DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA LA GESTIÓN DE ALIMENTOS

KAREN GIMENA ROJAS BOLAÑOS - 67000411

DEIVER CAMILO MENA - 67000248

EDWARD DANIEL LOPEZ ESPITIA - 67000380

DANIEL ALEJANDRO PACHECO ZÚÑIGA - 67000456

CRISTIAN ESTEBAN BLANCO DÍAZ - 67000093

UNIVERSIDAD CATÓLICA DE COLOMBIA

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA DE SISTEMAS

INFORMÁTICA SOCIAL

BOGOTÁ, D.C

2022

**TABLA DE CONTENIDOS**

[**11. Vista física**](#_heading=h.4bhcvlevlqet) **25**

INTRODUCCIÓN

La fundación el Claret se dedica al cuidado de los niños y adolescentes, los cuales se dirigen a ésta para recibir una alimentación. En este centro se manejan los menús y suministradores de estos alimentos en un formato de Excel. Es así como este método no es muy eficiente y/o seguro, ya que se ocasionan pérdidas o modificaciones de esta información.

Por lo anterior, el presente proyecto pretende optimizar el proceso de almacenamiento, mediante la gestión de los menús y suministradores, sistematizándolos con el fin de reducir las pérdidas de documentos y mejorar la seguridad de los mismos, para evitar la pérdida de la información de este.

Este documento proporciona una visión general de la arquitectura del sistema con la que el documento definirá el alcance, los objetivos de la arquitectura, los casos de uso y la representación arquitectónica.

RESUMEN

Un sistema de información que permite optimizar el proceso de almacenamiento mediante el manejo de alimentos sistematizados, para así evitar la pérdida y un buen manejo de la información, para lograr que la comunidad de Hogares el Claret pudiera tener el manejo de la información de una mejor manera.

Este trabajo también permitió llegar a una comunidad vulnerable que buscaba una ayuda en el manejo de su información, generando que tanto administrativos, como los niños poder obtener un menú que se acomodaba a cada uno, dando un alimento adecuado a ellos.

# PLANTEAMIENTO DEL PROBLEMA

En este centro se maneja un sistema de menús en excel con el cual se suministran los alimentos para los infantes, adultos y adolescentes de la fundación. Este método carece en eficiencia y seguridad, puesto a que se ocasionan pérdidas, daños del formato excel o modificaciones de esta información, sin previo aviso.

# OBJETIVOS

A Continuación se explicarán los objetivos a desarrollar.

## Objetivo General.

Desarrollar un sistema de información, el cual se encargue del manejo de alimentos.

## Objetivos Específicos

* Realizar el levantamiento de información e identificar los respectivos requerimientos del cliente para el funcionamiento del sistema.
* Desarrollar e implementar el sistema de información.

# JUSTIFICACIÓN

Mediante los diálogos que se realizaron con la encargada de la fundación, se propuso la idea de un sistema de información, en él se plantea y se busca agilizar, proteger y optimizar los procesos de recolección de datos de la fundación hogares claret para así beneficiar tanto al personal, como a los niños y adolescentes.

Este se fundamenta en como actualmente la fundación hogares claret está guardando y utilizando la información; puesto que esta es propensa a modificaciones de personas no autorizadas y tiene una poca eficiencia y practicidad en cuanto a la relación en sus tablas.

En la Fundación hogares claret se requiere controlar de una mejor manera la gestión de alimentos de los infantes y adolescentes, esto mejorará el manejo de la información evitando su pérdida, ayudando a que la fundación tenga más orden a la hora de manejar sus datos.

## 3.1. Impacto social

El presente trabajo logra beneficiar a La Fundación Hogares el Claret en donde los directivos, niños, jóvenes y trabajadores, teniendo un aproximado de unas siete sucursales en Colombia, siendo así un impacto a gran escala.

Este trabajo generó en nosotros un compromiso y busca de ayuda a los demás, en mejorar cómo se realiza un proceso para que así todo aquel que tuviera la necesidad de utilizar de este pudiera hacerlo de manera facil, comoda y concisa, sin tener ideas erróneas de una persona a otra, sino que esta información fuera clara. Este proyecto nos enseñó cómo, con un sistema de información se puede llegar a generar un cambio a una comunidad, beneficiando los y haciéndonos sentir que con nuestro trabajo, y nuestra carrera se puede generar una ayuda a los demás, dejando de un lado lo que es realizar un sistema en el que solo se trabaja desde una computadora, para permitirnos adentrándonos más en lo que este puede llegar a causar en los demás.

# 4. ALCANCES Y LIMITACIONES

A causa de diversos motivos el equipo de desarrollo de este proyecto cuenta con limitaciones que hacen cambiar los alcances del proyecto.

Por motivos externos el primer director del proyecto Alberto Guzman deja el proyecto. El equipo de trabajo se dirige ahora a la directora de la fundación Ana Milena Trujillo, quien en una reunión establece los siguientes parámetros con los que debe contar el sistema de información:

* Evitar el cuello de botella entre los usuarios y administradores del sistema de información, iniciándose así la creación de permisos a diferentes usuarios.
* El excel a utilizar para crear el sistema de información es usado por más de una casa hogar, por ende el sistema de información también.

El motivo ya antes presentado extiende el tiempo de desarrollo del proyecto. Teniendo en cuenta el poco para desarrollar un sistema de información robusto, para más de una casa hogar, el equipo de trabajo se reunió con el profesor a cargo Rafael Acosta en la cual se acordó reducir en módulos el alcance del proyecto con la condición de que al siguiente semestre, el profesor encargaría a un grupo continuar el desarrollo. El anterior acuerdo con el profesor también fue tratado con la directora Ana Milena Trujillo, quien accedió a esta petición. La documentación pertinente a la petición se encuentra en los anexos (nombre de anexo acta) y (nombre de anexo documento) del proyecto.

