

Actividad Experimental N°1 Primer Parcial

Nombre: Pablo Camacho

NRC: 1323

Fecha: 08/12/2024

Asignatura: Programación Orientada a Objetos

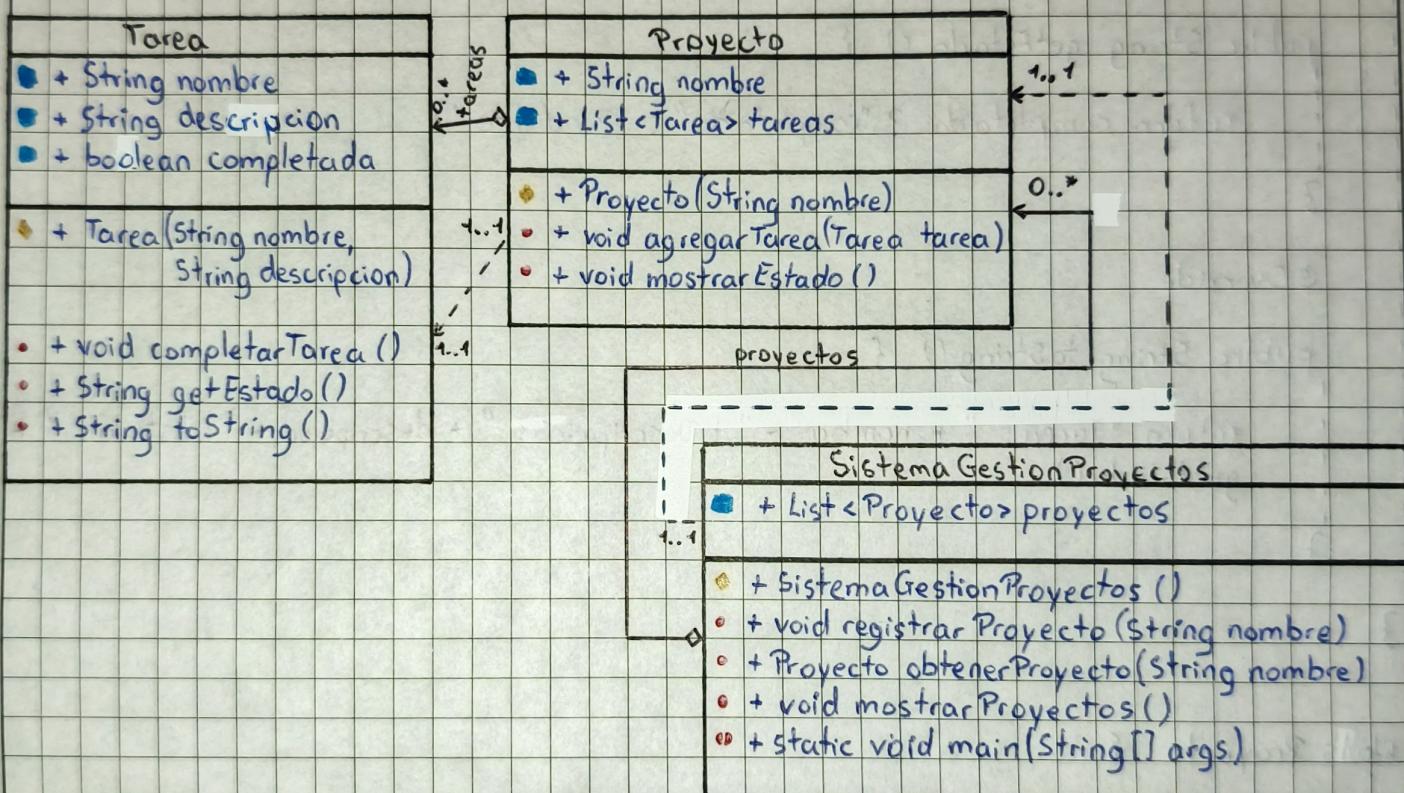
Tema de la Actividad: Sistema de Gestión de Proyectos de TIC

Tipo de Actividad: Desarrollo Práctico

Descripción de la Actividad:

Implemente un sistema de gestión de proyectos utilizando POO. Incluya las funcionalidades de registrar proyectos, asignar tareas y mostrar su estado.

