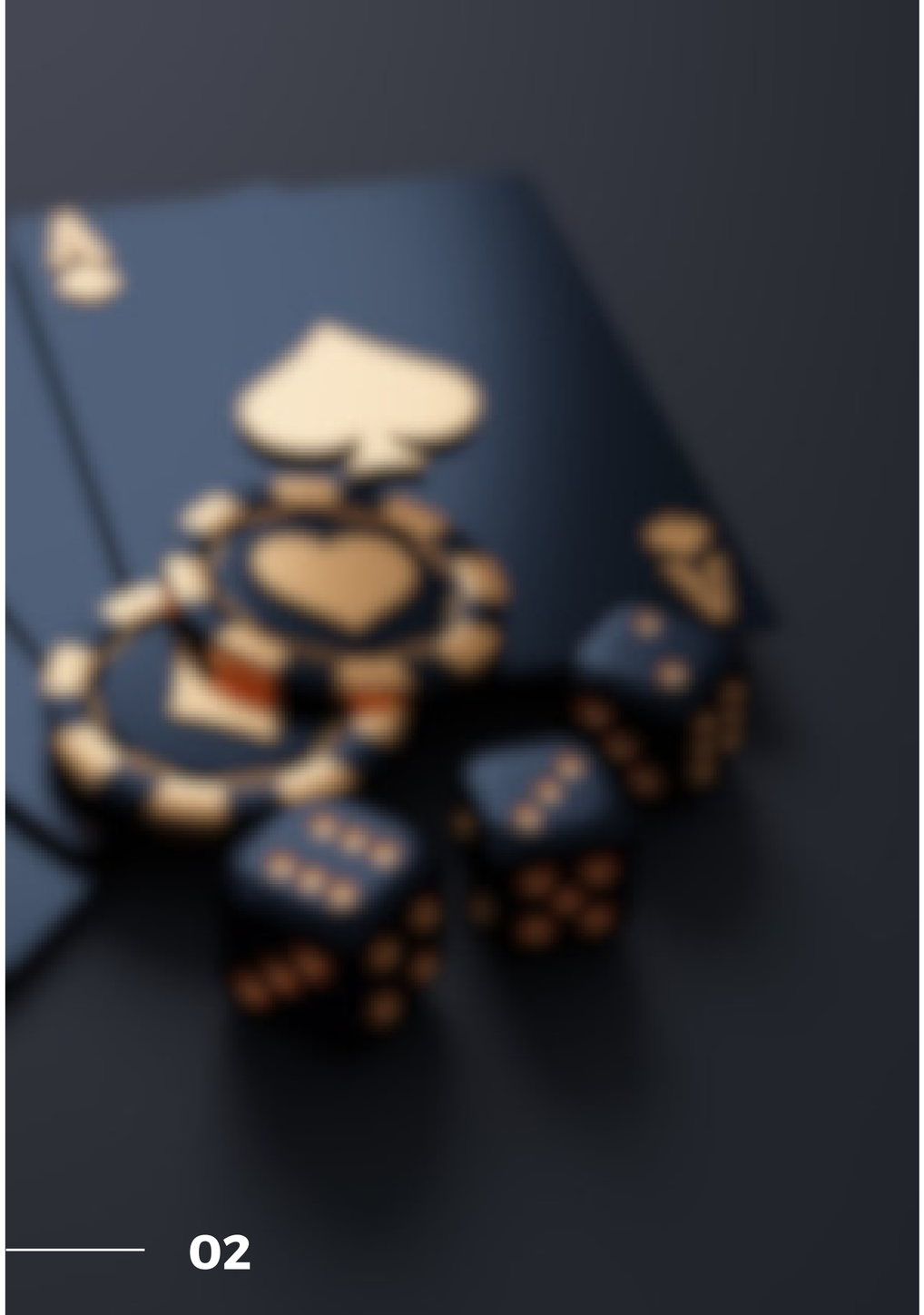
PROYECTO CASINO

Versión 1.0



EQUIPO DE DESARROLLO

Sonia Altamiranda

Desarrollador

Luz Niz

Desarrollador

Hugo Viqueira

Desarrollador

METODOLOGÍA DE TRABAJO

REPOSITORIO COLABORATIVO

Se usó Git Hub para crear un repositorio colaborativo, donde cada desarrollador subia su código a su branch, a través de GIT, donde después se realizó un merge con la branch main.

TRELLO

Utilización de Trello para la creación de un tablero, para asignar / organizar tareas y ver avances de cada desarrollador

MEETS

Utiización de la plataforma meet para realizar reuniones con el objeto de la creación la arquitectura, asignación de tareas y muestra de avances

SLACK

Cómo método de comunicación para consultar sobre el código, pedir asistencia o consultar sobre dudas.

