**TECNOLÓGICO NACIONAL DE MÉXICO**

**Instituto Tecnológico de León**



**INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**PROGRAMACIÓN WEB AVANZADA**

**7:00 – 8:40, martes, jueves, viernes**

**PROYECTO INDIVIDUAL**

**“Videojuegos”**

**PRESENTA:**

**JÉSSICA RAMÍREZ SÁNCHEZ**

****

**CON LA ASESORÍA DE:**

**LIC. ANGÉLICA MARÍA ORTIZ GAUCIN**

ÍNDICE

[DESCRIPCIÓN DEL PROBLEMA 4](#_Toc40803331)

[ANÁLISIS DEL PROBLEMA 4](#_Toc40803332)

[OBJETIVOS 4](#_Toc40803333)

[OBJETIVO GENERAL 4](#_Toc40803334)

[OBJETIVOS ESPECÍFICOS 4](#_Toc40803335)

[ALCANCES 4](#_Toc40803336)

[LIMITACIONES 4](#_Toc40803337)

[DIAGRAMA RELACIONAL DE LA BASE DE DATOS 5](#_Toc40803338)

[DICCIONARIO DE DATOS 5](#_Toc40803339)

[Tabla “videojuego” 5](#_Toc40803340)

[Tabla “usuario” 5](#_Toc40803341)

[SCRIPT DE LA BASE DE DATOS 5](#_Toc40803342)

[DESARROLLO DEL SISTEMA 6](#_Toc40803343)

[Carpeta model 7](#_Toc40803344)

[Clase ConectaBD.java 7](#_Toc40803345)

[Clase GestorBD.java 7](#_Toc40803346)

[Clase Usuario.java 12](#_Toc40803347)

[Clase Videojuego.java 14](#_Toc40803348)

[Carpeta controller 16](#_Toc40803349)

[Clase ControllerManagedBean.java 16](#_Toc40803350)

[Clase LoginController.java 20](#_Toc40803351)

[Carpeta filter 23](#_Toc40803352)

[Clase AuthorizationFilter.java 23](#_Toc40803353)

[Carpeta utils 24](#_Toc40803354)

[Clase SessionUtils.java 24](#_Toc40803355)

[Web Pages 25](#_Toc40803356)

[Vista login.xhtml 25](#_Toc40803357)

[Vista errorLogin.xhtml 26](#_Toc40803358)

[Vista videojuegos.xhtml 27](#_Toc40803359)

[Vista agregar.xhtml 29](#_Toc40803360)

[Vista localizar.xhtml 31](#_Toc40803361)

[Vista modificar.xhtml 32](#_Toc40803362)

[Vista borrar.xhtml 34](#_Toc40803363)

[Vista error.xhtml 35](#_Toc40803364)

# DESCRIPCIÓN DEL PROBLEMA

Se lleva el control de inventario de videojuegos de manera manual, pero ocurren problemas al momento de quién agrega los videojuegos.

# ANÁLISIS DEL PROBLEMA

Se llevará a cabo el desarrollo de un sistema que permita realizar las altas, bajas y modificaciones de los videojuegos, además, permitirá hacer la distinción de los entre dos roles de usuario dando permisos de las acciones del sistema.

# OBJETIVOS

# OBJETIVO GENERAL

Desarrollar un portal web para la gestión de videojuegos.

# OBJETIVOS ESPECÍFICOS

* Realizar el esquema de base de datos
* Implementar la autenticación de usuarios
* Autorizar permisos a los usuarios
* Diseñar un sistema amigable al usuario

# ALCANCES

El sistema realizará las altas, bajas y modificaciones únicamente de videojuegos. Los usuarios que tendrán acceso al sistema estarán registrados directamente desde la base de datos.

# LIMITACIONES

Los usuarios que deban tener acceso al sistema deberán ser registrados desde la base de datos, en una fase a futura se piensa desarrollar los módulos de usuarios y roles permitiendo mayor flexibilidad de acceso al sistema.

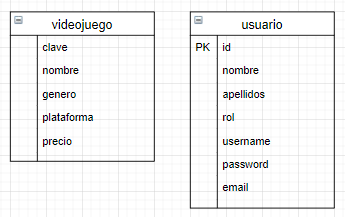
# REQUERIMIENTOS DE HARDWARE

* Ninguno en específico

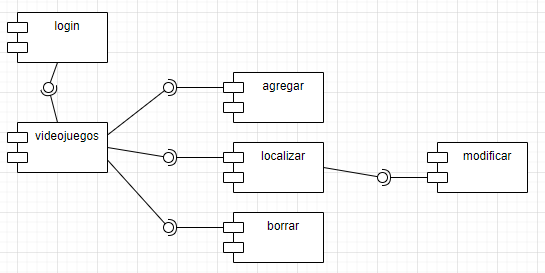
# REQUERIMIENTOS DE SOFTWARE

* Java Enterprise
* Servidor Apache
* Navegador

# DIAGRAMA RELACIONAL DE LA BASE DE DATOS



# DIAGRAMA DE COMPONENTES



# DICCIONARIO DE DATOS

## Tabla “videojuego”

|  |  |  |  |
| --- | --- | --- | --- |
| **Campo** | **Tipo** | **Longitud** | **Descripción** |
| clave | Int | 6 | Clave numérica del videojuego |
| nombre | Varchar | 150 | Nombre del videojuego |
| genero | Varchar | 150 | Género al que pertenece el videojuego |
| plataforma | Varchar | 150 | Plataforma en la que se puede jugar |
| precio | Varchar | 150 | Precio de venta |

