

**Desarrollo de un prototipo de software para la gestión y monitoreo de programas
de alimentación en entidades del sector público y privado.**

Andrés Felipe Barajas Wellman y Juan Pablo Jiménez Gómez

Trabajo de Grado para Optar al Título de Ingeniería de Sistemas

DIRECTOR

Msc. Emilio Justiniano Cárcamo Troconis

Escuela de Ingeniería de Sistemas e Informática

Universidad Industrial de Santander

Facultad de Ingeniería Físico-Mecánicas

Escuela de Ingeniería de Sistemas e Informática

Ingeniería de Sistemas

Bucaramanga

2023

Dedicatoria

Quiero expresar mi sincero agradecimiento a mi compañero Juan Pablo por su inquebrantable compromiso y dedicación a lo largo de este proyecto. Su constante entusiasmo, ideas innovadoras y arduo trabajo fueron fundamentales para llevar a cabo este trabajo de manera exitosa. Su colaboración y habilidades complementaron perfectamente a las mías, lo que nos permitió superar obstáculos y lograr resultados excepcionales.

Además, no puedo dejar de agradecer a mis familiares por su apoyo constante a lo largo de este viaje académico. Sus palabras de aliento, paciencia y comprensión han sido fundamentales para mantenerme enfocado y motivado durante la realización de este proyecto. Sin su apoyo incondicional, no habría sido posible llegar a este punto. Su amor y confianza han sido un pilar fundamental en mi camino hacia la culminación de este proyecto de grado.

Gracias a todos ellos tanto familiares y amigos cercanos, he tenido la fortuna de contar con un sólido respaldo tanto en el aspecto académico como en el personal. Estoy sinceramente agradecido por todos los momentos compartidos y por haber formado parte de esta experiencia enriquecedora. Este logro es el resultado del esfuerzo colectivo y el apoyo inquebrantable de las personas que me rodean. A todos, ¡les expreso mi más profundo agradecimiento!

Andrés Felipe Barajas Wellman

Dedicatoria

El siguiente trabajo de grado no es solo mi triunfo, sino también el de todas las personas (familiares, profesores y amigos) que creyeron en mí hasta el final. Un agradecimiento a mis padres por haberme dado la oportunidad de tener una buena educación y de darme la libertad de ser quien soy. A mi hermano César Luis por haberme guiado como mi amigo, mi mentor y mi padre, entendiendo siempre mis circunstancias e impulsándome a ser mejor, colocando siempre su fe en mí sin dudar jamás. Quiero agradecer de manera especial a mi compañero de grupo Andrés Barajas, porque el camino no fue fácil, los obstáculos fueron demasiados, y aún con muchas cosas en nuestra contra, hicimos una dupla increíble, recalco que su apoyo, dedicación e ingenio fueron imprescindibles para la realización de este proyecto. Agradezco también a mis dos hermanos el ing. Juan Camilo Marín, y futuro ing. Sergio Andrés Sánchez porque a pesar de no compartir sangre, nuestra hermandad y amistad nos llevó hasta ese punto. Gracias por haberme acompañado en todo este camino, por haberme enseñado tanto, por jamás haber dudado de mí persona, y por enseñarme a vivir. Finalmente agradezco de manera especial a Margarita Toloza, mi razón de ser y existir, sé que no puedes verme en estos momentos, pero solo quiero decirte que todo esto es gracias a ti, que puedes sentir todo el orgullo de tu nieto tal y como siempre lo recalcaste, y que por fin me siento feliz, contento y en paz.

Juan Pablo Jiménez Gómez

Agradecimientos

Manifestamos nuestro más sincero reconocimiento a la Universidad Industrial de Santander, por darnos las herramientas para adquirir conocimientos que fueron la base para el desarrollo de este proyecto

Expresamos nuestra gratitud de manera especial hacia el profesor Emilio Justiniano Cárcamo, cuya orientación ha sido importante para la exitosa realización de este trabajo. Las revisiones críticas y motivadoras que hemos realizado junto a él, han ampliado nuestra comprensión en áreas fundamentales de la carrera.

Nos gustaría destacar con profundo agradecimiento a nuestros padres y hermanos, cuyo respaldo incondicional, y afecto han constituido han sido una motivación inigualable a lo largo de nuestro camino académico.

Extendemos nuestro agradecimiento a todas aquellas personas que con las mejores de las intenciones han aportado en este proyecto, ya sea a través de consejos o compartiendo experiencias. Este logro es fruto del trabajo en equipo.

Tabla de contenido

Introducción	18
1 Justificación o planteamiento del problema	19
2 Objetivos	21
2.1 Objetivo general	21
2.2 Objetivos específicos.....	21
3 Marco de referencia.....	21
3.1 Marco teórico	21
3.2 Estado del arte	26
4 Fase de planeación	32
4.1 Marco metodológico	32
4.2 Definición de tecnologías.....	42
4.3 Proceso de diseño web para el Frontend del sistema	47
5 Fase de desarrollo e integración	55
5.1 Desarrollo del Backend	55
5.2 Desarrollo e implementación del Frontend	63
6 Fase de pruebas	74
6.1 Pruebas unitarias de desarrollador.....	74
6.2 Pruebas para rendimiento de peticiones http con Jmeter	74
7 Conclusiones	81
8 Trabajo futuro.....	83
Lista de referencias.....	84

SOFTWARE PARA GESTIÓN DE PROGRAMAS DE ALIMENTACIÓN

6

APÉNDICE..... 91

Lista de figuras

Figura 1. Funcionamiento del marco de trabajo ágil Scrum.	24
Figura 2. Estructura de manejo de JWT.....	25
Figura 3. Diagrama de gestión de procesos de los programas de alimentación escolar en Colombia.....	29
Figura 4 Revisión del marco metodológico	33
Figura 5 Diagrama de casos de uso para el prototipo de software enfocado a los programas de alimentación	35
Figura 6 Diagrama de funcionalidades para el software con conexión entre tecnologías	36
Figura 7 Base de datos relacional del nuevo programa de alimentación	38
Figura 8 Diagrama de taxonomías	43
Figura 9 Single Page Application(SPA).....	44
Figura 10 Angular Modelo Vista Controlador (MVC)	45
Figura 11 Conexión entre Angular y Node.js	46
Figura 12 Ecosistema de Node.js	47
Figura 13 Wireframe para el inicio de sesión del software.....	48
Figura 14 Wireframe para la interfaz de inserción de usuarios del software.....	48
Figura 15 Wireframe para la interfaz que muestra la lista de usuarios y permite actualizar información de estos.....	49
Figura 16 Wireframe para la interfaz de muestra la lista de servicios alimenticios.....	49

Figura 17 wireframe para la interfaz de permite crear y actualizar los servicios alimenticios	50
Figura 18 Wireframe para la interfaz de la lista de asistencia y control de usuarios del software	50
Figura 19 Wireframe para la interfaz de las gráficas (histórico) de consumo del software	51
Figura 20 Mockup para el inicio de sesión del software	52
Figura 21 Mockup para la interfaz de inserción de usuarios del software	52
Figura 22 Mockup para la interfaz que muestra la lista de usuarios y permite actualizar información de estos	53
Figura 23 Mockup para la interfaz de muestra la lista de servicios alimenticios.....	53
Figura 24 Mockup para la interfaz de permite crear y actualizar los servicios alimenticios	54
Figura 25 Mockup para la interfaz de la lista de asistencia y control de usuarios del software	54
Figura 26 Mockup para la interfaz de las gráficas (histórico) de consumo del software	55

* Trabajo de Grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática.

Director: Emilio Justiniano Cárcamo Troconis.

Figura 27 Índice de componentes y herramientas utilizados en la creación del backend	56
Figura 28 Funcionamiento del mapeo que realizan los DTO.....	58
Figura 29 Índice de los modelos de información del proyecto	59
Figura 30 Índice de servicios utilizados en el backend.....	59
Figura 31 Funcionamiento de los controladores dentro del patrón de arquitectura MVC	60
Figura 32 Token JWT concatenando Header, Payload y Signature.....	61
Figura 33 Proceso de autenticación de JWTbackend.....	62
Figura 34 Índice de componentes y herramientas utilizados en la creación del backend	62
Figura 35 Modulos y componentes de herramientas de PrimeNG.....	64
Figura 36 Creación de los componentes del software.....	64
Figura 37 Estructura de un componente en angular	65
Figura 38 Índice de componentes para el desarrollo de la interfaz del inicio de sesión	66
Figura 39 Prototipo del inicio de sesión del software	67
Figura 40 Barra de navegación del software de gestión de planes alimenticios	68

* Trabajo de Grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática.

Director: Emilio Justiniano Cárcamo Troconis.

Figura 41 Prototipo de la interfaz de inserción de usuarios del software 69

Figura 42 Prototipo de la interfaz del listado de usuarios y actualización de los mismos
..... 70

Figura 41 Prototipo de la interfaz de inserción de usuarios del software 70

Figura 43 Prototipo de la interfaz del listado de servicios alimenticios..... 71

Figura 44 Prototipo de la interfaz del listado de usuarios y actualización de estos 71

Figura 45 Prototipo de la interfaz de la lista de asistencia y control de usuarios del
software..... 72

Figura 46 Prototipo de la interfaz de las gráficas (histórico) de asistencia del software
..... 73

Figura 47 Configuración de prueba de estrés en Jmeter 76

Figura 48 Configuración del entorno para petición GET en Jmeter 77

Figura 49 Árbol de resultados con datos de respuesta correcta 77

Figura 50 Reporte resumen de la prueba de estrés realizada 78

Figura 51 Gráfico de de número de transacciones por tiempo de la prueba de estrés
realizada 78

Figura 52 Configuración del entorno para prueba de estabilidad en Jmeter 79

Figura 53 Reporte resumen de la prueba de estabilidad realizada 80

* Trabajo de Grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática.

Director: Emilio Justiniano Cárcamo Troconis.

Figura 54 Gráfico de número de transacciones por tiempo de la prueba de estabilidad realizada en Jmeter..... 81

Figura 55 Gráfica de número de peticiones activas por tiempo para prueba de estrés en Jmeter 92

Figura 56 Gráfica de número de peticiones activas por tiempo para prueba de estabilidad en Jmeter 93

* Trabajo de Grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática.

Director: Emilio Justiniano Cárcamo Troconis.

Lista de tablas

Tabla 1. Programación de sprints.....	39
Tabla 2. Programación de actividades sprint 1.	39
Tabla 3. Programación de actividades sprint 2.	40
Tabla 4. Programación de actividades sprint 3.	41
Tabla 5. Programación de actividades sprint 4.	42

Glosario

Programa de alimentación: Consisten en herramientas, planes o campañas especializadas para promover hábitos alimenticios saludables que apoyen el bienestar y el rendimiento de los usuarios inscritos al plan, pertenecientes a una entidad del sector público o privado.

Base de datos: Sistema que permite almacenar, recuperar y administrar datos, además de ser muy utilizada en el desarrollo de aplicaciones para gestionar todo tipo de información eficientemente.

Marco ágil: Metodología utilizada en la gestión de proyectos de software que se enfoca en la adaptabilidad, la colaboración, y la entrega de funcionalidades por ciclos de trabajo iterativos.

Ciclo de vida del software: Enfoque sistemático para la gestión del desarrollo y evolución del software, que consta de varias capas las cuales gestionan que el producto escale hasta ser funcional, eficiente y cumpla con los objetivos propuestos.

Arquitecturas de desarrollo: Estas arquitecturas determinan cómo los componentes del software se relacionan y se comunican entre sí. La elección de una arquitectura adecuada impacta en la mantenibilidad, escalabilidad y rendimiento del sistema.

* Trabajo de Grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática.

Director: Emilio Justiniano Cárcamo Troconis.

Fases de desarrollo web: Se compone de varias fases técnicas las cuales son esenciales para estructurar un desarrollo web adecuado que cumpla los estándares, y sea de alta calidad.

Wireframe: Son definidos como bosquejos iniciales del diseño web, ya definen la estructura de los componentes clave del sitio, además de definir el espacio de cada elemento, y conocer las limitaciones básicas de diseño.

Mockup: Su realización es necesaria para corregir errores de usabilidad, además de representar un modelo más fidedigno de lo que serán las interfaces a futuro teniendo en cuenta todos los elementos visuales del modelo a escala

Single Page Application(SPA): Es una aplicación web que funciona como una sola página en lugar de varias páginas individuales lo que facilita la experiencia de los usuarios al cargar solo ciertos componentes de la misma.

Autenticación: Proteger la integridad y la privacidad de los datos de los usuarios, previniendo accesos no autorizados y garantizando disponibilidad de la información.

Middleware: Es una herramienta de software útil para autenticación, comunicación de datos entre componentes y aplicaciones, gestión de datos y servicios en un software.

Controladores: Realizan la comunicación dentro del patrón de arquitectura MVC, ya que están conectados directamente con la vista, el modelo de datos, y el usuario.

* Trabajo de Grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática.

Director: Emilio Justiniano Cárcamo Troconis.

Servicios: Componentes que cumplen funciones de autorización, enviar o recibir registros, y colaborar con bases de datos, respecto a la necesidad que presenta el usuario.

Data transfer objects (DTO): Permiten la ágil y estructurada transferencia de datos, asegurando una comunicación sin problemas y un rendimiento óptimo entre diversos componentes de una aplicación.

Modelos de trabajo: Representación e interacción de la información de cada una de las tablas que contiene la base de datos.

Json Web Token (JWT): Herramienta utilizada en la gestión de procesos de seguridad para autenticar, autorizar, compartir e intercambiar información de usuarios en el desarrollo de aplicaciones.

Sprint: Fase del marco de desarrollo ágil que consiste en un periodo de tiempo en el cual se estima un tiempo determinado para realizar el desarrollo e integración de un componente del software que se esté realizando.

* Trabajo de Grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática.

Director: Emilio Justiniano Cárcamo Troconis.

Resumen

Título: Desarrollo de un prototipo de software para la gestión y monitoreo de programas de alimentación en entidades del sector público y privado.

Autores: Andres Felipe Barajas Wellman, Juan Pablo Jiménez Gómez**

Palabras Clave: Planes de alimentación, marco ágil, Backend, Frontend, ciclo de vida del software, arquitecturas de desarrollo, autenticación, fases de desarrollo web.

Descripción: En la industria alimentaria debido al aumento en la competencia que existe entre las empresas y entidades que hacen parte de esta, dentro de los últimos años se han dado a la tarea de automatizar para una mejor gestión de procesos, los cuales aún tienen problemas para ser implementados de forma adecuada.

El presente trabajo pretende diseñar y desarrollar un prototipo de software encargado de la gestión de programas de alimentación que cuente con la capacidad de autenticar, insertar y monitorear la gestión de consumo de los usuarios que se encuentran agregados al sistema.

El producto final mostrará el resultado de todas las metodologías implementadas, es decir un prototipo de software funcional que ha cumplido con todos los objetivos propuestos haciendo uso de las tecnologías de la información.

* Trabajo de Grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática.

Director: Emilio Justiniano Cárcamo Troconis.

Abstract

Title: Development of a software prototype for the management and monitoring of feeding programs in public and private sector entities.

Authors: Andres Felipe Barajas Wellman, Juan Pablo Jiménez Gómez**

Keywords: Food plans, agile framework, Backend, Frontend, software life cycle, development architectures, authentication, web development pase.

Description: In the food industry due to the increase in competition between companies and entities that are part of this, in recent years have been given the task of automating for better process management, which still have problems to be implemented properly.

This work aims to design and develop a software prototype for the management of food programs that has the ability to authenticate, insert and monitor the consumption management of users who are added to the system.

The final product will show the result of all the implemented methodologies, that to say, a functional software prototype that has fulfilled all the proposed objectives using information technologies.

Introducción

Con base en las implementaciones que se han llevado a cabo en programas de alimentación de los últimos años, diversas empresas y entidades pertenecientes a la industria alimentaria han adaptado los procesos de gestión de dichos programas. El objetivo de estas adaptaciones es asegurar que los beneficiarios, tanto del sector público como del sector privado, puedan recibir los alimentos de manera efectiva. Sin embargo, debido a la falta de planificación, estos programas no se han establecido correctamente. Por esta razón, se propone el desarrollo de un prototipo de software que automatice el proceso de gestión y control de los usuarios que forman parte del programa alimentario. De esta forma, los usuarios inscritos recibirán los alimentos en la periodicidad acordada.

Siguiendo esta línea de pensamiento, se obtuvieron varios resultados. En primer lugar, se asegura que el historial no puede ser adulterado, lo que garantiza la integridad de la información. Además, se mantiene la disponibilidad de información sobre los datos fundamentales para el seguimiento de los usuarios. Por último, se cumplen los estándares de la norma ISO 9000:2015, que establece "la capacidad de rastrear la historia, la aplicación o la ubicación de lo que está bajo consideración" (ISO, 2015).

Durante el desarrollo del proyecto se implementó cada uno de los pasos presentes en el ciclo de vida del software, lo que resultó en una planeación y prototipado adecuado. Además, se llevaron a cabo pruebas de concepto para elegir las tecnologías que permitieron mejorar el software, lo que por consiguiente lleva al uso de tecnologías enfocadas en la

gestión de tareas como lo es Scrum, que distribuye cada una de las actividades en un periodo base de tiempo dependiendo de su complejidad, asimismo, se aplicaron los conocimientos adquiridos durante la carrera para diseñar productos basados en software.

1 Justificación o planteamiento del problema

¿Cómo se puede mejorar la planificación y monitoreo de los programas de alimentación dirigidos a usuarios pertenecientes a diversas entidades (colegios, universidades, empresas, hospitales) mediante el uso de las tecnologías de la información?

En el contexto nacional, tanto entidades públicas como privadas han buscado diseñar programas de alimentación para que los usuarios pertenecientes a una institución puedan acceder a estos servicios de forma regular, de manera similar a una tiquetera. Sin embargo, la gestión de estos programas a través de sistemas de información aún no se ha generalizado debido a fallas en la planificación, ineficiencias en la administración de los procesos alimenticios y la falta de empoderamiento de las entidades involucradas en este proceso. Por lo tanto, es necesario implementar estrategias de transformación digital.

Se han explorado diferentes enfoques para garantizar que los usuarios que forman parte de un programa de alimentación reciben los alimentos de manera regular. La solución más adecuada para llevar a cabo este proceso implica trabajar con la disponibilidad de la información de los usuarios. Como menciona **del Mar Araluce Letamendia (2001)**, es necesario establecer mecanismos que permitan retroalimentar los procesos y desarrollar herramientas para recopilar información de los clientes. Siguiendo este enfoque, se puede asegurar un control adecuado del proceso.

Entre los desafíos planteados para desarrollar un sistema de seguimiento que automatice la gestión, la autenticación y salvaguardia de la información del usuario, es necesario administrar los servicios que se ofrecerán, como desayuno, almuerzo y cena. También se debe tener la capacidad de registrar en el sistema a todos los usuarios inscritos en el programa de alimentación, junto con la institución a la que pertenecen, con el propósito de monitorear su asistencia y verificar que efectivamente hayan recibido los alimentos.

Para garantizar la seguridad de la información y el correcto funcionamiento de los programas de alimentación, se utilizaron una amplia gama de herramientas y tecnologías, conocidas como sistemas de información. “Estos sistemas realizan cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información, con el fin de apoyar las actividades de una empresa o negocio” (**Figueredo Ramos, 2015**).

Este proyecto propuso un flujo de investigación que identifica las necesidades que impiden que el proceso de gestión se realice de manera adecuada. Como resultado, se desarrolló un software para la gestión idónea de los programas de alimentación, siguiendo las etapas del ciclo de vida del software y aplicando técnicas de ideación, prototipado y diseño de productos basados en software desde una perspectiva de ingeniería. Se pusieron en práctica los conocimientos adquiridos durante la carrera y se utilizaron conceptos y metodologías propias de una fábrica de software, como Scrum.

2 Objetivos

2.1 Objetivo general

Desarrollar un prototipo de herramienta de software que permita gestionar y monitorear programas de alimentación en empresas de la industria alimentaria.

2.2 Objetivos específicos

- 1) Definir los requerimientos para la gestión de los comedores que implementan las empresas operadoras dentro de la industria alimentaria.
- 2) Desarrollar el prototipo software backend encargado de la gestión de servicios para el manejo ordenado de datos de los usuarios beneficiarios del software.
- 3) Desarrollar un prototipo software frontend que cuente con interfaces amigables al usuario y acopladas al prototipo backend.
- 4) Realizar un plan de pruebas que verifique el apropiado funcionamiento del prototipo de software.

3 Marco de referencia

3.1 Marco teórico

La automatización de planes alimenticios es un enfoque innovador que busca mejorar la eficiencia, y precisión en la gestión de los mismos. Al establecer un sólido marco de referencia para este proyecto de investigación, se exploraron diversas tecnologías y procesos clave que desempeñaron un papel fundamental en el diseño, desarrollo e implementación para la creación de un sistema de automatización robusto siguiendo cada una de las etapas del ciclo de vida del software.

El marco de referencia de este proyecto de tesis se basa en la implementación de tecnologías avanzadas que permitirán una gestión eficiente y efectiva. En primer lugar, se utilizó una infraestructura como código (IaC), una metodología que permitió definir y gestionar la infraestructura de manera programática. Esto garantizó la flexibilidad y escalabilidad necesarias para adaptarse a las necesidades cambiantes del proyecto. Adicionalmente se emplearon tecnologías como Angular y Node.js para construir una interfaz web interactiva y dinámica, que permitió la creación del inicio de sesión para los usuarios administradores, así como las vistas para la inserción de los usuarios que harán parte del plan alimenticio, la vista que gestiona que los usuarios reciban el alimento en la periodicidad acordada, y la vista que muestra las gráficas a manera de histórico de consumo. De tal manera que los administradores gestionen los planes alimenticios de manera sencilla y eficiente.

Por otro lado, el control de versiones, especialmente utilizando Git, permitió mantener un historial completo de los cambios realizados en el código y la configuración, lo que facilitó la colaboración y la reversión de modificaciones. Estas tecnologías desempeñaron un papel crucial en la automatización y optimización de los procesos relacionados con los planes alimenticios. De esta forma se realizó la integración entre el control de versiones y la gestión de la configuración que permitió mantener la consistencia y la estabilidad de la configuración del entorno, además de asegurar que se mantenga un control riguroso sobre los cambios realizados en el sistema, lo que facilitó su mantenimiento y evolución a lo largo del tiempo.

En cuanto a la arquitectura de base de datos, se utilizó un enfoque escalable y eficiente para almacenar la información necesaria para los planes alimenticios. Se utilizó MySQL aplicando prácticas de modelado de datos que permitió almacenar y acceder a la información nutricional, los perfiles de los usuarios y otros datos relevantes para los planes alimenticios. La arquitectura de base de datos desempeñó un papel fundamental en la gestión y organización de la información relacionada con los planes alimenticios, permitiendo un acceso rápido y confiable a los datos necesarios para su generación.

El enfoque de desarrollo ágil fue fundamental en la implementación del proyecto de automatización de planes alimenticios. El desarrollo ágil se basa en la entrega iterativa e incremental de software funcional, lo que promueve la entrega continua de funcionalidades y la colaboración efectiva entre los miembros del equipo, además de responder de manera ágil a los cambios y requisitos del proyecto. Haciendo uso de la metodología ágil Scrum. La adopción de un enfoque ágil en el desarrollo del proyecto aseguró una mayor flexibilidad y capacidad de respuesta a medida que se descubran nuevas necesidades y se realicen ajustes durante el proceso.

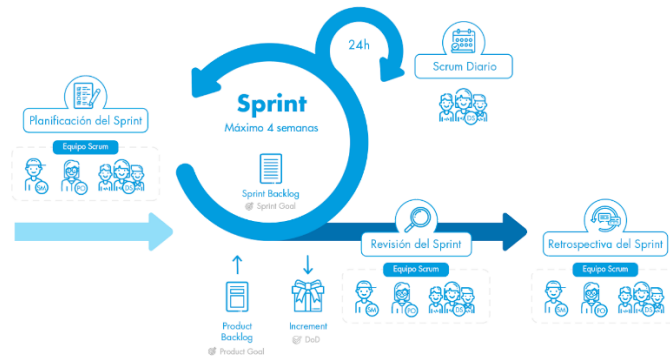


Figura 1. Funcionamiento del marco de trabajo ágil Scrum.

Fuente: <https://netmind.net/es/scrum-el-pasado-y-el-futuro/>

Para la seguridad de los datos y disponibilidad de la información se utilizó JSON Web Token mejor conocido como JWT es una herramienta muy utilizada en cuanto a desarrollo de aplicaciones se refiere, debido a su amplio uso en procesos de seguridad ya que JWT permite autenticar, autorizar, compartir e intercambiar información de usuarios en aplicaciones.

La estructura de JWT consta de tres elementos, el primero es el Header o encabezado el cual posee un algoritmo identificador y el token de tipo “JWT” que funciona como identificador, seguido a esto el segundo elemento es el Payload o carga útil el cual contiene las “claims” consideradas como declaraciones sobre una entidad como el usuario, afirmando que ciertos valores específicos de este son los que se autentican, y son únicos para cada usuario y sesión. Las claims siempre se definen por un nombre el cual siempre es una cadena, y un valor que es definido como un valor JSON.

El tercer elemento es “Signature” o firma la cual contiene un algoritmo “hash” igual al del “header” y que se conforma de recopilatorios de este último y de los “payload” del JWT. La firma es utilizada para generar la validación del JWT de tal manera que se le proporcione al usuario el acceso a recursos que deberían encontrarse protegidos dentro del software que se esté desarrollando, sea un ejemplo muy común el inicio de sesión de usuarios autorizados.

Estas tecnologías y procesos se trabajaron en conjunto para proporcionar los cimientos necesarios para el desarrollo exitoso de nuestro proyecto, aprovechando al máximo las capacidades de estas tecnologías y garantizando un sistema eficiente y de calidad.

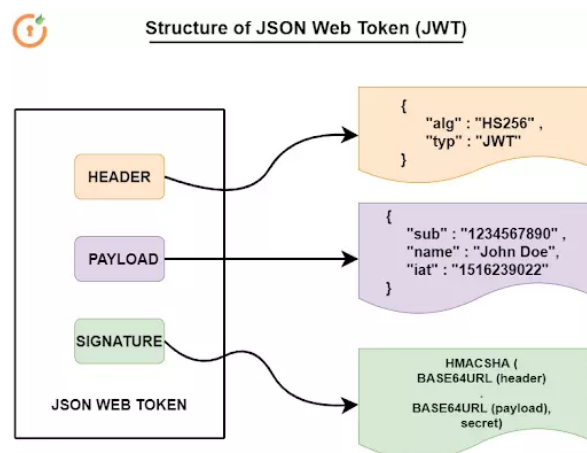


Figura 2. Estructura de manejo de JWT

Fuente: <https://blog.miniorange.com/what-is-jwt-json-web-token-how-does-jwt-authentication-work/>

3.2 Estado del arte

Actualmente gracias al constante avance tecnológico, y el desarrollo de la infraestructura, la automatización de procesos tecnológicos en la industria ha desplegado un sin número de herramientas y aplicaciones a diferentes áreas de trabajo como lo son la banca, la ciencia, la salud, el internet de las cosas (IOT), entre otros. Esto ha permitido que muchas empresas se acojan a estos nuevos sistemas, manejando no solo el tema de control de acceso y trazabilidad en las organizaciones, sino también ofreciendo nuevos servicios que faciliten el acceso virtual o físico del usuario, al igual que los procesos comerciales, pagos, solicitudes, y consultas.

Sin embargo, cabe resaltar el hecho de la constante evolución de las tecnologías de la información (TI), debido que en sus inicios no se empleaban de la misma forma los sistemas de automatización, para mecanizar procesos. “La revolución tecnológica actual se debe a los avances significativos en las TIC's. Los grandes cambios que caracterizan esencialmente esta nueva sociedad son: la generalización del uso de las tecnologías, las redes de comunicación, el rápido desenvolvimiento tecnológico y científico y la globalización de la información.” (*Tema 1.1 Evolución de las Tecnologías de Información - Instituto Consorcio Clavijero, s/f*). Fue de esta forma que se implementan estas tecnologías en diferentes áreas de trabajo como la banca, la medicina, la ciencia, el sector industrial, sector del comercio, industria alimentaria, etc. dejando de un lado procedimientos primitivos por sistemas más modernos y útiles.

Varios sectores del área alimenticia se han automatizado varios de sus servicios como por ejemplo la facturación, y la toma de pedidos. A partir de esto se han implementado herramientas que favorecen tanto al usuario como al empleado por medio de dispositivos y aplicaciones móviles. Una de estas herramientas es la compra de tiqueteras virtuales para la adquisición de un plan alimenticio. Algunos restaurantes usan el método contactless o pago móvil desde que inició la pandemia del COVID, el caso de KFC en los restaurantes de China “Ha sido en la ciudad china de Hangzhou donde Alipay ha estrenado su servicio de pago “Smile to Pay” con el que sólo necesitamos nuestro rostro y número de teléfono para pagar.” (Elpais, [JORGE SANZ FERNÁNDEZ](#), 2017).

Partiendo de la prestación de nuevos servicios que se menciona anteriormente, y haciendo énfasis en el proyecto en el que se está trabajando, el objeto de estudio son los programas de alimentación, ya que la automatización de estos da lugar al uso de un sistema de gestión que pueda tomar las operaciones principales que se llevan a cabo. Es de esta forma que se han venido automatizado varios de estos servicios, como por ejemplo el mantener el control del consumo de alimentos que estos llevan. A partir de esto se han implementado herramientas que favorecen tanto al usuario como al empleado por medio de dispositivos y aplicaciones.

Respecto a la idea de cómo asegurar la gestión y trazabilidad del cliente que va a consumir los alimentos se menciona el primer caso centrado en el área escolar, donde se ha automatizado el proceso de selección de candidatos beneficiarios de los programas de alimentación. Ese caso es el aplicado por el Ministerio de Educación Nacional Colombiano,

denominado como “PAE” (**ver figura 3**) el cual es un programa que se encarga no solo de cubrir las necesidades alimenticias de los estudiantes beneficiarios pertenecientes a un colegio oficial del sector que ha sido elegido como vulnerable, sino también de llevar un control y cumplimiento de dicho proceso por medio del registro de los estudiantes beneficiarios a la plataforma de matrícula (SIMAT). No obstante, debido a problemas de organización y repartición adecuada de los recursos que han sido destinados al programa. Se vulnera la seguridad del usuario, la disponibilidad de la información, y el hecho de que no se pueda lograr una trazabilidad respecto a los procesos de evolución del producto por etapas. Claro está que este proceso se ha ido implementado y así mismo automatizando de igual forma en varios países, sin embargo, cabe resaltar que en algunos funciona mejor que otros debido a ciertos parámetros implementados.

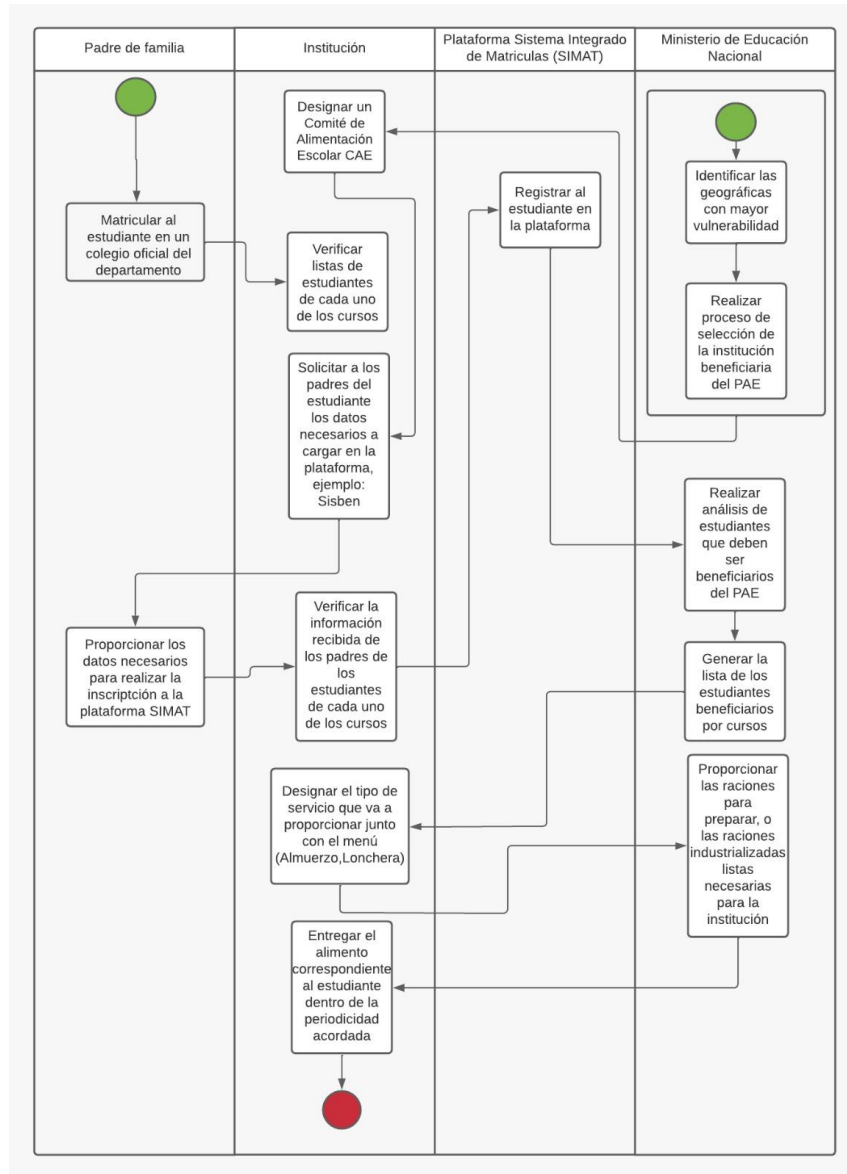


Figura 3. Diagrama de gestión de procesos de los programas de alimentación escolar en Colombia

Fuente: Autor

Un segundo caso de éxito relacionado al programa de alimentación escolar sería el que se está implementando en el país de Guatemala en el presente 2023, donde el ministerio de Educación (Minieduc) en relación con el Programa Mundial de Alimentos (PMA), presentaron una aplicación móvil piloto de un sistema automatizado para el programa de alimentación escolar “A través de esta plataforma digital se nos permitirá monitorear los avances, proveedores y entrega de la alimentación escolar a nivel nacional, dijo la ministra de Educación, **Claudia Ruíz**” (AGN - **Brenda Larios, 28 de julio del 2022**). Gracias a que el proyecto impulsa la realización de un adecuado control del programa del cual hace parte, la transparencia, y el hecho de que han automatizado totalmente sus servicios, “La aplicación digital que se presentó incluirá la información del [Programa de Alimentación Escolar](#), como el total de beneficiarios, los centros educativos dónde se entregan los alimentos, detalles de las compras, los proveedores y su ejecución en el país.”(AGN - **Brenda Larios, 28 de julio del 2022**).

