

UniversEcom: Tu tienda de informática

Carles Palau Girón

Resumen—Junto al desarrollo y popularización de internet, ha nacido comercialmente un nuevo aire optimista que prefiere ofrecer sus productos virtualmente acogiendo nuevas ventajas competitivas: comodidad, instantaneidad y universalidad. Pero no todo son beneficios en el comercio electrónico ya que éste impide al cliente conocer en primera persona el producto que va a comprar, formular quejas de manera eficaz y diligente ni tampoco garantiza la seguridad de los datos personales del comprador ante la existencia de piratas informáticos. UniversEcom se presenta, por tanto, como una tienda electrónica que pretende atender a esas citadas irregularidades del comercio online y destacar en el mundo del negocio innovando en su administración, de manera que el cliente pueda gestionar sus propios productos a través del móvil sin necesidad de tener que entrar al sitio Web.

Palabras clave—Tienda online, e-commerce, Webservice, REST, PHP, HTML, JavaScript, JSON, Gestión tienda, Seguridad, adaptabilidad, Comprador, Vendedor, Reparaciones, Recambios, Componentes informáticos, Posicionamiento, app, Android.

Abstract—Next to the development and popularization of Internet, has emerged commercially a new optimistic air which prefers offer its products in a virtual way hosting new competitive advantages: comfort, instantaneousness, and universality. But not everything are benefits in online retail because it prevents the customer from knowing first-hand the product that he goes to buy, from submitting complaints effectively and diligently and also ensuring the personal data security of the purchaser towards the existence of hackers. UniversEcom is therefore presented as an electronic store which pretends attend to those cited irregularities of the online business and seeks to point out in the world of business awarding for innovation in its administration, it will allow the customer to manage his own products over the mobile phone without the need to access the website.

Index Terms—Webshop, e-commerce, Webservice, REST, PHP, HTML, JavaScript, JSON, store management, security, responsive, purchaser, vendor, repairs, spare parts, computer components, positioning, app, Android.



1 INTRODUCCIÓN

UniversEcom, se presenta como una tienda electrónica que pretende atender a las actuales irregularidades del comercio online como pueden ser la falta de seguridad en el pago o en los datos, los precios abusivos, la recepción tardía del producto, la navegación complicada, páginas no adaptables al dispositivo o la existencia de webs mal estructuradas en las que resulta complicado realizar la compra. Pretende nuestra página, por tanto, destacar en el mundo de los negocios innovando en su administración, de manera que el cliente pueda gestionar sus propios productos a través del móvil sin necesidad de tener que entrar al sitio web.

Este proyecto confía en el éxito y viabilidad de la innovación gracias a la implementación de una pasarela de pago que facilita la fluidez económica entre comprador y vendedor y gracias, también, a la creación de una *app* para el sistema operativo Android que permite subir los productos a la página en menos de cinco sencillos pasos. A esto debemos añadir la comodidad, no sólo del cliente sino también del administrador, quien en una sola mano pre-

senta el beneficio de poder automatizar sus propias gestiones como si se tratara de un sistema CSM, pero completamente editable.

La estructura de este proyecto se divide en tres grandes módulos que interactúan entre sí. El primero de ellos es la página web, tanto la interfaz del usuario como la parte de administración por parte del dueño de la tienda. El segundo es una aplicación Android desarrollada en Java, encargada de gestionar la tienda. La tercera es un *webservice* cuyo objetivo es integrar la aplicación Android con la web de manera que puedan interactuar gracias a la gestión que hace el *webservice* de la base de datos.

En este documento, se detalla el problema a resolver y la solución a implantar determinando los objetivos y requisitos del proyecto así como su implementación final. También se explica la metodología que se ha seguido y la planificación del proyecto.

- E-mail de contacto: palau.carles@gmail.com
- Mención realizada: Ingeniería del Software
- Trabajo tutorado por: Yolanda Benítez Fernández (Departamento de Ciencias de la Computación)
- Curso 2016/17

2 ESTADO DEL ARTE

UniversEcom tiene como propósito innovar tanto en la edición de la tienda online por parte del cliente como en la gestión administrativa por parte del vendedor.

2.1 Situación actual comprador

Actualmente, en el mundo comercial existen dos vías posibles para llevar a cabo una operación de compra-venta:

Por una parte, tenemos las tiendas físicas, donde los usuarios más conservadores y tradicionales efectúan sus compras. Por otro lado, las tiendas online, donde los usuarios pueden realizar sus compras desde el ordenador de su casa sin tener que desplazarse a una tienda física.

- Tiendas Físicas:

Comprar en tiendas físicas es la forma de compra más antigua y tradicional que existe, pero no por ello la mejor, ya que nos encontramos con una serie de inconvenientes como pueden ser los precios abusivos o el bajo *stock* de productos. A ello le tenemos que añadir la inversión de tiempo que debe hacer el usuario para realizar la compra y la descuidada proximidad por parte del vendedor hacia el cliente a la hora de poder observar el producto que quiere comprar.



- Tiendas Online:

Cada vez es mayor el número de gente que opta por esta opción ya que su formato evita muchos de los problemas que tiene la compra en una tienda física como son la poca variedad de productos, la reducción de tiempo invertido por parte del comprador o la poca intimidad del cliente a la hora de revisar el producto. No obstante, son numerosas las tiendas online que presentan otros tipos de inconvenientes: la escasa seguridad en el pago, la tardía recepción del producto, la no examinación del producto en primera mano, la ardua complejidad que supone la operación de compra, la no adaptación de la página al dispositivo del usuario o la dificultad añadida en torno a la navegación.



- Mejoras que aplicará UniversEcom para el comprador

La estrategia seguida para sumar adeptos a nuestra página web es la de ofrecerles a los clientes algunos de los beneficios que tienen las tiendas físicas, como por ejemplo la examinación del producto (realizando fotos detalladas del mismo) o la facilitación de un método de devolución.

Y entre los beneficios otorgados a los usuarios que ya compran en Internet, cabe señalar la optimización en los servicios de transacción, lo cual incluye una sencilla navegación y una fácil gestión de compra que tienen como finalidad la creación de una página lo más cómoda e intuitiva posible. A esta confortabilidad se integran la adaptación de la página a cualquier dispositivo, la excelente seguridad en el pago y la comprometida entrega del producto dentro del periodo establecido.

2.2 Situación actual vendedor

Son cuantiosas las personas que han decidido y deciden a día de hoy emprender un negocio en internet debido al bajo coste económico que supone el mantenimiento de éste. A continuación, procedemos a presentar los distintos sistemas administrativos a los que se puede optar a la hora de contratar una tienda online:

- Tienda online con proveedor de hosting:

En el momento de contratación de un alojamiento web, el mismo alojamiento ofrece paquetes para que sea posible la implementación de nuestra tienda online. Si aceptamos el contrato con el proveedor de hosting se nos ofrecerá la posibilidad de elegir una plantilla que adaptaremos a nuestro gusto dentro de las posibilidades que se ofrezcan. Es la opción más rápida, pero no la más beneficiosa ya que a parte del desembolso económico que supone la contratación, las plantillas ofrecidas son muy poco flexibles, presentando un diseño muy básico. El inconveniente más notable, no obstante, es que si decidimos cambiar de proveedor de hosting perdemos nuestra tienda por completo.