UML



Código

```
package com.mycompany.sistemagestionproyectos;
```

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

```
import java.util.List;
```

```
class Tarea {
```

```
    public String nombre;
```

```
    public String descripcion;
```

```
public boolean completada;  
public Tarea(String nombre, String descripción) {  
    this.nombre = nombre;  
    this.descripcion = descripción;  
    this.completada = false;  
}  
  
public void completarTarea() {  
    this.completada = true;  
}  
  
public String getEstado() {  
    return completada ? "completada" : "Pendiente";  
}  
  
@Override  
public String toString() {  
    return "Tarea: " + nombre + " | Descripción: " + descripción + " | Estado: " +  
        getEstado();  
}  
  
}  
  
class Proyecto {  
    public String nombre;  
    public List<Tarea> tareas;  
    public Proyecto(String nombre) {  
        this.nombre = nombre;  
        this.tareas = new ArrayList<>();  
    }  
    public void agregarTarea(Tarea tarea) {  
        tareas.add(tarea);  
    }  
}
```

```
public void mostrarEstado () {
    System.out.println ("Proyecto: " + nombre);
    for (Tarea tarea : tareas) {
        System.out.println (tarea);
    }
}

public class SistemaGestionProyectos {
    public List<Proyecto> proyectos;
    public SistemaGestionProyectos () {
        this.proyectos = new ArrayList<> ();
    }
    public void registrarProyecto (String nombre) {
        proyectos.add (new Proyecto (nombre));
        System.out.println ("Proyecto " + nombre + " registrado con exito.");
    }
    public Proyecto obtenerProyecto (String nombre) {
        for (Proyecto proyecto : proyectos) {
            if (proyecto.nombre.equals (nombre)) {
                return proyecto;
            }
        }
        return null;
    }
    public void mostrarProyectos () {
        System.out.println ("Lista de proyectos:");
        for (Proyecto proyecto : proyectos) {
```

```
System.out.println("- " + proyecto.nombre);
}

}

public static void main(String[] args) {
    SistemaGestionProyectos sistema = new SistemaGestionProyectos();

    sistema.registrarProyecto("Proyecto 1");
    sistema.registrarProyecto("Proyecto 2");

    Proyecto proyecto1 = sistema.obtenerProyecto("Proyecto 1");
    if (proyecto1 != null) {
        proyecto1.agregarTarea(new Tarea("Diseño", "Crear el diseño inicial del sistema"));

        proyecto1.agregarTarea(new Tarea("Implementación", "Desarrollar las funcionalidades principales"));
    }

    Proyecto proyecto2 = sistema.obtenerProyecto("Proyecto 2");
    if (proyecto2 != null) {
        proyecto2.agregarTarea(new Tarea("Investigación", "Investigar tecnologías para el proyecto"));
    }

    sistema.mostrarProyectos();
    if (proyecto1 != null) proyecto1.mostrarEstado();
    if (proyecto2 != null) proyecto2.mostrarEstado();
}
```

Ejecución

Proyecto 'Proyecto 1' registrado con éxito

Proyecto 'Proyecto 2' registrado con éxito

Lista de proyectos:

- Proyecto 1

- Proyecto 2

Proyecto: Proyecto 1

Tarea: Diseño | Descripción: Crear el diseño inicial del Sistema | Estado: Pendiente

Tarea: Implementación | Descripción: Desarrollar las funcionalidades principales | Estado: Pendiente

Proyecto: Proyecto 2

Tarea: Investigación | Descripción: Investigar tecnologías para el proyecto | Estado: Pendiente



Universidad de las Fuerzas Armadas-ESPE

Programación orientada a objetos

Actividad Experimental N°1 Primer Parcial

Sistema de Gestión de Proyectos de TIC

Nombre:

Camacho Bravo Pablo Guber

NRC:

1323

Fecha de Entrega:

08/12/2024

Nombre del docente:

Jaramillo Montaño Luis Enrique

Tipo de Actividad

Desarrollo Practico

Desarrollo Práctico

Descripción de la Actividad: Implemente un sistema de gestión de proyectos utilizando POO. Incluya las funcionalidades de registrar proyectos, asignar tareas y mostrar su estado.

