

Talento Futuro

# **Ejercicio Final de Módulo 2: Sistema de Gestión de Tareas**

Por Fabiola Díaz

Profesores:

Javier Venegas

Marco Ojeda

Enero 2025

## Situación Inicial

En el marco del desarrollo del módulo de especialización en Java, se solicitó implementar un sistema de gestión de tareas básico que permita a los usuarios crear, organizar y priorizar tareas. Este sistema debía abordar conceptos clave como estructuras de datos, programación funcional, algoritmos, y diseño orientado a objetos, aplicando principios como SOLID.

El sistema debía funcionar como una aplicación de consola en la que el usuario pudiera interactuar a través de un menú.

## Objetivo

El objetivo del proyecto es desarrollar un sistema que facilite la gestión de tareas mediante un diseño robusto, eficiente y escalable. Este sistema debe cumplir con los siguientes requisitos:

- Almacenar tareas con propiedades como título, descripción, prioridad y estado.
- Permitir operaciones básicas de agregar, editar, eliminar, listar y buscar tareas.
- Implementar características orientadas a objetos y programación funcional.
- Proveer una interfaz sencilla para que los usuarios interactúen con el sistema.

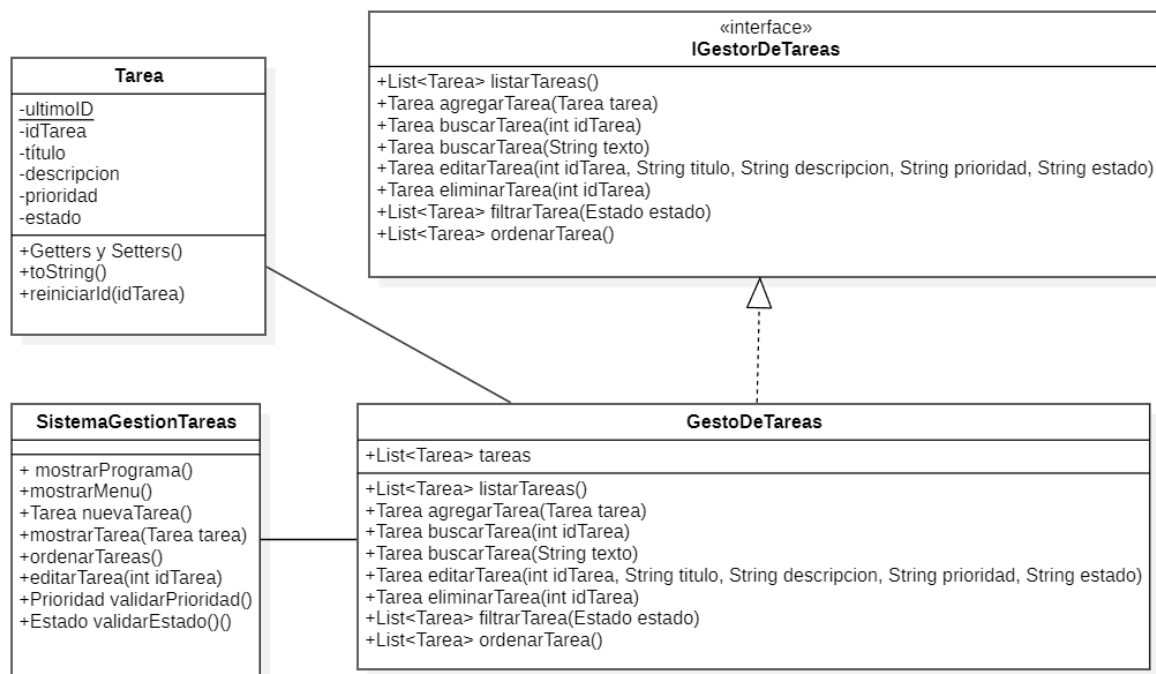
## Requerimientos

1. Almacenamiento y estructura de datos: Las tareas se almacenan en una lista enlazada, que garantiza flexibilidad y eficiencia en la manipulación de elementos.
2. Interacción con el usuario: Implementación de un menú en consola que facilite la ejecución de las operaciones principales.
3. Identificación de tareas: Cada tarea debe tener un identificador único, generado automáticamente para evitar duplicados.
4. Operaciones principales:
  - Agregar, editar y eliminar tareas.
  - Listar tareas organizadas por prioridad.
  - Buscar tareas por coincidencias en el título o descripción.