- Tienda online con gestor de contenido (CSM)

Un CSM (Content Management System) es un sistema de gestión que nos permite publicar y modificar contenidos. En el caso de una tienda online opera en la gestión de productos. Como ventaja principal de este sistema encontramos la automatización de éste, que permite a un usuario sin grandes nociones tecnológicas tener su propia tienda. Estos gestores tienen un coste reducido de contratación (hay algunos incluso gratuitos) y disponen de una comunidad sobresaliente en la reparabilidad de problemas. Es una alternativa productiva, pero no permite un diseño completo sobre la tienda, es decir, el administrador edita sobre una plantilla que únicamente posibilita la modificación de ciertos aspectos visuales, pero no del 100% de ellos. En nuestro caso, se busca instaurar una tienda totalmente editable con el fin de moldear la página exactamente atendiendo al gusto del cliente. Se pretende, por tanto, conseguir una gestión sencilla y para ello un CSM ofrece diversos módulos de los cuales sólo unos pocos sirven de utilidad.

- UniverEcom, tienda online a medida:

Esta es sin duda la opción más costosa en cuanto a administración pero también la más económica. No supone un desembolso económico y prevalece la libertad otorgada en torno a la edición del diseño: podemos añadir, quitar o modificar funcionalidades a nuestro gusto. Desde un inicio se nos presentan los módulos básicos de gestión de la tienda y con el tiempo podremos ir añadiendo más módulos según nuestras necesidades. También conocemos, a primera instancia, toda la estructura de nuestra plataforma y la seguridad será implantada por nosotros sin dejarla en manos de terceros.

La gran mayoría de personas interesadas en montar un negocio online no tiene grandes conocimientos en tecnología por lo que optan por sistemas o plantillas que faciliten la gestión de la tienda para así despreocuparse del diseño de la misma. El creador busca vender productos lo antes posible sin pensar cuales son las consecuencias de decidirse por una u otra alternativa. Seguramente el administrador, al emprender el negocio, ahorrará más tiempo eligiendo una de las dos primeras alternativas, pero en un futuro esta elección puede dificultar la posibilidad de editar la página.

Por lo que llegamos a la conclusión de que la mejor alternativa, a largo plazo, es montar nuestra tienda nosotros mismos partiendo completamente desde cero ya que conoceremos a la perfección la estructura de la misma y la podremos adaptar a nuestras necesidades sin un gran desembolso económico.

3 OBJETIVOS

Para poder lograr la perfecta creación de una tienda virtual que funcione por completo de manera automatizada, será necesario haber ido alcanzando, a lo largo del proyecto, una serie de objetivos. A continuación, pasamos a listar los objetivos del trabajo con sus respectivas prioridades:

Objetivo	Prioridad
Aprehensión de los requisitos de los usuarios finales en la fase inicial del proyecto.	→
Diseño e implementación del sitio Web con toda la interacción usuario-interfaz antes de la entrega del proyecto.	→
Creación de una pasarela de pagos para la venta de productos.	→
Aplicación de las técnicas de posicionamiento web para salir como primera opción en los motores de búsqueda.	→
Aplicación de técnicas de seguridad durante el desarrollo de las distintas partes de la web.	→
Testación de cada módulo con el fin de comprobar si realiza su función específica sin permitir la ejecución de acciones maliciosas.	→

Búsqueda y contratación del alojamiento y <i>hosting</i> en la fase inicial del proyecto.	→
Desarrollo de una <i>app</i> en Android para la gestión de la tienda.	→
Integración de las distintas partes del proyecto (web tienda <i>online</i> , <i>webservice</i> y <i>app</i> Android para la gestión de la tienda) a fin de conseguir que se comuniquen correctamente sin redundancia en la información.	→

Tabla 1: Objetivos del proyecto

4 METODOLOGÍA

Para conseguir los citados objetivos se ha seguido un desarrollo iterativo e incremental propio de la metodología *SCRUM*. Este proyecto ha sido compuesto por una sola persona por lo que resulta imposible ceñirse a la metodología *SCRUM* en su totalidad, pero sí que ha sido posible adaptar ciertas prácticas utilizadas en ella, puesto que se trata de un proyecto complejo donde existe la necesidad de obtener resultados de manera inmediata y donde los requisitos pueden ir variando con el tiempo.

La aplicación de estas prácticas nos ha aportado una mayor flexibilidad a la hora de resolver cambios que se hayan podido producir a lo largo del proyecto, así como una óptima calidad del software, una imprescindible predicción del tiempo y una reducción de riesgos. Pero, sobre todo, nos ha ayudado a mejorar la productividad dado que ha habido entregas constantes del producto.

El proyecto ha sido puesto en marcha en forma de iteraciones cuya duración era de dos semanas. Estas iteraciones se entienden como pequeños proyectos que facilitaban el cumplimiento de los objetivos de forma progresiva. En cada iteración se han ido realizando entregas parciales del producto final.

A continuación pasamos a describir de manera más detallada, la metodología utilizada:

- El primer paso ha sido crear un *product Backlog*, es decir, una lista de requisitos priorizada.
- El segundo paso consistía en elegir una tarea (o más) del *product Backlog* e iniciar el *sprint*.
 - Antes de finalizar dicho *sprint*, unos días antes, se pactó una reunión con el tutor para que destacase los posibles cambios que se podrían efectuar en el ítem entregable, disponiendo de margen para su modificación antes del cierre del *sprint*.
- Una vez finalizado un *sprint*, se evaluó que el entregable estuviera correcto y no hubiera errores. Estando todo exacto se finalizó el *sprint*, se eligió

otra tarea del *product Backlog* y se inició el siguiente *sprint*.

- En el último *sprint* se hizo un entregable completo del producto final junto con la documentación del *software*.

5 PLANIFICACIÓN

Este proyecto ha tenido una duración de aproximadamente tres meses comprendidos en el período del 02/10/2016 al 06/01/2017. Al principio se definió una planificación dejando dos semanas de margen ante cualquier imprevisto. Finalmente, hubo que recurrir a esas dos semanas para ampliar el tiempo del tercer *sprint*, para acabar de implementar la web y para ampliar el sexto *sprint* para ultimar la implementación de la aplicación Android.

A continuación, vemos lo entregables que se han ido entregando al cliente:

Sprint	Entregables
0	Informe Inicial
1	Encuesta, Prototipo Web, Prototipo <i>app</i>
2	Sitio Web, Modelo E-R
3	Sitio Web, Módulo pasarela de pagos, Creación BBDD, Informe de progreso I, Plan de viabilidad
3B	Aplicativo Web
4	Casos de uso, UML
5	Webservice, Informe de progreso II
6B	<i>App</i> Android
8	Entregar del aplicativo final

Tabla 2: Entregables del proyecto

Cabe mencionar, que sólo se muestran los *sprints* con entregables.

La planificación se detalla en el apéndice A1.

6 REQUISITOS

Gracias a la meticolosa recopilación de requisitos, que se mencionarán a continuación, se ha logrado el siguiente objetivo: aprehensión de los requisitos de los usuarios finales en la fase inicial del proyecto.

