

Título del Trabajo  
Fin de Grado

Grado en Ingeniería Multimedia



Trabajo Fin de Grado

Autor:

Francisco Ortega Guardia

Tutor/es:

Manolo Marco Such

Enero 2016



# Índice

1. Introducción
   1. Motivaciones
   2. Introducción al problema
   3. Definiciones, abreviaturas y acrónimos.
   4. Objetivos
   5. Relación con asignaturas
2. Análisis de la solución
   1. Análisis de Soluciones actuales
      1. Shopify
      2. Magento
      3. Prestashop
      4. WooComerce (Wordpress + paypal)
   2. Perfil del cliente
   3. Aproximación al problema
3. Modelo de negocio
4. Especificación de requisitos
   1. Casos de Uso
      1. Casos de uso Tienda Online
      2. Casos de uso Panel Control
   2. Requisitos funcionales
      1. Casos de uso Tienda Online
      2. Casos de uso Panel Control
   3. Requisitos no funcionales
   4. Requisitos de diseño
5. Estudio de viabilidad
   1. Planificación temporal
      1. Hitos
      2. Estimación de costes
   2. Estimación de riesgos
      1. Riesgos de tecnología
      2. Riesgos personales
      3. Riesgos estimaciones
6. Metodología
   1. Metodología de desarrollo
   2. Planificación y gestión
   3. Documentación
   4. Diseño
   5. Desarrollo
7. Diseño del sistema
   1. Diseño Interfaz
      1. Diseño Ideal vs Diseños comunes
      2. Diseño Inicial
      3. Evolución del diseño
      4. Diseño Final
   2. Diseño de la base de datos
   3. Diseño de la arquitectura
8. Desarrollo del sistema
   1. Desarrollo Hito 1
   2. Desarrollo Hito 2
9. Difusión del sistema
   1. Hosting
   2. SEO
   3. Redes sociales
10. Ley Orgánica de protección de datos
11. Problemas, errores y dificultades
12. Conclusión
    1. Conclusión sobre el sistema
    2. Conclusión sobre los objetivos marcados
    3. Mejoras y línea de futuro
13. Bibliografía
    1. Índice de figuras

# Introducción

## 1.1 Motivaciones

Desde el comienzo y durante el grado se nos ha dicho que el ingeniero tendrá un rol muy importante en el mundo laboral. Es cierto que nuestro rol puede llegar a ser primordial en dicho mundo dado que deberíamos ser capaces de dominar y poder actuar en muchos ámbitos de la tecnología social moderna. Esto es teóricamente cierto, pero honestamente considero que las expectativas son demasiado altas para el nivel que se ha exigido en el grado en innumerables ocasiones. La formación que nos han dado en algunos ámbitos es compleja y completamente apta para el mundo real pero hay otros espacios de conocimiento que han sido bastante deficientes ya sea por profesorado, por organización o por contenidos coincidiendo estos con el itinerario de gestión de contenidos que son el que yo he cursado. Una de mis motivaciones pues es la de comprobar cuan capacitado estoy para realizar el siguiente proyecto cuya dificultad y complejidad no es ni mucho menos excesiva, aunque tal vez su cantidad de trabajo para el tiempo disponible si lo sea. Hablaré de esto más adelante en el apartado de “Problemas, errores y dificultades” y en el de “conclusión”.

El método de desarrollo seguido casi en su absoluta mayoría durante el grado ha sido la programación desde cero sin ‘frameworks’. A la hora de desarrollar un sistema tan grande, como el que se planea realizar en este proyecto de final de grado, me parece muy interesante realizarlo siguiendo este procedimiento para poder ahondar más en la base del funcionamiento del sistema con el objeto estudiar qué puede mejorar mediante el uso de ‘frameworks’ y qué partes del sistema es mejor programar desde la capa más baja de la arquitectura. En síntesis, otra de mis motivaciones para realizar este proyecto es tomarlo como un estudio de las facilidades que proporcionan los ‘frameworks’ versus la programación base.

Por último, otra de las motivaciones es el cerrar un ciclo. El ciclo de estudio en el grado de Ingeniería Multimedia para poder abrir un nuevo ciclo a otros proyectos y metas personales.

## 1.2 Introducción al problema

No hace falta decir que desde hace unos años el internet se ha convertido en una pieza clave e indispensable en nuestras vidas. Gracias a él podemos comunicarnos de forma instantánea así como compartir nuestras ideas y nuestros proyectos con todo el mundo entre otras muchas aplicaciones. Es esta arista de internet la que han empleado los negocios para darse a conocer, captar y fidelizar clientes. Abrirnos paso en internet para colar nuestra empresa o negocio es uno de los factores esenciales para su éxito. La posibilidad de comprar mediante un solo click del ratón ha hecho posible y muy necesario que la mayoría de los comercios importantes dispongan de una página web para ofrecer sus servicios a su ya vasto público.

Pero, ¿qué hay del pequeño empresario que comienza un negocio y quiere dar sus primeros pasos en la red? Estos parten de un público mínimo, muy local o incluso nulo, de modo que a no ser que dispongan de un presupuesto para una campaña de publicidad deberán difundirse mediante el boca a boca.

En este proyecto vamos a dar forma a una tienda online sencilla para un cliente que acaba de estrenar un pequeño negocio.

## 1.3 Definiciones, abreviaciones y acrónimos

* Cliente: Persona o entidad que ha hecho el encargo de la plataforma.
* Usuario: Persona o entidad que en última instancia hará uso de nuestra plataforma para navegar o realizar sus compras.
* Framework: “Desde el punto de vista del desarrollo de software, un framework es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado.”-- Leandro Alegsa en <http://www.alegsa.com.ar/Dic/framework.php>
* Hosting: Espacio virtual en el que alojar nuestra página web.
* Domain / Dominio: Nombre o dirección por la que identificamos un grupo de dispositivos o equipos conectados a internet.
* User-friendly: Del Inglés, familiar para el usuario. Es decir, sencillo para el usuario para usar o manipular.
* SEO: Del inglés, Search Engine Optimization (Optimización del motor de búsqueda). Proceso técnico por el cual se realizan cambios en la estructura de los sitios web para lograr una mayor visibilidad y posicionamiento en los buscadores de internet.
* SEO-friendly: Referente al SEO, fácil de analizar para el SEO y que permite a éste tomar un sistema como adecuado para una mayor visibilidad.
* Self-Hosting: Técnica para convertir un ordenador personal en un host. Dicho de otra forma, convertir el ordenador de uno mismo en el host que almacenará el sitio web.
* Template: Plantilla de diseño predefinida.
* Crons: Procesos ejecutados automáticamente cada X tiempo.
* Logueado: Ha comenzado una sesión.
* Debug: Procesado de errores.

## 1.4 Objetivos

En síntesis los objetivos principales se pueden dos grandes bloques:

* Objetivos Formativos
  + Poner a prueba las capacidades obtenidas durante mi formación en el grado aplicando los conocimientos obtenidos para la resolución del problema.
  + Estudio de la ventaja de uso de frameworks y plataformas predefinidas sobre la programación base del sistema.
  + Estudio de la programación del sitio web “a pelo”.
  + Aplicación de metodologías de desarrollo ágil.
  + Adquisición de experiencia en herramientas no conocidas.
  + Seguimiento de la planificación y capacidad de improvisación ante problemas no contemplados en la planificación.
* Objetivos Profesionales
  + Desarrollar la plataforma encomendada.
  + Investigar y proporcionar al cliente información sobre los pagos adicionales que deberá afrontar (hosting, domain, etc.)
  + Proporcionar la mejor experiencia de usuario posible
    - Mediante una interfaz limpia e intuitiva
    - Mediante la correcta visualización en dispositivos móviles
    - Mediante una buena usabilidad
    - Tiempos de carga aceptables
    - Protección de sus datos
  + Permitir la difusión mediante redes sociales
  + Desarrollar el sistema de modo que sea tanto user-friendly como SEO-friendly para lograr un posicionamiento óptimo.
  + Minimizar el esfuerzo, la dificultad y los conocimientos avanzados y específicos necesarios para operar la administración del sistema al cliente.
  + Permitir la adición de módulos de funcionalidad al sistema.
  + Minimizar la intervención del administrador en algunos ámbitos del sistema.

## 1.5 Relación con asignaturas

Atendiendo al objetivo formativo número uno (“[Poner a prueba las capacidades obtenidas durante mi formación en el grado aplicando los conocimientos obtenidos para la resolución del problema.](#_Objetivos)“) voy a proceder a distinguir algunos de los bloques de trabajo realizados o por realizar con las asignaturas a las que han correspondido durante el grado.

* *Fundamentos del diseño gráfico (FDG)­ –* Análisis y diseño inicial del sitio web.
* *Fundamentos de las bases de datos (FDB)* y *Diseño de bases de datos Multimedia (DBDM)* — Análisis de lo solicitado por el cliente en cuanto a estructuras de datos se refiere. Diseño y construcción de la base de datos utilizada en la tienda y en el panel de control.
* *Análisis y especificación de sistemas multimedia* (AESM) – Adquisición y organización de los requisitos del proyecto.
* *Diseño de sistemas multimedia (DSM) –* Análisis y estructuración de los posibles patrones a implementar en el sistema.
* *Programación hipermedia I (PH1) y Programación Hipermedia 2 (PH2) –*  Uso de las tecnologías principales de este proyecto: HTML5, CSS3, JS (Javascript), PHP.
* *Servicios Multimedia Basados en Internet (SMBI) –* Análisis de la posibilidad de self-hosting y procesos en servidor.
* *Servicios Multimedia Avanzados (SMA) –* Aproximación a la solución del problema. Estructura general del sistema. Corrección de bugs y errores. Uso de APIs. Diseño, estructuración y manipulación de base de datos.
* *Negocios y Multimedia (NM) –* Adaptaciones del sistema para el SEO y GOOGLE Adsense y GOOGLE Analytics.

# Análisis de la solución

## 2.1 Análisis de las soluciones actuales

### Shopify Releases Revolutionary New Online Store-Building Platform2.1.1 Shopify

Figura 2.1.1.0 – Icono y logo de Shopify

#### Ventajas:

* Múltiples canales de venta – Shopify permite la venta a través de diferentes canales como pueden ser Facebook o Wordpress así como otros métodos más tradicionales como las tienda online o los blogs + tienda.
* SEO – Permite la optimización del SEO y ofrece ayudas sobre el así como para AdSense o GoogleAnalytics.
* Nube – Alojado en la nube nos ahorra problemas de self-hosting.

#### Inconvenientes:

* Cuotas – Exige una cuota mensual así como una comisión por operación comercial realizada aunque no finalice.
* Limitación del idioma: Administración está en inglés y su uso no se ajusta completa ni perfectamente a otros idiomas como el Castellano, dejando palabras o incluso páginas traducidas de forma incompleta que pueden interferir en la usabilidad de los usuarios.
* Externa a nuestra plataforma - A pesar de que cuenta con widgets y códigos para insertar objetos en nuestra página, la inserción completa en nuestra página y el control de los objetos puede llegar a ser complejo.

#### Conclusión

Las ventajas de esta plataforma son extraordinariamente útiles pero inconvenientes son muy difíciles de resolver. El cliente pierde dinero con las ventas ya que shopify se lleva una comisión por venta. Añadido a la falta de público inicialmente, el inicio sería demasiado complicado para el cliente.

### 2.1.2 Magento



Figura 2.1.2.0 – Icono y logo de Magento

#### Ventajas

* Código abierto – Todo el mundo puede desarrollar para esta plataforma por lo que las tiendas implementadas con Magento.
* Personalización avanzada – Al existir una gran cantidad de plugins que se podrían implementar o incluso auto-desarrollar para mejorar o añadir las funcionalidades de nuestra tienda las posibilidades son infinitas.
* Idiomas – A diferencia de la anterior opción, Magento tiene disponibles varios idiomas.
* SEO-Friendly – Permite la optimización del SEO.

#### Incovenientes

* Requiere de un presupuesto considerable
* Requiere un alto nivel de programación – Requiere de un equipo de gestión o un alto nivel de programación.

#### Conclusión

Dado el alto nivel de programación ya es una plataforma que queda descartada completamente. Aun así, atendiendo a sus otras características vemos que el presupuesto puede no salir rentable al cliente.

### 2.1.3 Prestashop



Figura 2.1.3.0 – Icono y logo de Prestashop

#### Ventajas

* Personalización Avanzada – Al igual que Magento, tener código abierto hace que sea una herramienta muy configurable.
* Fácil de Instalar – Muchos de los servicios de hosting actuales disponen de herramientas para instalarlo en un solo click.
* Idiomas – Varios idiomas a elegir.
* SEO-Friendly – Permite la optimización del SEO.

#### Inconvenientes

* Costes – Se debe tener en cuenta el coste requerido para el soporte técnico de prestashop así como para las actualizaciones de software. Además muchos de los plugins son de pago mientras que en otras plataformas serían gratuitos.
* Complicada Gestion – Prestashop es literalmente demasiado completo. El panel de administrador es un poco complicado de gestionar.
* Conocimientos Medios – Requiere de conocimientos medios en varias tecnologías (HTML o CSS3) para personalizar algunas de las características que nos ofrece.
* Los tiempos de lanzamiento son largos.