Basado en lo anterior, el proyecto dentro de su alcance aborda algunos de los temas ya mencionados además de resaltar el reglamento de trazabilidad que asiste como medida de supervisión a los programas de alimentación que prestan las organizaciones con respecto a los servicios de alimentación de los beneficiarios, garantizando una buena auditabilidad de sus etapas, el control y monitoreo del sistema. Como caso de uso dentro del contexto educativo, se hace énfasis en la gestión de asistencia del usuario por medio de una “checklist” ya sea que se encuentre afiliado a un comedor estudiantil, que pague una cuponera periódicamente, o porque está incluido en su matrícula, asegurando que reciba el

alimento de manera regular. Y en el contexto empresarial, del lado del administrador que gestiona el software, se puede tener tanto el seguimiento como el control del registro de todos los usuarios que pertenecen al plan, y gráficas que le permitan establecer tendencias para realizar mejoras.

4 Fase de planeación

4.1 Marco metodológico

Para el desarrollo del Proyecto se utilizó como hoja de ruta el marco de trabajo ágil Scrum como método de gestión de proyectos, de tal manera que el equipo de trabajo se organizó siguiendo un proceso de construcción de fases “Sprint” en el cual se definieron las tareas que se iban a desarrollar de manera prioritaria, evaluando su complejidad de los Backlog que los conforman para así poder establecer su periodo de tiempo, para esto se listó cada una de las tareas evaluándose por su necesidad, su complejidad y su tiempo creando así el sprint correspondiente. Las cuales se fueron desarrollando a lo largo del cronograma de actividades que fue planeado.

Cabe resaltar que todas las tareas que se realizaron en cada una de las fases “Sprint” del proyecto, se realizaron de tal manera que se cumpliera con cada una de las etapas del ciclo de vida del desarrollo del software, desde su planeación hasta su posterior entrega. Abarcando procesos de soporte, gestión de la configuración, y documentación del sistema. Manteniendo de esa forma un adecuado mantenimiento de calidad, verificación y posterior validación del software, revisión conjunta. Y dando como resultado una constante mejora en cuanto a la resolución de problemas se refiere, y a la gestión de la infraestructura y disponibilidad de la información.

Para la preparación y posterior desarrollo de cada una de estas tareas se utilizó Jira como gestor de tareas que brinda una amplia variedad de herramientas para simplificar la planificación y supervisión de proyectos y actividades, realizando un adecuado

seguimiento de las tareas desde que fueron asignadas hasta su posterior finalización y revisión; Así mismo supervisando el flujo de trabajo, los tiempos de realización, y una adecuada trazabilidad en lo que fueron el desarrollo de actividades.

La realización del proyecto se dividió en tres fases de desarrollo que abarcan cada una de las fases del ciclo de vida del software, esto se hizo con el propósito de ir realizando, evaluando y validando cada uno de los objetivos planteados dentro del alcance del proyecto.

Dentro de la siguiente fase, se realizó la planificación del alcance, y las restricciones del proyecto, al igual que la toma de decisiones sobre los recursos, herramientas e arquitectura que fueron necesarios para el desarrollo del mismo, la realización de una estimación para la definición y repartición de actividades y plazos asignados a cada desarrollador, además de llevar a cabo un análisis de riesgos y resolución de los problemas que podrían surgir durante el proyecto; dando como resultado el cumplimiento del objetivo específico 1 del proyecto (Revisar sección 3.2 - Objetivos específicos).



Figura 4 Revisión del marco metodológico

Fuente: Autor

4.1.1 Análisis de procesos de gestión de los programas alimentarios

La primera actividad que se realizó en la fase de planeación fue establecer qué funciones desarrollan los programas alimentarios, para eso como punto de partida se tomó como referencia para el análisis de procesos programas que son implementados en varios países (**vea el estado del arte**). El objetivo principal requería tener un entendimiento mucho más amplio de cómo relacionar el tratamiento de datos y los procesos de gestión que se manejan entre las empresas y las instituciones que se encuentran asociadas al programa. Por lo anterior, se diseñó un diagrama BPMN que explica el procedimiento que se hace paso a paso y la interacción que hay para cada uno de los diferentes autores del software en cada fase de gestión.

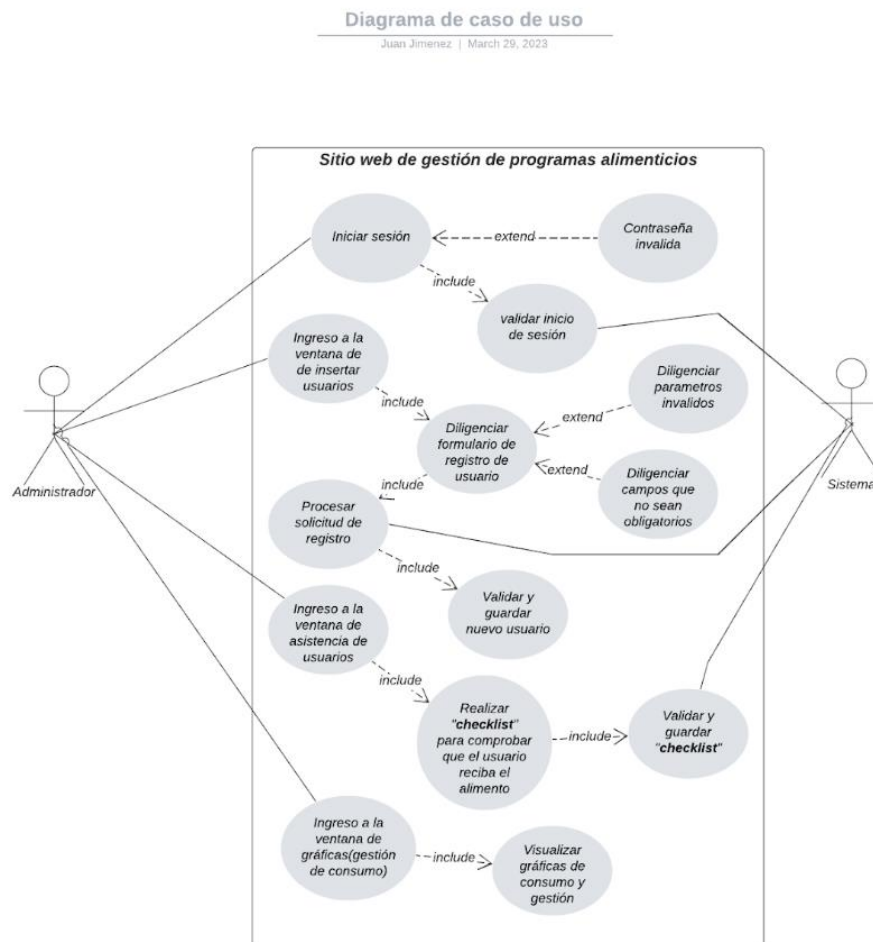


Figura 5 Diagrama de casos de uso para el prototipo de software enfocado a los programas de alimentación

Fuente: Autor

Luego de entender los protocolos, y funciones de gestión del programa de alimentación que va a manejar el software del proyecto, se establecieron varios análisis que conllevaron a un diagrama que contará con funcionalidades para el prototipo del software, para poder automatizar el tratamiento de datos de una forma más compacta de tal manera que

simplifique la dificultad a la hora de realizar estas operaciones, y de esta forma se asegure de esta forma la fiabilidad y disponibilidad de la información.

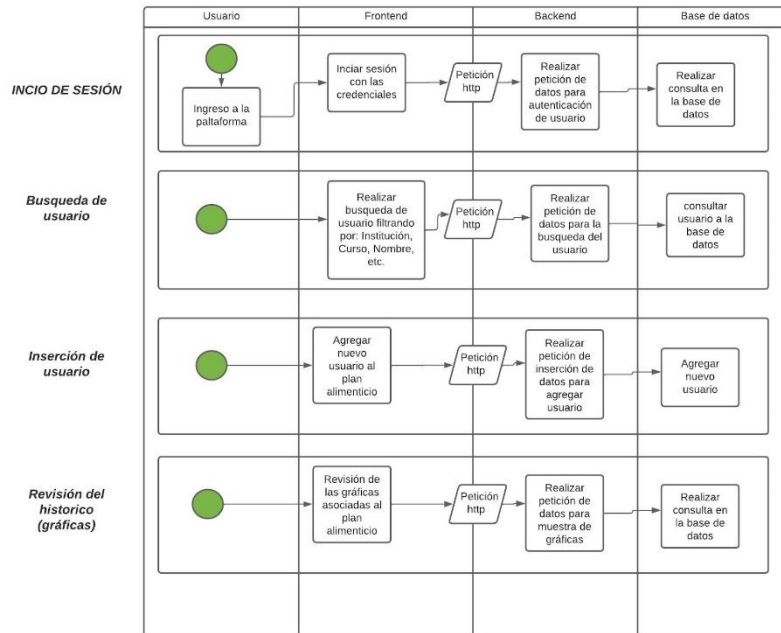


Figura 6 Diagrama de funcionalidades para el software con conexión entre tecnologías

Fuente: Autor

4.1.2 Modelado de datos para el desarrollo del software

Habiendo realizado un análisis más minucioso, y luego de haber recopilado los aspectos o características más importantes en cuanto al proceso de gestión de los programas alimentarios se refiere. Se planteó un modelo inicial para definir la arquitectura de desarrollo de software que asegure un adecuado flujo de información para los servicios que serían creados posteriormente luego de diseñar la base de datos.

La razón de optar principalmente por esta técnica viene precisamente de la idea de poder juntar todos los procesos de los programas de alimentación en un mismo centro y poder manejarlo de una manera más fluida y eficaz.

4.1.3 Diseño y desarrollo de la base de datos junto con los requerimientos del proyecto

Luego de elegir la arquitectura, lo siguiente fue el diseño y desarrollo de la base de datos del software. En este punto se tuvo que tener en cuenta el alcance en cuanto a las operaciones que iba a ejecutar el software, precisamente porque era necesario definir un modelo de datos que pudiera satisfacer todas las necesidades que se han ido planteando. A partir de esto se tomaron tres aspectos importantes para construir la base de datos del sistema. El primero de estos aspectos hace referencia a las instituciones ya que son el punto de partida de donde provienen los beneficiarios; razón por la cual dentro del modelo de datos se incluyen la gestión de instituciones vigentes geográficamente. El segundo aspecto hace referencia a los usuarios que hacen parte del programa alimentario y que obligatoriamente deben pertenecer a un curso para estar inscritos a estos beneficios. El último aspecto se relaciona en la forma en la que se va a automatizar el proceso para asegurar que el usuario beneficiario realmente haya recibido su alimento en la periodicidad acordada, y es de esta forma que en el modelo de datos se va a registrar el contrato entre la empresa o entidad que presta el servicio del programa de alimentación y la institución, donde esta última a partir del contrato maneja los servicios como el desayuno, almuerzo

o cena que le proporcionará al usuario, y que por medio de una “**checkList**” se verificará con fechas, la asistencia de los mismos.

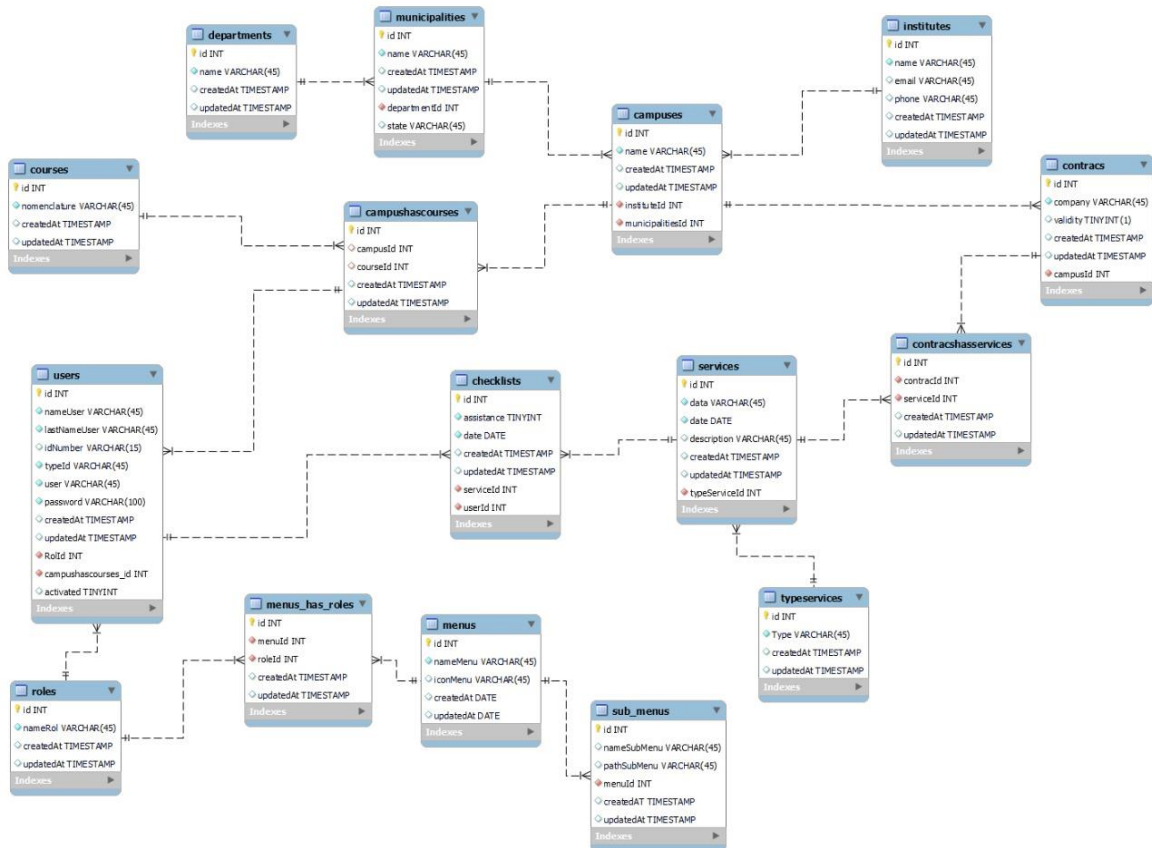


Figura 7 Base de datos relacional del nuevo programa de alimentación

Fuente: Autor

Luego de diseñar y construir la base de datos del proyecto, la siguiente actividad que se realizó fue la definición de los requerimientos del software para la gestión de los planes de alimentación que implementan las empresas operadoras dentro de la industria alimentaria.

Posterior a lo ya mencionado, y siguiendo el cronograma de planificación de las actividades a través del marco de trabajo ágil SCRUM, incluyendo la revisión de historias de Usuario, definición de cargas de trabajo y planificación de los sprints.

Tabla 1. Programación de sprints.

SPRINTS	
CANTIDAD	PERIODO DE DURACIÓN
4	4 SEMANAS

Tabla 2. Programación de actividades sprint 1.

Sprint 1		
NO.	Historia de usuario	Complejidad
1	Diseño y desarrollo de la base de datos	15
2	Creación del wireframe para inicio de sesión	2
3	Creación del wireframe para la barra de navegación	2
4	Creación del wireframe para la ventana de inserción de usuarios	2
5	Creación del wireframe para la ventana que permite actualizar información de los usuarios	2
6	Creación del wireframe para la ventana que muestra la lista de usuarios	2
7	Creación del wireframe para la ventana de generación de servicios alimenticios	2
8	Creación del wireframe para la ventana de actualización de servicios alimenticios	2
9	Creación del wireframe para la ventana de gestión del plan alimenticio o "checklist" del mismo	2
10	Creación del wireframe para la ventana de las gráficas del histórico	2
11	Conexión con la base de datos usando Sequelize	5
12	Configuración del proyecto en NodeJS	3
13	Creación e implementación de los DTO de los usuarios del sistema	6
14	Creación e implementación de los modelos de trabajo: Menu	7
15	Creación e implementación de los modelos de trabajo: MenuRol	7
16	Creación e implementación de los modelos de trabajo: Rol	7
17	Creación e implementación de los modelos de trabajo: Server	7

Sprint 1		
18	Creación e implementación de los modelos de trabajo: Submenu	7
19	Creación e implementación de los modelos de trabajo: User	7
TOTAL	20	89

Tabla 3. Programación de actividades sprint 2.

Sprint 2		
NO.	Historia de usuario	Complejidad
1	Creación e implementación de los servicios: MenuRol	8
2	Creación e implementación de los servicios: Rol	8
3	Creación e implementación de los servicios: User	8
4	Creación del Mockup para inicio de sesión	2
5	Creación del Mockup para la barra de navegación	2
6	Creación del Mockup para la ventana de inserción de usuarios	2
7	Creación del Mockup para la ventana que permite actualizar información de los usuarios	2
8	Creación del Mockup para la ventana que muestra la lista de usuarios	2
9	Creación del Mockup para la ventana de generación de servicios alimenticios	2
10	Creación del Mockup para la ventana de actualización de servicios alimenticios	2
11	Creación del Mockup para la ventana de gestión del plan alimenticio o "checklist" del mismo	2
12	Creación del Mockup para la ventana de las gráficas del histórico	2
13	Creación e implementación de los controladores: Autenticación	6
14	Creación e implementación de los controladores: MenuRol	6
15	Creación e implementación de los controladores: Rol	6
16	Creación e implementación de los controladores: User	6
TOTAL	16	50

Tabla 4. Programación de actividades sprint 3.

