



Diseño e Implementación de una Tienda Electrónica de Ropa

Emilio Domínguez Sánchez
Máster de Ingeniería Informática
Desarrollo de aplicaciones web

Ignasi Lorente Puchades
César Pablo Córcoles Briongos

07/01/2019



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Diseño e Implementación de una Tienda Electrónica de Ropa</i>
Nombre del autor:	<i>Emilio Domínguez Sánchez</i>
Nombre del consultor/a:	<i>Ignasi Lorente Puchades</i>
Nombre del PRA:	<i>César Pablo Córcoles Briongos</i>
Fecha de entrega (mm/aaaa):	01/2019
Titulación::	<i>Máster de Ingeniería Informática</i>
Área del Trabajo Final:	<i>TFM – Desarrollo de aplicaciones web</i>
Idioma del trabajo:	<i>Español.</i>
Palabras clave	<i>Tienda Ropa Java</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>Este proyecto consiste en la creación de una aplicación para vender productos de ropa a través de Internet. Se basa en la necesidad de una pequeña tienda de ropa en poder poner sus productos en Internet, al no ser desarrollada para una tienda en concreto esta aplicación se podrá usar en distintas tiendas.</p> <p>Para su desarrollo se ha seguido la metodología del proceso unificado de desarrollo de software dirigido a través de casos de uso y centrado en la arquitectura y es iterativo e incremental.</p> <p>En este proyecto se ha obtenido una aplicación totalmente funcional y disponible para poder vender a diferentes tiendas de ropa, que con algunos sencillos pasos se podrá adaptar fácilmente a las necesidades del cliente.</p> <p>A través de este proyecto se han aplicado los conocimientos adquiridos en este Máster de Ingeniería Informática como pueden ser las de un ingeniero de software, gestionando el proceso de desarrollo de aplicaciones para garantizar la calidad, usabilidad y satisfacción de los usuarios o las de un arquitecto de software diseñando aplicaciones y evaluando su rendimiento.</p>	
<p>Abstract (in English, 250 words or less):</p>	
<p>This project consists in the creation of an application to sell clothing products through the Internet. It's based on the need for a small clothing store to be able</p>	

to put their products on the Internet, since it's not developed for a specific store, this application can be used in different stores.

For its development the methodology of the unified process of software development guided through cases of use and focused on architecture has been followed and it's iterative and incremental.

In this project we have obtained a full functional and available application to be able to sell to different clothing stores, which with some simple steps can be easily adapted to the needs of the client.

Through this project, the knowledge acquired in this Master's Degree in Computer Engineering has been applied, such as those of a software engineer, managing the application development process to guarantee the quality, usability and satisfaction of the users or those of an architect. of software designing applications and evaluating their performance.

Índice

1. Introducción.....	1
1.1. Contexto y justificación del Trabajo	1
1.2. Objetivos del Trabajo	2
1.3. Enfoque y método seguido	3
1.4. Integración continua	4
1.5. Planificación del Trabajo.....	5
1.6. Breve resumen de productos obtenidos	8
1.7. Breve descripción de los otros capítulos de la memoria	8
2. Diseño de la aplicación	9
2.1. Diagramas UML	9
2.1.1. Diagramas de Casos de Uso.....	9
2.2. Arquitectura	32
2.2.1. Patrones arquitectónicos.....	32
2.2.2. Patrón de diseño.....	32
2.2.3. Arquitectura general.....	34
2.3. Prototipo del proyecto.....	34
3. Desarrollo de la aplicación.....	40
3.1.1. Implementación de la capa de persistencia	40
3.1.2. Implementación lógica de negocio	41
Descripción de los servicios	41
3.1.3. Implementación capa web	44
3.1.4. Test.....	45
Pruebas unitarias	45
Pruebas de rendimiento.....	47
3.1.5. Versiones de la aplicación/servicio.....	53
3.1.6. Bugs.....	54
3.1.7. Futuras líneas de trabajo	54
4. Conclusiones.	55
4.1. Objetivos marcados	55
4.2. Experiencia obtenida	56
5. Bibliografía	57
6. Anexos	58
6.1. Instalación.....	58
6.2. Manual de usuario.....	59
6.3. Manual del usuario administrador	66

1. Introducción

1.1. Contexto y justificación del Trabajo

En general, por comercio electrónico se entiende toda compra realizada a través de Internet, cualquiera que sea el método de pago utilizado. La característica básica del comercio electrónico se basa en la orden de compraventa, la cual tiene que realizarse a través de algún medio electrónico. La facturación del comercio electrónico en España durante el segundo trimestre del pasado año aumentó un 23,4% respecto al mismo periodo de 2016, hasta alcanzar una cifra de negocio total superior a 7.300 millones de euros, según los datos recogidos por el portal de la Comisión Nacional de los Mercados y la Competencia (CNMC).

Una gran parte de este crecimiento se explica por el incremento del 14,4% de las ventas procedentes de las agencias de viajes y operadores turísticos, por el avance de casi el 12% de las aerolíneas y por el aumento del 5,4% de las ventas 'online' de ropa. En este último sector es dónde nos vamos a fijar.

El perfil de la compradora online es una mujer entre los 25 y los 44 años de edad. Ellas realizaron el 60% de las compras de este sector que, en nuestro país, ya sólo se ve superado por gadgets; paquetes turísticos, billetes y viajes; y entradas para espectáculos.

Las grandes firmas como Mango, Zara, H&M, Desigual o Privalia, ya han desarrollado sus propias aplicaciones para facilitar las compras a través de dispositivos móviles, lo que consigue aumentar las ventas. Según el estudio de GBS Finanzas, que ha elegido a las 50 firmas de ropa, calzado y complementos con presencia online en España, más del 20% de los usuarios de *e-commerce* de moda eligen el *smartphone* para realizar sus compras a través de Internet.

El objetivo de este proyecto consiste en crear una Aplicación Web que sea capaz de acercar los productos de una tienda real a los usuarios de Internet, causando así un doble efecto, que será: por una parte, dar a conocer dicha tienda en todo el mundo, así como sus productos, y por otra parte, dar la posibilidad al cliente de comprar el producto a través de Internet, sin necesidad de personarse en el lugar de venta. La aplicación permitirá la realización de consultas a cualquier usuario, pero sólo los usuarios registrados podrán cerrar el proceso de compra.

Una ventaja adicional de esta aplicación es que, al no haber sido diseñada para una tienda en concreto, es posible utilizarla como base para diferentes tiendas, simplemente modificando estilos, imágenes y algunos textos.

Aunque es verdad que existen otros frameworks o programas para crear tiendas online, como puede ser PrestaShop o WooCommerce, en nuestra

solución damos una opción mucho más adaptable a una pequeña empresa que no una solución tan genérica como pueden ser los frameworks ya desarrollados. Además con esta implementación se le está dando al dueño de la tienda un desarrollo casi a medida que a pequeñas empresas si se lo encargan a una consultora les pudiese salir mucho más caro y la mayoría de las veces inaccesible. Además un desarrollo a medida nos dará una solución con una experiencia de compra completamente adaptada al producto. Eso significa, por supuesto, que la tienda estará totalmente personalizada con la imagen de marca y no será un clon mas del servicio de plantillas que pueda ocasionar una solución como Prestashop o WooCommerce.

Además, son soluciones con garantía de futuro, de manera que si se precisa de futuras implementaciones será mucho más sencillo que acudir a módulos prediseñados.

También es importante la ventaja que se obtiene en cuanto al posicionamiento orgánico (SEO) así como en marketing de búsqueda directa (SEM).

Por último comentar que con este desarrollo la aplicación se puede adaptar a lo que nos pida el cliente (o dueño de la tienda) de forma muy sencilla y por consiguiente a un precio muy razonable para ellos.

1.2. Objetivos del Trabajo

Teniendo en cuenta la aplicación a realizar, se marcan los siguientes objetivos concretos para así conseguir una aplicación totalmente funcional:

1. Visualizar los productos de una tienda de ropa, por cada producto se mostrará su nombre, descripción, marca, precio y su foto.
2. Permitir categorizar los productos.
3. Permitir realizar búsquedas a través de palabras clave de la ropa ofertada.
4. Carrito de la compra en el que el usuario añadirá los productos que quiera comprar.
5. Registro y logueo de usuarios en la aplicación para poder realizar sus pedidos.
6. Usuario especial administrador para gestionar el contenido de la tienda.
7. Blog propio dentro de la tienda para comunicarse mejor con los compradores.
8. Envío automático de correos electrónicos durante el proceso de compra.
9. Posibilidad de compartir productos en las redes sociales
10. Añadir comentarios o dudas sobre los productos.
11. Sistema de fidelización propio basado en puntos que podrán canjearse por descuentos en futuras compras.
12. Sistema de auto-recomendaciones de los productos de la tienda en base a compras pasadas de otros usuarios.
13. Envío de facturas al correo electrónico. Facturas en PDF.
14. Diseño amigable de toda la aplicación.
15. Publicación de la aplicación en un servidor Cloud.

16. Puesta en funcionamiento de un servidor de integración continua en un servidor Cloud.

1.3. Enfoque y método seguido

Para llevar a cabo el presente proyecto se ha utilizado como metodología el **Proceso Unificado de Desarrollo Software**. El Proceso Unificado de Desarrollo Software o simplemente Proceso Unificado, es un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental. El refinamiento más conocido y documentado del Proceso Unificado es el Proceso Unificado de Rational o simplemente RUP.

El Proceso Unificado no es simplemente un proceso, sino un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. De la misma forma, el Proceso Unificado de Rational, también es un marco de trabajo extensible, por lo que muchas veces resulta imposible decir si un refinamiento particular del proceso ha sido derivado del Proceso Unificado o del RUP. Por dicho motivo, los dos nombres suelen utilizarse para referirse a un mismo concepto.

El proceso unificado divide el trabajo de desarrollo de software en cuatro fases:

1. Fase de Inicio

En esta fase corresponde definir el negocio. Es la etapa donde se define la factibilidad del proyecto a realizar, se representa el modelo de negocio, visión y metas del proyecto, se identifican actores, conceptos de dominio y requisitos de usuario. Adicionalmente se complementa con la definición de la arquitectura preliminar, y estimaciones (imprecisas, preliminares) de plazos y costos. También se define la viabilidad del proyecto.

2. Fase de Elaboración

En la fase de elaboración se obtiene la visión refinada del proyecto a realizar, la implementación iterativa del núcleo central de la aplicación, la resolución de los riesgos más altos, la identificación de nuevos requisitos y nuevos alcances, y estimaciones más ajustadas. A esta altura existe la posibilidad de detener el proyecto por complejidad técnica.

3. Fase de Construcción

La fase de construcción es la implementación iterativa del resto de los requisitos de menor riesgo y elementos más sencillos. Es la evolución hasta convertirse en un producto listo, incluyendo todos los requisitos (100%), para entregarse al Cliente. Al final de esta fase el sistema contiene todos los casos de uso que el cliente y la dirección del proyecto han acordado. La mayoría de los casos de uso que no se desarrollaron en la fase anterior se desarrollan en iteraciones, en grupos de requisitos o casos de uso durante esta fase.

4. Fase de Transición

Es el periodo donde el producto es completamente entregado al cliente para ser probado y desplegado (instalado).

Algunas características a destacar según el proceso unificado son:

- a) Los proyectos se organizarán en una serie de mini-proyectos de corta duración (2 a 6 semanas), llamados iteraciones, que incluyen un conjunto reducido de requerimientos a implementar.
- b) El resultado de cada iteración es un sistema que puede ser probado, integrado y ejecutado.
- c) Rápida retroalimentación y asimilación de los cambios, posibilitada por el tamaño limitado de lo realizado en cada iteración.
- d) Se abordan, resuelven y prueban primeramente las decisiones de diseño críticas o de alto riesgo.
- e) Si no se logra cumplir lo previsto dentro del plazo estipulado, se aconseja transferir tareas o requisitos para una iteración posterior, pero no modificar la fecha de entrega de la iteración actual. Por lo tanto, el proceso iterativo permite una comprensión creciente de los requerimientos a la vez que se va haciendo crecer el sistema. Con esto se logra reducir los riesgos del proyecto y tener un subsistema ejecutable lo antes posible.

1.4. Integración continua

Esta aplicación se integrará en un servidor de integración continua.

La integración continua es una práctica habitual en el mundo del desarrollo actual, usada principalmente por equipos de desarrollo que han adoptado metodologías ágiles de trabajo. Básicamente, consiste en un proceso por el cual cualquier pequeña mejora en el software se integra rápidamente con el software que se llevará a producción.

El software usado para desarrollar este proceso será:

Jenkins. Es un software que se usa como un sistema de automatización, capaz de realizar cientos de tareas necesarias para asegurar la calidad del software y facilitar su despliegue o construcción, alguna de las tareas más habituales que se realizan con Jenkins son probar el software, revisar las métricas de calidad del software, despliegue y compilación del software.

En Jenkins usamos los plugins de maven, javadoc y junit para generar diferente material que será muy útil para ver el estado del proyecto y su evolución.

Git y GitHub. Para la integración continua se usa Git que es un software de control de versiones gratuito. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos. A su vez git hub es una plataforma de desarrollo colaborativo que permite alojar proyectos usando el control de versiones Git. Típicamente los proyectos que se alojan en GitHub se alojan de forma pública (como en este caso) pero también tiene versiones de pago para alojar proyectos de forma privada. Se usa este servicio por ser gratuito y por su integración con jenkins.

1.5. Planificación del Trabajo

Antes de comenzar el proyecto, se ha determinado el número de iteraciones en base a las funcionalidades que se deben implementar. En este proyecto van a ser las siguientes cuatro iteraciones:

- **Iteración 1**

Es la iteración más complicada que nos debe ofrecer la base para el resto de las iteraciones. Para ello se seleccionaron las funcionalidades que mejor podían ayudar a dar soporte a las demás. Se condifica, prueba y determina las funciones de registro y autenticación de usuario, así como muchos de los casos de uso para el administrador, ya que muchas funcionalidades dependen de la creación de datos, especialmente nuevos productos (y modificación de estos), categorías etc.

Iteración 2

En esta iteración se realizan las funcionalidades básicas para visualizar los productos de la tienda por parte del usuario así como el visionado según categorías y las opciones de búsqueda de la tienda.

- **Iteración 3**

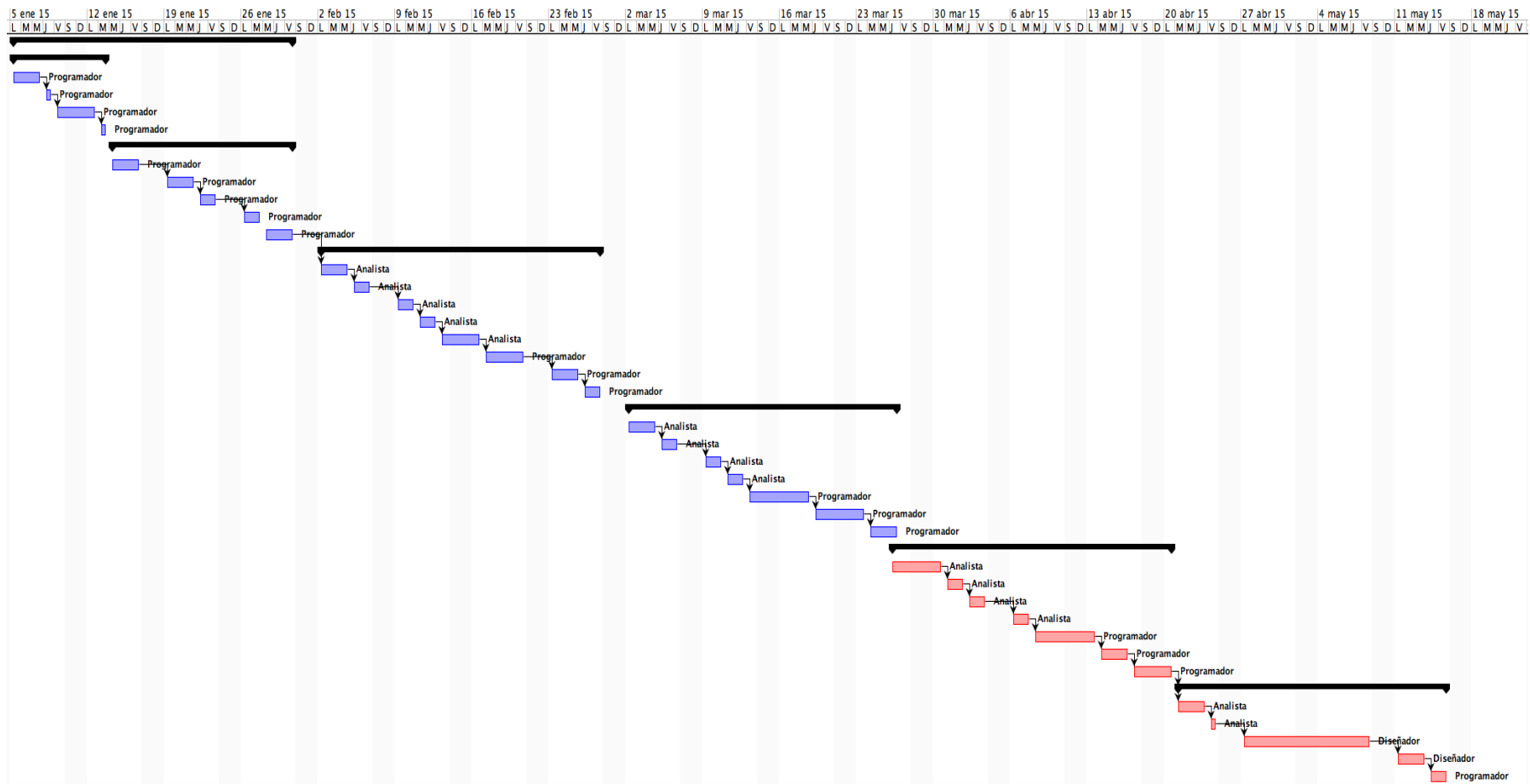
En esta iteración se realiza el proceso de compra completo por parte del usuario, además de añadir las funcionalidades extras de la tienda, comentarios, recomendaciones, etc.

- **Iteración 4**

En esta iteración se realiza la interfaz gráfica de toda la aplicación además se abordarán las funcionalidades que quedan pendientes y las pruebas de rendimiento de la aplicación.

A continuación se muestra un diagrama de Gant con la planificación inicial de lo que llevará la realización de este proyecto. En el se han tenido en cuenta 3 tipos diferentes de roles que realizarán los diferentes aspectos de la aplicación, Programador (encargado de realizar el código de la aplicación), Analista (encargado de analizar los diferentes paso) y Diseñador (encargado de la parte gráfica de la aplicación).

