SDI – Sistemas Distribuidos e Internet

PRÁCTICA 1 DE ENTREGA – SPRING

INFORME

Grupo 104-106

URL AWS	**http://ec2-18-217-40-214.us-east-2.compute.amazonaws.com/**
Nombre1:	Marcos
Apellidos1:	Canal López
Email1:	UO258899@uniovi.es
Cód. ID GIT	104
Nombre1:	Martín
Apellidos1:	Fernández Prieto
Email1:	UO258619@uniovi.es
Cód. ID GIT	106

Índice

Introducción

Desarrollamos una aplicación Web de compra-venta entre particulares (al estilo Wallapop) en el que existen perfiles de usuario de tipo: Público (Anónimo), Usuario Registrado (Administrador y Usuario Estándar). Cada uno de estos tienen distintas funcionalidades para interactuar con la aplicación.

El propósito de la aplicación es poner en contacto a vendedores de productos con usuarios que pueden estar interesados en ellos. Dichos usuarios pueden comunicarse entre sí para resolver cualquier duda que surja. Además, estos podrán comprar o vender productos y su saldo personal se vera modificado en dichos casos.

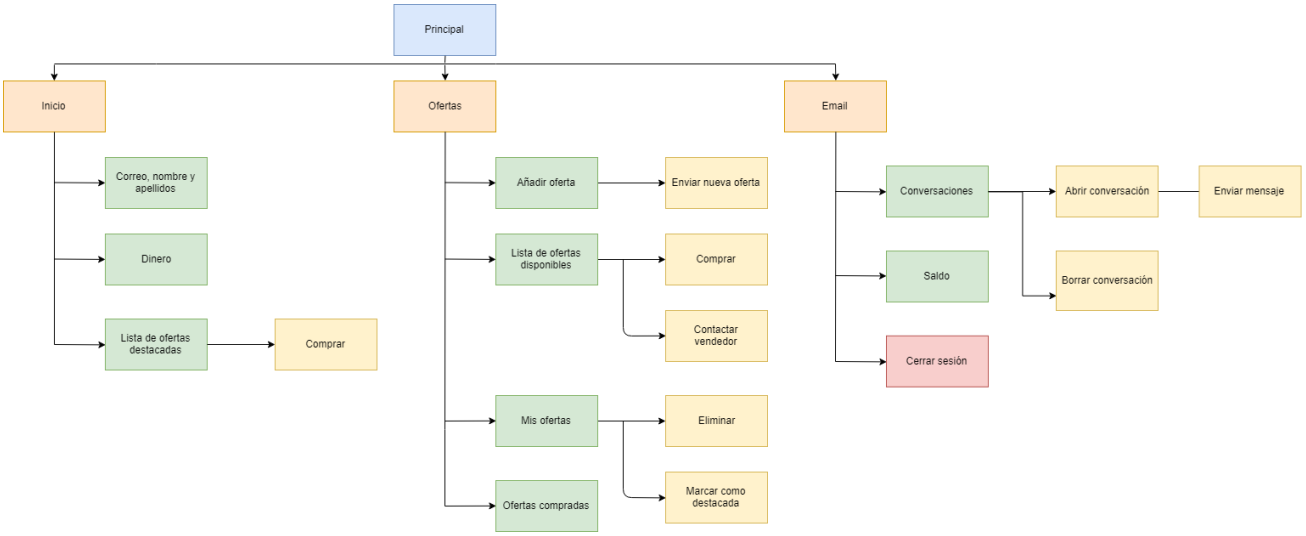
Como esta es una aplicación de prueba, los nuevos usuarios empezaran con un saldo de 100€, para permitirles comprar ofertas y ver su funcionamiento, además de por supuesto poder vender y también recibir el dinero de dicha compra.

Mapa de navegación

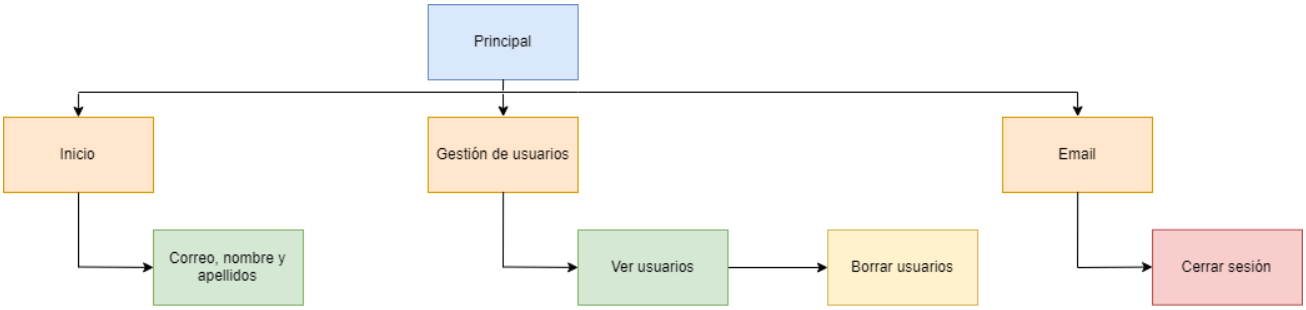
Se muestran a continuación los tres mapas de navegación para los distintos tipos de usuario: Usuario Registrado, Administrador y Usuario Anónimo.

