

Desarrollo de Sistemas con Tecnología

Java Emisión 15

Ejercicio 1:

Nombre: Noyola Nazario Alejandro

1. Usar cuatro tablas catálogo de su proyecto final

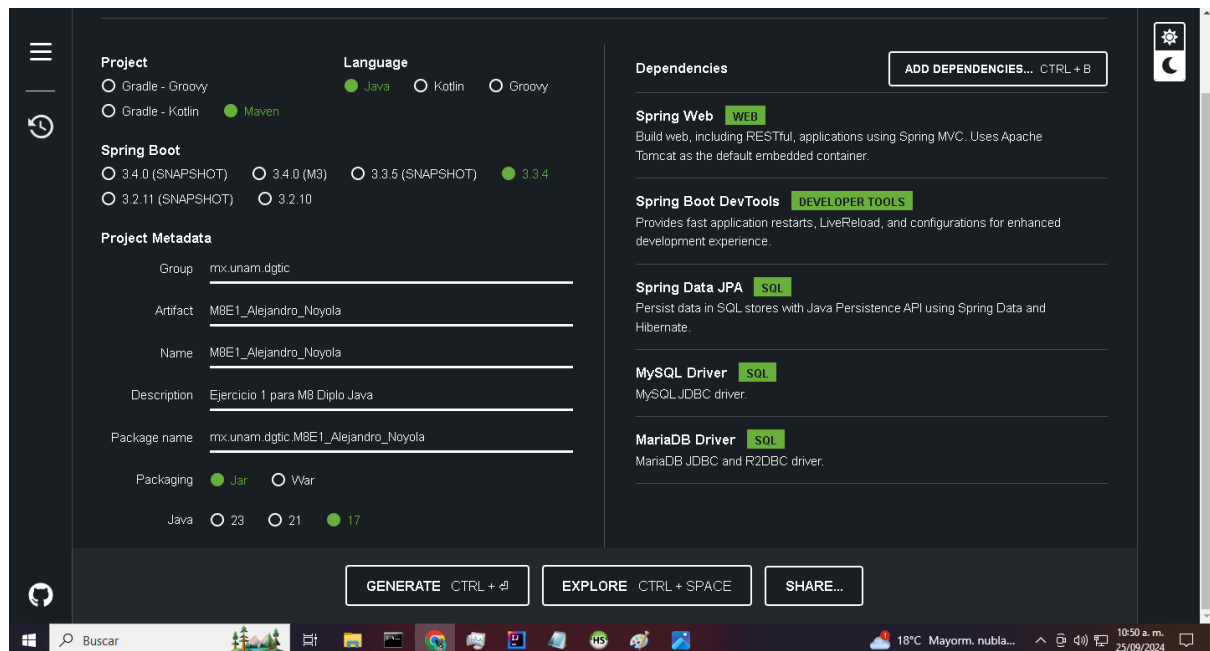
2. Se debe crear un proyecto Spring nuevo con las siguientes características:

1. Nombre del proyecto

2. M8E1<PrimerNombre> _<PrimerApellido>

3. Ejemplo: M8E1_Omar_Mendoza

3. Con soporte JPA y MariaDb



4. Para cada Entidad se debe implementar interfaces relacionadas con

1. CrudRepository

The screenshot shows an IDE with the project structure on the left. The package structure is: `mx.unam.dgtic` (containing `Comprador`, `Electronicos`, `Laptops`, `Vendedor`) and `datos` (containing `CompradorRepository`, `ElectronicosRepository`, `LaptopsRepository`, `VendedorRepository`). The `CompradorRepository.java` file is open, showing the following code:

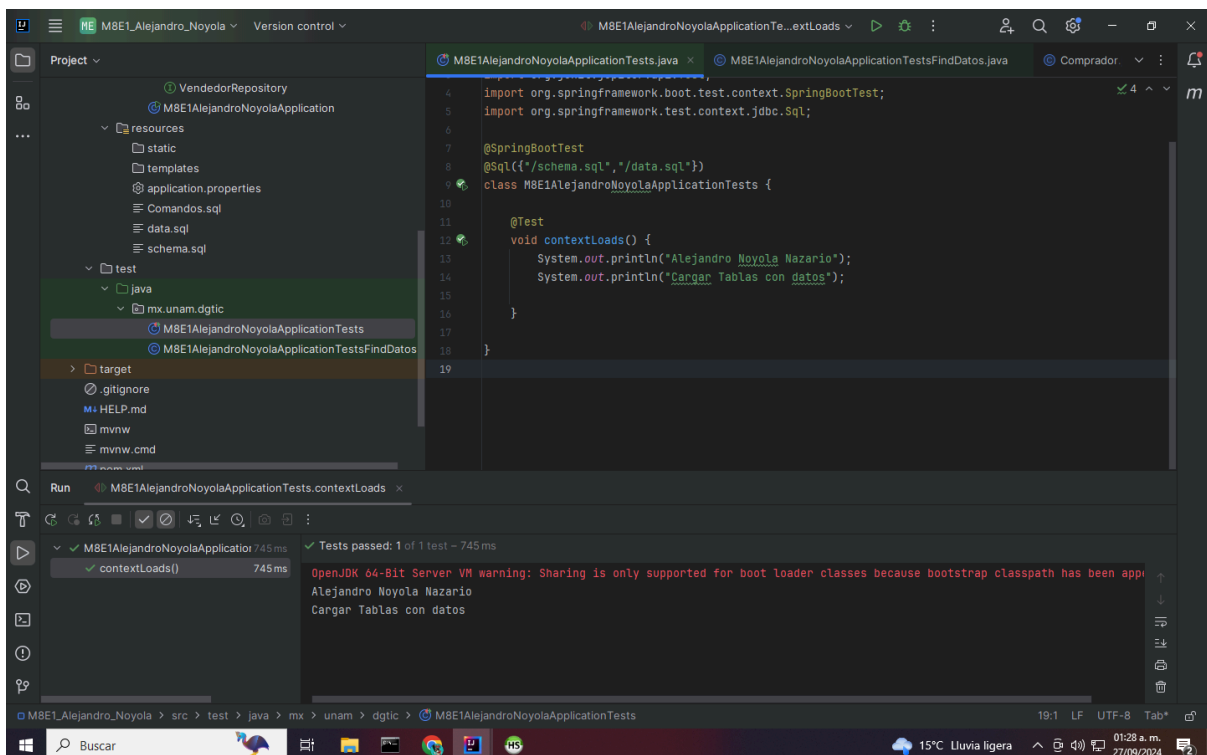
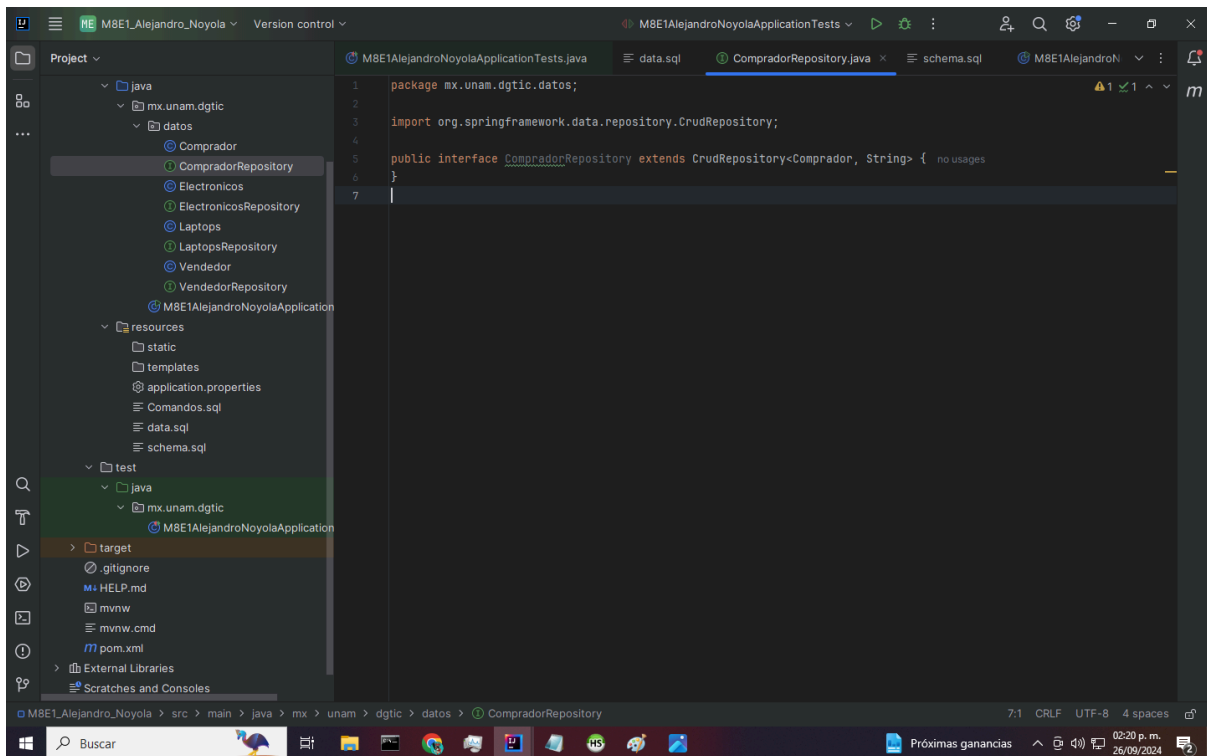
```
1 package mx.unam.dgtic.datos;
2
3 import org.springframework.data.repository.CrudRepository;
4
5 public interface CompradorRepository extends CrudRepository<Comprador, String> {
6 }
7
```