	🕒	Nombre	Duración	Inicio	Terminado	Predecesores	Nombres del Recurso
1		Estudio de las tecnologías a usar	20 days?	5/01/15 8:00	30/01/15 17:00		
2	👤	Revisión de lenguajes ya conocidos	7 days?	5/01/15 8:00	13/01/15 17:00		Programador
3		Java	3 days?	5/01/15 8:00	7/01/15 17:00		Programador
4		Html	1 day?	8/01/15 8:00	8/01/15 17:00	3	Programador
5		Css	2 days?	9/01/15 8:00	12/01/15 17:00	4	Programador
6		Javascript	1 day?	13/01/15 8:00	13/01/15 17:00	5	Programador
7		Estudio de librerías nuevas	13 days?	14/01/15 8:00	30/01/15 17:00		
8		Tapstry	3 days?	14/01/15 8:00	16/01/15 17:00		Programador
9		Hibernate	3 days?	19/01/15 8:00	21/01/15 17:00	8	Programador
10		Spring	2 days?	22/01/15 8:00	23/01/15 17:00	9	Programador
11		Maven	2 days?	26/01/15 8:00	27/01/15 17:00	10	Programador
12		Otras herramientas o librerías	3 days?	28/01/15 8:00	30/01/15 17:00		Programador
13		Iteración 1	20 days?	2/02/15 8:00	27/02/15 17:00		
14		Análisis	3 days?	2/02/15 8:00	4/02/15 17:00	12	Analista
15		Estudio del sistema a desarrollar	2 days?	5/02/15 8:00	6/02/15 17:00	14	Analista
16		Especificación de requisitos	2 days?	9/02/15 8:00	10/02/15 17:00	15	Analista
17		Estudio de mini-aplicación base	2 days?	11/02/15 8:00	12/02/15 17:00	16	Analista
18		Diseño (modelo, vista y controlador)	2 days?	13/02/15 8:00	16/02/15 17:00	17	Analista
19		Implementación	4 days?	17/02/15 8:00	20/02/15 17:00	18	Programador
20		Pruebas	3 days?	23/02/15 8:00	25/02/15 17:00	19	Programador
21		Revisión de toda la iteración	2 days?	26/02/15 8:00	27/02/15 17:00	20	Programador
22		Iteración 2	19 days?	2/03/15 8:00	26/03/15 17:00		
23		Análisis	3 days?	2/03/15 8:00	4/03/15 17:00		Analista
24		Estudio del sistema a desarrollar	2 days?	5/03/15 8:00	6/03/15 17:00	23	Analista
25		Especificación de requisitos	2 days?	9/03/15 8:00	10/03/15 17:00	24	Analista
26		Diseño (modelo, vista y controlador)	2 days?	11/03/15 8:00	12/03/15 17:00	25	Analista
27		Implementación	4 days?	13/03/15 8:00	18/03/15 17:00	26	Programador
28		Pruebas	3 days?	19/03/15 8:00	23/03/15 17:00	27	Programador
29		Revisión de toda la iteración	3 days?	24/03/15 8:00	26/03/15 17:00	28	Programador
30		Iteración 3	18 days?	26/03/15 8:00	20/04/15 17:00		
31		Análisis	3 days?	26/03/15 8:00	30/03/15 17:00		Analista
32		Estudio del sistema a desarrollar	2 days?	31/03/15 8:00	1/04/15 17:00	31	Analista
33		Especificación de requisitos	2 days?	2/04/15 8:00	3/04/15 17:00	32	Analista
34		Diseño (modelo, vista y controlador)	2 days?	6/04/15 8:00	7/04/15 17:00	33	Analista
35		Implementación	4 days?	8/04/15 8:00	13/04/15 17:00	34	Programador
36		Pruebas	3 days?	14/04/15 8:00	16/04/15 17:00	35	Programador
37		Revisión de toda la iteración	2 days?	17/04/15 8:00	20/04/15 17:00	36	Programador
38		Iteración 4	19 days?	21/04/15 8:00	15/05/15 17:00	37	
39		Análisis	3 days?	21/04/15 8:00	23/04/15 17:00	37	Analista
40		Diseño gráfico de la interfaz	1 day?	24/04/15 8:00	24/04/15 17:00	39	Analista
41		Implementación	10 days?	27/04/15 8:00	8/05/15 17:00	40	Diseñador
42		Pruebas	3 days?	11/05/15 8:00	13/05/15 17:00	41	Diseñador
43		Revisión de toda la iteración	2 days?	14/05/15 8:00	15/05/15 17:00	42	Programador



1.6. Breve resumen de productos obtenidos

No hay que entrar en detalle: la descripción detallada se hará en el resto de capítulos.

Este capítulo resume los productos que se han generado durante la realización de este proyecto.

- Memoria del proyecto. Este documento en el cual se resumirá las actividades más relevantes del proyecto, sus objetivos, su diseño, su planificación y las conclusiones finales
- Bases de datos del proyecto. Como este proyecto necesita de una base de datos, se ha obtenido las consultas para crearla y arrancarla.
- Código fuente del proyecto. El propio proyecto en sí, archivos java, xml, tml, etc. Que componen el proyecto, para poder modificarlo y mejorarlo.
- Documento de instalación. Documento que indica como instalar y ejecutar la aplicación entregada.
- War de la aplicación. Ejecutable de la aplicación.

1.7. Breve descripción de los otros capítulos de la memoria

La estructura que seguirá el proyecto se basa en 6 capítulos más. Diseño, desarrollo, conclusiones, bibliografía, glosario y anexos.

En el capítulo de diseño se hará énfasis en la estructura de la aplicación, los casos de uso esperados, patrones y arquitectura de la aplicación.

En el capítulo de desarrollo se especificará como se ha implementado el diseño explicado en el punto anterior.

En las conclusiones se comentará el proceso de creación del proyecto, lo conseguido con la realización de este proyecto en comparación con los objetivos marcados en un principio y lo que se podrá mejorar en un futuro además de lo aprendido durante su realización y el grado de satisfacción alcanzado con el proyecto.

En la bibliografía, glosario y anexos se comentará la documentación consultada para realizar este proyecto, los acrónimos o palabras más usadas y los anexos generados con este proyecto

2. Diseño de la aplicación

En esta sección se describirá el diseño en profundidad de la estructura interna de la aplicación con los patrones utilizados, haciendo especial hincapié en el patrón MVC y sus capas modelo, vista y controlador.

El diseño junto con la implementación constituye la fase en la que a partir de la especificación de sistema se obtiene un sistema ejecutable. En concreto el diseño permite obtener la estructura de un software que realiza la especificación.

Para realizar esta tarea, se elabora un documento en el cual se señala lo que el usuario necesita del sistema software. Dicho documento, es considerado muchas veces como un contrato entre el cliente y el equipo de desarrollo del sistema.

Actualmente, el desarrollo de software orientado a objetos y el uso de UML se han incrementado. Es por ese motivo que el empleo de casos de uso se está imponiendo frente a otras técnicas de especificación de requisitos.

2.1. Diagramas UML

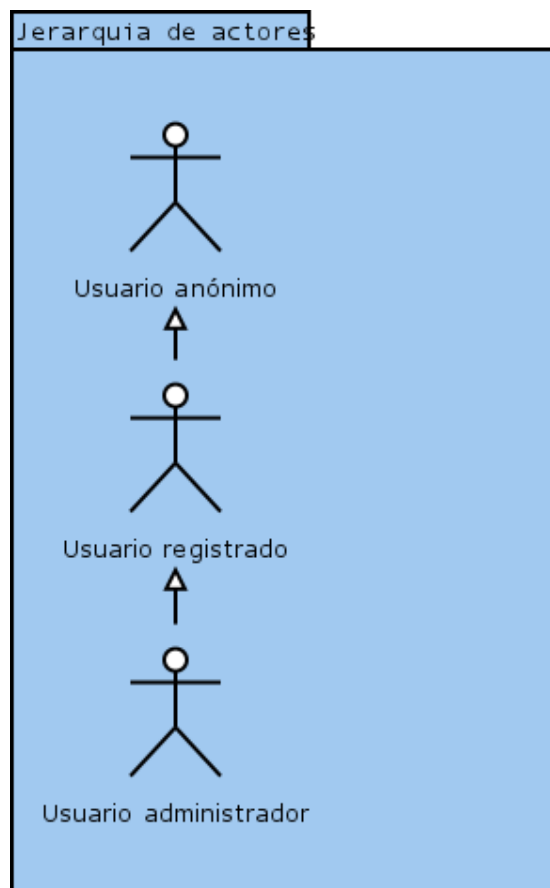
2.1.1. Diagramas de Casos de Uso.

En nuestro sistema se pueden identificar tres tipos de actores con distintos privilegios y opciones.

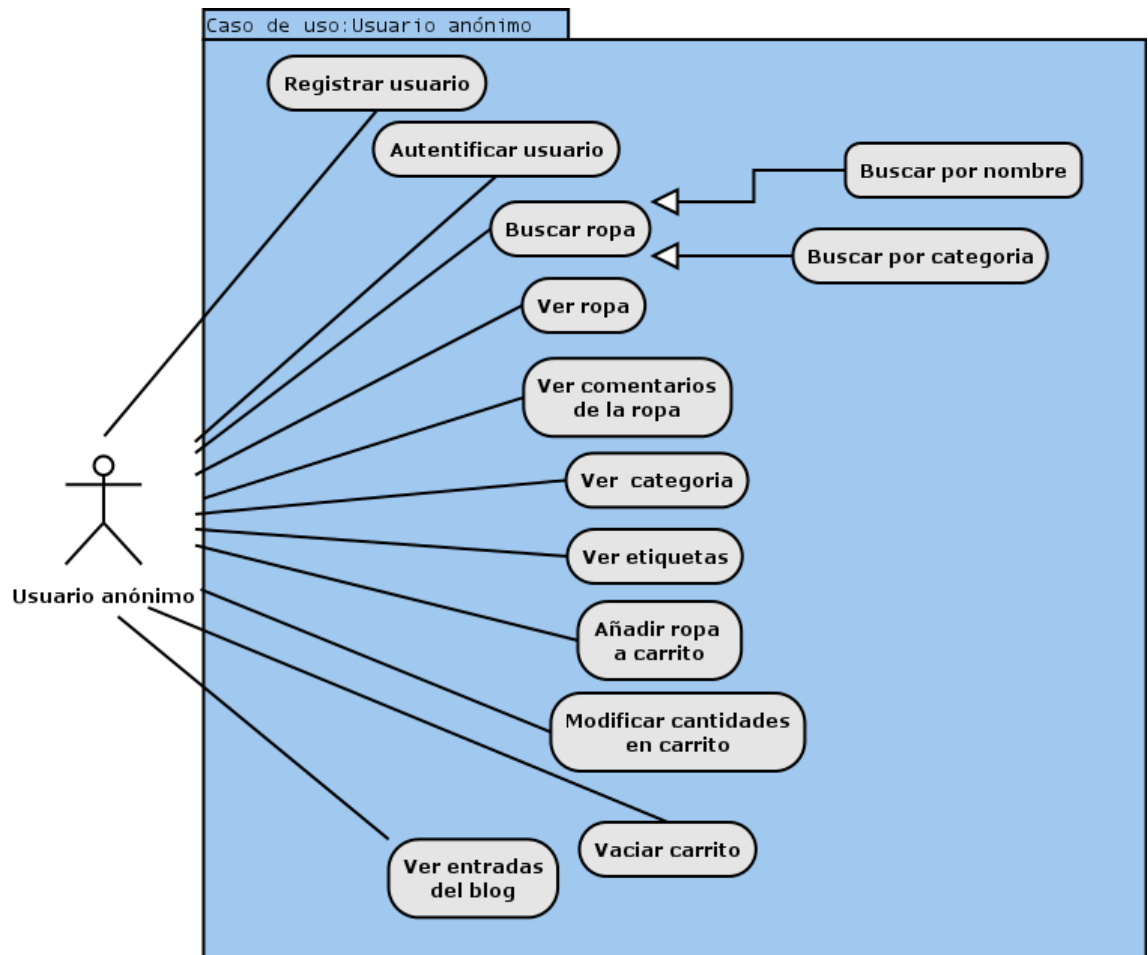
- **Usuario no autenticado o anónimo:** Es todo usuario que accede a la aplicación y que no se ha autenticado. Tendrá acceso a las opciones de búsqueda y a la visualización de la ropa, además podrá ver los comentarios de otros usuarios acerca de cada producto, también podrá añadir y modificar productos del carrito, aunque no podrá realizar la compra. Lógicamente también podrá registrarse.
- **Usuario autenticado:** Es todo usuario que accede a la aplicación y que está registrado y se ha autenticado satisfactoriamente en el sistema. Podrán modificar los datos de su perfil, realizar la compra de los productos que se hayan añadido al carrito, ver los datos de las compras realizadas y comentar acerca de los productos de la tienda.

- **Administrador:** Es un usuario especial, cuya función será la de gestionar la aplicación. Podrá crear, modificar y borrar los datos almacenados en la base de datos, lo que le permitirá un total control gestionando las categorías, las etiquetas, las entradas del blog, mandar correos a los usuarios, crear y modificar la ropa que le llega, así como asignarle un stock. Además podrá gestionar los pedidos que los usuarios le vayan solicitando.

Cada usuario hereda las funcionalidades del usuario anterior, dando lugar a una “Jerarquía de Actores” como se puede ver en la siguiente figura. En la que el usuario no autenticado o anónimo es el usuario básico. A partir de ahí, el usuario autenticado, puede realizar las acciones del usuario anónimo más otras especiales para él. Lo mismo ocurre con el administrador, que podrá realizar las acciones del usuario autenticado más todas las reservadas para este tipo de usuario.



A continuación se muestran los casos de uso para cada tipo de usuario.



Para cada caso de uso del **usuario anónimo** se comentará una breve descripción de él, sus precondiciones (si las tiene), su flujo normal para que el usuario ejecute dicho caso de uso, flujo alternativo (si aplica) y las postcondiciones del caso de uso comentado.

Caso de uso : Registrar usuario	
DESCRIPCIÓN	La aplicación permite a un visitante registrarse en la aplicación.
ACTORES	Usuario no autenticado
PRECONDICIONES	
FLUJO NORMAL	<ol style="list-style-type: none"> 1. El usuario pulsa en el enlace de autenticarse. 2. Se pulsa en crear cuenta. 3. Se muestra el formulario de registro que el usuario deberá rellenar.

	4. Se pulsa registrar para proceder con el registro.
FLUJO ALTERNATIVO	<ol style="list-style-type: none"> 1. El usuario crea un carrito. 2. El usuario accede al carrito a través el botón carrito. 3. Se pulsa registrarse. 4. Se muestra el formulario de registro que el usuario deberá rellenar. 5. Pulsa registrar.
POSTCONDICIONES	El usuario queda registrado en la aplicación, con un login y contraseña que puede usar para loguearse a la aplicación.

Caso de uso : Autenticar usuario	
DESCRIPCIÓN	Con esta funcionalidad un usuario (con los datos de registro) podrá introducir su nombre de usuario y contraseña, para autenticarse en la aplicación.
ACTORES	Usuario no autenticado
PRECONDICIONES	--
FLUJO NORMAL	<ol style="list-style-type: none"> 1. El usuario pulsa en el enlace de autenticarse. 2. Se rellena los campos de usuario y contraseña. 3. Pulsa entrar.
FLUJO ALTERNATIVO	<ol style="list-style-type: none"> 1. El usuario crea un carrito 2. El usuario accede al carrito a través el botón carrito. 3. Pulsa log-in. 4. Se rellena los campos de usuario y contraseña. 5. Pulsa Entrar.
POSTCONDICIONES	El usuario está logueado en la aplicación, tiene acceso a las pantallas de Actualizar info registro y mis pedidos, así como a Salir para desloguearse.

Caso de uso : Buscar ropa por nombre	
DESCRIPCIÓN	Introduciendo en el buscador de la aplicación, (disponible en la parte superior de todas las páginas) parte del nombre o de la descripción de un artículo de ropa, los usuarios obtendrán la lista de artículos de ropa que coinciden con su búsqueda.
ACTORES	Usuario no autenticado
PRECONDICIONES	--
FLUJO NORMAL	<ol style="list-style-type: none"> 1. El usuario escribe lo que quiere buscar en el buscador de la parte superior de una página. 2. Pulsa entrar.
FLUJO ALTERNATIVO	
POSTCONDICIONES	<p>Se le muestra al usuario un listado con los resultados de la búsqueda.</p> <p>Si no encuentra ningún resultado, se muestra lo siguiente "No se ha encontrado nada relacionado con lo que quiere buscar".</p>

Caso de uso : Buscar ropa por categoría	
DESCRIPCIÓN	Con esta funcionalidad el usuario podrá elegir entre dos formas de buscar, por categoría navegando a través del menú y seleccionando la que quiere consultar, o mediante el buscador, introduciendo un conjunto de palabras claves y seleccionando la categoría por la que quiere filtrar.
ACTORES	Usuario no autenticado
PRECONDICIONES	--
FLUJO NORMAL	<ol style="list-style-type: none"> 1. El usuario navega a través del menú de la aplicación y selecciona la categoría que desea ver.

FLUJO ALTERNATIVO	1. El usuario a través de la página de búsqueda, escribe lo que quiere buscar y en el filtro de categoría filtra por la categoría que desee.
POSTCONDICIONES	Se le muestra al usuario un listado con los resultados de la búsqueda. Si no encuentra ningún resultado, se muestra lo siguiente "No se ha encontrado nada relacionado con lo que quiere buscar".

Caso de uso : Ver Ropa	
DESCRIPCIÓN	Permite ver los detalles de cada producto de ropa.
ACTORES	Usuario no autenticado
PRECONDICIONES	--
FLUJO NORMAL	1. El usuario selecciona la prenda de ropa que quiere ver.
FLUJO ALTERNATIVO	
POSTCONDICIONES	Se le muestra al usuario la siguiente información de cada producto, el nombre de la prenda, su precio, su marca, su color, su descripción, el número de puntos que contiene (si los tiene) y una imagen del mismo.

Caso de uso : Ver Comentarios de la ropa	
DESCRIPCIÓN	Permite ver los comentarios que los usuarios logueados han realizado sobre los productos de la tienda.
ACTORES	Usuario no autenticado
PRECONDICIONES	Se selecciona una prenda de ropa con comentarios.
FLUJO NORMAL	1. El usuario selecciona la prenda de ropa que se indica en las precondiciones.

FLUJO ALTERNATIVO	
POSTCONDICIONES	Se le muestran al usuario los comentarios de la prenda seleccionada.