#### Conclusión

Prestashop es la herramienta ideal para una empresa con un presupuesto desahogado que pueda disponer de un encargado con conocimientos suficientes para manipular la plataforma sin problemas. Dispone de muchas opciones en su panel de gestión para controlar absolutamente todas las características propias de una tienda online y más. Sin quedarse corta, ofrece también la posibilidad de hacer cosas sorprendentes con un poco de HTML5 y CSS3. Lamentablemente esta opción queda descartada dada la imposibilidad de mantener a una persona a cargo de la plataforma por parte del cliente.

### 2.1.4 WordPress + Paypal

Figura 2.1.4.1 – Icono Wordpress y Figura 2.1.4.2 – Icono Paypal

Existe un plugin conocido como WooComerce que ofrece una solución que puede resumirse como WordPress + Paypal. Vamos a conocer sus ventajas y desventajas

#### Ventajas

* Gratuito
* Sencillo – Con base Wordpress, WooComerce es muy sencillo de instalar, emplear y de modificar.
* Paypal – Pasarela de pago con Paypal, una de las más recomendadas por los usuarios por su fiabilidad y sencillez.

#### Inconvenientes

* Las funcionalidades que permite WooComerce son muy sencillas y básicas.
* Limitación de artículos – No pueden instalarse muchos productos con este plugin de Wordpress.
* Incompatibilidades – Algunas incompatibilidades con temas visuales ofrecidos por Wordpress.

#### Conclusión

La conclusión que podemos sacar de este plugin es que es una de las mejores opciones para darse a conocer o que planean tener muy pocos artículos en venta. Sus limitaciones impiden desarrollar una web como la que se nos ha exigido, pero su sencillez permite no disponer de ningún tercero al cargo del sistema y su precio obviamente es inmejorable. En cuanto a nuestro proyecto, es una opción que no utilizaremos pero que bien podemos tomar como referencia. Intentaremos plasmar en nuestra arquitectura las características de WooComerce así como la pasarela de pago Paypal.

## 2.2 Perfil del cliente

El perfil actual de nuestro cliente es, mayoritariamente, un reflejo de lo que nos está pidiendo desarrollar. Es muy importante analizar este perfil para poder comprender y estimar muchos factores durante el desarrollo. A continuación tenemos datos del cliente que comentaremos posteriormente:

* Sexo: Varón.
* Edad: 40-45 años.
* Ocupación: Sector agrario y agrícola.
* Presupuesto: Medio-Bajo.
* Estudios: Básicos.
* Idiomas:
  + Castellano: nativo.
  + Valenciano: muy bajo.
* Conocimientos de informática (programación, diseño, etc): Ningunos.
* Manejo del ordenador: Muy bajo – Bajo.
* Manejo general de las tecnologías: Muy bajo.

El conjunto de datos representados nos indica que es una persona que se crió sin estudios y dedicándose desde temprana edad al trabajo, probablemente en el mismo sector al que se dedica actualmente. Podemos observar también que este cliente probablemente cae en el segmento de aquellas personas que han llevado un negocio durante mucho tiempo pero que ahora por circunstancias (favorables o desfavorables) quieren probar o se ven forzados a expandirse a un sector no tan local como al que estaban vendiendo hasta ahora.

Esta información, que tal vez no parezca relevante, puede llegar a ser crítica a la hora de realizar cualquier decisión, desde tener que castellanizar términos ingleses que empleamos a diario tal como hosting, a elegir un sistema u otro en función de su nivel de Inglés.

La globalización de la tecnología ha permitido la expansión de las herramientas que hemos comentado con anterioridad. Pero las consecuencias son que aquellos que no puedan permitirse económicamente un equipo o un gestor para aquellas que requieren un nivel del idioma en el que se distribuyeron deben adaptarse al resto de tecnologías. Dentro de las tecnologías que por idioma están alcance de nuestro cliente, por el mismo motivo que el grupo anterior, lamentablemente debemos rechazar también aquellas que requieren un nivel mínimo de conocimiento para poder gestionarlas una vez instaladas. De modo que, analizando esta información y teniendo en cuenta que el administrador sería el mismo cliente, sin importar cuales sean las ventajas de cada una de las plataformas podemos ver fácilmente que todas las opciones que hemos analizado, que actualmente son las mejores o las más convenientes del mercado, son imposibles de aplicar en este proyecto.

## 2.3 Aproximación al problema

Con lo anterior descrito y con objeto de cumplir el objetivo formativo número 2 y número 3 la aproximación se realizará de forma manual casi en su totalidad.

La plataforma consistirá básicamente en una página web que sea capaz de ofrecer las características mínimas para una buena navegación en el portal y permita realizar las transacciones pertinentes entre el sistema y los usuarios. Por otro lado dispondremos de un panel de control para el administrador que será desarrollado aparte y que permitirá al cliente realizar modificaciones sobre el contenido de su tienda.

Para el usuario, la sencillez e intuición en las acciones así como los tiempos de carga, usabilidad y accesibilidad deberán ser primordiales.

Para el administrador, la gestión también deberá presentarse de forma lo más sencilla posible y los datos deberán estar muy claros para evitar que su gestión sea problemática y de lugar a errores que podrían costar caro.

Dado que la personalización por “templates” o plantillas para la estética y visualización de la página no será posible y habrá que modificar el diseño a mano mediante CSS3, se permitirá otro tipo de personalización.

Aprovechando la flexibilidad que concede el programar en su totalidad el sistema por completo podremos añadir algunas características que no todas las tiendas online contemplan. Por ejemplo, el elemento social dentro de la tienda online. Es cierto que las más grandes tiendas de páginas lo implementan desde hace tiempo, pero gran parte de las webs de tiendas más pequeñas no incluyen un mínimo de conexión entre las redes sociales y los usuarios.

# Modelo de negocio

La ventaja de montar una tienda online partiendo de un negocio ya establecido es que permite ahorrar muchos problemas y sobre todo tiempo. Encontrar proveedores, establecer presupuestos, renegociar los presupuestos y otras operaciones ya han sido realizadas y por consiguiente podemos eliminar operaciones de la lista. No obstante, sigue haciendo falta resolver otras cuestiones de igual importancia:

Rehacer el estudio de mercado: Podemos claramente suponer que no va a ser lo mismo vender una serie de productos a un público local en físico que darse abrirse paso en internet donde el público es más amplio. En el ámbito local está muy claro a qué segmento se dirigió la actividad del cliente mientras que en internet tenemos la oportunidad de llegar a un público nuevo. Esto nos lleva a la siguiente cuestión que veremos a continuación.

Captura y fidelización de clientes: En el ámbito local el buen trato y unos precios competitivos o la falta de otras entidades que ofrezcan el mismo servicio son claves para el éxito. En el ámbito perteneciente a la red se añaden nuevos factores para la captura y fidelización de clientes. Para la captura de clientes emplearemos la difusión que nos proporcionan las redes sociales, mientras que para la fidelización deberemos proporcionar una buena experiencia de usuario, una buena experiencia de usuario a la hora de navegar por el portal, una buena experiencia de usuario a la hora del pago y una buena experiencia de usuario a la hora del envío de los bienes. Estos parámetros, junto con los ya descritos anteriormente deberían asegurar que los usuarios quedan satisfechos y están dispuestos a repetir la experiencia.

Establecer el modelo de negocio: Actualmente el cliente no tiene pensado ninguna forma de sacar beneficios de la tienda online puesto que no hay una base sólida de clientes online establecida todavía. Con vistas al futuro y con el supuesto de que la plataforma funcione en su empeño de atraer y retener clientes podría empezar a establecerse un modelo de negocio. Existen varias opciones pero tras conversaciones con el cliente las que más lo han convencido han sido las opciones de monetizar mediante sistemas como GOOGLE ADSENSE, registrarse en programas de afiliados o vender espacios publicitarios.



Figura 3.1. Funcionamiento de los programas de afiliados

# Especificación de requisitos

El objetivo principal de la especificación de requisitos es informar de las funcionalidades del sistema sin mencionar la forma en la que se implementarán. Es decir, únicamente describe el sistema sin explicar cómo se realiza.

El análisis de requisitos permite definir con detalle todos los aspectos funcionales y no funcionales que dispondrá nuestra plataforma. De esta forma tanto el cliente como el desarrollador pueden tener una clara definición de cada uno de las funcionalidades que se quieren llevar a cabo para luego evitar confusiones. Es por este motivo por el que los requisitos deben estar descritos con la mayor concreción y claridad posible.

Estas definiciones son las que se emplearán como guía de desarrollo del proyecto. Es muy probable que durante el desarrollo se encuentren problemas en la especificación de requisitos. Llevado de la mano de los posibles cambios de opinión que el cliente pueda tener podrían entorpecer gravemente el desarrollo del proyecto (debemos ser muy previsores con este aspecto pues los clientes rara vez son capaces de definir lo que desean a la perfección sin cambios a la primera, y menos un cliente con el perfil que describimos en el punto [2.1](#_Perfil_del_cliente) de este documento). Por esta razón la fase de especificación de requisitos se convierte en una de las fases más importantes del desarrollo de cualquier proyecto.

A continuación y durante los siguientes sub apartados de este bloque detallaremos un diagrama general del uso de la plataforma por parte de los usuarios y definiremos los requisitos que se establecieron al inicio de las negociaciones con el cliente.

## 4.1 Diagrama de flujo y casos de uso

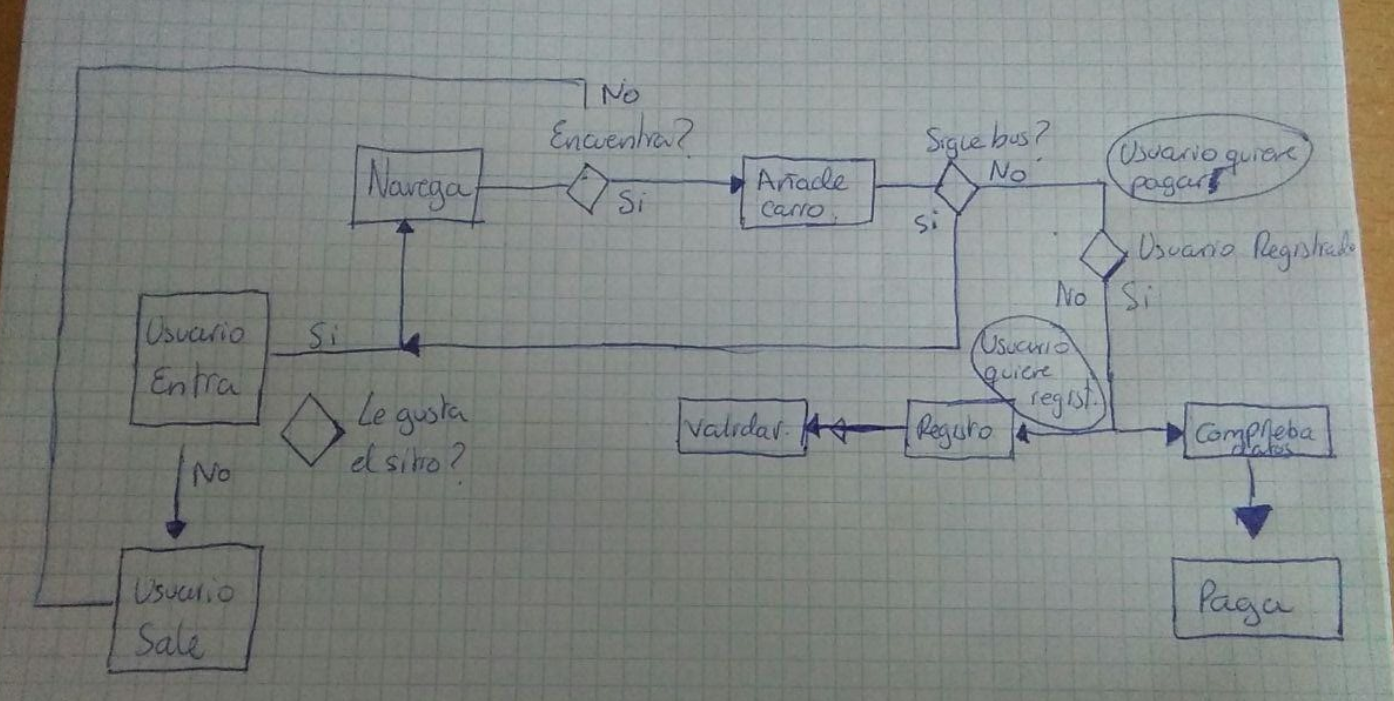
Durante las conversaciones con el cliente en las que se establecieron los casos de uso se propuso generar un pequeño diagrama de flujo que explicara a grandes rasgos el camino que seguiría el usuario en una tienda online cualquiera para poder dar así una idea al cliente de qué funcionalidades se podían encontrar en este tipo de sistemas. De esta manera el cliente pudo hacerse un claro esquema de cómo funcionaría el portal y podría modificar, añadir o eliminar los casos que él conviniese.

Figura 4.1.1 – Diagrama de flujo del recorrido básico del usuario. (A mano)

Pasado por software de generación de diagramas:

Figura 4.1.2 – Diagrama de flujo del recorrido básico del usuario (Pasado por software).

