

Segunda Práctica

Tecnologías Informáticas para la Web

Curso 2021-22

ÍNDICE

INTRODUCCIÓN	3
DESCRIPCIÓN DE LA APLICACIÓN	3
FUNCIONALIDADES DEL USUARIO	4
<i>Darse de alta en la aplicación</i>	4
Modificación de información/perfil de usuario	4
<i>Baja de la aplicación</i>	4
<i>Dar de alta un producto</i>	4
<i>Modificar datos del producto</i>	4
<i>Cambiar estado del producto</i>	4
<i>Dar de baja del producto</i>	5
<i>Realizar consultas de productos en venta</i>	5
Comprar un producto ofertado por otro vendedor	5
FUNCIONALIDADES DEL ADMINISTRADOR	5
<i>Gestionar usuarios: modificar y dar de baja</i>	5
<i>Gestionar productos del catálogo: modificar y dar de baja</i>	5
<i>Consultas con la aplicación financiera</i>	5
Desarrollo y Tecnologías	6
Microservicios	6
Desarrollo del Front-End	8
Bases de Datos	8
Base de Datos MySQL	9
Base de Datos MongoDB	9
NORMATIVA DE ENTREGA	10
CRITERIOS DE EVALUACIÓN	11
GESTIÓN DEL PROYECTO	11
ANEXO I	12
Anexo II	12

Introducción

El objetivo principal de la segunda práctica es demostrar y desarrollar todas las habilidades y conocimientos que se obtendrán en las clases magistrales y prácticas de la segunda mitad de la asignatura.

Para ello, se solicita a los alumnos desarrollar una Aplicación Web mediante una arquitectura basada en Microservicios usando el Framework Spring.

La temática de la Aplicación Web será la compra/venta de artículos de segunda mano.

En aras de conocer cuál debe ser el resultado que se ha de lograr en la práctica, los alumnos pueden tomar como referencia la aplicación **Wallapop** (o *vibbo en menor medida*).

No será necesario el desarrollo de un front para móvil, la aplicación será exclusivamente para Web.

Descripción de la Aplicación

Como se ha indicado anteriormente, el software a desarrollar es una aplicación web que permita la compra/venta de artículos de segunda mano.

Las funcionalidades principales del usuario/cliente serán:

- Darse de alta en la aplicación
- Modificación de su información/perfil de usuario
- Baja de la aplicación
- Dar de alta un producto
- Modificar datos de producto
- Cambiar estado del producto
- Dar de baja el producto
- Realizar consultas de productos en venta
- Comprar un producto ofertado por otro vendedor

Por otra parte, y con el fin de garantizar que los procesos de compra y venta se adecuen a la política de la aplicación, se implementarán las siguientes funcionalidades del administrador:

- Gestionar usuarios: modificar y dar de baja
- Gestionar productos del catálogo: modificar y dar de baja
- Comprobar pagos realizados

Los detalles de cada una de las funcionalidades se especifican a continuación:

Funcionalidades del usuario

Darse de alta en la aplicación

La aplicación deberá permitir a nuevos usuarios poder darse de alta ellos mismos en la plataforma para la compra/venta de productos. Para la creación de un usuario la aplicación dispondrá de un formulario de registro que incluirá como mínimo los siguientes campos:

- Correo electrónico del usuario
- Contraseña para el inicio de sesión
- Nombre real del usuario
- Apellidos reales del usuario
- Ciudad de residencia

Modificación de información/perfil de usuario

Una vez que el usuario haya iniciado sesión en la aplicación podrá actualizar en cualquier momento su información de perfil.

Baja de la aplicación

Los usuarios tendrán la posibilidad de darse de baja de la aplicación. El darse de baja de la aplicación implica la eliminación de todos los artículos que el usuario tenga en venta en ese momento.