Caso de uso : Ver Categorías	
DESCRIPCIÓN	Permite ver y navegar a través de las categorías de la aplicación, viendo la ropa que contienen.
ACTORES	Usuario no autenticado
PRECONDICIONES	
FLUJO NORMAL	1. El usuario a través del menú de la aplicación selecciona una categoría
FLUJO ALTERNATIVO	
POSTCONDICIONES	Se le muestran al usuario todas las prendas de ropa categorizadas con esa categoría.

Caso de uso : Ver Etiquetas	
DESCRIPCIÓN	Permite ver y navegar a través de las etiquetas de la aplicación, viendo la ropa que contienen.
ACTORES	Usuario no autenticado
PRECONDICIONES	
FLUJO NORMAL	1. El usuario a través del menú de la aplicación selecciona una etiqueta.
FLUJO ALTERNATIVO	
POSTCONDICIONES	Se le muestran al usuario todas las prendas de ropa que contiene esa etiqueta.

Caso de uso : Añadir ropa a carrito

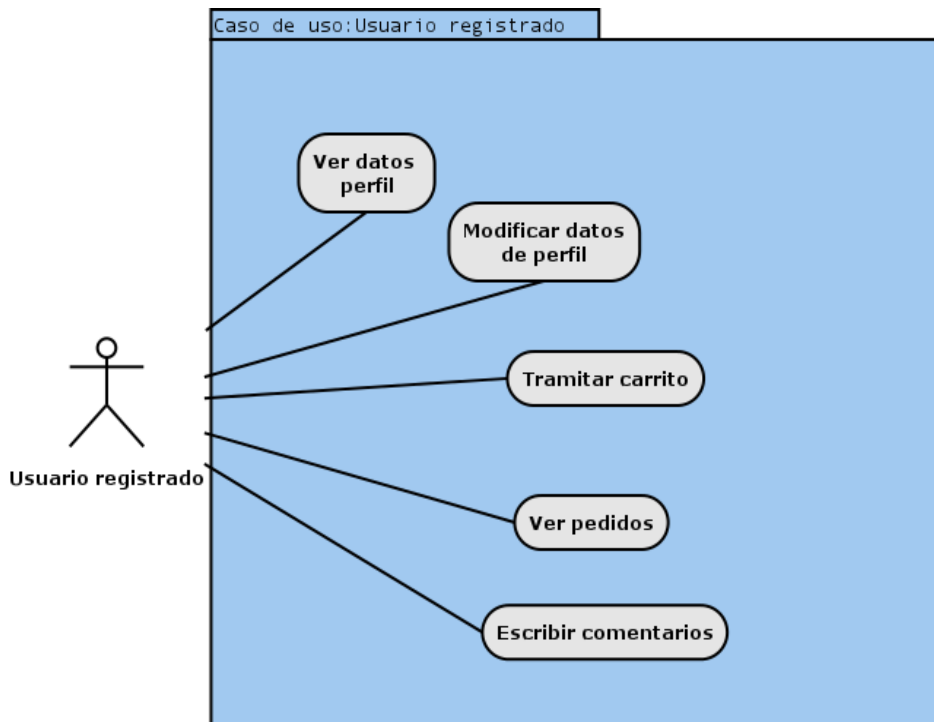
DESCRIPCIÓN	Permite a un usuario no autenticado agregar productos a su carrito (no persistente).
ACTORES	Usuario no autenticado
PRECONDICIONES	
FLUJO NORMAL	<ol style="list-style-type: none"> 1. El usuario selecciona una prenda que quiera añadir al carrito, así como su talla. 2. Pulsa en añadir a carrito.
FLUJO ALTERNATIVO	
POSTCONDICIONES	El producto se añade al carrito.

Caso de uso : Modificar cantidades en carrito	
DESCRIPCIÓN	Permite a un usuario invitado modificar las cantidades de su carrito.
ACTORES	Usuario no autenticado
PRECONDICIONES	Disponer de un carrito con productos
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Pulsar en Carrito 2. Modificar la cantidad seleccionando un número mayor que 0.
FLUJO ALTERNATIVO	<ol style="list-style-type: none"> 1. Pulsar en Carrito 2. Modificar la cantidad seleccionando 0.
POSTCONDICIONES	<p>Si se selecciona un número mayor que 0, la cantidad de unidades pedidas de esa unidad pasa a ser esa.</p> <p>Si se selecciona 0, se borra dicho producto del carrito.</p> <p>Si se selecciona un número menor que 0, no se actualiza la cantidad.</p>

Caso de uso : Vaciar carrito

DESCRIPCIÓN	Permite a un usuario invitado borrar todo su carrito.
ACTORES	Usuario no autenticado
PRECONDICIONES	Disponer de un carrito con productos.
FLUJO NORMAL	1. Pulsar en Vaciar Carrito
FLUJO ALTERNATIVO	1. Pulsar en Carrito 2. Modificar todas las cantidad seleccionando 0 en su cantidad.
POSTCONDICIONES	Se borra el carrito y todos los productos que el contenía.

Caso de uso : Ver entradas del blog	
DESCRIPCIÓN	Permite a un usuario invitado ver las entradas del blog de la aplicación.
ACTORES	Usuario no autenticado
PRECONDICIONES	
FLUJO NORMAL	1. Pulsar en blog en el menú de la aplicación. 2. Seleccionar la entrada que se quiere ver.
FLUJO ALTERNATIVO	1. Pulsar sobre la entrada que se quiere ver en la parte inferior de la aplicación.
POSTCONDICIONES	Se muestra todo el contenido de la entrada del blog.



Como en el apartado anterior, para cada caso de uso del usuario autenticado también se comentará una breve descripción de el, sus precondiciones (si las tiene), su flujo normal para que el usuario ejecute dicho caso de uso, flujo alternativo (si aplica) y las postcondiciones del caso de uso comentado.

Caso de uso : Ver datos del perfil	
DESCRIPCIÓN	Permite a un usuario registrado ver la información de su perfil
ACTORES	Usuario registrado
PRECONDICIONES	
FLUJO NORMAL	1. Pulsar en actualizar info registro.
FLUJO ALTERNATIVO	
POSTCONDICIONES	Se muestra la información con la que el usuario se registrón la aplicación.

Caso de uso : Modificar datos del perfil	
DESCRIPCIÓN	Permite a un usuario registrado modificar los datos de su perfil

ACTORES	Usuario registrado
PRECONDICIONES	
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Pulsar en actualizar info registro. 2. Cambiar los datos que se desean modificar. 3. Pulsar en registrar.
FLUJO ALTERNATIVO	
POSTCONDICIONES	Se actualiza correctamente la información que se cambió.

Caso de uso : Tramitar carrito	
DESCRIPCIÓN	Permite a un usuario registrado tramitar un carrito para así convertirlo en un pedido y formalizarlo para que le sea enviado
ACTORES	Usuario registrado
PRECONDICIONES	Poseer un carrito con productos añadidos previamente
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Pulsar en Carrito. 2. Seleccionar que se quiere hacer con los puntos del usuario: <ol style="list-style-type: none"> a. Acumular: Se le acumularán al usuario, los puntos que tienen los productos del carrito. b. Descontar: Se le descontarán los puntos de la cuenta de usuario más los acumulados en el carrito actual, tras descontarlos se mostrará el descuento asociado. 3. Pulsar en tramitar pedido.
FLUJO ALTERNATIVO	
POSTCONDICIONES	Se tramita correctamente el pedido y este pasa a estado creado.

Caso de uso : Escribir comentarios

DESCRIPCIÓN	Permite escribir comentarios sobre las diferentes prendas de la ropa.
ACTORES	Usuario registrado
PRECONDICIONES	
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Entrar en la ropa que queremos comentar. 2. Escribir en el apartado correspondiente el comentario que queremos hacer. 3. Pulsar en enviar.
FLUJO ALTERNATIVO	
POSTCONDICIONES	Se crea el comentario en la prenda indicada, además se muestra correctamente el nombre de usuario con el que se creó el comentario.



Como los apartado anteriores, para cada caso de uso del **usuario administrador** también se comentará una breve descripción de el, sus precondiciones (si las tiene), su flujo normal para que el usuario ejecute dicho caso de uso, flujo alternativo (si aplica) y las postcondiciones del caso de uso comentado.

Caso de uso : Crear ropa	
DESCRIPCIÓN	Esta funcionalidad permite registrar la ropa en la aplicación.

ACTORES	Usuario administrador
PRECONDICIONES	Toda ropa tiene que llevar al menos un stock talla (aunque el stock sea 0) y un adjunto asociado.
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Entrar en el menú de administración. 2. En la sección ropa pulsar sobre “Registrar Ropa” 3. Se abre el formulario con los datos a rellenar para registrar una prenda de ropa. 4. Rellenar todos los datos obligatorios y pulsar en registrar ropa.
FLUJO ALTERNATIVO	
POSTCONDICIONES	Se crea la prenda de ropa con todos los datos introducidos en el formulario, además de en la categoría que seleccionamos y con todas las tallas introducimos.

Caso de uso : Añadir stock	
DESCRIPCIÓN	Esta funcionalidad permite añadir stock a una prenda de ropa en la aplicación.
ACTORES	Usuario administrador
PRECONDICIONES	Disponer de una prenda de ropa
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Al crear una nueva prenda de ropa, seleccionar el número de tallas que dispongamos. 2. Se abren un número determinado de campos según las tallas que dispongamos, rellenar el campo de talla y stock. 3. Registrar la ropa.
FLUJO ALTERNATIVO	<ol style="list-style-type: none"> 1. Entrar en el menú de administración. 2. En la sección ropa pulsar sobre “Actualizar Ropa”. 3. Se abre un listado con todas las prendas de la aplicación, pulsar sobre “Añadir talla” en la prenda que queremos añadir el stock. <p>Escribir la talla y el stock del que dispongamos y</p>

	pulsar "Actualizar".
POSTCONDICIONES	Se crean las tallas que hemos decidido crear

Caso de uso : Modificar ropa	
DESCRIPCIÓN	Esta funcionalidad permite modificar los atributos de una prenda de ropa de la tienda.
ACTORES	Usuario administrador
PRECONDICIONES	Disponer de una prenda de ropa
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Entrar en el menú de administración. 2. En la sección ropa pulsar sobre "Actualizar Ropa". 3. Se abre un listado con todas las prendas de la aplicación, pulsar sobre "Modificar" en la prenda que queremos modificar. 4. Modificar los datos que queramos y pulsar en "Actualizar".
FLUJO ALTERNATIVO	
POSTCONDICIONES	Se modifican los datos que hemos cambiado en la prenda

Caso de uso : Modificar stock	
DESCRIPCIÓN	Esta funcionalidad permite modificar el stock de una prenda de ropa.
ACTORES	Usuario administrador
PRECONDICIONES	Disponer de una prenda de ropa con stock
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Entrar en el menú de administración. 2. En la sección ropa pulsar sobre "Actualizar Ropa". 3. Se abre un listado con todas las prendas de la aplicación, pulsar sobre "Modificar Stock" en la prenda que queremos modificar. 4. Modificar los datos que queramos y pulsar en "Actualizar".

FLUJO ALTERNATIVO	
POSTCONDICIONES	Se modifica tanto el stock como la talla que queramos en la prenda seleccionada.

Caso de uso : Crear categoría	
DESCRIPCIÓN	Esta funcionalidad permite crear categorías en la aplicación.
ACTORES	Usuario administrador
PRECONDICIONES	Solo se podrán crear subcategorías de las categorías padre, hombre o mujer.
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Entrar en el menú de administración. 2. En la sección Categoría pulsar sobre “Crear categoría”. 3. Seleccionar la categoría padre, y rellenar el campo de nombre. 4. Pulsar en “Actualizar”.
FLUJO ALTERNATIVO	
POSTCONDICIONES	Se crea la categoría y dicha categoría aparece en el menú en la categoría padre que se seleccionó.

Caso de uso : Modificar categoría	
DESCRIPCIÓN	Esta funcionalidad permite modificar categorías en la aplicación.
ACTORES	Usuario administrador
PRECONDICIONES	Solo se podrán crear subcategorías de las categorías padre, hombre o mujer.
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Entrar en el menú de administración. 2. En la sección Categoría pulsar sobre “Modificar categoría”. 3. Seleccionar la categoría a modificar. 4. En el formulario modificar la categoría.

	5. Pulsar en "Actualizar".
FLUJO ALTERNATIVO	Existen Categorías en la aplicación
POSTCONDICIONES	Se actualiza la categoría con los datos introducidos

Caso de uso : Crear etiqueta	
DESCRIPCIÓN	Esta funcionalidad permite crear etiquetas en la aplicación.
ACTORES	Usuario administrador
PRECONDICIONES	
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Entrar en el menú de administración. 2. En la sección Etiquetas pulsar sobre "Crear etiqueta" 3. Rellenar el campo de nombre y dejar en blanco el de porcentaje descuento 4. Pulsar en "Actualizar"
FLUJO ALTERNATIVO	<ol style="list-style-type: none"> 1. Entrar en el menú de administración. 2. En la sección Etiqueta pulsar sobre "Crear etiqueta". 3. Rellenar el campo de nombre y escribir el porcentaje de descuento. 4. Pulsar en "Actualizar".
POSTCONDICIONES	Según escribamos o no el porcentaje de descuento se crea la etiqueta en una zona del menú o no. Si lo creamos con descuento se creará la etiqueta en la sección "Descuentos" si no se creará en la sección Colecciones.

Caso de uso : Modificar etiqueta	
DESCRIPCIÓN	Esta funcionalidad permite modificar etiquetas en la aplicación.

ACTORES	Usuario administrador
PRECONDICIONES	Existen etiquetas en la aplicación
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Entrar en el menú de administración. 2. En la sección Etiquetas pulsar sobre “Modificar etiqueta”. 3. En el formulario modificar la etiqueta. 4. Pulsar en “Actualizar”.
FLUJO ALTERNATIVO	
POSTCONDICIONES	Se modifica los datos de la etiqueta con los nuevos introducidos en el formulario.

Caso de uso : Crear entrada del blog	
DESCRIPCIÓN	Esta funcionalidad permite crear entradas en el blog de la tienda.
ACTORES	Usuario administrador
PRECONDICIONES	
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Entrar en el menú de administración. 2. En la sección Blog pulsar sobre “Crear entrada”. 3. En el formulario introducir todos los datos solicitados. 4. Pulsar en “Crear”.
FLUJO ALTERNATIVO	
POSTCONDICIONES	Se crea una nueva entrada en el blog. Se puede acceder a esta nueva entrada tanto en la sección de inferior de la aplicación como el propio blog.

Caso de uso : Modificar entrada del blog	
DESCRIPCIÓN	Esta funcionalidad permite modificar entradas en el blog de la tienda.

ACTORES	Usuario administrador
PRECONDICIONES	Disponer de al menos una entrada del blog
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Entrar en el menú de administración. 2. En la sección Blog pulsar sobre “Modificar entrada”. 3. Seleccionar la entrada que queremos modificar. 4. En el formulario modificar la información que queremos. 5. Pulsar actualizar
FLUJO ALTERNATIVO	
POSTCONDICIONES	Se modifica la entrada del blog con los nuevos datos y esta pasa a ser la primera entrada tanto en el blog como en la sección inferior de la aplicación donde se ven las últimas 3 entradas del blog.

Caso de uso : Borrar entrada del blog	
DESCRIPCIÓN	Esta funcionalidad permite borrar entradas en el blog de la tienda.
ACTORES	Usuario administrador
PRECONDICIONES	Disponer de al menos una entrada del blog
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Entrar en el menú de administración. 2. En la sección Blog pulsar sobre “Borrar entrada” 3. Seleccionar la entrada que queremos borrar. 4. Pulsar Borrar entrada.
FLUJO ALTERNATIVO	
POSTCONDICIONES	Se borra la entrada del blog tanto de la sección de blog como de las 3 entradas del blog en todas las páginas de la aplicación (si estuviese en esa sección).

Caso de uso : Dar privilegios de administrador	
DESCRIPCIÓN	Esta funcionalidad permite darle a otros usuarios registrados en la aplicación la opción de ser administradores, pudiendo así estos realizar todas las operaciones y ver todas las páginas que solo pueden ver los administradores.
ACTORES	Usuario administrador
PRECONDICIONES	Disponer al menos un usuario que no sea administrador
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Entrar en el menú de administración. 2. Pulsar sobre "Gestión de usuarios". 3. Pulsar sobre dar privilegios de administrador sobre el usuario al que le queramos dar estos privilegios.
FLUJO ALTERNATIVO	
POSTCONDICIONES	El usuario que se seleccionó pasa a ser administrador.

