



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Portal web de una tienda virtual de productos para animales domésticos

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Pablo Peinado Abad

Tutor: Sergio Sáez Barona

2013-2014

Portal web de una tienda virtual de productos para animales domésticos



Resumen

Este proyecto se basa en el desarrollo de una tienda virtual de productos para animales domésticos utilizando el *framework* de PHP *Yii Framework* y el sistema gestor de bases de datos *MySQL*. También, como en todo desarrollo web, se hace uso del lenguaje de marcado de hipertexto HTML y de las hojas de estilo en cascada CSS. El desarrollo contempla la gestión de un catálogo de productos, la cesta de la compra y pedidos, así como la capacidad de gestionar las cuentas de usuario.

Palabras clave: *Framework* PHP, *Yii Framework*, web, PHP, *MySQL*, HTML, CSS, tienda online.

Tabla de contenidos

| | |
|--|----|
| 1. Introducción..... | 6 |
| 1.1 Objetivos | 6 |
| 1.2 Estructura..... | 7 |
| 2. Especificación de requisitos software..... | 8 |
| 2.1 Propósito | 8 |
| 2.2 Descripción general..... | 9 |
| 2.3 Requisitos funcionales | 10 |
| 2.4 Requisitos no funcionales | 14 |
| 3. Fase de análisis | 15 |
| 3.1 Diagrama entidad-relación | 15 |
| 3.2 Casos de uso..... | 16 |
| Fichas de casos de uso para usuario web..... | 18 |
| Fichas de casos de uso para administrador web..... | 26 |
| 4. Fase de diseño | 31 |
| 4.1 Capa de presentación | 31 |
| 4.2 Capa de persistencia | 35 |
| 4.3 Capa de lógica | 36 |
| 4.4 ¿Por qué Yii Framework? | 38 |
| 5. Detalles de implementación | 40 |
| 5.1 Estructura de ficheros y dependencias | 40 |
| 5.2 Tecnologías utilizadas en el desarrollo | 43 |
| 6. Pruebas | 46 |
| 6.1 Conformidad estándar W3C | 46 |
| 6.2 Visualización en navegadores..... | 48 |
| 6.3 Prueba de uso de la web..... | 60 |
| 7. Conclusiones | 74 |
| 7.1 Técnicas | 74 |
| 7.2 Personales | 74 |
| 7.3 Posibles ampliaciones del proyecto | 75 |
| 8. Bibliografía | 77 |
| 9. Glosario..... | 79 |

1. Introducción

Tras cursar el Grado en Ingeniería Informática en la rama de Tecnologías de la Información me quedó la sensación de que no se profundiza demasiado en el desarrollo de proyectos web con una complejidad mínima y utilizando lenguajes que permitieran elaborar sitios web dinámicos que manejaran bases de datos.

Las asignaturas cursadas abordan otro tipo de aspectos web como el diseño de sitios, o cómo interactúan algunas de las tecnologías web actuales. Analizando esto, y tras un vistazo a las ofertas de trabajo actuales es lo que hace que me decante a escoger este tipo de proyecto para poder, aunque de una forma en parte autodidacta, ampliar mis conocimientos en estas áreas que considero bastante demandadas para los pocos conocimientos adquiridos en ellas a lo largo del grado.

Para tal proyecto he escogido el binomio del lenguaje PHP junto al gestor de bases de datos *MySQL*, la programación en PHP en este caso viene agilizada por un *framework* para este lenguaje, ya que programar en PHP hoy en día puede resultar tedioso para un proyecto que cumpla unos estándares actuales mínimos.

Como *framework* de PHP para mi desarrollo he escogido *Yii Framework*. En la sección 4.4 trataré de explicar porque he tomado esta decisión.

1.1 Objetivos

Diseñar e implementar un sitio web que gestionará una tienda virtual para la venta por internet de productos para animales domésticos utilizando un *framework* PHP. La aplicación gestionará los usuarios, el catálogo de productos y los pedidos, no así las pasarelas de pago seguras ya que quedan en otro plano a este proyecto, todo esto poniendo atención a su adecuada presentación web. También se ofrecerá una búsqueda avanzada (categorías, razas, marcas...) así como la navegación por categorías y la gestión de la cesta de la compra.

Como objetivos personales están el aprender a utilizar un *framework* de PHP, en este caso *Yii Framework*, así como el manejo del propio lenguaje PHP, además de un gestor de base de datos como es el *MySQL*. También están el mejorar el conocimiento del lenguaje HTML, así como de las hojas de estilo CSS, para poder realizar desarrollos web con un nivel adecuado en cuanto a presentación y diseño de la interfaz.

1.2 Estructura

En este proyecto empezaré exponiendo los requisitos software que cumplirá mi desarrollo web según la norma del IEEE 830-1998, continuare con la fase de análisis en el punto tres donde realizaré un estudio de los casos de uso de todos los actores que intervendrían en la aplicación y veremos también el diagrama de clases.

Continuaré explicando la fase de diseño en el punto cuatro donde separaré mi desarrollo en 3 capas:

- La capa de presentación, que será diseñada a partir de unos bocetos digitales o *wireframes*.
- La capa de persistencia, donde presentaré su esquema lógico y comentaremos las relaciones entre tablas.
- Por último, para abordar la capa de lógica, se hará un estudio a nivel de control, donde veremos el flujo de trabajo de mi desarrollo mediante un diagrama.

Una vez concluidas las fases de diseño comentaremos en el punto cinco los detalles de implementación que he llevado a cabo para mi desarrollo web, en este punto echaremos un vistazo a la estructura de ficheros resultante, sus dependencias y funciones así como las tecnologías y el software que he utilizado para mi proyecto, omitiremos la instalación y configuración de este ya que excedería el volumen para los objetivos de esta memoria.

Pasaremos al punto seis, en el que probaremos el comportamiento del desarrollo bajo distintas pruebas, tanto de conformidad bajo el estándar del *w3c Markup Validation Service* <http://validator.w3.org/> que hará un chequeo del marcado correcto de los documentos HTML y CSS contenidos en mi trabajo para ver si atienden a la conformidad que dicta el w3c. En este punto también probaremos el resultado visual de probar la aplicación web bajo distintos navegadores y resoluciones, y, aunque no tiene una versión adaptada a dispositivos móviles, veremos si la navegabilidad se ve afectada desde estos.

Seguidamente, en este mismo punto haremos las pruebas de uso, donde probaremos toda la funcionalidad que ofrece la aplicación explicándola paso a paso y ayudándome de capturas de pantalla veremos el flujo sobre las distintas pantallas. Una vez concluidas las pruebas expondré las conclusiones, divididas en las conclusiones personales, y en conclusiones técnicas. Por último se incluye la bibliografía utilizada para llevar a cabo el desarrollo, donde se citarán los libros, manuales, URLs consultados y para terminar un glosario de términos que han aparecido en esta memoria ordenados de forma alfabética.

2. Especificación de requisitos software

2.1 Propósito

Esta sección va dedicada a la especificación de requisitos software bajo el estándar IEEE 830/1998 para mi proyecto de fin de grado.

Alcance

Especifica los requisitos software de la tienda virtual de productos para animales de compañía peipet.com

Personal involucrado

| | |
|-------------------------|--|
| Nombre | Pablo Peinado Abad |
| Rol | Único desarrollador de la aplicación |
| Categoría profesional | Estudiante de Grado en Ingeniería Informática UPV |
| Responsabilidades | Desarrollar la aplicación integra y hacer la memoria |
| Información de contacto | pabpeiab@inf.upv.es |

| | |
|-------------------------|--|
| Nombre | Sergio Sáez Barona |
| Rol | Director del proyecto |
| Categoría profesional | Doctor en Informática y Profesor Titular de la Universidad |
| Responsabilidades | Supervisar el TFG de Pablo Peinado Abad |
| Información de contacto | ssaez@disca.upv.es |

Referencias

| Referencia | Título | Ruta | Fecha | Autor |
|------------|--------------------------------|----------------|----------|--------------------|
| 1 | Memoria TFG Pablo Peinado Abad | Este documento | 12/08/14 | Pablo Peinado Abad |

2.2 Descripción general

Perspectiva del producto

Este desarrollo es una tienda online de productos para animales domésticos. Es independiente a cualquier otro desarrollo (no tiene dependencias externas), está alojada para acceso de forma local, y, excepcionalmente, de forma remota en peipet.dynsns-web.com para comprobaciones del tutor.

Funcionalidad del producto

Este portal pretende gestionar una tienda virtual para la venta por internet de productos para animales domésticos. Dado que esta página únicamente está realizada con fines didácticos como trabajo de fin de grado, su funcionalidad para la venta online estará limitada. La aplicación gestionará los usuarios, el catálogo de productos y los pedidos, no así las pasarelas de pago seguras ya que quedan en otro plano a este proyecto, todo esto poniendo atención a su adecuada presentación web. También se ofrecerá una búsqueda avanzada (razas, marcas, etc.) así como la navegación por categorías y la gestión de la cesta de la compra.

Características de los usuarios

| | |
|-----------------|-----------------------------|
| Tipo de usuario | Cliente de la tienda online |
| Formación | Varios perfiles diferentes |
| Habilidades | Varios perfiles diferentes |
| Actividades | Compra online de la tienda |

| | |
|-----------------|------------------------------------|
| Tipo de usuario | Administrador del portal |
| Formación | Grado en Ingeniería Informática |
| Habilidades | Especialista en administración web |
| Actividades | Administrar el sitio web |

Restricciones

El portal correrá bajo la máquina del alumno en el que fue desarrollado, y únicamente se le brindará acceso remoto al tutor a través de peipet.dynsns-web.com, que es un proveedor de resolución de nombres para IPs dinámicas, para labores de comprobación del trabajo por parte del tutor. El ancho de banda del servidor está limitado a 10 Megas de subida ya que es una conexión con fines domésticos y el portal no será explotado con fines comerciales.

Suposiciones y dependencias

La web está diseñada para correr en un servidor bajo la plataforma Apache/MySQL, y puede ser accedido por todos los navegadores de escritorio, dado que no está adaptado a dispositivos móviles desde estos se podrá acceder pero bajo restricciones de funcionalidad.

Evolución previsible del sistema

Dado que es un desarrollo como trabajo de fin de grado, en principio el desarrollo finaliza en el momento de la defensa del proyecto. Cabe la posibilidad de ampliación en un futuro para uso con fines comerciales, con lo que se desarrollaría la pasarela de pago y se mejoraría la gestión de los pedidos, así como funcionalidades adicionales, como la adaptación a dispositivos móviles, esto se comenta en profundidad en el punto 7.3, posibles ampliaciones del proyecto.

2.3 Requisitos funcionales

Enumeraremos aquí en forma de tabla los requisitos principales de la aplicación que deberá satisfacer el desarrollo

| | |
|-------------------------|---|
| Número de requisito | 1 |
| Nombre de requisito | Consultar catálogo de productos |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Requisitos TFG |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |
| Comentarios | Se podrá consultar el catálogo entero de productos por parte del usuario de la web sin tener que estar validado/dado de alta |

| | |
|-------------------------|---|
| Número de requisito | 2 |
| Nombre de requisito | Búsqueda en el catálogo por marca/raza/categoría o por palabras clave en el campo descripción. |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Requisitos TFG |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |

| | |
|-------------------------|---|
| Número de requisito | 3 |
| Nombre de requisito | Se deberá ordenar los productos del catálogo según un criterio de ordenación |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Requisitos TFG |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |
| Comentarios | Se deben poder ordenar los productos por precio. |

| | |
|-------------------------|---|
| Número de requisito | 4 |
| Nombre de requisito | Existirá una cesta funcional sin que el usuario esté validado en el sistema. |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Requisitos TFG |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |
| Comentarios | Además, esta permite añadir artículos y variar el número de productos que contiene mediante controles. |

| | |
|-------------------------|--|
| Número de requisito | 5 |
| Nombre de requisito | El sistema permitirá hacer un pedido a usuarios validados en el sistema |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Requisitos TFG |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |
| Comentarios | Si no está validado, se redirigirá a la pantalla de <i>login</i> y permitirá la posibilidad de registrarse. Una vez confirmado el pedido, el cliente recibe un email en su cuenta con un adjunto en PDF con los detalles del pedido. |

| | |
|-------------------------|---|
| Número de requisito | 6 |
| Nombre de requisito | El sistema permitirá registrarse como cliente a clientes no dados de alta. |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Requisitos TFG |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |
| Comentarios | Utilizando como nombre de usuario su email y cifrando la contraseña antes de almacenarla en la base de datos. |

Portal web de una tienda virtual de productos para animales domésticos

| | |
|-------------------------|--|
| Número de requisito | 7 |
| Nombre de requisito | El sistema permitirá actualizar los datos de usuarios ya dados de alta. |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Requisitos TFG |
| Prioridad del requisito | <input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional |
| Comentarios | Accediendo a su área personal o al confirmar un pedido. |

| | |
|-------------------------|--|
| Número de requisito | 8 |
| Nombre de requisito | El sistema permitirá mandar una solicitud de contacto por parte del cliente que llega al email corporativo de la empresa. |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Requisitos TFG |
| Prioridad del requisito | <input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional |
| Comentarios | En este caso a tiendapeipet@gmail.com , como medida de seguridad se incluirá un <i>captcha</i> , para evitar que se manden emails indiscriminados por medio de software malicioso. |

| | |
|-------------------------|--|
| Número de requisito | 9 |
| Nombre de requisito | El sistema da información sobre la localización de la empresa. |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Requisitos TFG |
| Prioridad del requisito | <input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional |
| Comentarios | Utilizará un widget con <i>Google Maps</i> indicando una localización ficticia que apuntará a la UPV. |

| | |
|-------------------------|--|
| Número de requisito | 10 |
| Nombre de requisito | El portal ofrecerá información sobre la tienda y ofertas en la página principal. |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Requisitos TFG |
| Prioridad del requisito | <input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/Opcional |
| Comentarios | A poder ser, se emplearán recursos visuales llamativos como el uso de <i>sliders</i> . |

| | |
|-------------------------|--|
| Número de requisito | 11 |
| Nombre de requisito | El sistema ofrecerá gestión de usuarios, con permisos asignados a sus distintos roles. |
| Tipo | <input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción |
| Fuente del requisito | Requisitos TFG |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional |
| Comentarios | Se discriminará entre usuarios validados y no validados, estos últimos, podrán consultar el catálogo de productos, llenar una cesta sin pasarla a pedido, registrarse como nuevos usuarios, mandar solicitud de contacto y obtener información comercial de la página. Los usuarios registrados, una vez validados además, pueden pasar una cesta a pedido y modificar sus datos personales. Por último, el administrador del sitio web puede hacer todo lo anterior y añadir nuevos productos al catálogo, añadir nuevas razas/marcas/categorías, administrar los usuarios registrados etc. |

| | |
|-------------------------|---|
| Número de requisito | 12 |
| Nombre de requisito | La web se podrá visionar de manera correcta en los principales navegadores como son: Internet Explorer, Chrome, Firefox y Safari. |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Requisitos TFG |
| Prioridad del requisito | <input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional |
| Comentarios | No se garantizará la correcta visualización en navegadores de dispositivos móviles, quedando este aspecto como una posible ampliación del proyecto. |

Requisitos comunes de los interfaces

El portal permite la introducción de datos de usuario por medio de los navegadores de forma remota y local por medio de la entrada estándar de teclado y ratón, así como la navegación por la interfaz.

Interfaces de usuario

Las interfaces deberán ser sencillas para que los usuarios naveguen por ellas de forma cómoda e intuitiva, además deberán ser en todo lo posible visualmente atractivas.