Dar de alta un producto

Los usuarios tendrán la posibilidad de dar de alta en la aplicación productos de segunda mano para su venta. Para dar de alta un nuevo producto el usuario deberá de rellenar un formulario con la siguiente información mínima:

- Título del producto
- Categoría del producto, a seleccionar entre un conjunto de categorías cerradas
- Descripción del producto (con una longitud máxima de 500 caracteres)
- Imagen/foto del producto
- Precio de venta

Al dar de alta un producto le será asignado un estado inicial de “Disponible”.

Modificar datos del producto

En cualquier momento los usuarios deberán poder modificar todos los datos de sus productos en venta.

Cambiar estado del producto

Los usuarios tendrán la posibilidad de modificar el estado de sus productos en venta, señalándoles como reservados o vendidos, o volviendo a hacerlos disponibles si la venta se cancela en el último momento, por ejemplo.

Dar de baja del producto

Los usuarios también podrán dar de baja un producto del catálogo a pesar de que no se haya completado el proceso de venta.

Realizar consultas de productos en venta

Solo los usuarios que hayan iniciado sesión en la aplicación podrán realizar consultas en el catálogo de productos disponibles. En la página principal se visualizarán algunos artículos al azar y una pequeña sección con un campo de búsqueda rápida. Esta pequeña sección estará compuesta por un único campo de texto, un botón buscar y un link a búsqueda avanzada. Los usuarios podrán realizar búsquedas rápidas introduciendo su título y pulsando el botón buscar. Por lo tanto, la aplicación mostrará todos los artículos que contengan el término o estructura de términos en el título o la descripción.

Además, se proporcionará la posibilidad de hacer búsqueda avanzadas en las que el usuario podrá filtrar como mínimo por categoría, ciudad, vendedor, título y descripción.

Comprar un producto ofertado por otro vendedor

Los usuarios tienen la opción de realizar el pago de un producto que deseen comprar a través de la propia aplicación. Para ello los productos contarán con un botón de “Compra” que al ser pulsado solicitará al usuario introducir la información de pago (tarjeta bancaria), para a continuación llamar al microservicio de Banco con el fin de simular el pago de la transacción. Si la transacción se ha efectuado de forma correcta se generará un mensaje automático al vendedor para informarle de la compra conteniendo además el código de identificación de la transacción devuelto por el banco. Será su responsabilidad modificar el estatus del producto.

Funcionalidades del administrador

Gestionar usuarios: modificar y dar de baja

El administrador deberá disponer de un registro de usuarios en la aplicación que le permita acceder a sus datos, modificarlos o darlos de baja.

Gestionar productos del catálogo: modificar y dar de baja

De la misma forma el administrador deberá poder consultar los productos en venta, modificarlos o eliminarlos de la misma.