The screenshot shows the same IDE with the `Vendedor.java` file open. The code defines the `Vendedor` entity with the following attributes and methods:

```
1 package mx.unam.dgtic.datos;
2
3
4 import ...
5
6 @Entity
7 @Table(name = "vendedor")
8 public class Vendedor {
9     @Id
10     @Column(name = "idVendedor")
11     private Integer idVendedor;
12     private String nombreVendedor;
13     private String apellidoVendedor;
14     private Integer edad;
15     private char genero;
16     private String correo;
17     private String usuarioVendedor;
18     private String contrasena;
19
20     public Vendedor() {
21     }
22
23     public Vendedor(Integer idVendedor) {
24         this.idVendedor = idVendedor;
25     }
26
27     public Vendedor(Integer idVendedor, String nombreVendedor, String apellidoVendedor, Integer edad, char genero, String correo, String usuarioVendedor, String contrasena) {
28         this.idVendedor = idVendedor;
29         this.nombreVendedor = nombreVendedor;
30         this.apellidoVendedor = apellidoVendedor;
31         this.edad = edad;
32         this.genero = genero;
33         this.correo = correo;
34         this.usuarioVendedor = usuarioVendedor;
35         this.contrasena = contrasena;
36     }
37 }
38
```

5. Implementar operaciones CRUD sobre cada una de las Clases

1. El esquema y el estado de la tablas se deben crear desde una prueba Unitaria



2. Generar pruebas unitarias que usen los métodos findAll, findBy

7. Probar las implementaciones en pruebas unitarias organizadas por tema (diferentes clases de prueba para cada tema que incluyen varias pruebas unitarias).

- insertarUnoCompradorTest[M8E1AlejandroNoyolaApplicationTestsFindDatosComprador]

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows the project structure with folders like `resources`, `test`, and `java`. The `test` folder is expanded, showing `java`.
- Editor:** Displays the code for `M8E1AlejandroNoyolaApplicationTestsFindDatosComprador.java`. The code is as follows:


```

12 public class M8E1AlejandroNoyolaApplicationTestsFindDatosComprador {
13     void insertarUnoCompradorTest(){
14
15         Optional<Comprador> optional = compradorRepository.findById(13579);
16         if (optional.isPresent()){
17             System.out.println("Recuperar comprador desde la BD");
18             System.out.println(optional.get());
19         } else {
20             System.out.println("Matricula 13579 no localizada");
21         }
22     }
23 }
      
```
- Run Console:** Shows the output of the test execution:


```

OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
ALEJANDRO NOYOLA NAZARIO
Insertar Comprador
Recuperar comprador desde la BD
Comprador{idComprador=13579, nombreComprador='Valerie', apellidoComprador='Noyola', edad=27, genero='F', correo='Vale1164@gmail.com', usuarioComprador='606
      
```
- Status Bar:** Shows the file path `M8E1AlejandroNoyolaApplicationTestsFindDatosComprador` and the encoding `UTF-8`.

