

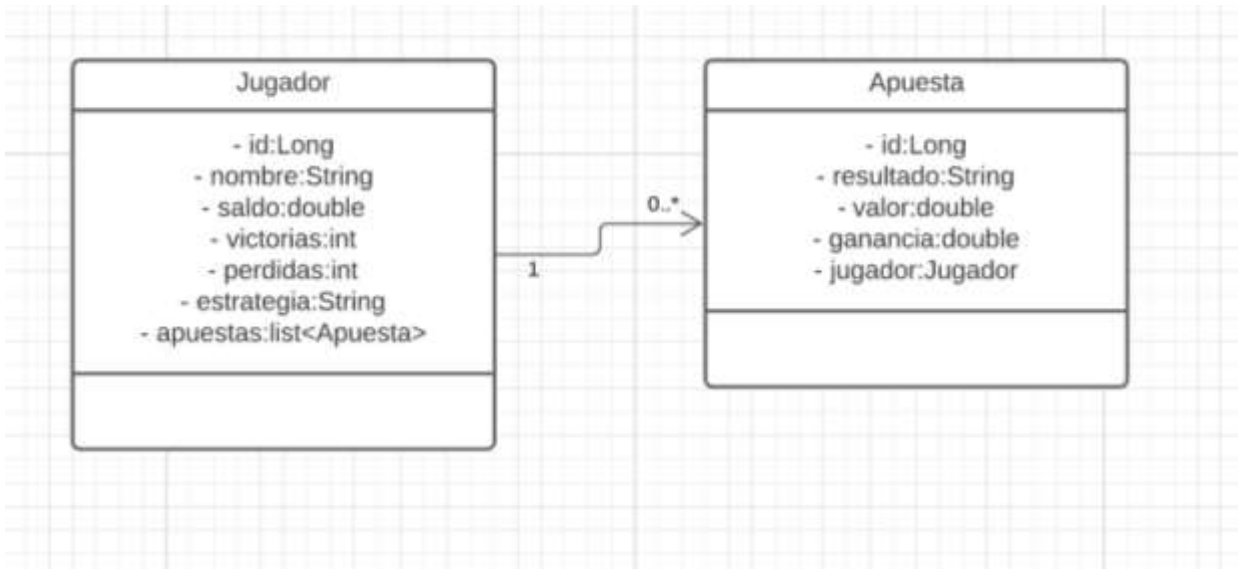
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

			PRÁCTICA DE LABORATORIO		
CARRERA: COMPUTACIÓN			ASIGNATURA: PROGRAMACIÓN APLICADA		
NRO. Examen	2	Examen Final			
OBJETIVO ALCANZADO: <ul style="list-style-type: none"> Consolidar los conocimientos adquiridos en clase sobre JPA. 					
ACTIVIDADES DESARROLLADAS					
<p>1. Se desea simular los posibles beneficios de diversas estrategias de juego en un casino. La ruleta francesa es un juego en el que hay una ruleta con 37 números (del 0 al 36). Cada 2000 (tiempo parametrizable) milisegundos el croupier saca un número al azar y los diversos hilos clientes apuestan para ver si ganan. Todos los hilos empiezan con 1.000 euros y la banca (que controla la ruleta) con 50.000. Cuando los jugadores pierden dinero, la banca incrementa su saldo.</p> <ul style="list-style-type: none"> Se puede jugar a un número concreto. Habrá 4 hilos clientes que eligen números al azar del 1 al 36 (no el 0) y restarán 10 euros de su saldo para apostar a ese ese número. Si sale su número su saldo se incrementa en 360 euros (36 veces lo apostado). Se puede jugar a par/impar. Habrá 4 hilos clientes que eligen al azar si apuestan a que saldrá un número par o un número impar. Siempre restan 10 euros para apostar y si ganan incrementan su saldo en 20 euros. Se puede jugar a la «martingala». Habrá 4 hilos que eligen números al azar. Elegirán un número y empezarán restando 10 euros de su saldo para apostar a ese número. Si ganan incrementan su saldo en 360 euros. Si pierden jugarán el doble de su apuesta anterior (es decir, 20, luego 40, luego 80, y así sucesivamente) La banca acepta todas las apuestas, pero nunca paga más dinero del que tiene. Si sale el 0, todo el mundo pierde y la banca se queda con todo el dinero. 					

