Diseño e implementación de una tienda electrónica de ropa

Autor: Emilio Domínguez Sánchez
Máster de Ingeniería Informática
Desarrollo de aplicaciones web.
Tutor: Ignasi Lorente Puchades / César Pablo Córcoles Briongos.

Índice de la presentación

INTRODUCCIÓN

“Vender a través de Internet es poder vender a todo el mundo a todas horas”

CONTEXTUALIZACIÓN:

- El e-commerce en el sector de la moda movió ya más 20.000 millones de dólares en Europa anuales.
- Todas las grandes firmas como Zara, Desigual, Mango ya venden en internet.
- En España, más de 4 millones de personas realizaron compras de ropa, de calzado y de accesorios de vestir a través de Internet.

SOLUCIONES PARA UNA TIENDA DE ROPA PEQUEÑA

Actualmente para que una tienda de ropa pequeña pueda vender sus productos a través de Internet tiene dos opciones :

1 – Usar un content management system (CMS) prefijado con muchas opciones que no se adaptan a su negocio. Como puede ser prestashop o eCommerce. Las cuales para muchas tiendas pequeñas es más engorro que una verdadera solución.

2 – Pagar mucho dinero a una consultora por una solución a su medida que cubra sus necesidades básicas y realice la misma funcionaldiad que realizan en la tienda física.

Objetivos del proyecto:

✓ **APLICACIÓN WEB QUE ACERQUE LOS PRODUCTOS DE UNA TIENDA DE MODA PEQUEÑA A LOS USUARIOS DE INTERNET**

El objetivo principal del proyecto es que cualquier tienda de moda que venda sus productos físicamente pueda hacerlo también a través de internet.

✓ **DAR A CONOCER LA TIENDA EN TODO EL MUNDO**

Que cualquier tienda pueda darse a conocer en todo el mundo.

✓ **POSIBILIDAD QUE EL CLIENTE PUEDA COMPRAR DESDE CUALQUIER PARTE DEL MUNDO, SIN PERSONARSE EN LA TIENDA**

Con esta tienda damos la posibilidad a los clientes que puedan comprar en una tienda física a través de cualquier parte del mundo y las 24 horas del día.

