Jose Dario Gomez Machado

**Documento de Especificación de Arquitectura**

Realizado por:Yuri Natalia López García

Giselle Dayanna Marbello

***HISTORIAL DE REVISIONES***

| **Fecha** | **Versión** | **Autor** | **Descripción** | **Revisado Por** |
| --- | --- | --- | --- | --- |
| <Escriba aquí> | <Escriba aquí> | <Escriba aquí> | <Escriba aquí> | <Escriba aquí> |

**Contenido**

[1.](#_heading=h.3znysh7) Documento de Arquitectura de Software 4

[1.1.](#_heading=h.2et92p0) Introducción 4

[1.2.](#_heading=h.tyjcwt) Propósito 4

[1.3.](#_heading=h.3dy6vkm) Alcance 4

[1.4.](#_heading=h.1t3h5sf) Referencias 5

[1.5.](#_heading=h.4d34og8) Definiciones acrónimos y abreviaciones 5

[2.](#_heading=h.2s8eyo1) Generalidades del Proyecto 5

[2.1.](#_heading=h.17dp8vu) Problema a Resolver 5

[2.2.](#_heading=h.3rdcrjn) Descripción General del Sistema a Desarrollar 5

[2.3.](#_heading=h.26in1rg) Identificación de los Stakeholders y sus responsabilidades 5

[3.](#_heading=h.lnxbz9) Vistas de la arquitectura 6

[3.1.](#_heading=h.35nkun2) Vista de Casos de Uso 6

[3.2.](#_heading=h.1ksv4uv) Vista de Procesos 6

[3.3.](#_heading=h.44sinio) Vista Lógica 6

[3.4.](#_heading=h.z337ya) Vista de Implementación 6

[3.5.](#_heading=h.3j2qqm3) Vista de Despliegue 6

[4.](#_heading=h.1y810tw) Arquitectura en capas 6

[5.](#_heading=h.4i7ojhp) Vista de Datos 7

[5.1.](#_heading=h.2xcytpi) Modelo Relacional 7

[6.](#_heading=h.1ci93xb) Definición de Interfaces de Usuario 7

[7.](#_heading=h.3whwml4) Características Generales de Calidad 7

[7.1.](#_heading=h.2bn6wsx) Tamaño y performance 7

[7.2.](#_heading=h.1pxezwc) Calidad 7

[7.3.](#_heading=h.49x2ik5) Usabilidad 7

[7.4.](#_heading=h.2p2csry) Eficiencia 7

[7.5.](#_heading=h.147n2zr) Seguridad 7

[7.6.](#_heading=h.3o7alnk) Confiabilidad 7

[7.7.](#_heading=h.23ckvvd) Mantenimiento 7

[7.8.](#_heading=h.ihv636) Estándares 7

# Documento de Arquitectura de Software

## Introducción

## Propósito (Describir el propósito de este documento)

El documento presenta la descripción de la arquitectura para el Sistema Hero Kids, en base al Modelo de Vistas “4+1”, que es de utilidad para los stakeholders (Interesados) debido al rol que desempeña cada uno en el desarrollo del sistema.

La arquitectura presentada corresponde a la solución propuesta para el sistema, proporcionando distintas vistas de la arquitectura según los diferentes puntos de vista de los involucrados en el desarrollo del aplicativo. El documento se compone de diagramas y diseños detallados, brindando un contexto y una descripción de las interacciones entre los usuarios y el sistema.

El informe está estructurado de acuerdo con la plantilla para la documentación de arquitecturas de software.

## Alcance (Describir el alcance del documento)

El documento contiene la representación del proyecto Hero kids, mediante el diseño de diagramas dado el análisis previo de los requisitos del sistema para darle solución a una problemática presentada por un cliente.

El documento está organizado mediante tres ideas principales:

* Características generales del diseño.
* Los requisitos funcionales y no funcionales para el diseño.
* Los modelos y vistas que lo representan.

Los modelos implementados para la representación del sistema, permiten analizar el funcionamiento del sistema desde distintos puntos de vista, de forma que podemos implementar el “Modelo de arquitectura 4+1”. Gracias a esto, se puede llegar a implementar *el análisis de los requisitos, la implementación del diseño de solución, así como para la especificación, construcción y despliegue del sistema en su ambiente de explotación.*

Cada uno de los modelos fueron implementados en la herramienta de modelado UML (Lenguaje de modelado unificado) mediante la herramienta Diagrams.net

## Referencias

1. Documento de Especificación de Requerimientos no funcionales.
2. Documento de Visión del Proyecto.
3. Plan de Proyecto del Sistema

## Definiciones acrónimos y abreviaciones

**ARQUITECTURA DE SOFTWARE:** conjunto de elementos estáticos, propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente, como al comportamiento del software en tiempo de ejecución. Naturalmente este diseño Arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto.

**DESCRIPCIÓN DE ARQUITECTURA**: colección de productos de documentación.

**VISTAS:** es una representación de un área de interés o perspectiva del sistema en alto nivel.

**TIPOS DE VISTAS:** especificación de una convención de cómo construir y usar una vista. Deben satisfacer la capacidad de creación y análisis de una vista.

**STAKEHOLDER:** Individuo, equipo u organización con intereses relativos al sistema.

# Generalidades del Proyecto

## Problema a Resolver

El registro de los nuevos usuarios, la tardanza al solicitar cupos, dificultad de realizar una cita, son algunas de las problemáticas con las que cuenta actualmente el jardín por ello las abordamos y les brindamos solución por medio de nuestro sistema.

## Descripción General del Sistema a Desarrollar

Es una aplicación que ayudará a obtener un mejor rendimiento interno en los jardines infantiles, beneficiando a las dos partes que se comunican directamente con el jardín, garantizando seguridad y usabilidad a los usuarios y sus diferentes

problemáticas.