Interfaces de hardware

En este desarrollo interviene la máquina que hace de servidor web, y las máquinas cliente que accede vía web al portal.

Interfaces de software

El cliente únicamente accede a la aplicación mediante un navegador de escritorio, y la interfaz es la de la propia página web, no hay otras interfaces.

Interfaces de comunicación

Se comunican el servidor web y el cliente mediante el protocolo HTTP.

2.4 Requisitos no funcionales

Requisitos de rendimiento

Dado que el volumen de usuarios no va a ser importante y el proyecto ha sido desarrollado únicamente con fines didácticos el sistema no soportará gran volumen de usuarios accediendo concurrentemente. Sí que debe permitir un buen rendimiento para unos pocos usuarios accediendo de forma local o remota para hacer pruebas.

Seguridad

- Se emplea un cifrado de las contraseñas cuando se almacenan en la base de datos para una mayor seguridad.
- Se emplean *captchas* para determinar que es un humano quien está introduciendo datos en la aplicación y no un *software* malicioso.
- Se requiere validación para emitir pedidos y ver o modificar datos personales.

Fiabilidad

El número de errores permisible debe ser mínimo, dada la pequeña envergadura del proyecto. Las operaciones permitidas deben desarrollarse sin incidentes.

Disponibilidad

No es requerida disponibilidad ya que es un proyecto con fines didácticos, únicamente la web deberá estar disponible para su comprobación local o remota cuando el tutor lo requiera.

Mantenibilidad

Las tareas de mantenimiento se basan exclusivamente en introducir nuevos productos si así se requiere. Las copias de seguridad, como es un proyecto que se da por concluido y no es previsible que crezca en complejidad, una vez hecho el respaldo oportuno al final de este, no es necesario que se hagan periódicamente.

Portabilidad

La portabilidad del proyecto se podrá llevar a cabo respaldando las carpetas “htdocs” y “mysql” contenidas en la raíz del directorio donde está instalado Xampp (en nuestro caso), el proceso deberá hacerse con los servicios Apache y MySQL parados. Estas carpetas nos permitirán tener operativa la página en otro servidor distinto que este configurado para ser un servidor web con Apache y MySQL instalados, copiándolas en el lugar oportuno.

3. Fase de análisis

3.1 Diagrama entidad-relación

Representaré las entidades relevantes de mi aplicación así como sus interrelaciones y propiedades mediante un diagrama de clases o entidad-relación, no he determinado los atributos que contienen las clases ya que lo dejo para la fase de diseño. Si mencionare aquí la relación entre las tablas y la multiplicidad.

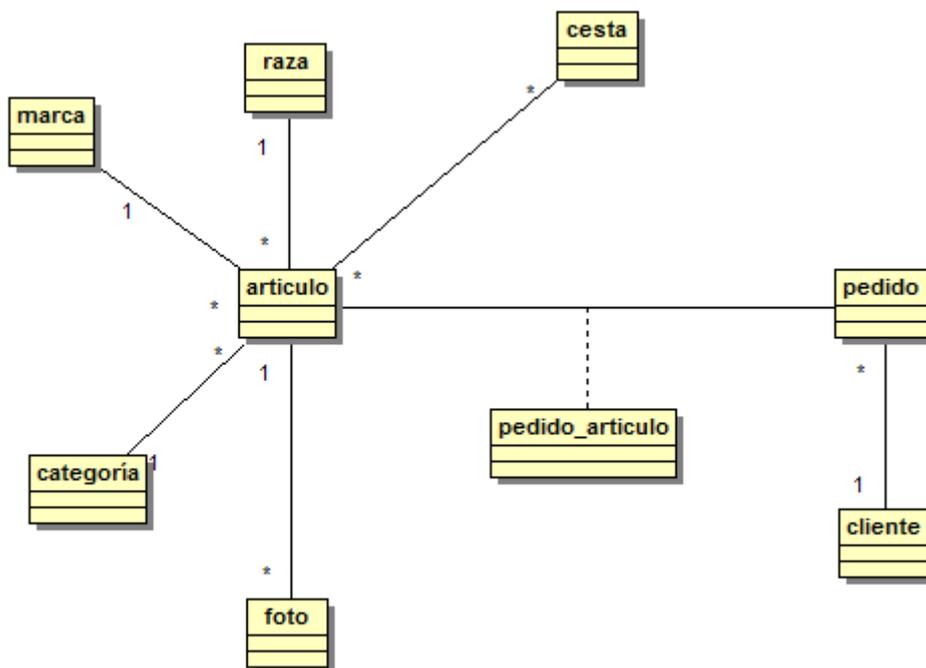


Fig.o Diagrama de clases.

Como comprobamos, un artículo sólo puede pertenecer a una raza, a una marca y a una categoría, mientras que cada una de estas tres clases pueden contener ninguno,

uno, o varios artículos. Un artículo puede tener ninguna, una, o varias fotos, pero una foto sólo puede pertenecer a un artículo. Una cesta puede tener contener ninguno, uno, o varios artículos y lo mismo ocurre con las cestas en relación con los artículos. Un cliente puede tener ninguno, uno, o varios pedidos mientras que cada pedido sólo puede pertenecer a un cliente y por último he de aclarar que la clase pedido_articulo, es una clase asociación, esto viene a representar que en un pedido pueden haber varios artículos, con lo que habrá un registro por cada artículo perteneciente a un pedido, es decir, cada registro contendrá únicamente los ID de las dos clases.

3.2 Casos de uso

En este apartado, en primer lugar, veremos los diagramas de casos de uso para luego hacer una descripción más detallada mediante una ficha por cada caso de uso. He de mencionar que he omitido el recuadro en los diagramas que agrupa todo bajo la aplicación web de Peipet por simplicidad, es decir el que aparece solo el actor y el cuadrado, ya que se sobreentiende que estos corresponden a este desarrollo.

Tenemos dos roles diferenciados, el usuario de la web, y el administrador web, empezemos por el diagrama del primero:

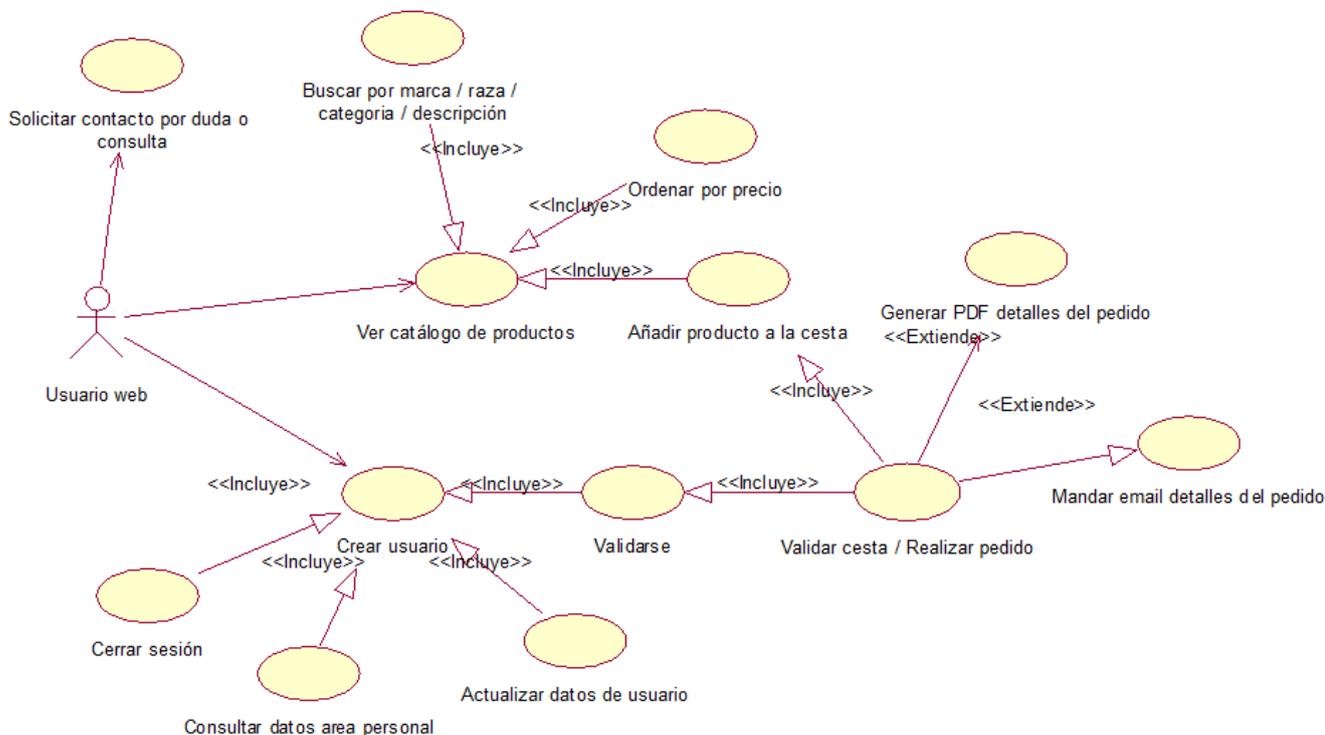


Fig.1 Diagrama de casos de uso del usuario web.

Pasaremos ahora a ver los diagramas del administrador web:

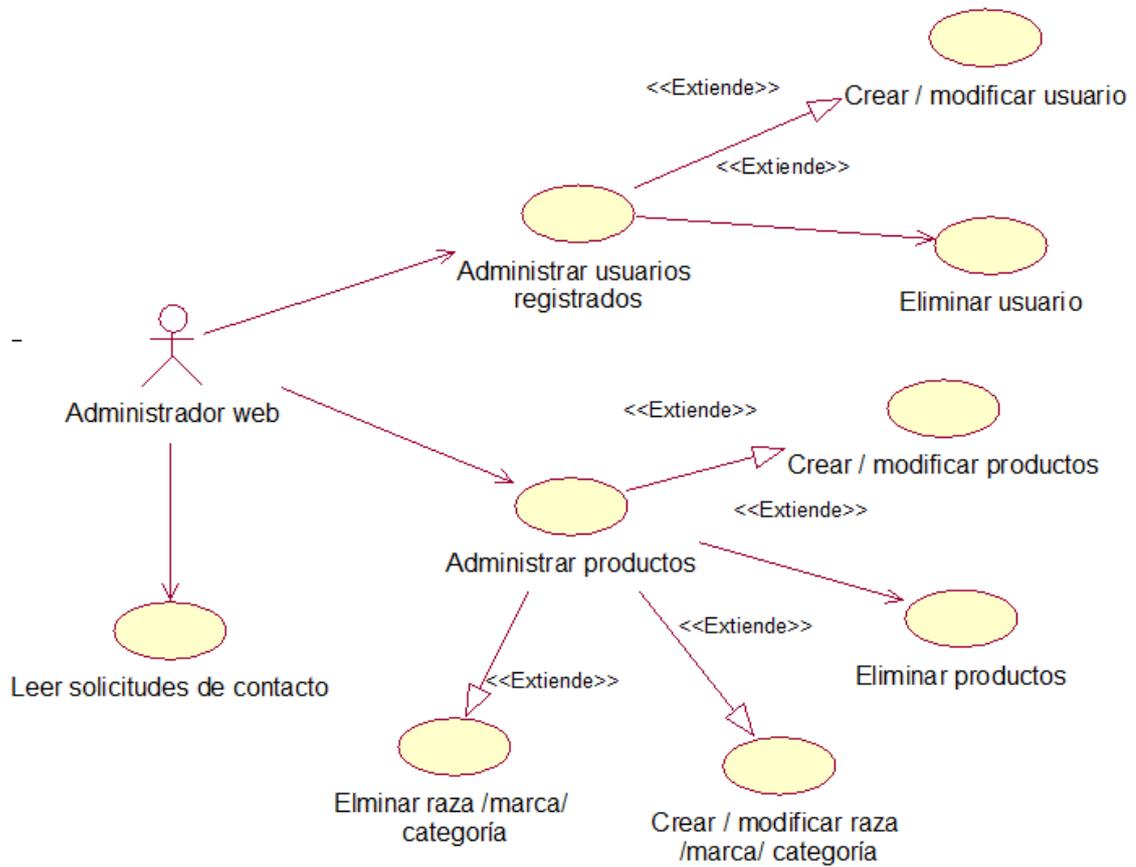


Fig.2 Diagrama de casos de uso del administrador web.

He dividido también las fichas entre los dos tipos de usuario del sistema, comenzaremos con las del usuario web:

Fichas de casos de uso para usuario web

| | | |
|-------------------------|---|--|
| RF- 01 | Ver catálogo de productos | |
| Descripción | El usuario navega por el catálogo de productos de la web. | |
| Precondición | Ninguno, cualquier usuario validado o no puede ver el catálogo. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario accede en la web. |
| | 2 | El usuario pincha en la opción “Productos” del menú principal. |
| Postcondición | Ninguna | |
| Excepciones | Paso | Acción |
| | 1 | No tiene. |
| Comentarios | - | |

| | | |
|-------------------------|--|--|
| RF- 02 | Buscar por marca / raza / categoría / descripción | |
| Descripción | El usuario navega por el catálogo de productos de la web, y tiene disponible el botón de búsqueda avanzada o los menús laterales con las opciones indicadas. | |
| Precondición | Seleccionar los criterios de búsqueda deseados. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario accede en la web. |
| | 2 | El usuario pincha en la opción “Productos” del menú principal. |
| | 3 | Pincha en “BÚSQUEDA AVANZADA” o bien en las opciones del menú lateral. |
| Postcondición | Ninguna | |
| Excepciones | Paso | Acción |
| | 1 | No tiene. |

| | | |
|-------------------------|---|--|
| RF- 03 | Ordenar por precio | |
| Descripción | El usuario puede ordenar los productos por precio en orden creciente/decreciente. | |
| Precondición | Ninguno, cualquier usuario validado o no puede ver el catálogo. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario accede en la web. |
| | 2 | El usuario pincha en la opción “Productos” del menú principal. |
| | 3 | El usuario pincha en “Precio” situado encima de los productos. |
| Postcondición | Los productos se ordenan por precio. | |
| Excepciones | Paso | Acción |
| | 1 | No tiene. |
| Comentarios | Alternando clicks se alterna entre orden creciente y decreciente. | |

| | | |
|-------------------------|--|--|
| RF- 04 | Añadir producto a la cesta | |
| Descripción | El usuario puede añadir un producto a su cesta, esta cesta funciona por sesión y es única para él. | |
| Precondición | Ninguno, cualquier usuario validado o no puede añadir productos a la cesta. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario accede en la web. |
| | 2 | El usuario pincha en la opción “Productos” del menú principal. |
| | 3 | El usuario pincha en el logotipo de cesta en el producto deseado y este se añade a su cesta. |
| Postcondición | El producto se añade a su cesta. | |
| | Paso | Acción |

| | | |
|--------------------|---|----------|
| | 1 | No tiene |
| Comentarios | El producto queda añadido a su cesta, si pincha repetidas veces añadirá más unidades de este producto, una vez. | |

| | | |
|-------------------------|--|--|
| RF- 05 | Validar cesta / realizar pedido | |
| Descripción | El usuario una vez comprobado que los artículos añadidos son los correctos, confirma la cesta, y esta pasa a pedido. | |
| Precondición | El usuario debe estar registrado y validado en el sistema. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario accede en la web. |
| | 2 | El usuario pincha en la opción “Productos” del menú principal. |
| | 3 | Añade los deseados a la cesta. |
| | 4 | Pincha en “Hacer pedido”. |
| 5 | Una vez allí se le mostrarán los detalles del pedido y podrá confirmarlos. | |
| Postcondición | El pedido queda generado, el usuario recibirá un email con los detalles del pedido en PDF. | |
| Excepciones | Paso | Acción |
| | 1 | El usuario debe estar registrado y validado. |
| Comentarios | El usuario puede imprimir los detalles del pedido a modo de presupuesto antes de generar el pedido. | |

