

Ajedrez++ (Ejercicio de Diseño)

El ajedrez ([Ajedrez](#)) es un juego muy antiguo y popular y, probablemente, el juego sobre el que se han inventado mayor cantidad de variantes ([Variante del ajedrez](#)). Las variantes suelen consistir en combinaciones de:

- modificar la topología del tablero o su cantidad (por ej. 1D, 3D, 2D pero con más o menos escaques, 2 tableros en espejo, escaques romboidales, hexagonales, etc.)
- modificar la cantidad, tipo de piezas y/o su disposición inicial (por ej. agregado de combinaciones alfil-caballo o torre-caballo, etc)
- modificar las reglas del juego (por ej. movimientos de piezas, reglas de captura, objetivos, número de jugadores, etc.)

Incluso dentro del ajedrez tradicional existen distintas modalidades (por ej. partidas simultáneas, torneos, con distintos ritmos o manejo de tiempo, contra IAs, etc.).

Y hasta hay algunas variaciones más extremas donde se combina con otros deportes ([Chess boxing](#)) que preferimos dejar de lado en este ejercicio... ¿Se referiría a eso el filósofo contemporáneo Lukas Podolski cuando dijo que *“el Fútbol es como el ajedrez, pero sin dados”*?

Objetivo

La idea del ejercicio es realizar un programa que permita a dos personas jugar al ajedrez tradicional, pero que sea lo suficientemente flexible en su diseño como para incorporar posibles variaciones (como más de un jugador, jugadores IA, distintas topologías de tableros, piezas nuevas, reglas diferentes, etc..).

Etapas de diseño

En esta primera etapa se pretende comenzar (siempre es un proceso iterativo) con un diseño **sin implementación**. Identificando los objetos con los que vamos a representar los distintos componentes del problema (encontrará que algunos se mapean literalmente a un

objeto del problema, pero otros no tienen una correspondencia directa).

Luego definiremos las clases con las que construiremos nuestros objetos, prestando atención a posibles jerarquías de herencia y uso de polimorfismo como herramientas para flexibilizar nuestra solución. Luego trataremos de delimitar las responsabilidades de cada objeto (qué necesitan los otros objetos de él y qué necesita él de los demás objetos) y con eso ya podemos empezar a bosquejar las interfaces

(aunque no sepamos con precisión argumentos o retornos todavía).



Etapa de implementación

Ahora podemos empezar a implementar nuestras clases. Probablemente se encontrará con algunas dificultades, que lo obliguen a volver a la etapa de diseño. Esto es normal y no hay por qué preocuparse. Como dijimos, es un proceso iterativo.

Por favor, no se complique con la representación visual del juego (no aporta gran cosa al problema, desde el punto de vista de POO). Basta con imprimir en ascii por la terminal para comprobar que funciona.

Extensión

Ahora viene lo más divertido. Si hicimos las cosas bien flexibles en las etapas anteriores, deberíamos poder ser capaces de extender nuestro programa para que incluya (además del ajedrez tradicional) otras variantes. Para probarlo, intentaremos incorporar alguna de las variantes que se mencionan en el artículo de wikipedia ([Variante del ajedrez](#)). Puede ser que su diseño no soporte alguna de estas variaciones porque son demasiado radicales, así que sugerimos que comience por alguna sencilla.