

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

		FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES	
CARRERA:		ASIGNATURA: Programación Aplicada	
NRO. PRÁCTICA:		TÍTULO PRÁCTICA: Examen_Final	
OBJETIVO ALCANZADO: Consolidar los conocimientos adquiridos en clase sobre Java.			
ACTIVIDADES DESARROLLADAS			
1. Se pidió crear una aplicación que simule varios juegos de un casino. Para la simulación del casino se crearon seis paquetes. <ul style="list-style-type: none"> • Ec.edu.ups.modelo • Ec.edu.ups.controlador • Ec.edu.ups.vista • META-INF • Ec.edu.ups.utils • Ec.edu.ups.imagenes 			
2. Ec.edu.ups.modelo En esta clase se crearon todos los modelos que interactuaran con los controladores. <ul style="list-style-type: none"> • Jugadores • Ruleta 			
3. Jugadores Para esta clase se utilizó el lenguaje de JPA y anotaciones para crear las columnas en la base de datos de PostgreSQL. Atributos: <ul style="list-style-type: none"> @Id <ul style="list-style-type: none"> • private int id; @Column <ul style="list-style-type: none"> • private double dinero; @Column <ul style="list-style-type: none"> • private int vecesPerdidas; @Column <ul style="list-style-type: none"> • private int numeroApostado; 			

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

@Column

- private String tipoParlImpar;

Se crearon todos los getter y setters para los atributos.

4. Ruleta

Para esta clase se utilizó el lenguaje de JPA y anotaciones para crear las columnas en la base de datos de PostgreSQL.

Atributos:

@Id

- private int id;

@Column

- private double valorDeCaja;

Se crearon todos los getter y setters para los atributos.

5. ec.edu.ups.Controlador

En este paquete se crearon todas las clases que interactúan con la interfaz del usuario.

- ControladorGenerico
- ControladorJugador
- ControladorRuleta

6. ControladorGenerico

El controlador genérico es una clase abstracta para poder reutilizar sus métodos en otros controladores.

Atributos:


- private Class<T> clase;
- private EntityManager em;

en esta clase se crearon dos constructores. En el primero constructor no se recibe ninguno objeto en su parámetro, y este inicializa los atributos de la misma clase. El segundo constructor recibe un objeto de tipo Entity Manager, y este inicializa todos los atributos de la misma clase.

Se crearon todos los getters y setters de los atributos de esta clase

Métodos:

- **Créate:** Este método recibe un objeto en su parámetro. Este método realiza la tarea de conseguir la base de datos y crear el objeto dentro de la base una vez realizada la tarea realiza un commit y finaliza.

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

- Read: Este método recibe un objeto en su parámetro. Este método realiza la tarea de buscar un objeto por su primary key, una vez que lo haya hallado lo retorna, caso contrario retorna un null.
- Update: Este método recibe un objeto en su parámetro. Este método realiza la tarea de actualizar el objeto dentro de la base de datos, una vez realizada la tarea realiza un commit y finaliza.
- Delete: Este método recibe un objeto en su parámetro. Este método realiza la tarea de eliminar el objeto dentro de la base de datos, una vez realizada la tarea realiza un commit y finaliza.
- findAll: Este método realiza la tarea de devolver una lista de objetos.

7. Controlador Jugador

Esta clase hereda de la clase Controlador Genérico, en si esta clase reutiliza todos los métodos del controlador genérico.

Atributos:

- private Random randomico;

métodos:

- elegirAlazar: Este método la única tarea que realiza es devolver un numero al azar entre 1-36
- elegirParImpar: Este método realiza la tarea de escoger una numero al azar entre 1-2, y de eso determina si devuelve un hilo que es "PAR" o "IMPAR"

8. Controlador Ruleta

Esta clase hereda de la clase Controlador Genérico, en si esta clase reutiliza todos los métodos del controlador genérico.

Atributos:

- private Random randomico;

métodos:

- girarRuleta: Este método la única tarea que realiza es devolver un numero al azar entre 0-36

9. Ec.edu.ups.utils

Este paquete solamente contiene la clase utils para poder conectar la base de datos con el controlador.

- Utils

10. Utils


Esta clase realiza la tarea de conseguir el Entity Manager.

Atributos:

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

métodos:

- getEntityManager: Este método solamente realiza la tarea de devolver el entity manager.

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

11. Ec.edu.ups.imagenes

Este paquete realiza la tarea de guardar las imágenes que se utilizaran para la interfaz.

12. META_INF

Este paquete contiene una clase de tipo persistince.xml.

13. Ec.edu.ups.vista

Este paquete contiene todas las interfaces y clases que se utilizaron para simular los juegos de casinos.

- HiloRuleta
- Jugar
- PanelRuleta
- VentanaPrueba

14. HiloRuleta

Este método hereda de la clase Thread.

Atributos:

- private int i =0;
- private boolean continuar;
- private PanelRuleta p;

Esta clase tiene solamente un constructor en el cual solamente recibe un panel en su parámetro y lo inicializa.

También se han creado todos los getters y setters de los atributos.

Métodos:


- Run: este método esta sobrescrito de la clase padre. Realiza la tarea de girar la imagen cada vez por 25. En caso de que el valor de la variable sea mayor que 500 solamente se manda a que el valor se reinicie a 0.

15. Jugar

Esta clase hereda de la clase Thread. Esta clase tiene la tarea de simular como los jugadores realizan los pagos por los juegos y el resultado del juego.

Atributos:

- private boolean continuar;
- private int juego;
- private VentanaPrueba ventanaP;
- private Ruleta r;
- private List<Jugadores> jugadores;
- private ControladorRuleta controladorR = new ControladorRuleta();
- private ControladorJugador controladorJ = new ControladorJugador();
- private DefaultListModel<String> modeloLista;
- private JList listaInterfaz;
- private JTextField texto;

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

El constructor de esta clase recibe en su parámetro un defaultModel, Jlist, y un JTextField. De la misma manera inicializa todos estos parámetros dentro de la clase.