Con la aclaración de las limitaciones y los acuerdos establecidos con el profesor Rafael Acosta y la directora Ana Milena Trujillo el alcance del presente proyecto es el siguiente:

## 4.1 PLANIFICACIÓN

* **Fase de Análisis - 100%**

La fase de análisis en la cual se cubren requerimientos funcionales, no funcionales, casos de uso y vistas de datos se realiza a plenitud.

* **Fase de Diseño - 100%**

La fase del diseño en la cual se desarrolla una página web en la que se gestione el sistema de información cuenta con toda la integración del porcentaje de la programación más un diseño no funcional del esquema del menú, para que así el siguiente grupo tenga una breve descripción de cómo se planea elaborar la visualización de los menús

* **Fase de Programación - 60%**

La fase de la programación es de las más importantes en el desarrollo del sistema de información y es de las más afectadas por los limitantes, por ello su alcance se taza en un 60% la cual conlleva en sí la base de datos, sistema de usuarios (crear, editar, dar permisos), separación de las casas hogar en la base de datos.

Debido a los motivos antes aclarados el siguiente grupo a continuar con el desarrollo del sistema de información deberá concluir con las siguientes fases:

* **Fase de desarrollo - 40%**

Debido a que el primer grupo de desarrollo del sistema de información plantea ejecutar el 60% , el grupo a continuar esta labor realizará el porcentaje correspondiente para llegar así al 100% de esta fase.

En esta fase el equipo a continuar se encargará de realizar la lógica de los menús con lo antes establecido, generar una orden de compra y gestionar datos pertinentes en la base de datos.

* **Fase de pruebas -100%**

Para esta fase el equipo a continuar el desarrollo deberá encargarse de corregir errores en la fase anterior.

* **Fase de despliegue**

Por último, en la fase de despliegue el grupo a continuar el desarrollo del aplicativo deberá enseñar a la organización como emplear el sistema de información creado.

**5. ESTADO DEL ARTE**

Mediante investigaciones realizadas para tener unas bases y fundamentos que se van a retroalimentar con conocimientos previos, se expone el estado del arte, donde y con algunos antecedentes relacionados a temas de almacenamiento de información, interfaz, representación arquitectónica, entre otros daremos a conocer estudios que corroboran a nuestra elección de los distintos aplicativos utilizados. Esto se dividió en 3 items:

**5.1 Aplicativos en representación de la arquitectura.**

**5.2 Aplicativos en representación de la interfaz.**

**5.3 Aplicativos en almacenamiento de información.**

Por otra parte, el estado del arte también contempla la toma de decisiones y la importancia que tiene la responsabilidad social universitaria, por lo que se va a identificar los principios y pasos a seguir para que la responsabilidad social no se quede solo en palabras y pueda ser realmente un hecho.

Desde luego la responsabilidad social se desarrolla cuando un organismo toma consciencia de sí mismo, de su entorno y de su papel en el entorno, presuponiendo así la superación de un enfoque egocéntrico. Además se trata de una [[1]](#footnote-0)voluntad ética e interesada a la vez de hacer las cosas “bien” para que todos los beneficiarios internos y externos de los servicios organizacionales se encuentren correctamente.

Se busca que la responsabilidad social universitaria deje atrás la marginalización institucional implementando en estas iniciativas como lo son los voluntarios, humanitarios, entre otros que luchan por la debilidad de un vínculo con la formación profesional, para así lograr que la didáctica universitaria logre una integración de la proyección social en el corazón, con el fin de controlar su calidad humanitaria.

La R.S.U. También exige que en su visión se puede articular las diversas partes de la institución como un proyecto que proporcione, de manera social los principios éticos y que así permita un desarrollo equitativo y sostenible, permitiendo la reducción y transmisión de la responsabilidad. En cuanto a los profesionales como ciudadanos, se busca igualmente responsables. Por lo que este esquema pide ayuda a visualizar el carácter global y central de la reforma desde la parte de la Universidad.

En conclusión, tiene como meta el orientar hacia la transformación de la Universidad en una pequeña comunidad ejemplar de democracia, teniendo así una equidad y sobre todo la segregación y corrección de los privilegios, para permitir una transformación, siendo política y económica; dando como esta un modelo de desarrollo sostenible y política que proporcione la protección al medio ambiente, como lo es con el uso del papel reciclado, tratamiento de las desechos entre otros, para tener así, pertenece a utilizando las herramientas.

Por lo que la responsabilidad social universitaria es más que un proyecto social, estaba generando cambios y no sólo en un comportamiento y gestión de cara a la sociedad, sino también hacia adentro de esta misma.

## 5.1 Aplicativos en representación de la arquitectura.

En este ítem se aborda [[2]](#footnote-1) El modelo 4+1 en el que se describe la arquitectura del software usando cinco vistas concurrentes, en donde para cada vista definimos un conjunto de elementos (componentes, contenedores y conectores), en donde captamos la forma y los patrones con que trabajan, y captamos la justificación y las restricciones, relacionando la arquitectura con algunos de sus requisitos.

En este encontramos la vista lógica que describe el modelo de objetos, la vista de procesos para describir los aspectos de concurrencia, la vista física (el mapa del software) y la vista de desarrollos con la organización del software en su ambiente de desarrollo, y por último los escenarios en los que los elementos de las cuatro vistas trabajan conjuntamente en forma natural.

## 5.2 Aplicativos en representación de la interfaz.

Se implementó un sistema de base de datos, en donde en la interfaz y el motor de base de datos, en donde el servicio SQL es el principal para almacenar, procesar y proteger los datos.

[[3]](#footnote-2)SQL (por sus siglas en inglés Structured Query Language; en español lenguaje de consulta estructurada) es un lenguaje de dominio específico, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales.

## 5.3 Aplicativos en representación de la información.

Se realizó una investigación de aplicativos que permitan realizar la misma función que nuestro sistema de información, aun así, no se logra evidenciar uno que cumpla con todos los requerimientos que el nuestro brinda.

