**SISTEMA DE INVENTARIO PARA EQUIPOS DE CÓMPUTO**

**(PC STOCK)**

**Integrantes:**

Juan Camilo Murillo Torres

Alvaro Omar Mejia Aparicio

Barney Anibal Vergara Perez

Byron Torres

**Orientador:**

Sergio Arturo Medina Castillo

Proyecto de software que permitirá llevar a cabo un control de inventario preciso sobre equipos de cómputo, así como de sus periféricos y componentes con todas sus respectivas características, y enfocado a empresas; permitiendo llevar un histórico de movimientos entre los responsables de los equipos y sus periféricos y componentes asociados.

Sprint 5 - Ciclo 3

Universidad Autónoma De Bucaramanga

Misión Tic 2022

1. Aplicación Backend:

package restful.Model;

import java.util.ArrayList;

import java.sql.Statement;

import java.sql.ResultSet;

import java.sql.PreparedStatement;

import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement

public class UsuarioModel {

private int codigo\_usu;

private String nombre\_usu;

private String usuario;

private String password;

private String email;

private String codigo\_tipo\_usu;

private String codigo\_estado\_usu;

public UsuarioModel() {

}

public UsuarioModel(int codigo\_usu, String nombre\_usu, String usuario, String password, String email, String codigo\_tipo\_usu, String codigo\_estado\_usu) {

this.codigo\_usu = codigo\_usu;

this.nombre\_usu = nombre\_usu;

this.usuario = usuario;

this.password = password;

this.email = email;

this.codigo\_tipo\_usu = codigo\_tipo\_usu;

this.codigo\_estado\_usu = codigo\_estado\_usu;

}

public int getCodigo\_usu() {

return codigo\_usu;

}

public String getNombre\_usu() {

return nombre\_usu;

}

public String getUsuario() {

return usuario;

}

public String getPassword() {

return password;

}

public String getEmail() {

return email;

}

public String getCodigo\_tipo\_usu() {

return codigo\_tipo\_usu;

}

public String getCodigo\_estado\_usu() {

return codigo\_estado\_usu;

}

public void setCodigo\_usu(int codigo\_usu) {

this.codigo\_usu = codigo\_usu;

}

public void setNombre\_usu(String nombre\_usu) {

this.nombre\_usu = nombre\_usu;

}

public void setUsuario(String usuario) {

this.usuario = usuario;

}

public void setPassword(String password) {

this.password = password;

}

public void setEmail(String email) {

this.email = email;

}

public void setCodigo\_tipo\_usu(String codigo\_tipo\_usu) {

this.codigo\_tipo\_usu = codigo\_tipo\_usu;

}

public void setCodigo\_estado\_usu(String codigo\_estado\_usu) {

this.codigo\_estado\_usu = codigo\_estado\_usu;

}

}

Restful service:

package restful.Service;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

import java.sql.SQLException;

import java.util.ArrayList;

import restful.Model.UsuarioModel;

import restful.Model.conexion;

public class UsuarioServices {

public ArrayList<UsuarioModel> getusuario() {

ArrayList<UsuarioModel> lista = new ArrayList<>();

conexion conn = new conexion();

String sql = "SELECT \* FROM usuario";

try {

Statement stm = conn.getCon().createStatement();

ResultSet rs = stm.executeQuery(sql);

while (rs.next()) {

// llegue hasta aqui

UsuarioModel usuario = new UsuarioModel();

usuario.setCodigo\_usu(rs.getInt("codigo\_usu"));

usuario.setNombre\_usu(rs.getString("nombre\_usu"));

usuario.setUsuario(rs.getString("usuario"));

usuario.setPassword(rs.getString("password"));

usuario.setEmail(rs.getString("email"));

usuario.setCodigo\_tipo\_usu(rs.getString("codigo\_tipo\_usu"));

usuario.setCodigo\_estado\_usu(rs.getString("codigo\_estado\_usu"));

lista.add(usuario);

}

} catch (SQLException e) {

}

return lista;

}

public UsuarioModel getUsuario(int codigo\_usu) {

UsuarioModel usuario = new UsuarioModel();

conexion conex = new conexion();

