

# Second Project Dominoes

*Proyecto de Programación II. Facultad de Matemática y Computación. Universidad de La Habana. Curso 2021.*

Dominoes es una aplicación que permite jugar Dominoes contra una serie de "inteligencias" lo **original** de esta aplicación es que hemos tratado de crear código cumpliendo con los principios SOLID de manera que implementar una nueva variante de domino o una nueva inteligencia requiera de la menor cantidad de cambios posibles, volviéndolo extensible.

Es una aplicación de Consola, desarrollada con tecnología .NET Core 6.0 y en el lenguaje C#.

La aplicación está dividida en dos componentes fundamentales:

- VisualDominoes es una aplicación de consola que renderiza la interfaz gráfica y sirve los resultados.
- DominoEngine es una biblioteca de clases donde está implementada lo necesario para la lógica del juego.

## Variaciones

Podemos variar las condiciones para ganar, la formas de finalizar el juego, Como se desarrollan las lógicas de un juego esto incluye como transcurre un turno como reparten las fichas las estrategias o formas, en las que juegan los players, la forma en la que se generan las fichas del juego, la forma en la que se reparten las fichas a cada jugador, entre otros aspectos

## Implementaciones de tipos de domino

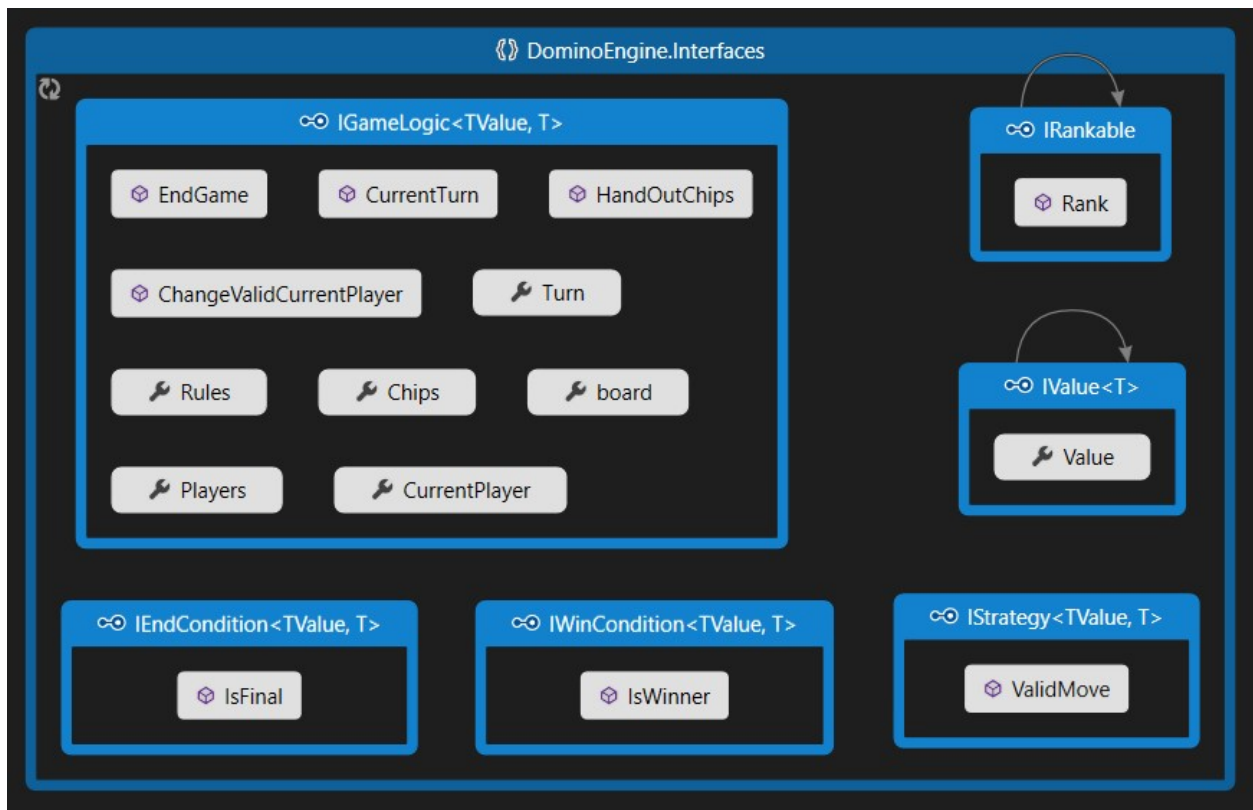
Implementamos tres variantes de domino dos con lógicas clásicas y uno con una lógica distinta las lógicas clásicas son aquellas en las que los juegos tienen un comportamiento ordinario como el domino clásico y la lógica distinta una lógica creada especialmente para jugar al robaito esto debido a que en este juego los turnos se desarrollan de formas diferentes o sea no terminan hasta que el jugador no para de robar,