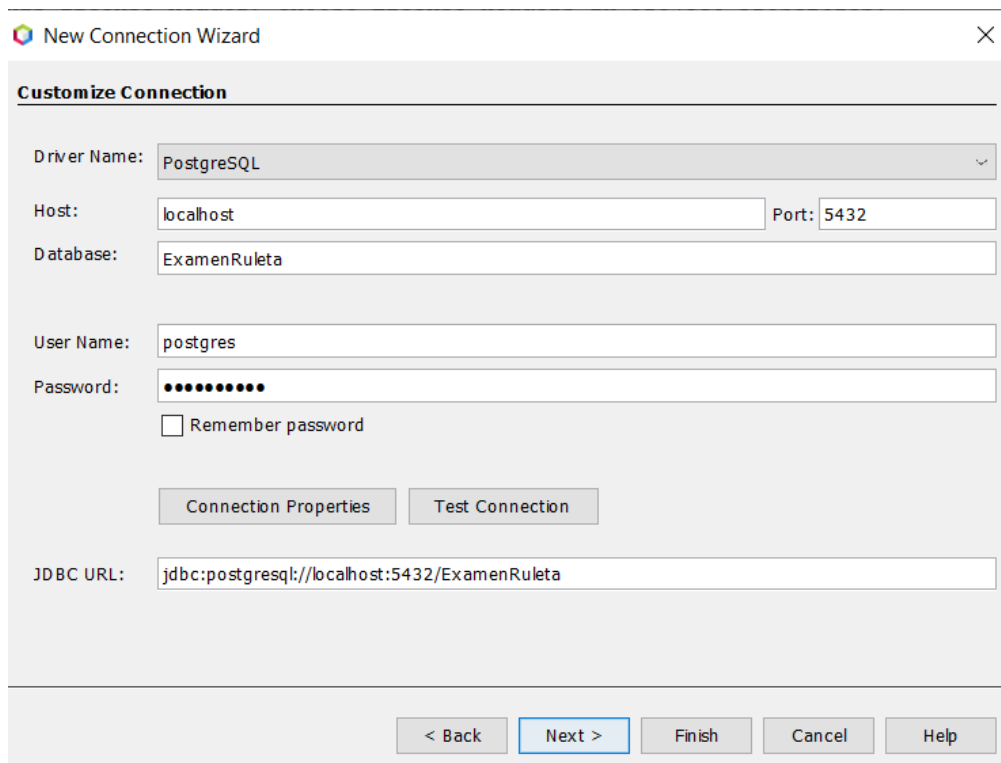
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

- **Desarrollo de la Aplicación**

1. Se desarrollo un diagrama de clases




2. Se creo una nueva base de datos con el nombre de ExamenRuleta



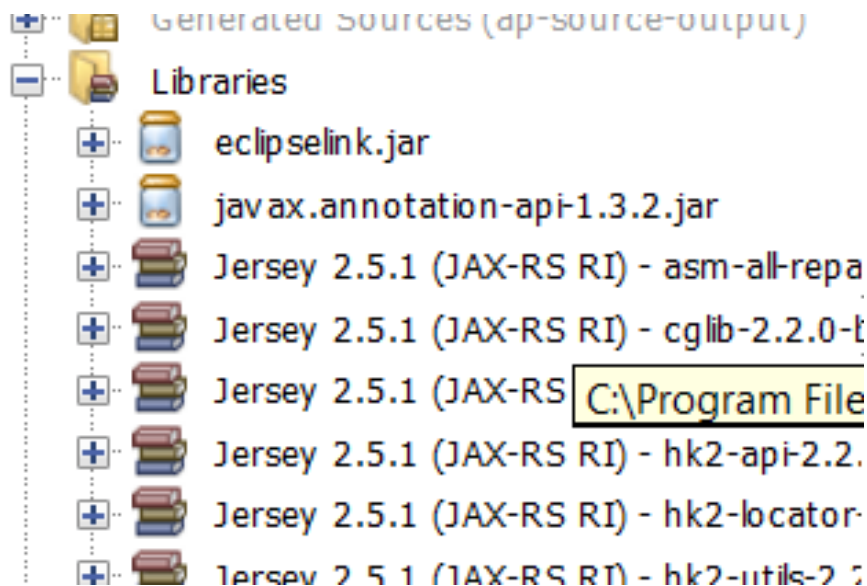
The screenshot shows the 'New Connection Wizard' dialog box, specifically the 'Customize Connection' tab. The fields are filled as follows:

- Driver Name:** PostgreSQL
- Host:** localhost
- Port:** 5432
- Database:** ExamenRuleta
- User Name:** postgres
- Password:** (masked with dots)
- ☐ Remember password
- Connection Properties** and **Test Connection** buttons are visible.
- JDBC URL:** jdbc:postgresql://localhost:5432/ExamenRuleta

At the bottom, there are navigation buttons: **< Back**, **Next >** (highlighted), **Finish**, **Cancel**, and **Help**.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

3. Se creó un nuevo proyecto con el nombre Examen(ProgramacionAplicada) en el cual se importaron las siguientes librerías:




4. Dentro del paquete ec.edu.ups.utils se creó la clase JPAUtils:

```
public class JPAUtils {

    private static final EntityManagerFactory emf = Persistence.createEntityManagerFactory("Examen_ProgramacionAplicada_PU");

    public static EntityManager getEntityManager() {
        return emf.createEntityManager();
    }

}
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

5. Dentro del paquete ec.edu.ups.modelo se crearon las entidades definidas en el diagrama de clases

- Apuesta

```

@Entity
public class Apuesta implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    @Column
    private double valor;

    @Column
    private String resultado;

    @Column
    private double ganancia;

    @ManyToOne
    @JoinColumn(name = "fk_jugador")
    private Jugador jugador;

    public Apuesta() {
    }
}

```

- Jugador

```
@Entity
public class Jugador implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    @Column
    private String nombre;

    @Column
    private double saldo;

    @Column
    private int victorias;

    @Column
    private int perdidas;

    @Column
    private String estrategia;

    @OneToMany(mappedBy = "jugador", cascade = CascadeType.ALL)
    private List<Apuesta> apuestas;
```

6. Dentro del paquete `ec.edu.ups.controlador` se desarrolló un controlador genérico para lo cual se usó los conocimientos adquiridos sobre programación genérica y reflexión :

```
public abstract class AbstractControlador<E> {
    private List<E> lista;
    private Class<E> clase;
    private EntityManager em;

    public AbstractControlador() {
        Type t = getClass().getGenericSuperclass();
        ParameterizedType pt = (ParameterizedType) t;
        clase = (Class) pt.getActualTypeArguments()[0];
        em = JPAUtils.getEntityManager();
        lista = buscarTodo();
    }

    public AbstractControlador(EntityManager em) {
        lista = new ArrayList<>();
        Type t = getClass().getGenericSuperclass();
        ParameterizedType pt = (ParameterizedType) t;
        clase = (Class) pt.getActualTypeArguments()[0];
        em = JPAUtils.getEntityManager();
        this.em = em;
    }

    public boolean crear(E objeto) {
        if (validar()) {
            em.getTransaction().begin();
            em.persist(objeto);
            em.getTransaction().commit();
            lista.add(objeto);
            return true;
        }
        return false;
    }