Sprint 3		
NO.	Historia de usuario	Complejidad
1	Configuración del proyecto en Angular	3
2	Creación e implementación de las rutas: Autenticación	9
3	Creación e implementación de las rutas: MenuRol	9
4	Creación e implementación de las rutas: Rol	9
5	Creación e implementación de las rutas: controladores	9
6	Desarrollo e implementación del prototipo para la interfaz del inicio de sesión del software	13
7	Desarrollo e implementación del prototipo de la barra de navegación para el software	6
8	Desarrollo e implementación del prototipo para la interfaz de inserción de usuarios del software	8
9	Desarrollo e implementación del prototipo para la interfaz de que permite actualizar información de los usuarios	8
10	Desarrollo e implementación del prototipo para la interfaz de que muestra la lista de usuarios	7
11	Desarrollo e implementación del prototipo para la interfaz de generación de servicios alimenticios	6
12	Desarrollo e implementación del prototipo para la interfaz de actualización de servicios alimenticios	6
13	Desarrollo e implementación del prototipo para la interfaz de gestión de planes alimenticios del software	12
14	Desarrollo e implementación del prototipo para la interfaz de las gráficas del histórico del software	13
TOTAL	14	118

Tabla 5. Programación de actividades sprint 4.

Sprint 4		
NO.	Historia de usuario	Complejidad
1	Pruebas unitarias de desarrollador para evaluar la funcionalidad del software	6
2	Pruebas unitarias en firebase para evaluación de código	8
3	Pruebas unitarias en jmeter para evaluar la funcionalidad del software	9
4	Arreglos y ajustes finales al frontend y backend	4
5	Documentación del código	2
TOTAL	5	29

4.2 Definición de tecnologías

Dentro de la definición de tecnologías se estableció como iba a ir estructurada e integrada la aplicación. Es por esto que se creó un diagrama taxonómico que estableciera la relación que iba a tener en la base de datos con el backend, la interacción que es ultimo iba a tener con el frontend, y el cómo estos iban a ir contruidos de la mano de un sistema de control de versiones como lo es Github en conjunto con herramientas de gestión CI/CD.

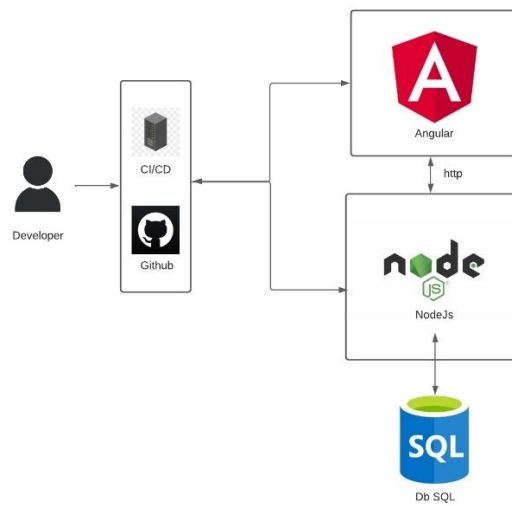


Figura 8 Diagrama de taxonomías

Fuente: Autor

4.2.1 Prueba de concepto: Angular

Para poder determinar qué tecnologías eran las más adecuadas a la hora de desarrollar el proyecto, se realizaron algunas pruebas de concepto. Partiendo de lo anterior, lo primero fue definir la tecnología más apropiada para el frontend, dentro de las opciones se comparó Angular, Vue y React. El framework elegido fue Angular, analizando en términos de flexibilidad, capacidad y herramientas además de señalar que Angular Además tiene la capacidad de ser Single Page Application (SPA) intercambiando vistas por medio de peticiones dentro de la misma página sin la necesidad de recargar la misma, haciendo que la aplicación sea más eficiente.

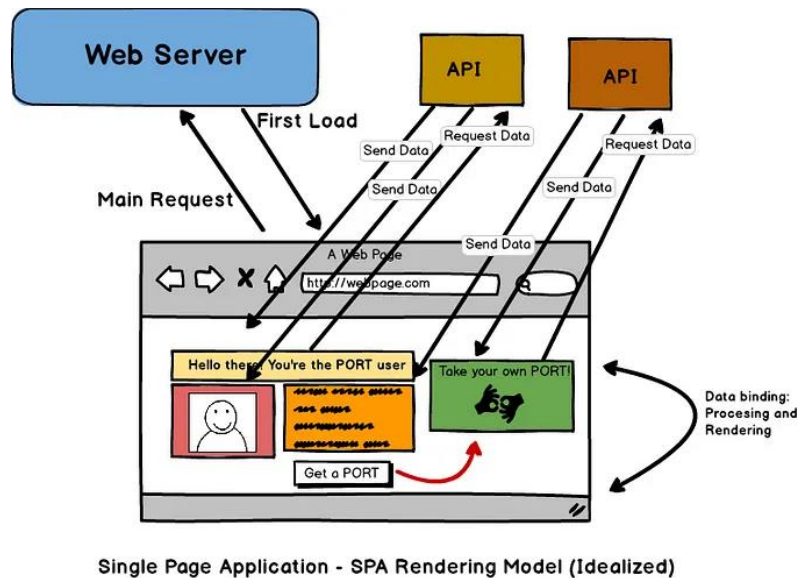


Figura 9 Single Page Application(SPA)

Fuente: [Single Page Application\(SPA\)](#)

Otras ventajas establecen que como la estructura de Angular está definida por el patrón de diseño Modelo-Vista-Controlador(MVC), facilita definir y conservar el software continuamente seguro, escalable y organizado. También es importante recalcar que maneja un enlace bidireccional que usa el framework para sincronizar componentes y datos pertenecientes a la misma aplicación.

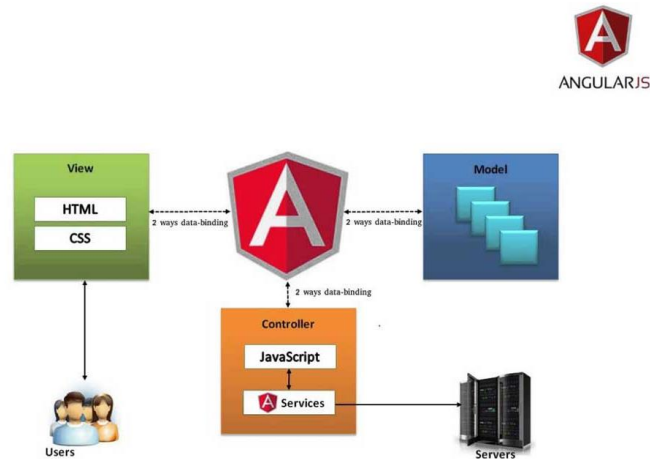


Figura 10 Angular Modelo Vista Controlador (MVC)

Fuente: [Angular Modelo Vista Controlador \(MVC\)](#)

4.2.2 Prueba de concepto: Node.js

Para la tecnología que se usaría en el Backend se eligió Node.js como entorno de ejecución para el uso de JavaScript como backend, debido a que dentro de la prueba de concepto se buscaba mantener una sinergia adecuada entre la tecnología del frontend y la del backend, en este caso Angular puede trabajar del lado del cliente y Node.js puede hacerlo del lado del servidor, lo cual es una ventaja considerable a la hora de realizar la integración de ambos entornos ya que se vuelve una actividad más fluida, evitando errores de compatibilidad, y generando una mejora progresiva en el rendimiento y el tiempo de ejecución al realizarse varias solicitudes de los usuarios al servidor.

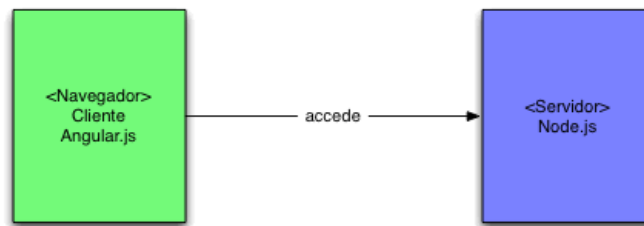


Figura 11 Conexión entre Angular y Node.js

Fuente: [Conexión entre Angular y Node.js](#)

En relación de lo ya mencionado, el uso de Node.js facilita el desarrollo del software de la aplicación por varias razones, la primera es la amplia serie de paquetes y módulos los cuales se pueden encontrar en la plataforma Node Package Manager (NPM) que dispone de varias soluciones de software a la hora de necesitar trabajar con diferentes componentes, bases de datos o realizar peticiones GET,POST,PUT, etc. Lo que hace que su ecosistema sea muy diverso, además cabe resaltar que Node.js cuenta con la capacidad de conectarse a otros servicios facilitando por ejemplo la fase de pruebas unitarias por medio de métodos de consumo de datos.

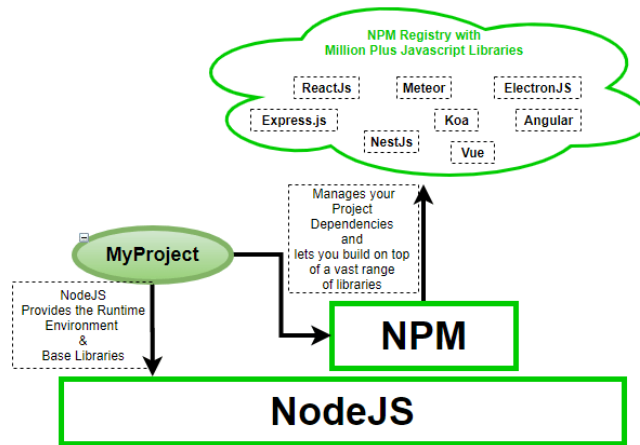


Figura 12 Ecosistema de Node.js

Fuente: [Ecosistema de Node.js](#)

4.3 Proceso de diseño web para el Frontend del sistema

4.3.1 Diseño de wireframes

Los wireframes son la primera fase del diseño web encargada de la construcción de los bosquejos o esqueletos iniciales del diseño web, ya que permitieron definir la estructura de los componentes clave del sitio, además de definir el espacio de cada elemento, y conocer las limitaciones básicas de diseño. También fueron necesarios para organizar el orden de cada una de las vistas optimizando el trabajo y simplificando errores de usabilidad para el usuario a futuro, ya que el diseño del wireframe funciona como guía visual a la hora de pasar a la siguiente fase que sería maquetar pantallas, y posterior desarrollo de las mismas. A Continuación, se mostrará cada uno de los wireframes diseñados, siguiendo el orden del manejo de procesos y jerarquías del software.

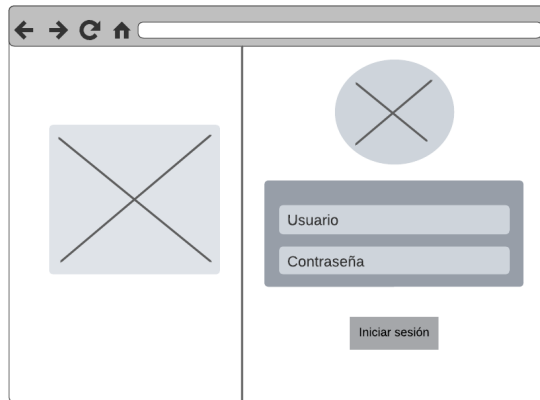


Figura 13 Wireframe para el inicio de sesión del software

Fuente: Autor

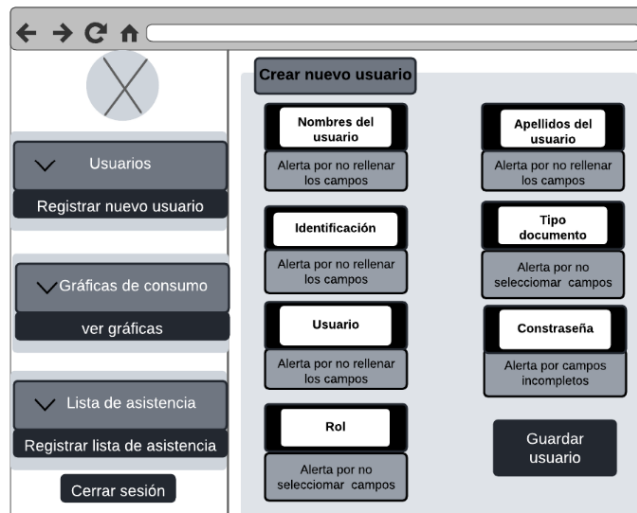


Figura 14 Wireframe para la interfaz de inserción de usuarios del software

Fuente: Autor

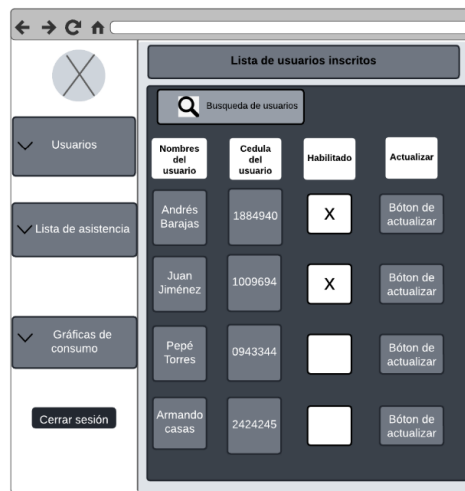


Figura 15 Wireframe para la interfaz que muestra la lista de usuarios y permite actualizar información de estos

Fuente: Autor



Figura 16 Wireframe para la interfaz de muestra la lista de servicios alimenticios

Fuente: Autor



Figura 17 wireframe para la interfaz de permite crear y actualizar los servicios alimenticios

Fuente: Autor

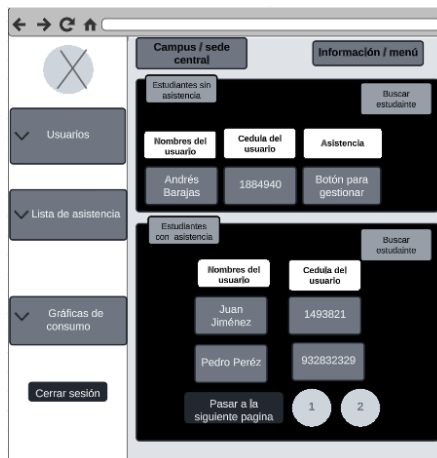


Figura 18 Wireframe para la interfaz de la lista de asistencia y control de usuarios del software

Fuente: Autor

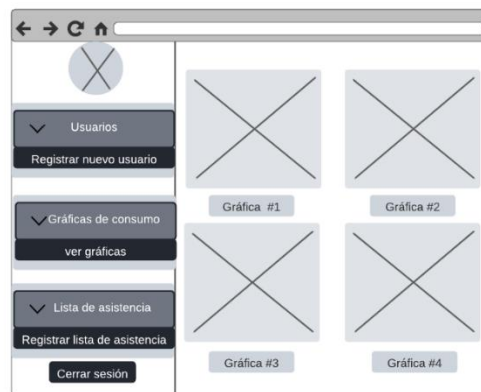


Figura 19 Wireframe para la interfaz de las gráficas (histórico) de consumo del software

Fuente: Autor

4.3.2 Diseño de mockups

Los mockups son la segunda fase del diseño web siendo estos una representación más fidedigna de las interfaces, debido a que su uso mostró el comportamiento y la estructura visual que llevará el sitio web. En esta fase se corrigieron errores de usabilidad, además de añadir todos los elementos visuales del modelo a escala que se habían elegido para cada una de las interfaces: colores, plantillas, iconografía, imágenes, textos, etc. Cabe resaltar sí es cierto que se intenta ser lo más apegado posible a la maqueta de diseños, para el prototipo funcional pueden realizarse algunos arreglos para mejorar la estructura visual del sitio web. Al igual que con la presentación visual de los wireframes (**ver sección 5.3.1**), también se mostrará cada uno de los mockups diseñados, siguiendo el orden del manejo de procesos y jerarquías del software.