Caso de uso : Quitar privilegios de administrador	
DESCRIPCIÓN	Esta funcionalidad permite quitar los privilegios de administración a los usuarios que lo tengan.
ACTORES	Usuario administrador
PRECONDICIONES	Disponer al menos dos usuarios que sean administrador.
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Entrar en el menú de administración. 2. Pulsar sobre "Gestión de usuarios". 3. Pulsar sobre Quitar privilegios de administrador sobre el usuario al que le queramos dar estos privilegios.
FLUJO ALTERNATIVO	
POSTCONDICIONES	El usuario que se seleccionó deja de ser administrador para ser un usuario registrado

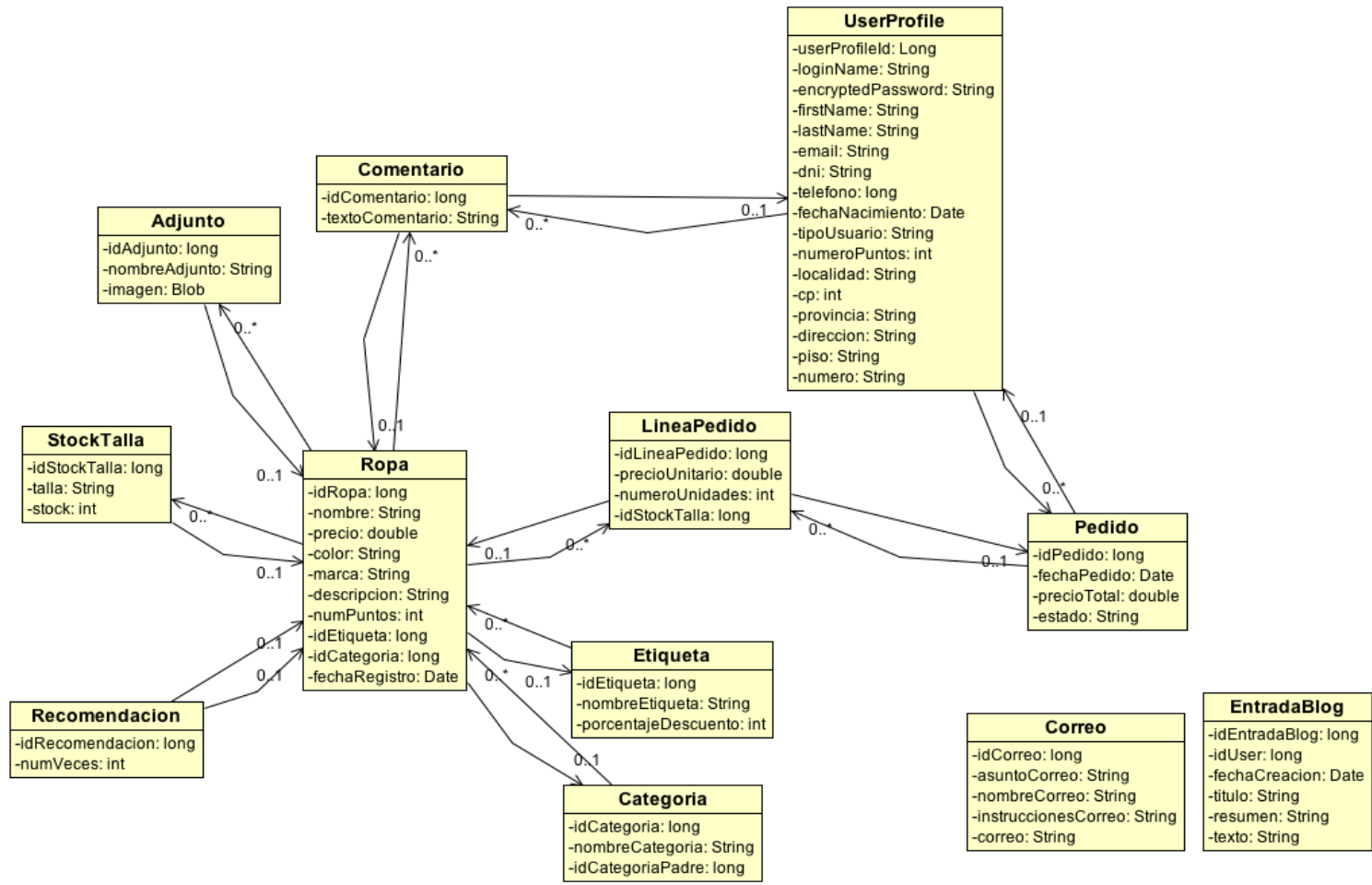
	normal.
--	---------

Caso de uso : Regalar puntos a usuario	
DESCRIPCIÓN	Esta funcionalidad permite regalarle puntos a un usuario.
ACTORES	Usuario administrador
PRECONDICIONES	Disponer al menos de un usuario.
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Entrar en el menú de administración. 2. Pulsar sobre "Gestión de usuarios". 3. Pulsar sobre Regalar puntos. 4. En el formulario rellenar el campo de número de puntos y pulsar en regalar.
FLUJO ALTERNATIVO	
POSTCONDICIONES	El usuario que se seleccionó se le suman de su cuenta de puntos el número de puntos que el administrador le regaló. Además al usuario en cuestión le llega un correo informándole del regalo de puntos.

Caso de uso : Gestionar pedidos	
DESCRIPCIÓN	Esta funcionalidad permite a los usuarios administradores gestionar los pedidos, comprobando el stock, enviándolo y cambiándole el estado del mismo según proceda.
ACTORES	Usuario administrador
PRECONDICIONES	Disponer al menos de un pedido en estado tramitación y otro en estado sin stock.
FLUJO NORMAL	<ol style="list-style-type: none"> 1. Entrar en el menú de administración. 2. Pulsar sobre "Ver Pedidos" 3. Pulsar sobre Tramitación para ver los pedidos en este estado.

	<ol style="list-style-type: none"> 4. Pulsar sobre ver productos en el pedido que queramos tramitar. 5. Pulsar en comprobar stock. 6. En la nueva ventana introducir la fecha de recepción por parte del usuario y el número de seguimiento del envío.
FLUJO ALTERNATIVO	<ol style="list-style-type: none"> 1. Entrar en el menú de administración. 2. Pulsar sobre “Ver Pedidos”. 3. Pulsar sobre Sin stock para ver los pedidos en este estado. 4. Comprobar que artículo no tiene stock. 5. Modificar el stock y talla de este artículo cuando dispongamos de él. 6. Pulsar en comprobar stock. 7. En la nueva ventana introducir la fecha de recepción por parte del usuario y el número de seguimiento del envío.
POSTCONDICIONES	<p>El pedido pasa a estado enviado. Y al usuario le llega un correo electrónico con la fecha de recepción del pedido y un número de seguimiento de la compañía con la cual se envía.</p>

A continuación se muestra el diseño para la capa de persistencia (base de datos) mediante un diagrama de clases:



2.2. Arquitectura

Para la realización de la aplicación se han seguido una serie de patrones que proporcionan al producto robustez, fiabilidad y modularidad. A continuación se comentan los principales patrones utilizados.

2.2.1. Patrones arquitectónicos

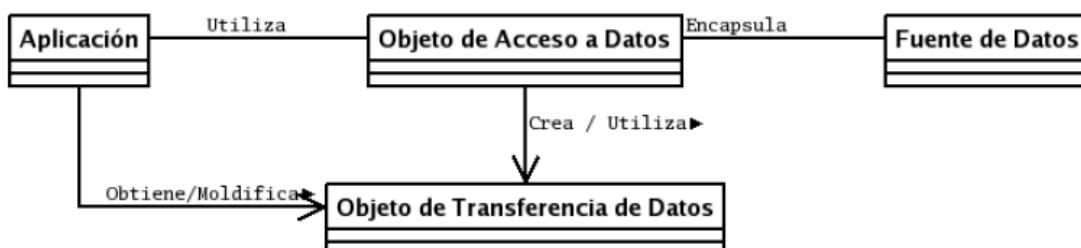
Tienen como objetivo aconsejar la arquitectura global que debe seguir una aplicación.

- **Patrón Model View Controller (MVC):** Patrón que proporciona una clara separación entre la lógica de negocio (modelo) y la interfaz gráfica con el usuario (vista). Por otro lado está un controlador que proporciona la interfaz entre la vista y el modelo. Este patrón favorece en gran medida la reutilización de clases, por ejemplo, para una misma lógica de negocio se pueden utilizar diferentes vistas. Por otro lado, junto con otros patrones de diseño, se consigue que los cambios en diferentes capas no sean apreciados por el resto.
- **Patrón Layers:** Ofrece soporte para la arquitectura modelo vista controlador. La aplicación se estructura en diferentes capas, ofreciendo transparencia hacia la implementación de cada capa. Esto favorece la separación de roles en la creación del producto software. En nuestro caso, la persona encargada del diseño gráfico de la aplicación apenas debe tener conocimientos de java. En esta aplicación, dentro de las tres capas definidas por el modelo MVC.

2.2.2. Patrón de diseño

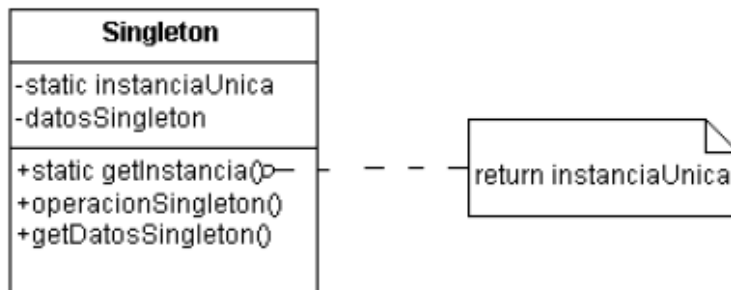
Los siguientes patrones explican cómo resolver un problema concreto de diseño.

- **Patrón Value Object (VO):** Representan el estado de los objetos de dominio (persistentes). Para ello tiene atributos privados y solo métodos get y set para los atributos modificables.
- **Patrón Data Access Object (DAO):** Patrón que permite desacoplar la lógica de negocio del acceso a la base de datos, ocultando el tipo de repositorio de datos utilizado. La interfaz DAO abstrae las operaciones sobre la fuente de datos, definiendo operaciones para insertar, borrar, actualizar y recuperar los objetos persistentes de un tipo. Para cada fuente de datos concreta se ofrecerá una implementación de la interfaz.



Patrón DAO

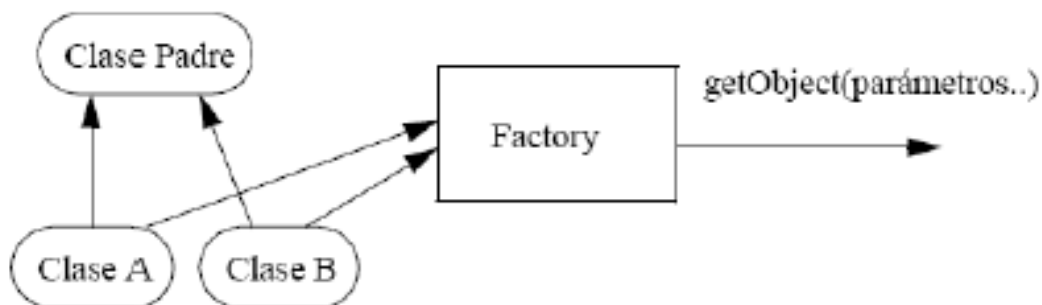
- **Patrón singleton:** Como características básicas se puede decir que este patrón está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. El patrón Singleton se implementa creando en la clase un método que crea una instancia del objeto sólo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se limita el alcance del constructor.



Patrón Singleton

Lo empleamos en nuestra aplicación para la sesión, para los DAOs y para los servicios.

- **Patrón Factory:** Este patrón (también denominado patrón Factoría) crea una instancia concreta de múltiples tipos, normalmente con el mismo padre, pero que se diferencian en algún aspecto. El objeto Factory tendrá un método para crear los objetos, este decidirá de qué tipo crea en función del parámetro que se le pase.



Patrón factory

Este patrón es muy útil ya que proporciona una manera transparente y fiable de construir objetos en función de algún criterio.

En el proyecto se usa para la creación de todos los beans de la aplicación como son los DAO y los servicios

- **Inyección de Dependencias:** es un patrón que tiene como finalidad conseguir un código más desacoplado, que nos facilitará las cosas a la hora de hacer Tests y además nos permite cambiar partes del sistema más fácilmente en caso de que fuese necesario. Tener el código desacoplado nos permite cambiar las dependencias en tiempo de ejecución basándonos en cualquier factor que considerásemos, para ello

necesitaríamos un Inyector o Contenedor que sería el encargado de inyectar las dependencias correctas en el momento necesario. Siguiendo el patrón de Inyección de Dependencias (DI, Dependency Injection) los componentes declaran sus dependencias, pero no se encargan de conseguirlas, ahí es donde entra el Contenedor de Spring, que en nuestras aplicaciones será el encargado de conseguir e inyectar las dependencias a los objetos.

2.2.3.Arquitectura general

La arquitectura JEE implica un modelo de aplicaciones distribuidas en diversas capas o niveles. Esta idea proporciona independencia entre las diferentes capas ofreciendo fiabilidad, mantenibilidad y flexibilidad.

Siguiendo el patrón MVC, la arquitectura global tendrá tres capas bien diferenciadas:

Modelo: En esta capa se encuentra la lógica de negocio de la aplicación. Se encarga de las transferencias entre la base de datos y la aplicación, asegurando la persistencia de los datos. Hacia el controlador ofrece una serie de fachadas que permite al controlador ejecutar las acciones que posee.

Controlador: Se puede decir que es la capa que une el modelo y la vista, recibiendo las peticiones de la vista y llamando al modelo para realizar la acción pertinente. A continuación recibe la información que el modelo le proporciona y la envía de forma adecuada a la vista.

Vista: Se puede considerar la interfaz con el usuario, permitiéndole realizar peticiones y mostrándole la información devuelta a través del controlador y modelo.

2.3. Prototipo del proyecto

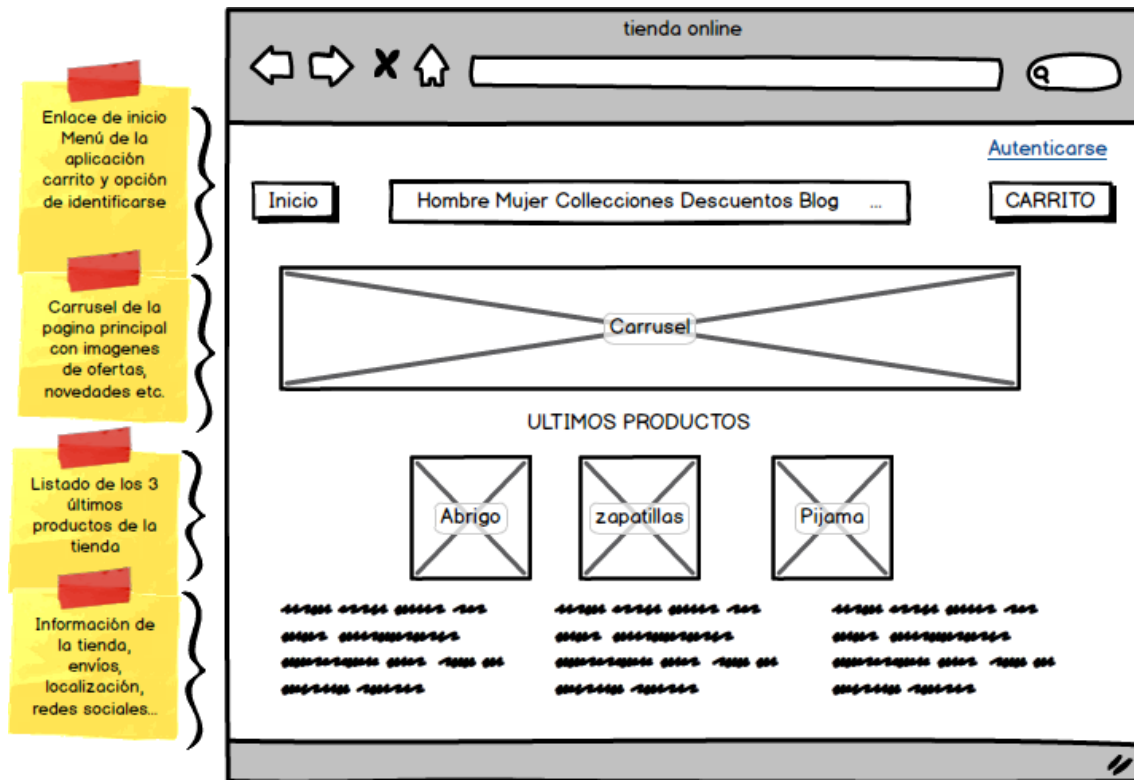
En esta sección mostraré los prototipos de las páginas mas relevantes que formarán la tienda de correo electrónico, a estos prototipos se les llama wireframe.

Un wireframe para un sitio web, es también conocido como un esquema de página. En resumen se trata de una guía visual que representa el esqueleto o estructura visual de un sitio web.

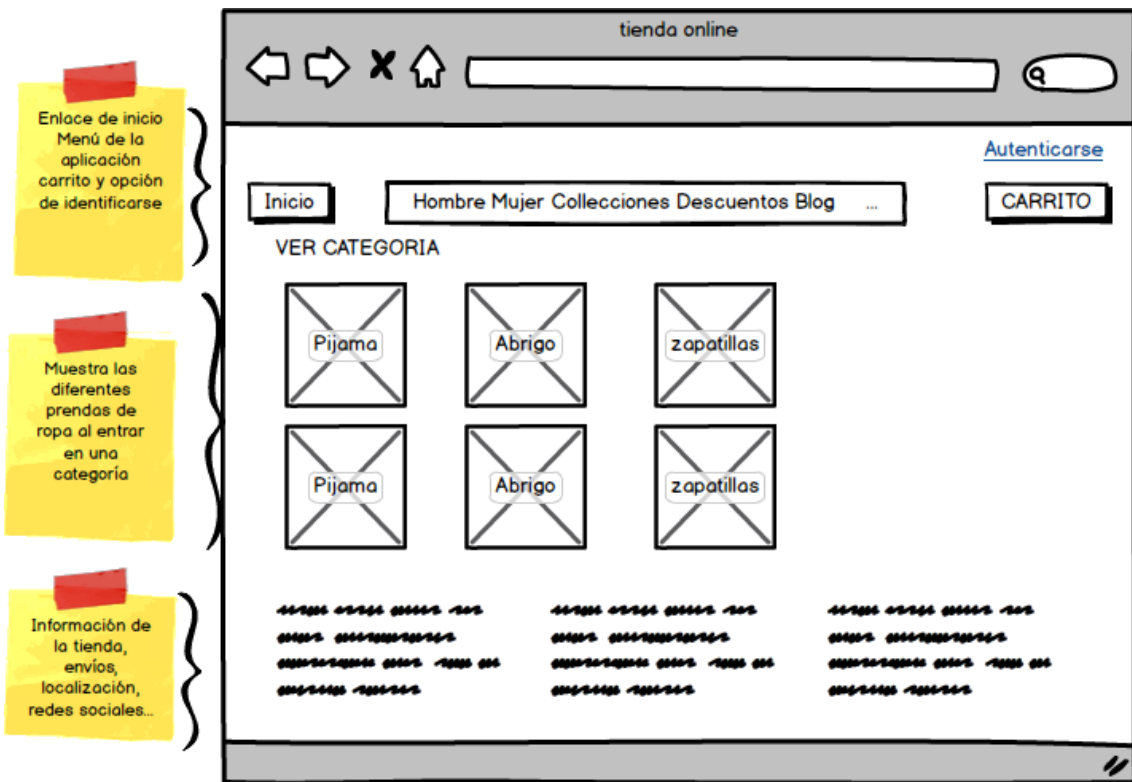
El wireframe esquematiza el diseño de página y la disposición del contenido del sitio, incluyendo elementos de la interfaz y sistemas de navegación, y cómo funcionan en conjunto. Así como algunas anotaciones que nos ayudan a desgranar los objetivos del site. Este esquema carece de estilo tipográfico, color o aplicaciones gráficas, ya que su principal objetivo reside en la funcionalidad, comportamiento y jerarquía de contenidos. Para presentar estos diseños esquemáticos han sido realizados utilizando una aplicación, en su versión gratuita, llamada Balsamiq Mockups. Se enfoca en “qué hace la pantalla, no cómo se ve.” Antiguamente los esquemas eran con lápiz y papel o esquemas en una pizarra. Actualmente hay muchas formas de hacerlos de forma automática, como el programa comentado, lo que facilita mucho el trabajo. Así como el poder compartirlos con equipos multidisciplinares y geográficamente dispersos.

A continuación mostraré los wireframes (junto a notas) de las páginas más relevantes de la aplicación.

