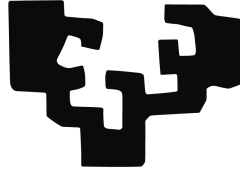


eman ta zabal zazu



Universidad      Euskal Herriko  
del País Vasco      Unibertsitatea

Departamento de Lenguajes y Sistemas Informáticos  
Escuela Universitaria de Ingeniería de Vitoria-Gasteiz

# KANALVET

## SERVICIO WEB E-COMMERCE

Memoria de desarrollo

presentada para optar al título de Ingeniero Técnico en Informática de

Gestión por

**Mikel Bravo Rodríguez**

Director

Pablo González Nalda



*A mis padres*





---

# Índice general

<b>Índice general</b>	<b>III</b>
<b>Índice de figuras</b>	<b>VII</b>
<b>Índice de tablas</b>	<b>IX</b>
<b>Resumen y organización de la memoria</b>	<b>XI</b>
Resumen . . . . .	XI
Organización . . . . .	XI
<b>I Alcance del Proyecto</b>	<b>1</b>
<b>1 Descripción y objetivos del proyecto</b>	<b>3</b>
1.1. Descripción del proyecto . . . . .	3
1.2. Objetivos del proyecto . . . . .	5
1.3. Cómo surgió la idea . . . . .	5
<b>2 Viabilidad</b>	<b>7</b>
2.1. Análisis de riesgos . . . . .	7
2.2. Planificación del tiempo del proyecto . . . . .	10
2.2.1. Estructura de descomposición del trabajo . . . . .	10
2.2.2. Fases, tareas y entregables . . . . .	13
2.2.2.1. Listado de fases tareas y entregables . . . . .	13
2.2.2.2. Descripción detallada de las tareas y entregables . . . . .	14
2.2.3. Recursos humanos y materiales . . . . .	20
2.2.4. Asignación de recursos y agenda del proyecto . . . . .	20
2.2.4.1. Planificación . . . . .	22
2.2.4.2. Diagrama de Gantt . . . . .	25
2.3. Estimación de costes del proyecto . . . . .	27
2.3.1. Recursos de trabajo . . . . .	27

2.3.2.	Recursos materiales . . . . .	27
2.3.3.	Costo de recursos de trabajo . . . . .	27
2.3.4.	Costo de recursos materiales . . . . .	28
2.3.5.	Amortizaciones . . . . .	29
2.3.6.	Presupuesto . . . . .	30
<b>II Conceptos generales, tecnologías y herramientas</b>		<b>31</b>
<b>3</b>	<b>Conceptos generales</b>	<b>33</b>
3.1.	Modelo / Vista / Controlador . . . . .	33
3.1.1.	Definición de las partes . . . . .	34
3.1.1.1.	El Modelo . . . . .	34
3.1.1.2.	La Vista . . . . .	34
3.1.1.3.	El Controlador . . . . .	35
3.1.2.	MVC en Kanalvet . . . . .	36
3.2.	Criptografía . . . . .	36
3.2.1.	Criptografía simétrica . . . . .	36
3.2.1.1.	Algoritmo RSA . . . . .	36
<b>4</b>	<b>Desarrollo técnico</b>	<b>39</b>
4.1.	Estudio de la tecnología . . . . .	39
4.1.1.	HTML . . . . .	39
4.1.2.	CSS . . . . .	40
4.1.3.	JavaScript . . . . .	40
4.1.3.1.	¿Qué es Javascript? . . . . .	41
4.1.4.	JQuery . . . . .	41
4.1.4.1.	¿Por qué JQuery? . . . . .	42
4.1.5.	PHP . . . . .	42
4.1.5.1.	¿Por qué PHP? . . . . .	43
4.1.6.	SQL . . . . .	43
4.1.6.1.	¿Por qué SQL? . . . . .	44
4.1.7.	WordPress . . . . .	44
4.1.7.1.	¿Por qué WordPress? . . . . .	45
4.1.8.	L <sup>A</sup> T <sub>E</sub> X . . . . .	45
4.1.8.1.	¿Por qué L <sup>A</sup> T <sub>E</sub> X? . . . . .	46
4.2.	Herramientas . . . . .	46
4.2.1.	Eclipse . . . . .	46
4.2.2.	Gimp 2.6 . . . . .	47
4.2.3.	Microsoft Visio 2010 . . . . .	47
4.2.4.	Microsoft Project 2010 . . . . .	48
4.2.5.	Sublime Text . . . . .	48

<b>III</b>	<b>La aplicación Kanalvet: diseño, estructura y gestión</b>	<b>51</b>
<b>5</b>	<b>Diseño y estructura</b>	<b>53</b>
5.1.	Hito 1: Diseño de la arquitectura de la aplicación . . . . .	53
5.1.1.	Descripción general . . . . .	53
5.2.	Hito 2: Diseño e implementación de un algoritmo de actualización de datos de la aplicación. . . . .	56
5.2.1.	Descripción General . . . . .	56
5.2.2.	Diseño Base de Datos . . . . .	56
5.2.3.	Diseño del algoritmo de carga de datos . . . . .	57
5.2.4.	Resumen . . . . .	59
5.3.	Hito 3: Función para navegar entre productos . . . . .	60
5.3.1.	Estructura y opciones . . . . .	60
5.4.	Hito 4: Pasarela de pago . . . . .	62
5.4.1.	Pasarela de pago con PayPal . . . . .	62
5.4.1.1.	Solución problema 1: Seguridad . . . . .	64
5.4.1.2.	Solución problema 2: Automatización . . . . .	64
5.4.1.3.	Resumen . . . . .	65
<b>6</b>	<b>Gestión del proyecto</b>	<b>67</b>
6.1.	Retrasos . . . . .	67
6.2.	Horas Planificadas vs Horas reales . . . . .	68
6.3.	Costes planificados vs Costes reales . . . . .	69
<b>IV</b>	<b>Conclusiones y trabajo futuro</b>	<b>71</b>
<b>7</b>	<b>Conclusiones y líneas futuras</b>	<b>73</b>
7.1.	Conclusiones . . . . .	73
7.2.	Líneas futuras . . . . .	74
<b>V</b>	<b>Apéndices y bibliografía</b>	<b>75</b>
<b>A</b>	<b>Pliego de condiciones</b>	<b>77</b>
A.1.	Condiciones del cliente . . . . .	77
A.2.	Condiciones del servidor y otros elementos hardware . . . . .	77
A.3.	Condiciones del software . . . . .	78
A.4.	Condiciones generales . . . . .	78
	<b>Bibliografía</b>	<b>79</b>







---

## Índice de figuras

2.1. Estructura de descomposición del trabajo I. . . . .	11
2.2. Estructura de descomposición del trabajo II. . . . .	12
2.3. Diagrama de Gantt . . . . .	26
3.1. Diagrama MVC . . . . .	35
5.1. Estructura Kanalvet . . . . .	54
5.2. Estructura control . . . . .	55
5.3. Entidad relación . . . . .	57
5.4. Ejemplo estructura árbol . . . . .	60






---

## Índice de tablas

2.1. Calendario laboral . . . . .	21
2.2. Listado de fases tareas y entregables I. . . . .	22
2.3. Listado de fases tareas y entregables II. . . . .	23
2.4. Asignación de recursos a tareas I. . . . .	24
2.5. Asignación de recursos a tareas II. . . . .	25
2.6. Recursos de trabajo . . . . .	27
2.7. Recursos materiales (Hardware) . . . . .	28
2.8. Recursos materiales (Software). . . . .	28
2.9. Costo de recursos de trabajo . . . . .	28
2.10. Costo de recursos materiales . . . . .	28
2.11. Amortizaciones . . . . .	29
2.12. Presupuesto . . . . .	30
5.1. Ejemplo Excel datos . . . . .	57
6.1. Horas planificadas vs horas reales . . . . .	68
6.2. Amortizaciones reales . . . . .	69
6.3. Coste real de los recursos de trabajo. . . . .	70
6.4. Costes planificados vs costes reales . . . . .	70





---

# Resumen y organización de la memoria

## Resumen

Este trabajo diseña y construye una aplicación web para una empresa cuya intención es expandir su negocio de venta de productos para clínicas veterinarias a través de internet, buscando captar clientes en el ámbito nacional.

Al ser un proyecto desarrollado en empresa, las funcionalidades que debe tener esta aplicación se ven dadas por el cliente que pide una web fácil y entendible para clientes que accedan a la web con intención de comprar, pero también pide un módulo de administrador sencillo para poder gestionar la web sin ayuda externa y sin tener grandes conocimientos sobre aplicaciones web. Por tener un módulo de administrador la aplicación se divide dos grandes apartados, el de administrador y el de usuario.

La aplicación se basa en el gestor de contenidos WordPress y aunque se utilizan algunas de las funcionalidades que da este gestor de contenidos, éste solo se usa como base que soporta las funcionalidades creadas para la aplicación web. Las funcionalidades creadas para realizar la aplicación se han desarrollado usando PHP y se ejecutan mediante un plugin de WordPress. Por ello se entiende que cualquier servidor que soporte WordPress es capaz de soportar la aplicación, dando así mayor flexibilidad a ésta.

Para desarrollar la aplicación se han tenido en cuenta las normas de accesibilidad y desarrollo web dados por la W3 además de usar técnicas de protección de datos para los usuarios registrados.

## Organización

La primera parte de esta memoria la forma el capítulo en el que se presenta el alcance del proyecto, donde se hace una descripción del mismo, se enuncian los objetivos que debe cumplir, se representa detalladamente el estudio de viabilidad realizado, riesgos, etc.

La segunda parte plantea el desarrollo técnico de la aplicación, enunciando el estudio de la tecnología realizado, y las herramientas utilizadas para poder desarrollarlo.

La tercera parte describe la aplicación Kanalvet, desarrollada en este trabajo y que expone el diseño y la estructura utilizada. Asimismo, se evalúa la gestión del proyecto mediante los retrasos que ha podido sufrir, el coste real que ha supuesto, etc. En la cuarta parte, se explica el modo de uso de la aplicación mediante su manual de usuario.

Por último, en la quinta parte se elaboran las conclusiones y se plantea el trabajo futuro.

Como información adicional, se incluyen en los anexos, el pliego de condiciones, archivos de configuración y la licencia del documento.

# **Parte I**

## **Alcance del Proyecto**





---

# Descripción y objetivos del proyecto

Este capítulo presenta una breve descripción del proyecto resaltando sus funcionalidades básicas, así como los objetivos principales de la aplicación, de dónde surgió la idea, desarrollos similares y un análisis de riesgos que especifica los posibles problemas que pueden surgir en la realización de la aplicación.

## 1.1. Descripción del proyecto

Kanalvet es una aplicación web desarrollada para una empresa de mismo nombre que pretende ampliar su mercado y empezar a vender sus productos vía web. Con ese propósito Kanalvet se ha construido de tal forma que muestre el amplio catálogo de productos del que dispone la empresa con imágenes, descripciones, especificaciones... para que los usuarios que visiten la página puedan acceder de una forma sencilla y clara a toda la información de los productos.

Un área importante de la web es la parte del usuario. Ya que Kanalvet es una aplicación web orientada a la venta online, necesita tener la opción de registro para los clientes que deseen comprar. Los usuarios registrados tienen la opción del carro de la compra donde pueden almacenar los diferentes productos que quieran comprar. Una vez hecha la compra ésta pasa a un apartado especial para usuarios registrados, donde se indican todos los pedidos que ha realizado ese usuario y sus estados.

Por otra parte tenemos la parte del administrador. Este módulo ha sido desarrollado de tal manera que el cliente que nos encargó la aplicación fuera capaz de gestionarla de una manera sencilla. Para ese motivo se crearon 3 módulos principales:

1. Gestión de productos: al ser una web de venta de productos para clínicas veterinarias, el cliente tenía la necesidad de poder insertar/ modificar/ eliminar cualquier producto en cualquier momento. Aprovechando que el cliente usaba un archivo con formato de hoja de cálculo excel para la gestión interna de productos, se optó por crear una funcionalidad que a partir de una hoja de cálculo excel pudiera insertar / borrar / eliminar productos de la aplicación web, de esta manera se evitan tener datos duplicados haciéndolo fácil para el cliente. Ésta funcionalidad consigue automatizar la carga de productos a la aplicación de forma que la gestión de la aplicación se vuelve muy sencilla.
2. Gestión de usuarios: este apartado se creó a petición del cliente ya que como el precio de los productos debía ser accesible solo por usuarios registrados, también quería tener la opción de validar los usuarios recién registrados el mismo, por lo que un usuario no está activo hasta que el administrador le da el visto bueno. Un usuario registrado pero no activo, no puede acceder a información detallada de los productos, ni hacer consultas sobre los productos a través de los formularios web. Para poder validar los usuarios se genera una tabla donde aparece la información básica que el usuario ha proporcionado al registrarse, de modo que el administrador después de comprobar los datos de los usuarios es capaz de validarlos con un solo “click”. En este módulo también se le da la opción de desactivar usuarios.
3. Gestión de pedidos: desde este módulo el administrador es capaz de ver una lista con todos los pedidos que han realizado los usuarios registrados a través de la aplicación. Cada pedido de la lista muestra el usuario que lo ha realizado, el valor total del pedido y el estado del pedido. El estado del pedido se puede modificar desde este mismo módulo para poder ir informando al usuario del estado de su pedido.

Estos módulos de administrador se han sido creados para la gestión de los datos de la aplicación (usuarios, productos, pedidos) pero para la gestión interna de la aplicación se ha usado el “backend” que nos ofrece WordPress. La administración de este “backend” no es tarea del responsable de Kanalvet, si no que ha sido utilizado únicamente por el desarrollador de la aplicación, de esta forma la interfaz para la gestión de los datos se reduce y se vuelve más sencilla.

En conclusión Kanalvet es una aplicación web que permite ver y comprar fácilmente los productos que ofrece la empresa y que además está desarrollada de tal forma que

el administrador pueda gestionarla sin ninguna dificultad.

## 1.2. Objetivos del proyecto

El objetivo de esta aplicación web es la de crear una aplicación web sencilla de gestionar y visualmente atractiva, con la cual la empresa pueda expandir su negocio vía web captando nuevos clientes dentro del ámbito nacional.

El objetivo como diseñador y constructor de la aplicación es el de satisfacer las necesidades del cliente, intentando siempre priorizar la sencillez y estética de la misma con tal de conseguir una web fácil de usar por los usuarios y sobre todo de fácil gestión para el cliente que no quiere tener dedicar excesivo tiempo en configurar la aplicación, ya que busca una herramienta que le ayude a expandir su negocio y no una nueva responsabilidad a la hora de llevar la empresa.

## 1.3. Cómo surgió la idea

La idea de crear esta aplicación me fue dada por la empresa solicitante que buscaba de una herramienta capaz de abrir las puertas de su negocio a un público más amplio.

