INTEGRANTES:
Huayllas Pinto Kevin
Chavez Fabian Kevin
Martinez Paredes Juan Rene
Pérez Maldonado Leandro Nicolás
Elena

Capturas de la conexión con la base de datos

Huayllas Pinto Kevin

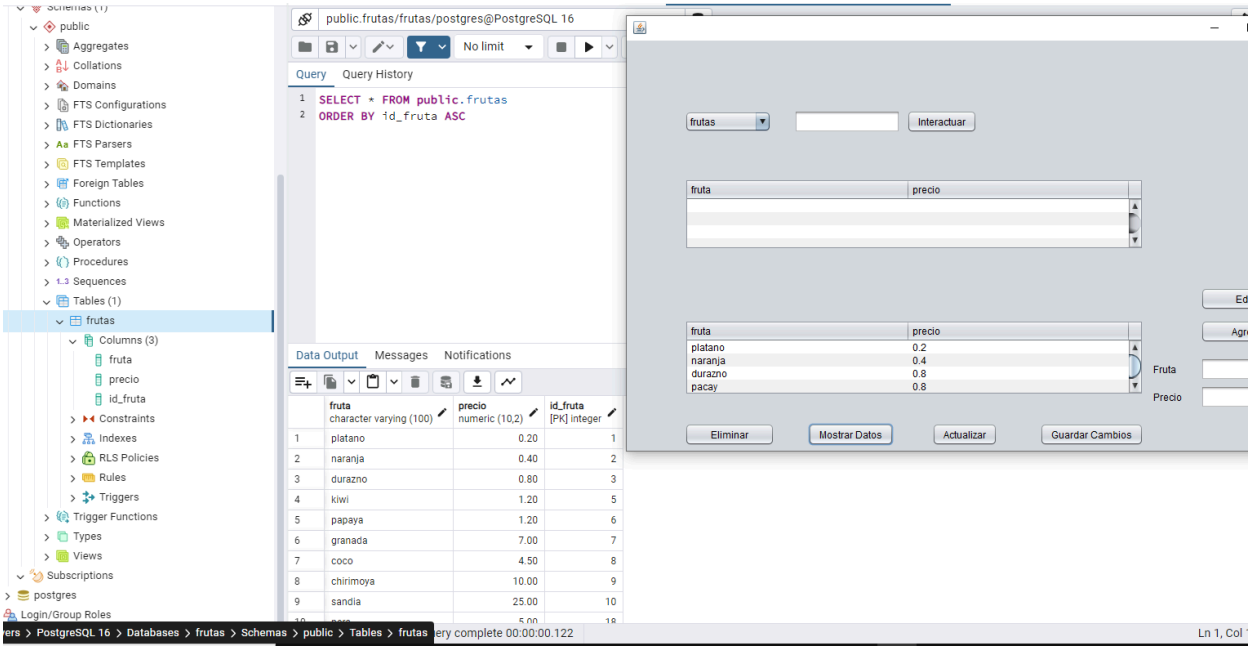
```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection {
    private static final String DB_URL = "jdbc:postgresql://oregon-postgres.render.com:5432/sisinfo2db";
    private static final String DB_USERNAME = "kevin_123";
    private static final String DB_PASSWORD = "8baO6G0RmY0HjY2PqSrScxaR3p4aI3We";

    public static Connection getConnection() throws SQLException {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            return DriverManager.getConnection(DB_URL, DB_USERNAME, DB_PASSWORD);
        } catch (ClassNotFoundException e) {
            System.err.println("Error al cargar el controlador JDBC: " + e.getMessage());
            throw new SQLException("Error al cargar el controlador JDBC", e);
        }
    }
}
```

Martinez Paredes Juan Rene

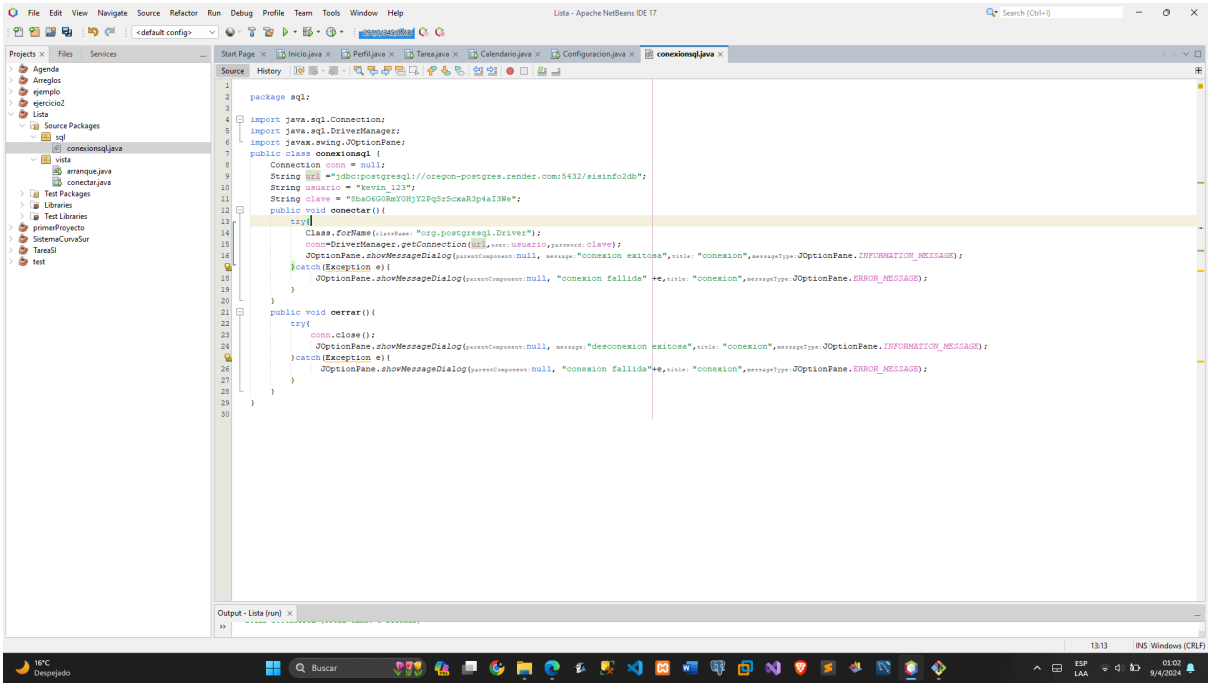
```java
*/
import java.sql.*;

public class ConexionBD {
    private static final String URL = "jdbc:postgresql://oregon-postgres.render.com:5432/sisinfo2db";
    private static final String USER = "kevin_123";
    private static final String PASSWORD = "8baO6G0RmY0HjY2PqSrScxaR3p4aI3We";

    public static Connection getConnection() throws SQLException {
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }

    public static void close(Connection con, Statement stmt, ResultSet rs) {
        try {
            if (rs != null) rs.close();
            if (stmt != null) stmt.close();
            if (con != null) con.close();
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    public static void closeConnection(Connection con) {
        try {
            if (con != null) con.close();
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }
}
```