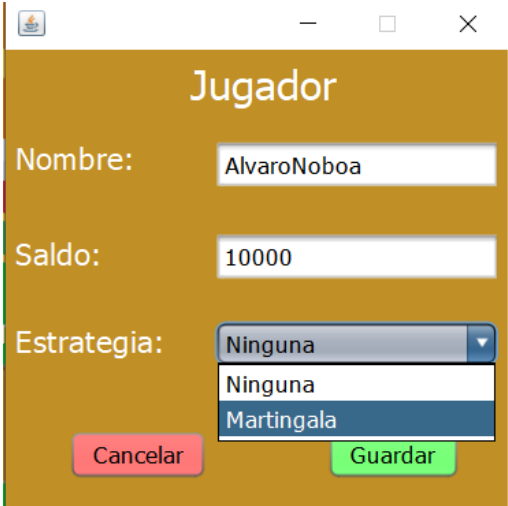
    public boolean eliminar(E objeto) {
        em.getTransaction().begin();
        em.remove(em.merge(objeto));
    }
}
```

7. Finalmente se diseñó la interfaz para el usuario:

- Ventana de Principal:



- Ventana para registrar un nuevo Jugador:



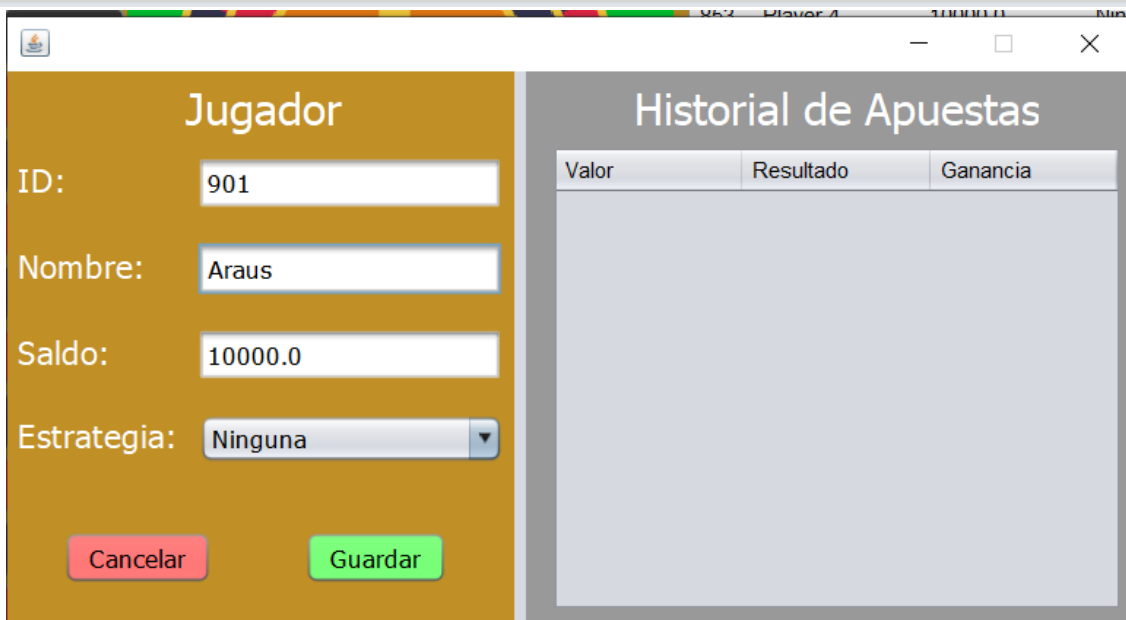
- Ventana para Modificar Jugador



The main interface features a roulette wheel in the center. To the left is a 36-number grid (0-36) with alternating red and black backgrounds. Below the grid is a 'Saldo de La Banca' display showing 50000.0. To the right of the wheel is a table of players:

ID	Jugador	Saldo	Estrategia
801	Player 1	10000.0	Ninguna
851	Player 2	10000.0	Ninguna
852	Player 3	10000.0	Ninguna
853	Player 4	10000.0	Ninguna
854	Fraille	10000.0	Ninguna
901	AleandroHobos	10000.0	Martingala

Buttons for 'Nuevo', 'Eliminar', and 'Modificar' are on the right. Below the player table is a 'Jugadores en mesa' section with input fields for 'Apuesta' and 'Numero' for players P1, P2, P3, and P4. At the bottom are buttons for 'Apostar Aleatoriamente', 'Eliminar', and 'Agregar'.




The 'Modificar Jugador' window has two panes. The left pane is titled 'Jugador' and contains the following fields:

- ID: 901
- Nombre: Araus
- Saldo: 10000.0
- Estrategia: Ninguna (dropdown menu)

Buttons for 'Cancelar' and 'Guardar' are at the bottom. The right pane is titled 'Historial de Apuestas' and contains a table with columns 'Valor', 'Resultado', and 'Ganancia'.



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

- Link Git:

<https://github.com/VazquezAriel/ExamenFinal-ProgramacionAplicada/tree/main>

RESULTADO(S) OBTENIDO(S):

- Conocimos el funcionamiento de JPA y su implementación en JAVA.

CONCLUSIONES:

- Pudimos reforzar el contenido aprendido durante todo el ciclo

RECOMENDACIONES:

- Revisar la información proporcionada por el docente previo a la práctica
- Haber asistido a las sesiones de clase.
- Consultar con el docente las dudas que puedan surgir al momento de realizar la práctica

Nombre de estudiante: Ariel Vazquez

Firma de estudiante:

