Modelo de Casos de Uso

**Testify**

OSLO

Ojeda Valeria – Sly Eduardo

Levipichun Emilio – Oyarzo Malena

**

El modelo de casos de uso describe la funcionalidad propuesta del nuevo sistema.

Este modelo se basa en la descripción de elementos o usuarios externos al sistema (actores) y de la funcionalidad del sistema (casos de uso). Un Modelo de Casos de Uso describe los requerimientos funcionales de un actor en términos de las interacciones, la utilización de este modelo presenta el sistema desde la perspectiva de su uso y esquematiza como proporcionará valor a sus usuarios.

El modelo de casos de uso sirve como acuerdo entre clientes y desarrolladores para limitar las funciones con que dispondrá el sistema.

Tabla de contenido

[Requerimientos 4](#_Toc178512868)

[Requerimientos Funcionales 4](#_Toc178512869)

[Requerimientos no Funcionales 5](#_Toc178512870)

[Actores 5](#_Toc178512871)

[Administrador 5](#_Toc178512872)

[Gestor de Prueba 6](#_Toc178512873)

[Tester 6](#_Toc178512874)

[Desarrollador 7](#_Toc178512875)

[Invitado 7](#_Toc178512876)

[Casos de Uso 8](#_Toc178512877)

[Diagramas Asociados 11](#_Toc178512878)

[Diagrama de casos de uso 11](#_Toc178512879)

[Diagrama de Clases 12](#_Toc178512880)

Modelo de Casos de Uso

Requerimientos

Requerimientos Funcionales

Los requerimientos funcionales del sistema describen la funcionalidad o los servicios que se espera que éste provea. En este apartado se describe lo que el sistema Testify tendrá que hacer, los factores que afectan al producto y satisfacen los requerimientos.

A continuación, se narran los requerimientos funcionales detectados y luego se detallan en tablas individuales para cada uno:

RF01: El sistema debe permitir el alta, modificación y aprobado de los casos de prueba.

RF02: El sistema debe permitir la creación de múltiples proyectos para un mismo usuario.

RF03: Un caso de prueba puede estar asignado a un hito.

RF04: Un hito puede estar asignado a más de un caso de prueba.

RF05: Cada caso de prueba puede contener uno o más precondiciones.

RF06: Las precondiciones pueden estar activas o desestimadas.

RF07: Cada caso de prueba puede contener uno o más pasos a seguir.

RF08: Las pruebas pueden estar activas o desestimadas.

RF09: Los casos de prueba deberán ser identificados según su tipo: Funcional, No Funcional, de Validación, Seguridad, Usabilidad, Mantenibilidad, Regresión e Integración.

RF10: El sistema contendrá los roles: Administrador, Gestor de Prueba, Tester e Invitado.

RF11: Los roles serán asignados a los usuarios y por proyecto, a excepción del rol de Administrador.

RF12: Solo el Rol Administrador podrá crear las cuentas de los usuarios.

RF13: El sistema deberá utilizar la API de google para acceder.

RF14: El gestor de prueba solo podrá visualizar los proyectos asignados a su usuario.

RF15: El Tester solo podrá visualizar los casos de prueba a los que fue asignado.

RF16: El Tester solo podrá actualizar el resultado de los casos de prueba y/o agregar comentarios.

RF17: La página principal del sistema debe mostrar los gráficos del proyecto de estado de pruebas y cantidad de pruebas, como su estado Aprobado o Pendiente.

RF18: El sistema debe permitir adjuntar documentos a los casos de prueba.

RF19: Cada proyecto puede tener configurado una tasa mínima de aprobación, el proyecto será marcado como Aprobado cuando el porcentaje de casos de uso aprobados supere la tasa.

RF20: Los casos de prueba que se tendrán en cuenta para el cálculo de aprobación serán aquellos que fueron asignados a un Tester.

RF21: El sistema permitirá exportar en un documento PDF los casos de prueba, gráficos y resultados.

Requerimientos no Funcionales

En este documento, se abordarán los requerimientos no funcionales del sistema, los cuales son cruciales para definir las características que pueden influir en el desempeño y las limitaciones del producto. Los requerimientos no funcionales se centran en aspectos que van más allá de las funcionalidades básicas del sistema y que impactan su rendimiento general, fiabilidad y usabilidad. En el caso de Testify, se identifican los siguientes:

RNF01: El sistema debe pensarse solamente para ser utilizado en computadoras de escritorio o notebooks, no deben considerarse la adaptación a otros dispositivos.

