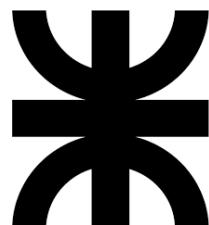


UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL ROSARIO



Lenguaje de Programación JAVA

Trabajo Práctico Integrador

Aplicación de reserva de entradas en el cine “Orfeo’s”

Integrantes:

| | |
|------------------------|-------|
| Buschittari, Nahuel | 47774 |
| Hernández, Juan Pablo | 47800 |
| Taborra, Facundo | 47999 |
| Vivas, Facundo Ignacio | 47844 |

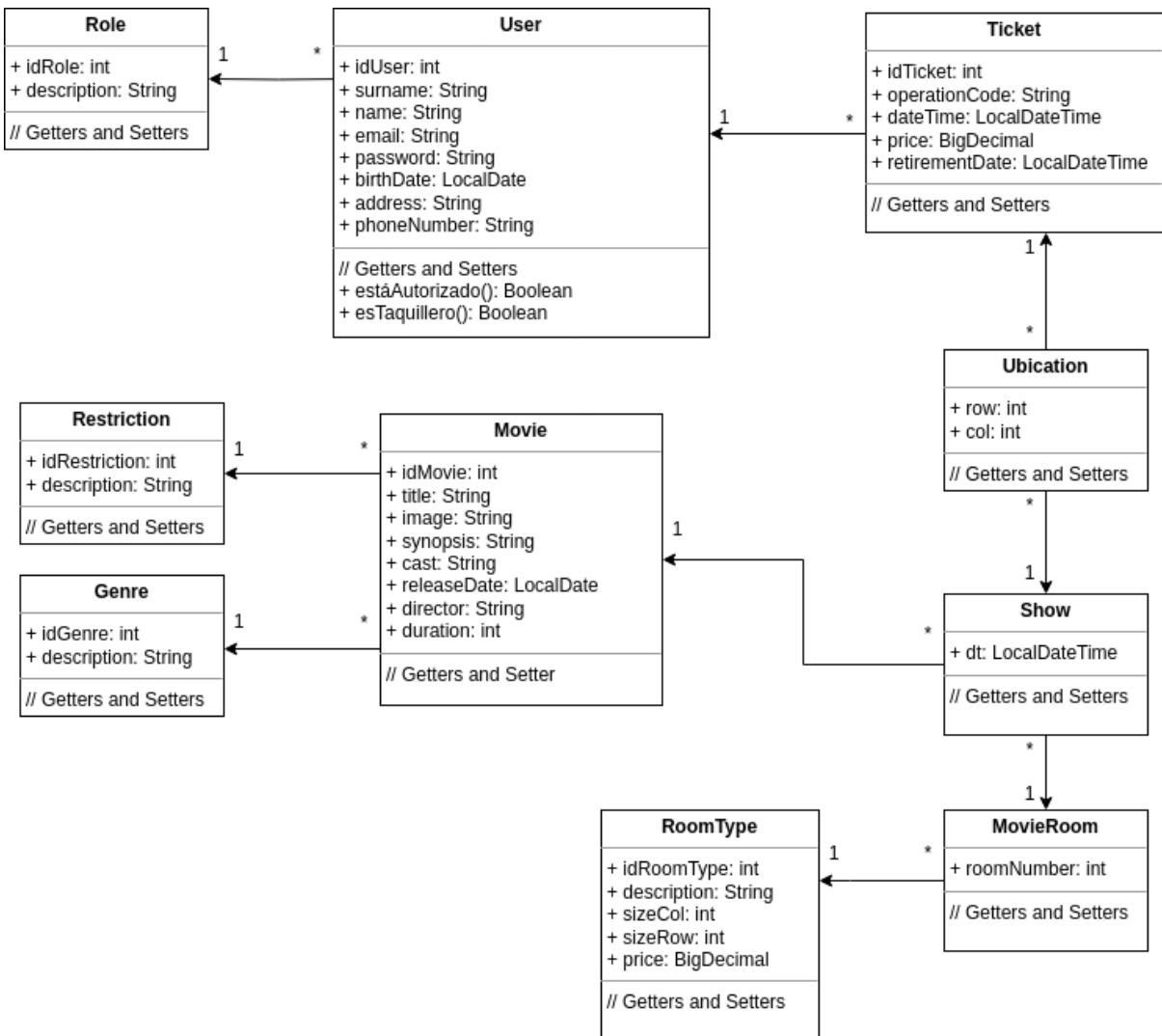
Docentes:

Tabacman, Ricardo David
Meca, Adrián

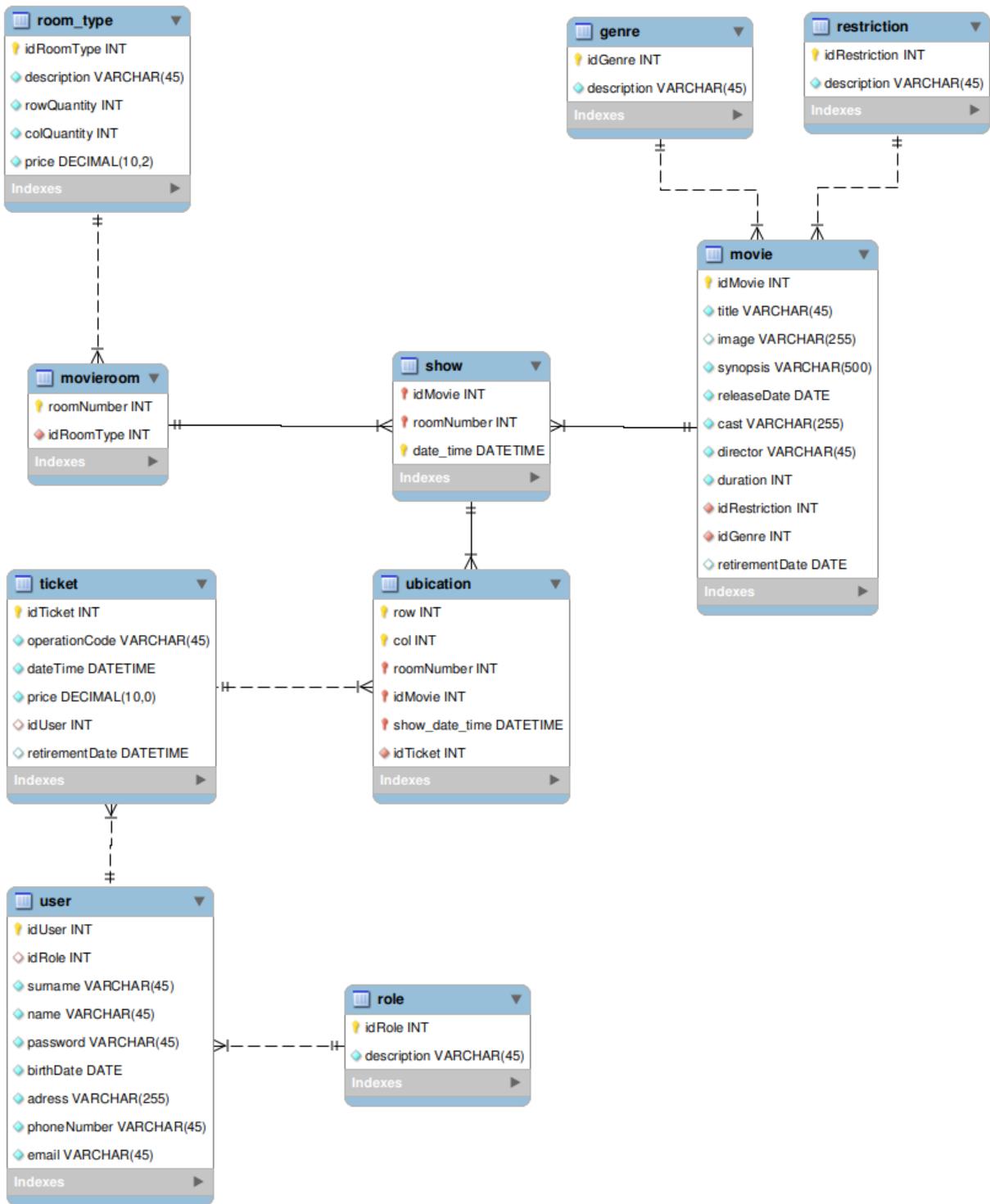
Índice

| | |
|---|-----------|
| Índice | 2 |
| Diagrama de Clases | 3 |
| Modelo de Datos | 4 |
| Casos de Uso | 5 |
| CUU01 - Obtener entradas | 5 |
| CUU02 - Crear funciones | 7 |
| CUU03 - Cancelar reserva | 8 |
| CUR01 - Reservar Entradas | 9 |
| Capturas de pantalla del sistema | 10 |
| Fragmentos del código | 15 |
| entradas.jsp | 15 |
| selecAsiento.jsp | 16 |
| entradasServlet.java | 17 |
| showController.java | 18 |
| dataShow.java | 19 |
| Repositorio en GitHub | 21 |

Diagrama de Clases



Modelo de Datos



Casos de Uso

CUU01 - Obtener entradas

Dimensiones:

| Alcance | Nivel | Estructura | Interacción | Instanciación | Caja |
|---------|---------|-----------------|-------------|---------------|-------|
| Sistema | Usuario | Sin estructurar | Semántica | Real | Negra |

Meta: Conseguir entradas para ver una película

Actores: Primario e iniciador: Cliente

Precondiciones: El cliente está logueado en su cuenta

Disparador: Cliente quiere reservar una entrada

Camino Básico

1. El cliente selecciona la opción de reservar entradas. El sistema muestra las películas disponibles en la cartelera.
2. El cliente selecciona la película que quiere ver. El sistema muestra los tipos de salas disponibles para esa película.
3. El cliente selecciona el tipo de sala y la cantidad de entradas que quiere reservar. El sistema muestra las fechas y horarios disponibles para las funciones disponibles.
4. El cliente selecciona la fecha y hora conveniente. El sistema muestra una grilla con todos los asientos disponibles.
5. El cliente selecciona el/los asientos donde se quiere ubicar. El sistema muestra un resumen de todos los datos de la reserva y calcula el total a pagar.
6. El cliente confirma la reserva. El sistema envía un correo electrónico con los datos de la reserva al mail asociado a su cuenta

Caminos alternativos

- 2.a <durante> No hay salas disponibles para esa función:
 - 2.a.1 El sistema muestra un mensaje. Fin CU.
- 3.a <durante> No hay funciones disponibles:
 - 3.a.1 El sistema muestra un mensaje. Fin CU.
- 4.a <durante> Todos los asientos están reservados
 - 4.a.1 Fin CU.
- 5.a <posterior> El cliente quiere cancelar la reserva