String Sql = "SELECT \* FROM usuario WHERE codigo\_usu = ?";

try {

PreparedStatement pstm = conex.getCon().prepareStatement(Sql);

pstm.setInt(1, codigo\_usu);

ResultSet rs = pstm.executeQuery();

while (rs.next()) {

usuario.setCodigo\_usu(rs.getInt("codigo\_usu"));

usuario.setNombre\_usu(rs.getString("nombre\_usu"));

usuario.setUsuario(rs.getString("usuario"));

usuario.setPassword(rs.getString("password"));

usuario.setEmail(rs.getString("email"));

usuario.setCodigo\_tipo\_usu(rs.getString("codigo\_tipo\_usu"));

usuario.setCodigo\_estado\_usu(rs.getString("codigo\_estado\_usu"));

}

} catch (SQLException e) {

System.out.println(e);

}

return usuario;

}

public UsuarioModel addUsuario(UsuarioModel usuario) {

conexion conex = new conexion();

String Sql = "INSERT INTO usuario(codigo\_usu,nombre\_usu,usuario,password,email,codigo\_tipo\_usu,codigo\_estado\_usu)";

Sql = Sql + "values (?,?,?,?,?,?,?)";

try {

PreparedStatement pstm = conex.getCon().prepareStatement(Sql);

pstm.setInt(1, usuario.getCodigo\_usu());

pstm.setString(2, usuario.getNombre\_usu());

pstm.setString(3, usuario.getUsuario());

pstm.setString(4, usuario.getPassword());

pstm.setString(5, usuario.getEmail());

pstm.setString(6, usuario.getCodigo\_tipo\_usu());

pstm.setString(7, usuario.getCodigo\_estado\_usu());

pstm.executeUpdate();

} catch (SQLException e) {

System.out.println(e);

return null;

}

return usuario;

}

public UsuarioModel updateUsuario(UsuarioModel usuario) {

conexion conn = new conexion();

String sql = "UPDATE usuario SET nombre\_usu=?,usuario=?,password=?,email=?,codigo\_tipo\_usu=?,Codigo\_estado\_usu=? WHERE codigo\_usu= ?";

try {

PreparedStatement pstm = conn.getCon().prepareStatement(sql);

pstm.setInt(1, usuario.getCodigo\_usu());

pstm.setString(2, usuario.getNombre\_usu());

pstm.setString(3, usuario.getUsuario());

pstm.setString(4, usuario.getPassword());

pstm.setString(5, usuario.getEmail());

pstm.setString(6, usuario.getCodigo\_tipo\_usu());

pstm.setString(7, usuario.getCodigo\_estado\_usu());

pstm.executeUpdate();

} catch (SQLException excepcion) {

System.out.println("Ha ocurrido un error al eliminar " + excepcion.getMessage());

return null;

}

return usuario;

}

public String delUsuario(int codigo\_usu) {

conexion conn = new conexion();

String sql = "DELETE FROM usuario WHERE codigo\_usu= ?";

try {

PreparedStatement pstm = conn.getCon().prepareStatement(sql);

pstm.setInt(1, codigo\_usu);

pstm.executeUpdate();

} catch (SQLException excepcion) {

System.out.println("Ha ocurrido un error al eliminar " + excepcion.getMessage());

return "{\"Accion\":\"Error\"}";

}

return "{\"Accion\":\"Registro Borrado\"}";

}

}

Restful recurso:

package restful.Resources;

import java.util.ArrayList;

import javax.ws.rs.Consumes;

import javax.ws.rs.Produces;

import javax.ws.rs.GET;

import javax.ws.rs.POST;

import javax.ws.rs.Path;

import javax.ws.rs.PathParam;

import javax.ws.rs.core.MediaType;

import restful.Model.UsuarioModel;

import restful.Service.UsuarioServices;

import com.google.gson.Gson;

import com.google.gson.GsonBuilder;

import javax.ws.rs.DELETE;

import javax.ws.rs.PUT;

@Path("usuarios")

@Produces(MediaType.APPLICATION\_JSON)