Para permitir la realización de este sistema de información se tuvo que tener en cuenta el manejo de los alimentos, entre otros requerimientos como los son el manejo de las medidas de las distribuciones de cada tipo tanto para los niños como para los adolescentes. Para esto también se tuvo que analizar las tablas de composición de alimentos, para permitir una ingesta recomendada de acuerdo a lo anterior comentado y con sus instrumentos básicos de acuerdo a una dieta.

Por otra parte, se implementan las tablas como una herramienta dinámica para la edición, en la que se va a poder realizar y ampliar el compromiso nutricional de los alimentos. Estas tablas también se plantearon de acuerdo a los días de la semana y semanas del mes, teniendo así un menú variado y único para cada semana, sin repetir la siguiente, para que así la alimentación sea más agradable para los niños y adolescentes.

# 6. REPRESENTACIÓN ARQUITECTÓNICA

Se detalla la arquitectura utilizada, las vistas del modelo de 4+1 de Kruchten, estas vistas utilizadas para documentar el sistema.

## 6.1 Vista Lógica

En esta vista se representará la estructura y funcionalidad del sistema que concede a los usuarios, los servicios que debe ofrecer, lo que el sistema debe hacer, las funciones, etc.

Este proyecto representa la funcionalidad de un sistema que proporciona a los usuarios sus historiales y datos clínicos. El sistema de información se encargará de controlar el manejo del menú de los niños y adolescentes; también del manejo de los proveedores.

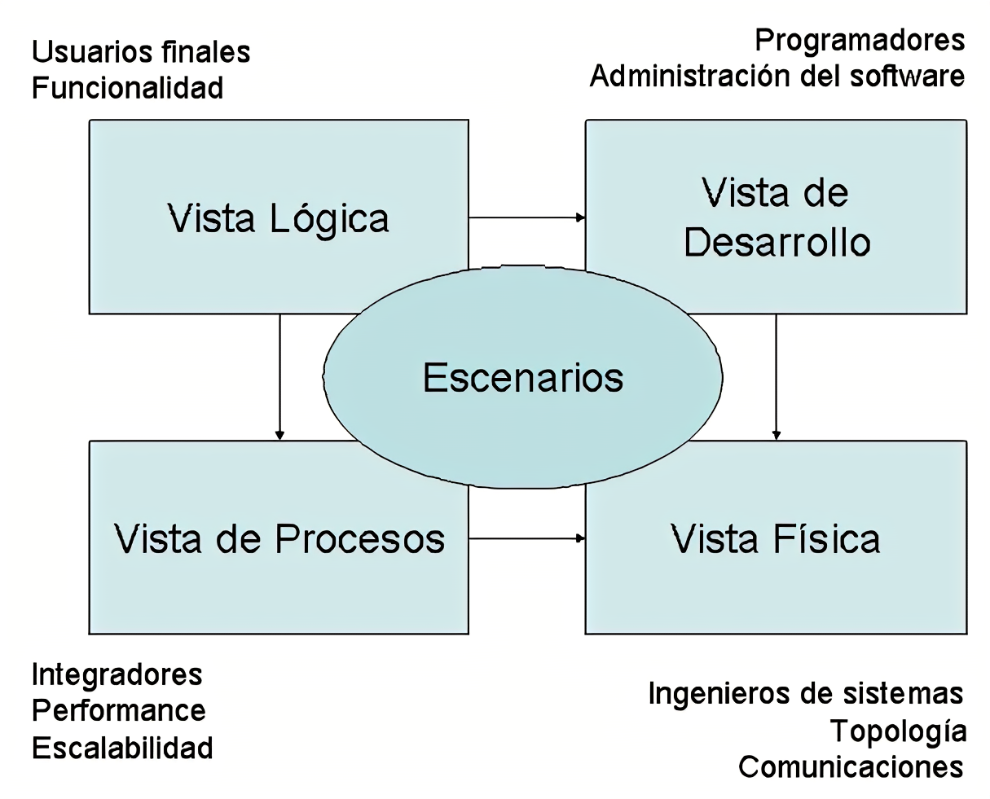
****

Figura 1. Vista Lógica

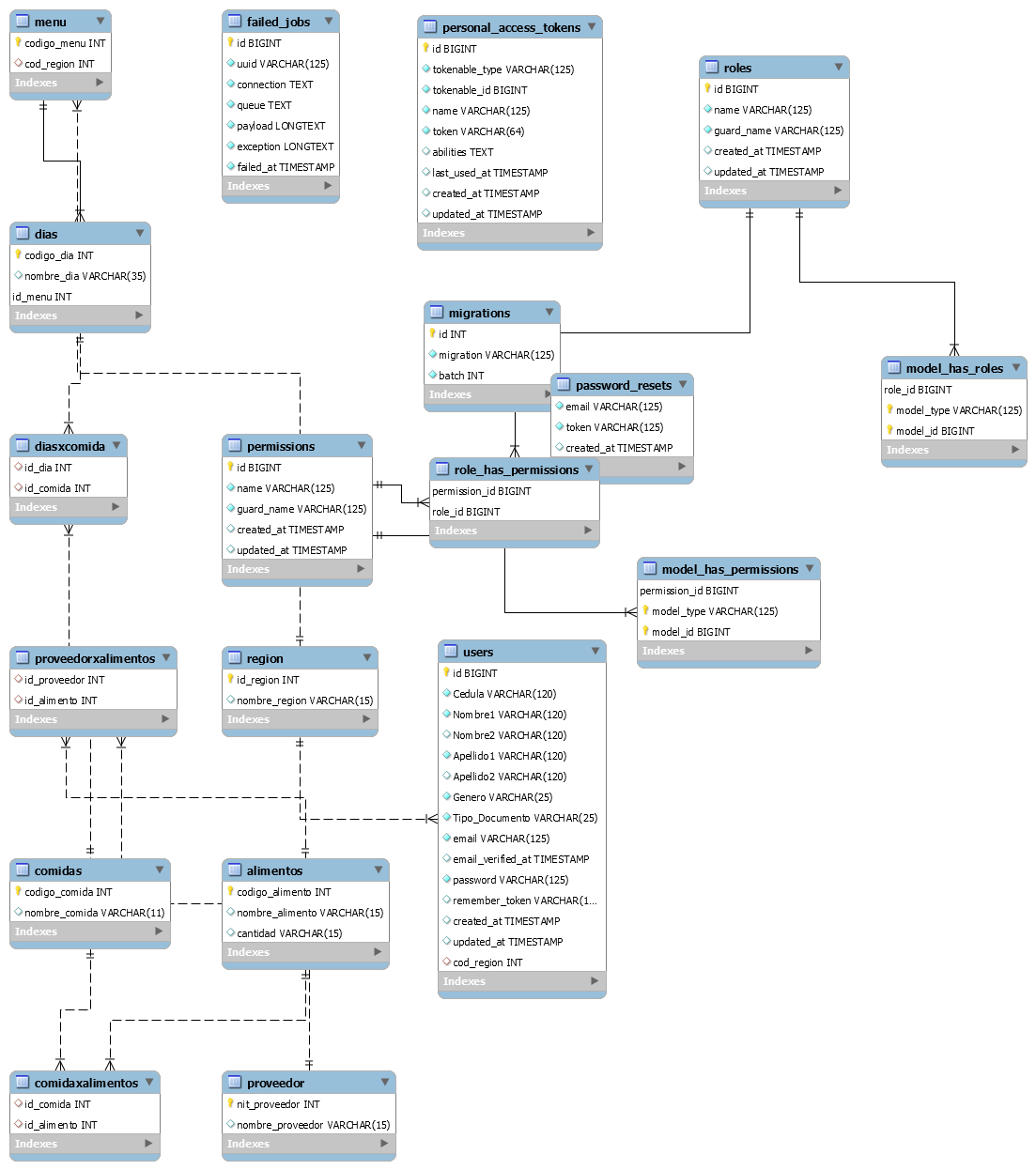
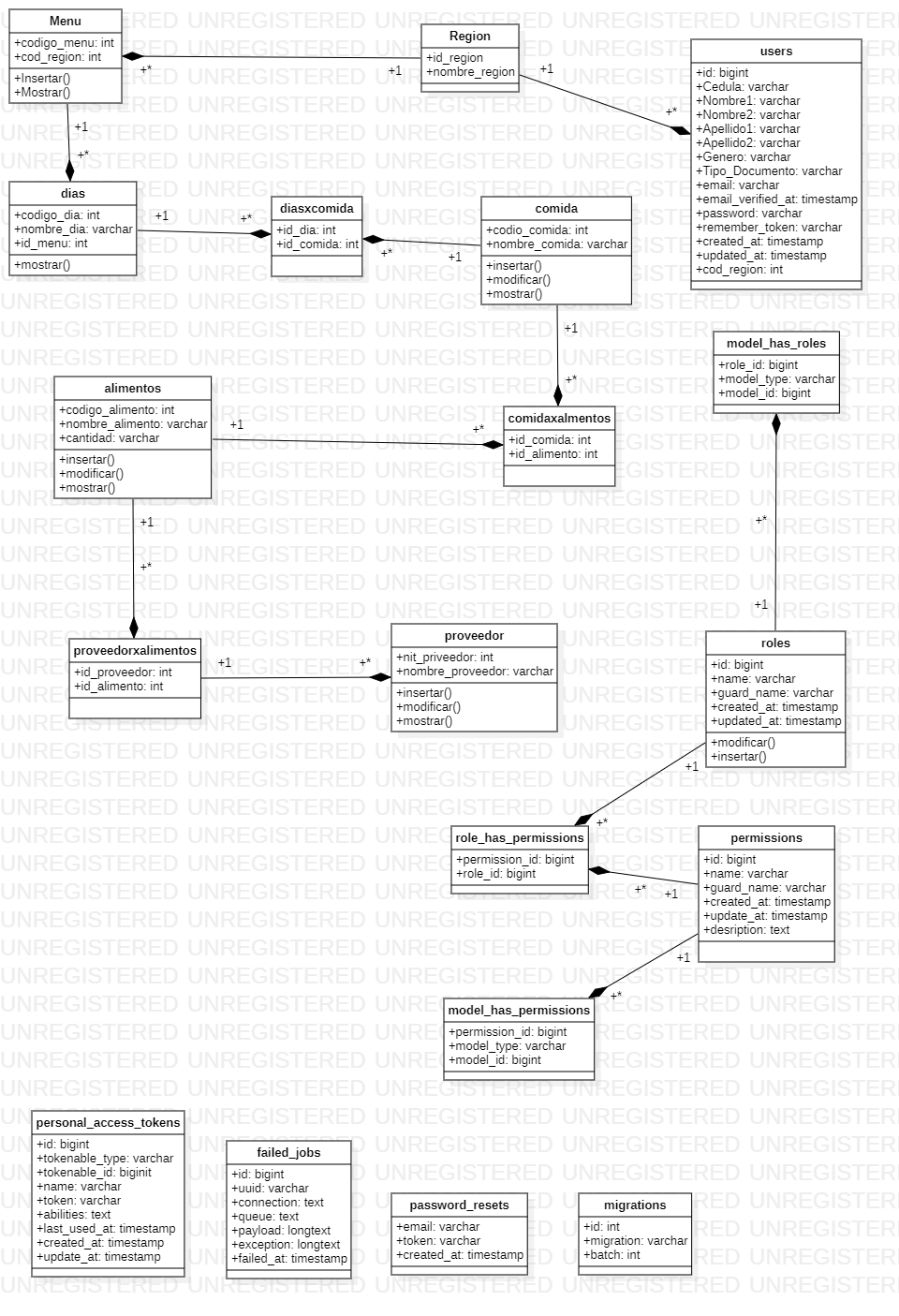
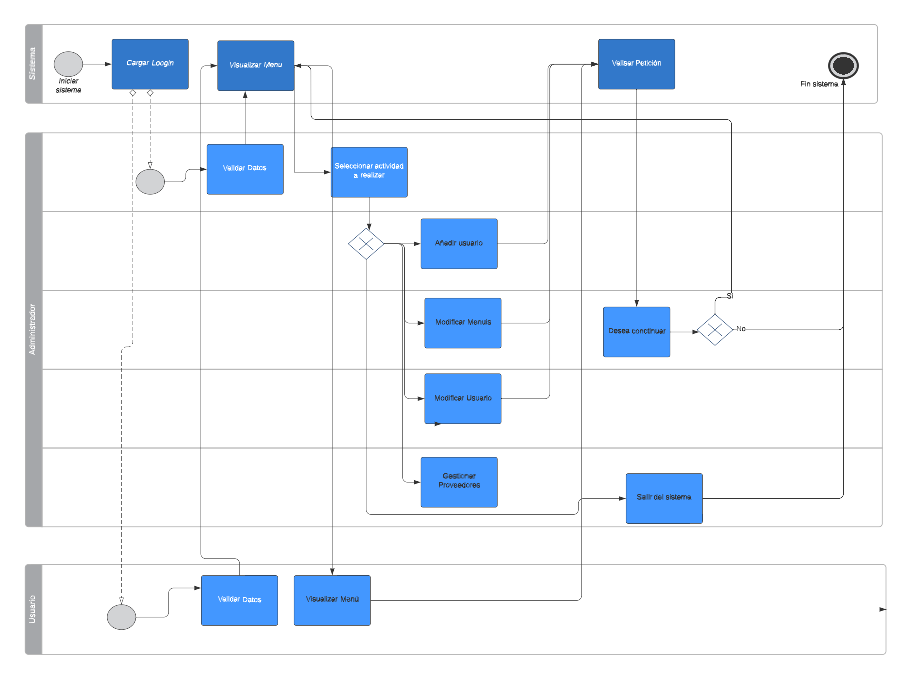