RNF02: El acceso al sistema debe ser vía Web y online.

Actores

En el sistema Testify, los actores representan los diferentes usuarios y sistemas que interactúan con las funcionalidades del sistema. Cada actor tiene un conjunto de características y relaciones específicas que definen cómo se relacionan con los casos de uso del sistema. A continuación, se describen los actores identificados:

Administrador

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | | Administrador | A01 |
| **Descripción** | *El Administrador tiene acceso completo al sistema y es responsable de gestionar usuarios, roles, permisos y proyectos. Además, puede crear, modificar y eliminar casos de prueba, hitos y otras entidades del sistema* | | |
| **Características** | *Acceso completo a todas las funcionalidades del sistema, incluido el mantenimiento de usuarios y roles.* | | |
| **Relaciones** | *Se relaciona con los casos de uso de gestión de usuarios, roles, permisos, proyectos e hitos.* | | |
| **Referencias** | *Accede a todos los casos de uso del sistema, diagramas de secuencia, etc.* | | |

|  |
| --- |
| Comentarios |
| *Solo el Administrador puede crear cuentas de usuario.* |

Gestor de Prueba

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | | Gestor de Prueba | A02 |
| **Descripción** | *El Gestor de Prueba está encargado de administrar los casos de uso asignados a su proyecto, y gestionar el progreso de las pruebas. Tiene permisos para crear y asignar casos de prueba a otros usuarios.* | | |
| **Características** | *Acceso limitado a los proyectos y casos de prueba asignados a su usuario.* | | |
| **Relaciones** | *Relación con la gestión de casos de prueba, asignación de pruebas y consulta de proyectos.* | | |
| **Referencias** | *Participa en la creación y seguimiento de pruebas.* | | |

|  |
| --- |
| Comentarios |
| *Solo puede visualizar y administrar los proyectos y casos de uso asignados.* |

Tester

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | | Tester | A03 |
| **Descripción** | *El Tester está encargado de ejecutar pruebas en los casos de uso que se le han asignado y de reportar resultados. Puede agregar comentarios y actualizar el estado de las pruebas.* | | |
| **Características** | *Puede consultar y actualizar el resultado de las pruebas asignadas a su usuario.* | | |
| **Relaciones** | *Relación con los casos de prueba y la actualización de resultados de pruebas.* | | |
| **Referencias** | *Participa en la ejecución de pruebas y en la modificación de resultados.* | | |

|  |
| --- |
| Comentarios |
| *Solo tiene acceso a los casos de uso que le han sido asignados.* |

Invitado

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | | Invitado | A05 |
| **Descripción** | *El Invitado tiene acceso limitado al sistema, solo puede consultar información básica de los proyectos y los casos de uso sin poder realizar modificaciones.* | | |
| **Características** | *Solo puede consultar información. No tiene permisos de modificación ni de creación.* | | |
| **Relaciones** | *Relación con la consulta de casos de uso y proyectos.* | | |
| **Referencias** | *Solo participa en la consulta de datos.* | | |

|  |
| --- |
| Comentarios |
| *Tiene acceso limitado a la información del sistema.* |

Casos de Uso

En este apartado se detallan los Casos de Uso del sistema, los cuales describen las interacciones clave entre los usuarios y el sistema para alcanzar objetivos específicos. Cada caso de prueba especifica una funcionalidad particular del sistema, describiendo los pasos que un actor, o conjunto de actores, realiza para completar una tarea. Además, se identifican los actores involucrados en cada caso de prueba, ya sean personas o sistemas externos, que interactúan con la plataforma.