## Tabla “usuario”

|  |  |  |  |
| --- | --- | --- | --- |
| **Campo** | **Tipo** | **Longitud** | **Descripción** |
| id | Bigint | 64 | Identificador de usuario |
| nombre | Varchar | 100 | Nombre del usuario |
| apellidos | Varchar | 100 | Apellidos de usuario |
| rol | Varchar | 20 | Nombre del rol de usuario |
| username | Varchar | 100 | Username |
| password | Varchar | 250 | Contraseña del usuario |
| email | Varchar | 100 | Email del usuario |

# SCRIPT DE LA BASE DE DATOS

CREATE DATABASE videojuegos;

USE videojuegos;

CREATE TABLE videojuego(

clave int(6),

nombre varchar(150),

genero varchar(150),

plataforma varchar(150),

precio varchar(150)

);

INSERT INTO videojuego VALUES(029010, 'Cities Skylines', 'Constructor', 'PC', '300');

INSERT INTO videojuego VALUES(108249, 'Celeste', 'Indie', 'PC/Nintendo Switch', '500');

INSERT INTO videojuego VALUES(120945, 'Bad North Jotunn Edition', 'Combate', 'PC', '200');

INSERT INTO videojuego VALUES(092375, 'Fox Game', 'RPG', 'PC', '900');

INSERT INTO videojuego VALUES(834902, 'Mortal Kombat', 'Combate', 'PC', '599');

CREATE TABLE usuario(

id bigint not null auto\_increment primary key,

nombre varchar(100) not null,

apellidos varchar(100) not null,

rol varchar(20) not null,

username varchar(100) not null,

password varchar(250) not null,

email varchar(100) not null

);

INSERT INTO usuario(nombre, apellidos, rol, username, password, email)

VALUES('Jessica', 'Ramirez', 'admin', 'jessica', 'una\_facil', 'jessicaramsa@gmail.com');

INSERT INTO usuario(nombre, apellidos, rol, username, password, email)

VALUES('Angélica', 'Ortiz', 'prueba', 'prueba', 'una\_facil', 'prueba@gmail.com');

# ESTRUCTURA MVC

La gestión del código estará compuesta por la estructura de MVC con ayuda de algunos otros paquetes que nos serán de utilidad.



La gestión de las vistas no tendrá ningún orden especial, debido a que todas se encontrarán en la carpeta de Web Pages de nuestro proyecto web.

# DESARROLLO DEL SISTEMA

El proyecto será una aplicación web JSF. Tendremos una distribución de las capas al estilo MVC tal que el árbol de directorios quede de la siguiente manera:



## Carpeta model

### Clase ConectaBD.java

En esta clase realizaremos la conexión a la base de datos.

public class ConectaBD {

private Connection conexion = null;

private String servidor = "localhost";

private String database = "videojuegos";

private String usuario = "root";

private String password = "";

private String url = "";

public ConectaBD() {

try {

Class.forName("com.mysql.jdbc.Driver");

url = "jdbc:mysql://" + servidor + "/" + database;

conexion = DriverManager.getConnection(url, usuario, password);

} catch(SQLException | ClassNotFoundException ex) {

System.out.println(ex);

}

}

public Connection getConexion() {

return conexion;

}

public Connection cerrarConexion() {

try {

conexion.close();

} catch(SQLException sqle) {

System.out.println(sqle);

}

conexion = null;

return conexion;

}

}

### Clase GestorBD.java

Esta clase se encargará de realizar las operaciones con la base de datos recibiendo los debidos parámetros para hacer las altas, bajas, modificaciones y consultas.

public class GestorBD {

private Connection conexion = null;

private Statement stm = null;

private ResultSet videojuegoResultSet, usuarioResultSet;

private Integer clave;

private String nombre, genero, plataforma, precio;

private int resultUpdate = 0;

public ArrayList<Videojuego> leerVideojuegos() {

ArrayList<Videojuego> videojuego = new ArrayList<>();

Videojuego videojuegoEncontrado;

try {

ConectaBD conectaBD = new ConectaBD();

conexion = conectaBD.getConexion();

stm = conexion.createStatement();

videojuegoResultSet = stm.executeQuery("SELECT \* FROM videojuego;");

if (!videojuegoResultSet.next()) {

System.out.println("No se encontraron registros.");

conexion.close();

return null;

} else {

do {

clave = videojuegoResultSet.getInt("clave");

nombre = videojuegoResultSet.getString("nombre");

genero = videojuegoResultSet.getString("genero");

plataforma = videojuegoResultSet.getString("plataforma");

precio = videojuegoResultSet.getString("precio");

videojuegoEncontrado = new Videojuego(clave, nombre, genero, plataforma, precio);

videojuego.add(videojuegoEncontrado);

} while(videojuegoResultSet.next());

conexion.close();

return videojuego;

}

} catch(SQLException sqle) {

System.out.println("Error en la base de datos.");

sqle.printStackTrace();

return null;

}

}

public boolean guardarVideojuego(Videojuego videojuegoNuevo) {

try {

ConectaBD conectaBD = new ConectaBD();

conexion = conectaBD.getConexion();

stm = conexion.createStatement();

resultUpdate = stm.executeUpdate("INSERT INTO videojuego VALUES("

