

## **Instituto Técnico - Educación Media**

Proyecto Integrador

Registro de Inasistencias Docentes

Grupo: 2MA

Nombre del grupo: Grupo 1

Integrantes: Natasha Ramos

Asignaturas: Programación Avanzada

Docentes a cargo: Marcela Mederos

Fecha de entrega: 7-11-2025

# Índice

Índice	2
Imagen de la aplicación	3
Introducción	3
Organización del Equipo y Análisis FODA	4
1. Definición de Roles	5
2. Reglamento interno	5
3. Relevamiento de necesidades	5
4. Propuesta funcional	6
5. Análisis del usuario	6
6. Capa Lógica y Persistencia	7
7. Diagramas UML	7
8. Prototipo en Figma	8
9. Conclusiones	9
10. Referencias	9

## Introducción

Este proyecto tiene como objetivo desarrollar una aplicación de escritorio que registre y publique inasistencias docentes, especialmente por licencias médicas. Será desarrollada en Java (Swing), utilizando una base de datos MySQL, y su arquitectura de red será simulada en Packet Tracer. La aplicación permitirá que alumnos y docentes consulten de forma pública las licencias activas, mientras que el personal administrativo podrá gestionar los registros.



# Organización del Equipo y Análisis FODA

El grupo original estuvo compuesto por tres integrantes: Natasha Ramos, Thiago Castillo y Francis Cedres. Sin embargo, debido a la falta de compromiso y participación de los dos últimos, el proyecto fue desarrollado íntegramente por Natasha Ramos, quien asumió todos los roles técnicos y documentales.

## Análisis FODA

- Fortalezas: Capacidad de autogestión, organización del tiempo, dominio de Java y comprensión de las capas del sistema. Alta responsabilidad y dedicación para cumplir con los objetivos del proyecto.
- Oportunidades: Mejora continua en el uso de herramientas profesionales (NetBeans, Draw.io y Figma), aprendizaje en documentación técnica formal y desarrollo de proyectos reales.
- Debilidades: Sobrecarga de tareas al trabajar individualmente, menor revisión por pares y limitación de tiempo para realizar pruebas exhaustivas.
- Amenazas: Riesgo de errores por carga individual de trabajo y dificultades para coordinar sin un equipo activo.

## 1. Definición de Roles

Rol desempeñado: Programadora principal, diseñadora de interfaz, gestora de base de datos y documentadora.

## 2. Reglamento interno

Responsabilidad: El único integrante responsable adoptó cada rol de los participantes anteriores.

Mejora continua: Se pueden mejorar errores en el proyecto y documentación

Tiempos de entrega: Se definirán plazos internos para evitar retrasos.

Comunicación: No se usó ninguno en esta entrega

Herramientas: Google Drive.

Toma de decisiones: Por mayoría con justificación técnica.

Nombre de la app: Docente Gestiones.

## 3. Relevamiento de necesidades

El proyecto responde a la necesidad institucional de contar con un sistema que gestione las licencias docentes de forma digital, segura y eficiente. Se busca reducir errores humanos, mejorar el acceso a la información y facilitar el registro de licencias médicas y administrativas.

## 4. Propuesta funcional

El sistema se estructura en tres capas principales:

- **Capa Lógica:** Contiene las clases **Docente**, **Licencia** y **ClaseLogica**. Estas clases representan los datos del sistema y coordinan la lógica de negocio. **ClaseLogica** actúa como intermediaria entre la interfaz gráfica y la base de datos, encapsulando la creación de objetos y la delegación de operaciones.
- **Capa de Persistencia:** Incluye las clases **GestorDAO** y **ConexionBD**. **GestorDAO** se encarga de ejecutar las operaciones CRUD (crear, leer, actualizar, eliminar) sobre la base de datos MySQL. **ConexionBD** establece y gestiona la conexión con el servidor local mediante JDBC.
- **Capa Gráfica:** Representada por las interfaces **VentanaDocente** y **Registro**. **VentanaDocente** permite ingresar y guardar datos de licencias docentes. **Registro** muestra los registros existentes en una tabla y permite eliminar filas específicas.

## 5. Análisis del usuario

La aplicación está diseñada para dos perfiles de usuario:

### a) Personal administrativo

- **Rol:** Gestionar el registro, edición y eliminación de inasistencias docentes.
- **Requisitos:** Conocimientos básicos de informática y manejo de formularios.
- **Interacción:** Accede a la interfaz completa, incluyendo formularios de carga, edición y visualización de registros.

