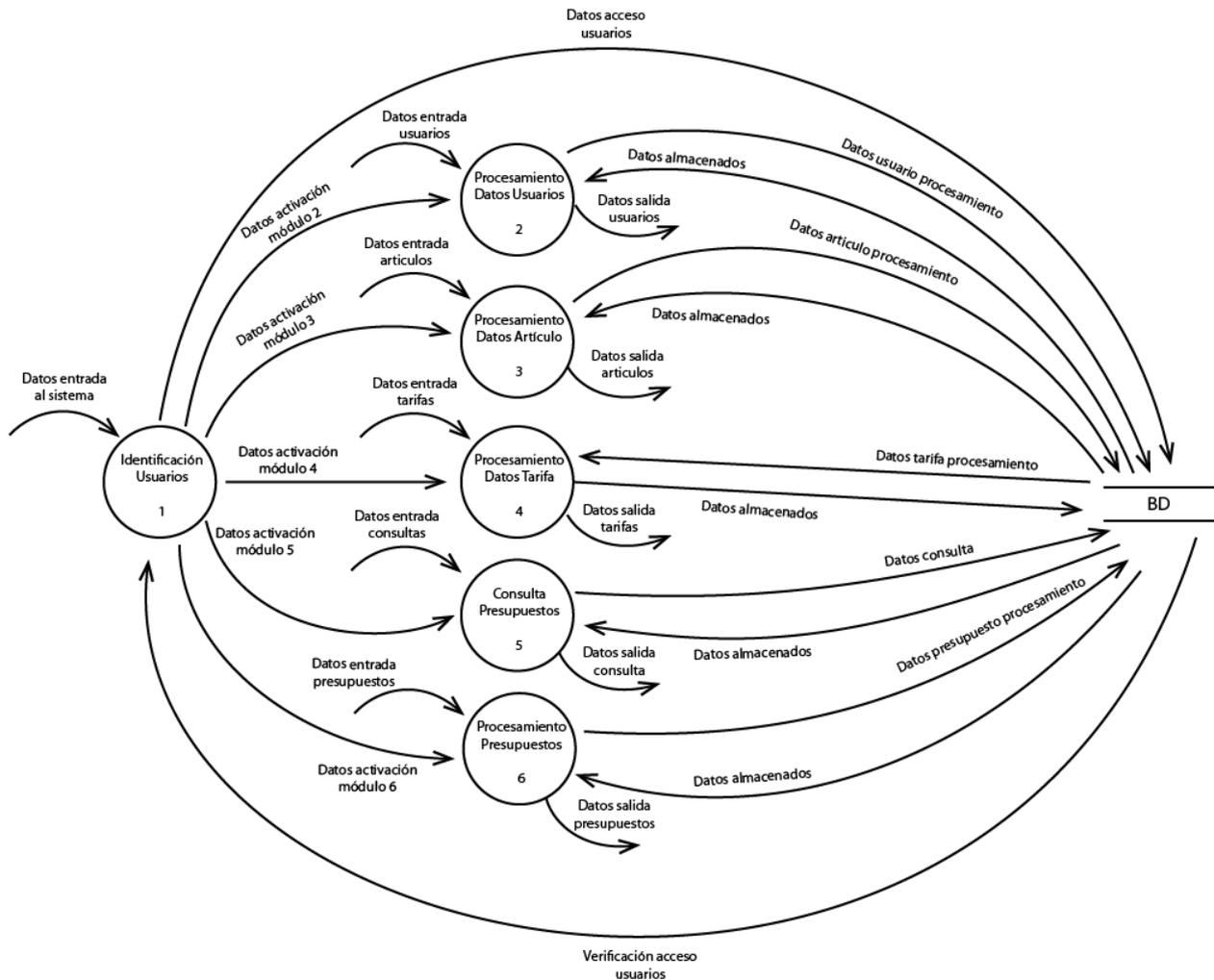


Figura 10: Nivel de abstracción 1: Diagrama de subsistemas

- Subsistema **Identificación Usuarios**: este proceso se encargará de verificar los datos que el usuario introduce en el sistema para su acceso en el mismo. Tras la introducción del login y la contraseña con las entradas del ratón y el teclado, éstos serán verificados en la base de datos. En caso de que sean correctos, el usuario accederá a las diferentes partes del sistema, según sea administrador o usuario cliente. En caso contrario, deberá introducir los datos válidos.
- Subsistema **Procesamiento Datos Usuario**: constará de todas las funciones necesarias para

que el administrador pueda gestionar los datos de los usuarios. De esta manera, podrá realizar el registro de los usuarios en el sistema, además de otros procesos como la modificación, visualización y eliminación de los datos de los usuarios.

- Subsistema **Procesamiento Datos Artículo**: constará de todas las funciones necesarias para que el administrador pueda introducir nuevos artículos en la base de datos y eliminarlos, así como la modificación de los datos y su visualización.
- Subsistema **Procesamiento Datos Tarifa**: permitirá introducir las tarifas y sus correspondientes valores, que tendrá asociado cada usuario. También se podrá modificar estos valores, así como eliminar una determinada tarifa y la visualización de todas las tarifas existentes.
- Subsistema **Consultas Presupuestos**: permitirá consultar los presupuestos realizados según diferentes criterios introducidos.
- Subsistema **Procesamiento Presupuestos**: en él se podrá elaborar presupuestos, modificar sus datos y eliminar aquellos que se deseen. Mientras que se realizan estas acciones, permitirá la visualización de la parte del presupuesto elaborada hasta el momento.

El almacén de datos que se refleja en el diagrama es la base de datos que recoge toda la información que la aplicación debe mantener, es decir, la información de los usuarios, artículos, tarifas y presupuestos. La estructura de este almacén de datos está ampliamente descrita en el apartado 7.2 de este documento.

### **7.3.1.3. Nivel de abstracción 2: Diagrama de funciones de los subsistemas**

En estos diagramas se representan con un menor nivel de abstracción los procesos del DFD de subsistemas que no son primitivos.

#### **Subsistema Procesamiento Datos Usuario**

Constará de una serie de procedimientos que permitirán al administrador controlar todas las funciones relacionadas con la gestión de los usuarios, como:

- Función **Crear Usuario**: se introducirán todos los datos referentes al usuario. Una vez rellenados todos los campos informativos, se almacenarán en la base de datos. Si hay algún error en los datos, se mostrará un mensaje de validación alertando del fallo. Si todo el proceso se realiza con éxito, aparecerá el usuario creado en el listado total de usuarios.

- Función **Eliminar Usuario**: permitirá el borrado permanente en la base de datos de toda la información de un usuario. Para eliminar el usuario deseado, bastará con pulsar el botón de borrado mostrado para tal efecto en el listado de usuarios. Dicho registro estará asociado al identificador del usuario a eliminar. Una vez eliminado el usuario, se recargará la lista del resto de usuarios existentes en la base de datos.
- Función **Modificar Usuario**: permitirá la modificación en la base de datos de la información de un usuario, exceptuando su identificador login. Para modificar el usuario deseado, bastará con pulsar el botón de edición mostrado para tal efecto en el listado de usuarios. Dicho registro estará asociado al identificador del usuario a editar. Una vez realizado el proceso, el cambio se hará efectivo en la base de datos. Si la acción ha sido correcta, se mostrará el listado de todos los usuarios, incluido el usuario modificado . En el caso de que hubiera algún error, aparecerá un mensaje de validación que especificará tal hecho.
- Función **Visualizar Usuarios**: mostrará todos los datos de todos los usuarios registrados en el sistema. Toda esta información es recuperada de la base de datos.

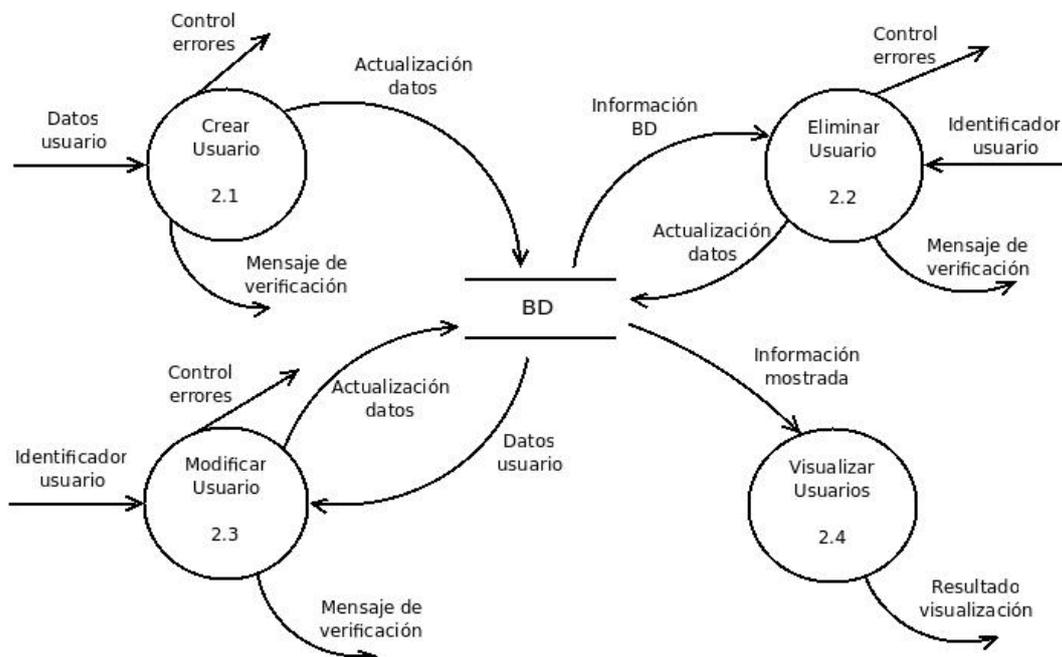
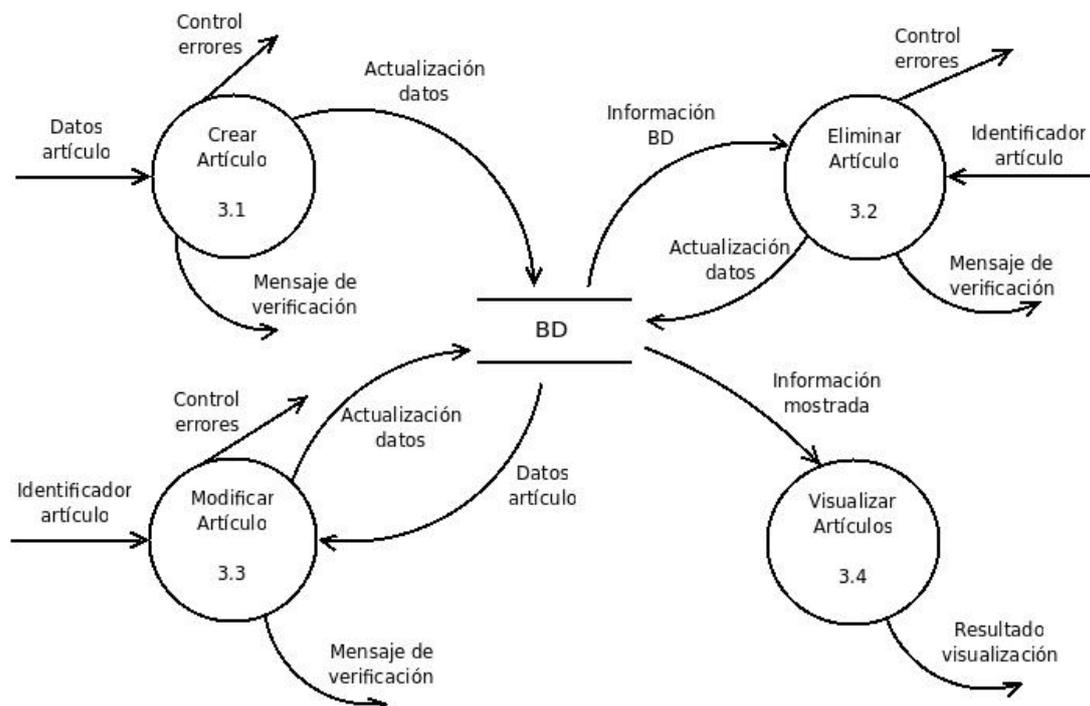


Figura 11: Nivel de abstracción 2: Refinamiento *Procesamiento Datos Usuario*

### Subsistema Procesamiento Datos Artículo

Constará de una serie de procedimientos que permitirán al administrador controlar todas las funciones relacionadas con la gestión de los artículos, como:

- Función **Crear Artículo**: se introducirán todos los datos referentes al artículo. Una vez rellenados todos los campos informativos, se almacenarán en la base de datos. Si hay algún error en los datos, se mostrará un mensaje de validación alertando del fallo. Si todo el proceso se realiza con éxito, aparecerá el usuario creado en el listado total de artículos.
- Función **Eliminar Artículo**: permitirá el borrado permanente en la base de datos de toda la información de un artículo. Para eliminar el artículo deseado, bastará con pulsar el botón de borrado mostrado para tal efecto en el listado de artículos. Dicho registro estará asociado al identificador del artículo a eliminar. Una vez eliminado el artículo, se recargará la lista del resto de artículos existentes en la base de datos.
- Función **Modificar Artículo**: permitirá la modificación en la base de datos de la información de un artículo, exceptuando su identificador. Para modificar el artículo deseado, bastará con pulsar el botón de edición mostrado para tal efecto en el listado de artículos. Dicho registro estará asociado al identificador del artículo a editar. Una vez realizado el proceso, el cambio se hará efectivo en la base de datos. Si la acción ha sido correcta, se mostrará el listado de todos los artículos, incluido el artículo modificado . En el caso de que hubiera algún error, aparecerá un mensaje de validación que especificará tal hecho.
- Función **Visualizar Artículos**: mostrará todos los datos de todos los artículos registrados en el sistema. Toda esta información es recuperada de la base de datos.

Figura 12: Nivel de abstracción 2: Refinamiento *Procesamiento Datos Artículo*

### Subsistema Procesamiento Datos Tarifa

Constará de una serie de procedimientos que permitirán al administrador controlar todas las funciones relacionadas con la gestión de las tarifas, como:

- Función **Crear Tarifa**: se introducirán todos los datos referentes a la tarifa. Una vez rellenos todos los campos informativos, se almacenarán en la base de datos. Si hay algún error en los datos, se mostrará un mensaje de validación alertando del fallo. Si todo el proceso se realiza con éxito, aparecerá la tarifa creada en el listado total de tarifas.
- Función **Eliminar Tarifa**: permitirá el borrado permanente en la base de datos de toda la información de una tarifa. Para eliminar la tarifa deseada, bastará con pulsar el botón de borrado mostrado para tal efecto en el listado de tarifas. Dicho registro estará asociado al identificador de la tarifa a eliminar. Una vez eliminado la tarifa, se recargará la lista del resto de tarifas existentes en la base de datos.
- Función **Modificar Tarifa**: permitirá la modificación en la base de datos de la información de una tarifa, exceptuando su identificador. Para modificar la tarifa deseado, bastará con pulsar el botón de edición mostrado para tal efecto en el listado de tarifas. Dicho registro estará asociado

al identificador de la tarifa a editar. Una vez realizado el proceso, el cambio se hará efectivo en la base de datos. Si la acción ha sido correcta, se mostrará el listado de todas las tarifas, incluido la tarifa modificada. En el caso de que hubiera algún error, aparecerá un mensaje de validación que especificará tal hecho.

- Función **Visualizar Tarifa**: mostrará todos los datos de todas las tarifas registradas en el sistema. Toda esta información es recuperada de la base de datos.

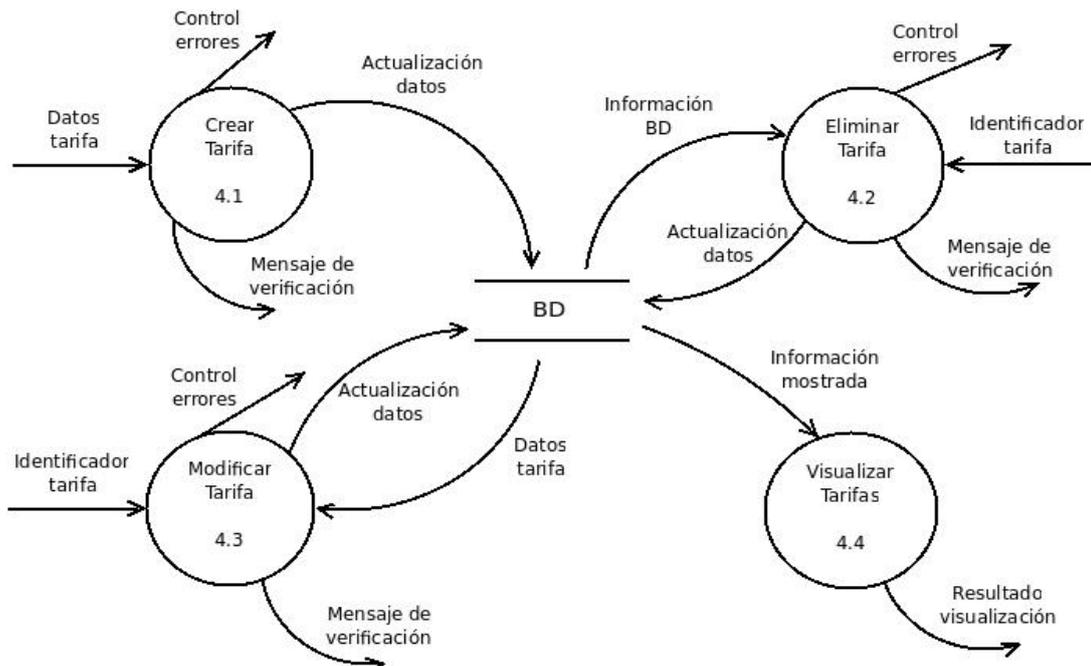


Figura 13: Nivel de abstracción 2: Refinamiento *Procesamiento Datos Tarifa*

### Subsistema Consultas Presupuestos

Permitirá al administrador consultar los presupuestos realizados por los clientes en base a unos criterios de búsqueda introducidos.

- Función **Buscar Presupuestos**: proceso de búsqueda de un presupuesto mediante criterio de fechas. Este filtrado se buscará en la base de datos. Si no se encuentra ninguna coincidencia, el listado se mostrará vacío.
- Función **Visualizar Presupuestos**: se mostrará un listado de todos los presupuestos y de los resultantes en el filtrado de búsqueda cuando se realice. Toda esta información se extrae de la base de datos y se muestra.

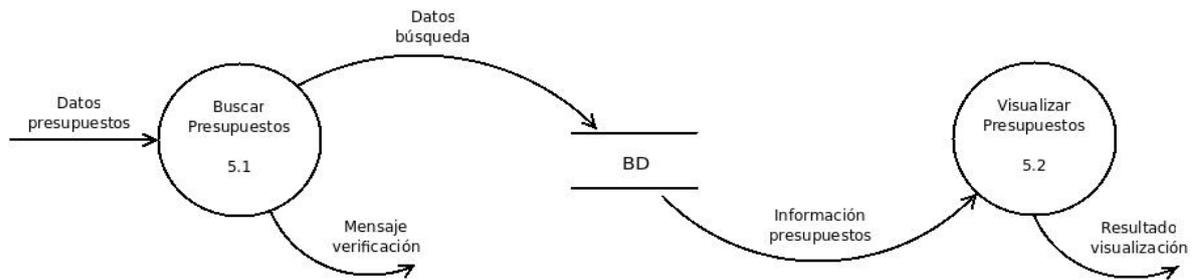


Figura 14: Nivel de abstracción 2: Refinamiento *Consultas Presupuestos*

### Subsistema Procesamiento Presupuestos

Permitirá a los usuarios clientes gestionar su propio presupuesto. Para ello podrán crear un nuevo presupuesto, modificar los datos de los artículos pertenecientes a él y borrar un artículo o varios. A la misma vez que el usuario va realizando estas operaciones, estará viendo en todo momento el resultado del presupuesto que está elaborando.

- Función **Crear Presupuesto**: se introducirán los datos referentes a un artículo a través de una búsqueda con concordancia de los artículos almacenados en la base de datos. Los datos correspondientes al artículo seleccionado (referencia, descripción y pvp) se mostrarán en el presupuesto, así como la tarifa correspondiente al usuario logueado. El usuario tendrá que indicar el número de unidades de dicho artículo. Con toda esta información, se calculará el importe del artículo respecto a la cantidad del mismo, el subtotal de los importes de todos los artículos introducidos y el importe total (calculado en base a la tarifa del usuario). Una vez terminado el presupuesto, el usuario podrá guardar dicha información y se almacenarán en la base de datos. Si hay algún error en los datos, se mostrará un mensaje alertando del fallo.
- Función **Eliminar Artículo Presupuesto**: permitirá el borrado de cualquier artículo perteneciente al presupuesto. Para ello, el usuario sólo tendrá que pulsar el botón de borrado asociado a cada artículo.
- Función **Modificar Presupuesto**: en cualquier momento de la elaboración de presupuesto, el usuario podrá modificar los datos de los artículos, así como añadir o eliminar artículos al mismo.
- Función **Visualizar Presupuesto**: mostrará en tiempo real los datos de elaboración del presupuesto.

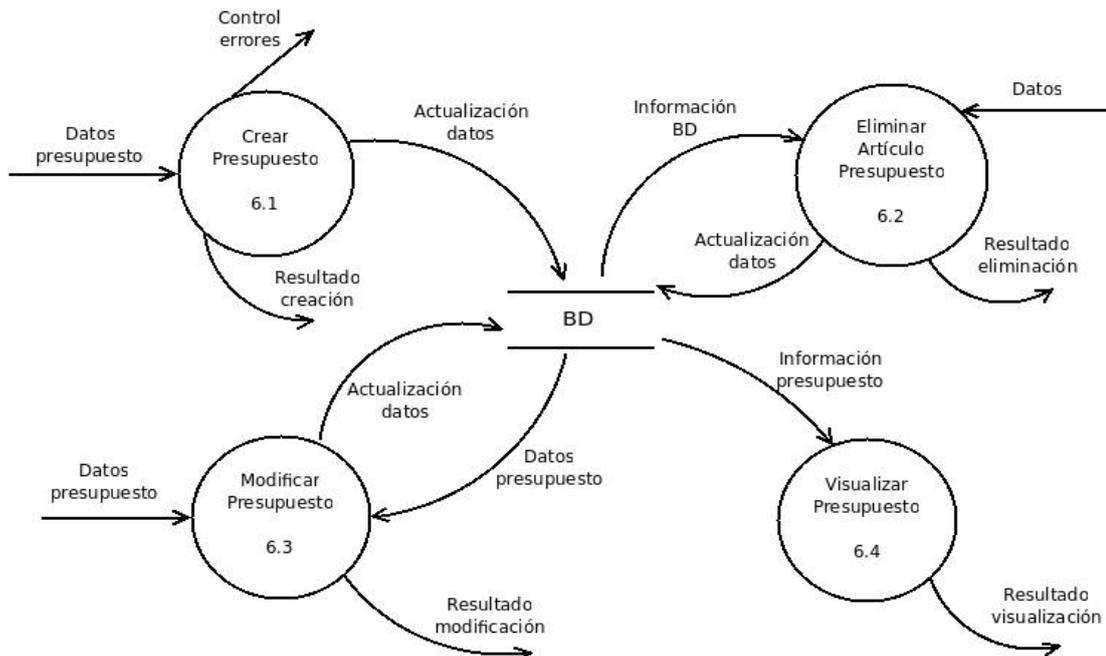


Figura 15: Nivel de abstracción 2: Refinamiento *Procesamiento Datos Presupuesto*

### 7.3.1.4. Nivel de abstracción 3: Diagrama de subfunciones asociadas a cada función

A continuación se detallan las subfunciones integrantes de las funciones del nivel anterior.

#### Función Crear Usuario

- Subfunción **Introducir Datos Usuario:** permitirá cumplimentar todos los campos referentes a la información del usuario, que son login, clave, nombre e idtarifa.
- Subfunción **Almacenar Datos:** almacenará en la base de datos todos los datos introducidos anteriormente.

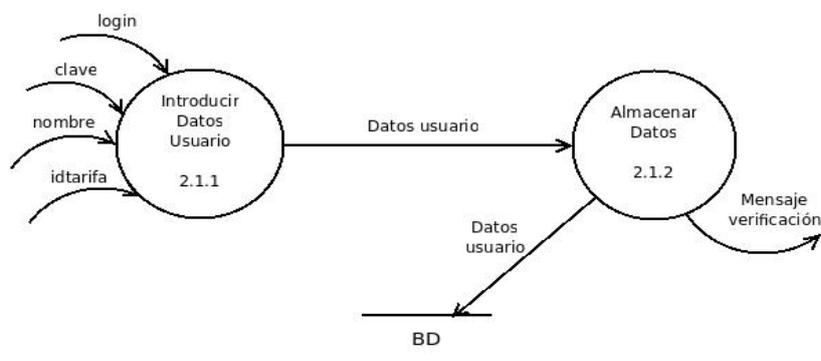


Figura 16: Nivel de abstracción 3: Refinamiento *Crear Usuario*

### Función Eliminar Usuario

- Subfunción **Introducir Identificador Usuario**: se encargará de recoger, a través de un botón para tal efecto asociado a cada usuario, el login del usuario para posteriormente buscar su registro en la base de datos.
- Subfunción **Buscar Usuario**: permitirá buscar los datos del usuario cuyo login se ha introducido en la base de datos.
- Subfunción **Eliminar Usuario**: permitirá borrar permanentemente de la base de datos toda la información referente al usuario indicado y que está almacenado en ella.

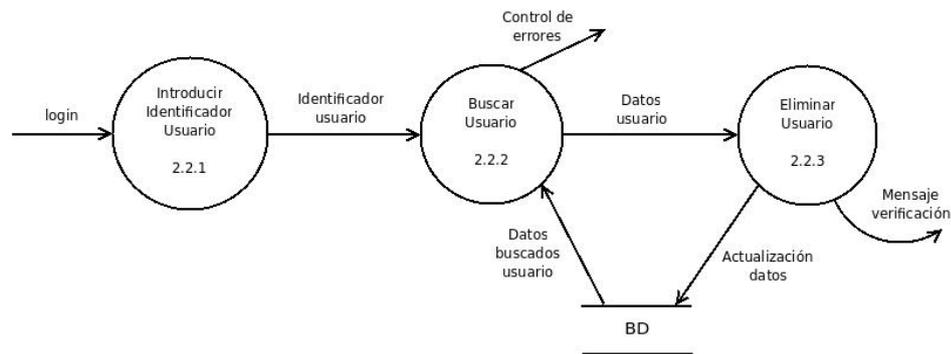


Figura 17: Nivel de abstracción 3: Refinamiento *Eliminar Usuario*

### Función Modificar Usuario

- Subfunción **Introducir Identificador Usuario**: se encargará de recoger, a través de un botón para tal efecto asociado a cada usuario, el login del usuario para posteriormente buscar su registro en la base de datos.
- Subfunción **Buscar Usuario**: permitirá buscar los datos del usuario cuyo login se ha introducido en la base de datos.
- Subfunción **Modificar Clave**: permitirá sustituir la clave incorrecta del usuario por la nueva clave.
- Subfunción **Modificar Nombre**: permitirá sustituir el nombre incorrecto del usuario por el nuevo nombre.
- Subfunción **Modificar Idtarifa**: permitirá sustituir el idtarifa incorrecto del usuario por el nuevo idtarifa.

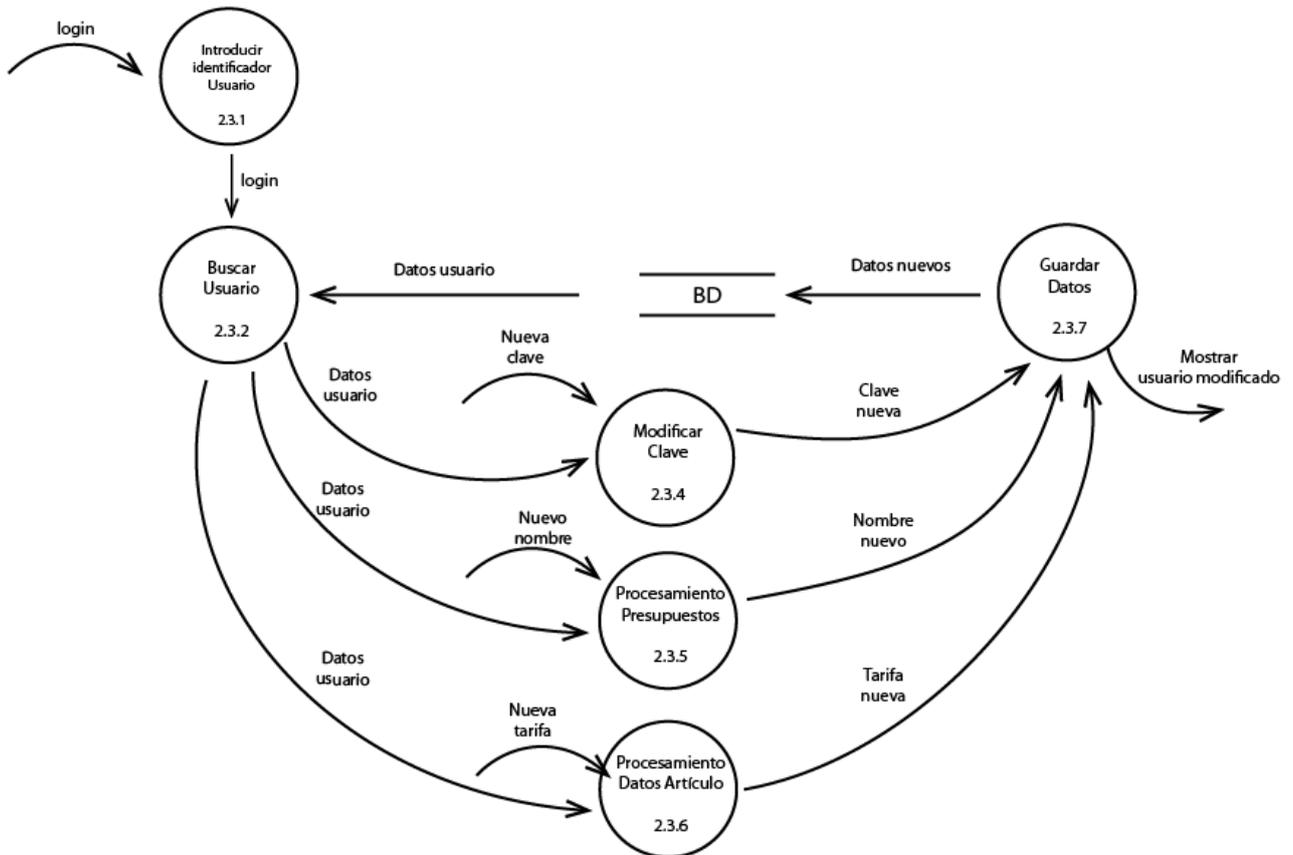


Figura 18: Nivel de abstracción 3: Refinamiento *Modificar Usuario*

**Función Crear Artículo**

- Subfunción **Introducir Datos Artículo**: permitirá cumplimentar todos los campos referentes a la información del artículo, que son referencia, tipo, descripción, pvp y desperdicio.
- Subfunción **Almacenar Datos**: almacenará en la base de datos todos los datos introducidos anteriormente.

