



Objetivo:

- Consolidar los conocimientos adquiridos en clase sobre Java.

Enunciado:

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.

- 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 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.

Adicionalmente, se deberá generar un sistema de base de datos con JPA en donde puede gestionar a los clientes o hilos jugadores, con cada una de las apuestas realizadas, los valores que se están manejando tanto de la banca como de cada cliente y gestionar la simulación es decir se puede iniciar y parar en cualquier intervalo de tiempo en la simulación, además de poder cambiar a cualquier cliente con un nuevo o un anterior y en que modalidad va a jugar. Por otro lado, es parametrizable el tiempo que se demora dar la vuelta a la ruleta con el proceso de apuesta.

Es importante destacar que debe existir un sistema de simulación visual y un sistema de gestión de jugadores, transacciones y apuestas en donde se evidencia la apuesta, el jugador, la ruleta el número generado y como varían los saldos de los que intervienen dentro del juego.

Por ultimo se debe presentar dos reportes o tablas de los datos:

1. Clientes y la banca con el número de transacciones o apuestas realizadas, el valor de total y cuantas a perdido y cuantas veces a ganado además de la modalidad de juego.
2. Dentro de cada cliente se puede acceder al historial de apuestas y transacciones realizadas.



Examen

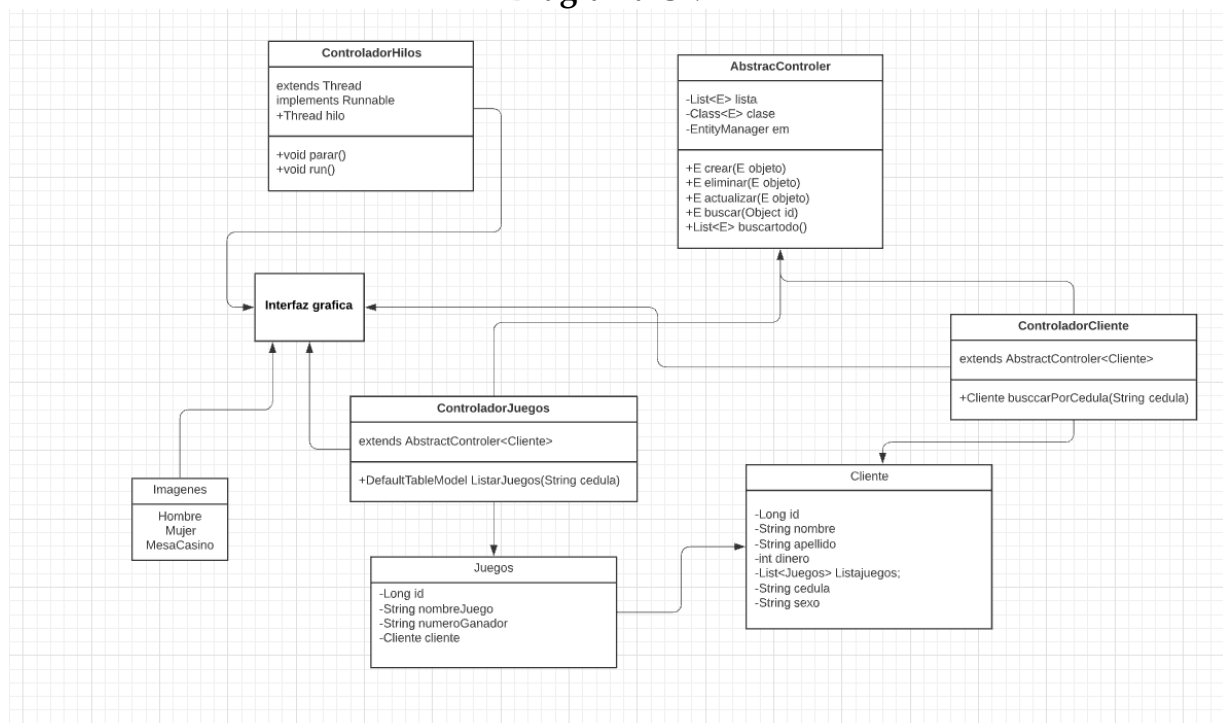
29/01/2021

Se calificará el avance con los siguientes criterios de evaluación:

- JPA: 25%
- Hilos (Thread): 15 %
- Excepciones: 5%
- MVC: 10%
- Diagrama de clases: 10%
- Usabilidad – Vista - Simulación: 25%
- Programación genérica, Java 8, reflexión: 10%

Entrega: Subir al Git el documento en formato PDF de la práctica y código hasta las **23:55** del domingo **31 de Enero del 2021**.