Una vez con la aplicación propuesta la aplicación, el módulo al que más importancia le daba la empresa era el módulo de gestión de productos. La idea de implementar este módulo en base al archivo con formato de hoja de cálculo excel que usaba la empresa surgió cuando se vio que el archivo que usaba estaba realmente bien estructurado y con todo lo necesario para poder publicar en la aplicación. Además la propuesta de esa implementación fue muy bien aceptada por la empresa ya que la cantidad de productos que ofrece supera los mil, organizados en más de quinientas familias y sub-familias, por lo que tener datos duplicados para la gestión interna de productos y para datos almacenados en la aplicación podría ser caótico.

Muchas de otras ideas usadas para el desarrollo de la aplicación como pueden ser el estilo o las opciones ofrecidas a los usuarios vienen dadas por la empresa que tenía bastante claro que es lo que buscaba en esta aplicación. Parte de los retrasos que hemos sufrido también se han debido a las ideas que nos trajo la empresa ya que una vez implementadas han visto que no eran lo que esperaban y hemos tenido que modificar cosas con el consiguiente aumento del tiempo establecido. Dos grandes cambios que

hubo que hacer sobre lo inicialmente planteado fueron: la pasarela de pago con PayPal y la apariencia de los menús desplegables.

---

## Viabilidad

### 2.1. Análisis de riesgos

El análisis de riesgos es el proceso que permite la identificación de las amenazas que pueden perjudicar al desarrollo del proyecto y determinar el impacto o grado de perjuicio que pueden ocasionar. Implica analizar las amenazas y vulnerabilidades que puedan darse y se valoran en términos de probabilidad de ocurrencia y gravedad de impacto sobre el proyecto.

A continuación se establece una escala (de menor a mayor gravedad) para valorar el impacto que causarían los riesgos que se describen a continuación:

- **Muy grave:** De consecuencias fatales para el desarrollo del proyecto, requeriría replantearse, incluso, si merece la pena continuar con el proyecto.
- **Grave:** Problemas que podrían imposibilitar terminar el proyecto en el plazo previsto. Requeriría realizar una profunda replanificación ampliando el número de horas a dedicar o reduciendo el trabajo a realizar.
- **Perjudicial:** Pequeños retrasos relativamente sencillos de solventar, aunque pueden resultar peligrosos si no se identifican y subsanan a tiempo.

Los posibles riesgos que se tienen en cuenta en la realización del proyecto los siguientes:

Elección equivocada de las herramientas de trabajo:

- Descripción: Es posible que con alguna de las tecnologías elegidas no sea posible cumplir los objetivos planteados o dar a la aplicación la funcionalidad deseada.
- Probabilidad: 10
- Alcance: Muy grave, ya que supondría replantear por completo el proyecto.
- Medidas preventivas: Realizar un estudio profundo sobre las diferentes tecnologías y tener planteadas tecnologías alternativas.
- Medidas correctoras: Comenzar la implementación con otras tecnologías en el menor tiempo posible.

Desconocimiento del software elegido:

- Descripción: En este proyecto se van a usar varias herramientas con las que el desarrollador no está nada familiarizado y esto podría ocasionar retrasos a la hora de realizar la aplicación.
- Probabilidad: 90
- Alcance: Perjudicial, impediría avanzar en el tiempo estimado.
- Medidas preventivas: Planificar una correcta formación en el uso de las tecnologías que sean nuevas y tener preparado de antemano documentación al respecto.
- Medidas correctoras: Estudiar la documentación y usar ejemplos para adaptarse a las nuevas herramientas en un tiempo breve.

Estancamiento en la codificación:

- Descripción: Podría llegar a darse algún problema de difícil solución que paralizaría momentáneamente el avance del proyecto.
- Probabilidad: 20
- Alcance: Perjudicial, impediría avanzar en el tiempo estimado.
- Medidas preventivas: Tener bien estudiadas las herramientas de trabajo y ajustar los plazos a las necesidades del desarrollador.
- Medidas correctoras: Re planificar las tareas posteriores y dedicar más formación a las herramientas.

Pérdida de los datos del proyecto por fallo del hardware o software maligno:

- Descripción: Debido a un virus o a un fallo en el disco duro del ordenador en el que se desarrolla el proyecto se podrían perder todos los datos.
- Probabilidad: 10
- Alcance: Muy grave, pues supondría empezar de cero.
- Medidas preventivas: Tener copias de seguridad del proyecto en distintos ordenadores y dispositivos externos e instalar un antivirus y cortafuegos en el sistema.
- Medidas correctoras: Intentar recuperar la información de los medios de almacenamiento.

Problema con los recursos materiales:

- Descripción: Los ordenadores usados para desarrollar el proyecto pueden no superar el rendimiento necesario para realizar la aplicación e incluso sufrir algún percance que los deje inservibles temporal o permanentemente.
- Probabilidad: 10
- Alcance: Perjudicial, ya que retrasarían el proyecto de una forma indefinida aunque no impedirían su desarrollo.
- Medidas preventivas: Realizar un mantenimiento frecuente de las computadoras así como hacer uso correcto de ellas y tener preparados más dispositivos que se puedan usar como alternativa en caso de emergencia.
- Medidas correctoras: Buscar una solución rápidamente ya sea reparando las computadoras o adquiriendo otras nuevas.

Baja del desarrollador de la aplicación:

- Descripción: Podría suceder que por enfermedad o accidente, el desarrollador del proyecto quede incapacitado durante una temporada.
- Probabilidad: 5
- Alcance: Perjudicial, puesto que no impide la realización del proyecto pero sí lo puede retrasar.

- Medidas preventivas: No previsible.
- Medidas correctoras: Dependiendo de la gravedad del problema sería posible continuar avanzando en el proyecto.

Análisis o planificación muy optimistas:

- Descripción: Una mala planificación podría acarrear retrasos en el desarrollo del proyecto.
- Probabilidad: 30
- Alcance: Perjudicial, supondría un retraso del proyecto respecto a la planificación.
- Medidas preventivas: Realizar una planificación lo más realista posible sin estar demasiado ajustada y teniendo en cuenta las probabilidades de los riesgos que se están analizando.
- Medidas correctoras: Modificar la planificación alargando la duración del proyecto o reduciendo las pretensiones iniciales.

## 2.2. Planificación del tiempo del proyecto

Este apartado describe la estructura de descomposición del trabajo utilizada en la realización de la aplicación, las fases, tareas y entregables del proyecto, así como la agenda de trabajo y los recursos materiales empleados.

### 2.2.1. Estructura de descomposición del trabajo

La EDT (Estructura de descomposición del trabajo) pretende estructurar de manera exhaustiva, jerárquica y descendente los entregables y las tareas necesarias para completar el proyecto. Por tanto, la planificación del proyecto quedaría según las figuras [2.1](#) y [2.2](#)



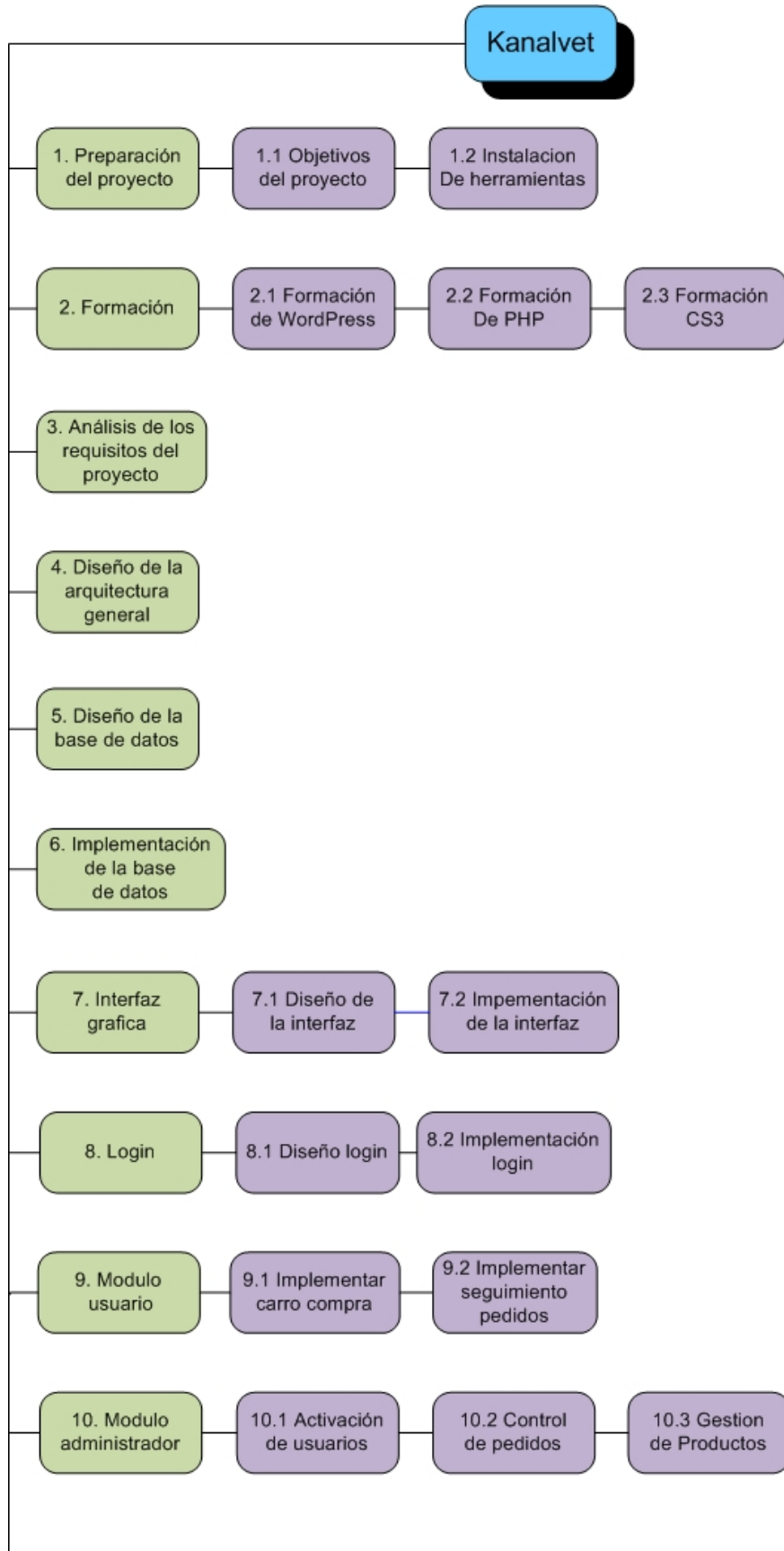


Figura 2.1: Estructura de descomposición del trabajo I.

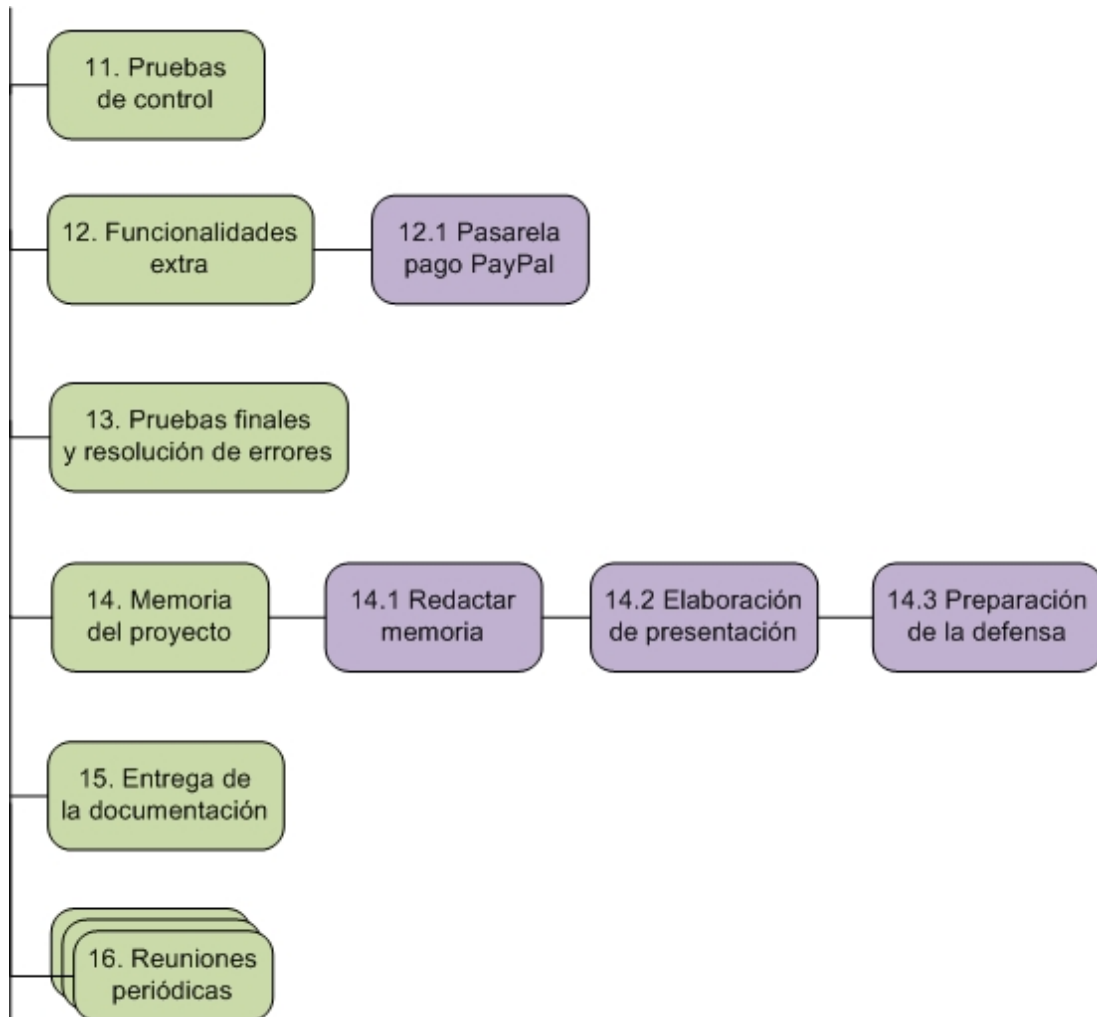


Figura 2.2: Estructura de descomposición del trabajo II.

### 2.2.2. Fases, tareas y entregables

Esta sección ilustra y describe las fases, tareas y entregables planificados para la realización del proyecto.

