



UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE  
Programación Orientada a Objetos

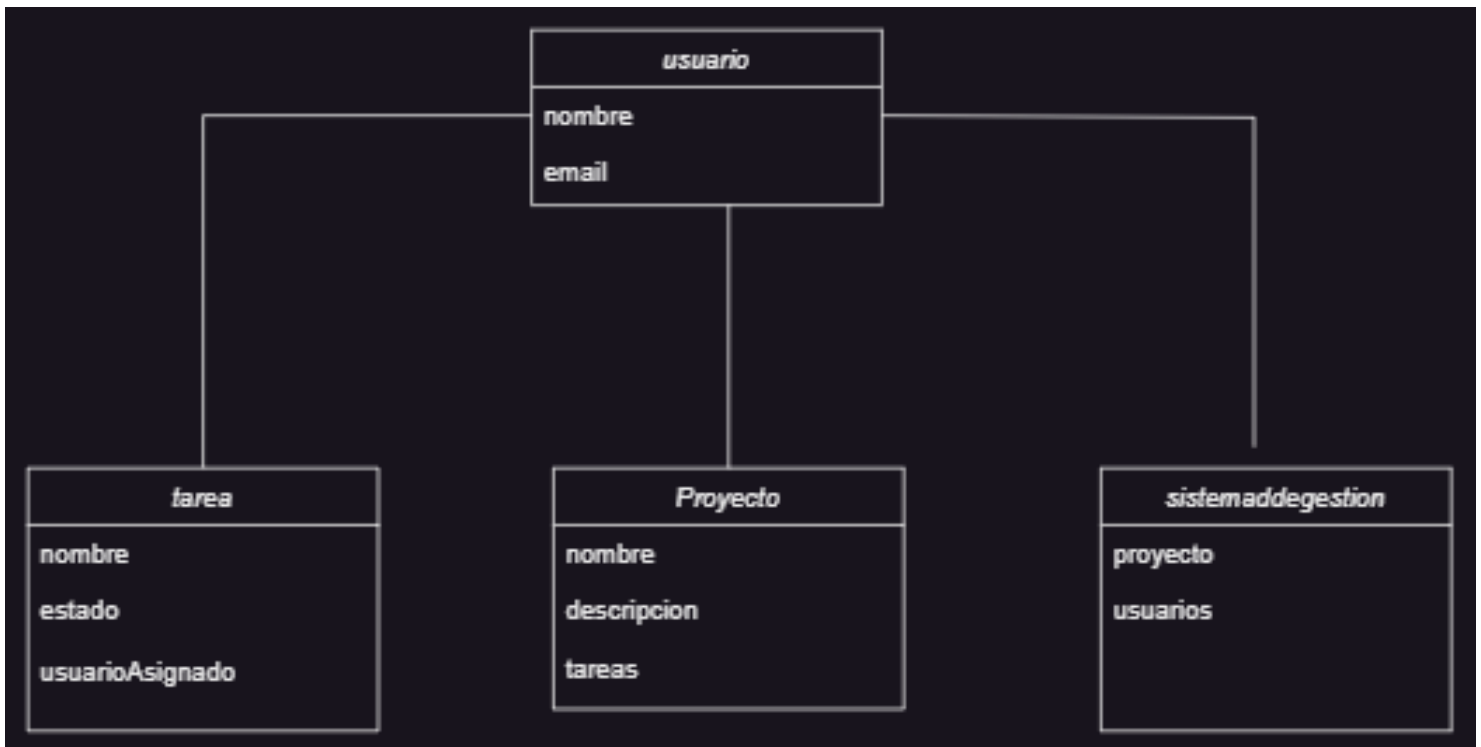
<b>APELLIDOS:</b> TIPANTIZA TIPANTIZA	<b>NRC:</b> 1322
<b>NOMBRES:</b> JEFFERSON ADRIAN	<b>FECHA DE ENTREGA:</b> 8/12/2024

**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 del proyecto**



Empezamos creando el proyecto lo cual tiene como atributos o datos.

- Nombres, Descripción y tareas.

