

**Plataforma para gestión logística de
productos para comedores comunitarios**

Jhonatan Steven González
Juan David Cortés

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
BOGOTÁ, D.C.
2020

Plataforma para gestión logística de productos para comedores comunitarios

Autor:

Jhonatan Steven González
Juan David Cortés

MEMORIA DEL TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO
DE LOS REQUISITOS PARA OPTAR AL TÍTULO DE
MAGÍSTER EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

Directora

Angela Carrillo Ramos

Comité de Evaluación del Trabajo de Grado

Clara Mabel Solano

Diana Lancheros

Página web del Trabajo de Grado

<https://livejaverianaedu.sharepoint.com/sites/Ingsis/TGMISC/201003/>

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERIA DE SISTEMAS Y COMPUTACIÓN
BOGOTÁ, D.C.
Mayo,2020

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

Rector Magnífico

Jorge Humberto Peláez, S.J.

Decano Facultad de Ingeniería

Ingeniero Lope Hugo Barrero Solano

Director Maestría en Ingeniería de Sistemas y Computación

Ingeniera Angela Carrillo Ramos

Director Departamento de Ingeniería de Sistemas

Ingeniero Efraín Ortíz Pabón

Artículo 23 de la Resolución No. 1 de junio de 1946

“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia”

AGRADECIMIENTOS

Jhonatan Gonzales

En memoria a mi padre Edgar González. Agradezco a Juan Cortes por su apoyo en este proceso de tesis y su desarrollo de la misma, a Fanny Peña, amigos y familia por la ayuda y estar presente en mi proceso de maestría; también a nuestra directora de tesis Ángela Carrillo por estar tan pendiente de mi salud y de nuestro proceso de tesis, a las personas de Ingeniería Industrial que nos ofrecieron su ayuda en el desarrollo de la misma y por último a la maestría en ingeniería de sistemas de la Pontificia Universidad Javeriana por lo conocimientos para desarrollar esta tesis.

Juan Cortés

Agradezco a Jhonatan Gonzales por el apoyo recibido durante este trabajo de grado conociendo la situación en la que se encuentra, pues siempre representó un apoyo. Agradezco a mis padres Juan Carlos Cortés y a Margarita Olaya porque sin el apoyo de ellos no habría podido realizar este proyecto de tesis debido a su persistencia e insistencia soy el hombre que soy hoy en día. Agradezco a la directora Angela Carrillo por su ayuda y dirección en la presentación de esta tesis y por su amistad a lo largo de estos 8 años de estudio en la Universidad Javeriana pues me permitió consolidarme como el profesional que soy. Finalmente agradezco a toda la comunidad educativa de la Universidad Javeriana por permitirme formarme a través de conocimientos importantes en el área de informática que se verán a lo largo de este trabajo de grado

ABSTRACT

Currently in Colombia the fulfillment of the right to food of unfavorable children is not guaranteed, this is visible both in the number of community canteens that have been closed, as well as in the volume of food that is discarded annually. Therefore, the solution proposed in this report consists of an information system that allows humanitarian organizations, canteens and transport companies to propose strategies to combat the problem of malnutrition and hunger in children; This system allows the storage of stocks, the control of products and suppliers as well as the order of food to supply requests, they can modify the construction of the same according to the need for a usable and understandable interface for multiple users.

RESUMEN

Actualmente en Colombia no se garantiza el cumplimiento del derecho a la alimentación de niños en situación desfavorable, esto es visible tanto en la cantidad de comedores comunitarios existentes que ya han sido cerrados, así como en el volumen de alimentos que se desechan anualmente. Por ello, la solución planteada en esta memoria consiste en un sistema de información que permita a organizaciones humanitarias, comedores y empresas transportadoras plantear estrategias para combatir la problemática de la desnutrición y el hambre en niños; este sistema permite el almacenamiento de existencias, el control de productos y proveedores así como el pedido de alimentos para suplir solicitudes, permitiendo la construcción de las mismas según la necesidad a partir de una interfaz usable y entendible para múltiples usuarios.

RESUMEN EJECUTIVO

En los últimos años se ha visto la falta de incumplimiento en cuanto al derecho a la alimentación al interior de Colombia; actualmente, existe gran cantidad de desperdicio de alimento y falta de posibilidades de distribución de éste hacia las poblaciones con menos recursos, el hambre es un tema presente y real que se vive día a día como una problemática invisible ante los ojos de la sociedad [22] [23].