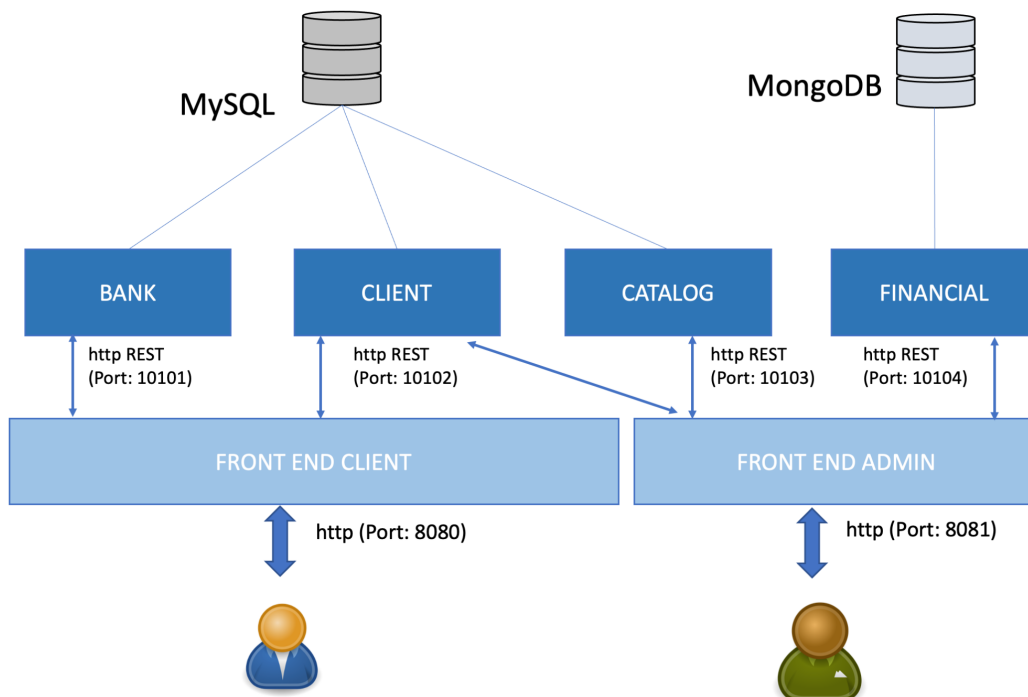
Consultas con la aplicación financiera

El administrador podrá consultar la información de las transacciones de compra realizadas a través de la aplicación: transacción, fecha, vendedor, comprador...

Desarrollo y Tecnologías

Microservicios

La aplicación final será desarrollada haciendo uso del Framework Spring y distribuyéndose en Microservicios que seguirán el principio de responsabilidad única. Teniendo como resultado un diagrama de contexto parecido al siguiente:



Nota: los puertos en la imagen son solo un ejemplo, se deberá seguir la numeración de puertos según grupo explicada en este documento.

Así:

- El **Microservicio de Cliente** expondrá los servicios Restful para el CRUD (*Create, Read, Update and Delete*) de los clientes.
- El **Microservicio Catálogo** expondrá en igual medida los servicios RestFul para la gestión del catálogo de productos.
- El **Microservicio Banco**, simulará el cobro de las transacciones. Más específicamente, deberá proveer un servicio que:
 - Reciba el cargo de la compra al vendedor y los datos de la tarjeta a la que será carga
 - Verifique que el número de tarjeta es correcto.

- El número de tarjeta consta de 16 números, divisible entre 3
 - La fecha es posterior a la actual
 - El CV2 tiene 3 cifras
 - Genere un código de identificación de la transacción de compra.
 - Si el proceso se completa con éxito además del código de la transacción se devolverá el código de HTTP 200 y en caso de error el código HTTP 402.
- El **Microservicio Financiera**, almacenará el registro de todas las transacciones llevadas a cabo por la aplicación. Este servicio almacenará la información en una base de datos Mongo.

En total, la aplicación deberá incluir al menos **4 proyectos de Microservicios**, cada uno de los cuales:

- Tendrá asociado un puerto determinado de acceso.
- Todas las comunicaciones con cada uno de estos Microservicios se harán mediante JSON a controladores REST (Representational State Transfer) que hará uso de la anotación “@Controller” o “@RestController”.
- Se deberán seguir las recomendaciones y best practices para el desarrollo de servicios RESTful disponibles en Aula Global (empleo de códigos de estado HTTP, nomenclatura de los servicios, etc)
- Para el acceso a la base de datos será obligatorio el uso de la dependencia Spring-data para manejar los DAOs (Data Access Object).

Como resultado final del uso de este Framework con una arquitectura en Microservicios los alumnos dispondrán de varios proyectos que deberán interactuar entre ellos. De tal manera, que si alguno deja de funcionar todavía seguirá funcionando parte de la funcionalidad del sistema.

Numeración de los puertos de los microservicios

Los puertos asignados a los proyectos basados en Microservicios deberán seguir el siguiente patrón: 1YYXX donde YY es el identificador de grupo de prácticas e XX serán dígitos arbitrarios decididos por el grupo.