6.1 Recopilación de requisitos

El proyecto desarrollado está enfocado plenamente al cliente. De la respuesta que los usuarios tuvieron respecto a la tienda surgió el triunfo del proyecto, por lo que ellos han sido nuestro foco de atención principal.

Se les realizó a los usuarios una encuesta de once preguntas en las cuales se cuestionaba desde los inconvenientes que se encontraban a la hora de hacer una compra en una página web hasta los métodos de pagos preferibles. Entre el resto de cuestiones debieron responder también sobre los aspectos por los que preferían comprar *online* o de manera presencial.

¿Qué es lo que valora más a la hora de hacer una compra por Internet?
(60 respuestas)

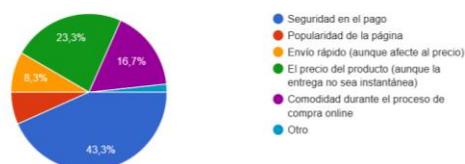


Figura 1: Pregunta incluida en la entrevista

6.2 Requisitos funcionales

A continuación, se especifican los requisitos funcionales junto con su importancia, clasificados por el tipo de plataforma así como también por el aspecto de haber sido o no realizados:

6.2.1 Requisitos funcionales web

Se clasifican los requisitos funcionales de la web atendiendo de a la finalidad del comprador y del vendedor.

➤ Comprador:

Importancia	Requisitos
Esencial	Realizar compras mediante el sistema de pago PayPal y tarjeta de crédito. ✓
	Una vez efectuada una compra se dispondrá de un sistema de devolución. ✓
	Mostrar factura al terminar compra. ✓
	Envío de formulario para solicitar presupuesto sobre una reparación. ✓
	Comunicación con los administradores de la tienda mediante formulario. ✓
	Registrarse en la Web. ✓
	Iniciar o cerrar la sesión. ✓
	Navegar y consultar productos. ✓
	Listar productor por categorías. ✓
	Añadir/Quitar/Consultar carrito de compra. ✓
Condicional	Darse de baja. ✓
	Recordar contraseña. ✓
	Modificar información registro ✗
Opcional	Realizar una búsqueda de productos. ✓
	Múltiples idiomas. ✗
	Seguimiento de productos marcados como deseos. ✗
	Notificación de producto marcado como deseo en caso de variación precio. ✗

Tabla 3: Requisitos funcionales de la web (comprador)

➤ Vendedor:

Importancia	Requisitos
Esencial	Subir producto. ✓
	Borrar producto. ✓
	Subir compra. ✓
	Borrar compra. ✓
	Consultar compras. ✓
	Consultar ventas. ✓
	Consultar usuarios. ✓

Condicional	Modificar producto.	✗
	Modificar compra.	✗
	Consultar productos.	✗
Opcional	Consultar ganancias.	✓
	Consultar perdidas.	✓
	Consultar total usuarios.	✓
	Exportar información.	✗

Tabla 4: Requisitos funcionales de la web (vendedor)

6.2.2 Requisitos funcionales App

Se exponen las condiciones funcionales de la app.

Importancia	Requisitos	
Esencial	Subir producto.	✓
	Borrar producto.	✓
	Subir compra.	✓
	Borrar compra.	✓
	Consultar compras.	✓
	Consultar ventas.	✓
	Consultar usuarios.	✓
Condicional	Modificar producto.	✓
	Modificar compra.	✓
	Consultar productos.	✓
Opcional	Consultar ganancias.	✓
	Consultar perdidas.	✓
	Consultar total usuarios	✓
	Exportar información.	✗

Tabla 5: Requisitos funcionales de la app

6.3 Requisitos no funcionales

A continuación, se especifican los requisitos no funcionales junto con su importancia, clasificados por el tipo de plataforma así como también por el aspecto de haber sido o no realizados:

6.3.1 Requisitos no funcionales web

Listado de requerimientos no funcionales de la web.

Importancia	Requisitos	
Esencial	Se debe procesar la petición en la base de datos en menos de 7 segundos después de realizar el clic.	✓
	El sistema debe ser capaz de operar correctamente con hasta 25 usuarios con sesiones abiertas operando de forma concurrente.	✓
	Los datos modificados en la base de datos han de ser actualizados sin duplicidad.	✓
	El sistema debe mostrar mensajes de errores que describan de manera clara lo que está haciendo mal el usuario.	✓
	El sistema debe utilizar un diseño adaptable para garantizar la visualización en cualquier dispositivo.	✓
	Los datos de los usuarios no serán revelados.	✓

	Encriptación de contraseñas sha256.	✓
Condicional	Diariamente se creará una copia de seguridad de la base de datos.	✓
Opcional	Se debe publicitar la web.	✗
	Debemos aplicar técnicas de Posicionamiento SEO para aparecer en motores de búsqueda.	✗

Tabla 6: Requisitos no funcionales de la web

6.3.2 Requisitos no funcionales App

Se formulan, en la siguiente tabla, las condiciones no funcionales de la app.

Importancia	Requisitos	
Esencial	Se debe procesar la petición en la base de datos en menos de 7 segundos después de realizar el clic.	✓
	Los datos modificados en la base de datos han de ser actualizados sin duplicidad.	✓
Condicional	El diseño ha de ser intuitivo.	✓
Opcional	Interfaz agradable.	✓

Tabla 7: Requisitos no funcionales de la app

6.4 Restricciones

A continuación, se especifican restricciones del sistema clasificadas según su tipología:

Tipo	Restricciones
Software	La app ha sido desarrollada con Java para Android.
	Se ha utilizado un lenguaje PHP7 para la mayoría de funcionalidades de la web.
Sistema	Se ha creado una cuenta de PayPal con modalidad de vendedores.
Tiempo	La fecha de entrega del proyecto será el siete de febrero.
Rendimiento	El proyecto tiene alta complejidad al ser ejecutado por sólo una persona.
Legal	No se ha vulnerado la Ley 3/2014.
	La Ley 15/1999 de Protección de Datos de Carácter Personal no ha sido transgredida.
	No se ha violado la Ley 34/2002 de Servicios de la Sociedad de la información y el comercio electrónico.

Tabla 8: Restricciones

6.5 Mockup

Una vez obtenida la opinión mayoritaria, se ha llevado a cabo la realización de dos mockups: uno para el aplicativo web y otro para la app en Android, cada uno con un objetivo distinto.

El prototipo de la aplicación va dirigido exclusivamente a los encuestados, por lo que una vez creado se les presentó el modelo a varios de estos usuarios con el fin de extraer

conclusiones y poder mejorar la implementación de la página.