5.a.1 El sistema cancela la reserva y redirige a la página principal

Postcondiciones de sistema

Éxito: El sistema registró la reserva correctamente

CUU02 - Crear funciones

Dimensiones:

| Alcance | Nivel | Estructura | Interacción | Instanciación | Caja |
|---------|---------|-----------------|-------------|---------------|-------|
| Sistema | Usuario | Sin estructurar | Semántica | Real | Negra |

Meta: Dar de alta las funciones disponibles en boletería

Actores: Primario e iniciador: Taquillero

Precondiciones: El taquillero está logueado en su cuenta con accesos

Las películas ya están cargadas en el sistema

Disparador: Taquillero desea cargar las funciones

Camino Básico:

1. El taquillero ingresa la opción de administrar las funciones. El sistema muestra las funciones ya cargadas.
2. El taquillero ingresa a la opción de añadir una nueva función. El sistema muestra las películas disponibles.
3. El taquillero selecciona la película, le asigna una fecha y hora, y una sala determinada. El sistema registra la nueva función.

Postcondiciones de sistema

Éxito: El sistema registró las funciones correctamente.

CUU03 - Cancelar reserva

Dimensiones:

| Alcance | Nivel | Estructura | Interacción | Instanciación | Caja |
|---------|---------|-----------------|-------------|---------------|-------|
| Sistema | Usuario | Sin estructurar | Semántica | Real | Negra |

Meta: Eliminar una reserva.

Actores: Primario e iniciador: Cliente

Precondiciones: El cliente está logueado.

El cliente tiene reservas activas

Disparador: Cliente desea cancelar una reserva.

Camino Básico:

1. El cliente accede a la sección de su perfil. El sistema muestra todas las reservas activas y las caducadas.
2. El cliente selecciona la opción de eliminar una reserva

Postcondiciones de sistema

Éxito: El sistema registró la cancelación correctamente.

CUR01 - Reservar Entradas

Dimensiones:

| Alcance | Nivel | Estructura | Interacción | Instanciación | Caja |
|---------|---------|-----------------|-------------|---------------|-------|
| Sistema | Resumen | Re-estructurado | Semántica | Real | Negra |

Meta: Ver una película

Actores: Primario e iniciador: Cliente

Precondiciones: El cliente está logueado en su cuenta

Camino Básico:

1. Todos los jueves, el taquillero carga las funciones de la semana. <CUU2 - Crear funciones>
2. El cliente saca sus entradas a través del sistema <CUU1 - Obtener entradas>

Caminos Alternativos:

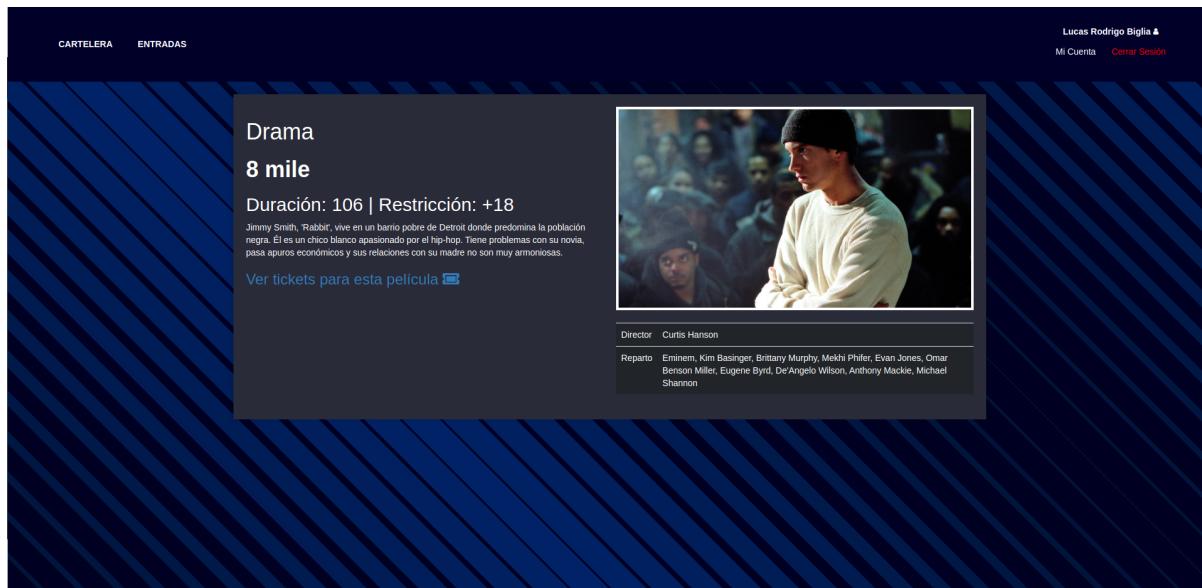
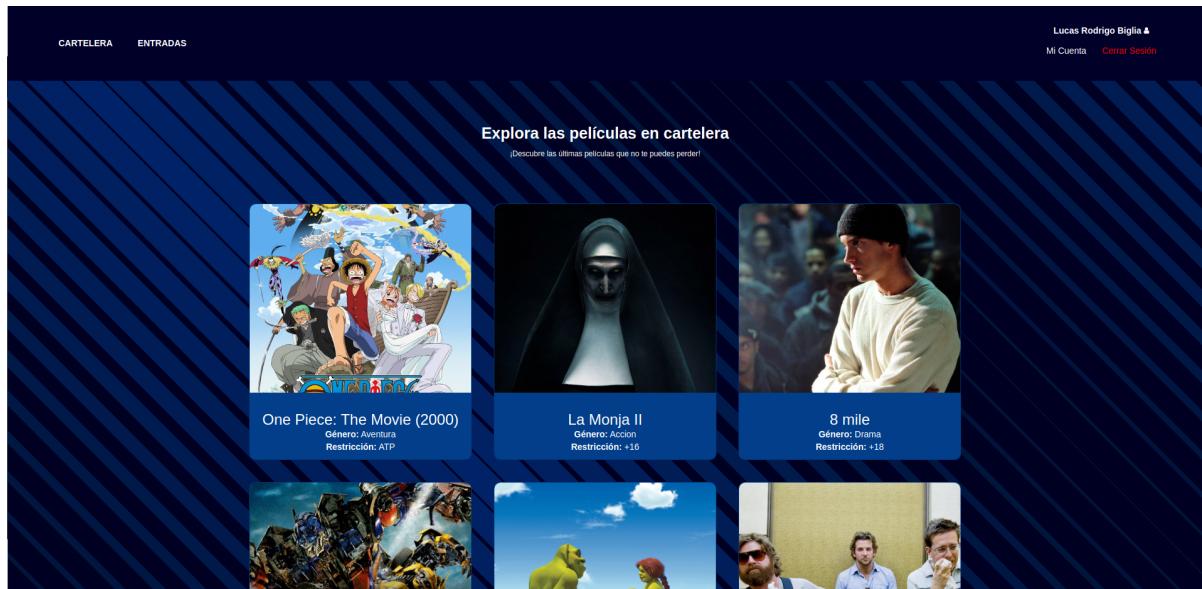
- 2.a <posterior> El cliente se arrepiente y quiere cancelar su ticket:
 - 2.a.1 <CUU3 - Cancelar reserva>

