

LOS FÓSILES TRES EN RAYA

Integrantes:

- Hermes Alberto Delgado Diaz
- Kevin Steve Quezada Ordoñez
- Emiliano Sebastian Paredes Gonzalez

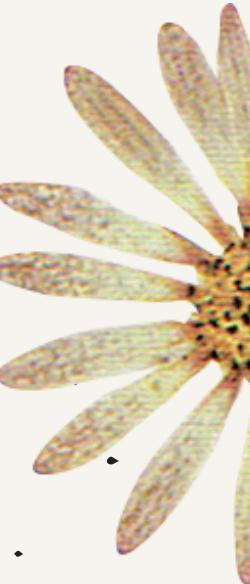


Problemática

La Secretaría de Educación Pública(SEP) quiere crear un juego para alumnos de educación básica. Su intención es que este juego ayude a mejorar cognitivamente a los alumnos de toda la república.

Para esto, la SEP lanzó una convocatoria donde puedes lanzar propuestas para el juego, además de participar en el desarrollo de este.

El equipo fósiles pensó en un juego que se llama Tres en raya.



Beneficios

El juego del tres en raya, o tic-tac-toe, es un juego simple pero que puede tener beneficios cognitivos, especialmente para los niños y adolescentes en etapa de desarrollo. Algunas formas en las que puede ayudar cognitivamente son:

- **Desarrollo de habilidades de resolución de problemas.**
- **Mejora de la concentración y la atención.**
- **Aprendizaje de conceptos de estrategia.**
- **Estimulación del pensamiento lógico.**



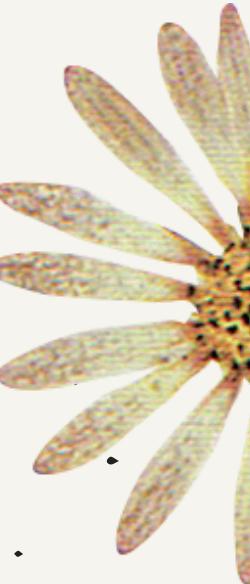
Patrones de Diseño utilizados

Los patrones de diseño utilizados son:

- **Modelo-Vista-Controlador.**
- **Observer**
- **Strategy**
- **Template Method**
- **Command**

DESIGN PATTERNS

DESIGN PATTERNS EVERYWHERE

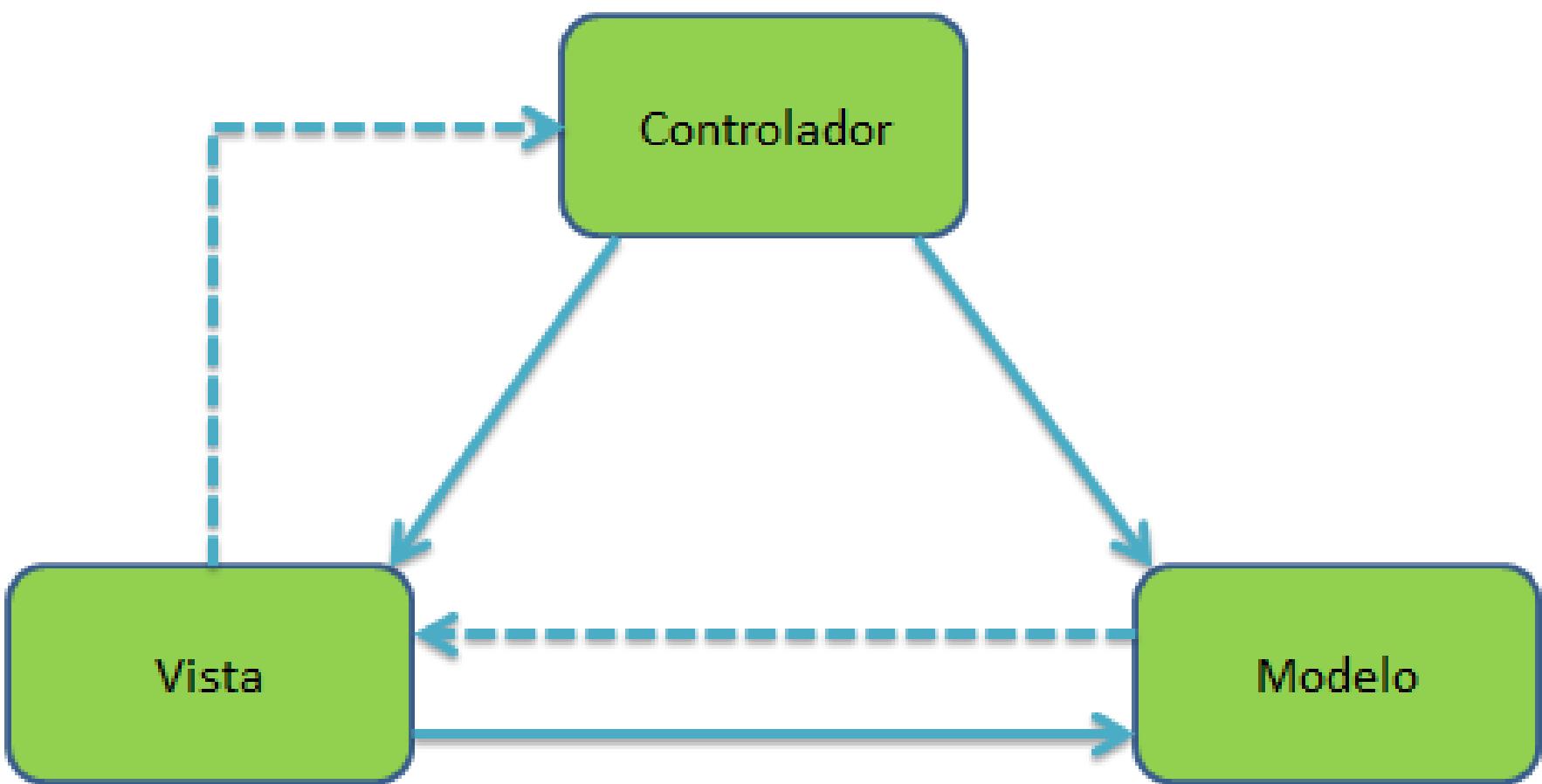


MVC

La implementación de modelo se utilizo en las clases Tablero, Tablero1v1 y Tablero1vPC.