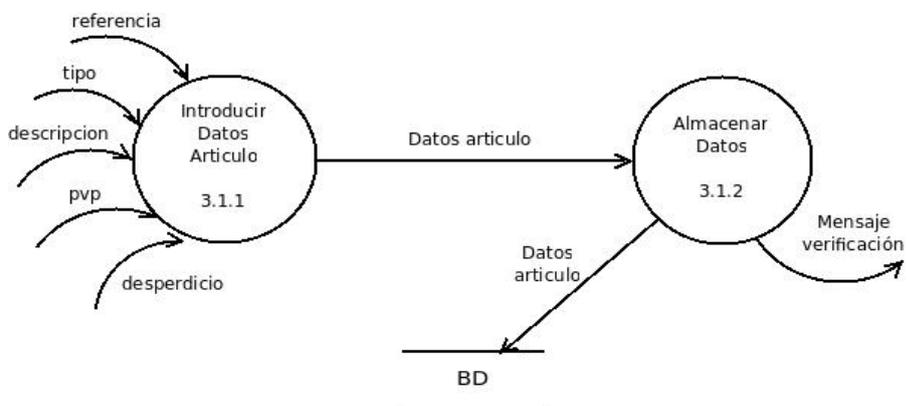


Figura 19: Nivel de abstracción 3: Refinamiento *Crear Artículo*

### Función Eliminar Artículo

- Subfunción **Introducir Identificador Artículo**: se encargará de recoger, a través de un botón para tal efecto asociado a cada artículo, la referencia del artículo para posteriormente buscar su registro en la base de datos.
- Subfunción **Buscar Artículo**: permitirá buscar los datos del artículo cuya referencia se ha introducido en la base de datos. Si el identificador introducido no está registrado en el sistema, se informará de la incidencia.
- Subfunción **Eliminar Artículo**: permitirá borrar permanentemente de la base de datos toda la información referente al artículo buscado y que está almacenado en ella. Se mostrará un mensaje confirmando si la operación se ha realizado correctamente.

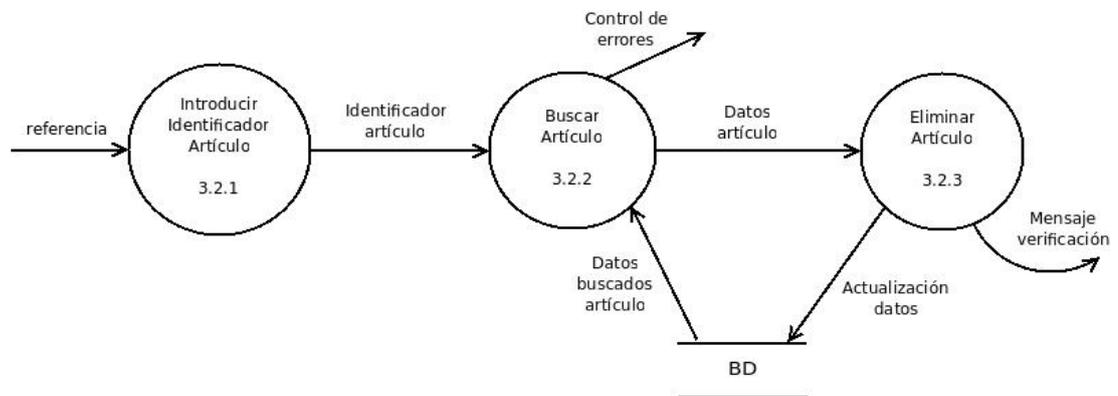


Figura 20: Nivel de abstracción 3: Refinamiento *Eliminar Artículo*

### Función Modificar Artículo

- Subfunción **Introducir Identificador Artículo**: se encargará de recoger, a través de un botón para tal efecto asociado a cada artículo, la referencia del artículo para posteriormente buscar su registro en la base de datos.
- Subfunción **Buscar Artículo**: permitirá buscar los datos del artículo cuya referencia se ha introducido en la base de datos.
- Subfunción **Modificar Tipo**: permitirá sustituir el tipo de artículo equivocado por el nuevo tipo.
- Subfunción **Modificar Descripción**: permitirá sustituir la descripción equivocada del artículo por la nuevo descripción.
- Subfunción **Modificar pvp**: permitirá sustituir el precio equivocado del artículo por el nuevo precio.
- Subfunción **Modificar Desperdicio**: permitirá sustituir el desperdicio equivocado del artículo por el nuevo desperdicio.

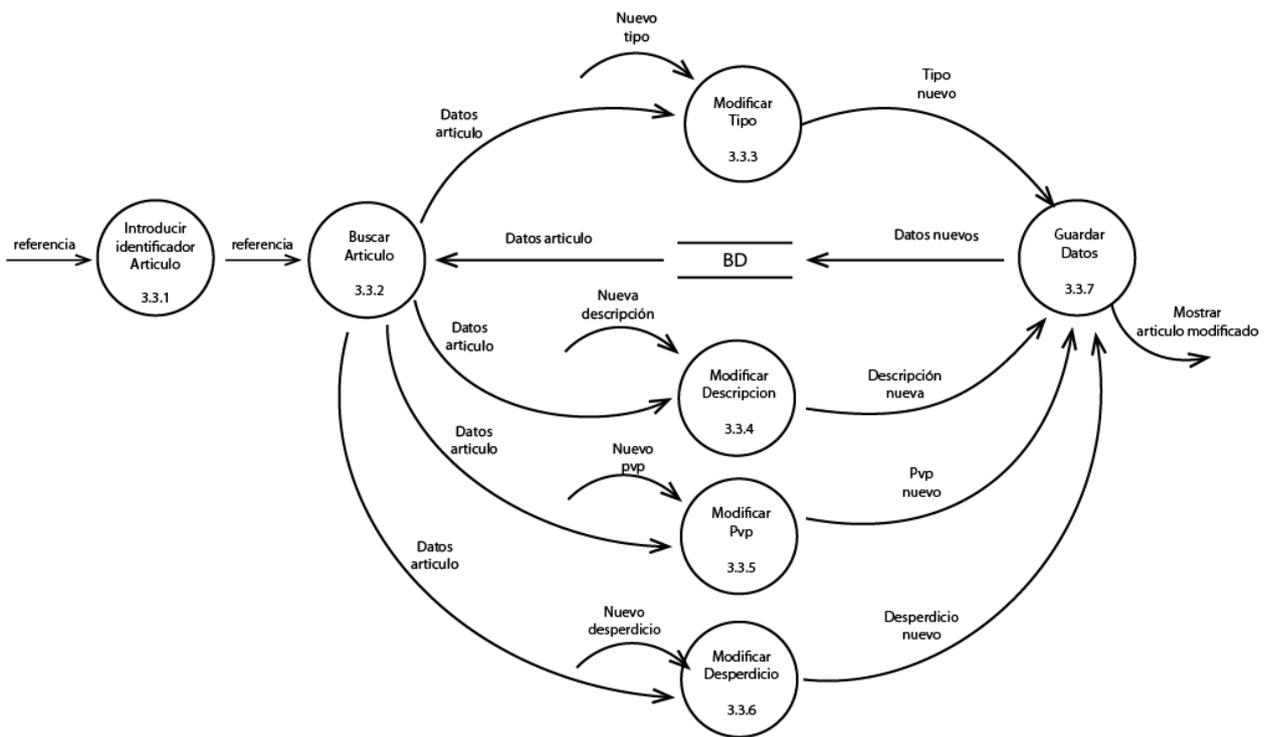


Figura 21: Nivel de abstracción 3: Refinamiento *Modificar Artículo*

**Función Crear Tarifa**

- Subfunción **Introducir Datos Tarifa**: permitirá cumplimentar todos los campos referentes a la información de la tarifa, que son idtarifa y valor.
- Subfunción **Almacenar Datos**: almacenará en la base de datos todos los datos introducidos anteriormente.

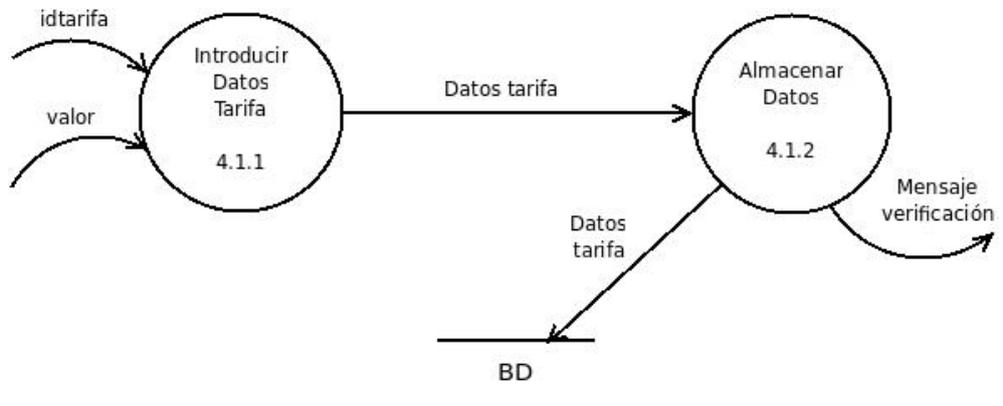


Figura 22: Nivel de abstracción 3: Refinamiento *Crear Tarifa*

### Función Eliminar Tarifa

- Subfunción **Introducir Identificador Tarifa**: se encargará de recoger, a través de un botón para tal efecto asociado a cada tarifa, el idtarifa de la tarifa para posteriormente buscar su registro en la base de datos.
- Subfunción **Buscar Artículo**: permitirá buscar los datos de la tarifa cuyo idtarifa se ha introducido en la base de datos. Si el identificador introducido no está registrado en el sistema, se informará de la incidencia.
- Subfunción **Eliminar Tarifa**: permitirá borrar permanentemente de la base de datos toda la información referente a la tarifa buscada y que está almacenada en ella. Se mostrará un mensaje confirmando si la operación se ha realizado correctamente.

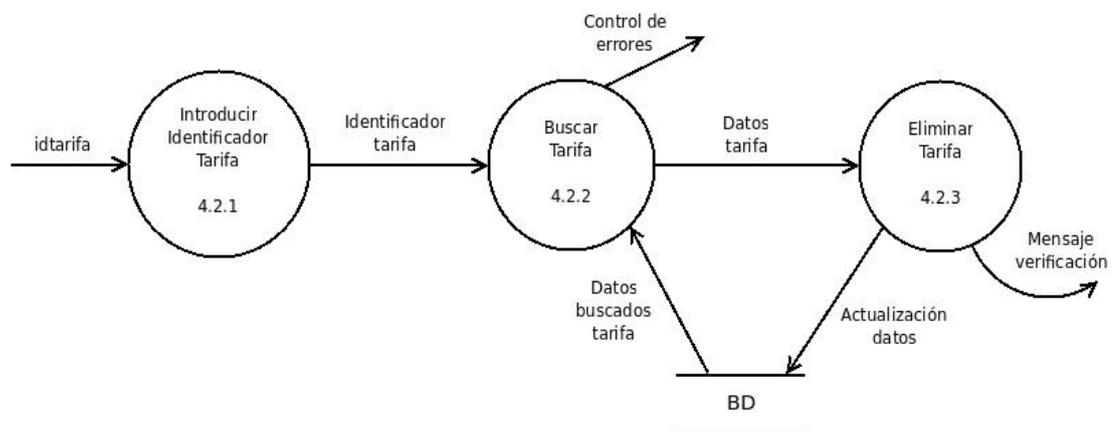


Figura 23: Nivel de abstracción 3: Refinamiento *Eliminar Tarifa*

### Función Modificar Tarifa

- Subfunción **Introducir Identificador Tarifa**: se encargará de recoger, a través de un botón para tal efecto asociado a cada tarifa, el idtarifa de la tarifa para posteriormente buscar su registro en la base de datos.
- Subfunción **Buscar Tarifa**: permitirá buscar los datos de la tarifa cuyo idtarifa se ha introducido en la base de datos.
- Subfunción **Modificar Idtarifa**: permitirá sustituir el idtarifa equivocado de la tarifa por el nuevo idtarifa.
- Subfunción **Modificar Valor**: permitirá sustituir el valor equivocado de la tarifa por el nuevo valor.

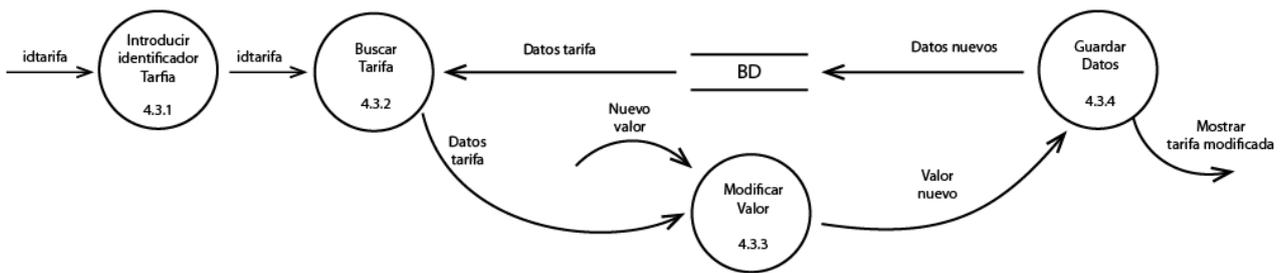


Figura 24: Nivel de abstracción 3: Refinamiento *Modificar Tarifa*

### Función Crear Presupuesto

- Subfunción **Introducir Datos Presupuesto**: permitirá cumplimentar todos los campos referentes a la información del presupuesto, que son la referencia y cantidad de los artículos. Los datos idpresupuesto, descripción y pvp del artículo, tarifa, fecha y login los introducirá el sistema, así como los datos importe, subtotal y total se calculará a partir de una determinada fórmula.
- Subfunción **Almacenar Datos**: almacenará en la base de datos todos los datos introducidos anteriormente.