- editarUnoElectronicosTest[M8E1AlejandroNoyolaApplicationTestsFindDatosElectronicos]

The screenshot shows an IDE with the following components:

- Project Explorer:** Shows the project structure with folders like `mx.unam.dgtic` and `datos`. The `datos` folder is expanded, showing `Comprador`, `CompradorRepository`, `Electronicos`, `ElectronicosRepository`, `Laptops`, `LaptopsRepository`, `Vendedor`, and `VendedorRepository`.
- Editor:** Displays the code for `M8E1AlejandroNoyolaApplicationTestsFindDatosElectronicos.java`. The code is as follows:

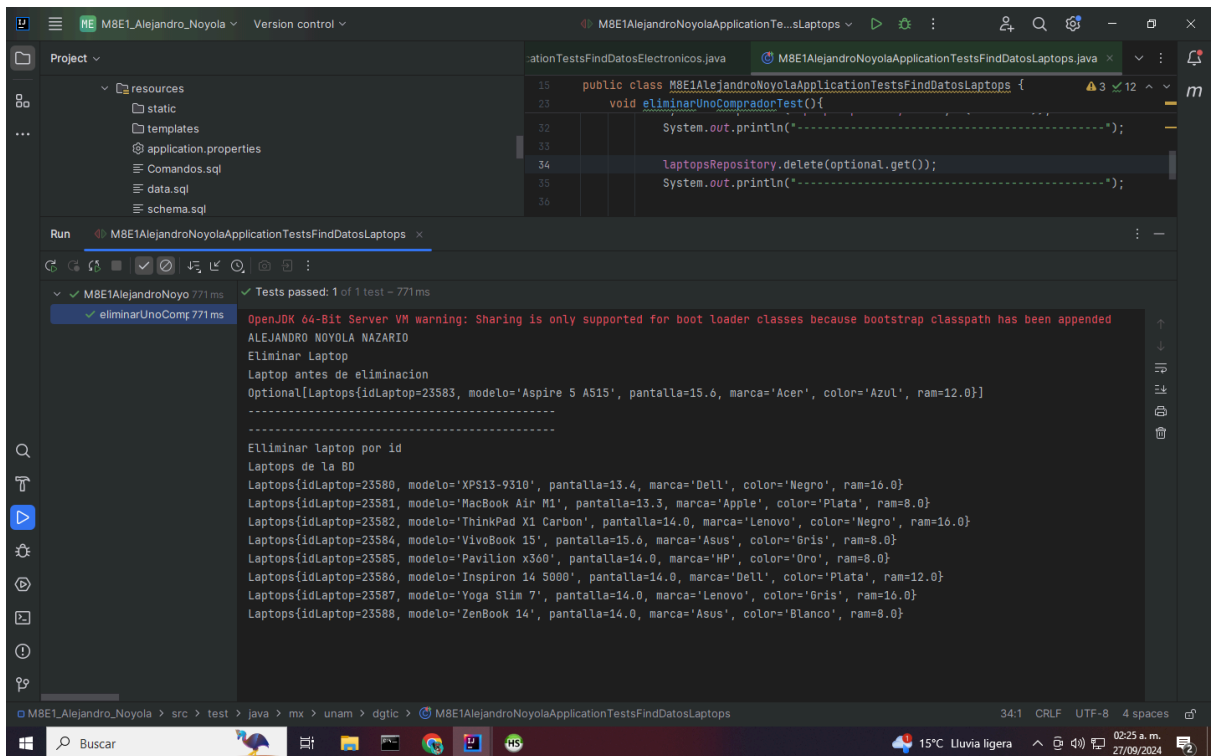

```

12 public class M8E1AlejandroNoyolaApplicationTestsFindDatosElectronicos {
13     void editarUnoCompradorTest(){
14
15         electronicos.setNombreElectronico("Xbox Series S");
16         electronicos.setPrecio(9299.00);
17         electronicos.setExistencias(75);
18         electronicos.setPrecioDeVenta(999.99);
19         electronicos.setCostoDeProduccion(900.500);
20
21         electronicosRepository.save(electronicos);
22         System.out.println("Electronico despues de actualizar");
23         System.out.println(electronicos);
24     }
25 }
      
```
- Run Console:** Shows the output of the test execution:

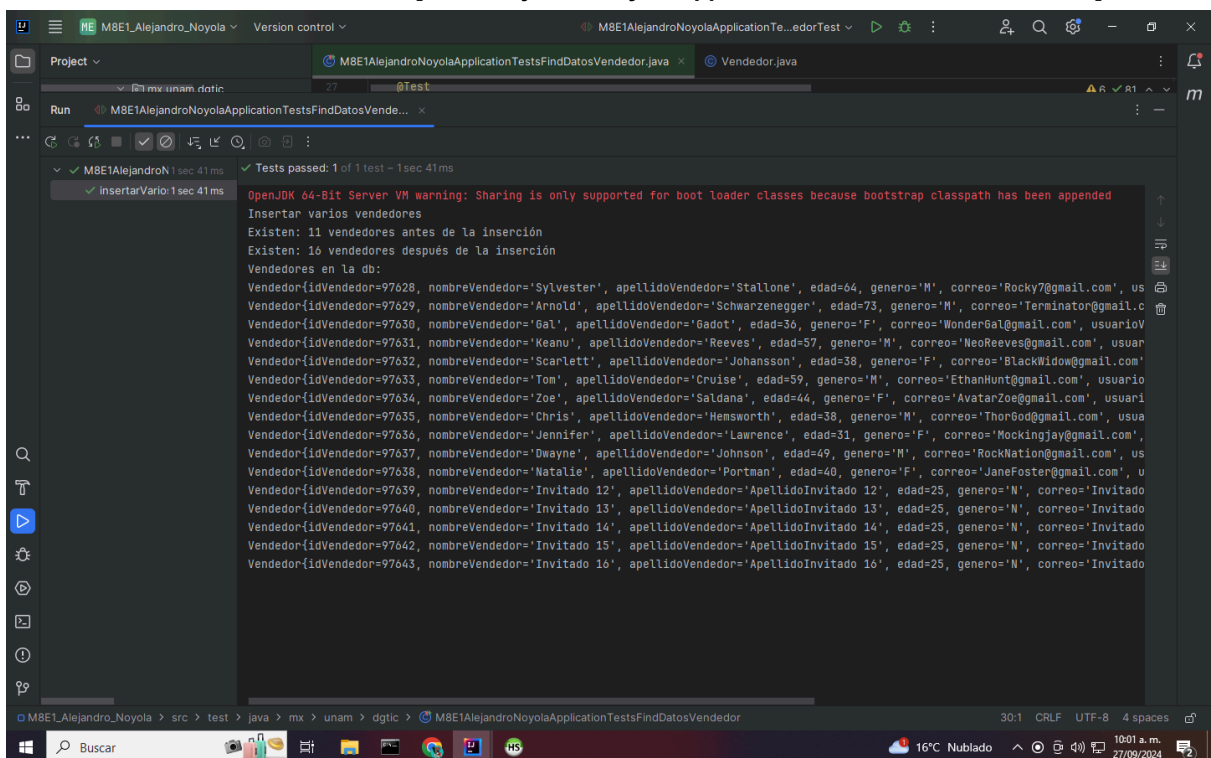

```

OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
ALEJANDRO NOYOLA NAZARIO
Editar Alumno
Electronico antes de actualizar
Electronicos{idElectronico=24680, nombreElectronico='Xbox Series X', precio=10299.0, existencias=100, codigoDeBarras=1234567890123, precioDeVenta=1299.99, c
Electronico despues de actualizar
Electronicos{idElectronico=24680, nombreElectronico='Xbox Series S', precio=9299.0, existencias=75, codigoDeBarras=1234567890123, precioDeVenta=999.99, cost
      
```
- Status Bar:** Shows the file path `M8E1AlejandroNoyolaApplicationTestsFindDatosElectronicos` and the encoding `UTF-8`.

