

UNIVERSIDAD TECNOLÓGICA NACIONAL

Facultad Regional Rosario



Informe Trabajo Práctico

Gestión de un boliche

Cátedra: Lenguaje de Programación JAVA

Comisión: 303 - ISI

Profesores:

- Adrián Meca
- Ricardo Tabacman

Integrantes:

- Glocer, Martin - 48165
- Spini, Santiago - 49799

Fecha de presentación: 14/02/2025

Índice

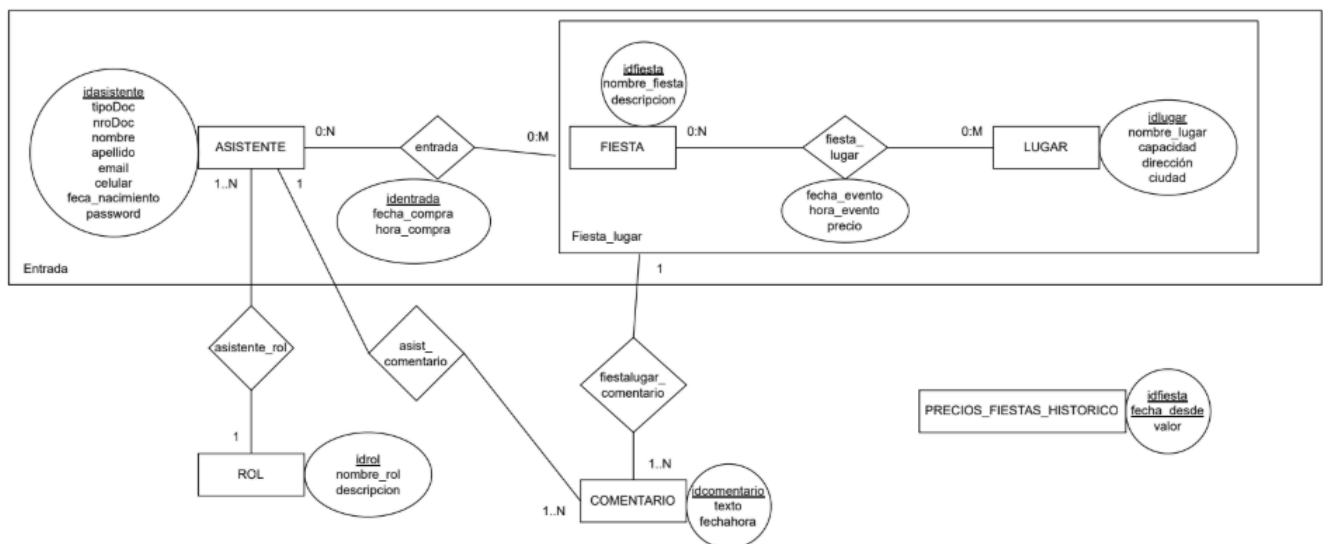
Índice	2
Narrativa	3
Modelo de Datos (D-E-R)	3
Modelo de Datos	4
Caso de uso resumen	4
Casos de uso de usuario	5
CUU01 - Vender entrada	5
CUU02 - Realizar comentario	9
Imágenes (capturas de pantalla del sistema)	12
Interfaz del Usuario	12
Interfaz del Administrador	13

Narrativa

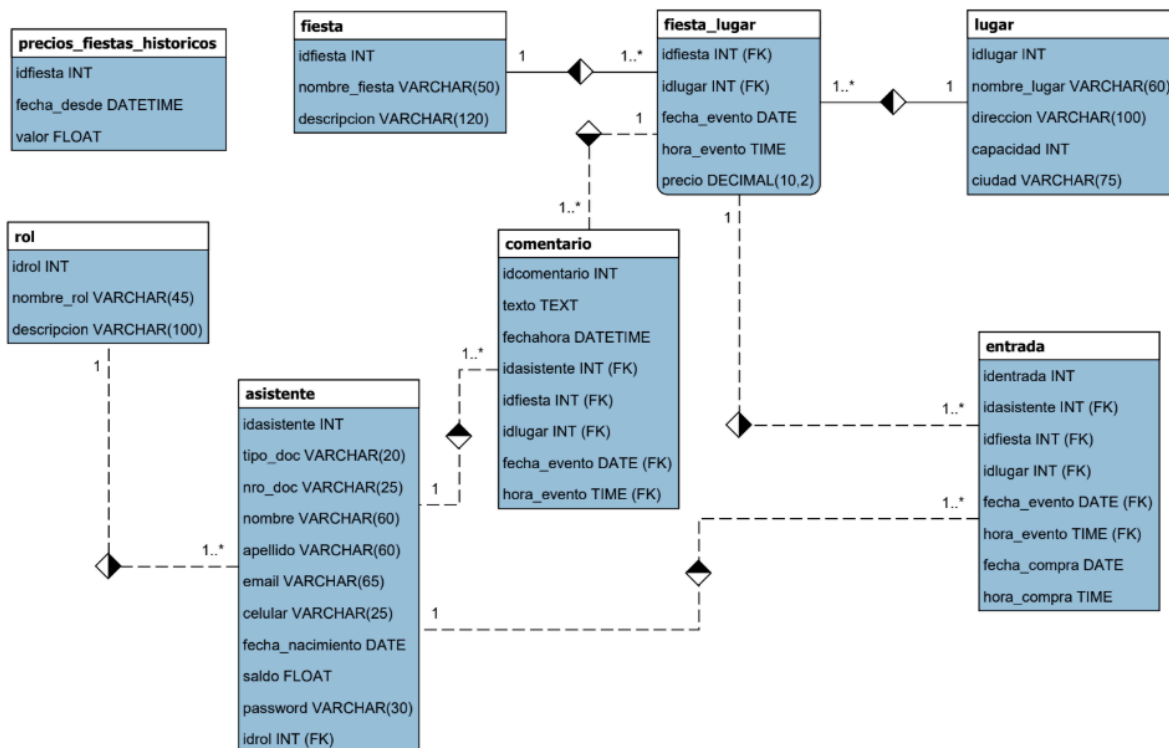
Esta aplicación Java facilita la gestión integral de un boliche, permitiendo a los usuarios comprar entradas para eventos de manera sencilla a través de una integración con Spark. Los usuarios también pueden interactuar con la comunidad a través de comentarios y acceder a información actualizada sobre los eventos. La plataforma está diseñada para optimizar la experiencia del boliche y sus asistentes, mejorando la gestión de eventos y brindando una forma ágil de adquirir entradas online.

El proyecto tiene como objetivo mejorar la experiencia de los usuarios, agilizar la compra de entradas, y fomentar la interacción entre el boliche y su comunidad a través de funcionalidades como la compra de entradas, la gestión de eventos y la interacción mediante comentarios.

Modelo de Datos (D-E-R)



Modelo de Datos



Caso de uso resumen

CU - Caso de Uso - Boliche

Nivel de la meta: **Resumen** - Alcance del Caso de Uso: **Sistema** - Caja: **Negra**

Instanciación: **Real** - Interacción: **Semántica** - Usabilidad: **No Contemplada**

ACTORES Primario: **Usuario** - Iniciador: **Usuario**

PRECONDICIONES: -

DISPARADOR: El usuario desea comprar una entrada para un evento futuro.

POSTCONDICIONES: Éxito: La venta de la entrada fue registrada.

Éxito alternativo: El usuario y la venta de la entrada fueron registrados.

Fracaso: No se registró una nueva entrada ni un nuevo usuario.

