Maria Clara Aponte 202320052

Daniel Vergara 202320392

**Documento de Análisis y Sustento: Sistema de Recomendación de Caminos de Aprendizaje**

**Diagrama de clases de la interfaz gráfica:**

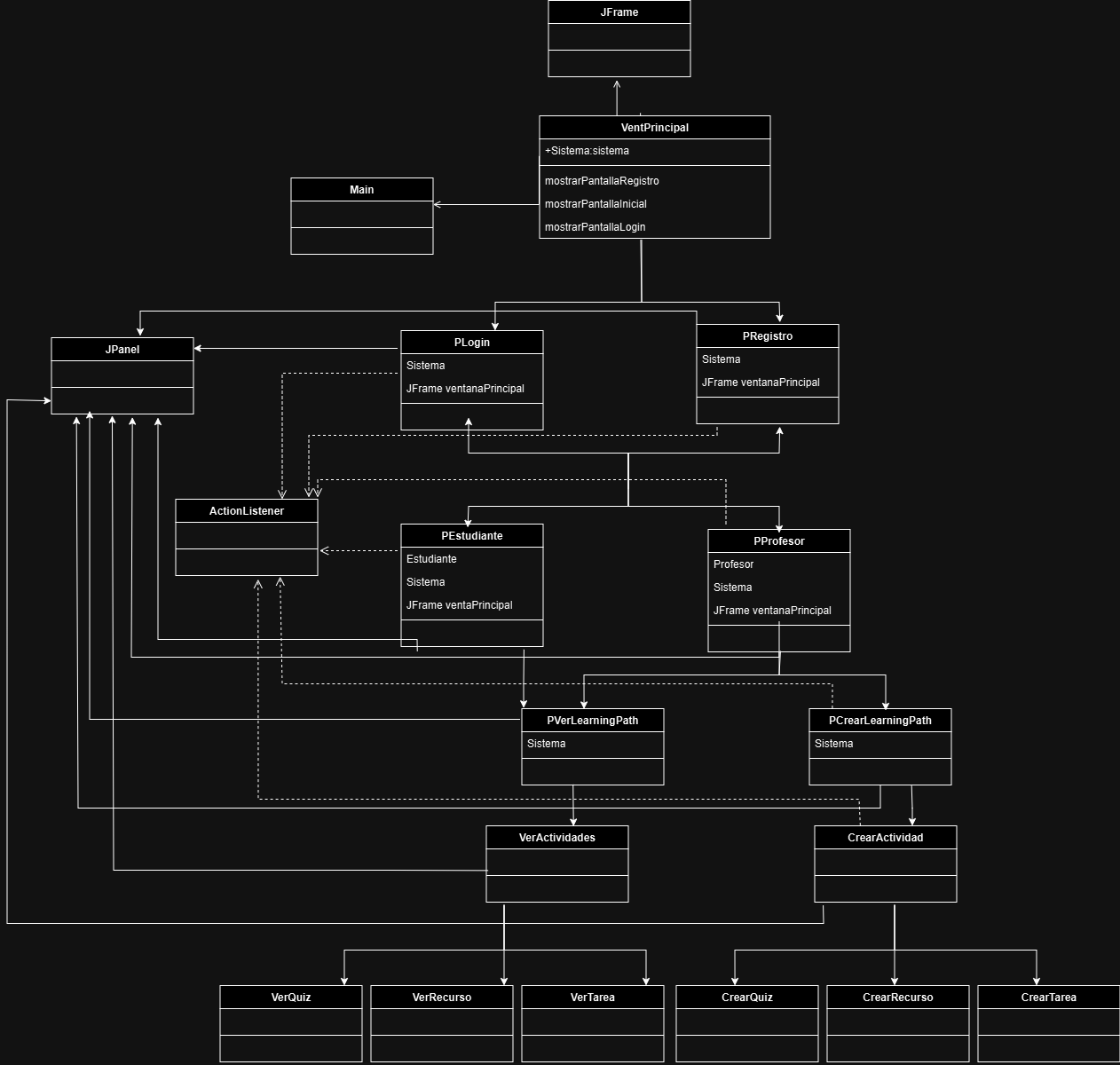


Diagrama de clases completo:

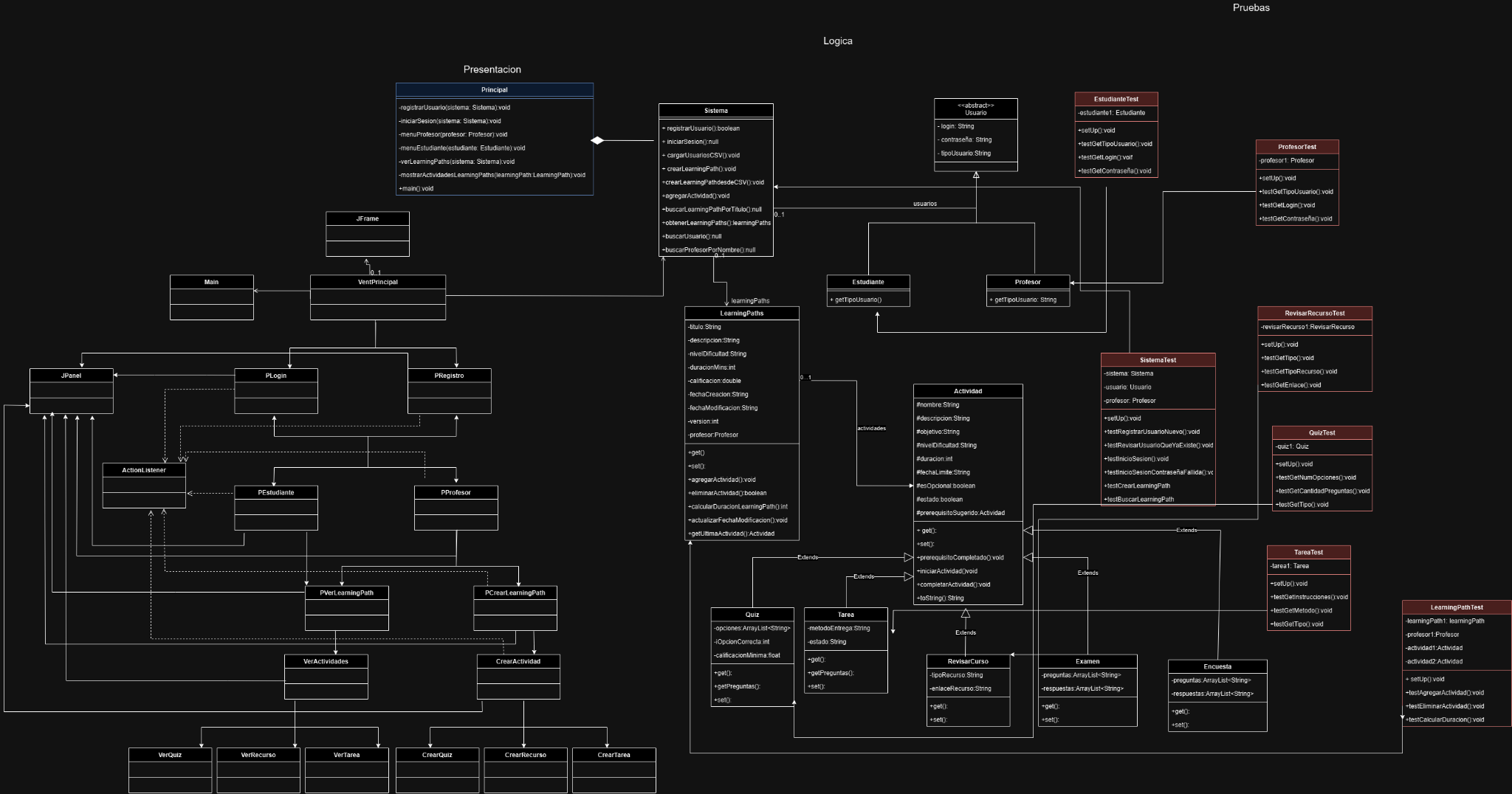