diagrama de clases



## 6.2 Vista de Procesos

Este proceso de vista explica cómo se comunican los sistemas entre sí, para esto se va a utilizar MySQL para el archivo de los datos, ya que esta consideración se requiere para el desarrollo, además de Laravel de PHP para el backend. Lo más importante es que el sistema se comporte de una manera eficiente cuando esté en ejecución para que los aspectos de concurrencia como los son distribución y rendimiento integren adecuadamente el sistema.

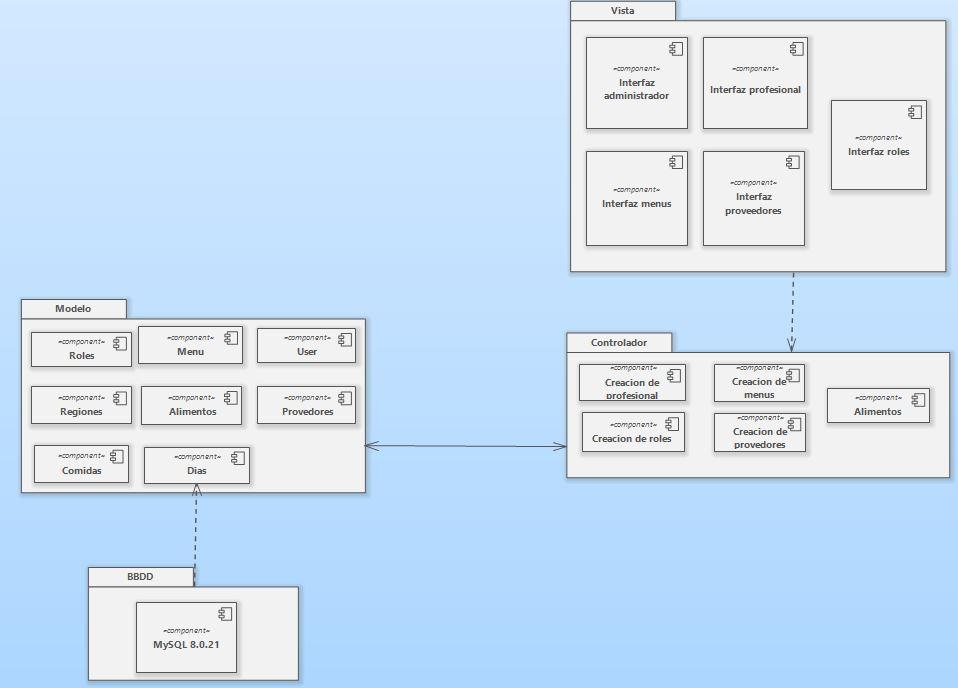


## 6.3 Vista de Desarrollo

Se ocupa de la gestión del software, en esta vista de desarrollo se centra la organización real de los módulos de software en el ambiente de desarrollo del software, allí se explica el sistema desde la óptica del programador describiendo los componentes y dependencias entre ellos. Entre los componentes se encuentran características como: que sea reutilizable, intercambiable y que posea interfaces definidas. Esto ayudará a que la programación sea más organizada, a que se establezcan relaciones entre clases que no estén relacionadas y a que algunas clases utilicen los mismos métodos.

En esta sección se describe la organización de los módulos de software necesarios para el entorno de desarrollo. El sistema debe interactuar con una base de datos, por lo tanto en el diagrama de componentes basándose en la arquitectura MVC tendremos el modelo, la vista y el controlador. En la figura 17 se muestra el diagrama de componentes del sistema.

**Resumen**

****

* **Componente Vista**

Este componente es el responsable de la presentación de la información y del modelo de una manera adecuada para la interacción, ya que este genera las diferentes interfaces de usuario por medio de las cuales el usuario interactúa con el sistema.

* **Servicios prestados**: genera y despliega las diferentes vistas que el usuario requiere.
* **Servicios requeridos**: Servicios prestados por el controlador.
* **Componente Controlador**

Este se encarga de responder y procesar los eventos generados por el usuario del sistema, además debe invocar los cambios que se generen en el modelo y la vista.