ENTORNO DE DESARROLLO

IDE

Visual Studio Code

LENGUAJE

TypeScript - JavaScript

ENTORNO DE EJECUCIÓN

Node.js

GESTOR DE PAQUETES

npm

PAQUETES INSTALADOS

- readlineSync
- clear
- colorette

SISTEMA DE CONTROL DE VERSIONES

GIT

REPOSITORIO REMOTO

GIT Hub

CASO DE USO

OBJETIVO

Jugar juego de casino

ACTORES

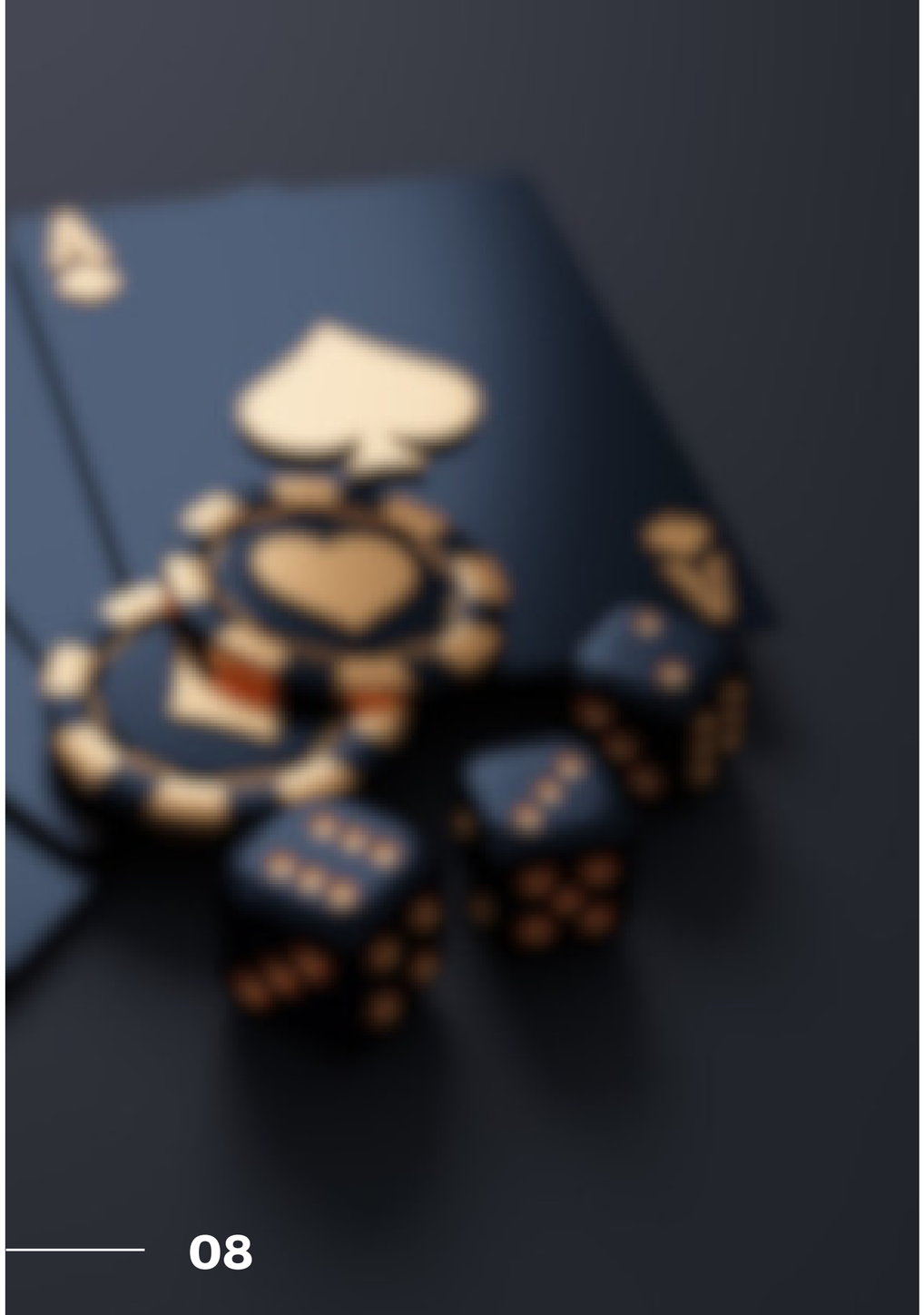
Jugador

DESCRIPCIÓN

El usuario desea jugar a los juegos de un casino en un programa por consola

PRECONDICIONES

El usuario debe ser mayor de la edad impuesta por el casino como edad minima para ingresar a jugar.
Debe ingresar un monto de dinero para apostar



FLUJO BÁSICO

1- El usuario ingresa su edad. Si la edad es mayor o igual a la admitida, puede continuar;

2- Ingresa su nombre;

4- El casino le da la bienvenida;

5- Ingresa el monto de dinero con el que quiere jugar;

6- El casino le muestra las opciones de juego;

7- El usuario elige el juego con el que desea jugar;

8- El usuario ingresa la cantidad de dinero que desea apostar;

9- El casino muestra si el jugador ganó o perdió. Deposita o debita el dinero según el caso;