Figuras 4.1.1 Diagrama flujo básico realizado con el cliente a mano

Figuras 4.1.2 Diagrama flujo básico realizado con el cliente pasado por software

De este diagrama el cliente entendió las funcionalidades básicas que debería incluir toda tienda online, que son:

* La navegación entre categorías: De la navegación sacó en claro que debía establecer una lista de categorías y artículos que serían expuestos en la tienda online.
* La necesidad de un carro de la compra: Al igual que en la vida real se necesita un elemento en el que depositar los artículos que se van seleccionando para comprar.
* La necesidad del registro de usuarios: Para poder saber quién está comprando, qué está comprando y a dónde hay que enviar los artículos se requiere una cuenta de usuario.
* Pasarela de Pago: La parte más importante del negocio, la pasarela de pago en la que los usuarios puedan abonar sus importes y el cliente pueda recibir el dinero.

Tras esto, el cliente ya tiene una idea de lo que significan los casos de uso y procede a detallarme algunos de los que quiere que se incluyan. Del conjunto total, los que no ha incluido el cliente han sido incluidos por motivos académicos o formativos como ya comentamos al principio de este documento. Así pues a continuación detallaremos los casos de uso que se han estado incluyendo hasta ahora siguiendo una estructura en forma de tabla donde se mostrarán:

* El nombre de la funcionalidad (Nombre)
* La acción desencadenante (Desencadenante)
* Los procesos realizados (Procesos)
* Las acciones desencadenadas (Consecuencias)

### 4.1.1 Casos de Uso Tienda Online

|  |  |
| --- | --- |
| CU001 | Carga inicial del documento |
| Precondición | El usuario ha introducido la URL de nuestro portal o bien ha accedido por resultados de un buscador. |
| Descripción | Se muestra el documento HTML al usuario. |
| Secuencia Normal | 1. El sistema carga el documento HTML por defecto (índice). |
| Post-Condición | El usuario puede ver el documento y está listo para navegar por él. |
| Excepciones | * Errores en el servidor. * Errores en los documentos. * El usuario no introduce bien nuestra dirección URL. |
| Comentarios |  |

Figura 4.1.1.1 – Caso de uso 001

|  |  |
| --- | --- |
| CU002 | Navegación por el portal |
| Precondición | El usuario ha accedido al portal y los documentos cargan correctamente. |
| Descripción | El usuario puede explorar las diferentes categorías y documentos que componen la tienda online. |
| Secuencia Normal | 1. El usuario selecciona la categoría o el elemento que quiere explorar. 2. El sistema carga el documento HTML correspondiente. |
| Post-Condición | El usuario puede visualizar el documento deseado. |
| Excepciones | * Errores en los documentos. * Error en el servidor. |
| Comentarios |  |

Figura 4.1.1.2 – Caso de uso 002

|  |  |
| --- | --- |
| CU003 | Registro |
| Precondición | El usuario accede correctamente al servicio de registro de nuestro sistema mediante los enlaces proporcionados en los documentos HTML. |
| Descripción | Permite al usuario crear una cuenta en nuestro sistema. |
| Secuencia Normal | 1. El usuario accede al sistema de registro. 2. El sistema le proporciona un formulario de registro. 3. El usuario rellena el formulario de registro correctamente. 4. El sistema da de alta al usuario en temporalidad. 5. El sistema envía un email de validación al usuario. |
| Post-Condición | El usuario queda registrado pero en el grado de temporalmente y debe validar su cuenta mediante el email enviado |
| Excepciones | * Errores en el servidor. * Errores en los documentos. * Error al rellenar el documento. * Error en el servidor de email. |
| Comentarios | El usuario quedará registrado en una tabla temporal de la base de datos. Si en un periodo de tiempo (todavía por establecer) no ha activado su cuenta los datos del usuario serán eliminados automáticamente por los crons que se definirán más adelante en las funcionalidades. |

Figura 4.1.1.3 – Caso de uso 003

|  |  |
| --- | --- |
| CU004 | Validación |
| Precondición | * El usuario ha rellenado el formulario de registro correctamente. * El usuario ha recibido el email de validación enviado por el sistema. |
| Descripción | El usuario quiere validar su cuenta en la plataforma. El sistema deberá comprobar la validación e informar al usuario del resultado del proceso. |
| Secuencia Normal | 1. El usuario accede a su cuenta de email. 2. El usuario pincha en el enlace proporcionado en el email. 3. El sistema cargará el documento correspondiente. 4. El sistema procesará la información 5. El sistema actualizará el estado del usuario. 6. El sistema mostrará un mensaje informativo al usuario. |
| Post-Condición | El usuario es validado correctamente y puede hacer uso de las funcionalidades propias de los usuarios registrados en nuestro sistema. |
| Excepciones | * Errores en el servidor. * Errores en los documentos. * El código de validación del email está caducado. * El usuario no ha recibido el email. |
| Comentarios |  |

Figura 4.1.1.4 – Caso de uso 004

|  |  |
| --- | --- |
| CU005 | Acceso a cuenta de usuario |
| Precondición | El usuario ha sido registrado y validado satisfactoriamente. |
| Descripción | El usuario accede a su cuenta de usuario para poder revisar la información personal o los datos de sus pedidos, etc. |
| Secuencia Normal | 1. El usuario accede al menú de acceso. 2. El sistema le muestra el formulario correspondiente. 3. El usuario rellena el formulario. 4. El sistema comprueba los datos. 5. El sistema permite el acceso al usuario a su cuenta. |
| Post-Condición | El sistema crea una sesión para el usuario y le permite acceder a sus datos mientras permanezca en el sitio. |
| Excepciones | * Errores en el servidor. * Errores en los documentos. * El usuario no introduce los datos correctos. * El usuario intenta acceder sin cuenta o sin haber validado su cuenta. |
| Comentarios |  |

Figura 4.1.1.5 – Caso de uso 005

|  |  |
| --- | --- |
| CU006 | Modificación datos personales |
| Precondición | * El usuario está logueado. * El usuario accede a su perfil y a sus datos personales. |
| Descripción | El usuario puede alterar (o no) sus datos personales y consultar otros datos de relevancia de su perfil. |
| Secuencia Normal | 1. El usuario accede a su perfil. 2. El sistema carga sus datos personales y los muestra. 3. El usuario modifica los datos personales. 4. El sistema registra los cambios y actualiza. |
| Post-Condición | Los datos que el usuario ha modificado quedan registrados y actualizados.  El sistema refresca la sección mostrando los cambios realizados al usuario. |
| Excepciones | * Errores en el servidor. * Errores en los documentos. * El usuario no introduce correctamente los nuevos datos. * El usuario no realiza ningún cambio en los datos. * Error en el sistema de actualización |
| Comentarios | Si el usuario no introduce la contraseña actual a la hora de realizar los cambios en los datos, éstos no se guardarán. |

Figura 4.1.1.6 – Caso de uso 006

|  |  |
| --- | --- |
| CU006 | Comparar artículos |
| Precondición |  |
| Descripción | El usuario quiere comparar artículos y pincha en el botón correspondiente. El sistema guardará el artículo en comparación para luego mostrarlos. |
| Secuencia Normal | 1. El usuario selecciona los artículos a comparar. 2. El sistema guarda la información del / de los artículos. 3. El usuario manda al sistema compararlos. 4. El sistema manda los artículos a un nuevo documento. |
| Post-Condición | Los artículos quedan exhibidos en el nuevo documento de forma que el usuario puede ver todas las características de ellos a la vez. |
| Excepciones | * Errores en el servidor. * Errores en los documentos. |
| Comentarios | El máximo de artículos a comparar en el sistema serán 3. Los elementos ocuparán un espacio proporcional al total disponible del documento. Es decir: 100(%)/x donde x es el número de elementos. Por el momento el cliente ha querido que los artículos se vean de esta manera. Es una manera muy simple de comparación. Es probable que en el futuro se exija añadir parámetros de comparación o incluso resaltar las características destacadas de cada producto. |

Figura 4.1.1.6 – Caso de uso 006

|  |  |
| --- | --- |
| CU007 | Valorar y comentar artículos |
| Precondición | * El usuario está registrado y logueado. * El usuario accede a la vista de un artículo. |
| Descripción | El usuario registrado tiene la capacidad de valorar y comentar artículos. |
| Secuencia Normal | 1. El usuario abre el artículo. 2. El sistema muestra la información del artículo. Además muestra la posibilidad de comentar pulsando un botón. 3. El usuario introduce su valoración y/o comentario. 4. El sistema registra la información. |
| Post-Condición | El sistema refresca la sección mostrando la información introducida. |
| Excepciones | * Errores en el servidor. * Errores en los documentos. * Error en la actualización de datos |
| Comentarios | El máximo número de comentarios y valoraciones que puede realizar el usuario es de 1 por artículo. |

Figura 4.1.1.7 – Caso de uso 007

|  |  |
| --- | --- |
| CU008 | Añadir al carrito |
| Precondición | * El usuario está registrado y logueado. |
| Descripción | El usuario puede añadir artículos al carrito de la compra para después realizar el pago. |
| Secuencia Normal | 1. El usuario selecciona un artículo. 2. El sistema añade al carrito dicho artículo. 3. El sistema actualiza los parámetros relacionados. |
| Post-Condición | El usuario ya está listo para realizar el pago. |
| Excepciones | * Errores en el servidor. * Errores en los documentos. * No hay artículos en stock. |
| Comentarios | No hay máximo número de artículos en el carrito. El máximo de cada artículo está delimitado por el máximo de unidades disponibles en stock en ese momento. |

Figura 4.1.1.8 – Caso de uso 008

|  |  |
| --- | --- |
| CU009 | Editar carrito |
| Precondición | * El usuario está registrado y logueado. * Mínimo de 1 objeto en carrito. |
| Descripción | El usuario puede eliminar artículos del carrito o incrementar el número de unidades de un artículo. |
| Secuencia Normal | 1. El usuario realiza el cambio deseado. 2. El sistema actualiza el carrito. |
| Post-Condición | El sistema muestra los cambios en el carrito. |
| Excepciones | * Errores en el servidor. * Errores en los documentos. * El carrito está vacío. |
| Comentarios | Los cambios se someten a la misma norma que al añadir al carrito. |

Figura 4.1.1.9 – Caso de uso 009

|  |  |
| --- | --- |
| CU010 | Realizar compra |
| Precondición | * El usuario está registrado y logueado. * Mínimo de 1 objeto en carrito. * Mínimo de 1 dirección de envío en el perfil del usuario * Mínimo de 1 forma de pago. |
| Descripción | El usuario ha terminado de seleccionar los artículos que desea comprar y se dispone a realizar el pago. |
| Secuencia Normal | 1. El usuario comprueba los datos del carrito. 2. El usuario comprueba los datos de dirección de envío. 3. El usuario comprueba los datos de método de envío. 4. El usuario selecciona PAGAR. 5. El sistema abre la pasarela de pago. 6. El usuario introduce sus datos. 7. El usuario confirma el pago. |
| Post-Condición | * El sistema registra el pedido. * El sistema registra el pago. * El usuario es reconducido con un mensaje informativo. |
| Excepciones | * Errores en el servidor. * Errores en los documentos. * El carrito está vacío. No hay direcciones de envío o método de envío. * Error en la pasarela de pago. * Error en el pago. * El usuario cancela el pago. |
| Comentarios |  |

Figura 4.1.1.9 – Caso de uso 009

### 4.1.2 Casos de uso Panel de Control

A partir de este punto y hasta el final del bloque de “Casos de uso Panel de Control” tomaremos por usuario al administrador del sitio.

|  |  |
| --- | --- |
| CU001CP | Acceso |
| Precondición | El usuario ha introducido la URL del panel de control. |
| Descripción | Se intenta acceder al panel de configuración de la tienda online. |
| Secuencia Normal | 1. El sistema carga un formulario de acceso. 2. El usuario rellena el formulario. 3. El sistema comprobará los datos del usuario. 4. El sistema permitirá el acceso. 5. El sistema cargará el documento HTML. |
| Post-Condición | El usuario es permitido al panel de control y quedará registrada su sesión. |
| Excepciones | * Errores en el servidor. * Errores en los documentos. * El usuario no introduce bien la dirección URL. * El usuario no introduce los datos correctos. |
| Comentarios |  |

Figura 4.1.2.1 – Caso de uso 001 CP

Durante la gestión de operaciones en el panel de control, la mayoría de las funcionalidades van a ser muy similares para cada uno de los menús. Las más comunes serán mostrar, cargar, añadir, eliminar o editar (éstas 3 últimas podremos agruparlas simplemente en “gestión”). Por tanto podremos resumir los casos de uso en una sola tabla que agrupa las posibles opciones para cada menú.

|  |  |
| --- | --- |
| CU002CP | Gestión de menús |
| Precondición | El usuario ha accedido al panel de control |
| Descripción | En conjunto hace referencia a las posibles operaciones que se pueden realizar en cada uno de los menús. Estas operaciones son comunes para cada uno de los menús. |
| Secuencia Normal | 1. El usuario selecciona la operación que desea realizar. 2. El sistema carga los elementos y parámetros necesarios para realizar la operación. 3. \*El usuario rellena los datos necesarios. 4. El sistema realiza los cambios. 5. El sistema refresca la sección mostrando los cambios. |
| Post-Condición | Las operaciones quedan realizadas. |
| Excepciones | * Errores en el servidor. * Errores en los documentos. * Error en los parámetros. * Error en los datos introducidos. * Error en la conexión a la base de datos. * Error en la actualización. * El usuario no introduce los datos correctos. |
| Comentarios | (\*)El punto 3 de la secuencia normal hace referencia a la posible demanda de parámetros o información por parte del sistema hacia el usuario en determinadas operaciones. Por lo general las operaciones de adición o de edición de los elementos base que conforman el conjunto de la temática del menú específico en el que nos encontremos. |

Figura 4.1.2.2 – Caso de uso 002 CP

## 4.2 Requisitos funcionales

Los requisitos funcionales describen lo que el sistema debe hacer y cómo debe hacerlo. El comportamiento del sistema lo hemos definido mediante los casos de uso, así que los requisitos funcionales describirán cómo se desarrollarán cada uno de los casos de uso del sistema. Por tanto definiremos los requisitos funcionales de la siguiente manera:

* ID: Identificador del requisito funcional.
* CU: Caso de uso de referencia.
* Secuencia: Lista de procesos que conformará el comportamiento del requisito. Encontramos aquí tanto la entrada de datos como la salida de datos.

