Práctica 2 : ApiRest Gestion de Usuarios

Asignatura: Sistemas Distribuidos

Autores: Daniel Requena Garrido
Raúl Rodríguez López-Rey

INDICE

Esta memoria corresponde con el proyecto que se tiene que ejecutar en el puerto 8081, ApiRest Gestión de Usuarios

Resumen de la arquitectura:

La arquitectura de este proyecto se basa en:

- 1 entidade: Se encuentran dentro del paquete model (Usuario.java).
- 1 ApiRest Controller: UsuarioRestController.java(se encuentra en el paquete restController).
- 1 repositorio: Encontrado en el paquete repository (RepoUsuario.java)
- 1 service: En el paquete service UserService. Aquí se implementan algunos métodos que nos ayuda a trabajar con los repositorios.
- 1 inicializador de Bases de Datos: En el paquete principal y denominado DBInitializer.

Entidad:

- Usuario: Sus atributos son una variable tipo long que será el id (generado de manera automática),login, nombre, contraseña, estado y fechaAlta, estos últimos de tipo String.

  El constructor recibe como parámetros 3 strings, login, nombre y contraseña y un integer para indicar el salgo . El estado por defecto se pone en activo y la fecha de alta se coge el momento en el que es creado el objeto Usuario

  Geters y seters de los atributos.

Repositorio:

- RepoUsuario: Interface que extiende JpaRepository<Usuario,Long>.

Service:

En los services en resumidas cuentas usamos métodos de los mismos repositorios o métodos que implican varios métodos del repositorio y los juntamos en uno solo. En el controladore usamos métodos de los repositorios y services indistintivamente.

Controlador:

UsuarioRestController:

Se realizan los métodos CRUD que se especifican, se encuentran en la url api/usuarios(las urls que mencionamos a continuación llevan delante esta url).

El listado general se realiza haciendo un get con la url “/”.

Para buscar un solo usuario se hace un get con la url “/{id}” siendo id el identificador único del usuario

Para añadir un nuevo usuario , con la ayuda de un software como *Postman*, hacemos un post a la url “/” poniendo en el body los datos del usuario que queremos agregar.

Para actualizar los parámetros , con un put y la url “/{id}” y nuevamente con *Postman* y metemos en el body de la petición los datos que queremos cambiar, los datos que no se especifiquen en la petición no se cambiarán, dejando los valores que tenía el usuario anteriormente.

El método de borrar se hace con delete “/{id}” este método realmente lo que hace es poner en inactivo el usuario, por lo que se podría haber hecho con put.

El método para realizar el pago de la reserva se utiliza con put “/pagos/{id}”. Este método resta el doble del valor de lo que vale alquilar la bici (1.5 *2) . El valor de 1,5€ lo hemos cogido de manera arbitraria.

Las devoluciones se realizan con put “/devoluciones/{id}” . Este método devuelve 1.5€ al usuario

Modelo Conceptual Base de datos: