

MINTIC UNIVERSIDAD DEL BOSQUE

Proyecto Ciclo 3

Tienda de Productos para mascotas

Grupo 27

Profesor: Ing. Germán Niño Tovar

Integrantes

Andrés Romero
Hector Urrego
Rosario Florez
Ricardo Samboni
Jorge Suárez

Septiembre de 2021

Sprint 1 Tienda Mascotas

Trello

Description

Edit

MÓDULO DE LOGIN DEL SISTEMA

Requisitos:


- Necesito ingresar al sistema con el usuario inicial admininicial y la contraseña admin123456.
- Necesito crear nuevos usuarios que entrarán al sistema
- Necesito consultar los datos del usuario, por medio de la cédula
- Necesito actualizar los datos de los usuarios que estén en el sistema
- Necesito borrar los datos de usuarios que estén en el sistema

Google Drive

×

Enable Google Drive

Access your Drive files for a project directly from its card, or create and attach new Drive files to a card.



Pantalla_Bienvenida_Color.jpg

...

Google Drive • Sep 17

Attachments

Dates

Attachment

Custom Fields

Add dropdowns, text fields, dates, and more to your cards.

Start free trial

POWER-UPS

+ Add Power-Ups

AUTOMATION

+ Add button

ACTIONS

→ Move

Copy

Make template

Watch

Archive

Share

Checklist

Hide checked items

Delete

13%

☒

 Crear Landing page

☐

 Crear formulario☐☐☐☐☐☐

Add an item

Creación de base de datos

Tabla T_Usuarios

```
create database Tienda_Mascotas27;  
use Tienda_Mascotas27;  
Create table T_Usuarios  
(  
    cedula_usuario bigint(20),  
    nombre_usuario varchar(40),  
    email_usuario varchar(70),  
    usuario varchar(20),  
    clave varchar(20)  
);  
select * from t_usuarios;
```

Tabla T_Clientes

```
Create table T_Clientes  
(  
    cedula_cliente bigint(20),  
    nombre_cliente varchar(50),  
    direccion_cliente varchar(100),  
    telefono_cliente varchar(100),  
    email_cliente varchar(70)  
);  
select * from t_clientes;
```

Tabla T_Proveedores

```
Create table T_Proveedores
(
    nit_proveedor bigint(20),
    nombre_proveedor varchar(100),
    direccion_proveedor varchar(100),
    telefono_proveedor varchar(25),
    ciudad_proveedor varchar(25)
);
select * from t_proveedores;
```

Tabla T_Productos

```
Create table T_Productos
(
    codigo_producto bigint(20),
    Iva_compra double,
    nit_proveedor bigint(20),
    nombre_producto varchar(100),
    precio_compra double,
    precio_venta double
);
select * from t_productos;
```

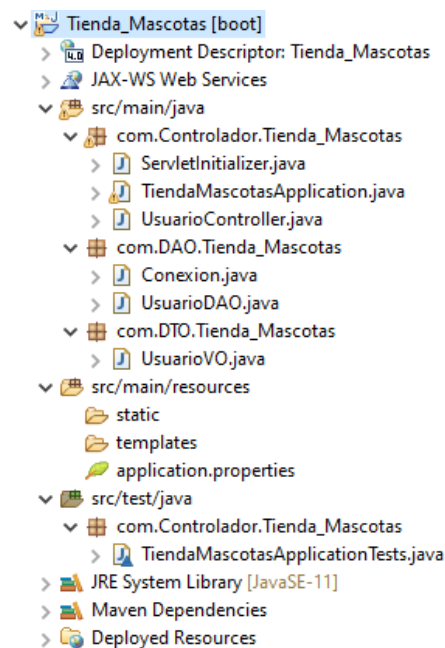
Tabla T_Ventas

```
Create table T_Ventas
(
    codigo_venta bigint(20),
    cedula_cliente bigint(20),
    cedula_usuario bigint(20),
    Iva_venta double,
    total_venta double,
    valor_venta double
);
select * from t_ventas;
```

Tabla T_Detalle_Ventas

```
Create table T_detalle_Ventas
(
    codigo_detalle_venta bigint(20),
    cantidad_producto int(12),
    codigo_producto bigint(20),
    codigo_venta bigint(20),
    valor_total double,
    valor_venta double,
    valor_Iva double
);
select * from T_detalle_ventas;
```

