NOMBRE: ALFONSO ARAUJO

FECHA: 02-04-2023

1) Realizar una pequeña aplicación web en la que se implemente un login. La web constará de un formulario de login (usuario y password) que se validará contra una base de datos, una página de bienvenida en la que se mostrará el nombre del usuario que ha realizado el login correctamente y un proceso de logout que se puede implementar mediante un enlace o un botón en la propia página de bienvenida.

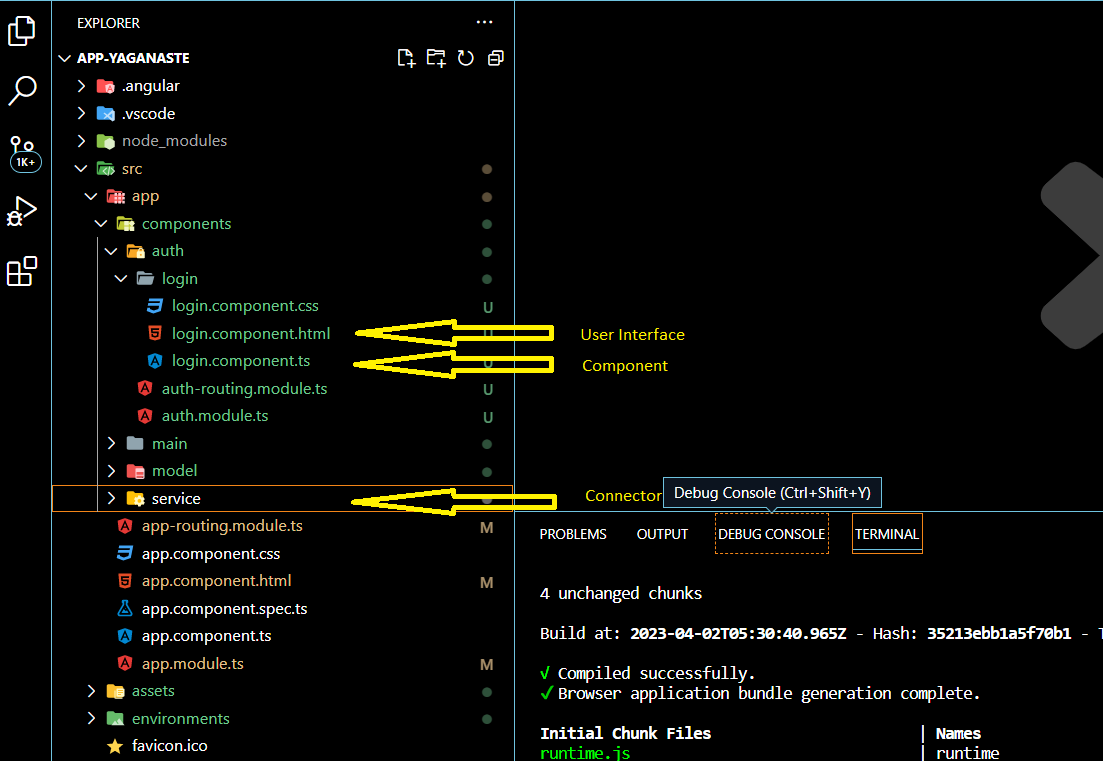
Empleando la tecnología o framework para diseño de aplicaciones web y/o servicios REST que mejor se conozca, explicar en qué parte o capas tanto front como back se dividirá la aplicación y explicar el funcionamiento de cada una, de forma general.

Solución del Problema:

**Es Importante recalcar que para la solución de todos los problemas que se requería conectar a una base de datos para este caso se usó Postgresql.**

1.- Creación de aplicación web

1. Se utilizo Angular Framework en su versión 14 de desarrollo para Javascript de código abierto, que sirve para generar aplicaciones web (front-end), en particular aplicaciones de una sola página.
2. Impulsa el uso del patron de diseño de software MVC que reparte el código del programa en componentes.
3. Divide a la aplicación web en 3 capaz:
   1. User Interface: Corresponde a los elementos de pantalla o controles codificados en html
   2. Component: Es el código que reacciona a los eventos de los elementos de la User Interface
   3. Connector: Es el representante del servidor en el cliente, lo que hace es pedir los datos al servidor.