@Consumes(MediaType.APPLICATION\_JSON)

public class UsuarioResouces {

UsuarioServices servicio=new UsuarioServices();

@GET

public ArrayList<UsuarioModel> getUsuario() {

return servicio.getusuario();

}

@Path("/{codigo\_usu}")

@GET

public UsuarioModel getUsuario(@PathParam("codigo\_usu") int codigo\_usu) {

return servicio.getUsuario(codigo\_usu);

}

@POST

public UsuarioModel addUsuario(String JSON) {

GsonBuilder builder = new GsonBuilder();

builder.setPrettyPrinting();

Gson gson = builder.create();

UsuarioModel usuario = gson.fromJson(JSON, UsuarioModel.class);

return servicio.addUsuario(usuario);

}

@DELETE

@Path("/{codigo\_usu}")

public String delUsuario(@PathParam("codigo\_usu") int codigo\_usu) {

return servicio.delUsuario(codigo\_usu);

}

@PUT

public UsuarioModel updateUsuario(String JSON) {

GsonBuilder builder = new GsonBuilder();

builder.setPrettyPrinting();

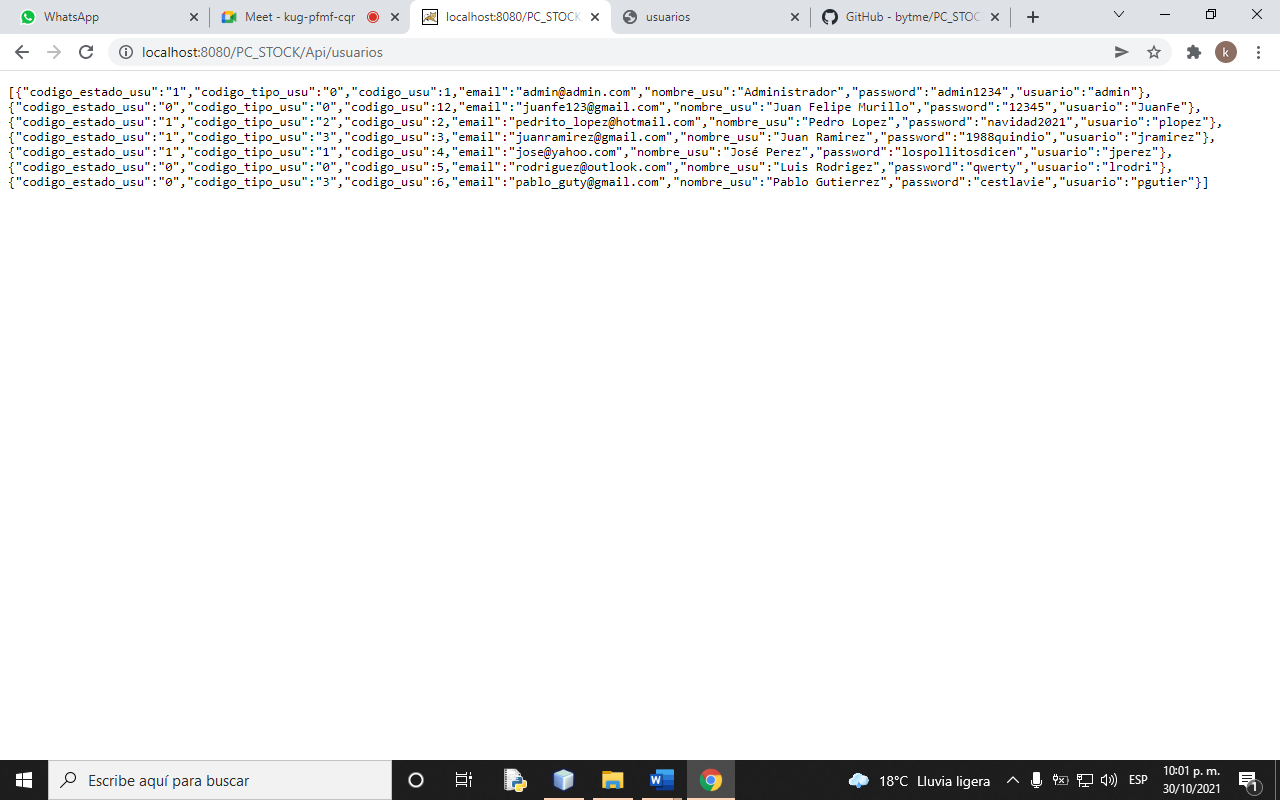
Gson gson = builder.create();

