

**Universidad de Santiago de Chile**  
**Facultad de Ingeniería**  
**Departamento de Ingeniería Informática**  
**Paradigmas de Programación**

**Paradigmas de Programación**  
**Proyecto Semestral de Laboratorio**  
Laboratorio 3  
Paradigma Orientado a Objetos - Java

Alumno: Benjamin Antonio Ortega Quinteros

Profesor: Edmundo Leiva

Carrera: Ingeniería de Ejecución en Computación e Informática

Fecha: 26-12-2024

# Índice

- 1. Introducción**
- 2. Descripción del Problema**
- 3. Análisis del Problema**
- 4. Diseño de la Solución**
- 5. Aspectos de Implementación e Instrucciones de Uso**
- 6. Resultados y Autoevaluación**
- 7. Conclusión**
- 8. Referencias y Anexo**

# Introducción

En el presente informe se aborda la problemática de implementar el juego Conecta 4 utilizando el paradigma orientado a objetos en el lenguaje de programación Java. Este proyecto tiene como objetivo resolver el problema planteado aplicando los principios fundamentales de la programación orientada a objetos, aprovechando la abstracción y modularidad que este paradigma ofrece.

La estructura del informe comienza con una introducción que contextualiza el problema, seguida de una descripción detallada del mismo y de los conceptos esenciales del paradigma orientado a objetos. Posteriormente, se presenta el análisis del problema, el diseño de la solución, y los aspectos más relevantes de la implementación. El informe también incluye instrucciones de uso, resultados obtenidos, una autoevaluación del proceso, las conclusiones del proyecto, referencias bibliográficas, y anexos para complementar la información.

## **Descripción del Problema**

Conecta 4 es un juego de estrategia para dos jugadores que se juega en un tablero vertical de 6 filas y 7 columnas. Cada jugador tiene a su disposición 21 fichas, cada jugador tiene asociado un color diferente. Los jugadores se turnan para dejar caer fichas de su color desde la parte superior del tablero. Las fichas caen hasta la posición más baja disponible en la columna seleccionada. El objetivo del juego es ser el primero en formar una línea de cuatro fichas del mismo color, ya sea horizontal, vertical o diagonalmente.

El problema planteado es implementar el juego Conecta 4 en el paradigma de programación orientado a objetos a través del cumplimiento de ciertos requerimientos que deben ser abordados bajo este paradigma y otras condiciones propuestas.

# Descripción del Paradigma

El paradigma orientado a objetos es un enfoque de programación que organiza el código en entidades llamadas "objetos", los cuales combinan datos (atributos) y comportamientos (métodos). Esta organización permite una estructura más clara y comprensible, que facilita la modularidad, la reutilización de código y el mantenimiento del software.

Cada objeto representa una instancia de una "clase", que define las características y comportamientos comunes de los objetos de ese tipo. Además de su estructura básica, la programación orientada a objetos promueve varios principios clave que optimizan el desarrollo y la resolución de problemas:

- **Encapsulación:** Este principio permite ocultar los detalles internos de los objetos y exponer solo la interfaz necesaria para interactuar con ellos.
- **Herencia:** A través de la herencia, una clase puede heredar atributos y métodos de otra, lo que fomenta la reutilización de código.
- **Polimorfismo:** El polimorfismo permite que una interfaz o método pueda ser utilizado de diferentes maneras según el tipo de objeto, promoviendo la flexibilidad en el diseño del sistema.

El paradigma orientado a objetos ofrece múltiples beneficios para el desarrollo de aplicaciones avanzadas y sostenibles. Además de facilitar la modularidad y reutilización de código, promueve una organización más lógica y comprensible del software, permitiendo a los programadores crear sistemas de manera más intuitiva.