## Implementación de estrategias

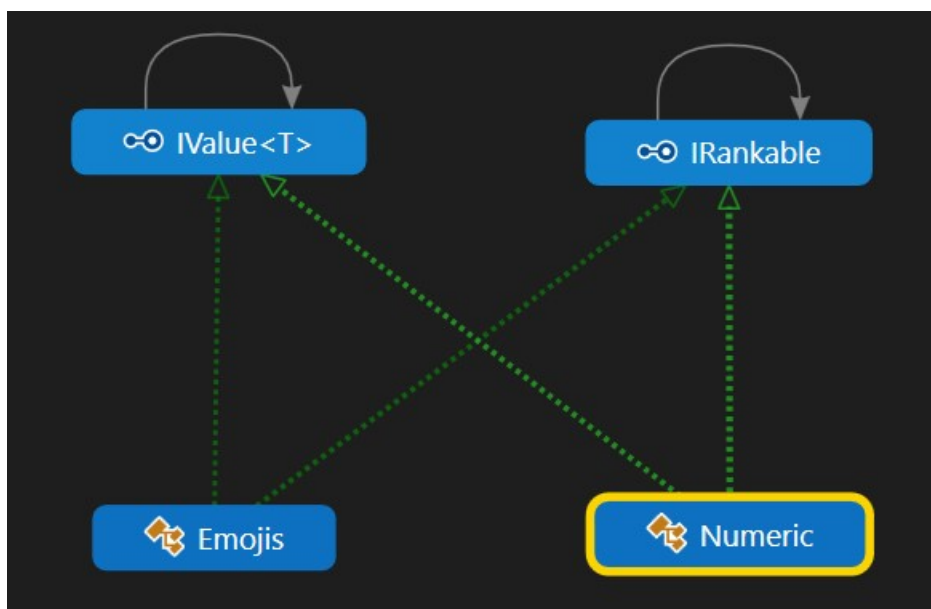
Implementamos una forma de juego para el usuario, un jugador que devuelve la ficha que más ranking tiene, uno que juega de forma aleatoria y otro que trata de buscar una forma que creemos que puede ser optima

## Sobre la Implementación

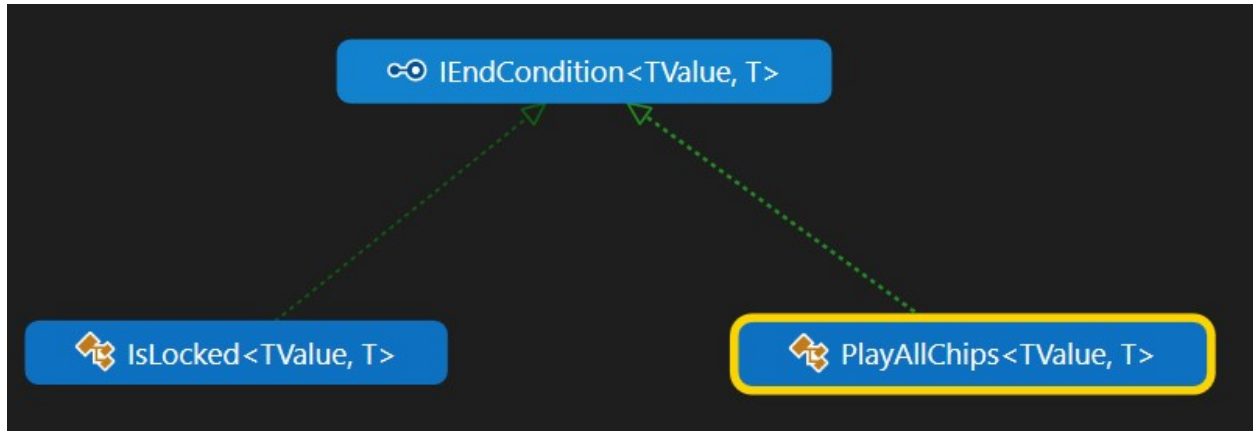
Interfaces implementadas para modelar los aspectos variables del juego