+ videojuegoNuevo.getClave() + ", "

+ "'" + videojuegoNuevo.getNombre() + "', "

+ "'" + videojuegoNuevo.getGenero() + "', "

+ "'" + videojuegoNuevo.getPlataforma() + "', "

+ "'" + videojuegoNuevo.getPrecio() + "');"

);

if (resultUpdate != 0) {

conexion.close();

return true;

} else {

conexion.close();

System.out.println("No se pudo insertar el videojuego.");

return false;

}

} catch(SQLException sqle) {

System.out.println("Error en la base de datos.");

sqle.printStackTrace();

return false;

}

}

public boolean borrarVideojuego(Videojuego videojuegoABorrar) {

try {

ConectaBD conectaBD = new ConectaBD();

conexion = conectaBD.getConexion();

stm = conexion.createStatement();

resultUpdate = stm.executeUpdate("DELETE FROM videojuego WHERE("

+ "clave = " + videojuegoABorrar.getClave()

+ " AND nombre = '" + videojuegoABorrar.getNombre() + "');"

);

if (resultUpdate != 0) {

conexion.close();

return true;

} else {

conexion.close();

System.out.println("No se pudo borrar el videojuego.");

return false;

}

} catch(SQLException sqle) {

System.out.println("Error en la base de datos.");

sqle.printStackTrace();

return false;

}

}

public boolean localizaVideojuego(Integer clave, String nombre) {

try {

ConectaBD conectaBD = new ConectaBD();

conexion = conectaBD.getConexion();

stm = conexion.createStatement();

videojuegoResultSet = stm.executeQuery("SELECT \* FROM videojuego"

+ " WHERE (clave = " + clave

+ " AND nombre = '" + nombre + "');"

);

if (!videojuegoResultSet.next()) {

System.out.println("No se encontraron registros.");

conexion.close();

return false;

} else {

conexion.close();

return true;

}

} catch(SQLException sqle) {

System.out.println("Error en la base de datos.");

sqle.printStackTrace();

return false;

}

}

public Videojuego localizaVideojuegoCompleto(Integer clave, String nombre) {

try {

ConectaBD conectaBD = new ConectaBD();

conexion = conectaBD.getConexion();

stm = conexion.createStatement();

videojuegoResultSet = stm.executeQuery("SELECT \* FROM videojuego"

+ " WHERE (clave = " + clave

+ " AND nombre = '" + nombre + "');"

);

if (!videojuegoResultSet.next()) {

System.out.println("No se encontraron registros.");

conexion.close();

return null;

} else {

Videojuego videojuegoNuevo = new Videojuego(

clave,

nombre,

videojuegoResultSet.getString("genero"),

videojuegoResultSet.getString("plataforma"),

videojuegoResultSet.getString("precio")

);

conexion.close();

return videojuegoNuevo;

}

} catch(SQLException sqle) {

System.out.println("Error en la base de datos.");

sqle.printStackTrace();

return null;

}

}

public boolean modificarVideojuego(Videojuego videojuegoACambiar) {

try {

ConectaBD conectaBD = new ConectaBD();

conexion = conectaBD.getConexion();

stm = conexion.createStatement();

resultUpdate = stm.executeUpdate("UPDATE videojuego SET "

+ "nombre = '" + videojuegoACambiar.getNombre() + "', "

+ "genero = '" + videojuegoACambiar.getGenero() + "', "

+ "plataforma = '" + videojuegoACambiar.getPlataforma() + "', "

+ "precio = '" + videojuegoACambiar.getPrecio() + "' "

+ "WHERE clave = " + videojuegoACambiar.getClave() + ";"

);

if (resultUpdate != 0) {

conexion.close();

return true;

} else {

conexion.close();

System.out.println("No se pudo modificar el videojuego.");

return false;

}

} catch(SQLException sqle) {

System.out.println("Error en la base de datos.");

sqle.printStackTrace();

return false;

}

}

public boolean localizaUsuario(String username, String password) {

try {

ConectaBD conectaBD = new ConectaBD();

conexion = conectaBD.getConexion();

stm = conexion.createStatement();

usuarioResultSet = stm.executeQuery("SELECT \* FROM usuario"

+ " WHERE (username = '" + username + "'"

+ " AND password = '" + password + "');"

);

if (!usuarioResultSet.next()) {

System.out.println("No se encontraron registros.");

conexion.close();

return false;

} else {

conexion.close();

return true;

}

} catch(SQLException sqle) {

System.out.println("Error en la base de datos.");

sqle.printStackTrace();

return false;

}

}

public String localizaRolUsuario(String username) {

try {

ConectaBD conectaBD = new ConectaBD();

conexion = conectaBD.getConexion();

stm = conexion.createStatement();

usuarioResultSet = stm.executeQuery("SELECT \* FROM usuario"

+ " WHERE username = '" + username + "';"

);

if (!usuarioResultSet.next()) {

System.out.println("No se encontraron registros.");

conexion.close();

return null;

} else {

String rol = usuarioResultSet.getString("rol");

conexion.close();

return rol;

}

} catch(SQLException sqle) {

System.out.println("Error en la base de datos.");

sqle.printStackTrace();

return null;

}

}

}

### Clase Usuario.java

Esta clase nos ayudará a mantener la persistencia con la base de datos de acuerdo a las instancias de Alumno que se utilizarán para las diferentes operaciones.