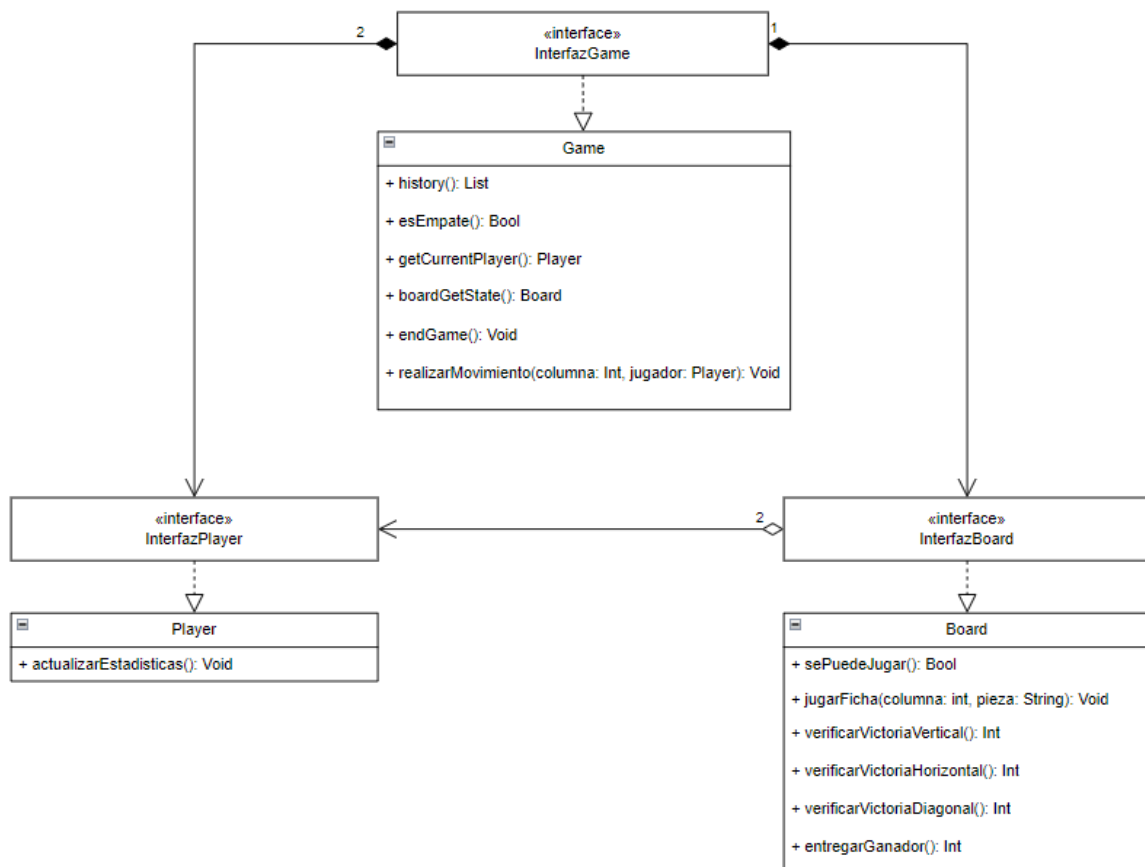
# Análisis del Problema

Para abordar este problema, es necesario identificar y definir los TDAs básicos que se requieren. En este caso, se han seleccionado tres TDAs fundamentales: **Board**, **Player** y **Game**. Cada uno de estos TDAs desempeña un papel clave en la gestión de la lógica del juego:

- **Board**: Representa el tablero de juego, donde se almacenan las piezas y se gestionan las jugadas.
- **Player**: Representa a cada jugador, gestionando sus atributos y las piezas disponibles para jugar.
- **Game**: Controla el flujo del juego, gestionando las interacciones entre los jugadores, el tablero y el estado del juego.

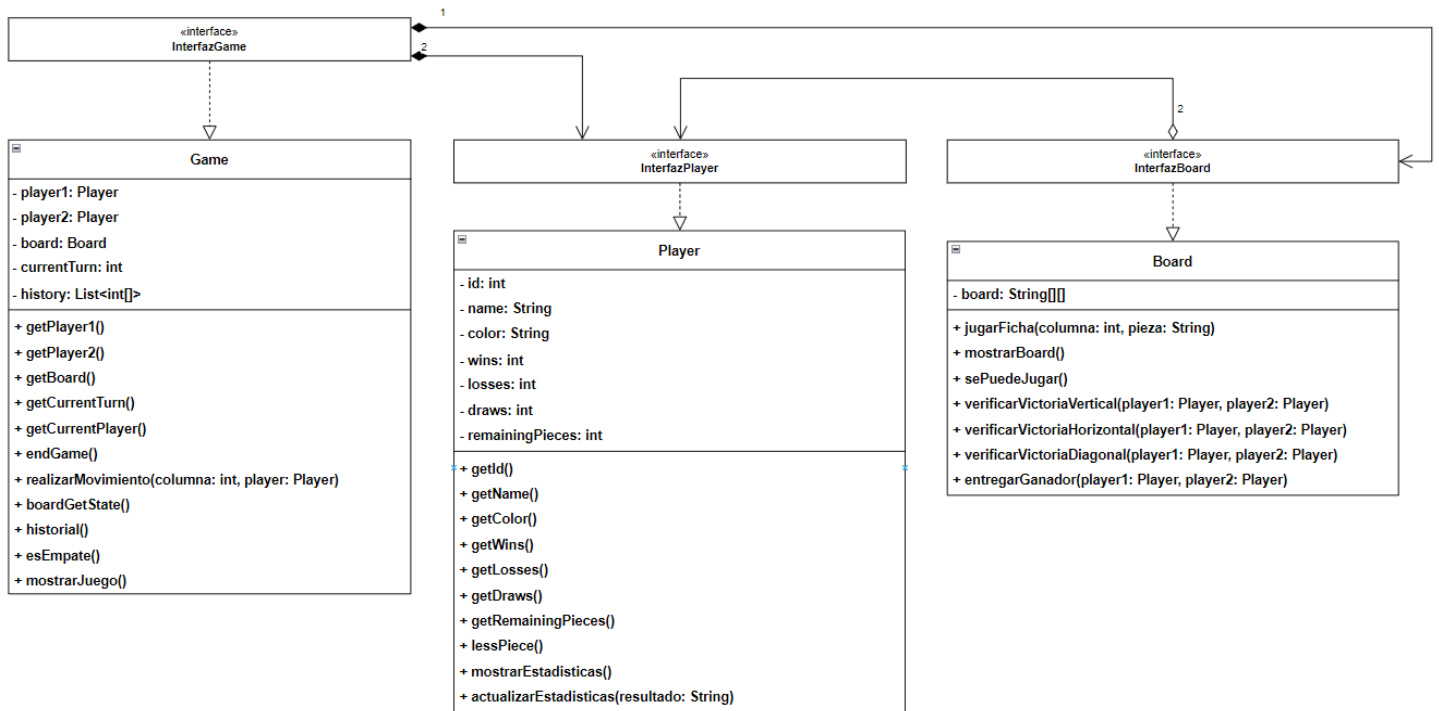
Se requiere implementar estos TDAs usando el paradigma orientado a objetos, con una implementación basada en java, por lo que se deben expresar los TDAs necesarios a través de interfaces y/o herencia que luego deben ser implementadas mediante clases.

A continuación se presenta un diagrama UML de análisis:



# Diseño de la Solución

La implementación se basa en el uso de clases y sus respectivos métodos para organizar los elementos principales del juego así como para lograr implementar parámetros del juego en cuestión. Dicho esto, se presenta un diagrama UML con la implementación realizada:



Como se muestra en el diagrama, se implementaron los tres TDAs clave identificados en el análisis del problema como clases: **Game**, **Player** y **Board**. La clase **Game** actúa como una composición de las otras dos clases, lo que significa que gestiona tanto el tablero como los jugadores, siendo responsable de controlar el flujo del juego. Por otro lado, la clase **Player** se relaciona con la clase **Board** mediante una agregación, ya que cada jugador tiene acceso al tablero para realizar movimientos y tomar decisiones durante la partida.

