



UNIVERSIDAD DE LAS FUERZAS ARMADAS- ESPE

PROGRAMACIÓN ORIENTADA A OBJETIVOS

Tarea:

Sistema de Gestión de Proyectos de TIC

Estudiante

Juan Fabricio Pilataxi Tipán

Janina Viviana Puga Monserrat

2024

Clases Principales:

Proyecto:

Atributos:

- id (int): Identificador único del proyecto
- nombre (str): Nombre del proyecto
- descripcion (str): Descripción detallada
- fechaInicio (date): Fecha de inicio
- fechaFin (date): Fecha estimada de finalización
- estado (str): Estado actual del proyecto (en progreso, completado, etc.)

Métodos:

- agregarTarea(Tarea tarea): Agrega una tarea al proyecto
- eliminarTarea(Tarea tarea): Elimina una tarea del proyecto
- cambiarEstado(str nuevoEstado): Cambia el estado del proyecto

Tarea:

Atributos:

- id (int): Identificador único de la tarea
- descripcion (str): Descripción de la tarea
- asignadoA (Usuario): Usuario al que está asignada la tarea
- fechaLimite (date): Fecha límite para completar la tarea
- estado (str): Estado actual de la tarea (pendiente, en progreso, completada)

Métodos:

- marcarCompletada(): Cambia el estado de la tarea a completada

Usuario:

Atributos:

- id (int): Identificador único del usuario
- nombre (str): Nombre del usuario
- correo (str): Correo electrónico
- rol (str): Rol del usuario (administrador, miembro del equipo)

Métodos:

- asignarTarea(Tarea tarea): Asigna una tarea al usuario

Funcionalidades Detalladas

Registrar Proyectos:

- El usuario crea un nuevo objeto de tipo Proyecto y proporciona los datos necesarios (nombre, descripción, fechas, etc.).
- El sistema asigna un ID único al proyecto y lo almacena en una lista o base de datos.

Asignar Tareas:

- El usuario crea una nueva tarea y la asocia a un proyecto existente.
- El sistema asigna un ID único a la tarea y la agrega a la lista de tareas del proyecto.
- El usuario puede asignar la tarea a un usuario específico.

Mostrar Estado:

- El sistema puede mostrar el estado general de un proyecto (porcentaje de tareas completadas, tareas pendientes, etc.).
- También puede mostrar el estado de cada tarea individual, incluyendo el usuario asignado y la fecha límite.