Empiezo con la página principal, en la que se mostrará un menú con las diferentes categorías de la tienda. Un carrusel y los últimos productos de la tienda.

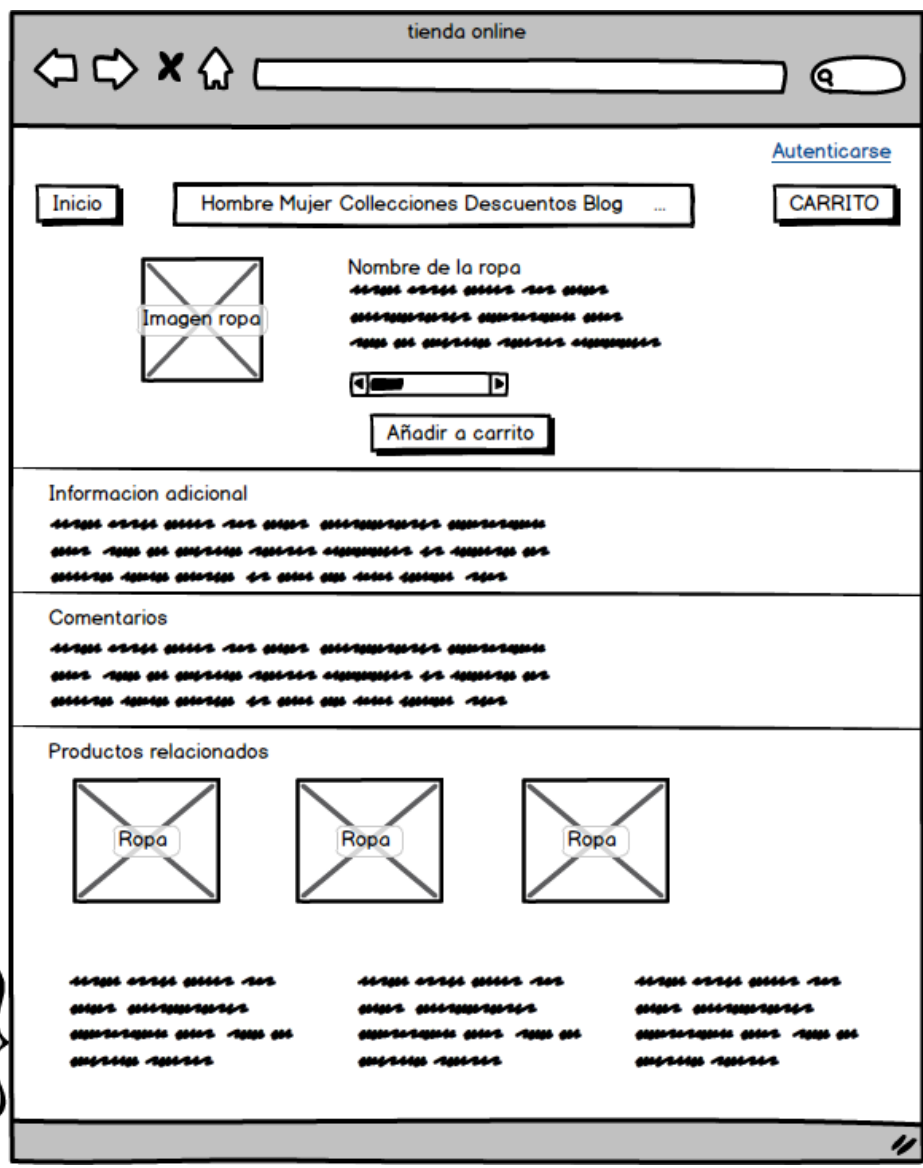


A continuación muestro el listado de productos que se mostrará cuando se pulse sobre una categoría.

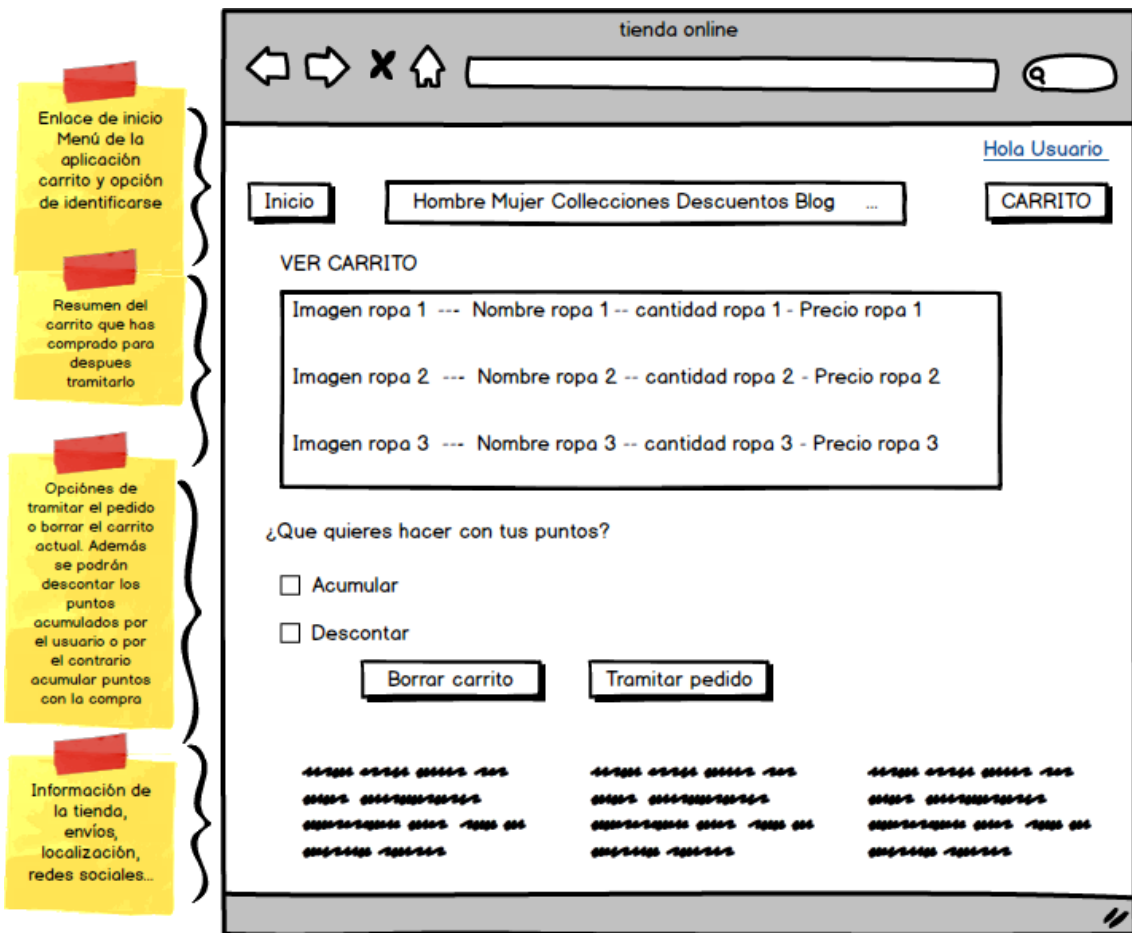


Después muestro el prototipo o wireframe que se mostrará cuando se selecciona una prenda de ropa dentro del listado de categoría o en cualquier otro enlace de la página.

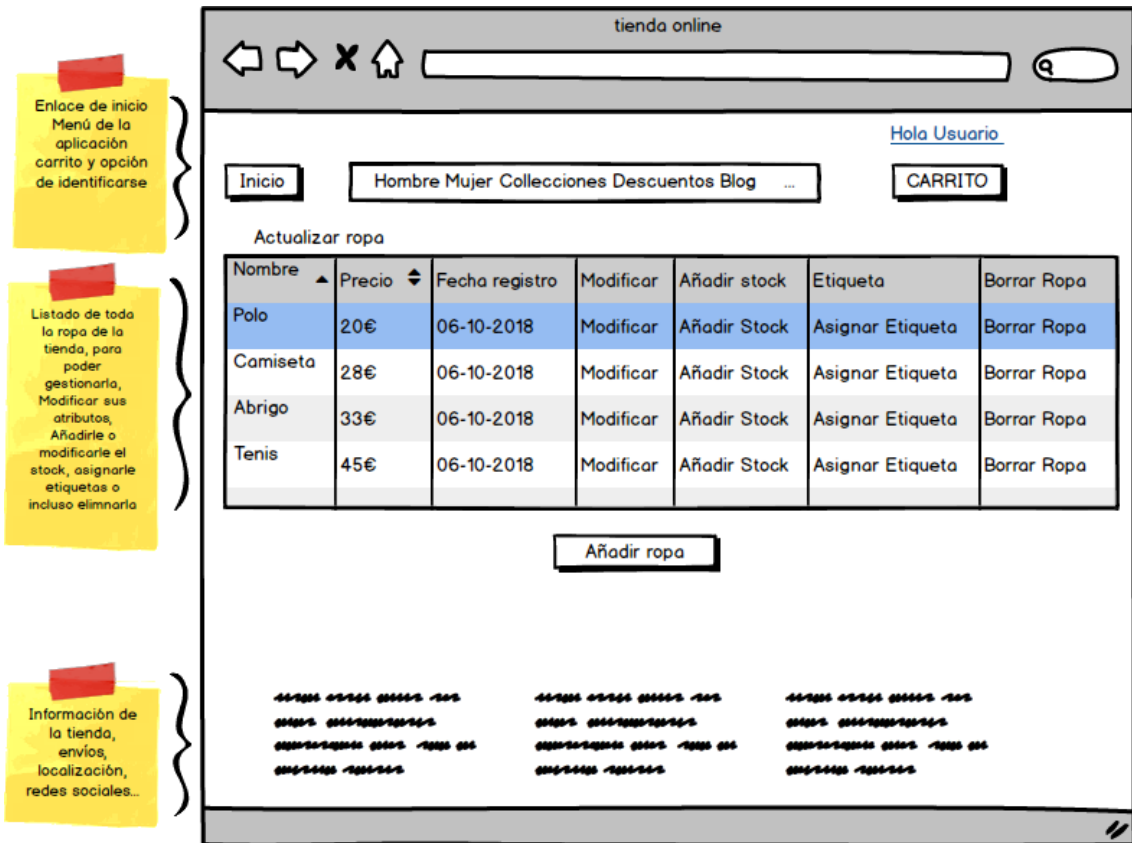
- Enlace de inicio Menú de la aplicación carrito y opción de identificarse
- Descripción del producto en el que entramos. Foto, Nombre, breve descripción. Listado de tallas disponibles y botón de añadir al carrito
- 3 desplegables en el primero información adicional del producto en el segundo comentarios de otros compradores sobre el producto y en el último desplegable productos que otros usuarios han comprado junto al que estamos mostrando.
- Información de la tienda, envíos, localización, redes sociales...



Después muestro la página que gestiona el carrito una vez se hayan elegidos los productos que se quieren comprar.



Ahora mostraré una de las páginas de la administración de la tienda, en ella se muestra, el listado de la ropa de la tienda así como las opciones que tiene el administrador sobre ellas. Como esta página será toda la sección de administración que permitirá gestionar, ropa, categorías o etiquetas.



Por último muestro el prototipo del listado de entradas del blog que contendrá la tienda online.



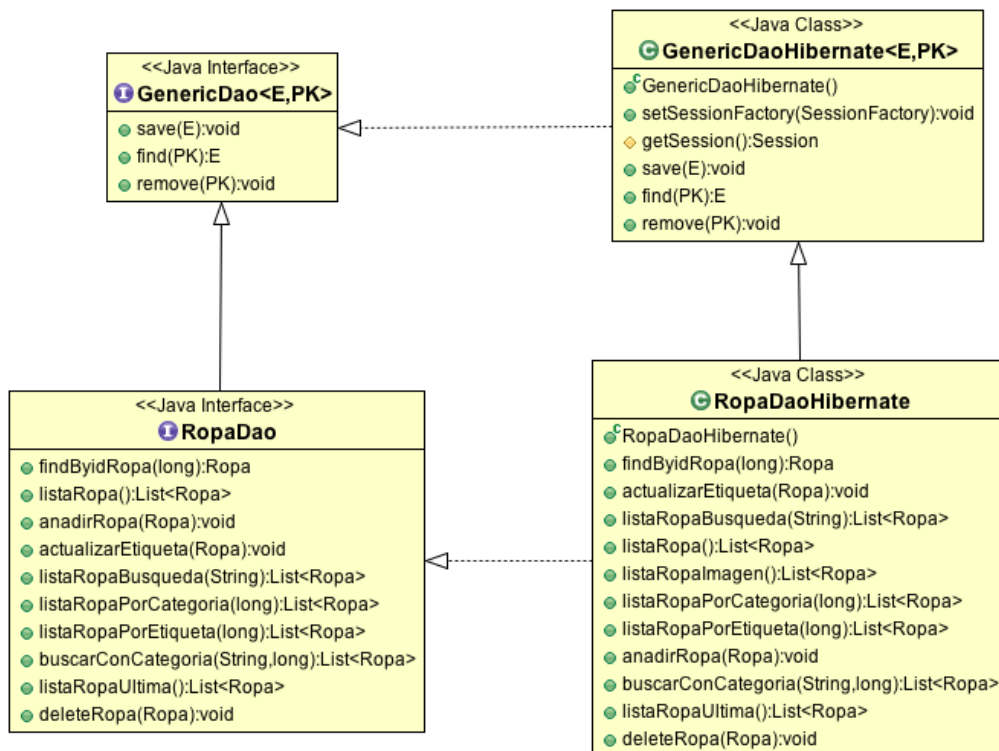
3. Desarrollo de la aplicación

A continuación se va a explicar de forma general como se ha desarrollado la aplicación. Al tratarse de una arquitectura orientada al servidor y un MVC (modelo vista controlador), se explicará donde están definidas cada uno de los elementos que intervienen: Programas del servidor, modelo de datos, vistas, controladores, etc.

3.1.1. Implementación de la capa de persistencia

La capa de persistencia está implementada usando el framework de Mapeo objeto-relacional (ORM) **Hibernate**. Todas las clases de persistencia nombradas anteriormente presentan una interfaz de acceso a los datos siguiendo el dictamen del patrón de diseño Data Access Object. Junto con las interfaces, se ha utilizado la interfaz genérica y su implementación concreta, en nuestro caso de Hibernate, que implementa las operaciones CRUD de cada clase.

En el siguiente diagrama vamos a ver cómo está implementada la capa de acceso a datos, para ello vemos de qué manera están diseñados los DAO (forma de acceso a base de datos). Disponemos tanto de una interfaz como de una implementación genéricas **GenericDao** y **Generic-DaoHibernate**, que son empleadas para la especialización de otros DAO. Empleamos como ejemplo el de **Ropa** ya que en el resto de clases se implementará de la misma manera.



Dentro de la implementación de esta capa, para todas las claves foráneas presentes en la base de datos se utiliza la estrategia Lazy. Esta estrategia es muy usada. Lo que realiza LAZY consiste en que cuando lanzamos una consulta, Hibernate por defecto intenta cargar el mínimo de datos posible ya que quizás no los necesitemos todos y sea una pérdida de recursos.

3.1.2. Implementación lógica de negocio

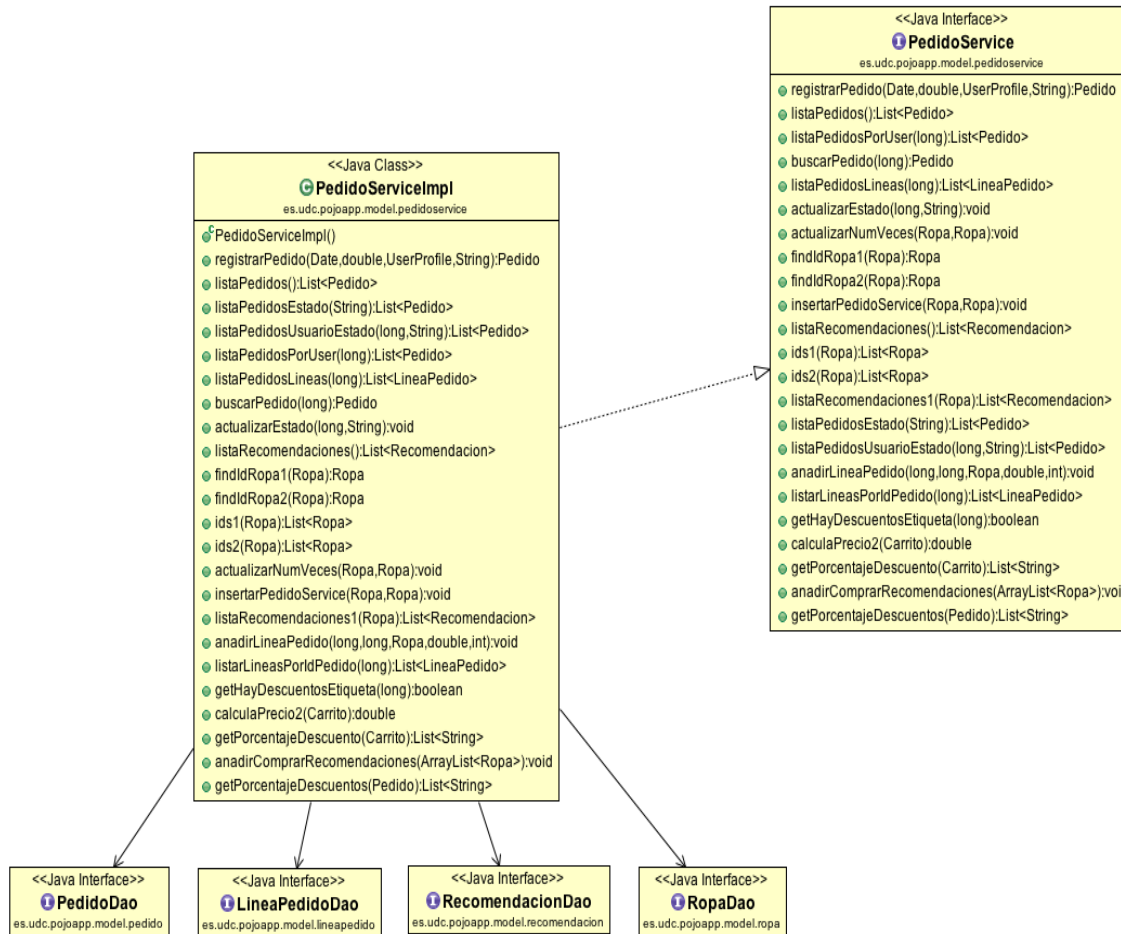
Para explicar la lógica de negocio de la aplicación, se han juntado los métodos en fachadas que agrupan funcionalidades relacionadas. Además también se ha tenido en cuenta agruparlas para poder usarlas en otro contexto.