Informe con Evidencias

El desarrollo del sistema de gestión de proyectos en Java se realizó siguiendo un enfoque estructurado basado en los principios de (POO). A continuación, describo el proceso llevado a cabo.

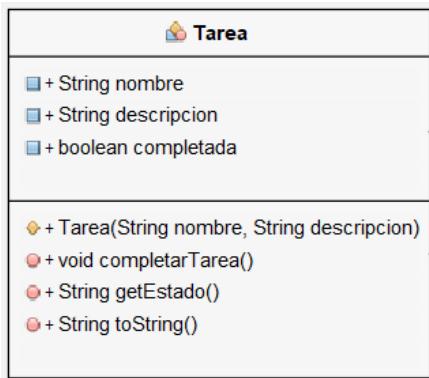
1. El primer paso fue identificar los requisitos funcionales del sistema. Se definieron las siguientes funcionalidades principales:
 - Registrar proyectos.
 - Asignar tareas a los proyectos.
 - Mostrar el estado de los proyectos y sus tareas.

Además, considere la necesidad de que el sistema sea modular, escalable y fácil de mantener, lo que llevó a la elección de POO como paradigma de desarrollo. Este enfoque permite modelar los elementos del sistema como objetos con atributos y comportamientos específicos.

2. El diseño del sistema se basó en la identificación de las entidades principales y sus relaciones. Se definieron las siguientes clases:

- 1) **Clase Tarea:**

- Representa una tarea dentro de un proyecto.
- Contiene atributos como el nombre, descripción y estado de la tarea (pendiente o completada).
- Métodos principales:
 - `completarTarea()`: Cambia el estado de la tarea a completada.
 - `getEstado()`: Devuelve el estado actual de la tarea.
 - `toString()`: Proporciona una representación legible de la tarea.



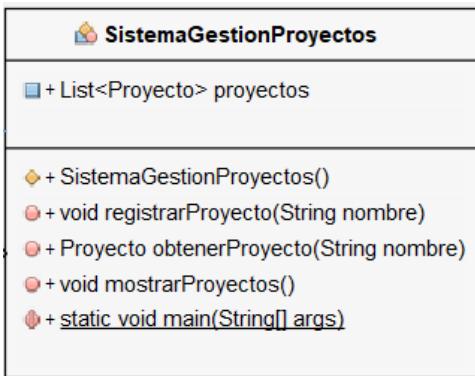
2) Clase Proyecto:

- Representa un proyecto que contiene una lista de tareas.
- Contiene atributos como el nombre del proyecto y una lista de tareas asociadas.
- Métodos principales:
 - agregarTarea(Tarea tarea): Permite agregar una nueva tarea al proyecto.
 - mostrarEstado(): Muestra el estado de todas las tareas del proyecto.



3) Clase SistemaGestionProyectos:

- Gestiona múltiples proyectos y actúa como el controlador principal del sistema.
- Contiene una lista de proyectos registrados.
- Métodos principales:
 - registrarProyecto(String nombre): Permite registrar un nuevo proyecto.
 - obtenerProyecto(String nombre): Busca y devuelve un proyecto por su nombre.
 - mostrarProyectos(): Lista todos los proyectos registrados en el sistema.



3. La implementación se realizó en el lenguaje de programación Java, siguiendo las prácticas de POO. Utilice una estructura modular para garantizar la separación de responsabilidades entre las clases.