#### 2.2.2.1. Listado de fases tareas y entregables

A continuación se muestra el listado de fases y tareas de las que dispondrá el proyecto:

0 Kanalvet

#### **1 Preparación del proyecto**

- 1.1 Objetivos del proyecto
- 1.2 Instalación de herramientas

#### **2 Formación**

- 2.1 Formación PHP
- 2.2 Formación WordPress
- 2.3 Formación GIMP 2.6

#### **3 Análisis de los requisitos del proyecto**

#### **4 Diseño de la arquitectura general del proyecto**

#### **5 Diseño de la base de datos**

#### **6 Implementación de la base de datos**

#### **7 Interfaz Gráfica**

- 7.1 Diseño de la interfaz
- 7.2 Implementación de la interfaz

#### **8 Login**

- 8.1 Diseño Login
- 8.2 Implementación Login

#### **9 Módulo Usuario**

- 9.1 Función Carro Compra
- 9.2 Función Seguimiento Pedidos

#### **10 Módulo administrador**

- 10.1 Función activar usuarios
- 10.2 Función Control pedidos
- 10.3 Función gestión productos

#### **11 Pruebas de control**

#### **12 Funcionalidades Extra**

12.1 Pasarela pago PayPal

### **13 Pruebas finales y resolución de errores**

### **14 Memoria del proyecto**

14.1 Redactar memoria

14.2 Elaboración de la presentación

14.3 Preparación de la defensa

### **15 Entrega de la documentación**

### **16 Reuniones periódicas**

16.1 Reuniones periódicas 1

16.2 Reuniones periódicas 2

16.3 Reuniones periódicas 3

16.4 Reuniones periódicas 4

16.5 Reuniones periódicas 5

16.6 Reuniones periódicas 6

16.7 Reuniones periódicas 7

16.8 Reuniones periódicas 8

16.9 Reuniones periódicas 9

16.10 Reuniones periódicas 10

16.11 Reuniones periódicas 11

16.12 Reuniones periódicas 12

16.13 Reuniones periódicas 13

16.14 Reuniones periódicas 14

16.15 Reuniones periódicas 15

16.16 Reuniones periódicas 16

#### 2.2.2.2. Descripción detallada de las tareas y entregables

A continuación se muestra una ficha para cada tarea identificada, donde se anota su número, nombre, una breve descripción y el trabajo estimado:

Nombre : Objetivos del proyecto

Número: 1.1

Descripción: Hacer un análisis previo para determinar los objetivos que se intentarán alcanzar con este proyecto

Trabajo estimado: 8 horas

Nombre : Instalación de herramientas  
Número: 1.2  
Descripción: Instalar en el equipo de trabajo las herramientas necesarias para el desarrollo del proyecto  
Trabajo estimado: 8 horas

Nombre : Formación PHP  
Número: 2.1  
Descripción: Realizar un estudio exhaustivo para familiarizarse con la herramienta y adquirir los conocimientos necesarios sobre PHP  
Trabajo estimado: 24 horas

Nombre : Formación WordPress  
Número: 2.2  
Descripción: Realizar un estudio exhaustivo para familiarizarse con la herramienta y adquirir los conocimientos necesarios sobre el gestor de contenidos WordPress  
Trabajo estimado: 9 horas

Nombre : Formación Photoshop GIMP 2.6  
Número: 2.3  
Descripción: Realizar un estudio exhaustivo para familiarizarse con la herramienta y adquirir los conocimientos necesarios sobre GIMP 2.6  
Trabajo estimado: 10 horas

Nombre : Análisis de los requisitos del proyecto  
Número: 3  
Descripción: Definición de los requisitos funcionales que va a tener el proyecto para cumplir los requisitos.  
Trabajo estimado: 10 horas

Nombre : Diseño de la arquitectura general del proyecto

Número: 4

Descripción: Especificaciones de la estructura interna de la aplicación

Trabajo estimado: 14 horas

Nombre : Diseño de la base de datos

Número: 5

Descripción: Determinar cuál va a ser la estructura de la base de datos, número de tablas, atributos de cada una y relaciones.

Trabajo estimado: 4 horas

Nombre : Implementación de la base de datos

Número: 6

Descripción: Implementación en SQL de la base de datos según las especificaciones que se han hecho en el diseño

Trabajo estimado: 4 horas

Nombre : Diseño de la interfaz

Número: 7.1

Descripción: Se especifica cómo va a ser el entorno visual por el que se van a mover los usuarios

Trabajo estimado: 40 horas

Nombre : Implementación de la interfaz

Número: 7.2

Descripción: Implementación de la interfaz siguiendo el diseño realizado

Trabajo estimado: 60 horas

Nombre : Diseño login

Número: 8.1

Descripción: Se especifica la manera en la que los usuarios podrán darse de alta y autenticarse dentro de la aplicación

Trabajo estimado: 16 horas

Nombre : Implementación login

Número: 8.2

Descripción: Implementación del login siguiendo el diseño realizado.

Trabajo estimado: 22 horas

Nombre : Función carro compra

Número: 9.1

Descripción: Desarrollo de funcionalidad de carro de compra para usuarios registrados.

Trabajo estimado: 30 horas

Nombre : Función seguimiento pedidos

Número: 9.2

Descripción: Desarrollo de funcionalidad de seguimiento de pedidos para usuarios registrados.

Trabajo estimado: 15 horas

Nombre : Función activar usuarios

Número: 10.1

Descripción: Desarrollo de funcionalidad dedicada la activación/desactivación de usuarios registrados dentro de la aplicación.

Trabajo estimado: 16 horas

Nombre : Función control pedidos

Número: 10.2

Descripción: Desarrollo de funcionalidad para la gestión de pedidos realizados por usuarios.

Trabajo estimado: 16 horas

Nombre : Función gestión productos

Número: 10.3

Descripción: Desarrollo de funcionalidad de gestión de productos encargada de actualizar toda la información referente a los productos.

Trabajo estimado: 90 horas

Nombre : Pruebas de control

Número: 11

Descripción: Pruebas de control de permisos para diferentes usuarios y administradores.

Trabajo estimado: 30 horas

Nombre : Pasarela de pago PayPal

Número: 12.1

Descripción: Desarrollo de funcionalidad que realiza una pasarela de pago vía PayPal para el pago de productos comprados.

Trabajo estimado: 80 horas

Nombre : Pruebas finales y resolución de errores

Número: 13

Descripción: Plan de pruebas definitivo para comprobar la correcta interacción de todos los elementos del proyecto y el buen funcionamiento.

Trabajo estimado: 30 horas



Nombre : Redactar memoria

Número: 14.1

Descripción: Redactar memoria de la aplicación especificando todos los detalles sobre la aplicación y su creación.

Trabajo estimado: 70 horas

Nombre : Elaboración de la presentación

Número: 14.2

Descripción: Elaborar las diapositivas necesarias para la defensa del proyecto

Trabajo estimado: 16 horas

Nombre : Preparación de la defensa

Número: 14.3

Descripción: Realizar ensayos previos a la presentación para conseguir una buena exposición final del proyecto.

Trabajo estimado: 10 horas

Nombre: Entrega de la documentación.

Número: 7.1

Descripción: Entrega de la documentación.

Trabajo estimado: 0 horas

Nombre: Reuniones periódicas

Número: 16.1 -16.16

Descripción: Reunión semanal todos los viernes para organizar el proyecto, plantear problemas, soluciones, etc.

Trabajo estimado: 2 horas por reunión.

### 2.2.3. Recursos humanos y materiales

Como recurso humano para la realización de este proyecto se dispone de una única persona que desarrolla tanto trabajo de analista como de programador (Mikel Bravo Rodríguez).

Como recursos materiales, se ha establecido 1 ordenador de sobremesa clónico sin sistema operativo integrado que aporta modularidad, genericidad y escalabilidad al hardware; así como un servidor web mantenido por terceros, donde se alojará finalmente la aplicación y en el cuál se realizarán las pruebas de la misma.

A continuación se muestra una lista detallada de las características de cada uno de los recursos:

**PC Sobremesa:**

Modelo: Clónico

Intel Core 2 duo E6550 2.33Ghz

RAM: 2GB

HD: SATA de 500GB

Gráfica: nVidia 8600GT

**Servidor Web:**

SO: CentOS (Linux)

Version PHP: 5.2.9

Version Apache: 2.2.17

Version MySQL: 5.0.92-community

### 2.2.4. Asignación de recursos y agenda del proyecto

Se estima que la realización del proyecto durará del día 3 de septiembre de 2010 al 29 de de diciembre de 2010. Esta planificación está adaptada a las necesidades del desarrollador ya que deberá compatibilizar el proyecto con las clases en la universidad. Por otra parte, se deja margen para llegado el caso de requerir más tiempo del previsto en alguna tarea, hay posibilidad de realizarla.

Siguiendo el calendario laboral [2.1](#) oficial BOE para Bizkaia se han establecido los

<b>Fecha</b>	<b>Evento</b>
1 de enero	Año Nuevo
6 de enero	Epifanía del Señor
19 de marzo	San José
1 de abril	Jueves Santo
2 de abril	Viernes Santo
5 de abril	Lunes de Pascua
1 de mayo	Fiesta del Trabajo
1 de julio	San Ignacio de Loyola
20 de agosto	Viernes de la Semana Grande
12 de octubre	Día de la Hispanidad
1 de noviembre	Todos los Santos
6 de diciembre	Día de la Constitución
8 de diciembre	La Inmaculada Concepción
25 de diciembre	Navidad

Tabla 2.1: Calendario laboral

siguientes días como festivos.

A continuación se muestra la lista de tareas con la asignación de recursos y el trabajo estimado de cada una así como el diagrama de Gantt. De dichas tablas se han obtenido los siguientes datos:

Trabajo total estimado: 632 horas

Coste estimado: 4.572,60 €

## 2.2.4.1. Planificación

A continuación se muestra el listado de fases y tareas de las que dispondrá el proyecto, así como los recursos asignados a cada tarea.

EDT	Nombre de tarea	Duración	Comienzo	Fin	Costo
0	Kanalvet	632 horas	vie 03/09/10	mié 29/12/10	3.376,60 €
1	Preparación del proyecto	16 horas	vie 03/09/10	lun 06/09/10	86,30 €
1.1	Objetivos del proyecto	8 horas	vie 03/09/10	vie 03/09/10	146,40 €
1.2	Instalación de herramientas	8 horas	lun 06/09/10	lun 06/09/10	43,90 €
2	Formación	43 horas	mar 07/09/10	mar 14/09/10	232,40 €
2.1	Formación PHP	24 horas	mar 07/09/10	jue 09/09/10	128,70 €
2.2	Formación WordPress	9 horas	vie 10/09/10	lun 13/09/10	49,20 €
2.3	Formación GIMP2.6	10 horas	lun 13/09/10	mar 14/09/10	54,50 €
3	Análisis de los requisitos del proyecto	10 horas	mar 14/09/10	mié 15/09/10	183,00 €
4	Diseño de la arquitectura general del proyecto	14 horas	mié 15/09/10	vie 17/09/10	256,20 €
5	Diseño de la base de datos	4 horas	vie 17/09/10	vie 17/09/10	73,20 €
6	Implementación de la base de datos	4 horas	vie 17/09/10	lun 20/09/10	22,70 €
7	Interfaz Gráfica	100 horas	lun 20/09/10	mié 06/10/10	531,50 €
7.1	Diseño de la interfaz	40 horas	lun 20/09/10	lun 27/09/10	732,00 €
7.2	Implementación de la interfaz	60 horas	lun 27/09/10	mié 06/10/10	319,50 €
8	Login	38 horas	mié 06/10/10	jue 14/10/10	202,90 €
8.1	Diseño login	16 horas	mié 06/10/10	vie 08/10/10	292,80 €
8.2	Implementación login	22 horas	vie 08/10/10	jue 14/10/10	118,10 €
9	Módulo Usuario	45 horas	jue 14/10/10	vie 22/10/10	241,50 €
9.1	Función Carro Compra	30 horas	jue 14/10/10	mié 20/10/10	160,50 €

Tabla 2.2: Listado de fases tareas y entregables I.

EDT	Nombre de tarea	Duración	Comienzo	Fin	Costo
9.2	Función Seguimiento Pedidos	15 horas	mié 20/10/10	vie 22/10/10	81,00 €
10	Módulo administrador	122 horas	vie 22/10/10	lun 15/11/10	651,10 €
10.1	Función activar usuarios	16 horas	vie 22/10/10	mar 26/10/10	86,30 €
10.2	Función Control pedidos	16 horas	mar 26/10/10	jue 28/10/10	86,30 €
10.3	Función gestión productos	90 horas	jue 28/10/10	lun 15/11/10	478,50 €
11	Pruebas de control	30 horas	lun 15/11/10	vie 19/11/10	160,50 €
12	Funcionalidades Extra	80 horas	vie 19/11/10	vie 03/12/10	425,50 €
12.1	Pasarela pago PayPal	80 horas	vie 19/11/10	vie 03/12/10	425,50 €
13	Pruebas finales y resolución de errores	30 horas	vie 03/12/10	lun 13/12/10	160,50 €
14	Memoria del proyecto	96 horas	mar 14/12/10	mié 29/12/10	513,30 €
14.1	Redactar memoria	70 horas	mar 14/12/10	vie 24/12/10	372,50 €
14.2	Elaboración de la presentación	16 horas	vie 24/12/10	mar 28/12/10	86,30 €
14.3	Preparación de la defensa	10 horas	mar 28/12/10	mié 29/12/10	54,50 €
15	Entrega de la documentación	0 horas	mié 29/12/10	mié 29/12/10	0,00 €
16	Reuniones periódicas	562 horas	vie 03/09/10	vie 17/12/10	0,00 €

Tabla 2.3: Listado de fases tareas y entregables II.

Nombre de tarea	Nombres de los recursos
Kanalvet	
Preparación del proyecto	Analista
Objetivos del proyecto	Programador;PC Sobremesa[1]
Instalación de herramientas	Programador;PC Sobremesa[1]
Formación	Programador;PC Sobremesa[1]
Formación PHP	Programador;PC Sobremesa[1]
Formación WordPress	Programador;PC Sobremesa[1]
Formación CS3	Analista
Análisis de los requisitos del proyecto	Analista
Diseño de la arquitectura general del proyecto	Analista
Diseño de la base de datos	Analista
Implementación de la base de datos	Programador;PC Sobremesa[1]
Interfaz Gráfica	Analista
Diseño de la interfaz	Programador;PC Sobremesa[1]
Implementación de la interfaz	Analista
Login	Programador;PC Sobremesa[1]
Diseño login	Analista
Implementación login	Programador;PC Sobremesa[1]
Módulo Usuario	Programador;PC Sobremesa[1]
Función Carro Compra	Programador;PC Sobremesa[1]

Tabla 2.4: Asignación de recursos a tareas I.

Nombre de tarea	Nombres de los recursos
Función Seguimiento Pedidos	Programador;PC Sobremesa[1]
Módulo administrador	Programador;PC Sobremesa[1]
Función activar usuarios	Programador;PC Sobremesa[1]
Función Control pedidos	Programador;PC Sobremesa[1]
Función gestión productos	Programador;PC Sobremesa[1]
Pruebas de control	Programador;PC Sobremesa[1]
Funcionalidades Extra	Programador;PC Sobremesa[1]
Pasarela pago PayPal	Programador;PC Sobremesa[1]
Pruebas finales y resolución de errores	Programador;PC Sobremesa[1]
Memoria del proyecto	Programador;PC Sobremesa[1]
Redactar memoria	Programador;PC Sobremesa[1]
Elaboración de la presentación	Programador;PC Sobremesa[1]
Preparación de la defensa	Programador;PC Sobremesa[1]
Entrega de la documentación	Programador;PC Sobremesa[1]
Reuniones periódicas	Programador;PC Sobremesa[1]

Tabla 2.5: Asignación de recursos a tareas II.