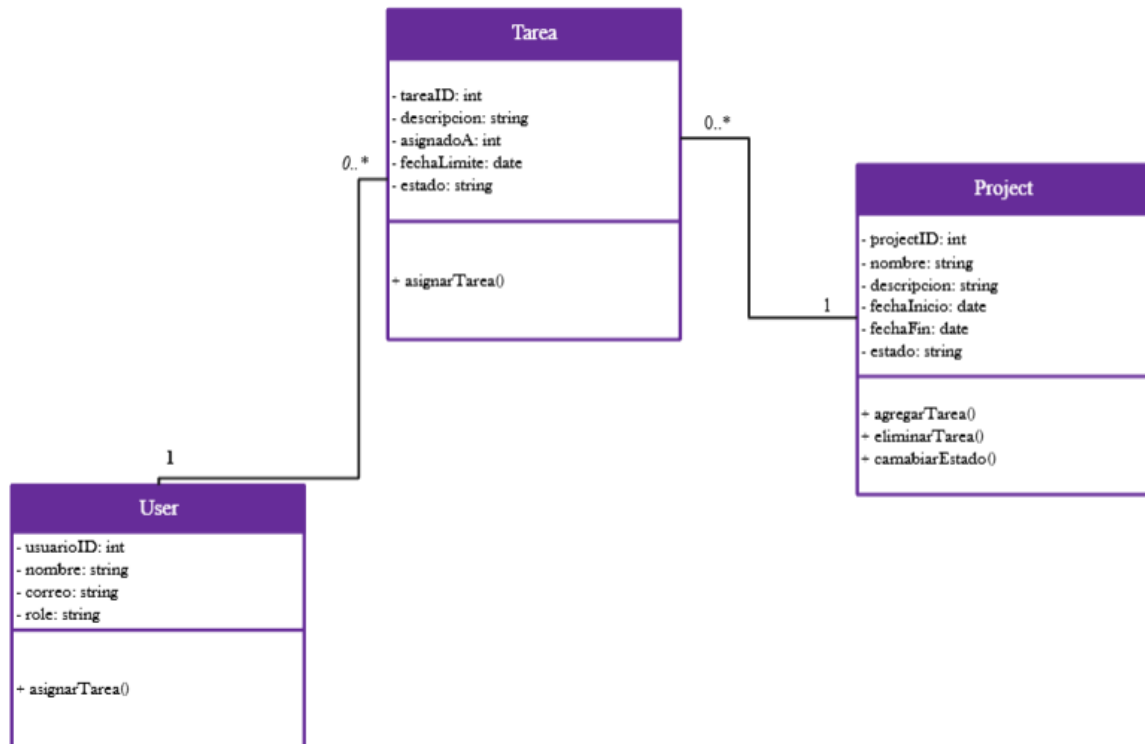
Relaciones entre Clases

- **Proyecto - Tarea:**
 - **Relación:** Un proyecto **tiene muchas** tareas. Esta es una relación de **composición**, ya que las tareas no existen independientemente del proyecto y son destruidas cuando el proyecto es eliminado.
 - **Representación en UML:** Una línea sólida con un rombo en el extremo de la clase Proyecto, apuntando a la clase Tarea.
- **Tarea - Usuario:**
 - **Relación:** Una tarea es **asignada a un** usuario. Esta es una relación de **asociación**, ya que una tarea puede existir sin un usuario asignado, y un usuario puede tener muchas tareas asignadas.
 - **Representación en UML:** Una línea sólida simple entre las clases Tarea y Usuario.

Explicación del Diagrama:

- **Clases:** Cada clase se representa como un rectángulo con el nombre de la clase en la parte superior.
- **Atributos:** Los atributos de cada clase se enumeran dentro de la clase.
- **Métodos:** Los métodos de cada clase se enumeran debajo de los atributos.
- **Relaciones:**
 - La línea sólida con un rombo representa la composición entre Proyecto y Tarea.
 - La línea sólida simple representa la asociación entre Tarea y Usuario.

UML



Código

```
import java.util.ArrayList;
import java.util.Date;

public class Proyecto {
    private int id;
    private String nombre;
    private String descripcion;
    private Date fechaInicio;
    private Date fechaFin;
    private String estado;
    private ArrayList<Tarea> tareas = new ArrayList<>();

    // Constructor y getters/setters

    public void agregarTarea(Tarea tarea) {
        tareas.add(tarea);
    }

    public void eliminarTarea(Tarea tarea) {
        tareas.remove(tarea);
    }

    public void cambiarEstado(String nuevoEstado) {
        this.estado = nuevoEstado;
    }
}

public class Tarea {
    private int id;
    private String descripcion;
```

```

    private Usuario asignadoA;
    private Date fechaLimite;
    private String estado;

    // Constructor y getters/setters

    public void marcarCompletada() {
        this.estado = "Completada";
    }
}

public class Usuario {
    private int id;
    private String nombre;
    private String correo;
    private String rol;

    // Constructor y getters/setters

    public void asignarTarea(Tarea tarea) {
        // Lógica para asignar la tarea al usuario, podría implicar
        actualizar la tarea
    }
}

```

Explicación:

- **Clases:** Se han definido las clases Proyecto, Tarea y Usuario según la especificación.
- **Atributos:** Cada clase tiene los atributos correspondientes: identificador, nombre, descripción, fechas, estado, etc.
- **Métodos:** Se han implementado los métodos necesarios para cada clase: agregar tareas, eliminar tareas, cambiar estados, asignar tareas, etc.
- **ArrayList:** Se utiliza un ArrayList para almacenar las tareas asociadas a un proyecto.
- **Constructor y getters/setters:** Aunque no se muestran explícitamente en el código, es recomendable incluir un constructor para inicializar los objetos y métodos getters y setters para acceder y modificar los atributos de las clases.