1) Clase Tarea:

- Implemente los atributos y métodos necesarios para gestionar el estado de las tareas.
- Utilice el método `toString()` para facilitar la visualización de la información de las tareas.

```

class Tarea {
    public String nombre;
    public String descripcion;
    public boolean completada;

    public Tarea(String nombre, String descripcion) {
        this.nombre = nombre;
        this.descripcion = descripcion;
        this.completada = false;
    }

    public void completarTarea() {
        this.completada = true;
    }

    public String getEstado() {
        return completada ? "Completada" : "Pendiente";
    }

    @Override
    public String toString() {
        return "Tarea: " + nombre + " | Descripcion: " + descripcion + " | Estado: " + getEstado();
    }
}

```

2) Clase Proyecto:

- Implemente una lista de tareas como atributo principal.
- Desarrolle métodos para agregar tareas y mostrar el estado del proyecto.

```

class Proyecto {
    public String nombre;
    public List<Tarea> tareas;

    public Proyecto(String nombre) {
        this.nombre = nombre;
        this.tareas = new ArrayList<>();
    }

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

    public void mostrarEstado() {
        System.out.println("Proyecto: " + nombre);
        for (Tarea tarea : tareas) {
            System.out.println(tarea);
        }
    }
}

```

3) Clase SistemaGestionProyectos:

- Implemente una lista de proyectos como atributo principal.
- Desarrolle métodos para registrar proyectos, buscar proyectos por nombre y mostrar la lista de proyectos.

```

public class SistemaGestionProyectos {
    public List<Proyecto> proyectos;

    public SistemaGestionProyectos() {
        this.proyectos = new ArrayList<>();
    }

    public void registrarProyecto(String nombre) {
        proyectos.add(new Proyecto(nombre));
        System.out.println("Proyecto '" + nombre + "' registrado con éxito.");
    }

    public Proyecto obtenerProyecto(String nombre) {
        for (Proyecto proyecto : proyectos) {
            if (proyecto.nombre.equals(nombre)) {
                return proyecto;
            }
        }
        return null;
    }

    public void mostrarProyectos() {
        System.out.println("Lista de proyectos:");
        for (Proyecto proyecto : proyectos) {
            System.out.println("- " + proyecto.nombre);
        }
    }
}

```

4) Clase main:

- Cree un método principal para simular el uso del sistema.
- Incluí ejemplos de registro de proyectos, asignación de tareas y visualización del estado de los proyectos.

```

public static void main(String[] args) {
    SistemaGestionProyectos sistema = new SistemaGestionProyectos();

    sistema.registrarProyecto("Proyecto 1");
    sistema.registrarProyecto("Proyecto 2");

    Proyecto proyecto1 = sistema.obtenerProyecto("Proyecto 1");
    if (proyecto1 != null) {
        proyecto1.agregarTarea(new Tarea("Diseño", "Crear el diseño inicial del sistema"));
        proyecto1.agregarTarea(new Tarea("Implementacion", "Desarrollar las funcionalidades principales"));
    }

    Proyecto proyecto2 = sistema.obtenerProyecto("Proyecto 2");
    if (proyecto2 != null) {
        proyecto2.agregarTarea(new Tarea("Investigacion", "Investigar tecnologias para el proyecto"));
    }

    sistema.mostrarProyectos();
    if (proyecto1 != null) proyecto1.mostrarEstado();
    if (proyecto2 != null) proyecto2.mostrarEstado();
}
}

```

4. Realice pruebas funcionales para verificar que el sistema cumpliera con los requisitos establecidos. Las pruebas incluyeron:

- Registro de proyectos con nombres únicos.
- Asignación de múltiples tareas a un proyecto.
- Visualización del estado de los proyectos y sus tareas.

Salida generada durante las pruebas:

```

[exec:3.1.0:exec (default-cli) @ SistemaGestionProyectos ---]
  Proyecto 'Proyecto 1' registrado con exito.
  Proyecto 'Proyecto 2' registrado con exito.
  Lista de proyectos:
  - Proyecto 1
  - Proyecto 2
  Proyecto: Proyecto 1
  Tarea: Diseño | Descripcion: Crear el diseño inicial del sistema | Estado: Pendiente
  Tarea: Implementacion | Descripcion: Desarrollar las funcionalidades principales | Estado: Pendiente
  Proyecto: Proyecto 2
  Tarea: Investigacion | Descripcion: Investigar tecnologias para el proyecto | Estado: Pendiente
  -----
  BUILD SUCCESS
  -----
  Total time: 1.545 s
  Finished at: 2024-12-08T17:00:51-05:00
  -----

```