Estructura del proyecto Tienda Mascotas



Package com.Controlador.Tienda_Mascotas

TiendaMascotasApplication.java

```
1 package com.Controlador.Tienda_Mascotas;
2
3 import org.springframework.boot.SpringApplication;
4
5 @SpringBootApplication
6 public class TiendaMascotasApplication {
7
8     public static void main(String[] args) {
9         SpringApplication.run(TiendaMascotasApplication.class, args);
10
11         Conexion conn = new Conexion();
12         //conn.getCon();
13     }
14 }
15
```

UsuarioController.java

```
1 package com.Controlador.Tienda_Mascotas;
2
3 import java.util.ArrayList;
4
5 @RestController
6 public class UsuarioController {
7
8     @RequestMapping("/listarUsuarios")
9
10     public ArrayList<UsuarioVO> listarUsuarios()
11     {
12         UsuarioDAO Dao = new UsuarioDAO();
13
14         return Dao.listarUsuarios();
15     }
16 }
17
```

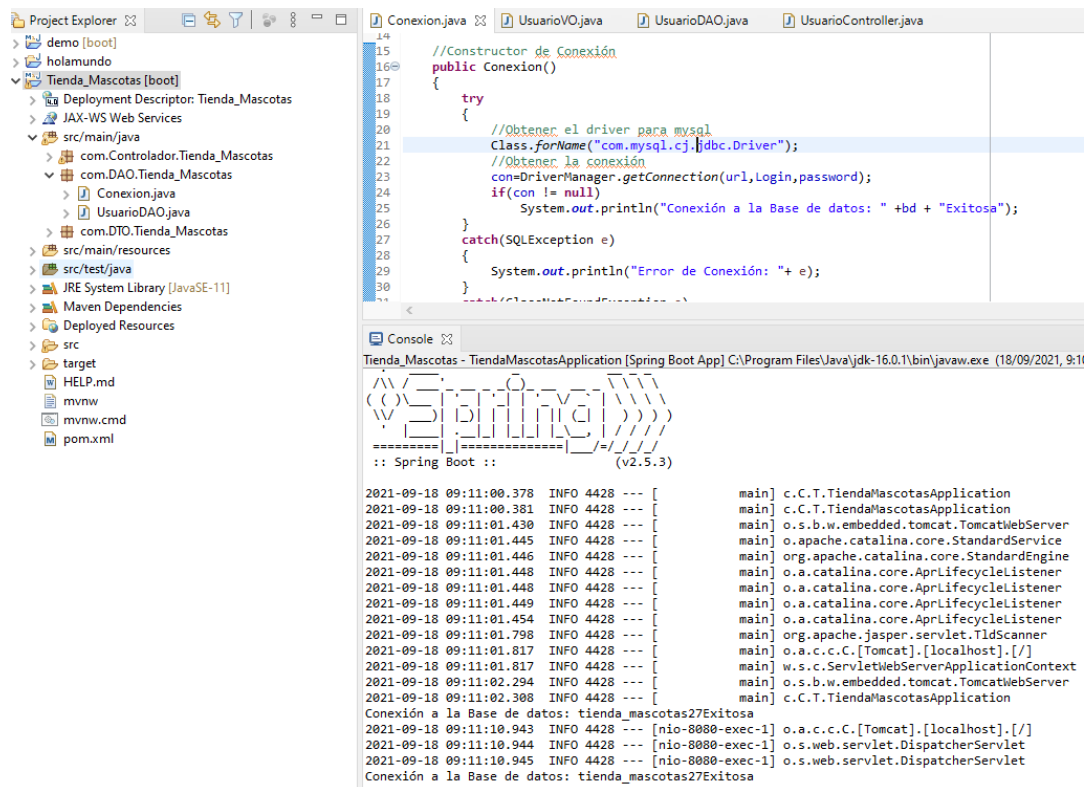
ServerInitiallizer.java

```
1 package com.Controlador.Tienda_Mascotas;
2
3 import org.springframework.boot.builder.SpringApplicationBuilder;
4
5
6 public class ServletInitializer extends SpringBootServletInitializer {
7
8     @Override
9     protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
10         return application.sources(TiendaMascotasApplication.class);
11     }
12
13 }
14
```