UsuarioModel usuario = gson.fromJson(JSON, UsuarioModel.class);

return servicio.updateUsuario(usuario);

}

}



1. Aplicación frontEnd.

Pagina Index

<!DOCTYPE html>

<html lang="es">

    <head>

        <title>usuarios</title>

        <meta charset="uft-8">

        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KyZXEAg3QhqLMpG8r+8fhAXLRk2vvoC2f3B09zVXn8CA5QIVfZOJ3BCsw2P0p/We" crossorigin="anonymous">

        <link rel="stylesheet" href="css/alertify.min.css" />

    </head>

    <body>

        <div class="container mt-4 shadow-lg p-3 mb-5  rounded ">

            <button id="btnCrear" type="button" class="btn btn-primary">Crear Usuario</button>

            <table id="pedidos" class="table mt-2 table-bordered table-striped">

              <thead>

                <tr class="text-center">

                  <th>CODIGO</th>

                  <th>NOMBRE</th>

                  <th>USUARIO</th>

                  <th>PASSSWORD</th>

                  <th>EMAIL</th>

                  <th>TIPO</th>

                  <th>ESTADO</th>

                  <th>ACCIONES</th>

                </tr>

              </thead>

              <tbody>

              </tbody>

            </table>

          </div>

          <div class="modal fade" id="modalUsuario" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

            <div class="modal-dialog">

              <div class="modal-content">

                <div class="modal-header bg-primary text-white">

                  <h5 class="modal-title" id="exampleModalLabel">Usuarios</h5>

                  <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

                </div>

                <div class="modal-body">

                  <form>

                    <div class="mb-3">

                      <label for="codigo\_usu" class="col-form-label">Codigo usuario :</label>

                      <input id="id" type="number" placeholder="Digite el código" class="form-control" id="recipient-name"

                        required autocomplete="off" oninvalid="setCustomValidity('Digite un Código')"

                        oninput="setCustomValidity('')">

                    </div>

                    <div class="mb-3">

                      <label for="nombre\_usu" class="col-form-label">Nombre usuario :</label>

                      <input id="nombre\_usu" type="text" placeholder="Digite el Nombre"  class="form-control" id="recipient-name"

                        required autocomplete="off" oninvalid="setCustomValidity('Digite el Nombre')"

                        oninput="setCustomValidity('')">

                    </div>

                    <div class="mb-3">

                      <label for="usuario" class="col-form-label">usuario :</label>

                      <input id="usuario" type="text" placeholder="Digite un usuario"  class="form-control"

                        id="recipient-name" required autocomplete="off" oninvalid="setCustomValidity('Digite un Usuario')"

                        oninput="setCustomValidity('')">

                    </div>

                    <div class="mb-3">

                        <label for="password" class="col-form-label">password :</label>

                        <input id="password" type="text" placeholder="Digite una clave"  class="form-control"

                          id="recipient-name" required autocomplete="off" oninvalid="setCustomValidity('Digite una clave')"

                          oninput="setCustomValidity('')">

                      </div>

                    </div>

                    <div class="mb-3">

                        <label for="email" class="col-form-label">Email :</label>

                        <input id="email" type="text" placeholder="Digite un Email"  class="form-control"

                          id="recipient-name" required autocomplete="off" oninvalid="setCustomValidity('Digite un Email')"

                          oninput="setCustomValidity('')">

                      </div>

                      <div class="mb-3">

                        <label for="codigo\_tipo\_usu" class="col-form-label">Tipo Usuario :</label>

                        <input id="codigo\_tipo\_usu" type="text" placeholder="Digite un Tipo Usuario"  class="form-control"

                          id="recipient-name" required autocomplete="off" oninvalid="setCustomValidity('Digite un Tipo Usuario')"