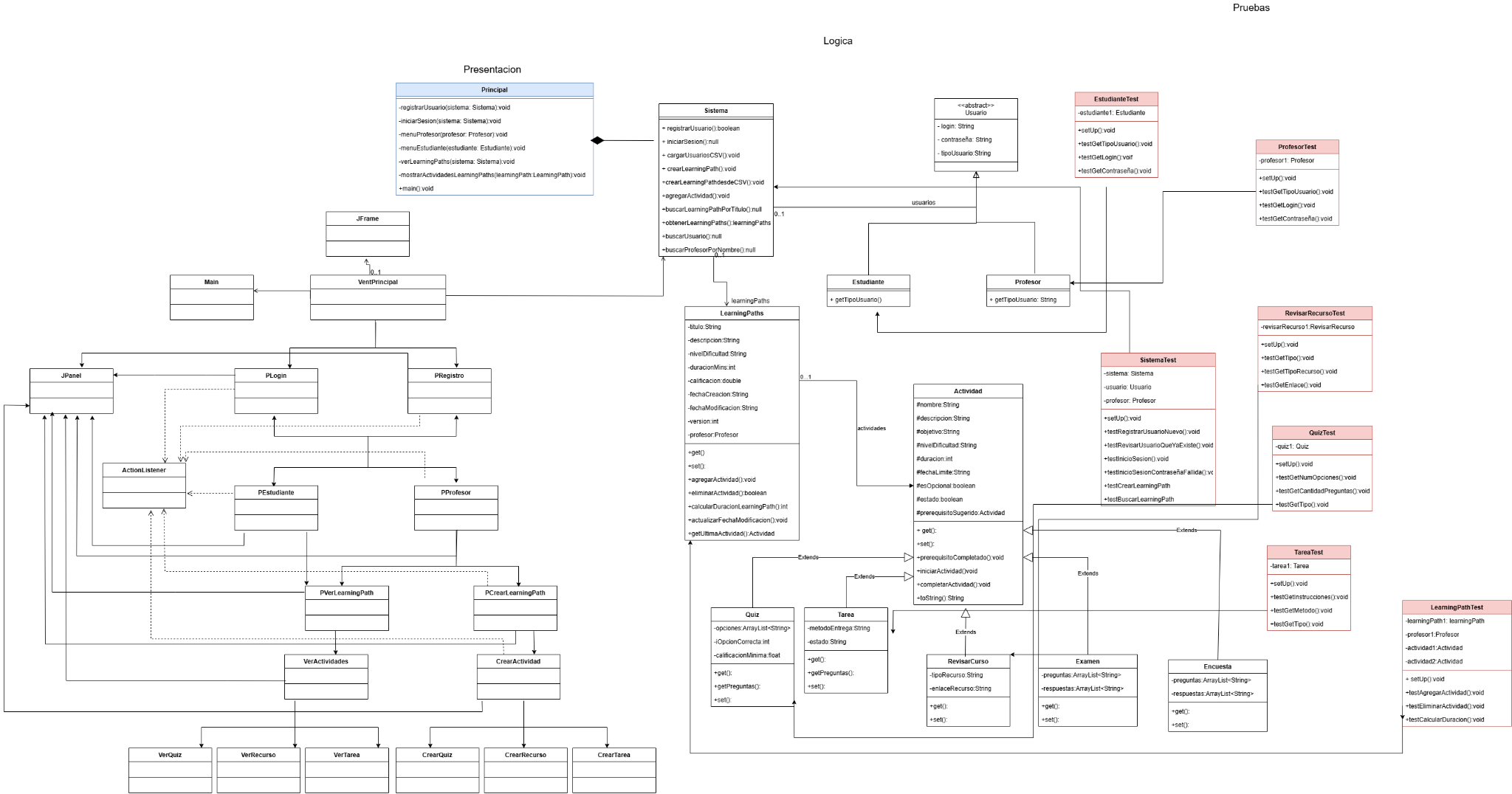