Por ende, el motivo de este proyecto de grado es dar una solución viable y óptima respecto a los costos y ante el grado de desperdicio que se vive día a día en la sociedad, esto permitiendo apoyarnos en las ventajas que los sistemas de información ofrecen a través de la automatización y la mejora en cuanto a velocidad de procesos de ejecución que permite monitorear y facilitar la ejecución de tareas.

Es por lo anterior que en conjunto con el departamento de ingeniería industrial se propone un sistema que permita el control de las variables necesarias dentro del proceso de ejecución de la cadena de distribución alimenticia; dicho sistema, a través de la integración de información de transporte (vehículos, características de estos y empresas), alimentos (precios y disponibilidad) así como distribución, permitirá a los comedores comunitarios realizar pedidos de productos para satisfacer las necesidades alimenticias de las poblaciones en situación de pobreza, esto basándose en un sistema de recomendación que será alimentado por un modelo de optimización del vecino más cercano en cuanto a la distribución de los insumos alimenticios; de igual manera, permitirá a las partes interesadas de World Vision y el Banco Mundial de Alimentos llevar un control acerca de procesos y ejecución de tareas basados en los pedidos realizados [20][21].

El sistema propone, de igual manera, un sistema de recetas que permite almacenar agrupamiento de ingredientes en la base de datos para la creación de platos específicos y facilitar el pedido de los comedores de una manera más inmediata; estas recetas pueden ser modificadas según la necesidad a nivel individual de los comedores; de igual manera, los pedidos consideran el número de niños que están inscritos a cada comedor comunitario, lo que facilita determinar los requerimientos de cada comedor y la elaboración de cada almuerzo supliendo de esta forma la necesidad alimentaria de cada persona.

En cuanto a tecnologías se empleó .NET, más concretamente el lenguaje de C#, ya que permite una ejecución más rápida a nivel de sistema operativo y los programadores tienen más experiencia con el mismo; este lenguaje fue el que se usó para la implementación del backend, en conjunto con servicios Rest y seguridad de JWT para el manejo de peticiones; en cuanto a la base de datos se empleó el motor de SQL Server ya que permite gran compenetración con el lenguaje seleccionado y la conexión es más adaptable empleando Entity Framework para el desacoplamiento de dependencias en entidades tanto en código como en el modelo entidad-relación. Por otro lado, en la parte del FrontEnd se realizó el desarrollo usando el framework de angular, implementando de manera modular cada uno de los servicios a través de empaquetados en funcionalidades para separar la lógica de llamados a servicios con la lógica de negocio. La elección de este framework se debe a que permite la mejora en cuanto a usabilidad, facilitando el desarrollo con respecto a funcionalidad, diseño y experiencia del usuario.

Para la implementación del sistema se construyó a través de una arquitectura Multinivel ya que permite desacoplamiento de la visualización del sistema con la lógica de negocio y el acceso a datos; esto permite que cada parte sea modular y se desarrolle de manera paralela sin afectarse entre ellas.

De igual manera, dentro de la parte lógica, se desarrolló la separación del negocio usando una arquitectura de capas ya que permite un mayor nivel de abstracción y división de tareas, así como una ejecución más segura en cuanto a la transmisión de los datos y separación de lógica de procesos en cada una de las capas. Estas capas que se desarrollaron fueron las siguientes:

- Capa de acceso: que permitirá el ingreso a la lógica de acceso a partir de firmas en métodos; esta capa está compuesta de cada uno de los controladores del sistema distribuidos a modo de entidades.
- Capa de Lógica de negocio (Business Logic): que permitirá el desarrollo de las funcionalidades y procesos de cada entidad según la petición enviada por el usuario en la vista.
- Capa de acceso a datos: que permitirá la conexión con la base de datos y el tratamiento de las entidades a modo de registros según las acciones del usuario.

Finalmente, como conclusión se aprecia que la solución planteada por el equipo de trabajo consigue solucionar temas de eficiencia en cuanto a costos de transporte y de producto, de distancia y de existencias permitiendo a World Vision y al Banco Mundial de Alimentos realizar pedidos disminuyendo costos y centralizando la información de distintos proveedores en un único punto de acceso disminuyendo el esfuerzo humano para realizar el proceso del negocio, así mismo se plantea dicho sistema para que tanto el comedor como cada uno de los proveedores puedan comunicarse por un mismo medio, permitiendo aceptar o rechazar pedidos y que la información de dicha respuesta sea visible inmediatamente a los comedores agilizando los procesos logísticos.

INTRODUCCIÓN

Actualmente en Colombia no se garantiza el cumplimiento de los derechos sociales con respecto a alimentación de la población; esto se evidencia si se considera que el país cuenta con una población de 48.258.494 habitantes, de los cuales el 4.8% sufre de hambre, es decir, 2.4 millones de colombianos se encuentran en esta condición (FAO, 2019; DANE, 2018). A pesar de estas cifras de forma contradictoria, se observa que diariamente los comedores comunitarios, entidades cuyo objetivo principal es mitigar este problema, son cerrados por la falta de apoyo del gobierno nacional (Personería de Bogotá, 2014). De acuerdo a una entrevista realizada a la señora Clara Albarracín, directora de la Fundación Manos Creadoras (que trabaja con la Fundación América Solidaria y con la Pontificia Universidad Javeriana, en el marco de Proyecto Social Universitario de Ingeniería Industrial, por la estudiante de ingeniería industrial Laura Gisselle Rodríguez Martínez en agosto, 2019), ella establece que recibió el apoyo de donaciones internacionales hasta el gobierno distrital de Gustavo Petro, momento en el cual, este alcalde discontinuó los programas de donación por parte de entidades internacionales.

En el año 2018 fueron cerrados 110 comedores comunitarios, dejando de atender a 28.000 personas en situación de pobreza, siendo los más afectados los niños en cuanto a factor nutricional. En total se atienden sólo 35.700 personas de un total de 7.181.000 que se encuentran en estado de inseguridad alimentaria.

En contraste, se evidencia otra situación crítica: alto volumen de desperdicios y pérdidas de alimentos por parte de los restaurantes. El desperdicio de alimentos (FAO, 2011) se evidencia en su disminución durante los procesos de distribución, *retail* (también venta al detalle o Comercio minorista) y consumo. Adicionalmente, se observa un alto nivel de alimentos desperdiciados y perdidos por parte de los restaurantes; solamente en 2010 se perdieron en el país 1.426.932 toneladas de frutas y verduras en la etapa de post-cosecha, siendo esta cifra el equivalente al 39 % de la oferta total de frutas y verduras de este año. Lo anterior se genera de comportamientos, hábitos de compra, consumo, y la manipulación de alimentos (FAO, 2011). Por otro lado, las pérdidas se observan en la disminución de la masa de alimentos disponibles para consumo humano en las fases de producción agropecuaria, postcosecha, almacenamiento y procesamiento industrial y evidencian ineficiencias en las cadenas de producción. De lo anterior, se puede constatar que en Colombia se pierde un total de 9.76 millones de toneladas de alimentos al año, evidenciando que esta suma podría alimentar cerca de 8 millones de personas en un año.

Actualmente existen organizaciones sin ánimo de lucro, como el Banco Mundial de Alimentos y Visión Mundial, que reciben alimentos, así como productos de aseo, higiene personal y protección femenina, bienes y servicios, los cuales los distribuye a organizaciones sin ánimo de lucro que atienden población vulnerable, con el apoyo de la academia, el sector privado y público. (Banco de Alimentos 2018).

Teniendo en cuenta lo anterior, el motivo del presente trabajo es dar soporte y brindar información acerca de los procesos efectuados por dichas entidades a través de una plataforma informática, que permite:

- Presentar información sobre los productos que serán entregados a cada uno de los comedores comunitarios.
- Presentar mejores proveedores tanto alimenticios como de transporte
- Coordinar entrega de alimentos con respecto a los menús nutricionales que ofrezca cada comedor comunitario.
- Manejo de proveedores: su información de base, así como los productos que venden, su inventario y notificaciones o alertas de vencimiento.

1. DESCRIPCIÓN GENERAL

1.1 Oportunidad y problemática

Para dar una solución al Banco Mundial de Alimentos y Visión Mundial, sobre su gestión logística, que consiste en la gobernanza de las funciones de la cadena de suministro, incluyendo actividades como gestión de transporte, gestión de flotas, el almacenamiento, la manipulación de materiales, la gestión de inventario y la gestión de proveedores de logística externos, se desarrolló una plataforma informática que a través de la información ingresada y el contexto de movilidad presenta recomendaciones sobre decisiones relacionadas con la disminución de costos de logística, mejores proveedores, gestión de productos, inventarios y proveedores. Todo lo anterior, la plataforma desarrollada apoya la labor de suplir las necesidades nutricionales que estas organizaciones sin ánimo de lucro y comedores comunitarios en comunidades de escasos recursos. [22]

1.2 Formulación del problema que se resolvió

La gran problemática que se atendió con la plataforma es la de apoyar a las entidades que se dedican a mitigar el hambre y la desnutrición de los niños de escasos recursos que se encuentran en condición de hambre y pobreza, y también a disminuir la gran cantidad de alimentos que se desperdician año tras año por las grandes empresas distribuidoras de alimentos, permitiendo tratar dos problemáticas de manera simultánea y correctiva en conjunto. Es así que la plataforma responde a la pregunta de investigación planteada en el proyecto así:

¿Cómo con una plataforma TIC se puede contribuir a disminuir el déficit alimenticio en las poblaciones de bajos recursos y evitar que tantos alimentos de empresas se desperdicien año tras año?

1.3 Justificación de la problemática

Para responder la pregunta anterior surge como respuesta la necesidad de crear una plataforma informática que contemple las necesidades de los comedores comunitarios, donde se distribuyen actualmente los alimentos y los proveedores alimenticios que son aquellos que se ven en la necesidad de desechar alimentos, y así apoyar el proceso logístico de distribuir los alimentos

que requiere esta población y que las partes interesadas de Visión Mundial (VM) tengan una herramienta que les facilite gestionar este proceso de manera integral[22].

Se diseña un modelo de adaptación que plantea adaptarse al proceso actual, permitiendo al interesado (VM) controlar vehículos, productos, proveedores y comedores, permitiendo reflejar a través de un aplicativo el proceso en general que se ejecuta normalmente pero de una manera más eficiente permitiendo evidenciar las variables a un nivel global y que de esta manera los procesos de envíos sean menos costosos para VM, permitiendo destinar dicho dinero a la mejora de otros aspectos operativos del proceso. De igual manera los comedores comunitarios podrán realizar pedidos directamente teniendo en cuenta cálculos de precios en productos, cantidad de personas dentro del comedor comunitario, cantidades para recetas y gestión de conocimiento en cuanto al proceso de pedidos. De esta manera se plantea una herramienta de logística que permita mejorar la toma de decisiones y contribuir a la disminución de hambre y desperdicios de comida en el país. [23] [24]

Se tiene intención que la herramienta sea un apoyo para las partes interesadas sin embargo se entiende que posteriormente se pueden presentar cambios o mejoras en cuanto a la plataforma y por ende se dejara la misma a disposición de las partes interesadas con el objetivo que en el futuro dicha herramienta permita mejorar aún más los procesos partiendo de una herramienta y una premisa base.

1.4 Impacto Esperado

La solución propuesta se enfoca en la disminución y mitigación de los problemas de hambre en los sectores donde tenga control VM, así como en la prevención de desperdicios de alimentos por parte de los proveedores de alimentos, que pueden ayudar a personas de escasos recursos mediante su donación o venta a precios bajos para los comedores.

Para lograr los objetivos planteados se desarrollaron tres (3) elementos importantes:

El primer elemento corresponde a un modelo de adaptación que permite a los usuarios utilizar la herramienta para solucionar la problemática que ocurre en su contexto, con lo anterior se destaca que la herramienta apoya los procesos de los comedores comunitarios, sin alterarlos. Lo anterior en conjunto con una interfaz usable, sencilla de entender y que ejecuta las tareas core del negocio permite mejorar el flujo del proceso y ofrece ventajas a los niños y a los proveedores automatizando tareas y permitiendo una visibilidad global de las variables que determinan los costos, mejorando la calidad y velocidad de la ejecución del proceso logístico.