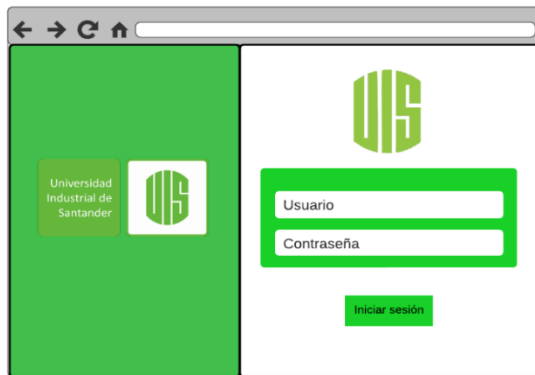


Figura 20 Mockup para el inicio de sesión del software

Fuente: Autor

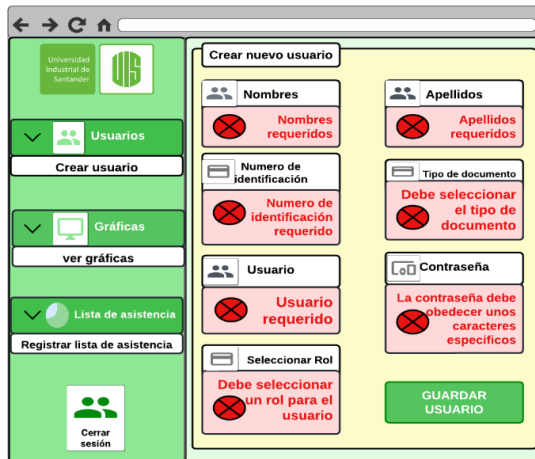


Figura 21 Mockup para la interfaz de inserción de usuarios del software

Fuente: Autor

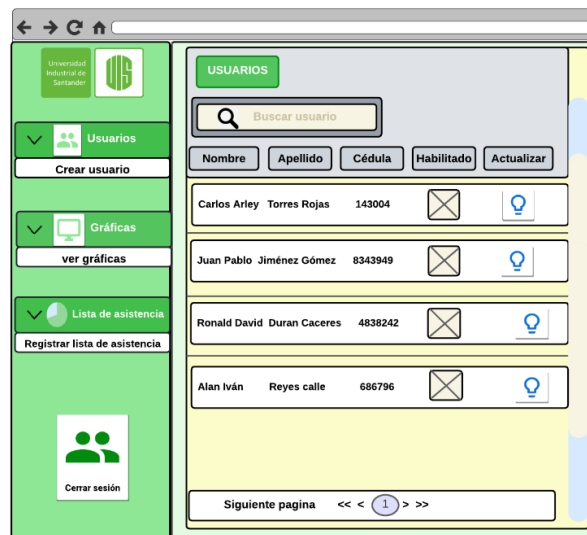


Figura 22 Mockup para la interfaz que muestra la lista de usuarios y permite actualizar información de estos

Fuente: Autor



Figura 23 Mockup para la interfaz de muestra la lista de servicios alimenticios

Fuente: Autor

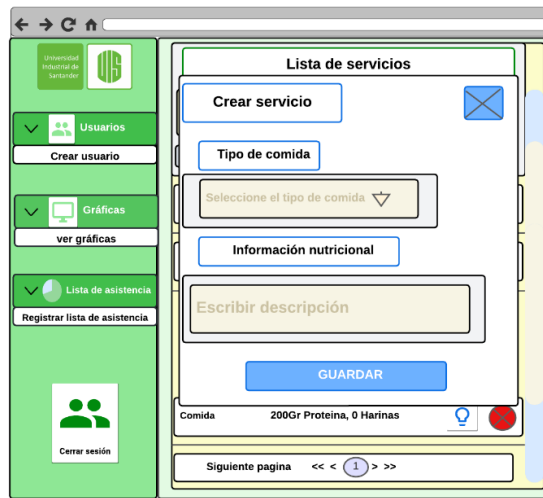


Figura 24 Mockup para la interfaz de permite crear y actualizar los servicios alimenticios

Fuente: Autor

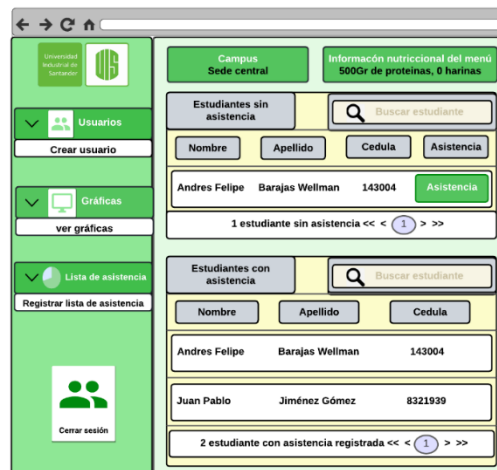


Figura 25 Mockup para la interfaz de la lista de asistencia y control de usuarios del software

Fuente: Autor



Figura 26 Mockup para la interfaz de las gráficas (histórico) de consumo del software

Fuente: Autor

5 Fase de desarrollo e integración

Durante el desarrollo de estas fases se completaron los objetivos específicos 2 y 3 (**Ver sección 3.2**), debido a que se realizó la implementación de la base de datos, y posterior construcción tanto del backend como del frontend de sistema que gestiona los planes de alimentación, esto con el fin de que el producto no tenga inconvenientes a la hora del acoplamiento de ambas tecnologías, logrando así un funcionamiento íntegro y preparándolo para la fase de pruebas unitarias.

5.1 Desarrollo del Backend

En la realización del Backend se utilizó “Express” como “framework” de trabajo de “Node.js”, esto debido a que “Express” proporciona una amplia gama de herramientas a la

hora de desarrollar aplicaciones, y mejorando el rendimiento de la aplicación. Otra de las razones de su uso es que facilitó la configuración de entornos de desarrollo, y creación e implementación de middlewares como reglas de estructura y funcionamiento de peticiones, además de presentar una ventaja frente a otros frameworks ya que proporcionó una forma más eficiente de manejar solicitudes HTTP para las diferentes rutas que con las que se está trabajando.

Adicional a esto, el primer paso que se realizó fue la creación de todos los componentes que fueron utilizados para desarrollar las funcionalidades de la aplicación, entre estos componentes se encuentran los DTO, Middlewares, controladores, rutas y los servicios.

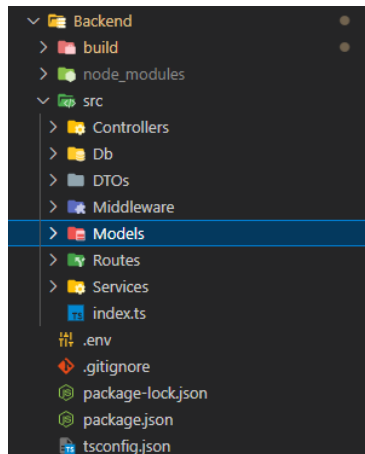


Figura 27 Índice de componentes y herramientas utilizados en la creación del backend

Fuente: Autor

5.1.1 Conexión con la base de datos

La primera actividades que se realizó al desarrollar el backend fue realizar la conexión con la base de datos (**ver figura 7**), para esto se utilizó una clase denominada como “Sequelize” la cual es una herramienta ORM (Object-Relational mapping) proveniente de Node.js, que permite transformar en entidades de un lenguaje de programación orientado a objetos las tablas que son seleccionadas de una base de datos relacional, facilitando de esta forma el acceso a los registros, y teniendo como resultado una base de datos de objetos virtuales.

5.1.2 Implementación de los data transfer objects (DTO)

Los data transfer objects (DTO) son patrones de diseño muy utilizados debido a que permiten el traspaso de información entre las diferentes secciones en las cuales se está desarrollando el software. Es de esta forma que el servidor selecciona las tablas que quiera utilizar de la base de datos, y realiza un mapeo haciendo uso de los DTO, para recopilar aquellos registros u información importante de una o varias tablas, y mostrarlos al realizar una consulta. Lo que viene siendo una herramienta eficiente a la hora de realizar filtros o búsquedas por usuarios.

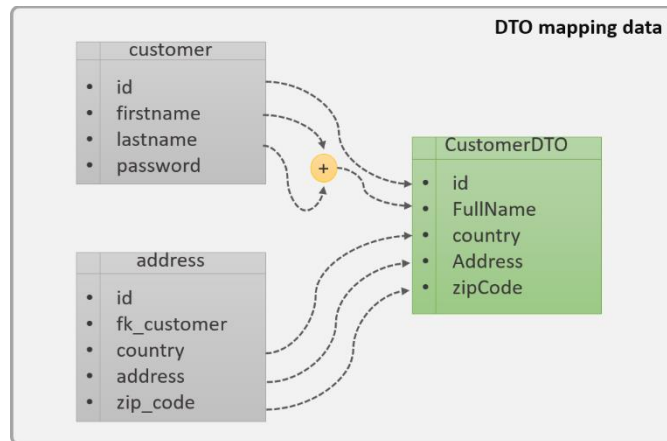


Figura 28 Funcionamiento del mapeo que realizan los DTO

Fuente: [Data Transfer Object \(DTO\)](#)

5.1.3 Definición y desarrollo de los modelos de trabajo

Dentro del desarrollo del backend los modelos son definidos como la representación e interacción de las tablas que contiene la base de datos. Cabe resaltar que los modelos van ligado a la herramienta “Sequelize” debido a que los modelos le proporcionan a esta herramienta los datos de la tabla a la que pertenece, como por ejemplo el título de la tabla, los atributos de la misma, y también las directrices sobre las cuales se va a estructurar toda la información que sea tanto recibida como enviada.

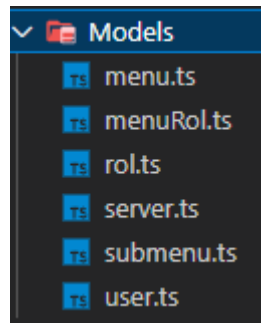


Figura 29 Índice de los modelos de información del proyecto

Fuente: Autor

5.1.4 Construcción de los servicios y funciones operacionales

Se debe establecer una conexión entre SQL y los datos que utiliza Sequelize. Adicional a esto también se importa la base de datos. Se define un nuevo modelo que representa los roles y que guarda los atributos los cuales se encuentran definidos en una clase, y adicional a esto se define el tipo de dato que posee cada atributo. Es importante verificar que se aplique la llave primaria que en este caso es el id.

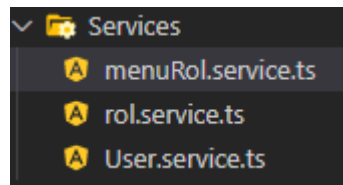


Figura 30 Índice de servicios utilizados en el backend

Fuente: Autor

5.1.5 Implementación de los controladores

Lo primero a tener en cuenta es que es muy importante importar los modelos sobre los cuales va a actuar este servicio y las labores que va a desarrollar. Posteriormente se realizó

la configuración del servicio para obtener la información, es por esto que se crean funciones para que se puedan obtener todos los datos de entrada dependiendo del rol que se esté utilizando, y adicional a esto como parámetro se debe recibir el id que posee dicho rol. Posteriormente se definieron los atributos los cuales se enviarán como respuesta a la función que se está intentando satisfacer.

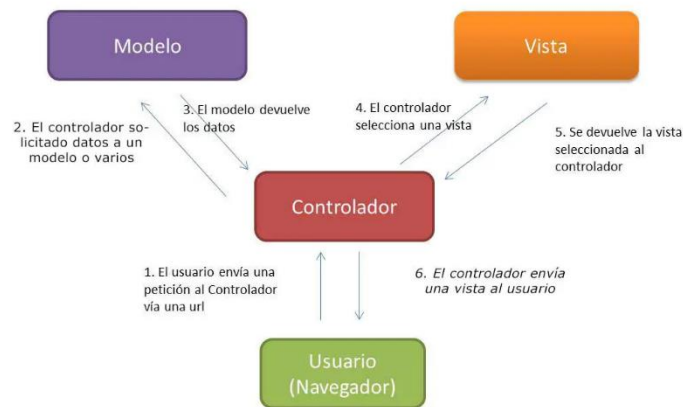


Figura 31 Funcionamiento de los controladores dentro del patrón de arquitectura

MVC

Fuente: [Patrón MVC](#)

Sumando a esta metodología también se implementó un método de seguridad para encriptar los datos que hacen parte de los usuarios inscritos, importando la librería bcryptjs, para encriptar la contraseña según el número de giros que se le solicite, enviando una alerta al usuario con la información de que los datos han sido guardados con éxito. Cabe resaltar que, en términos de usabilidad, este método se utiliza generalmente para enviar

información delicada del usuario como las contraseñas. y adicionalmente se hace uso de JWT como configuración que tendrá la encriptación del token.

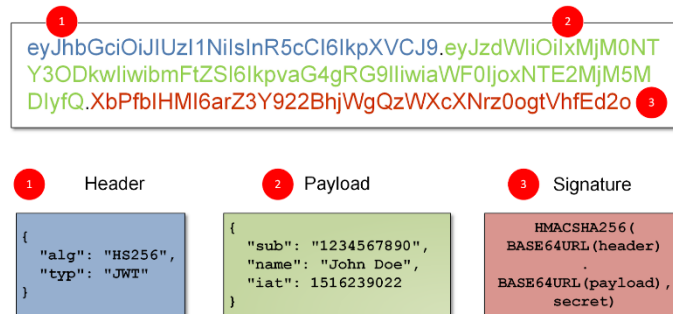


Figura 32 Token JWT concatenando Header, Payload y Signature

Fuente: [JWT Token Explanation](#)

5.1.6 Implementación de middlewares para validación de tokens

Entonces lo primero que se puede apreciar es que de hecho el middleware enfocado en la validación de tokens JWT tiene una estructura similar por no decir igual a la del controlador, esto debido a la capacidad que tienen los middleware como herramienta de software para la autenticación, comunicación de datos entre componentes y aplicaciones, gestión de datos y servicios en un software. Teniendo en cuenta que en el controlador de autenticación se crea el token que se va a consumir en el middleware, este último maneja una funcionalidad para el programa siendo la de validar el token para que sea el correcto; este proceso se lleva a cabo cuando el token del usuario es recuperado por el servidor, y realiza una comparación entre la firma establecida en el controlador y la que es generada por la clave privada. En el caso de que no sea así, se emitirá la respuesta para dar a entender que el token no ha sido encontrado o que simplemente no existe. En caso de que el token

sea válido, se enviarán al realizar una solicitud http, lo que por consiguiente permite que el servidor autentique al usuario por medio de ese token, y envíe una respuesta al cliente, obteniendo como resultado que el middleware como servicio de JWT sea implementado para el consumo de las rutas.

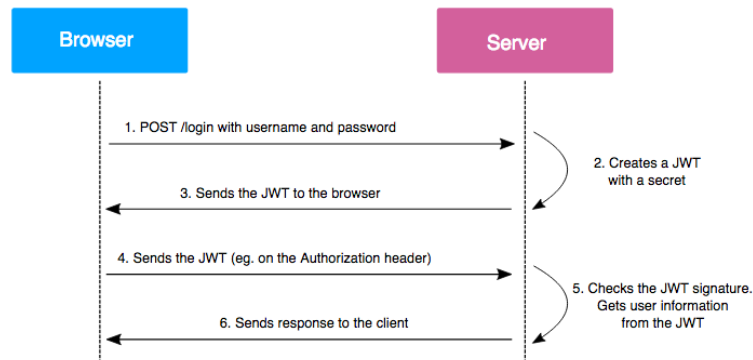


Figura 33 Proceso de autenticación de JWTbackend

Fuente: [Autenticación de JWT](#)

5.1.7 Construcción de las rutas operativas

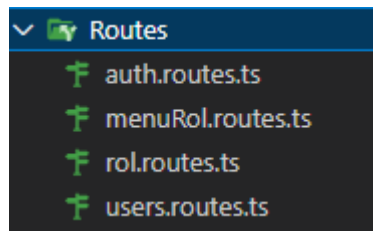


Figura 34 Índice de componentes y herramientas utilizados en la creación del backend

Fuente: Autor

Para realizar la protección de las rutas, se utilizaron los middlewares los cuales se aplicaron a cada una de las rutas realizándose la autenticación y posterior verificación de

los tokens JWT para evitar los accesos no autorizados. Si el usuario no tiene o no ha creado un token no puede realizar consultas o registrar datos, en caso de que la validación haya sido la correcta y el usuario sea autorizado, se realiza un seguimiento a las actividades que el usuario haga dentro de la vista que le proporciona la ruta, y esos cambios son guardados en una solicitud. Es importante resaltar que para realizar solicitudes a las rutas protegidas se puede usar Postman como herramienta de comunicación.

5.2 Desarrollo e implementación del Frontend

El prototipado es la tercera y última fase del desarrollo web. Son los modelos más próximos por utilizar como interfaz funcional dentro del software. Con estos modelos se definió el comportamiento del producto. Sin embargo, para la creación de los prototipos, se realizaron varias actividades dentro del desarrollo.

La primera actividad por realizar fue la elección de librería que se usó en conjunto con angular para construir la parte visual del software. Esta fue PrimeNG, una librería de código abierto con una amplia gama de componentes para crear interfaces de usuario personalizadas y funcionales, disminuyendo el tiempo de desarrollo debido a que no se tenían que crear vistas desde cero, lo cual ayudó a minimizar errores y aumentó la calidad tanto del producto como la productividad del equipo de trabajo durante la fase de desarrollo.

Name	Country	Representative	Date	Status	Activity
James Butt	Algeria	Ioni Bowcher		UNQUALIFIED	
Josephine Darakjy	Egypt	Amy Elsner		PROPOSAL	
Art Venere	Panama	Asiya Javayant		QUALIFIED	
Lenna Paprocki	Slovenia	Xuxue Feng		NEW	
Donette Foller	South Africa	Asiya Javayant	2016-05-20	PROPOSAL	
Simona Morasca	Egypt	Ivan Magalhaes	2018-02-16	QUALIFIED	
Mitsue Tollner	Paraguay	Ivan Magalhaes	2018-02-19	RENEWAL	
Leota Dillard	Serbia	Onyama Limba	2019-08-13	RENEWAL	
Sage Wieser	Egypt	Ivan Magalhaes	2018-11-21	UNQUALIFIED	
Kris Marrier	Mexico	Onyama Limba	2015-07-07	PROPOSAL	

Figura 35 Modulos y componentes de herramientas de PrimeNG

Fuente: [Interfaces PrimeNG](#)

5.2.1 Creación y configuración del proyecto

Luego de haber seleccionado las tecnologías a utilizar en el desarrollo, las librerías para diseño e implementación, y gestionar la lista de actividades que se han creado utilizando la metodología ágil Scrum. La siguiente tarea fue crear el proyecto en angular, e inmediatamente instalar las dependencias de **primeng** junto con **primeicons** para comenzar a utilizar los componentes de la librería.

```
ng generate component login
ng generate component nav
ng generate component user-create
```

Figura 36 Creación de los componentes del software

Fuente: Autor

Antes de entrar en lo que fue todo el desarrollo de las interfaces del software y funcionalidad de las mismas, explicará los componentes en angular, y es que cada uno de estos se encuentra compuesto por los siguientes archivos: El primero será el template, el cual es un archivo HTML donde se va a visualizar la interfaz del usuario. El segundo elemento es un archivo CSS en el cual se encuentran organizados cada uno de los estilos, métricas, colores, etc. Que se aplicarán a las vistas. El tercer archivo es de tipo Typescript es decir que aquí irán el conjunto de clases, funcionalidades, y propiedades que se vayan a ejecutar en la interfaz HTML realizada.