Se han creado los métodos getters y setters de cada atributo en la clase.

Métodos:

- **cobroNumeroConcreto:** Este método solamente realiza la tarea de cobrar a cada jugador 10 por cada vez que vaya a jugar numero concreto.
- **resultadoNumeroC:** Este método llama al controlador ruleta y pide que le consiga un numero alzar entre 0-36. En caso de que el numero que conseguí es 0 manda para imprimir en pantalla que todos los jugadores han perdido. En caso de que uno o mas jugadores hayan ganado llama al controlador jugador y actualiza la cuenta del jugador con 360 dólares, de la misma manera llama al controlador ruleta y resta los 360 dólares de la cuenta de la ruleta. En caso de que no hay obtenido el mismo numero que la ruleta solamente manda a imprimir en pantalla que el jugador ha perdido.
- **cobroParImpar:** Este método solamente realiza la tarea de cobrar a cada jugador 10 por cada vez que vaya a jugar par/impar.
- **resultadoParImpar:** Este método llama al controlador ruleta y solicita que le devuelva un numero al azar entre 0-36. Si el numero que consiguió es 0 solamente manda a que se visualice en la tabla que todos los jugadores han perdido. En caso de que el jugador hay apostado por un numero par y la ruleta hay conseguido un numero par llama al controlador jugador y actualiza la cuenta del jugador con 20 dólares mas. En caso de que el jugador haya apostado por impar y la ruleta haya conseguido un numero impar se llama al controlador jugador y se actualiza la cuenta del jugador con 20 dólares mas. En caso de que ninguno de los dos casos se cumpla solamente se visualizara en pantalla que el jugador ha perdido.
- **cobroMartingala:** Este método calcula el pago de los jugadores por las veces perdidas. Se llama al controlador jugador y se le resta la suma a pagar. Por ultimo se llama al controlador ruleta y se suma es valor a la caja de la ruleta.
- **resultadoMartingala:** El juego se simula de la misma manera que el juego de numero concreto. En caso de que el jugador pierda solamente se incrementa el numero de veces perdidas.
- **run:** Este método esta sobrescrito de la clase padre. Este método basándose del juego que se ha elegido correrá el proceso.

16. panelRuleta

Este panel solamente se utiliza para mover la imagen de forma circular.

Atributos:

- `BufferedImage img;`
- `double rotateAngle;`

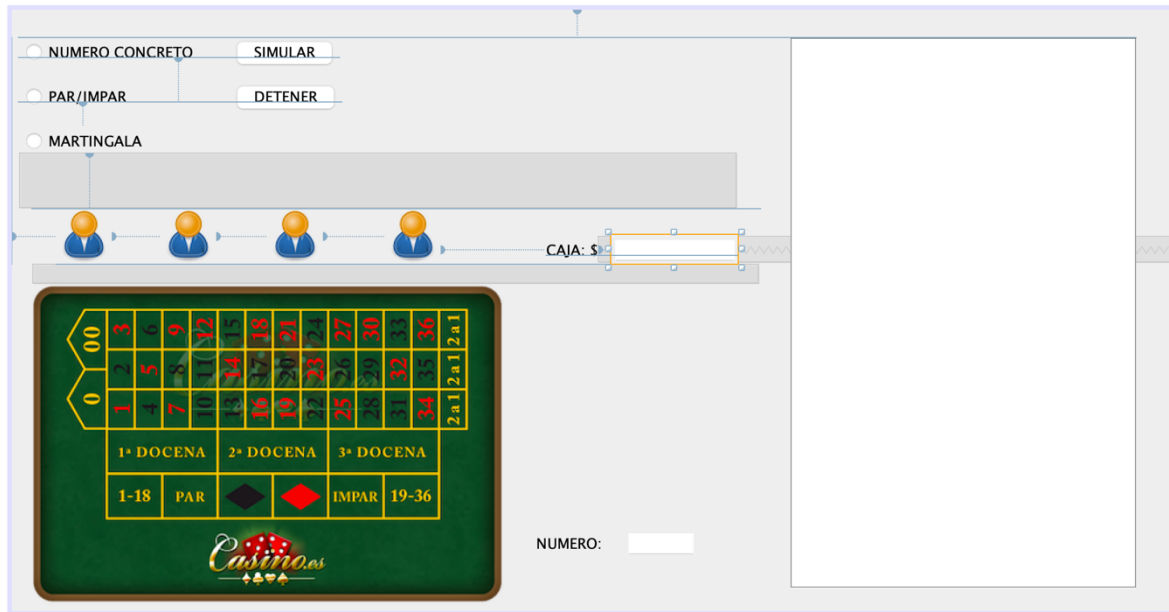
Esta clase solamente tiene un constructor donde solamente inicializa el buffered image.

Métodos:

- **loadImage:** Este método realiza la tarea de inicializar el atributo de buffered image con el path de la imagen que obtiene del parámetro.


- rotate: este método recibe un objeto de tipo integer en su parámetro. Después realiza la tarea de repintar la imagen con el nuevo angulo.
- Paint: Este método realiza la tarea de rotar la imagen.

17. ventanaPrueba



Este interfaz visualiza la simulación del casino. Se debe seleccionar un juego y presionar un botón de simulación. En case de que quiera detener una simulación solo se presionar el botón de "DETENER". Los datos de los jugadores se deberían de cargar en el tablero.

RESULTADO(S) OBTENIDO(S):

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

RECOMENDACIONES:

Nombre de estudiante: Denys Dutan

Firma de estudiante: 