En todas las interfaces se ha hecho uso de la anotación de **Spring @Transactional** para la gestión de transacciones. Una transacción de base de datos es un conjunto de instrucciones que se ejecutan en bloque.

Por ejemplo, realizar una consulta, modificar un registro A en la base de datos y eliminar un registro B. Si en alguna de estas instrucciones se produce un error, todo el proceso se echa atrás. De esta manera si posteriormente consultamos la base de datos, El registro A no habrá sido alterado. Este proceso de “deshacer” las instrucciones realizadas se denomina rollback. También hemos utilizado la anotación **@Transactional (readOnly = true)**, es decir la anotación **@Transactional** marcada como **readOnly**. Esto lo hacemos para los métodos donde no queremos escribir en base de datos, solo leer, para que así esta transacción no modifique datos y sea más óptima (sólo incurrirá en bloqueos de lectura).

Descripción de los servicios

A continuación se mostrará como están implementados los servicios que presenta la aplicación. En la siguiente figura se muestra un ejemplo de la arquitectura que presentan todos los servicios. Se ha utilizado el servicio de pedido como ejemplo. Se puede observar cómo se define la funcionalidad de un servicio a través de una interfaz, que será implementada por una clase concreta. La implementación del servicio se realiza en términos de DAOs para gestionar la persistencia de las entidades y de esta forma definir la lógica de negocio asociada a cada caso de uso.



Resumen de funcionalidad del Servicio pedido service: Este servicio recoge las funcionalidades relativas al pedido, tal como registrarlo, calcular su precio, añadir las recomendaciones o listarlos según su estado o usuario. En este servicio se declaran los daos, ropaDao, pedidoDao, recomendacionDao o etiquetaDao.

Descripción de métodos:

- **registrarPedido:** Método que a partir de unos datos facilitados, tales como la fecha, el precio el usuario registra un nuevo pedido.
- **listaPedidosPorUser:** Obtiene una lista de pedidos de un usuario en concreto.
- **buscarPedido:** Busca un pedido a través de su id.
- **actualizarEstado:** Actualizar el estado de un pedido.
- **actualizarNumVeces:** En una recomendación si se repite la recomendación que se pasa en lo parámetros de este método, se actualiza el número de veces de la recomendación sumándole uno.
- **listaRecomendaciones** y **listaRecomendacionTodas:** Devuelven una lista de recomendaciones según la ropa que se quiera obtener su lista de ropa recomendada.
- **listaPedidosEstados** y **listaPedidosEstadosUsuario:** Estos métodos obtienen una lista de pedidos según el estado del mismo y según el usuario que lo haya realizado.
- **anadirLineaPedido:** Se añade una línea de pedido con los datos solicitados, como el precio de la línea, la cantidad, el pedido al que está asociado o la ropa que tiene dicha línea.
- **listarLineasPorIdPedido:** Lista todas las líneas de un pedido.
- **getHayDescuentosEtiqueta:** Con este método obtenemos si hay o no descuentos a través de algún descuento asociado a una etiqueta.
- **getPorcentajeDescuento:** Con este método sabremos si el usuario ha usado sus puntos para obtener un descuento en su pedido.
- **calculaPrecio:** Con este método obtenemos el precio del carrito que se le pasa como parámetro.
- **anadirComprarRecomendaciones:** Añade las recomendaciones oportunas según los ids de los pedidos de la ropa que está asociada al pedido que se compra.

- **listarLineasPedidoPorRopa**: Obtiene todas las líneas de pedido de una prenda de ropa en concreto.
- **borrarPedido**: Borra un pedido a partir de su id.
- **borrarLineaPedido**: Borra una línea de pedido a través de su id.
- **borrarRecomendacion**: Borra una recomendación a través de su id.

3.1.3. Implementación capa web

Ahora abordaremos el diseño final que ven los usuarios de la aplicación es decir, la capa vista. Se realiza una introducción a las pantallas con las que los usuarios interactúan.

Para la implementación de la vista se usa el framework Tapestry. En esta implementación son necesarios dos archivos por cada página de las que consta nuestra aplicación.

En primer lugar hace falta una clase .java que es la que define el comportamiento de la página al interactuar con ella.

En segundo lugar es necesaria una plantilla, muy similar a un documento HTML, que sólo se diferencia en unas anotaciones empleadas exclusivamente por Tapestry, que sirven como referencias a los valores de variables, funciones y otros elementos definidos en la clase .java de la que hablamos previamente. Estas plantillas tienen extensión .tml que son las iniciales de Tapestry Markup Language. Es importante que ambos ficheros, el .java y el .tml se llamen igual. A su vez existe un tercer fichero (o grupo de ficheros) en la implementación de la vista. Este fichero es el encargado de la internacionalización, y tiene una extensión .properties (puede existir uno por idioma para el que existan textos internacionalizados en la aplicación).

A excepción de los datos almacenados en la base de datos, todas las cadenas de texto de la aplicación están almacenadas en estos ficheros de configuración. Por otro lado todas las páginas están compuestas de un layout, que es un componente creado para que todas las páginas usen una misma apariencia general y para que toda la aplicación tenga dos partes comunes, que son la cabecera con el menú principal de la tienda y el footer, con información relevante de nuestra tienda.

En todas las cabeceras de las páginas en el archivo .tml se define el layout de la siguiente manera:

```
<html xmlns:t="http://tapestry.apache.org/schema/tapestry_5_3.xs"
      xmlns:p="tapestry:parameter" t:type="Layout" t:pageTitle="title">
```

La primera línea especifica el esquema de Tapestry usado, en nuestro caso Tapestry 5. En la segunda línea se especifica el tipo de la página, que es lo mismo que decir la plantilla usaremos. También en esta línea se especifica el título de la página mostrado por el navegador. Es posible tener más de una

plantilla creada y usar la que más convenga en cada caso. En nuestra aplicación solo fue necesario crear una.

Para crear este layout (y todos los estilos de la aplicación) nos ayudamos de Bootstrap que nos hace más sencillo por ejemplo los efectos del menú o para hacer responsiva la aplicación (es decir que cambie la forma de distribuir los objetos según el tamaño de la pantalla y adaptando toda la página a este tamaño).

Otro aspectos a tener en cuenta de la capa web son los filtrados usados del framework Tapestry, que es el que se encarga de gestionar las peticiones web, para entre otras cosas el correcto funcionamiento del acceso o no a páginas según los permisos del usuario que intenta acceder.

En nuestra aplicación usamos los siguientes filtros del framework:

SessionFilter: Este filtro lo usamos para mantener la sesión de usuarios y carritos abierta es decir, mientras no expire, la sesión se mantendrá abierta para el usuario y si este crea un carrito también para su carrito. Mantener la sesión abierta ayudará al usuario a que no tenga que loguearse cada vez que salga de la aplicación.

PageRenderAuthenticationFilter: Este filtro gestiona el control de acceso a las páginas antes de que estas sean cargadas por el usuario. Hace uso de la clase (AuthenticationValidator) que valida si el usuario tiene acceso o no a la página en cuestión según unos criterios. Es decir tenemos los siguientes tipos de acceso:

- **USUARIOS AUTENTICADOS:** Para restringir el acceso a las páginas a estos usuarios, deberemos anotar la clase página de la siguiente forma: [@AuthenticationPolicy\(AuthenticationPolicyType.AUTHENTICATED_USERS\)](#).

Si un usuario intenta entrar en una página sólo accesible para usuarios registrados la aplicación lo redirigirá a la página de login.

- **ADMINISTRADORES:** Para limitar las páginas para que solo puedan verlas los administradores. Se crea la anotación @Private, si una página tiene esta anotación al principio de la clase, el AccessController lo controlará y comprobará si el usuario es de tipo administrador, sino lo es, lo redirigirá a la página principal de la tienda

3.1.4.Test

A continuación se detallarán las pruebas que se llevarán a cabo en la aplicación para comprobar su funcionamiento, su rendimiento y su usabilidad. I

Pruebas unitarias

En programación, una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. Para que una prueba unitaria sea buena se deben cumplir los siguientes requisitos:

- **Automatizable:** no debería requerirse una intervención manual. Esto es especialmente útil para integración continúa.
- **Completas:** deben cubrir la mayor cantidad de código.
- **Repetibles o Reutilizables:** no se deben crear pruebas que sólo puedan ser ejecutadas una sola vez. También es útil para integración continua.
- **Independientes:** la ejecución de una prueba no debe afectar a la ejecución de otra.
- **Profesionales:** las pruebas deben ser consideradas igual que el código, con la misma profesionalidad, documentación, etc.

Aunque estos requisitos no tienen que ser cumplidos a rajatabla, se recomienda seguirlos o de lo contrario las pruebas pierden parte de su función. El objetivo de las pruebas unitarias es aislar, cada parte del programa y mostrar que las partes individuales son correctas. Proporcionan un contrato escrito que el trozo de código debe satisfacer. Estas pruebas aisladas proporcionan cinco ventajas básicas:

1. **Fomentan el cambio:** Las pruebas unitarias facilitan que el programador cambie el código para mejorar su estructura (lo que se ha dado en llamar refactorización), puesto que permiten hacer pruebas sobre los cambios y así asegurarse de que los nuevos cambios no han introducido errores.
2. **Simplifica la integración:** Puesto que permiten llegar a la fase de integración con un grado alto de seguridad de que el código está funcionando correctamente. De esta manera se facilitan las pruebas de integración.
3. **Documenta el código:** Las propias pruebas son documentación del código puesto que ahí se puede ver cómo utilizarlo
4. **Separación de la interfaz y la implementación.**
5. **Los errores están más acotados y son más fáciles de localizar:** dado que tenemos pruebas unitarias que pueden desenmascararlos.

Es importante darse cuenta de que las pruebas unitarias no descubrirán todos los errores del código. Por definición, sólo prueban las unidades por sí solas. Por lo tanto, no descubrirán errores de integración, problemas de rendimiento y otros problemas que afectan a todo el sistema en su conjunto. Además, puede no ser trivial anticipar todos los casos especiales de entradas que puede recibir en realidad la unidad de programa bajo estudio. Las pruebas unitarias sólo son efectivas si se usan en conjunto con otras pruebas de software.

Existe software dedicado a automatizar estas pruebas unitarias, el más conocido en el entorno Java es JUnit [11] que se puede automatizar mediante el uso de Maven.

En la siguiente imagen se muestra el ejemplo del código de una de las pruebas unitarias realizadas en la aplicación:

```

@Test
public void testFindComentario() throws InstanceNotFoundException {
    Random rand = new Random();
    Ropa ropa = ropaService.findRopa(1L);
    UserProfile userProfile2 = userService.findUserProfile(7L);
    String comentarioString = "ComentarioPrueba-" + rand.nextInt(9999);
    Comentario comentario = userService.registrarComentario(comentarioString,
        userProfile2, ropa);
    Comentario aux = userService.findComentario(comentario.getIdComentario());
    /* Check data. */
    assertEquals(comentario, aux);
    ropaService.borrarComentario(aux.getIdComentario());
}

```

La prueba que se muestra se realiza para el método findComentario del servicio userService, en ella estamos creando aleatoriamente un nuevo comentario, y a través del método findComentario recuperandolo después comprobamos que son idénticos. Por último borramos el comentario registrado. Como se puede ver con el ejemplo, la idea de las pruebas unitarias es que son independientes unas de otras, es decir esta prueba no tiene nada que ver con las otras realizadas. Únicamente validan el caso de uso a probar y en el caso de que se hayan creado datos (como es el caso del ejemplo) se borran.

Maven también nos permite automatizar la obtención de la cobertura realizada en nuestras pruebas unitarias mediante JUnit.

La cobertura es el porcentaje de código que cubren las pruebas unitarias. Lo recomendable para estas pruebas es que cubra el 80% del código de los servicios.

A continuación se muestra una imagen de la cobertura de la aplicación.

Coverage Report - All Packages

Package /	# Classes	Line Coverage	Branch Coverage	Complexity
All Packages	114	35% 949/2643	17% 81/454	1,351
es.udc.pojoapp.model.adjunto	3	85% 29/34	N/A N/A	1
es.udc.pojoapp.model.administradorservice	2	87% 48/55	N/A N/A	1,057
es.udc.pojoapp.model.categoria	3	85% 30/35	50% 1/2	1,095
es.udc.pojoapp.model.comentario	3	93% 27/29	N/A N/A	1
es.udc.pojoapp.model.correo	3	84% 27/32	50% 1/2	1,111
es.udc.pojoapp.model.entradablog	3	92% 38/41	50% 1/2	1,083
es.udc.pojoapp.model.etiqueta	3	82% 55/68	75% 6/8	1,2
es.udc.pojoapp.model.lineapedido	3	60% 27/45	0% 0/2	1,13
es.udc.pojoapp.model.pedido	3	81% 45/55	0% 0/4	1,097
es.udc.pojoapp.model.pedidoservice	2	73% 84/114	62% 35/56	1,518
es.udc.pojoapp.model.recomendacion	3	73% 33/45	25% 2/8	1,269
es.udc.pojoapp.model.ropa	3	89% 87/97	50% 1/2	1,037
es.udc.pojoapp.model.ropaservice	2	82% 57/69	N/A N/A	1
es.udc.pojoapp.model.stocktalla	3	96% 30/31	N/A N/A	1
es.udc.pojoapp.model.userprofile	3	98% 85/86	100% 2/2	1,044
es.udc.pojoapp.model.userservice	4	93% 77/82	87% 7/8	1,231
es.udc.pojoapp.model.userservice.util	2	94% 147/155	82% 23/28	1,933

Pruebas de rendimiento

El rendimiento de nuestra aplicación depende del número de peticiones simultáneas que se realizan a ella.

Se utiliza el software Apache Jmeter para realizar pruebas de rendimiento sobre la aplicación. Esta herramienta (gratuita) permite simular la navegación

de varios usuarios (denominados en la aplicación threads o hilos). Con esta aplicación podemos simular una carga de peticiones contra nuestra aplicación y así obtener el tiempo de respuesta de la aplicación a diferentes acciones de los usuarios.

Se han creado dos script de navegación (ayudándonos de la aplicación BadBoy, que a través una simple navegación captura los datos para importarlos a continuación, en Jmeter). Los dos scripts recorren 6 páginas de la aplicación. En la primera prueba respuesta ha sido más que satisfactoria como se puede ver a continuación:

Informe Agregado

Nombre: Informe Agregado

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Etiqueta	# Muestras	Media	Mediana	Linea de 9...	Mín	Máx	% Error	Rendi...	Kb/sec
http://localhost/pojo-app/	4200	139	38	59	33	277371	0,00%	1,0/sec	18,4
http://localhost/pojo-app/ropa/vercategoria/3	2100	69	31	52	27	8646	0,00%	30,6/...	8,8
http://localhost/pojo-app/ropa/ver/61	2100	72	43	70	38	1569	0,00%	30,6/...	10,6
http://localhost/pojo-app/ropa/ver/60;jsessionid=qt3...	2100	80	44	69	39	4815	0,00%	30,6/...	11,4
http://localhost/pojo-app/user/login	2100	53	23	35	20	2906	0,00%	30,6/...	9,2
http://localhost/pojo-app/ropa/busqueda/a/3	2100	60	31	49	27	1362	0,00%	30,6/...	10,2
Total	14700	88	38	60	20	277371	0,00%	3,6/sec	68,5

La carga de trabajo en esta prueba llegó a un total de 14.700 muestras (las muestras son el número de páginas visitadas). Esto se realizó mediante 100 hilos (o usuarios), que navegaban a través de las páginas y que se iban lanzando cada 10 segundos. Cada usuario recorría el bucle que se puede ver en la imagen 21 veces, (excepto la página principal que se recorría 2 veces por iteración de bucle), como se puede ver en la imagen el error es del 0%, además los tiempos de respuesta son más que aceptables.

A continuación se decide hacer una prueba con más carga de usuarios y de navegación, usando 200 usuarios (o hilos en la aplicación) y un bucle de repeticiones de 80 veces cada página, cada usuario también se lanza cada 10 segundos.

Informe Agregado

Nombre: Informe Agregado

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Etiqueta	# Muestras	Media	Mediana	Linea de 9...	Mín	Máx	% Error	Rendimien...	Kb/sec
http://localhost/pojo-app/	32000	58	33	36	31	943	0,00%	3,2/sec	57,8
http://localhost/pojo-app/ropa/ver/60	16000	58	39	41	36	1217	0,00%	1,6/sec	33,6
http://localhost/pojo-app/user/login	16000	42	20	22	19	1242	0,00%	1,6/sec	29,0
http://localhost/pojo-app/ropa/ver/33	16000	51	38	41	36	900	0,00%	1,6/sec	35,8
http://localhost/pojo-app/ropa/vercategoria/2	16000	46	27	29	25	1251	0,00%	1,6/sec	27,7
http://localhost/pojo-app/ropa/busqueda/a/3	16000	44	27	29	25	955	0,00%	1,6/sec	32,0
Total	112000	51	33	40	19	1251	0,00%	11,2/sec	215,8

Como se puede ver el error sigue en 0% y el rendimiento sigue siendo muy aceptable y eso que la carga de muestras o páginas vistas en esta prueba de rendimiento ascendió a 112.000.

Es recomendable realizar un estudio periódico del número de usuarios que pueden llegar a estar conectados simultáneamente a nuestra aplicación para

ampliar el servidor en caso de que sea necesario para poder atender un mayor número de peticiones simultáneas.

Pruebas de usabilidad.

Usabilidad: Es la eficiencia con la que se ofrece al usuario el servicio o información que este requiere. Cualquier website enfocado en estrategias de marketing online debe ser altamente usable, sobre todo para su público objetivo

Se aplicará la evaluación heurística para evaluar la página web, con esta técnica identificaremos los problemas, verificaremos que las normas de usabilidad son respetadas y además nos permitirá explicar de forma consensuada los problemas de usabilidad observados.