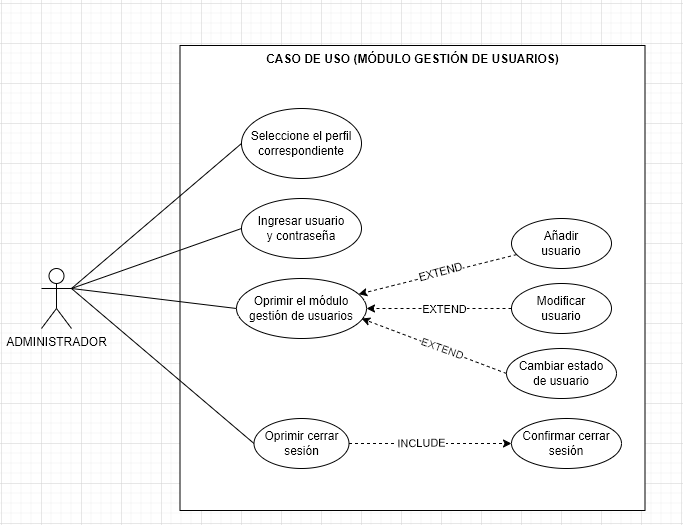
Dentro de la aplicación aparecerán 2 módulos los cuales son: **Módulo gestión usuarios** y **Módulo trámites**.

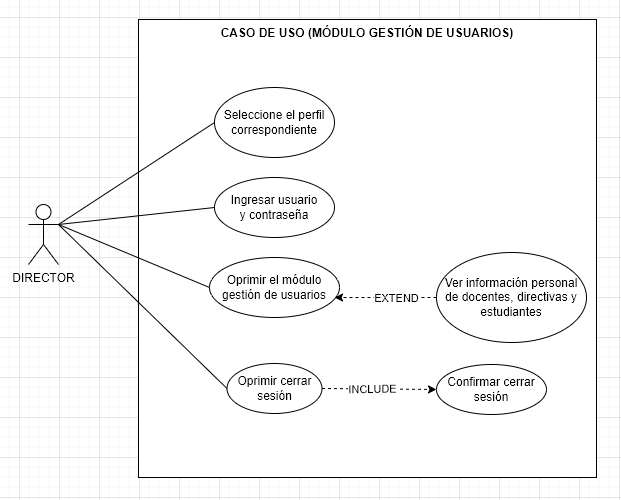
## Identificación de los Stakeholders y sus responsabilidades

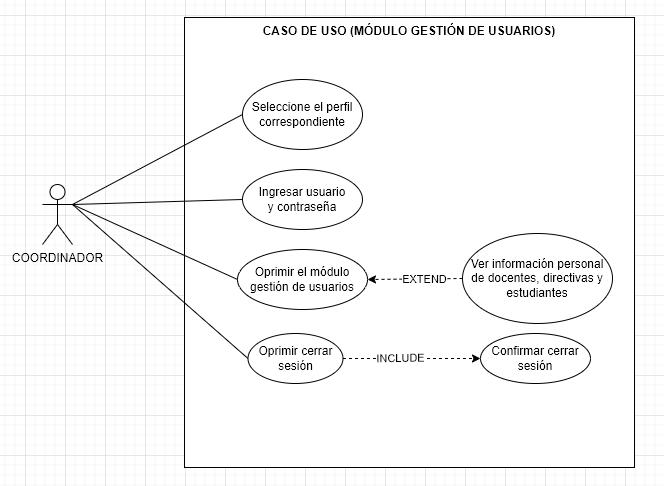
| **STAKEHOLDER** | **DESCRIPCIÓN** | **ESCENARIO** | **Caso de Uso** |
| --- | --- | --- | --- |
| Administrador | Es el usuario que está  encargado de la  gestión de los usuarios dentro de la aplicación. | * Escenario de   gestión de usuarios | * CU\_añadir usuario * CU\_modificar usuario * CU\_cambiar estado de usuario |
| Director | Es el usuario que está pendiente del funcionamiento general del jardín por medio de la aplicación. | * Escenario de gestion de usuarios * Escenario de trámites | * CU\_observar registros de cupos * CU\_realizar citación |
| Coordinador | Es el usuario que está pendiente del funcionamiento general del jardín por medio de la aplicación. | * Escenario de gestion de usuarios * Escenario de trámites | * CU\_observar registros de cupos * CU\_realizar citación |
| Docente | Es el usuario que informa a los acudientes de cualquier evento que se vaya a realizar en el jardín por medio del calendario y cita a los acudientes en el momento que sea necesario. | * Escenario trámites (opción citación) | * CU\_ver información personal de las directivas y estudiantes * CU\_realizar citación |
| Acudiente | Es el usuario que se mantiene informado por medio de la aplicación, observa la información personal de los integrantes del jardín por medio del módulo gestión usuarios, también realiza la solicitud de los cupos y crea una citación por medio del módulo trámites, además observa los eventos próximos del jardín por medio del módulo calendario. | * Escenario de trámites * Escenario de gestion usuarios | * CU\_ver información personal de los integrantes del jardín * CU\_solicitar cupo * CU\_realizar citación |