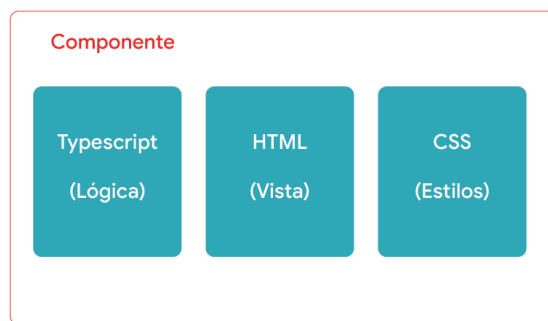


Figura 37 Estructura de un componente en angular

Fuente: [Componentes de angular](#)

5.2.2 Desarrollo de la interfaz del inicio de sesión para los usuarios

La primera actividad dentro del desarrollo de las vistas fue el inicio de sesión del cual sólo los usuarios autorizados de las empresas que ofrecen el plan alimenticio podrán tener acceso a la aplicación. Cabe resaltar que los mockups realizados en la fase de planeación fueron de mucha ayuda ya que a partir de estos ya se tenía establecido el comportamiento

y la estructura visual del sitio web. Posterior a esto y teniendo como hoja de ruta los mockups, se crearon los estilos css que se aplicarían a las interfaces, además de agregar las funcionalidades principales que alojadas en los métodos del archivo typescript, y todas las directivas fueron estructuradas en el archivo html del proyecto.

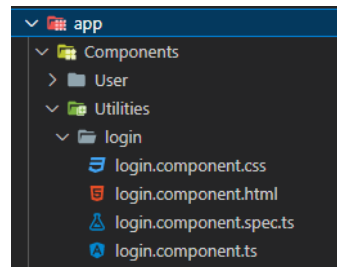


Figura 38 Índice de componentes para el desarrollo de la interfaz del inicio de sesión

Fuente: Autor

De esta forma el primer módulo a realizar fue la creación e implementación de la vista de inicio de sesión del sistema, el cual le permitirles únicamente a los usuarios autorizados que pertenecen a las empresas que manejan los planes alimenticios realizar su autenticación de identidad como método fundamental para garantizar la seguridad del software de gestión. Además de mantener un control respecto a las acciones que puede realizar el usuario en el software o los datos a los que este puede acceder. Cabe resaltar que siguiendo las etapas del desarrollo web, se diseñó una interfaz personalizada para proporcionarle a los usuarios autorizados no solo un acceso seguro sino también eficiente e intuitivo de usar.

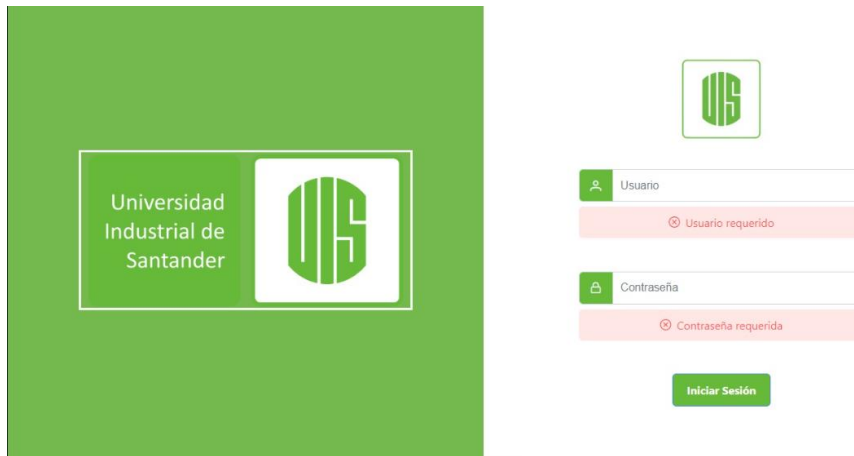


Figura 39 Prototipo del inicio de sesión del software

Fuente: Autor

5.2.3 Desarrollo de la barra de navegación para gestión de operaciones

La siguiente actividad que se desarrolló fue la creación de un complemento vital para las vistas que se desarrollaron posteriormente, y este fue la implementación de una barra de navegación. Que le permite al usuario desplazarse de una manera fácil e intuitiva por el software lo cual aumenta la usabilidad y eficiencia del mismo, ya que permite encontrar funciones específicas como lo son para este proyecto el apartado para insertar usuarios nuevos que vayan a hacer parte de un plan alimenticio, el apartado que se encarga de gestionar la asistencia de los usuarios que hacen parte del plan, y el apartado de las gráficas donde se pueden revisar y analizar los datos de consumo de los usuarios.



Figura 40 Barra de navegación del software de gestión de planes alimenticios

Fuente: Autor

5.2.4 Desarrollo de la interfaz de inserción de usuarios

Luego de haber de haber implementado la barra de navegación, se realizaron los módulos operativos que llevan todas las funciones del software, el primero fue la vista de inserción de usuarios el cual consiste en el diligenciamiento de un formulario diseñado para que solos aquellos usuarios autorizados que lleven a cabo este proceso puedan insertar los datos de los usuarios que hacen parte del plan alimenticio de una manera eficiente e intuitiva, protegiendo de esta forma la disponibilidad de la información, y facilitando de esta forma la gestión de los datos de la cual se hablará más adelante.

El prototipo de la interfaz de inserción de usuarios del software se divide en dos secciones principales. A la izquierda, un menú lateral con fondo verde claro muestra tres categorías: 'Usuarios' con el botón 'Crear Usuario', 'Graficas' con 'Ver Graficas', y 'Lista de asistencia' con 'Registrar lista de asistencia'. En la parte inferior del menú hay un ícono de usuario y el texto 'Cerrar sesion'. A la derecha, un formulario titulado 'Crear nuevo usuario' contiene los siguientes campos y mensajes de error:

- Nombres:** Mensaje de error: "El campo nombres no puede estar vacío".
- Apellidos:** Mensaje de error: "El campo apellidos no puede estar vacío".
- Numero de Identificación:** Mensaje de error: "El numero de identificación debe registrarse con solo numeros y deben ser minimo 8 caracteres".
- Documento:** Mensaje de error: "Debe seleccionar el tipo de documento".
- Usuario:** Mensaje de error: "El usuario es requerido".
- Contraseña:** Mensaje de error: "La contraseña debe contener minimo 8 caracteres una, letra mayúscula y un numero".
- Rol de usuario:** Mensaje de error: "Debe seleccionar un rol para el usuario".

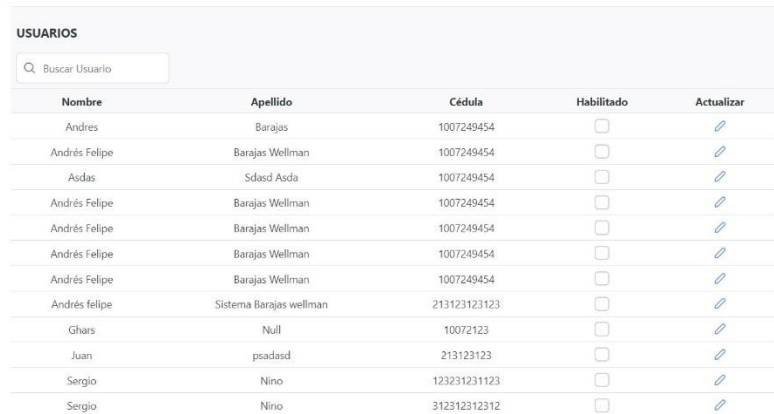
En la parte inferior derecha del formulario hay un botón verde que dice 'Guardar Usuario'.

Figura 41 Prototipo de la interfaz de inserción de usuarios del software

Fuente: Autor

5.2.5 Desarrollo de la interfaz que muestra la lista de usuarios y permite actualizar información de estos

Para el manejo dinámico de la información personal del usuario inscrito en el plan, se creó la siguiente interfaz, la cual permite revisar en tiempo real los datos de los usuarios, y así mismo cuenta con la opción de actualizar estos. Facilitando la sincronización instantánea de la información en la base de datos. La ventaja de esta implementación reside en asegurar la exactitud y disponibilidad de la información para con los perfiles de los usuarios.

Figura 42 Prototipo de la interfaz del listado de usuarios y actualización de los mismos

Nombre	Apellido	Cédula	Habilitado	Actualizar
Andres	Barajas	1007249454	<input type="checkbox"/>	✎
Andrés Felipe	Barajas Wellman	1007249454	<input type="checkbox"/>	✎
Asdas	Sdasd Asda	1007249454	<input type="checkbox"/>	✎
Andrés Felipe	Barajas Wellman	1007249454	<input type="checkbox"/>	✎
Andrés Felipe	Barajas Wellman	1007249454	<input type="checkbox"/>	✎
Andrés Felipe	Barajas Wellman	1007249454	<input type="checkbox"/>	✎
Andrés Felipe	Barajas Wellman	1007249454	<input type="checkbox"/>	✎
Andrés Felipe	Sistema Barajas wellman	213123123123	<input type="checkbox"/>	✎
Ghars	Null	10072123	<input type="checkbox"/>	✎
Juan	psadasd	213123123	<input type="checkbox"/>	✎
Sergio	Nino	123231231123	<input type="checkbox"/>	✎
Sergio	Nino	312312312312	<input type="checkbox"/>	✎

Figura 41 Prototipo de la interfaz de inserción de usuarios del software

Fuente: Autor

5.2.6 Desarrollo de la interfaz que muestra la lista de servicios alimenticios

De la misma forma que se implementó una interfaz para el listado y gestión de los datos de los usuarios, también se realizó el mismo procedimiento para los servicios ofrecidos dentro de los planes de alimentación. Dependiendo del plan en el que se encuentre afiliado el usuario, se le asignara un servicio diferente (desayuno, almuerzo, cena, etc.), la interfaz también contará con una breve descripción nutricional de lo que se va consumir para poder manejarlo como si de un menú estándar se tratase.

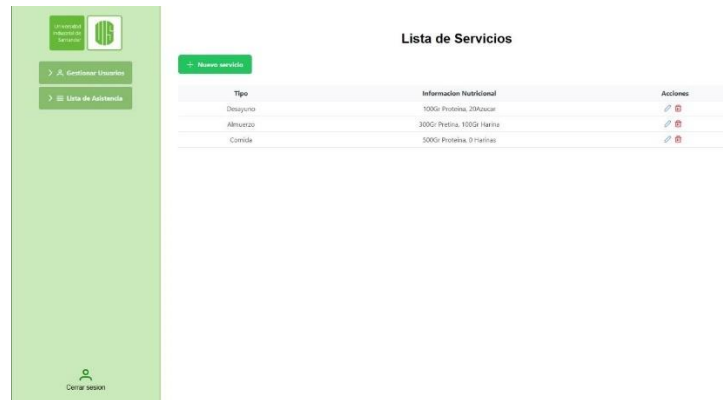


Figura 43 Prototipo de la interfaz del listado de servicios alimenticios

Fuente: Autor

5.2.7 Desarrollo de la interfaz que permite crear y actualizar los servicios alimenticios

La siguiente interfaz es un complemento de la interfaz presentada en el punto anterior, ya que la funcionalidad de esta reside en crear y así mismo poder actualizar los servicios de alimentación para los usuarios inscritos.

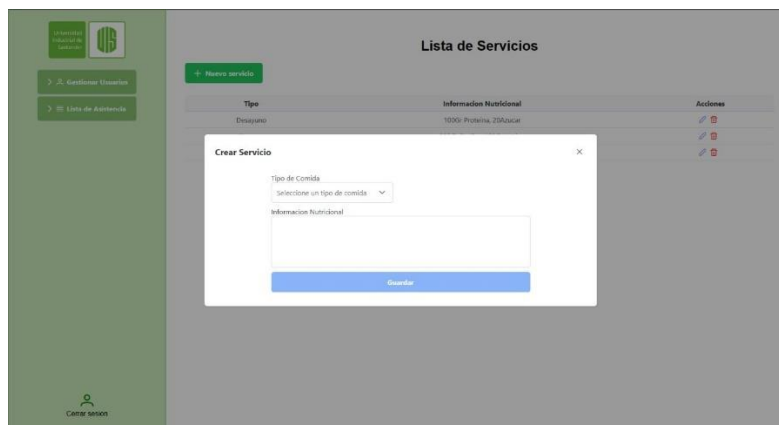


Figura 44 Prototipo de la interfaz del listado de usuarios y actualización de estos

Fuente: Autor

5.2.8 Desarrollo de la interfaz asistencia y gestión de usuarios

La creación de formularios para la inserción de usuarios fue un paso importante, ya que al guardarse todos los registros en la base de datos el usuario autorizado tiene acceso a la búsqueda y recuperación de los mismos; por consiguiente, se desarrolló la vista que gestiona la toma de asistencia y control de los usuarios que hacen parte del plan alimenticio. De esta forma se puede visualizar la información perteneciente a los inscritos, así como el lugar al cual se encuentran inscritos, entre otros aspectos importantes como el tipo de servicio que prestan los comedores, y mediante una “**checklist**” determinar si los usuarios han recibido el alimento en la periodicidad acordada.

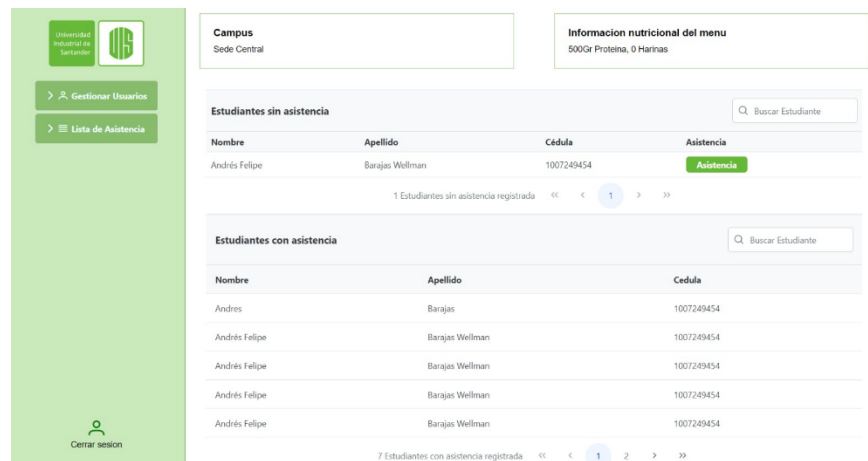


Figura 45 Prototipo de la interfaz de la lista de asistencia y control de usuarios del software

Fuente: Autor

6 Fase de pruebas

6.1 Pruebas unitarias de desarrollador

Con relación al seguimiento que se le está dando a las fases del ciclo de vida del software con la construcción del prototipo del software, la siguiente capa a abordar fueron las pruebas. Los primeros testeos en realizarse fueron las pruebas unitarias de desarrollador, las cuales buscan evaluar que cada uno de las componentes y funcionalidades integradas en el software tanto para el backend como para el frontend funcionen de la manera adecuada. Cabe resaltar que esta fase no necesariamente mantiene el orden del ciclo de vida, ya que durante la fase de desarrollo se prueba cada componente en un ambiente individual para determinar si hay errores, y de ser así corregirlos de manera temprana. Para un mejor entendimiento de lo que se realizó en las pruebas unitarias se adjuntara el siguiente vídeo el cual enseña de manera completa el funcionamiento del software para gestión de planes de alimentación, junto con cada una de sus funciones: [Prototipo de software para la gestión y monitoreo de programas de alimentación. - YouTube](#)

6.2 Pruebas para rendimiento de peticiones http con Jmeter

Apache Jmeter es una herramienta muy utilizada que permite realizar pruebas de estrés o estabilidad, de tal manera que se pueda medir el rendimiento del software dentro de un entorno de carga simulada. Siguiendo este contexto, se realizaron pruebas para verificar el tiempo de peticiones concurrentes.

6.2.1 Configuración de entorno de desarrollo

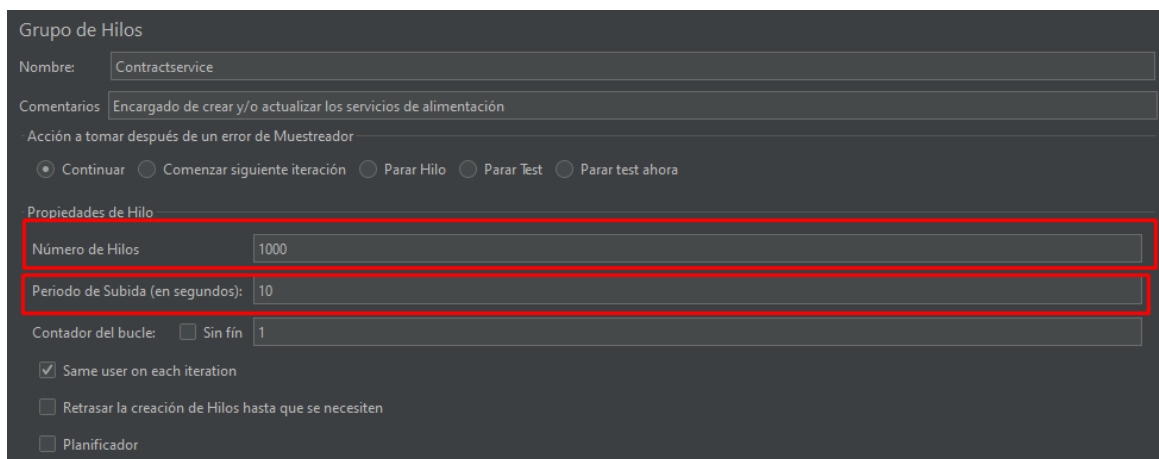
Lo primero que se realizó fue la configuración del ambiente, manipulando el Thread Group (grupo de usuarios) configurando cada uno de los “samplers” HTTP, que en este caso serán peticiones tanto GET, POST, UPDATE y DELETE para incluir tanto los parámetros específicos de cada escenario, como las URL de gestión. Seguido a esto se debe estipular un número de hilos (Grupo de usuarios que hacen peticiones), el intervalo de tiempo incremental, y las repeticiones. En el caso exclusivo de las peticiones POST se incluyó todos aquellos parámetros y datos que emularían la transferencia de información al servidor en los Sampler HTTP. Finalmente se debe incluir para la verificación de las pruebas, los elementos de monitoreo, denominados como “Listener” dentro de los cuales se resaltan los View Results Tree (Arbol de resultados) que muestra el trámite de registros, y el Summary Report (informe de resultados) el cual muestra dentro del entorno de ejecución un análisis en tiempo real de los tiempos de respuesta y las tasas de error.