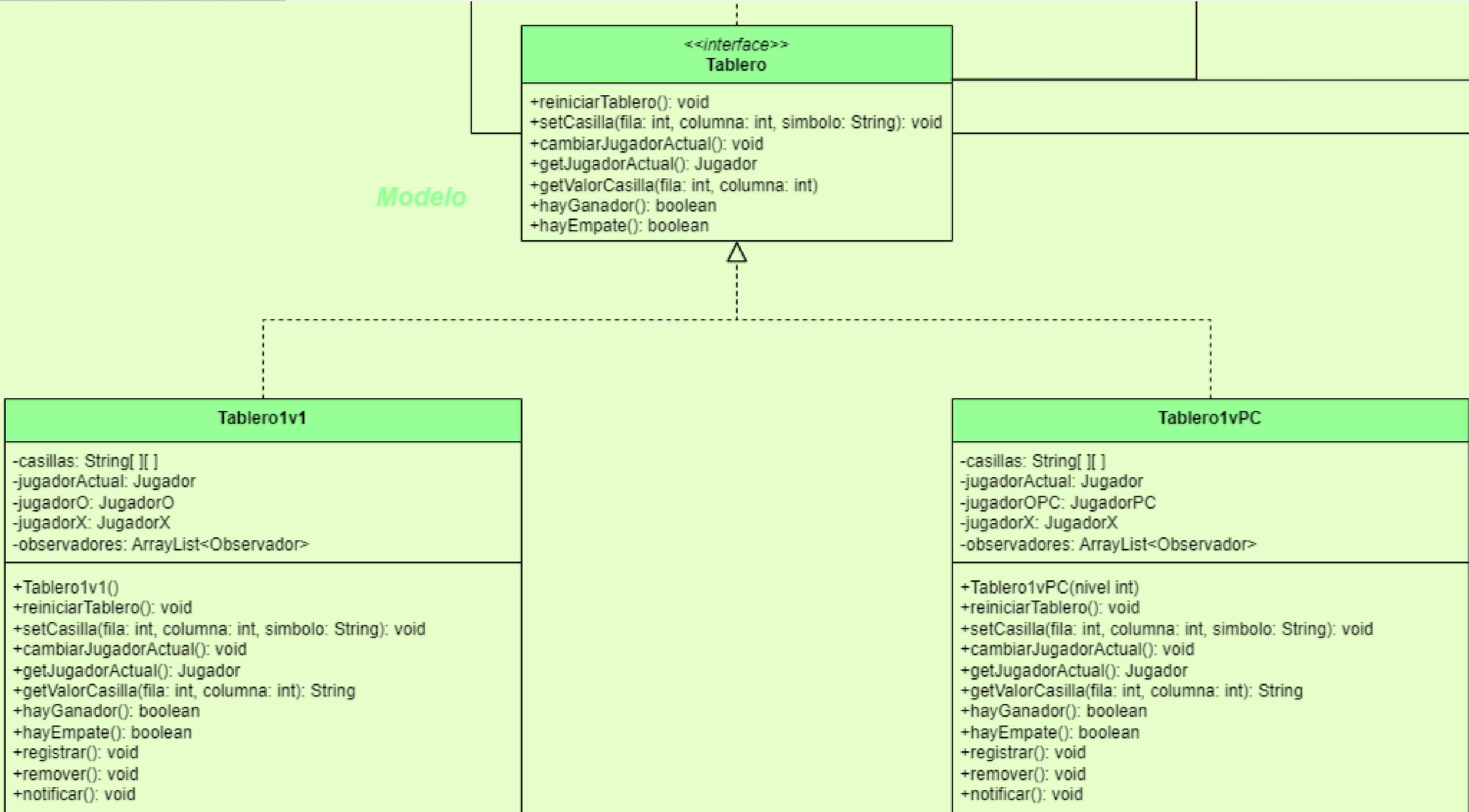
La implementación de vista se utilizo en las clases InicioVista, TableroVista y Tablero VistaPC.

La implemetación de controlador se utilizo en las clases ControladorInicio, Controlador1v1 y Controlador1vPC.