Ese Patrón se repite por cada componente o funcionalidad que necesitemos agregar.

Adicional a esto angular divide el proyecto en módulos (puede ser uno o N) y cada modulo este compuesto por N cantidad de componentes. Para nuestro caso de estudio la distribución quedo de la siguiente manera:

Modulo Principal (app.module)

Component (app.component)

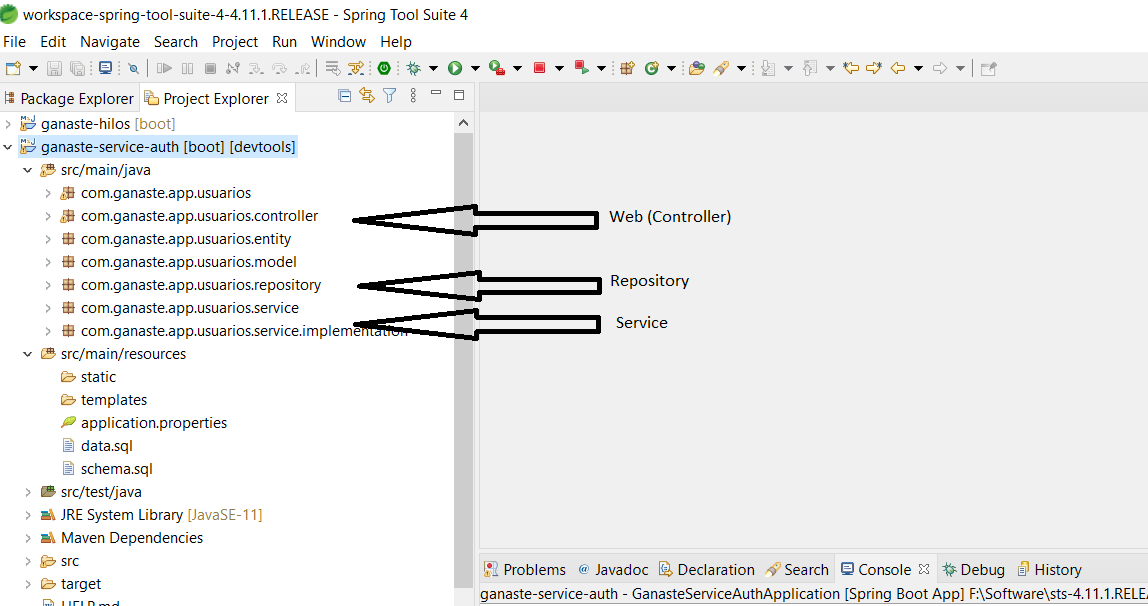
Enrutamiento

Modulo (auth.module)

Component (auth.component)

1.- Creación de aplicación web api

1. Se utilizó Spring Boot Framework en su versión 4.11.1.RELEASE de desarrollo para Java, sirve para desarrollar aplicaciones con arquitecturas de microservicios y API REST
2. Impulsa el uso del patrón MVC como marco de desarrollo de las API REST.
3. Implementa la arquitectura multicapa:
   1. Repository para la capa de repositorio:
   2. Service para la capa de servicios
   3. Web para de controladores.