| | | |
|-------------------------|---|--|
| RF- 06 | Generar PDF detalles del pedido | |
| Descripción | El usuario puede imprimir un PDF a modo de presupuesto antes de validar el pedido. | |
| Precondición | Validar cesta. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario accede en la web. |
| | 2 | El usuario pincha en la opción “Productos” del menú principal. |
| | 3 | Añade los deseados a la cesta. |
| | 4 | Pincha en “Hacer pedido”. |
| | 5 | Pincha en “Imprimir pedido”. |
| Postcondición | Sale en pantalla el PDF generado. | |
| Excepciones | Paso | Acción |
| | 1 | El usuario debe estar registrado y validado. |
| Comentarios | Sale en pantalla el PDF generado, el usuario puede guardarlo en su disco o imprimirlo, a modo de presupuesto. | |

| | | |
|-------------------------|---|--|
| RF- 07 | Mandar emails con detalles del pedido | |
| Descripción | El usuario recibe un email con un PDF con los detalles de su pedido | |
| Precondición | Haber hecho un pedido. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario accede en la web. |
| | 2 | El usuario pincha en la opción “Productos” del menú principal. |
| | 3 | Añade los deseados a la cesta. |

| | | |
|----------------------|-------------|---|
| | 4 | Pincha en “Hacer pedido”. |
| | 4 | Pincha en “Realizar pedido”. |
| Postcondición | Ninguna | |
| Excepciones | Paso | Acción |
| | 1 | Usuario registrado y validado que genera un pedido. |
| Comentarios | - | |

| | | |
|-------------------------|--|---|
| RF- 08 | Solicitar contacto por duda o consulta | |
| Descripción | El usuario puede solicitar información mediante un formulario de contacto. | |
| Precondición | Ninguna. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario accede en la web. |
| | 2 | El usuario pincha en la opción “Contacto” del menú principal. |
| | 3 | Rellena formulario y envía. |
| Postcondición | Llega un email al administrador del sitio con la consulta del usuario. | |
| Excepciones | Paso | Acción |
| | 1 | No tiene. |
| Comentarios | - | |

| | | |
|-------------------------|---|---|
| RF- 09 | Validarse | |
| Descripción | El usuario hace <i>login</i> en la web. | |
| Precondición | Tener un usuario ya registrado. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario accede en la web. |
| | 2 | El usuario pincha en la opción “Acceso área clientes” del menú principal. |
| | 3 | El usuario introduce sus credenciales y acepta. |
| Postcondición | El usuario inicia sesión en la web. | |
| Excepciones | Paso | Acción |
| | 1 | Si no está registrado el sistema le avisará y no le dejará entrar. |
| Comentarios | El usuario tiene el botón para registrarse a la vista en la ventana de <i>login</i> . | |

| | | |
|-------------------------|-------------------------------|--|
| RF- 10 | Cerrar sesión | |
| Descripción | El usuario cierra sesión. | |
| Precondición | Estar con una sesión abierta. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario esta validado. |
| | 2 | El usuario pincha en la opción cerrar sesión del menú principal. |
| Postcondición | Se cierra la sesión. | |
| Excepciones | Paso | Acción |
| | 1 | El usuario no estaba dentro de una sesión. |
| Comentarios | - | |

| | | |
|-------------------------|--|---|
| RF- 11 | Crear usuario | |
| Descripción | El usuario puede crear un nuevo usuario en el sistema. | |
| Precondición | No tener un usuario registrado en el sistema. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario accede en la web. |
| | 2 | El usuario pincha en la opción “Acceso área clientes” del menú principal. |
| | 3 | Al usuario se le da la opción de registro mediante un botón en esta pantalla. |
| | 4 | El usuario rellena un formulario de registro y acepta. |
| Postcondición | El usuario queda registrado. | |
| Excepciones | Paso | Acción |
| | 1 | Si no rellena los campos requeridos no puede registrarse. |
| Comentarios | - | |

| | | |
|-------------------------|--|--|
| RF- 12 | Actualizar datos de usuario | |
| Descripción | El usuario puede modificar sus datos siendo un usuario registrado. | |
| Precondición | Ser un usuario registrado. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario accede en la web. |
| | 2 | El usuario accede a su área personal. |
| | 3 | El usuario confirma sus datos, si no son correctos pincha en “Modificar datos personales”. |

| | | |
|----------------------|--|---|
| | 4 | Aparece un formulario con sus datos ya rellenos, modifica los datos que desea y acepta. |
| Postcondición | Los datos actualizados quedan registrados. | |
| Excepciones | Paso | Acción |
| | 1 | No deja aceptar si no se rellenan todos los campos requeridos. |
| Comentarios | - | |

| | | |
|-------------------------|--|---------------------------------------|
| RF- 13 | Consultar datos área personal | |
| Descripción | El usuario puede consultar sus datos siendo un usuario registrado. | |
| Precondición | Ser un usuario registrado. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El usuario accede en la web. |
| | 2 | El usuario accede a su área personal. |
| Postcondición | Ninguna | |
| Excepciones | Paso | Acción |
| | 1 | Ninguna. |
| Comentarios | - | |

Fichas de casos de uso para administrador web

| | | |
|-------------------------|--|---|
| RF- 14 | Crear / modificar usuario | |
| Descripción | El administrador podría crear o modificar usuario en caso de que le fuera comunicado algún problema con la cuenta. | |
| Precondición | En el caso de modificar, que el usuario que se modifica este dado de alta en el sistema. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El administrador accede a la ruta de administración de usuarios no accesible mediante la interfaz web convencional y sin permisos de acceso para usuarios normales. |
| | 2 | Crea o modifica el usuario. |
| Postcondición | El usuario queda creado o modificado. | |
| Excepciones | Paso | Acción |
| | 1 | Debe rellenar todos los datos que se requieran, como en el caso de ser una creación modificación de una cuenta normal. |
| Comentarios | - | |

| | | |
|-------------------------|--|---|
| RF- 15 | Eliminar usuario | |
| Descripción | El administrador podría eliminar usuarios en caso de que le fuera comunicado algún problema con la cuenta. | |
| Precondición | La cuenta de usuario debe estar registrada en el sistema. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El administrador accede a la ruta de administración de usuarios no accesible mediante la interfaz web convencional y sin permisos de acceso para usuarios normales. |

| | | |
|----------------------|-----------------------------|---------------------|
| | 2 | Elimina el usuario. |
| Postcondición | El usuario queda eliminado. | |
| Excepciones | Paso | Acción |
| | 1 | Ninguna. |
| Comentarios | - | |

| | | |
|-------------------------|---|--|
| RF- 16 | Crear / modificar productos | |
| Descripción | El administrador puede crear o modificar productos para mantener actualizado el catálogo. | |
| Precondición | En el caso de modificar, que el producto que se modifica debe estar dado de alta en el sistema. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El administrador accede a la ruta de administración de productos no accesible mediante la interfaz web convencional y sin permisos de acceso para usuarios normales. |
| | 2 | Crea o modifica el producto. |
| Postcondición | El producto queda creado o modificado. | |
| Excepciones | Paso | Acción |
| | 1 | Debe rellenar todos los datos que se requieran, como en el caso de ser una creación modificación de producto. |
| Comentarios | - | |

| | | |
|-------------------------|--|--|
| RF- 17 | Eliminar productos | |
| Descripción | El administrador puede eliminar productos para mantener actualizado el catálogo. | |
| Precondición | El producto que se modifica debe estar dado de alta en el sistema. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El administrador accede a la ruta de administración de productos no accesible mediante la interfaz web convencional y sin permisos de acceso para usuarios normales. |
| | 2 | Elimina el producto. |
| Postcondición | El producto queda eliminado. | |
| Excepciones | Paso | Acción |
| | 1 | Ninguna. |
| Comentarios | - | |

| | | |
|-------------------------|---|--|
| RF- 18 | Crear / modificar raza / marca / categoría | |
| Descripción | El administrador crear / modificar razas / marcas / categorías para mantener actualizado el catálogo. | |
| Precondición | En caso de modificar la raza / marca / categoría debe estar dada de alta en el sistema. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El administrador accede a la ruta de administración de productos no accesible mediante la interfaz web convencional y sin permisos de acceso para usuarios normales. |
| | 2 | Crear / modificar raza / marca / categoría. |
| Postcondición | La raza / marca / categoría queda creada / modificada. | |
| Excepciones | Paso | Acción |
| | 1 | Ninguna. |

| | | |
|-------------------------|--|--|
| RF- 19 | Eliminar raza / marca / categoría | |
| Descripción | El administrador eliminar razas / marcas / categorías para mantener actualizado el catálogo. | |
| Precondición | La raza / marca / categoría debe estar dada de alta en el sistema. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El administrador accede a la ruta de administración de productos no accesible mediante la interfaz web convencional y sin permisos de acceso para usuarios normales. |
| | 2 | Elimina raza / marca / categoría. |
| Postcondición | La raza / marca / categoría queda eliminada. | |
| Excepciones | Paso | Acción |
| | 1 | La raza / marca / categoría no existe en el sistema. |
| Comentarios | - | |

| | | |
|-------------------------|--|---|
| RF- 20 | Leer solicitudes de contacto | |
| Descripción | El administrador puede leer solicitudes de contacto, en el caso de que sea su función. | |
| Precondición | La solicitud ha sido generada por el usuario en la pestaña de contacto. | |
| Secuencia Normal | Paso | Acción |
| | 1 | El administrador accede a su correo y lee el email que ha generado la aplicación automáticamente. |
| Postcondición | Responde la solicitud. | |
| Excepciones | Paso | Acción |
| | 1 | Ninguna. |

| | |
|--------------------|--|
| Comentarios | En el caso de que el administrador se encargue de contestar las solicitudes. |
|--------------------|--|

4. Fase de diseño

Pasamos a la fase de diseño, esta fase se ha dividido en tres secciones; presentación, donde mostraré los bocetos del *layout* de la web hechos mediante una herramienta software (*Balsamiq Mockups*), para pasar a la persistencia, donde mostraré el diseño de la base de datos mediante un diagrama de clases que representará las tablas extraído de la aplicación phpMyAdmin, y por último mostraremos un diagrama de flujo o un mapa web para representar la navegación por esta.

Esta fase adapta el diseño de la aplicación al patrón MVC (Modelo, Vista, Controlador), explicaré con más detalle en el punto 5.1 este modelo de diseño, pero para resumir podríamos decir que divide la aplicación en tres capas, el modelo, que agrupa las características y atributos de los datos, las vistas, que son las encargadas de representar la información en la pantalla, y los controladores, que actúan como mediadores entre la solicitud del usuario y los modelos. Esta es una terminología que utiliza *Yii framework*, pero quizá, nosotros estamos más familiarizados por su uso en los estudios de grado, a utilizar las tres capas con los siguientes nombres: Presentación, lógica y persistencia, que es cómo voy a nombrar los apartados de esta fase.

4.1 Capa de presentación

Empezamos con los bocetos o *wireframes* realizados para diseñar la distribución de los ítems en la página. Incluiré solo los principales, el objetivo de esta sección es tener una idea de cómo será la distribución de los elementos a lo largo de todo el sitio web.

Pantalla principal

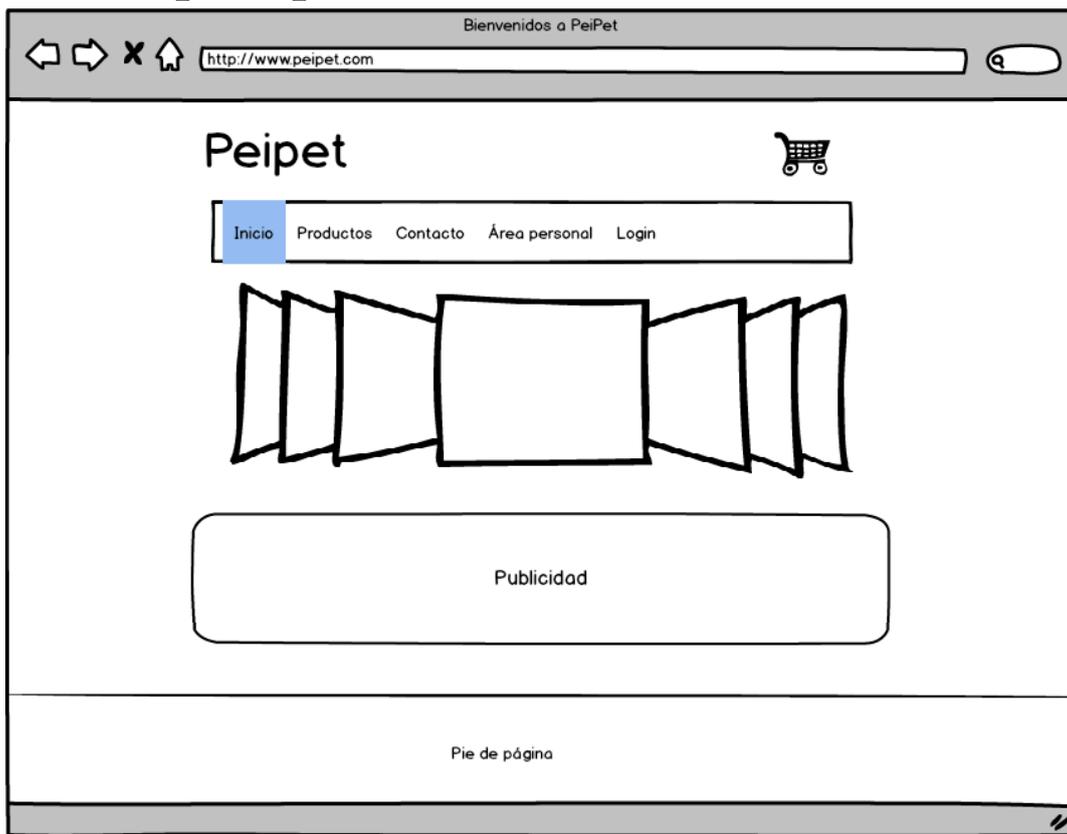


Fig.3 Wireframe pantalla principal

Productos

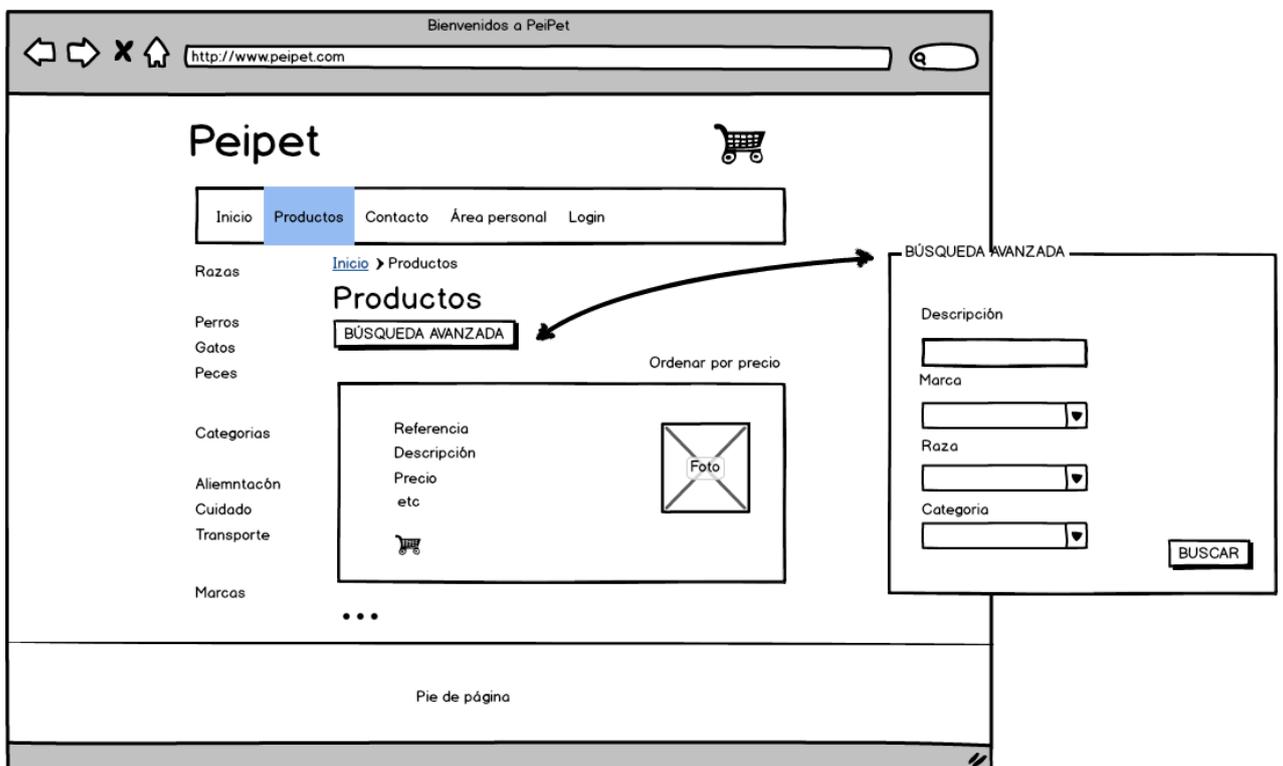


Fig.4 Wireframe pantalla productos.