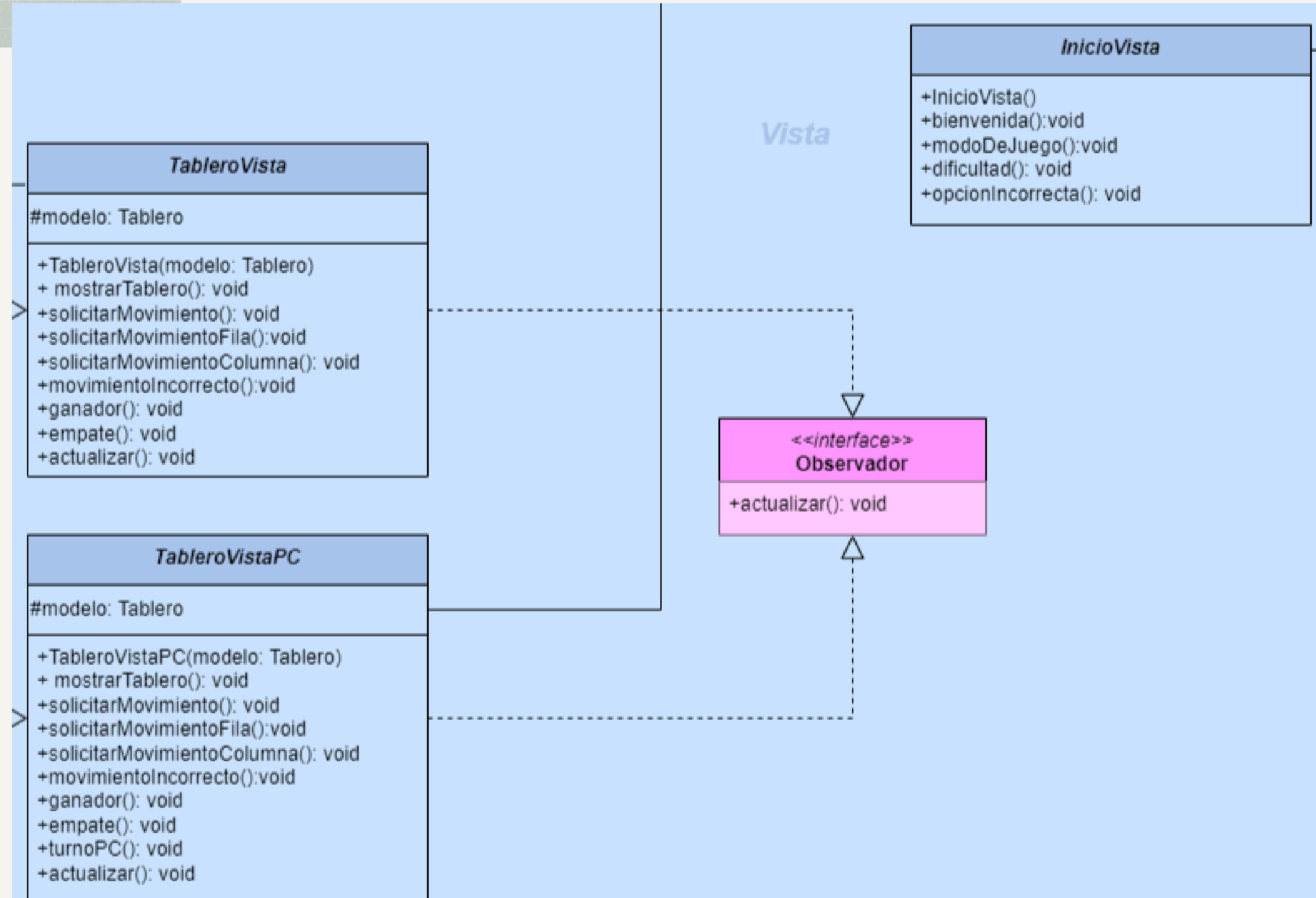


Modelo

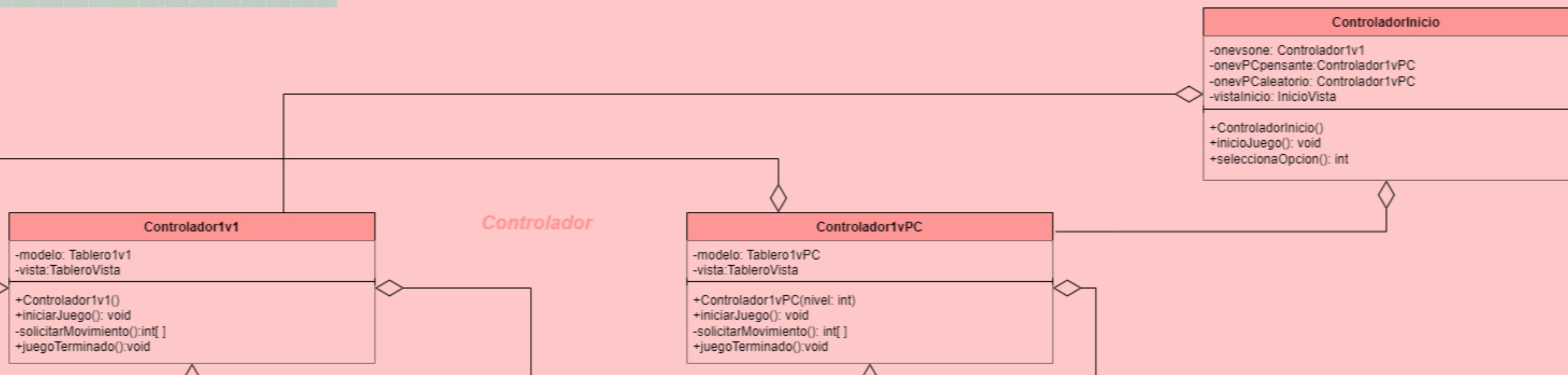
Modelo



Vista

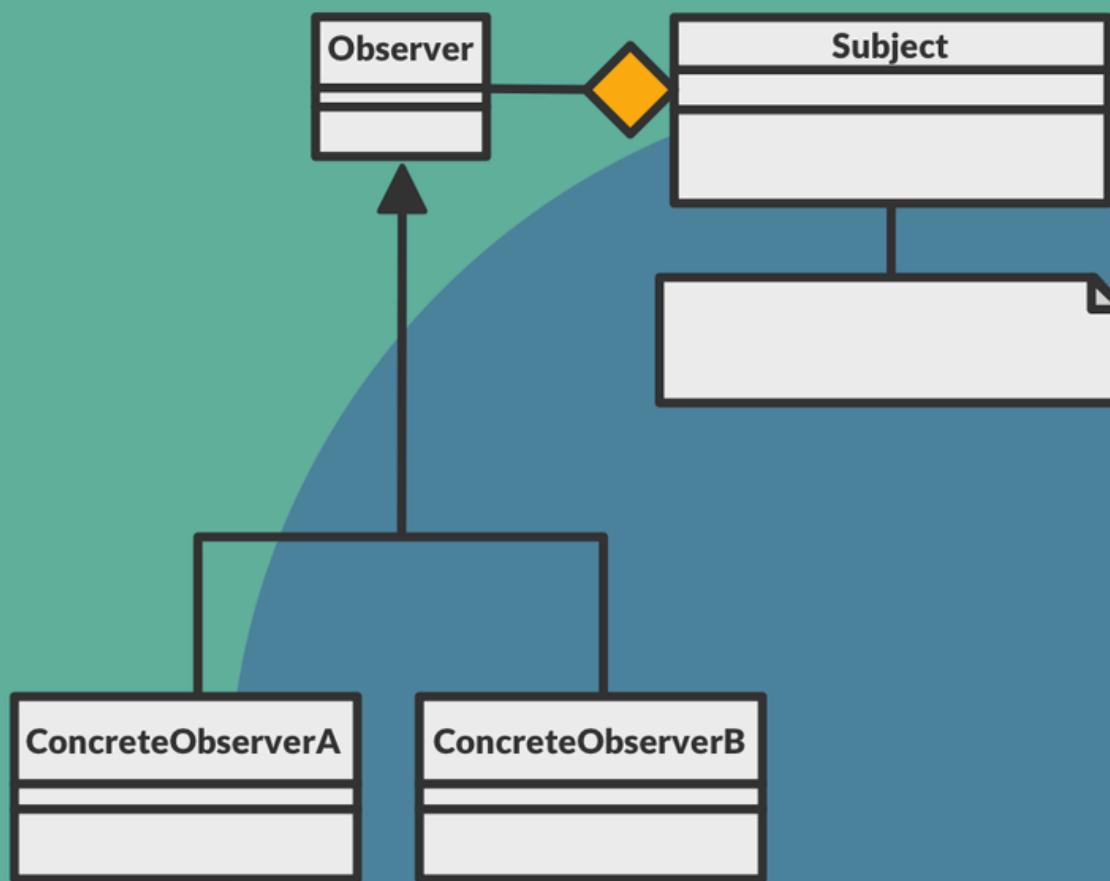


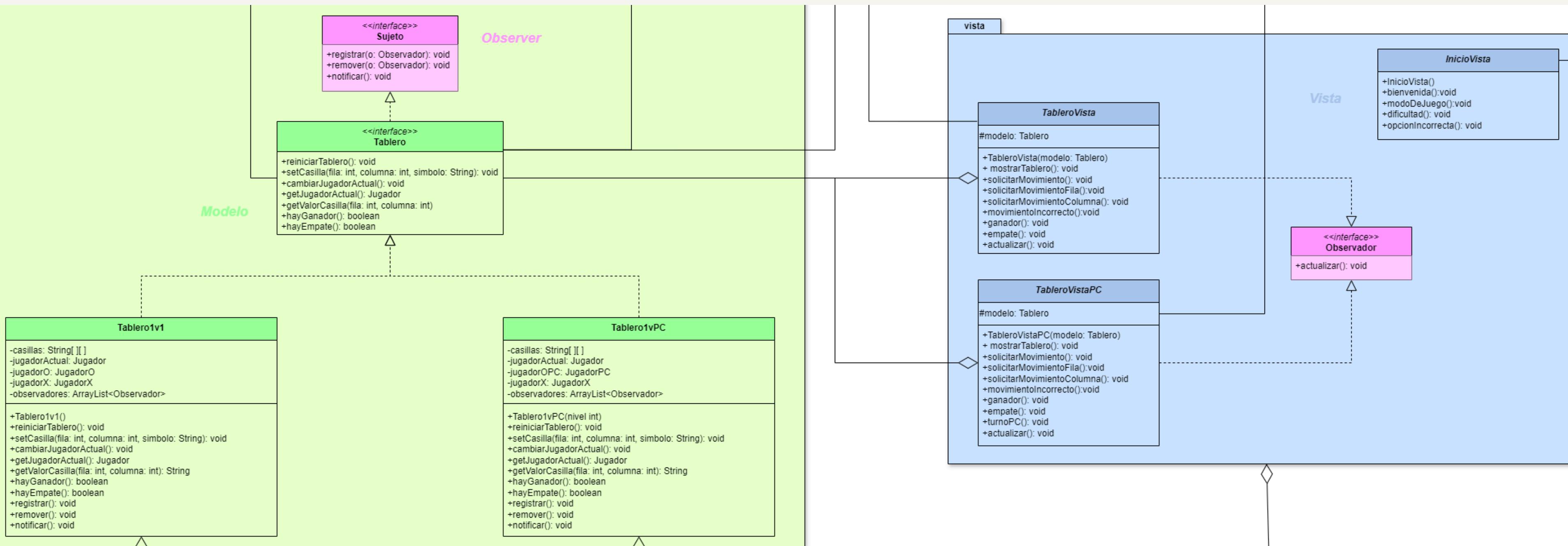
Controlador



Observer

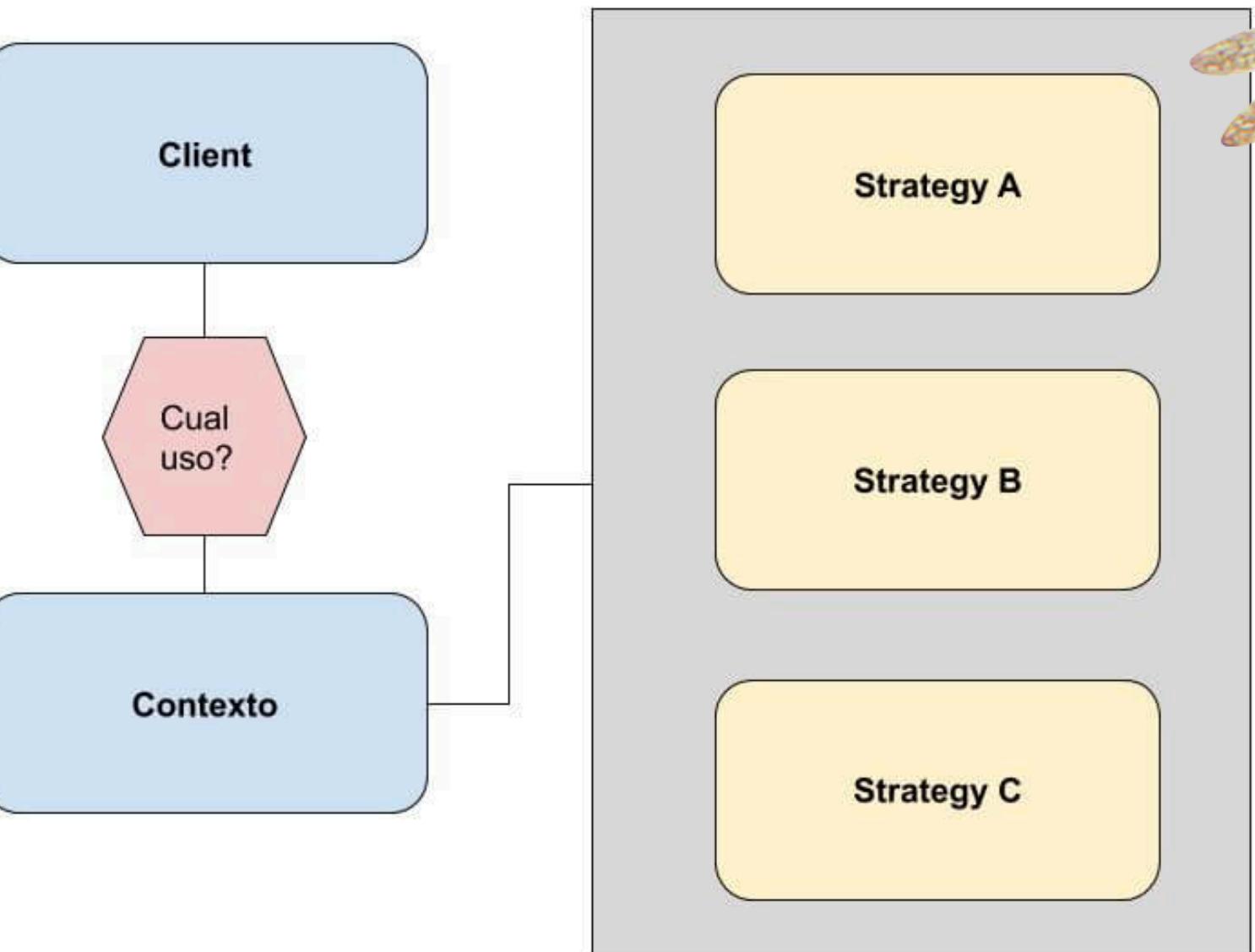
Este patrón se utilizó para poder comunicar las vistas con los modelos, cada vez que los modelos cambien, en este caso los tableros, las vistas mostrarán el tablero modificado haciendo una actualización de este mismo.