Paso	Acción
1	El usuario se loguea en el sistema invocando a CUU01 - Iniciar sesión
2	El usuario selecciona uno de los eventos futuros y adquiere la entrada invocando a CUU02 - Vender entrada
3	El usuario hace comentarios sobre un evento invocando a CUU03 - Realizar comentario

Casos de uso de usuario

CUU01 - Vender entrada

Caso de Uso 01 - Vender entrada

Nivel de la meta: **Usuario** - Alcance del Caso de Uso: **Sistema** - Caja: **Negra**

Instanciación: **Real** - Interacción: **Dialogal** - Usabilidad: **No Contemplada**

ACTORES Primario: **Usuario** - Iniciador: **Usuario**

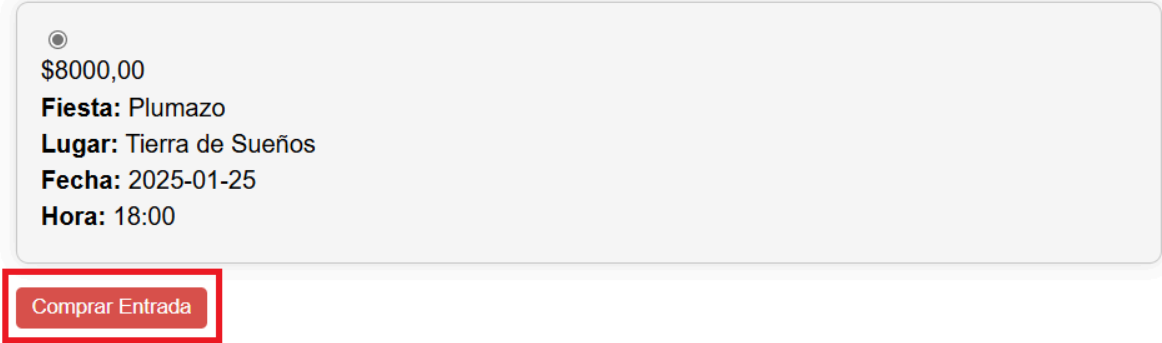
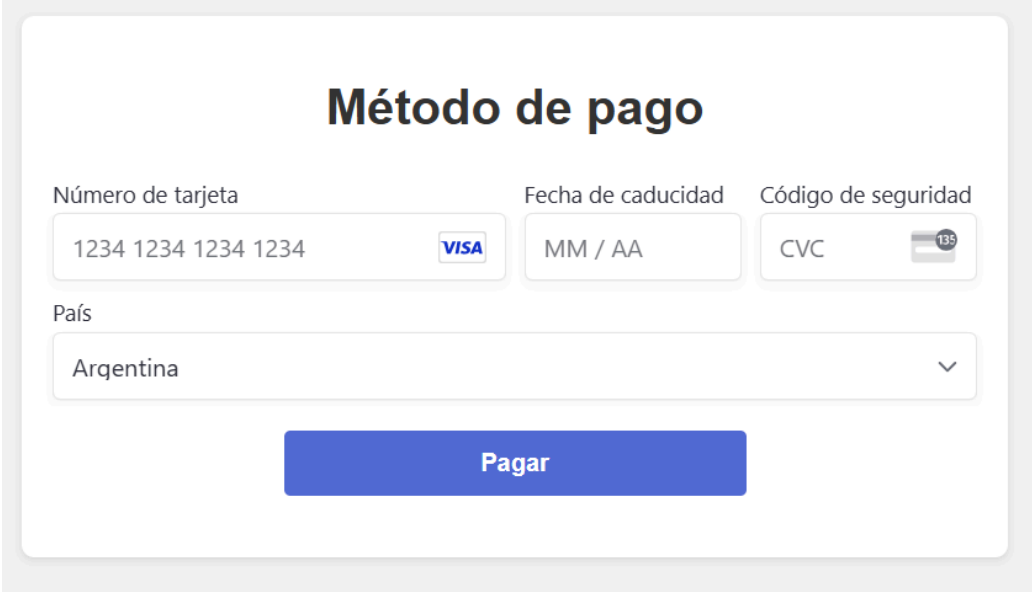
PRECONDICIONES: El usuario se encuentra logueado.





DISPARADOR: El usuario desea comprar una entrada para un evento futuro.

POSTCONDICIONES: Éxito: La venta de la entrada fue registrada.

Éxito alternativo: -

Fracaso: No se registró una nueva entrada.

Paso	Usuario / Condicion	Sistema
1		Muestra los futuros eventos disponibles.
	Sist	
2	El usuario selecciona el evento al que quiere asistir. Presiona el botón "Comprar Entrada".	Muestra nueva página para ingresar la información del método de pago.
	Usu	
	Sist	
2.a	El usuario no seleccionó un evento. Presiona el botón "Comprar Entrada"	Muestra mensaje informativo. Vuelve al paso 2.

	Sist	<p>Por favor seleccione un evento</p>	
3		El usuario ingresa el número de tarjeta, la fecha de caducidad, el código de seguridad y el país.	Valida los datos ingresados. Registra la nueva entrada comprada. Informa la compra exitosa al usuario.
	Usu	<div> <h3>Método de pago</h3> <div> <div>Número de tarjeta</div> <div>4242 4242 4242 4242 </div> </div> <div> <div>Fecha de caducidad</div> <div>06 / 26</div> </div> <div> <div>Código de seguridad</div> <div>397 </div> </div> <div> <div>País</div> <div>Argentina </div> </div> <div>Pagar</div> </div>	
	Sist	<div>  <h3>¡Pago exitoso!</h3> <p>Tu pago se ha procesado correctamente</p> <div>OK</div> </div>	
3.a		El usuario no ingresó el número de la tarjeta.	Informa la situación. Vuelve al paso 3.

	Sist	<div> <h3>Método de pago</h3> <div> <div>Número de tarjeta</div> <div> 1234 1234 1234 1234  </div> </div> <div> <div>Fecha de caducidad</div> <div>06 / 26</div> </div> <div> <div>Código de seguridad</div> <div>397 </div> </div> <div>El número de tu tarjeta está incompleto.</div> <div>País</div> <div>Argentina </div> <div>Pagar</div> <div>El número de tu tarjeta está incompleto.</div> </div>	
3.b	El usuario no ingresó la fecha de caducidad.	Informa la situación.	Vuelve al paso 3.
	Sist	<div> <h3>Método de pago</h3> <div> <div>Número de tarjeta</div> <div> 4242 4242 4242 4242  </div> </div> <div> <div>Fecha de caducidad</div> <div>MM / AA</div> </div> <div> <div>Código de seguridad</div> <div>397 </div> </div> <div>La fecha de caducidad de tu tarjeta está incompleta.</div> <div>País</div> <div>Argentina </div> <div>Pagar</div> <div>La fecha de caducidad de tu tarjeta está incompleta.</div> </div>	
3.c	El usuario no ingresó el código de seguridad.	Informa la situación.	Vuelve al paso 3.

	Sist	<div><h3>Método de pago</h3><div><div>Número de tarjeta</div><div>4242 4242 4242 4242 </div></div><div><div>Fecha de caducidad</div><div>06 / 26</div></div><div><div>Código de seguridad</div><div>CVC </div><div>El código de seguridad de tu tarjeta está incompleto.</div></div><div><div>País</div><div>Argentina </div></div><div><div>Pagar</div></div><div>El código de seguridad de tu tarjeta está incompleto.</div></div>
--	------	---

CUU02 - Realizar comentario

Caso de Uso 02 - Realizar comentario

Nivel de la meta: **Usuario** - Alcance del Caso de Uso: **Sistema** - Caja: **Negra**

Instanciación: **Real** - Interacción: **Dialogal** - Usabilidad: **No Contemplada**

ACTORES Primario: **Usuario** - Iniciador: **Usuario**

PRECONDICIONES: El usuario se encuentra logueado.

DISPARADOR: El usuario desea realizar un comentario sobre un evento futuro.

POSTCONDICIONES: Éxito: El comentario fue registrado.

Éxito alternativo: -

Fracaso: No se registró ningún comentario nuevo.

Paso	Usuario / Condicion	Sistema
1		Muestra los futuros eventos disponibles.