### 4.2.1 Requisitos funcionales Tienda Online

|  |  |
| --- | --- |
| ID | RF001 |
| CU | CU001 |
| Secuencia | * JS - Comprobación de las variables de sesión. * PHP - Comprobación de las variables de error. * JS - Generación html/php. * Salida: documento HTML. |

Figura 4.2.1.1 – Requisito Funcional 001;

|  |  |
| --- | --- |
| ID | RF002 |
| CU | CU002 |
| Secuencia | * Entrada: evento - elemento de navegación. * JS - Obtención de datos. * JS - Carga de datos. * JS - Generación código. * Salida: documento HTML |

Figura 4.2.1.2 – Requisito Funcional 002;

|  |  |
| --- | --- |
| ID | RF003 |
| CU | CU003 |
| Secuencia | * Entrada: evento - acceso a menú de registro. * Salida: documento HTML. * \*Espera entrada\* * Entrada: datos introducidos por el usuario. * JS - Obtención de datos. * JS - Procesado de datos. * PHP - Actualización de base de datos. * Salida: email de validación, nuevo documento HTML informativo. |

Figura 4.2.1.3 – Requisito Funcional 003;

|  |  |
| --- | --- |
| ID | RF004 |
| CU | CU004 |
| Secuencia | * Entrada: código de validación obtenido del enlace enviado por mail al usuario tras el registro correcto. * JS - Procesado de datos. * PHP - Actualización de base de datos. * Salida: documento HTML informativo. |

Figura 4.2.1.4 – Requisito Funcional 004;

|  |  |
| --- | --- |
| ID | RF005 |
| CU | CU005 |
| Secuencia | * Entrada: evento - menú de acceso a cuenta de usuario. * Salida: documento HTML (form). * \*Espera entrada\* * Entrada: datos introducidos por el usuario. * JS - Procesado de datos * PHP - Consulta a base de datos * Salida: documento HTML informativo. * Salida: Acceso a menú de acceso a cuenta de usuario. |

Figura 4.2.1.5 – Requisito Funcional 005;

|  |  |
| --- | --- |
| ID | RF006 |
| CU | CU006 |
| Secuencia | * Entrada: evento - menú cuenta personal. * Salida: documento HTML (form). * \*Espera entrada\* * Entrada: datos introducidos por el usuario. * JS - Procesado de datos. * PHP – Actualización datos en base de datos * Salida: documento HTML informativo (refresco). |

Figura 4.2.1.6 – Requisito Funcional 006;

|  |  |
| --- | --- |
| ID | RF007 |
| CU | CU007 |
| Secuencia | * Entrada: evento – selección de un artículo. * JS – Inclusión de artículo en <contenedor>. * Entrada: evento – comparación artículos. * JS – Generación de código HTML. * Salida: documento HTML. |

Figura 4.2.1.7 – Requisito Funcional 007;

|  |  |
| --- | --- |
| ID | RF008 |
| CU | CU008 |
| Secuencia | * Entrada: evento – selección de un artículo. * \*Espera entrada\* * Entrada: Datos introducidos por el usuario. * JS – Procesa los datos. * PHP – datos introducidos en base de datos. * Salida: documento HTML (refresco). |

Figura 4.2.1.8 – Requisito Funcional 008;

|  |  |
| --- | --- |
| ID | RF009 |
| CU | CU009 |
| Secuencia | * Entrada: evento – selección de un artículo. * JS – Añade el artículo. * JS – Actualiza datos locales. * Salida: ninguna. |

Figura 4.2.1.9 – Requisito Funcional 009;

|  |  |
| --- | --- |
| ID | RF010 |
| CU | CU010 |
| Secuencia | * Entrada: evento – alteración carrito. * JS – actualización de los datos locales. * JS – alteración objetivo del carrito. * Salida: documento HTML (refresco). |

Figura 4.2.1.10 – Requisito Funcional 010;

|  |  |
| --- | --- |
| ID | RF011 |
| CU | CU011 |
| Secuencia | * Entrada: evento – acción de continuar hacia compra. * JS – comprobación datos previos. * \*Espera entrada\* * JS – Recogida de datos del usuario (confirmación datos). * JS – Generación de datos. * JS – Carga de datos en HTML. * PHP – Generación datos en base de datos. * PHP – Envío de datos a la pasarela de pago. * PHP – apertura pasarela de pago. * \*Espera entrada (pago)\* * PHP – Actualiza datos base de datos. |

Figura 4.2.1.11 – Requisito Funcional 011;

### 4.2.2 Requisitos funcionales panel de control

|  |  |
| --- | --- |
| ID | RF001CP |
| CU | CU010CP |
| Secuencia | (Apertura de menús)   * Entrada: evento – selección de menús. * PHP – obtención de datos de la base de datos. * JS – cargar de datos y generación de HTML. * Salida: documento HTML. |

Figura 4.2.2.1 – Requisito Funcional 01;

|  |  |
| --- | --- |
| ID | RF002CP |
| CU | CU010CP |
| Secuencia | (Adición, Edición, Eliminación de elementos de los menús).   * Entrada: evento – selección de operación. * \*Espera entrada\* (opcional según la operación) * ^JS – Obtención de datos introducidos. (paso dependiente del tipo de operación). * PHP – Actualización base de datos * Salida: documento HTML (refresco). |

Figura 4.2.2.2 – Requisito Funcional 02;

## Requisitos no funcionales

Los requisitos no funcionales, como su nombre indica son aquellos que no van dirigido a las funcionalidades si no a otros aspectos o propiedades que debe cumplir la plataforma. Los podemos exhibir mediante tablas (al igual que los casos de uso y los requisitos funcionales) con los siguientes parámetros:

* ID: Identificador
* Nombre del requisito no funcional.
* Descripción breve del requisito.

En nuestro proyecto en un principio se han determinado que los requisitos no funcionales que se deben cumplir son:

|  |  |
| --- | --- |
| ID | RNF001 |
| Nombre | Costes Temporales |
| Descripción | Para proporcionar una buena experiencia de usuario es imprescindible mantener los costes temporales de los procesos al nivel más bajo posible. |

Figura 4.3.1 – Requisito No Funcional 01;

|  |  |
| --- | --- |
| ID | RNF002 |
| Nombre | Seguridad |
| Descripción | Los datos que se almacenan en la base de datos son datos sensibles y deben mantener un nivel de seguridad alto. Además debe informarse a los usuarios que sus datos están siendo guardados para poder cumplir la ley orgánica de protección de datos. |

Figura 4.3.2 – Requisito No Funcional 02;

|  |  |
| --- | --- |
| ID | RNF003 |
| Nombre | Sostenibilidad |
| Descripción | El objetivo principal del portal web es aumentar el número de ventas del cliente. Por tanto, es imperativo que los componentes de hosting, dominio así como el mantenimiento del sitio web sean de calidad. |

Figura 4.3.3 – Requisito No Funcional 03;

|  |  |
| --- | --- |
| ID | RNF004 |
| Nombre | Servicios Externos |
| Descripción | Para algunas funciones dependeremos de servicios externos como podrían ser, por ejemplo, el servicio de mensajería (email) |

Figura 4.3.4– Requisito No Funcional 04;

|  |  |
| --- | --- |
| ID | RNF005 |
| Nombre | Capacidad |
| Descripción | Aunque no es imperativo ni mucho menos pues el cliente no espera llegar a niveles de difusión tan altos, es recomendable pensar en definir un sistema desde el principio que pueda soportar un número suficiente de clientes, artículos, etc. en base de datos. |

Figura 4.3.5– Requisito No Funcional 05;

|  |  |
| --- | --- |
| ID | RNF005 |
| Nombre | Mantenimiento |
| Descripción | El cliente desea que el poco mantenimiento que se desee realizar sobre su tienda online (gestión de recursos principalmente) sea lo suficientemente fácil e intuitivo como para que una persona de con su perfil sea capaz de gestionarlo. |

Figura 4.3.6– Requisito No Funcional 06;

## Requisitos de diseño

Los requisitos de diseño hacen referencia a todo lo relacionado con el aspecto visual y de interfaz del sistema. Definimos a continuación los requisitos de diseño siguiendo el mismo esquema que los requisitos no funcionales (identificador, nombre, descripción).

|  |  |
| --- | --- |
| ID | RD001 |
| Nombre | Adaptable |
| Descripción | Hoy en día los dispositivos Smartphone son una herramienta muy usada para navegar por internet. Esto genera gran énfasis en que el diseño de interfaz de nuestro sistema sea fluido y adaptable a todos los dispositivos. |

Figura 4.4.6– Requisito de diseño 01;

|  |  |
| --- | --- |
| ID | RD002 |
| Nombre | Usable |
| Descripción | Una clave para el éxito de un portal web es la usabilidad de su diseño, por tanto la usabilidad de la interfaz del sistema es muy importante. |

Figura 4.4.6– Requisito de diseño 02;

|  |  |
| --- | --- |
| ID | RD003 |
| Nombre | Visualmente estético |
| Descripción | El diseño debe además de ser usable y adaptable, ser visualmente atractivo. O al menos lo suficientemente abstracto como para no generar una mala experiencia a los usuarios. |

Figura 4.4.6– Requisito de diseño 03;

# Estudio de viabilidad

En el estudio de viabilidad se analiza si el proyecto podrá desarrollarse correctamente o no en base a unos parámetros y datos concretos ya sean de naturaleza empírica u objetiva. Actualmente podemos asegurar que dados los problemas existentes en la realización del proyecto y primeramente en la definición del mismo, a la hora de entregar este documento el proyecto no estará completo. Pero por ahora vamos a poner en exposición la planificación actual.

## Planificación temporal

### 5.1.1 Hitos

El desarrollo de la plataforma se ha dividido en hitos en los que se busca un balance entre la carga de tareas y el tiempo de cada hito.

Además la planificación de cada hito y la asignación de prioridades irá sujeta a dos factores concretos: el ámbito académico y el ámbito profesional.

En el ámbito profesional podemos encontrar que la prioridad de las tareas vendrá organizada por las demandas del cliente.

En el ámbito académico la prioridad en las tareas vendrá dictada por la asignación de eventos relativos al trabajo de final de grado y su evaluación.

Para poder encontrar el balance entre los dos factores que dictan el ritmo de desarrollo del proyecto la prioridad de las tareas oscilará a medida que se avance en el desarrollo. De modo que probablemente los datos recogidos en este documento puedan no coincidir con lo llevado a cabo.

* Hito 0: Definición del proyecto. Hito de carácter académico en el que se necesitaba definir un proyecto concreto para la realización del trabajo de final de carrera.
* Hito 1: Comienzo del desarrollo. Hito de carácter profesional en el que se desarrollan las funcionalidades básicas y principales de la plataforma web por parte del cliente. Es decir, se da forma y base a la tienda online como tal.
* Hito 2: Presentación de Septiembre. Hito de carácter tanto académico como profesional. En este hito se desarrollarán las funcionalidades básicas y principales del sistema de gestión, es decir del panel de control del administrador. Al finalizar este hito se presentará el desarrollo del proyecto ante un tribunal para su evaluación. Pero el trabajo continuará.
* Hito 3: Primera corrección. Hito de carácter profesional. Tras el resultado de la evaluación ante el tribunal en la universidad el trabajo continuará para dar lugar a las primeras correcciones. En este hito se desarrollarán las funcionalidades que falten y comenzará la búsqueda de bugs y fallos del sistema.
* Hito 4: Hito Final. Hito de carácter profesional. Una vez concluido el hito 3, concluidas todas las funcionalidades y superadas las primeras pruebas, se realizará una segunda búsqueda y corrección de errores. Se pulirán los detalles y el diseño.

### 5.1.2 Estimación de costes

Como el nombre indica, la estimación de costes es una estimación aproximada del esfuerzo y de los recursos empleados en el desarrollo del proyecto. Tomando como esfuerzo el número de horas trabajadas por el desarrollador y los recursos como los elementos externos necesarios para el desarrollo, definiremos a continuación figuras representativas de los costes de cada uno. Dado que el resumen de las horas totales invertidas muy probablemente variará con respecto a las horas planeadas dedicaremos un apartado en la sección 11 a ese tema.