## Tecnologías Utilizadas

- Lenguaje de programación: Java
- Gestor de dependencias: Maven
- Pruebas unitarias: JUnit
- IDE: IntelliJ IDEA (puede usarse cualquier IDE compatible con Maven)

## Diagrama de clases



## Estructura del Código

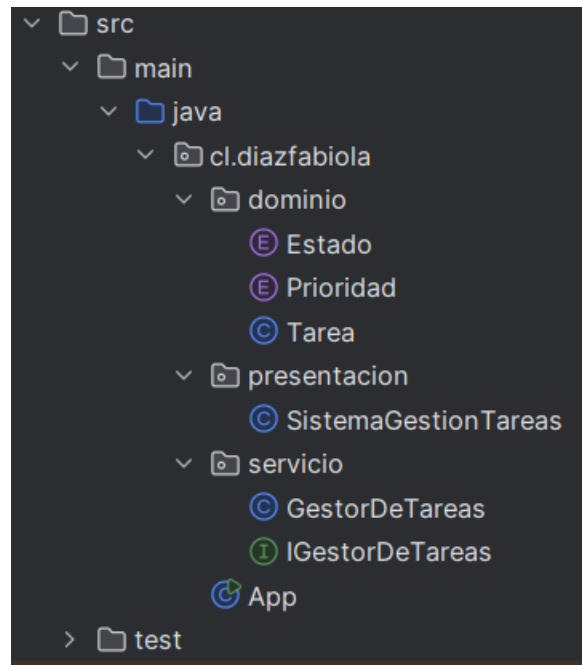
Diseño modular, la implementación está dividida en capas:

1. **Dominio:** Representado por la clase *Tarea*. Es el modelo principal que define los atributos y métodos de una tarea
2. **Servicio:** Representado por la Interface *IGestorDeTareas* que define las operaciones (como agregar, buscar, editar, eliminar, listar, filtrar y ordenar tareas) que se deben implementar para gestionar las tareas. Es la capa de servicio que abstrae la lógica de negocio.

También se encuentra la clase *GestorDeTareas* que implementa la interface *IGestorDeTareas* y contiene la lista de tareas. Maneja toda la lógica de negocio relacionada con la gestión de las tareas.

3. **Presentación:** Representada por *SistemaGestionTareas* que Interactúa con el usuario a través de la consola y utiliza una instancia de la clase *GestorDeTareas* para ejecutar las operaciones de gestión de tareas.

Este diseño facilita la escalabilidad y el mantenimiento.



## Consideraciones y Decisiones

### 1. **Generación de IDs automáticos:**

Decidí implementar un mecanismo de generación automática e incremental de identificadores únicos para cada tarea. Esto asegura que cada tarea sea única, incluso si el título o la descripción son iguales.

- Justificación: Al contar con un ID único, no es necesario realizar validaciones para detectar tareas repetidas por título o descripción.

### 2. **Búsqueda de tareas:**

Al buscar una tarea por título o descripción, se muestra la primera coincidencia que encuentra el sistema.

- Justificación: Este enfoque es simple y eficiente, dado que el objetivo es realizar búsquedas rápidas sin necesidad de recorrer toda la lista en busca de múltiples coincidencias.

## Demostración del programa

La aplicación cuenta con un menú de inicio, donde se muestra una bienvenida y un menú con las siguientes opciones:

```
-----
BIENVENIDO AL SISTEMA DE GESTIÓN DE TAREAS
-----
Ingrese una opción
1.- Agregar Tarea
2.- Editar Tarea
3.- Eliminar Tarea
4.- Listar Tarea
5.- Buscar Tarea
6.- Salir
|
```

### 1. Agregar Tarea:

La opción Agregar Tarea permite al usuario registrar una nueva tarea en el sistema mediante la introducción de sus datos principales: título, descripción, prioridad y estado. A continuación, se detalla el funcionamiento:

- **Validación de Campos Vacíos:** Si el título o la descripción se dejan en blanco, se muestra un mensaje al usuario indicando el error *“Error al agregar Tarea. Debe ingresar todos los campos.”* En este caso, no se crea la tarea, y el sistema regresa automáticamente al menú principal.
- **Gestión de Valores Inválidos o en Blanco:** Si los datos ingresados para prioridad o estado son incorrectos o se dejan en blanco, la tarea se crea con los siguientes valores por defecto (Prioridad: Media, Estado: Pendiente)
- **Creación de la Nueva Tarea:** Si todos los campos obligatorios se completan correctamente, la tarea se genera exitosamente con un identificador único (ID) auto incrementable, asegurando que no haya conflictos entre las tareas, incluso si otras tareas comparten el mismo título o descripción.
- **Retorno al Menú Principal:** Una vez completada la operación, el sistema notifica al usuario que la tarea fue creada exitosamente y regresa al menú para continuar con otras acciones.

```
Ingrese una opción
1.- Agregar Tarea
2.- Editar Tarea
3.- Eliminar Tarea
4.- Listar Tarea
5.- Buscar Tarea
6.- Salir
1
Ingresa los siguientes datos:
TÍTULO:
Tarea 1

DESCRIPCIÓN:
Descripción de la tarea 1

PRIORIDAD:
1. ALTA
2. MEDIA
3. BAJA
1

ESTADO:
1. PENDIENTE
2. EN_PROGRESO
3. COMPLETADA
1
La tarea con ID: 1 fue agregada con éxito
```

## 2. Editar Tarea:

La opción Editar Tarea permite al usuario modificar los datos de una tarea existente en el sistema. A continuación, se describe su funcionamiento:

- Ingreso del ID de la Tarea: El usuario debe ingresar el ID de la tarea que desea editar. Si el ID no corresponde a ninguna tarea registrada, se muestra un mensaje de error como *"La tarea con ID 2 no existe"* y el sistema retorna al menú principal.
- Visualización de Datos de la Tarea: Una vez que se verifica la existencia de la tarea, el sistema muestra todos los datos actuales de la tarea seleccionada.
- Submenú de Edición: Se presenta un submenú con las siguientes opciones:
  - Editar Título.
  - Editar Descripción.
  - Editar Prioridad.
  - Editar Estado.
  - Terminar Edición.
- Edición de Datos: El usuario puede seleccionar cualquiera de las opciones (1-4) para editar los campos correspondientes, no se permiten datos nulos o vacíos, en caso de ingresar un valor vacío o no válido se muestra un mensaje y volverá al submenú. Para finalizar la edición se debe seleccionar la opción 5. Una vez terminada la edición, el sistema notifica al usuario que los cambios han sido guardados exitosamente.

```
Ingrese una opción
1.- Agregar Tarea
2.- Editar Tarea
3.- Eliminar Tarea
4.- Listar Tarea
5.- Buscar Tarea
6.- Salir
2
Ingrese ID de la Tarea que quiere editar:
1
ID: 1
TITULO: Tarea 1
DESCRIPCIÓN: Esta es la descripción de la primera tarea
PRIORIDAD: ALTA
ESTADO: PENDIENTE
-----
¿Que quiere editar?
1.- Título
2.- Descripción
3.- Prioridad
4. Estado
5. Terminar Edición
|
```



### 3. Eliminar Tarea

Para eliminar una tarea se solicita el ID de la tarea, si la tarea existe se muestra un mensaje que la tarea fue eliminada, en caso que contrario se muestra un mensaje diciendo que la tarea no existe.

```
Ingrese una opción
1.- Agregar Tarea
2.- Editar Tarea
3.- Eliminar Tarea
4.- Listar Tarea
5.- Buscar Tarea
6.- Salir
3
Ingrese ID de la Tarea:
1
La tarea con ID 1 eliminada

Presione cualquier tecla para continuar...
```

### 4. Listar Tareas

Esta opción despliega un submenú para mostrar la lista de tareas según ID, filtrar por estado u ordenar por prioridad.

```
Ingrese una opción
1.- Agregar Tarea
2.- Editar Tarea
3.- Eliminar Tarea
4.- Listar Tarea
5.- Buscar Tarea
6.- Salir
4
1.- Mostrar Tareas por ID
2.- Filtrar por Estado
3.- Ordenar por Prioridad
|
```

Aquí se selecciona la opción 1, por lo que las tareas se muestran según su ID de forma ascendente.

```
1.- Mostrar Tareas por ID
2.- Filtrar por Estado
3.- Ordenar por Prioridad
1
ID: 1
TITULO: Primera Tarea
DESCRIPCIÓN: Esta es la descripci?n de la primera tarea
PRIORIDAD: ALTA
ESTADO: PENDIENTE
-----

ID: 2
TITULO: Segunda Tarea
DESCRIPCIÓN: Esta es la descripci?n de la segunda tarea
PRIORIDAD: MEDIA
ESTADO: EN_PROGRESO
-----

ID: 3
TITULO: Tercera tarea
DESCRIPCIÓN: Esta es la descripci?n de la tercera tarea
PRIORIDAD: BAJA
ESTADO: COMPLETADA
-----

Presione cualquier tecla para continuar...
```