Cesta de la compra

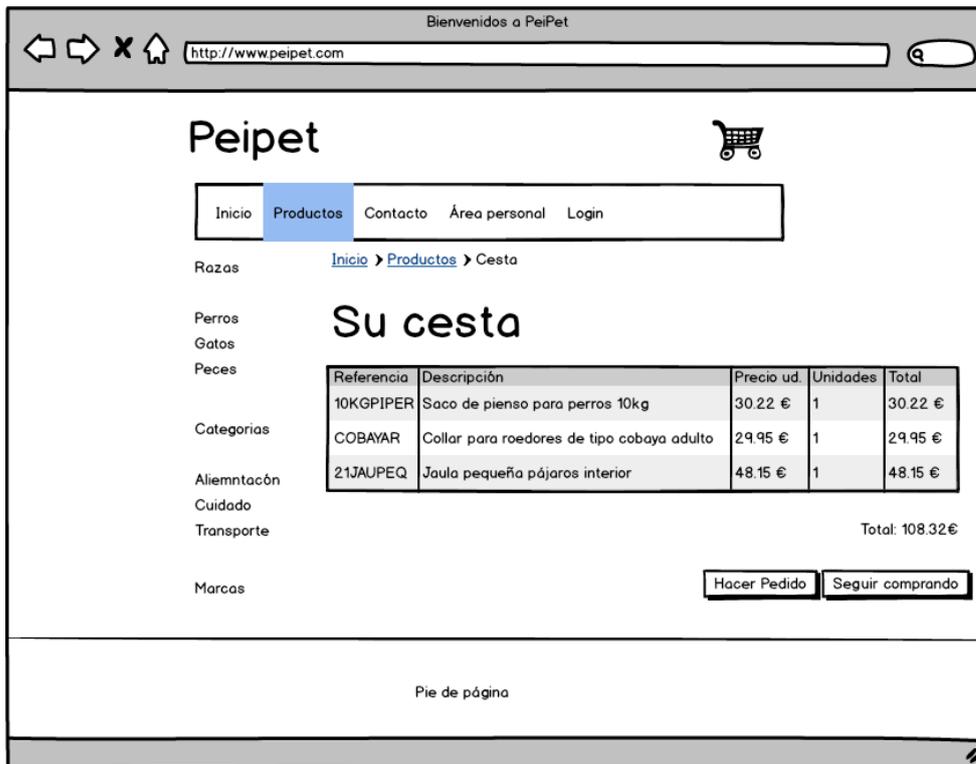


Fig.5 Wireframe de la cesta de la compra.

Login

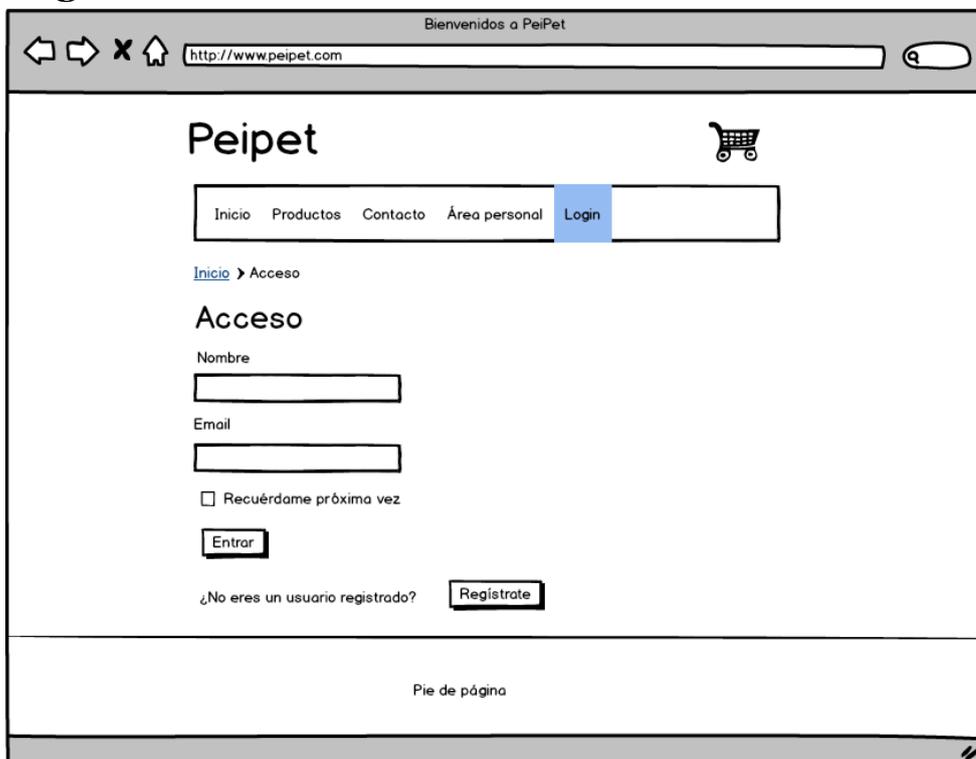


Fig.6 Wireframe pantalla de validación

Área personal



Fig.7 Wireframe pantalla área personal.

Contacto

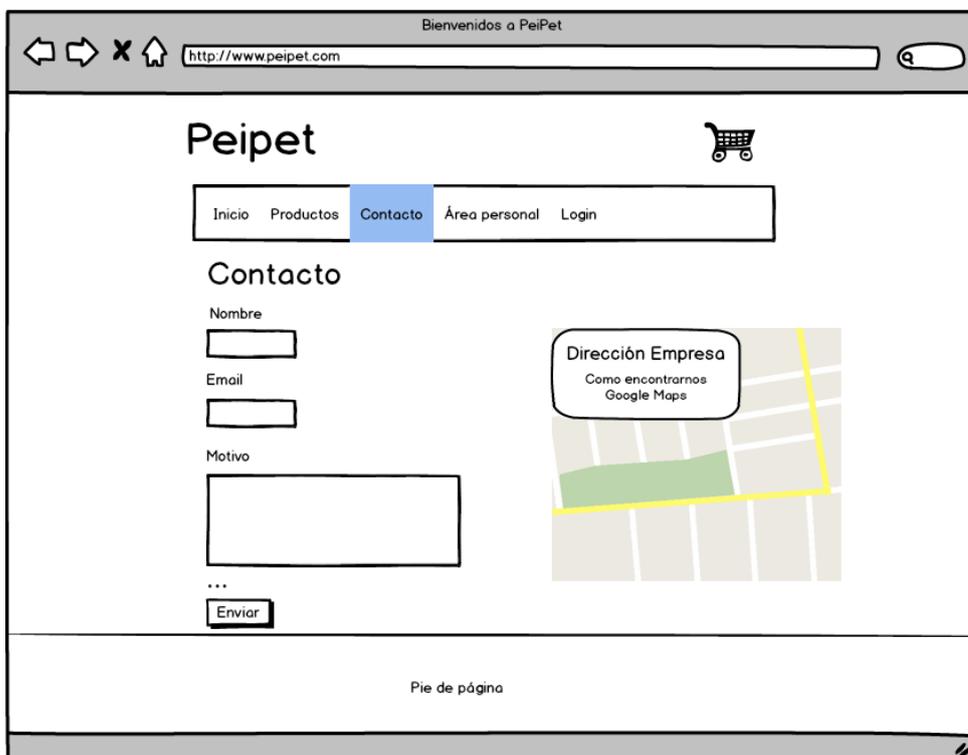


Fig.8 Wireframe pantalla de contacto.

4.2 Capa de persistencia

A la hora de diseñar las bases de datos, y tras un análisis de las tablas que podría necesitar, el esquema de la base de datos queda así:

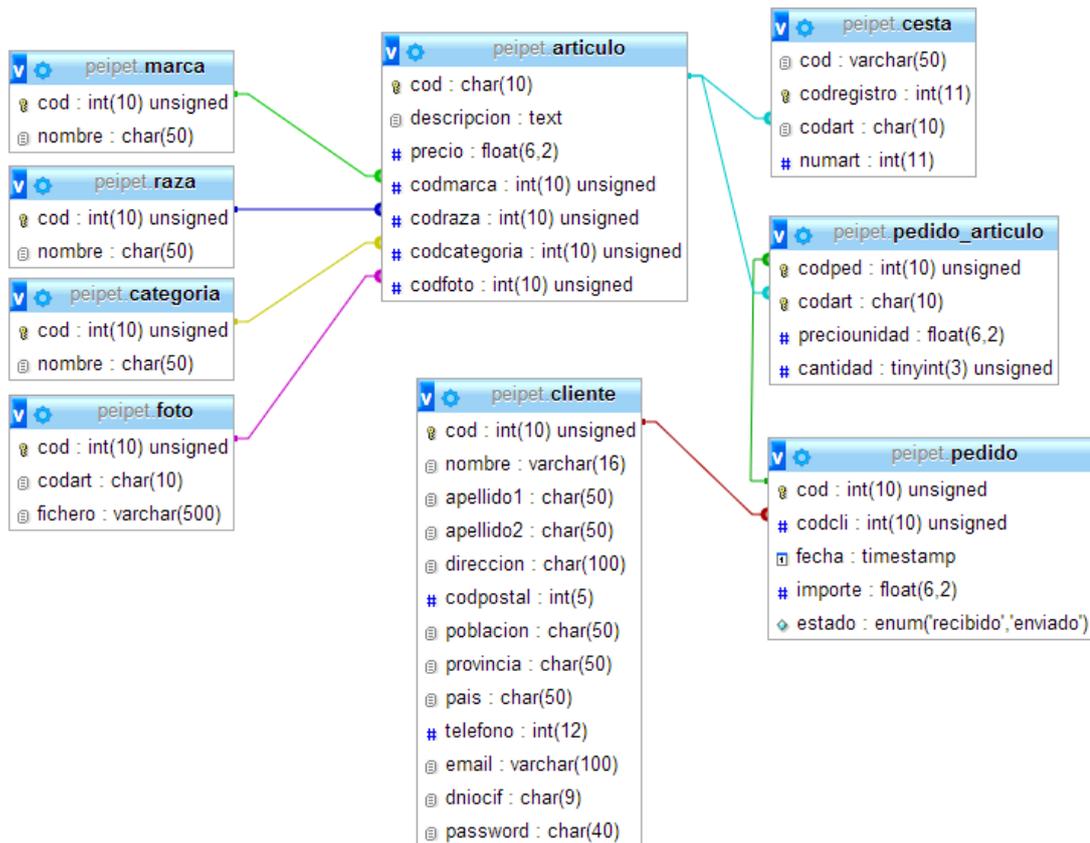


Fig.9 Esquema de la base de datos, extraído de *phpMyAdmin*

En este apartado tengo que aclarar unos aspectos, tras revisar de nuevo la memoria al acabar el desarrollo de la aplicación he comprobado que las tablas relacionadas con los pedidos no he llegado a utilizarlas, se implementaron en un principio para tener un registro de los pedidos que se hacían en la web, pero el trabajo se ha extendido bastante en el tiempo para llegar a esa parte, con lo que los pedidos en cuanto se cursan, solo queda un registro en el email del cliente con los detalles del pedido en formato PDF.

No he querido borrarlas para que quede constancia de lo que quería abarcar en el proyecto inicial, y como posible extensión a este. Las tablas afectadas en este caso son “pedido” y “pedido_articulo”, donde en “pedido” registraría los pedidos realizados por los clientes para tener constancia de ellos y para que en el área personal de estos también pudieran consultarlos a modo de historial.

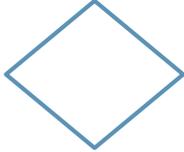
En cuanto a la tabla “cesta” tampoco la utilizo, ya que conseguí hacerla funcionar por medio de sesiones para cada usuario registrado, y al final del proyecto, me he dado

cuenta que permanecía ahí sin usar. Tampoco he querido tocarla por un posible uso en alguna extensión del proyecto como podría ser el guardado de cestas de usuario para que persista en almacenamiento aun cuando el usuario haya cerrado sesión.

En cuanto a la forma de trabajar con la persistencia, he de decir, que *Yii framework* ayuda mucho a la hora de trabajar con esta, ya que este genera un *Active Record (AR)*, que es una técnica que utiliza *Yii* para hacer un Mapeo Objeto-Relacional (ORM). Cada clase AR representa una tabla de la base de datos (o vista), cuyos atributos son representados como las propiedades de la clase AR, y una instancia AR representa una fila en esa tabla. Las operaciones CRUD (Create, Read, Update, Delete) también son implementadas como métodos de las clases AR, con esto conseguimos que el acceso a los datos de una manera Orientada a Objetos (OO), lo que simplifica mucho el trabajar con la base de datos ya que automatiza las operaciones CRUD. Para entendernos, y en resumidas palabras, *Yii* crea una base de datos orientada a objetos virtual, sobre la base de datos relacional, lo que simplifica mucho el trabajo con la persistencia.

