

---

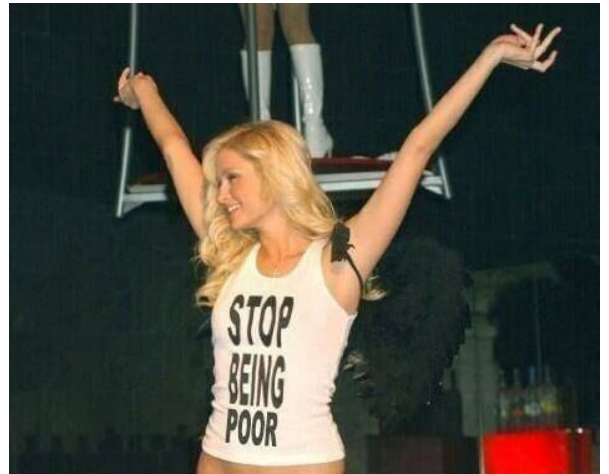
# TP N° 4 LABORATORIO II

---

Roberta Gabor

DIVISIÓN 2ª  
TURNO MAÑANA  
FECHA DE ENTREGA LUNES 23/11  
PROFESOR MAXIMILIANO NEINER

# CASINO

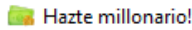


# EL AFANO

Sistema de apuestas donde uno compra monedas y/o boletos y juega por dinero; se asume que en algún momento se puso una tarjeta de crédito, y la plata que se suma en la billetera del casino se descuenta de la tarjeta

## Forms:

### Form Inicial "CASINO"

 Hazte millonario! ×

## CASINO

	dni	saldo	variacion	transaccion
▶	1	595	120	gana
	2	0	-1080	pierde
	4	220	45	pierde
*				

COMPRAR MONEDAS

JUGAR!

SALIR

En el form inicial se podrá ver la ultima jugada de cada usuario.

DNI: Identificador de cada usuario único irrepetible.

saldo: saldo actual, es el reflejo de la cantidad de monedas en billetera pasado a pesos

variación: si compra o gana aparecerá la ganancia pasada a pesos, y si se pierde

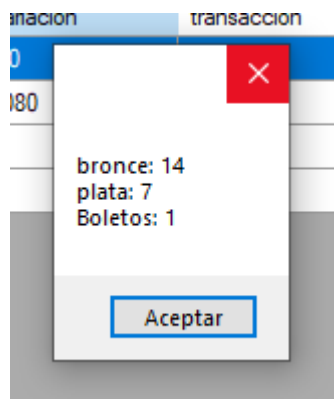
aparecerá con un -.

transacción: muestra si GANO, PERDIO, O COMPRO.

El DT reflejado se lee desde un DT guardado en XML (Apéndice al final menciona implementaciones), igual que el guardado.

No se podrá cerrar a menos que los demás forms están cerrados.

Si se da doble clic en el row del usuario nos informara como se compone su billetera:



## Form “COMPRAR”

Desde el form Inicial se apreta comprar:

A screenshot of a form titled "¡A Gastar!". The form has three main sections: "ID JUGADOR" with a text input field, "TIPO DE MONEDA" with a dropdown menu showing "oro", and "CANTIDAD" with a text input field. Below these fields is a question mark icon. At the bottom, there are two buttons: "CANCELAR" and "ACEPTAR".

FORM DE TIPO MODAL

El ID aka DNI podrá ya existir, de ya existir se agregará la compra a la billetera de ese usuario.

**Se pueden comprar 2 productos distintos: *Monedas y boletos.***  
***Monedas se divide en tipos: Oro, Bronce, Plata.***

No se podrá comprar si→

El DNI no es un numero valido.

La cantidad comprada no es válido.

Si se quiere comprar boletos y no tiene las monedas de bronce suficiente.

Antes de jugar si o si se debe comprar monedas para poder apostar y registrar el usuario en el casino y la base de datos. No se podrán comprar boletos en primera instancia tampoco, ya que cuestan 5 monedas de bronce.

**Si da doble clic en “?” aparecerán los precios correspondientes a cada moneda**

Si compra un boleto se le descontarán 5 monedas de bronce de la billetera y bajará su dinero en cartera.

## Form “Jugar”

## Form modal

Al jugar si o si debe estar registrado el usuario; la cantidad de monedas y el tipo debe coincidir con lo disponible en la cartera del este

Al apretar el radio botón se mostrarán (si ya se escribió el DNI y se encuentra registrado, sino seguirá blanco) la cantidad de boletos disponibles.

Si se cumple todo al apretar jugar se abrirá la ruleta en otro form. Este form no se podrá cerrar hasta que termine su ejecución e hilo la ruleta. Si al frenar el hilo con stop no se cierra y se intenta cerrar este form, ruleta se cerrará automáticamente.

Si se apretó el radio botón de boletos y se juega se descontará de su cartera.