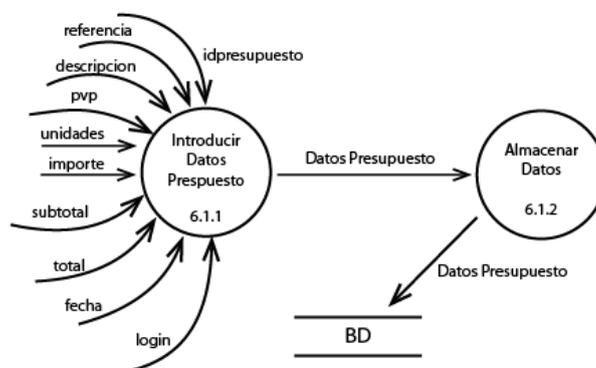


Figura 25: Nivel de abstracción 3: Refinamiento *Crear Presupuesto*

### Función Eliminar Artículo Presupuesto

- Subfunción **Buscar Artículo**: mediante la selección del artículo que se desea eliminar a través de un botón para tal efecto asociado a cada artículo, el usuario podrá acceder a dicho artículo.
- Subfunción **Eliminar Artículo**: permitirá eliminar el artículo seleccionado.

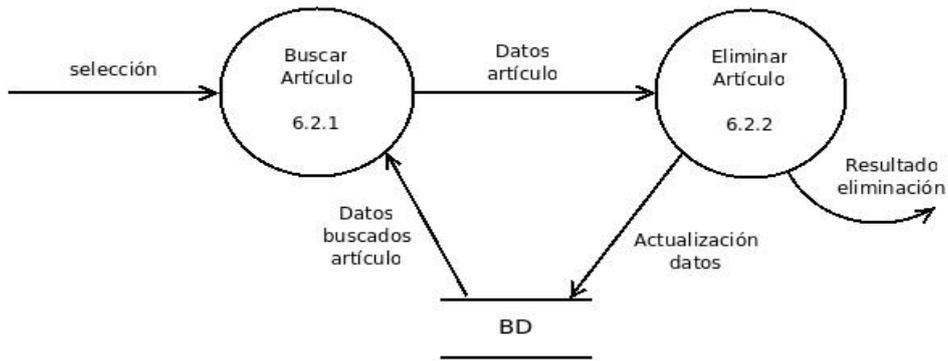


Figura 26: Nivel de abstracción 3: Refinamiento *Eliminar Artículo Presupuesto*

**Función Modificar Presupuesto**

- Subfunción **Modificar Unidades**: permitirá sustituir la unidad equivocada del artículo por la nueva unidad.
- Subfunción **Modificar Importe, Subtotal y Total**: se recalcurán los nuevos importe, subtotal y total según los anteriores nuevos datos introducidos.

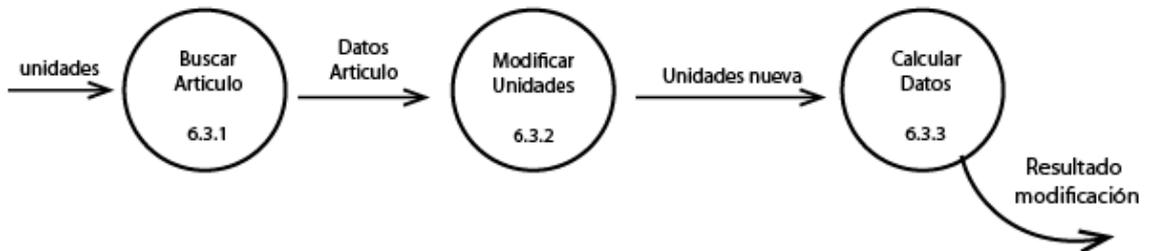


Figura 27: Nivel de abstracción 3: Refinamiento *Modificar Presupuesto*

**7.4. Especificación de requisitos de la interfaz**

La interfaz de cualquier aplicación es la parte visible de la misma y, por tanto, lo que el usuario percibe a primera vista. La interfaz es la encargada de la interacción entre el hardware, software y el usuario, es decir, es la que se ocupa de la comunicación entre el usuario y la aplicación. De ahí que sea uno de los elementos más importantes dentro de cualquier proyecto informático.

El usuario de la aplicación es el que determina en gran medida el éxito de la misma, ya que será el que la utilice y vea las ventajas e inconvenientes de manejarla. Si el usuario se encuentra a gusto con la aplicación y su uso le resulta fácil, el éxito de la misma está prácticamente asegurado.

La interfaz que se pretende desarrollar es una interfaz de tipo web, es decir, una interfaz que se visualizará a través de cualquier navegador web.

### **7.4.1. Ergonomía de la interfaz**

La interfaz que se pretende desarrollar deberá cumplir una serie de características en función de la solución por la que se ha optado para resolver el problema y de la naturaleza del problema en sí. Además, una interfaz bien diseñada mejora la percepción del contenido. Así pues, a continuación se propondrá las características que se creen convenientes para dotar a la aplicación de una interfaz bien estructurada y ergonómica.

### **7.4.2. Ubicación de la aplicación en la pantalla**

Por el uso que se le puede dar a la aplicación, la ubicación de la ventana ocupará toda la pantalla. No obstante, el usuario podrá redimensionarla, maximizándola o minimizándola, cuando desee, tal y como se puede hacer en cualquier aplicación del mercado.

### **7.4.3. Información a mostrar, formato y situación de la misma en la pantalla**

El objetivo de la interfaz es que el usuario pueda interactuar con ella de forma sencilla, eficiente y amigablemente, evitando que tenga que memorizar las diferentes opciones que se le presentan. Para ello, se mostrará la información de la siguiente forma:

- La cabecera, el pie de página y el menú tendrán la misma estructura en los diferentes subsistemas de la aplicación, variando el contenido del menú de acuerdo a las distintas opciones de las que disponen los dos tipos de usuario y el nombre de la sección en la que se encuentre el usuario. La parte central de la aplicación se destinará para el grueso de las interacciones y acciones a realizar.
- El estilo y formato de la interfaz será consistente y atractivo, presentando la información ordenadamente.
- Los iconos y botones se colocarán de forma intuitiva para el usuario y los enlaces a otras páginas serán representativos con el contenido al que direccionan.
- La información que se visualizará en cada pantalla será muy significativa y relevante. No se mostrarán datos carentes de sentido en el contexto en el que se presentan, minimizando así la cantidad de información que el usuario debe memorizar.

- La información se mostrará de manera clara y concisa, resaltando los datos importantes y con un correcto uso de las tabulaciones, las mayúsculas y las minúsculas.
- El usuario podrá acceder a las diferentes partes de la aplicación desde los enlaces de los menús, en la parte izquierda de la pantalla.

Por lo que se puede comprobar, toda la información que se presenta seguirá una estructura casi o muy similar en todas las páginas, para que el usuario pueda localizar los datos rápida y cómodamente.

#### **7.4.4. Validación de los datos de entrada**

La interfaz debe contener una protección contra errores de entrada y recoger los datos reduciendo el número de teclas pulsadas por el usuario. En determinados momentos el usuario dispondrá de elementos de selección de opciones que muestren únicamente los valores permitidos para minimizar el número de entradas en los formularios.

Se controlará los datos introducidos por el usuario para que tengan el formato correcto y coincidente con el de los campos de la base de datos.

En el caso de que se viole alguna restricción, se mostrará mensajes indicando al usuario que ha realizado una acción no válida, describiéndole el problema.

## 8. Especificación del sistema

En este apartado se describirá la manera en que debe realizarse el diseño del sistema para garantizar que el mismo cumpla la especificación de requisitos anteriormente detallada.

### 8.1. Diseño de datos

Los objetos de datos definidos durante el análisis de requisitos del software fueron modelados utilizando el Diagrama Entidad-Relación y el Diccionario de Datos en el capítulo anterior. La actividad de diseño de datos traduce esos elementos del modelo de requisitos en estructuras de datos a nivel de los componentes del software y en arquitectura de base de datos a nivel de aplicación.

En primer lugar se realizará la representación lógica de la estructura de datos definitiva con ayuda del modelo relacional. Seguidamente, se normalizará el modelo para que sea más consistente y así evitar futuros errores de referencia y de integridad en la información. En último lugar, se presentará la definición sintáctica de la estructura de datos utilizada en el sistema

#### 8.1.1. Modelo relacional

El objetivo de este apartado es definir lógicamente y físicamente el dominio de la información, partiendo de la especificación realizada en la fase de análisis de requisitos

A partir del diagrama Entidad-Interrelación (ver apartado 7) se va a proceder a su transformación en el modelo relacional del presente problema.

A continuación se detallarán cada una de las tablas obtenidas mediante la aplicación de las reglas de transformación para pasar del modelo conceptual al modelo relacional.

##### **Tabla Tarifa**

Esta tabla se forma a partir del tipo de entidad *Tarifa*, tomando de éste todos sus atributos: *idtarifa* y *valor* (regla RTECAR-1) (Luque Ruiz, 2001). La clave principal de la tabla es el atributo *idtarifa*, no considerándose ningún otro atributo como clave alterna.

Ninguno de los atributos de esta tabla mantiene referencia alguna con atributos de las otras tablas. Por lo tanto, la tabla *Tarifa* queda de la forma siguiente:

**Tarifa** (idtarifa, valor)

**Tabla Usuario**

Esta tabla se forma a partir del tipo de entidad del mismo nombre (regla RTECAR-1) (Luque Ruiz, 2001). Esta tabla tiene los atributos *login*, *clave*, *nombre* y *nivel*. El identificador de esta tabla es el atributo *login*.

Además, la tabla *Usuario* incorpora el atributo *idtarifa* del tipo de entidad *Tarifa* por ser el identificador de ésta, ya que mantiene una relación uno a muchos en el tipo de interrelación (U-T) con este tipo de entidad, y a través de este atributo mantiene referencia con la tabla *Tarifa* (regla RTECAR-3.1) (Luque Ruiz, 2001). Por tanto, la tabla *Usuario* queda de la forma:

**Usuario** (login, clave, nombre, nivel, idtarifa)

**Tabla Artículo**

Esta tabla se forma a partir del tipo de entidad *Articulo*, tomando de éste todos sus atributos: *referencia*, *tipo*, *descripcion*, *pvp* y *desperdicio* (regla RTECAR-1) (Luque Ruiz, 2001). La clave principal de la tabla es el atributo *referencia*, no considerándose ningún otro atributo como clave alterna.

Ninguno de los atributos de esta tabla mantiene referencia alguna con atributos de las otras tablas. Por lo tanto, la tabla *Articulo* queda de la forma siguiente:

**Articulo** (referencia, tipo, descripcion, pvp, desperdicio)

**Tabla Presupuesto**

Esta tabla se forma a partir del tipo de entidad del mismo nombre (regla RTECAR-1) (Luque Ruiz, 2001). Esta tabla tiene los atributos *idpresupuesto*, *subtotal*, *tarifa*, *total* y *fecha*. El identificador de esta tabla es el atributo *idpresupuesto*.

Además, la tabla *Presupuesto* incorpora el atributo *login* del tipo de entidad *Usuario* por ser el identificador de ésta, ya que mantiene una relación uno a muchos en el tipo de interrelación (U-P) con este tipo de entidad, y a través de este atributo mantiene referencia con la tabla *Usuario* (regla RTECAR-3.1) (Luque Ruiz, 2001). Por tanto, la tabla *Presupuesto* queda de la forma:

**Presupuesto** (idpresupuesto, subtotal, tarifa, total, fecha, login)

### Tabla *Detalle*

Esta tabla se forma a partir del tipo de entidad del mismo nombre (regla RTECAR-1) (Luque Ruiz, 2001). Esta tabla tiene los atributos *id*, *unidades*, *pvp* e *importe*. El identificador de esta tabla es el atributo *id*.

Además, la tabla *Detalle* incorpora

- el atributo *login* del tipo de entidad *Usuario* por ser el identificador de ésta, ya que mantiene una relación uno a muchos en el tipo de interrelación (*U-P*) con este tipo de entidad, y a través de este atributo mantiene referencia con la tabla *Usuario* (regla RTECAR-3.1) (Luque Ruiz, 2001).
- el atributo *referencia* del tipo de entidad *Articulo* por ser el identificador de ésta, ya que mantiene una relación uno a muchos en el tipo de interrelación (*D-A*) con este tipo de entidad, y a través de este atributo mantiene referencia con la tabla *Articulo* (regla RTECAR-3.1) (Luque Ruiz, 2001).
- el atributo *idpresupuesto* del tipo de entidad *Presupuesto* por ser el identificador de ésta, ya que mantiene una relación uno a muchos en el tipo de interrelación (*D-P*) con este tipo de entidad, y a través de este atributo mantiene referencia con la tabla *Presupuesto* (regla RTECAR-3.1) (Luque Ruiz, 2001).

Por tanto, la tabla *Detalle* queda de la forma:

**Detalle** (id, unidades, pvp, importe, idpresupuesto, referencia, login)

## 8.1.2. Normalización del modelo

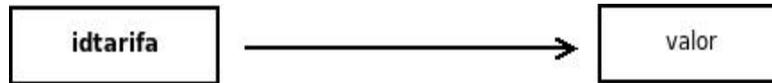
El modelo relacional está basado en el principio básico de normalización, el cual debe aplicarse independientemente del sistema de gestión de bases de datos utilizado.

Los principios de normalización eliminan redundancias superfluas, reduciendo el espacio requerido para el almacenamiento de la información y, por tanto, disminuyendo los posibles problemas de integridad de la información almacenada en la base de datos.

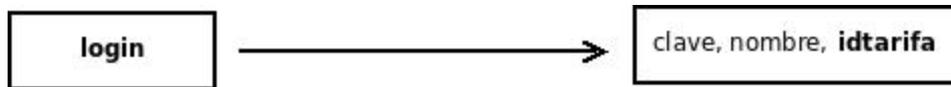
Se va a realizar el proceso de normalización sobre cada una de las tablas obtenidas en el modelo relacional, y se dará por concluido en el momento en el que todas las tablas se encuentren en Forma Normal de Boyce-Codd (FNBC), ya que así se considera que el esquema relacional obtenido es suficientemente consistente.

Para llevar a cabo este proceso se hará uso de los Diagramas de Dependencias Funcionales, que permiten comprobar las dependencias funcionales existentes entre los atributos de las tablas.

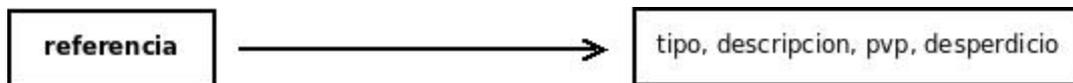
**Tabla *Tarifa*:** esta tabla se encuentra en FNBC, ya que todos los atributos son atómicos, es decir, son valores únicos (FN1), y las únicas dependencias existentes son entre los atributos no primos y los primos.



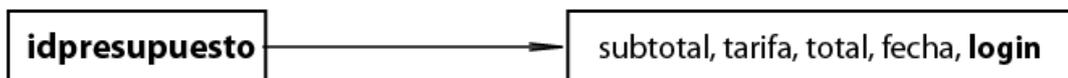
**Tabla *Usuario*:** esta tabla se encuentra en FNBC, ya que todos los atributos son atómicos, es decir, son valores únicos (FN1), y las únicas dependencias existentes son entre los atributos no primos y los primos.