Finalmente, observamos las distintas acciones posibles del gestor en la web:

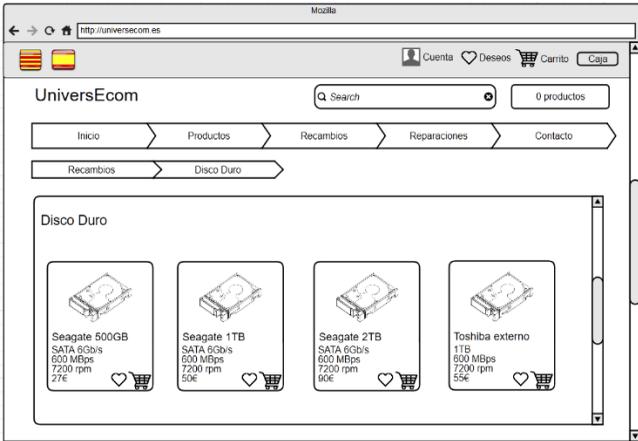


Figura 2: Mockup web

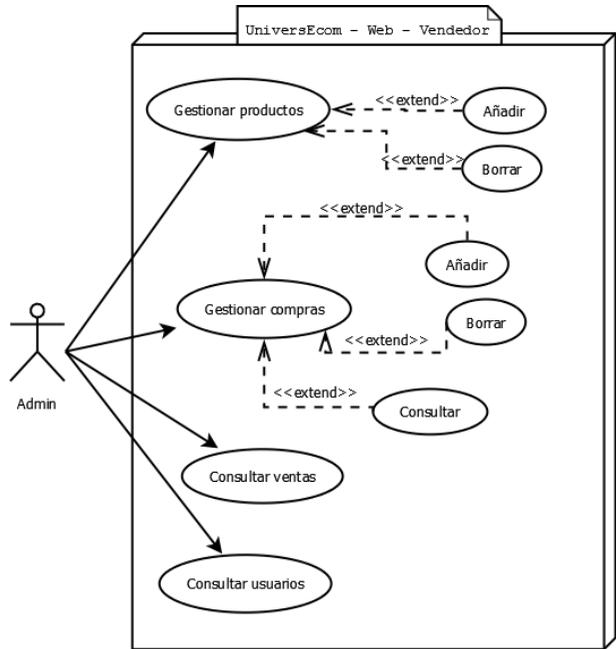


Figura 4: Casos de uso de la web (vendedor)

El prototipo de la aplicación en Android se ha desarrollado únicamente con el fin de establecer una estructura técnica y estética en nuestra página y, a su vez, para facilitar nuestra psicología de diseño a la hora de crear dicho patrón.

6.6 Casos de uso

Procedemos a establecer los casos de uso que han determinado las acciones que podrán hacer tanto comprador como vendedor en nuestra plataforma.

Vemos, por tanto, las actividades a las que opta un comprador en la web:

7 DISEÑO

Una vez reunidos los requisitos y el prototipo, se ha procedido a crear la base de datos *mysql*. A continuación, mostramos el UML resultante del diseño de la base de datos:

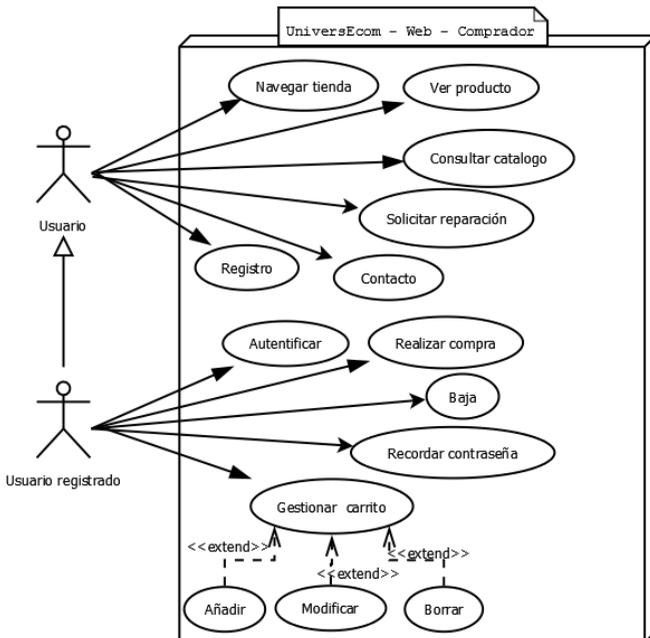


Figura 3: Casos de uso de la web (comprador)

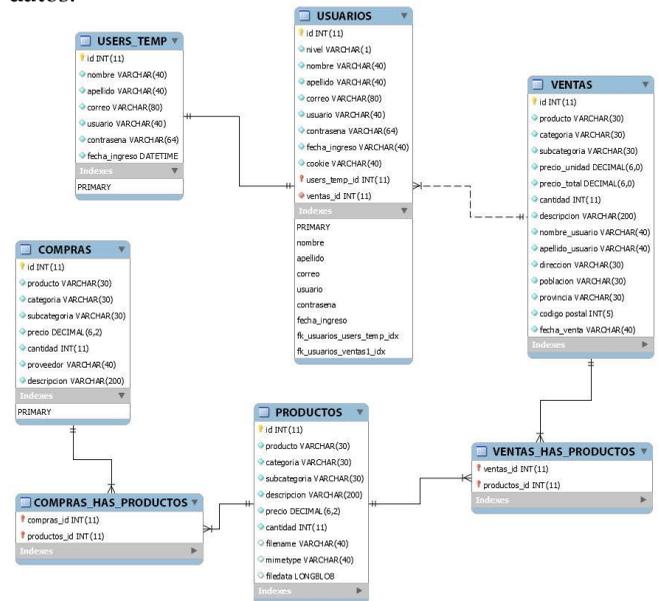


Figura 5: UML

En el Apéndice A2 se refleja el diagrama entidad-relación.

8 IMPLEMENTACIÓN

8.1 Web

8.1.1 Maquetación: HTML, CSS, Bootstrap

Se ha optado por un diseño sencillo, no muy recargado, aunque presenta alguna combinación de colores llamativa.

Para poder cumplir el requisito sobre la adaptabilidad de la web se ha optado por la utilización del *framework Bootstrap* que permite crear diseños simples, limpios e intuitivos. Este *framework* presenta varios elementos con estilos ya predefinidos que son adaptables a cualquier dispositivo. En la siguiente imagen vemos cómo se ha utilizado una clase ya definida en *Bootstrap* en la cual se asigna una imagen y *Bootstrap* la adapta al contenido.

```
<a href="https://www.universecom.es/index.php"></a>
```

Figura 6: Clase *img-responsive* de *Bootstrap*

También se definen estilos dependiendo de la resolución de la pantalla utilizada con el propósito de que el sitio web se adapte a todos ellos sin mayor problema.

8.1.2 Funcionalidad: PHP7, JavaScript, Ajax

Se ha implementado toda la funcionalidad del sitio web con PHP7 ayudándonos en ocasiones por funciones en JavaScript o Ajax. Hay infinitud de aspectos de la web tratados con estas tecnologías.

Se exponen, a continuación, las funcionalidades más importantes clasificadas por categorías:

- Gestión usuario

Los módulos necesarios han sido desarrollados para permitir una plena interacción del usuario con su cuenta, permitiendo su registro en la web y facilitando un método de baja en caso de estar descontento con la página. También se permite recuperar la contraseña, si el usuario la ha olvidado, y un acceso total a la web en caso de loguearse en ella.

En esta interacción que realizan dichos módulos con el usuario, en caso de ser necesaria la notificación por *mail* de avisos, como es el caso del registro (donde se envía un mail de confirmación) o en el caso de recuperar la contraseña (que se envía una nueva al mail del cliente) se utiliza la clase *Phpmailer*, desarrollada en PHP.