|  |  |
| --- | --- |
| **CU01 - CRUD Casos de Prueba** | |
| Actores | Administrador, Gestor de Prueba |
| Descripción | Los usuarios pueden crear, modificar, eliminar y consultar casos de prueba dentro de un proyecto. |

|  |  |
| --- | --- |
| **CU02 - Consultar Casos de Prueba** | |
| Actores | Invitado, Desarrollador, Tester |
| Descripción | Los usuarios pueden visualizar todos los casos de prueba. |

|  |  |
| --- | --- |
| **CU03 - Asignar Caso de Prueba** | |
| Actores | Administrador, Gestor de Prueba, Desarrollador |
| Descripción | El Desarrollador asigna casos de uso a Testers o el Administrador asigna casos de prueba al Desarrollador para su evaluación o implementación. |

|  |  |
| --- | --- |
| **CU04 - Adjuntar Documentos** | |
| Actores | Administrador, Tester, Desarrollador, Gestor de Prueba |
| Descripción | El usuario puede subir y adjuntar documentos a los casos de prueba para respaldar los resultados de las pruebas. |

|  |  |
| --- | --- |
| **CU05 - Comentar Caso de Prueba** | |
| Actores | Administrador, Gestor de Prueba, Desarrollador, Tester |
| Descripción | El usuario puede agregar comentarios en los casos de prueba asignados. |

|  |  |
| --- | --- |
| **CU06 - Exportar Casos de Prueba y Resultados** | |
| Actores | Gestor de Prueba, Administrador |
| Descripción | El usuario puede exportar los casos de prueba, gráficos y resultados del proyecto en un archivo PDF. |

|  |  |
| --- | --- |
| **CU07 - Consultar Proyectos** | |
| Actores | Invitado |
| Descripción | El usuario puede visualizar la información general de todos los proyectos. Al consultar la sección de proyectos, pueden acceder a los detalles relevantes y seleccionar el proyecto que deseen revisar. |

|  |  |
| --- | --- |
| **CU08 - CRUD Proyectos** | |
| Actores | Administrador |
| Descripción | El usuario administra la creación, modificación, eliminación y búsqueda de proyectos en el sistema  El usuario selecciona el proyecto y establece el porcentaje requerido para marcar el proyecto como satisfactorio. |

|  |  |
| --- | --- |
| **CU09 - CRUD Permisos** | |
| Actores | Administrador |
| Descripción | El usuario administra los permisos asignados a los diferentes roles dentro del sistema, pudiendo crear, modificar, eliminar y buscar permisos. |

|  |  |
| --- | --- |
| **CU10 - CRUD Usuarios** | |
| Actores | Administrador |
| Descripción | El usuario administra los usuarios del sistema, pudiendo crear, modificar, buscar y eliminar, y el sistema actualiza la lista de usuarios. |

|  |  |
| --- | --- |
| **CU11 - CRUD Hitos** | |
| Actores | Administrador |
| Descripción | El usuario administra los hitos del sistema, pudiendo elegir entre crear, modificar, eliminar y consultar hitos. |

|  |  |
| --- | --- |
| **CU12 - CRUD Roles** | |
| Actores | Administrador |
| Descripción | El usuario administra los roles en el sistema, pudiendo elegir entre crear, modificar, eliminar y buscar roles. |

|  |  |
| --- | --- |
| **CU13 - Gestionar Rol** | |
| Actores | Administrador |
| Descripción | El usuario gestiona los roles asignados a un usuario específico. Desde esta sección, puede modificar, eliminar y buscar roles asignados, asegurando que cada usuario tenga los permisos adecuados para sus tareas. |

|  |  |
| --- | --- |
| **CU14 - Consultar Hitos** | |
| Actores | Invitado |
| Descripción | El usuario puede visualizar los hitos de un proyecto específico. Después de seleccionar un proyecto, acceden a la lista de hitos asociados, obteniendo una visión clara de su progreso. |

Diagramas Asociados

En el marco del desarrollo del sistema Testify, es fundamental describir cómo se organizan y relacionan sus distintos componentes a través de diagramas de UML. Estos diagramas no solo ofrecen una visión clara de la arquitectura interna del sistema, sino que también facilitan la comprensión de las interacciones entre los elementos que lo componen. A continuación, se presentan los diagramas de casos de uso, de paquetes, de componentes y de clases, los cuales permiten visualizar las diferentes capas de abstracción del sistema.

Diagrama de casos de uso

El diagrama de casos de uso representa gráficamente los casos de uso que tiene un sistema. Se define un caso de uso como cada interacción supuesta con el sistema a desarrollar, donde se representan los requisitos funcionales. Es decir, se está diciendo lo que tiene que hacer un sistema y cómo.

En la siguiente figura se muestra un ejemplo de casos de uso, donde se muestran cinco actores (Administrador, Gestor de Prueba, Tester e Invitado) y las operaciones que pueden realizar sus roles.