* **Servicios prestados**: Gestiona eventos de entrada y salida del sistema.
* **Servicios requeridos**: Requiere de los servicios del modelo y la vista.
* **Componente Modelo**

Este contiene toda la lógica del sistema, las consultas que pueden ser realizadas sobre los datos de este.

* **Servicios prestados**:Contiene la funcionalidad y los datos más relevantes para el usuario, los cuales son necesarios para la presentación de las vistas.
* **Servicios requeridos**: Requiere de los servicios del controlador.
* **Componente BBDD**

Este componente es el encargado de permitir la comunicación entre la lógica y la base de datos del sistema.

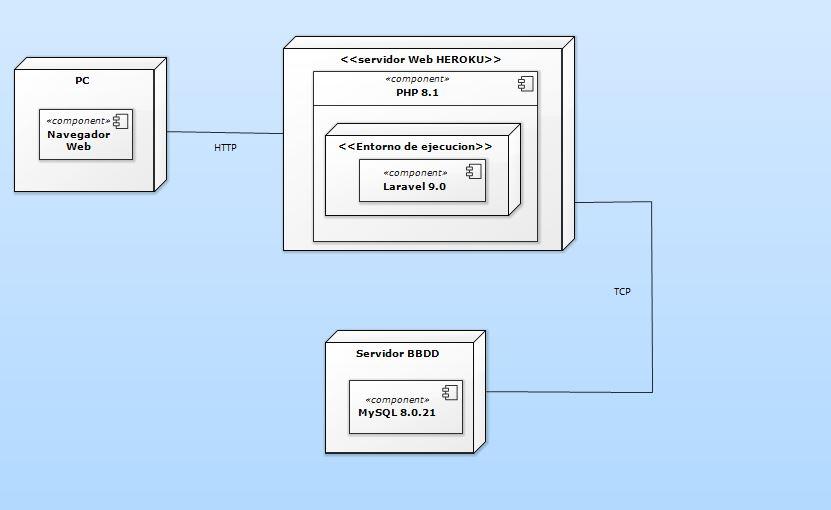
* **Servicios prestados**: Gestiona la comunicación entre el componente BBDD y el Componente Modelo.
* **Servicios requeridos**: Requiere de los servicios del modelo que contiene los procedimientos necesarios.

## 6.4 Vista Física

En esta se busca representar la distribución de los componentes entre los distintos equipos que conforman en los diagramas de despliegues en el cual se relacionan los diferentes nodos que serán usadas en el aplicativo de reservas medidas, generando la separación por capas que ayuda a visualizar el proyecto, en función con página web, su aplicativo y la base de datos.

Describe el sistema desde el punto de vista de un ingeniero de sistemas. Incluye todos los componentes físicos del sistema, así como las conexiones entre estos componentes que conforman la solución (incluyendo servicios).

En esta sección la vista de implementación o física será representada por medio de un diagrama de despliegue donde se distribuyen los componentes entre los distintos equipos, como se ve en la figura 16.

****

El sistema está diseñado para que se ejecute en un servidor en este caso Heroku que posee un contenedor web para el sistema, el entorno de ejecución será realizado en laravel en su versión 9.0 y este tendrá una conexión a un servidor para la BBDD en el cual se utilizará MySQL en su versión 8.0.21.

* **PHP 8.1:** El procesador de php sobre el que se ejecuta el sistema de información para la gestión de alimentos, es el subsistema base que permite ejecutar todas las instrucciones del programa.
* **Laravel**: Es el framework que conforma la base sobre la que se construye la aplicación. Facilita y acelera el desarrollo mediante sus funciones ya implementadas, permitiendo centrarse en el desarrollo de la aplicación.
* **Navegador Web**: Sirve de ventana entre el usuario y la aplicación construida. El sistema ha sido diseñado para ser ejecutado en un servidor web, por lo que los clientes que se conecten a la misma utilizaran un navegador web como primera herramienta para utilizarla.
* **Servidor BBDD**: Conforma un entorno físico sobre el que ejecutar un sistema de gestión de base de datos, se comunicara con el servidor web a través del protocolo TCP. Este servicio podría ser provisto desde el mismo hardware que el servidor web Heroku.
* **MySQL 8.0.21**: Es el SGBD que se ha elegido para el desarrollo y puesta en producción. Sirve como el almacén de información principal del sistema, mediante el cual se pueden obtener los datos necesarios para gestionar las peticiones que llegan al sistema.

# 7. VISTA DE CASOS DE USO Y ESPECIFICACIÓN DE REQUERIMIENTOS

Antes de entrar en detalle de cómo se desarrollará mediante frameworks y otras herramientas la aplicación , los ingenieros deben tener totalmente claro cómo interactúa el usuario con la aplicación que plantea realizar, por ello debemos realizar un diagrama de casos de uso en la que se define gráficamente la interacción del sistema y el usuario.