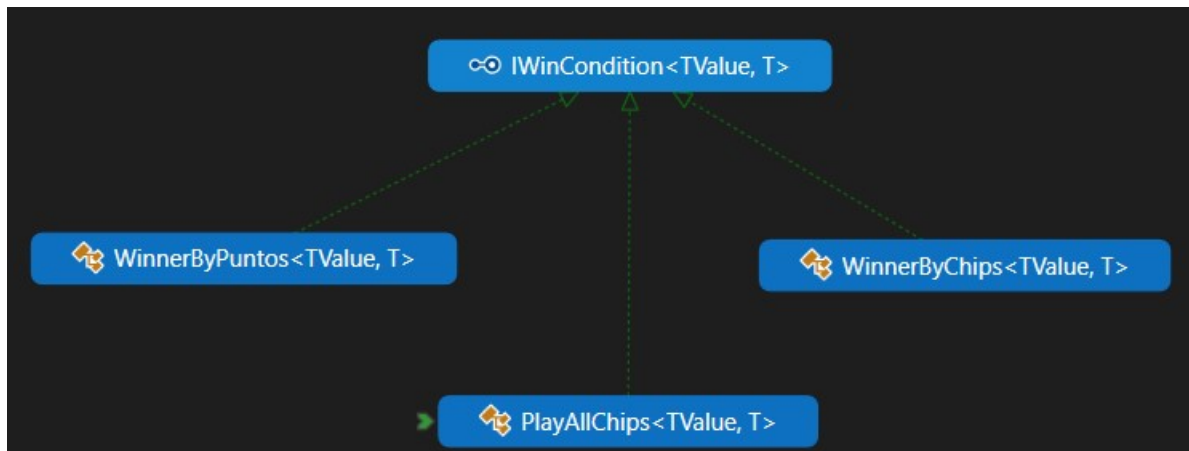
IValue e IRankable fueron creadas con el objetivo de construir un objeto que formara parte de las fichas constituye una de las caras de esta. Hacer variar, cuando variamos el valor podemos tener fichas de figuras 🎲, colores 🔴, números 1 etc...



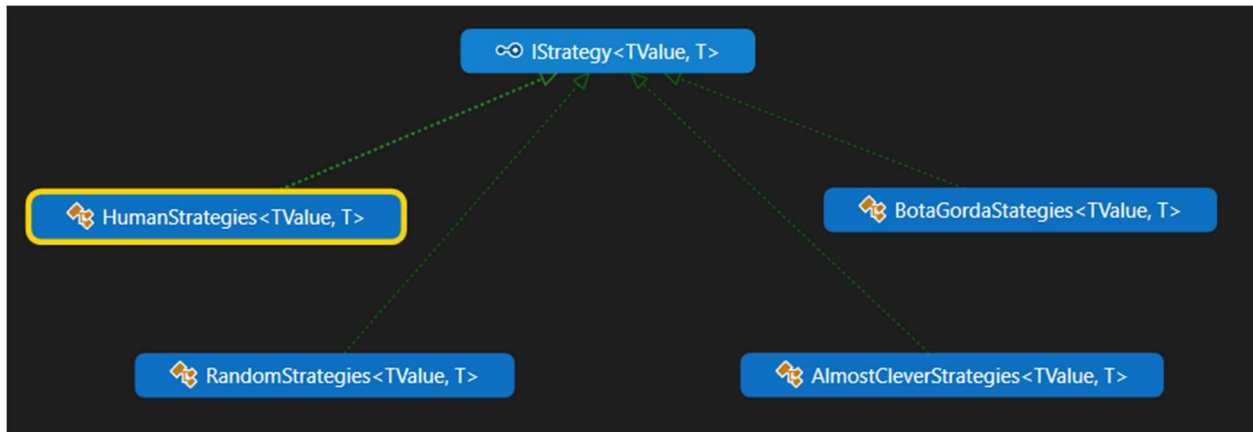
Implementamos dos clases que heredan de IEndConditions una es el criterio de finalización todos los jugadores pasados o juego trancado y la otra es todas las fichas jugadas esto nos permitió modelar las reglas de los juegos