Nombres de los proyectos

Los nombres de los proyectos de los microservicios seguirán el siguiente esquema:

- Nombre del proyecto: CCCCyXX

§ cccc -> CLIE / BANK / CATAL / FINA

§ yy -> es el identificador del grupo de clase (80, 81, 82 o 89)

§ XX -> es el identificador del grupo de proyecto (ej: 10)

Desarrollo del Front-End

Los Microservicios implementados serán consumidos por 2 **aplicaciones distintas de front end: una para clientes y otra para el administrador.**

Las tecnologías a utilizar para los desarrollos de estas aplicaciones serán:

- **Aplicación JEE** y despliegue mediante el servidor de aplicaciones Payara
- **Spring Boot y Thymeleaf:** se creará un proyecto de Spring Boot que hará uso de la dependencia Thymeleaf como template de las páginas. El controlador de este proyecto hará de cliente servicios REST.

Los alumnos podrán escoger entre desarrollar ambas aplicaciones empleando la misma tecnología o utilizar una distinta para cada front-end, pero se valorará en la evaluación el empleo de Spring Boot/Thymeleaf en al menos una de las aplicaciones.

Numeración de los puertos de las aplicaciones del Front-End

- Se utilizará el puerto 8080 para JEE
- Se utilizará el puerto 8081 para el front end Spring Boot.

Nombres de los proyectos

Los nombres de los proyectos del front-end seguirán el siguiente esquema:

- Aplicación del cliente: CLIENTyyxx:

§ yy -> es el identificador del grupo de clase (80, 81, 82 o 89)

§ XX -> es el identificador del grupo de proyecto (ej: 10)

- Aplicación del administrador: ADMINyyxx

§ yy -> es el identificador del grupo de clase (80, 81, 82 o 89)

§ XX -> es el identificador del grupo de proyecto (ej: 10)

Bases de Datos

Los microservicios **Cliente y Catálogo** emplearán como medio de almacenamiento una base de datos **MySQL**, y el microservicio **Financiera** hará uso de **MongoDB**.

Base de Datos MySQL

El alumno realizará el diseño de la BD con todas las tablas necesarias para poder implementar las funcionalidades requeridas. En la entrega de la práctica se incluirá al menos:

- Diagrama E/R
- Script de creación
- Script de carga inicial de datos: para insertar datos necesarios con los cuales llevar a cabo las pruebas de las funcionalidades

Para facilitar a los profesores identificar las distintas bases de datos de los alumnos el nombre la base de datos deberá seguir el siguiente esquema:

MYSQLyy_XX_nombreBD

Donde:

- yy -> es el identificador del grupo de clase (80, 81, 82 o 89)
- XX -> es el identificador del grupo de proyecto (ej: 10)

Ejemplo: `MYSQL89_10_secondHand`

Igualmente, para facilitar la instalación de los microservicios el usuario y la contraseña de acceso especificados en el fichero `application.properties` serán:

`spring.datasource.username = root`

`spring.datasource.password = admin`

Base de Datos MongoDB

El alumno realizará el diseño del conjunto de colecciones de la base de datos Mongo. En la entrega deberá incluir el json asociado a la colección.

El nombre la base de datos deberá seguir el siguiente esquema:

- *MONGOyy_XX_nombreBD*

- Donde:
 - yy -> es el identificador del grupo de clase (80, 81, 82 o 89)
 - XX -> es el identificador del grupo de proyecto (ej: 10)
- Ejemplo: MONGO89_10_secondHand

Normativa de Entrega

La entrega deberá respetar las siguientes normas:

- La entrega se realizará a través de Aula Global en la actividad habilitada para el caso.
- **ÚLTIMO DÍA DE ENTREGA: 12/12/2021**
- El coordinador de cada grupo deberá entregar un único fichero comprimido .zip o .rar que contenga todos los archivos de la entrega
- El nombre del fichero comprimido deberá seguir el siguiente formato de nombre:

GR82.XX.Practica2.zip

dónde XX corresponde al identificador individual del grupo de

IMPORTANTE. En ningún caso, se aceptarán prácticas entregadas por correo electrónico

Dentro del fichero comprimido .zip o .rar deberá estar los siguientes elementos:

- Todos los proyectos que forman la práctica.