Diagrama de clases proyecto 2:

## 

Diagrama de clases aplicación completa. ( archivo png en el repositorio)

## **Diagrama de clases proyecto 1:**

Diagrama de las clases del negocio

## **\* reviews en learning path**

## **1. Visión general de SISTEMA**

* **El diagrama muestra la base para un sistema de gestión de aprendizaje, que facilita la creación de learningPaths, la gestión de actividades, y el seguimiento de los usuarios**
* **La clase Sistema actúa como interfaz principal para las operaciones, gestionando tanto usuarios como Learning Paths.**

## **2. Cumplimiento de Requerimientos del Negocio**

### **2.1 Gestión de Usuarios**

* **Clases**
  + **Usuario (abstracta)**
  + **Estudiante**
  + **Profesor**
* **Implementación:**
  + **Usuario es una clase abstracta que define los atributos login, contraseña y tipo de usuario**
  + **Las subclases Profesor y Estudiante implementan getTipoUsuario, cumpliendo con lo que dice la clase sistema**
  + **Relación: La clase Sistema mantiene una colección de usuarios mediante la asociación 0…1, lo que demuestra que en esta clase se permite gestionar múltiples usuarios**

### **2.2 Caminos de Aprendizaje (Learning Paths)**

* **Clase principal:LearningPaths representa las rutas de aprendizaje personalizadas**
* **Atributos clave:**
  + **Titulo**
  + **Descripción**
  + **Nivel de dificultad**
  + **Duración en minutos**
  + **Versión**
  + **Profesor**
* **Métodos**
  + **AgregarActividad()**
  + **Eliminarctividad()**
  + **BuscarProfesorPorNombre()**
* **Relaciones**
  + **Cada learning path es gestionado por 1 profesor(1…0)**
  + **Los estudiantes pueden seguir múltiples caminos**

### **1.3 Actividades Diversas**

* **Clase abstracta: Actividad define los atributos y métodos básicos para cada actividad a realizar**
* **Atributos clave:**
  + **Nombre**
  + **Descripción**
  + **Objetivo**
  + **Duracion**
  + **FechaLimite**
  + **esOpcional**
  + **prerequisitoSuperado**
* **Métodos relevantes**
  + **CompletarActividad()**
  + **IniciarActividad()**
* **Subclases específicas**
  + **Quiz: Actividad evaluativa con opciones múltiples y calificacionMinima para pasar**
  + **Tares: Puede tener distintos estados como “envidada”, “exitosa”, o “no exitosa”**
  + **RevisarRecurso: Explorar recursos como videos, libros o sitios web**
  + **Examen: Incluya preguntas abiertas y sus respectivas respuestas**
  + **Encuesta: Permite recopilar opciones de los estudiantes mediante un conjunto de preguntas**

### **2.4 Seguimiento del Progreso y prerequisitos**

* **Atributo importante: prerequisitoSuperado en la clase Actividad, lo que indica que algunas actividades pueden requerir la finalización de otras para acceder a ellas, o por lo menos recibe una advertencia**
* **La clase Progreso permite rastrear el avance de los estudiantes en cada camino de aprendizaje, cumpliendo con el requisito de monitorear el progreso individual.**
* **Estado: Indica si una actividad ha sido o no completada**

## **3. Funcionales Clave del Sistema (clase sistema)**

### **La clase sistema centraliza todas las opciones fundamentales para poder gestionar las funciones necesarias para los usuarios, Learning Paths y actividades**

### **3.1 Métodos de la clase Sistema**

* **registrarUsuario(): Registra un nuevo usuario**
* **iniciarSesion(): Autenticación de usuarios**
* **cargarUsuarioCSV():Carga los usuarios a su respectivo archivo CSV**
* **crearLearningPath(): Crea el LearningPath**
* **buscarLearningPath():Busca el LearningPath**
* **eliminarLearningPath():Elimina el LearningPath seleccionado**
* **buscarProfesorPorNombre():Busca el profesor por su nombre**

**Esta clase centraliza la interacción entre los diferentes módulos**

## **4. Reglas de Dominio**

### **4.1 Estructura de Usuarios**

* **Regla: Todo usuario debe ser o Profesor o Estudiante.**
* **Implementación: Usuario es una clase abstracta, forzando que cada instancia sea de tipo Profesor o Estudiante.**

### **4.2 Composición de Caminos de Aprendizaje**

* **Regla: Un Camino de Aprendizaje debe contener una o más Actividades.**
* **Implementación: Relación de composición entre CaminoAprendizaje y Actividad (1..\*).**

### **4.3 Tipos de Actividades**

* **Regla: Toda Actividad debe ser de un tipo específico.**
* **Implementación: Actividad es una clase abstracta con subclases concretas para cada tipo de actividad.**