El segundo elemento es el empleo de un modelo de optimización, planteado en conjunto por el equipo de Ingeniería Industrial que permite ofrecer a través de la interfaz de usuario los mejores proveedores y productos para la ejecución del proceso. Este modelo considera las siguientes variables: Precio de productos, Costos de envío, distancia de comedores, cantidad de productos. A través del análisis de estas variables en un modelo Simplex se presentan las variables que contemplan la mejor ejecución posible del proceso, automatizando los pedidos manuales y optimizando los tiempos de las operaciones de logística.

El último elemento es el desarrollo de la documentación de los aspectos del desarrollo, manuales y características del proceso de implementación para las partes interesadas (VM), para que en un futuro se puedan mejorar características del software y aumentar funcionalidades y servicios de este, esto con el objetivo de que se vuelva una herramienta útil para la resolución de la problemática y que ayude a disminuir los índices de hambre y desperdicio en Bogotá.

Este trabajo de grado se constituye en una herramienta que permite mejorar los procesos de la cadena de distribución considerando sus costos y logística, para facilitar a VM, tomar decisiones que finalmente ayuden a mejorar la calidad de vida de las personas más desfavorecidas.

2. DESCRIPCIÓN DEL PROYECTO

2.1. Objetivo general

Diseñar una plataforma informática que cubra la gestión logística de proveedores, productos y rutas de transporte de alimentos, para comedores comunitarios aledaños a Bogotá, como apoyo al banco mundial de alimentos y visión mundial.

2.2 Objetivos específicos

1. Contextualizar la logística de negocio según información suministrada
2. Definir el modelo y proceso de adaptación que será utilizado en la construcción de la aplicación.
3. Analizar e integrar los procesos de gestión logística y el modelo de adaptación para orquestar las funcionalidades de la plataforma informática.
4. Diseñar una plataforma informática que integre el modelo de adaptación y los procesos de gestión.
5. Probar la plataforma a través de un prototipo funcional, que será evaluado por los usuarios funcionales finales

2.3 Fases de desarrollo

En esta sección se presentan las cinco (5) fases con las cuales se desarrolló el proyecto.

2.3.1 Contextualización de problemática

Identificación del problema y de las características necesarias a considerar, comprendiendo la problemática y factores del riesgo dentro del mismo.

Actividades

- Contextualización documentos levantados por el equipo de Ingeniería Industrial.
- Recepción de modelo de optimización.

Resultados esperados

- Documento de contextualización a cerca del contexto y la problemática.

2.3.2 Proceso de modelado de adaptación

Definir el modelo y proceso de adaptación a considerar en la construcción de la aplicación.

Actividades

- Recepción de encuestas de parte del equipo de Ingeniería Industrial.
- Mockups para pre-visualización y guía del modelo final del sistema.

Resultados Esperados

- Modelado de perfiles y contexto.
- Estrategias de adaptación según el contexto.

2.3.3 Proceso de modelado logístico

Analizar e integrar los procesos de gestión logística y el modelo de adaptación para orquestar las funcionalidades de la plataforma informática.

Actividades

- Análisis y cotejamiento de procesos de logística vs posibles soluciones.
- Revisión modelos y procesos.

Resultados Esperados

- Seudocódigo de modelos de optimización planteados por el equipo de Ingeniería Industrial.
- Seudocódigo de implementación con las integraciones de proveedores.

2.3.4 Implementación del Sistema

Diseñar una plataforma informática que integre el modelo de adaptación y los procesos de gestión.

Actividades

- Refinamiento de requerimientos.
- Implementación del proceso.
- Desarrollo de métricas para eficiencia del sistema.
- Desarrollo pruebas internas de equipo de desarrollo.
- Entregas periódicas de proceso de implementación actual.

Resultados esperados:

- Algoritmos adaptativos obtenidos a partir de la recopilación de datos de los modelos de optimización.
- Herramienta informática con funcionalidades unificadas de modelo de adaptación y de modelo de optimización.
- Documento SRS formulado a partir de los requerimientos obtenidos.
- Documento de arquitectura.

2.3.5 Pruebas

Probar la plataforma a través de un prototipo funcional, que será evaluado por los usuarios funcionales finales.

