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](#_Toc258856753)

[Requerimientos Funcionales 4](#_Toc258856754)

[Requerimientos no Funcionales 5](#_Toc258856755)

[Actores 5](#_Toc258856756)

[[Actor 1] 5](#_Toc258856757)

[[Actor 2] 7](#_Toc258856758)

[Casos de Uso 7](#_Toc258856759)

[[Caso de Uso 1] 7](#_Toc258856760)

[Descripción 7](#_Toc258856761)

[Diagramas Asociados 7](#_Toc258856762)

[Diagrama de casos de uso 7](#_Toc258856763)

[Diagramas de Paquetes 8](#_Toc258856764)

[Diagrama de componentes 9](#_Toc258856765)

[Diagrama de Clases 10](#_Toc258856766)

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 uso.

RF02: El sistema debe permitir la creación de múltiples proyectos para un mismo usuario. RF03: Un proyecto estará dividido en Hitos.

RF04: Cada Hito contendrá 1 o más de un caso de uso.

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

RF06: Las precondiciones pueden estar activas o desestimadas.

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

RF08: Los pasos a seguir pueden estar activos o desestimados.

RF09: Los casos de uso deberán ser identificados según su tipo: Documentación o Código.

RF10: El sistema contendrá los roles: Administrador, Gestor de Prueba, Tester, desarrollador 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 uso a los que fue asignado.

RF16: El desarrollador solo podrá visualizar los casos de uso a los que fue asignado.

RF17: El Tester solo podrá actualizar el resultado de los casos de uso y/o agregar comentarios.

RF18: El desarrollador podrá agregar comentarios a los casos de uso a los que fue asignado.

RF19: 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.

RF20: El sistema debe permitir adjuntar documentos a los casos de uso.

RF21: 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.

RF22: Los casos de uso que se tendrán en cuenta para el cálculo de aprobación serán aquellos que fueron asignados a un Tester.

RF23: El sistema permitirá exportar en un documento PDF los casos de uso, 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 y 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.* |

Desarrollador

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | | Desarrollador | A04 |
| **Descripción** | *El Desarrollador trabaja en los casos de uso asignados para su desarrollo o corrección. Puede agregar comentarios y consultar el progreso de sus casos de uso asignados.* | | |
| **Características** | *Puede agregar comentarios sobre los casos de uso a los que está asignado y seguir su progreso.* | | |
| **Relaciones** | *Relación con la gestión de casos de uso y el seguimiento del avance en las pruebas.* | | |
| **Referencias** | *Participa en el desarrollo y análisis de los casos de uso.* | | |

|  |
| --- |
| 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

[Caso de Uso 1]

Descripción

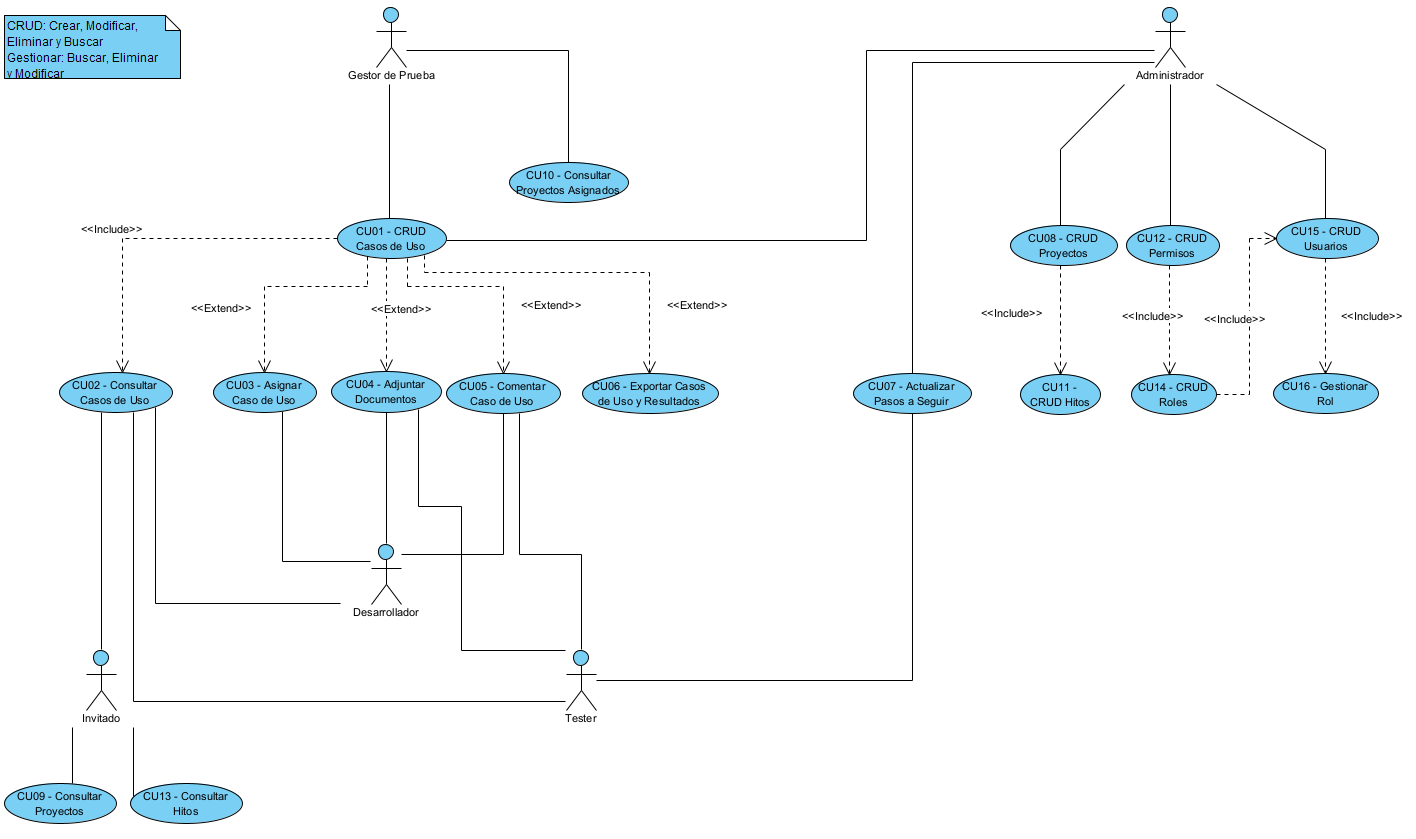
[Explicar brevemente el propósito del caso de uso]

Diagramas Asociados

[Incluir aquí todos los diagramas UML asociados a este documento.]

Diagrama de casos de uso

El diagrama de casos de usos 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, Desarrollador e Invitado) y las operaciones que pueden realizar sus roles.

Diagramas de Paquetes

[Los diagramas de Paquetes se usan para reflejar la organización de paquetes y sus elementos. Los usos más comunes de estos diagramas, son para organizar diagramas de casos de uso y diagramas de clases, estos paquetes son como grandes contenedores de clases.

Los elementos contenidos en un paquete comparten el mismo espacio de nombres, esto significa que los elementos contenidos en un mismo espacio de nombres específico deben tener nombres únicos.]



Diagrama de componentes

[Lo que distingue el diagrama de componentes de otro tipo de diagramas es sin duda su contenido. Normalmente contiene componentes, interfaces y relaciones entre ellos.

Los componentes pertenecen a un mundo físico, es decir, representan a un bloque de construcción al modelar aspectos físicos de un sistema.

Cada componente debe tener un nombre que lo distinga de los demás. Al igual que las clases los componentes pueden enriquecerse con compartimientos adicionales que muestran sus detalles.]



Diagrama de Clases

[En UML el diagrama de clases es uno de los tipos de diagramas o símbolo estático y tiene como fin describir la estructura de un sistema mostrando sus clases, atributos y relaciones entre ellos. Estos diagramas son utilizados durante el proceso de análisis y diseño de los sistemas informáticos, en donde se intentan conformar el diagrama conceptual de la información que se manejará en el sistema. Como ya sabemos UML es un modelado de sistema Orientados a Objetos, por ende, los conceptos de este paradigma se incorporan a este lenguaje de modelado.

Los diagramas de clases tienen las siguientes características:

* Las clases define el ámbito de definición de un conjunto de objetos.
* Cada objeto pertenece a una clase.
* Los objetos se crean por instanciación de las clases.

En su representación gráfica contamos con:

* Nombre de la Clase.
* Atributos de la Clase.
* Operations con las Clases.]
* 