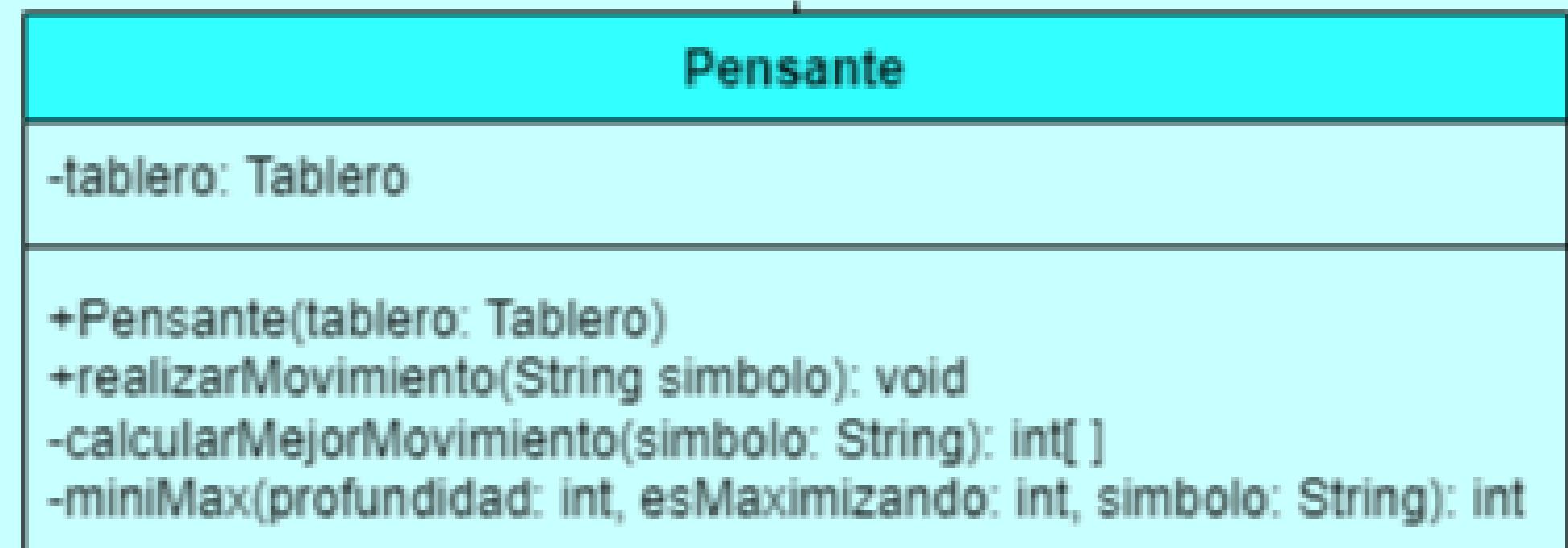
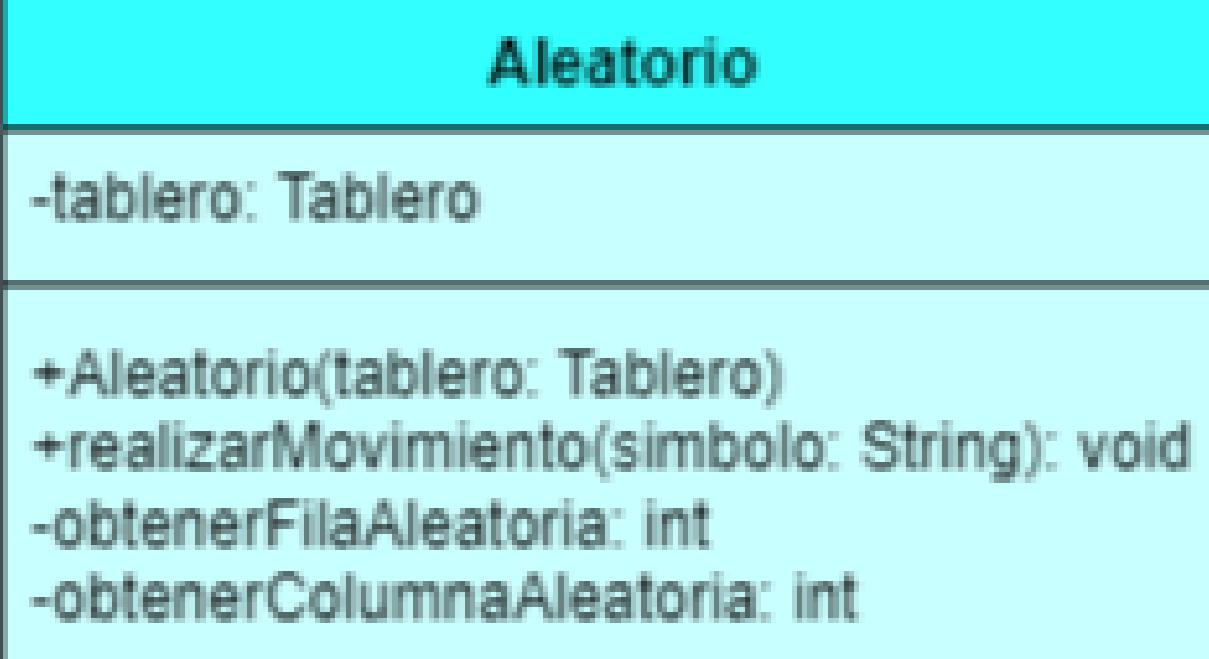
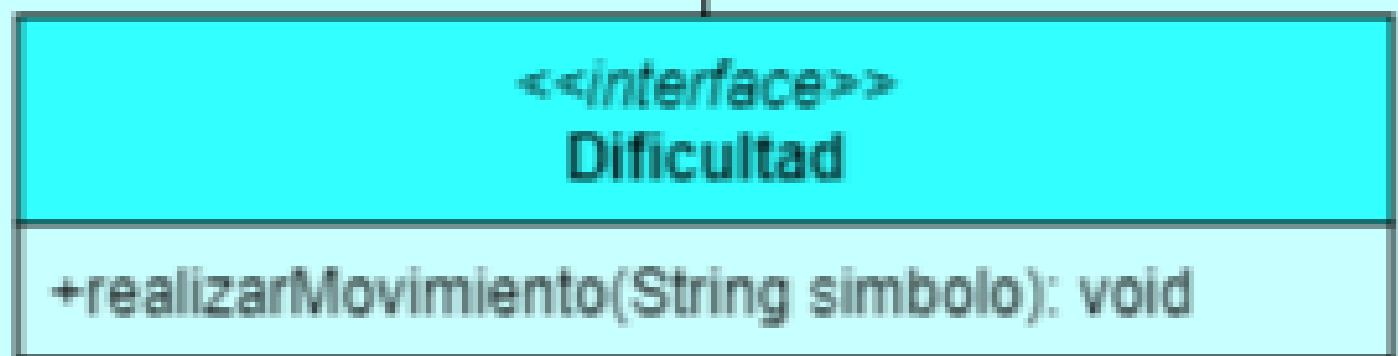
Strategy