Deberán de poder ser desplegados de forma independiente e incluirán el código fuente de la aplicación para que el profesor de prácticas pueda analizarlo.
- Información de la/las BBDD.

Capturas de pantalla o imágenes de los diagramas de base de datos que permita conocer todas las tablas/colecciones que componen la base de datos, sus campos y sus relaciones.
- Scripts SQL/Colecciones de MongoDB.

Ficheros con comandos SQL que permita crear la BBDD de datos e importar una información mínima y ficheros que contengan colecciones de MongoDB con datos mínimos.
- Documentación.

Breve descripción de todos los servicios implementados, incluyendo propósito, parámetros (si los tiene) y los posibles códigos HTTP de respuesta. En el segundo apéndice se adjunta una tabla a modo de ayuda para la redacción de este documento.

- División de tareas y carga de trabajo

Documento o tabla resumen con la división de tareas de la práctica y reparto de carga de trabajo entre los miembros del equipo. o Información de instalación y prueba: se deberá detallar cualquier información adicional necesaria para la instalación y ejecución del proyecto, así como los usuarios de la aplicación sugeridos para realizar las pruebas y sus claves.

Criterios de Evaluación

La calificación final se compone de una calificación compartida por todos los miembros del grupo y una calificación individual. Para la calificación compartida se tendrán en cuenta los siguientes criterios:

- Correcto funcionamiento de la aplicación.
- Satisfacción de todos los requisitos recogidos en este documento.
- Estructura de la solución que facilite el mantenimiento y adaptación del sistema.
- Claridad y diseño del interfaz. Se deberá procurar un aspecto profesional a la interfaz con el usuario.
- Claridad y documentación del código.
- Empleo de Spring Boot/Thymeleaf en al menos una de las aplicaciones del Front End.
- Calidad de la gestión del proyecto (Recomendado usar SCRUM).
- Posibles ampliaciones (Siempre que se cumplan los requisitos mínimos de la práctica).

La calificación individual se basará en los siguientes criterios:

- Defensa de la práctica. Todos los miembros deberán ser capaces de responder a preguntas relacionadas con la implementación de cualquiera de las funcionalidades de la aplicación.
- Posibles diferencias de carga de trabajo entre los distintos miembros del grupo.
- Conocimiento de lo desarrollado por parte de todos.

El profesor del grupo reducido acordará con cada grupo una fecha y hora para llevar a cabo la defensa de la práctica. Se requiere la presencia de todos los miembros del grupo durante la defensa de la práctica. Aquellos integrantes del equipo que no se presenten a la defensa no puntuarán en la entrega.

Gestión del Proyecto

Se recomienda (opcional) disponer de un sistema de gestión de la configuración (GIT) y una herramienta colaborativa (Trello, Taiga,...) que permita ver la evolución del backlog del producto, el avance y la descomposición de las tareas a lo largo del tiempo de desarrollo de la práctica.

Anexo I

En los actuales navegadores está bloqueado el acceso a recursos que se encuentran en diferentes dominios de los que proporcionaron la página desde javascript por cuestiones de seguridad. Para resolver esto se puede configurar el navegador (no recomendable por cuestiones de seguridad) o incluir en la cabecera de http la posibilidad CORS (*Cross-Origin Resource Sharing*).

En Spring Boot basta con incluir la anotación `@CrossOrigin` en el controlador:

https://developer.mozilla.org/es/docs/Web/HTTP/Access_control_CORS

Anexo II

Insert a new User

This endpoint allows storing a new user in the system.

POST	/Path/end/point/{path_variable}	
	Additional content: Parameter, Request_Body, Path_variable	
	200	Content that is returned
	402	Content that is included in the result