Casos de uso

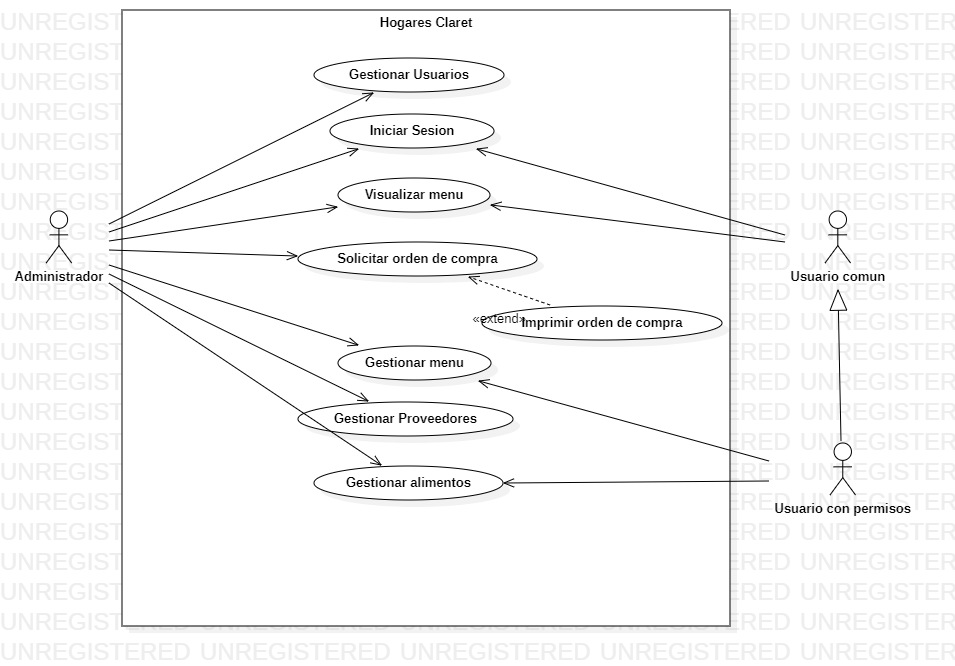
****

Figura 2: Diagrama casos de uso del sistema.

## 7.1 ACTORES DEL SISTEMA

**Administrador:** este autor es aquel que puede gestionar los menús del sistema, a los usuarios, proveedores y alimentos, también tener la posibilidad de imprimir una orden de compra.

**Usuario Común:** será el encargado de visualizar el menú y navegar en las diferentes pestañas del sistema sin poder realizar modificaciones.

**Usuario con privilegios:** Este actor cuenta con los mismos privilegios del usuario común, con la diferencia de poder gestionar menús y alimentos.

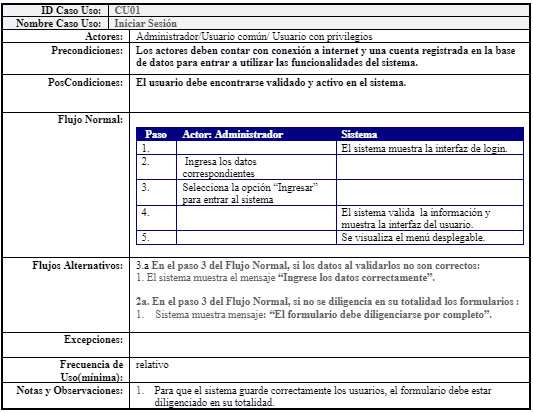
**Sistema:** Es el actor que ejecuta acciones que pide el usuario, en base a la programación y modelos de vistas creados.

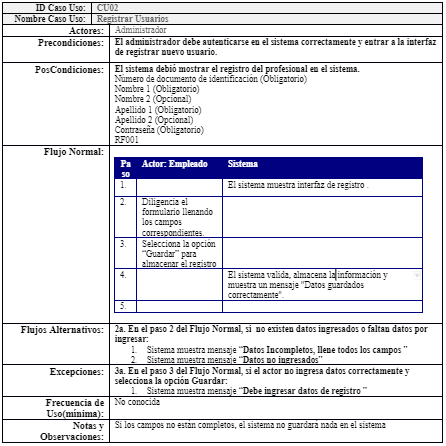
## 7.2. REQUERIMIENTOS FUNCIONALES

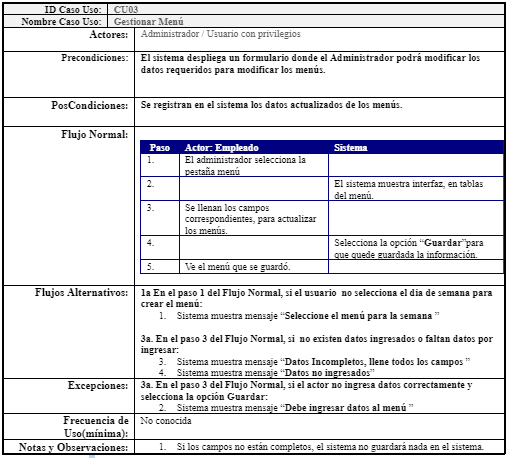
A Continuación se mostrará el listado de los requerimientos funcionales y los casos de uso a los que accede cada actor en el sistema.

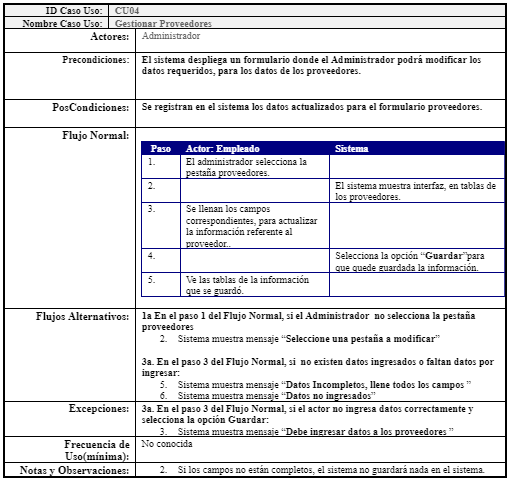
| **ID requerimiento** | **Actor** | **Caso de uso** |
| --- | --- | --- |
| CU01 | * Administrador * Usuario Común * Usuario con privilegios | Iniciar Sesión |
| CU02 | * Administrador | Registrar Usuario |
| CU03 | * Administrador * Usuario con privilegios | Gestionar Menú |
| CU04 | * Administrador * Usuario con privilegios | Gestionar Proveedores |
| CU05 | * Administrador * Usuario Común * Usuario con privilegios | Visualizar Menú |
| CU06 | * Administrador | Solicitar orden de compra |
| CU07 | * Sistema | Imprimir orden de compra |
| CU08 | * Administrador * Usuario con privilegios | Gestionar alimentos |