Postcondiciones de sistema

Éxito: El sistema registró la reserva correctamente

Éxito alternativo: El sistema registró la cancelación correctamente

Capturas de pantalla del sistema



CARTELERA ENTRADAS

Lucas Rodrigo Biglia  Mi Cuenta Cerrar Sesión

← BOLETERÍA VIRTUAL
Venta de Entradas

Seleccione una película 8 mile ▾

Aceptar

CARTELERA ENTRADAS

Lucas Rodrigo Biglia  Mi Cuenta Cerrar Sesión

← BOLETERÍA VIRTUAL
Venta de Entradas

Seleccione una película 8 mile ▾

Seleccione el tipo de sala IMAX Subtitulado \$1600.00 ▾

Seleccione cantidad de entradas 3

Aceptar

CARTELERA ENTRADAS

Lucas Rodrigo Biglia  Mi Cuenta Cerrar Sesión

← BOLETERÍA VIRTUAL
Venta de Entradas

Seleccione una película 8 mile ▾

Seleccione el tipo de sala 3D Subtitulado \$1200.00 ▾

Seleccione cantidad de entradas 1

Seleccione fecha y hora 26/11/2023 22:35 ▾

Aceptar

CARTELERA ENTRADAS

Lucas Rodrigo Biglia 
Mi Cuenta Cerrar Sesión

←

BOLETERÍA VIRTUAL

Venta de Entradas

Seleccione un Asiento

| PANTALLA | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|--|
| A-1  | A-2  | A-3  | A-4  | A-5  | A-6  | A-7  | A-8  | A-9  | A-10  |
| B-1  | B-2  | B-3  | B-4  | B-5  | B-6  | B-7  | B-8  | B-9  | B-10  |
| C-1  | C-2  | C-3  | C-4  | C-5  | C-6  | C-7  | C-8  | C-9  | C-10  |
| D-1  | D-2  | D-3  | D-4  | D-5  | D-6  | D-7  | D-8  | D-9  | D-10  |
| E-1  | E-2  | E-3  | E-4  | E-5  | E-6  | E-7  | E-8  | E-9  | E-10  |
| F-1  | F-2  | F-3  | F-4  | F-5  | F-6  | F-7  | F-8  | F-9  | F-10  |
| G-1  | G-2  | G-3  | G-4  | G-5  | G-6  | G-7  | G-8  | G-9  | G-10  |
| H-1  | H-2  | H-3  | H-4  | H-5  | H-6  | H-7  | H-8  | H-9  | H-10  |
| I-1  | I-2  | I-3  | I-4  | I-5  | I-6  | I-7  | I-8  | I-9  | I-10  |
| J-1  | J-2  | J-3  | J-4  | J-5  | J-6  | J-7  | J-8  | J-9  | J-10  |

CARTELERA ENTRADAS

Lucas Rodrigo Biglia 
Mi Cuenta Cerrar Sesión

←

BOLETERÍA VIRTUAL

Venta de Entradas

Resumen

Usted ha seleccionado:

| Pelicula | Sala | Numero Sala | Ubicacion | Fecha y Hora | Cantidad Entradas | Total |
|----------|----------------|-------------|-----------|------------------|-------------------|-----------|
| 8 mile | 3D Subtitulado | 1 | G-8 | 26/11/2023 22:35 | 1 | \$1200.00 |

¿Desea confirmar la reserva?

[Confirmar](#) [Cancelar](#)

CARTELERA ENTRADAS ADMINISTRAR FUNCIONES ADMINISTRAR PELICULAS PANEL ADMINISTRADOR

Facundo Vivas 
Mi Cuenta Cerrar Sesión

←

BOLETERÍA VIRTUAL

Venta de Entradas

Sus tickets fueron registrados con éxito!

Recibirá sus tickets por correo electrónico. Revise su casilla de Spam

Con el código de operación podrá retirarlos

Gracias por elegirnos!

