

Pontificia Universidad Católica Madre y Maestra



Programación web avanzada ISC-517-T

Presentado por:
Cristian Bueno
2015-1256

Presentado al profesor:
Carlos Camacho

Tema:
Practica Docker Basico

Fecha de entrega:
Sábado, 13 de junio del 2020

Introducción

Un grupo de emprendedores de la PUCMM inicia una empresa con una idea innovadora, la cual tiene como modelo de negocio el alquiler de equipos electrónicos de consumo para el público en general, el proceso principal consiste en dar de alta a los clientes, tener un inventario de los equipos y procesar el alquiler de los equipos; el alquiler se completa cuando todos los elementos alquilados son entregados, la facturación se genera cuando el cliente va entregando los equipos, el costo a cobrar al cliente depende de la tarifa que tiene cada uno de los equipos

Link: <https://github.com/cristianbg11/alquiler>

Desarrollo

En ausencia del uso una base de datos, vamos a trabajar los objetos creados en colecciones estáticas que permitirán visualizar la información independiente de la sesión que realice la petición, para ellos, estaremos trabajando con el siguiente objeto propuesto:

En esta práctica se espera que realicen un mantenimiento para la estructura planteada en el cual realicen operaciones de insertar, actualizar, borrar y consultar. Debemos tener una pagina para presentar el listado de todos los estudiantes registrados, en dicha pagina tendrán enlaces que permitirán crear un estudiante, el cual nos enviará aun formulario para estos fines. De la misma forma, para las acciones de borrado y actualización. A modo de ejemplo anexo imágenes que explican lo realizado. Se debe implementar una plantilla para la vistosidad del proyecto.

Un grupo de emprendedores de la PUCMM inicia una empresa con una idea innovadora, la cual tiene como modelo de negocio el alquiler de equipos electrónicos de consumo para el público en general, el proceso principal consiste en dar de alta a los clientes, tener un inventario de los equipos y procesar el alquiler de los equipos; el alquiler se completa cuando todos los elementos alquilados son entregados, la facturación se genera cuando el cliente va entregando los equipos, el costo a cobrar al cliente depende de la tarifa que tiene cada uno de los equipos. Dado el escenario planteado realice:

1. El sistema utilizará como motor de base de datos H2, utilizando ORM bajo JPA. Una vez inicializado el sistema se debe crear un usuario administrador, el cual será el único que podrá crear usuarios nuevos y asignar roles de permisos a dicho usuario.
2. El sistema debe permitir administrar los clientes, la cual debe tener la información básica siendo obligatorio indicar la cédula y la opción de subir una foto del cliente.
3. El sistema debe permitir administrar los equipos que serán alquilados, un equipo debe ser asociado a una familia y una sub-familia depende de una familia de equipo. Cada equipo tendrá una cantidad de existencia la cual será controlada en el proceso de alquiler e indicando el costo por alquiler por día.
4. Para alquilar los equipo, se debe debe registrar la fecha, el cliente, fecha

de promesa de entrega y los equipos. El sistema debe validar que la existencia sea valida. En el proceso de alquiler se debe poder visualizar la imagen del equipo.

5. El calculo de la facturación será realizada en el momento de la entrega de los equipos, donde indicará el costo por día de uso. Puede ser que el cliente entregue de forma parcial los equipos alquilados.

6. El sistema debe mostrar un listado de los equipos alquilados que aun no son devueltos indicando el cliente y los días alquilados ordenado del alquiler más viejo al nuevo.

7. Se debe mostrar un historial de alquileres por cliente y un gráfico de promedio de días alquilado por familia y sub-familia de los equipos.