public class Usuario {

private BigInteger id;

private String nombre;

private String apellidos;

private String username;

private String email;

private String password;

private String rol;

public Usuario(String nombre, String apellidos, String username, String email, String password, String rol) {

this.nombre = nombre;

this.apellidos = apellidos;

this.username = username;

this.email = email;

this.password = password;

this.rol = rol;

}

public BigInteger getId() {

return id;

}

public void setId(BigInteger id) {

this.id = id;

}

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public String getApellidos() {

return apellidos;

}

public void setApellidos(String apellidos) {

this.apellidos = apellidos;

}

public String getEmail() {

return email;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getRol() {

return rol;

}

public void setRol(String rol) {

this.rol = rol;

}

}

### Clase Videojuego.java

Esta clase nos ayudará a mantener la persistencia con la base de datos de acuerdo a las instancias de Videojuego que se utilizarán para las diferentes operaciones.

public class Videojuego {

private Integer clave;

private String nombre;

private String genero;

private String plataforma;

private String precio;

public Videojuego(Integer clave, String nombre, String genero, String plataforma, String precio) {

this.clave = clave;

this.nombre = nombre;

this.genero = genero;

this.plataforma = plataforma;

this.precio = precio;

}

public Integer getClave() {

return clave;

}

public void setClave(Integer clave) {

this.clave = clave;

}

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public String getGenero() {

return genero;

}

public void setGenero(String genero) {

this.genero = genero;

}

public String getPlataforma() {

return plataforma;

}

public void setPlataforma(String plataforma) {

this.plataforma = plataforma;

}

public String getPrecio() {

return precio;

}

public void setPrecio(String precio) {

this.precio = precio;

}

}

## Carpeta controller

Esta carpeta contendrá los controladores. Clase que se encargará de comunicarse con la clase modelo de GestorBD y realizar las peticiones con la base de datos.

### Clase ControllerManagedBean.java

@ManagedBean

@SessionScoped

public class ControllerManagedBean {

private Integer clave;

private String nombre;

private String genero;

private String plataforma;

private String precio;

private GestorBD gestorBD;

private static ArrayList<Videojuego> videojuegosList;

public ControllerManagedBean() {

gestorBD = new GestorBD();

videojuegosList = gestorBD.leerVideojuegos();

}

public Integer getClave() {

return clave;

}

public String getNombre() {

return nombre;

}

public String getGenero() {

return genero;

}

public String getPlataforma() {

return plataforma;

}

public String getPrecio() {

return precio;

}

public ArrayList<Videojuego> getVideojuegosList() {

return videojuegosList;

}

public void setClave(Integer clave) {

this.clave = clave;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public void setGenero(String genero) {

this.genero = genero;

}

public void setPlataforma(String plataforma) {

this.plataforma = plataforma;

}

public void setPrecio(String precio) {

this.precio = precio;

}

public GestorBD getGestorBD() {

return gestorBD;

}

public void setGestorBD(GestorBD gestorBD) {

this.gestorBD = gestorBD;

}

public void setVideojuegosList(ArrayList<Videojuego> videojuegosList) {

ControllerManagedBean.videojuegosList = videojuegosList;

}

public void pedirDatosVideojuego\_aAgregar() {

try {

FacesContext.getCurrentInstance()

.getExternalContext()

.redirect("agregar.xhtml");

} catch(IOException ioe) {

Logger.getLogger(ControllerManagedBean.class.getName()).log(Level.SEVERE, null, ioe);

}

}

public void guardarVideojuego() {

Videojuego videojuegoNuevo = new Videojuego(clave, nombre, genero, plataforma, precio);

if (gestorBD.guardarVideojuego(videojuegoNuevo)) {

try {

videojuegosList = gestorBD.leerVideojuegos();

FacesContext.getCurrentInstance()

.getExternalContext()

.redirect("videojuegos.xhtml");

} catch(IOException ioe) {

Logger.getLogger(ControllerManagedBean.class.getName()).log(Level.SEVERE, null, ioe);

}

}

}

public void pedirDatosVideojuego\_aBorrar() {

try {

FacesContext.getCurrentInstance()

.getExternalContext()

.redirect("borrar.xhtml");

} catch(IOException ioe) {

Logger.getLogger(ControllerManagedBean.class.getName()).log(Level.SEVERE, null, ioe);

}

}

public void borrarVideojuego() {

Videojuego videojuegoABorrar = new Videojuego(clave, nombre, genero, plataforma, precio);

if (gestorBD.borrarVideojuego(videojuegoABorrar)) {

try {

videojuegosList = gestorBD.leerVideojuegos();

FacesContext.getCurrentInstance()

.getExternalContext()

.redirect("videojuegos.xhtml");

} catch(IOException ioe) {

Logger.getLogger(ControllerManagedBean.class.getName()).log(Level.SEVERE, null, ioe);

}

}

}

public void pedirDatosVideojuego\_aLocalizar() {

try {

FacesContext.getCurrentInstance()

.getExternalContext()

.redirect("localizar.xhtml");

} catch(IOException ioe) {

Logger.getLogger(ControllerManagedBean.class.getName()).log(Level.SEVERE, null, ioe);