Screenshot of an email inbox showing a ticket confirmation message from "cinesorfeo@outlook.com".

Tus Tickets Recibidos

cinesorfeo@outlook.com para mí

Tu ticket

Gracias por utilizar nuestro sistema de tickets, a continuación se muestra el ticket que ha generado.

Código de Operación: 123456 123456
Película: 8 mille
Fecha y Hora: 26/11/2023 22:35
Sala: 1
Asientos: 36 37
Monto Total: \$2400.00

Con estos códigos de operación podrá abonar y retirar sus tickets en boletería.

Disfrute su película
Orfeo Cines

Responder Reenviar

Screenshot of a user's account profile page titled "Mi Cuenta".

CARTELERA ENTRADAS Lucas Rodrigo Biglia ▾
Mi Cuenta Cerrar Sesión

Mi Cuenta

| | |
|------------------------------|--|
| Nombre: Lucas Rodrigo | Dirección: Mendoza 3657, Rosario, Santa Fe |
| Apellido: Biglia | Teléfono: 987654 |
| Email: Biglia5@outlook.com | Cambiar Contraseña |
| Fecha Nacimiento: 30/01/1986 | Confirmar nueva Contraseña |

[Editar mi Cuenta](#)

Mis Tickets Activos

| | | |
|--|--|---------------------------------|
|  8 milie | Fecha: 26/11/2023 22:35 Sala: 1 Fila:F Asiento: 6 Precio: 1200 | Cancelar Ticket |
|  | Fecha: 26/11/2023 22:35 Sala: 1 Fila:I Asiento: 6 Precio: 1200 | |

CARTELERA ENTRADAS ADMINISTRAR FUNCIONES ADMINISTRAR PELÍCULAS PANEL ADMINISTRADOR

Facundo Vivas Mi Cuenta Cerrar Sesión

Ingrese Título de la película: Relatos Salvajes

Restricción: +16 Genero: none Aceptar

Relatos Salvajes

Drama
Restricción: +16
Damian Szifron
Ricardo Darín, Grandinetti
Fecha de estreno: 2014-08-21 Duración: 123

The pianist

Drama
Restricción: +16
Roman Polanski
Adrien Brody, Thomas Kretschmann, Frank Finlay, Maureen Lipman, Emilia Fox, Ed Stoppard, Julia Rayner, Jessica Kate Meyer
Fecha de estreno: 2003-03-06 Duración: 151

Inglourious Basterds

Acción
Restricción: +12
Quentin Tarantino
Brad Pitt, Christoph Waltz, Meryl Streep, Eli Roth, Diane Kruger, Daniel Brühl, Til Schweiger, Michael Fassbender
Fecha de estreno: 2009-06-31 Duración: 153

CARTELERA ENTRADAS ADMINISTRAR FUNCIONES ADMINISTRAR PELÍCULAS PANEL ADMINISTRADOR

Facundo Vivas Mi Cuenta Cerrar Sesión

Fecha: dd/mm/aaaa Ingrese Título de la película: Aceptar

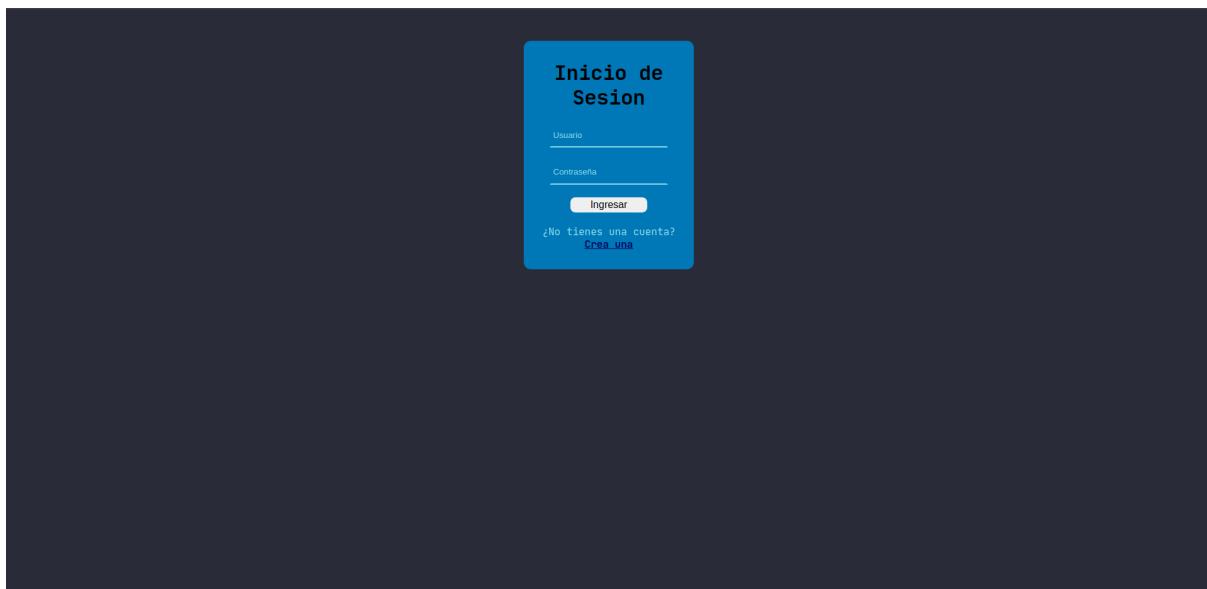
| Fecha y Hora | Película | Sala | + |
|------------------|-------------------------------------|------|---|
| 01/09/2022 13:30 | Transformers: Revenge of the Fallen | 2 | <input checked="" type="checkbox"/> <input type="checkbox"/> |
| 01/09/2022 16:30 | The pianist | 3 | <input checked="" type="checkbox"/> <input type="checkbox"/> |
| 02/09/2022 18:00 | Deadpool | 4 | <input checked="" type="checkbox"/> <input type="checkbox"/> |
| 02/09/2022 19:45 | The pianist | 3 | <input checked="" type="checkbox"/> <input type="checkbox"/> |
| 03/09/2022 22:00 | Inglourious Basterds | 1 | <input checked="" type="checkbox"/> <input type="checkbox"/> |
| 05/09/2022 21:30 | Thor: Love and Thunder | 2 | <input checked="" type="checkbox"/> <input type="checkbox"/> |
| 17/10/2022 15:47 | 8 mile | 3 | <input checked="" type="checkbox"/> <input type="checkbox"/> |