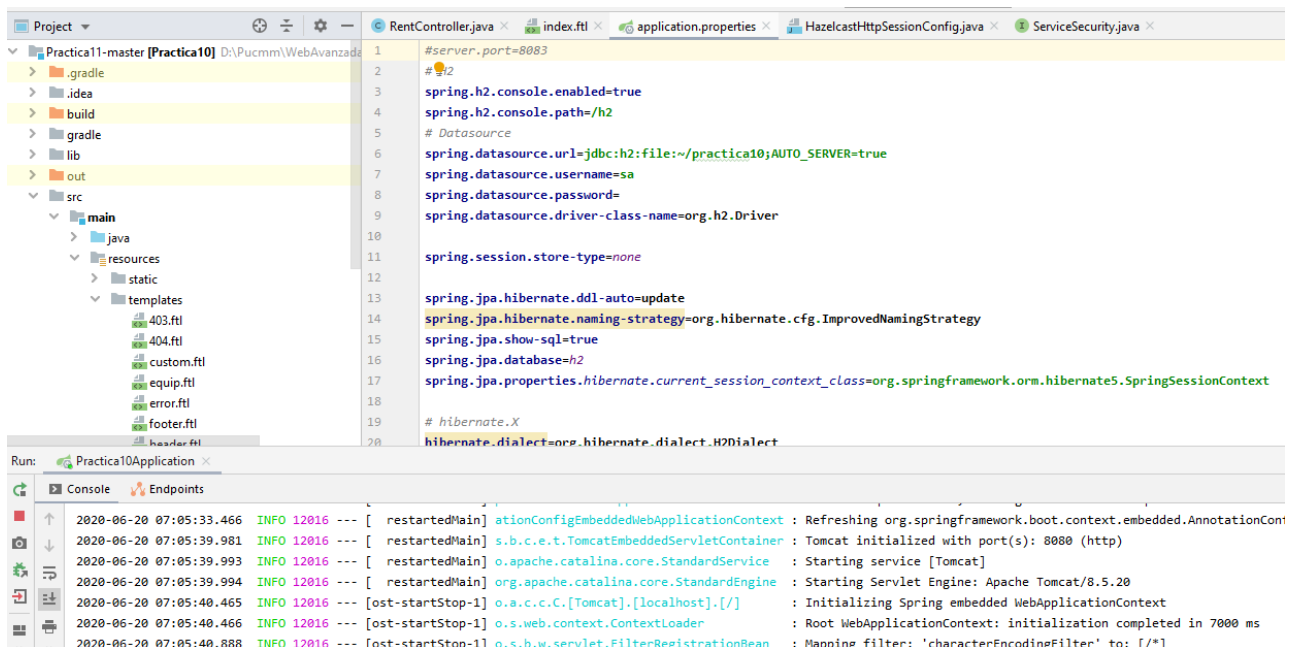
8. El sistema debe implementar i18n, para dos idiomas mínimo.

9. El sistema debe implementar plantillas basadas en HTML5 y CSS, con capacidad de web responsiva.

10. El sistema debe implementar Spring Security para el manejo de seguridad.

De nuestra práctica #1, sobre el proyecto de Spring Boot, vamos a incluir la funcionalidad al proyecto que permita contener la aplicación en Docker y publicarla en Docker Hub. La imagen debe tener las siguientes características:

- 1. Utilizar una imagen base de Java que ocupe la menor cantidad de espacio.*
- 2. Utilizar un volumen donde se guarde los logs y la base de datos H2.*
- 3. Utilizar variable de ambiente en la definición de la imagen para cambiar el puerto de arranque de la aplicación.*



The screenshot shows an IDE with the following components:

- Project Explorer:** Displays the project structure for 'Practica11-master'. The 'src/main/resources/templates' directory contains files: 403.ftl, 404.ftl, custom.ftl, equip.ftl, error.ftl, footer.ftl, and hazelcast.ftl.
- Properties File:** The 'application.properties' file is open, showing configuration for a Spring Boot application. Key settings include:
 - `server.port=8083`
 - `spring.h2.console.enabled=true`
 - `spring.h2.console.path=/h2`
 - `spring.datasource.url=jdbc:h2:file:~/practica10;AUTO_SERVER=true`
 - `spring.datasource.username=sa`
 - `spring.datasource.password=`
 - `spring.datasource.driver-class-name=org.h2.Driver`
 - `spring.session.store-type=none`
 - `spring.jpa.hibernate.ddl-auto=update`
 - `spring.jpa.hibernate.naming-strategy=org.hibernate.cfg.ImprovedNamingStrategy`
 - `spring.jpa.show-sql=true`
 - `spring.jpa.database=h2`
 - `spring.jpa.properties.hibernate.current_session_context_class=org.springframework.orm.hibernate5.SpringSessionContext`
 - `hibernate.dialect=org.hibernate.dialect.H2Dialect`
- Console:** Displays the application logs. The logs show the application starting successfully on port 8080 (http) and initializing the Spring embedded WebApplicationContext. The logs also show the application starting the Tomcat service and the Servlet Engine: Apache Tomcat/8.5.20.

Conclusión

Esto nos va facultando para acercarme más a manejar todas las herramientas necesarias para desarrollar una aplicación web avanzada, dónde se pueda desarrollar en el backend de una página web la manipulación de información que se mostraría en una página dinámica.

Como hemos podido observar, mediante Spring Boot es posible crear aplicaciones completas que se ejecuten en un entorno reducido, pues no es necesario tener un servidor de aplicaciones completo que nos consumirá una gran cantidad de recursos.

Bibliografia

https://freemarker.apache.org/docs/ref_directive_list.html

<https://freefrontend.com/css-tabs/>