6.2.2 Pruebas de estrés

Las pruebas de estrés se encargan de determinar la solides del sistema. La configuración del ambiente de ejecución es simple, se tienen en cuenta dos parámetros de entrada, el primero de estos es la configuración de un número de hilos que tal y como se menciona anteriormente se entiende como las peticiones que se hacen al sistema. Es de esta forma que, a mayor número de hilos, más peticiones se realizaran al sistema. El segundo parámetro de entrada es la variación de tiempo. Por ejemplo: Si se agregan 100 hilos y un tiempo variable de 10, se da a entender que cada 1 segundos entran 10 usuarios. Es así

como se determina si el sistema es capaz de aguantar esa cantidad X de peticiones, o si presenta errores.

Para el ejemplo se evaluará una cantidad de hilos de 1000, dentro de las cuales irán en un periodo de subida de 10 segundos, es decir que cada segundo ingresó 100 solicitudes.



Grupo de Hilos

Nombre: Contractservice

Comentarios: Encargado de crear y/o actualizar los servicios de alimentación

Acción a tomar después de un error de Muestreador

Continuar Comenzar siguiente iteración Parar Hilo Parar Test Parar test ahora

Propiedades de Hilo

Número de Hilos: 1000

Periodo de Subida (en segundos): 10

Contador del bucle: Sin fin 1

Same user on each iteration

Retrasar la creación de Hilos hasta que se necesiten

Planificador

Figura 47 Configuración de prueba de estrés en Jmeter

Fuente: Autor

Posteriormente se realiza la configuración del tipo de petición que se quiere realizar, cabe resaltar que se realizaron peticiones a cada servicio, y con diferentes solicitudes como GET, POST, PATCH, PUT, DELETE. Pero en este caso la prueba que se mostrará fuera al servicio de contrato, es decir el encargado de gestionar la creación y actualización de servicios como lo son el desayuno, el almuerzo o la cena.

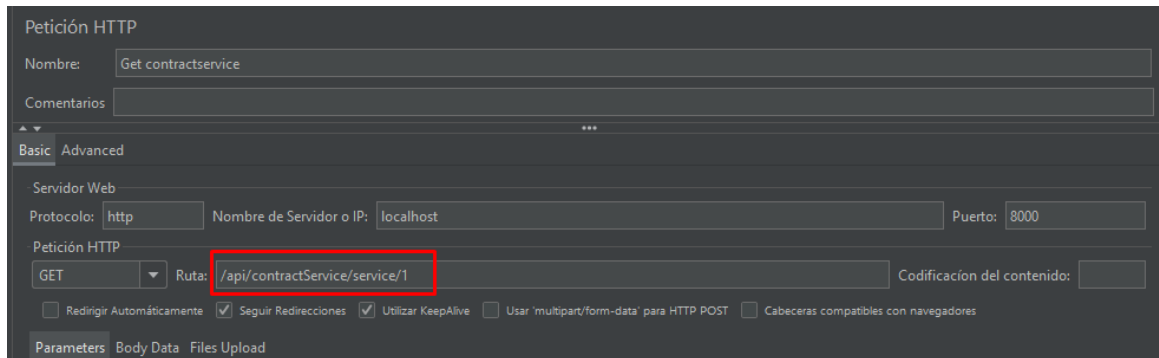


Figura 48 Configuración del entorno para petición GET en Jmeter

Fuente: Autor

Luego de haber realizado la prueba de estrés, se podrá evidenciar en el árbol de resultados que las 1000 peticiones fueron realizadas de manera, mostrando los datos de respuesta.

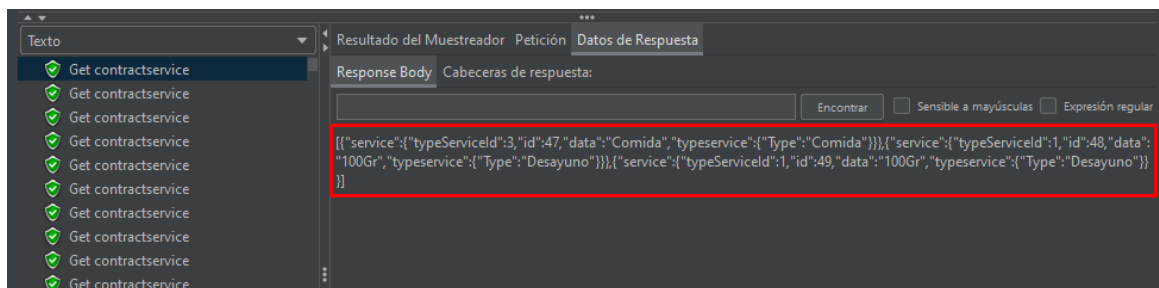


Figura 49 Árbol de resultados con datos de respuesta correcta

Fuente: Autor

En el reporte resumen, se revisa las estadísticas de las pruebas de estrés realizadas tales como el rendimiento de ejecución de las 1000 muestras, el porcentaje de error que en este caso no aplica y variables como la media de ejecución y la desviación estándar.

Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Byt...
Get contract...	1000	2	0	10	0,72	0,00%	100,1/sec	52,38	17,59	536,0
Total	1000	2	0	10	0,72	0,00%	100,1/sec	52,38	17,59	536,0

Figura 50 Reporte resumen de la prueba de estrés realizada

Fuente: Autor

Finalmente se muestra el gráfico de resultados de transacciones por segundo, donde se puede apreciar que el rendimiento del servicio se aumenta en los primeros segundos, y luego se mantiene constante aún para la configuración del entorno que se programó. En caso de que las fallas en la transacción fueran mayor a las transacciones exitosas, se daría a entender que el software tiene un problema con el almacenamiento de los servicios. Siendo de esta forma que, según las pruebas de estrés realizadas a cada servicio, verifica que el software es funcional aún en condiciones que exijan un rendimiento considerable.

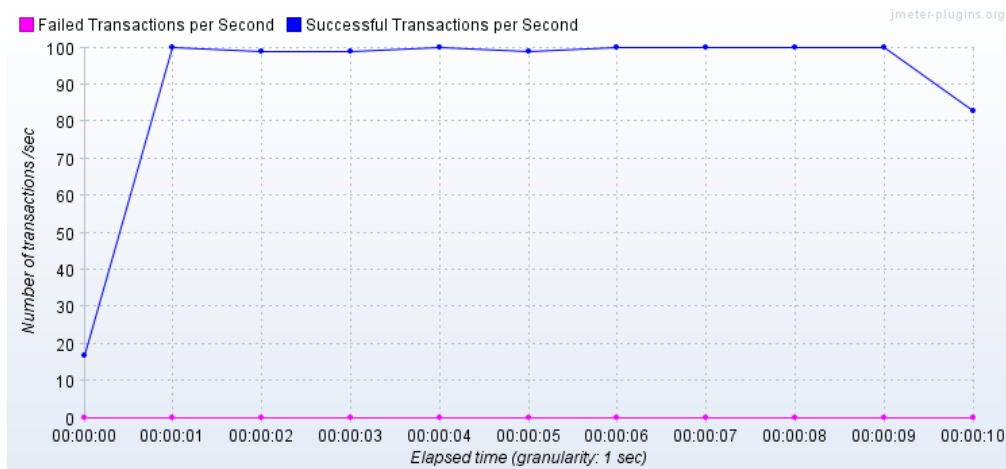


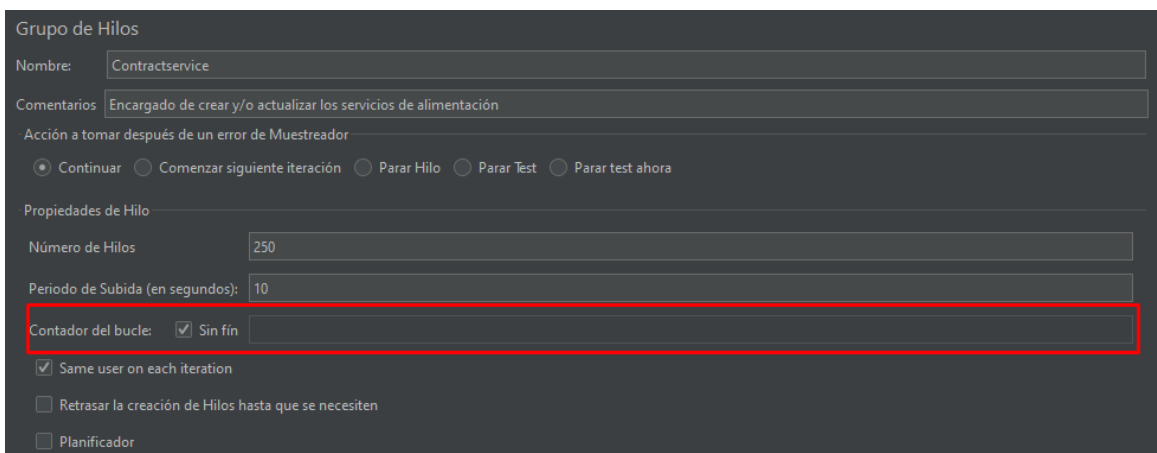
Figura 51 Gráfico de de número de transacciones por tiempo de la prueba de estrés realizada

Fuente: Autor

6.2.3 Pruebas de estabilidad

Las pruebas de estabilidad se utilizaron para determinar si el software aguantara una un tiempo continuo de cargas continuas de repeticiones, es decir, que, si el software aguanta 15 hilos, estos se ingresan en un ciclo infinito, donde cada vez que se realicen las 15 peticiones, estas se doblaran y ahora serán 30 siendo esta la forma en la que mide el límite de solicitudes, y para determinar si hay fugas en el software.

Para el siguiente ejemplo también realizado al contractService se disminuyó el número de hilos a 250, cuyo intervalo será de 10 segundos, pero cuyo contador del bucle no tendrá fin, es decir que luego de ejecutarse esas 250 solicitudes se volverá a realizar el mismo procedimiento una y otra vez hasta que se ponga pausa a la prueba.



Grupo de Hilos

Nombre:

Comentarios

Acción a tomar después de un error de Muestreador

Continuar Comenzar siguiente iteración Parar Hilo Parar Test Parar test ahora

Propiedades de Hilo

Número de Hilos

Periodo de Subida (en segundos):

Contador del bucle: Sin fin

Same user on each iteration

Retrasar la creación de Hilos hasta que se necesiten

Planificador

Figura 52 Configuración del entorno para prueba de estabilidad en Jmeter

Fuente: Autor

Tal y como se evidencia en la siguiente imagen, se evaluaron un total de 20003 muestras antes de que se detuviera la prueba de manera manual, es por esto mismo que el porcentaje de error es de 1.25%, ya que como se detuvo a la fuerza, algunas peticiones no se alcanzaron a ejecutar.

Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Byt...
Get contract...	20003	206	0	324	71,36	1,25%	907,6/sec	497,44	157,54	561,2
Total	20003	206	0	324	71,36	1,25%	907,6/sec	497,44	157,54	561,2

Figura 53 Reporte resumen de la prueba de estabilidad realizada

Fuente: Autor

En la siguiente imagen a diferencia de la prueba de estrés, las muestras son demasiado grandes; sin embargo, aún se puede apreciar que las transacciones a mayor número de muestras aumentan, lo que establece una relación directamente proporcional, y se mantiene por encima de las fallas, lo que da a entender que la realización de las pruebas para cada servicio verifica que el software es estable. El número de fallas que se hace presente al final es debido a que como la ejecución del entorno se hizo dentro de un bucle, al detenerse forzosamente para apreciar los resultados, se entiende que hay peticiones que no se pueden enviar, lo que por ende genera un error; sin embargo, esto no genera un problema en la prueba realizada

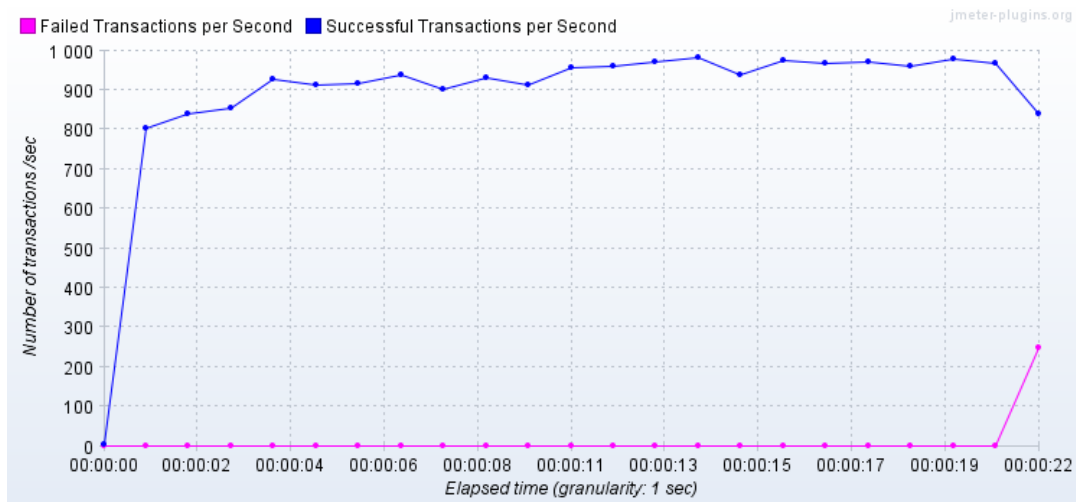


Figura 54 Gráfico de número de transacciones por tiempo de la prueba de estabilidad realizada en Jmeter

Fuente: Autor

7 Conclusiones

Luego de todo el trabajo realizado durante el desarrollo del proyecto de grado, puede determinarse que si se dio cumplimiento del objetivo principal: Desarrollar un prototipo de herramienta de software que permita gestionar y monitorear programas de alimentación en empresas de la industria alimentaria.

La realización del proyecto involucró un arduo trabajo de ingeniería en conjunto con la aplicación de cada una de las fases del ciclo de vida del software estableciendo una hoja de ruta para el cumplimiento de los demás objetivos propuestos.

La automatización de actividades es un enfoque cada vez más utilizado para administrar, agilizar y sobre todo supervisar procesos que antes se realizaban manualmente. El automatizar los programas de alimentación mejoran la eficiencia, la precisión y la calidad

de la recopilación y análisis de datos importantes para gestionar un programa, entregando a las entidades que acogen este sistema una herramienta que les proporciona ventajas técnicas significativas como la gestión centralizada para el manejo de información, la personalización de los menús, la trazabilidad de cada uno de los consumos realizados por los usuarios, la seguridad de los datos, la escalabilidad según la organización que lo acoja, y la optimización de recursos en cuanto a una herramienta robusta y eficiente.

Es importante recalcar que, durante todo el proceso de construcción del proyecto, existieron varias situaciones que probaron las capacidades de ambos autores como futuros ingenieros de sistemas para completar cada una de las actividades relacionadas al ciclo de vida del software impartidas, fortaleciendo las habilidades de análisis, investigación, y programación, para encontrar las alternativas óptimas para la culminación de un proyecto de grado adecuado. La fase de pruebas fue de vital importancia ya permitió probar de manera tanto conjunta como individual en diferentes escenarios controlados, cada uno de los módulos que componen el software, comprobando hallazgos que fueron intervenidos de manera temprana, y comprobando la eficiencia de la herramienta.

La experiencia recogida ha sido muy enriquecedora, no solo para poder aplicar cada uno de los conocimientos vistos durante la carrera, sino también para aprender sobre cómo funciona la industria del desarrollo del software, al utilizar herramientas, servicios, y prácticas de trabajo que se utilizan día a día durante este sector laboral.

8 Trabajo futuro

Como trabajo futuro se plantea el expandir el software a una escala más global, para abarcar otras áreas de atención modificando el prototipo de software para adaptarse a hoteles, cafeterías y/o restaurantes del sector.

Como principal punto a trabajar para trabajado futuro es que la automatización de las gráficas también se personalice para cada usuario inscrito en la plataforma, de tal forma que se puedan obtener estadísticas individuales de consumo.

Otro punto adicional es que se busca mejorar el software para que en caso de que se necesite subir algún documento relacionado a la planilla de inscritos, o históricos de las tarifas de consumo, exista una función que permita cargar dichos documentos, para llevar un control de la información de la información ya sea de elementos propios de la empresa, o de los usuarios que hacen parte de esta.

Adicionalmente, y enfocado al frontend del software, se mejorarían cada una de las vistas agregando algunos elementos estéticos como las fotos de los usuarios que hacen parte del plan alimenticio, o imágenes que identifiquen cada una de las comidas asociadas a los servicios de desayuno, almuerzo y cena.