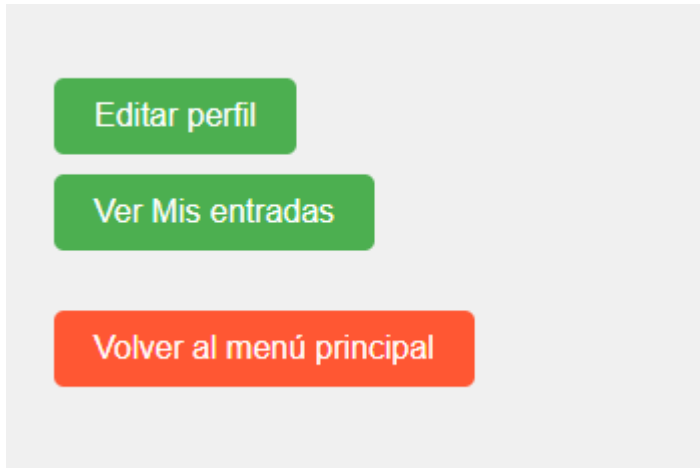
	Sist	<table><tr><th>Fecha</th><th>Hora</th><th>Fiesta</th><th>Lugar</th><th>Direccion</th><th>Ciudad</th><th>Precio</th><th>Comprar Entrada</th><th>Comentarios</th></tr><tr><td>2025-01-25</td><td>18:00</td><td>Plumazo</td><td>Tierra de Sueños</td><td>Funes 5437</td><td>Rosario</td><td>8000.0</td><td>Comprar Entrada</td><td>Ver comentarios</td></tr><tr><td>2025-02-12</td><td>23:00</td><td>Lova</td><td>Isla</td><td>Isla Rio 2431</td><td>Rosario</td><td>10000.0</td><td>Comprar Entrada</td><td>Ver comentarios</td></tr></table>		Fecha	Hora	Fiesta	Lugar	Direccion	Ciudad	Precio	Comprar Entrada	Comentarios	2025-01-25	18:00	Plumazo	Tierra de Sueños	Funes 5437	Rosario	8000.0	Comprar Entrada	Ver comentarios	2025-02-12	23:00	Lova	Isla	Isla Rio 2431	Rosario	10000.0	Comprar Entrada	Ver comentarios
Fecha	Hora	Fiesta	Lugar	Direccion	Ciudad	Precio	Comprar Entrada	Comentarios																						
2025-01-25	18:00	Plumazo	Tierra de Sueños	Funes 5437	Rosario	8000.0	Comprar Entrada	Ver comentarios																						
2025-02-12	23:00	Lova	Isla	Isla Rio 2431	Rosario	10000.0	Comprar Entrada	Ver comentarios																						
2		El usuario selecciona el link “Ver comentarios” del evento sobre el que quiere hacer comentarios.	Muestra nueva página para ver y realizar comentarios.																											
	Usu	<table><tr><th>Fecha</th><th>Hora</th><th>Fiesta</th><th>Lugar</th><th>Direccion</th><th>Ciudad</th><th>Precio</th><th>Comprar Entrada</th><th>Comentarios</th></tr><tr><td>2025-01-25</td><td>18:00</td><td>Plumazo</td><td>Tierra de Sueños</td><td>Funes 5437</td><td>Rosario</td><td>8000.0</td><td>Comprar Entrada</td><td>Ver comentarios</td></tr><tr><td>2025-02-12</td><td>23:00</td><td>Lova</td><td>Isla</td><td>Isla Rio 2431</td><td>Rosario</td><td>10000.0</td><td>Comprar Entrada</td><td>Ver comentarios</td></tr></table>		Fecha	Hora	Fiesta	Lugar	Direccion	Ciudad	Precio	Comprar Entrada	Comentarios	2025-01-25	18:00	Plumazo	Tierra de Sueños	Funes 5437	Rosario	8000.0	Comprar Entrada	Ver comentarios	2025-02-12	23:00	Lova	Isla	Isla Rio 2431	Rosario	10000.0	Comprar Entrada	Ver comentarios
Fecha	Hora	Fiesta	Lugar	Direccion	Ciudad	Precio	Comprar Entrada	Comentarios																						
2025-01-25	18:00	Plumazo	Tierra de Sueños	Funes 5437	Rosario	8000.0	Comprar Entrada	Ver comentarios																						
2025-02-12	23:00	Lova	Isla	Isla Rio 2431	Rosario	10000.0	Comprar Entrada	Ver comentarios																						
	Sist	<div><h3>Comentarios del evento: Plumazo</h3><p>Fecha y hora: 2025-01-25 a las 18:00 Lugar: Tierra de Sueños</p><p>Ale (2024-11-20T12:53:20): Con muchas ganas de asistir a este nuevo plumazo! Tengo la sensación de que va a estar muy bueno.</p><p>Julia (2024-11-20T13:34:30): Para mí no debería ser en Tierra de Sueños.</p><h4>Agregar un nuevo comentario</h4><div><div></div><div>Enviar comentario</div></div></div>																												
3		El usuario ingresa el contenido del comentario. Presiona el botón “Enviar comentario”.	Valida que se ingresen los datos. Registra el nuevo comentario. Muestra el comentario junto con el resto de comentarios de otros usuarios.																											
	Usu	<div><h4>Agregar un nuevo comentario</h4><div><div>Ya no falta nada! No veo la hora de que sea el día.</div><div>Enviar comentario</div></div></div>																												

	Sist	<div><h3>Comentarios del evento: Plumazo</h3><p>Fecha y hora: 2025-01-25 a las 18:00 Lugar: Tierra de Sueños</p><p>Ale (2024-11-20T12:53:20): Con muchas ganas de asistir a este nuevo plumazo! Tengo la sensación de que va a estar muy bueno.</p><p>Julia (2024-11-20T13:34:30): Para mí no debería ser en Tierra de Sueños.</p><p>Ale (2024-12-11T12:47:40): Ya no falta nada! No veo la hora de que sea el día.</p><h4>Agregar un nuevo comentario</h4><div><div></div><div>Enviar comentario</div></div></div>	
	3.a	El usuario no ingresó el contenido del comentario. Presiona el botón "Enviar comentario".	Informa la situación. Vuelve al paso 3.
	Sist	<div><h4>Agregar un nuevo comentario</h4><div><div></div><div>Enviar comentario</div><div><div>!</div><div>Rellena este campo.</div></div></div></div>	

Imágenes (capturas de pantalla del sistema)