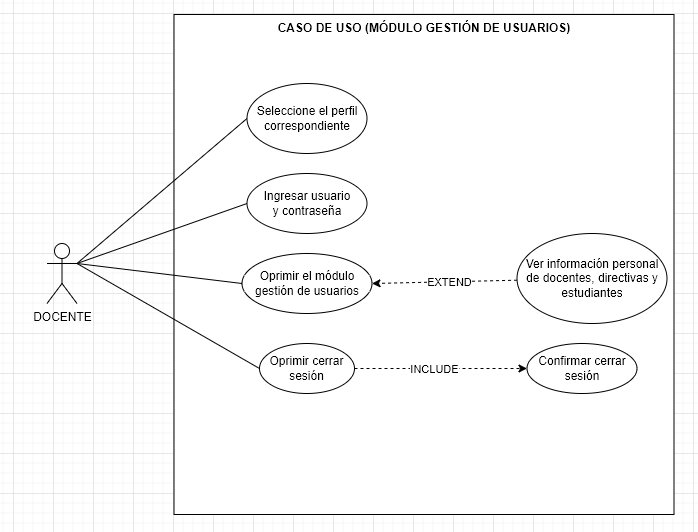
# Vistas de la arquitectura

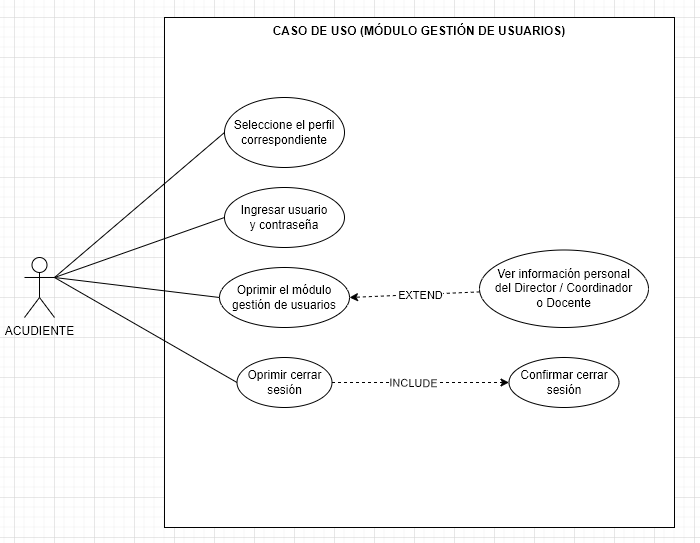
## Vista de Casos de Uso

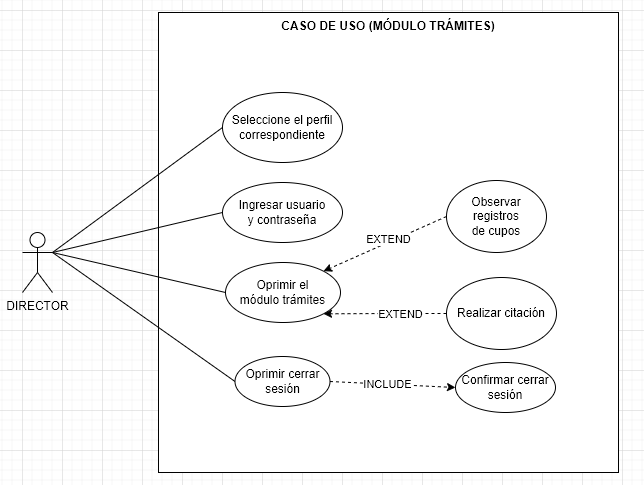


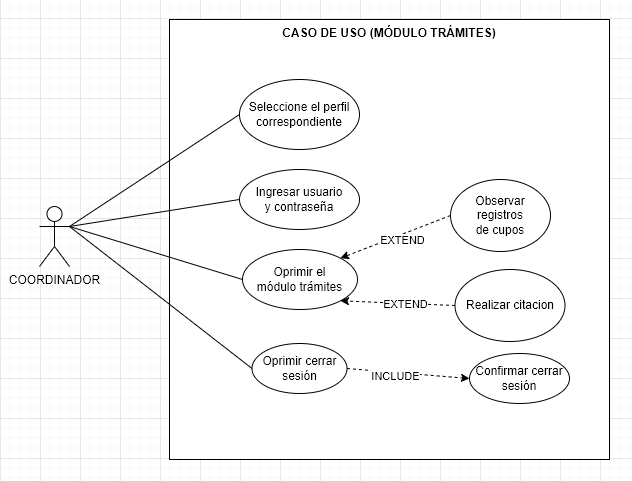


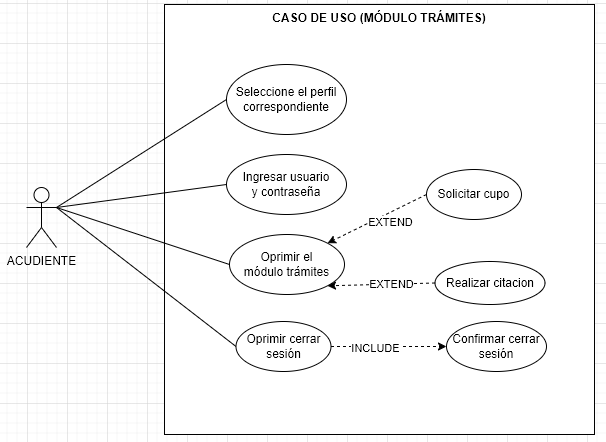


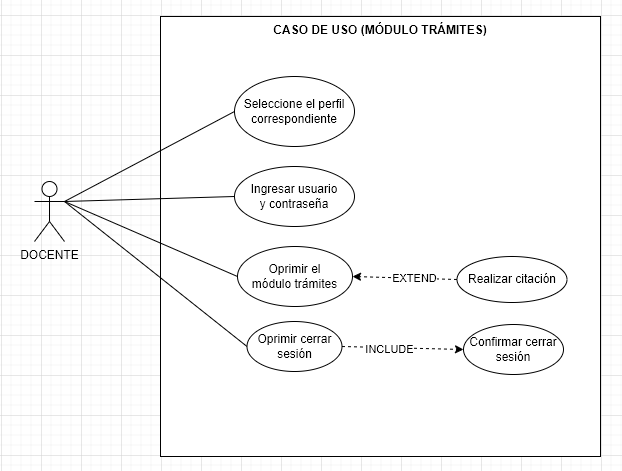






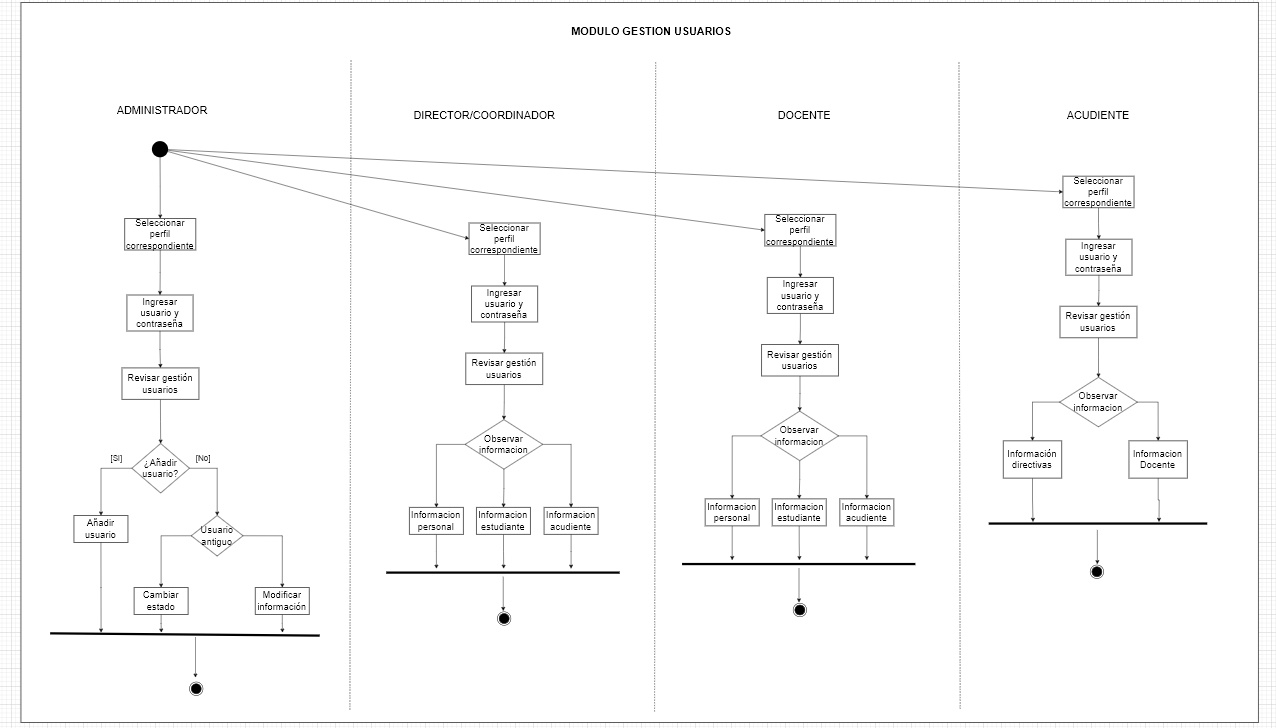


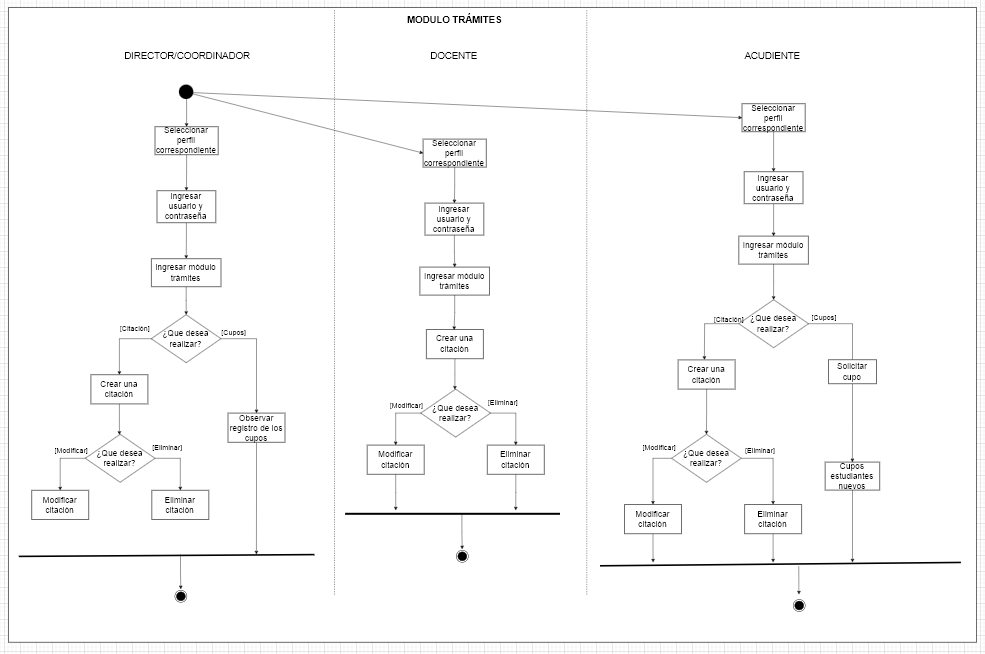




## Vista de Procesos

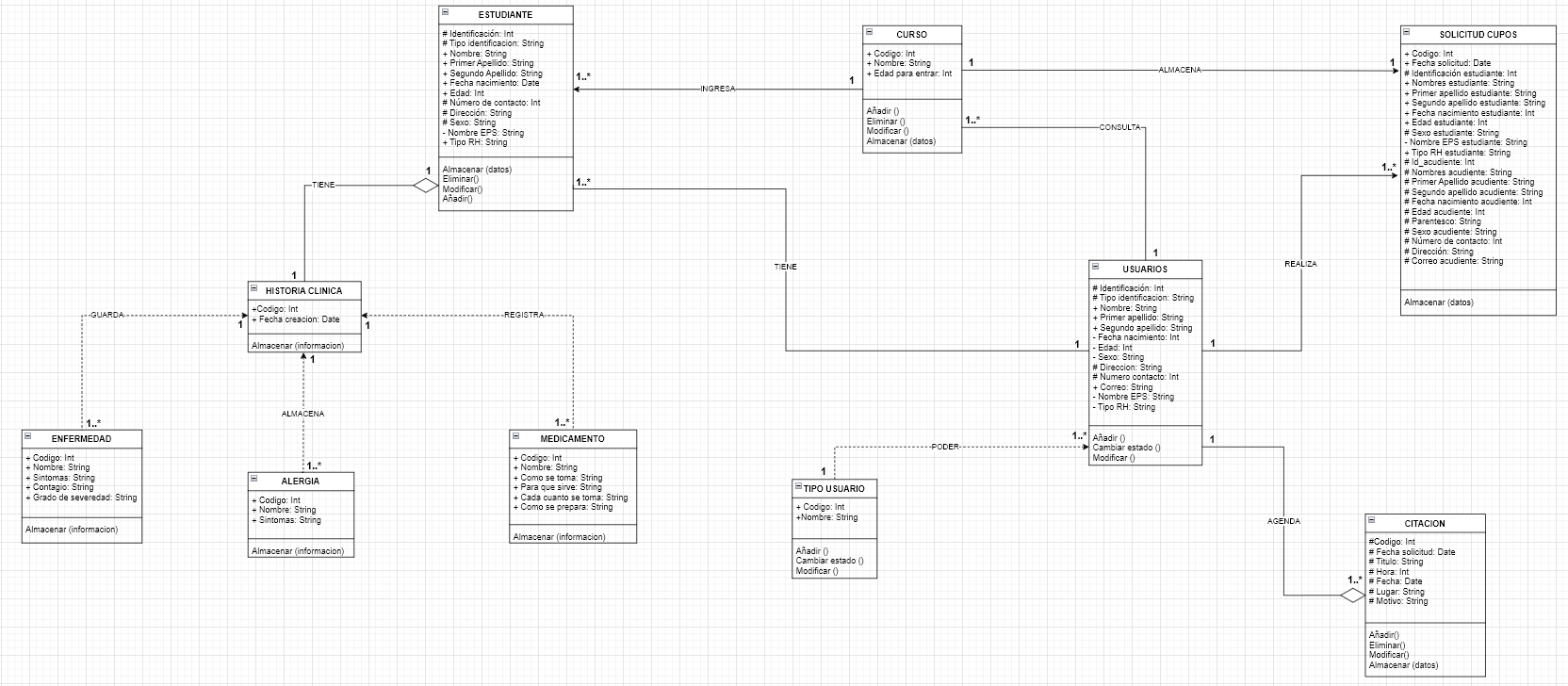
### Diagrama de Actividades



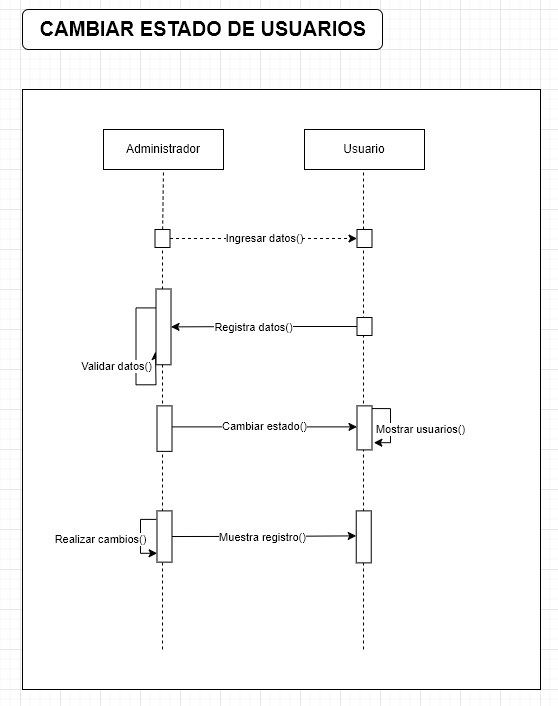


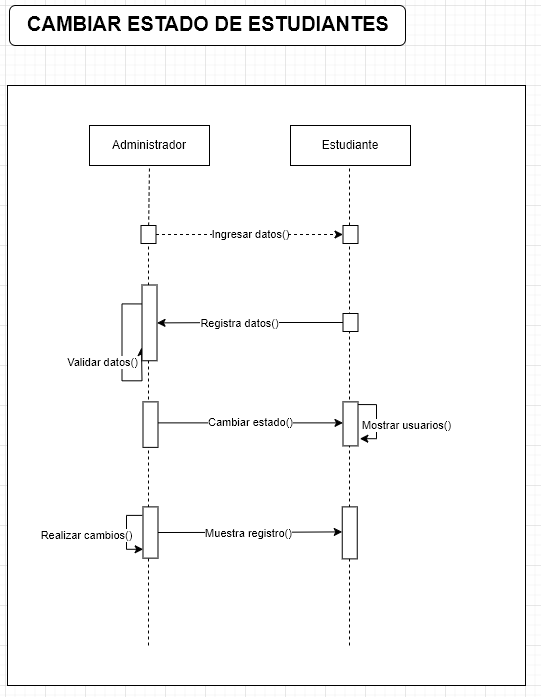
## Vista Lógica

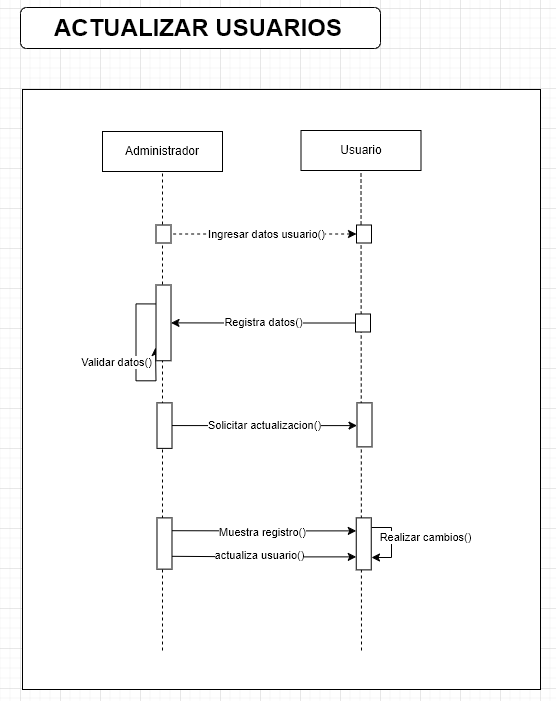