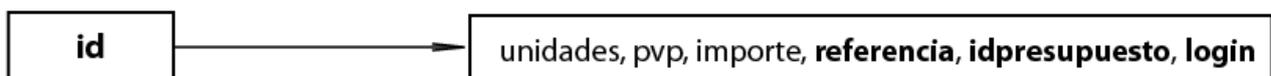
**Tabla *Articulo*:** esta tabla se encuentra en FNBC, ya que todos los atributos son atómicos, es decir, son valores únicos (FN1), y las únicas dependencias existentes son entre los atributos no primos y los primos.



**Tabla *Presupuesto*:** esta tabla se encuentra en FNBC, ya que todos los atributos son atómicos, es decir, son valores únicos (FN1), y las únicas dependencias existentes son entre los atributos no primos y los primos.



**Tabla *Detalle*:** esta tabla se encuentra en FNBC, ya que todos los atributos son atómicos, es decir, son valores únicos (FN1), y las únicas dependencias existentes son entre los atributos no primos y los primos.



Todas las relaciones se encuentran normalizadas en FNBC, por lo que no ha sido necesario realizar ninguna operación de normalización sobre las tablas definidas.

Por tanto, el esquema relacional puede definirse ahora en base a las tablas derivadas en la sección anterior:

**Tarifa** (idtarifa, valor)

**Usuario** (login, clave, nombre, **idtarifa**)

**Articulo** (referencia, tipo, descripcion, pvp, desperdicio)

**Presupuesto** (idpresupuesto, subtotal, tarifa, total, fecha, **login**)

**Detalle** (id, unidades, pvp, importe, **referencia**, **idpresupuesto**, **login**)

### 8.1.3. Definición sintáctica de las tablas

El esquema relacional se definirá ahora en base a las tablas derivadas en el apartado anterior, tablas normalizadas en FNBC, haciendo uso del lenguaje MySQL.

/\* Inicialmente se borran las tablas por si éstas han sido creadas con anterioridad. Las tablas deben ser borradas teniendo en cuenta las relaciones entre ellas. Como norma general es conveniente borrarlas en el orden inverso a como se crean\*/

```
DROP TABLE detalle;
DROP TABLE presupuesto;
DROP TABLE articulo;
DROP TABLE usuario;
DROP TABLE tarifa;
```

```
-----
--
-- Estructura de tabla para la tabla `tarifa`
--
```

```
CREATE TABLE `tarifa` (
  `idtarifa` int(11) NOT NULL,
  `valor` float NOT NULL,
  `estado` enum('0','1') NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-----  
  
--  
-- Estructura de tabla para la tabla `usuario`  
--  
  
CREATE TABLE `usuario` (  
  `login` varchar(20) NOT NULL,  
  `clave` varchar(40) NOT NULL,  
  `nombre` varchar(50) NOT NULL,  
  `idtarifa` int(11) NOT NULL,  
  `nivel` int(3) NOT NULL,  
  `estado` enum('0','1') NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
--  
-- Base de datos: `spw_bd`  
--  
  
-----  
  
--  
-- Estructura de tabla para la tabla `articulo`  
--  
  
CREATE TABLE `articulo` (  
  `referenciaArt` varchar(10) NOT NULL,  
  `tipo` varchar(10) NOT NULL,  
  `descripcion` varchar(50) NOT NULL,  
  `pvp` float NOT NULL,  
  `desperdicio` float NOT NULL,  
  `estado` enum('0','1') NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
-----  
  
--  
-- Estructura de tabla para la tabla `presupuestos`  
--  
  
CREATE TABLE `presupuestos` (  
  `idpresupuesto` int(11) NOT NULL,  
  `subtotal` float DEFAULT NULL,  
  `tarifa` float DEFAULT NULL,  
  `total` float DEFAULT NULL,  
  `fecha` date DEFAULT NULL,  
  `login` varchar(15) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-----  
  
--  
-- Estructura de tabla para la tabla `detalle`  
--  
  
CREATE TABLE `detalle` (  
  `id` int(11) NOT NULL,  
  `referenciaArt` varchar(8) DEFAULT NULL,  
  `idpresupuesto` int(11) DEFAULT NULL,  
  `unidades` int(11) DEFAULT NULL,  
  `pvp` float DEFAULT NULL,  
  `importe` float DEFAULT NULL,  
  `login` varchar(20) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
--  
-- Índices para tablas volcadas  
--  
  
--  
-- Indices de la tabla `tarifa`  
--  
ALTER TABLE `tarifa`  
  ADD PRIMARY KEY (`idtarifa`);  
  
--  
-- Indices de la tabla `usuario`  
--  
ALTER TABLE `usuario`  
  ADD PRIMARY KEY (`login`),  
  ADD KEY `idtarifa` (`idtarifa`);  
  
--  
-- Indices de la tabla `articulo`  
--  
ALTER TABLE `articulo`  
  ADD PRIMARY KEY (`referenciaArt`);  
  
--  
-- Indices de la tabla `presupuestos`  
--  
ALTER TABLE `presupuestos`  
  ADD PRIMARY KEY (`idpresupuesto`),  
  ADD KEY `fk_usuario_presupuesto_idx` (`login`);
```

```
--  
-- Indices de la tabla `detalle`  
--  
ALTER TABLE `detalle`  
  ADD PRIMARY KEY (`id`),  
  ADD KEY `fk_presupuesto` (`idpresupuesto`),  
  ADD KEY `fk_referencia` (`referenciaArt`);  
  
--  
-- AUTO_INCREMENT de las tablas volcadas  
--  
--  
-- AUTO_INCREMENT de la tabla `detalle`  
--  
ALTER TABLE `detalle`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=17;  
  
--  
-- AUTO_INCREMENT de la tabla `presupuestos`  
--  
ALTER TABLE `presupuestos`  
  MODIFY `idpresupuesto` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;  
  
--  
-- Restricciones para tablas volcadas  
--  
--  
-- Filtros para la tabla `usuario`  
--  
ALTER TABLE `usuario`  
  ADD CONSTRAINT `usuario_ibfk_1` FOREIGN KEY (`idtarifa`) REFERENCES `tarifa`  
  (`idtarifa`) ON DELETE CASCADE ON UPDATE CASCADE;  
  
--  
-- Filtros para la tabla `presupuestos`  
--  
ALTER TABLE `presupuestos`  
  ADD CONSTRAINT `fk_usuario_presupuesto` FOREIGN KEY (`login`) REFERENCES `usuario`  
  (`login`) ON DELETE NO ACTION ON UPDATE NO ACTION;
```

```

--
-- Filtros para la tabla `detalle`
--
ALTER TABLE `detalle`
  ADD CONSTRAINT `fk_presupuesto` FOREIGN KEY (`idpresupuesto`) REFERENCES
`presupuestos` (`idpresupuesto`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `fk_referencia` FOREIGN KEY (`referenciaArt`) REFERENCES `articulo`
(`referenciaArt`) ON DELETE NO ACTION ON UPDATE NO ACTION;

```

## 8.2. Diseño arquitectónico

El objetivo principal del diseño arquitectónico es desarrollar una estructura de programa modular y representar las relaciones de control entre los módulos. Además, el diseño arquitectónico mezcla la estructura del programa y la estructura de datos, y define las interfaces que facilitan el flujo de datos a lo largo del programa.

Son diversas las propuestas dentro de la Ingeniería del Software que facilitan el proceso de especificación en el ámbito arquitectónico del sistema. El común denominador de todas ellas es el desarrollo de Diagramas de Descomposición Funcional (DDF), los cuales serán de diferentes tipos atendiendo a los distintos niveles de abstracción que en ellos se apliquen y al detalle con que traten la información que representan.

Entre todas las propuestas anteriormente mencionadas, se ha optado por los Diagramas de Jackson, los cuales exponen una estructura modular jerarquizada.

El diagrama representado a continuación muestra la estructura general del programa.

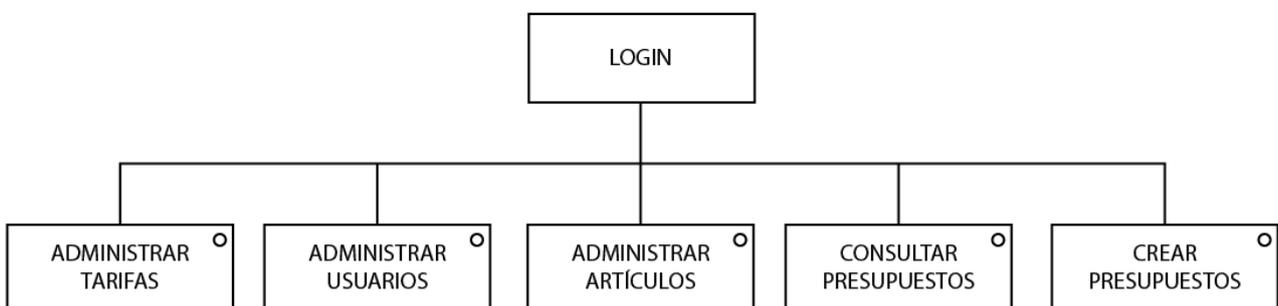


Figura 28: Diagrama de Descomposición general

El sistema desarrollado se ha diseñado de forma que fuera lo más modular posible. Así, tal y como se muestra en la anterior, se representa un módulo principal, denominado *LOGIN*, el cual corresponde a la página principal de la aplicación, la cual da acceso a los restantes módulos.

Los módulos dependientes del módulo principal corresponden a las distintas opciones con que se puede encontrar un determinado usuario.

En la siguiente figura se representa el diagrama de correspondiente al módulo *ADMINISTRAR USUARIOS*.

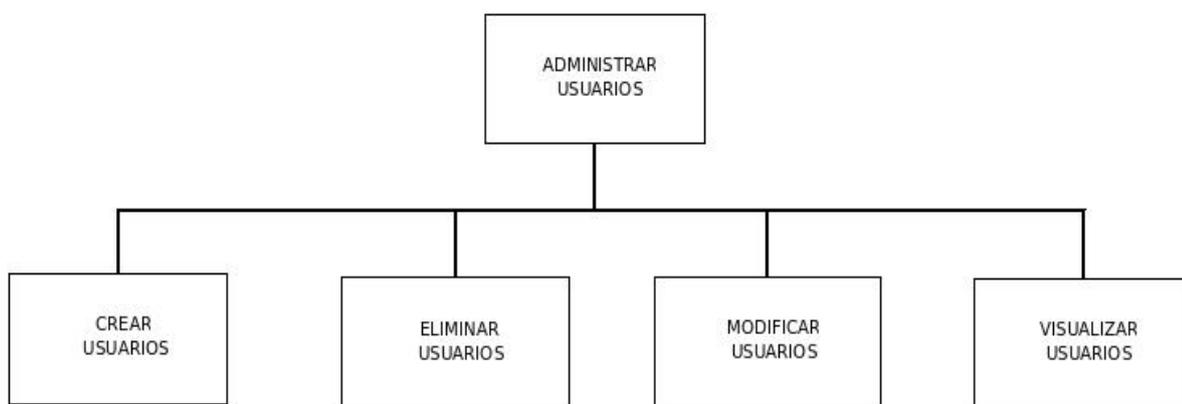


Figura 29: Diagrama de Descomposición de ADMINISTRAR USUARIOS

Este módulo está disponible sólo para el administrador del sistema. En él podrá crear nuevos usuarios, así como eliminarlos y modificar los datos de aquellos que ya estén registrados. Por último, podrá visualizar el listado total de usuarios registrados.

Los diagramas de descomposición para los módulos *ADMINISTRAR ARTÍCULOS*, *ADMINISTRAR TARIFAS* se van a omitir por ser iguales al diagrama de la figura anterior. En ellos se podrán realizar las mismas acciones explicadas en el anterior módulos, pero cada uno con sus correspondientes datos. A su vez, los diagramas de descomposición para los módulos *CONSULTAR PRESUPUESTOS* y *CREAR PRESUPUESTOS* también se van a omitir por tener más descomposición de lo representado en la figura anterior. La única diferencia es que al módulo *CREAR PRESUPUESTOS* van a acceder los usuarios registrados por el administrador. En él podrán realizar los presupuestos, modificar los datos de los artículos y borrar artículos de los presupuestos.

### 8.3. Diseño de la interfaz

En este apartado se presenta las características de las diferentes ventanas existentes en el sistema, utilizando para ello prototipos y capturas de pantalla.

La aplicación permite realizar múltiples acciones y es capaz de manejar cierta cantidad de información. Para que el usuario tenga constancia de lo que hace el programa en un momento determinado, éste deberá mostrar información en la pantalla. Para que el usuario pueda interactuar con la aplicación, éste debe proporcionar los mecanismos para que el usuario pueda introducir información, modificarla, borrarla o realizar acciones con la misma.

Para que todas estas posibilidades que ofrece el programa no resulten confusas, se debe diseñar una interfaz entendible y fácil de usar, en la que todas las ventanas sigan un mismo esquema y cuyos elementos se encuentren dispuestos de forma simple, clara y ordenada. Por todo esto, se creará una interfaz gráfica que facilite al usuario el aprendizaje y manejo de la aplicación

#### 8.3.1. Descripción de la interfaz

Como se ha mencionado anteriormente, todos los módulos de la aplicación seguirán una misma estructura para facilitar la comprensión y el manejo de la aplicación (a excepción de la pantalla inicial de acceso a la aplicación que estará formada por un formulario de logueo). Por ello es por lo que se va a mostrar un esquema del entorno de trabajo que los usuarios deberán conocer para moverse con celeridad por la aplicación.

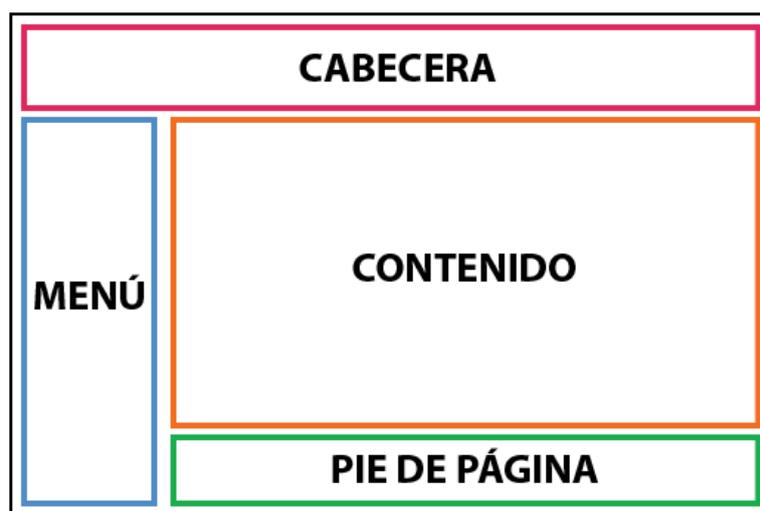


Figura 30: Esquema de interfaz

Este esquema está formado por las siguientes partes:

- **Cabecera:** en ella se encontrará, situado a la izquierda y con un tamaño de letra superior al resto, el nombre simplificado de la aplicación. A la derecha aparecerá la zona de sesión del usuario, en la que se verá su nombre y podrá cerrar su sesión para salir de la aplicación.
- **Contenido:** en este espacio estará alojado todo el grueso de la aplicación. Es decir, es el lugar donde aparecerán todos los formularios de datos y toda la información que se va a manejar del sistema.
- **Menú:** en esta parte estarán ubicados los distintos enlaces que permiten la navegación y acceso a todo el sistema.
- **Pie de página:** en la parte izquierda se encontrará la información general acerca de la aplicación.