Bienvenido Facundo Vivas al PANEL ADMINISTRADOR

Desde aquí podrá gestionar los datos

Usuarios Roles Tickets Funciones Películas Restricciones Géneros Salas de Cine Tipos de Salas Ubicaciones

Volver a la app Cerrar Sesión

A screenshot of a mobile application's registration screen. The background is light blue. At the top left is a dark blue navigation bar with white text: "CARTELERA" and "ENTRADAS" on the left, and "Inicia Sesión!" with a user icon on the right. Below the navigation bar is a white rectangular form area with rounded corners. The title "Registrarse" is centered at the top in bold black font. The form contains six input fields arranged in two columns: "Nombre" and "Apellido" in the first row, "Email" and "Fecha Nacimiento" in the second row, and "Direccion" and "Telefono" in the third row. The "Fecha Nacimiento" field includes a date picker icon. At the bottom of the form is a grey "Crear mi Cuenta" button.

Fragmentos del código

entradas.jsp

```
<% if((String)request.getParameter("step") == null) { %>

<%
    request.getSession().removeAttribute("show");
    request.getSession().removeAttribute("showsDateTime");
    request.getSession().removeAttribute("tipoSala");
    request.getSession().removeAttribute("cantidad");
%>

<jsp:include page="/views/partials/selecPelicula.jsp"></jsp:include>

<%} else { %>

<% switch ((String)request.getParameter("step")) {

    case "selecSala": %>

        <jsp:include page="/views/partials/selecPelicula.jsp"></jsp:include>
        <jsp:include page="/views/partials/selecHabitacion.jsp"></jsp:include>

        <% break;

    case "selecHora": %>

        <jsp:include page="/views/partials/selecPelicula.jsp"></jsp:include>
        <jsp:include page="/views/partials/selecHabitacion.jsp"></jsp:include>
        <jsp:include page="/views/partials/selecDiaHora.jsp"></jsp:include>

        <% break;

    case "selecAsiento": %>

        <jsp:include page="/views/partials/selecAsiento.jsp"></jsp:include>

        <% break;

    case "confirmar": %>

        <jsp:include page="/views/partials/confirmarEntrada.jsp"></jsp:include>

        <% break;

    case "informarExito": %>

        <jsp:include page="/views/partials/informarExito.jsp"></jsp:include>

<%}%>
<%}>
```

selecAsiento.jsp

```
<%>
User user = (User)session.getAttribute("usuario");

if(user == null){
    response.sendRedirect("/TPI-Cine/views/pages/login.jsp");
}

Show s = (Show)session.getAttribute("show");

RoomType rt = (RoomType)session.getAttribute("tipoSala");
RoomTypeController ctrlRT = new RoomTypeController();

rt = ctrlRT.getOne(rt);

UbicationController ctrlUbi = new UbicationController();
LinkedList<Ubication> ubicaciones = ctrlUbi.getByShow(s);

Ubication[][] asientos = new Ubication[rt.getSizeRow()+1][rt.getSizeCol()+1];

for(int i=1; i <= rt.getSizeRow(); i++){
    for(int j=1; j <= rt.getSizeCol(); j++){
        Ubication u = new Ubication();
        u.setRow(i);
        u.setCol(j);
        asientos[i][j] = u;
    }
}

for(Ubication u : ubicaciones){
    asientos[u.getRow()][u.getCol()] = u;
}

request.setAttribute("asientos", asientos);

LinkedList<Ubication> ubicElegidas =
(LinkedList<Ubication>)request.getSession(false).getAttribute("ubicElegidas");
%>


### Seleccione un Asiento



| PANTALLA                                                     |                                                                                                                                                                                                                                         |
|--------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <%=Character.toString(fila+i)%>-<%=asientos[i][j].getCol()%> | <a href="/TPI-Cine/src/servlet/EntradasServlet?step=asiento&amp;fila=&lt;%=asientos[i][j].getRow()%&gt;&amp;col=&lt;%=asientos[i][j].getCol()%&gt;" style="color: blue" type="submit">&lt;i class="fa-solid fa-couch"&gt;&lt;/i&gt;</a> |