## Aspectos de la Implementación

La estructura del proyecto se compone por 7 archivos, en donde cuatro son las clases: Game, Player, Board y Main, y los tres restantes son las interfaces de cada clase exceptuando la clase Main.

Para la implementación de la solución se usó el lenguaje de programación Java, con JDK (Java Development Kit) versión 11.0.25, con la herramienta de compilación Gradle versión 8.5.

Para el desarrollo se usó el IDE IntelliJ IDEA Community Edition 2024.1 en el sistema operativo Windows 11. No se usaron librerías externas a las integradas en el core de Java.

## Instrucciones de Uso

Para compilar el programa se requiere que Gradle esté instalado en el dispositivo, luego, con el proyecto descargado, abrimos el terminal del respectivo sistema operativo y nos dirigimos a la ruta en donde se encuentra el proyecto. Hecho esto debemos ingresar los siguientes comandos:

### Linux/Unix:

Compilación: **./gradlew build**

Ejecución: **./gradlew run**

### Windows(cmd):

Compilación: **gradlew.bat build**

Ejecución: **gradlew.bat run**

Se recomienda agregar el comando **--console=plain** a la ejecución para que no estorbe la barra de progreso de Gradle, tal que el comando de ejecución sea **gradlew.bat run --console=plain** o **./gradlew.bat run --console=plain** en su defecto.

Hecho esto se ejecutará el programa que será operado por consola. El programa está preparado para errores de entrada por lo cual si se ingresa un dato erróneo al que se pide se da retroalimentación por consola para ingresar nuevamente el dato, incluyendo el tipo de dato y su intervalo.



## Resultados y Autoevaluación

La implementación realizada cumple con todos los requerimientos definidos en la descripción del problema y las expectativas esperadas de los requerimientos funcionales y no funcionales, además se incorporaron medidas preventivas para gestionar posibles errores a la hora de ingresar datos en el programa. La implementación realizada por consola se ve:

```
### Conecta4 - Menu Principal ###
Bienvenido al juego Conecta4
Seleccione una opcion:

1. Crear nuevo juego
2. Visualizar estado actual
3. Realizar jugada
4. Ver estadisticas generales
5. Ver juego completo
6. Salir del juego

Ingrese su opcion: |
```

```
### Realizar Jugada ###
Turno de: Gaspar (Rojo)
Fichas restantes: 9
Seleccione columna (0-6): 0
Movimiento realizado:
```

	1	2	3	4	5	6	7
1							
2							
3							
4							
5							
6	R						

En un futuro se podría seguir desarrollando esta implementación, agregando una interfaz gráfica que sea más intuitiva para el usuario y lograr una experiencia mucho más satisfactoria a la hora de jugar Conecta4.

# Conclusión

Podemos concluir que la implementación del juego Conecta-4 utilizando el paradigma de programación orientado a objetos ha sido exitosa. Todos los requisitos funcionales fueron cumplidos de manera adecuada, y se logró una solución eficiente gracias a la correcta aplicación de condiciones en cada entrada del usuario, lo que ayuda a reducir errores y mejorar la experiencia de juego.

En comparación con otros paradigmas estudiados en el curso, el uso de clases y sus métodos en la programación orientada a objetos resultó ser una herramienta poderosa, ya que facilita la modularidad, la reutilización de código y la claridad en la estructura del sistema. La posibilidad de organizar el código en clases y objetos hace que el desarrollo sea más manejable y que la solución sea escalable y mantenible. Sin embargo, la mayor dificultad fue comprender cómo utilizar eficazmente las clases y los métodos para optimizar el desarrollo.

## Referencias

López, M. (2023, diciembre 9). *Aplicaciones prácticas de la programación orientada a objetos*. IMMUNE.  
<https://immune.institute/blog/programacion-orientada-a-objetos/>

EDteam. (2022, octubre 5). *¿Qué es la programación orientada a objetos (POO)?* EDteam.  
<https://ed.team/blog/que-es-la-programacion-orientada-a-objetos-poo>