

## INFORME TÉCNICO

**Proyecto Tres en Raya (Tic Tac Toe)**

### **PORTADA**

**UNIVERSIDAD DE PAMPLONA**

**Facultad de Ingenierías y Arquitecturas**

**Programa:** Ingeniería de Sistemas

**Semestre:** Segundo Semestre – 2025

### **INFORME TÉCNICO**

**Proyecto:** Juego Tres en Raya (Tic Tac Toe)

### **ELABORADO POR:**

- **Jhon Jairo Vera Acevedo** – Developer
- **Aylin Yuleni Estupiñán González** – Developer
- **Danna Sofia Clavijo Plata** – Scrum Master
- **Laura Sofía Monterrey Gutiérrez** – Product Owner

### **FECHA:**

Diciembre de 2025

### **ÍNDICE**

1. Introducción
2. Objetivos del Proyecto
3. Alcance del Sistema
4. Tecnologías Utilizadas

5. Arquitectura del Sistema
6. Descripción de Clases
7. Flujo del Programa
8. Pruebas Realizadas
9. Resultados Obtenidos
10. Conclusiones
11. Recomendaciones

## 1. INTRODUCCIÓN

El presente **Informe Técnico** documenta el desarrollo del proyecto **Tres en Raya**, una aplicación de consola desarrollada en **Java**, cuyo propósito es aplicar los conceptos fundamentales de **Programación Orientada a Objetos**, arquitectura de software y control de versiones.

## 2. OBJETIVOS DEL PROYECTO

### Objetivo General

Desarrollar un juego de Tres en Raya funcional en Java, aplicando principios de programación estructurada y orientada a objetos.

### Objetivos Específicos

- Implementar un tablero 3x3 con validación de movimientos
- Aplicar herencia y polimorfismo
- Desarrollar una inteligencia artificial básica
- Manejar errores de entrada por consola
- Implementar control de versiones con Git

## 3. ALCANCE DEL SISTEMA

El sistema permite:

- Juego entre dos jugadores humanos
- Juego contra la computadora
- Detección automática de ganador o empate
- Interfaz de consola con colores

No incluye interfaz gráfica ni almacenamiento en base de datos.

#### 4. TECNOLOGÍAS UTILIZADAS

- **Lenguaje:** Java SE 17
- **IDE:** IntelliJ IDEA Community Edition
- **Control de versiones:** Git y GitHub
- **Paradigma:** Programación Orientada a Objetos

#### 5. ARQUITECTURA DEL SISTEMA

El proyecto sigue una arquitectura **MVC no estricta**, separando responsabilidades entre:

- **Modelo:** Tablero, EstadoJuego
- **Controlador:** TresEnRaya
- **Vista:** Consola (salida por terminal)

#### 6. DESCRIPCIÓN DE CLASES PRINCIPALES

##### **Main.java**

Clase principal que inicia la aplicación.

##### **TresEnRaya.java**

Controlador que gestiona el flujo del juego.

##### **Tablero.java**

Representa el tablero y valida movimientos.

##### **Jugador.java**

Clase abstracta base para los jugadores.

#### **JugadorHumano.java**

Obtiene movimientos desde la consola.

#### **JugadorComputadora.java**

Implementa una inteligencia artificial básica.

### **7. FLUJO DEL PROGRAMA**

1. Inicio del programa
2. Configuración del juego
3. Ciclo de turnos
4. Validación de movimientos
5. Verificación de ganador o empate
6. Finalización de la partida

### **8. PRUEBAS REALIZADAS**

- Validación de coordenadas
- Detección de victoria (horizontal, vertical y diagonal)
- Manejo de casillas ocupadas
- Empate por tablero lleno

### **9. RESULTADOS OBTENIDOS**

El sistema funciona correctamente, responde a entradas inválidas y muestra resultados confiables, cumpliendo con los objetivos planteados.

### **10. CONCLUSIONES**

El desarrollo del proyecto permitió fortalecer conocimientos en Java, POO y trabajo colaborativo, logrando un sistema funcional y escalable.

## **11. RECOMENDACIONES**

- Implementar pruebas automatizadas con JUnit
- Añadir interfaz gráfica con JavaFX
- Incorporar niveles de dificultad
- Migrar a Maven o Gradle