Se evaluará la página a través de los diez principios heurísticos de Nielsen, estos diez principios son los siguientes:

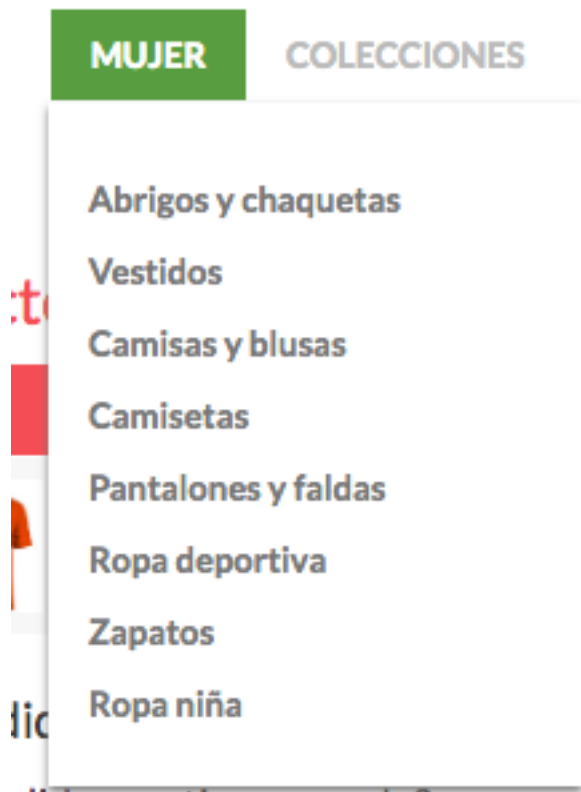
- Visibilidad del estado del sistema.
- Adecuación entre el sistema y el mundo real
- Libertad y control por parte del usuario
- Consistencia y estándares
- Prevención de errores
- Reconocimiento antes que recuerdo
- Flexibilidad y eficiencia en el uso
- Diseño estético y minimalista
- Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores
- Ayuda y documentación

1. Visibilidad del estado del sistema: El usuario tiene que saber siempre lo que está sucediendo y que está haciendo en el sistema. La web cumple con este objetivo, por ejemplo en el caso de tramitar un pedido, la propia web informa de que se tramitará el pedido con un mensaje como el siguiente:

Este pedido pasara a tramitarse, esta seguro?



2. Adecuación entre el sistema y el mundo real: Como podemos comprobar la web usa un lenguaje adecuado para una tienda, como podemos ver en la siguiente imagen, que nos vale de ejemplo las categorías en los que se clasifica la web, bien podrían ser los de una tienda física.



- Libertad y control por parte del usuario: Nielsen con esto nos quiere decir que en caso de elegir una página por error el usuario debería disponer de una “salida de emergencia” que será fácil de identificar y no disponga de un diálogo largo y complicado con la página web, en este caso, la aplicación por ejemplo en la cesta nos permitió eliminar cualquier artículo de este o modificar su cantidad

Mi Carrito

Foto	Nombre	Cantidad	Precio
	Polo	1 	20.0 €

QUE QUIERES HACER CON LOS PUNTOS?

Acumular Descontar

Vas a acumular 0 puntos en tu cuenta.


BORRAR CARRITO

TRAMITAR PEDIDO

- Consistencia y estándares. Los usuarios no deben saber que diferentes palabras, situaciones o acciones significan lo mismo. Es conveniente seguir convenciones. La página web intenta seguir los estándares de otras tiendas online, por ejemplo dispone como muchas de un buscador en el cual se puede buscar por el nombre del artículo que se quiere comprar.

5. Prevención de errores. Como se puede ver al tramitar el pedido, la aplicación prevee que pueda haber un error en la dirección por eso previamente a seguir tramitando el pedido informa al usuario de la dirección de envío:

Datos de pedido creado

Foto	Cantidad	Precio
	1	20.0 €

EL PRECIO FINAL DEL PEDIDO ES: 20.0 EUROS

ESTOS SON LOS DATOS A LOS QUE SE LE ENVIARÁ EL PEDIDO :

Nombre: Admin123 Dominguez
Correo electronico: emilio89@gmail.com
Telefono: 981861593
Dni: 99999999s
Provincia: A Coruña:aa
Localidad: Santiago de Compostela
Numero: 10
Tus Puntos ahora son: 0

BORRAR PEDIDO


TRAMITAR PEDIDO

6. Reconocimiento antes que recuerdo. El usuario no tendrá que reconocer ningún proceso ya que estos son sencillos de reconocer y usar.
7. Flexibilidad y eficiencia en el uso. La aplicación guía al usuario a través de todo el proceso de compra, informando de los pasos que se están llevando a cabo a través de la aplicación y en algún caso también a través de correos electrónicos. Por eso creo que cumple bien con este objetivo.
8. Diseño estético y minimalista. Como se puede ver en la portada de la página esta sigue un diseño minimalista, solo mostrándonos la información más relevante, como puede ser el buscador, las diferentes categorías y subcategorías y los últimos productos añadidos a la tienda.

Bienvenido a la Tienda [Autenticarse](#)

INICIO | [HOMBRE](#) | [MUJER](#) | [COLECCIONES](#) | [DESCUENTOS](#) | [BLOG](#)

CARRITO: 0.0 € (0 PRODC.)
[VACIAR CARRITO](#)




NUEVAS LLEGADAS.


VER TODAS LAS NUEVAS TENDENCIAS

[COMPRAR AHORA](#)


ÚLTIMOS PRODUCTOS



Zapatos de piel
65.0€




Zapatillas deportivas
55.0€



Polo
20.0€

SOBRE NOSOTROS

Nuestra tienda es la marca de confianza y la mejor elección en el mundo del diseño de moda. La amplia gama de productos se dirige a personas que valoran el buen gusto y la variedad con la mejor relación calidad precio y que buscan una experiencia de compra fácil y agradable.



Nuestra tienda

Calle A Coruña,
Santiago de compostela, nº 1.

Telefono: +34 981 981 981
FAX: +34 981 981 981
E-mail: info@tiendaonline.com



Entradas en blog

Lo que te perdiste esta semana- la sección preferida

Aquí la sección resumen sobre lo que pasó esta semana, tanto en la tienda, como en la moda en general

ESTA ES LA TERCERA ENTRADA DEL BLOG, Y LO ESTOY PROBANDO

Este es el TERCERA resumen del blog y si también lo estoy probando

ESTA ES LA SEGUNDA ENTRADA DEL BLOG, Y LO ESTOY PROBANDO

Este es el SEGUNDO resumen del blog y si también lo estoy probando

Support

[FAQ](#)
[Pagos](#)
[Seguimiento de paquete](#)
[Sobre nosotros](#)
[Contactanos](#)

- Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores. Como nos dice Nielsen . Los mensajes de error deben expresarse en un lenguaje común y sencillo, que indique con precisión el problema y sugiera las posibles alternativas o soluciones. Esto nos sucede cuando un usuario, por ejemplo quiere registrarse y no ha indicado un valor marcado como obligatorio, la aplicación lo marca de forma muy intuitiva tal y como se puede mostrar a continuación

REGISTRO DE USUARIOS

Identificador usuario



Debe introducir un valor para Identificador de usuario.

10. Ayuda y documentación. Como se puede ver en la siguiente imagen, la aplicación en el footer de la página dispone de una sección para ayuda y documentación, tanto en los procesos de, compra de pago etc.

Support

[FAQ](#)

[Pagos](#)

[Seguimiento de paquete](#)

[Sobre nosotros](#)

[Contactanos](#)

Por todo esto creo que la aplicación tiene una usabilidad más que aceptable ya que revisando todos los principios de Nielsen a través de la aplicación no hemos encontrado fallos significativos y lo más destacable es que tiene un proceso fácil e intuitivo de compra.

3.1.5. Versiones de la aplicación/servicio

Con esta entrega, se adjuntará la versión 1.0.0 de la aplicación la cual es una versión estable preparada para poner en un entorno de producción y vender productos de una tienda de ropa con ella.

3.1.6.Bugs.

No se encuentran bugs o errores destacables en la aplicación.

3.1.7.Futuras líneas de trabajo

En mi opinión, una aplicación nunca está terminada, siempre puede ser mejorada incorporando nuevas funcionalidades y mejoras que faciliten y automaticen el trabajo a los usuarios. Las posibles mejoras que no han sido incorporadas por estar fuera de los objetivos y tiempo han sido:

- Añadir la implementación del sistema de pago por PayPal para automatizar el pago de los productos comprados en nuestra tienda y así añadir un tipo de compra a nuestra tienda, la compra contrarrembolso (ya implementada) y el pago por el servicio PayPal.
- Añadir más roles a la aplicación. Actualmente solo existen dos roles de usuarios registrados: los normales y los administradores. Sería muy útil que exista otro tipo de usuarios como pueden ser “Dinamizadores”, con la capacidad de registrar y llevar el stock de la tienda.
- Añadir una aplicación móvil a la tienda. Esto supondría bastante desarrollo, pero como hoy en día las tecnologías giran en torno a dispositivos móviles (smartphone o tablets) sería un trabajo que daría mucha más utilidad y facilidades a los clientes de la tienda

4. Conclusiones.

Se explicará en este capítulo los objetivos alcanzados gracias a este proyecto y las conclusiones obtenidas a lo largo de todo el proceso.

4.1. Objetivos marcados

En este proyecto se han alcanzado todos los objetivos marcados y algunos adicionales, que son los siguientes:

- ✓ Realización de la planificación y el seguimiento dentro de unos márgenes aceptables en tiempo, coste y esfuerzo.
- ✓ Se emplearon las tecnologías Spring, Tapestry 5, Hibernate, MySQL y Maven.
- ✓ Se ha conseguido analizar, diseñar, implementar y probar una aplicación web cuya función es la venta y gestión de una tienda online de venta de ropa, y es resultado plenamente funcional para su empleo real.
- ✓ En cada fase iterativa se llevaron a cabo las labores de análisis, diseño, implementación y pruebas correspondientes. Esta división del desarrollo mediante iteraciones facilitó en gran medida la reunión de requisitos del usuario, se hizo más fácil la realización de cada una de las fases gracias al paradigma de "Divide y vencerás".
- ✓ Se ha reducido el impacto de cambios en las especificaciones de requisitos del cliente gracias a la metodología empleada.

- ✓ El diseño se efectuó usando patrones y estándares (documentados con el uso de diagramas UML), destacando en dificultad el diseño e implementación de la vista con Tapestry, ya que es un framework que no está muy extendido y por lo tanto no se encontró documentación técnica oficial extendida. Destacar también la compatibilidad de la aplicación con cualquier otro sistema gestor de base de datos. Bastaría con cambiar el URI, el class driver y actualizar correctamente las sentencias de creación de tablas.
- ✓ Se han superado diversas dificultades de diseño, integración e implementación a lo largo del desarrollo gracias a documentación indicada en la bibliografía, y documentación obtenida en internet.

Por lo tanto podemos afirmar que se ha adquirido experiencia en la resolución de problemas no estructurados. Se han empleado múltiples tecnologías, Eclipse me ha facilitado el trabajo de desarrollo, las funcionalidades y la potencia de MySQL Workbench me ayudó en gran medida con la creación y edición de bases de datos

De los resultados obtenidos concluir que el conjunto de funcionalidades proporcionadas por la aplicación cubren perfectamente todas las funcionalidades básicas ofrecidas por otras aplicaciones similares de comercio online.

4.2. Experiencia obtenida

La elección de tecnologías utilizadas y el conjunto de tareas de planificación, análisis, diseño y programación que se han tenido que llevar a cabo a lo largo de este proyecto me han proporcionado una experiencia muy útil y muy valorada hoy en día por muchas empresas en el ámbito del desarrollo de aplicaciones, más concretamente en aplicaciones web de comercio electrónico, que era uno de los objetivos principales que quería alcanzar con este proyecto.

En particular, he obtenido una experiencia muy satisfactoria a la hora de integrar diversas tecnologías que hicieron posible el funcionamiento de una tienda online. Se ha obtenido una experiencia muy valorada hoy en día en algunas tecnologías como Java, Spring, Hibernate, Maven, AJAX, Tapestry, etc. que era para mí el objetivo primordial del proyecto de fin de carrera.

5. Bibliografía

Alur, D., Crupi, J. & Malks, D. (2003). *Core J2EE patterns : best practices and design strategies*. Upper Saddle River, NJ: Prentice Hall PTR.

Bauer, C., King, G. & Gregory, G. (2016). *Java persistence with Hibernate*. Shelter Island, NY: Manning Publications.

DuBois, P. (2013). *MySQL*. Upper Saddle River, NJ: Addison-Wesley.

Eckel, B. (2007). *Piensa en Java*. Madrid: Pearson Prentice Hall.

Ka, I. (2005). *Enjoying web development with Tapestry*. Place of publication not identified: TipTec Development.

Walls, C. (2015). *Spring*. Madrid: Anaya Multimedia.

Hibernate. <http://www.hibernate.org>

Tapestry. <http://tapestry.apache.org/>

Spring, <http://www.springsource.org>

JQuery, <https://jquery.com/>

Junit, <http://junit.org/>

Apache Jmeter, <http://jmeter.apache.org/>

Eclipse. <http://www.eclipse.org>

Maven. <https://maven.apache.org/>

Apache Tomcat. <http://tomcat.apache.org/>

MySQL. <http://www.mysql.com>

OpenProj. <http://openproj.org/openproj>

BootStrap, <http://getbootstrap.com/>

6. Anexos

6.1. Instalación.

Lo primero que deberemos hacer es configurar una variable de entorno con el nombre `JAVA_HOME` y en valor, el directorio de instalación de nuestra distribución J2SE.

Además, se necesita añadir a la variable de entorno `PATH` la ruta hacia la carpeta `bin` del directorio de instalación de Maven.

Una vez realizadas estas dos tareas, hay que dirigirse al directorio de la aplicación.

A continuación se comentan las instrucciones básicas de Maven para las diferentes opciones de compilación.

- Creación del war de la aplicación, ejecución automatizada de pruebas e instalación en el repositorio maven local de los componentes generados.

`mvn package`

- Creación una base de datos llamada `tiendaOnline` y un usuario con permisos sobre ella también llamado `tiendaOnline`:

-

```
$mysqladmin create tiendaOnline -uroot -p
```

```
$mysql -uroot -p root
```

```
mysql> CREATE USER tiendaOnline IDENTIFIED BY tiendaOnline;
```

```
mysql> GRANT ALL PRIVILEGES ON tiendaOnline to  
tiendaOnline@localhost IDENTIFIED BY tiendaOnline;
```

Ejecutar los scripts de base de datos facilitados, para crear las tablas correspondientes y que insertan los datos de ejemplo de la aplicación:

- `mvn sql:execute`

A continuación ejecutar los siguientes comandos maven para instalar correctamente el proyecto en eclipse:

1. `mvn clean`
2. `mvn install`
3. `mvn package`
4. `mvn eclipse:clean`
5. `mvn eclipse:eclipse`

Para ejecutar el proyecto con jetty ir a la carpeta donde se aloja el proyecto y ejecutar el siguiente comando:

```
mvn jetty:run
```

A continuación, acceder con un navegador a la aplicación.

Si queremos ejecutar la aplicación en un servidor Tomcat lo primero que tenemos que hacer es disponer del .war de la aplicación, y copiarlo en la carpeta webapps (que se encuentra dentro de la carpeta de instalación de nuestro Tomcat).

Después, desde una consola acceder a la carpeta bin de nuestro Tomcat y arrancarlo.

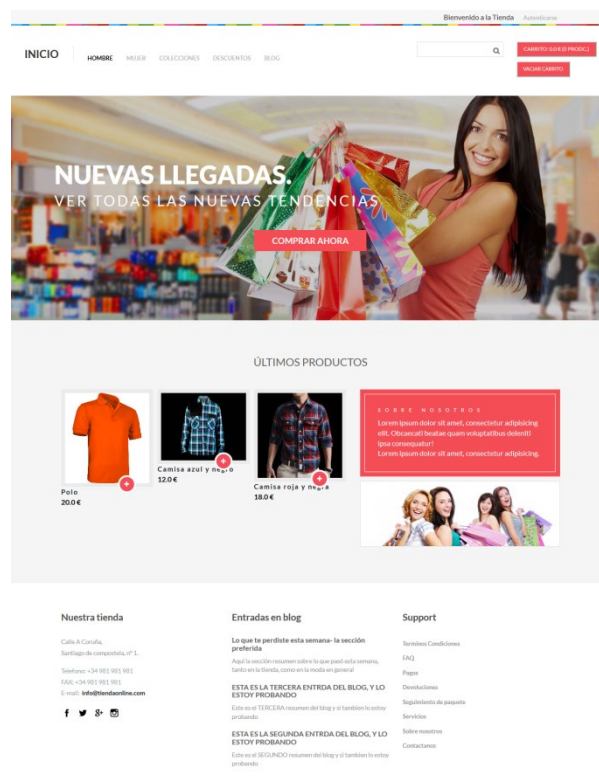
Por último a través del navegador ya podremos acceder a la aplicación.

6.2. Manual de usuario

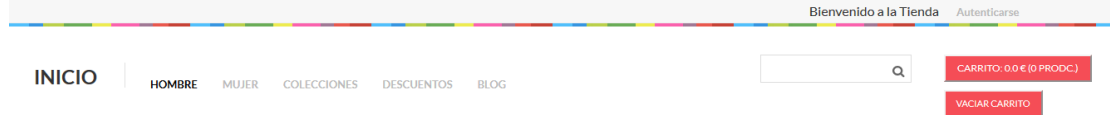
En esta sección se explicará cómo deberán realizar los usuarios el proceso de compra y como deben gestionar los administradores la aplicación.

En el siguiente texto se explicarán los diferentes pasos que tiene que realizar el usuario para comprar productos a través de la aplicación. Se explica paso por paso y mostrando lo que se muestra por pantalla.

1. Lo primero que el usuario deberá hacer es acceder a la aplicación a través de su navegador web preferido. La aplicación está optimizada para trabajar a través de Internet Explorer 11 (o posteriores), Google Chrome, Mozilla Firefox o Safari. Para otros navegadores no se asegura una total compatibilidad de sus funciones. En este caso de ejemplo se usó la versión 46.0.1 de Mozilla Firefox. Una vez elegido el navegador, el usuario tendrá que escribir la dirección web de la aplicación y accederá a la página principal.



A continuación se detallan las diferentes secciones de la página principal:



En la zona de más arriba podemos ver el menú de la tienda. En ella están las diferentes categorías y etiquetas de la tienda, además del acceso al blog propio de la aplicación.

Las categorías se dividen en 2 categorías padres (hombre y mujer) y las etiquetas también en 2, Colecciones y descuentos. Las etiquetas con colecciones no disponen de descuento, pero las de descuentos sí que disponen de un descuento (en porcentaje).

A continuación tiene un buscador por palabra. Dicho buscador busca en toda la tienda, si el texto introducido está en un título de una Ropa o en su descripción.

Después del buscador, el usuario puede ver el carrito que tiene actualmente. Nada más entrar en la tienda está vacía, esto son dos botones, uno para acceder al carrito y otro para vaciarlo.

Por último, arriba a la derecha hay un enlace “Autenticarse” que lleva al usuario a la página de login de la aplicación.



En esta sección, tenemos un botón para acceder a toda la ropa de la tienda, ordenada según la fecha en que fue añadida a la misma.

Además de un ejemplo con los 3 últimos productos añadidos.

Nuestra tienda

Calle A Coruña,
Santiago de compostela, nº 1.