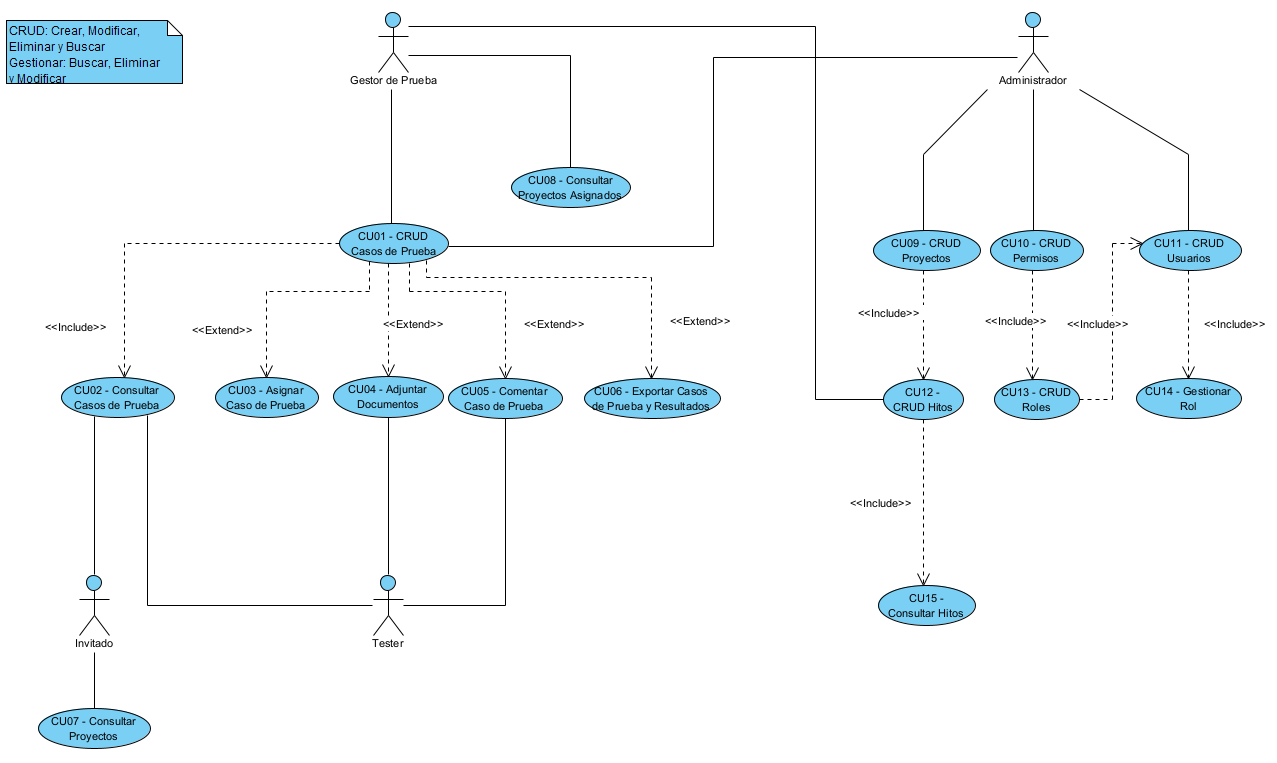


Diagrama de Clases

El diagrama que se presenta es un diagrama de clases UML, el cual modela la estructura de un sistema de gestión de proyectos, roles, casos de prueba y permisos. A continuación, se describen las principales características y relaciones:

Clases Principales

1. Proyecto

Atributos:

* id: Identificador único del proyecto.
* nombre: Nombre del proyecto.
* estado: Tipo enumerado EstadoProyectoEnum que puede ser pendiente, enProceso o completado.
* aprobacion: Representa el porcentaje de aprobación.

Métodos:

* agregarCasoDePrueba(Revision): Permite agregar un caso de prueba al proyecto.
* calcularTasaAprobacion(): Método para calcular la tasa de aprobación del proyecto.

Relaciones:

* Relación uno a muchos con Hito: Un proyecto contiene múltiples hitos.
* Relación de muchos a muchos con Usuario a través de RolesAsignados.
* Relación uno a muchos con Categoria: Un proyecto puede tener múltiples categorías.