### **4.4 Prerrequisitos de Actividades**

* **Regla: Las actividades pueden tener prerrequisitos sugeridos.**
* **Implementación: Atributo prerequisitoSugerido en la clase Actividad.**

### **4.5 Progreso del Estudiante**

* **Regla: El progreso se calcula por estudiante y por camino de aprendizaje.**
* **Implementación: Clase Progreso relacionada tanto con Estudiante como con CaminoAprendizaje.**

## **5. Principal (consola)**

**La aplicación cuenta con una clase de presentación principal, el cual se puede ejecutar como un Java Application e interactuar a través de la consola. Dentro de esta, se encuentran varios métodos que hacen funcionar la aplicación de forma eficiente:**

1. **Autenticación**
2. **Main profesor (menú)**
3. **Main estudiante (menú)**

**Aparte de estos tres principales métodos, hay otros que funcionan como un apoyo o extensión de estos.**

## **6. Main (Interfaz gráfica JFrame)**

**También se implementó una interfaz gráfica que permite navegar las funciones del programa, ejecutando el archivo main como Java Application. Esta tiene las mismas funciones que el archivo Principal, por lo que cuenta con un sistema de autenticación y, dependiendo del usuario, menús interactivos diferentes.**

**La función principal de este archivo es cargar al sistema la información necesaria que se encuentra en los archivos csv, además de invocar a VentPrincipal que es la ventana principal e inicial de la aplicación.**

**Para poder desarrollar paso a paso la interfaz, primero se construyó un boceto en draw.io, adjuntado en el repositorio GIT.**

## **7. Historias de usuario**

**Las historias de usuario son una herramienta importante para entender el funcionamiento de la aplicación y, sobre todo, su propósito. Se detallarán las historias para los dos usuarios principales del sistema: Profesores (creando y gestionando learning paths) y estudiantes (haciendo las actividades). A continuación se detallará información de usuarios potenciales:**

1. **Martha Camila Aponte**

**Es una profesora de la universidad de Los Andes, del departamento de ingeniería de sistemas y computación. Martha es la coordinadora del curso Introducción a la programación. Para poder usar los learning paths de manera adecuada debe tener un usuario (asignado como profesor) en el sistema (mc.aponte2) y una contraseña para poder autenticarse. Al iniciar sesión, la profesora Martha necesita crear un learning path para apoyar el material del curso, este se llamará introducción a la programación. Ya creado, puede agregar tres tipos de actividades principales para que los estudiantes las realicen y las revisen. Por ejemplo, asignar una revisión de recurso, luego una tarea y por último un quiz para comprobar el entendimiento del tema. En un futuro, podrá monitorear a cada estudiante en su proceso de aprendizaje que lleve en el learning path.**

1. **Santiago Vergara**

**Santiago es un estudiante de primer semestre que está muy interesado en aprender a programar. Para poder acceder al learning path creado por la profesora Martha, se le suministra un login (o se le solicita crear una cuenta) con el atributo de estudiante. Al entrar, tendrá acceso a los learning paths disponibles en el sistema y podrá entrar a cada uno, donde podrá mirar las actividades propuestas por el profesor, por ejemplo, una tarea. Se recomienda hacerlas en orden, pero el estudiante es libre de seguir su propio camino para completar el learning path.**

1. **David Vergara**

**David es un profesor del área de ciencias interesado en crear un learning path para que los estudiantes de su facultad puedan aprovechar esta herramienta. Este profesor tiene un problema, olvidó su contraseña. Este problema hará que el sistema no lo deje autenticarse y no podrá usar los servicios de la aplicación hasta resolver el tema de la contraseña.**

## **8. Conclusión**

**El diagrama UML ofrece una estructura en la cual nos basamos para crear el programa. La clase Sistema como núcleo permite centralizar las funciones y garantizar una experiencia fluida al programarlo. Además, la implementación final de la interfaz gráfica permite mostrar el trabajo realizado de manera visual e interactiva.**

**URL REPOSITORIO PROGRAMA:**

[**https://github.com/DPOO-G9-202420/Proyecto1\_DPO.git**](https://github.com/DPOO-G9-202420/Proyecto1_DPO.git)