}

}

public void localizarVideojuego() {

if (gestorBD.localizaVideojuego(clave, nombre)) {

try {

Videojuego videojuegoNuevo = gestorBD.localizaVideojuegoCompleto(clave, nombre);

genero = videojuegoNuevo.getGenero();

plataforma = videojuegoNuevo.getPlataforma();

precio = videojuegoNuevo.getPrecio();

FacesContext.getCurrentInstance()

.getExternalContext()

.redirect("modificar.xhtml");

} catch(IOException ioe) {

Logger.getLogger(ControllerManagedBean.class.getName()).log(Level.SEVERE, null, ioe);

}

} else {

try {

videojuegosList = gestorBD.leerVideojuegos();

FacesContext.getCurrentInstance()

.getExternalContext()

.redirect("error.xhtml");

} catch(IOException ioe) {

Logger.getLogger(ControllerManagedBean.class.getName()).log(Level.SEVERE, null, ioe);

}

}

}

public void modificarVideojuego() {

Videojuego videojuegoACambiar = new Videojuego(clave, nombre, genero, plataforma, precio);

if (gestorBD.modificarVideojuego(videojuegoACambiar)) {

try {

videojuegosList = gestorBD.leerVideojuegos();

FacesContext.getCurrentInstance()

.getExternalContext()

.redirect("videojuegos.xhtml");

} catch(IOException ioe) {

Logger.getLogger(ControllerManagedBean.class.getName()).log(Level.SEVERE, null, ioe);

}

}

}

}

### Clase LoginController.java

@ManagedBean

@SessionScoped

public class LoginController {

private BigInteger id;

private String nombre;

private String apellidos;

private String rol;

private String password;

private String username;

private String email;

private GestorBD gestorBD;

public LoginController() {

gestorBD = new GestorBD();

}

public BigInteger getId() {

return id;

}

public void setId(BigInteger id) {

this.id = id;

}

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public String getApellidos() {

return apellidos;

}

public void setApellidos(String apellidos) {

this.apellidos = apellidos;

}

public String getRol() {

return rol;

}

public void setRol(String rol) {

this.rol = rol;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public GestorBD getGestorBD() {

return gestorBD;

}

public void setGestorBD(GestorBD gestorBD) {

this.gestorBD = gestorBD;

}

public String validarLogin() {

GestorBD gestorBD = new GestorBD();

boolean valid = gestorBD.localizaUsuario(username, password);

if (valid) {

rol = gestorBD.localizaRolUsuario(username);

HttpSession session = SessionUtils.getSession();

session.setAttribute("username", username);

session.setAttribute("rol", rol);

return "videojuegos";

} else {

FacesContext.getCurrentInstance().addMessage(

null,

new FacesMessage(

FacesMessage.SEVERITY\_WARN,

"Usuario o contraseña incorrectos",

"Por favor introduce el username y cotraseña"

)

);

return "login";

}

}

public boolean validarRol(String rol) {

return true;

}

public String logout() {

HttpSession session = SessionUtils.getSession();

session.invalidate();

return "login";

}

}

## Carpeta filter

### Clase AuthorizationFilter.java

Esta clase realiza el filtro de autorización de los usuarios.

public class AuthorizationFilter implements Filter {

private static final boolean debug = true;

private FilterConfig filterConfig = null;

public AuthorizationFilter() { }

@Override

public void doFilter(ServletRequest request, ServletResponse response,

FilterChain chain) throws IOException, ServletException {

try {

HttpServletRequest req = (HttpServletRequest) request;

HttpServletResponse res = (HttpServletResponse) response;

HttpSession session = req.getSession(false);

String reqURI = req.getRequestURI();

if (reqURI.indexOf("/login.xhtml") >= 0

|| (session != null && session.getAttribute("username") != null)

|| reqURI.indexOf("/public/") >= 0

|| reqURI.contains("javax.faces.resource")) {

chain.doFilter(request, response);

} else {

res.sendRedirect(req.getContextPath() + "/faces/login.xhtml");

}

} catch (IOException | ServletException ex) {

System.out.println(ex.getMessage());

}

}

public FilterConfig getFilterConfig() {

return (this.filterConfig);

}

public void setFilterConfig(FilterConfig filterConfig) {

this.filterConfig = filterConfig;

}

@Override

public void destroy() {}

@Override

public void init(FilterConfig filterConfig) { }

@Override

public String toString() {

if (filterConfig == null) {

return ("AuthorizationFilter()");

}

StringBuffer sb = new StringBuffer("AuthorizationFilter(");

sb.append(filterConfig);

sb.append(")");

return (sb.toString());

}

}

## Carpeta utils

### Clase SessionUtils.java

Se encargará de realizar las peticiones para obtener el id del usuario y loguearse.

public class SessionUtils {

public static HttpSession getSession() {

return (HttpSession) FacesContext.getCurrentInstance()

.getExternalContext().getSession(false);

}

public static HttpServletRequest getRequest() {

return (HttpServletRequest) FacesContext.getCurrentInstance()

.getExternalContext().getRequest();

}

public static String getUsername() {

HttpSession session = (HttpSession) FacesContext.getCurrentInstance()

.getExternalContext().getSession(false);

return session.getAttribute("username").toString();

}

public static String getUserId() {

HttpSession session = getSession();

if (session != null) {

return (String) session.getAttribute("userid");

} else {

return null;

}

}

}

## Web Pages

Todas las vistas son descriptivas en su nombre.

### Vista login.xhtml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://xmlns.jcp.org/jsf/html">

<head>

<title>Login</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous"/>

<link rel="stylesheet" href="#{request.contextPath}/resources/styles/bootstrap.min.css"/>

<style>

form {

padding: 2em;

margin-top: 2em;

border-radius: 5px;

}

</style>

</head>

<body>

<div class="container">

<div class="row justify-content-center">

<div class="col-12 col-md-6">

<h:form>

<h1 class="text-center">Iniciar sesión</h1><br/>

<div class="form-group">

<h:outputLabel for="username">Username</h:outputLabel>

<h:inputText id="username" value="#{loginController.username}" class="form-control" required="true"/>

</div>

<div class="form-group">

<h:outputLabel for="password">Contraseña</h:outputLabel>

<h:inputSecret id="password" value="#{loginController.password}" class="form-control" required="true"/>

</div>

<h:commandButton value="Login" type="submit" action="#{loginController.validarLogin}" class="btn btn-primary btn-block"/>

</h:form>

</div>

</div>

</div>

</body>

</html>

### Vista errorLogin.xhtml

Página de error en caso de que en el login se introduzcan datos incorrectos.

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://xmlns.jcp.org/jsf/html">

<head>

<title>Error</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous"/>

<link rel="stylesheet" href="#{request.contextPath}/resources/styles/bootstrap.min.css"/>

<style>

form {

padding: 2em;

margin-top: 2em;

border-radius: 5px;

}

</style>

</head>

<body>

<div class="row justify-content-center">

<div class="col-12 col-md-6">

<h:form id="form">

<img src="#{request.contextPath}/resources/imgs/error.png" width="80" height="80"/>

<h2 class="text-center">ERROR, no se encontró un usuario con:</h2>

<div class="form-group">

<span>Username</span>

<h:outputText value="#{LoginController.username}" class="form-control"/>

</div>

</h:form>

<h:form class="form-group">

<h:commandButton value="Regresar" action="faces/login.xhtml" class="btn btn-primary btn-block"/>

</h:form>

</div>

</div>

</body>

</html>

### Vista videojuegos.xhtml

Vista de tabla de videojuegos.

<?xml version='1.0' encoding='UTF-8' ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://xmlns.jcp.org/jsf/html"

xmlns:f="http://xmlns.jcp.org/jsf/core"

xmlns:c="http://xmlns.jcp.org/jsp/jstl/core"

xmlns:ui="http://java.sun.com/jsf/facelets">

<head>

<title>Videojuegos</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous"/>

<link rel="stylesheet" href="#{request.contextPath}/resources/styles/bootstrap.min.css"/>

<style>

form {

padding: 2em;

margin-top: 2em;

border-radius: 5px;

}

</style>

</head>

<body>

<nav class="navbar navbar-dark bg-dark fixed-top">

<p class="navbar-brand">¡Hola, #{loginController.username}!</p>

<h:form style="margin-top: -1em" class="form-inline navbar-text my-2 my-lg-0">

<h:commandLink value="Videojuegos" action="faces/videojuegos.xhtml" class="nav-item nav-link"/>

</h:form>

<h:form class="form-inline navbar-text my-2 my-lg-0">

<h:commandLink action="#{loginController.logout}" value="Cerrar sesión"

class="my-2 my-sm-0"/>

</h:form>

</nav>

<div class="container" style="padding-top: 2em">

<div class="row justify-content-center">

<div class="">

<h:form>

<h:dataTable value="#{controllerManagedBean.getVideojuegosList()}"

var="videojuego"

styleClass="table table-striped table-dark"

headerClass="order-table-header"

rowClasses="order-table-odd-row,order-table-even-row">

<h:column>

<f:facet name="header">Clave</f:facet>

#{videojuego.clave}

</h:column>

<h:column>

<f:facet name="header">Nombre</f:facet>

#{videojuego.nombre}

</h:column>

<h:column>

<f:facet name="header">Género</f:facet>

#{videojuego.genero}

</h:column>

<h:column>

<f:facet name="header">Plataforma</f:facet>

#{videojuego.plataforma}

</h:column>

<h:column>

<f:facet name="header">Precio</f:facet>

#{videojuego.precio}

</h:column>

</h:dataTable>

<h:commandButton value="Agregar videojuego" type="submit"

action="#{controllerManagedBean.pedirDatosVideojuego\_aAgregar()}"

class="btn btn-success btn-block"/>

<c:if test="#{loginController.rol eq 'admin'}">

<h:commandButton value="Borrar videojuego" type="submit"

action="#{controllerManagedBean.pedirDatosVideojuego\_aBorrar()}"

class="btn btn-danger btn-block"/>

<h:commandButton value="Modificar videojuego" type="submit"

action="#{controllerManagedBean.pedirDatosVideojuego\_aLocalizar()}"

class="btn btn-info btn-block"/>

</c:if>

</h:form>

</div>

</div>

</div>

</body>

</html>

### Vista agregar.xhtml

Vista para agregar un videojuego.