4.3 Capa de lógica

En esta sección, vamos a ver mediante un diagrama de flujo las acciones que podemos realizar en la página. Voy a explicar un poco el diagrama mediante una tabla antes de presentarlo:

| | |
|---|---|
|  | Con este icono determinaré las páginas principales de la web, que son: Inicio, productos, contacto, área personal y validar usuario. |
|  | Con esta forma represento acciones que puede realizar el usuario web, algunas de estas pueden realizarse inmediatamente y otras vendrán encaminadas por una condición previa. |
|  | Este es el icono clásico de toma de decisión, aquí la aplicación aplicará la lógica para determinar qué acciones toma o dónde redirige al usuario |
|  | Terminador, he establecido que el usuario finaliza su navegación por la página una vez realiza el pedido, aunque este podría seguir navegando por ella si así lo deseara. |

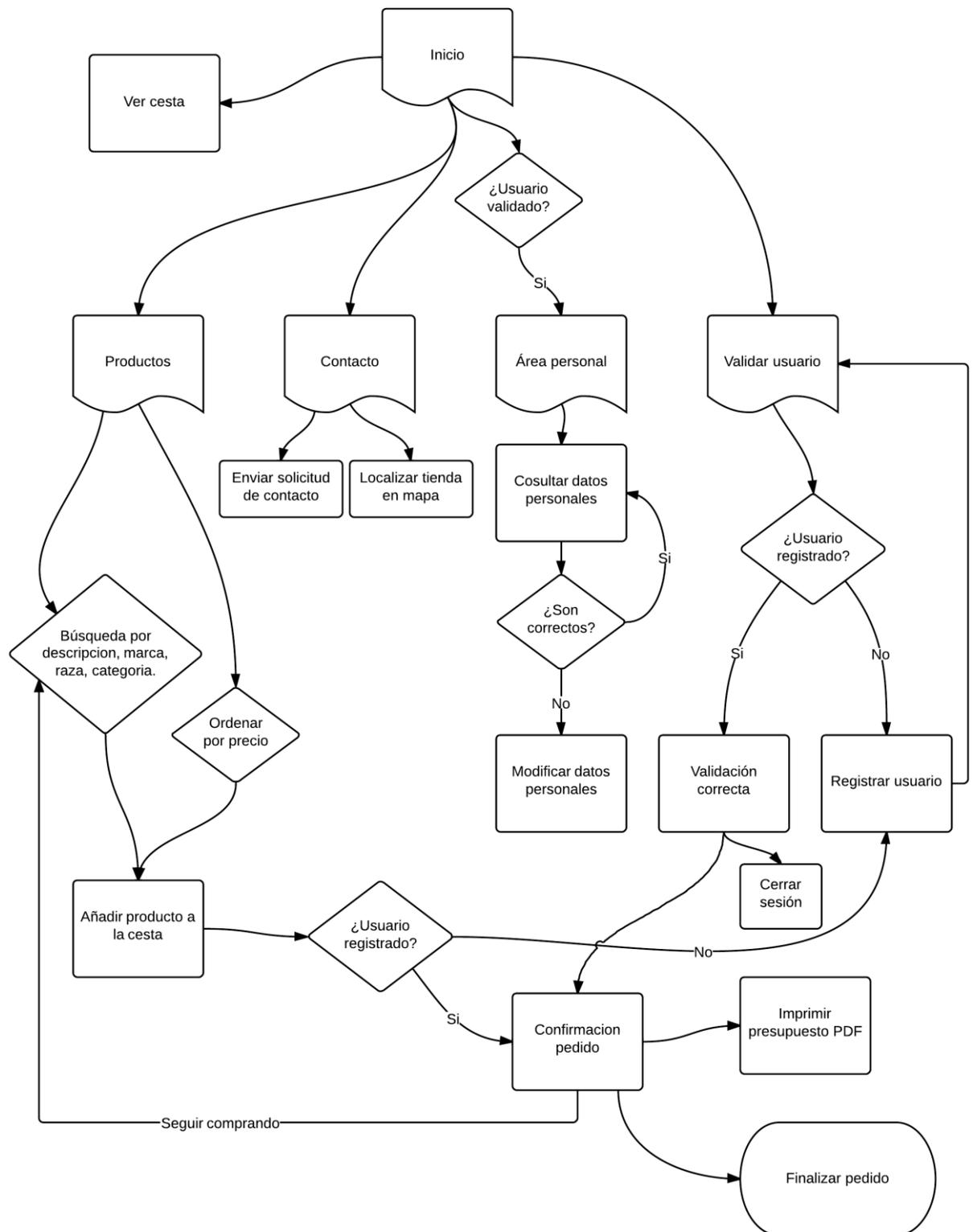


Fig.10 Mapa de la web. Diagrama de flujo.

4.4 ¿Por qué Yii Framework?

Para tomar tal decisión, estuve haciendo un estudio sobre la web para concretar cuál sería la opción más adecuada. Primero, intenté descubrir la tendencia de uso de *frameworks* PHP actuales. En internet puedes encontrar mucha información, lo importante es saber escoger lo importante o representativo. Según Google Trends que es una herramienta de Google para medir los términos más buscados en el famoso buscador, obtuve la siguiente gráfica sobre los datos del 2013.

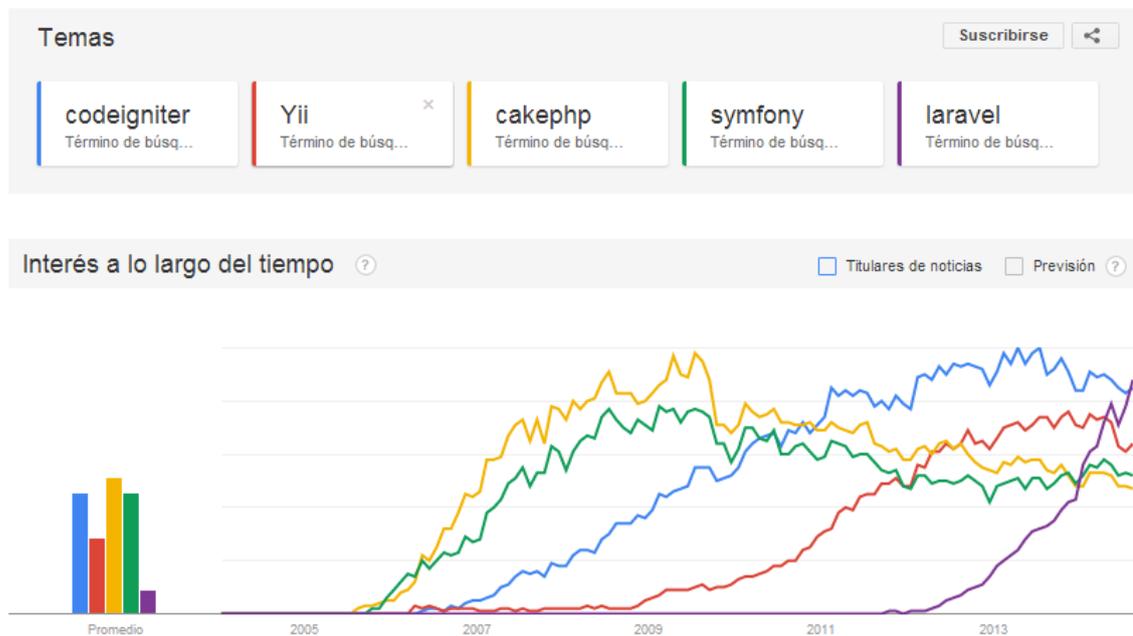


Fig. A Frameworks más buscados según Google Trends en 2013

Si bien esta gráfica revela que actualmente *CodeIgniter* y *Laravel* son los dos frameworks que actualmente despiertan más interés en la red de redes, explicaré porque deseché al final del estudio estas dos opciones, en primer lugar *CodeIgniter* aunque con mucha comunidad de usuarios que trabajan con él, lo descarté porque teniendo en cuenta presenta una curva de aprendizaje más pronunciada que los demás (en poco tiempo se aprende mucho), este se utiliza sobre todo en pequeños proyectos que no son demasiado complejos, quedando limitado para proyectos que toman cierta complejidad.

Laravel está siendo el último “Boom” en cuanto a *frameworks PHP*, los profesionales hablan muy bien de él, tiene una curva de aprendizaje también muy pronunciada, es muy potente a diferencia de *CodeIgniter* que no lo es tanto, y usa herramientas avanzadas como *Composer*, (el administrador de dependencias de PHP).

¿Qué es lo que hizo que no me decidiera por este último? - Que al ser relativamente nuevo, la comunidad de usuarios no es tan grande como pueden ser la de otros *frameworks*, tampoco hay demasiados proyectos realizados, lo que implica que dispondremos de un número menor de extensiones de terceros para utilizar y además encontraremos mucha menos documentación sobre este.

A todo esto hay que añadir, que bajo mi poca experiencia, en el mundo de informática, ser un “*Early adopter*” (gente que adopta las nuevas tecnologías recién salidas al mercado) a veces es un poco arriesgado, estas pueden dejar de utilizarse y pasar de moda tan rápido como aparecen, por eso prefiero escoger algo que lleva ya una cierta trayectoria de uso, y tiene una previsión aparentemente buena para el futuro.

Por curiosidad, y por ver cómo está el mercado laboral en cuanto a desarrollo web utilizando *frameworks* PHP, escogí uno de los portales más importantes a nivel nacional de empleo, Infojobs www.infojobs.net, para hacer una comparación informal de la distribución de empleos según los distintos *frameworks*. La lista la elaboré a partir de datos a nivel nacional y fueron extraídos el mismo día, a principios de agosto de 2014, la distribución en número de empleos queda así:

| Empleos según <i>framework</i> PHP Infojobs.net | |
|--|---------------------|
| <i>Framework</i> PHP | Núm. empleos |
| Symfony | 47 |
| CodeIgniter | 24 |
| Yii | 14 |
| Laravel | 12 |
| CakePHP | 11 |

Fig. B Ofertas de empleo a nivel nacional según Infojobs.com a día 11/08/14

He abandonado también la idea de desarrollar en Symfony, ya que últimamente aunque en las ofertas de trabajo no lo demuestran, está cayendo en desuso dado que es posiblemente uno de los *frameworks* más complicados de utilizar según lo que he podido comprobar en la red, algo totalmente desaconsejado como primer *framework* a utilizar y para introducirme en esta forma de desarrollo.

Una vez visto esto, y tras invertir muchas horas en internet, el análisis de los *frameworks* PHP que a día de hoy están disponibles, he considerado que era el más adecuado para mi proyecto por su gran documentación y comunidad de usuarios es *Yii Framework*, lo que facilita que estén disponibles bastantes módulos o extensiones que facilitan el desarrollo, todo esto viene acompañado por una curva de aprendizaje bastante razonable, que me permite aprender el uso de la herramienta para cumplir mis objetivos en un tiempo adecuado. Como primer *framework* que utilizo, considero que el compromiso entre todos los aspectos estudiados hacen de él, el más apropiado para mi proyecto.

5. Detalles de implementación

A continuación veremos un poco como he desarrollado el proyecto a nivel de código y de estructura de ficheros, como también las dependencias, para ver cómo está estructurada internamente la aplicación. En el siguiente apartado veremos que tecnologías y que software he empleado en mi proyecto.

5.1 Estructura de ficheros y dependencias

Básicamente partimos del directorio por defecto donde Xampp instala todos los archivos necesarios para que puedan ser ejecutados los servicios que ofrece, en mi caso, está instalado en la ruta por defecto (C:/Xampp) y dentro de esta carpeta encontramos todos los archivos de configuración, las bases de datos y los archivos .php junto con otros archivos que son necesarios para la visualización de la web.

Las carpetas que nos interesan son “mysql” y “htdocs” dentro del directorio Xampp en la raíz del disco duro. La base de datos está contenida dentro de la carpeta “mysql”, pero no trabajamos directamente sobre ella, es mediante la propia aplicación web la que la gestiona, donde esta podemos crear, modificar o borrar registros de las tablas que atañen a la base de datos de la aplicación. También podemos modificar la base de datos mediante la interfaz web que nos proporciona *MySQL* llamada *phpMyAdmin*, donde podremos hacer tareas de mantenimiento o reconfiguración de la base de datos. En todo caso no deberemos tocar esta carpeta ni sus archivos directamente y deberemos respaldarla con los servicios *Apache* y *MySQL* parados para que se pueda copiar correctamente.

En la carpeta “htdocs” es donde residen los archivos que serán publicados vía web y conformaran la web propiamente dicha, en terminología de servidores web este sería nuestro *webroot* o raíz del sitio. Dentro de esta carpeta se encuentra la carpeta “peipet”, que es la raíz de todos los archivos de nuestro proyecto. *Yii Framework* crea una estructura de archivos bajo un patrón MVC (Modelo, Vista, Controlador) una vez generamos un nuevo proyecto vamos a tratar de explicar el modelo a continuación.

Modelo Vista Controlador (MVC)

El Modelo Vista Controlador o MVC es un patrón de diseño que separa los tres componentes de una aplicación para estructurar la implementación y facilitar su posterior mantenimiento. Estos tres componentes son:

- El **modelo** representa la lógica de la aplicación, es donde se definen los datos y en las reglas del negocio, en *Yii* estos modelos quedan representados por un archivo `.php` para cada objeto que interviene en la lógica y están contenidos en la carpeta “models”.
- La **vista** hace referencia a la interfaz de usuario, a la presentación de la información, en *Yii* las vistas quedan agrupadas para cada modelo en una carpeta con su nombre dentro de la carpeta “views”.
- El **controlador** actúa como mediador entre la solicitud del usuario y los modelos, y, las vistas involucradas en la ejecución. En la carpeta “controllers” tenemos un controlador para cada objeto y otro controlador del sitio, que genera *Yii* automáticamente, como es “SiteController.php”

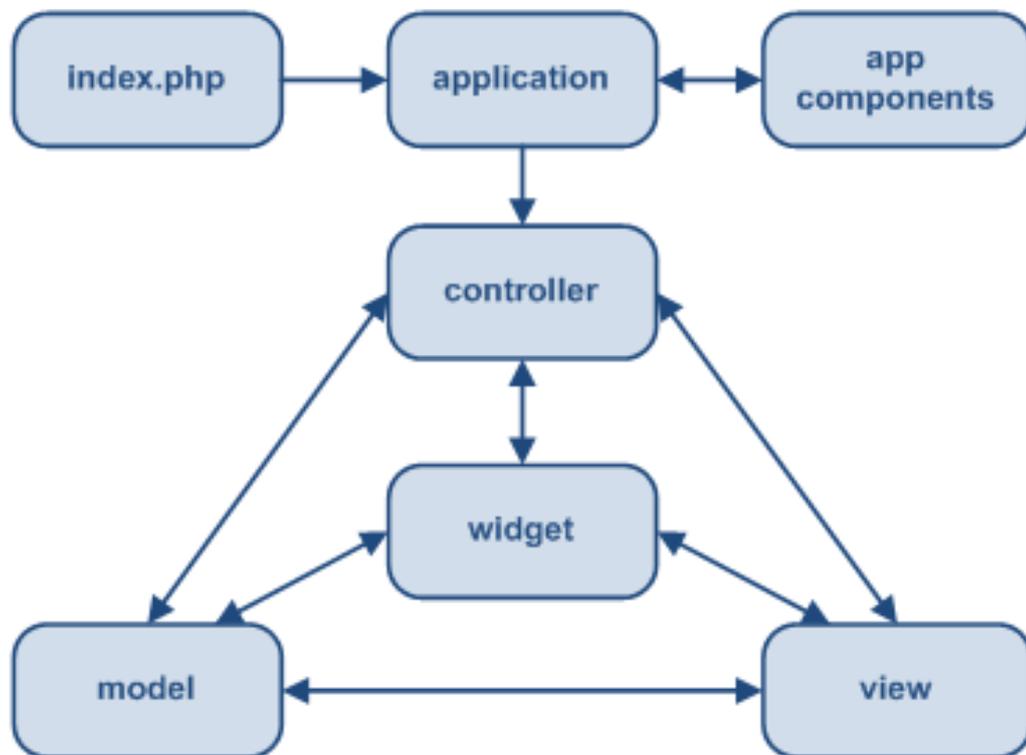


Fig.11 Esquema del Modelo Vista Controlador (MVC)

Estas tres capas, junto a otros archivos de configuración del *framework* quedan almacenados dentro de la carpeta “protected”, que como su nombre indica, está protegida del acceso mediante el uso de navegadores web por medio de un archivo “.htaccess”, que se usa en el servidor Apache para limitar el acceso a la carpeta que contenga este archivo y todas las que queden en la jerarquía por debajo de esta. Hasta ahora, hemos comentado el MVC que se encuentra dentro de “protected”, vamos a ver otros archivos y carpetas relevantes que utiliza Yii para su funcionamiento. En la siguiente captura tomada del NetBeans, vemos la separación de las tres capas, *controllers* (controladores), *models* (modelos) y *views* (vistas).