Actividades

- Desarrollo de pruebas en vivo.
- Anotaciones y observaciones por parte del usuario final.

Resultados esperados:

- Documento obtenido a partir de las pruebas realizadas en vivo.
- Documento de análisis de funcionamiento de arquitectura en cuanto a atributos de calidad.

3. MARCO TEÓRICO / ESTADO DEL ARTE

3.1 Marco Conceptual

En Colombia se presentan problemas relacionados con la distribución alimenticia en comedores comunitarios y el desperdicio de productos, esto se ve reflejado en las altas tasas de pérdida y desecho de alimentos [1] y en las tasas de desnutrición infantil [2], esto en gran medida a que actualmente no se posee un mecanismo de control de vencimiento o de baja tasa de demanda de estos.

Teniendo en cuenta el anterior contexto a continuación se presentan conceptos clave tales como: cadena de distribución, modelos de optimización, tecnologías de información y sistemas de información.

3.2 Conceptos básicos

Con la finalidad de comprender mejor el contexto y funcionalidad del sistema, se exponen conceptos y definiciones, relacionados con el mismo. De igual manera se explican conceptos de adaptación para evidenciar como los mismos afectan la usabilidad del sistema.

- Cadena de distribución: “La cadena de distribución es el conjunto de actuaciones que llevamos a cabo desde que terminamos la fabricación de un bien, hasta que dicho bien se adquiere por el consumidor final.” [3]

- Modelos: “Un modelo es la representación abstracta de un proceso en particular o de la realidad en general.” [4]
- Modelo de Optimización: “Un modelo de optimización es la representación matemática de un problema real en el cual uno conoce el impacto de cada una de las variables y uno busca encontrar el mínimo valor (o máximo) posible de una función objetivo: costo, ventas, nivel de servicio, entre otros.” [5]
- Conservación de alimentos:” La Conservación de alimentos es un conjunto de procedimientos y recursos para preparar y envasar los productos alimenticios con el fin de guardarlos y consumirlos mucho tiempo después.” [6]
- Hambre: “carencia de alimentos e inseguridad ali-mentaria que afecta a todos los grupos de población. La desnutrición es el resultado fisiológico del hambre y/o la enfermedad y se manifiesta en un amplio déficit de macro y micronutrientes.” [7]
- Sistema de información:” Es un conjunto ordenado de mecanismos que tienen como fin la administración de datos y de información, de manera que puedan ser recuperados y procesados fácil y rápidamente.[8]
- Arquitectura de software:” se refiere a la estructuración del sistema que, idealmente, se crea en etapas tempranas del desarrollo. Esta estructuración representa un diseño de alto nivel del sistema que tiene dos propósitos primarios: satisfacer los atributos de calidad (desempeño, seguridad, modificable), y servir como guía en el desarrollo.” [9]
- Backend: “Backend es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos. El Backend también accede al servidor, que es una aplicación especializada que entiende la forma como el navegador solicita cosas.” [10]
- FrontEnd: “Frontend es la parte de un programa o dispositivo a la que un usuario puede acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.” [10]

A continuación (imagen 1), se presentan los conceptos y sus relaciones según su importancia en el trabajo de grado, a través de un mapa conceptual.