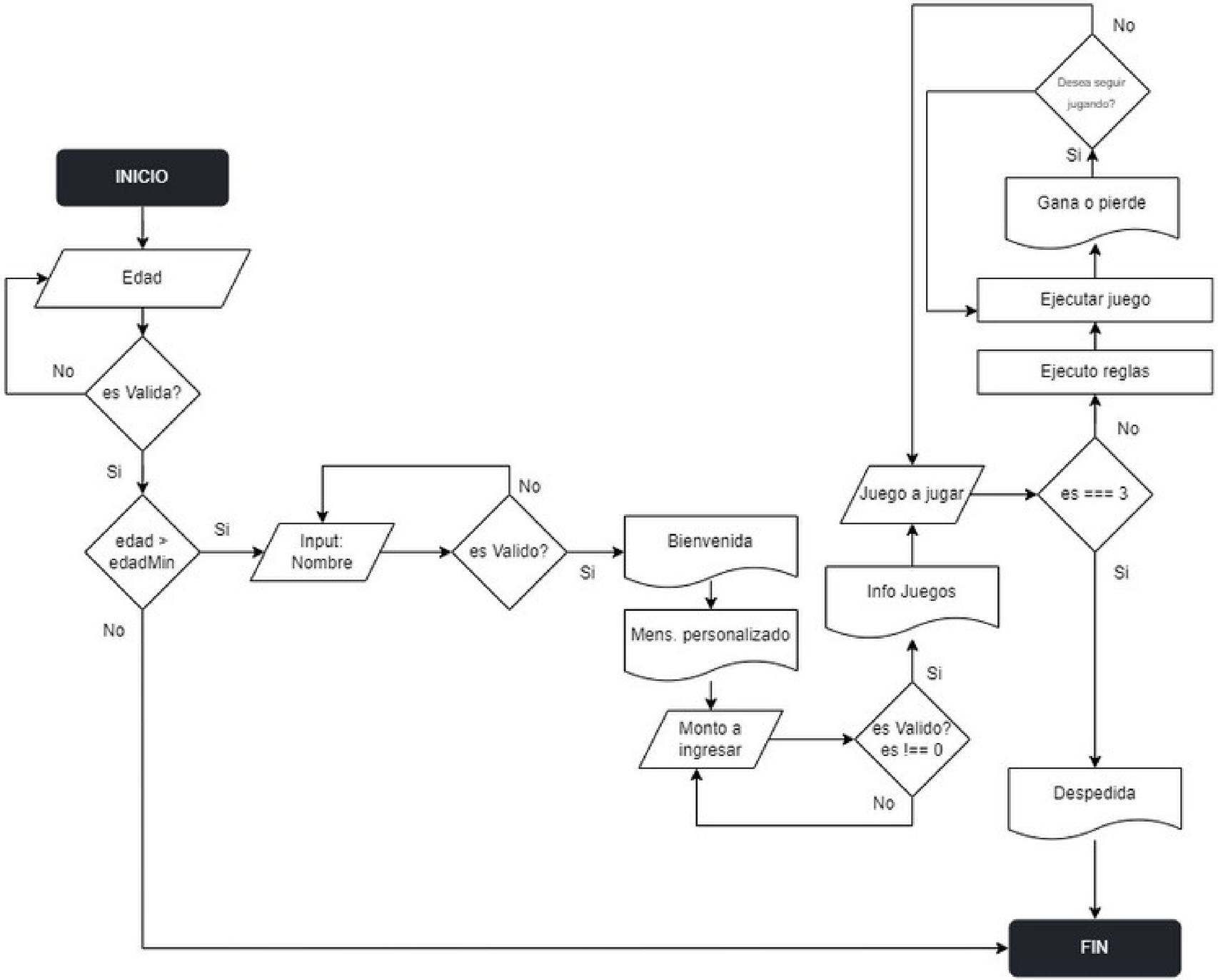
10- El casino pregunta si quiere seguir jugando al mismo juego;

11- El casino pregunta si quiere cambiar de juego;

12- El jugador cobra su premio;

13- El jugador sale del programa;

14- El casino despide al jugador;