                          oninput="setCustomValidity('')">

                      </div>

                      <div class="mb-3">

                        <label for="codigo\_estado\_usu" class="col-form-label">Estado Usuario  :</label>

                        <input id="codigo\_estado\_usu" type="text" placeholder="Digite el Estado Usuario"  class="form-control"

                          id="recipient-name" required autocomplete="off" oninvalid="setCustomValidity('Digite el Estado Usuario')"

                          oninput="setCustomValidity('')">

                      </div>

                    <div class="modal-footer">

                      <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</button>

                      <button type="submit" class="btn btn-primary">Guardar</button>

                    </div>

                  </form>

                </div>

              </div>

            </div>

          </div>

        <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"

         integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"

         crossorigin="anonymous"></script>

        <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"

         integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx"

         crossorigin="anonymous"></script>

        <script src="js/alertify.min.js"></script>

        <script src="js/usuarios.js"></script>

    </body>

</html>

Pagina usuarios.js

const url = "http://localhost:8080/PC\_STOCK/Api/usuarios"

const contenedor = document.querySelector('tbody')

let resultados = ''

const modalUsuarios = new bootstrap.Modal(document.getElementById('modalUsuario'))

const formUsuarios = document.querySelector('form')

const nombreUsuario = document.getElementById('nombre\_usu')

const usuarioUsuario = document.getElementById('usuario')

const passwordUsuario = document.getElementById('password')

const emailUsuario = document.getElementById('email')

const tipoUsuario = document.getElementById('codigo\_tipo\_usu')

const estadoUsuario = document.getElementById('codigo\_estado\_usu')

const codigoUsuario = document.getElementById('id')

let opcion = ''

btnCrear.addEventListener('click', () => {

    nombreUsuario.value = ''

    usuarioUsuario.value = ''

    passwordUsuario.value = ''

    emailUsuario.value = ''

    tipoUsuario.value = ''

    estadoUsuario.value = ''

    codigoUsuario.value = ''

    codigoUsuario.disabled = false

    modalUsuarios.show()

    opcion = 'crear'

})

const ajax = (options) => {

    let { url, method, success, error, data } = options;

    const xhr = new XMLHttpRequest();

    xhr.addEventListener("readystatechange", (e) => {

        if (xhr.readyState !== 4) return;

        if (xhr.status >= 200 && xhr.status < 300) {

            let json = JSON.parse(xhr.responseText);

            success(json);

        } else {

            let message = xhr.statusText || "Ocurrió un error";

            error(`Error ${xhr.status}: ${message}`);

        }

    });

    xhr.open(method || "GET", url);

    xhr.setRequestHeader("Content-type", "application/json; charset=utf-8");

    xhr.send(JSON.stringify(data));

};

const getAll = () => {

    ajax({

        url: url,

        method: "GET",

        success: (res) => {

            console.log(res);

            res.forEach((usuarios) => {

                resultados += `<tr>

                        <td width="10%">${usuarios.codigo\_usu}</td>

                        <td width="15%">${usuarios.nombre\_usu}</td>

                        <td width="25%">${usuarios.usuario}</td>

                        <td width="25%">${usuarios.password}</td>

                        <td width="25%">${usuarios.email}</td>

                        <td width="25%">${usuarios.codigo\_tipo\_usu}</td>

                        <td width="25%">${usuarios.codigo\_estado\_usu}</td>

                        <td class="text-center" width="20%"><a class="btnEditar btn btn-primary">Editar</a><a class="btnBorrar btn btn-danger">Borrar</a></td>

                    </tr>`

            });

            contenedor.innerHTML = resultados

        },

        error: (err) => {

            console.log(err);

            $table.insertAdjacentHTML("afterend", `<p><b>${err}</b></p>`);

        },

    });

};

document.addEventListener("DOMContentLoaded", getAll);

document.addEventListener("click", (e) => {

    if (e.target.matches(".btnBorrar")) {

        const fila = e.target.parentNode.parentNode

        const id = fila.firstElementChild.innerHTML

        console.log(id)

        alertify.confirm(`Estas seguro de eliminar el id ${id}?`,

            function () {

                ajax({

                    url: url + "/" + id,

                    method: "DELETE",

                    headers: {

                        'Content-Type': 'application/json'

