

# Modelo de Diseño

## Testify

## OSLO

Ojeda Valeria – Sly Eduardo  
Levipichun Emilio – Oyarzo Malena



El propósito del Modelo de Diseño es empezar a realizar los casos de uso desarrollados durante las etapas anteriores en especial la etapa de captura de requerimientos. Es decir, tomar el Modelo de Casos de Uso y las Especificaciones Suplementarias creadas con anterioridad entre otros insumos y generar un modelo de diseño que pueda ser usado por los desarrolladores durante la etapa de implementación.

## Tabla de contenido

|  |           |
|--|-----------|
| <b>Introducción .....</b>                          | <b>4</b>  |
| <i>Propósito .....</i>                             | <i>4</i>  |
| <i>Alcance .....</i>                               | <i>4</i>  |
| <i>Definiciones, siglas y abreviaturas. ....</i>   | <i>4</i>  |
| <i>Visión general.....</i>                         | <i>5</i>  |
| <b>Diseño de Casos de Uso .....</b>                | <b>6</b>  |
| <i>Diseño del Caso de Uso CRUD Escenario .....</i> | <i>7</i>  |
| Diagrama de paquetes.....                          | 7         |
| Diagrama de secuencia .....                        | 8         |
| <b>Diseño de Subsistemas .....</b>                 | <b>8</b>  |
| <i>Subsistemas Específicos .....</i>               | <i>8</i>  |
| Subsistema de Proyectos .....                      | 8         |
| Subsistema de Usuarios .....                       | 9         |
| Subsistema de Escenario .....                      | 9         |
| Subsistema de Gestión de Documentos Adjuntos.....  | 10        |
| Subsistema de Exportación de Informes .....        | 10        |
| Subsistema de Iteraciones .....                    | 10        |
| Subsistema de Roles y Permisos .....               | 11        |
| Subsistema de Tipos, Subtipos y Categorías.....    | 11        |
| Dependencias .....                                 | 12        |
| Recursos.....                                      | 13        |
| <b>Diagramas .....</b>                             | <b>14</b> |
| <i>Diagrama de Clases.....</i>                     | <i>14</i> |
| <i>Diagrama de Colaboración.....</i>               | <i>15</i> |

# Modelo de Diseño

---

## Introducción

El presente documento es el Modelo de Diseño para el sistema Testify, que se encarga de la gestión de casos de prueba en proyectos de software. Aquí se detallan los aspectos técnicos y funcionales que guían la implementación de las diferentes funcionalidades descritas en las etapas anteriores de análisis y captura de requerimientos. Este documento es clave para alinear el desarrollo y la arquitectura del sistema con los requerimientos funcionales y no funcionales, asegurando que cada componente interactúe correctamente en el entorno del sistema.

## Propósito

El propósito del Modelo de Diseño es proporcionar una descripción detallada de cómo se implementarán los componentes del sistema Testify. Este documento está dirigido principalmente a los desarrolladores, diseñadores de software y arquitectos de sistemas que son responsables de la implementación. Además, sirve como una guía técnica para los interesados en comprender cómo el sistema ejecutará las funcionalidades requeridas. Los diagramas, especificaciones de interacción y descripciones incluidas en este documento son cruciales para la fase de desarrollo.

## Alcance

El Modelo de Diseño se aplica a todo el sistema Testify, afectando directamente las áreas de gestión de usuarios, asignación de roles, administración de proyectos, iteraciones y escenarios de prueba. Este modelo cubre todas las capas de software involucradas, desde la interacción del usuario hasta la persistencia de datos en la base de datos. Abarca tanto la implementación del frontend como del backend del sistema, proporcionando una visión integral de las interacciones entre sus componentes.

## Definiciones, siglas y abreviaturas.

- API: Application Programming Interface. Conjunto de funciones y procedimientos para interactuar con los componentes del sistema.
- CRUD: Create, Read, Update, Delete. Operaciones básicas de manejo de datos en el sistema.
- Desarrollador: Rol encargado de agregar comentarios a los casos de uso y escenarios que le fueron asignados, y también de revisar los resultados de las pruebas.
- Documento Adjunto: Archivos que pueden ser asociados a los casos de uso para

proporcionar contexto adicional o evidencia del proceso de prueba.

- Escenario: Configuración estructurada que permite organizar y ejecutar pruebas bajo condiciones controladas para validar aspectos específicos del sistema en desarrollo.
- Gestor de Prueba: Usuario encargado de gestionar las pruebas de software y su progreso.
- GUI: Graphical User Interface. Interfaz gráfica de usuario para la interacción visual con el sistema.
- Hito: Una etapa importante dentro del ciclo de vida del proyecto, que agrupa uno o más casos de uso. Un proyecto está compuesto por varios hitos.
- Precondición: Un estado o conjunto de requisitos que deben cumplirse antes de ejecutar un caso de uso o escenario. Pueden estar activas o desestimadas según su relevancia.
- Resultado Esperado: El resultado que se espera obtener al seguir los pasos de un caso de uso o escenario.
- Resultado Obtenido: El resultado real que se obtiene tras ejecutar los pasos del caso de uso o escenario. Se compara con el resultado esperado para determinar el éxito o fallo de la prueba.
- Tasa de Aprobación: El porcentaje de casos de uso aprobados dentro de un proyecto, que determina si el proyecto ha alcanzado el umbral mínimo para considerarse aprobado.
- Tester: Rol responsable de ejecutar pruebas y actualizar los resultados de los casos de uso asignados.
- Testify: Sistema de gestión de casos de prueba para proyectos de desarrollo de software.
- UML: Unified Modeling Language. Lenguaje estándar utilizado para representar el diseño y la arquitectura del sistema.

## Visión general

El documento está estructurado en varias secciones que detallan el diseño de los casos de uso del sistema, la arquitectura del software y las interacciones entre los diferentes componentes. Las principales secciones incluyen:

- Diseño de Casos de Uso: Describe la implementación técnica de los casos de uso definidos en la etapa de análisis.
- Diseño de Subsistemas: Detalla los subsistemas clave y sus interacciones.
- Diagramas de Paquetes: Representa las divisiones de responsabilidad y los componentes del sistema.
- Diagramas de Interacción: Muestra cómo interactúan los distintos componentes en la ejecución de casos de uso.

- **Requerimientos Especiales:** Consideraciones técnicas adicionales que influyen en la implementación, como limitaciones de rendimiento y consideraciones de seguridad.

## Diseño de Casos de Uso

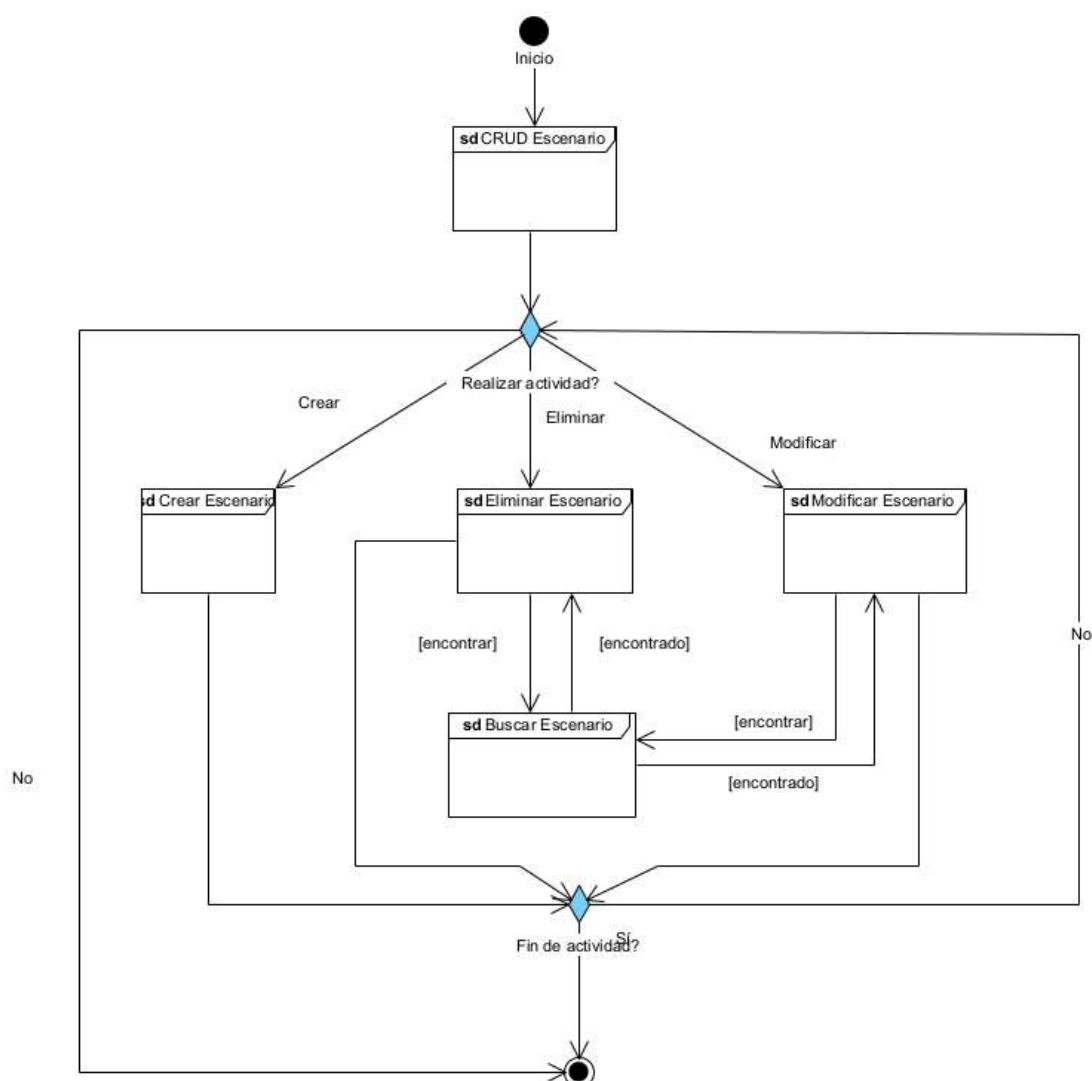
En esta sección se especifica el comportamiento de los casos de uso mediante la definición de los subsistemas y objetos de diseño que interactúan para ejecutar cada escenario. A continuación, se definen los componentes involucrados en el sistema:

- **Objetos de Diseño:** Cada objeto de diseño es una abstracción que encapsula una parte de la lógica del sistema. Estos incluyen clases como Usuario, Proyecto, Escenario, Iteración, Checklist, entre otros.
- **Operaciones:** Se especifican las operaciones que cada objeto puede realizar. Por ejemplo, el objeto Escenario tiene operaciones para crear, actualizar, eliminar, y consultar escenarios. Estas operaciones interactúan con otros objetos como Usuario y Proyecto para asegurar una correcta gestión de pruebas.
- **Interfaces:** Se describen las interfaces de usuario (UI) y de programación (API) que permitirán la interacción entre los componentes. La interfaz gráfica (GUI) facilita la interacción con los actores del sistema, mientras que las API permiten la integración con otros servicios.

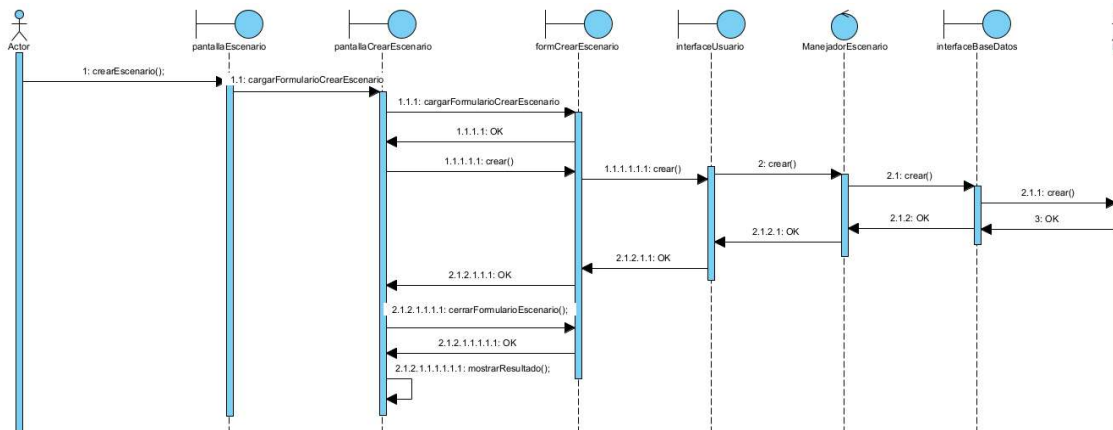
Los diagramas de paquetes, de flujo de eventos, de interacción y los requerimientos especiales son utilizados para detallar cómo se implementan estos objetos en el sistema. A continuación, se proporciona un diagrama de paquetes para entender las relaciones entre los subsistemas, seguido de un diagrama de interacción que detalla el flujo de mensajes en un caso de uso específico (CRUD Escenario).