Interfaz del Usuario

Pestaña de perfil



Pestaña editar perfil

Datos del perfil

Tipo de documento

dni

Número de documento

12333

Nombre

Ale

Apellido

Glozer

Email

aleglozer@gmail.com

Fecha de nacimiento

19/06/1998

Celular

34119

Saldo

5000.0

Rol

2

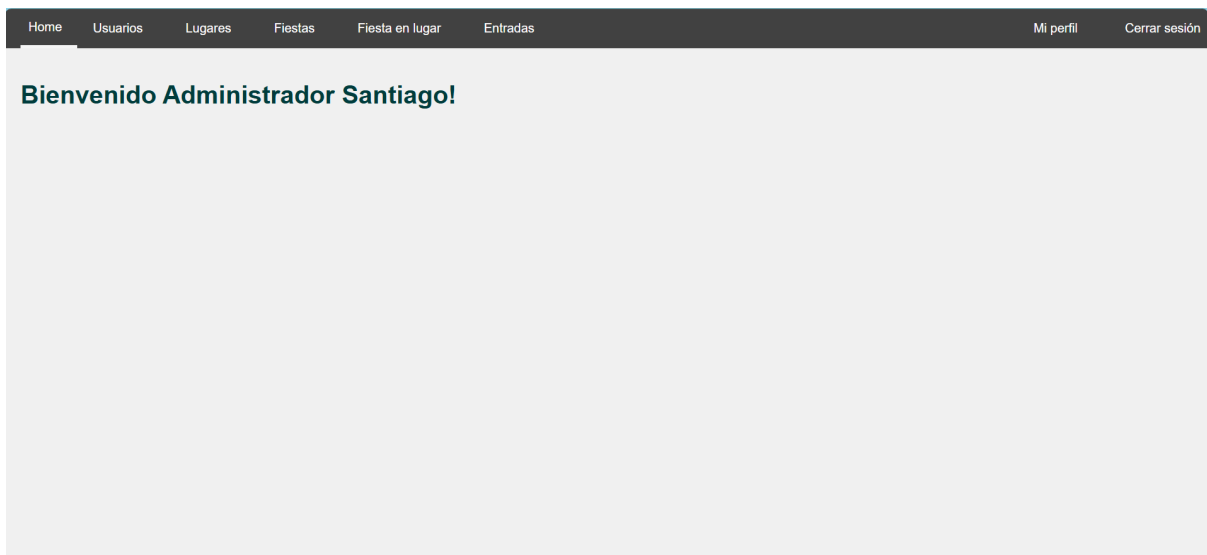
Contraseña

3210

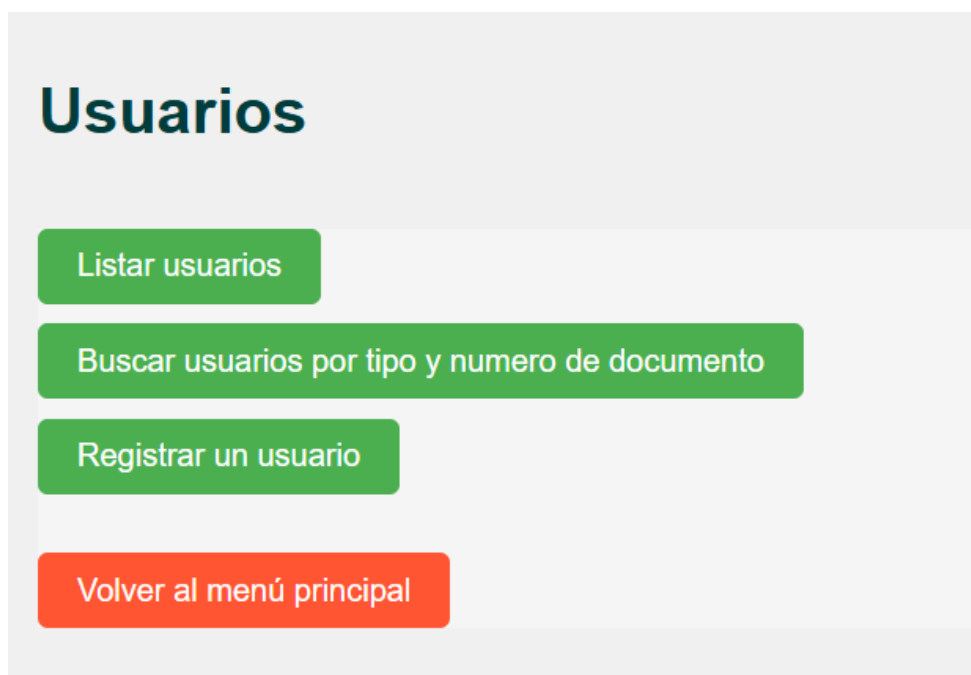
Guardar cambios

Interfaz del Administrador

Página inicial de bienvenida



Formato de menú para cada pestaña (es similar para Lugares, Fiestas, Fiesta en lugar, Entradas)



Lista de fiestas en lugares registradas

Lista de fiestas registradas

Buscar por nombre de fiesta:

Fecha	Hora	Fiesta	Lugar	Direccion	Ciudad	Precio	modificar	eliminar
2023-11-11	13:00	Crean	Isla	Isla Rio 2431	Rosario	12500.0	Editar	<input type="button" value="Borrar"/>
2023-11-25	15:00	Coco	Tierra de Sueños	Funes 5437	Rosario	0.0	Editar	<input type="button" value="Borrar"/>
2023-11-25	23:00	Lova	Metropolitano	Junín 501	Rosario	0.0	Editar	<input type="button" value="Borrar"/>
2023-12-09	13:00	Crean	Club CAE	3 de febrero 608	Santa Fe	0.0	Editar	<input type="button" value="Borrar"/>
2023-12-16	14:00	Plumazo	Club CAE	3 de febrero 608	Santa Fe	0.0	Editar	<input type="button" value="Borrar"/>
2023-12-20	18:00	Coco	Metropolitano	Junín 501	Rosario	0.0	Editar	<input type="button" value="Borrar"/>
2023-12-27	20:30	Plumazo	Isla	Isla Rio 2431	Rosario	0.0	Editar	<input type="button" value="Borrar"/>

Pestaña para editar una fiesta en un lugar

Datos de fiesta_lugar


Id Fiesta

1


Id Lugar

1

Fecha actual del evento

Elegir fecha nueva del evento 

Hora actual del evento

Elegir hora nueva del evento 

Precio actual del evento

12500,0

Elegir nuevo precio del evento

Lista de entradas registradas

Lista de todas las entradas

Buscar por nombre o apellido

Buscar

Id entrada	Tipo documento	Número documento	Nombre	Apellido	Fiesta	Lugar	Dirección	Ciudad	Fecha evento	Hora evento	Fecha compra	Hora compra	eliminar
1	25	44630606	Santiago	Spini	Plumazo	Tierra de Sueños	Funes 5437	Rosario	2025-01-25	18:00	2024-11-23	17:30	<div>Borrar</div>
2	18	12333	Ale	Glocer	Plumazo	Tierra de Sueños	Funes 5437	Rosario	2025-01-25	18:00	2024-11-20	19:19:10	<div>Borrar</div>
23	20	38214678	Julia	Roberts	Plumazo	Tierra de Sueños	Funes 5437	Rosario	2025-01-25	18:00	2025-01-15	20:59:30	<div>Borrar</div>

Pestaña para editar una entrada

Datos de la entrada

Id Entrada

16

Tipo de documento

dni

Número de documento

12333

Id de la fiesta

4

Id del lugar

2

Fecha del evento

25 / 01 / 2025

Hora del evento

18 : 00

Fecha de compra

21 / 11 / 2024

Elegir nueva fecha de compra

dd / mm / aaaa

Hora de compra

16 : 43 : 09

Elegir nueva fecha de compra

-- : --

Guardar cambios

Fragmentos de código - CUU02 - Vender entrada

Entrada.java

```
package entidades;

import java.time.LocalDate;

public class Entrada {

    private int identrada;
    private Asistente asistente;
    private Fiesta_lugar fiesta_lugar;
    private LocalDate fecha_compra;
    private LocalTime hora_compra;

    public Entrada() {}

    public Entrada(Asistente asistente, Fiesta_lugar fiesta_lugar, LocalDate fecha_compra, LocalTime hora_compra) {
        super();
        this.asistente = asistente;
        this.fiesta_lugar = fiesta_lugar;
        this.fecha_compra = fecha_compra;
        this.hora_compra = hora_compra;
    }

    public Entrada(int identrada, Asistente asistente, Fiesta_lugar fiesta_lugar, LocalDate fecha_compra, LocalTime hora_compra) {
        super();
        this.identrada = identrada;
        this.asistente = asistente;
        this.fiesta_lugar = fiesta_lugar;
        this.fecha_compra = fecha_compra;
        this.hora_compra = hora_compra;
    }

    public Asistente getAsistente() {
        return asistente;
    }

    public void setAsistente(Asistente asistente) {
        this.asistente = asistente;
    }

    public Fiesta_lugar getFiesta_lugar() {
        return fiesta_lugar;
    }

    public void setFiesta_lugar(Fiesta_lugar fiesta_lugar) {
        this.fiesta_lugar = fiesta_lugar;
    }
}
```

```

public void setFiesta_lugar(Fiesta_lugar fiesta_lugar) {
    this.fiesta_lugar = fiesta_lugar;
}

public int getIdentrada() {
    return entrada;
}

public void setIdentrada(int entrada) {
    this.entrada = entrada;
}

public LocalDate getFecha_compra() {
    return fecha_compra;
}

public void setFecha_compra(LocalDate fecha_compra) {
    this.fecha_compra = fecha_compra;
}

public LocalTime getHora_compra() {
    return hora_compra;
}

public void setHora_compra(LocalTime hora_compra) {
    this.hora_compra = hora_compra;
}

@Override
public String toString() {
    return "Entrada [entrada=" + entrada + ", idasistente=" + asistente.getIdasistente() + ", idfiesta="
        + fiesta_lugar.getFiesta().getIdfiesta() + ", idlugar=" + fiesta_lugar.getLugar().getIdlugar()
        + ", fecha_evento=" + fiesta_lugar.getFecha_fiesta() + ", hora_evento=" + fiesta_lugar.getHora_fiesta()
        + ", fecha_compra=" + fecha_compra + ", hora_compra=" + hora_compra + "]";
}