El coste temporal será especificado en días en el diagrama de Gantt realizado en Microsoft Project 2013. No obstante la medida básica serán horas. 1 día ~ 6h de trabajo.

A continuación desglosaremos en más detalles representados visualmente con gráficas para cada uno de los hitos los siguientes apartados:

* Diagrama de Gantt.
* Costes temporales diseño.
* Costes temporales funcionalidades.
* Distribuciones de costes en días.
* Distribuciones de costes en horas.

### Hito 0 e Hito 1

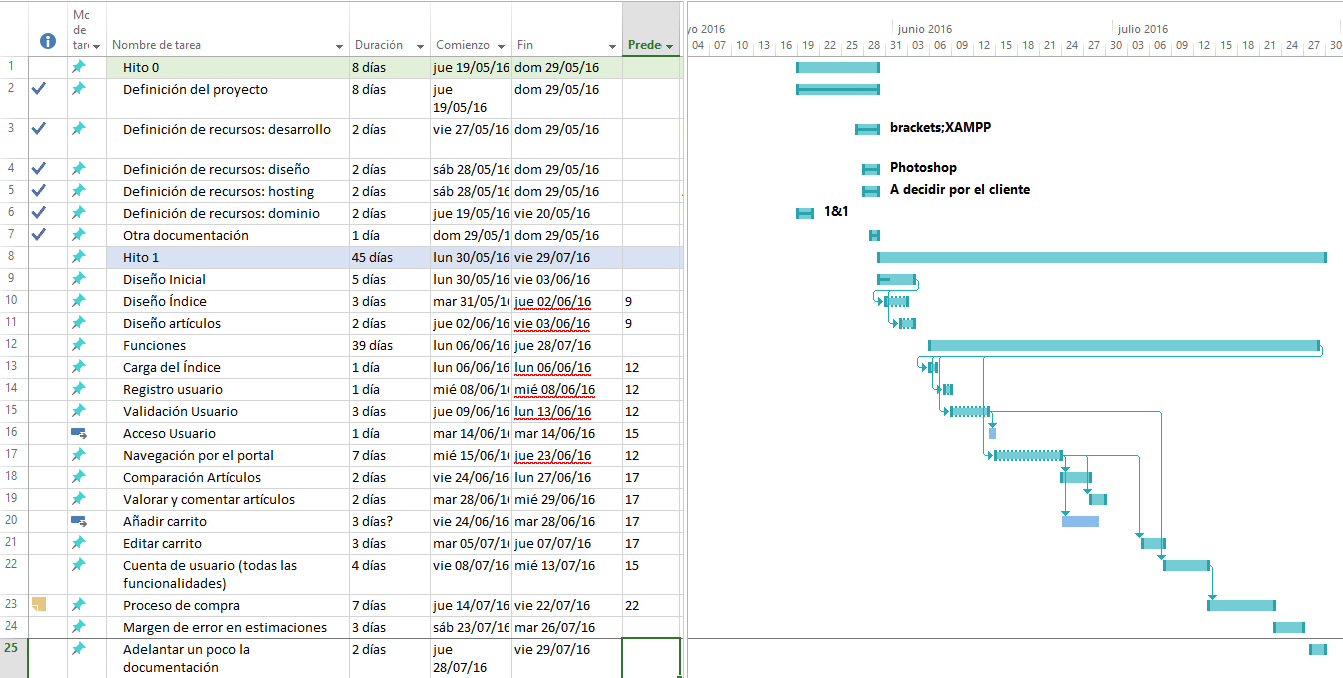


Figura 5.1.2.1 Diagrama de Gantt hito 0 e hito 1.

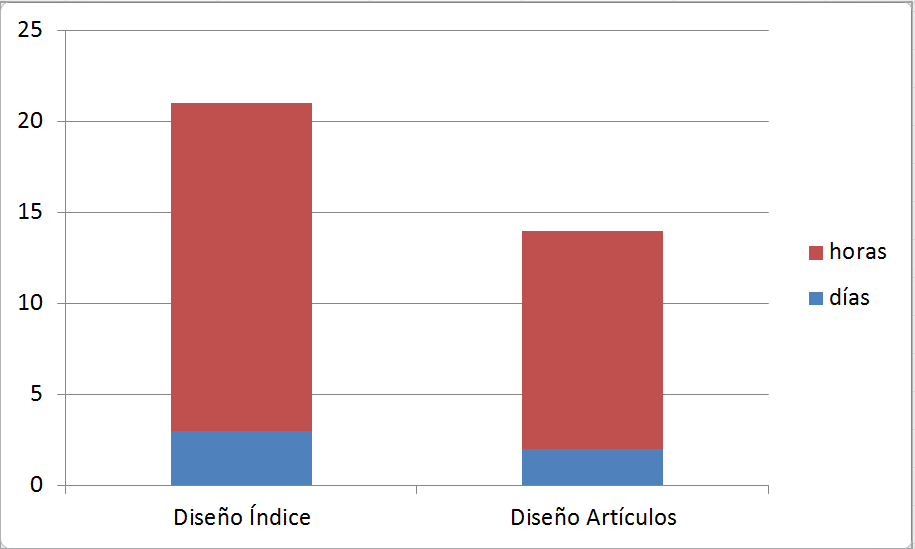


Figura 5.1.2.2 – Costes temporales diseño hito 0 y Figura 5.1.2.3 – Costes temporales diseño hito 0.

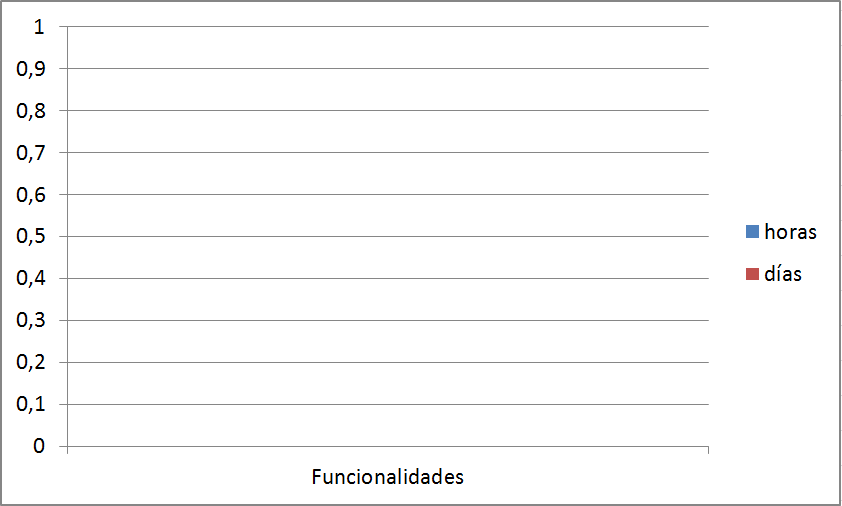


Figura 5.1.2.4 – Costes temporales funcionalidades hito 0

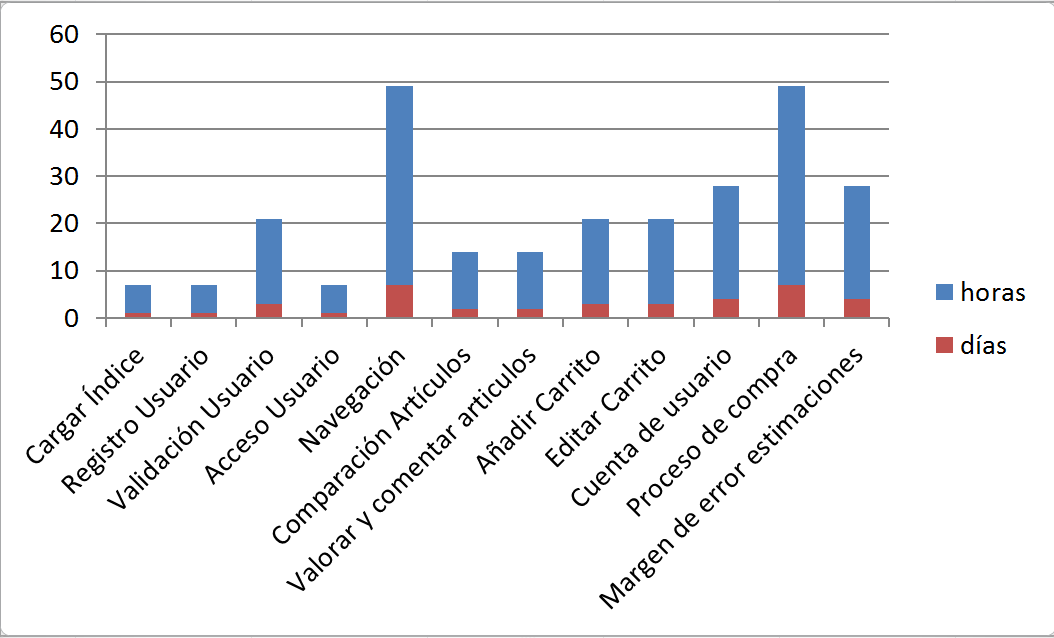


Figura 5.1.2.5 – Costes temporales funcionalidades hito 1

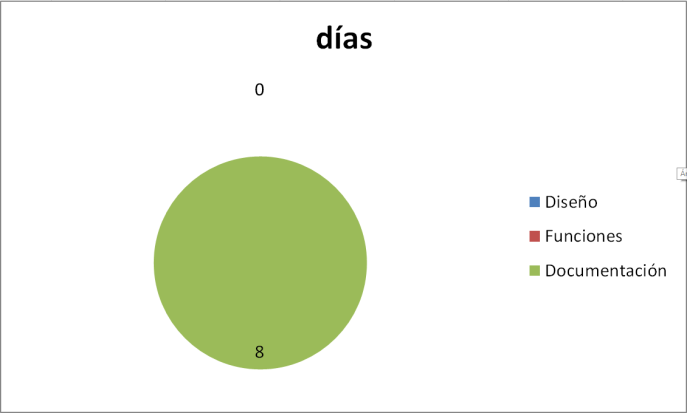


Figura 5.1.2.6– División de costes temporales de las tareas en horas hito 0



Figura 5.1.2.7 – División de costes temporales de las tareas en días hito 0

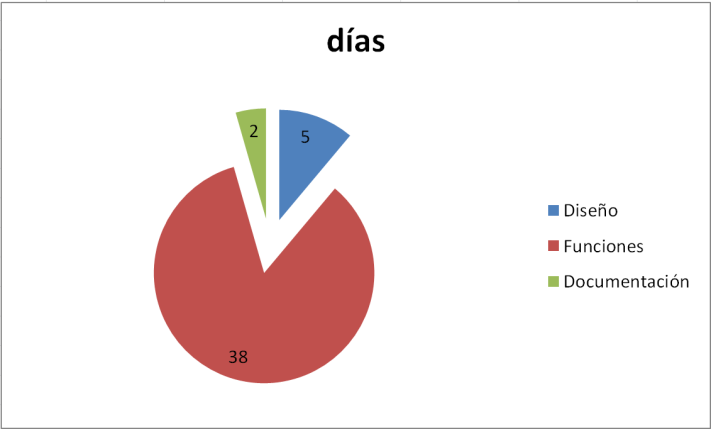


Figura 5.1.2.8 – División de costes temporales de las tareas en días hito 1

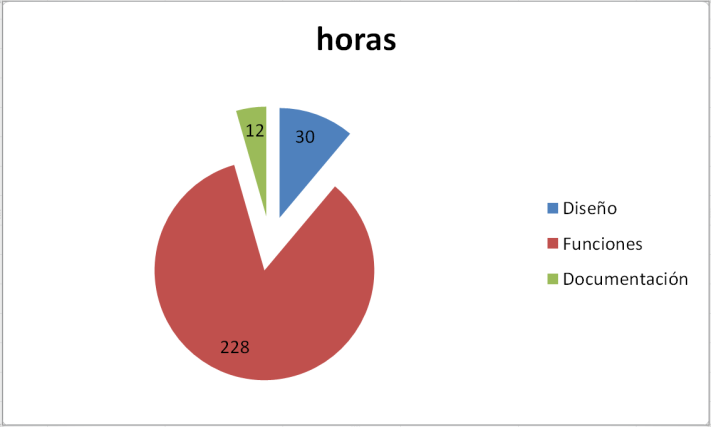


Figura 5.1.2.9 – División de costes temporales de las tareas en días hito 1

### Hito 2

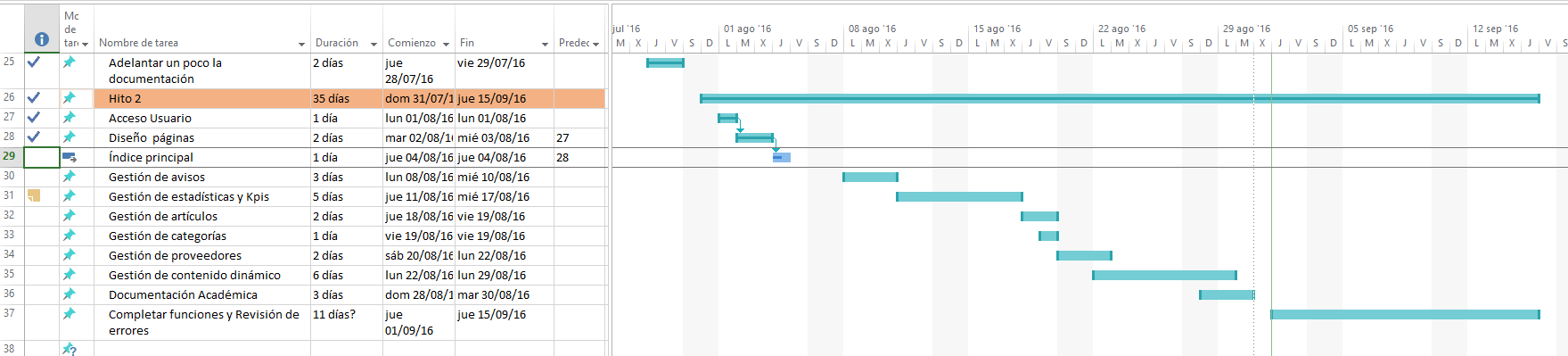


Figura 5.1.2.10 – Diagrama Gantt hito 2

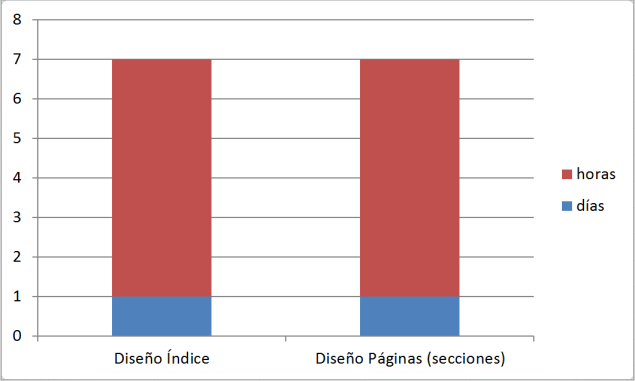
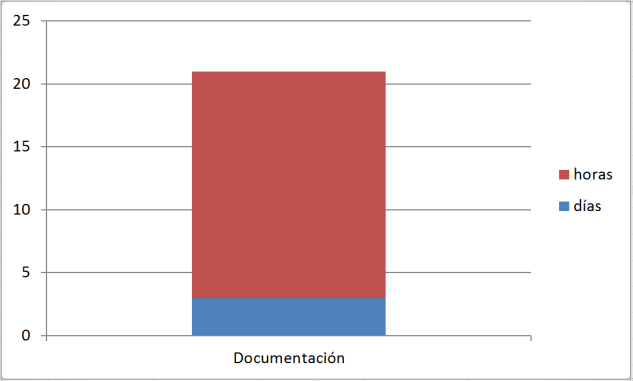


Figura 5.1.2.11 – Costes temporales documentación hito 2

Figura 5.1.2.12 – Costes temporales diseño hito 2

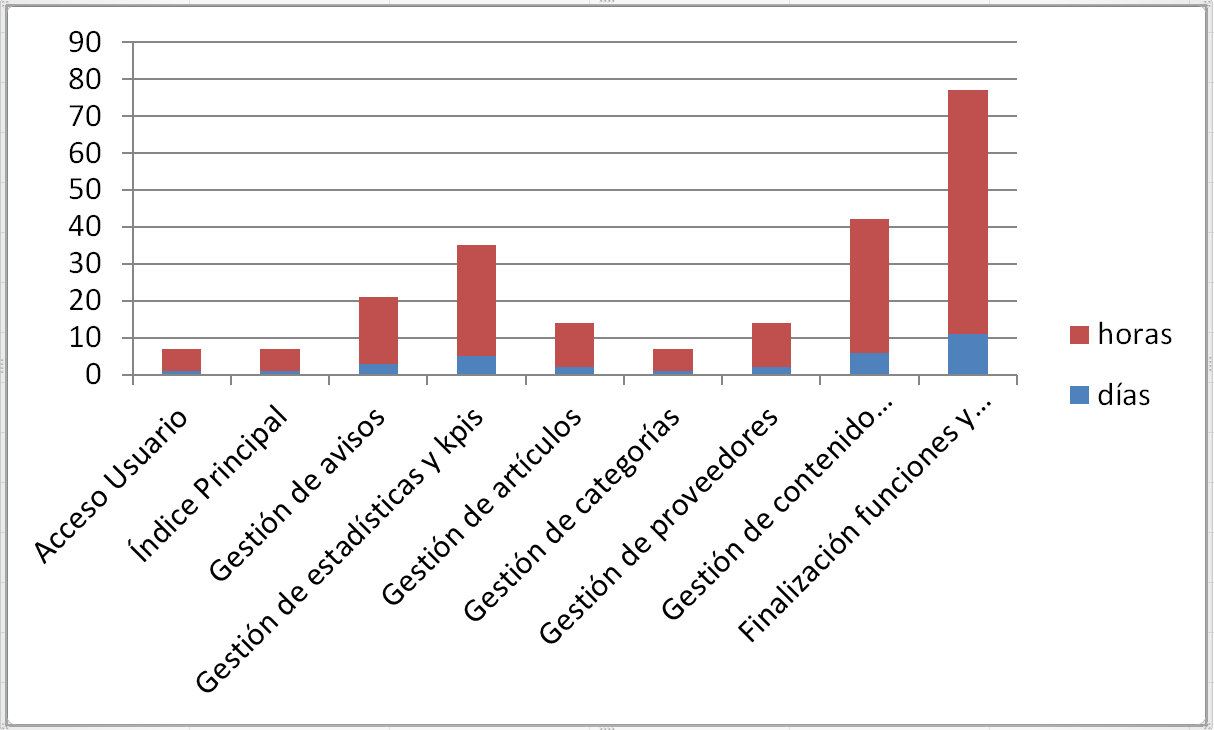


Figura 5.1.2.13 – Costes temporales funcionalidades hito 2

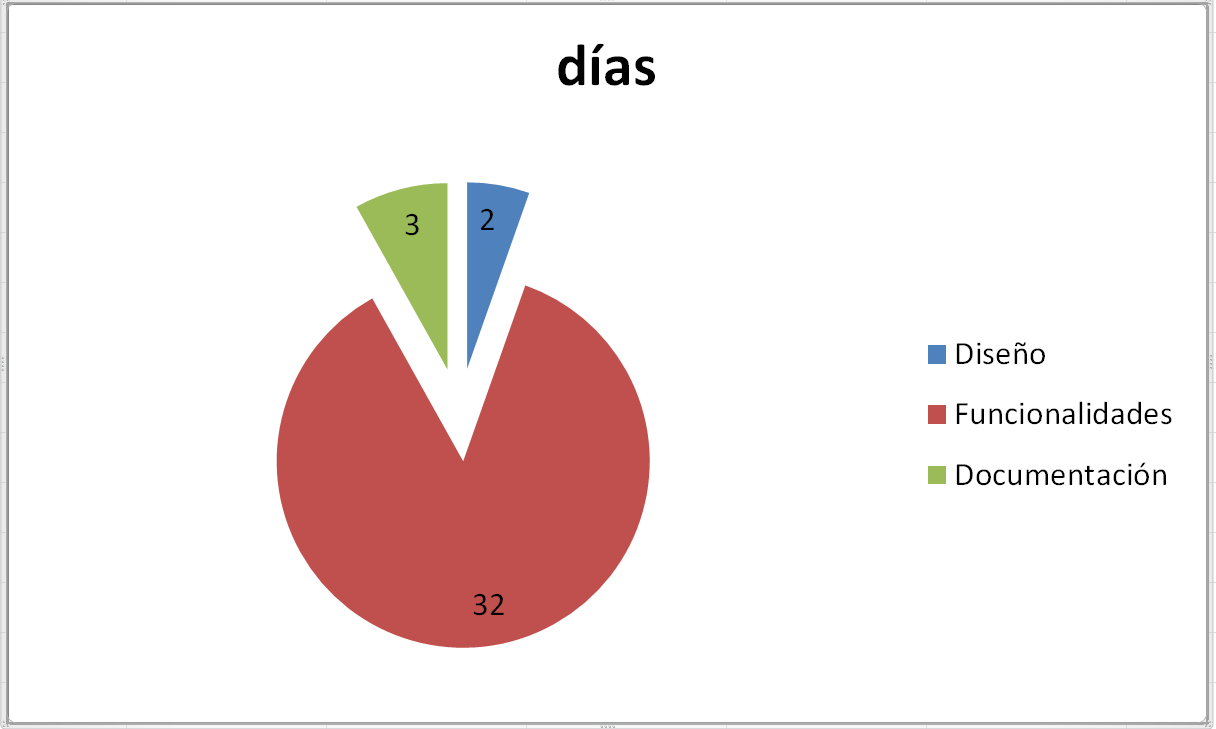


Figura 5.1.2.14 – Reparto de costes temporales de las tareas en días

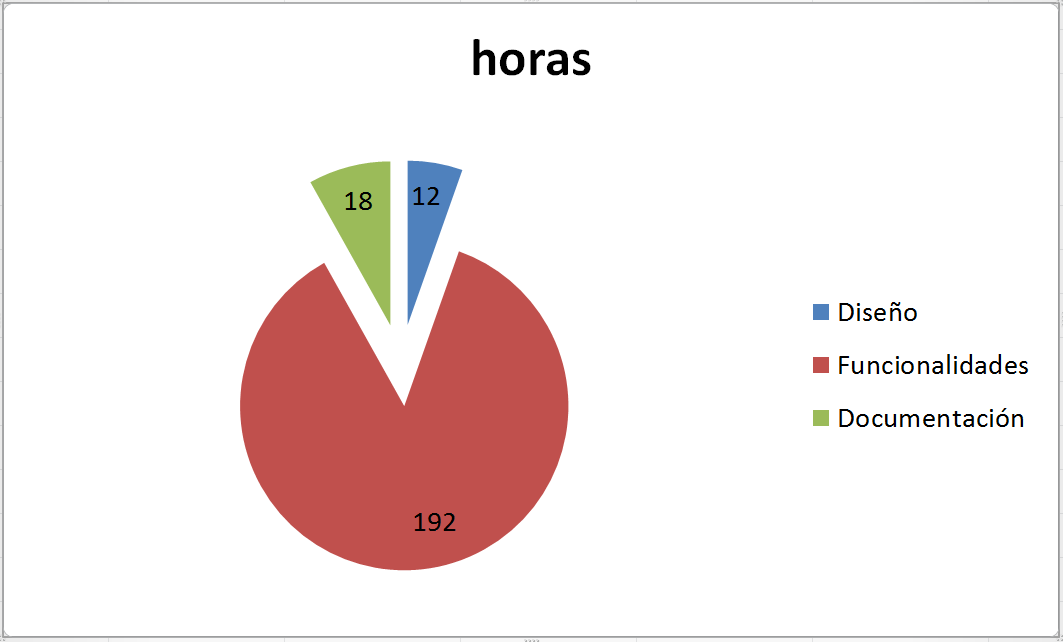


Figura 5.1.2.15 – Reparto de costes temporales de las tareas en horas

### Hitos 3 y 4 (extra académicos)

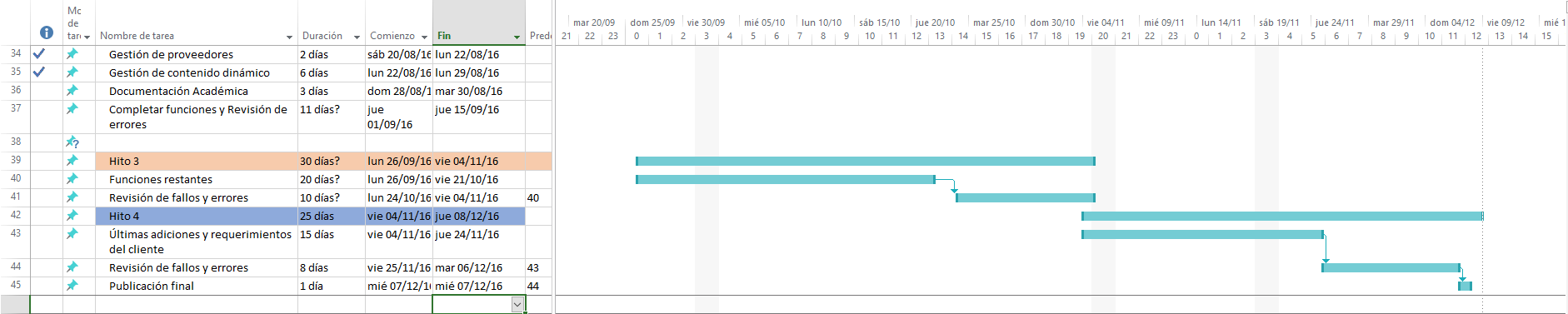


Figura 5.1.2.16 – Diagrama de Gantt de los hitos 3 y 4

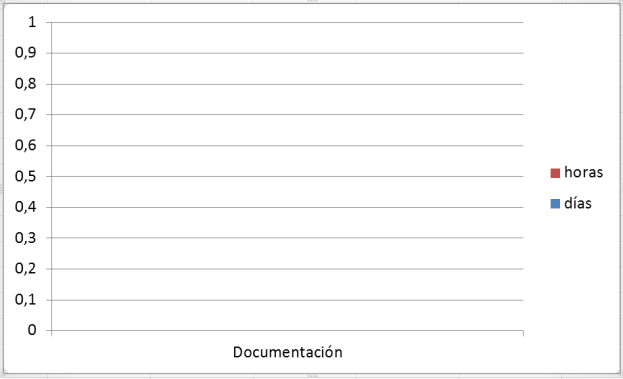
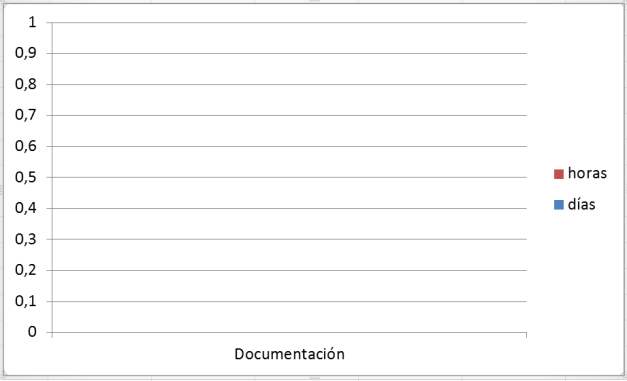