- Feedback con el cliente:

El cliente también tiene la posibilidad de solicitar un presupuesto de un producto que esté estropeado. También nos puede dar su opinión, formular quejas o posibles mejoras. Finalmente, permite un sistema de devoluciones. Todo esto está gestionado a través de formularios que dependiendo del asunto envía la información a uno de nuestros correos.



Figura 7: Cliente solicitando una reparación

- Compras:

Para la operación de compra hay una sección de productos y otra de recambios, las cuales muestran todos los productos disponibles e intercambiables. Si entramos dentro podemos filtrar según el tipo de producto. Una vez dentro de los productos, se nos listan junto a una foto, el título del producto, una descripción y un precio. También hay dos botones: añadir al carrito y ver ficha. En la web vemos de manera permanente el carrito en la parte superior derecha. En esta sección, podemos gestionar nuestra compra eliminando productos y se actualizará inmediatamente gracias a una función referenciada a través de Ajax que asiste a la actualización dinámica de la información del carrito. Al acceder a otra página, recupera la información del carrito; actualiza gracias a un código presente en todas las páginas que recupera la sesión del carrito y lo recorre archivando las compras del cliente:

```
session_start();
$compras=$_SESSION['carrito'];

for($i=0;$i<count($compras)-1;$i++){
$total_unidades=$total_unidades + intval($compras[$i]['cantidad']);
$total_precio=($total_precio + $compras[$i]['precio'] * $compras[$i]['cantidad']);
}
$articulos=$total_unidades;
$gastado=number_format($total_precio,2);
```

Figura 8: Actualizando el carrito al acceder a otras paginas

Una vez seguro de lo que se desea comprar, se accede al carrito donde se muestra toda la información de la compra de forma detallada, mostrando producto y detalles del mismo junto con la opción de 'incrementar más unidades de ese producto' o de 'eliminarlo'.

Cuando esté el cliente seguro del contenido de su carrito pasará a la pasarela interna de *UniversEcom*, donde aparecerán tanto el usuario, nombre, apellido y correo de registro del usuario, sin posibilidad de ser modificados los 3 primeros parámetros, pero sí el correo. Además deberá agregar información de envío; para ello debe pulsar en 'pagar' y pasará toda la información de manera oculta a *PayPal* para finalizar la compra y guardar la venta:

test facilitator's Test Store

Resumen de su pedido	
Descripciones	Importe
Compra en Universecom	€524,00
Precio del artículo: €524,00	
Cantidad: 1	
<hr/>	
Importe total a pagar	€524,00
Total €524,00 EUR	

Figura 9: Pasarela de pago de paypal

- Seguridad:

Todos los módulos recogen la información con formularios vía POST. Sólo un aspecto de la web se trata con GET: el acceder a la ficha de un producto, por lo que no afectaría a la seguridad.

Las contraseñas de usuario se han encriptado con el algoritmo *sha256*.

Las consultas están realizadas en PDO (parametrizadas) con lo cual evitan entre otras cosas ataques de SQL *Inyection*.

```

$sql = "SELECT * FROM usuarios WHERE usuario = :usuario
AND contrasena = :contrasena AND correo = :correo";
$stmt = $conexion->prepare($sql);
$stmt->bindParam(':usuario', $usuario, PDO::PARAM_STR);
$stmt->bindParam(':contrasena', $contrasena, PDO::
PARAM_STR);
$stmt->bindParam(':correo', $correo, PDO::PARAM_STR);

```

Figura 10: Consulta en PDO

Se crean números aleatorios con PHP (*mtrand*) y se asignan a variables de sesión de manera que cuando pasamos de página en la web se comprueba que sea el mismo número.

Se ha aplicado seguridad evitando que los usuarios puedan acceder a secciones de las páginas directamente por la URL gracias al envío de campos ocultos que enviamos de una página a otra, de manera que si entramos directamente a la URL esos campos no existen.

En el *login*, se comprueba el usuario, la contraseña y el nivel de usuario. Si la información coincide y el nivel es cero (por defecto), entonces el usuario entra en su sesión. Si la información coincide pero, por el contrario, el nivel es igual a 1 entonces significa que el usuario es administrador y pasamos a la parte del *backend* para gestionar la tienda. En esta parte también hay un *checkbox* para recordar la contraseña. Si éste es pulsado se genera una *cookie* con un número aleatorio que permite guardar la sesión.

```

//Comprobamos si el usuario esta registrado con una cookie
guardada
if (isset($_COOKIE["id_user_sdc"]) && isset($_COOKIE[
"num_ale_user_sdc"])) {
//Si hay cookies guardadas comprobamos que no estén vacías
if ($_COOKIE["id_user_sdc"] != "" || $_COOKIE[
"num_ale_user_sdc"] != "") {

```

Figura 11: Cookies

A raíz de todas estas implementaciones se ha conseguido el objetivo de aplicar técnicas de seguridad durante el desarrollo de las distintas partes de la web.

- Backend:

Se accede a esta parte de la web cuando el usuario se loguea y está registrado con el nivel 1, por lo que significa que es administrador. La parte del *backend* permite, como ya se ha comentado en los requisitos, subir o eliminar compras/productos, así como listar usuarios y ventas. Se ha realizado con métodos ya comentados en PHP y PDO para insertar registros en la base de datos.

8.2 Webservice

Antes de crear la aplicación para gestionar la tienda se necesita un intermediario que consulte la base de datos y muestre en un enlace web el resultado en JSON, de manera que la aplicación se pueda conectar y consultar los datos. Este intermediario está desarrollado con PHP y transmite los datos sobre *http* gracias a la arquitectura REST.

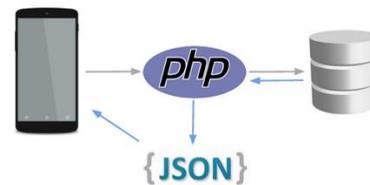


Figura 12: Webservice

La estructura del *webservice* ha sido creada de la siguiente manera: hay un servicio central para cada tabla de la base de datos. Por ejemplo cuando obtenemos el stock de productos, nuestro servicio central es *ws_productos* donde se definen la clase llamada *productos* y sus distintas funciones (listar, insertar, actualizar y borrar).

```

public static function getAll()
{
    $consulta = "SELECT id, producto, categoria, subcategoria,
tripcion, precio, cantidad FROM productos";
    try {
        // Preparar sentencia
        $comando = Database::getInstance()->getDb()->prepare($consulta);
        // Ejecutar sentencia preparada
        $comando->execute();
        return $comando->fetchAll(PDO::FETCH_ASSOC);
    }
}

```

Figura 13: Servicio *ws_productos* (función listar productos)

Después hay un servicio adicional para cada función que únicamente se encarga de llamar a la clase de nuestro servicio central y finalmente mostrar el resultado.

```

include 'ws_productos.php';
if ($_SERVER['REQUEST_METHOD'] == 'GET') {
    // Manejar petición GET
    $producto = Productos::getAll();
    if ($producto) {
        $datos["estado"] = 1;
        $datos["producto"] = $producto;
        print json_encode($datos);
    } else {
        print json_encode(array(
            "estado" => 2,
            "mensaje" => "Ha ocurrido un error"
        ));
    }
}

```

Figura 14: Servicio *ws_obtener_productos*

Por ejemplo, como vemos en la imagen anterior el servicio `ws_obtener_productos` se ejecuta y llama a la clase `Productos`, función `getAll` (listar), situada en el servicio central `ws_productos` y finalmente los presenta en la ruta `http` que le hayamos indicado, en formato JSON.