#### 2.2.4.2. Diagrama de Gantt

El diagrama de Gantt 2.3 muestra el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo del tiempo total determinado para el proyecto.

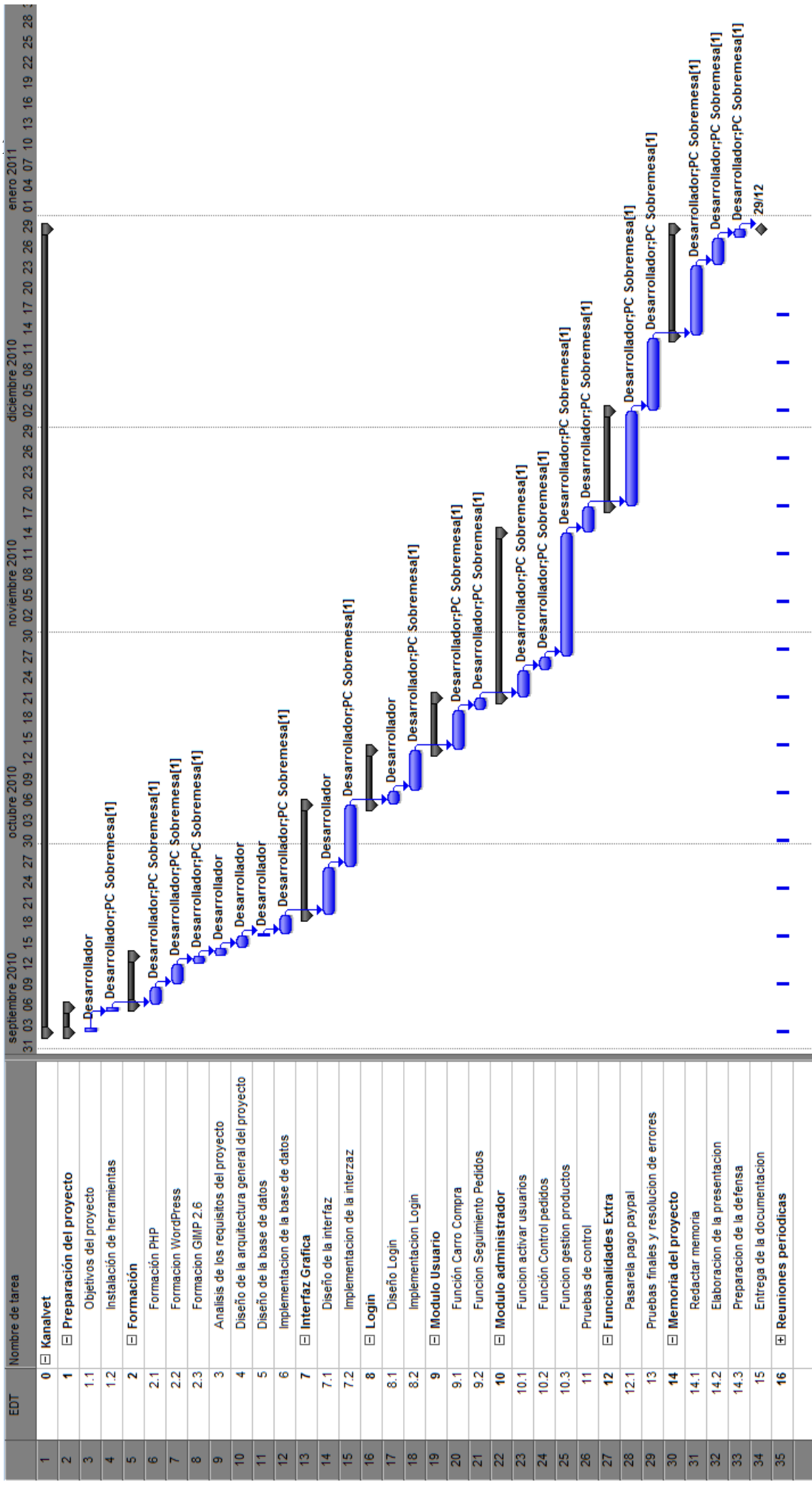


Figura 2.3: Diagrama de Gantt



## 2.3. Estimación de costes del proyecto

Para realizar la estimación de costos se toman como referencia los datos que aparecen en las tablas adjuntas. En ellas se indica lo que cobra cada uno de los distintos perfiles de los recursos humanos. Tienen una tasa estándar y una tasa de horas extra que es en todos los casos la tasa estándar más un 10%. La tasa estándar se ha calculado a partir de las tablas de sueldos del CEDS (Centro de Estudios y Diseño de Sistemas) con los siguientes datos:

Analista informático:  $41.000\text{€} \div 14 \text{ pagas} \div 20 \text{ días} \div 8 \text{ horas} = 18.30\text{€/h}$

Programador informático:  $11.872\text{€} \div 14 \text{ pagas} \div 20 \text{ días} \div 8 \text{ horas} = 5.30 \text{ €/h}$

Los recursos materiales se consideran cada uno de ellos con un coste por uso de 1,50€ en concepto de luz y otros gastos que se cobran independientemente del tiempo de uso.

En la realización de este presupuesto se tienen en cuenta tanto los recursos materiales como las licencias de software necesarias para el desarrollo del proyecto.

### 2.3.1. Recursos de trabajo

La tabla 2.6 recursos de trabajo, muestra los trabajadores del proyecto así como su perfil y el importe que genera cada uno de ellos.

Concepto	Tasa estándar	Tasa horas extra
Analista	18,30€/hora	24,70€/hora
Programador	5,30€/hora	7,15€/hora

Tabla 2.6: Recursos de trabajo

### 2.3.2. Recursos materiales

En las tablas 2.7 y 2.8 se muestran los recursos materiales hardware y software empleados en el proyecto.

### 2.3.3. Costo de recursos de trabajo

La tabla 2.9 muestra el costo de los recursos de trabajo.

Concepto	Coste unitario
PC Sobremesa	478,90€
Hosting Web	120,00€

Tabla 2.7: Recursos materiales (Hardware)

Concepto	Coste unitario	Num. de licencias
MS Windows 7 ultimate	319,00€	1
MS Project 2010	775,00€	1
MS Visio 2010	330,00€	1
Sublime Text 2	40,00€	1
Antivirus Nod32	39,95€	1
Linux Mint 11	0,00€	1
Gimp 2.6	0,00€	1
Eclipse 3.6	0,00€	1
Filezilla	000€	1

Tabla 2.8: Recursos materiales (Software).

Concepto	Trabajo	Coste	Importe
Analista	92 horas	18,30€	1.683,60€
Programador	540 horas	5,30€	2.862,00€
<b>Total</b>			<b>4.545,60€</b>

Tabla 2.9: Costo de recursos de trabajo

#### 2.3.4. Costo de recursos materiales

La tabla 2.10 muestra el costo de los recursos materiales.

Concepto	Unidades	Precio	Importe
PC Sobremesa	18	1,50€	27,00€
<b>Total</b>			<b>27,00€</b>

Tabla 2.10: Costo de recursos materiales

### 2.3.5. Amortizaciones

Para el cálculo de las amortizaciones (Véase la tabla 2.11) se ha considerado un tiempo de amortización de 3 años. Teniendo en cuenta que el número de horas laborables al año es de 200, el tiempo de amortización (en horas) se obtiene de la siguiente forma:

$$4800 \text{ horas} = 3 \text{ años} \times 200 \text{ días laborables} \times 8 \text{ horas}$$

Concepto	Coste Unitario	tiempo de amortización	Coste unitario de amortización	Tiempo de uso	Importe
PC Sobre mesa	478,90 €	4800	0,09977 €	480h	47,88 €
MS Windows 7 Ultimate	319,00 €	4800	0,06645 €	21h	1,34 €
MS Project 2010	755,00 €	4800	0,16146 €	15h	2,42 €
MS Visio 2010	330,00 €	4800	0,06875 €	6h	0,41 €
Antivirus Nod32	39,95 €	4800	0,00832 €	21h	0,17 €
Sublime Text 2	40,00 €	4800	0,00833 €	70h	0,58 €
Eclipse 3.6	0,00 €	4800	0,00000 €	315h	0,00 €
Gimp 2.6 3.6	0,00 €	4800	0,00000 €	20h	0,00 €
Filezilla	0,00 €	4800	0,00000 €	20h	0,00 €
<b>Total</b>					<b>52,77€</b>

Tabla 2.11: Amortizaciones

### 2.3.6. Presupuesto

En la tabla 2.12 se recogen los resultados de las tablas 2.6 a 2.11 a los que se les añade un 10% en concepto de gastos generales y un 15% extra en concepto de beneficios para la empresa. Al total obtenido se le suma el 18% de IVA.

<b>Concepto</b>	<b>Importe</b>
Recursos de trabajo	4.545,60 €
Recursos materiales	27,00 €
Amortizaciones	52,77 €
<b>Total</b>	<b>4.625,37 €</b>
Gastos Generales(10%)	462,53 €
Beneficio(15%)	693,80 €
<b>Subtotal</b>	<b>5.781,70 €</b>
IVA(18%)	1.040,70 €
<b>Total</b>	<b>6.822,40 €</b>

Tabla 2.12: Presupuesto

Por tanto, el presupuesto asciende a seis mil ochocientos veintidós euros con cuarenta céntimos.

Estimando que los datos aportados en el presente presupuesto son suficientes, se somete el proyecto a su aprobación por los organismos oficiales correspondientes.

Firma: Mikel Bravo Rodríguez

Vitoria-Gasteiz, a 14 de Mayo de 2010

## **Parte II**

# **Conceptos generales, tecnologías y herramientas**



---

## Conceptos generales

Este capítulo recoge información elemental sobre la arquitectura modelo vista controlador, empleada en Kanalvet para el diseño de la aplicación de una forma que los cambios realizados en la misma se puedan realizar sin mucha dificultad. También se hace una breve descripción de la criptografía, utilizada en la pasarela de PayPal.

### 3.1. Modelo / Vista / Controlador

Esta arquitectura de software separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón de llamada y retorno MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página.

La arquitectura MVC (Model/View/Controller) fue introducida como parte de la versión Smalltalk-80 del lenguaje de programación Smalltalk. Fue diseñada para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas. Este modelo de arquitectura se puede emplear en sistemas de representación gráfica de datos, como se ha citado, o en sistemas CAD, en donde se presentan partes del diseño con diferente escala de aumento, en ventanas separadas. En la figura siguiente, vemos la arquitectura MVC en su forma más general. Hay un Modelo, múltiples Controladores que manipulan ese Modelo, y hay varias Vistas de los datos del Modelo, que cambian cuando cambia el estado de ese Modelo. Este modelo de

arquitectura presenta varias ventajas:

- Hay una clara separación entre los componentes de un programa; lo cual nos permite implementarlos por separado
- Hay un API muy bien definido; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

Al incorporar el modelo de arquitectura MVC a un diseño, las piezas de un programa se pueden construir por separado y luego unirlos en tiempo de ejecución. Si uno de los Componentes, posteriormente, se observa que funciona mal, puede reemplazarse sin que las otras piezas se vean afectadas. Este escenario contrasta con la aproximación monolítica típica de muchos programas Java. Todos tienen un Frame que contiene todos los elementos, un controlador de eventos, un montón de cálculos y la presentación del resultado. Ante esta perspectiva, hacer un cambio aquí no es nada trivial.

### 3.1.1. Definición de las partes

#### 3.1.1.1. El Modelo

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

#### 3.1.1.2. La Vista

La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.



## 3.1.1.3. El Controlador

El Controlador es el objeto que proporciona significado a las ordenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo. Vamos a mostrar un ejemplo concreto. Consideremos como tal el sistema descrito en la introducción a este capítulo, una pieza geométrica en tres dimensiones, que representamos en la figura siguiente:

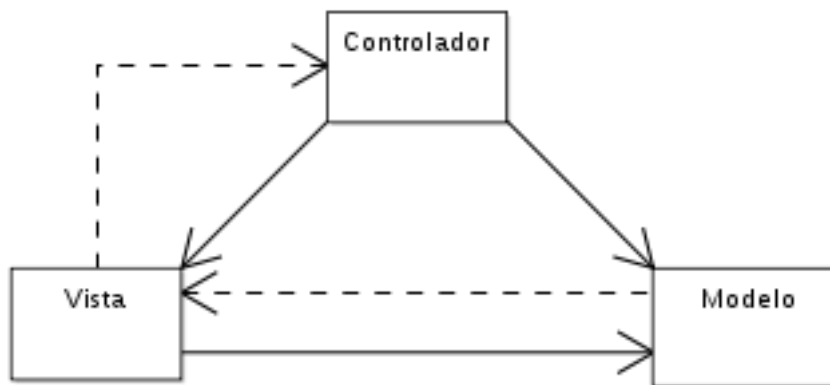


Figura 3.1: Diagrama MVC

En este caso, la pieza central de la escena en tres dimensiones es el Modelo. El Modelo es una descripción matemática de los vértices y las caras que componen la escena. Los datos que describen cada vértice o cara pueden modificarse (quizás como resultado de una acción del usuario, o una distorsión de la escena, o un algoritmo de sombreado). Sin embargo, no tiene noción del punto de vista, método de presentación, perspectiva o fuente de luz. El Modelo es una representación pura de los elementos que componen la escena. La porción del programa que transforma los datos dentro del Modelo en una presentación gráfica es la Vista. La Vista incorpora la visión del Modelo a la escena; es la representación gráfica de la escena desde un punto de vista determinado, bajo condiciones de iluminación determinadas. El Controlador sabe que puede hacer el Modelo e implementa el interface de usuario que permite iniciar la acción. En este ejemplo, un panel de datos de entrada es lo único que se necesita, para permitir añadir, modificar o borrar vértices o caras de la figura

### 3.1.2. MVC en Kanalvet

Para poder desarrollar Kanalvet como una aplicación de MVC nos ayudamos mucho del CSM Wordpress ya que este nos daba la vista dejándonos para los desarrolladores la realización del modelo y el controlador.

El modelo fue llevado a cabo creando objetos llamados DAO (Data Access Object). La función que tienen estos objetos no es más que la de interactuar con la base de datos y brindarnos los datos almacenados en la base de datos de forma que desde el resto de componentes tengan una total abstracción de la estructura interna de la base de datos, y en caso de necesitar hacer algún cambio en la base de datos, con solo modificar estos archivos el resto de la aplicación siga funcionando correctamente.

Para el controlador se han usado una serie de funciones que se encargan de controlar en qué estado de la aplicación se encuentra el usuario para así mediante instrucciones a la vista y al modelo poder generar la pagina resultado de las necesidades del usuario.

## 3.2. Criptografía

### 3.2.1. Criptografía simétrica

La criptografía simétrica es un método criptográfico en el cual se usa una misma clave para cifrar y descifrar mensajes. Las dos partes que se comunican han de ponerse de acuerdo de antemano sobre la clave a usar. Una vez ambas tienen acceso a esta clave, el remitente cifra un mensaje usándola, lo envía al destinatario, y éste lo descifra con la misma.

#### 3.2.1.1. Algoritmo RSA

En criptografía, RSA (Rivest, Shamir y Adleman) es un sistema criptográfico de clave pública desarrollado en 1977. En la actualidad, RSA es el primer y más utilizado algoritmo de este tipo y es válido tanto para cifrar como para firmar digitalmente.

La seguridad de este algoritmo radica en el problema de la factorización de números enteros. Los mensajes enviados se representan mediante números, y el funcionamiento se basa en el producto, conocido, de dos números primos grandes elegidos al azar y mantenidos en secreto. Actualmente estos primos son del orden de 10200 ,

y se prevé que su tamaño aumente con el aumento de la capacidad de cálculo de los ordenadores.

Como en todo sistema de clave pública, cada usuario posee dos claves de cifrado: una pública y otra privada. Cuando se quiere enviar un mensaje, el emisor busca la clave pública del receptor, cifra su mensaje con esa clave, y una vez que el mensaje cifrado llega al receptor, este se ocupa de descifrarlo usando su clave privada.

Se cree que RSA será seguro mientras no se conozcan formas rápidas de descomponer un número grande en producto de primos. La computación cuántica podría proveer de una solución a este problema de factorización.

Kanalvet usa este sistema criptográfico en la pasarela de pago de PayPal, como se puede ver en la sección [5.4](#)



---

## Desarrollo técnico

En este capítulo se describe brevemente el objetivo de las tecnologías que se usan en este trabajo, para qué se usan y por qué se han elegido frente a otras.

### 4.1. Estudio de la tecnología

#### 4.1.1. HTML

HTML es el acrónimo de HyperText Markup Language (Lenguaje de Marcado de Hipertexto) y se define como un lenguaje de programación del lado del cliente (los navegadores son los encargados de interpretar estos códigos).

HTML es el lenguaje con el que se definen las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web.

El HTML se creó en un principio con objetivos divulgativos de información con texto y algunas imágenes. No se pensó que llegara a ser utilizado para crear área de ocio y consulta con carácter multimedia (lo que es actualmente la web), de modo que, el HTML se creó sin dar respuesta a todos los posibles usos que se le iba a dar y a todos los colectivos de gente que lo utilizarían en un futuro. Sin embargo, pese a esta deficiente planificación, si que se han ido incorporando modificaciones con el tiempo, estos son los estándares del HTML. Numerosos estándares se han presentado ya. El HTML 5 es el último estándar a Junio de 2011.

HTML es el estándar que da forma a cualquier página web por lo que para Kanalvet

es una herramienta imprescindible para su desarrollo.

Aunque el último estándar HTML5 salió en Junio de 2011 a la hora de realizar Kanalvet se han dejado de lado las características de este nuevo estándar ya que al desarrollar Kanalvet era un estándar muy joven y para el cual no todos los navegadores tenían soporte.

### 4.1.2. CSS

CSS es el acrónimo de Cascading Style Sheets ( Hojas de estilo en Cascada) y se define como un lenguaje de programación del lado del cliente.

CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

CSS se usa en Kanalvet para dotar a la aplicación de una estética atractiva y una experiencia de usuario agradable.

Al igual que HTML, el estándar CSS se actualizó en Junio de 2011 y por la falta de compatibilidad con navegadores web se descartaron las nuevas características del nuevo estándar usando técnicas antiguas para poder llegar a más gente.

### 4.1.3. JavaScript

Javascript es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.

Muchos confunden el Javascript con el Java pero ambos lenguajes son diferentes y tienen sus características singulares. Javascript tiene la ventaja de ser incorporado en cualquier página web, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado.

Java por su parte tiene como principal característica ser un lenguaje independiente de la plataforma. Se puede crear todo tipo de programa que puede ser ejecutado en cualquier ordenador del mercado: Linux, Windows, Apple, etc. Debido a sus características también es muy utilizado para internet.

Como síntesis se puede decir que Javascript es un lenguaje interpretado, basado en prototipos, mientras que Java es un lenguaje más orientado a objetos.

#### 4.1.3.1. ¿Qué es Javascript?

Javascript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con Javascript podemos crear diferentes efectos e interactuar con nuestros usuarios.

Aunque Javascript tiene un potencial enorme, en Kanalvet se usa únicamente para la verificación de los formularios. Se optó por verificarlos mediante javascript porque así se consigue una web más dinámica con la cual interactúa el usuario sin tener la necesidad de estar recargando la página entera para comprobar ciertos campos. Por lo que hemos hecho uso de esta tecnología para crear una mejor experiencia de usuario.

Como se explica arriba con Javascript hemos conseguido dinamismo a la hora de interactuar con el usuario en los formularios, pero también queríamos aprovechar esta herramienta para hacer la web más vistosa y atractiva para lo cual no hemos usado Javascript “puro”, nos hemos basado en un gran framework basado en Javascript como se explica en el punto [4.1.4](#)

#### 4.1.4. JQuery

jQuery es un framework Javascript. Un framework es un producto que sirve como base para la programación avanzada de aplicaciones, que aporta una serie de funciones o códigos para realizar tareas habituales. Por decirlo de otra manera, framework son unas librerías de código que contienen procesos o rutinas ya listos para usar. Los programadores utilizan los frameworks para no tener que desarrollar ellos mismos las tareas más básicas, puesto que en el propio framework ya hay implementaciones que están probadas, funcionan y no se necesitan volver a programar.

Así pues, este framework Javascript, nos ofrece una infraestructura con la que tendremos mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Por ejemplo, en Kanalvet se ha usado para la creación de menús desplegables, zoom en imágenes... características que se ejecutan de parte del cliente y que consiguen crear una experiencia muy buena y agradable a la hora de navegar por la web y ver los diferentes productos.

Además, todas estas ventajas que sin duda son muy de agradecer, con jQuery las obtenemos de manera gratuita, ya que el framework tiene licencia para uso en cualquier tipo de plataforma, personal o comercial.

#### 4.1.4.1. ¿Por qué JQuery?

Jquery no es el único framework que existe pero finalmente nos decantamos por él porque es uno de los framework mas usados, o por lo menos uno de los que tiene mayo comunidad trabajando detrás y del cual se encuentra más información en internet. Lo elegimos, porque no queríamos construir una web basada únicamente en JQuery si no que lo que queríamos conseguir con esto era darle un poco de dinamismo a Kanalvet con efectos, menús desplegable... pero tampoco centrándonos en ello, y al tener una gran comunidad detrás y con la gran cantidad de información que se puede encontrar por internet nos permitió hacer trabajar fácilmente con él sin necesidad de dedicarle muchas horas o meternos de lleno a desarrollar efectos con él.

#### 4.1.5. PHP

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

PHP se escribe dentro del código HTML, lo que lo hace realmente fácil de utilizar, al igual que ocurre con el popular ASP de Microsoft, pero con algunas ventajas como su gratuidad, independencia de plataforma, rapidez y seguridad. Cualquiera puede descargar a través de la página principal de PHP y de manera gratuita, un módulo que hace que nuestro servidor web comprenda los scripts realizados en este lenguaje. Es independiente de plataforma, puesto que existe un módulo de PHP para casi cualquier servidor web. Esto hace que cualquier sistema pueda ser compatible con el lenguaje y



significa una ventaja importante, ya que permite portar el sitio desarrollado en PHP de un sistema a otro sin prácticamente ningún trabajo.

Por último la seguridad, en este punto también es importante el hecho de que en muchas ocasiones PHP se encuentra instalado sobre servidores Unix o Linux, que son de sobra conocidos como más veloces y seguros que el sistema operativo donde se ejecuta las ASP, Windows NT o 2000. Además, PHP permite configurar el servidor de modo que se permita o rechacen diferentes usos, lo que puede hacer al lenguaje más o menos seguro dependiendo de las necesidades de cada cual.

Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por poner dos ejemplos.

#### 4.1.5.1. ¿Por qué PHP?

PHP posee muchas cualidades, pero las que más nos interesaban y por las cuales nos decantamos por usar PHP fue por ser libre y multiplataforma. Esto nos permite hacer funcionar correctamente la aplicación en cualquier tipo de servidor, y aunque sí teníamos claro que íbamos a usar un servidor basado en Linux esto nos permite que en caso de tener que migrar a otro tipo de servidor en un futuro, la elección de tecnologías usadas en las aplicaciones web no sea un inconveniente a la hora de migrar.

#### 4.1.6. SQL

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés structured query language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.

Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar, de una forma sencilla, información de interés de una base de datos, así como también hacer cambios sobre ella.

El SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales permitiendo gran variedad de operaciones en éstos últimos. Es un lenguaje declarativo de alto nivel o de no procedimiento, que gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros, y no a registros individuales, permite una alta productividad en codificación y la orientación

a objetos. De esta forma una sola sentencia puede equivaler a uno o más programas que se utilizarían en un lenguaje de bajo nivel orientado a registros.

#### 4.1.6.1. ¿Por qué SQL?

Para la realización de Kanalvet se optó por usar SQL, concretamente MySQL, frente a otras opciones ya que es una alternativa libre bastante usada en aplicaciones web y desde la cual en PHP tenemos soporte para poder trabajar con ella.

Las 2 opciones principales que barajamos fueron usar MySQL o PostgreSQL. Finalmente nos decantamos por usar MySQL porque aunque Kanalvet guarda una cantidad enorme de datos en la base de datos, la estructura de esta no es excesivamente compleja por lo que las dos opciones eran igualmente válidas, y al tener más experiencia en bases de datos MySQL optamos por usar estas.

#### 4.1.7. WordPress

WordPress es un sistema de gestión de contenido enfocado a la creación de blogs (sitios web periódicamente actualizados). Desarrollado en PHP y MySQL, bajo licencia GPL.

WordPress es seguramente la plataforma más utilizada para crear y mantener blogs. Si lo usamos correctamente, podemos convertirlo en un CMS (Content Management System) en un alojamiento de páginas web, compartido y profesional.

Un Gestor de Contenidos es una plataforma de desarrollo como corazón de nuestro sitio web. Contiene todos los elementos necesarios, para la administración de sus capacidades, y el mantenimiento y publicación de nuestros contenidos.

WordPress es usado por páginas personales pero también está presente en 500 de las empresas más grandes en Internet. Su robustez y estabilidad, así como la facilidad de mantener las últimas versiones al día lo han hecho un favorito. La clave de que pueda ser usado para múltiples propósitos y temáticas es debido a:

- Facilidad en la construcción y mantenimiento de temas (templates) de los que dispone de una gran oferta de temas, libres, profesionales, a medida. Es posible diseñar Temas a medida, o encontrar una oferta de Temas para WordPress comerciales o gratuitos.
- Código abierto (disponibilidad para la modificación de código)

- Numerosos plugins y widgets Se encuentran muchos Plugins y cientos de widgets fáciles de instalar y que ayudan a caracterizar un web.
- Campos a medida en los POST
- Gran comunidad de usuarios, manuales de ayuda, castellanizado.

#### 4.1.7.1. ¿Por qué WordPress?

Wordpress es una herramienta perfecta para Kanalvet porque, aparte de ser Open-Source de coste 0€, está basada en otras PHP y MySql herramientas que habíamos decidido usar en Kanalvet. Por otra parte Wordpress nos ofrece una estructura desde la cual trabajar y un backend desde el cual poder configurar una gran variedad de aspectos básicos de la aplicación de una forma sencilla

#### 4.1.8. L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X es un sistema de composición de textos que está formado mayoritariamente por órdenes (macros) construidas a partir de comandos de TeX un lenguaje de bajo nivel, en el sentido de que sus acciones últimas son muy elementales pero con la ventaja añadida, en palabras de Lamport, de “poder aumentar las capacidades de L<sup>A</sup>T<sub>E</sub>X utilizando comandos propios del TeX descritos en The TeXbook”. Esto es lo que convierte a L<sup>A</sup>T<sub>E</sub>X en una herramienta práctica y útil pues, a su facilidad de uso, se une toda la potencia de TeX. Estas características hicieron que L<sup>A</sup>T<sub>E</sub>X se extendiese rápidamente entre un amplio sector científico y técnico, hasta el punto de convertirse en uso obligado en comunicaciones y congresos, y requerido por determinadas revistas a la hora de entregar artículos académicos.

Con todo, además de todas las nuevas extensiones, la característica más relevante de este esfuerzo de re estandarización fue la arquitectura modular: se estableció un núcleo central (el compilador) que mantiene las funcionalidades de la versión anterior pero permite incrementar su potencia y versatilidad por medio de diferentes paquetes que sólo se cargan si son necesarios. De ese modo, L<sup>A</sup>T<sub>E</sub>X dispone ahora de innumerables paquetes para todo tipo de objetivos, muchos dentro de la distribución oficial, y otros realizados por terceros, en algunos casos para usos especializados.

#### 4.1.8.1. ¿Por qué $\text{\LaTeX}$ ?

$\text{\LaTeX}$  se ha usado para escribir este documento referente a Kanalvet. Lo normal habría sido usar herramientas como Word de Microsoft Office o alternativas libres como Writter de LibreOffice pero al final me decidí a usar  $\text{\LaTeX}$  para escribir la memoria del proyecto primero por recomendación del profesor director del proyecto y segundo por la inquietud de aprender.

$\text{\LaTeX}$  es una forma completamente diferente de redactar textos a la que estamos acostumbrados, pero no por ello más difícil. Si es cierto que la curva de aprendizaje al principio puede que sea más muy inclinada y que el principio cueste, pero una vez aprendido son todas ventajas ya que posee una gran cantidad de extensiones y características que ayudan mucho en la realización de documentos.

## 4.2. Herramientas

### 4.2.1. Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama “Aplicaciones de Cliente Enriquecido”, opuesto a las aplicaciones “Cliente-liviano” basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

El entorno de desarrollo integrado (IDE) de Eclipse emplea módulos (plug-in) para proporcionar toda su funcionalidad al frente de la plataforma de cliente enriquecido, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de software. Adicionalmente a permitirle a Eclipse extenderse usando otros lenguajes de programación como son C/C++, Python y PHP, permite a Eclipse trabajar con lenguajes para procesado de texto como  $\text{\LaTeX}$ , aplicaciones en red como Telnet y Sistema de gestión de base de datos. La arquitectura plugin permite escribir cualquier extensión deseada en el ambiente, como sería Gestión de la configuración. Se provee soporte para Java y CVS en el SDK de Eclipse. Y no tiene por qué ser usado únicamente para soportar otros lenguajes de programación.

La definición que da el proyecto Eclipse acerca de su software es: “una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular”

#### 4.2.2. Gimp 2.6

GNU Image Manipulation Program(GIMP) es un programa de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías. Es un programa libre y gratuito. Está englobado en el proyecto GNU y disponible bajo la Licencia pública general de GNU.

Es el programa de manipulación de gráficos disponible en más sistemas operativos, como ser, Unix, GNU/Linux, Windows, Mac OS X, entre otros, además se incluye en muchas distribuciones GNU/Linux.

La interfaz de GIMP está disponible en varios idiomas, entre ellos: Español, alemán, inglés, catalán, gallego, euskera, francés, italiano, ruso, sueco, noruego, coreano, neerlandés y en otras lenguas adicionales.