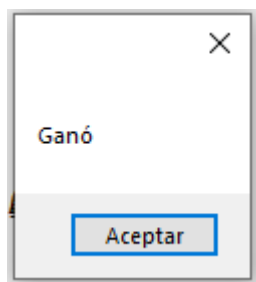
## Form “Ruleta”



Este form aplicará un nuevo hilo, que lo que hace es girar una foto cada 150 ms. Al apretar PARAR enviará un evento de FRENACION que llegará hasta el manejador en el WF principal (donde se declara el hilo, y se comienza) que abortará en hilo. El form de JUGAR no se podrá cerrar mientras este siga vigente. Si mientras se está ejecutando el hilo se apreta X (closing) hará lo mismo que STOP, si se apreta el X luego de haber apretado STOP para no generar error y abortar 2 veces, se identifica que tipo de motivo tiene X, identificando si el EventArgs es Empty o no.

Si se cierra sin poner stop no se ejecutará ninguna jugada ni se ganará ni perderá.

Al apretar parar saldrá un cartel si se ganó o no: (con un sonido adecuado)



Aclaración:

Al ser información instantánea que se efectúa inmediatamente en la billetera, al efectuar cualquier tipo de movimiento se verá reflejado en el DataTable y en el DataBase.

## TEMAS VISTOS APENDICE

### 15: EXCEPCIONES

Declaraciones: Proyecto Excepciones

`cantidadInvalidaException()` Cuando se quiere poner una cantidad diferente a un número natural

`ArchivosException()` Cuando no se puede leer o guardar un archivo

`dniInvalidoException()` Cuando se quiere poner una cantidad diferente a un número natural entre los parámetros establecidos es validación de DNI

`insuficienteParaBoletoException()` Cuando se quiere comprar boletos y no alcanzan las monedas de bronce

### 16:TEST UNITARIOS

Se generan 3: Verifica que dos jugadores no sean iguales, Que un jugador no se pueda guardar repetido en el casino, que el DNI sea valido

### 17: TIPOS GENERICOS

### 18: INTERFACES

Se utilizan en serialización de Archivos en proyecto Archivos.

### 19: ARCHIVOS Y SERIALIZACION

XML: se guarda un casino y se lee un casino; serialización del datatable.

Texto: se guardan todas las partidas realizadas con append, en un archivo por cada jugador en una carpeta llamada Jugadas en FormBase/bin/Debug

### 21 Y 22: BASE DE DATOS Y SQL

En la carpeta de la solución se encuentran los 2 archivos log para guardarla en la base master (donde luego se ira a buscar para hacer la conexión). Se guardarán las ultimas jugadas de cada jugador. En Entidades/Acceso datos se establecen las conexiones.

**datos\_Jugadas**

### 23: HILOS

Hilo en el cual una imagen rotará. Se establece y se inicia en el form base. Mientras se ejecuta se establece el random de perdida o victoria y se arma la jugada.

### 24: EVENTOS

Se declaran en el form ruleta y manejarán la vida del hilo. El manejador de encuentra en el form principal que trabaja con el hilo.

### 25: METODOS DE EXTENSION

Entidades/extensión muestra los precios según el tipo del enumerado.

**Casino**  
Clase

- Campos
  - jugadas
  - jugadores
- Propiedades
  - Jugadas
  - Jugadores
- Métodos
  - BuscarJugador...
  - Casino
  - operator !=
  - operator + (+ 1 ...)
  - operator ==

**Jugada**  
Clase

- Campos
  - movimiento
  - varianza
  - victima
- Propiedades
  - Movimiento
  - Varianza
  - Victima
- Métodos
  - CalcularVarianza
  - Jugada (+ 2 so...)
  - Mostrar
  - ToString

**Jugador**  
Clase

- Campos
  - billetera
  - boletos
  - dni
  - saldo
- Propiedades
  - Billetera
  - Boletos
  - DNI
  - Saldo
  - StringToDNI
- Métodos
  - BuscarMoneda
  - CantidadMone...
  - Jugador (+ 3 so...)
  - Mostrar
  - operator -
  - operator !=
  - operator +
  - operator ==
  - SacarSaldo
  - ToString
  - ValidarDni (+ 1 ...)

**ETipoMoneda**  
Enum

- bronce
- plata
- oro

**ETipoTransaccion**  
Enum

- pierde
- gana
- compra

**Extension**  
Static Class

- Métodos
  - MostrarMonedas

**AccesoDatos**  
Clase

- Campos
  - cn
  - comando
- Métodos
  - AccesoDatos
  - InsertarJugada
  - ModificarJuego

**SerializacionJug...**  
Clase

- Métodos
  - Guardar
  - Leer

**SerializacionPar...**  
Clase

- Métodos
  - Guardar

**Moneda**  
Clase

- Campos
  - cantidad
  - ganancia
  - moneda
  - precio
- Propiedades
  - Cantidad
  - Ganancia
  - Moneyda
  - Precio
- Métodos
  - Equals
  - GetHashCode
  - Moneda (+ 1 s...
  - operator - (+ 1 ...)
  - operator !=
  - operator + (+ 1 ...)
  - operator ==
  - SacarGanancia
  - SacarPrecio
  - SacarSaldo
  - ToString
  - ValidarCantidad

**BoletoChances**  
Clase

- Campos
  - cantidadBoletos
- Propiedades
  - Cantidad
- Métodos
  - BoletoChances...
  - explicit operato...
  - GastoBoleto
  - operator -
  - operator +