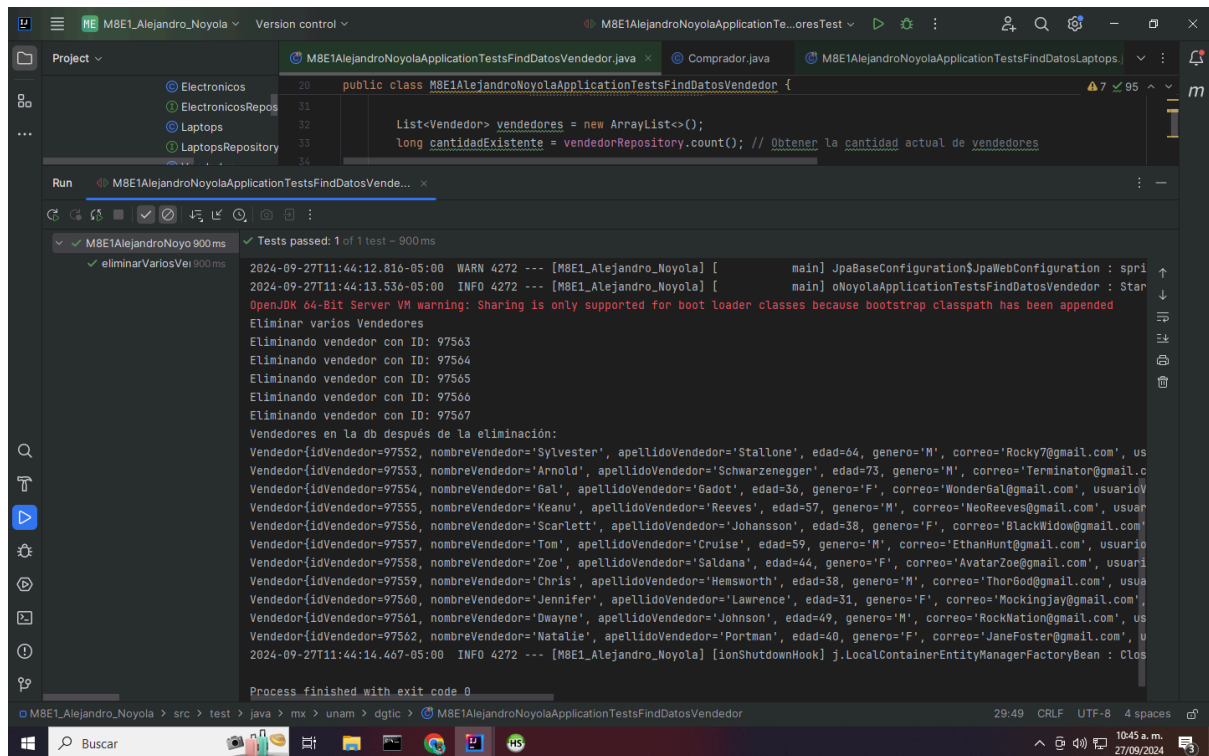
- eliminarUnoLaptopsTest[M8E1AlejandroNoyolaApplicationTestsFindDatosLaptops]



- insertarVariosVendedorTest[MBE1AlejandroNoyolaApplicationTestsFindDatosVendedor]



- eliminarVariosVendedoresTest[MBE1AlejandroNoyolaApplicationTestsFindDatosVendedor]



8. Entregar

- El archivo zip del proyecto IntelliJ Idea CE listo para descomprimir, abrir y ejecutar.
- Un documento PDF con evidencias de construcción y ejecución de cada uno de los puntos, agregando una descripción en cada punto
- El documento PDF no debe estar incluido en el archivo zip