Figura 5.1.2.17 – Coste temporal de documentación del hito 3

Figura 5.1.2.18 – Coste temporal de documentación del hito 4



Figura 5.1.2.19 – Coste temporal de diseño del hito 3

Figura 5.1.2.20 – Coste temporal de diseño del hito 4

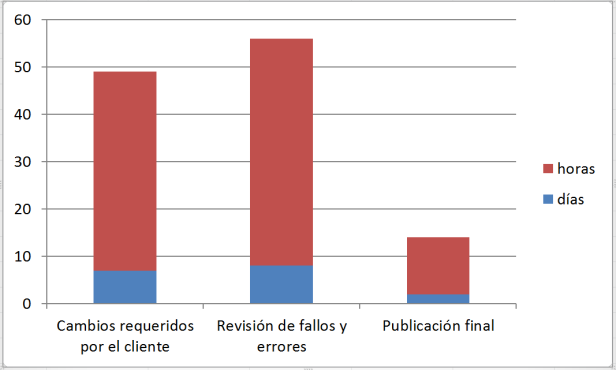
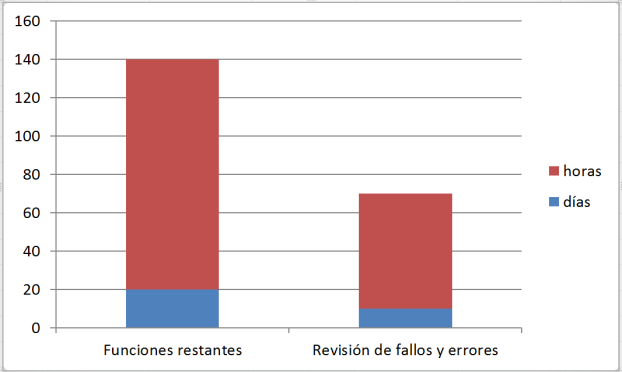


Figura 5.1.2.21 – Coste temporal de funcionalidades del hito 3

Figura 5.1.2.22 – Coste temporal de funcionalidades del hito 4

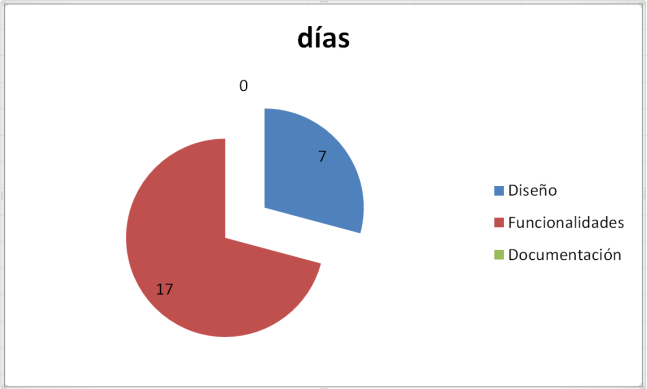
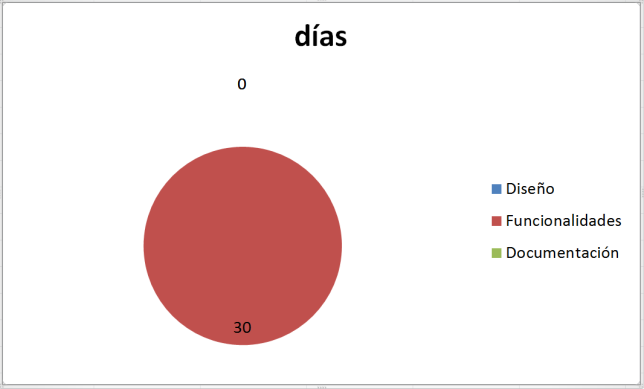


Figura 5.1.2.23 – Coste temporal de funcionalidades del hito 3

Figura 5.1.2.24 – Coste temporal de funcionalidades del hito 4

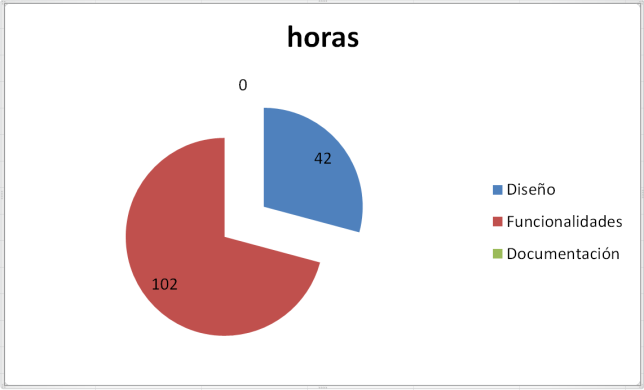
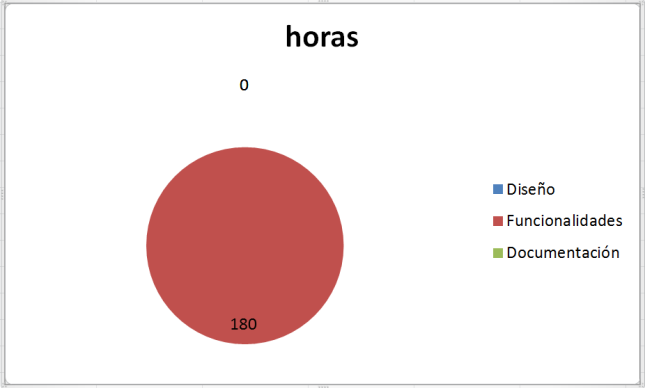


Figura 5.1.2.25 – Coste temporal de funcionalidades del hito 3

Figura 5.1.2.26 – Coste temporal de funcionalidades del hito 4

#### Estimación de costes en Recursos

Los recursos externos que se han especificado durante la definición del proyecto así como sus costes son los siguientes: hosting, dominio, tecnologías.

##### Hosting

El hosting es el espacio en el que se alojará nuestro sistema y desde el que será accesible a los usuarios. En un principio se estudiaron varias opciones:

* Self-Hosting: Habilitar una máquina privada para que haga de servicio de almacenamiento. Esta opción era poco viable tanto en el aspecto técnico como físico. Independientemente de si el cliente tuviese un equipo con el que formar el servidor, las condiciones de la población en la que se ubica no eran propicias para esta opción (fallos de corriente, poca velocidad de internet, etc.). Y por otro lado, aunque todo funcionase a la perfección el cliente no tendría conocimientos para mantener el sistema en caso de mala operatividad.
* Hosting Gratuito: Disponer de espacios de almacenamiento gratuitos como el ofrecido por 000webhost era una posibilidad perfecta para el comienzo del desarrollo, pero el cliente no ha accedido a esta opción, dadas las limitaciones en espacio y en ancho de banda que ofrecen estos servicios. Estas limitaciones podrían afectar gravemente a la experiencia de usuario.
* Hosting de pago: Almacenamiento en un servidor privado de pago en el que las características ofrecidas sean de una calidad media alta. El ancho de banda sería el adecuado para una experiencia de usuario aceptable y no habría que preocuparse del espacio ofrecido. Además cuentan con servicio técnico y muchas herramientas útiles para el cliente.

La opción elegida fue el Hosting de pago y de entre todos los servicios disponibles en el mercado, tanto desarrollador como cliente coincidieron en que uno de los mejores servicios por calidad / precio (según la experiencia del desarrollador) fue [1&1](https://www.1and1.es/alojamiento-web?ac=OM.WE.WEo42K356263T7073a).

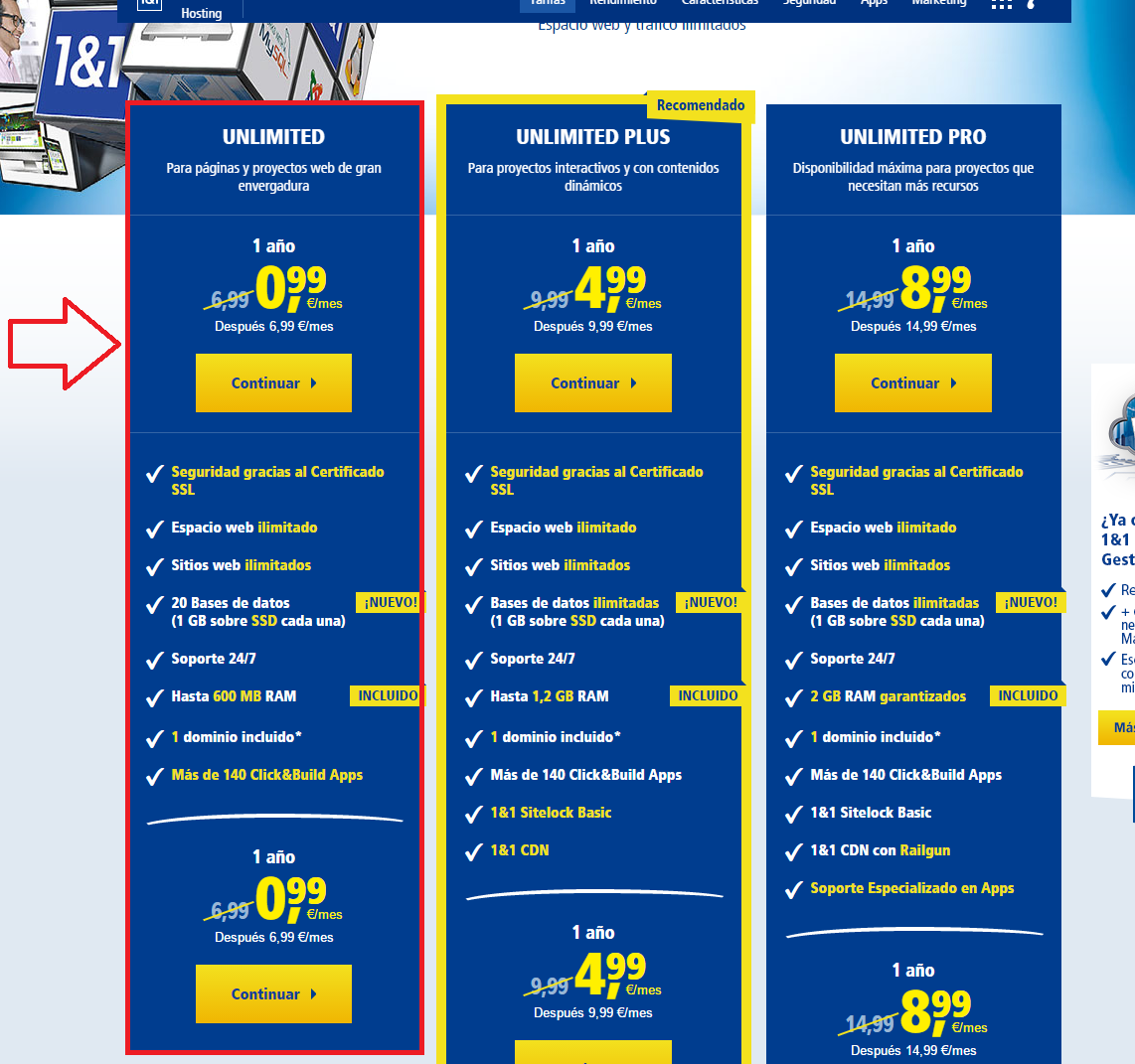


Figura 5.1.2.27 – Servicio de hosting

##### Dominio

El dominio es la dirección mediante la cual acceder al sitio web. Potencialmente, el domino final será, a ser posible, algo así como [www.energiapenolite.com](http://www.energiapenolite.com).

El coste del dominio entra en el precio del paquete de compra del servicio hosting por lo que no sería necesario realizar ningún pago adicional. En caso de querer más dominios se intentarían comprar a la misma empresa que el servicio de alojamiento con tal de mantener todo dentro del mismo organismo y evitar más trabajo al cliente (migraciones, complejas configuraciones, etc.)