Este patrón se utilizó en las dificultades del jugadorPC, tenemos las dificultades Aleatoria y Pensante, las dos realizan el mismo movimiento, pero cada una a su propia manera.



dificultad

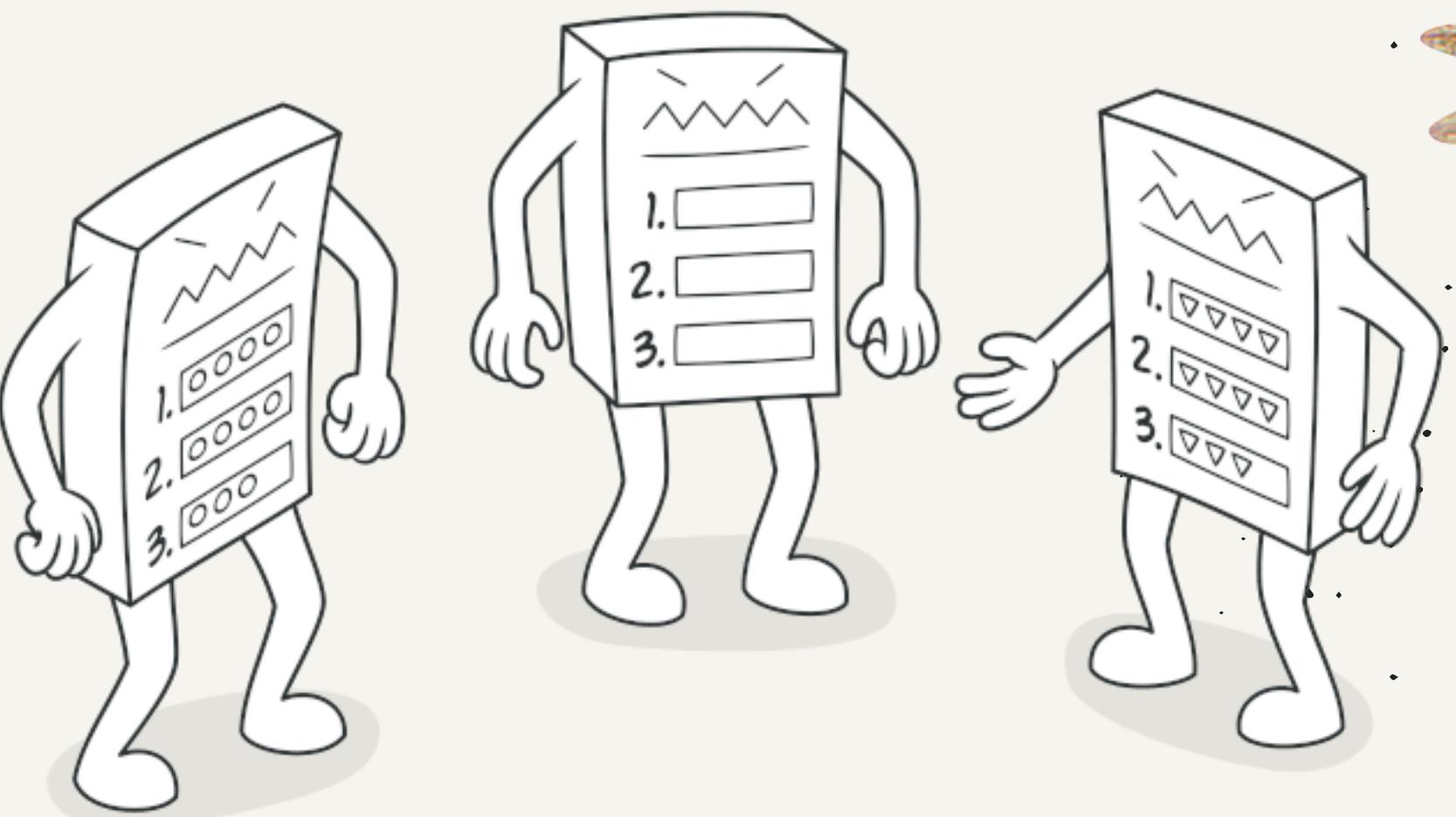
Strategy



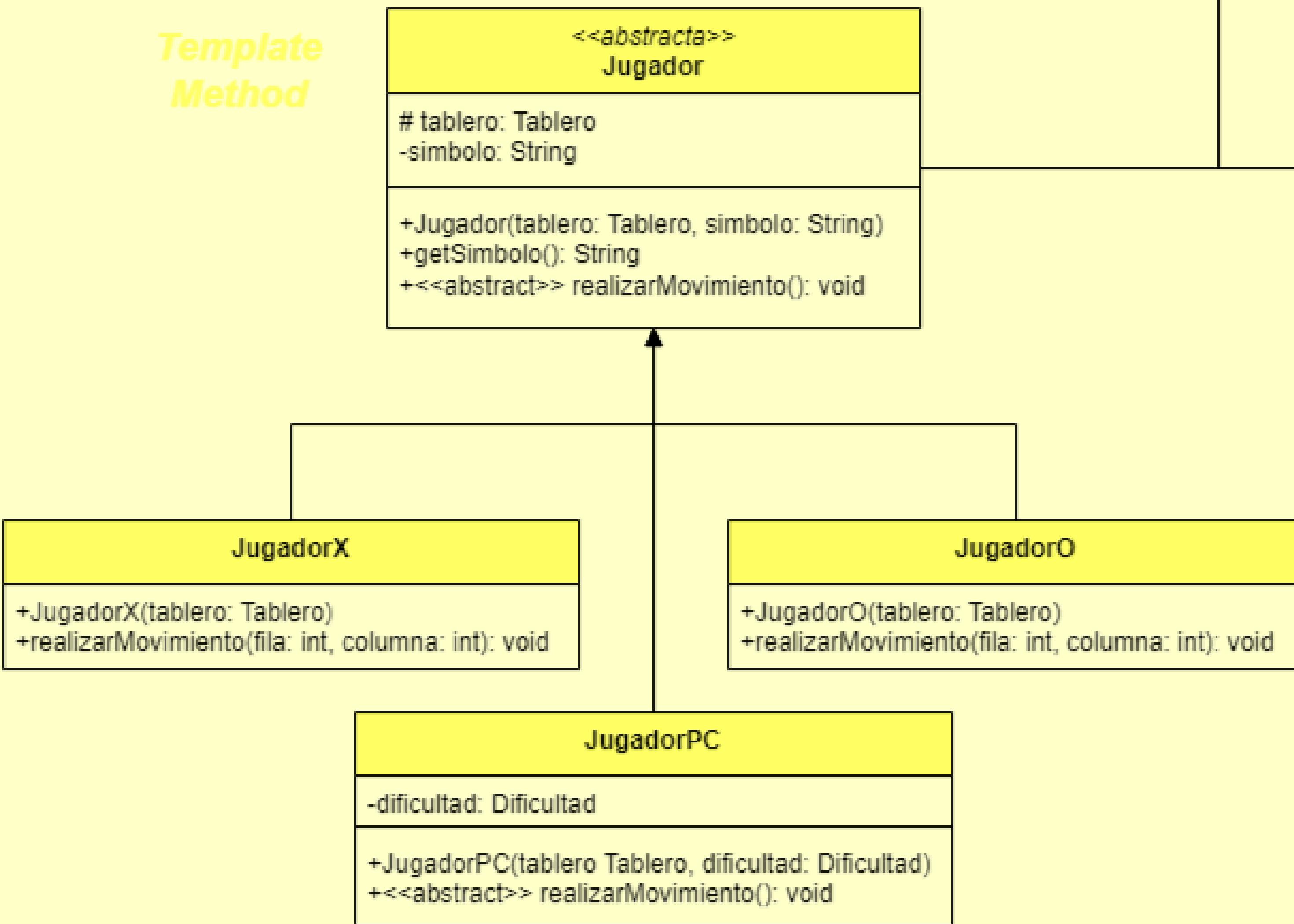
Template Method

El patrón Template Method sirve para extender pasos particulares de algún algoritmo, pero no todo en concreto o la estructura. En este caso la clase Jugador tiene un método `realizarMovimiento()` y las subclases JugadorX, JugadorO, JugadorPC lo implementa cada una a su manera. Además se utilizo ya que el algoritmo es casi idéntico, solo tiene pequeñas diferencias entre si.

El patrón Template Method fue investigado por cuenta propia, nos basamos en la página web llamada 'Refactoring Guru' [5] además de el repositorio de github 'Java Design Patterns' [6].

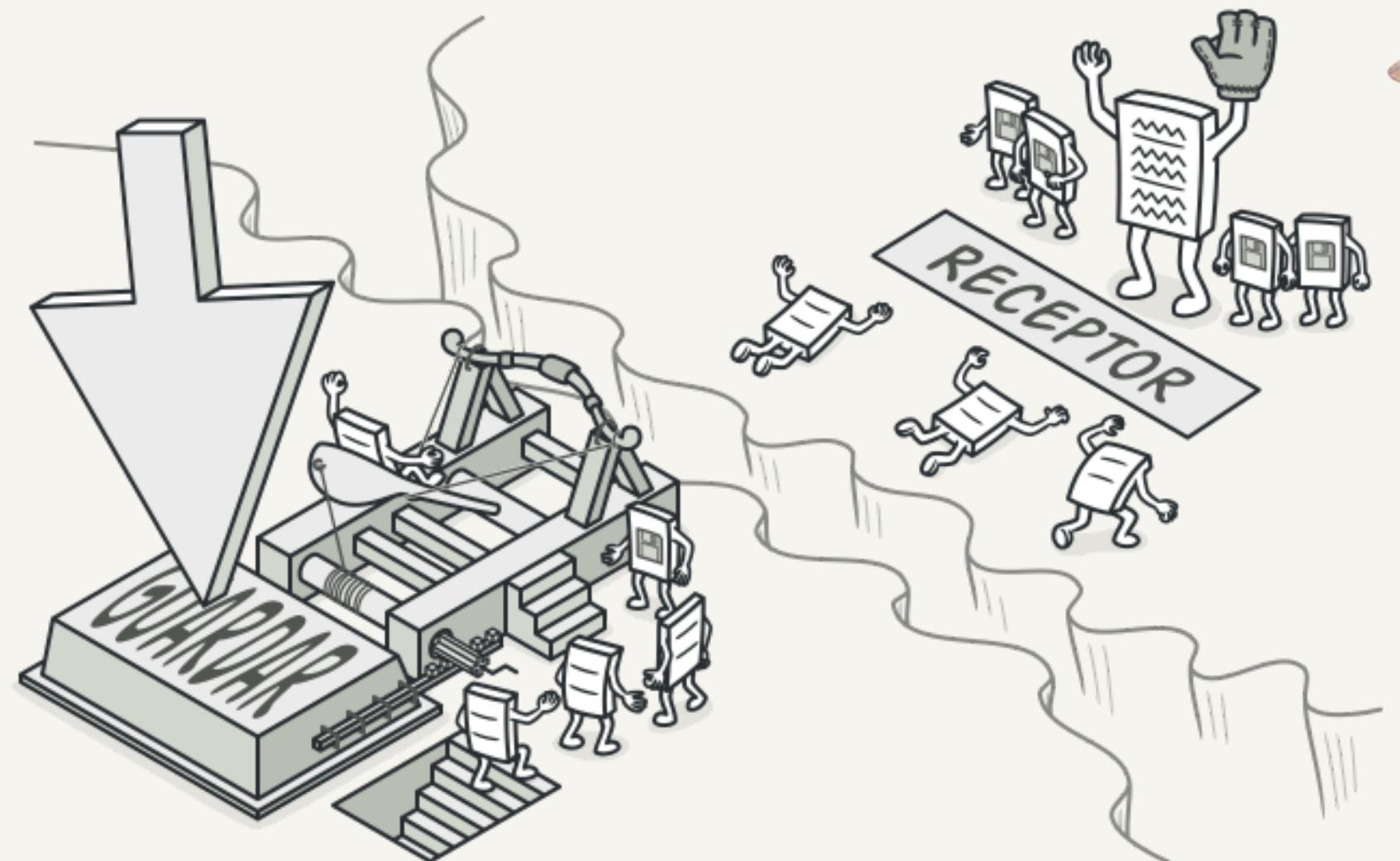


Template Method



Command

Este patrón se utilizo ya que Command es un patrón de diseño de comportamiento que convierte una solicitud en un objeto independiente que contiene toda la información sobre la solicitud. En este caso utilizamos command para realizar los movimientos. De esta manera el controladorPC y controlador1v1 no saben como se realiza el movimiento y no lo necesitan saber, de eso se encarga la clase MovimientoCommand que con su método execute(), realizamos el movimiento del jugador además de notificaciones. La información de este patrón se saco dela página web llamada 'Refactoring Guru' [4] además de el repositorio de github 'Java Design Patterns' [6].