## Diseño del Caso de Uso CRUD Escenario

### Diagrama de paquetes



## Diagrama de secuencia



## Diseño de Subsistemas

### Subsistemas Específicos

#### Subsistema de Proyectos

##### Propósito

Su existencia es necesaria para permitir la creación, modificación, consulta y eliminación de proyectos, así como la asignación de recursos y usuarios a cada proyecto. Se enfoca en mantener la jerarquía de los proyectos y en relacionarlos con iteraciones y escenarios.

##### Función

Este subsistema transforma las entradas del usuario (como la creación de proyectos y asignación de recursos) en proyectos bien definidos con hitos y recursos asignados. Produce salidas como proyectos estructurados y asignados correctamente con sus elementos relacionados (iteraciones, escenarios, etc.).

##### Subordinados

- Objeto Proyecto: Representa la estructura del proyecto y su relación con otros elementos (iteraciones, usuarios, etc.).
- Objeto Iteración: Relaciona el proyecto con las diferentes iteraciones que lo componen.



## **Subsistema de Usuarios**

### **Propósito**

Gestiona la administración de los usuarios del sistema, incluyendo su autenticación y autorización, así como la asignación de roles y permisos dentro de proyectos específicos. Fue creado para asegurar que cada usuario tenga un acceso adecuado basado en su rol y proyecto asignado.

### **Función**

Recibe como entrada las credenciales del usuario, datos de registro y roles asignados, y produce salidas como la autorización de acceso, la gestión de permisos y la asignación de roles en proyectos.

### **Subordinados**

- Objeto Usuario: Define los atributos del usuario, sus roles y su relación con los proyectos y escenarios.
- Objeto Rol: Asigna los permisos y define qué acciones puede realizar el usuario en el sistema.

## **Subsistema de Escenario**

### **Propósito**

Gestiona la creación, modificación, consulta y eliminación de escenarios de prueba dentro de cada proyecto. El subsistema de Escenarios es clave para realizar las pruebas planificadas y llevar un seguimiento del progreso y resultados de los mismos.

### **Función**

Transforma entradas como la definición de escenarios, pasos de prueba, comentarios y documentos adjuntos en un escenario gestionado. Produce salidas como escenarios actualizados, con resultados de prueba y comentarios.

### **Subordinados**

- Objeto Escenario: Maneja la estructura de los escenarios de prueba, incluyendo pasos, precondiciones y comentarios.
- Objeto Iteración: Vincula los escenarios con las iteraciones correspondientes.
- Objeto Documento Adjunto: Gestiona los archivos adjuntos que respaldan el escenario.

## **Subsistema de Gestión de Documentos Adjuntos**

### **Propósito**

Este subsistema se creó para permitir la vinculación de documentos con escenarios y otros elementos dentro de los proyectos, proporcionando contexto adicional y evidencia dentro del proceso de pruebas. Es un subsistema necesario para manejar de manera eficiente la documentación asociada a las pruebas.

### **Función**

Recibe documentos como entrada (PDFs, imágenes, archivos) y los adjunta a escenarios específicos. La salida es un conjunto de escenarios y proyectos que contienen la documentación adjunta necesaria para las pruebas.

### **Subordinados**

- Objeto Documento Adjunto: Responsable de gestionar los archivos relacionados con los escenarios.
- Objeto Escenario: Los documentos están vinculados a escenarios específicos.

## **Subsistema de Exportación de Informes**

### **Propósito**

Fue creado para generar y exportar informes detallados en formato PDF que contengan los resultados de las pruebas y escenarios. Esto es fundamental para presentar los resultados de pruebas a partes interesadas de una manera formal y organizada.

### **Función**

Transforma los resultados de las pruebas, comentarios y documentos en informes exportables en PDF. Las salidas incluyen informes detallados de pruebas y resultados.

### **Subordinados**

- Objeto Informe: Define la estructura del informe a exportar.
- Objeto Escenario: Recopila los datos necesarios de los escenarios para ser incluidos en el informe.

## **Subsistema de Iteraciones**

### **Propósito**

Gestiona las iteraciones dentro de un proyecto, lo que permite organizar las pruebas en fases de desarrollo. Cada iteración agrupa un conjunto de escenarios que deben ser ejecutados en una fase del proyecto.

## **Función**

Transforma entradas como fechas de inicio y fin, y asignación de escenarios, en iteraciones bien definidas. Las salidas incluyen el progreso de las pruebas dentro de cada iteración.

## **Subordinados**

- Objeto Iteración: Maneja la estructura de la iteración, las fechas y los escenarios asociados.
- Objeto Escenario: Relaciona los escenarios con cada iteración del proyecto.