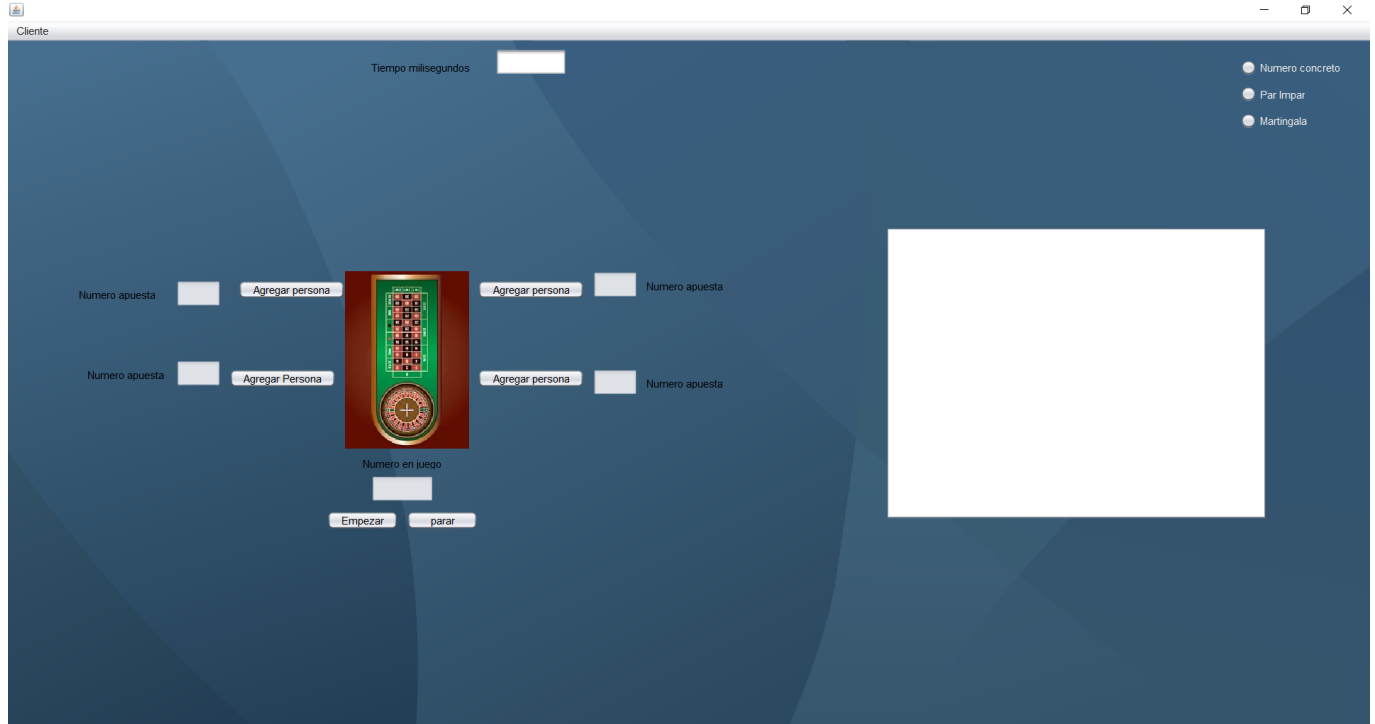
Diagrama UML



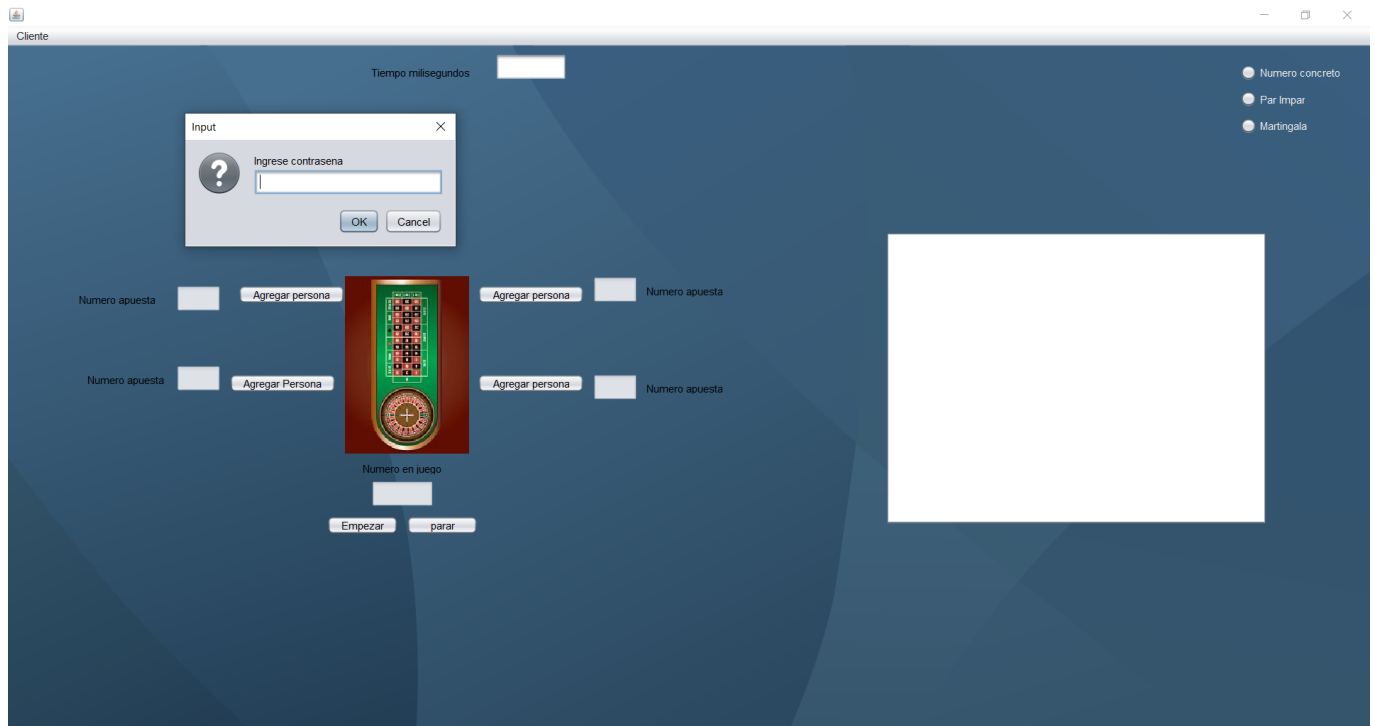


Funcionamiento

Pantalla principal



Para iniciar la simulación se deberá crear jugadores para esto se deberá conocer la clave que es 3103



Después de poner la clave podremos crear los clientes



Examen

29/01/2021

Cliente

Registrar Cliente

Ingrese cedula

Ingrese nombre

Ingrese apellido

☐ Hombre ☐ Mujer

Registrar

Una vez creado como mínimo 4 jugadores pasaremos a ingresarlos en el juego mediante su numero de cedula

Numero apuesta **Adrian Lopez**

Numero apuesta **Juan Perez** Numero apuesta

Numero apuesta **Tracy Cardenas**

Julia Arizaga Numero apuesta

Numero en juego

Empezar **parar**

En la esquina superior derecha podremos escoger el modo de juego



Examen

29/01/2021

☒ Numero concreto

☐ Par Impar

☐ Martingala

Una vez seleccionado podremos dar en empezar donde en un JTextArea se mostrará quien va ganando y se ira actualizando el dinero de los jugadores y el crupier