```

DataEntrada.java

```

package data;

import java.sql.Connection;

public class DataEntrada {

    public LinkedList<Entrada> getAll() {
        Statement stmt=null;
        ResultSet rs=null;
        LinkedList<Entrada> ents = new LinkedList<>();

        try {
            stmt= DbConnector.getInstance().getConn().createStatement();
            rs= stmt.executeQuery("select ent.identrada, asis.idasistente \r\n"
                + " , asis.tipo_doc, asis.nro_doc, asis.nombre, asis.apellido\r\n"
                + " , f.idfiesta, f.nombre_fiesta\r\n"
                + " , l.idlugar, l.nombre_lugar, l.direccion, l.ciudad\r\n"
                + " , fl.fecha_evento, fl.hora_evento\r\n"
                + " , ent.fecha_compra, ent.hora_compra\r\n"
                + "from entrada ent \r\n"
                + "left join fiesta_lugar fl \r\n"
                + " on ent.idfiesta = fl.idfiesta \r\n"
                + " and ent.idlugar = fl.idlugar \r\n"
                + " and ent.fecha_evento = fl.fecha_evento \r\n"
                + " and ent.hora_evento = fl.hora_evento\r\n"
                + "left join fiesta f\r\n"
                + " on ent.idfiesta=f.idfiesta\r\n"
                + "left join lugar l\r\n"
                + " on fl.idlugar=l.idlugar\r\n"
                + "left join asistente asis \r\n"
                + " //+ " #on ent.tipo_doc = asis.tipo_doc \r\n"
                + " on ent.idasistente = asis.idasistente \r\n"
                + "order by ent.identrada, fl.fecha_evento asc, fl.hora_evento asc;");

            if(rs!=null) {
                while(rs.next()) {
                    Entrada ent = new Entrada();
                    ent.setIdentrada(rs.getInt("ent.identrada"));
                    ent.setFecha_compra(rs.getObject("ent.fecha_compra", LocalDate.class));
                    ent.setHora_compra(rs.getObject("ent.hora_compra", LocalTime.class));
                    Asistente asis = new Asistente();
                    asis.setIdasistente(rs.getInt("asis.idasistente"));
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

```

        asis.setTipo_doc(rs.getString("asis.tipo_doc"));
        asis.setNro_doc(rs.getInt("asis.nro_doc"));
        asis.setNombre(rs.getString("asis.nombre"));
        asis.setApellido(rs.getString("asis.apellido"));
        ent.setAsistente(asis);
        Fiesta f = new Fiesta();
        f.setIdfiesta(rs.getInt("f.idfiesta"));
        f.setNombre_fiesta(rs.getString("f.nombre_fiesta"));
        Lugar l = new Lugar();
        l.setIdlugar(rs.getInt("l.idlugar"));
        l.setNombre_lugar(rs.getString("l.nombre_lugar"));
        l.setDireccion(rs.getString("l.direccion"));
        l.setCiudad(rs.getString("l.ciudad"));
        Fiesta_lugar fl = new Fiesta_lugar();
        fl.setFecha_fiesta(rs.getObject("fl.fecha_evento", LocalDate.class));
        fl.setHora_fiesta(rs.getObject("fl.hora_evento", LocalTime.class));
        fl.setFiesta(f);
        fl.setLugar(l);
        ent.setFiesta_lugar(fl);

        System.out.println("IdEntrada: " + ent.getIdentrada() + ", Tipo_doc: " + asis.getTipo_doc() + ", Nro_doc: " + asis.getNro_doc());
        ents.add(ent);
    }
}

} catch (SQLException e) {
    e.printStackTrace();
} finally {
    try {
        if(rs!=null) {rs.close();}
        if(stmt!=null) {stmt.close();}
        DbConnector.getInstancia().releaseConn();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

return ents;
}

```

```

public void add(Entrada en) {
    Connection conn = null;
    PreparedStatement stmt = null;
    ResultSet keyResultSet = null;

    try {
        // Obtener conexión directamente
        conn = DbConnector.getInstancia().getConn();

        // Verificar que la conexión no sea nula
        if (conn == null) {
            System.out.println("ERROR: Conexión a base de datos es nula");
            return;
        }

        // Imprimir estado de autocommit
        System.out.println("Auto-commit status: " + conn.getAutoCommit());

        stmt = conn.prepareStatement(
            "INSERT INTO Entrada(idasistente, idfiesta, idlugar, fecha_evento, hora_evento, fecha_compra, hora_compra) " +
            "VALUES(?, ?, ?, ?, ?, ?, ?)",
            PreparedStatement.RETURN_GENERATED_KEYS
        );

        Asistente asis = en.getAsistente();
        Fiesta_lugar fl = en.getFiesta_lugar();
        Fiesta f = fl.getFiesta();
        Lugar l = fl.getLugar();

        stmt.setInt(1, asis.getIdasistente());
        stmt.setInt(2, f.getIdfiesta());
        stmt.setInt(3, l.getIdlugar());
        stmt.setObject(4, fl.getFecha_fiesta());
        stmt.setObject(5, fl.getHora_fiesta());
        stmt.setObject(6, en.getFecha_compra());
        stmt.setObject(7, en.getHora_compra());

        // Imprimir la consulta SQL (opcional, para depuración)
        System.out.println("Ejecutando consulta SQL: " + stmt.toString());

        int affectedRows = stmt.executeUpdate();
        System.out.println("Filas afectadas: " + affectedRows);

        keyResultSet = stmt.getGeneratedKeys();
    }
}

```

```

// keyResultSet = stmt.executeQuery();
if (keyResultSet != null && keyResultSet.next()) {
    int generatedId = keyResultSet.getInt(1);
    en.setIdentrada(generatedId);
    System.out.println("Entrada generada con ID: " + generatedId);
} else {
    System.out.println("No se generaron claves");
}

} catch (SQLException e) {
    System.out.println("ERROR EN INSERCIÓN:");
    System.out.println("Mensaje: " + e.getMessage());
    System.out.println("Código de error SQL: " + e.getErrorCode());
    System.out.println("Estado SQL: " + e.getSQLState());
    e.printStackTrace();
} finally {
    try {
        if (keyResultSet != null) keyResultSet.close();
        if (stmt != null) stmt.close();
        DbConnector.getInstancia().releaseConn();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}
}

```

```

public Entrada getById(Entrada en) {

    Entrada ent=null;
    DataFiesta df = new DataFiesta();
    DataLugar dl = new DataLugar();
    DataAsistente da = new DataAsistente();
    PreparedStatement stmt=null;
    ResultSet rs=null;
    try {
        stmt=DbConnector.getInstance().getConn().prepareStatement(
            "select e.identrada, a.tipo_doc, a.nro_doc, e.idfiesta, e.idlugar, e.fecha_evento,"
            + " e.hora_evento, e.fecha_compra, e.hora_compra\r\n"
            + "from entrada e \r\n"
            + "inner join asistente a \r\n"
            + "    on a.idasistente = e.idasistente \r\n"
            + "where identrada = ?");

        stmt.setInt(1, en.getIdentrada());
        rs=stmt.executeQuery();
        if(rs!=null && rs.next()) {
            ent=new Entrada();
            ent.setIdentrada(rs.getInt("e.identrada"));

            Asistente asis = new Asistente();
            asis.setTipo_doc(rs.getString("a.tipo_doc"));
            asis.setNro_doc(rs.getInt("a.nro_doc"));
            Asistente a = da.getByDocumento(asis);
            ent.setAsistente(a);

            Fiesta f = df.getById(rs.getInt("e.idfiesta"));

            Lugar l = dl.getById(rs.getInt("e.idlugar"));

            Fiesta_lugar fiesta_lug = new Fiesta_lugar();
            fiesta_lug.setFiesta(f);
            fiesta_lug.setLugar(l);
            fiesta_lug.setFecha_fiesta(rs.getObject("fecha_evento", LocalDate.class));
            fiesta_lug.setHora_fiesta(rs.getObject("hora_evento", LocalTime.class));
            ent.setFiesta_lugar(fiesta_lug);

            ent.setFecha_compra(rs.getObject("fecha_compra", LocalDate.class));
            ent.setHora_compra(rs.getObject("hora_compra", LocalTime.class));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

```

        e.printStackTrace();
    }finally {
        try {
            if(rs!=null) {rs.close();}
            if(stmt!=null) {stmt.close();}
            DbConnector.getInstance().releaseConn();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    System.out.println(ent);

    return ent;
}

```

```

public LinkedList<Entrada> getByUser(Asistente a){
    PreparedStatement stmt=null;
    ResultSet rs=null;
    LinkedList<Entrada> ents = new LinkedList<>();

    try {
        stmt=DbConnector.getInstancia().getConn().prepareStatement("select ent.identrada, asis.idasistente \r\n"
            + " , asis.tipo_doc, asis.nro_doc, asis.nombre, asis.apellido\r\n"
            + " , f.idfiesta, f.nombre_fiesta\r\n"
            + " , l.idlugar, l.nombre_lugar, l.direccion, l.ciudad\r\n"
            + " , fl.fecha_evento, fl.hora_evento\r\n"
            + " , ent.fecha_compra, ent.hora_compra\r\n"
            + "from entrada ent \r\n"
            + "left join fiesta_lugar fl \r\n"
            + " on ent.idfiesta = fl.idfiesta \r\n"
            + " and ent.idlugar = fl.idlugar \r\n"
            + " and ent.fecha_evento = fl.fecha_evento \r\n"
            + " and ent.hora_evento = fl.hora_evento\r\n"
            + "left join fiesta f\r\n"
            + " on ent.idfiesta=f.idfiesta\r\n"
            + "left join lugar l\r\n"
            + " on fl.idlugar=l.idlugar\r\n"
            + "left join asistente asis \r\n"
            + " //+ " #on ent.tipo_doc = asis.tipo_doc \r\n"
            + " on ent.idasistente = asis.idasistente \r\n"
            + "where ent.idasistente = ? \r\n"
            + "order by fl.fecha_evento asc, fl.hora_evento asc;");

        stmt.setInt(1, a.getIdasistente());
        rs=stmt.executeQuery();
    }
}

```

```

        if(rs!=null) {
            while(rs.next()) {
                Entrada ent = new Entrada();
                ent.setIdentrada(rs.getInt("ent.identrada"));
                ent.setFecha_compra(rs.getObject("ent.fecha_compra", LocalDate.class));
                ent.setHora_compra(rs.getObject("ent.hora_compra", LocalTime.class));
                Asistente asis = new Asistente();
                asis.setIdasistente(rs.getInt("asis.idasistente"));
                asis.setTipo_doc(rs.getString("asis.tipo_doc"));
                asis.setNro_doc(rs.getInt("asis.nro_doc"));
                asis.setNombre(rs.getString("asis.nombre"));
                asis.setApellido(rs.getString("asis.apellido"));
                ent.setAsistente(asis);
                Fiesta f = new Fiesta();
                f.setIdfiesta(rs.getInt("f.idfiesta"));
                f.setNombre_fiesta(rs.getString("f.nombre_fiesta"));
                Lugar l = new Lugar();
                l.setIdlugar(rs.getInt("l.idlugar"));
                l.setNombre_lugar(rs.getString("l.nombre_lugar"));
                l.setDireccion(rs.getString("l.direccion"));
                l.setCiudad(rs.getString("l.ciudad"));
                Fiesta_lugar fl = new Fiesta_lugar();
                fl.setFecha_fiesta(rs.getObject("fl.fecha_evento", LocalDate.class));
                fl.setHora_fiesta(rs.getObject("fl.hora_evento", LocalTime.class));
                fl.setFiesta(f);
                fl.setLugar(l);
                ent.setFiesta_lugar(fl);

                System.out.println("IdEntrada: " + ent.getIdentrada() + ", Tipo_doc: " + asis.getTipo_doc()
                    + ", Nro_doc: " + asis.getNro_doc());
                ents.add(ent);
            }
        }

    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        try {
            if(rs!=null) {rs.close();}
            if(stmt!=null) {stmt.close();}
            DbConnector.getInstance().releaseConn();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    return ents;
}

```



```

public void actualizarEntrada(Entrada ent, LocalDate fc_nueva, LocalTime hc_nueva) {
    PreparedStatement stmt = null;
    try {
        stmt = DbConnector.getInstance().getConn().
            prepareStatement("update entrada set fecha_compra = ?, hora_compra = ? where identrada = ?");

        stmt.setObject(1, fc_nueva);
        stmt.setObject(2, hc_nueva);
        stmt.setInt(3, ent.getIdentrada());

        System.out.println(ent.getIdentrada());
        System.out.println(fc_nueva);
        System.out.println(hc_nueva);

        stmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
        System.out.println("Se produjo un error, revise los datos ingresados");
    } finally {
        try {
            if (stmt != null) {
                stmt.close();
            }
            DbConnector.getInstance().releaseConn();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

```

public void deleteByID(Entrada en) {
    PreparedStatement stmt = null;

    try {
        stmt = DbConnector.getInstance().getConn().prepareStatement("delete from entrada where identrada=? ");

        stmt.setInt(1, en.getIdentrada());
        stmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
        System.out.println("Entrada inexistente");
    } finally {
        try {
            if (stmt != null) {
                stmt.close();
            }
            DbConnector.getInstance().releaseConn();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

indexEntrada.jsp

```

<% page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<% page import="entities.Asistente" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="<%= request.getContextPath() %>/Estilos/index.css">
<title>Entradas</title>
</head>
<body>
<%
// HttpSession session = request.getSession();
Asistente loggedInUser = (Asistente) session.getAttribute("user");
if (loggedInUser == null) {
    response.sendRedirect("index.jsp");
} else { %>
<% int isAdmin = (loggedInUser.getIdrol() == 1) ? 1 : 2;
if (isAdmin == 1) { %>
<jsp:include page="menu_cabecera_admin.jsp" />
<h1>Entrada</h1>
<div>
<nav>
<ul class="opc">
<li><a class="button-link href="indexEntrada">Listar todas las entradas</a></li> <!-- admin -->
<li><a class="button-link href="Sv_fiesta_lugar_disponibles_para_entradas">Adquirir entrada</a></li> <!-- admin -->
<li><a class="button-link back-to-main" href="menu.jsp">Volver al menú principal</a></li>
</ul>
</nav>
</div>
</div>
</div>
<% } else if (isAdmin == 2) { %>
<jsp:include page="menu_cabecera_usuario.jsp" />
<h1>Entrada</h1>
<div>
<nav>
<ul class="opc">
<li><a class="button-link href="Sv_fiesta_lugar_disponibles_para_entradas">Adquirir entrada</a></li> <!-- USUARIO (CU Epic) -->
<li><a class="button-link back-to-main" href="menu.jsp">Volver al menú principal</a></li>
</ul>
</nav>
</div>
<% } %>
<% } %>
</body>
</html>

```

registrarEntrada.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="entities.Asistente, entities.Fiesta_lugar, java.util.LinkedList" %>
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="<%= request.getContextPath() %>/Estilos/estilo2.css">
<link rel="stylesheet" type="text/css" href="<%= request.getContextPath() %>/Estilos/estilo_cards.css">
<meta charset="UTF-8">
<title>Comprar entrada</title>
<style>
/* ... (mantener los estilos existentes) ... */
</style>
</head>
<body>
<%
Asistente loggedInUser = (Asistente) session.getAttribute("user");
if (loggedInUser == null) {
    response.sendRedirect("index.jsp");
} else {
    boolean isAdmin = (Boolean) session.getAttribute("isAdmin");
%>