Lista de referencias

Amazon. (s.f.). *amazon.com*. Obtenido de <https://aws.amazon.com/es/devops/what-is-devops/>

Amazon. (s.f.). *amazon.com*. Obtenido de <https://aws.amazon.com/es/microservices/>

Atlassian.com. (s.f.). *Atlassian*. Obtenido de <https://www.atlassian.com/es/git/tutorials/what-is-version-control>

Bigelow, S. J. (Agosto de 2021). *computerweekly.com*. Obtenido de

<https://www.computerweekly.com/es/definicion/Gestion-de-la-configuracion>

Ceupe. (s.f.). *ceupe.com*. Obtenido de <https://www.ceupe.com/blog/que-es-y-en-que-consiste-la-trazabilidad.html>

decidesoluciones. (03 de Septiembre de 2019). *decidesoluciones.es*. Obtenido de

[https://decidesoluciones.es/arquitectura-de-](https://decidesoluciones.es/arquitectura-de-microservicios/#:~:text=%C2%BFQu%C3%A9%20es%20una%20arquitectura%20de%20microservicios%3F%20La%20arquitectura,y%20aut%C3%B3noma%2C%20proporcionando%20una%20funcionalidad%20de%20negocio%20completa.)

[microservicios/#:~:text=%C2%BFQu%C3%A9%20es%20una%20arquitectura%20de%20microservicios%3F%20La%20arquitectura,y%20aut%C3%B3noma%2C%20proporcionando%20una%20funcionalidad%20de%20negocio%20completa.](https://decidesoluciones.es/arquitectura-de-microservicios/#:~:text=%C2%BFQu%C3%A9%20es%20una%20arquitectura%20de%20microservicios%3F%20La%20arquitectura,y%20aut%C3%B3noma%2C%20proporcionando%20una%20funcionalidad%20de%20negocio%20completa.)

Digite. (s.f.). *digite.com*. Obtenido de <https://www.digite.com/agile/scrum-methodology/>

Ekon. (30 de Junio de 2020). *Ekon.es*. Obtenido de <https://www.ekon.es/blog/tipos-trazabilidad-tecnologia/>

F5 GLOSSARY. (s.f.). *f5.com*. Obtenido de

https://www.f5.com/es_es/services/resources/glossary/what-is-infrastructure-as-code

ISO. (2015). *iso.org*. Obtenido de <https://www.iso.org/obp/ui/es/#iso:std:iso:9000:ed-4:v1:en>

Larios, B. (28 de julio de 2022). *AGN*. Obtenido de <https://agn.gt/presentan-aplicacion-movil-para-el-programa-de-alimentacion-escolar/>

María, A., & López, H. (s/f). propuesta de protocolo de trazabilidad sanitaria para los menús ofertados por los servicios de alimentación pontificia universidad javeriana. Edu.co. Recuperado el 12 de octubre de 2023, de <https://repository.javeriana.edu.co/bitstream/handle/10554/50017/Trabajo%20de%20grado%20ana%20maria%20hurtado%20final%202020.pdf?sequence=1&isAllowed=y>

Microsoft. (s.f.). *microsoft.com*. Obtenido de <https://azure.microsoft.com/es-es/topic/what-is-kubernetes/#overview>

Microsoft. (s.f.). *microsoft.com*. Obtenido de <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-devops/>

Netapp. (s.f.). *netapp.com*. Obtenido de <https://www.netapp.com/es/devops-solutions/what-is-devops/>

Powerdata.es. (26 de Noviembre de 2018). *blog.powerdata.es*. Obtenido de <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/cloud-computing-ejemplos-y-casos-de-uso-en-empresas>

proyectosagiles. (s.f.). *proyectosagiles.org*. Obtenido de <https://proyectosagiles.org/que-es-scrum/>

Redhat. (31 de Enero de 2018). *redhat.com*. Obtenido de <https://www.redhat.com/es/topics/devops/what-is-ci-cd>

redhat. (11 de Mayo de 2022). *redhat.com*. Obtenido de

<https://www.redhat.com/es/topics/automation/what-is-infrastructure-as-code-iac>

Redhat. (19 de Julio de 2022). *redhat.com*. Obtenido de

<https://www.redhat.com/es/devops/what-is-agile-methodology>

Roche, J. (s.f.). *deloitte.com*. Obtenido de

<https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html>

Santos, J. M. (13 de 7 de 2022). project-management. Obtenido de [https://project-](https://project-management.com/the-definitive-guide-to-devops/)

[management.com/the-definitive-guide-to-devops/](https://project-management.com/the-definitive-guide-to-devops/)

Jurídica, S., De, M., & Nacion~u, E. (s/f). REPUBLICA DE COLOMBIA . 'RESIDENCIA

OE U\ HEPÚBLICA. Gov.co. Recuperado el 21 de febrero de 2023, de

<https://dapre.presidencia.gov.co/normativa/normativa/DECRETO%20218%20DEL%2014%20DE%20FEBRERO%20DE%202020.pdf>

Nota técnica ESTRATEGIA DE TRANSPARENCIA Y EFICIENCIA EN EL PROGRAMA

DE ALIMENTACIÓN ESCOLAR (PAE). (s/f). Gov.co. Recuperado el 21 de febrero

de 2023, de https://www.mineducacion.gov.co/1780/articles-363488_recurso_8.pdf

Alimentos Para Aprender. (s/f). Gov.co. Recuperado el 21 de febrero de 2023, de

<https://www.alimentosparaaprender.gov.co/>

Atlassian. (s/f). *Comparación entre la arquitectura monolítica y la arquitectura de*

microservicios. Atlassian. Recuperado el 12 de marzo de 2023, de

<https://www.atlassian.com/es/microservices/microservices-architecture/microservices-vs-monolith>

De sistemas monolíticos a microservicios. (s/f). Aplyca Tecnología SAS. Recuperado el 13 de marzo de 2023, de <https://www.aplyca.com/blog/de-sistemas-monoliticos-microservicios>

Atlassian. (s/f-b). *Jira Software: Funciones.* Atlassian. Recuperado el 13 de marzo de 2023, de <https://www.atlassian.com/es/software/jira/features>

Asana. (s/f). *Qué es Scrum y cómo aplicarlo en gestión de proyectos.* Asana. Recuperado el 13 de marzo de 2023, de <https://asana.com/es/resources/what-is-scrum>

El Tiempo, T. (2020, April 29). *Así funciona el programa de alimentación escolar PAE.* El Tiempo. <https://www.eltiempo.com/vida/educacion/quienes-son-beneficiarios-del-programa-de-alimentacion-escolar-pae-490024>

Don Bosco Popayán, I. (2022, April 19). *Así funciona el Programa de Alimentación Escolar (PAE) en nuestra institución.* Salesianos Popayán.

<https://www.donbosco.edu.co/post/as%C3%AD-funciona-el-programa-de-alimentaci%C3%B3n-escolar-pae-en-nuestra-instituci%C3%B3n>

(N.d.). Gov.Co. Retrieved March 14, 2023, from

https://www.mineducacion.gov.co/1621/articles-235135_archivo_pdf_lineamientos_tecnicos.pdf

Arquitectura de Monolítico. (n.d.). Reactiveprogramming.io. Retrieved March 15, 2023, from

<https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/monolitico>

¿*Qué es una base de datos relacional (RDBMS)?* (n.d.). Google Cloud. Retrieved March 16,

2023, from <https://cloud.google.com/learn/what-is-a-relational-database?hl=es-419>

¿*Qué es una SPA? (Angular)*. (n.d.). Programaenlinea.net. Retrieved March 16, 2023, from

<https://www.programaenlinea.net/una-spa-angular/>

Single Page Application: Un viaje a las SPA a través de Angular y JavaScript. (2018,

December 31). Medium. [https://davidjguru.medium.com/single-page-application-un-](https://davidjguru.medium.com/single-page-application-un-viaje-a-las-spa-a-trav%C3%A9s-de-angular-y-javascript-337a2d18532)

[viaje-a-las-spa-a-trav%C3%A9s-de-angular-y-javascript-337a2d18532](https://davidjguru.medium.com/single-page-application-un-viaje-a-las-spa-a-trav%C3%A9s-de-angular-y-javascript-337a2d18532)

Abacus Consulting News. (2018, January 30). *Desarrollo de aplicaciones para web, tabletas*

y móviles con AngularJS. Abacus Consulting. [https://abacus-consulting.net/desarrollo-](https://abacus-consulting.net/desarrollo-de-aplicaciones-para-web-tabletas-y-moviles-con-angulars-js/)

[de-aplicaciones-para-web-tabletas-y-moviles-con-angulars-js/](https://abacus-consulting.net/desarrollo-de-aplicaciones-para-web-tabletas-y-moviles-con-angulars-js/)

Villa, M. (2021, November 23). *Securing your Applications in Node.js - Part 1/3*. The

NodeSource Blog - Node.js Tutorials, Guides, and Updates; The NodeSource Blog

#shoptalk | The Enterprise Node Company™ Providing Enterprise Node.js Training,

Support, Software & Consulting, Worldwide. [https://nodesource.com/blog/NodeJS-](https://nodesource.com/blog/NodeJS-securing-your-applications-1)

[securing-your-applications-1](https://nodesource.com/blog/NodeJS-securing-your-applications-1)

Caules, C. Á. (2015, June 12). *Angular.js vs Node.js*. Arquitectura Java.

<https://www.arquitecturajava.com/angular-js-vs-node-js/>

Colectiva, N. (2020, October 3). 10 Ventajas de usar Node JS – Parte 1. *Nubecolectiva.com*.

<https://blog.nubecolectiva.com/10-ventajas-de-usar-node-js-parte-1/>

Fernández, J. S. (2017, September 4). *China estrena el pago mediante reconocimiento facial en restaurantes*. Cinco Días.

https://cincodias.elpais.com/cincodias/2017/09/04/lifestyle/1504529247_096440.html

Auth. (n.d.). *JSON Web Token Claims*. Auth0 Docs. Retrieved March 22, 2023, from

<https://auth0.com/docs/secure/tokens/json-web-tokens/json-web-token-claims>

What is JWT (JSON Web Token)? How does JWT Authentication work? (2021, December 8).

Blog - MiniOrange. <https://blog.miniorange.com/what-is-jwt-json-web-token-how-does-jwt-authentication-work/>

JWT tokens and security - working principles and use cases. (2016, December 20).

VAADATA - Ethical Hacking Services. <https://www.vaadata.com/blog/jwt-tokens-and-security-working-principles-and-use-cases/>

Pasos para crear un JWT. (n.d.). Readthedocs.Io. Retrieved March 22, 2023, from

<https://processa-aspen.readthedocs.io/en/latest/JWT-Build/>

Salgado, C. (n.d.). *Mosaic*. Uoc.edu. Retrieved March 23, 2023, from

<https://mosaic.uoc.edu/2015/09/15/proceso-de-desarrollo-de-un-proyecto-digital/>

Publicado por pmoinformatica.com. (n.d.). *Elementos del diagrama de casos de uso*.

Pmoinformatica.com. Retrieved March 30, 2023, from

<http://www.pmoinformatica.com/2021/02/elementos-diagrama-casos-de-uso.html>

Abiztar. (n.d.). *Casos a incluir y casos a extender*. Com.mx. Retrieved March 30, 2023, from

<https://www.abiztar.com.mx/articulos/casos-a-incluir-casos-a-extender.html>

Casos De Uso “Incluir” Y “Extender” - Visual Paradigm Blog Español. (2022, February 9).

Visual Paradigm Blog Español. <https://blog.visual-paradigm.com/es/include-and-extend-use-cases/>

Diseño e implementación del front-end para la plataforma de internet de las cosas Uaoiot Juan

camilo Garcés viveros código: 2130444 Ivan Dario Mosquera Triviño código:

2130125 universidad autónoma de occidente facultad de ingeniería departamento de

operaciones y sistemas programa ingeniería multimedia Santiago de Cali 2019. (n.d.).

Edu.Co. Retrieved March 30, 2023, from

<https://red.uao.edu.co/bitstream/handle/10614/11290/T08676.pdf?sequence=5&isAllowed=y>

Agiliza tu desarrollo en Nodejs con el ORM sequelize. (n.d.). ..Ed.Team. Retrieved April 10,

2023, from <https://ed.team/blog/agiliza-tu-desarrollo-en-nodejs-con-el-orm-sequelize>

Model instances. (n.d.). Sequelize.org. Retrieved April 11, 2023, from

<https://sequelize.org/docs/v6/core-concepts/model-instances/>

Model basics. (n.d.). Sequelize.org. Retrieved April 11, 2023, from

<https://sequelize.org/docs/v6/core-concepts/model-basics/>

Data Transfer Object (DTO) – Patrón de diseño. (2018, November 30). Oscar Blancarte -

Software Architecture. <https://www.oscarblancarteblog.com/2018/11/30/data-transfer-object-dto-patron-diseno/>

APÉNDICE

Anexo A

En el siguiente anexo se dejará una ruta para visualizar de una manera más clara y concisa el diagrama de bases de datos del proyecto. [Imagen base de datos proyecto de programas de alimentación para el sector público y privado](#)

Anexo B

En el siguiente anexo se dejará una ruta al repositorio del proyecto el cual contiene tanto los registros de la base de datos, como el backend y el frontend que se desarrolló durante todo el proyecto. [Repositorio en GitHub con el contenido del proyecto](#)

Anexo C

En el siguiente anexo se encuentra una gráfica adicional de las pruebas de estrés que fueron realizadas en Jmeter.

La siguiente gráfica hace referencia a los hilos o peticiones activas con respecto al tiempo, mostrando que siempre se mantiene de manera constante sin presentar errores en un contador igual a uno.

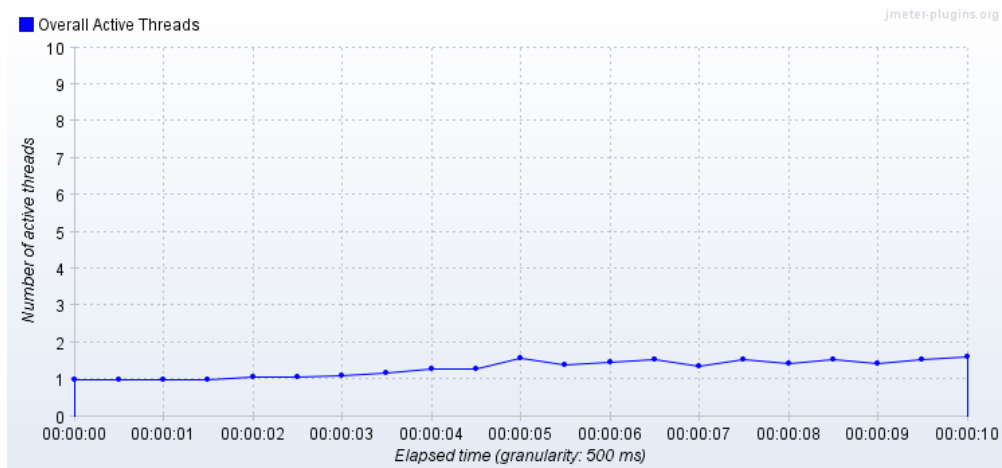


Figura 55 Gráfica de número de peticiones activas por tiempo para prueba de estrés en Jmeter

Anexo D

En el siguiente anexo se encuentra una gráfica adicional de las pruebas de Estabilidad que fueron realizadas en Jmeter.

La siguiente gráfica hace referencia a los hilos o peticiones activas con respecto al tiempo. En este caso como el entorno de ejecución se corre dentro de un bucle, las peticiones aumentan los primeros segundos debido a la latencia, para luego estabilizarse y finalmente caer debido a que la ejecución se detuvo de manera forzosa, y algunas peticiones no fueron enviadas lo que por ende da a entender que genera errores.

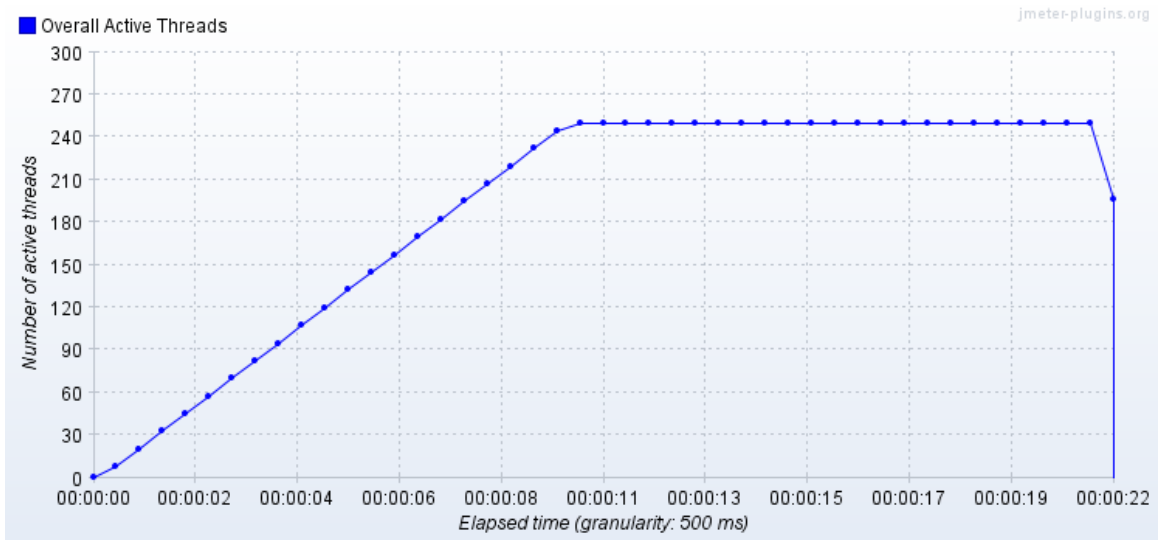


Figura 56 Gráfica de número de peticiones activas por tiempo para prueba de estabilidad en Jmeter