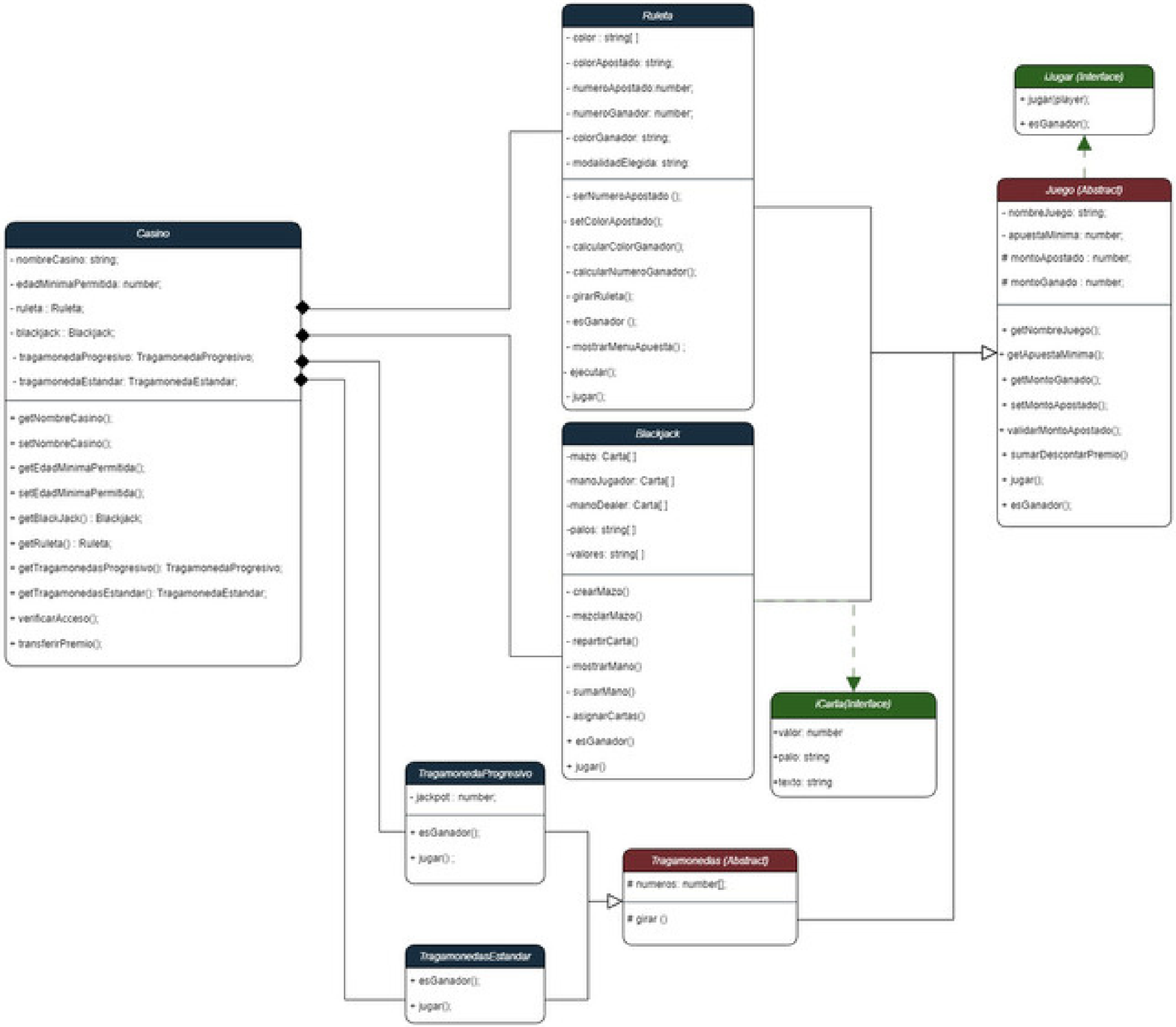

INICIO
Edad
No
es Valida?
Si
edad > edadMin
Si
Input: Nombre
No
es Valido?
Si
Bienvenida
Mens. personalizado
Monto a ingresar
es Valido? es !== 0
No
Si
Info Juegos
Juego a jugar
es === 3
No
Si
Ejecuto reglas
Ejecutar juego
Gana o pierde
Si
Desea seguir jugando?
No
Despedida
FIN

ARQUITECTURA

Diseño del programa

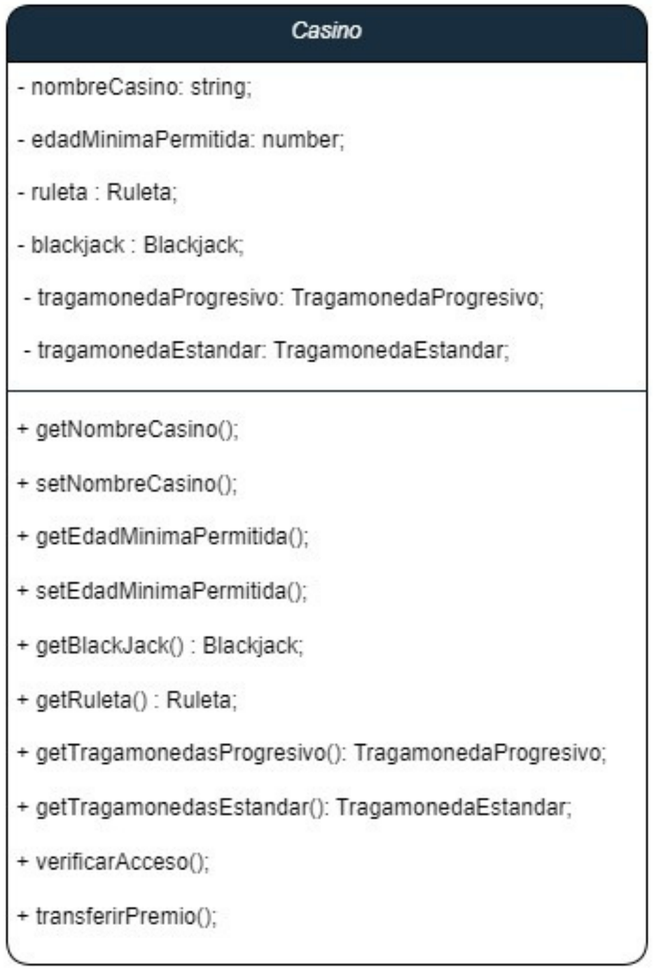
PROYECTO NODE.JS 2023

UML



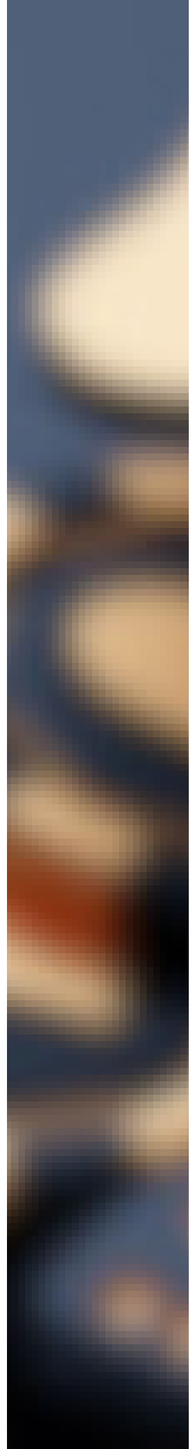
CLASE CASINO

Es la clase que crea el casino, esta compuesta por los juegos que se pueden jugar en el casino.