Cliente

Tiempo milisegundos 2000

☒ Numero concreto
☐ Par Impar
☐ Martingala

Numero apuesta 5 Adnan Lopez 15 Juan Perez Numero apuesta

Numero apuesta 18 Tracy Cardenas 29 Julia Arizaaga Numero apuesta

Numero en juego

Empezar parar

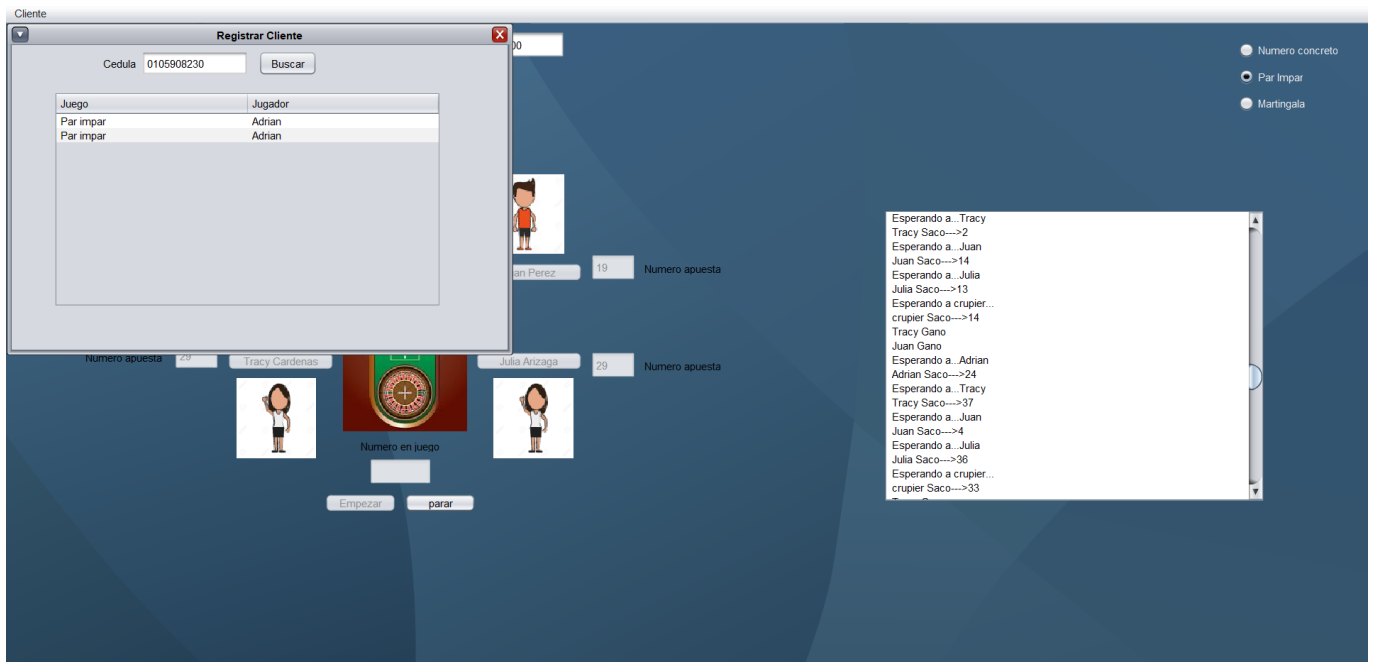
Esperando a Adnan
Adrian Saco-->19
Esperando a Tracy
Tracy Saco-->35
Esperando a Juan
Juan Saco-->30
Esperando a Julia
Julia Saco-->37
Esperando a crupier...
crupier Saco-->2
Esperando a Adnan
Adrian Saco-->28
Esperando a Tracy
Tracy Saco-->32
Esperando a Juan
Juan Saco-->36
Esperando a Julia
Julia Saco-->30
Esperando a crupier...
crupier Saco-->1

Podremos observar cuantos juegos a ganado cada jugador y que tipo de juegos a ganado eso en la pestaña cliente esto se observara con la cedula de los clientes

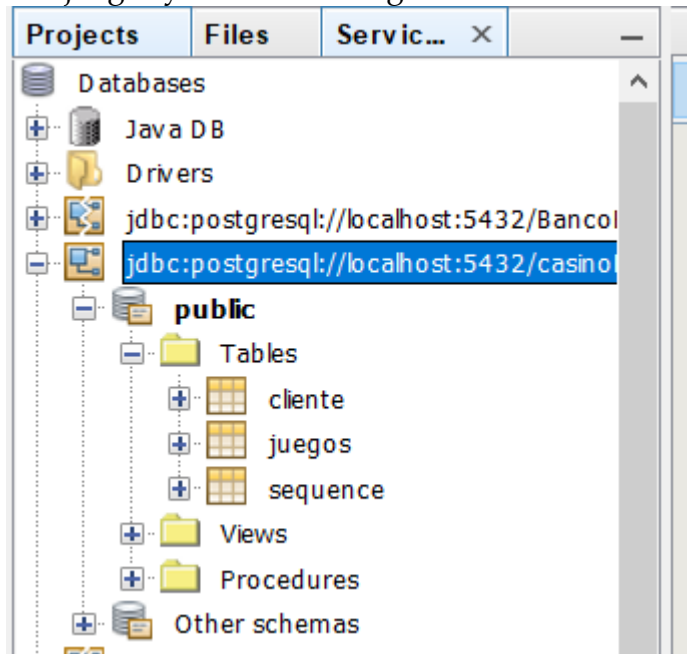


Examen

29/01/2021



Los juegos y clientes están guardados en una base de datos PostgreSQL



Mediante JPA



Programación Aplicada

Tema: Java Aplicado.

Examen

29/01/2021

The screenshot shows an IDE window with multiple tabs open, including `persistence.xml`, `Cliente.java`, `Juegos.java`, `PantallaRegistrarCliente.java`, `ControladorCliente.java`, `AbstractControler.java`, `PantallaJuegosGanados.java`, and `ControladorJuego...`. The `persistence.xml` tab is active, displaying the XML configuration for a Persistence Unit named `CasinoPU`.

The `General` tab of the `CasinoPU` configuration is shown, with the following settings:

- Persistence Unit Name:** Please wait...
- Persistence Provider:** (Empty dropdown)
- JDBC Connection:** No Available Connection
- ☒ Use Java Transaction APIs
- Table Generation Strategy:** ☐ Create ☐ Drop and Create ☐ None
- ☐ Include All Entity Classes in "(0)" Module
- Include Entity Classes:** (Empty list)

The `Properties tab is also visible at the bottom.`