✓ **ADAPTABILIDAD DEL PROYECTO A CUALQUIER TIENDA**

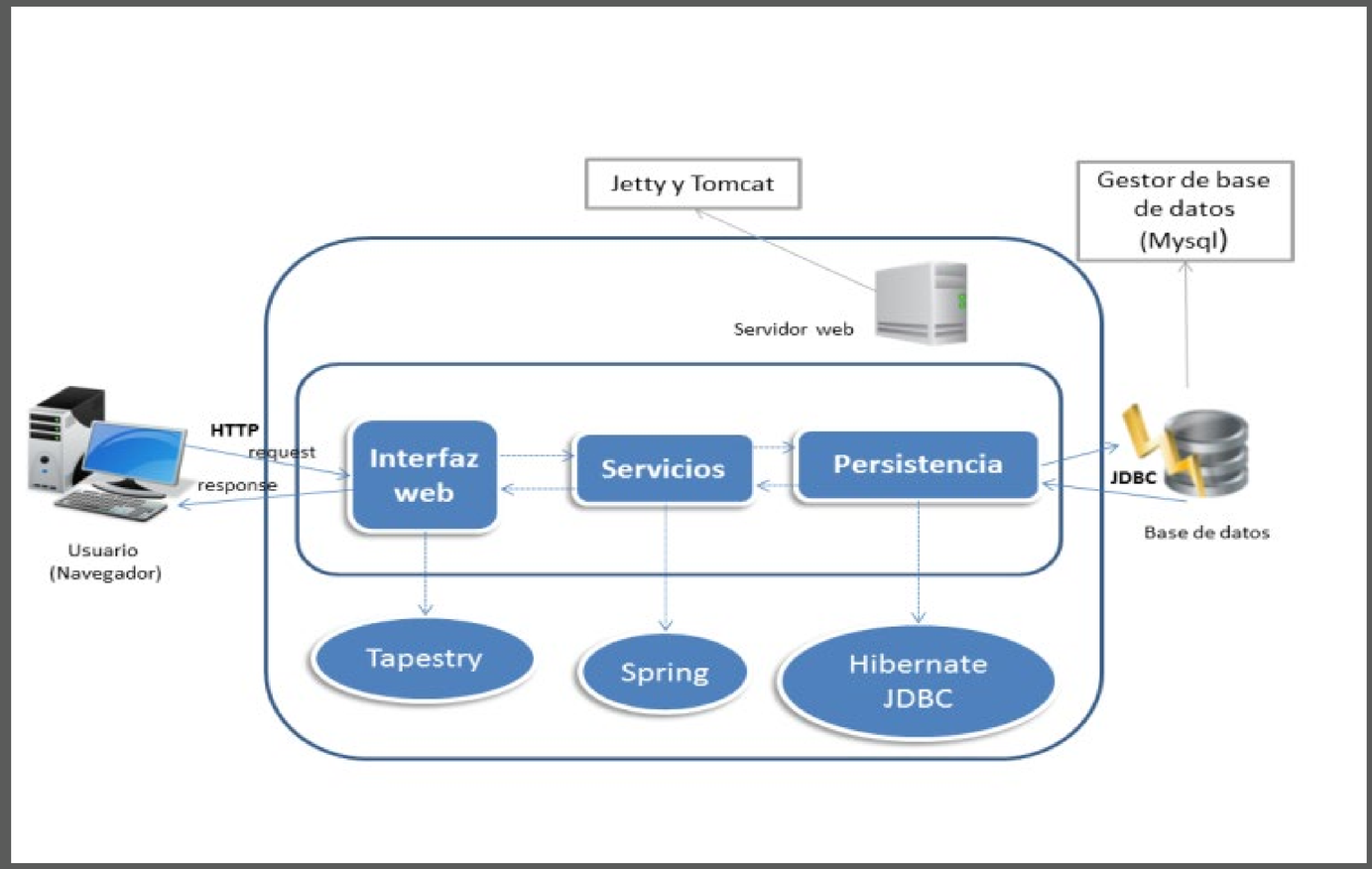
Al no haber sido diseñada para una tienda en concreto, este proyecto puede ser adaptado con sencillos pasos de programación a cualquier tienda real.

Funcionalidades de la tienda

- ✓ VISUALIZAR LOS PRODUCTOS DE UNA TIENDA DE ROPA, POR CADA PRODUCTO SE MOSTRARÁ SU NOMBRE, DESCRIPCIÓN, MARCA, PRECIO Y SU FOTO.
- ✓ PERMITIR CATEGORIZAR LOS PRODUCTOS
- ✓ PERMITIR REALIZAR BÚSQUEDAS A TRAVÉS DE PALABRAS CLAVE DE LA ROPA OFERTADA.
- ✓ CARRITO DE LA COMPRA EN EL QUE EL USUARIO AÑADIRÁ LOS PRODUCTOS QUE QUIERA COMPRAR.
- ✓ REGISTRO Y LOGUEO DE USUARIOS EN LA APLICACIÓN PARA PODER REALIZAR SUS PEDIDOS.
- ✓ USUARIO ESPECIAL ADMINISTRADOR PARA GESTIONAR EL CONTENIDO DE LA TIENDA.
- ✓ BLOG PROPIO DENTRO DE LA TIENDA PARA COMUNICARSE MEJOR CON LOS COMPRADORES.
- ✓ ENVÍO AUTOMÁTICO DE CORREOS ELECTRÓNICOS DURANTE EL PROCESO DE COMPRA.
- ✓ POSIBILIDAD DE COMPARTIR PRODUCTOS EN LAS REDES SOCIALES
- ✓ AÑADIR COMENTARIOS O DUDAS SOBRE LOS PRODUCTOS.
- ✓ SISTEMA DE FIDELIZACIÓN PROPIO BASADO EN PUNTOS QUE PODRÁN CANJEARSE POR DESCUENTOS EN FUTURAS COMPRAS.
- ✓ SISTEMA DE AUTO-RECOMENDACIONES DE LOS PRODUCTOS DE LA TIENDA EN BASE A COMPRAS PASADAS DE OTROS USUARIOS.
- ✓ DISEÑO AMIGABLE DE TODA LA APLICACIÓN.

TECNOLOGÍAS USADAS PARA EL DESARROLLO DE LA APLICACIÓN

En el gráfico se observa las diferentes tecnologías usadas para el desarrollo de la aplicación.
Desde el motor de base de datos que está realizado con MYSQL.
La persistencia realizada con Hibernate.
Los servicios realizados con Spring
La vista realizada con Tapestry.
Para ejecutar la aplicación se usa como servidor web o de aplicaciones jetty (para desarrollo) y tomcat para su puesto en producción



Desarrollo realizado

Iteraciones llevadas a cabo para la realización del proyecto

Iteración 1
- Base del proyecto, funcionalidades que nos dan soporte a las demás.
- Registro y autenticación de usuarios y casos de uso de administrador.

Iteración 2
- Visualización de productos de la tienda.
- Busqueda por categorías y por palabras clave.

Iteración 3
- Proceso de compra completo.
- Funcionalidades extras como pueden ser comentarios o recomendaciones.

Iteración 4
- Funcionalidades adicionales, como blog o gestión de correos
- Interfaz gráfica completa.

Desarrollo realizado

Resumen de los patrones usados

Patrones arquitectónicos

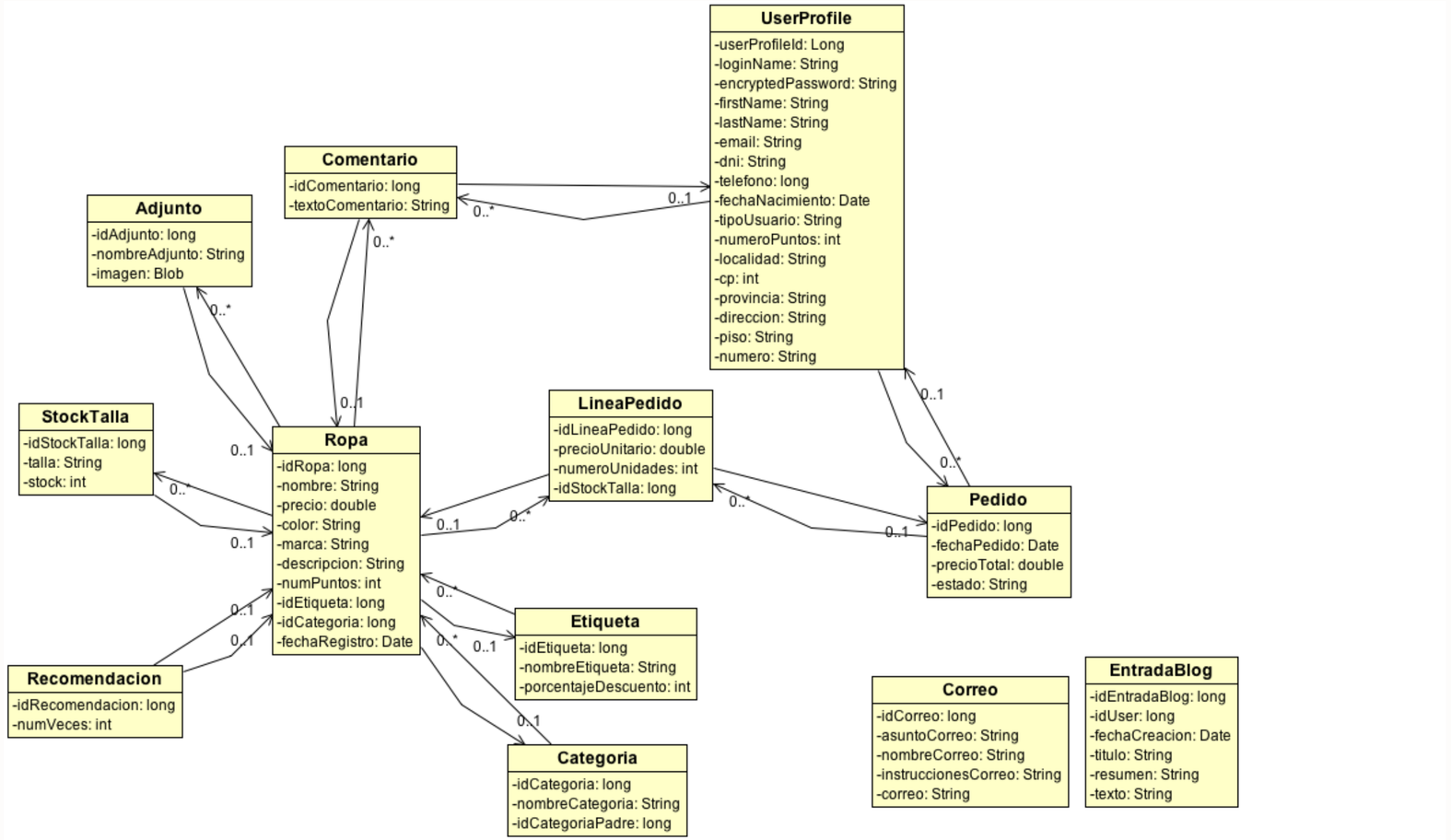
- **Model View Controller** – Patrón que crea separación entre la lógica de negocio (modelo) y la interfaz gráfica (vista). El controlador proporciona la interfaz entrel a vista y el modelo.
- **Patrón Layers** – Aplicación en diferentes capas, esto ofrece separación en la creación del producto software.

Patrones de diseño

- **Value Object (VO)** – Estado objetos de dominio (persistentes). Atributos privados y métodos get y set privados.
- **Data Access Object (DAO)-** Desacopla la lógica de negocio del acceso a la base de datos
- **Singleton –** Restringe la creación de objetos. En nuestra aplicación se usa para la sesión para los DAOs y para los servicios
- **Factory –** Crea instancia concreta de múltiples tipos, normalmente con el mismo padre. Se usa para la creación de todos los beans.
- **Inyección Dependencias** – Nos permite un código más desacoplado. Nos facilitará las cosas a la hora de hacer Tests. Los componentes declaran sus dependencias pero no se encargan de conseguirlas, esto se encarga el Contenedor de Spring

Desarrollo realizado

Modelo de datos de la aplicación



Desarrollo realizado

Pecularidades del modelo de datos

- ✓**Implementado usando el framework de Mapeo-objeto-Relacional Hibernate.**
- ✓**Siguen el dictamen del patrón de diseño Data Access Object.**
- ✓**Uso de la estrategia LAZY, cuando se lanza una consulta Hibernate carga el mínimo de datos posibles, ya que quizás no necesitemos todos.**

Desarrollo realizado

Lógica de Negocio – Se agrupan los métodos en fachadas que agrupan funciones relacionadas, las más importantes son:

AdministradorService

- Recopila las funcionalidades del administrador sobre la aplicación. Se resumen en darle contenido a la aplicación

User Service

- Controla el registro y login del usuario. Usa el paquete útil (passowrdEncrypter).

RopaService

- Son los métodos que nos permiten acceder a la ropa, agrupaciones de estas, a los adjuntos (imágenes) o al stock y talla

PedidoService

- Servicio encargado de todo lo relacionado con un pedido, gestionarlo, cambiarle el estado, calcular el precio o añadir recomendaciones

Desarrollo realizado

Pecularidades de la lógica de negocio

- ✓**Uso de la anotación de Spring @Transactional, para la gestión de las transacciones. Una transacción en base de datos es un conjunto de instrucciones que se usan en bloque. Con la anotación @Transactional si una de ellas produce un error, todo el proceso se echa para atrás.**

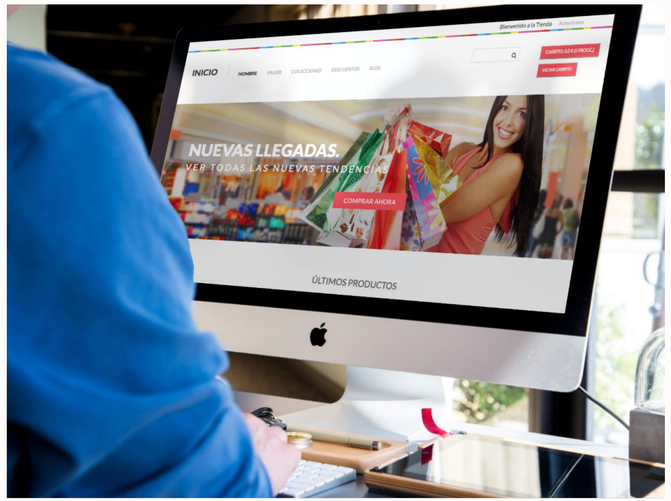
- ✓**También se usa @Transactional (readOnly=True), para métodos donde no queremos escribir en base de datos.**

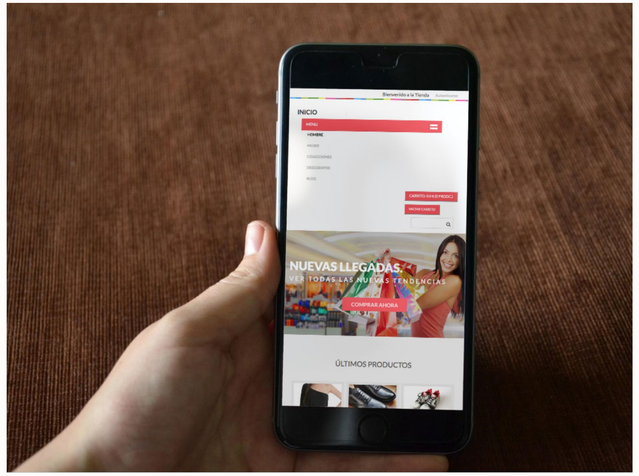
Desarrollo realizado

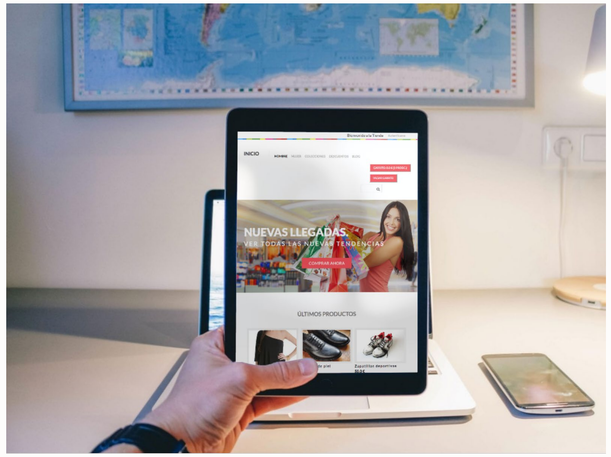
Capa web

- ✓**Uso del framework Tapestry.**
- ✓**Dos archivos por página con el mismo nombre. Un archivo .java y un archivo .tml.**
- ✓**En las cabeceras del archivo .tml se define el layout de la aplicación.**
- ✓**Ayuda de Jquery y Bootstrap para realizar los estilos de la aplicación.**

Aplicación en diferentes dispositivos







Como se puede ver la tienda dispone de una interfaz responsiva que se adapta a diferentes dispositivos y con ella una interfaz sencilla de usar

CONCLUSIONES

- ✓ Se obtiene un resultado totalmente funcional. Cubriendo todas las funcionalidades básicas que podría tener una tienda de moda.
- ✓ Analizar, diseñar e implementar una aplicación web para vender productos de ropa a través de internet
- ✓ Para el diseño se usa patrones y estándares
- ✓ Se han empleado correctamente las tecnologías: Java, Spring, Tapestry 5, Hibernate, MySQL y Maven

TRABAJOS FUTUROS

- ✓ Añadir la implementación del sistema de pago por PayPal para automatizar el pago de productos.
- ✓ Añadir más roles a la aplicación. Dinamizadores, gestores de pedidos etc.
- ✓ Añadir una aplicación móvil a la tienda.
- ✓ Añadir mas ficheros de configuración para facilitar la tarea de implementación en diferentes tiendas.

DEMO DE LA APLICACIÓN

MUCHAS GRACIAS