**Ejercicio POO**

**Herencia, polimorfismo, sobreescritura, sobrecarga**

**Practica de: diseño diagrama de clases y relaciones entre entidades.**

**BIENVENIDOS AL *GON$INO* (Casino de Gonzo)**

Su programa debe crear el entorno virtual (programa en C++) para que el usuario pueda jugar en el casino. La moneda de este casino son los Gonzos. A continuación se detalla el funcionamiento del casino.

**Proceso – Ingresar al casino**

* Cuando entra al casino un usuario el sistema debe preguntar el nombre, el identificador y el dinero en pesos. El sistema valida que el dinero en pesos sea mayor a 0.
* El sistema agrega el jugador al casino y calcula automáticamente la cantidad de Gonzos a los que equivale el dinero del usuario.

**Proceso – jugar .**

1. Antes de iniciar a jugar el sistema le pregunta al jugador su identificación. El jugador debe existir en el sistema, sino existe le muestra un error al usuario
2. El sistema debe preguntar cuántos Gonzos desea apostar el usuario. El sistema debe validar que el jugador tenga la cantidad suficiente de Gonzos para apostar, si no los tiene entonces el sistema debe avisarl para que ajuste el valor de su apuesta.
3. El sistema debe preguntar al usuario cuál de los juegos desea jugar.
4. El usuario selecciona el juego y el sistema verifica que el juego exista.
5. Si el juego existe el sistema inicia el juego. Sino el sistema muestra un error que indica que no existe.
6. Una vez que el usuario termina de jugar, el sistema le mostrar si el jugador gano o perdió, cuánto Gonzos ganó o perdió y a cuánto dinero equivalen esos gonzos. Además, el sistema le debe preguntar al usuario si desea jugar de nuevo. Si el usuario desea jugar entonces el sistema continúa con el paso 2.
7. El sistema debe llevar la cuenta de cuántas veces ha jugado el jugador. Se cuenta por cada vez que juega, es decir, si por ejemplo tiró el dado 3 veces en el juego 2, entonces jugó 3 veces.

**Juego 1, El mayor de 13:**

* **Mecánica de juego:** el sistema genera un número aleatorio para el usuario entre 1 y 13. Ahí el usuario debe tener 2 opciones para continuar, la primera: es darse por perdido antes de que se genere el número que le corresponde a casino (en este caso perderá la mitad del dinero apostado). La segunda: es jugar, es decir, que se genere el número aleatorio del casino.
* **CalcularResultado: e**n caso de que el usuario saque un número menor o igual que el casino, perderá todo lo apostado, en otro caso el usuario ganará el doble de lo apostado (es decir, apostado X2).

**Juego 2, Dos colores:**

* **Mecánica de juego:** el usuario tira un dedo (de 6 caras) y escoge entre 2 colores (Blanco o Negro) la idea es coincidir con el valor del dado y el color que luego le saldrá al casino.

El sistema le pregunta al usuario el color a elegir (Blanco o negro) y el valor apostado, luego generará un color y un número aleatorio para el casino.

* **CalcularResultado:** si coincide tanto el valor del dado como el color, el usuario ganará 4 veces lo apostado. Si el usuario coincide sólo con el valor del dado ganará 0.5 veces lo apostado, si el usuario coincide sólo con el color elegido no gana ni pierde nada, si el usuario no coincide en nada, pierde todo el dinero apostado.

**Nuevo juego**

Agregue un nuevo juego al casino y pruébelo funcionando. Su nuevo juego debe heredar de la clase Juego.

**Proceso – ver Información usuario**

* El sistema pregunta el id del jugador a consultar
* Si el jugador existe el sistema presenta el identificador, nombre y cantidad de Gonzos que tiene disponibles el jugador.
* Si el jugador no existe el sistema presenta un mensaje de error

**Proceso - vender Gonzos**

* El sistema pregunta el id del jugador que quiere recargar
* Si el jugador existe el sistema solicita el dinero a recargar
* El sistema valida que el valor a recargar sea mayor a 0.
* El sistema actualiza los Gonzos del jugador.
* Si el jugador no existe el sistema presenta un mensaje de error

**Proceso – retirar usuario**

* El sistema pregunta que usuario desea retirar
* El sistema recibe el id del usuario
* El sistema valida si el usuario existe, si existe elimina al jugador del casino. Si el jugador no existe el sistema muestra un error.
* El sistema muestra la información del usuario a retirar.

**Datos específicos:**

**Conversión de pesos a Gonzos**

10 000 pesos equivalen a 100 gonzos.

.

**Jugador**

Tiene una identificación, un nombre, saldo en Gonzos

**El casino**

Un casino tiene jugadores y juegos disponibles.

* El casino calcula a cuantos Gonzos equivale cierta cantidad de pesos y también es capaz de calcular a cuántos pesos equivalen cierta cantidad de Gonzos.

**Requerimientos no funcionales**

* Debe usar un mapa para guardar los usuarios
* Su programa debe tener manejo de excepciones.
* Su programa debe tener menú para acceder a las funcionalidades.
* Su programa se debe ajustar al diseño definido por el arquitecto. A fin de acelerar el desarrollo el arquitecto provee el repositorio con los archivos ya definidos y algunas funcionalidades implementadas. Es su labor terminar el desarrollo para cubrir los requisitos solicitados. Para correr este proyecto se recomienda tener Clion instalado, sino usa Clion debe hacer su makefile.

**Pasos**

* Explore el diagrama de clases, los requerimientos y el código que le fue entregado. Luego inicie la implementación.

Graphical user interface, application

Description automatically generated