```

entradasServlet.java

```
● ● ●

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    Show s = new Show();
    Movie m = new Movie();
    RoomType rt = new RoomType();
    MovieController ctrlMovie = new MovieController();
    ShowController ctrlShow = new ShowController();
    RoomTypeController ctrlRT = new RoomTypeController();
    MovieRoomController ctrlMovieRoom = new MovieRoomController();

    switch ((String)request.getParameter("step")) {
        case "pelicula":
            try {
                if (request.getParameter("movie") != null) {
                    m.setIdMovie(Integer.parseInt((String)request.getParameter("movie")));
                    m = ctrlMovie.getByIdMovie(m);
                    s.setMovie(m);
                    request.getSession(false).setAttribute("show", s);
                    response.sendRedirect("/TPI-Cine/views/pages/entradas.jsp?step=selecSala");
                }
                break;
            } catch (Exception e) {
                request.setAttribute("error", e.getMessage());
                request.getRequestDispatcher("/views/pages/error.jsp").forward(request, response);
            }
        case "tipoDeSala":
            try {
                s = (Show) request.getSession(false).getAttribute("show");
                if(request.getParameter("tipo") != null && s != null){
                    rt.setIdRoomType(Integer.parseInt((String)request.getParameter("tipo")));
                    m = s.getMovie();
                    LinkedList<Show> shows = ctrlShow.getDateByRoomTypeMovie(m,rt);
                    rt = ctrlRT.getOne(rt);
                    request.getSession().setAttribute("cantidad",
                        Integer.parseInt(request.getParameter("cantidad")));
                    BigDecimal total = new
                        BigDecimal(Integer.parseInt(request.getParameter("cantidad")));
                    request.getSession().setAttribute("total", total);
                    request.getSession().setAttribute("tipoSala", rt);
                    request.getSession().setAttribute("showsDateTime", shows);
                    response.sendRedirect("/TPI-Cine/views/pages/entradas.jsp?step=selecHora");
                }
                break;
            } catch (Exception e) {
                request.setAttribute("error", e.getMessage());
                request.getRequestDispatcher("/views/pages/error.jsp").forward(request, response);
            }
        case "diaHora":
            try {
                String date_time = (String) request.getParameter("date_time");
                s = (Show)request.getSession(false).getAttribute("show");
                s.setDt(LocalDateTime.parse(date_time));
                Show s_traido = new Show();
                s_traido = ctrlShow.getRoomByMovieDateTime(s);
                s_traido.setMovieroom(ctrlMovieRoom.getOne(s_traido.getMovieroom()));
                s_traido.getMovieroom().setRt((RoomType)request.getSession().getAttribute("tipoSala"));
                s.setMovieroom(s_traido.getMovieroom());
                request.getSession(false).removeAttribute("show");
                request.getSession(false).setAttribute("show", s);
                LinkedList<Ubication> ubicElegidas = new LinkedList<Ubication>();
                request.getSession(false).setAttribute("ubicElegidas", ubicElegidas);
                response.sendRedirect("/TPI-Cine/views/pages/entradas.jsp?step=selecAsiento");
                break;
            } catch (Exception e) {
                request.setAttribute("error", e.getMessage());
                request.getRequestDispatcher("/views/pages/error.jsp").forward(request, response);
            }
        case "asiento":
            try {
                LinkedList<Ubication> ubicElegidas =
                    (LinkedList<Ubication>)request.getSession(false).getAttribute("ubicElegidas");
                request.getSession(false).setAttribute
                ("cantidad", (Integer)request.getSession(false).getAttribute("cantidad") - 1);
                if((Integer)request.getSession(false).getAttribute("cantidad") >= 0) {
                    Ubication u_elegida = new Ubication();
                    int fila = Integer.parseInt( (String)request.getParameter("fila"));
                    int colu = Integer.parseInt( (String)request.getParameter("col"));
                    u_elegida.setRow(fila);
                    u_elegida.setCol(colu);
                    ubicElegidas.add(u_elegida);
                    request.getSession(false).setAttribute("ubicElegidas", ubicElegidas);
                    if ((Integer)request.getSession(false).getAttribute("cantidad") != 0) {
                        response.sendRedirect("/TPI-Cine/views/pages/entradas.jsp?step=selecAsiento");
                    } else {
                        request.getSession(false).setAttribute("ubicElegidas", ubicElegidas);
                        response.sendRedirect("/TPI-Cine/views/pages/entradas.jsp?step=confirmar");
                    }
                }
                break;
            } catch (Exception e) {
                request.setAttribute("error", e.getMessage());
                request.getRequestDispatcher("/views/pages/error.jsp").forward(request, response);
            }
        default:
            break;
    }
}
```

showController.java

```
● ● ●

package logic;

import java.sql.SQLException;
import java.util.LinkedList;
import data.DataShow;
import entities.Movie;
import entities.RoomType;
import entities.Show;

public class ShowController {
    private DataShow ds;

    public ShowController(){
        ds= new DataShow();
    }
    public Show getOne(Show s) throws SQLException {
        return ds.getOne(s);
    }
    public void deleteOne(Show s) throws SQLException {
        ds.delete(s);
    }
    public Show editOne(Show s, Show sw) throws SQLException {
        return ds.update(s,sw);
    }
    public Show addOne(Show s) throws SQLException {
        return ds.createOne(s);
    }
    public LinkedList<Show> getAll() throws SQLException{
        return ds.getList();
    }
    public LinkedList<Show> getDateByRoomTypeMovie(Movie m, RoomType rt) throws SQLException {
        return ds.getDate(m,rt);
    }
    public Show getRoomByMovieDateTime(Show s) throws SQLException {
        return ds.getRoomByMovieDateTime(s);
    }
    public LinkedList<Show> getByTit(Movie m) throws SQLException{
        return ds.getShowsbyMovie(m);
    }
    public LinkedList<Show> getByTityFech(Show s, Movie m) throws SQLException{
        return ds.getShowsbyFechaMovie(s,m);
    }
    public LinkedList<Show> getByFech(Show s) throws SQLException{
        return ds.getShowsbyFecha(s);
    }
}
```