### 8.3.2. Prototipo

El diseño preliminar es una actividad muy importante, ya que permite concretar todos y cada uno de los elementos que van a aparecer en las pantallas que componen el software.

Según el esquema anteriormente comentado conforme al cual se va a diseñar las ventanas, la pantalla inicial tendrá la siguiente estructura:

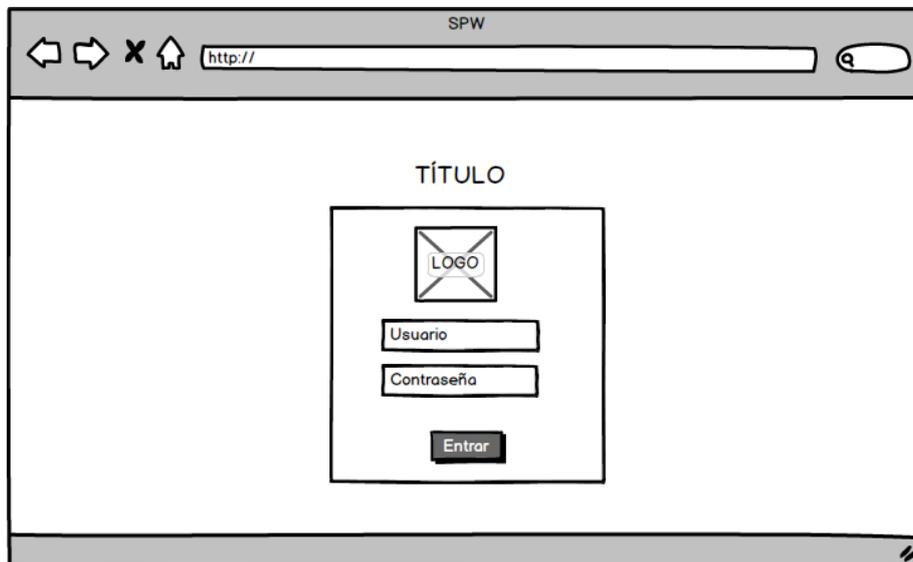


Figura 31: Prototipo de la página de acceso (Login)

Como se puede apreciar en la imagen, la pantalla inicial del portal contiene en la parte central el logotipo, el título de la aplicación y el formulario de logueo, en el que cualquiera de los dos tipos de usuario van a tener que introducir su nombre de usuario o login y su clave para acceder al sistema. Si hay un error

algún error en los datos introducidos, se mostrará un mensaje de validación y el usuario no podrá tener acceso hasta que no introduzca los nuevos datos.

Como se ha venido comentando hasta ahora, según el tipo de usuario logueado (administrador o cliente/usuario) el sistema mostrará una parte de la aplicación u otra.

### Parte Administración

Una vez establecidos los criterios para el diseño de la interfaz y la estructura básica de cada una de las pantallas, se puede pasar a su construcción.

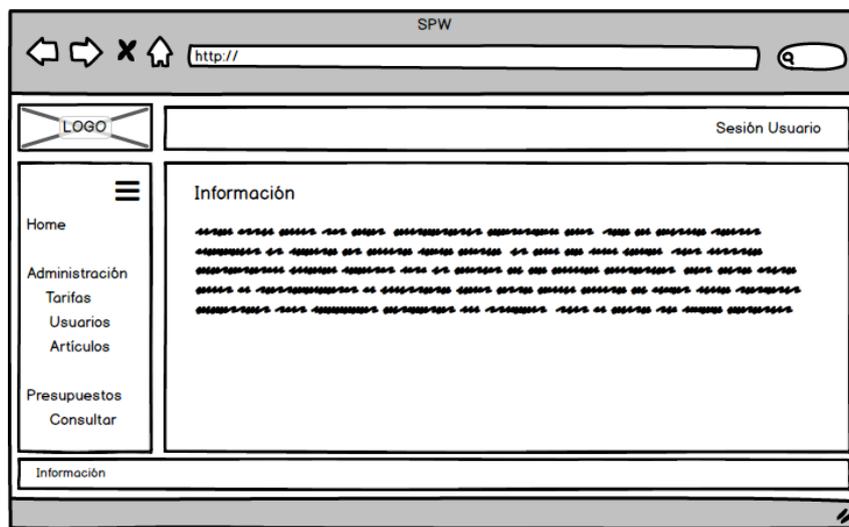


Figura 32: Prototipo de la página inicial (parte Administración).

Ésta es la primera pantalla con la que se encuentra el administrador. En la parte superior se encuentra su nombre de sesión y a donde podrá dirigirse para cerrar la sesión para salir del sistema. En la parte de la izquierda se encontrará los distintos menú por lo que podrá navegar para acceder a una opción u otra. Estas opciones serán: Administración de Tarifas (listado, creación, edición y eliminación de tarifas), Administración de Usuarios (listado, creación, edición y eliminación de usuarios), Administración de Artículos (listado, creación, edición y eliminación de artículos) y Administración de Presupuestos (Consulta de Presupuestos). También se tenía previsto, dentro de la administración de presupuestos que el administrador pudiera eliminar cualquiera de los presupuestos almacenados; así como hacer otro módulo de gestión de la base de datos, en la que se pudiera limpiar la base de datos (limpiar las tablas) y realizar copias de seguridad y restaurarlas. Pero por cuestiones de tiempo, no se han podido implementar.

En la parte central se mostrará información estática para el administrador. Esa era la idea inicial, pero para compensar los módulos no realizados, se ha decidido situar unos componentes que informan sobre el número de tarifas, usuarios, artículos y presupuestos almacenados en el sistema. A través de un enlace

situado en cada componente, el administrador podrá acceder a cada una de las administraciones de estos ítems.

Como los módulos de administración de tarifas, usuarios y artículos realizan las mismas acciones, se mostrará un prototipo genérico y común a todas ellas para las distintas tareas.

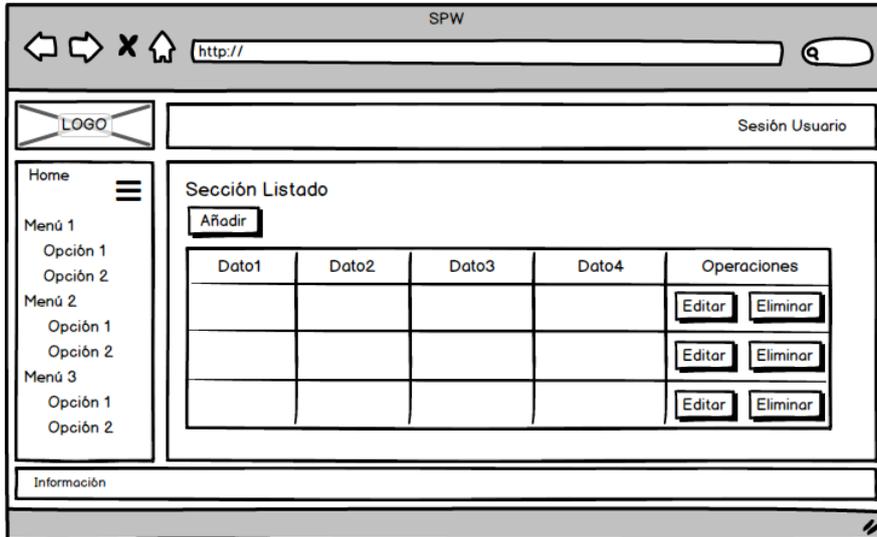


Figura 33: Prototipo de la página Listado (parte Administración).

A esta parte se llega al seleccionar cualquiera de las tres opciones de menú de Administración. El administrador podrá ver un listado de todos los registros almacenados en la base de datos del ítem seleccionado. A través de esta pantalla se puede hacer toda la gestión de un ítem: crear nuevos ítems, edítalos y eliminarlos.

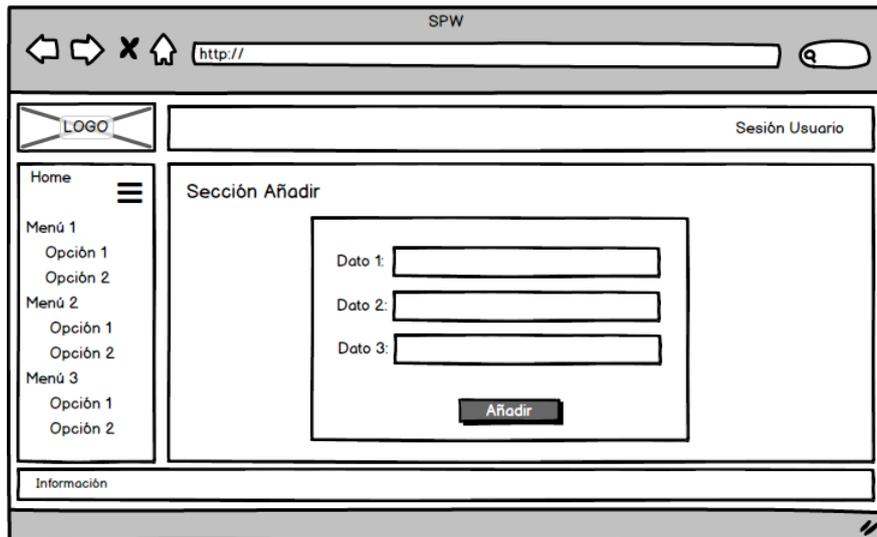


Figura 34: Prototipo de la página Añadir (parte Administración).

Este formulario aparece cuando el administrador pulsa el botón de Añadir situado en la parte superior del listado (figura 33). En él se podrán introducir todos los datos referentes al ítem seleccionado. En caso de que los datos introducidos no sean los esperados, se mostrará un mensaje informando de tal error.

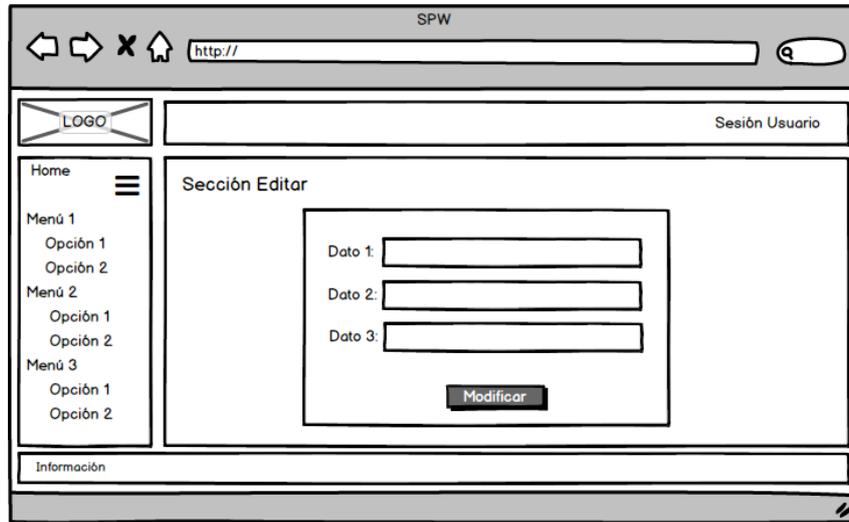


Figura 35: Prototipo de la página Editar (parte Administración).

Desde la página del listado y pulsando sobre el botón Editar de cada ítem del listado, el administrador podrá cambiar los datos erróneos del mismo registro. Una vez introducidos los nuevos datos, se deberá pulsar sobre el botón Modificar para guardar los cambios en la base de datos. El sistema regresará a la pantalla de listado y se mostrará el registro modificado. En caso de que haya algún error en los datos introducidos, se mostrará un mensaje informando de tal caso.

Desde la página del listado, también se podrá eliminar cualquier ítem que se desee. Simplemente basta con pulsar el botón Eliminar asociado a cada ítem. El registro se borrará automáticamente de la base de datos y de la pantalla, mostrándose de nuevo el listado sin dicho registro.

Para la parte de Administración de Presupuestos, el administrador podrá acceder a la opción de Consultas de Presupuestos.

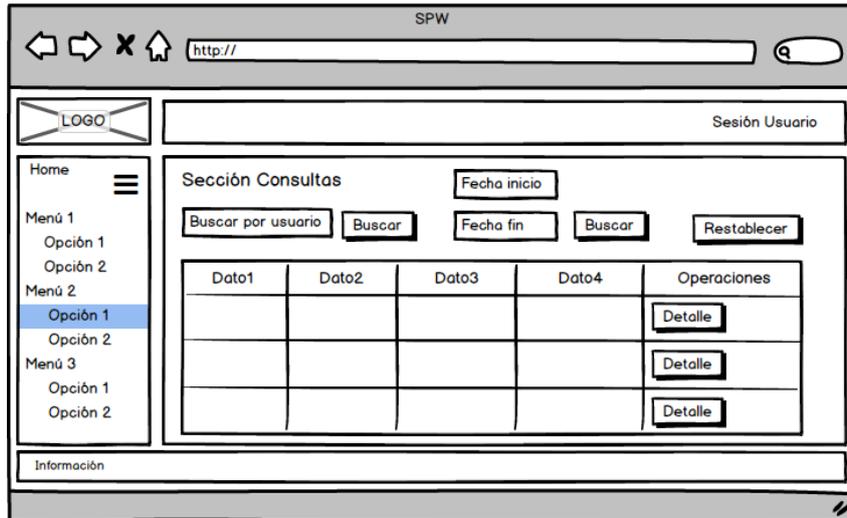


Figura 36: Prototipo de la página Consultas de Presupuestos (parte Administración).

Aquí, el administrador verá un listado resumen de los distintos presupuestos almacenados en la base de datos, mostrando sólo los datos más relevantes de cada presupuesto. Con ese listado se puede buscar los presupuestos requeridos en base a un criterio de búsqueda: por fechas. Introduciendo una fecha de inicio y una fecha de fin, se listarán los presupuestos realizados entre dicho rango de fechas. Para volver al listado general, se deberá pulsar el botón Restablecer.

Además, por cada resumen de presupuesto se puede ver su detalle, es decir, toda la información del presupuesto y con una apariencia más atractiva y comercial. Para ello, se deberá pulsar el botón Detalle y se abrirá una ventana modal como la de la siguiente figura.