                    },

                    success: (\_res) => location.reload(),

                    error: (err) => alert(err),

                });

                alertify.success('Registro eliminado')

            },

            function () {

                alertify.error('Cancel')

            });

    }

    if (e.target.matches(".btnEditar")) {

        const fila = e.target.parentNode.parentNode

        codigoUsuario.value = fila.children[0].innerHTML

        nombreUsuario.value = fila.children[1].innerHTML

        usuarioUsuario.value = fila.children[2].innerHTML

        passwordUsuario.value = fila.children[3].innerHTML

        emailUsuario.value = fila.children[4].innerHTML

        tipoUsuario.value = fila.children[5].innerHTML

        estadoUsuario.value = fila.children[6].innerHTML

        codigoUsuario.disabled = true

        opcion = 'editar'

        modalUsuarios.show()

    }

})

const fila = e.target.parentNode.parentNode

codigoUsuario.value = fila.children[0].innerHTML

formUsuarios.addEventListener('submit', (e) => {

    e.preventDefault()

    let metodo = "POST"

    if (opcion == 'editar') {

        metodo = "PUT"

    }

    ajax({

        url: url,

        method: metodo,

        headers: {

            'Content-Type': 'application/json'

        },

        success: (\_res) => location.reload(),

        error: (err) =>

            $form.insertAdjacentHTML("afterend", `<p><b>${err}</b></p>`),

        data: {

            "codigo\_estado\_usu": estadoUsuario.value,

            "codigo\_tipo\_usu": tipoUsuario.value,

            "codigo\_usu": codigoUsuario.value,

            "email": emailUsuario.value,

            "nombre\_usu": nombreUsuario.value,

            "password": passwordUsuario.value,

            "usuario": usuarioUsuario.value

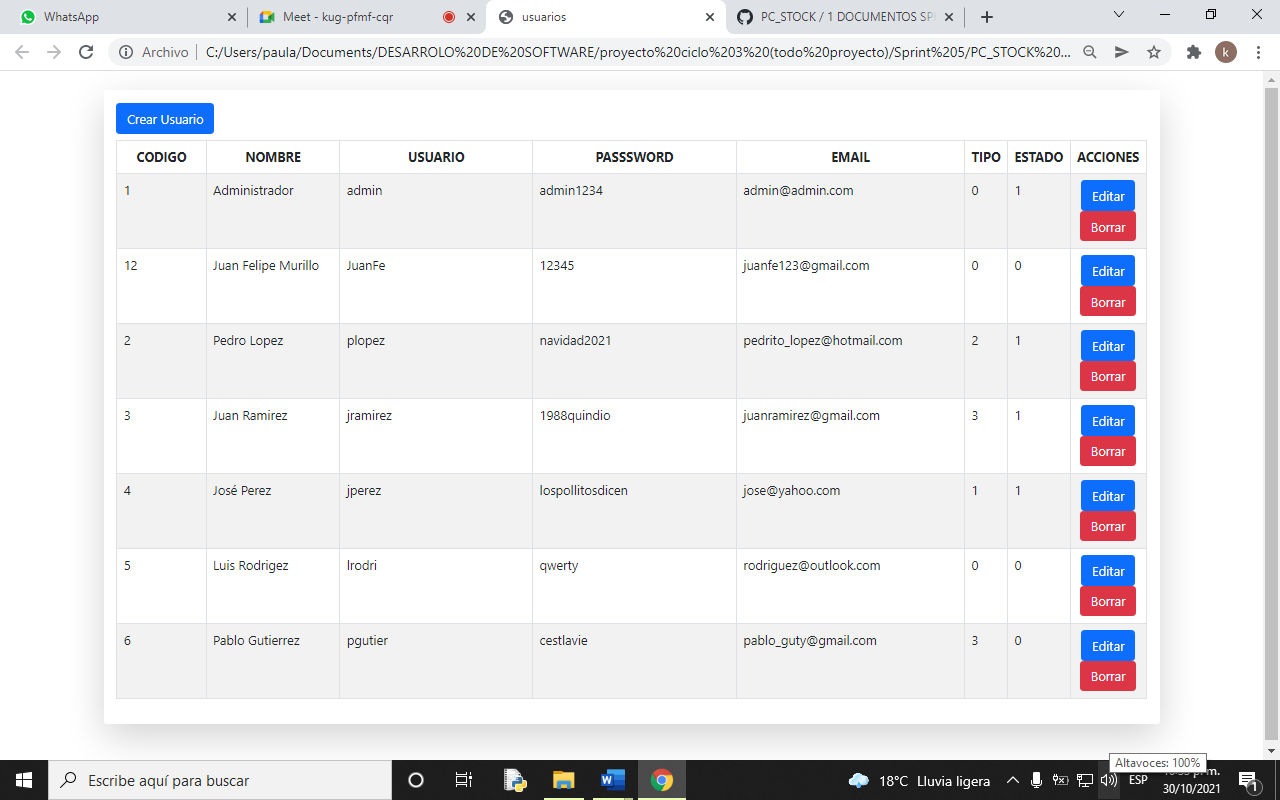
                },

    });

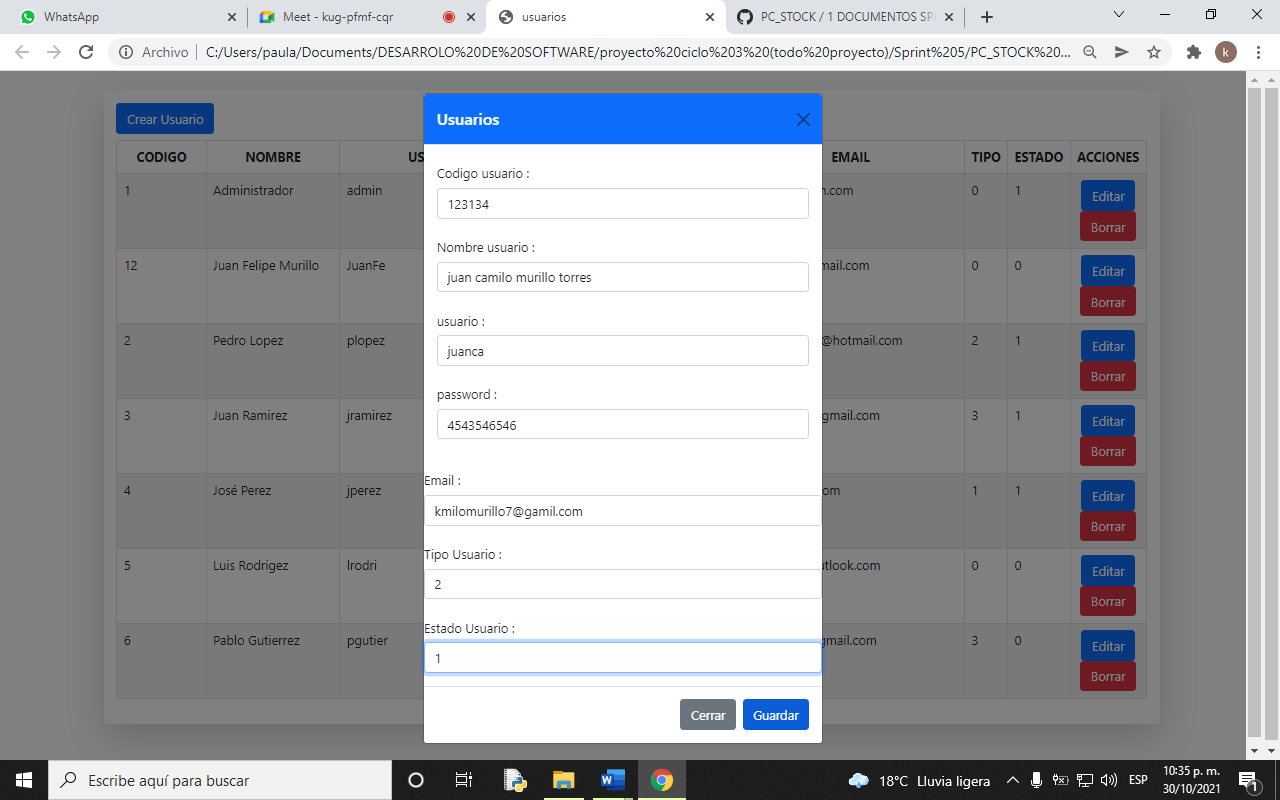
    modalUsuarios.hide()