Creamos una clase usuario.

- Sus atributos son datos: nombre y email.

Se crea la clase tarea.

- Con los datos: nombre, estado y usuarioAsignado. Lo cual nos permite llevar un control de tareas para cada usuario.

La clase Sistema de gestión

- Lleva por dato: proyecto y usuario.



## UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

### Programación Orientada a Objetos

- **Código**

Crear un nuevo proyecto en NetBeans

Link de repositorio de GitHub:

<https://github.com/JEFFERSON712/GestionProyectos>

Implementamos las clases

```
public class Usuario {  
    private String nombre;  
    private String email;  
  
    public Usuario(String nombre, String email) {  
        this.nombre = nombre;  
        this.email = email;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public String getEmail() {  
        return email;  
    }  
  
    public void setEmail(String email) {  
        this.email = email;  
    }  
  
    @Override  
    public String toString() {  
        return "Usuario{" +  
            "nombre='" + nombre + '\'' +  
            ", email='" + email + '\'' +  
            '}';  
    }  
}
```



# UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

## Programación Orientada a Objetos

```
public class Tarea {
    private String nombre;
    private String estado; // Ejemplo: "Pendiente", "En Progreso", "Completada"
    private Usuario usuarioAsignado;

    public Tarea(String nombre) {
        this.nombre = nombre;
        this.estado = "Pendiente";
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getEstado() {
        return estado;
    }

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

    public Usuario getUsuarioAsignado() {
        return usuarioAsignado;
    }

    public void asignarUsuario(Usuario usuario) {
        this.usuarioAsignado = usuario;
    }

    @Override
    public String toString() {
        return "Tarea(" +
            "nombre=" + nombre + '\'' +
            ", usuarioAsignado=" + (usuarioAsignado != null ? usuarioAsignado.getNombre() : "Ninguno") +
            ')';
    }
}
```



# UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

## Programación Orientada a Objetos

```
Usuario.java × Tarea.java × Proyecto.java × SistemaGestion.java × Main.java ×

import java.util.ArrayList;
import java.util.List;

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

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

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

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

    public List<Tarea> getTareas() {
        return tareas;
    }

    @Override
    public String toString() {
        return "Proyecto{" +
            "nombre='" + nombre + '\'' +
            ", descripcion='" + descripcion + '\'' +
            ", tareas=" + tareas +
            '}';
    }
}
```



UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE  
Programación Orientada a Objetos

```
Usuario.java × Tarea.java × Proyecto.java × SistemaGestion.java × Main.java ×

import java.util.ArrayList;
import java.util.List;

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

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

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

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

    public List<Tarea> getTareas() {
        return tareas;
    }

    @Override
    public String toString() {
        return "Proyecto{" +
            "nombre='" + nombre + '\'' +
            ", descripcion='" + descripcion + '\'' +
            ", tareas=" + tareas +
            '}';
    }
}
```



# UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

## Programación Orientada a Objetos

```
Usuario.java x Tarea.java x Proyecto.java x SistemaGestion.java x Main.java x

import java.util.ArrayList;
import java.util.List;

public class SistemaGestion {
    private final List<Proyecto> proyectos;
    private final List<Usuario> usuarios;

    public SistemaGestion() {
        proyectos = new ArrayList<>();
        usuarios = new ArrayList<>();
    }

    public void agregarUsuario(Usuario usuario) {
        usuarios.add(usuario);
    }

    public void agregarProyecto(Proyecto proyecto) {
        proyectos.add(proyecto);
    }

    public List<Proyecto> getProyectos() {
        return proyectos;
    }

    public List<Usuario> getUsuarios() {
        return usuarios;
    }

    public Proyecto buscarProyectoPorNombre(String nombre) {
        for (Proyecto proyecto : proyectos) {
            if (proyecto.getNombre().equalsIgnoreCase(nombre)) {
                return proyecto;
            }
        }
        return null;
    }

    public Usuario buscarUsuarioPorNombre(String nombre) {
        for (Usuario usuario : usuarios) {
            if (usuario.getNombre().equalsIgnoreCase(nombre)) {
                return usuario;
            }
        }
        return null;
    }
}
```



# UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

## Programación Orientada a Objetos

```
Usuario.java x Tarea.java x Proyecto.java x SistemaGestion.java x Main.java x
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        SistemaGestion sistema = new SistemaGestion();
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\n--- Sistema de Gestión de Proyectos ---");
            System.out.println("1. Crear usuario");
            System.out.println("2. Crear proyecto");
            System.out.println("3. Agregar tarea a proyecto");
            System.out.println("4. Listar proyectos");
            System.out.println("5. Salir");
            System.out.print("Selecciona una opción: ");
            int opcion = scanner.nextInt();
            scanner.nextLine(); // Consumir la línea

            switch (opcion) {
                case 1 -> {
                    System.out.print("Nombre del usuario: ");
                    String nombreUsuario = scanner.nextLine();
                    System.out.print("Email del usuario: ");
                    String email = scanner.nextLine();
                    sistema.agregarUsuario(new Usuario(nombreUsuario, email));
                }
                case 2 -> {
                    System.out.print("Nombre del proyecto: ");
                    String nombreProyecto = scanner.nextLine();
                    System.out.print("Descripción del proyecto: ");
                    String descripcion = scanner.nextLine();
                    sistema.agregarProyecto(new Proyecto(nombreProyecto, descripcion));
                }
                case 3 -> {
                    System.out.print("Nombre del proyecto: ");
                    String proyectoTarea = scanner.nextLine();
                    Proyecto proyecto = sistema.buscarProyectoPorNombre(proyectoTarea);
                    if (proyecto != null) {
                        System.out.print("Nombre de la tarea: ");
                        String nombreTarea = scanner.nextLine();
                        Tarea tarea = new Tarea(nombreTarea);
                        proyecto.agregarTarea(tarea);
                        System.out.println("Tarea agregada.");
                    } else {
                        System.out.println("Proyecto no encontrado.");
                    }
                }
                case 4 -> {
                    for (Proyecto p : sistema.getProyectos()) {
                        System.out.println(p);
                    }
                }
                case 5 -> {
                    System.out.println("Saliendo...");
                    scanner.close();
                    return;
                }
                default -> System.out.println("Opción inválida.");
            }
        }
    }
}
```



## Informe

### Estructura del Sistema

El sistema está compuesto por las siguientes clases:

#### 2.1. Clase Usuario

Representa a un usuario que puede ser asignado a una tarea.

##### Atributos:

- **nombre:** Nombre del usuario.
- **email:** Dirección de correo electrónico del usuario.

##### Métodos principales:

- Constructor para inicializar nombre y email.
- Métodos getter y setter.
- Método toString para representar al usuario como texto.

#### Clase Tarea

- Representa una tarea dentro de un proyecto.
- Atributos:
  - **nombre:** Nombre de la tarea.
  - **estado:** Estado de la tarea (por ejemplo, *Pendiente*, *En Progreso*, *Completada*).
  - **usuarioAsignado:** Referencia al usuario asignado a la tarea.

#### Clase Proyecto

- Representa un proyecto que agrupa múltiples tareas.
- Atributos:
  - **nombre:** Nombre del proyecto.
  - **descripcion:** Breve descripción del proyecto.
  - **tareas:** Lista de tareas asociadas al proyecto.
- Métodos principales:
  - Constructor para inicializar el proyecto.

#### Clase SistemaGestion





UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE  
Programación Orientada a Objetos

- **Es el núcleo del sistema y gestiona los proyectos y usuarios.**
- **Atributos:**
  - **proyectos:** Lista de proyectos registrados en el sistema.
  - **usuarios:** Lista de usuarios registrados en el sistema.
- **Métodos principales:**
  - **Métodos para agregar usuarios y proyectos.**
  - **Métodos para buscar usuarios o proyectos por nombre.**
  - **Métodos getter para acceder a las listas de usuarios y proyectos.**

**Clase Main**

- **Es el punto de entrada del programa.**