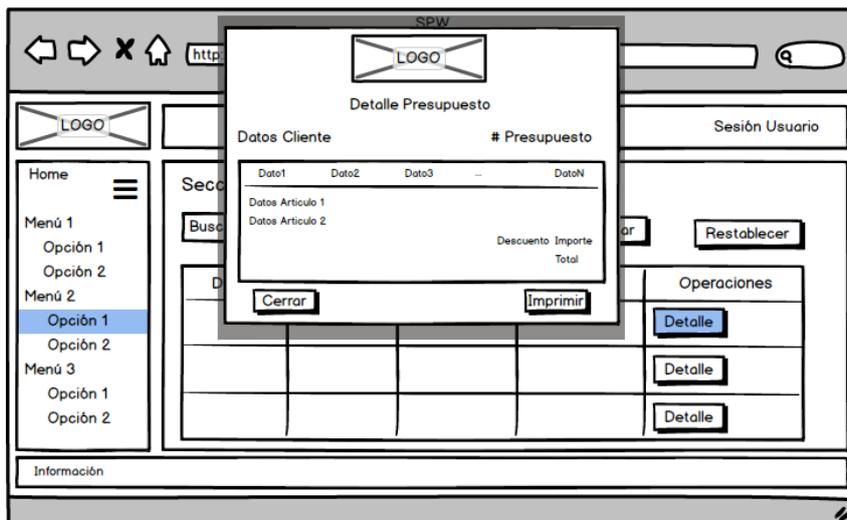


Figura 37: Prototipo de la página Detalle de Presupuestos (parte Administración).

Esa ventana modal ofrecerá la posibilidad de imprimir (en impresora, si se tiene configurada una, o en formato PDF) el presupuesto.

En un principio se tenía pensado implementar la búsqueda de presupuestos por el usuario que los creó, pero por tiempo no se ha podido realizar. Pero contrarrestando esto, se ha implementado la funcionalidad de exportar el listado resumen de presupuestos a formato Excel y PDF, que en un principio no se tenía contemplado realizarlo.

Hasta aquí es el subsistema que va a manejar el administrador del sistema. A continuación se comentará el subsistema de Creación de Presupuestos, destinado al usuario cliente.

### Parte Creación de Presupuestos

Después de que el usuario cliente se loguee con éxito, el sistema lo dirigirá a la siguiente pantalla:

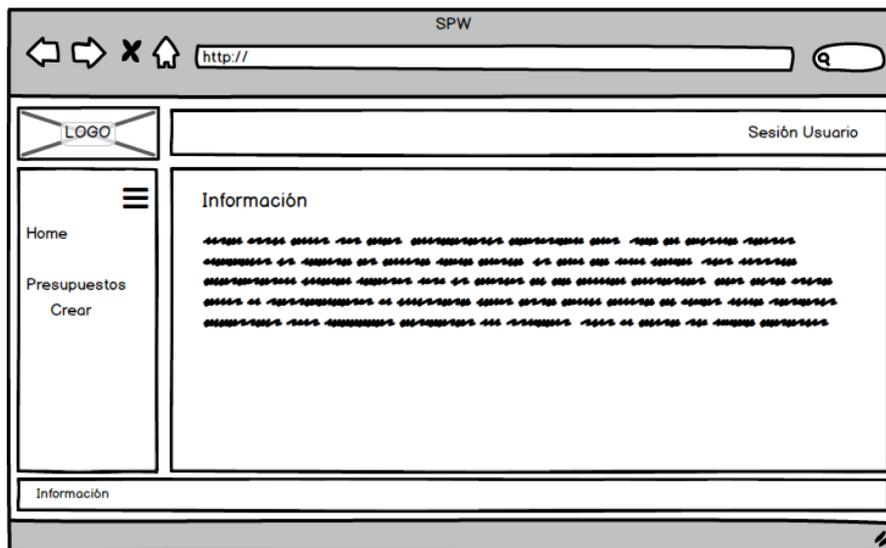


Figura 38: Prototipo de la página inicial (parte Creación de Presupuestos).

En esta página sólo se mostrará información estática. El usuario podrá comenzar a crear su presupuesto a través de la opción correspondiente situada en el menú de la parte izquierda.

Para la creación del presupuestos, el usuario deberá introducir la referencia del artículo que quiere incluir. Esta búsqueda se facilitará con un autocompletado, en el que aparecerán las referencias coincidentes a lo que el usuario va introduciendo. Una vez encontrado el artículo deseado, se deberá pulsar el botón Añadir para incluirlo en el presupuesto. A la vez que se carga la referencia del artículo, se mostrarán los demás datos del mismo necesarios para el presupuesto. El usuario tendrá que introducir el número de unidades de cada artículo añadido al presupuesto. Dinámicamente, se van calculando los valores necesarios para el cómputo total del presupuesto. El usuario podrá cambiar las unidades cuando desee. Además podrá eliminar cualquier artículo añadido, pulsando el botón eliminar asociado a cada artículo.

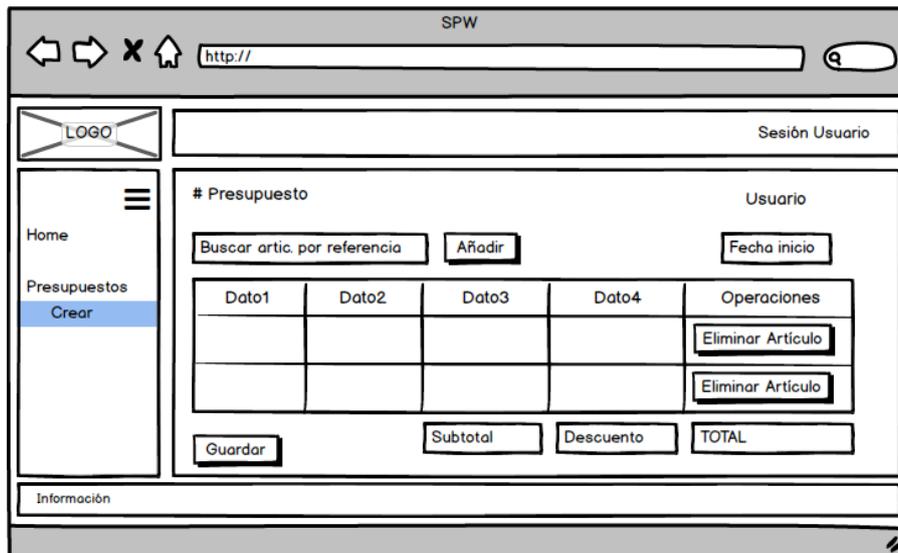


Figura 39: Prototipo de la página Crear Presupuesto (parte Creación de Presupuestos).

Se pretendía que, cuando el usuario le diera al botón Guardar para terminar el presupuesto, se generara el presupuesto en formato PDF, pero no se ha podido implementar. Una vez guardado el presupuesto, el sistema volverá a mostrar un presupuesto vacío por si se quiere realizar otro.

### 8.3.3. Diseño final

Una vez establecidos los criterios para el diseño de la interfaz y la estructura básica de cada una de las pantallas, se puede pasar a su construcción.

A continuación se muestran los diseños definitivos de los prototipos comentados anteriormente.

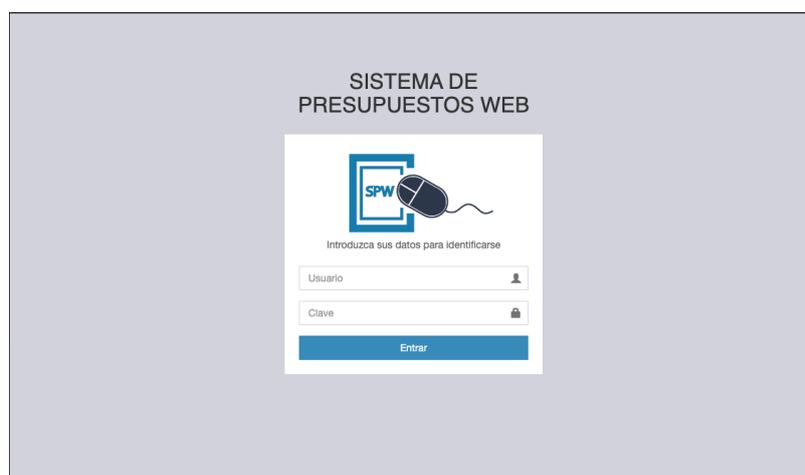


Figura 40: Página de acceso (Login)

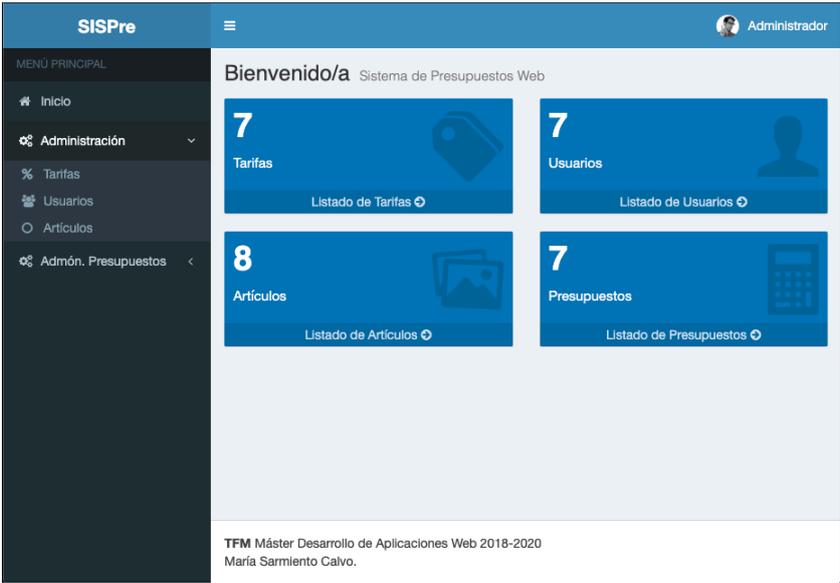


Figura 41: Página inicial (parte Administración).

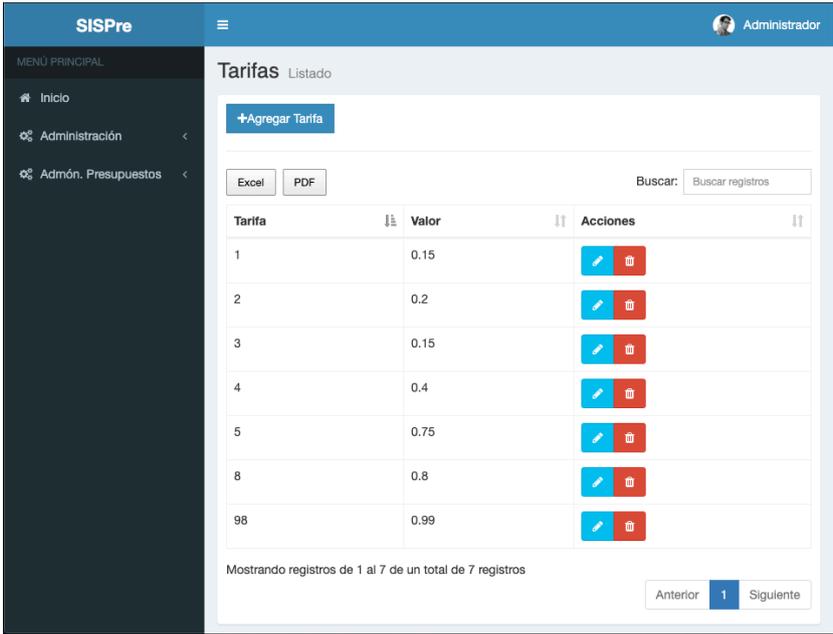


Figura 42: Página Listado (parte Administración).

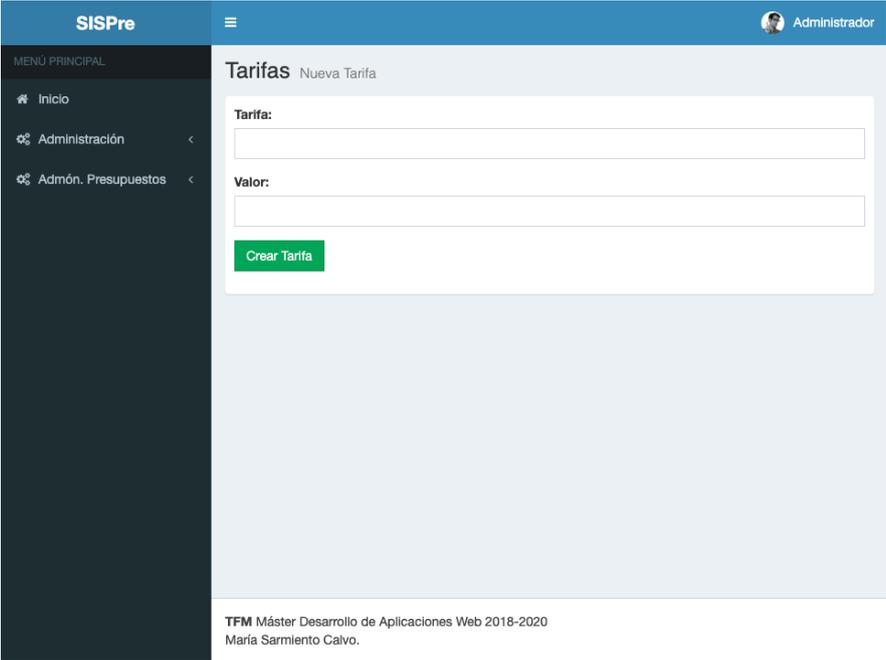


Figura 43: Página Añadir Tarifa (parte Administración).

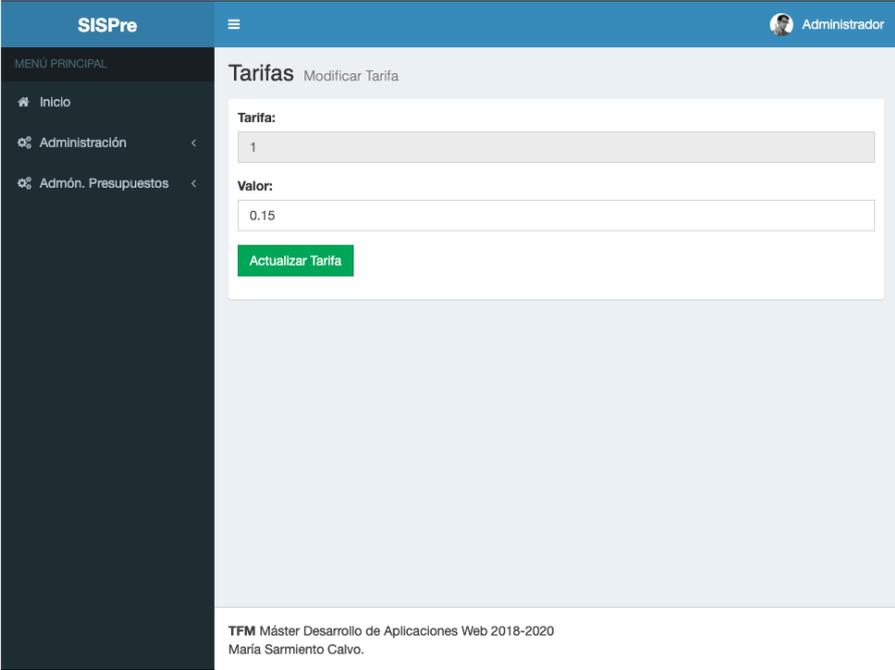


Figura 43: Página Editar Tarifa (parte Administración).