Gimp es una alternativa firme, potente y rápida a Photoshop para muchos usos, aunque no se ha desarrollado como un clon de él y posee una interfaz bastante diferente.

Este programa ha sido adquirido y utilizado para la realización del desarrollo y diseño del logotipo así como el tratamiento gráfico de otros muchos componentes de la aplicación. Al ser totalmente gratuito, abarata el presupuesto del proyecto de manera considerable.

#### 4.2.3. Microsoft Visio 2010

Microsoft Visio es un software de dibujo vectorial para Microsoft Windows. Visio comenzó a formar parte de los productos de Microsoft cuando fue adquirida la compañía Visio en el año 2000.

Las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML, y más, que permiten iniciar al usuario en los lenguajes de programación.

Aunque originalmente apuntaba a ser una aplicación para dibujo técnico para el campo de ingeniería y arquitectura; con añadidos para desarrollar diagramas de negocios, su adquisición por Microsoft implicó drásticos cambios de directrices de tal

forma que a partir de la versión de Visio para Microsoft Office 2003 el desarrollo de diagramas para negocios pasó de añadido a ser el núcleo central de negocio, minimizando las funciones para desarrollo de planos de ingeniería y arquitectura que se habían mantenido como principales hasta antes de la compra. Una prueba de ello es la desaparición de la función *property line* tan útil para trabajos de agrimensura y localización de puntos por radiación, así como el suprimir la característica de *ghost shape* que facilitaba la ubicación de los objetos en dibujos técnicos. Al parecer Microsoft decidió que el futuro del programa residía en el mundo corporativo de los negocios y no en las mesas de dibujo de arquitectos e ingenieros compitiendo con productos como AutoCad, DesignCad, Microstation, etc.

Microsoft Visio ha sido utilizada para la realización de esquemas y gráficos por ser muy intuitiva y permitir la creación de todo tipo de diagramas de manera fácil y rápida.

#### 4.2.4. Microsoft Project 2010

Microsoft Project (o MSP) es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.

El software Microsoft Office Project en todas sus versiones (la versión 2010 es la más reciente) es útil para la gestión de proyectos, aplicando procedimientos descritos en el PMBoK (Management Body of Knowledge) del PMI (Project Management Institute).

Esta herramienta ha sido utilizada por su uso extendido, lo que facilita la formación, además de permitir llevar a cabo gran parte del proceso de gestión del proyecto

#### 4.2.5. Sublime Text

Sublime Text es un editor de código multiplataforma, ligero y con pocas concesiones a las florituras. Es una herramienta concebida para programar sin distracciones. Su interfaz de color oscuro y la riqueza de coloreado de la sintaxis, centra nuestra atención completamente.

Sublime Text permite tener varios documentos abiertos mediante pestañas, e incluso emplear varios paneles para aquellos que utilicen más de un monitor. Dispone de

modo de pantalla completa, para aprovechar al máximo el espacio visual disponible de la pantalla.

El programa cuenta “de serie” con 22 combinaciones de color posibles, aunque se pueden conseguir más. Para navegar por el código cuenta con Minimap, un panel que permite moverse por el código de forma rápida.

El sistema de resaltado de sintaxis de Sublime Text soporta un gran número de lenguajes (C, C++, C#, CSS, D, Erlang, HTML, Groovy, Haskell, HTML, Java, JavaScript,  $\text{\LaTeX}$ , Lisp, Lua, Markdown, Matlab, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile and XML).

He utilizado Sublime Text 2 para la realización de este documento. El documento se ha creado utilizando  $\text{\LaTeX}$  para el cual que Sublime tiene una combinación de colores, aparte de que nos da la posibilidad de compilar directamente desde el propio programa. Esto hace de Sublime Text una alternativa liviana a la hora de crear documentos en  $\text{\LaTeX}$ .





## **Parte III**

### **La aplicación Kanalvet: diseño, estructura y gestión**



---

## Diseño y estructura

Este capítulo describe los principales hitos de desarrollo de Kanalvet, los métodos utilizados para alcanzar dichos hitos, la arquitectura empleada, problemas a los se ha tenido que enfrentar el desarrollador y sus posibles soluciones.

### 5.1. Hito 1: Diseño de la arquitectura de la aplicación

La aplicación Kanalvet se basa en es gestor de contenidos WordPress, por lo que uno de los primeros objetivos de esta aplicación era el de especificar y definir que características de WordPress podían ser útiles para el desarrollo de la aplicación y cuales habría que reescribir.

#### 5.1.1. Descripción general

Antes de profundizar con la estructura usada en Kanalvet es imprescindible conocer el modo en el que funciona el gestor de contenidos que soporta la aplicación.

WordPress es un gestor de contenidos principalmente orientado a aplicaciones web de tipo blog, por lo que su estructura se basa es un único archivo “index” donde se va cargando los diferentes contenidos. Esta estructura se presenta bastante favorable para Kanalvet ya que nos permite mantener una estructura general de la aplicación modificando un solo archivo, pero que se mantiene para cualquier tipo de contenido que se desee mostrar. Para diferenciar los diferentes contenidos que puede tener la aplicación WordPress usa algo que llama “páginas” en las cuales está la información que se quiere mostrar.

Por lo tanto Kanalvet se crea con la siguiente estructura (figura 5.1) :

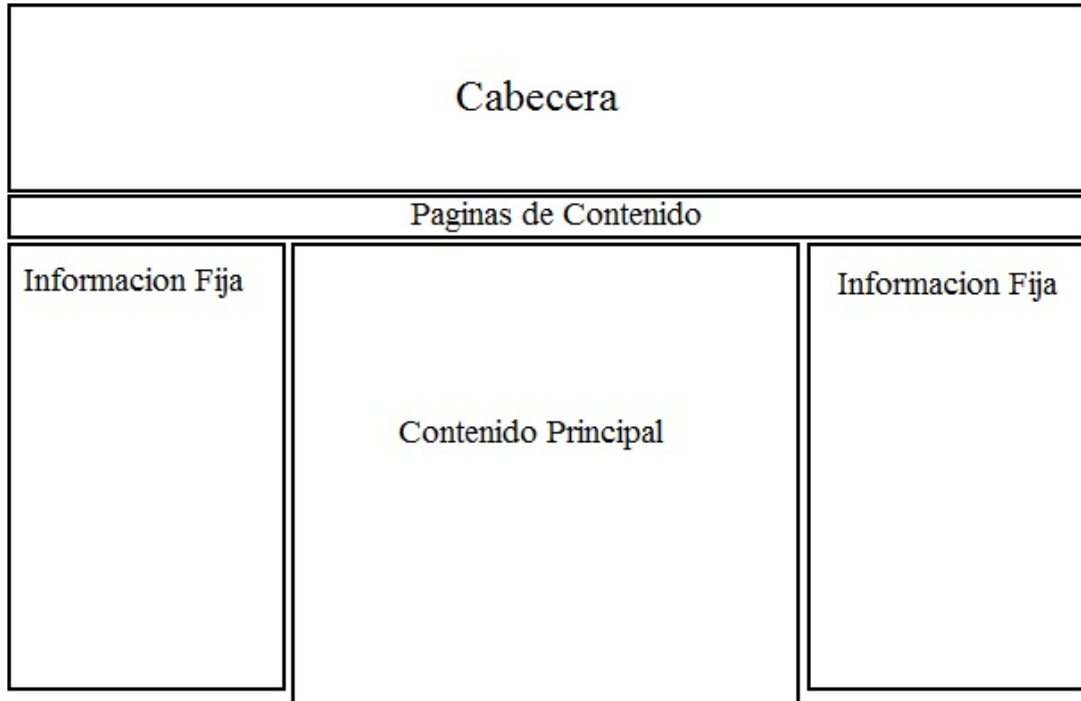


Figura 5.1: Estructura Kanalvet

WordPress, al ser un gestor de contenidos orientado a la creación de Blogs, no trae por defecto la posibilidad de introducir código PHP dentro de sus páginas. Para esa tarea existen muchos plugins, entre los cuales instalamos “PHP Execution” que nos permite introducir nuestro código PHP en cada página de la aplicación.

Para la mejor comprensión y posible cambio del código creado para la aplicación se optó por usar las “páginas de contenido” como simples contenedores en los que se realizan llamadas a las funciones que realmente cargan el contenido de cada páginas, y así tener un único archivo de control desde el cual poder gestionar todas las funciones que componen la aplicación. Esta estructura 5.2 lo que nos permite es poder controlar la mayor parte de la aplicación usando únicamente dos archivos que controlan todo el flujo de control.

Por otra parte, WordPress trae un sistema de login propio, que da la posibilidad de registrar nuevos usuarios, pero este tipo de login guarda poca información del usuario registrado y está muy orientado para cuentas de usuario capaces de cambiar el contenido de la aplicación, por lo que se decidió crear un nuevo sistema de login propio y

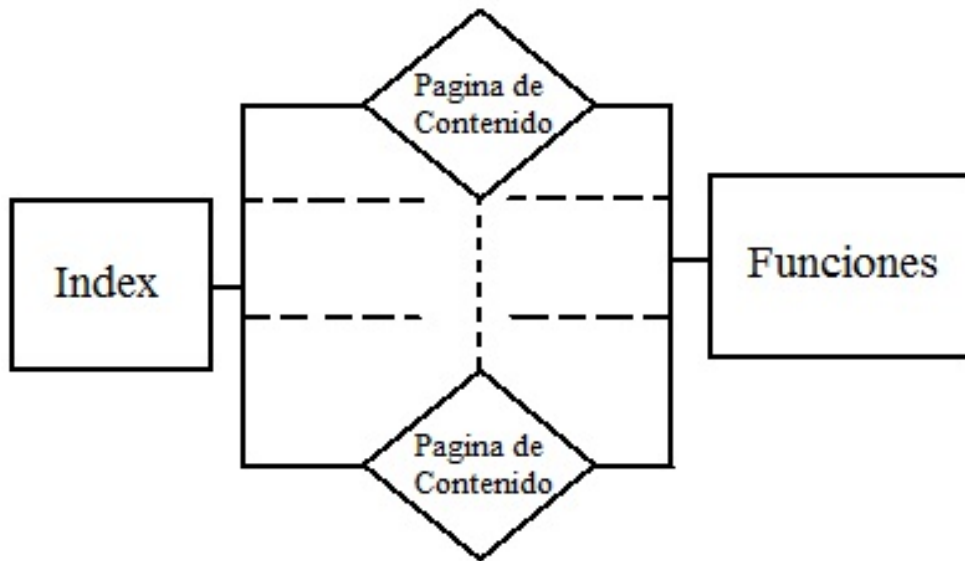


Figura 5.2: Estructura control

personalizado, dejando el login de WordPress únicamente para el desarrollador de la aplicación.

Este nuevo sistema de login y registro creado da la posibilidad de almacenar todos los datos correspondientes a los usuarios, siempre manteniendo la privacidad de sus datos, como por ejemplo la contraseña, que se guardan encriptados en base de datos. Además esta creado con la posibilidad de diferenciar dos tipos de usuarios diferentes: Usuario y administrador.

Por lo tanto tenemos un nuevo login totalmente personalizado dando salida a las necesidades de la empresa. Dando todas las opciones necesarias a los administradores para gestionar los datos de la aplicación de una manera sencilla, y ofreciendo un login sencillo y seguro a los usuarios de la aplicación, dejando de lado el login de WordPress usado por el desarrollador la aplicación para modificar la estructura de páginas, plugins, etc. Evitando así que por error esta estructura sea modificada y se altere la estructura y forma original de la aplicación.

## 5.2. Hito 2: Diseño e implementación de un algoritmo de actualización de datos de la aplicación.

### 5.2.1. Descripción General

Kanalvet es una aplicación que pretende que la empresa solicitante amplie su rango de mercado vía web, por lo que es imprescindible que la aplicación contenga información detallada y actualizada de todos los productos que se ofrecen en ella. Para conseguir todo esto se pide una funcionalidad con la cual personas sin conocimientos de desarrollo web puedan gestionar toda la información contenida en la aplicación sin dificultades.

Teniendo en cuenta las características que esta funcionalidad debe tener y sabiendo que la empresa lleva una gestión de todos sus productos mediante un archivo de excel, se toma la decisión de desarrollar un algoritmo que sea capaz de leer la información referente a los productos y en base a esa información pueda actualizar los datos almacenados en la aplicación. De esta manera se consigue tener la aplicación actualizada con los datos reales de la empresa, sin tener datos duplicados.

### 5.2.2. Diseño Base de Datos

Lo más importante para el desarrollo de esta funcionalidad es definir correctamente cómo se van a guardar los datos en la base de datos de la aplicación. La empresa que solicita la aplicación cuenta con un catalogo de más de 1500 productos separados en más de 550 familias y subfamilias, por lo que mantener una buena relación entre productos y familias es un punto clave para el desarrollo de la aplicación.

Una de las cosas a tener en cuenta es que cada familia no tiene un grupo determinado de subfamilias y la relación entre ellas debe ser flexible y resistente a cambios ya que puede que se introduzcan nuevas familias.

Por lo tanto el diagrama Entidad-Relación queda de la siguiente forma [5.3](#) :

#### **Familia:**

- **id:** Identificador único que identifica la familia
- **nombre:** Nombre de la familia
- **padre:** Identificador de la familia superior. En caso de ser raíz es igual a 0

**Producto:**

- **kod:** Código único que identifica el producto
- **nombre:** Nombre del producto
- **precio:** Precio del producto
- **descripción:** pequeña descripción del producto

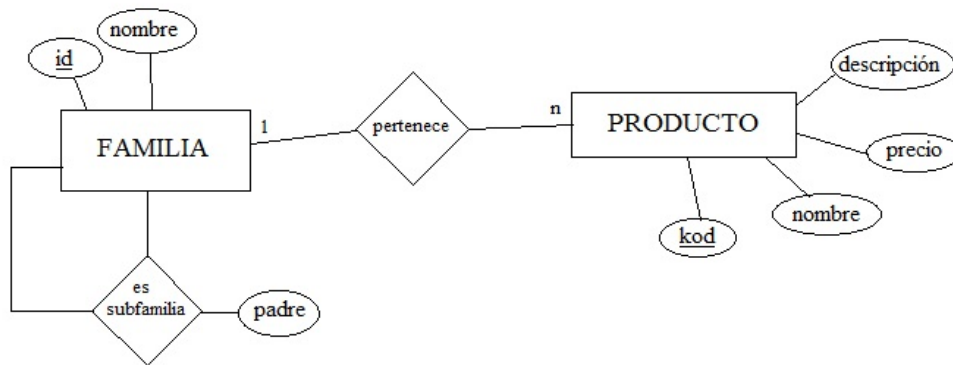


Figura 5.3: Entidad relación

5.2.3. Diseño del algoritmo de carga de datos

Una vez con las bases de datos creadas y las relaciones entre productos, familias y subfamilias definidas analizamos el excel de la empresa para ver de qué modo se deben tratar los datos almacenados en él para la posterior carga en la base de datos.