Telefono: +34 981 981 981
FAX: +34 981 981 981
E-mail: info@tiendaonline.com



Entradas en blog

Lo que te perdiste esta semana- la sección preferida

Aquí la sección resumen sobre lo que pasó esta semana, tanto en la tienda, como en la moda en general

ESTA ES LA TERCERA ENTRADA DEL BLOG, Y LO ESTOY PROBANDO

Este es el TERCERA resumen del blog y si tambien lo estoy probando

ESTA ES LA SEGUNDA ENTRADA DEL BLOG, Y LO ESTOY PROBANDO

Este es el SEGUNDO resumen del blog y si tambien lo estoy probando

Soporte

Terminos Condiciones

FAQ

Pagos

Devoluciones

Seguimiento de paquete

Servicios

Sobre nosotros

Contactanos

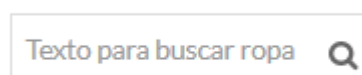
En esta última sección (que se divide en 3 apartados) tenemos la información de la tienda, dirección fiscal, teléfono fax, email y posible acceso a redes sociales de la tienda. Una sección con las 3 últimas entradas al blog propio de la tienda, y una sección de Soporte, donde el usuario podrá encontrar diferentes enlaces a páginas de información o ayuda en sus compras.

2. Lo primero que tiene que hacer un usuario para comprar, es buscar o seleccionar la ropa que quiera añadir a su carrito, para lo que tiene varias opciones:

- a. Navegar a través de las categorías del menú, como se puede ver en la siguiente imagen.



- b. Buscando a través del buscador. Buscando en toda la ropa de la tienda:

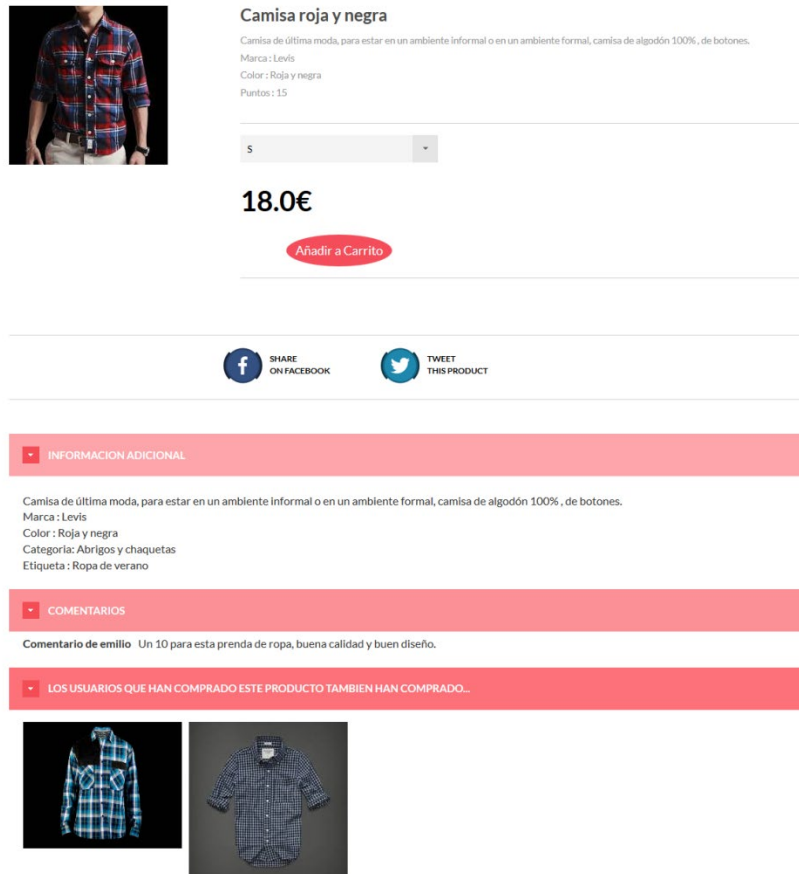


O buscando y filtrando por categoría:

BUSQUEDA POR CATEGORIA

Texto a buscar Categoría

3. Una vez seleccionada la ropa que quiere comprar, entrará en la página donde se detalla dicha prenda de ropa:



The screenshot shows a product page for a 'Camisa roja y negra' (Red and black shirt). It includes a product image, a size selector set to 'S', a price of 18.0€, and an 'Añadir a Carrito' button. Below the product are social sharing options for Facebook and Twitter. The page also features three expandable sections: 'INFORMACION ADICIONAL' with product details, 'COMENTARIOS' with a user comment, and 'LOS USUARIOS QUE HAN COMPRADO ESTE PRODUCTO TAMBIEN HAN COMPRADO...' with two recommended product images.

Camisa roja y negra
Camisa de última moda, para estar en un ambiente informal o en un ambiente formal, camisa de algodón 100%, de botones.
Marca : Levis
Color : Roja y negra
Puntos : 15

S

18.0€

Añadir a Carrito

SHARE ON FACEBOOK TWEET THIS PRODUCT

INFORMACION ADICIONAL

Camisa de última moda, para estar en un ambiente informal o en un ambiente formal, camisa de algodón 100%, de botones.
Marca : Levis
Color : Roja y negra
Categoría: Abrigos y chaquetas
Etiqueta : Ropa de verano

COMENTARIOS

Comentario de emilio Un 10 para esta prenda de ropa, buena calidad y buen diseño.

LOS USUARIOS QUE HAN COMPRADO ESTE PRODUCTO TAMBIEN HAN COMPRADO...

En esta página además de ver la información relevante de la ropa que hemos seleccionado también podemos ver los comentarios de otros usuarios, así como Auto-recomendaciones que nos hace la aplicación. Estas recomendaciones son los otros productos de la tienda que más han sido comprados con la prenda seleccionada (dentro del mismo pedido).

Para añadir ropa al carrito el usuario primero tendrá que seleccionar la talla que desea comprar y luego pulsar sobre “Añadir a Carrito”.

S

18.0€

Añadir a Carrito

4. Una vez creado su carrito, el usuario deberá pulsar en el botón

CARRITO: 18.0 € (1 PRODC.)

que está situado arriba a la derecha en todas las páginas de la aplicación.

Así accederá a la página del carrito:

Mi Carrito

Foto	Nombre	Cantidad	Precio
	Camisa roja y negra	<input type="text" value="1"/> 	18.0 €

Para tramitar este carrito debe [Iniciar - sesion](#) o [registrarse](#)

En esta página se verá el carrito. Cuando el usuario no está autenticado, para poder tramitarlo y conseguir finalizar el pedido el usuario deberá registrarse, para ello puede pulsar en Iniciar – sesión (si ya tiene usuario) o en registrarse, para crear uno nuevo.

a. Si ya tiene usuario, pulsar en Iniciar – Sesión se irá a la siguiente página:

AUTENTICACIÓN

Identificador de usuario *

Nombre de usuario

Contraseña *

•••

Recordar contraseña (las cookies deben estar habilitadas)

ENTRAR

REGISTRARSE (USUARIO NO REGISTRADO)

CREAR CUENTA

Donde el usuario deberá escribir su nombre de usuario y su contraseña.

b. Si no tiene usuario deberá pulsar en Registrarse e irá a la siguiente página.

REGISTRO DE USUARIOS



Identificador usuario	<input type="text"/>	Fecha de nacimiento	<input type="text"/>
Contraseña	<input type="password"/>	Localidad	<input type="text"/>
Confirmar contraseña	<input type="password"/>	Código Postal	<input type="text"/>
Nombre	<input type="text"/>	Provincia	<input type="text"/>
Apellidos	<input type="text"/>	Dirección	<input type="text"/>
Email	<input type="text"/>	Piso	<input type="text"/>
DNI	<input type="text"/>	Número	<input type="text"/>
Teléfono	<input type="text"/>		

REGISTRAR

Donde el usuario no autenticado podrá crear un usuario para tramitar sus pedidos.

5. Una vez autenticado, el usuario verá su carrito de la siguiente forma:

Mi Carrito

Foto	Nombre	Cantidad	Precio
	Camisa roja y negra	<input type="text"/> 	18,0 €

QUE QUIERES HACER CON LOS PUNTOS?

Acumular Descontar

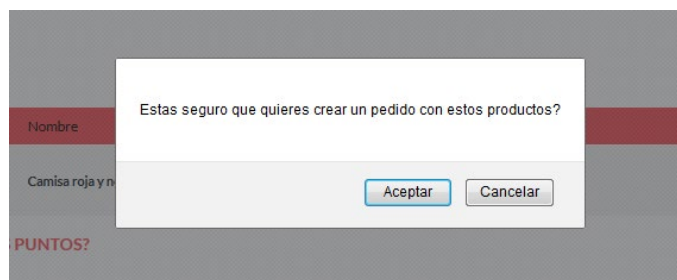
Vas a acumular 15 puntos en tu cuenta.

BORRAR CARRITO

TRAMITAR PEDIDO

En esta página se puede modificar la cantidad de unidades de cada ropa comprada, si se pone a 0 se borra dicha prenda del carrito.


A su vez, se le pregunta al usuario que desea hacer con sus puntos, si desea acumular los puntos de los que dispone en el carrito, o si prefiere descontar todos sus puntos más los del carrito de este pedido. Si pulsa en Tramitar Pedido, pasará a crear un pedido con su carrito, además se le pedirá un mensaje de confirmación como el siguiente:



Una vez aceptado, su carrito pasará a ser un Pedido en estado Creado.

6. Una vez creado el pedido, se verá la siguiente información:

Datos de pedido creado

Foto	Cantidad	Precio
	1	18.0 €

EL PRECIO FINAL DEL PEDIDO ES: 7.0 EUROS

ESTOS SON LOS DATOS A LOS QUE SE LE ENVIARÁ EL PEDIDO :

Nombre: Admin123 Dominguez
Correo electronico: emilio89@gmail.com
Telefono: 981861593
Dni: 9999999s
Provincia: A Coruña
Localidad: Santiago de Compostela
Numero: 10
Tus Puntos ahora son: 0

BORRAR PEDIDO

TRAMITAR PEDIDO

Si se pulsa en borrar pedido, el pedido se pondrá en estado borrado. En cambio sí se tramita el pedido, y se acepta el mensaje de confirmación que sale a continuación, se cambiará el estado de este a tramitación y se enviará un correo al correo electrónico del usuario con toda la información de dicho pedido.

En dicho correo se puede ver toda la información del pedido que se acaba de tramitar.

- Una vez que un administrador de la tienda vea ese pedido y la disponibilidad del stock (si hay stock) enviará ese pedido (que pasará a estado enviado) y al usuario le llegará un correo electrónico con el número de seguimiento del paquete del pedido y una fecha aproximada de llegada.
- El usuario podría ver siempre el estado de sus pedidos pulsando en la sección Mis pedidos que está arriba en todas las páginas de la tienda online.

Lista de sus pedidos

Filtrar pedidos por estado: **Tramitación** Creado Enviado Borrado

Esto son los pedidos que estan en estado de tramitacion, en breve nosotros lo atenderemos y lo marcaremos como enviado

Fecha Pedido	Precio Total	Estado	Ver Productos
04-jun-2016	7.0	Tramitación	Ver

6.3. Manual del usuario administrador

El usuario administrador es el encargado de darle contenido a la tienda, así como de gestionar los pedidos. Para ello tendrá que ir a la página principal de administración a la que se accede añadiendo “/administración” a la dirección de la página principal de la tienda. Si el usuario tiene los permisos adecuados (permisos de administración) se irá a la siguiente página.

PÁGINA DE ADMINISTRACIÓN

En el siguiente menú estarán disponibles todas las opciones para administrar la tienda

ROPA CATEGORIA ETIQUETAS VER PEDIDOS GESTION DE USUARIOS BLOG GESTION CORREOS

En la sección de ropa, el administrador podrá registrar nueva ropa así como actualizar información de la ya existente.



Si pulsa sobre registrar ropa, se le mostrará la siguiente página.

REGISTRO DE ROPA

Nombre	<input type="text"/>	Descripción	<input type="text"/>
Precio	<input type="text" value="0"/>	Selecciona Categoría	<input type="text" value="Abrigos y chaquetas"/>
Añadir imagen	<input type="button" value="Examinar..."/> No se ha sele...	Color	<input type="text"/>
Marca	<input type="text"/>	Numero de tallas:	<input type="text" value="1"/>
Indique el stock de cada una de las tallas:			
Nombre Talla - 1	<input type="text" value="Talla 1"/>	Stock Talla - 1	<input type="text" value="0"/>

Añadiendo los campos que se indican, el administrador creará una nueva prenda de ropa en la tienda. Además podrá registrar todas las tallas que le lleguen seleccionando en el campo número de tallas, el número de las diferentes tallas que tenga.

Si pulsa en actualizar ropa, el administrador verá lo siguiente.

Nombre	Precio	Fecha Registro	Modificar	Stock	Etiqueta	Modificar Stock	Borrar
Polo	20.0	10-abr-2016	Modificar	Añadir Talla	Asignar Etiqueta	Modificar Stock	Borrar Ropa
Camisa azul y negro	12.0	29-may-2016	Modificar	Añadir Talla	Asignar Etiqueta	Modificar Stock	Borrar Ropa
Camisa roja y negra	18.0	30-may-2016	Modificar	Añadir Talla	Asignar Etiqueta	Modificar Stock	Borrar Ropa
Camisa de cuadros pequeños	28.0	30-may-2016	Modificar	Añadir Talla	Asignar Etiqueta	Modificar Stock	Borrar Ropa

Donde podrá modificar la ropa existente (sus atributos), añadir tallas, asignarle etiquetas, modificar el stock o borrar dicha ropa. Hay que tener en cuenta que

borrar una ropa borraría todas las dependencias de esta tallas, comentarios, recomendaciones etc.

En el menú de categoría el administrador podrá crear o modificar las categorías de la tienda. Análogamente en el menú de etiquetas el administrador también creará o modificará las etiquetas de la aplicación, asignándole a las que crea conveniente un descuento.

En la sección de Ver Pedidos, el administrador podrá ver todos los pedidos realizados en la aplicación, tal que así:

Lista de pedidos

Filtrar pedidos por estado: **Tramitación** Creado Enviado Borrado Sin Stock


Id Pedido	Fecha Pedido	Precio Total	Estado	Ver Productos
7	04-jun-2016	7.0	Tramitación	Ver

Por cada estado, al pulsar en Ver Productos, se verán unas acciones u otras, pasamos a continuación a detallarlas.

Si pulsa en Ver Productos en un pedido en estado en Tramitación, se verá lo siguiente:

Lista de pedidos

Este pedido esta en estado: **Tramitación**

Foto	Nombre	Cantidad	Precio
	Camisa roja y negra	1	18.0 €

El usuario ha usado los puntos de su cuenta para obtener un descuento.

El precio de este pedido es de, 7.0 euros


COMPROBAR STOCK

En comprobar stock, la aplicación comprobará el stock de las prendas del pedido. Si hay suficientes se pasará el pedido a estado enviado, si no se pasará a estado Sin stock.

Si se cambia el estado del pedido a enviado, se pasará a la pantalla siguiente:

Lista de pedidos

Este pedido esta en estado: **Enviado**

Foto	Nombre	Cantidad	Precio
	Camisa roja y negra	1	18.0 €

El usuario ha usado los puntos de su cuenta para obtener un descuento.

El precio de este pedido es de, 7.0 euros

Si se ha hecho ya, porfavor envíele al usuario de este pedido, la fecha aproximada de recepcion y el Numero de seguimiento del paquete.

Fecha de recepcion aprox. :

Numero de seguimiento


ENVIAR CORREO

Donde el administrador le puede enviar un correo al usuario que realizó el pedido, con la fecha aproximada de recepción y con el número de seguimiento de la compañía que lo va a realizar.

En caso de que el pedido no tenga stock, la página le redirige a la siguiente página:

Lista de pedidos

Este pedido esta en estado: **Camisa azul y negro SIN STOCK**

Foto	Nombre	Cantidad	Precio
	Camisa azul y negro	1	12.0 €

El precio de este pedido es de, 12.0 euros

Cuando vuelva a tener stock de dicho producto el administrador deberá registrarlo en la página de Actualizar ropa y volver a enviar este pedido.

A su vez en la página de ver pedidos, el administrador también podrá Ver los pedidos, en estado creados (y no tramitados por el usuario), los enviados (y ya finalizados) y los borrados por los usuarios.

En la página de gestión de usuarios, el administrador podrá gestionar algunas acciones sobre los usuarios registrados en la aplicación, para ello tendrá que ir a la siguiente página.

Lista de usuarios

Identificador de usuario	Nombre	Apellidos	Dirección correo-e	Tipo Usuario	Numero Puntos	Regalar Puntos	Dar Privilegios	Quitar Privilegios	Mandar Correo
emilio	Admin123	Dominguez	emilio89@gmail.com	Administrador	0	Regalar Puntos	Dar Privilegios	Quitar Privilegios	Mandar Correo
emilio2	Emilio	Dom	emilio@mai.com		60	Regalar Puntos	Dar Privilegios	Quitar Privilegios	Mandar Correo
emilioEntero	Emilio	Dominguez	emilio@si.com		0	Regalar Puntos	Dar Privilegios	Quitar Privilegios	Mandar Correo
111	qwef	2345	wert@werg.ee		0	Regalar Puntos	Dar Privilegios	Quitar Privilegios	Mandar Correo
emilio89	Emilio	Dominguez Sanchez	emilio89@gmail.com		23	Regalar Puntos	Dar Privilegios	Quitar Privilegios	Mandar Correo
tania	tania	dom	taniadominguez11@gmail.com		15	Regalar Puntos	Dar Privilegios	Quitar Privilegios	Mandar Correo

Las opciones que el administrador tiene sobre los usuarios son, Regalar puntos, Dar o quitar privilegios o Mandar Correo.

En la opción de Regalar puntos, el administrador podrá regalarle puntos a un usuario para así incentivarlo a que compre en la página, ya que después esos puntos los podrá canjear por descuento cuando el administrador le regale puntos a un usuario. Este usuario recibirá un correo electrónico con los puntos que se le han regalado.

Las opciones de dar o quitar privilegios son para dar o quitar privilegios de administrador, es decir, si a un usuario se le da los privilegios de administrador pasará a ser un administrador más de la tienda.

Además también se le puede mandar un correo al usuario a través de la aplicación.

En la sección Blog el usuario podrá gestionar las entradas del blog, añadiendo una nueva o modificando o borrando una existente.

Por último en la sección de Gestión Correos el administrador gestionará los correos que se mandan automáticamente a los usuarios, las instrucciones para hacer estas modificaciones están en cada uno de los correos a modificar, ya que dichos correos envían datos a los usuarios.