El resultado es el siguiente:

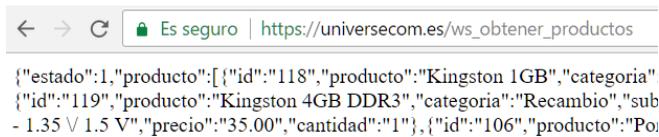


Figura 15: Datos mostrados tras la ejecución del webservices

En las funciones de insertar/modificar o eliminar, se lleva a cabo la misma operativa, pero en este caso en vez de recuperar datos, los edita por lo que previamente en nuestros servicios web debemos dejar bien clara la estructura de la base de datos a utilizar.

8.3 App

La *app* se ha desarrollado con Java y ofrece una gran amplitud de posibilidades, comentadas ya, en los requisitos. Todas las funcionalidades siguen la misma estructura: primero hay una pantalla que, según la opción que se escoja realizar, dirige a una *activity* o a otra. En cuanto a la interacción con el *webservice*, todo está programado en base a la ejecución de tareas en segundo plano. Al tratarse de operaciones cuya velocidad de carga depende del tamaño de la base de datos, se ha procedido a la creación de un hilo secundario que se ejecute dejando el hilo principal libre con tal de que nuestra interfaz de usuario actualizara sin problemas. Para lograr lo citado ha sido creada una clase que extiende de la clase `AsyncTask` que nos ofrece Android y de la cual han sido utilizados sus métodos para gestionar las tareas. Por ejemplo, siguiendo con el ejemplo del apartado anterior donde el *webservice* facilitaba vía web los productos en formato JSON, vemos como gestiona la *app* para obtener esos datos y mostrarlos de forma correcta:

1. Muy importante: definimos las variables junto con la ruta desde donde la *app* consulta la información facilitada por el *webservice*:

```
// IP de mi Url
String IP = "https://universecom.es";
// Rutas de los Web Services
String GET = IP + "/ws_obtener_productos";
```

Figura 16: Ruta servicio web

2. El método `onCreate` entrará en funcionamiento una vez pasemos de *layout* y será lo primero que se ejecutará. Por lo que aquí, hemos creado nuestro "thread" e instanciamos a nuestro método que es el encargado de obtener la información. Este método está dentro de otro; `doInBackground`, para ejecutarse de manera silenciosa para el usuario.
3. En `doInBackground` se ejecutan las tareas de recopilación y parseo de datos, facilitados por el *webservice*. Para eso, establecemos la conexión y pone-

mos los datos en un *bufferReader*. Se comprueba que el estado es igual a 1 ya que es un indicador de que hay datos proporcionados por el *webservice* que han sido parseados.

```
for (int i=0; i<usuariosJSON.length(); i++) {
    devuelve = devuelve + "ID producto: " + usuariosJSON.getJSONObject(i).getString("id") + "\n" + "Categoria: " + usuariosJSON.getJSONObject(i).getString("categoria") + "\n" + "Subcategoria: " + usuariosJSON.getJSONObject(i).getString("subcategoria") + "\n" + "Precio: " + usuariosJSON.getJSONObject(i).getString("precio") + "\n" + "Cantidad: " + usuariosJSON.getJSONObject(i).getString("cantidad") + "\n\n";
}
```

Figura 17: Parseando los datos

4. Finalmente, en el método `onPostExecute` mostraremos el resultado de la tarea ejecutada.

De esta manera, obtenemos un aplicativo que responde rápidamente a las acciones realizadas por el gestor.

En caso de insertar información en la base de datos, en lugar de obtenerla: primero se crea la estructura correcta para mandar los datos introducidos por el usuario al *webservice* y, en segundo lugar, éste los introducirá en la base de datos.

También se permite seleccionar una imagen de la galería o realizar una fotografía gracias a la creación de métodos que permiten la interacción con el dispositivo móvil.

```
private void openGallery() { //Método para abrir la galería y seleccionar imagen
    Intent gallery = new Intent(Intent.ACTION_PICK, MediaStore.Images.Media.INTERNAL_CONTENT_URI);
    startActivityForResult(gallery, PICK_IMAGE);
}
```

Figura 18: Función para abrir la galería de Android

Como resultado de toda la parte de implementación se han obtenido los objetivos referentes tanto al desarrollo de la web como al desarrollo de la *app*, así como también el objetivo de la integración de los distintos módulos y el objetivo referente a la seguridad.

9 PRUEBAS

La parte del testeo de la web, se ha dividido en: *manual testing* y *automation testing*.

- Manual testing:

Se han realizado pruebas de caja negra para evaluar el cumplimiento de los requisitos funcionales. Entrando en detalle, se han realizado casos de prueba a partir de los casos de uso para comprobar que no se puede acceder a partes de la web en las que no se está autorizado. También se ha intentado forzar el error en los formularios introduciendo datos incoherentes, incompletos o vacíos.

- Automation testing:

Ésta posibilidad nos permite aumentar el número de test en un tiempo limitado. Se ha utilizado un *plugin* (*Sql inject me*) de Firefox que realiza una gran diversidad de pruebas para intentar acceder a la base de datos a través de ataques de tipo *sql injection*.

Con todo esto se consigue el objetivo de 'testación de cada módulo' con el fin de comprobar si el módulo realiza su función específica sin permitir la ejecución de acciones maliciosas.

10 RESULTADOS

En UniversEcom se han podido ir poniendo en práctica los conocimientos adquiridos a lo largo de la carrera de ingeniería informática. Como resultado se ha obtenido una página web que beneficia tanto al usuario final, haciendo que la experiencia de compra sea lo más agradable y sencilla posible, y al gestor de la tienda, ya que con la app puede realizar sus gestiones en cualquier momento desde su teléfono móvil, de una forma muy visual e intuitiva. Se presenta, por tanto, como una tienda online que no sólo ha reparado atención en la figura del cliente, sino también en la figura del administrador.

IMAGEN	PRODUCTO	PRECIO	CANTIDAD	TOTAL
	Sobremesa HP 1	250.00	1	250.00
	Sobremesa Acer	150.00	1	150.00
	Monitor Acer 19,5	79.00	3	237.00
	Kingston 1GB	15.00	1	15.00
TOTAL A PAGAR:				€ 652.00 Euros
<input type="button" value="Enviar Pedido"/>				

Figura 19: Web (carrito)

Figura 20: Web (backend)



Figura 21: Imágenes app

11 CONCLUSIÓN Y LÍNEAS FUTURAS

Gracias a la conseguida constancia y a las ansias de competitividad, se puede decir que UniversEcom se erige como una tienda online innovadora, la cual ha sido construida sobre las necesidades y exigencias de sus clientes. Todo esto ha sido posible gracias a una rigurosa planificación la cual ha sido seguida meticulosamente.

