

# **Programación II: Orientada a Objetos Avanzada**

## **Caso de Estudio CompuWork**

**Jhon Jairo Rivera Menco**

Desarrollo de software

4to semestre

### **Grupo**

PREICA2501B010023

### **Docente**

CAROLINA MARQUEZ

**8 de Abril**

**IU Digital de Antioquia**

2025

## Introducción

La presente actividad tiene como objetivo implementar un sistema de administración de empleados y departamentos, aplicando los principios de la Programación Orientada a Objetos (POO). A través del desarrollo de clases específicas, pruebas unitarias y una interfaz gráfica usando JavaFX, se busca fortalecer las habilidades de diseño, codificación y validación de software.

## Explicación del sistema desarrollado

El sistema **CompuWork** permite gestionar empleados de tipo **permanente** y **temporal**, agrupándolos por departamentos. Cada tipo de empleado tiene su propia lógica para calcular el desempeño. El sistema está diseñado para ser **escalable**, **modular** y **mantenible**, utilizando herencia, abstracción y listas dinámicas.

Se desarrollaron las siguientes clases principales:

- **Empleado**: clase abstracta base.
- **EmpleadoPermanente**: representa empleados con años de antigüedad.
- **EmpleadoTemporal**: representa empleados por proyectos.
- **Departamento**: contiene una lista de empleados.
- **MainApp**: interfaz gráfica desarrollada con JavaFX.

También se realizaron pruebas unitarias utilizando **JUnit 5**, verificando el correcto cálculo del desempeño de cada tipo de empleado.

## Capturas de pantalla

el código no se pudo ejecutar en mi pc, es una tostadora.

<https://github.com/jirm23/compuwork3-integracion#>

## Conclusión

Durante esta actividad aprendí a aplicar conceptos avanzados de programación orientada a objetos en un sistema real, mejorando mi lógica y diseño modular. Me familiaricé con la creación de interfaces gráficas en JavaFX y la importancia de realizar pruebas automatizadas con JUnit para garantizar la calidad del código.

Uno de los principales retos fue configurar correctamente el entorno JavaFX y solucionar el error del **QuantumRenderer**, lo cual resolví ajustando los argumentos de ejecución y asegurando la compatibilidad del SDK. También fue retador integrar las clases y pruebas en un repositorio limpio y organizado.

Este proyecto me permitió afianzar conocimientos clave para futuros desarrollos profesionales.

## Referencias (APA)

Oracle. (2024). *JavaFX documentation*. <https://openjfx.io>

Oracle. (2024). *Java 24 Documentation*. <https://docs.oracle.com/en/java/javase/24/>

JUnit. (2024). *JUnit 5 User Guide*. <https://junit.org/junit5/docs/current/user-guide/>

GeeksforGeeks. (2023). *Introduction to OOP in Java*.

<https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/>