Package com.DAO.Tienda_Mascotas

Conexion.java

```
Conexion.java UsuarioVO.java UsuarioDAO.java UsuarioController.java
14
15 //Constructor de Conexión
16 public Conexion()
17 {
18     try
19     {
20         //Obtener el driver para mysql
21         Class.forName("com.mysql.cj.jdbc.Driver");
22         //Obtener la conexión
23         con=DriverManager.getConnection(url,Login,password);
24         if(con != null)
25             System.out.println("Conexión a la Base de datos: " +bd + "Exitosa");
26     }
27     catch(SQLException e)
28     {
29         System.out.println("Error de Conexión: "+ e);
30     }
31     catch(ClassNotFoundException e)
32     {
33         System.out.println("Error de Conexión: "+ e);
34     }
35     catch(Exception e)
36     {
37         System.out.println("Error de Conexión: "+ e);
38     }
39 }
40
41 //Metodo para retornar la conexión
42 public Connection getCon()
43 {
44     return con;
45 }
46
47 //Metodo para desconectar
48 public void desconectar()
49 {
50     con = null;
51 }
52
53 }
```



UsuarioDAO.java

```

public class UsuarioDAO {

    //Metodo para consultar los usuarios

    public ArrayList<UsuarioVO> listaUsuarios()
    {
        ArrayList<UsuarioVO> milista = new ArrayList<UsuarioVO>();
        Conexion con = new Conexion();

        try
        {
            PreparedStatement consulta = con.getCon().prepareStatement("SELECT * FROM t_usuarios");
            ResultSet rs = consulta.executeQuery();
            while(rs.next())
            {
                UsuarioVO persona = new UsuarioVO();
                persona.setCedula(Integer.parseInt(rs.getString("cedula_usuario")));
                persona.setNombre(rs.getString("nombre_usuario"));
                persona.setCorreo(rs.getString("email_usuario"));
                persona.setUsuario(rs.getString("usuario"));
                persona.setClave(rs.getString("clave"));

                milista.add(persona);
            }
            rs.close();
            consulta.close();
            con.desconectar();

        }
        catch(Exception e) {

            System.out.println("Error No se pudo Conectar" + e);

        }

        return milista;
    }
}

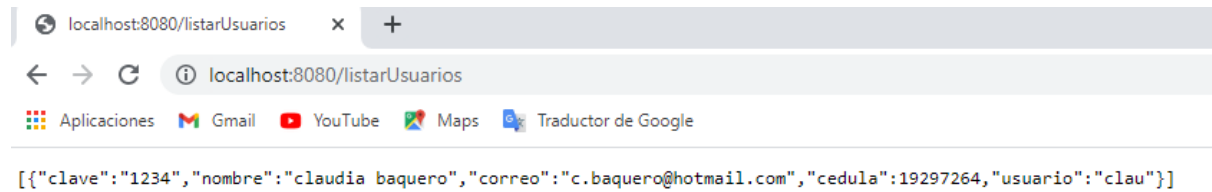
```


Package com.DTO.Tienda_Mascotas

UsuarioVO.java

```
1 package com.DTO.Tienda_Mascotas;
2
3 public class UsuarioVO {
4
5     private int Cedula;
6     private String Nombre;
7     private String Correo;
8     private String Usuario;
9     private String Clave;
10    public int getCedula() {
11        return Cedula;
12    }
13    public void setCedula(int cedula) {
14        Cedula = cedula;
15    }
16    public String getNombre() {
17        return Nombre;
18    }
19    public void setNombre(String nombre) {
20        Nombre = nombre;
21    }
22    public String getCorreo() {
23        return Correo;
24    }
25    public void setCorreo(String correo) {
26        Correo = correo;
27    }
28    public String getUsuario() {
29        return Usuario;
30    }
31    public void setUsuario(String usuario) {
32        Usuario = usuario;
33    }
34    public String getClave() {
35        return Clave;
36    }
37    public void setClave(String clave) {
38        Clave = clave;
39    }
40}
```

Ejecución del programa



Base de datos MySQL