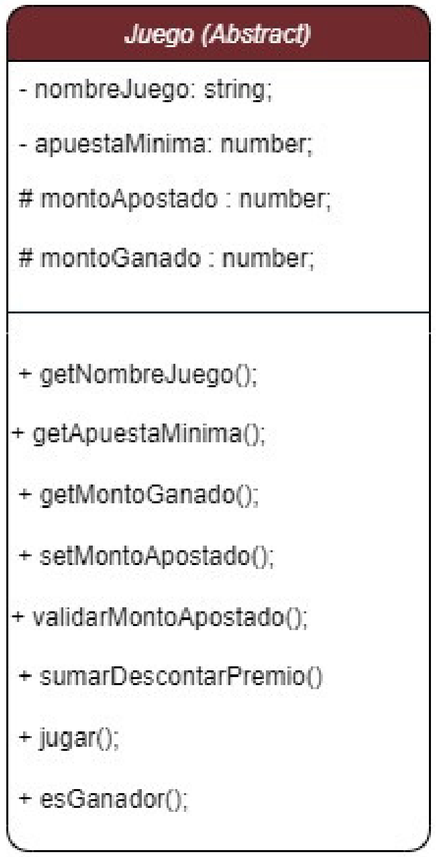
MÉTODOS

+getNombreCasino(): obtiene el nombre del casino.
+setNombreCasino(): cambia el nombre del casino.
+getEdadMinimaPermitida(): obtiene la edad minima permitida por el casino.
+setEdadMinimaPermitida(): cambia la edad minima permitida por el casino.
+getBlackJack(): Retorna el juego balckjack
+getRuleta(): Retorna el juego ruleta
+getTragamonedasProgresivo(): Retorna el juego tragamonedas de tipo progresivo
+getTragamonedasEstandar(): Retorna el juego tragamonedas de tipo estandar
+validarAcceso(): verifica si el jugador tiene la edad minima para jugar.
+transferirPremio(): Transfiere el dinero al jugador.



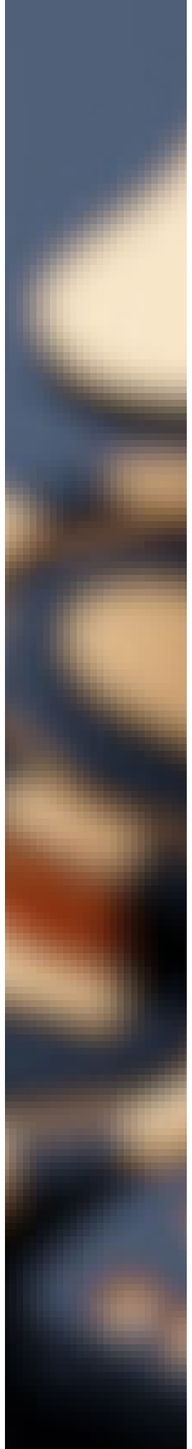
CLASE JUEGO

Es la clase PADRE de los juegos, es abtracta e implementa la interfaz iJugar



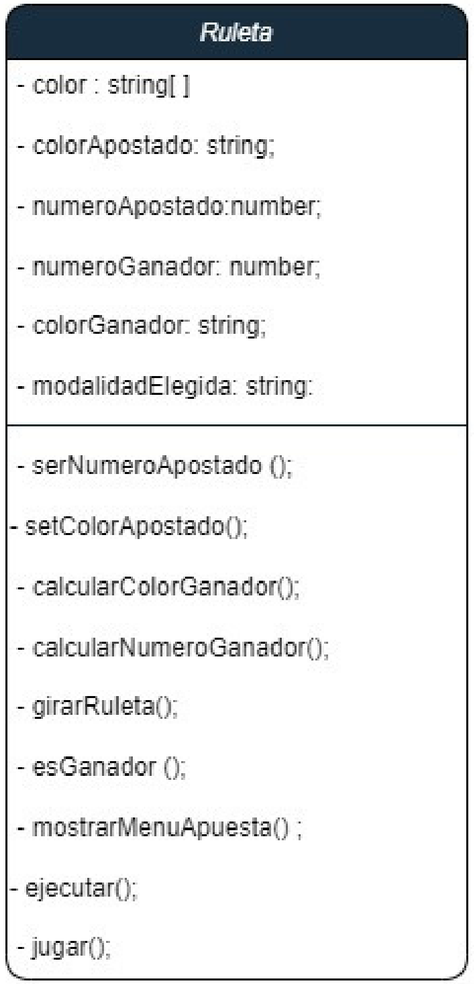
MÉTODOS

+getNombreJuego(): obtiene el nombre del juego.
+getApuestaMinima(): obtiene la apuesta minima del juego.
+getMontoGanado(): obtiene el monto ganado por el jugador
+setMontoApostado(): Ingresa el monto apostado por el jugador.
+validarMontoApostado(): valida el monto apostado por el jugador para que no pueda realizar una apuesta invalida.
+sumarDescontarPremio: Suma o descuenta del monto ganado, según el resultado del juego.
+jugar(): Implementa metodo de la interfaz, que será sobreescrito por las clases hijas.
+esGanador(): Implementa metodo de la interfaz, que será sobreescrito por las clases hijas.