<% if (isAdmin) { %>
<%@ include file="menu_cabecera_admin.jsp" %>
<% } else { %>
<%@ include file="menu_cabecera_usuario.jsp" %>
<% } %>

<h1>Compra tu entrada</h1>

<div>
<form id="purchase-form" action="pago.jsp" method="post">

<input type="hidden" name="id_user" value="<%= loggedInUser.getIdasistente() %>">
<!-- Campos ocultos para los detalles del evento -->
<input type="hidden" name="id_fiesta" id="id_fiesta">
<input type="hidden" name="id_lugar" id="id_lugar">
<input type="hidden" name="fecha_evento" id="fecha_evento">
<input type="hidden" name="hora_evento" id="hora_evento">

```

```

<div id="fiestas">
<%
LinkedList<Fiesta_lugar> eventos = (LinkedList<Fiesta_lugar>) session.getAttribute("fiestasDisponibles");
if (eventos != null && !eventos.isEmpty()) {
    for (Fiesta_lugar evento : eventos) {
%>
<div class="card">
<input type="radio" name="evento"
value="<%= evento.getFiesta().getIdfiesta() %> <%= evento.getLugar().getIdlugar() %>
<%= evento.getFecha_fiesta() %> <%= evento.getHora_fiesta() %> <%= evento.getPrecio() %>"
data-precio="<%= evento.getPrecio() %>">
<div class="precio"><%= String.format("%.2f", evento.getPrecio()) %></div>
<div class="detalles-evento">
<p><strong>Fiesta:</strong> <%= evento.getFiesta().getNombre_fiesta() %></p>
<p><strong>Lugar:</strong> <%= evento.getLugar().getNombre_lugar() %></p>
<p><strong>Fecha:</strong> <%= evento.getFecha_fiesta() %></p>
<p><strong>Hora:</strong> <%= evento.getHora_fiesta() %></p>
</div>
</div>
<%
    }
} else {
%>
<p>No hay fiestas disponibles en este momento.</p>
<%
    }
%>
</div>

<div id="error-message" class="error-message"></div>
<div id="success-message" class="success-message"></div>

<button type="submit" id="payment-button" class="payment-button" disabled>
    Comprar Entrada
</button>
</form>
</div>

```

```

<script>

document.addEventListener('DOMContentLoaded', () => {
  const form = document.getElementById('purchase-form');
  const errorDiv = document.getElementById('error-message');
  const submitButton = document.getElementById('payment-button');

  // Habilitar botón al seleccionar evento
  document.querySelectorAll('input[name="evento"]').forEach(radio => {
    radio.addEventListener('change', () => {
      const [idFiesta, idLugar, fechaEvento, horaEvento] = radio.value.split('_');
      document.getElementById('id_fiesta').value = idFiesta;
      document.getElementById('id_lugar').value = idLugar;
      document.getElementById('fecha_evento').value = fechaEvento;
      document.getElementById('hora_evento').value = horaEvento;

      submitButton.disabled = false;
      errorDiv.style.display = 'none';
    });
  });

  // Validación del formulario
  form.addEventListener('submit', (e) => {
    const selectedEvent = document.querySelector('input[name="evento"]:checked');
    if (!selectedEvent) {
      e.preventDefault();
      showError('Por favor seleccione un evento.');
```

pago.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="entities.Asistente" %>
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="<%= request.getContextPath() %>/Estilos/estilo1.css">
    <meta charset="UTF-8">
    <title>Pago con Stripe</title>
    <script src="https://js.stripe.com/v3/"></script>
    <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
    <style>
        .payment-container {
            max-width: 600px;
            margin: 40px auto;
            padding: 20px;
            background-color: #ffffff;
            border-radius: 8px;
            box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
        }
        #payment-element {
            margin: 20px 0;
        }
        #submit {
            background-color: #5469d4;
            color: white;
            border: none;
            border-radius: 4px;
            padding: 12px 24px;
            font-size: 16px;
            font-weight: 600;
            cursor: pointer;
            display: block;
            width: 100%;
            max-width: 300px;
            margin: 20px auto;
            transition: all 0.2s ease;
        }
        #submit:hover {
            background-color: #4456b3;
            transform: translateY(-1px);
        }
        #submit:disabled {
            opacity: 0.7;
            cursor: not-allowed;
        }
    </style>

```

```

        .payment-message {
            color: #dc3545;
            text-align: center;
            margin-top: 10px;
            font-size: 14px;
        }
        h1 {
            text-align: center;
            color: #333;
            margin-bottom: 30px;
        }
    }
</style>
</head>
<body>
    <%
        Asistente loggedInUser = (Asistente) session.getAttribute("user");
        if (loggedInUser == null) {
            response.sendRedirect(request.getContextPath() + "/index.jsp");
            return;
        }
        // Capturar los parámetros
        String idFiesta = request.getParameter("id_fiesta");
        String idLugar = request.getParameter("id_lugar");
        String fechaEvento = request.getParameter("fecha_evento");
        String horaEvento = request.getParameter("hora_evento");

        // Validar que los parámetros existan
        if (idFiesta == null || idLugar == null || fechaEvento == null || horaEvento == null) {
            response.sendRedirect("Sv_fiesta_lugar_disponibles_para_entradas");
            return;
        }
        int isAdmin = (loggedInUser.getIdrol() == 1) ? 1 : 2;
        if (isAdmin == 1) { %>
            <%@ include file="menu_cabecera_admin.jsp" %>
        } else if (isAdmin == 2) { %>
            <%@ include file="menu_cabecera_usuario.jsp" %>
        } %>
    %>

```

```

<div class="payment-container">
    <h1>Método de pago</h1>
    <input type="hidden" id="id_fiesta" value="<%= idFiesta %>">
    <input type="hidden" id="id_lugar" value="<%= idLugar %>">
    <input type="hidden" id="fecha_evento" value="<%= fechaEvento %>">
    <input type="hidden" id="hora_evento" value="<%= horaEvento %>">

    <div id="payment-element"></div>
    <button id="submit">Pagar</button>
    <div id="payment-message" class="payment-message"></div>
</div>

```

```

<script>
const stripe = Stripe('pk_test_51QNLTpAgf9XwlpIyUb2NyWlxMSZ1RfuYsBKQnaKLoIL18kP9Yyl3fmgQU8oj2kFkQ1RJ7kmM8OoN1lkpoxlpznS700WYWps3S0');
let elements;

var id_user = '<%= loggedInUser.getIdasistente() %>';

initialize();
async function initialize() {
  try {
    const response = await fetch('<%= request.getContextPath() %>/SvCreatePaymentIntent', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({
        amount: 1099,
        // Agregar los detalles del evento
        id_fiesta: document.getElementById('id_fiesta').value,
        id_lugar: document.getElementById('id_lugar').value,
        fecha_evento: document.getElementById('fecha_evento').value,
        hora_evento: document.getElementById('hora_evento').value
      })
    });
    if (!response.ok) {
      throw new Error('Error al crear el PaymentIntent');
    }
    const data = await response.json();

    elements = stripe.elements({
      clientSecret: data.clientSecret,
      appearance: {
        theme: 'stripe',
        variables: {
          colorPrimary: '#5469d4',
        }
      }
    });

    const paymentElement = elements.create('payment');
    await paymentElement.mount('#payment-element');
  } catch (error) {
    console.error('Error:', error);
    showMessage('Error al inicializar el pago: ' + error.message);
  }
}