LEYENDA: Azul (pagina principal de la aplicación), Naranja (Elementos del Nav), Verdes (Desplegables del nav o elementos de la vista), Amarillo (Acción) y Rojo (Cierrre de sesión).

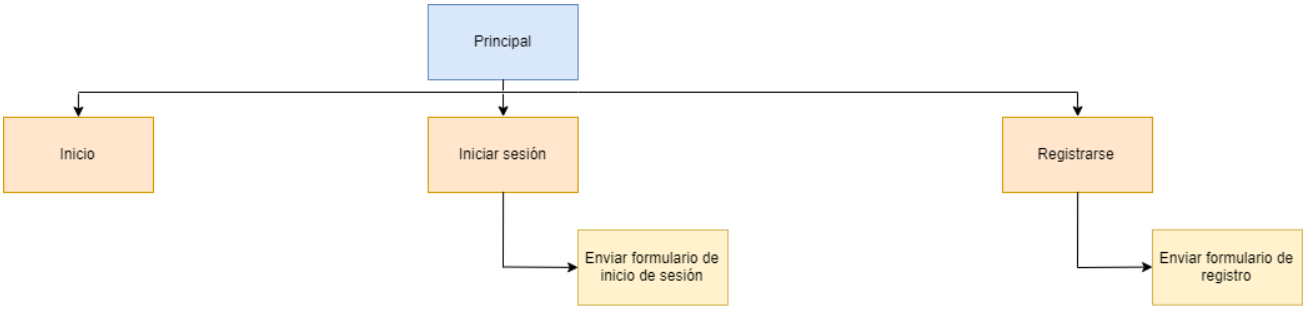
1. Mapa de navegación Usuario Registrado



2. Mapa de navegación de Administrador



3. Mapa de navegación de publico (Anónimo)



Aspectos técnicos y de diseño relevantes

Desarrollamos el proyecto usando Java (Java 8), en el entorno de desarrollo Spring Tool Suite 3.97. Como base de datos usamos HSQLB 2.4.1 y desplegamos la aplicación en un servidor EC2 de AWS (Amazon Web Services) usando el sistema operativo Windows 2019 Server.

Usamos como patrón para la capa de presentación el MVC (Modelo Vista Controlador) visto en clase.

Información necesaria para el despliegue y ejecución

Para desplegar la aplicación se deberá descargar el proyecto de GitHub (https://github.com/martinlacorrona/sdi-entrega1-104-106), una vez descargado se importara en el workspace de Spring Tool Suite (a partir de ahora STS) con la versión 3.9.7 (es con la que funciona, ya que fue en la que se hizo). Una vez que tengamos abierto el proyecto en STS procederemos a abrir la carpeta "hsqldb/bin" y ejecutamos el script "resetDB.bat" (para limpiar la base de datos en caso de que ya se haya abierto), y después ejecutaremos el script "runServer.bat". Una vez abierta la base de datos correctamente procederemos a arrancar el proyecto en STS, "Run As", "Java Aplicattion" y seleccionamos la clase "MyWallapop.java" en caso de que nos lo pida.

Una vez realizado esto la aplicación debería de estar disponible ya en http://localhost:8090.

Para la ejecución de las pruebas podrás ejecutarlas ejecutando la clase como Junit en la carpeta "src/test/java/com/uniovi/tests" llamada "MyWallapopTests".

Las **pruebas solo se pueden ejecutar localmente**. La razón es que las pruebas se hicieron basándose en el PDF sobre los comentarios de la practica en los que hablaba sobre el reinicio de las pruebas antes de la ejecución de cada una. Por lo que **todas las pruebas son independientes** (tal cual como se pidió), lo cual nos produce un **problema** al ejecutarlas **remotamente**. En los test se usa el servicio de usuarios, y de roles, pero del servicio local, es decir, del que usamos si tenemos la app abierta de manera local. Ya que si por ejemplo ejecutamos el método añadir del servicio de usuarios no nos va a funcionar, ya que se guarda en la app local, pero no en la remota, ya que a ese servicio de la remota no tiene acceso. Aunque si pueda interactuar con la interfaz de la remota, no podrá ejecutar las pruebas ya que los servicios que se inyectan son los de la local no los de la remota.

Además, estas no se podrían ejecutar varias veces. Por ejemplo, si borro a pedro una vez, no lo podré borrar más veces, ya que ya estará borrado, debido a que la base de datos de la aplicación remota es persistente.

Aun así, puedes encontrar la Web desplegada en:

http://ec2-18-217-40-214.us-east-2.compute.amazonaws.com. Esta tiene los valores por defecto, pero como está abierta al publico y puede ser modificada.