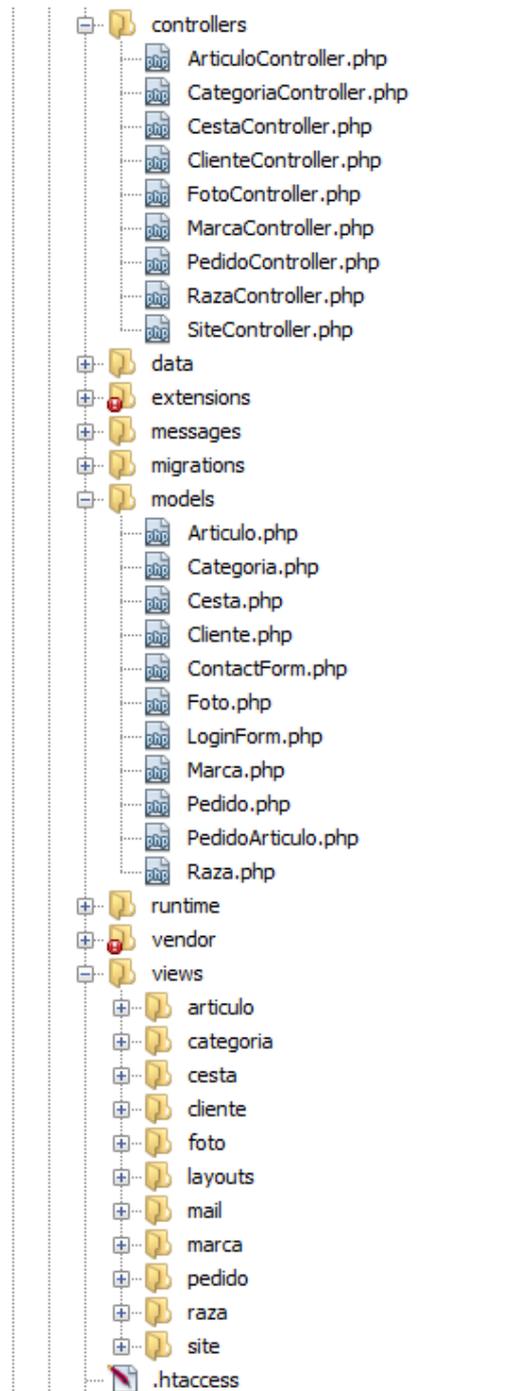


Fig.12 Captura de un extracto de la jerarquía de archivos que utiliza Yii obtenida del IDE Netbeans

En cuanto al **controlador frontal** (*Front Controller pattern*), este está representado por el archivo “index.php”, contenido en la raíz del proyecto y es el único punto de acceso del usuario a la aplicación web mediante la creación de una única instancia de la aplicación. Su función es la de recibir requerimientos del usuario y remitirlos a los controladores apropiados para su posterior procesamiento. Omitiré más detalles del funcionamiento interno de *Yii* ya que no es el objetivo de esta memoria.

En mi proyecto he utilizado tres extensiones y un tema visual, las extensiones se configuran en el archivo de configuración principal “main.php” dentro de “../protected/config” y los archivos necesarios para sus funcionamiento quedan dentro de “../protected/extensions/”, teniendo una carpeta por cada extensión instalada con su nombre. Los temas se configuran en “main.php” también, y se almacenan en “../themes”, en la raíz del proyecto, con lo que todo queda bastante estructurado ordenado. Hablaré de las extensiones y del tema en el siguiente punto.

5.2 Tecnologías utilizadas en el desarrollo

En este apartado empezaremos por mencionar las tecnologías utilizadas para luego pasar a ver cada una de las herramientas software utilizadas en el desarrollo, con unos breves comentarios sobre su versión y la funcionalidad proporcionada.

Básicamente, la aplicación está asentada sobre dos tecnologías principales que determinan la naturaleza del proyecto, éstas son PHP en su versión 5.5.11, y MySQL en su versión 5.6.16, ambas tecnologías integradas en el paquete Xampp en su versión 1.8.3. En cuanto a PHP se ha agilizado la programación usando un *Framework* PHP en este caso *Yii Framework*, en su versión 1.1.15, no he querido usar la 2.0 por no estar en fase estable y por lo tanto esta menos probada y cuenta con una documentación menos extensa.

Estas son los pilares de la aplicación, pero me he ayudado de otras cuantas aplicaciones para el desarrollo como son editores de texto, *IDEs*, gestores web de bases de datos, herramientas de diseño gráfico, etc. Vamos a verlas:

- Como editor principal he usado el IDE **Netbeans** en su versión 8.0 (Build 201403101706) para Windows de 64 bits. He de decir que ofrece mucha ayuda al programador en PHP ya que tiene funciones de autocompletado, incluso autocompleta y valida código de *Yii Framework*, ya que podemos configurarlo para ello.
- También he utilizado como editor el **Sublime Text** Build 3059 para Windows 64 bits. Para proyectos pequeños es una maravilla, presenta el código con colores sobre fondo negro a mi entender muy ergonómica y fácil de usar. A la hora de tratar con proyectos más grandes con *frameworks* y



Portal web de una tienda virtual de productos para animales domésticos

estructuras de datos más complejas ya se queda un poco corto y hay que pasarse a un IDE más completo.

- Para ayudarme un poco con las hojas de estilo y para el diseño me he ayudado de **Dreamweaver** CC 13.2 Build 6466 para Windows 32 bits, creo que para esta parte está bastante bien, pero no para el desarrollo de una aplicación con un *framework* PHP.
- Con los diagramas de casos de uso he utilizado **Rational Rose** de IBM en su versión 7.0 para Windows 32 bits.
- Para los diagramas de clases utilizo **Bouml** en su versión 4.23 patch 7 para Windows 32 bits.
- Para la fase de diseño de la web he empleado una aplicación para desarrollar los bocetos de forma muy visual e intuitiva, se llama **Balsamiq Mockups** y la he utilizado en su versión 2.2.21 para Windows 32 bits.
- Como herramienta para diseñar el logo corporativo utilicé **Sothink Logo Maker** Professional 4.0 Build 4081 para Windows 32 bits.
- Para el maquetado de las imágenes de la web he empleado **GIMP** en su versión 2.8.10 para Windows 64 bits.
- Para los diagramas de flujo de trabajo por la web me he ayudado de la herramienta online de www.lucidchart.com.
- Con las bases de datos, tras probar **MySQLWorkbench**, la abandoné en favor de la administración vía web con **phpMyAdmin** 4.1.12 porque para la envergadura de este proyecto es más que suficiente.
- El tema empleado para la web se llama **Hebo!**, es un tema HTML5 compatible con Yii, y lo podemos encontrar en <http://www.webapplicationthemes.com/hebo-responsive-html5-theme/>.
- He utilizado una extensión que me aporta la funcionalidad de cesta de la compra, y que me aporta una interfaz de código para trabajar con ella que se llama **Yii Shopping Cart**, la cual la podemos encontrar en <https://github.com/yiiext/shopping-cart-component>.
- También he utilizado otra extensión para generar PDFs llamada **Yii-PDF** que la podemos encontrar en <http://www.yiiframework.com/extension/pdf/>.

- Y la última extensión utilizada es la encargada de enviar emails, llamada SwiftMailer y la podemos encontrar en la siguiente dirección <http://www.yiiframework.com/extension/mail/>.

Por último mencionar que la plataforma de trabajo ha sido un PC con Windows 8.1 Professional 64 bits y agradecer que las empresas con software de pago faciliten versiones con licencia para estudiante o versiones *Trial* suficientes para el desarrollo de este proyecto.



6. Pruebas

Veremos ahora cómo se comporta la aplicación ya desarrollada bajo distintas pruebas para depurar su desarrollo si es oportuno. Empezaremos con unas pruebas de conformidad para ver si la aplicación se ajusta a los estándares del w3c, y pasaremos a ver cómo se comporta bajo los 3 principales navegadores. Por curiosidad, lo probaremos bajo un navegador de un *smartphone*, y sobre un *tablet*, aunque, no he desarrollado una versión específica para navegadores *mobile*.

6.1 Conformidad estándar W3C

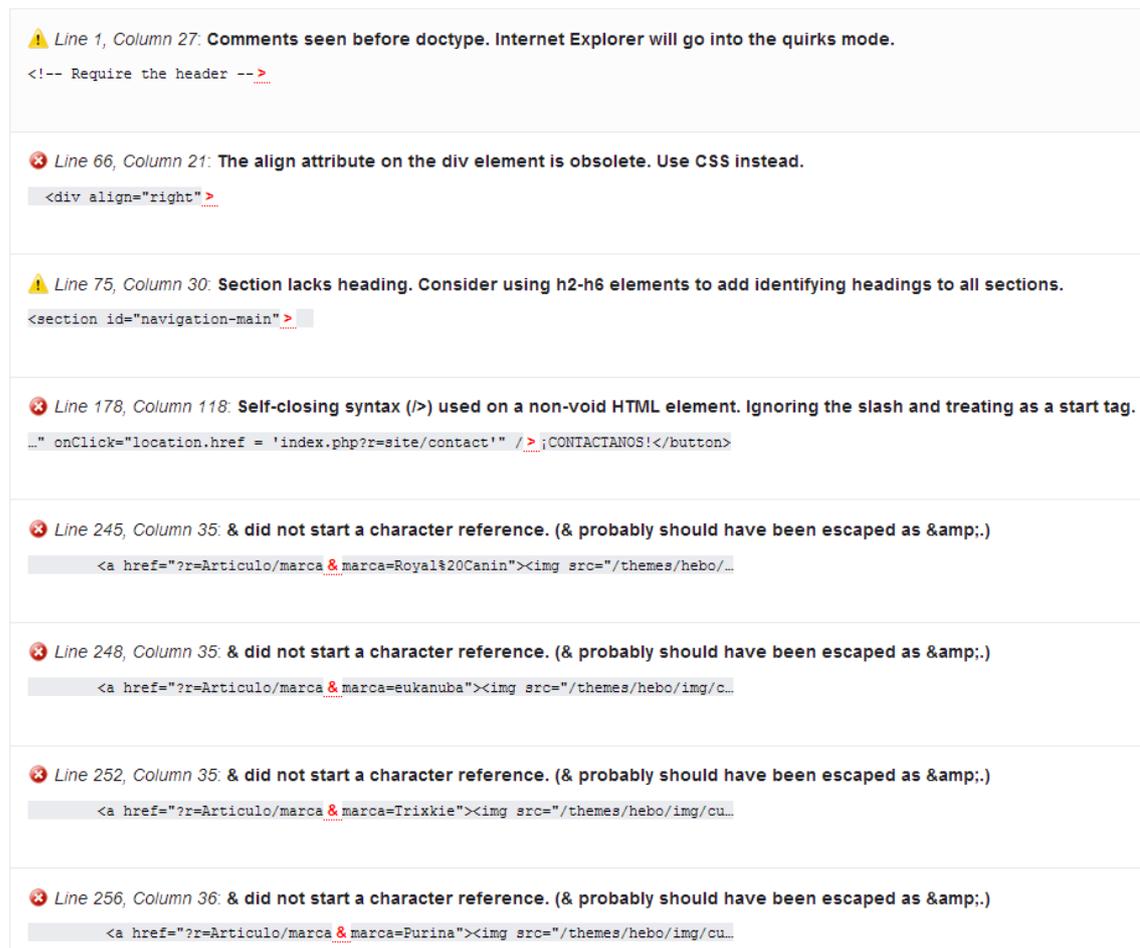
Empezamos con las pruebas del W3C, estas son unas pruebas que se hacen desde el portal del propio consorcio en <http://validator.w3.org/>, que, tras introducir la URL de tu página hace un chequeo de si el lenguaje de marcado que has utilizado para tu web se ciñe totalmente al estándar del W3C. Al validar mi web por primera vez en su portal me reportó lo siguiente:

| Errors found while checking this document as HTML5! | |
|---|--|
| Result: | 6 Errors, 3 warning(s) |
| Address : | <input type="text" value="http://peipet.dyndns-web.com/"/> |
| Encoding : | utf-8 <input type="text" value="(detect automatically)"/> |
| Doctype : | HTML5 <input type="text" value="(detect automatically)"/> |
| Root Element: | html |

Fig.13 Errores encontrados por el validador del W3c.

Y este es el desglose de los errores encontrados:

Validation Output: 6 Errors



The screenshot displays the output of a W3C Validator, listing six errors. Each error entry includes a warning icon (yellow triangle with exclamation mark) or an error icon (red circle with X), a line and column reference, a description of the error, and a code snippet showing the problematic HTML. The errors are:

- Line 1, Column 27: Comments seen before doctype. Internet Explorer will go into the quirks mode.** Code: `<!-- Require the header -->`
- Line 66, Column 21: The align attribute on the div element is obsolete. Use CSS instead.** Code: `<div align="right">`
- Line 75, Column 30: Section lacks heading. Consider using h2-h6 elements to add identifying headings to all sections.** Code: `<section id="navigation-main">`
- Line 178, Column 118: Self-closing syntax (>) used on a non-void HTML element. Ignoring the slash and treating as a start tag.** Code: `... onClick="location.href = 'index.php?r=site/contact'" />;CONTACTANOS!</button>`
- Line 245, Column 35: & did not start a character reference. (& probably should have been escaped as &.)** Code: `Total: | | | | 106.61 € |

Su dirección:
Nombre: Juan
Apellido1: Alonso
Apellido2: González
Dirección: C/Republica Argentina 56, pta 8
Código Postal: 89474
Población: Cáceres
Provincia: Cáceres
País: España
Teléfono: 678521458

Buttons at the bottom: Realizar pedido, Seguir comprando, Imprimir PDF.

Fig. 36 Detalle de la confirmación del pedido.

Una vez pulsado el botón “Hacer pedido” se nos redirigirá a esta página donde podremos comprobar si los datos personales son correctos, para proceder a realizar el pedido definitivamente. Si al comprobarlos estos no son correctos, podremos pinchar en “Modificar los datos personales” que nos llevará a la pantalla de registro pero con todos los datos actuales rellenos. En esta pantalla, modificaremos los datos convenientes y pincharemos en el botón “Actualizar”, con lo que las actualizaciones sobre estos quedan realizadas, y podremos continuar con el pedido con los datos ya corregidos.