### b) Alumnos y docentes

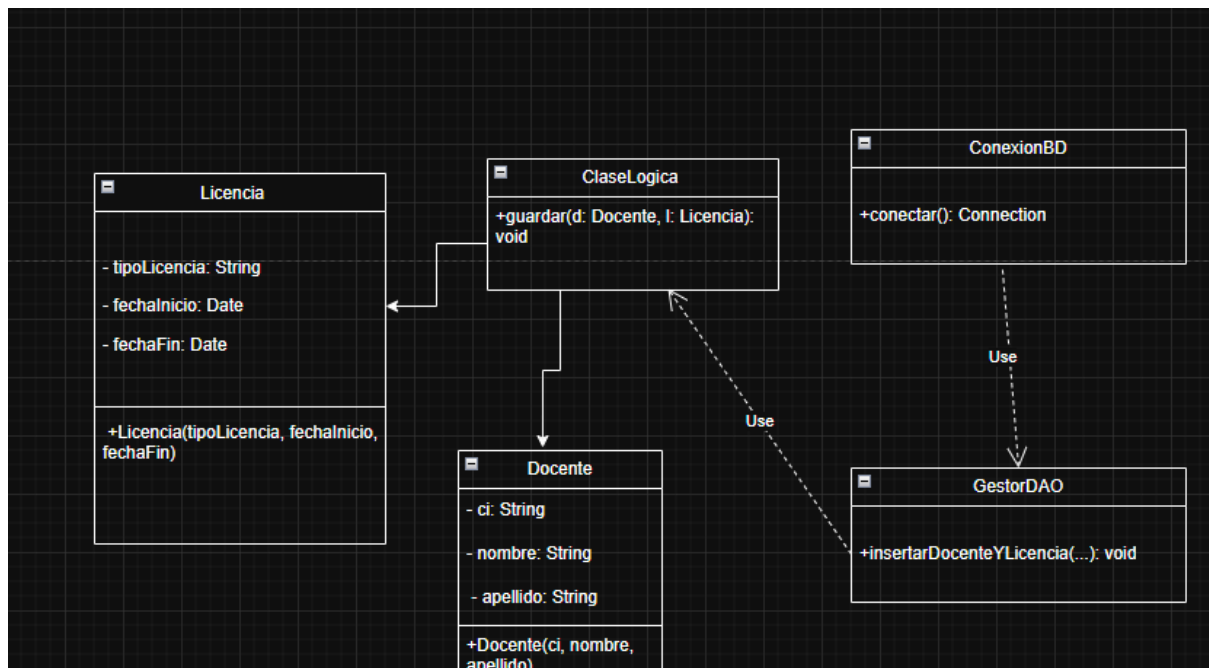
- **Rol:** Consultar inasistencias activas.
- **Requisitos:** No se requiere conocimiento técnico.
- **Interacción:** Acceden a un panel de solo lectura donde se visualiza el listado de licencias filtradas por turno o grupo.

## 6. Capa Lógica y Persistencia

Las clases desarrolladas en Java implementan la lógica del negocio y la conexión con MySQL mediante JDBC.

- **ClaseLogica** coordina las operaciones entre las entidades **Docente** y **Licencia**, y el gestor **GestorDAO**.
- **ConexionBD** establece la conexión con la base de datos y verifica el acceso al servidor local.
- **GestorDAO** ejecuta las operaciones SQL necesarias para insertar y eliminar registros.
- Las clases **Docente** y **Licencia** encapsulan los atributos y métodos necesarios para representar cada entidad del sistema.

## 7. Diagramas UML



Se elaboraron diagramas UML que representan la estructura y relaciones entre las clases del sistema:

- **Docente**: contiene atributos como **ci**, **nombre**, **apellido**, **curso**, **grupo**, **turno**.
- **Licencia**: incluye **tipoLicencia**, **fechaInicio**, **fechaFin**.
- **ClaseLogica**: método **guardar()** que recibe datos y coordina la creación de objetos y su persistencia.
- **GestorDAO**: método **insertarDocenteYLicencia()** que ejecuta la inserción en la base de datos.
- **ConexionBD**: método **conectar()** que devuelve una conexión activa.

## 8. Prototipo en Figma



The image shows a Figma prototype of a web form titled "Gestión de Licencias Docentes". The form is contained within a light gray rounded rectangle. It features several input fields for personal and professional information, a dropdown menu for license type, and date pickers for start and end dates. At the bottom, there are three buttons: "Guardar" (light blue), "Limpiar" (light gray), and "Ver tabla" (blue). The Figma window's standard controls (minimize, maximize, close) are visible in the top right corner.

**Gestión de Licencias Docentes**

CI

Nombre

Apellido

Curso

Grupo

Turno

Tipo de licencia

Fecha inicio

Fecha fin

Se diseñó un prototipo funcional en Figma que representa la interfaz principal de la aplicación. El formulario incluye los siguientes campos:

**CI**

**Nombre**

**Apellido**

**Curso**

**Grupo**

**Turno**

**Tipo de licencia**

**Fecha de inicio**



## 9. Conclusiones

En esta tercera entrega se alcanzó un nivel completo de documentación y desarrollo. Se corrigieron las observaciones de la docente, incorporando los diagramas UML, el prototipo Figma y la presentación en formato APA. El proyecto demuestra una comprensión sólida de la programación orientada a objetos, la conexión con bases de datos y la organización del trabajo técnico individual. En futuras mejoras se busca ampliar la funcionalidad de búsqueda y actualización de licencias.

## 10. Referencias

- Bibliografía sobre Swing, JDBC, UML, ISO-9001, gestión educativa.

IA's: Chat GPT y Copilot