Aquí se selecciona la opción 2. Por lo que se muestra la lista de tareas según el estado seleccionado. En este caso se selecciona opción 1 “PENDIENTE” y solo tenemos una tarea con este estado.

```
Ingresa una opción
1.- Agregar Tarea
2.- Editar Tarea
3.- Eliminar Tarea
4.- Listar Tarea
5.- Buscar Tarea
6.- Salir
4
1.- Mostrar Tareas por ID
2.- Filtrar por Estado
3.- Ordenar por Prioridad
2
Ingresa el estado
1. PENDIENTE
2. EN_PROGRESO
3. COMPLETADA
1
ID: 1
TITULO: Primera Tarea
DESCRIPCIÓN: Esta es la descripción de la primera tarea
PRIORIDAD: ALTA
ESTADO: PENDIENTE
-----

Presione cualquier tecla para continuar...
```

Por último, aquí se selecciona la opción 3 que muestras la Tareas según Orden de Prioridad (ALTA, MEDIA, BAJA)

```
4
1.- Mostrar Tareas por ID
2.- Filtrar por Estado
3.- Ordenar por Prioridad
3
ID: 1
TITULO: Primera Tarea
DESCRIPCIÓN: Esta es la descripci?n de la primera tarea
PRIORIDAD: ALTA
ESTADO: PENDIENTE
-----

ID: 2
TITULO: Segunda Tarea
DESCRIPCIÓN: Esta es la descripci?n de la segunda tarea
PRIORIDAD: MEDIA
ESTADO: EN_PROGRESO
-----

ID: 3
TITULO: Tercera tarea
DESCRIPCIÓN: Esta es la descripci?n de la tercera tarea
PRIORIDAD: BAJA
ESTADO: COMPLETADA
-----

Presione cualquier tecla para continuar...
```

## 5. Buscar Tarea:

El sistema solicita al usuario que ingrese un texto relacionado con el título o la descripción de la tarea que desea buscar. El sistema recorre las tareas registradas y busca coincidencias según el criterio ingresado.

En caso de no se encuentra ninguna coincidencia, se muestra el mensaje: *"Tarea no encontrada."* . Si hay varias tareas con el mismo título o descripción, el sistema devuelve la primera coincidencia encontrada.

```
Ingresa una opción
1.- Agregar Tarea
2.- Editar Tarea
3.- Eliminar Tarea
4.- Listar Tarea
5.- Buscar Tarea
6.- Salir
5
Ingresa título o descripción de la tarea:
primera tarea
ID: 1
TITULO: Primera Tarea
DESCRIPCIÓN: Esta es la descripción de la primera tarea
PRIORIDAD: ALTA
ESTADO: PENDIENTE
-----

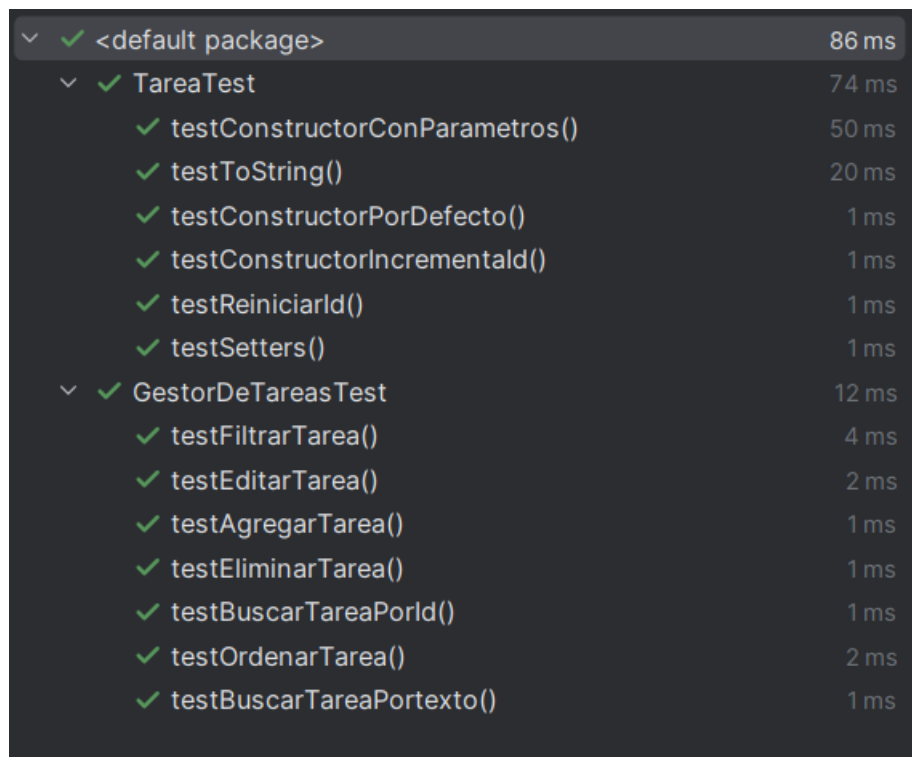
Presione cualquier tecla para continuar...
```