Actualizar los datos personales antes de realizar pedido

The screenshot shows a web browser window with the URL `peipet.dyndns-web.com/index.php?r=cliente/update&id=5`. The page title is "Actualizar datos de jualgo@yahoo.com". The form contains the following fields:

- Nombre: Juan
- Apellido1: Alonso
- Apellido2: González
- Dirección: C/Republica Argentina 55, pta 8
- Código postal: 89474
- Población: Caceres
- Provincia: Caceres
- País: España
- Teléfono: 678521458
- Email: jualgo@yahoo.com
- Dni/coif: 28758321X
- Contraseña: [oculto]

There are "Actualizar" and "Volver" buttons at the bottom of the form. The left sidebar contains navigation links for "Perros", "Gatos", "Pajaros", "Peces", "Roedores", and "Reptiles", as well as "Categorías" (Alimentación, Cuidado y salud, Casetas, Transporte, Abrigos, Collares, Peces, Jaulas) and "Nuestras marcas" (Royal Canin, Eukanuba, Trookie, Purina, Flural, Hills, Nayeco, Living World).

Fig. 37 Pantalla para actualizar los datos del usuario.

También tenemos la posibilidad de generar un PDF con los detalles a modo de presupuesto, con los datos de salida también correctamente formateados e incluyendo el logo de la empresa, como muestro en la captura de pantalla a continuación.

Imprimir PDF



Fig. 38 PDF generado por la aplicación con el botón “Imprimir en PDF” antes de la confirmación final del pedido.

Una vez que tengamos claro que queremos realizar el pedido pulsando en “Realizar pedido”, este quedará generado y recibiremos en nuestro email (el que hayamos introducido en el registro del usuario) un email confirmándolo con los detalles en un adjunto en PDF.

Pantalla de pedido realizado correctamente

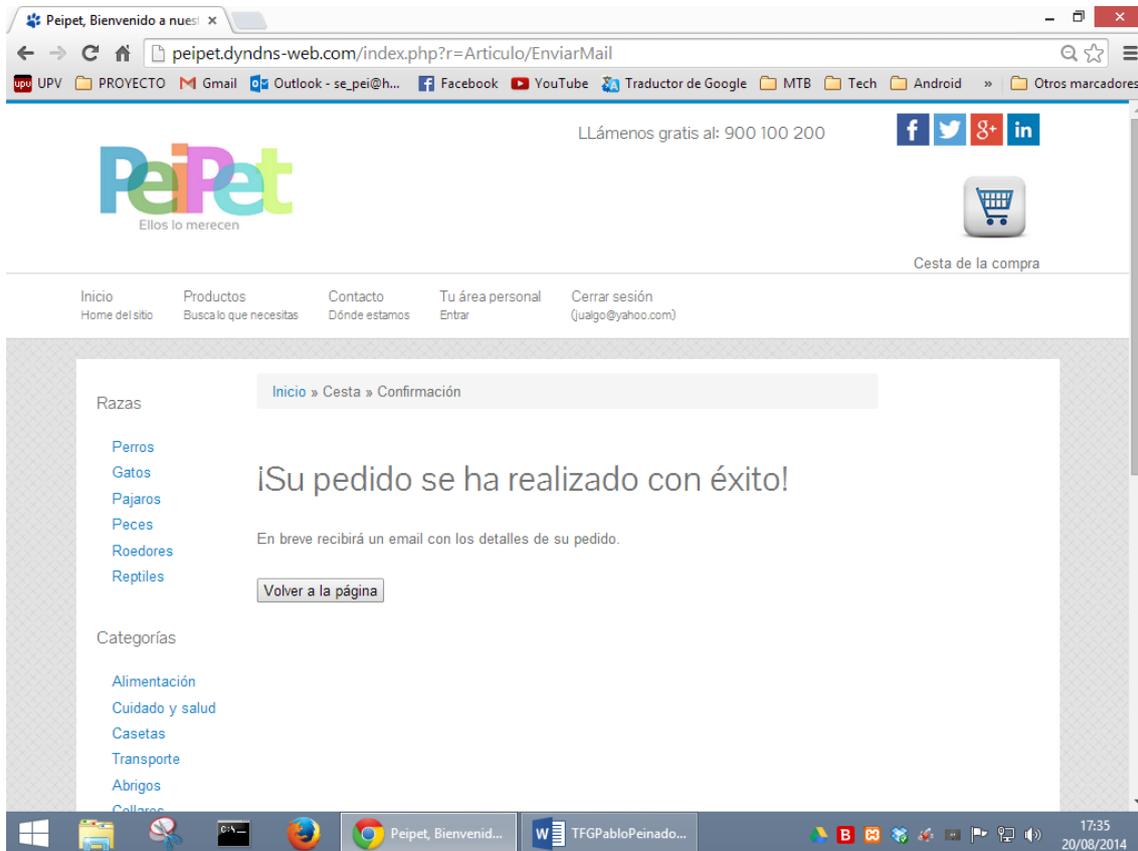


Fig. 39 Mensaje de confirmación de pedido realizado correctamente.

Seguidamente, podremos ir a nuestro mail para ver el mail que nos ha llegado con el pedido realizado, una vez más comprobamos que este llega correctamente formateado y con el adjunto de los detalles también en PDF, como vemos en la pantalla a continuación.

Email recibido con los detalles del pedido

The screenshot shows a Gmail interface with a search bar at the top. The email is from Peipet <tiendapeipet@gmail.com> received on August 6th. The subject is 'Su pedido en Peipet'. The main content is a confirmation of the order with the following details:

Confirmación del pedido

Código De Cliente: 6
Nombre De Usuario: tiendapeipet@gmail.com
Dni: 29200166x

Productos seleccionados:

| Referencia: | Descripción: | Precio ud. | Unidades: | Total: |
|---------------|-------------------------------|------------|-----------|---------------|
| PECRS50LTS | Pecera 50L peces tropicales | 431.00 € | 4 | 1724 € |
| ABRIGAT | Abrigo de invierno para gatos | 23.00 € | 1 | 23 € |
| Total: | | | | 1747 € |

Su dirección:

Nombre: Pablo
Apellido1: Peinado
Apellido2: Abad
Dirección: C/ Luis Milan 3, pta 2
Código Postal: 46021
Población: Valencia
Provincia: Valencia
País: España
Teléfono: 633199320

At the bottom, there is a chat bubble icon with the text 'No hay chats recientes' and a link 'Iniciar uno nuevo'. Below that is a PDF attachment icon labeled 'pedido.pdf'.

Fig. 40 Detalle del email recibido por el cliente al realizar el pedido con los detalles también en un adjunto en PDF.

Nota: En el momento de escribir esta memoria tengo problemas para que el email que llega tenga como remitente la dirección introducida por el usuario.

Pasamos ahora a ver cómo funciona el login y el registro de usuarios. En cualquier momento, el usuario puede validarse desde el menú principal pinchando sobre la opción del menú de “Acceso”, donde se le mostrará la siguiente pantalla de login:

Acceso para clientes registrados

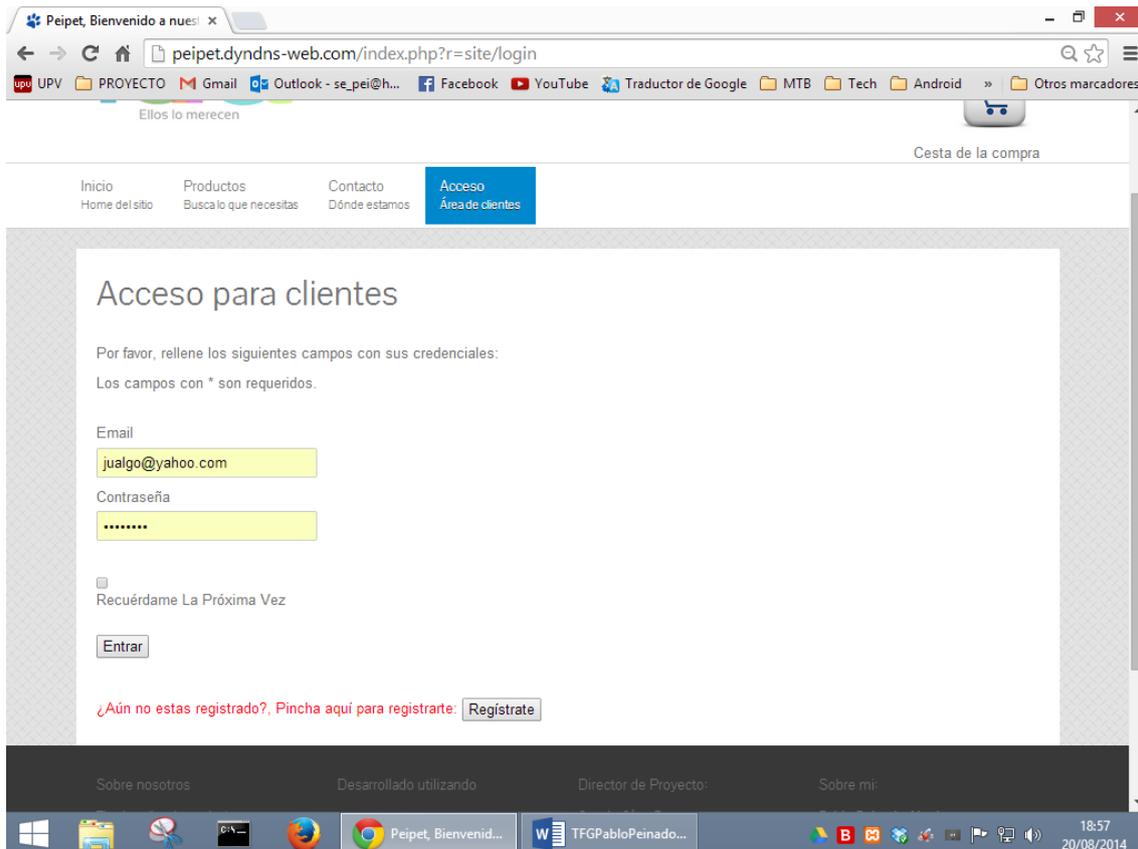


Fig. 41 Pantalla de acceso para los usuarios.

La aplicación validará si los campos están rellenos y en ese caso si los datos proporcionados son correctos, si el usuario se valida entra en sesión, y el botón del menú principal cambia de mensaje avisando de que el usuario está correctamente validado, y, además, ahora aparece un nuevo botón en el menú que le permite acceder a su área personal, como podemos ver a continuación.

Los menús cambian si el cliente se valida

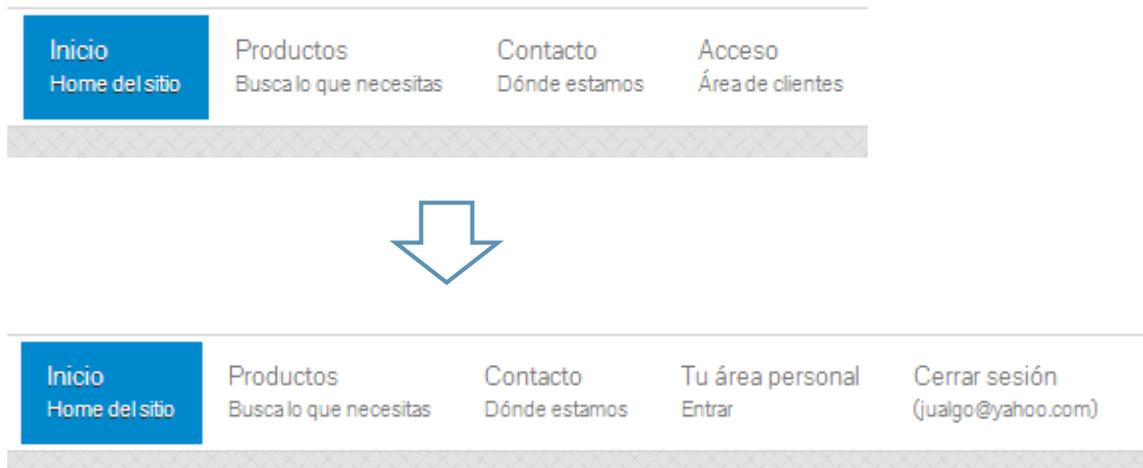


Fig. 42 Detalle de cómo cambia el menú para el usuario validado correctamente.

En su área personal podremos ver sus datos personales, así como modificarlos, lo que nos llevaría de nuevo a la pantalla de actualizar datos personales vista previamente en la Fig. 37.

Para registrar un nuevo usuario, la aplicación pregunta en la pantalla de login si no estamos registrados, y tenemos un botón para registrarnos, donde se accederá a la siguiente pantalla donde podremos dar de alta un nuevo usuario. La aplicación comprueba si todos los campos requeridos están completos, y si no lo están, no nos deja continuar lanzando el mensaje de error oportuno. También, hay que decir que las contraseñas se cifran mediante un algoritmo MD5 antes de ser almacenadas en la base de datos, con lo que si alguien accediera a esta, no tendría acceso a las contraseñas. Esto es lo que se almacena en la base de datos:

Contraseñas cifradas en la base de datos

| email | dniocif | password |
|------------------------|-----------|---------------|
| jualgo@yahoo.com | 28756321X | saCa1UuiWIGWQ |
| tiendapeipet@gmail.com | 29200166x | saCa1UuiWIGWQ |
| admin@peipet.com | 29200166x | saCa1UuiWIGWQ |
| pegasan@yahoo.com | 13413432Z | sar/1z2WGfFP. |
| pealsa@hotmail.com | 25466879Z | saEL4rUJqITJU |

Fig. 43 Detalle de las contraseñas cifradas en la base de datos.

Como vemos las contraseñas están cifradas y los caracteres que vemos no se corresponden con los que verdaderamente hemos introducido.

Por último, echaremos un vistazo a la pantalla de contacto, desde esta pantalla cualquier usuario registrado o no puede solicitar información a la tienda sobre cualquier cuestión. El usuario rellena los campos solicitados, y se le avisará al validarlos si se ha dejado alguno en blanco para que lo complete, también avisará si el formato de email introducido no es un formato válido. Para asegurarnos de que el mensaje no está generado por software malicioso el usuario debe introducir un *captcha*, este es nuevo cada vez que se carga la página, y se puede solicitar que se genere uno nuevo por parte del usuario pinchando sobre “Obtenga un nuevo código”, si el actual resulta poco legible. Por último, al enviar, se envía un email con los todos los datos del usuario al representante de la tienda, en este caso lo recibe la cuenta que he creado al efecto para hacer pruebas tiendapeipet@gmail.com.

Por último, en esta pantalla se muestra el *widget* de *Google Maps*, indicándonos la localización de la tienda, en este caso como la tienda es ficticia apunta a la Universidad Politécnica de Valencia. Este *widget*, nos permite trabajar como lo hacemos cargando directamente *Google Maps*, incluso redirigiéndonos a este con la dirección ya fijada.