### Diagramas – Clases

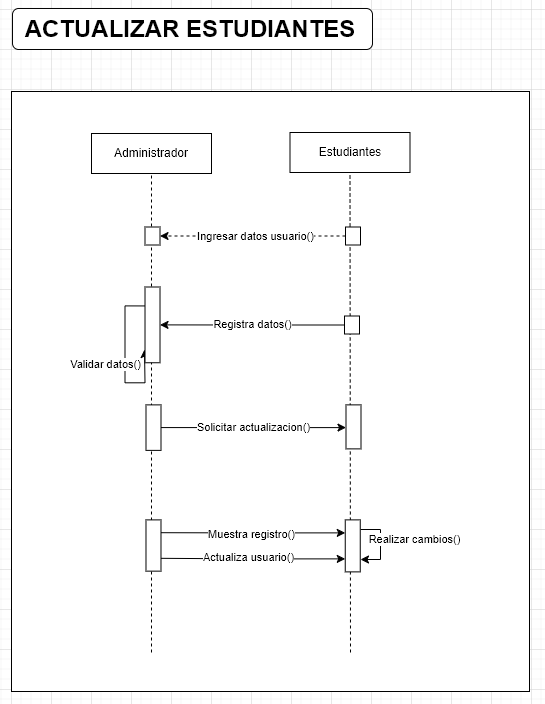


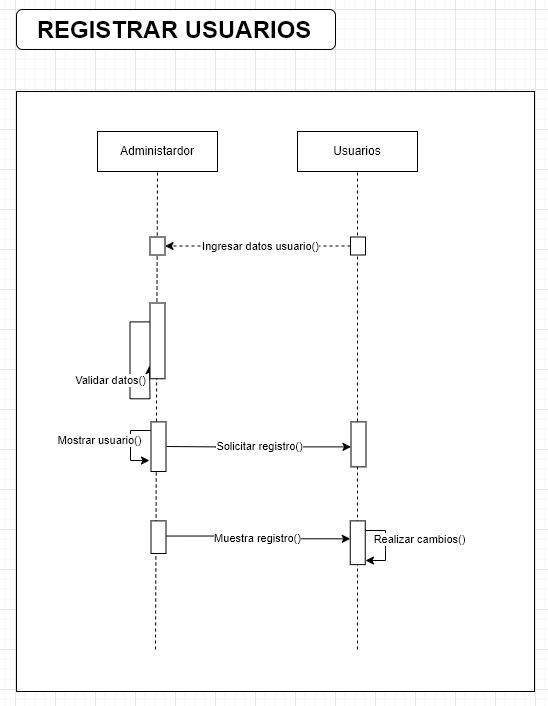
### Diagramas - Secuencia

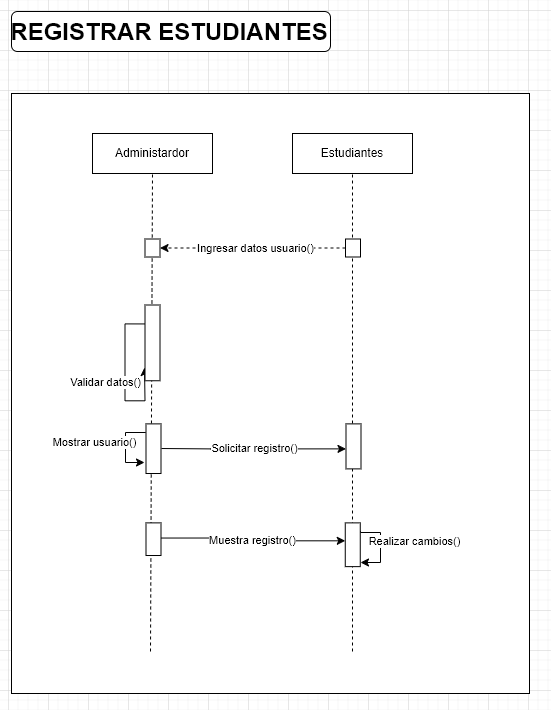


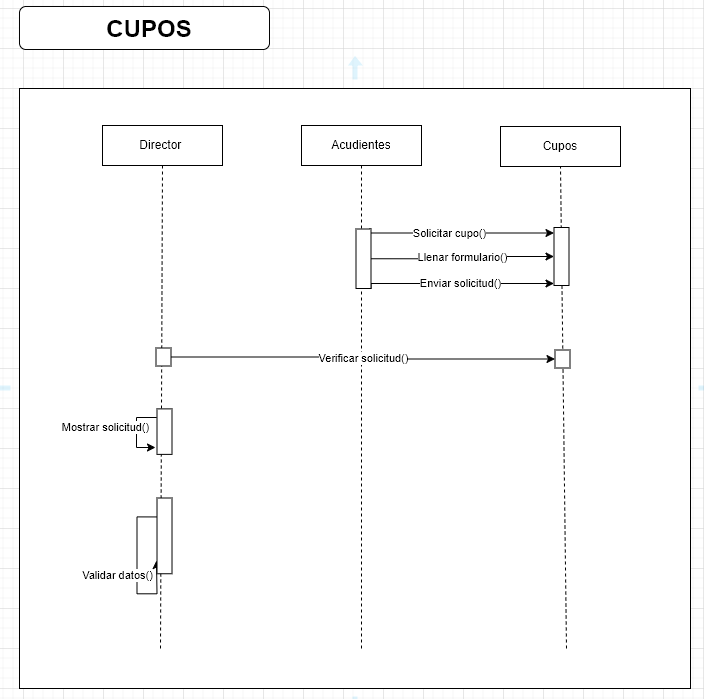


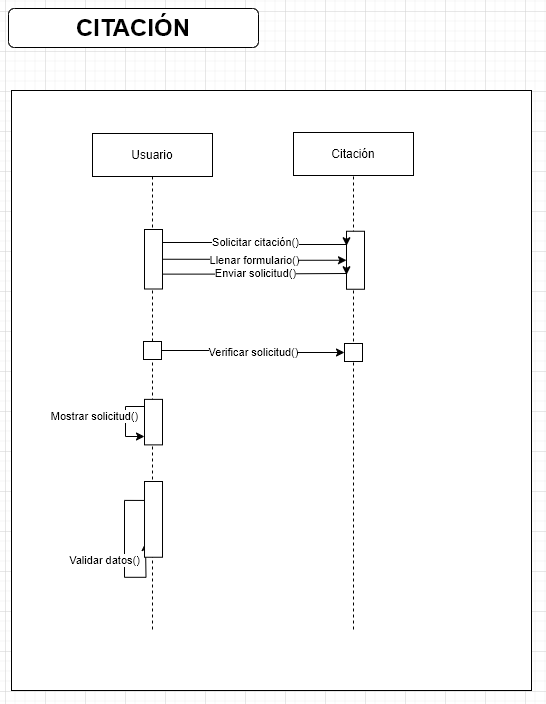






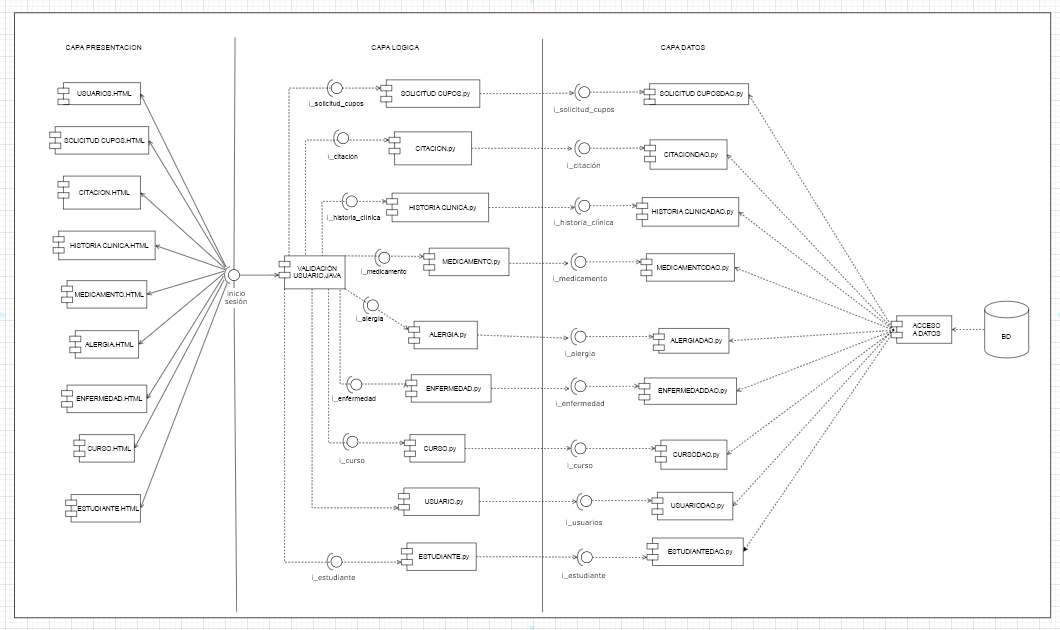




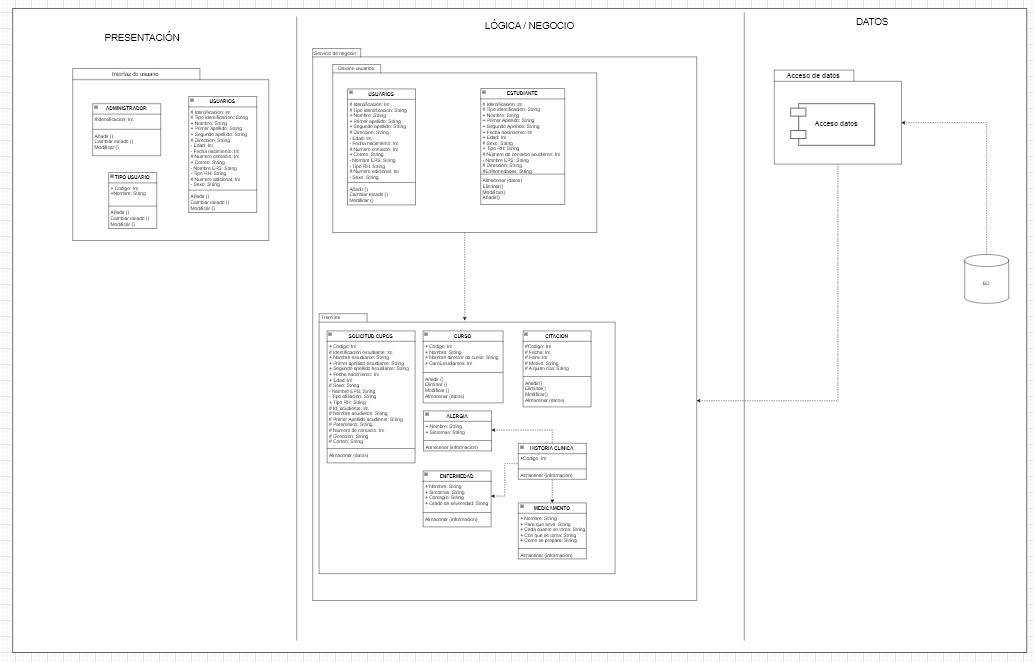