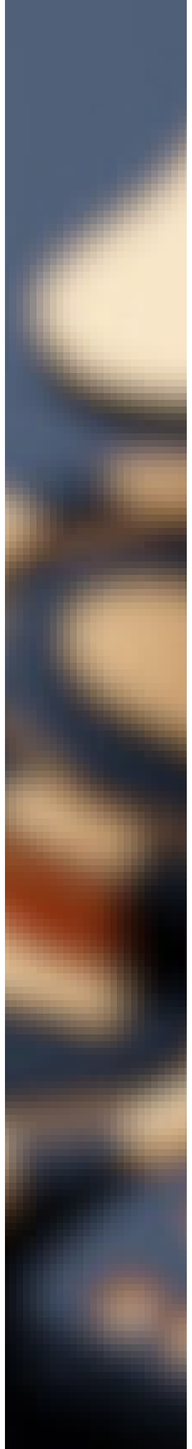
CLASE RULETA

Es la clase que permite jugar a la Ruleta. Esta clase HEREDA de la CLASE JUEGO



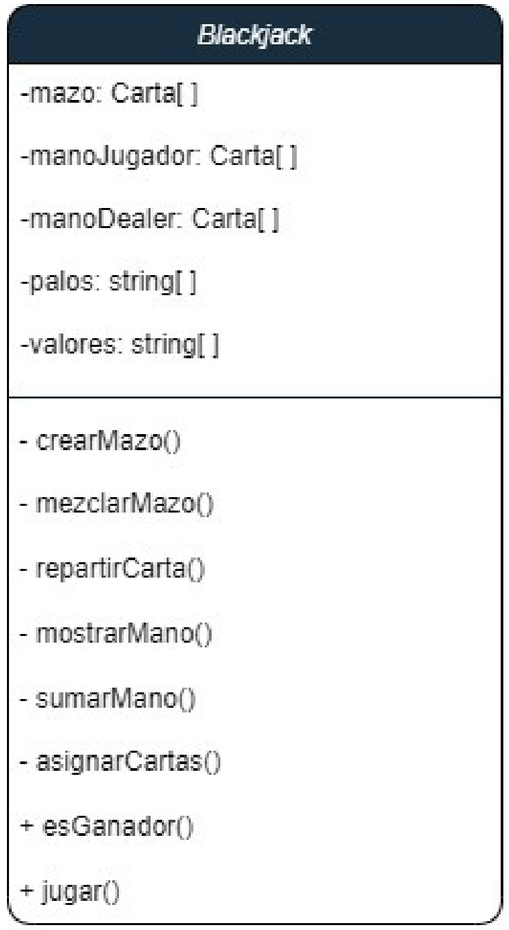
MÉTODOS

+setNumeroApostado(): Ingresa el numero apostado por el jugador.
+setColorApostado(): Ingresa el color apostado por el jugador.
+calcularColorGanador(): calcula el valor ganador y lo devuelve.
+calcularNumeroGanador(): calcula el numero ganador y lo devuelve.
+girarRuleta(): según la apuesta modalidad de apuesta del juegador asigna y calcula el numero / color ganardor.
+mostrarMenuApuesta(): Le muestra el menu al jugador para que seleccione la modalidad de apuesta.
+ejecutar(): Ejecuta la seleccion de menu.
+esGanador(): metodo heredado de Juego (a su vez implementado de iJugar). Que sobreescribe según las reglas de su juego.
+jugar(): ejecuta el juego. Metodo heredado de Juego (a su vez implementado de iJugar). Que sobreescribe según su propia ejecución.



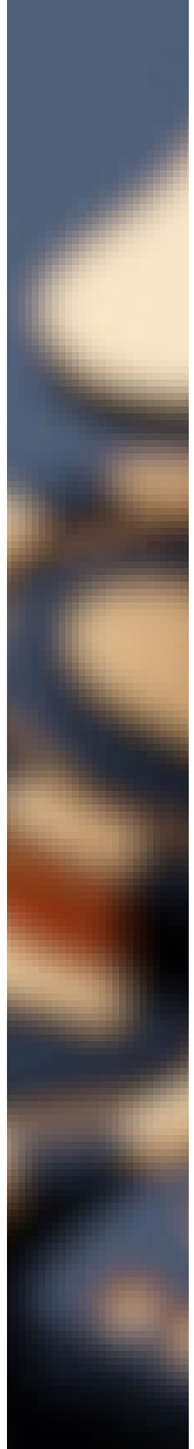
CLASE BLACKJACK

Es la clase que permite jugar a al Blackjack. Esta clase HEREDA de la CLASE JUEGO



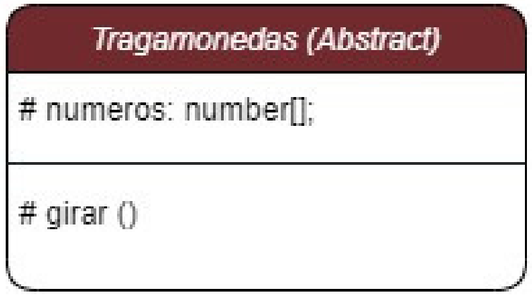
MÉTODOS

+crearMazo(): Crea el mazo de la partida.
+mezclarMazo(): Mezcla el mazo creado en la partida.
+repartirCarta(): Devuelve una carta.
+mostarMano(): muestra la mano del jugador o el dealer.
+sumarMano(): suma el valor de las cartas.
+asignarCartas(): Reparte 2 cartas al Jugador y 2 al casino.
+esGanador(): metodo heredado de Juego (a su vez implementado de iJugar). Que sobreescribe según las reglas de su juego.
+jugar(): ejecuta el juego. Metodo heredado de Juego (a su vez implementado de iJugar). Que sobreescribe según su propia ejecución.