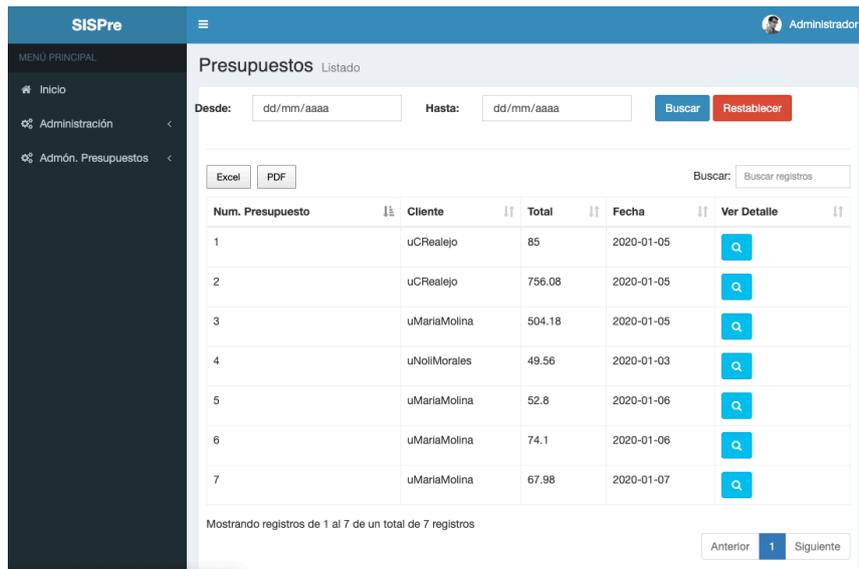


Figura 44: Página Consultas de Presupuestos (parte Administración).

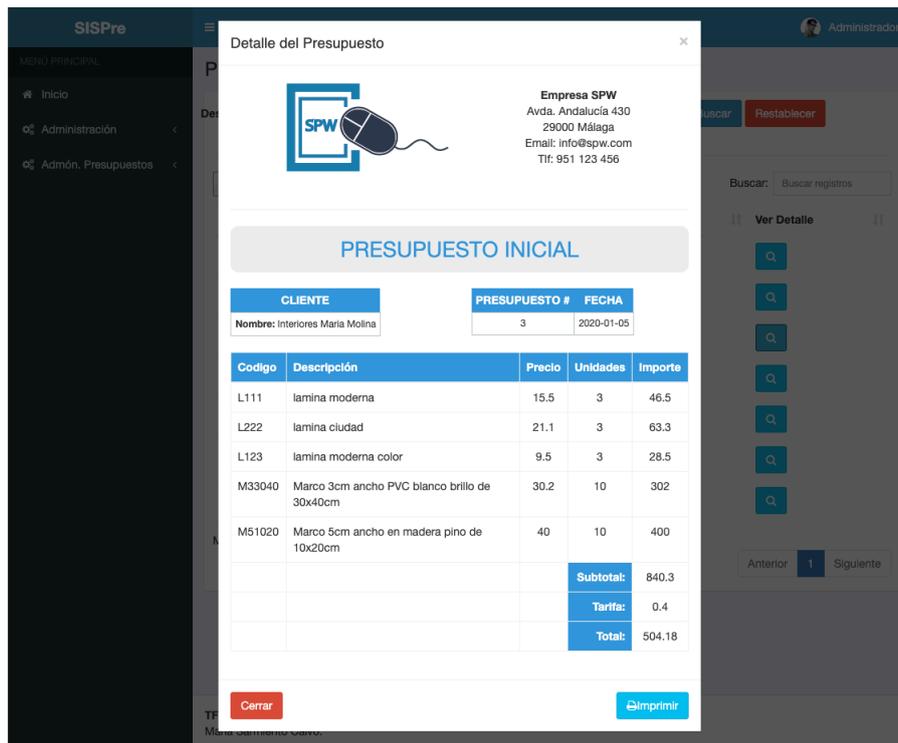


Figura 45: Prototipo de la página Detalle de Presupuestos (parte Administración).

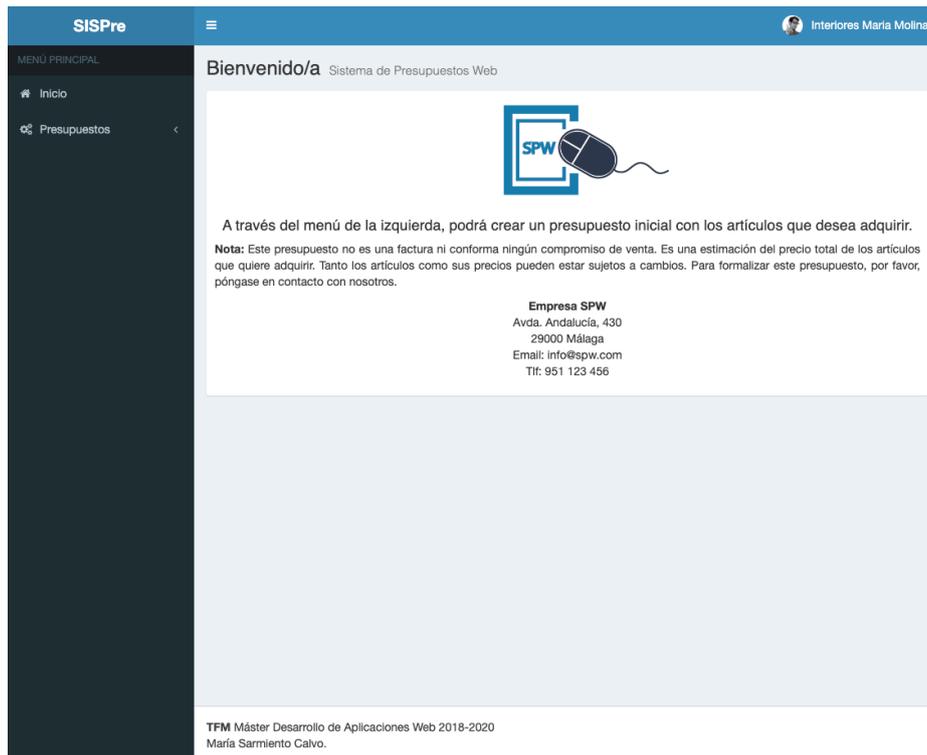


Figura 46: Página Detalle de Presupuestos (parte Creación de Presupuestos).

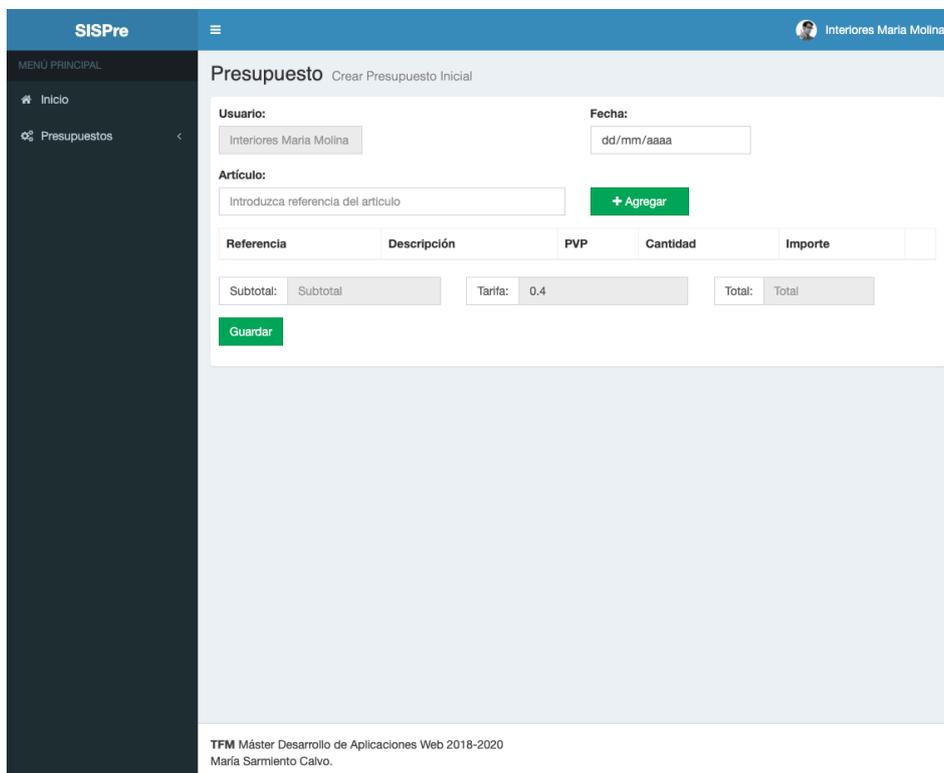


Figura 47: Página Crear Presupuesto (parte Creación de Presupuestos).

## 9. Pruebas

Las pruebas aseguran que el software desarrollado tenga una calidad mínima, puesto que permiten descubrir errores en la aplicación antes de que ésta entre en funcionamiento.

Seguidamente se describirán algunas de las pruebas a las que ha sido sometida la aplicación, describiendo los problemas encontrados, si los hubo, y las soluciones aportadas para resolver los mismos.

### 9.1. Casos de prueba

#### 9.1.1. Pruebas de las herramientas de desarrollo

Antes de comenzar a desarrollar cualquier aplicación es necesario conocer las herramientas con las que se va a trabajar. Por este motivo, se comprueba el funcionamiento de cada una de esas herramientas.

- Prueba realizada: **Inicio del servidor web Apache.**
  - *Resultados esperados:* su correcto funcionamiento.
  - *Problemas encontrados:* ninguno.
  - *Soluciones adoptadas:* ninguna.
  
- Prueba realizada: **Conexión con la base de datos**
  - *Resultados esperados:* su correcto funcionamiento.
  - *Problemas encontrados:* ninguno.
  - *Soluciones adoptadas:* ninguna.

#### 9.1.2. Pruebas de funcionamiento de la aplicación

Cabe mencionar que a lo largo de la fase de desarrollo se ha ido realizando pruebas para comprobar el correcto funcionamiento de la aplicación. Conforme surgían los errores, tanto a nivel de estructura interna como en la salida de los datos proporcionados por el sistema, se iba corrigiendo y así poder proseguir en el desarrollo. Algunas de las pruebas de funcionamiento realizadas a la aplicación son las siguientes:

- Prueba realizada: **Características en los formularios de creación de usuarios y tarifas**
  - *Resultados esperados*: su correcto funcionamiento.
  - *Problemas encontrados*: ninguno.
  - *Soluciones adoptadas*: ninguna.
  
- Prueba realizada: **Introducción de datos erróneos en formularios**
  - *Resultados esperados*: deberá detectar el error y mostrar un mensaje alertando del mismo e indicando al usuario que lo corrija.
  - *Problemas encontrados*: algunos campos de texto no tenían codificada bien la validación en cuanto al tipo de dato que se debía de introducir.
  - *Soluciones adoptadas*: se ha introducido validaciones más concretas en cuanto a tipología de dato.
  
- Prueba realizada: **Comprobación del diseño responsive**
  - *Resultados esperados*: deberá ajustar el diseño de la aplicación a los distintos tamaños de pantalla.
  - *Problemas encontrados*: ninguno.
  - *Soluciones adoptadas*: ninguna.
  
- Prueba realizada: **Examinar el funcionamiento de la aplicación, utilizando los navegadores web Firefox y Safari.**
  - *Resultados esperados*: su correcto funcionamiento.
  - *Problemas encontrados*: ninguno.
  - *Soluciones adoptadas*: ninguna.

## 10. Futuras mejoras

En cuanto a las posibles futuras mejoras que se pueden realizar sobre el proyecto, se consideran las siguientes:

- Añadir una nueva y más completa forma de presupuestar los artículos, según se vendan por unidades, por perímetro o metros lineales, o por superficie o metros cuadrados; así como los nuevos cálculos propios de ellos para su facturación.
- Realizar una mayor monitorización de los usuarios, como reflejar la fecha y hora de la conexión de los mismos y contabilizar los usuarios que están en línea en cada instante.
- Exportar los presupuestos a ficheros en formato PDF.
- Ampliar la administración de los presupuestos para que el administrador pueda eliminar presupuestos del sistema.
- Implementar un módulo para la gestión de la base de datos (copias de seguridad, restauración, etc.).

# 11. Conclusiones

Una vez finalizado el desarrollo del proyecto, se puede extraer las siguientes conclusiones del mismo:

- Se ha desarrollado un sistema software que permite elaborar presupuestos, a través de Internet, de una serie de productos que ofrece una empresa.
- Los productos pueden actualizarse desde la sede central de la empresa, mejorando sensiblemente la actualización de las tarifas de precios y disminuyendo el tiempo que tiene que dedicar la empresa a la actividad presupuestaria, derivando esta tarea a cada uno de los clientes interesados.
- No se ha realizado el proyecto como se definió originariamente, teniendo que omitir ciertos módulos debido a la restricción de tiempo impuesta y a la restricción propia del proceso de aprendizaje; pero ha habido partes que se han mejorado con respecto a la idea inicial.

Como conclusión personal se puede destacar que el aprendizaje de las tecnologías usadas, principalmente CodeIgniter, ha sido un reto, ya que esta materia se ha visto de una forma básica durante la asignatura Desarrollo Back-End con PHP (aspecto lógico debido a la propia temporalidad del máster) y que estoy satisfecha con lo aprendido y el resultado del proyecto. Además, esta formación tiene un extra añadido porque tiene una gran utilidad profesional.

## Anexo 1. Librerías externas utilizadas

Para el desarrollo del proyecto se han usado las siguientes librerías, plugins y/o utilidades:

- **Plantilla AdminLTE** basada en Bootstrap. Web: <https://github.com/ColorlibHQ/AdminLTE>
- **Iconos Ionicons** para usar iconos gratuitos. Web: <https://ionicons.com/>
- **jQuery UI**, para las interacciones de la interfaz de usuario. Web: <https://jqueryui.com/>
- **jQuery Print Plugin**, plugin para la impresión de páginas. Web: <https://github.com/DoersGuild/jquery.print>
- 
- **Datatables**, para la paginación y contabilización y listado de registros. Web: <https://datatables.net/>
- **DataTables Export**, plugin para la exportación de ficheros tipo Excel y PDF de las tablas listado. Web: <https://datatables.net/extensions/buttons/examples/initialisation/export>

## Anexo 2. Bibliografía

(Enero de 2020). Obtenido de HTML5 Introduction - W3Schools:

[https://www.w3schools.com/html/html5\\_intro.asp](https://www.w3schools.com/html/html5_intro.asp)

(Enero de 2020). Obtenido de CSS Tutorial - W3Schools: <https://www.w3schools.com/css/>

(Enero de 2020). Obtenido de jQuery: <https://jquery.com/>

(Enero de 2020). Obtenido de CodeIgniter: <https://codeigniter.com/>

(Enero de 2020). Obtenido de MySQL: <https://www.mysql.com/>

(Enero de 2020). Obtenido de World Wide Web Consortium (W3C): <https://www.w3.org/>

(Enero de 2020). Obtenido de RIB Spain: <https://www.rib-software.es/>

(Enero de 2020). Obtenido de Eneboo Open Source ERP: <https://www.eneboo.org/>

(Enero de 2020). Obtenido de Sage FacturaPlus: <http://www.sage.com>

*ndreessen*. Retrieved from Wired: [http://www.wired.com/epicenter/2012/04/ff\\_andreessen/](http://www.wired.com/epicenter/2012/04/ff_andreessen/)

Luque Ruiz, I. (2001). *Bases de Datos. Desde Chen hasta Codd con Oracle*. Ra-Ma.