Familia Raíz	Nivel 1	Nivel 2	Código	Nombre	Precio	Descripción
Fam 1	Fam 2	Fam 3	0000	Prod 1	00	Desc 1
		Fam 4	0000	Prod 2	00	Desc2
	Fam 5		0000	Prod 3	00	Desc3
	Fam 6		0000	Prod 4	00	Desc4
Fam 7			0000	Prod 5	00	Desc5

Tabla 5.1: Ejemplo Excel datos

Tal y como muestra la tabla ejemplo anterior 5.1, aunque no se sabe el número de familias y subfamilias que tiene cada familia raíz solo contamos con 2 niveles de profundidad. Además la estructura del excel tiene definida de una forma bastante ordenada desde la cual se diferencia claramente que subfamilias pertenecen a que familias y cuáles son las familias finales a las que pertenecen los productos. El diseño de la función queda de la siguiente forma:

**Parámetros**

*archivo.excel*

**Declarar**

*nivel0, nivel1, nivel2, famProducto;*

**Empezar**

*ParaCada lineaDelExcel hasta finDocumento*

*Si columna0 != vacio entonces*

*nivel0 = columna0*

*insertarFamiliaBBDD con padre = 0*

*famProducto=nivel0*

*Fin Si*

*Si columna1 != vacio entonces*

*nivel1 = columna1*

*insertarFamiliaBBDD con padre = nivel0*

*famProducto=nivel1*

*Fin Si*

*Si columna2 != vacio entonces*

*nivel2 = columna2*

*insertarFamiliaBBDD con padre = nivel1*

*famProducto=nivel2*

*Fin Si*

*Insertar o actualizar producto con familia= famProducto*

*Fin Para Cada*

**Fin**

Como vemos en el diseño de la función, se va recorriendo cada línea de la hoja de cánculo excel analizando así todos los datos que contiene el archivo. Cada línea pertenece a un producto diferente por lo que es importante saber en todo momento cual es la familia a la que realmente permanece el producto de cada línea.



Pero la función no solo se dedica a insertar cada producto en la tabla de “producto” de la base de datos sino que además inserta las familias de la tabla “familia” manteniendo una estructura ordenada de familias y subfamilias. Por ello conseguimos introducir todos los productos y al mismo tiempo dejarlos organizados en familias, algo que ayuda mucho al usuario final que visita la web ya que la empresa dispone de más de 1500 productos y una buena organización y fácil acceso a los productos ayuda a los usuarios a navegar por la aplicación y encontrar los productos que están buscando.

#### 5.2.4. Resumen

Una vez implementada esta funcionalidad de la aplicación creamos un apartado para usuarios registrados como Administrador desde el cual a través de un formulario pueden subir el archivo excel correspondiente a los productos de la empresa.

Con todo esto se consigue que la empresa solicitante de la aplicación es capaz de gestionar todos los datos referentes a los productos que oferta ordenados de una manera clara y ordenada. Además cualquier responsable de la empresa puede ser capaz de actualizar el contenido de la web ya que solamente *logueándose* en la aplicación con una cuenta de administrador y con el archivo excel haciendo un par de clicks consigue renovar la información que se muestra.

Como ventaja también cuenta que la completa actualización del contenido se realiza en poco menos de 3 segundos, por lo que la completa modificación de todos los datos no supondría mas que un pequeño instante de trabajo, y la aplicación queda actualizada enseguida sin necesidad de deshabilitar momentáneamente la web para realizar trabajos de actualización de información.

Cabe destacar que la empresa solicitante ha quedó muy satisfecha ya que gestionar todo el contenido de la web no les supone ningún trabajo porque el excel que se usa para actualizar la aplicación es el mismo que se usa para la gestión interna de productos, y con la gestión de productos llevada hasta ahora tienen también control total sobre Kanalvet.

### 5.3. Hito 3: Función para navegar entre productos

Llegados a este punto ya tenemos la estructura que tendrá la aplicación, la forma en la que guardaremos los datos de los productos y una funcionalidad que nos permite cargar todos los datos referentes a los productos y sus familias. Pero aún queda una de las cosas más importantes, que es mostrar toda esta información de una manera clara y sencilla desde la cual los usuarios de la web puedan navegar sin complicaciones para encontrar los productos que desean.

#### 5.3.1. Estructura y opciones

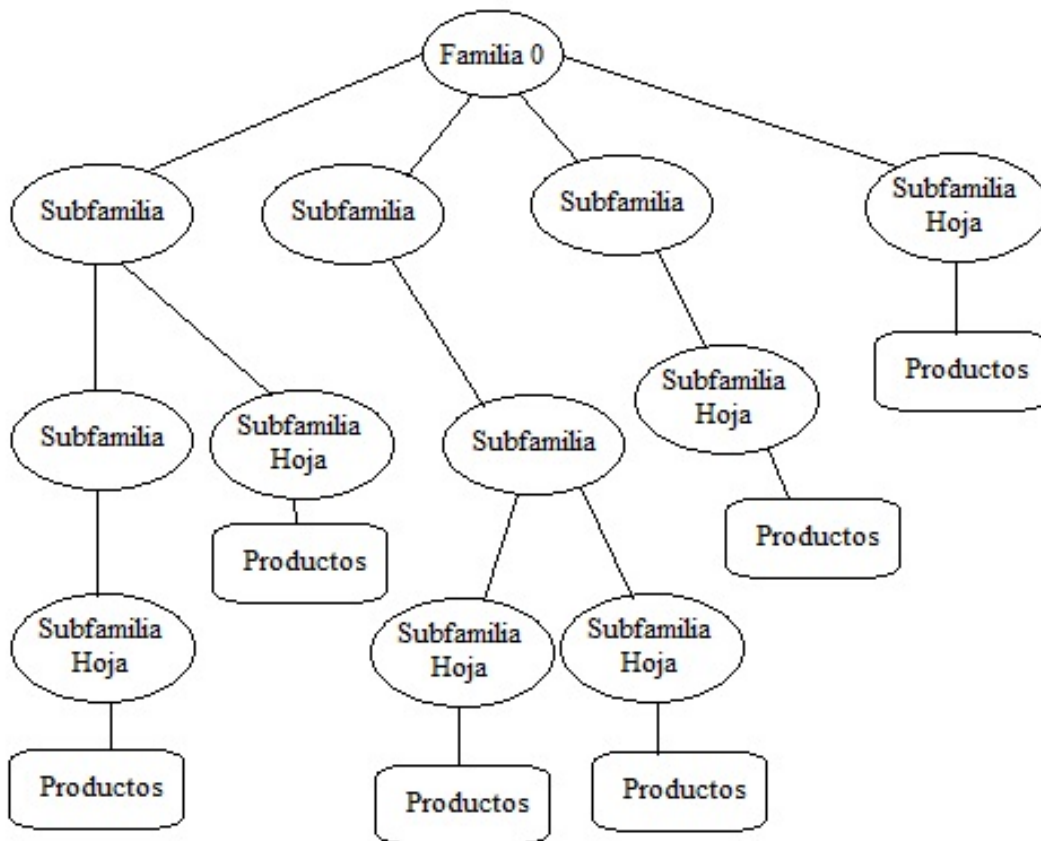


Figura 5.4: Ejemplo estructura árbol

Un detalle muy importante a tener en cuenta la hora de crear la función que mostrará las diferentes familias y los productos que éstas contienen, es la estructura que tiene

nuestra base de datos y como los tenemos almacenados. La estructura de familias con la cual hemos guardado los datos en la base de datos tiene una estructura de “árbol” en la cual podríamos tomar la familia ficticia con id 0 como nodo raíz y los nodos hoja serían familias finales en las que encontramos los productos. En la figura 5.4 observamos un ejemplo de cómo sería la estructura de tipo “árbol” que se ha utilizado para gestionar los diferentes productos en la aplicación Kanalvet.

Una de las opciones para llevar a cabo la visualización de los productos podría ser usar una implementación diferente para cada familia raíz. Pero dado que el número de familias y subfamilias puede variar y aprovechando la estructura de árbol, se optó por desarrollar una función recursiva que dado el id de una familia muestre todas las subfamilias que contiene esa familia y en caso de ser una subfamilia hoja muestre los productos.

El diseño queda de la siguiente manera:

***Función MostrarProductos***

***Parámetros***

*idFamilia*

***Empezar***

*Si idFamilia es SubfamiliaHoja entonces*

*mostrarProductos*

*Si No*

*MostrarSubfamiliasRecursivo(idFamilia)*

*Fin Si*

***Fin***

***Función MostrarSubfamiliasRecursivo***

***Parámetros***

*idFamilia*

***Empezar***

*Si IdFamilia es SubFamiliaHoja entonces*

*mostrarSubfamilia*

*Si no*

*Para Cada Subfamilia de idFamilia*

*MostrarSubfamiliasRecursivo*

*Fin Para Cada*

*Fin si*

***Fin***

Con esta función recursiva podemos navegar por cualquier familia y subfamilia de las que se compone la aplicación de una forma sencilla y reutilizando código. Algo que hace que la aplicación sea muy flexible a cambios ya que tan solo cambiando una función podemos alterar las vistas de productos y familias de las que se compone la aplicación.

## 5.4. Hito 4: Pasarela de pago

Kanalvet, al ser una aplicación web pensada para desarrollar una actividad empresarial de venta de productos se propuso una forma de automatizar el proceso de cobro de los productos. Para automatizar el proceso de cobro se utiliza lo que se llama “pasarela de pago” que se encargan de mover dinero de la cuenta bancaria del comprador a la del vendedor. Las pasarelas de pago usadas suelen ser de Visa, MasterCard y PayPal. Nosotros nos decantamos por esta última ya que PayPal es un servicio que cada día se usa más y que da ciertas garantías al comprador, además era la única pasarela de pago que contaba con un *sandbox* o banco de pruebas desde el cual realizar pruebas antes de realizar la implementación final.

### 5.4.1. Pasarela de pago con PayPal

Realizar una pasarela de pago con PayPal es una tarea relativamente sencilla ya que PayPal tanto con el banco de pruebas como con información muy detallada da muchas ayudas a los desarrolladores para llevar a cabo la pasarela.

La base de la pasarela se basa en enviar a la cuenta de PayPal del vendedor un formulario con ciertos valores. La cantidad de valores que se le envía depende de las necesidades de cada uno, pueden enviarse formularios solo con el valor total de la compra, separando cada artículo y añadiéndole IVA o descuento a cada uno o añadiéndole IVA o descuentos al global... así que nos da un gran abanico de posibilidades para mostrarle al usuario en la página de pago de PayPal el precio de los artículos. Los únicos valores fijos que hay que enviarle son precio total y moneda en la que se envía el precio. Un ejemplo de formulario de pago para PayPal podría ser el siguiente:

```
<form action="https://www.sandbox.paypal.com/cgi-bin/webscr"
method="post">

<input name="cmd" type="hidden" value="_cart">

<input name="upload" type="hidden" value="1">
<input name="business" type="hidden" value="tucodigodevendedor">
<input name="shopping_url" type="hidden" value="ladirecciondetuweb">
<input name="currency_code" type="hidden" value="EUR">
<input name="return" type="hidden"
value="http://mipagina.com/mensaje_de_exito">
<input name="notify_url" type="hidden"
value="http://mipagina.com/paypal_ipn.php">
<input name="rm" type="hidden" value="2">
#GALLETAS5 ; Nombre: Galletas ; Valor : 5.00 , Cantidad : 2
<input name="item_number_1" type="hidden" value="GALLETAS5">
<input name="item_name_1" type="hidden" value="Galleta">
<input name="amount_1" type="hidden" value="5.00">
<input name="quantity_1" type="hidden" value="2">
#PIRULETAS3 ; Nombre: Piruletas ; Valor : 3.00 , Cantidad : 5
<input name="item_number_2" type="hidden" value="PIRULETAS3">
<input name="item_name_2" type="hidden" value="Piruletas">
<input name="amount_2" type="hidden" value="3.00">
<input name="quantity_2" type="hidden" value="5">
<input type="submit" value="PayPal SandBox">

</form>
```

Con esto tendríamos creada nuestra pasarela de pago pero nos planteaba un par de problemas: No parecía una forma muy segura la de mostrar todos los datos en el código HTML y que todo el mundo pudiera verlo, y que de esta manera desde la web no teníamos forma automática de saber si el pago se había realizado o no.

#### 5.4.1.1. Solución problema 1: Seguridad

El problema de la seguridad en el pago es el que más nos importaba ya que con un formulario a la vista de todo el mundo puede intentar realizar algún tipo de engaño. Para ello PayPal nos ofrece una clave pública con la cual usando el algoritmo RSA (ver sección 3.2.1.1) fuimos capaces de encriptar todo el contenido del formulario y enviarlo completamente encriptado a nuestra cuenta de vendedor de PayPal.

Para mayor seguridad a parte de enviar encriptados todos los datos referentes a la compra de cada usuario, configuramos la cuenta de vendedor de PayPal para que solo aceptara peticiones de pago encriptadas.

#### 5.4.1.2. Solución problema 2: Automatización

Llegados a este punto contábamos con una pasarela de pago funcional y segura que permitía hacer pagos de pedidos con total seguridad, pero no teníamos un sistema automático, ya que una vez enviado el formulario encriptado a PayPal no sabíamos nada sobre el estado del mismo. Una vez enviado el formulario el pedido podía ser cancelado, podía surgir errores o podría llevarse a cabo con total normalidad pero desde la aplicación no teníamos forma de saberlo por lo que eso era tarea de gestión de Kanalvet.

Para solucionar esto activamos otra de las muchas características de PayPal llamada IPN (Instant Payment Notification). Al activar la opción IPN PayPal nos devuelve información referente al estado del pago a una página web que nosotros le indicamos, desde la cual podemos controlar automáticamente el proceso de pago.

En caso de error o de que el usuario cancele el pago, esa información llegara a Kanalvet y lo procesaría indicando al usuario que su pago no se ha realizado. En caso de que el pago fuera correcto, IPN enviaría todos los datos a Kanalvet donde se procesaría la respuesta y en caso de que todo este correcto se le envía de nuevo a PayPal una confirmación para que se lleve a cabo el traspaso de dinero de una cuenta a otra. Si algo fallara en este proceso de confirmación el pago sería negativo y no se llevaría a cabo el traspaso de dinero.

Por seguridad la información que devuelve el IPN de PayPal también nos la envía encriptada para lo cual según el algoritmo RSA debemos generar nuestra clave pública y subirla a la cuenta de PayPal para que este pueda encriptar la información que nos va a enviar y que más tarde desencriptaremos para verificarla.

5.4.1.3. Resumen

Resumiendo tenemos una pasarela de pago segura encriptada según el algoritmo RSA y automatizada 100% que nos garantiza que siempre sabremos si el pago de los productos se ha realizado correctamente. Y tenemos la certeza de que esta pasarela de pago funciona perfectamente gracias al banco de pruebas que nos brinda la posibilidad de hacer todo tipo de pruebas con dinero ficticio antes de implantar la pasarela de pago final en un sistema real.