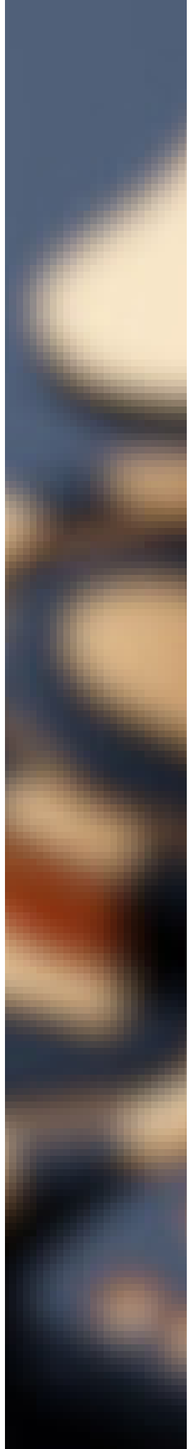
CLASE TRAGAMONEDAS

Es la clase PADRE de las tragamonedas. Es una clase ABSTRACTA. Hereda de la clase JUEGO



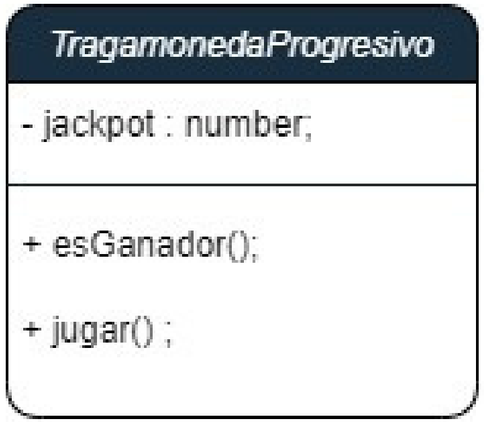
MÉTODOS

#girar(): Gira los tambores y asigna los numeros al array.



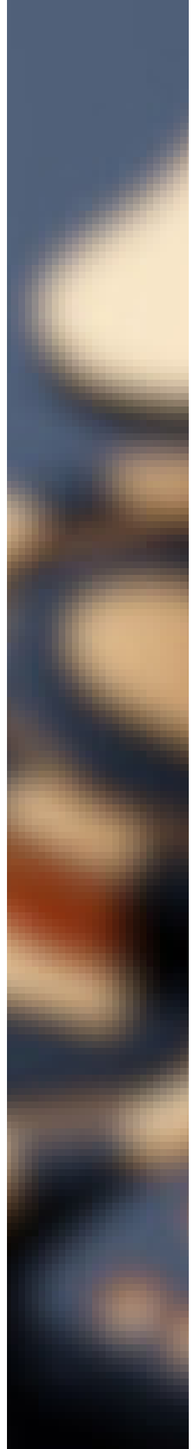
CLASE TRAGAMONEDAS PROGRESIVO

Es la clase HIJA de la clase tragamonedas.



MÉTODOS

+esGanador(): metodo heredado de Juego (a su vez implementado de iJugar). Que sobreescribe según las reglas de su juego.
+jugar(): ejecuta el juego. Metodo heredado de Juego (a su vez implementado de iJugar). Que sobreescribe según su propia ejecución.

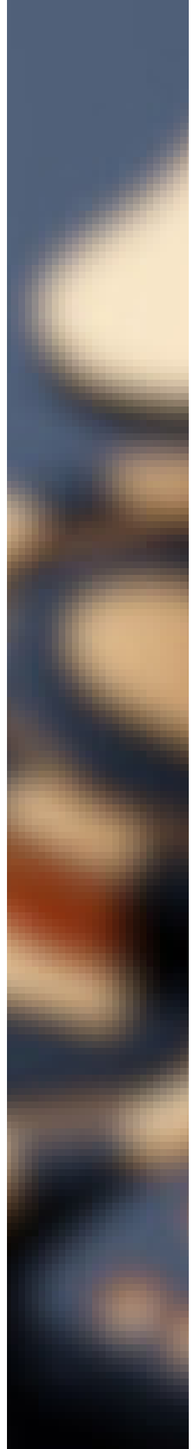


CLASE TRAGAMONEDAS ESTANDAR

Es la clase HIJA de la clase tragamonedas.

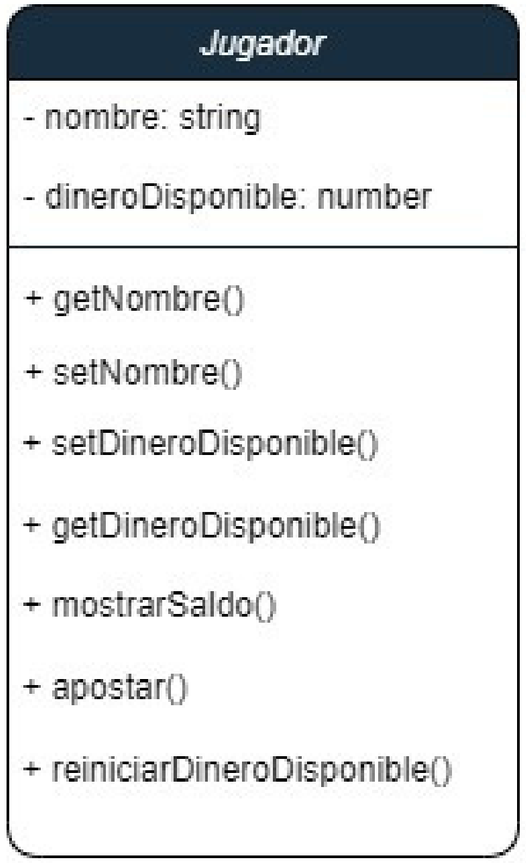
MÉTODOS

+esGanador(): metodo heredado de Juego (a su vez implementado de iJugar). Que sobreescribe según las reglas de su juego.
+jugar(): ejecuta el juego. Metodo heredado de Juego (a su vez implementado de iJugar). Que sobreescribe según su propia ejecución.