## **Subsistema de Roles y Permisos**

### **Propósito**

Se encarga de la gestión de roles y permisos, controlando el acceso y las acciones que los usuarios pueden realizar dentro de Testify. Asegura que solo los usuarios autorizados puedan realizar ciertas funciones según su rol.

## **Función**

Transforma las credenciales del usuario y los roles asignados en permisos para realizar ciertas acciones dentro del sistema. Produce como salida el control de acceso a funcionalidades según el rol del usuario.

## **Subordinados**

- Objeto Rol: Define los roles y las acciones permitidas para cada uno.
- Objeto Permiso: Define los permisos específicos asociados a cada rol.

## **Subsistema de Tipos, Subtipos y Categorías**

### **Propósito**

Este subsistema permite la clasificación de los escenarios y otros elementos dentro del sistema, lo que facilita la organización, filtrado y manejo de los elementos dentro de un proyecto. Fue creado para proveer una estructura de categorización flexible y escalable que pueda ser aplicada a diferentes elementos dentro de Testify.

## **Función**

Recibe como entrada la definición de tipos, subtipos y categorías, y permite asociar estos elementos con escenarios o proyectos específicos. Las salidas incluyen la correcta clasificación de los escenarios y otros elementos dentro del sistema.

## Subordinados

- Objeto Tipo: Representa la definición de tipos para clasificar escenarios.
- Objeto Subtipo: Se usa para clasificar con mayor detalle dentro de un tipo.
- Objeto Categoría: Permite la asignación de categorías a los elementos, facilitando la organización y el filtrado dentro del sistema.

## Dependencias

| Subsistema del que depende                   | Naturaleza de interacción   | Características   |
|--|---|---|
| Subsistema de Proyectos                      | Recupera los proyectos asociados al usuario y su estado   | Consulta y gestión de los datos del proyecto al que pertenece el escenario.   |
| Subsistema de Usuarios                       | Se relaciona con el subsistema de roles y permisos para controlar el acceso y con el subsistema de proyectos para gestionar a qué proyectos están asignados | Gestiona a los usuarios del sistema, incluyendo su autenticación y autorización, así como su relación con proyectos y roles.                                      |
| Subsistema de Escenario                      | Relación de iteraciones con escenarios  | Asigna y organiza escenarios dentro de una iteración.   |
| Subsistema de Gestión de Documentos Adjuntos | Permite a los usuarios adjuntar documentos a los escenarios y pruebas para proporcionar evidencia y contexto adicional en el proceso de pruebas.            | Depende del subsistema de escenarios para relacionar los documentos con las pruebas específicas y del sistema de archivos o base de datos para el almacenamiento. |
| Subsistema de Exportación de Informes        | Depende del subsistema de escenarios y proyectos para obtener los datos a exportar.   | Genera informes de resultados de las pruebas y exporta estos informes en formatos como PDF. Facilita la documentación y el  |

|  |  |   |
|--|--|---|
|  |  | análisis de los resultados obtenidos en las pruebas.  |
| Subsistema de Iteraciones                  | Permisos para gestionar iteraciones  | Define qué roles pueden gestionar iteraciones en el ciclo de vida de los proyectos.   |
| Subsistema de Roles y Permisos             | Interactúa con el subsistema de usuarios para asignar roles y el subsistema de proyectos para definir permisos específicos por proyecto. | Controla el acceso de los usuarios y define qué acciones pueden realizar dentro del sistema en función de sus roles (Administrador, Gestor de Pruebas, Tester, Invitado). |
| Subsistema de Tipos, Subtipos y Categorías | Se relaciona con el subsistema de escenarios para asignar clasificaciones a los elementos de prueba.                                     | Clasifica los elementos dentro del sistema (escenarios, pruebas, etc.) mediante tipos, subtipos y categorías, lo que facilita la organización y filtrado de información.  |

## Recursos

Los subsistemas utilizan varios recursos externos para cumplir su función:

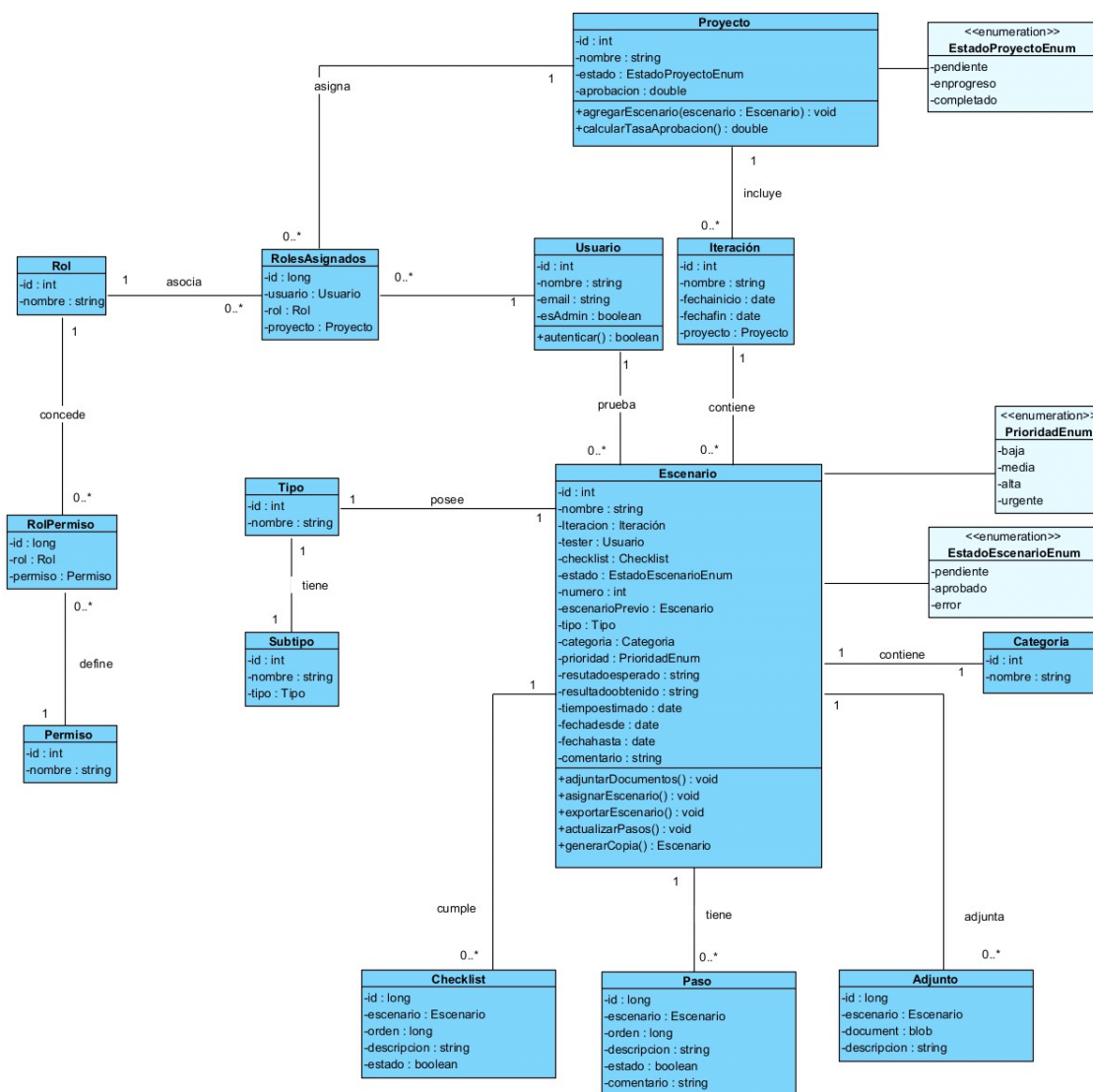
- Librerías de Exportación de PDF: Utilizadas para exportar escenarios y resultados de prueba en formato PDF.
- Servicios de Base de Datos: Necesarios para almacenar y consultar los datos de escenarios, pasos, precondiciones y resultados.
- Recursos de procesamiento: Uso moderado de ciclos de CPU y memoria para la manipulación de datos y generación de informes. El tiempo de proceso depende del número de escenarios y complejidad de las pruebas realizadas.
- Servicios de Autenticación: Utilizados para validar las credenciales de los usuarios y determinar sus roles en cada proyecto.

Estas dependencias aseguran que el subsistema funcione de manera eficiente dentro de Testify, gestionando correctamente los escenarios y permitiendo la trazabilidad de las pruebas realizadas en los proyectos.

## Diagramas

### Diagrama de Clases

El diagrama de clases de Testify proporciona una representación visual de las entidades clave que componen el sistema y sus relaciones. En este diagrama, se destacan las clases que participan en la gestión de proyectos, iteraciones, escenarios y usuarios, además de los roles, permisos y otros componentes esenciales para la funcionalidad del sistema.



## Diagrama de Colaboración