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://xmlns.jcp.org/jsf/html"

xmlns:f="http://xmlns.jcp.org/jsf/core">

<head>

<title>Agregar videojuego</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous"/>

<link rel="stylesheet" href="#{request.contextPath}/resources/styles/bootstrap.min.css"/>

<style>

form {

padding: 2em;

margin-top: 2em;

border-radius: 5px;

}

</style>

</head>

<body>

<nav class="navbar navbar-dark bg-dark fixed-top">

<p class="navbar-brand">¡Hola, #{loginController.username}!</p>

<h:form style="margin-top: -1em" class="form-inline navbar-text my-2 my-lg-0">

<h:commandLink value="Videojuegos" action="faces/videojuegos.xhtml" class="nav-item nav-link"/>

</h:form>

<h:form class="form-inline navbar-text my-2 my-lg-0">

<h:commandLink action="#{loginController.logout}" value="Cerrar sesión"

class="my-2 my-sm-0"/>

</h:form>

</nav>

<div class="container">

<div class="row justify-content-center">

<div class="col-12 col-md-6">

<h:form id="form">

<h2 class="text-center" style="padding-top: .5em">

Agregar videojuego

</h2>

<div class="form-group">

<span>Clave</span>

<h:inputText value="#{controllerManagedBean.clave}" class="form-control" required="true">

<f:validateLength maximum="6"/>

<f:validateLongRange minimum="000001"/>

</h:inputText>

</div>

<div class="form-group">

<span>Nombre</span>

<h:inputText value="#{controllerManagedBean.nombre}" class="form-control" required="true"/>

</div>

<div class="form-group">

<span>Género</span>

<h:inputText value="#{controllerManagedBean.genero}" class="form-control"/>

</div>

<div class="form-group">

<span>Plataforma</span>

<h:inputText value="#{controllerManagedBean.plataforma}" class="form-control"/>

</div>

<div class="form-group">

<span>Precio</span>

<h:inputText value="#{controllerManagedBean.precio}" class="form-control">

<f:validateDoubleRange minimum="0.0"/>

</h:inputText>

</div>

<h:commandButton value="Guardar" type="submit"

action="#{controllerManagedBean.guardarVideojuego()}"

class="btn btn-primary btn-block"/>

</h:form>

<h:form>

<h:commandButton value="Regresar" action="faces/videojuegos.xhtml"

class="btn btn-danger btn-block"

style="margin-top: -5em"/>

</h:form>

</div>

</div>

</div>

</body>

</html>

### Vista localizar.xhtml

Vista para localizar un videojuego.

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://xmlns.jcp.org/jsf/html">

<head>

<title>Localizar videojuego</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous"/>

<link rel="stylesheet" href="#{request.contextPath}/resources/styles/bootstrap.min.css"/>

<style>

form {

padding: 2em;

margin-top: 2em;

border-radius: 5px;

}

</style>

</head>

<body>

<nav class="navbar navbar-dark bg-dark fixed-top">

<p class="navbar-brand">¡Hola, #{loginController.username}!</p>

<h:form style="margin-top: -1em" class="form-inline navbar-text my-2 my-lg-0">

<h:commandLink value="Videojuegos" action="faces/videojuegos.xhtml" class="nav-item nav-link"/>

</h:form>

<h:form class="form-inline navbar-text my-2 my-lg-0">

<h:commandLink action="#{loginController.logout}" value="Cerrar sesión"

class="my-2 my-sm-0"/>

</h:form>

</nav>

<div class="container">

<div class="row justify-content-center">

<div class="col-12 col-md-6">

<h:form id="form">

<h2 class="text-center" style="padding-top: .5em">Localizar videojuego</h2>

<div class="form-group">

<span>Clave</span>

<h:inputText value="#{controllerManagedBean.clave}" class="form-control" required="true"/>

</div>

<div class="form-group">

<span>Nombre</span>

<h:inputText value="#{controllerManagedBean.nombre}" class="form-control" required="true"/>

</div>

<h:commandButton value="Localizar" type="submit"

action="#{controllerManagedBean.localizarVideojuego()}"

class="btn btn-primary btn-block"/>

</h:form>

<h:form style="margin-top: -3em">

<h:commandButton value="Regresar" action="faces/videojuegos.xhtml" class="btn btn-danger btn-block"/>

</h:form>

</div>

</div>

</div>

</body>

</html>

### Vista modificar.xhtml

Vista para modificar un videojuego.

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://xmlns.jcp.org/jsf/html"

xmlns:f="http://xmlns.jcp.org/jsf/core">

<head>

<title>Modificar videojuego</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous"/>

<link rel="stylesheet" href="#{request.contextPath}/resources/styles/bootstrap.min.css"/>

<style>

form {

padding: 2em;

margin-top: 2em;

border-radius: 5px;

}

</style>

</head>

<body>

<nav class="navbar navbar-dark bg-dark fixed-top">

<p class="navbar-brand">¡Hola, #{loginController.username}!</p>

<h:form style="margin-top: -1em" class="form-inline navbar-text my-2 my-lg-0">

<h:commandLink value="Videojuegos" action="faces/videojuegos.xhtml" class="nav-item nav-link"/>

</h:form>

<h:form class="form-inline navbar-text my-2 my-lg-0">

<h:commandLink action="#{loginController.logout}" value="Cerrar sesión"

class="my-2 my-sm-0"/>

</h:form>

</nav>

<div class="container">

<div class="row justify-content-center">

<div class="col-12 col-md-6">

<h:form id="form">

<h2 class="text-center" style="padding-top: .5em">Modificar videojuego</h2>

<div class="form-group">

<span>Clave</span>

<h:inputText value="#{controllerManagedBean.clave}" class="form-control"

required="true" disabled="true"/>

</div>

<div class="form-group">

<span>Nombre</span>

<h:inputText value="#{controllerManagedBean.nombre}" class="form-control" required="true"/>

