

Modelo de Casos de Uso

Kairos

NexTech

Centurión Valeria, Escalante Guillermo, Maldonado Agustina, Mendez Florencia, Ulloa Gonzalo.

**

**



**Tabla de contenido**

[**Requerimientos 4**](#_8yiyoahdyrk1)

[*Requerimientos Funcionales 4*](#_uzm3jk75mqka)

[*Requerimientos no Funcionales 4*](#_92eu00mr8e5w)

[**Actores 4**](#_y60w7npywvp1)

[*[Actor 1] 4*](#_64ru9o49elt2)

[*[Actor 2] 5*](#_drsue8w3ogm)

[**Casos de Uso 5**](#_t85dzrj1p781)

[*[Caso de Uso 1] 5*](#_xhtuc3zdmej4)

[*Descripción 5*](#_kxwy06ehnfw1)

[**Diagramas Asociados 5**](#_m8fv9l3cdq5p)

[*Diagrama de casos de uso 5*](#_v899yggkv21g)

[*Diagramas de Paquetes 6*](#_3kq4tovwr1ll)

[*Diagrama de componentes 7*](#_7kybs5rwciex)

[*Diagrama de Clases 7*](#_hkjtnvqez4pj)

Modelo de Casos de Uso

# Requerimientos

## Requerimientos Funcionales

### Gestión de usuarios y roles

* **RF01:** El sistema deberá contar con autenticación de usuarios mediante cuenta de Google.
* **RF02:** El sistema deberá permitir al Administrador asignar a los líder/es de proyecto/s a cada proyecto.
* **RF03:** El sistema deberá realizar la creación de un perfil de usuario con los atributos ‘nombre’ y ‘correo electrónico'.
* **RF04:** El sistema deberá permitir la asignación de múltiples roles a un mismo usuario dentro de un proyecto.

### Gestión de proyectos

* **RF23:** El sistema deberá permitir al Administrador crear y acceder a proyectos.
* **RF05:** El sistema deberá permitir a los usuarios modificar y acceder a proyectos.
* **RF06:** El sistema deberá permitir al Administrador asignar los usuarios participantes de cada proyecto.

### Planificación

* **RF07:** El sistema deberá permitir planificar las etapas de un proyecto.
* **RF08:** El sistema deberá permitir planificar iteraciones dentro de cada etapa.

### Gestión de tareas

* **RF09:** El sistema debe permitir al líder del proyecto crear tareas.
* **RF10:** El sistema deberá permitir realizar estimaciones sobre las tareas creadas.
* **RF11:** El sistema deberá permitir a los líderes de proyecto añadir, modificar y eliminar categorías de tareas (por ejemplo, *Codificación*).
* **RF12:** El sistema deberá permitir que los miembros propongan tareas fuera de la planificación. Estas tareas deberán quedar en estado “pendiente de aprobación” hasta que el líder de proyecto las acepte o rechace.
* **RF13:** El sistema deberá permitir asignar dependencias entre tareas.
* **RF14:** El sistema no deberá permitir el inicio de una tarea que dependa de otra que aún no se encuentre finalizada.
* **RF15:** El sistema deberá permitir modificar el estado de una tarea.
* **RF16:**El sistema deberá permitir visualizar los diferentes estados de cada tarea.
* **RF17:**El sistema deberá permitir realizar comentarios sobre una tarea.

### Registro de tiempos y seguimiento

* **RF18:**El sistema deberá permitir registrar el tiempo real trabajado en las actividades en el momento en que se realizan.
* **RF19:**El sistema deberá permitir al usuario ingresar manualmente el tiempo trabajado en una tarea cuando este se haya realizado fuera del sistema.
* **RF20:** El líder del proyecto puede asignar una tarea a más de un responsable.
* **RF21:**El sistema deberá brindar una visualización global de los avances alcanzados durante el desarrollo del proyecto.
* **RF22:**El sistema deberá permitir realizar exportaciones de información en distintos formatos (PDF, Excel).

## Requerimientos no Funcionales

## Usabilidad

* **RNF1:** La interfaz debe ser intuitiva y de uso rápido, permitiendo registrar tareas en pocos segundos.
* **RNF2:** El sistema debe estar disponible en español.

## Rendimiento

* **RNF3:** El sistema debe responder a las acciones del usuario en menos de 2 segundos en condiciones normales de carga.

## Seguridad

* **RNF4:** El acceso al sistema debe estar controlado mediante autenticación con UARGFlow/Google.
* **RNF5:** Los datos deben almacenarse de forma segura, con contraseñas encriptadas.

## Compatibilidad

* **RNF6:** El sistema debe ser compatible con los navegadores web modernos (Chrome, Firefox, Edge) en sus versiones actuales.

## Mantenibilidad

* **RNF7:** El código debe documentarse usando Javadoc y TypeDoc para facilitar el mantenimiento.

## Escalabilidad

* **RNF8:** La arquitectura debe permitir escalar para soportar múltiples proyectos y usuarios concurrentes.

# Actores

## Administrador

| Actor | | Administrador | << Identificador >> |
| --- | --- | --- | --- |
| **Descripción** | Usuario del sistema encargado de la gestión inicial de proyectos y usuarios. | | |
| **Características** | Tiene permisos de gestión global, pero no interviene en las tareas internas de los equipos | | |
| **Relaciones** | Se relaciona con los usuarios y proyectos para registrarlos, asignarlos y establecer un líder | | |
| **Referencias** |  | | |

| Comentarios |
| --- |
| *<< Comentarios adicionales sobre el actor >>* |

## Líder de Proyecto

| Actor | | Líder del Proyecto | << Identificador >> |
| --- | --- | --- | --- |
| **Descripción** | Usuario del sistema encargado de la gestión inicial de etapas, usuarios y roles de cada proyecto. | | |
| **Características** | Tiene permisos de gestión en los proyectos, etapas, tareas, reportes y miembros | | |
| **Relaciones** | Se relaciona con los miembros del proyecto | | |
| **Referencias** | El actor se encuentra involucrado en los siguientes CU:  CU ##( agrupacion abm) Gestionar Proyectos, CU03: Asignar Rol, CU07: Ingresar a Proyecto, CU## Gestionar etapas, CU ##Gestionar Iteraciones | | |

| Comentarios |
| --- |
| *<< Comentarios adicionales sobre el actor >>* |

## Miembro

| Actor | | Miembro | << Identificador >> |
| --- | --- | --- | --- |
| **Descripción** | Usuario del sistema encargado del registro de tiempos efectivos aplicados en cada tarea. | | |
| **Características** | Tiene permisos de ingresar tiempos a las tareas, realizar propuestas de tareas pendientes de aprobación | | |
| **Relaciones** | Se relaciona con los líderes de proyecto | | |
| **Referencias** | *<< Elementos del desarrollo en los que interviene el Actor (Caso de Uso, Diagrama de secuencia, ... >>* | | |

| Comentarios |
| --- |
| *<< Comentarios adicionales sobre el actor >>* |

# Casos de Uso

## [Caso de Uso 1]

### Descripción

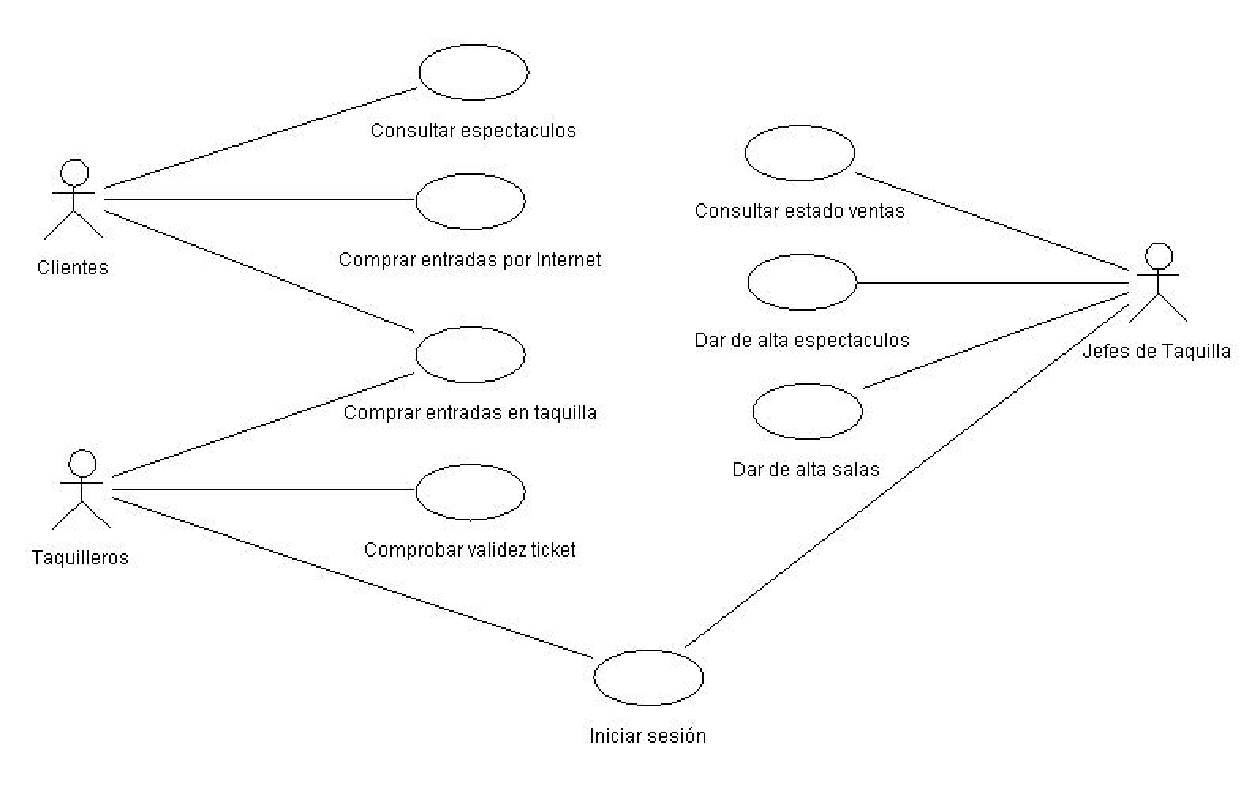
[Explicar brevemente el propósito del caso de uso]

# Diagramas Asociados

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

## Diagrama de casos de uso

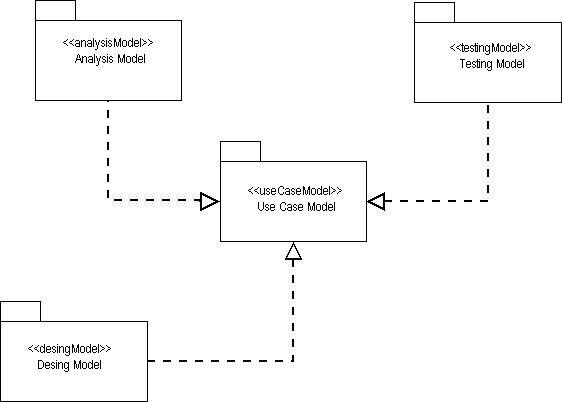
[El diagrama de casos de usos representa gráficamente los casos de uso que tiene un sistema. Se define un caso de uso como cada interacción supuesta con el sistema a desarrollar, donde se representan los requisitos funcionales. Es decir, se está diciendo lo que tiene que hacer un sistema y cómo. En la siguiente figura se muestra un ejemplo de casos de uso, donde se muestran tres actores (los clientes, los taquilleros y los jefes de taquilla) 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.]

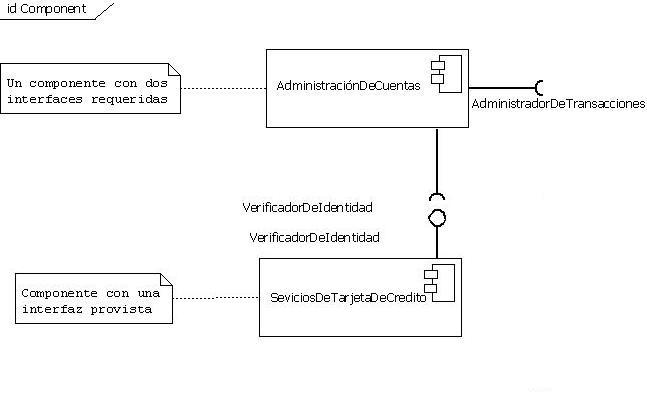
**

## 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

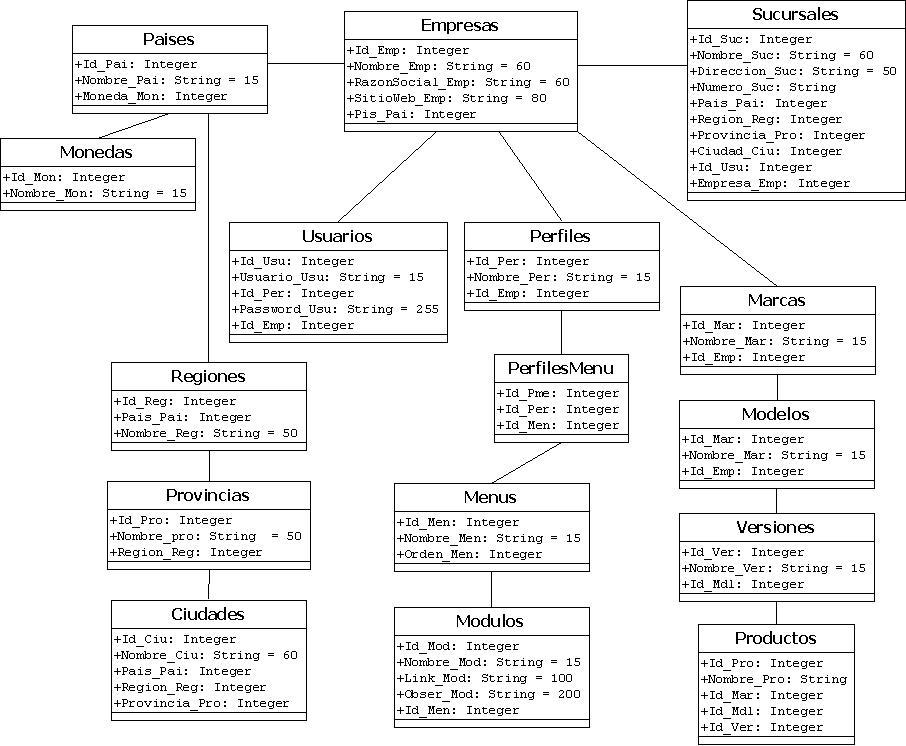
[En UML el diagrama de clases es uno de los tipos de diagramas o símbolo estático y tiene como fin describir la estructura de un sistema mostrando sus clases, atributos y relaciones entre ellos. Estos diagramas son utilizados durante el proceso de análisis y diseño de los sistemas informáticos, en donde 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.
* Operaciones con las Clases.]

**