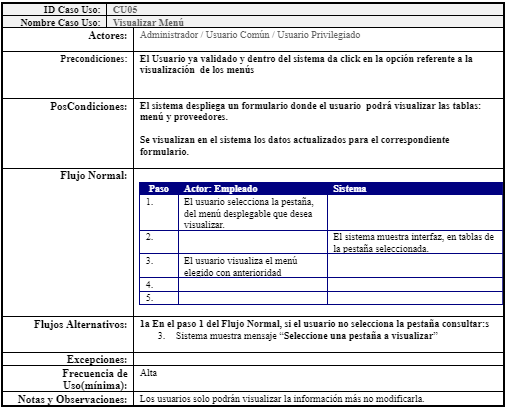
## 7.3 Especificación de requerimientos funcionales

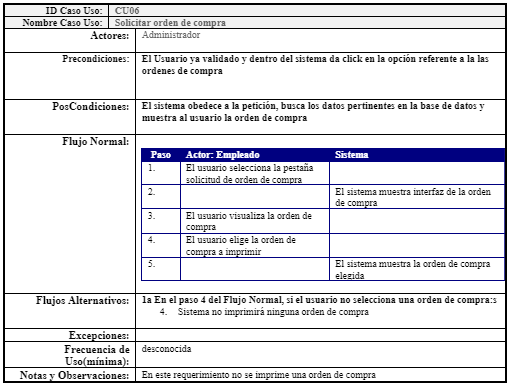


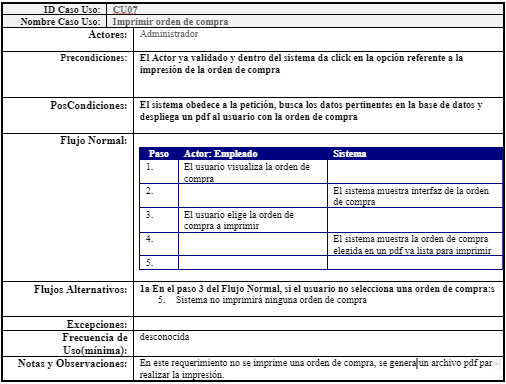


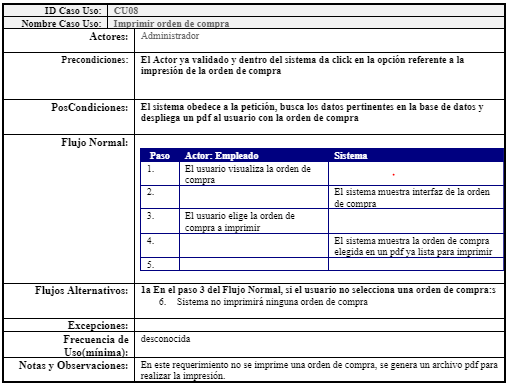












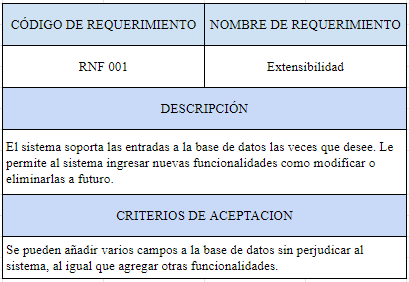
# 8. REQUERIMIENTOS NO FUNCIONALES

Antes de entrar en detalle de los requerimientos no funcionales, se mostrará el listado de estos.

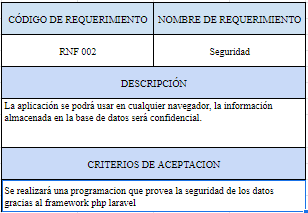
* RNF 001 - Extensibilidad
* RNF 002 - Seguridad de la aplicación
* RNF 003 - Confiabilidad
* RNF 004 - Usabilidad
* RNF 005 - Concurrencia
* RNF 006 - Portabilidad
* RNF 007 - Rendimiento
* RNF 008 - Mantenibilidad
* RNF 009 - Adecuación funcional
* RNF 010 - Escalabilidad

## 8.1 Especificación de requerimientos no funcionales

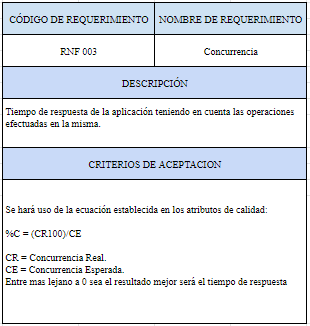
RNF 001 - Extensibilidad

****

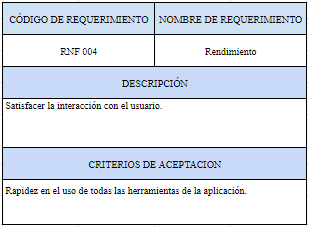
RNF 002 - Seguridad de la aplicación

****

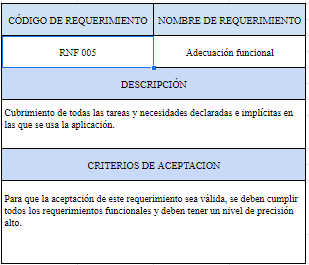
RNF 003 - Concurrencia

****

RNF 004 - Rendimiento

****

RNF 008 - Adecuación funcional

****

**BIBLIOGRAFÍA**

* <https://iconk.org/docs/guiaea.pdf> pagina 17
* <http://materias.fi.uba.ar/7510/practica/zips/Modelo4_1Krutchen.pdf>
* [draw.io](https://www.draw.io/)

1. [15939 Realidad y reflexion.indd (redicces.org.sv)](http://redicces.org.sv/jspui/bitstream/10972/91/1/Responsabilidad%20social%20universitaria.pdf) [↑](#footnote-ref-0)
2. Kruchten, P. (1995). Planos Arquitectónicos: El Modelo de 4+ 1 Vistas de la Arquitectura del Software. *IEEE Software*, *12*(6), 42-50. [↑](#footnote-ref-1)
3. Perez, M. (2011). *Microsoft SQL Server 2008 R2. Motor de base de datos y administración*. RC Libros. [↑](#footnote-ref-2)