A modo personal, este proyecto me ha permitido investigar sobre los *webservices* y sus distintas arquitecturas, así como también conseguir la integración de distintas plataformas de manera que trabajasen conjuntamente. Por otra parte, he podido comprobar la complejidad que presenta la ejecución de un proyecto junto con todos los inconvenientes que uno se puede encontrar en su realización, los cuales han de ser superados para poder proseguir con la planificación establecida.

No obstante, la perfección no se ha alcanzado y la página deberá ir evolucionando mediante la introducción de requisitos opcionales no implementados como son: la modificación informativa sobre el registro del usuario, la opcionalidad en cuanto al idioma, la implementación de un módulo de seguimiento de los productos deseados por el cliente, un sistema de notificaciones que alerte sobre una disminución en el precio del producto, la posibilidad de editar un producto desde la página web, la exportación a un Excel de los datos de compra y venta, la aplicación de técnicas de posicionamiento SEO y, por último, la realización de otro gran número de *testings*.

12 BIBLIOGRAFÍA

[1] Emprendedores. (2016). *¿Es viable mi proyecto de negocio?*. [online] www.emprendedores.es. Available at: <http://www.emprendedores.es/crear-una-empresa/viabilidad-negocio> [Accessed 29 Oct. 2016].

[2] Moragón, R. (2016). *Lo que debes saber sobre la nueva normativa de ecommerce*. [online] www.increta.com. Available at: <http://www.increta.com/es/blog/nueva-normativa-de-ecommerce/> [Accessed 1 Nov. 2016].

[3] Php.net. (2017). *PHP: PDO::query - Manual*. [online] www.php.net. Available at: <http://php.net/manual/es/pdo.query.php> [Accessed 3 Nov. 2016].

[4] Caules, C. (2016). *Introducción Servicios REST - Java*. [online] www.arquitecturajava.com. Available at: <http://www.arquitecturajava.com/servicios-rest/> [Accessed 7 Dec. 2016].

[5] Developer.android.com. (2017). *Android Developers*. [online] [www.developer.android.com](https://developer.android.com/index.html). Available at: <https://developer.android.com/index.html> [Accessed 19 Dec. 2016].

En el apéndice A3 y A4 se reflejan más imágenes tanto de la web como de la app.

APÉNDICE

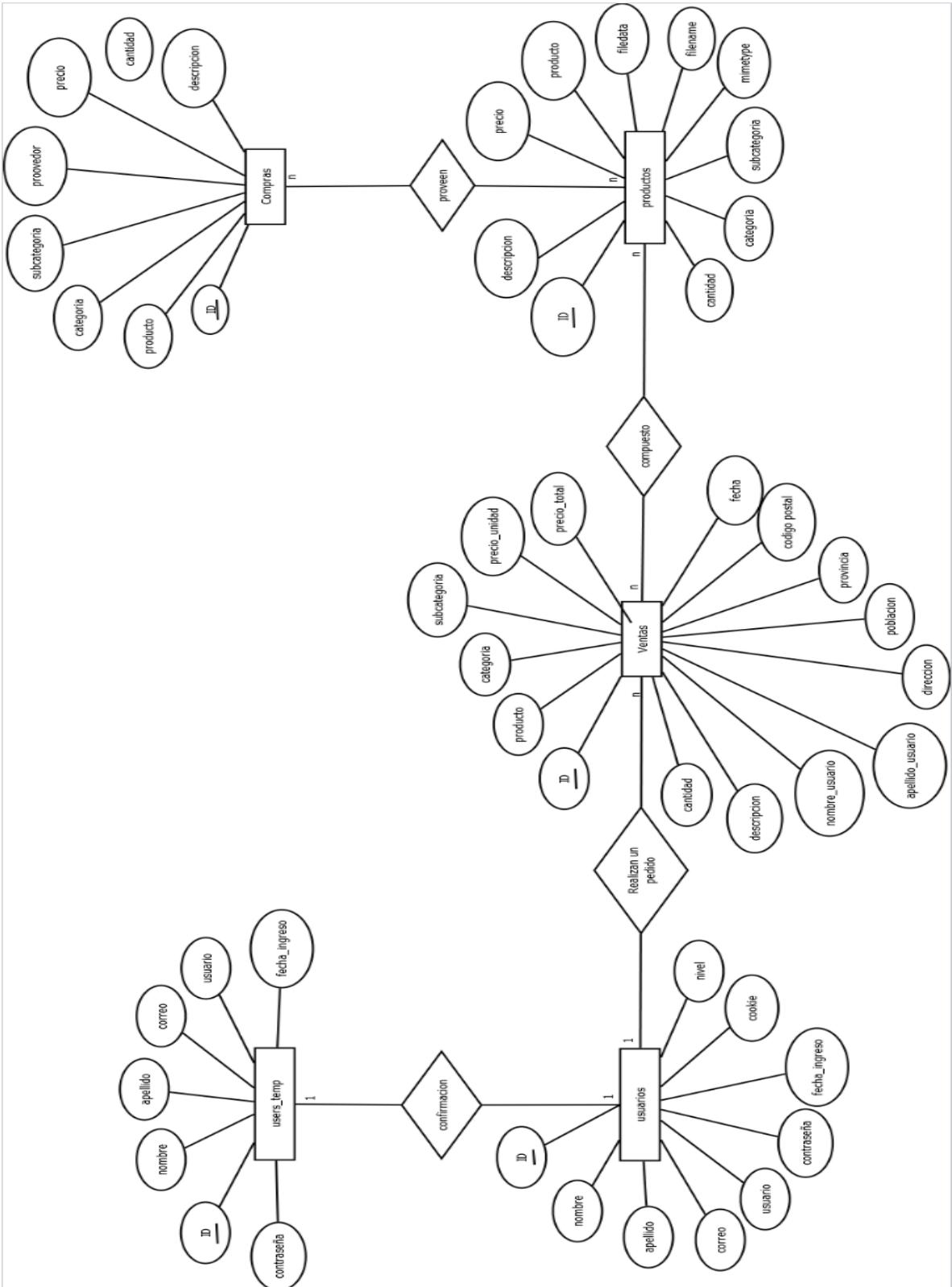
A1. Planificación del proyecto

En este apéndice se añade la planificación seguida que se ha seguido en los distintos *sprints* del proyecto.

SP	Sub-Tarea	Fecha Inicio	Fecha Fin
1	Informe Inicial	02/10/16	02/10/16
1	Captación de requisitos	03/10/16	17/10/16
1	Elección <i>hosting</i> y alojamiento	03/10/16	17/10/16
1	Tareas de documentación e investigación	03/10/16	17/10/16
2	Diseño e implementación sitio web	17/10/16	31/10/16
2	Tareas de documentación e investigación	17/10/16	31/10/16
3	Informe de progreso I	06/11/16	06/11/16
3	Diseño e implementación sitio web	31/10/16	14/11/16
3	Insertar pasarela de pagos	31/10/16	14/11/16
3	Tareas de documentación e investigación	31/10/16	14/11/16
3.1	Diseño web: carrito	14/11/16	21/11/16
4	Posicionamiento web	21/11/16	05/12/16
4	Seguridad web	21/11/16	05/12/16
4	Tareas de documentación e investigación	21/11/16	05/12/16
5	<i>App</i> Android: <i>webservice</i>	05/12/16	19/12/16
5	Tareas de documentación e investigación	05/12/16	19/12/16
5	Informe de progreso II	18/12/16	18/12/16
6	<i>App</i> Android	19/12/16	02/01/17
6	Tareas de documentación e investigación	19/12/16	02/01/17
6.1	Acabar detalles <i>App</i> Android	02/01/17	09/01/17
7	Integración de las tres plataformas	09/01/17	23/01/17
7	Testeo	09/01/17	23/01/17
7	Propuesta informe final	22/01/17	22/01/17
8	Documentación	23/01/17	06/02/17
8	Propuesta presentación	05/02/17	05/02/17
	Entrega final	07/02/17	07/02/17