Chavez Kevin Fabian

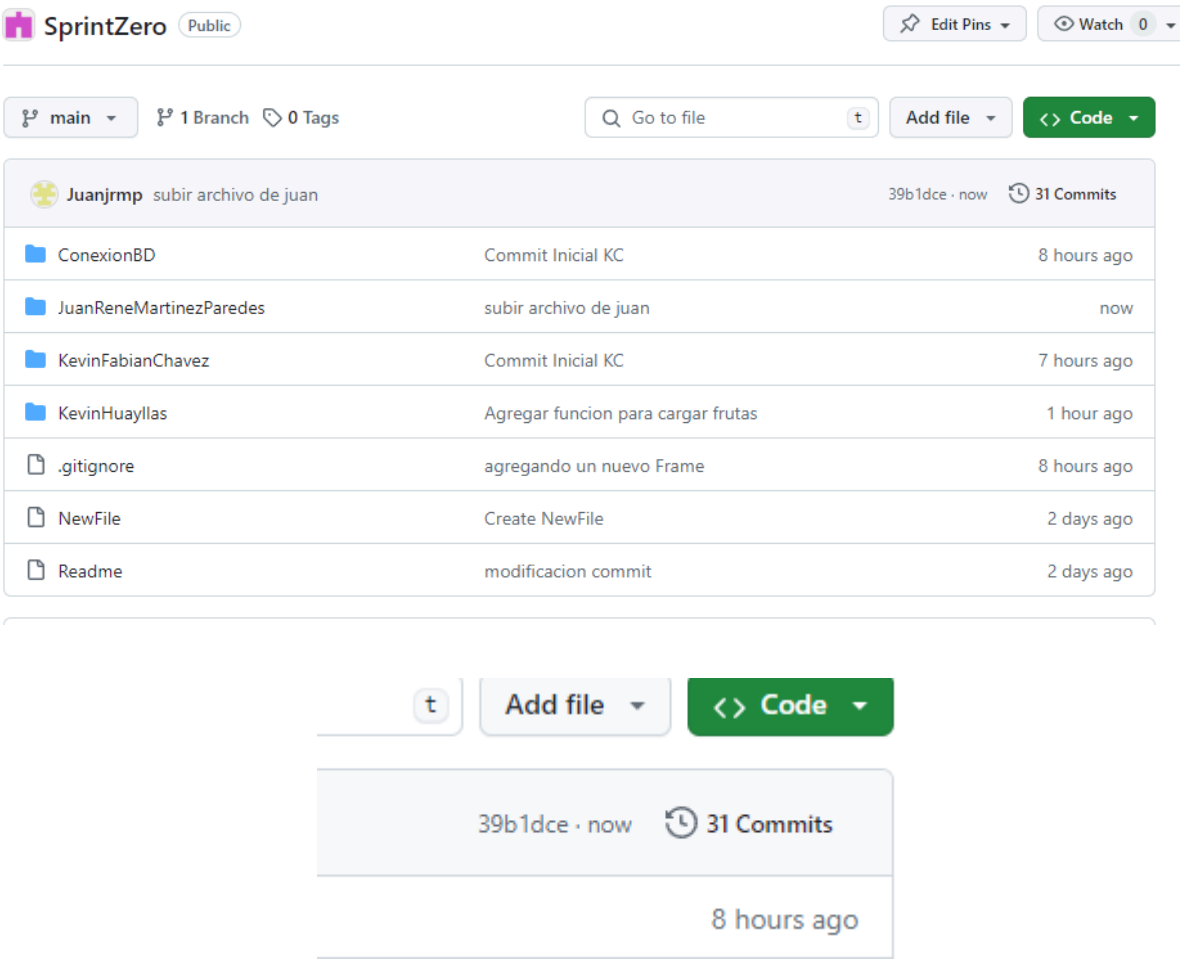
```java
10  import java.sql.*;
11
12  public class ConexionBD {
13      private static final String URL = "jdbc:postgresql://oregon-postgres.render.com:5432/sisinfo2db";
14      private static final String USER = "kevin_123";
15      private static final String PASSWORD = "8baO6G0RmY0HjY2PqSrScxaR3p4aI3We";
16
17      public static Connection getConnection() throws SQLException {
18          return DriverManager.getConnection( url: URL,  user: USER,  password: PASSWORD);
19      }
20
21      public static void close(Connection con, Statement stmt, ResultSet rs) {
22          try {
23              if (rs != null) rs.close();
24              if (stmt != null) stmt.close();
25              if (con != null) con.close();
26          } catch (SQLException ex) {
28              ex.printStackTrace();
29          }
30
31      public static void closeConnection(Connection con) {
32          try {
```


public.frutas/frutas/postgres@PostgreSQL 16
No limit
Query
Query History
SELECT * FROM public.frutas
ORDER BY id_fruta ASC
Data Output
Messages
Notifications
fruta
precio
id_fruta
platano 0.20 1
naranja 0.40 2
durazno 0.80 3
kiwi 1.20 5
papaya 1.20 6
granada 7.00 7
coco 4.50 8
chirimoya 10.00 9
sandia 25.00 10
frutas
Interactuar
Fruta
Precio
Eliminar
Mostrar Datos
Actualizar
Guardar Cambios