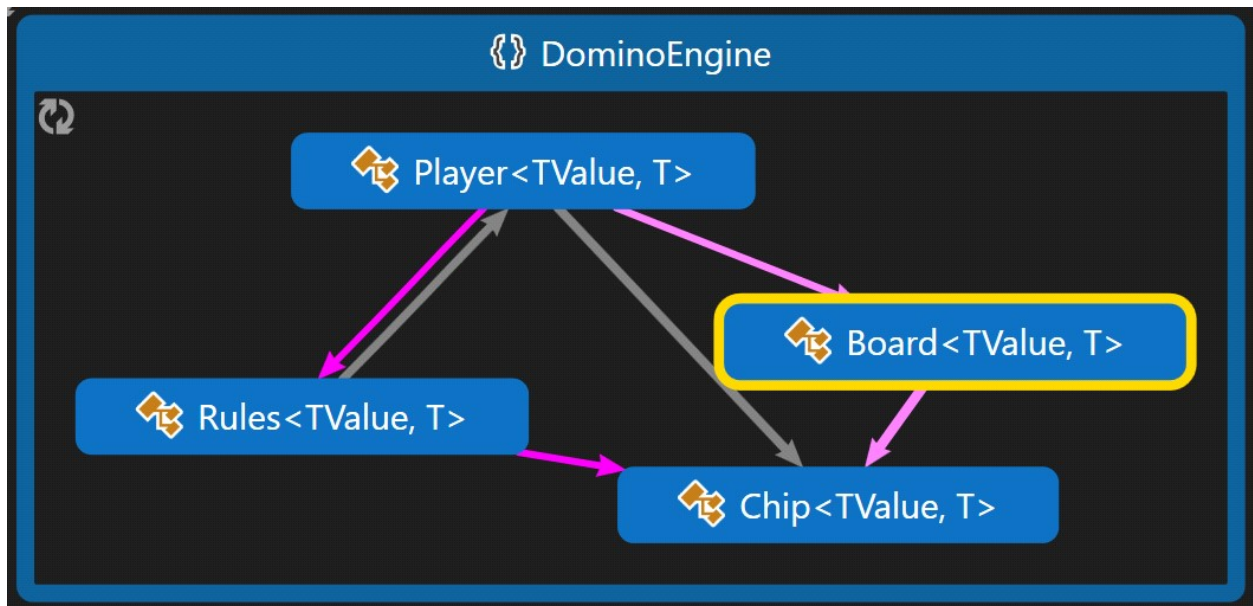
Implementamos clases que heredan de IWinConditions una es el criterio de ganador por puntos ganador por cantidad de fichas y ganador por todas jugadas esto nos permitió mas libertades a la hora de conformar las reglas del juego



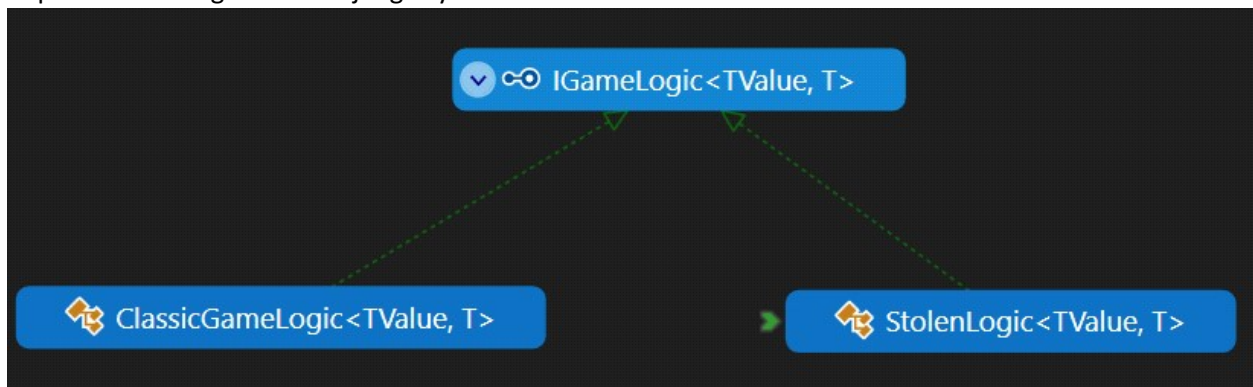
Las clases que heredan de IStrategy nos da una jugada dependiendo de criterios y formas de jugar al dominó



Estas clases no heredan de otras puesto que no nos vimos en la necesidad de variar estas estructuras en su totalidad



Representa las lógicas de los juegos y son modeladas usando las clases anteriores



## **Posibles mejoras a deficiencias**

- Como posibles mejoras esta la optimización e implementación de más variantes de juegos.
- Reajustar el funcionamiento y aumentar la abstracción para implementar juegos más potentes.
- Implementar un jugador de domino más inteligente que los actuales.
- Implementar una interfaz gráfica.