```

```

document.getElementById('submit').addEventListener('click', async (e) => {
  e.preventDefault();
  const button = e.target;
  button.disabled = true;
  try {
    const { error, paymentIntent } = await stripe.confirmPayment({
      elements,
      redirect: 'if_required',
      confirmParams: {
        return_url: window.location.origin + '<%= request.getContextPath() %>/confirmacion-pago.jsp'
      }
    });
    if (error) {
      console.error('Error de pago:', error);
      showMessage(error.message);
      button.disabled = false;
      return;
    }

    if (paymentIntent && paymentIntent.status === 'succeeded') {
      const eventData = {
        id_user: '<%= loggedInUser.getIdasistente() %>',
        id_fiesta: document.getElementById('id_fiesta').value,
        id_lugar: document.getElementById('id_lugar').value,
        fecha_evento: document.getElementById('fecha_evento').value,
        hora_evento: document.getElementById('hora_evento').value
      };
      // Descomenta esto para debugging
      console.log('Datos del evento a enviar:', eventData);
    }
  }
}

```

```

    try {
      const response = await fetch('<%= request.getContextPath() %>/RegistrarEntrada', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/x-www-form-urlencoded'
        },
        body: new URLSearchParams({
          id_user: eventData.id_user,
          id_fiesta: eventData.id_fiesta,
          id_lugar: eventData.id_lugar,
          fecha_evento: eventData.fecha_evento,
          hora_evento: eventData.hora_evento
        })
      });

      if (!response.ok) {
        throw new Error('Error al registrar la entrada: ' + response.statusText);
      }

      await Swal.fire({
        icon: 'success',
        title: '¡Pago exitoso!',
        text: 'Tu entrada ha sido registrada correctamente.',
        confirmButtonText: 'OK',
        confirmButtonColor: '#4CAF50'
      });

      window.location.href = '<%= request.getContextPath() %>/SvMisEntradas?id_user=' + eventData.id_user;

    } catch (registroError) {
      console.error('Error al registrar entrada:', registroError);
      throw new Error('Error al procesar el registro de la entrada');
    }
  }
} catch (err) {
  console.error('Error completo:', err);
  await Swal.fire({
    icon: 'error',
    title: 'Error de pago',
    text: 'No se pudo procesar el pago. Por favor, intenta de nuevo.',
    confirmButtonText: 'OK'
  });
  button.disabled = false;
}
});
});

```

```

<script>
  function showMessage(message) {
    const messageElement = document.getElementById('payment-message');
    messageElement.textContent = message;
    setTimeout(() => {
      messageElement.textContent = '';
    }, 7000);
  }
</script>
</body>
</html>

```

misEntradas.jsp


```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page import = "java.util.LinkedList"%>
<%@page import = "entities.Entrada"%>
<%@page import = "entities.Lugar"%>
<%@page import = "entities.Fiesta"%>
<%@page import = "entities.Fiesta_lugar"%>
<%@page import = "entities.Asistente"%>
<%@page import = "data.DataAsistente"%>
<%@page import = "data.DataFiesta"%>
<%@page import = "data.DataLugar"%>
<%@page import = "jakarta.servlet.http.HttpSession"%>
<%@page import = "java.util.stream.Collectors"%>
<%@page import = "java.time.LocalDate"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
    <link rel="stylesheet" type="text/css" href="<%= request.getContextPath() %>/Estilos/estilo1.css">
    <title>Mostrar entradas</title>
</head>
<body>

    <%
        // HttpSession session = request.getSession();
        Asistente loggedInUser = (Asistente) session.getAttribute("user");
        if (loggedInUser == null) {
            response.sendRedirect("index.jsp");
        } else {
            <% int isAdmin = (loggedInUser.getIdrol() == 1) ? 1 : 2;
            if (isAdmin == 1) {
                <%@ include file="menu_cabecera_admin.jsp" %>
            } else if (isAdmin == 2) {
                <%@ include file="menu_cabecera_usuario.jsp" %>
            } %>
            <h1>Lista de todas mis entradas</h1>
            <%
                LinkedList<Entrada> listaEntradas = (LinkedList<Entrada>) request.getSession().getAttribute("listaEntradas");
                String searchQuery = request.getParameter("searchQuery");

                if (searchQuery != null && !searchQuery.isEmpty()) {
                    listaEntradas = listaEntradas.stream()
                        .filter(le -> le.getFiesta_lugar().getFiesta().getNombre_fiesta().toLowerCase().contains(searchQuery.toLowerCase()) ||
                            le.getFiesta_lugar().getLugar().getNombre_lugar().toLowerCase().contains(searchQuery.toLowerCase()))
                        .collect(Collectors.toCollection(LinkedList::new));
                }
            %>
            <!-- Formulario de búsqueda -->
            <form method="get" action="">
                <input type="text" name="searchQuery" placeholder="Buscar por nombre de fiesta o lugar"
                    value="<%= searchQuery != null ? searchQuery : "" %>">
                <button type="submit">Buscar</button>
            </form>
    %>

```

```

<div>
  <table>
    <thead>
      <tr>
        <th>Id entrada</th>
        <th>Tipo documento</th>
        <th>Número documento</th>
        <th>Nombre</th>
        <th>Apellido</th>
        <th>Fiesta</th>
        <th>Lugar</th>
        <th>Dirección</th>
        <th>Ciudad</th>
        <th>Fecha evento</th>
        <th>Hora evento</th>
        <th>Fecha compra</th>
        <th>Hora compra</th>
        <th>eliminar</th>
      </tr>
    </thead>
    <tbody>
      <% for (Entrada ent : listaEntradas) {
        DataAsistente da = new DataAsistente();
        DataFiesta df = new DataFiesta();
        DataLugar dl = new DataLugar();
        Asistente asist = ent.getAsistente();
        Fiesta_lugar fl = ent.getFiesta_lugar();
        Fiesta fies = fl.getFiesta();
        Lugar lug = fl.getLugar();

        %>
        <tr>
          <td><%=ent.getIdentrada() %></td>
          <td><%=asist.getTipo_doc() %></td>
          <td><%=asist.getNro_doc() %></td>
          <td><%=asist.getNombre() %></td>
          <td><%=asist.getApellido() %></td>
          <td><%=fies.getNombre_fiesta() %></td>
          <td><%=lug.getNombre_lugar() %></td>
          <td><%=lug.getDireccion() %></td>
          <td><%=lug.getCiudad() %></td>
          <td><%=fl.getFecha_fiesta() %></td>
          <td><%=fl.getHora_fiesta() %></td>
          <td><%=ent.getFecha_compra() %></td>
          <td><%=ent.getHora_compra() %></td>
          <td>
            <form action="SvEliminarEntrada" method="post"
              onsubmit="return confirm('¿Estás seguro de que deseas eliminar esta entrada?');">
              <input type="hidden" name="identrada" value="<%=ent.getIdentrada() %>">
              <button type="submit"> Borrar </button>
            </form>
          </td>
        </tr>
      <% } %>
    </tbody>
  </table>
</div>
<% } %>
</body>
</html>

```

Conclusión

A lo largo del desarrollo de este trabajo práctico, hemos puesto en práctica los conocimientos adquiridos durante el curso de "Lenguaje de Programación JAVA". La aplicación web resultante nos ha permitido profundizar en el uso de tecnologías como JavaScript, JSP y Servlets, así como en la implementación de bases de datos MySQL y la conexión con aplicaciones Java mediante JDBC.

El proceso de desarrollo no ha estado exento de desafíos. La configuración del entorno de desarrollo, la resolución de errores de compilación y la integración de la API de Stripe han sido algunos de los obstáculos que hemos superado. Sin embargo, cada desafío nos ha brindado la oportunidad de aprender y fortalecer nuestras habilidades como desarrolladores.

A pesar de las dificultades, hemos logrado crear una aplicación web funcional que cumple con los requisitos del trabajo práctico. La aplicación permite a los usuarios registrarse, iniciar sesión, comprar entradas para eventos, realizar comentarios y gestionar la información de su perfil. Además, la integración con Stripe proporciona una forma segura y eficiente de procesar pagos online.

En general, consideramos que este trabajo práctico ha sido una experiencia de aprendizaje valiosa que nos ha permitido aplicar nuestros conocimientos y desarrollar nuevas habilidades en el desarrollo de aplicaciones web con Java.

Código fuente

- Repositorio git: https://github.com/martinglocer/Java_TP_Boliche.git