---

## Gestión del proyecto

En este capítulo se analizan las desviaciones que se han producido en cuanto a coste y trabajo respecto a la planificación inicial del proyecto.

### 6.1. Retrasos

Los retrasos en el desarrollo del proyecto se han producido principalmente por encontrar dificultades al programar, debido al desconocimiento por parte del desarrollador de las principales herramientas que se utilizan en la implementación de la aplicación. Aunque otro gran causante de retrasos en el proyecto han sido los continuos cambios por parte de la empresa en lo referente a la interfaz gráfica. Aunque en un principio se especificó bien lo que se quería y la empresa tenía muy claro lo que buscaba en un principio, a medida que la interfaz iba tomando forma lo especificado inicialmente no gustó mucho y hubo que hacer muchos cambios en todo lo referente a la interfaz gráfica y el modo en el que los diferentes contenidos y opciones se mostraban por pantalla.

También podría llegar a tomarse como un retraso funcionalidades que se implementaron pero que finalmente se descartaron en la aplicación final. Así es el caso de la pasarela de pago con PayPal. Esta se desarrollo e implemento según lo planificado pero al final por motivos externos a la implementación realizada la empresa decidió que prefería contactar directamente con los compradores interesados. Esto se debe a que, al vender productos para clínicas veterinarias se podrían realizar pedidos de sumas muy altas, y la empresa prefiere realizar todos los tramites con personas físicas para evitar cualquier tipo de engaño.

## 6.2. Horas Planificadas vs Horas reales

En la tabla 6.1 se puede ver la variación existente entre las horas estimadas para cada una de las tareas con las horas reales.

Nombre de tarea	Trabajo estimado	Trabajo real	Desviación
Preparación del proyecto	16 horas	13 horas	-3 horas
Objetivos del proyecto	8 horas	8 horas	8 horas
Instalación de herramientas	8 horas	5 horas	-3 horas
Formación	43 horas	43 horas	0 horas
Formación PHP	24 horas	24 horas	0 horas
Formación WordPress	9 horas	9 horas	0 horas
Formación GIMP 2.6	10 horas	10 horas	0 horas
Análisis de los requisitos del proyecto	10 horas	10 horas	0 horas
Diseño de la arquitectura general del proyecto	14 horas	12 horas	-2 horas
Diseño de la base de datos	4 horas	4 horas	0 horas
Implementación de la base de datos	4 horas	6 horas	2 horas
Interfaz Gráfica	100 horas	180 horas	80 horas
Diseño de la interfaz	40 horas	50 horas	10 horas
Implementación de la interfaz	60 horas	130 horas	70 horas
Login	38 horas	45 horas	7 horas
Diseño login	16 horas	16 horas	0 horas
Implementación login	22 horas	29 horas	7 horas
Módulo Usuario	45 horas	65 horas	10 horas
Función Carro Compra	30 horas	35 horas	5 horas
Función Seguimiento Pedidos	15 horas	20 horas	5 horas
módulo administrador	122 horas	178 horas	56 horas
Función activar usuarios	16 horas	24 horas	8 horas
Función Control pedidos	16 horas	34 horas	18 horas
Función gestión productos	90 horas	120 horas	30 horas
Pruebas de control	30 horas	30 horas	0 horas
Funcionalidades Extra	80 horas	95 horas	15 horas
Pasarela pago paypal	80 horas	95 horas	15 horas
Pruebas finales y resolución de errores	30 horas	30 horas	0 horas
Memoria del proyecto	96 horas	96 horas	0 horas
Redactar memoria	70 horas	70 horas	0 horas
Elaboración de la presentación	16 horas	16 horas	0 horas
Preparación de la defensa	10 horas	10 horas	0 horas
Entrega de la documentación	0 horas	0 horas	0 horas
<b>Total</b>	<b>632 horas</b>	<b>817 horas</b>	<b>185 horas</b>

Tabla 6.1: Horas planificadas vs horas reales

**Horas Planificadas: 632 horas    Horas reales: 817 horas    Desviación: +185horas**

### 6.3. Costes planificados vs Costes reales

Debido al incremento de horas en la realización del proyecto en comparación con las horas planificadas, se ha optado por alargar la duración del proyecto en vez de realizar horas extras, con el único fin de abaratar costes.

Concepto	Coste Unitario	tiempo de amortización	Coste unitario de amortización	Tiempo de uso	Importe
PC Sobre mesa	478,90 €	4800	0,09977 €	665h	66.34 €
MS Windows 7 Ultimate	319,00 €	4800	0,06645 €	21h	1,34 €
MS Project 2010	755,00 €	4800	0,16146 €	15h	2,42 €
MS Visio 2010	330,00 €	4800	0,06875 €	6h	0,41 €
Antivirus Nod32	39,95 €	4800	0,00832 €	21h	0,17 €
Sublime Text 2	40,00 €	4800	0,00833 €	70h	0,58 €
Eclipse 3.6	0,00 €	4800	0,00000 €	31.5h	0,00 €
Gimp 2.6 3.6	0,00 €	4800	0,00000 €	20h	0,00 €
Filezilla	0,00 €	4800	0,00000 €	20h	0,00 €
<b>Total</b>					<b>71,26 €</b>

Tabla 6.2: Amortizaciones reales

Concepto	Trabajo	Coste	Importe
Analista	100 horas	18,30€	1.830,00€
Programador	717 horas	5,30 €	3.800,10 €
<b>Total</b>			<b>5.630,10 €</b>

Tabla 6.3: Coste real de los recursos de trabajo.

Los costes de los recursos materiales se mantienen igual que en la planificación ya que su coste se calcula por uso y no ha sido necesario modificar la asignación de recursos de las tareas.

**Coste real de recursos materiales: 27,00 €**

Concepto	Planificado	Real	Desviación
Recursos de trabajo	4.545,60 €	5.630,10 €	1.084,50 €
Recursos materiales	27,00 €	27,00 €	0,00 €
Amortizaciones	52,77 €	71,26 €	18,49 €
<b>Total</b>	<b>4.625,37€</b>	5.728,36 €	1.102,99 €
Gastos Generales(10%)	462,53 €	572,83 €	110,30 €
Beneficio(15%)	693,80 €	859,25 €	165,45 €
<b>Subtotal</b>	<b>5.781,70 €</b>	<b>7.160,44 €</b>	<b>1.378,74 €</b>
IVA(18%)	1.040,70 €	1.288,87 €	248,17 €
<b>Total</b>	<b>6.822,40 €</b>	<b>8.449,31 €</b>	<b>1.626,91 €</b>

Tabla 6.4: Costes planificados vs costes reales

Finalmente como muestra la tabla 6.4 los retrasos en el desarrollo del proyecto respecto al tiempo planificado suponen un aumento del coste final de 1.626,91 €, resultando en un coste total de 8.449,31 €

## **Parte IV**

### **Conclusiones y trabajo futuro**



---

## Conclusiones y líneas futuras

Este capítulo muestra la conclusión a la que se ha llegado sobre la aplicación desarrollada, y comenta las líneas de trabajo futuro que puede llegar a tener.

### 7.1. Conclusiones

Teniendo en cuenta que Kanalvet es una aplicación web orientada a la venta online de productos, de las cuales hoy en día existen casi en cualquier empresa o negocio ya que es un formato que cada día más en auge, es una aplicación muy completa y que cumple todas las expectativas del cliente.

Kanalvet es una aplicación que se ha diseñado pensando mucho en el usuario final, intentando que la navegación por la página se haga de forma sencilla, visual y sin muchas complicaciones ya que un usuario que busca realizar una compra a través de internet no tiene porque tener conocimientos avanzados sobre el medio y cuantas más facilidades se le dé y mejor sea la experiencia de usuario es más fácil conseguir compradores potenciales que llegan a la web de forma casual. Por eso se le ha dado mucha importancia a la parte visual de la web, para intentar atraer al mayor número de compradores de una forma visual y luego conseguir mantenerlos con un diseño elegante y fácil de usar unido con una web ágil y que funciona sin problemas.

Pero no solo se ha diseñado pensando en el usuario final si no también pensando en la gestión de la página, ya que al tener muchos productos diferentes podría llegar a volverse algo caótico sino se le hubiera dado mucha importancia al apartado de gestión de la aplicación.

Por lo que en conclusión y como opinión propia y personal creo que se ha construido una aplicación visualmente atractiva fácil de usar, muy fácil de gestionar y que cumple con todas las expectativas que se habían planteado desde un principio. Y aunque ha habido retrasos importantes a la hora de desarrollar tampoco han sido muy determinantes y la posibilidad por parte del cliente de retrasar la fecha final de entrega de la aplicación nos ha permitido crear una web terminada, bien testada y sin errores.

## 7.2. Líneas futuras

Kanalvet es un proyecto bastante terminado y con bastantes funcionalidades , aun así todavía puede mejorar un poco añadiéndole nuevas funcionalidades como pueden ser:

- Pasarela de Pago: La aplicación está muy bien construida pero no incorpora una pasarela de pago por la cual poder realizar el pago de los pedidos realizados. Ahora mismo el pago de los pedidos de una forma más personal a través del contacto con el comprador pero la facilidad de uso y comodidad que da el contar con una pasarela de pago en este tipo de aplicaciones es muy importante para este tipo de negocios. Y aunque se desarrolló esta funcionalidad, en el último momento se decidió no implementarla en la versión final de la aplicación y creo que es de las pocas cosas que le falta a la web y algo muy a tener en cuenta para implementar en un futuro.
- Control de stock de productos: Un control de stock en la web proporcionaría al usuario final que va a comprar un producto en concreto saber antes de realizar la compra si el producto que desea comprar está en stock o no, y así saber si su pedido llegará en un tiempo relativamente corto o si tendrá que esperar a que se reponga el stock de ese producto. La consulta de stock actualmente se puede hacer llamando al teléfono gratuito que se muestra en la web o realizando una consulta mediante los formularios que disponen todos los productos, pero una funcionalidad que fuera capaz de mostrar el stock de productos sería algo a tener en cuenta.



## **Parte V**

### **Apéndices y bibliografía**



---

## Pliego de condiciones

### A.1. Condiciones del cliente

Este apartado hace alusión a las condiciones mínimas requeridas para hacer uso de la aplicación. Las características mínimas que se requieren en el ordenador para hacer uso del servicio son las siguientes:

- Pentium 233 MHz (Recomendado: Pentium 500MHz o superior)
- 64 MB RAM (Recomendado: 128 MB RAM o más)
- 52 MB de espacio en disco
- Monitor de 256 colores con capacidad para mostrar una resolución de 800x600 píxeles, aunque se recomienda una resolución de 1024x768 píxeles.

Ratón: Es recomendable su utilización para una mayor comodidad del usuario, aunque su uso no es obligatorio.

Monitor: La aplicación ha sido realizada en una resolución de 1024x768 píxeles, luego esta será la resolución recomendada para una mejor visualización de la aplicación, aunque no es obligatorio.

### A.2. Condiciones del servidor y otros elementos hardware

Las características recomendadas para el ordenador que ejerza de servidor y para un correcto funcionamiento son los siguientes:

- Un mínimo de un servidor con dos o más procesadores cada uno.
- Los servidores deben soportar todos los estándares de Internet como TCP/IP, DNS, NTP, DHCP entre otros.
- Microprocesadores de 1.3 GHz o superior.
- Memoria caché L2/L3 de 3MB o superior por procesador.
- 2GB de memoria RAM.

### A.3. Condiciones del software

**Cliente:**

- Navegador Web (IE6+, Firefox, Chrome, Opera, Safari)

**Servidor:**

- Apache 2
- MySQL 5.1
- PHP 4

### A.4. Condiciones generales

La presente aplicación ha sido diseñada como una propuesta de la empresa I3 aplicaciones informáticas. Con ella se ha pretendido diseñar una aplicación útil y amigable para poder realizar venta de productos online de forma que la gestión y actualización de la misma pueda realizarse de una forma automatizada.

Se busca facilitar el trabajo del usuario final, que tendrá a su disposición una aplicación en la que podrá realizar búsquedas de productos de forma sencilla para la posterior compra.

Estimando que los datos aportados en el presente pliego de condiciones son suficientes, se somete el proyecto a su aprobación por la empresa Kanalvet.

Vitoria-Gasteiz, a 14 de Mayo de 2010



---

## Bibliografía

ALVAREZ, M. A., 2001a. Qué es HTML. URL <http://www.desarrolloweb.com/articulos/que-es-html.html>.

ALVAREZ, M. A., 2001b. Qué es PHP. URL <http://www.desarrolloweb.com/articulos/392.php/>.

ALVAREZ, M. A., 2009. Introducción a jQuery. URL <http://www.desarrolloweb.com/articulos/introduccion-jquery.html>.

CIKAMELO, 2010. Microsoft Visio. URL <http://www.slideshare.net/cikamelomicrosoft-visio-4502404>.

F.MANUEL, 2012. Sublime Text, un sofisticado editor de código multiplataforma. URL <http://www.genbeta.com/herramientas/sublime-text-un-sofisticado-editor-de-codigo-multiplataforma>.

SERVER, I., 2009. ¿Que es WordPress? URL <http://islaserver.com/articulos/aplicaciones/que-es-wordpress.html>.

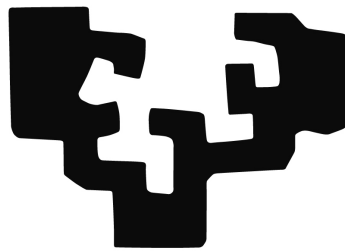
SUR, C., 2010. Herramientas utilizadas: GIMP. URL <http://www.codigosur.org/staticpage/herramientas-utilizadas/>.

VALDÉS, D. P., 2007. ¿Qué es JavaScript? URL <http://www.maestrosdelweb.com/editorial/pt/>.

WIKIPEDIA, 2012a. Criptografía simétrica. URL <http://es.wikipedia.org/wiki/Criptograf>.

- WIKIPEDIA, 2012b. Eclipse (software). URL [http://es.wikipedia.org/wiki/Eclipse\\_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software)).
- WIKIPEDIA, 2012c. Hojas de estilo en cascada. URL [http://es.wikipedia.org/wiki/Hojas\\_de\\_estilo\\_en\\_cascada](http://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada).
- WIKIPEDIA, 2012d. LaTeX. URL <http://es.wikipedia.org/wiki/LaTeX>.
- WIKIPEDIA, 2012e. Microsoft Project. URL [http://es.wikipedia.org/wiki/Microsoft\\_Project](http://es.wikipedia.org/wiki/Microsoft_Project).
- WIKIPEDIA, 2012f. Modelo Vista Controlador. URL [http://es.wikipedia.org/wiki/Modelo\\_Vista\\_Controlador](http://es.wikipedia.org/wiki/Modelo_Vista_Controlador).
- WIKIPEDIA, 2012g. RSA. URL <http://es.wikipedia.org/wiki/RSA>.
- WIKIPEDIA, 2012h. SQL. URL <http://es.wikipedia.org/wiki/SQL>.

eman ta zabal zazu



Universidad  
del País Vasco

Euskal Herriko  
Unibertsitatea