1. Usuario

Atributos:

* id: Identificador único del usuario.
* nombre: Nombre del usuario.
* email: Correo electrónico del usuario.
* esAdmin: Indica si el usuario tiene privilegios de administrador.
* autenticado: Booleano que indica si el usuario está autenticado.

Relaciones:

* Relación uno a muchos con RolesAsignados: Un usuario puede tener varios roles asignados en diferentes proyectos.
* Relación uno a muchos con Revision: Un usuario puede estar asignado como desarrollador o tester en varias revisiones.

1. Rol

Atributos:

* id: Identificador del rol.
* nombre: Nombre del rol (Administrador, Tester, Desarrollador, etc.).
* Relaciones:
* Relación uno a muchos con RolesAsignados: Un rol puede estar asignado a varios usuarios en diferentes proyectos.

1. Permiso

Atributos:

* id: Identificador del permiso.
* nombre: Nombre del permiso.

Relaciones:

* Relación muchos a muchos con Rol a través de RolPermiso: Un rol puede tener varios permisos, y un permiso puede estar asignado a varios roles.

1. RolesAsignados

Atributos:

* id: Identificador único.
* usuario: Relación con la clase Usuario.
* rol: Relación con la clase Rol.
* proyecto: Relación con la clase Proyecto.

Relaciones:

* Relación de asignación entre un usuario, un rol y un proyecto.

1. Hito

Atributos:

* id: Identificador único.
* nombre: Nombre del hito.
* fechaInicio: Fecha de inicio del hito.
* fechaFin: Fecha de finalización del hito.

Relaciones:

* Relación muchos a uno con Proyecto: Un hito pertenece a un proyecto.
* Relación uno a muchos con Revision: Un hito puede contener varias revisiones.

1. Revision

Atributos:

* id: Identificador único de la revisión.
* nombre: Nombre de la revisión.
* hito: Relación con Hito.
* tester y desarrollador: Relaciones con Usuario que representan a las personas asignadas como tester y desarrollador de esa revisión.
* precondicion: Relación con la clase Checklist.
* estado: Tipo enumerado EstadoCasoPruebaEnum, con valores pendiente, enPruebas, corregir, aprobado.
* resultadoEsperado: Resultado esperado de la revisión.
* resultadoObtenido: Resultado obtenido tras las pruebas.
* Otros atributos como prioridad y fechas relevantes a la revisión.

Métodos:

* adjuntarDocumentos(): Permite adjuntar documentos a la revisión.
* asignarCasoDePrueba(): Asigna un caso de prueba.
* exportarCasoDePrueba(): Exporta el caso de prueba.

Relaciones:

* Relación uno a muchos con Checklist: Las revisiones tienen varias precondiciones.
* Relación uno a muchos con Pasos: Una revisión puede tener varios pasos a seguir.

1. Checklist

Atributos:

* id: Identificador único.
* descripcion: Descripción del checklist.
* estado: Estado del checklist (activo o no).

Relaciones:

* Relación uno a muchos con Revision: Una revisión puede tener múltiples elementos en su checklist.

1. Paso

Atributos:

* id: Identificador único.
* descripcion: Descripción del paso.

Relaciones:

* Relación uno a muchos con Revision: Los pasos están relacionados con una revisión.

1. Categoria

Atributos:

* id: Identificador único.
* nombre: Nombre de la categoría.
* proyecto: Relación con la clase Proyecto.

Relaciones:

* Relación muchos a uno con Proyecto: Las categorías están asociadas a proyectos.

**Enumeraciones:**

* EstadoProyectoEnum: Representa el estado del proyecto, con valores pendiente, enProceso, completado.
* EstadoCasoPruebaEnum: Define el estado del caso de prueba, con valores como pendiente, enPruebas, corregir, aprobado.
* PrioridadEnum: Representa la prioridad de la revisión, con valores baja, media, alta.

**Relación entre entidades:**

* La relación clave es entre Usuario, Proyecto, y Rol, donde la clase RolesAsignados funciona como entidad puente para asignar roles específicos a usuarios dentro de proyectos.
* La clase Revision está relacionada con hitos (Hito) y usuarios (Usuario como tester y desarrollador), y es central en el seguimiento de las pruebas dentro de un proyecto.
* La clase Categoria agrupa revisiones por proyecto, permitiendo una organización adicional dentro del sistema.