2) Bases de Datos. Tenemos las siguientes tablas en una base de datos:

Tabla PROVINCIAS. Tiene como clave primaria IDPROVINCIA

|  |  |
| --- | --- |
| **IDPROVINCIA** | **DESCRIPCION** |
| 1 | Zaragoza |
| 2 | Huesca |
| 3 | Teruel |

Tabla PRODUCTOS. Tiene como clave primaria IDPRODUCTO

|  |  |  |
| --- | --- | --- |
| **IDPRODUCTO** | **DESCRIPCION** | **PRECIO** |
| A | Playmobil | 5,00 |
| B | Puzzle | 10,25 |
| C | Peonza | 3,65 |

Tabla CLIENTES. Tiene como clave primaria IDCLIENTE. El campo IDPROVINCIA apunta al campo IDPROVINCIA de la tabla PROVINCIAS.

|  |  |  |
| --- | --- | --- |
| **IDCLIENTE** | **NOMBRE** | **IDPROVINCIA** |
| 1 | Juan Palomo | 1 |
| 2 | Armando Ruido | 2 |
| 3 | Carmelo Cotón | 1 |
| 4 | Dolores Fuertes | 3 |
| 5 | Alberto Mate | 3 |

Tabla COMPRAS. Tiene como clave primaria IDCOMPRA. El campo IDCLIENTE apunta al campo IDCLIENTE de la tabla CLIENTES. El campo IDPRODUCTO apunta al campo IDPRODUCTO de la tabla PRODUCTOS.

|  |  |  |  |
| --- | --- | --- | --- |
| **IDCOMPRA** | **IDCLIENTE** | **IDPRODUCTO** | **FECHA** |
| 1 | 1 | C | 01/01/2022 |
| 2 | 2 | B | 15/01/2022 |
| 3 | 2 | C | 22/01/2022 |
| 4 | 4 | A | 03/02/2022 |
| 5 | 3 | A | 05/02/2022 |
| 6 | 1 | B | 16/02/2022 |
| 7 | 1 | B | 21/02/2022 |
| 8 | 4 | A | 21/02/2022 |
| 9 | 5 | C | 01/03/2022 |
| 10 | 3 | A | 01/03/2022 |
| 11 | 3 | C | 05/03/2022 |
| 12 | 2 | B | 07/03/2022 |
| 13 | 2 | B | 11/03/2022 |
| 14 | 1 | A | 18/03/2022 |
| 15 | 1 | C | 29/03/2022 |
| 16 | 5 | B | 08/04/2022 |
| 17 | 5 | C | 09/04/2022 |
| 18 | 4 | C | 09/04/2022 |
| 19 | 1 | A | 12/04/2022 |
| 20 | 2 | A | 19/04/2022 |

Crear las consultas SQL necesarias para obtener los siguientes datos:

1. Todas las compras detalladas con los datos del cliente, de los productos y de cada una de las ventas (código de cliente, nombre de cliente, nombre de provincia, producto, importe, fecha de la venta)
2. Las compras detalladas de los clientes de Teruel.
3. Las compras detalladas de los clientes de Huesca y Zaragoza en el primer trimestre de 2015
4. Las compras agrupadas por producto de todos los clientes mostrando el número de compras y el importe total para cada producto por cada uno de los clientes (código de cliente, provincia, producto, número de ventas, importe total)
5. Número de peonzas totales que se han comprado en el mes de marzo por los clientes de Zaragoza (número de peonzas e importe total)
6. Las compras realizadas por todos los clientes agrupadas por mes (código de cliente, nombre, provincia, mes, número de compras, importe total)

Detalle de todas las ventas agrupadas por día del mes (día del mes, producto, número de compras, importe total). Por ejemplo, los días 1 de enero, febrero, marzo y abril se agruparán como día del mes 1

Solución del Problema:

En el folder Solución 2 se encuentran los scripts necesarios para la solución del problema con un archivo de notas donde especifican los pasos a seguir.

3) Bases de Datos.

Don Pepe Muelas, gran aficionado la música, debido al alto volumen de grabaciones que posee desea construir una base de datos que gestione toda su discografía. Diseñar el modelo entidad-relación de la base de datos necesaria para almacenar las grabaciones, intérpretes y compañías que editan cada grabación, teniendo en cuenta que:

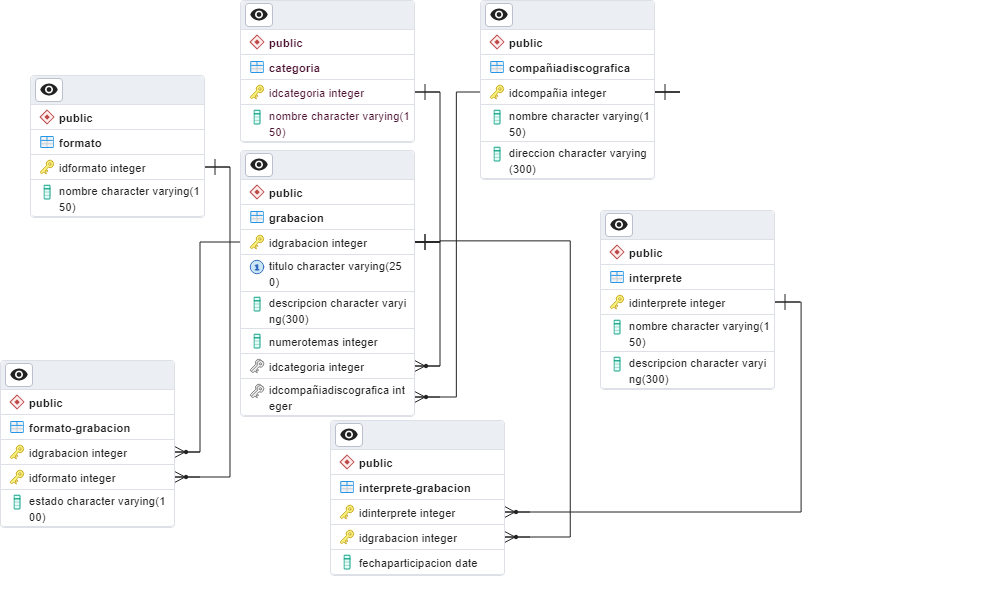
* De cada grabación se debe almacenar el título, la categoría musical a la que pertenece (jazz, rock, ...), el número de temas que lo componen y un campo descripción que pueda almacenar aspectos especiales de la grabación. Además, cada grabación se distingue de las otras por su título.
* Cada grabación tiene un formato asociado (CD, cinta, ...) y una grabación puede existir en diferentes formatos. Para cada una de ellas se desea conocer su estado de conservación (bueno, malo, regular).
* De los intérpretes se desea registrar su nombre y una breve descripción que contenga los aspectos más relevantes de su discografía, así como las grabaciones en las que ha participado cada uno y en qué fecha participó.

Toda grabación es editada por una compañía discográfica, la cual se desea almacenar.

Una compañía discográfica se diferencia de otra por un Identificador de compañía y tiene un nombre y una dirección.

Solución del Problema:

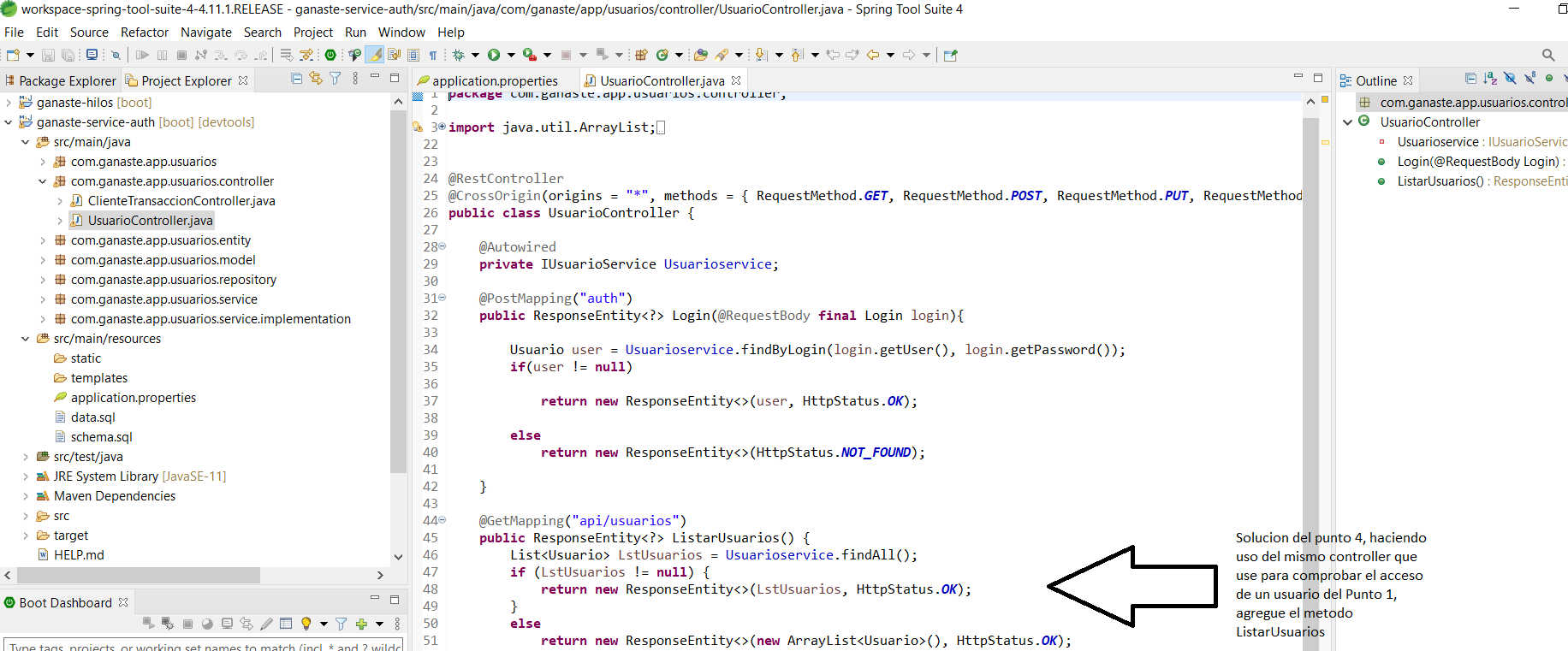
En el folder Solución 3 se encuentra la imagen del MER



4) Realizar la implementación de una clase Controller de un api Rest que incluya un endpoint “/api/usuarios” tipo “get” para listar los usuarios de una tabla USUARIOS de una bd, incluyendo la capa de servicio con el acceso a la bd para obtener el select de usuarios mediante jdbctemplate.

Solución del Problema:

En este caso hice uso del api del punto uno para crear el endpoint que haciendo uso de JDBC Template me consulte todos los usuarios de la tabla de Usuarios



Para realizar la prueba del endpoint en PostMan o cualquier otro programa de su preferencia realice lo siguiente:

Haciendo uso del método get escriba lo siguiente:

http://localhost:8596/api/usuarios

5) Dada una tabla CLIENTES con los campos CLIENTE y SALDO, se solicita implementar un método que mediante dos updates reste un importe del campo saldo del cliente1 y lo sume al saldo del cliente en una sola transacción, de forma que en caso de excepción el saldo de los dos clientes vuelva a su valor anterior.

private void trasnferencia(Connection conn, BigDecimal importe, int cliente1, int cliente2) {  
}

Solución del Problema:

En este caso hice uso del api del punto uno adicione un entidad Cliente con su servicio, controller y repository, creando en la misma base de dato del punto la tabla Cliente con unos importes iniciales, para probar que el saldo del cliente 1 disminuya y la del cliente 2 aumente en una transacción, realice en postman lo siguiente:

Por el método post:

http://localhost:8596/transaccion

y en el body agregue el siguiente json

{

    "cliente1":1,

    "cliente2":2,

    "importe": "25000"

}

Donde el valor del importe puede varias a gusto del usuario.

6) Si tienes dos hilos concurrentes que necesitan acceder a valores de una clase global de la aplicación, como lo harías de forma que mientras un hilo accede a la variable el otro quede bloqueado y no pueda acceder hasta que el primero haga uso de la variable.

Solución del Problema:

En la carpeta Solución 6 se encuentra la solución del problema.