## Vista de Implementación

### Diagrama de Componentes

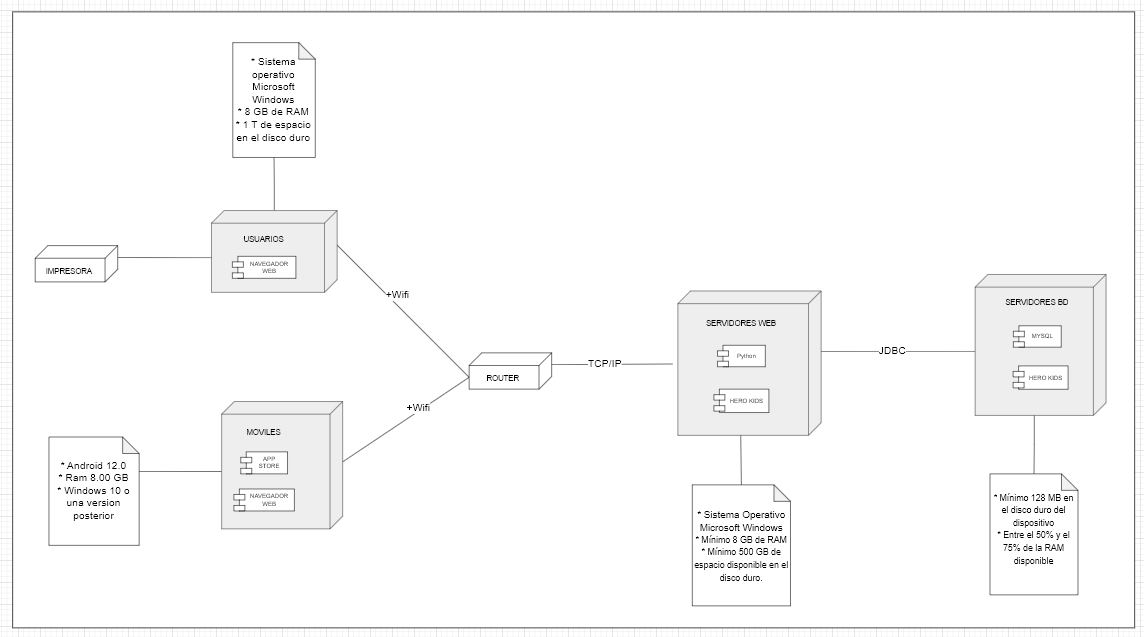


### Diagrama de Paquetes



## Vista de Despliegue

### Diagrama de despliegue



# Arquitectura en capas (N\_Capas, plataforma a utilizar)

Utilizamos el modelo de Vistas “4+1”, para describir la arquitectura de sistemas software, basado en el uso de múltiples vistas. Las vistas suelen describir el sistema desde el punto de vista de diferentes interesados, tales como usuarios finales, desarrolladores o directores de proyecto. Las cuatro vistas del modelo son: vista lógica, vista de desarrollo, vista de proceso y vista física.