})

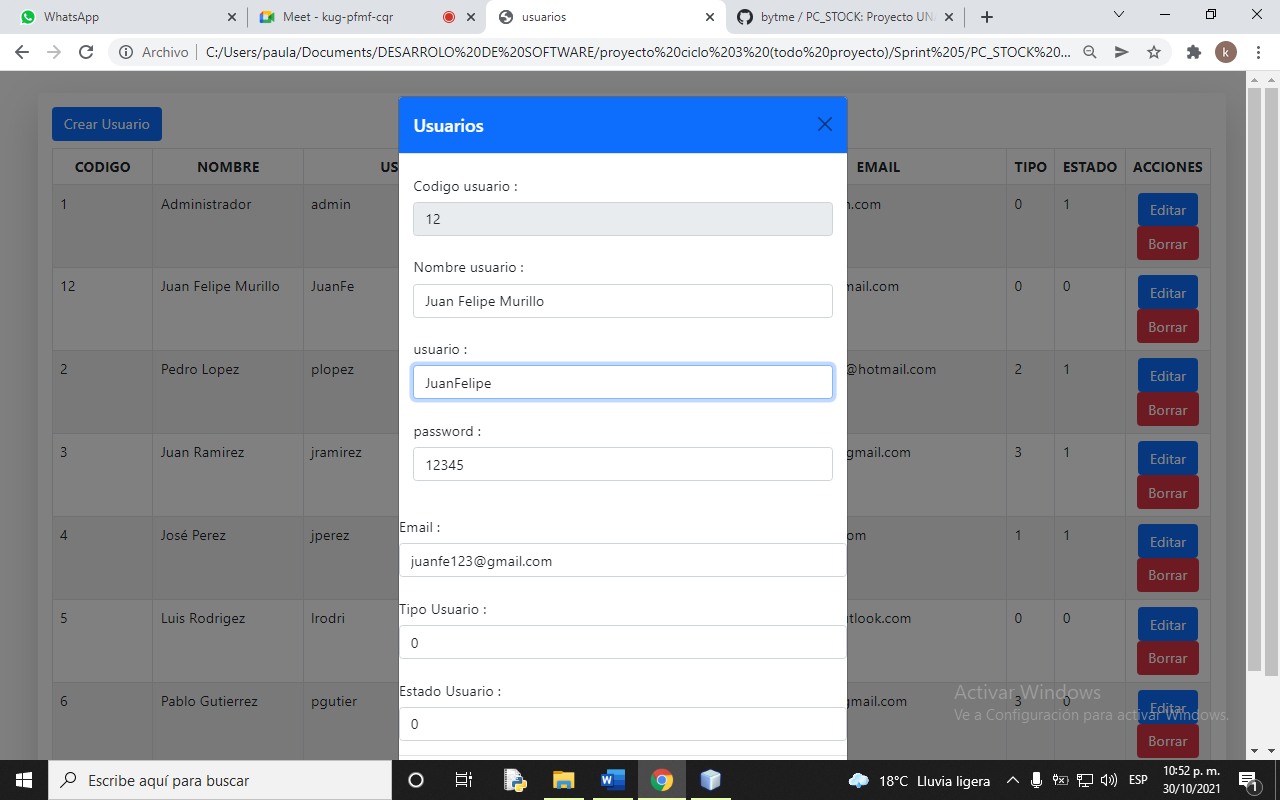
1. Imágenes evidencias.
2. Backend y Frontend en Github:
3. Pantallazos ejecucion del backend:
4. Pantallazos ejecucion del frontend:



Crear usuario



Modificar usuario



Eliminar usuario.