## Resumen de ejecución de pruebas unitarias

Se realizaron pruebas unitarias para evaluar el correcto funcionamiento de las clases y métodos implementados en el sistema de gestión de tareas.

En la primera imagen se detalla el conjunto de pruebas unitarias ejecutadas, clasificadas en dos grupos principales:

- TareaTest: Incluye pruebas de los métodos de la clase Tarea, verificando el comportamiento de los constructores, métodos toString, reinicio de ID, y los setters. Todas las pruebas de este grupo fueron exitosas.
- GestorDeTareasTest: Evalúa las funcionalidades del gestor de tareas, tales como agregar, editar, eliminar, buscar, ordenar tareas, y filtrar resultados. Todas las pruebas de este grupo también se ejecutaron correctamente.



✓ <default package>	86 ms
✓ TareaTest	74 ms
✓ testConstructorConParametros()	50 ms
✓ testToString()	20 ms
✓ testConstructorPorDefecto()	1 ms
✓ testConstructorIncrementalId()	1 ms
✓ testReiniciarId()	1 ms
✓ testSetters()	1 ms
✓ GestorDeTareasTest	12 ms
✓ testFiltrarTarea()	4 ms
✓ testEditarTarea()	2 ms
✓ testAgregarTarea()	1 ms
✓ testEliminarTarea()	1 ms
✓ testBuscarTareaPorId()	1 ms
✓ testOrdenarTarea()	2 ms
✓ testBuscarTareaPortexto()	1 ms

En la segunda imagen se presenta el análisis de cobertura del código, evidenciando los porcentajes alcanzados en diferentes niveles:

- Clases: Cobertura del 66% en general, con el dominio (Estado, Prioridad, y Tarea) y el servicio (GestorDeTareas) alcanzando una cobertura completa del 100%.
- Métodos: Se logró un 72% de cobertura, destacando la cobertura total (100%) en métodos de clases críticas como GestorDeTareas y las entidades del dominio.

Element ^	Class, %	Method, %	Line, %	Branch, %
✓ cl.diazfabiola	66% (4/6)	72% (27/37)	31% (62/200)	18% (16/88)
✓ dominio	100% (3/3)	94% (18/19)	95% (23/24)	100% (0/0)
Estado	100% (1/1)	100% (2/2)	100% (2/2)	100% (0/0)
Prioridad	100% (1/1)	100% (2/2)	100% (2/2)	100% (0/0)
Tarea	100% (1/1)	93% (14/15)	95% (19/20)	100% (0/0)
presentacion	0% (0/1)	0% (0/8)	0% (0/130)	0% (0/60)
SistemaGestionTareas	0% (0/1)	0% (0/8)	0% (0/130)	0% (0/60)
servicio	100% (1/1)	100% (9/9)	88% (39/44)	57% (16/28)
GestorDeTareas	100% (1/1)	100% (9/9)	88% (39/44)	57% (16/28)
IGestorDeTareas	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
App	0% (0/1)	0% (0/1)	0% (0/2)	100% (0/0)

## **Conclusión**

En general, el trabajo realizado cumple con los requerimientos establecidos. Se logró la implementación de las estructuras de datos, uso de programación funcional y aplicación de los principios de diseño orientado a objetos. Además, las pruebas unitarias alcanzaron una buena cobertura, lo que permitió validar el funcionamiento general del sistema. Sin embargo, durante el desarrollo y las pruebas, se pudo identificar errores y áreas de mejora. Los test resultaron ser una herramienta clave para detectar problemas y garantizar la calidad del código. Este proceso me permitió darme cuenta sobre posibles optimizaciones y nuevas implementaciones que podrían hacerse para mejorar el proyecto que permiten que sea más eficiente y mantenible.