Además utilizamos la arquitectura en capas a cual se basa en dividir la aplicación en capas, con la intención de que cada capa tenga un rol muy definido, como podría ser, una capa de presentación (UI), una capa de reglas de negocio (servicios) y una capa de acceso a datos (DAO). Las capas que utilizamos son:

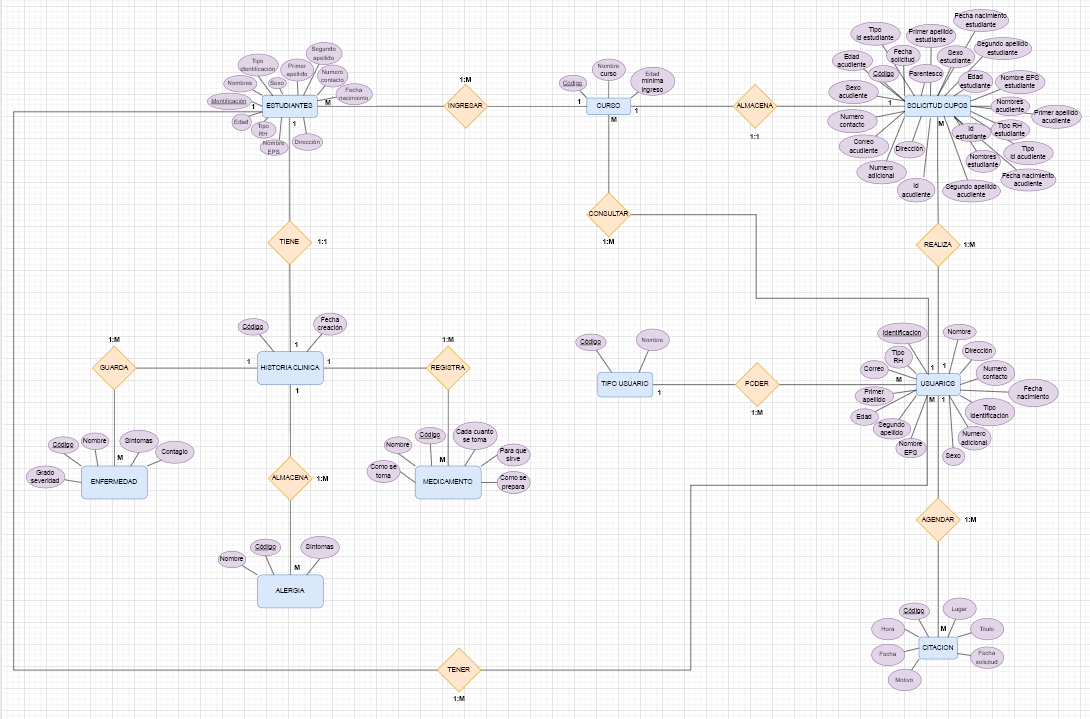
* Capa de presentación
* Capa lógica o de negocio
* Capa de datos

Las plataformas tecnológicas que utilizamos fueron:

* HTML y CSS
* Python
* Visual Studio Code
* XAMPP
* Bootstrap
* GitHub
* MySQL

# Vista de Datos

## Modelo Relacional



# Definición de Interfaces de Usuario

Demostración de las interfaces

[**https://senauti.github.io/gui-gaes-1-hero-kids/**](https://senauti.github.io/gui-gaes-1-hero-kids/)

# Características Generales de Calidad

## Tamaño y performance

* Tiempo de respuesta en el acceso a la Base de Datos

# El tiempo de respuesta en el acceso a la base de datos se demora 2 segundos aproximadamente

* Tiempo de respuesta de transacciones

El tiempo de respuesta de transacciones es entre 5 a 10 seg

* Espacio en disco para el cliente

Para descargar la aplicación en App Store se requiere de un dispositivo móvil de Android 12.0 y una RAM de mínimo 8.00 GB. Para utilizar la aplicación en el navegador web se necesita un sistema operativo Microsoft Windows con mínimo 1 GB de RAM y mínimo 1 GB de espacio disponible en el disco duro.

* Espacio en disco para el servidor de Base de datos

Para utilizar el servidor de Base de datos se necesitan como mínimo 128 MB en el disco duro del dispositivo y entre el 50% y el 75% de la RAM disponible

## Usabilidad

La aplicación permite que el usuario interactúe con ella de manera sencilla, será funcional tanto en dispositivos móviles y servidores web. Además ayudará al usuario a cumplir con las necesidades de manera rápida.

## Eficiencia

La aplicación será rápida a la hora de proporcionarle una respuesta al usuario en el momento que requiera de realizar algún proceso dentro de la aplicación, también contará con varios recursos adecuados para que se puedan resolver las problemáticas del usuario de manera inmediata.

## Seguridad

Los usuarios que interactúen con la aplicación tendrán la seguridad de que su información personal estará protegida y solo la podrán observar y modificar las personas o sistemas autorizados.

## Confiabilidad

La aplicación será confiable a la hora de proporcionar una respuesta al usuario en el momento que requiera de realizar algún proceso dentro de la aplicación, también aportará seguridad a la información personal de los usuarios.

## Mantenimiento

La aplicación se irá actualizando dependiendo de las nuevas funcionalidades del jardín, además se realizarán pruebas en cada actualización y podrá ser modificada en cualquier momento.

## Estándares

**ISO 9001 - Sistemas de Gestión de Calidad:** Se trata de una norma que incide en el enfoque de las empresas hacia el cliente para ofrecer unos productos y servicios de mayor calidad. Un sistema de gestión de calidad ISO 9001 ayuda a las empresas a controlar de forma continuada la calidad en todos sus procesos.

**ISO 25000:** Tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software.