Elena


Lista - Apache NetBeans IDE 17
conexionsql.java
package sql;
import java.sql.Connection;
import java.sql.DriverManager;
import javax.swing.JOptionPane;
public class conexionsql {
Connection conn = null;
String url ="jdbc:postgresql://oregon-postgres.render.com:5432/sisinfo2db";
String usuario = "kevin_123";
String clave = "8baO6G0RmY0HjY2PqSrScxaR3p4aI3We";
public void conectar(){
try{
Class.forName("org.postgresql.Driver");
conn=DriverManager.getConnection(url,usuario,clave);
JOptionPane.showMessageDialog(null,"conexion exitosa","conexion",JOptionPane.INFORMATION_MESSAGE);
}catch(Exception e){
JOptionPane.showMessageDialog(null, "conexion fallida" +e,"conexion",JOptionPane.ERROR_MESSAGE);
}
}
public void cerrar(){
try{
conn.close();
JOptionPane.showMessageDialog(null,"desconexion exitosa","conexion",JOptionPane.INFORMATION_MESSAGE);
}catch(Exception e){
JOptionPane.showMessageDialog(null, "conexion fallida"+e,"conexion",JOptionPane.ERROR_MESSAGE);
}
}
}

Cuadro de muestras de commits



SCRIPT DE LA BASE DE DATOS

```
    -- Table: public.frutas

-- DROP TABLE IF EXISTS public.frutas;

CREATE TABLE IF NOT EXISTS public.frutas
(
    id_fruta integer NOT NULL DEFAULT
nextval('frutas_id_fruta_seq'::regclass),
    nombre character varying(30) COLLATE
pg_catalog."default" NOT NULL,
    precio numeric NOT NULL,
```

```
    CONSTRAINT frutas_pkey PRIMARY KEY (id_fruta),
    CONSTRAINT frutas_nombre_key UNIQUE (nombre)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.frutas
    OWNER to postgres;

ALTER TABLE frutas ALTER COLUMN precio TYPE DECIMAL;


INSERT INTO frutas(nombre,precio)
VALUES('platano',0.20);
INSERT INTO frutas(nombre,precio)
VALUES('naranja',0.40);
INSERT INTO frutas(nombre,precio)
VALUES('durazno',0.80);
INSERT INTO frutas(nombre,precio) VALUES('pacay',0.80);
INSERT INTO frutas(nombre,precio) VALUES('kiwi',1.20);
INSERT INTO frutas(nombre,precio) VALUES('papaya',3);
INSERT INTO frutas(nombre,precio)
VALUES('granada',4.5);
INSERT INTO frutas(nombre,precio) VALUES('coco',7);
INSERT INTO frutas(nombre,precio)
VALUES('chirimoya',10);
INSERT INTO frutas(nombre,precio) VALUES('sandia',25);

CREATE OR REPLACE FUNCTION getFruits()
returns table (nombre VARCHAR,precio INT) AS $func$
begin
return query
select nombre,precio from Frutas
ORDER by precio;
```

```
end; $func$ LANGUAGE plpgsql;
```

TABLA DE LA BASE DE DATOS

Query | Query History

```
1  SELECT * FROM frutas;
2
```

Data Output | Messages | Notifications

	id_fruta [PK] integer	fruta character varying (30)	precio numeric
1	1	platano	0.2
2	2	naranja	0.4
3	3	durazno	0.8
4	4	pacay	0.8
5	5	kiwi	1.2
6	6	papaya	1.2
7	7	coco	4.5
8	8	granada	7
9	9	chirimoya	10
10	10	sandia	25