dataShow.java

```
public LinkedList<Show> getDateTime(Movie m, RoomType rt) throws SQLException {
    PreparedStatement stmt = null;
    ResultSet rs = null;
    LinkedList<Show> fechas = new LinkedList<Show>();
    try {
        stmt=DbConnector.getInstancia().getConn().prepareStatement(
            "select date_time from\n" +
            "cine_tpjava.show s\n" +
            "inner join movieRoom mr\n" +
            "on mr.roomNumber=s.roomNumber\n" +
            "where s.idMovie=?\n" +
            "and mr.idRoomType=?\n" +
            "and s.date_time >= CURRENT_DATE;" );
        stmt.setInt(1, m.getIdMovie());
        stmt.setInt(2, rt.getIdRoomType());
        rs=stmt.executeQuery();
        if (rs != null) {
            while (rs.next()) {
                Show s = new Show();
                s.setDt(rs.getObject("date_time",LocalDateTime.class));
                fechas.add(s);
            }
        }
    } catch (SQLException e) {
        throw e;
    } finally {
        try {
            if(rs!=null) {rs.close();}
            if(stmt!=null) {stmt.close();}
            DbConnector.getInstancia().releaseConn();
        } catch (SQLException e) {
            throw e;
        }
    }
    return fechas;
}
public Show getRoomByMovieDateTime(Show s) throws SQLException {
    PreparedStatement stmt=null;
    ResultSet rs=null;
    MovieRoom mr = new MovieRoom();
    try {
        stmt=DbConnector.getInstancia().getConn().prepareStatement
            ("select roomNumber\n" +
            "from cine_tpjava.show s\n" +
            "where s.date_time=?\n" +
            "and s.idMovie=?");
        stmt.setString(1, s.getDt().toString());
        stmt.setInt(2, s.getMovie().getIdMovie());
        rs=stmt.executeQuery();
        if(rs!=null && rs.next()) {
            mr.setRoomNumber(rs.getInt("roomNumber"));
            s.setMovieroom(mr);
        }
    }catch (SQLException e) {
        throw e;
    } finally {
        try {
            if(rs!=null) {rs.close();}
            if(stmt!=null) {stmt.close();}
            DbConnector.getInstancia().releaseConn();
        } catch (SQLException e) {
            throw e;
        }
    }
    return s;
}
```

```
● ● ●
```

```
public LinkedList<Show> getShowsbyMovie(Movie m) throws SQLException {
    PreparedStatement stmt = null;
    ResultSet rs = null;
    LinkedList<Show> shows = new LinkedList<Show>();
    try {
        stmt=DbConnector.getInstancia().getConn().prepareStatement(
            "select s.idMovie, s.roomNumber, s.date_time \n"
            + "from cine_tpjava.show s\n"
            + "inner join movie m\n"
            + "on m.idMovie=s.idMovie\n"
            + "where m.title like ?"
            + "order by s.date_time;"
        );
        stmt.setString(1, "%" + m.getTitle() + "%");
        rs=stmt.executeQuery();
        if (rs != null) {
            while (rs.next()) {
                MovieRoom mr=new MovieRoom();
                Movie mo=new Movie();
                Show sw = new Show();
                sw.setDt(rs.getObject("date_time",LocalDateTime.class));
                mr.setRoomNumber(rs.getInt("roomNumber"));
                sw.setMovieroom(mr);
                mo.setIdMovie(rs.getInt("idMovie"));
                sw.setMovie(mo);
                shows.add(sw);
            }
        }
    } catch (SQLException e) {
        throw e;
    } finally {
        try {
            if(rs!=null) {rs.close();}
            if(stmt!=null) {stmt.close();}
            DbConnector.getInstancia().releaseConn();
        } catch (SQLException e) {
            throw e;
        }
    }
    return shows;
}
public LinkedList<Show> getShowsbyFechaMovie(Show s, Movie m) throws SQLException {
    PreparedStatement stmt = null;
    ResultSet rs = null;
    LinkedList<Show> shows = new LinkedList<Show>();
    try {
        stmt=DbConnector.getInstancia().getConn().prepareStatement(
            "select s.idMovie, s.roomNumber, s.date_time \n"
            + "from cine_tpjava.show s\n"
            + "inner join movie m\n"
            + "on m.idMovie=s.idMovie \n"
            + "where m.title like ? and CAST(s.date_time as DATE)=?"
            + "order by s.idMovie, s.date_time;"
        );
        stmt.setString(1, "%" + m.getTitle() + "%");
        stmt.setObject(2, s.getDt());
        rs=stmt.executeQuery();
        if (rs != null) {
            while (rs.next()) {
                MovieRoom mr=new MovieRoom();
                Movie mo=new Movie();
                Show sw = new Show();
                sw.setDt(rs.getObject("date_time",LocalDateTime.class));
                mr.setRoomNumber(rs.getInt("roomNumber"));
                sw.setMovieroom(mr);
                mo.setIdMovie(rs.getInt("idMovie"));
                sw.setMovie(mo);
                shows.add(sw);
            }
        }
    } catch (SQLException e) {
        throw e;
    } finally {
        try {
            if(rs!=null) {rs.close();}
            if(stmt!=null) {stmt.close();}
            DbConnector.getInstancia().releaseConn();
        } catch (SQLException e) {
            throw e;
        }
    }
    return shows;
}
```

Repositorio en GitHub



<https://github.com/flechaVivas/TPI-Cine>