Gestión de Trabajo Empresarial

Proyecto compuesto por:

- Salvador Lopez Trigueros
- Alejandro Sierra Diaz
- Manuel Rossi Domínguez
- Manuel Ávila Dugo
- Jose Manuel Sanchez Rosal

1. Introducción

Este proyecto ha sido desarrollado como parte de la asignatura **Entornos de Desarrollo**, cursada en el primer año del Ciclo Formativo de Grado Superior en **Desarrollo de Aplicaciones Multiplataforma (DAM)**. El propósito principal del trabajo es aplicar los conocimientos adquiridos sobre análisis y diseño orientado a objetos, empleando **UML (Unified Modeling Language)** como herramienta de modelado.

El resultado es un sistema básico de gestión de proyectos en el que se pueden representar trabajadores, asignarlos a distintos proyectos, y organizar tareas en función de cargos, fechas y roles definidos.

2. Objetivos del Proyecto

Los principales objetivos establecidos para este proyecto fueron:

- Analizar un sistema realista de gestión empresarial y descomponerlo en entidades, relaciones y comportamientos relevantes.
- Elaborar un **diagrama de clases UML** completo que represente con precisión la estructura del sistema.
- Utilizar una herramienta de modelado como **Modelio 5** para la generación automática de código.
- Implementar parcialmente la lógica del sistema en el lenguaje de programación Java.
- Presentar una documentación clara, bien estructurada y en formato Markdown.
- Preparar una breve exposición para defender el diseño del sistema.

3. Análisis del Problema

En el análisis inicial se planteó la necesidad de una herramienta que permita:

- Gestionar **proyectos** empresariales.
- Registrar y organizar **trabajadores**.
- Asignar trabajadores a proyectos concretos.

- Realizar seguimiento de fechas y estados de desarrollo.

De esta manera, se identificaron las **entidades clave** del sistema:

- `Trabajadores` (clase abstracta): modelo genérico para cualquier trabajador.
- `Empleado`: representa un trabajador que ocupa un puesto determinado en la empresa.
- `Proyecto`: representa un proyecto empresarial, con participantes y fechas clave.

Este enfoque permite reflejar una estructura básica de recursos humanos, útil para cualquier organización que quiera llevar un control de sus equipos de trabajo y proyectos activos.

4. Diseño UML

El diseño se ha realizado utilizando la herramienta **Modelio 5**, que facilita tanto el modelado como la generación automática del esqueleto de código en Java.

Elementos destacados del diagrama de clases:

- **Herencia**: `Empleado` extiende de `Trabajadores`.
- **Encapsulamiento**: todos los atributos son `private` y se accede a ellos mediante métodos públicos.
- **Asociación**: `Proyecto` mantiene una lista de empleados participantes (array).
- **Tipos de datos**: uso de `Date` para representar fechas de ingreso y `String` para fechas en el proyecto (simplificado).
- **Anotaciones UML**: uso de metadatos como `@objid` que permite organizar mejor el diseño.

Este diseño sigue principios SOLID básicos como la segregación de responsabilidades y la reutilización de código mediante herencia.

5. Implementación en Java

La estructura de clases generada ha sido organizada de la siguiente forma:

5.1. Clase abstracta `Trabajadores`

Esta clase actúa como base para otras clases derivadas. Sus atributos encapsulan la información general del trabajador:

- `String nombre`
- `String apellido`
- `String DNI`
- `int edad`
- `String direccion`
- `Date fechaIngreso`
- `int sueldoBase`

Además, define un método **abstracto** `mostrarInformacion()`, que obliga a las subclases a proporcionar su propia implementación para mostrar detalles del trabajador.

5.2. Clase `Empleado`

Hereda de `Trabajadores` y agrega información más específica:

- `String cargo`
- `String puestoTrabajo`

Implementa el método `mostrarInformacion()` sobreescribiendo la versión abstracta e incluye un **constructor parametrizado** para facilitar la creación de objetos.

5.3. Clase `Proyecto`

Define las características de un proyecto empresarial:

- `String nombre`
- `String descripcion`
- `String[] listaPersonas`
- `String estadoProyecto`
- `String fechaInicio`
- `String fechaFin`

Incluye un **constructor completo** que permite inicializar todos los atributos de una vez, representando así proyectos con participantes definidos, fechas establecidas y una descripción clara del objetivo.

6. Herramientas Utilizadas

Durante el desarrollo del proyecto se han empleado diversas herramientas:

- **Modelio 5**: Software de modelado UML profesional. Permite crear diagramas, gestionar clases, relaciones y generar directamente el código en Java.
- **Java (JDK 17)**: Lenguaje de programación orientado a objetos, ideal para implementar los conceptos del diseño.
- **Markdown (.md)**: Lenguaje de marcado ligero utilizado para redactar esta documentación de manera clara y estructurada.
- **GitHub**: Alojamiento del repositorio del proyecto (Repositorio aquí).

7. Conclusiones

El proyecto ha permitido poner en práctica habilidades fundamentales en el diseño de software: desde el análisis del problema, pasando por el modelado UML, hasta la implementación inicial del sistema.

El uso de herramientas profesionales como Modelio y GitHub ha contribuido a mejorar la organización y calidad del código.

A través de este ejercicio, se ha comprendido la importancia de una buena estructura de clases y de aplicar principios sólidos de la programación orientada a objetos, preparando así una base sólida para el desarrollo de aplicaciones más complejas en el futuro.