A2. Diseño de la base de datos

En este apéndice se adjunta el modelo entidad-relación perteneciente a la fase de diseño del proyecto:



A3. Resultados web cliente

En este apéndice se amplía el apartado de resultados, mostrando más imágenes de la web:

INDICE:



MENUS:



CATALOGO:



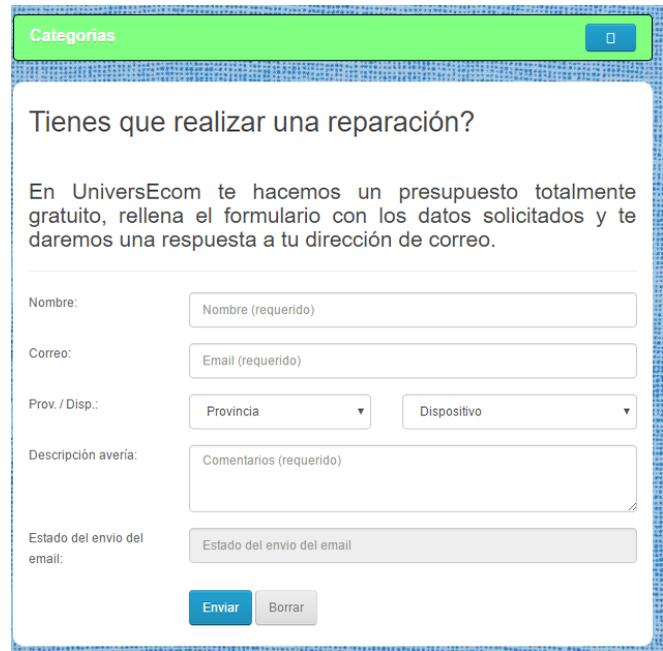
PASARELA DE PAGOS:

Este es el resumen de su compra, verifique su pedido uo modifique sus datos

Usuario	user		
Nombre	user		
Apellido	user		
Dirección	C/ sallent 2		
Poblacion	Terrassa		
Provincia	Barcelona		
Código Postal	08222		
Teléfono	937889856		
Correo	user@user.com		

CANTIDAD	PRECIO	PRODUCTO	TOTAL
1	250.00	Sobremesa HP 1	250.00
1	150.00	Sobremesa Acer	150.00
3	79.00	Monitor Acer 19,5	237.00
1	15.00	Kingston TGB	15.00
TOTAL A PAGAR:			€ 652.00 Euros

FORMULARIOS:



CARRITO EN TODAS LAS PÁGINAS:



EDITAR CARRITO EN CUALQUIE PÁGINA:



A4. Resultados web gestión tienda

En este apéndice se adjuntan imágenes de la parte del *backend* para gestionar la tienda e imágenes adicionales de la app de gestión:

A4.1 Gestión web

INICIO BACKEND:



Bienvenido al panel de administración:

Desde aquí podras gestionar tu tienda online de manera muy sencilla.

Productos: Gestiona los productos de tu tienda:

Subir Producto: Añadir un producto que quieras poner a la venta
 Modificar Producto: Modifica cualquier característica del producto
 Borrar Producto: Borra un producto

Compras: Revisa todas tus compras

Ventas: Revisa todas tus ventas

Usuarios: Gestiona los usuarios de tu tienda

BORRAR PRODUCTO:



Borrar producto

ID	Producto	Categoría	Subcategoría	Precio	Cantidad	Descripción
118	Kingston 1GB	Recambio	Memoria	15.00	1	Kingston Technology System Specific Memory 1GB 667MHz
119	Kingston 4GB DDR3	Recambio	Memoria	35.00	1	Kingston HyperX Impact Black Series - DDR3L - 4 GB - DIMM SO de 204 pines - 1800 MHz / FC3L-12800 - CL9 - 1.35 / 1.5 V
106	Portátil Acer 15,6	Producto	Portátil	430.00	1	Portátil - Acer ES1521-02WL, A6-6310, 16GB RAM, AMD Radeon R4 tamaño : 15.6" (39.6 cm) Modelo de procesador: A6-6310 RAM: 16GB Capacidad de Disco: 1 TB
107	Monitor Acer 19,5	Producto	Pantalla	79.00	1	Monitor - ACER V206HQLAB, HD+ Tamaño (Pulgadas): 19.5 pulgadas Contraste : 100M:1 Tiempo de respuesta : 5ms

CONSULTAR COMPRAS:



Consultar compra

ID	Producto	Categoría	Subcategoría	Precio	Cantidad
1	compra1	Producto	Sobremesa	12.00	1
3	ASUS	Producto	Sobremesa	10.00	12
4	hahs	Producto	Sobremesa	1.00	1
11	nuevacompra	Producto	Sobremesa	11.00	1
13	12	Producto	Sobremesa	12.00	12

TOTAL : 46.00

USUARIOS:

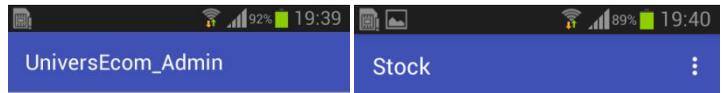


Usuarios

Usuario 0 **admin** - Nombre **admin** - Apellido **admin** - Correo **admin@admin.es**
 Usuario 1 **facon** - Nombre **Facon** - Apellido **News** - Correo **faconnews@gmail.com**
 Usuario 2 **carles** - Nombre **Carles** - Apellido **Palau** - Correo **palau.carles@gmail.com**
 Usuario 3 **user** - Nombre **user** - Apellido **user** - Correo **user@user.com**

TOTAL : 4

A4.2 App



Login



User

Password

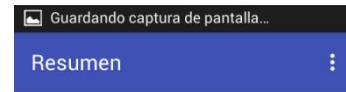
LOGIN CANCEL

Actualmente dispones de estos productos en tu almacén:

ID producto: 118
 Producto: Kingston 1GB
 Categoría: Recambio
 Subcategoría: Memoria
 Precio: 15.00
 Cantidad: 1

ID producto: 119
 Producto: Kingston 4GB DDR3
 Categoría: Recambio
 Subcategoría: Memoria
 Precio: 35.00
 Cantidad: 1

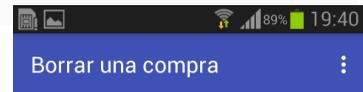
ID producto: 106
 Producto: Portátil Acer 15,6
 Categoría: Producto
 Subcategoría: Portátil
 Precio: 430.00
 Cantidad: 1



Número de usuarios registrados en la web: 4



Ir a la web de UniversEcom



Ingrese el ID de la compra que quiere eliminar:

ID

 BORRAR COMPRA