</div>

<div class="form-group">

<span>Género:</span>

<h:inputText value="#{controllerManagedBean.genero}" class="form-control"/>

</div>

<div class="form-group">

<span>Plataforma</span>

<h:inputText value="#{controllerManagedBean.plataforma}" class="form-control"/>

</div>

<div class="form-group">

<span>Precio</span>

<h:inputText value="#{controllerManagedBean.precio}" class="form-control">

<f:validateDoubleRange minimum="0.0"/>

</h:inputText>

</div>

<h:commandButton value="Guardar" type="submit"

action="#{controllerManagedBean.modificarVideojuego()}"

class="btn btn-primary btn-block"/>

</h:form>

<h:form style="margin-top: -3em">

<h:commandButton value="Regresar" action="faces/localizar.xhtml"

class="btn btn-danger btn-block"/>

</h:form>

</div>

</div>

</div>

</body>

</html>

### Vista borrar.xhtml

Vista para borrar un videojuego.

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://xmlns.jcp.org/jsf/html">

<head>

<title>Borrar videojuego</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous"/>

<link rel="stylesheet" href="#{request.contextPath}/resources/styles/bootstrap.min.css"/>

<style>

form {

padding: 2em;

margin-top: 2em;

border-radius: 5px;

}

</style>

</head>

<body>

<nav class="navbar navbar-dark bg-dark fixed-top">

<p class="navbar-brand">¡Hola, #{loginController.username}!</p>

<h:form style="margin-top: -1em" class="form-inline navbar-text my-2 my-lg-0">

<h:commandLink value="Videojuegos" action="faces/videojuegos.xhtml" class="nav-item nav-link"/>

</h:form>

<h:form class="form-inline navbar-text my-2 my-lg-0">

<h:commandLink action="#{loginController.logout}" value="Cerrar sesión"

class="my-2 my-sm-0"/>

</h:form>

</nav>

<div class="container">

<div class="row justify-content-center">

<div class="col-12 col-md-6">

<h:form id="form">

<h2 class="text-center" style="padding-top: .5em">

Borrar videojuego

</h2>

<div class="form-group">

<span>Clave</span>

<h:inputText value="#{controllerManagedBean.clave}" class="form-control" required="true"/>

</div>

<div class="form-group">

<span>Nombre</span>

<h:inputText value="#{controllerManagedBean.nombre}" class="form-control" required="true"/>

</div>

<span>

<h:commandButton value="Eliminar" type="submit"

action="#{controllerManagedBean.borrarVideojuego()}"

class="btn btn-danger btn-block"/>

</span>

</h:form>

<h:form>

<h:commandButton value="Regresar" action="faces/videojuegos.xhtml"

class="btn btn-primary btn-block"

style="margin-top: -5em"/>

</h:form>

</div>

</div>

</div>

</body>

</html>

### Vista error.xhtml

Vista de error en caso de querer localizar un videojuego que no existe dentro de los registros de la base de datos.

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://xmlns.jcp.org/jsf/html">

<head>

<title>Error</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous"/>

<link rel="stylesheet" href="#{request.contextPath}/resources/styles/bootstrap.min.css"/>

<style>

form {

padding: 2em;

margin-top: 2em;

border-radius: 5px;

}

</style>

</head>

<body>

<nav class="navbar navbar-dark bg-dark fixed-top">

<p class="navbar-brand">¡Hola, #{loginController.username}!</p>

<h:form style="margin-top: -1em" class="form-inline navbar-text my-2 my-lg-0">

<h:commandLink value="Videojuegos" action="faces/videojuegos.xhtml" class="nav-item nav-link"/>

</h:form>

<h:form class="form-inline navbar-text my-2 my-lg-0">

<h:commandLink action="#{loginController.logout}" value="Cerrar sesión"

class="my-2 my-sm-0"/>

</h:form>

</nav>

<div class="row justify-content-center">

<div class="col-12 col-md-6">

<h:form id="form">

<img src="#{request.contextPath}/resources/imgs/error.png"

width="80" height="80" class=""/>

<h2 class="text-center">ERROR, no se encontró un videjuego con:</h2>

<div class="form-group">

<span>Clave</span>

<h:outputText value="#{controllerManagedBean.clave}" class="form-control"/>

</div>

<div class="form-group">

<span>Nombre</span>

<h:outputText value="#{controllerManagedBean.nombre}" class="form-control"/>

</div>

</h:form>

<h:form class="form-group" style="margin-top: -4em">

<h:commandButton value="Regresar" action="faces/videojuegos.xhtml" class="btn btn-primary btn-block"/>

</h:form>

</div>

</div>

</body>

</html>

# CONCLUSIONES

El desarrollo de una aplicación web con el lenguaje Java en su versión Enterprise nos permite desarrollar aplicaciones web bastante complejas.

Java fue por muchos años considerado para el desarrollo de aplicaciones que necesitaban robustez y su seguridad en su diseño y estructura. El SDK nos provee diferentes utilidades que facilitan al desarrollador la codificación.

Si bien el desarrollo se puede ver beneficiado con la seguridad de la aplicación hay que ser conscientes de que puede haber una generación de código bastante grande que solo contribuye a que nuestra aplicación crezca y no tengamos una organización correcta.