<<interface>>

Command

+execute(): void

Command

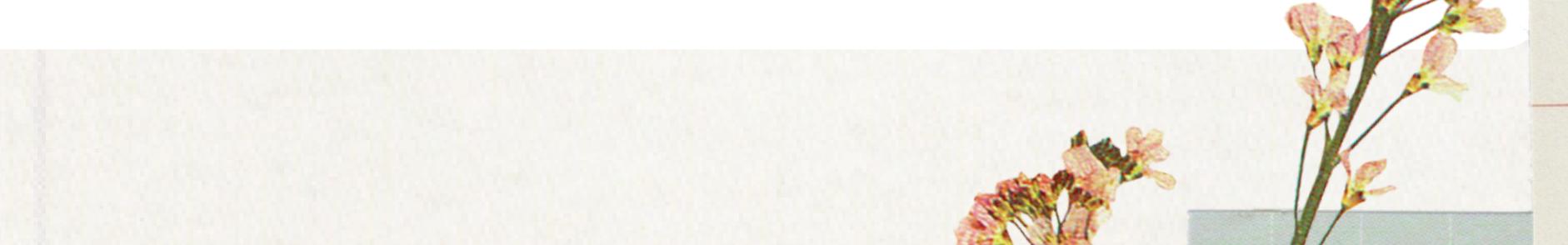
MovimientoCommand

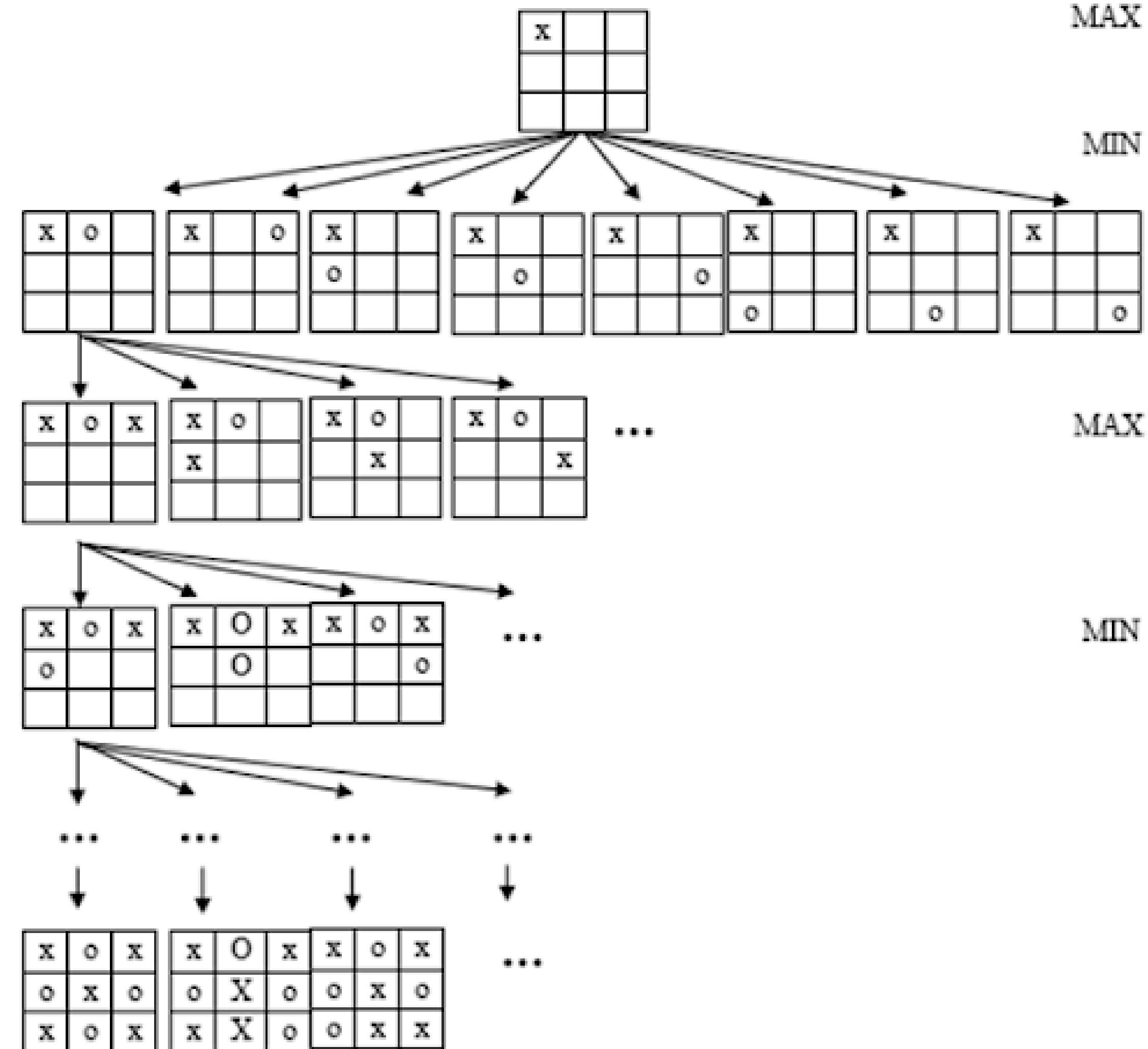
-tablero: Tablero
-fila: int
-columna: int

+MovimientoCommand(fila:int, columna: int, tablero: Tablero)
+execute(): void

Implementación de IA en Tres en Raya

El jugadorPC realiza el movimiento buscando el mejor movimiento de todos los posibles a rea lizar, se utilizo el algoritmo minimax. Para entender el proceso de este algoritmo nos basamos en el video 'Simple Explanation of the Minimax Algorithm with Tic-Tac-Toe' [1] y para poder implementarlo nos basamos en el video 'Tic-Tac-Toe AI Player using the Minimax Algorithm: A Step-By-Step Python Coding Tutorial' [2]. Además buscamos información del libro 'Digital Notes of Artificial Intelligence' [3].





Referencias

1. Sciencie Buddies. Simple Explanation of the Minimax Algorithm with Tic-Tac-Toe. Consultado en: 2024. 2024. url: <https://www.youtube.com/watch?v=5y2a0Zhgq0U>.
2. Sciencie Buddies. Tic-Tac-Toe AI Player using the Minimax Algorithm: A Step-By-Step Python Coding Tutorial. Consultado en: 2024. 2024. url: <https://www.youtube.com/watch?v=JWRxdoPrk7M>.
3. SRI INDU COLLEGE OF ENGINEERING y TECHNOLOGY. DIGITAL NOTES ON ARTIFICIAL INTELLIGENCE. Autonomous Institution - UGC, Govt. of India, 2023.
4. Refactoring Guru. Command. Consultado en: 2024. 2024. url: <https://refactoring.guru/es/design-patterns/command/java/example>.
5. Refactoring Guru. Template Method. Consultado en: 2024. 2024. url: <https://refactoring.guru/es/design-patterns/template-method>.
6. Mkejeiri. Java design patterns. Consultado en: 2024. 2020. url: <https://github.com/mkejeiri/Java-Design-Pattern>.