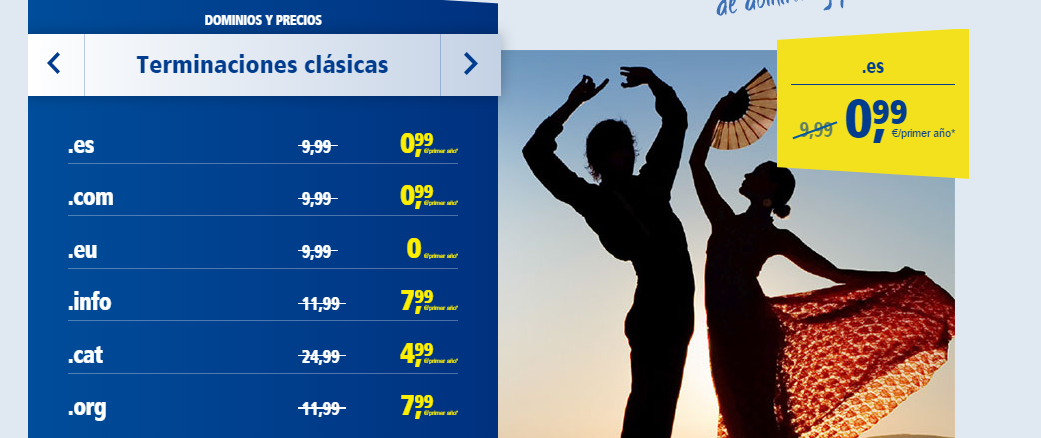


Figura 5.1.2.28 – Dominio

##### Costes de herramientas

Por último contemplaremos el coste de las herramientas a utilizar durante el proyecto. Actualmente las herramientas planificadas para las distintas fases y operaciones del proyecto son las siguientes:

|  |  |  |
| --- | --- | --- |
| Nombre | Utilidad | Coste |
| Microsoft Project 2013 | Planificación / Gestión | 0€ (Versión MSDN) |
| Microsoft Word 2010 | Documentación | 0€ (Versión MSDN) |
| Microsoft Excel 2010 | Documentación | 0€ (Versión MSDN) |
| Atlassian Sourcetree | Gestión versiones | 0€ (Gratuito) |
| Adobe Photoshop | Diseño | 0€ (Versión de prueba) |
| XAMPP | Desarrollo | 0€ (Gratuito) |
| Mozilla Firefox | Desarrollo y Pruebas | 0€ (Gratuito) |
| Google Chrome | Desarrollo y Pruebas | 0€ (Gratuito) |
| Adobe Brackets | Desarrollo | 0€ (Gratuito) |

Figura 5.1.2.29 – Costes de las herramientas de desarrollo

Para finalizar este apartado, podemos resumir la estimación de costes de recursos mediante la siguiente gráfica.

Figura 5.1.2.30 – Costes el primer año. Figura 5.1.2.31 – Costes a partir del primero año

## Estimación de riesgos

Durante el desarrollo de cualquier proyecto existen riesgos que lo pueden retrasar o hacer fracasar. Estos riesgos son causados por imprevistos, problemas de desarrollo o incluso problemas en la planificación del mismo. Con objeto de prevenirlos es imprescindible realizar una estimación de los problemas que puedan acontecer durante el proceso y describir características tales como su probabilidad de aparición o cómo abordar o rodearlo.

En esta sección expondremos los posibles riesgos atendiendo a varias categorías:

### Riesgos tecnológicos.

Las tecnologías empleadas para el desarrollo de cualquier sistema pueden sufrir cambios o incompatibilidades con los operativos o incluso con otras herramientas que estemos empleando. Es por ello que si se diese alguna incompatibilidad con los softwares utilizados deberemos rápidamente investigar una solución. Previendo esta situación los caminos más aconsejables a seguir son:

* Solucionar la incompatibilidad: Si se dispone de tiempo suficiente como para intentarlo, se debe procurar solucionar la incompatibilidad para poder continuar con el transcurso del proyecto.
* Desechar software y buscar sustituto: Solución no óptima porque se corre el riesgo de sufrir otro tipo de incompatibilidades o de pérdida de información.

### Personales

Es probable que durante el desarrollo al personal encargado del progreso de la plataforma le surjan complicaciones o bajas por diversos motivos (enfermedad, asuntos familiares, etc.).

Prevenir este tipo de riesgos es muy difícil. Para nuestro proyecto las consecuencias supondrían un retraso muy difícil de solucionar si el periodo de inactividad es alto dado que el número de desarrolladores es exactamente uno.

Dada la dificultad de prevenir este tipo de cuestiones, la solución más acorde a la situación sería aumentar el número de horas diarias y el sacrificio del tiempo personal del desarrollador.

### Estimaciones

Considerando la poca experiencia en estimaciones del personal a cargo del proyecto una mala estimación de las horas /días de trabajo a dedicar en una tarea es un riesgo más que probable. Para evitar el retraso que se sufriría en caso de que la estimación hubiese quedado corta (se estimaban menos horas de las que ha llevado completar la tarea) se establece un plan de recorte de las funcionalidades más superficiales o poco relevantes de ese hito o un balance con las tareas en que la estimación era superior al tiempo realmente dedicado.

# Metodología

## Metodología de desarrollo

Las metodologías de desarrollo indican de qué manera se va a llevar a cabo en función de características como:

* Tiempo de desarrollo.
* Número de entregas.
* Número de reuniones.
* Tamaño del equipo de desarrollo.
* Roles
* Planificación del proyecto.

En el caso concreto de nuestra plataforma la metodología no corresponderá cien por cien con ninguna establecida. A continuación definiremos la metodología de la que se nutre nuestra aproximación para el desarrollo del software.

* Scrum es el nombre con el que se denomina a los marcos de desarrollo ágiles caracterizados por:
  + Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
  + Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto organizados, que en la calidad de los procesos empleados.
  + Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o en cascada.

(Definción extraida de Wikipedia, enlace en bibliografía)

Una de las características principales de Scrum es que se celebran reuniones todos los días en los que los desarrolladores responden a 3 preguntas esenciales:

* ¿Qué has hecho desde ayer?
* ¿Has tenido algún problema durante el progreso de la tarea?
* ¿Qué es lo que haré hoy?

Esta es una de las características que emplearemos en nuestra metodología, pero con la salvedad de que no se celebrará reunión, simplemente todas las mañanas se responderán a estas preguntas para guiar el desarrollo del proyecto en función del progreso. Esto permite tener una visión más realista, aceptando que el problema no puede ser definido a la exactitud desde el principio y maximizando la capacidad del equipo de desarrollar y actuar frente a los imprevistos. Otra de las características que cumplirá nuestro proyecto será el carácter incremental del sistema.

De este modo observando las características de nuestro proyecto:

* Tiempo de desarrollo MUY limitado.
* Cuatro entregas planificadas.
* Imposibilidad de reuniones debido a los factores que afectan tanto a desarrollador como cliente y el tutor (tiempo, distancia, etc.).
* Trabajo individual.
* Roles:
  + Desarrollador. (Developer team)
  + Tutor. (Scrum Master).
  + Cliente. (Product owner).
* Planificación del proyecto.

Podemos definir nuestra metodología como:

* Una metodología ágil basada en una visión realista que se centra en maximizar la capacidad de desarrollo frente a la posibilidad de que los casos de uso y las funcionalidades pueden variar así como el surgimiento de imprevistos.
* El progreso del proyecto guía su desarrollo.
* Focalización y organización diaria.
* Siguiendo un modelo iterativo en el que se diferencian dos grandes iteraciones:
  + La tienda Online.
  + El panel de control.
* Ambas iteraciones dispondrán de sus fases de desarrollo que seguirán las marcadas por las metodologías de desarrollo de software
  + Análisis del problema: en el que se establecerán todas las funcionalidades obtenidas en los requisitos funcionales y los casos de uso así como su forma de afrontarlas..
  + Diseño: Utilizaremos esta fase para dar lugar al boceto inicial sobre el que se sustentará la estructura general del sistema.
  + Implementación: Desarrollo de todo lo descrito en el análisis del problema.
  + Despliegue: Fase de pruebas y corrección de errores

Algunas de las características, como la planificación del proyecto o las fases de desarrollo secuenciadas y diferenciadas, suenan contradictorias a lo que dicta Scrum. Recordamos que nuestra metodología, por sus características, se nutre de Scrum pero no la cumple en su plenitud.

## Gestión y planificación

La gestión y planificación del proyecto se llevó a cabo en base a la experiencia en la realización de tareas similares en el pasado. La planificación se realizó también con la metodología explicada anteriormente en cuenta de modo que sigan las fases establecidas y además mantengan una carga de trabajo equilibrada en base al tiempo de cada hito.

Para la gestión se utilizarán los siguientes softwares:

* Microsoft Project 2013: Potente herramienta de gestión de proyectos con innumerables herramientas útiles para el seguimiento del mismo. La emplearemos para el diagrama de Gantt y el seguimiento de las tareas a lo largo del desarrollo.
* Atlassian Sourcetree: Herramienta gratuita para la gestión de versiones. Consiste en un cliente que proporciona una interfaz visual para los repositorios GIT y Mercurial.

## Documentación

Como documentación, principal y prácticamente solo encontramos este documento. No obstante, no solo encontramos software de generación de documentos sino también software de tablas de cálculo para las gráficas que podremos encontrar en este documento. Por tanto, los programas empleados para la generación de documentación son:

* Microsoft Word 2010: Se trata de un procesador de texto de los más completos que hay en el mercado. Actualmente de pago pero que podemos encontrar entre los artículos disponibles en la MSDN de la Escuela Politécnica Superior de Alicante de forma gratuita.
* Microsoft Excel 2010: También de Microsoft y también disponible en el MSDN de la Escuela Politécnica Superior de la Universidad de Alicante. Esta herramienta, caracterizada por ser un software de hojas de cálculo empleado sobre todo para la contabilidad, la hemos usado para la generación de los gráficos que encontramos en este documento.

## Diseño

El diseño de esta plataforma abarca diferentes categorías diferentes por tanto vamos a necesitar unas herramientas especializadas para cada una de las clases de diseño.

* Diseño Gráfico: Emplearemos Adobe Photoshop en su versión de evaluación. Esto nos permitirá definir gráficamente algunos elementos que queramos incluir en la interfaz del sistema.
* Diseño de Flujos: Emplearemos herramientas online para la generación de diagramas de flujo como el definido al principio del documento en la [figura 4.1.2](#_4.1__Diagrama). El software empleado es Gliffy. Aunque al ser una versión de prueba de 15 días, transcurrido este tiempo se realizarán mediante la herramienta VISIO (perteneciente a la suite de Microsoft) o clónicos gratuitos como LucidChart.
* Diseño de mockups: Los mockups son bocetos sencillos con los que se define la estructura de la interfaz visual a grandes rasgos. Estos bocetos los realizaremos mediante la herramienta online gratuita llamada FrameBox.

## Desarrollo

La planificación del desarrollo ya ha sido detallada en los apartados anteriores por lo que en este apartado detallaremos el software necesario y que se empleará para la construcción de la plataforma. Cada una de las herramientas listadas a continuación suplirá una necesidad básica en el entorno de trabajo.

* XAMPP: Es un conjunto de herramientas multiplataforma cuyo objetivo es simular un servidor online mediante sus diferentes sistemas: Sistema de gestión de base de datos; servidor web Apache; Intérpretes de lenguajes de Script como PHP y PERL. La necesidad básica que cubre este sistema es la de proporcionar un entorno de trabajo mediante la simulación del servidor.
* Brackets: Es un procesador de texto capaz de reconocer los distintos lenguajes de programación y de diseño empleados. Mediante la coloración visual y algunas de sus herramientas podemos obtener una ayuda adecuada a la hora de desarrollar. Este software ha sido creado desde cero especialmente para desarrolladores web y desarrolladores front-end. De este modo, suple la necesidad de un medio por el cual construir el sistema.
* Google Chrome: Es un navegador web desarrollado por Google. El rol del navegador será primordial en dos aspectos. El aspecto principal es el renderizado del sistema, es decir, la forma en la que se va a ver el sistema. Por otra parte también suple la necesidad de un entorno de pruebas en el que poder detectar los errores. Las principales herramientas que dispone este navegador y que emplearemos para el seguimiento del rendimiento del sistema serán:
  + Consola: Un terminal de desarrollador donde ver todos los errores que se generan en la página web. También es informativa de otros procesos que se realizan como por ejemplo las peticiones AJAX.
  + La pestaña de Elementos: Permite observar el código HTML de la página así como las propiedades CSS de cada uno de los elementos.
  + Network: Una de las herramientas más potentes y que seguramente emplearemos más a menudo durante el progreso del proyecto. Esta pestaña nos permite observar todas las peticiones de todos los tipos que se realizan y datos muy relevantes como los errores que se producen en ellas, los tiempos de carga y ejecución, etc.
  + Fuentes: Aunque hay más herramientas en el inspector, esta será la última que comentaremos pues las demás apenas les daremos uso si se le da alguno en absoluto. Esta pestaña la emplearemos principalmente para el debug. Permite utilizar puntos de fuga, eventos y ejecución paso-a-paso.
* GitHub: Por último, pero no menos importante, dispondremos de un repositorio público en la plataforma para repositorios GitHub, que emplea tecnología Git que usaremos de forma visual mediante el cliente Atlassian Sourcetree.

# Diseño del sistema

## Diseño del Interfaz

### Diseño Ideal vs Diseños Comunes

### 

# Diseño Interfaz

## Diseño Ideal vs Diseños comunes

## Diseño Inicial

## Evolución del diseño

## Diseño Final