Página de contacto

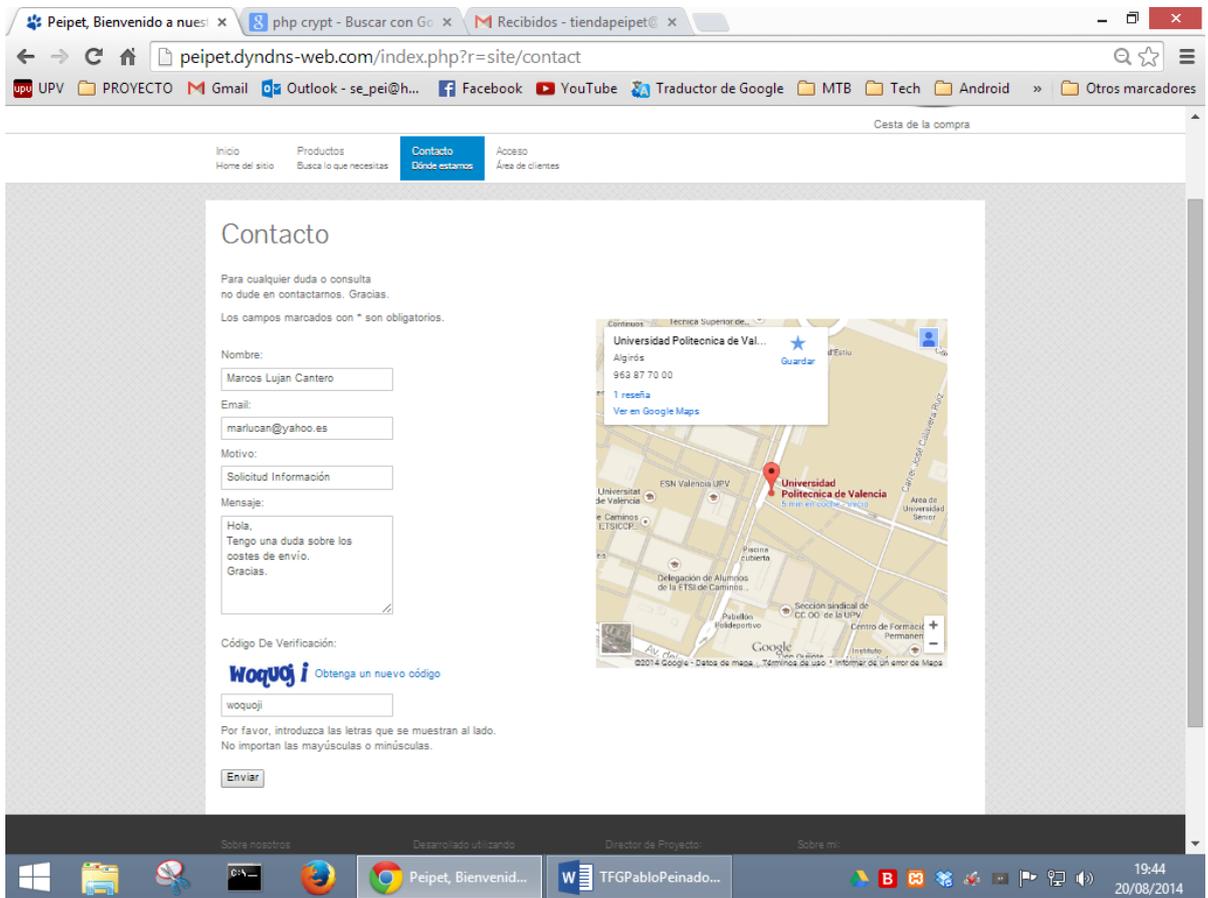


Fig. 44 Detalle de la pantalla de contacto.

7. Conclusiones

Acabo esta memoria exponiendo las conclusiones del trabajo, las he dividido en técnicas y personales, ya que considero que merecen esta diferenciación. Seguidamente he añadido las posibles ampliaciones del proyecto.

7.1 Técnicas

En este trabajo he aprendido a manejarme con un framework para PHP como es *Yii Framework*, antes de introducirme en proyecto, apenas sabía lo que era un framework de PHP, es más, las primeras versiones de la web las desarrollé sin ayuda de uno, lo cual, hizo que el desarrollo fuera mucho más lento y con mucho más proceso de *debugging*, ya que todo el código era escrito desde cero. Esto me ha ayudado a comprender lo necesarias que son este tipo de herramientas para proyectos modernos, que requieren unos ciertos estándares de calidad, seguridad y sobre todo tienen que ser desarrollados en un tiempo limitado, con lo que hoy en día se hace obligatorio el uso de un *framework*.

Me ha dejado también un buen sabor de boca aprender a trabajar con las otras herramientas que he utilizado para el proyecto, *MySQL* es bastante fácil de aprender y a la vez potente si profundizas en detalle, pero para el nivel del trabajo y con la ayuda de la interfaz de gestión web *phpMyAdmin* el trabajo sobre la base de datos bastante accesible. Es de agradecer, que paquetes como *Xampp* integren todas estas herramientas que son instaladas a la vez, y además se puedan configurar de forma relativamente sencilla.

7.2 Personales

Creo que este trabajo me ha dado una base para poder afrontar desarrollos web más complejos, como ya mencione en la sección de objetivos al principio de la memoria, creo que en los estudios de Grado en Ingeniería Informática no se le da tanto peso a desarrollos web de este tipo (incluso en la rama de Tecnologías de la Información, como la que yo he cursado), que luego tienen una parte importante de ofertas de trabajo en el mercado laboral moderno.

Pasar de programar el proyecto con PHP a un *framework* me ha demostrado toda la potencia y la simplicidad que te puede aportar una herramienta de este tipo, aunque también he de decir que su uso no es trivial, requiere de un esfuerzo para aprenderlo, pero una vez que dominas la interfaz hace las cosas mucho más fáciles.

También he visto como un proyecto web se puede hacer eterno, sobre todo para gente sin experiencia como yo, cuando crees que te vas acercando al final siempre salen cosas nuevas que implementar y detalles por corregir, lo que te hace comprender que para hacer un proyecto profesional que aparentemente parece sencillo por detrás bastante trabajo y detalles a tener en cuenta, más si no tienes unos requisitos previos establecidos.

7.3 Posibles ampliaciones del proyecto

Voy a enumerar una lista de posibles ampliaciones del proyecto, algunas fueron planeadas para ser implementadas en este trabajo, pero me ha faltado tiempo y no las he podido incluir, y otras, en cambio, son unas propuestas de ampliación que me he formulado.

Empecemos con las que en principio quería incluir en el trabajo:

- Revisar concordancia de contraseñas (introducir la contraseña dos veces) y revisar el formato de email en el registro de usuario.
- También en el formulario de registro de nuevos usuarios selección de País/Provincia/Ciudad mediante una base de datos mostrada mediante menús desplegables dependientes unos de otros.
- Registrar los pedidos efectuados por los clientes en la base de datos y mostrar un historial en su área personal para cada cliente.
- El mensaje de email desde “Contacto” no llega con el remitente especificado, esto más que una ampliación es un problema que no he tenido tiempo de resolver.

Pasamos a posibles ampliaciones no previstas en la planificación inicial del trabajo:

- Menús personalizados para el administrador web que le permite gestionar la web de forma directa sin introducir direcciones de gestión en la URL de la web.
- Adquisición de un dominio y alojamiento de la web en un proveedor de *hosting*. Actualmente tengo configurada una dirección web para una dirección IP dinámica para que el tutor pueda revisar la web de forma



Portal web de una tienda virtual de productos para animales domésticos

remota en www.peipet.dyndns-web.com, pero de forma temporal dado que tengo un plazo gratuito de prueba y este expirará en breve.

- Cálculo de impuestos (IVA), y aplicación de descuentos sobre los pedidos de los clientes, dado que es un trabajo de ámbito no comercial no he incluido impuestos.
- Pasarelas de pago seguras. Si el proyecto pasara a fase comercial tendría que incluirlas también.
- Sesiones cifradas para los visitantes validados mediante HTTPS.
- Revisar aspectos de seguridad, como la inyección de código SQL mediante la URL.
- Enlaces a las redes sociales operativos con cuentas en cada una de ellas creadas.
- Estudio y aplicación de posicionamiento web.
- Foro para usuarios registrados para que compartan información y experiencia con los productos.
- Traducción de la web a otros idiomas.
- Valoración de los productos mediante puntuación y comentarios a usuarios que los hayan adquirido.

8. Bibliografía

Libros:

Welling, L. y Thomson, L. (2da. Ed.) (Diciembre 2013). *Desarrollo web con PHP y MySQL*. Madrid: Anaya Multimedia.

Winesett J. (2da. Ed.) (Noviembre 2012). *Web Application Development with Yii and PHP*. Birmingham: Packt Publishing Ltd.

Duckett J. (2011). *HTML & CSS, design and build websites*. Indianápolis: Wiley.

Makarov A. (Agosto 2011). *Yii 1.1 Application Development Cookbook*. Birmingham: Packt Publishing Ltd.

Documentos web:

Varios autores. (2008-2014). La guía definitiva para Yii. Consultada a lo largo del desarrollo del TFG (mar-ago de 2014), en <http://www.yiiframework.com/doc/guide/1.1/es/index>

Varios autores. (2008-2014). *Class Reference*. Consultada a lo largo del desarrollo del TFG (mar-ago de 2014), en <http://www.yiiframework.com/doc/api/>

Varios autores. (Agosto 2014). Manual de PHP. Consultada a lo largo del desarrollo del TFG (mar-ago de 2014) en <http://php.net/manual/es/>

Chitnis M., Pravin T. & Ananthamurthy L. (Marzo 2003) *Creating UML Use Case Diagrams*. Consultada a lo largo del desarrollo del TFG (mar-ago de 2014) en <http://www.developer.com/design/article.php/2109801/Creating-Use-Case-Diagrams.htm>

MySQL con clase. (2005). Curso de MySQL. Consultada a lo largo del desarrollo del TFG (mar-ago de 2014) en <http://mysql.conclase.net/curso/index.php>

Medvedev V. (Diciembre 2012). *Shopping Cart*. Consultada a lo largo del desarrollo del TFG (mar-ago de 2014) en <https://github.com/yiiext/shopping-cart-component>

Akshay K. (Enero 2014). *How to install brand new theme on yii app?. Shopping Cart*. Consultada a lo largo del desarrollo del TFG (mar-ago de 2014) en <https://www.youtube.com/watch?v=rAvm51t5Cmo>

Salgado G. (Junio 2013). Curso de Yii Framework. Consultado a lo largo del desarrollo del TFG (mar-ago de 2014) en <http://codigofacilito.com/cursos/yii>

9. Glosario

Listado de términos ordenados por alfabético, la mayoría de las definiciones han sido extraídas de la *Wikipedia* estando aquí reunidas por conveniencia.

Breadcrumbs: La miga de pan (del inglés *breadcrumb*), es una técnica de navegación usada en muchas interfaces gráficas de usuario, además de páginas web, consiste en una línea de texto en la que se indica el recorrido seguido y la forma de regresar. Permite que el usuario conozca la ruta de su ubicación en directorios y subdirectorios, y navegue a través de ella.

Captcha: Son las siglas de *Completely Automated Public Turing test to tell Computers and Humans Apart* (prueba de Turing completamente automática y pública para diferenciar computadoras (ordenadores) de humanos). Se utiliza para determinar cuándo el usuario es o no humano. Se usa en páginas web sobre todo para evitar que software malicioso pueda manipular formularios.

CRUD: En computación CRUD es el acrónimo de Crear, Obtener, Actualizar y Borrar (del original en inglés: *Create, Read, Update and Delete*). Se usa para referirse a las funciones básicas en bases de datos o la capa de persistencia en un software.

CSS: Hoja de estilo en cascada o CSS (siglas en inglés de *cascading style sheets*) es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML. *El World Wide Web Consortium (W3C)* es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

Debugging: O depurar un programa, es el proceso de identificar y corregir errores de programación. El término en inglés se asemeja a la eliminación de bichos (*bugs*), manera en que se conoce informalmente a los errores de programación.

Framework PHP: Representa una arquitectura de software, en este caso sobre el lenguaje PHP, que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

Front Controller pattern: Es un patrón de diseño que se basa en usar un controlador como punto inicial para la gestión de las peticiones. El controlador gestiona estas peticiones, y realiza algunas funciones como: comprobación de



restricciones de seguridad, manejo de errores, mapear y delegación de las peticiones a otros componentes de la aplicación que se encargarán de generar la vista adecuada para el usuario.

GNU GPL: La Licencia Pública General de GNU o más conocida por su nombre en inglés *GNU General Public License* (o simplemente sus siglas del inglés GNU GPL) es la licencia más ampliamente usada. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Hosting: El alojamiento web (en inglés *web hosting*) es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web.

HTML: Siglas de *HyperText Markup Language* («lenguaje de marcas de hipertexto»), hace referencia al lenguaje de marcado para la elaboración de páginas web. Define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, etc. Es un estándar a cargo de la W3C.

HTTP: *Hypertext Transfer Protocol* (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la *World Wide Web*.

IEEE 830-1998: Estándar del IEEE (*Institute of Electrical and Electronics Engineers*), que se utiliza para hacer una Especificación de Requisitos de Software (ERS), es decir, una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software. Los casos de uso también son conocidos como requisitos funcionales. Además de los casos de uso, la ERS también contiene requisitos no funcionales (o complementarios).

IDE: Un entorno de desarrollo integrado, llamado también IDE (sigla en inglés de *integrated development environment*), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios. Suele consistir en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

Layout: Esquema de distribución de los elementos dentro un diseño, en este caso en el diseño de una página web, en este sentido, puede ser una especie de plantilla que presenta tablas o espacios en blanco. La idea es que, a partir del *layout*, la página web comience a desarrollarse con sus contenidos específicos.

Mapeo Objeto-Relacional (ORM): El mapeo objeto-relacional es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional como motor de persistencia. En la práctica esto crea una base de datos orientada a objetos virtual, sobre la base de datos relacional.

MVC: Siglas de *Model View Controler* o Modelo Vista Controlador, modelo que estructura por capas la programación. Los modelos agrupan y manejan las operaciones

lógicas, las vistas están encargadas de representar la información en pantalla y los controladores hacen de intermediario entre ambos, controlando el acceso a estos y haciendo de interfaz para archivos, scripts, programas o usuarios.

MySQL: Sistema de gestión de bases de datos relacional, multihilo y multiusuario ofrecida por *Oracle Corporation* en un esquema de licenciamiento dual, por un lado ofrece licencia GNU GPL y por otro lado mediante licencia en caso de vaya a ser utilizado con fines comerciales.

PHP: Lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.

Posicionamiento web: Es el proceso de mejorar la visibilidad de un sitio web en los resultados de los diferentes buscadores. También es frecuente nombrarlo por su título inglés, SEO (*Search Engine Optimization*).

Programación Orientada a Objetos (POO): La programación orientada a objetos o POO (OOP según sus siglas en inglés) es un paradigma de programación que usa los objetos en sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento. Los objetos son entidades que tienen un determinado estado, comportamiento (método) e identidad.

Slider: Elemento web que “desliza” o hace una transición entre una secuencia de imágenes mostrándolas con un tiempo y un efecto determinado.

W3c: El World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce recomendaciones para la World Wide Web. Fue creado en octubre de 1994 y está dirigida por Tim Berners-Lee, el creador original de URL (Uniform Resource Locator, Localizador Uniforme de Recursos), HTTP (HyperText Transfer Protocol, Protocolo de Transferencia de HiperTexto) y HTML (Hyper Text Markup Language, Lenguaje de Marcado de HiperTexto) que son las principales tecnologías sobre las que se basa la Web.

Widget: Un *widget* o artilugio, es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de *widgets* o *Widget Engine*. Entre sus objetivos está el dar fácil acceso a funciones frecuentemente usadas y proveer de información visual. Aunque no es condición indispensable, los widgets suelen ser utilizados para ser "empotrados" en otra página web, copiando el código que el mismo widget pone a disposición del usuario.

Wireframe: En diseño web, es una representación esquemática de una página web sin elementos gráficos que muestran contenido y comportamiento de las páginas. Sirven como herramienta de comunicación y discusión entre arquitectos de información, programadores, diseñadores y clientes. También se pueden utilizar para comprobar la usabilidad de un sitio web.



Portal web de una tienda virtual de productos para animales domésticos

Yii Framework: *Framework* PHP creado por Qiang Xue, el cual empezó su desarrollo en enero de 2008. Actualmente se encuentra en la versión estable 1.1.15 y es uno de los *framework* PHP más utilizados a día de hoy.