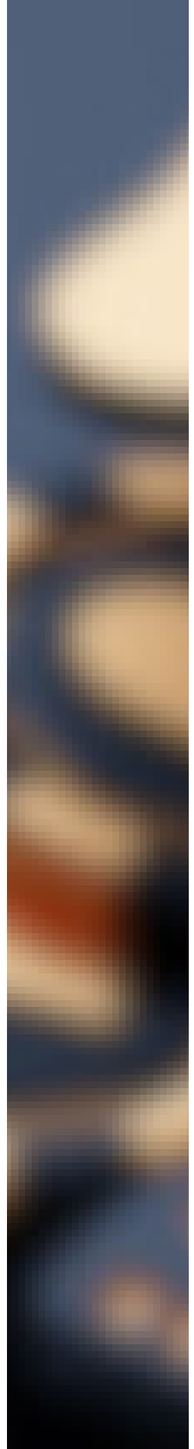
CLASE JUGADOR

Es la clase que permite guardar los datos del jugador



MÉTODOS

+getNombre(): obtiene el nombre del jugador.
+setNombre(): cambia el nombre del jugador.
+setDineroDisponible(): carga dinero en la cuenta del jugador;
+getDineroDisponible(): obtiene el dinero que el jugador tiene disponible en su cuenta.
+mostrarSaldo(): muestra el saldo del jugador.
+apostar: le pregunta el usuario el monto que desea apostar, hace la primera validación (que no pueda apostar $0).
+reiniciarDineroDisponible(): Reinicia el dinero disponible del jugador a $0.



INTERFACE IJUGAR

MÉTODOS

+jugar(): cada juego la implementará como necesite para ejecutarse.

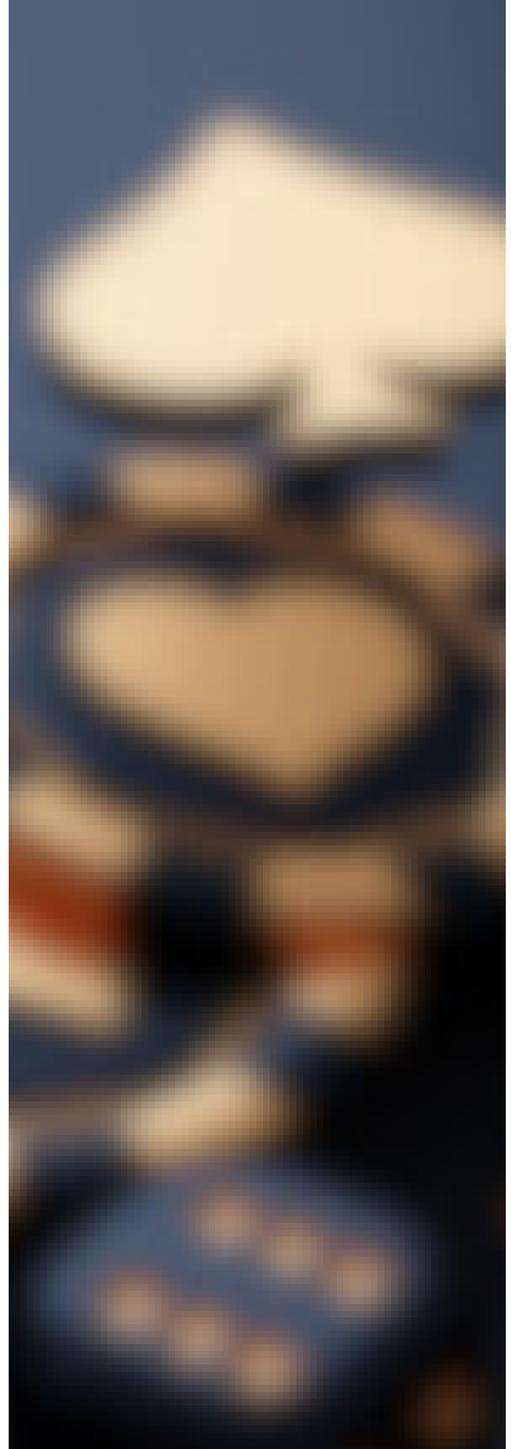
+esGanador(): cada juego la implementará como necesite para verificar si el jugador gano en base a sus propias reglas.

Es la interface que contiente la declaración de 2 métodos que serán usuados por las clases que lo implementen

AVANCES Y MEJORAS PLANIFICADAS

MEJORAS PARA PROXIMAS VERSIONES

- Versiones VIP de los juegos;
- Optimizar algunas funciones del programa;
- Funciones para la clase tragamonedas con métodos de Array;
- Premios especiales;
- Regalo de bienvenida si el usuario ingresa por primera vez;



PROYECTO NODE.JS 2023