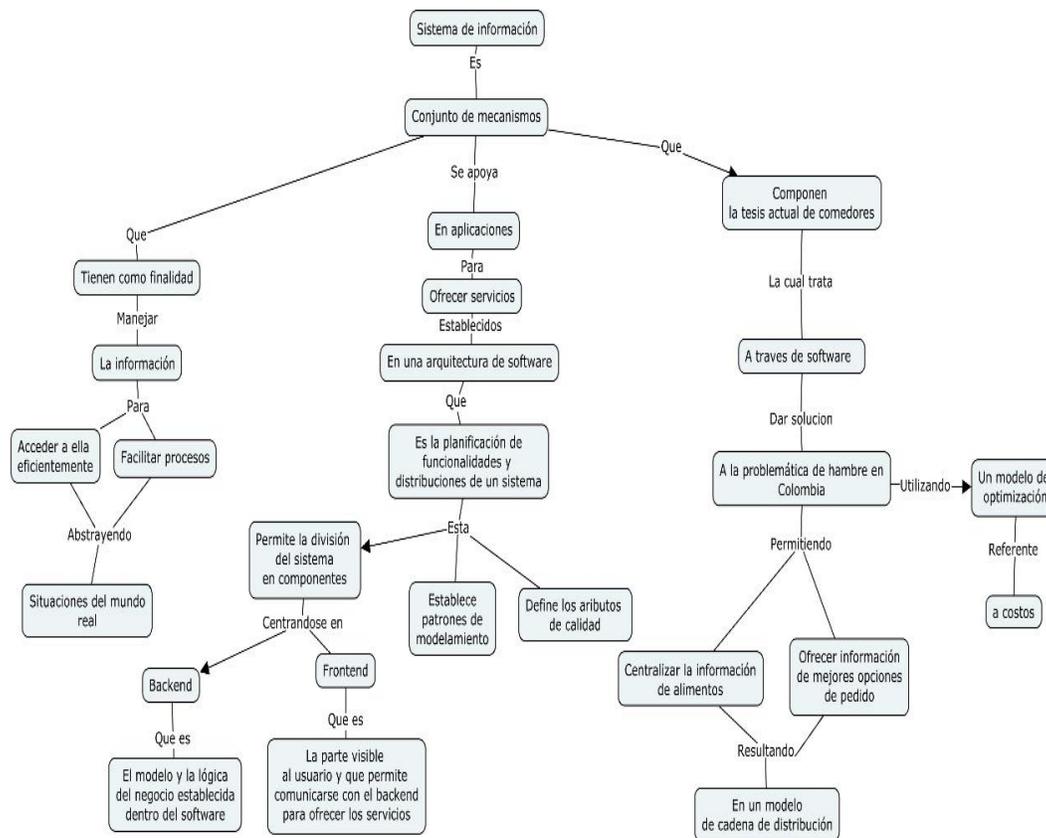


Imagen 1 Mapa conceptual

3.3 Adaptación

Según el contexto expuesto anteriormente, se debe tener en cuenta la construcción de un sistema que se rija según la situación actual de la problemática y que de igual manera provea las funcionalidades específicas a cada uno de los usuarios del sistema.

Para esta última parte se planteó el perfilamiento de los usuarios según sus funciones específicas y se identificaron los siguientes casos:

- Proveedor de alimentos: usuario que introdujera la cantidad de existencias del producto que tenga disponible y que permitirá la selección de estos productos por parte de los comedores comunitarios.
- Proveedor de transporte: usuario que podrá insertar la existencia de vehículos de transporte así como sus características al sistema.
- Comedor: usuario principal del sistema que podrá realizar pedidos; el sistema ofrecerá sugerencias para el pedido teniendo en cuenta el proveedor de alimentos y el proveedor de transporte.
- Administrador: podrá apreciar la totalidad de funcionalidad del sistema, teniendo acceso a la información completa del mismo.

El sistema a su vez tendrá presente el algoritmo de adaptación que permitirá, dados los atributos y características de cada proveedor, ofrecer las opciones de pedido menos costosas para el comedor teniendo en cuenta el costo de envío y entrega de los productos; sin embargo, el mismo usuario podrá seleccionar el proveedor que se recomiende o seleccionar uno de su preferencia.

Un aspecto clave para el sistema será la posibilidad de estandarizar pedidos o “recetas”, siendo éstas introducidas a la base de datos y será posible precargarla para que el domicilio se estandarice; estos pedidos también serán personalizables según la necesidad del comedor y podrán ser modificados de manera manual o de manera automática mediante el consumo de un archivo de Excel con la información.

4. TRABAJOS RELACIONADOS

Teniendo en cuenta el contexto de la plataforma y los objetivos que se quieren conseguir para dar solución a la problemática, se investigaron aplicativos y herramientas similares que suplieran de manera parcial o parecida la necesidad objetivo del nicho objetivo de este trabajo de grado, según su índole se dividieron los trabajos relacionados según sus funcionalidades, teniendo en cuenta esto se categorizaron como se describe a continuación:

- Aplicaciones de control de inventario: son aquellas aplicaciones que permiten el control de productos y sus existencias a manera de bodega virtual, son útiles para controlar flujos y cadenas de distribución según producto que se necesite por el cliente o sencillamente para almacenaje.
- Aplicaciones de domicilio: aplicativos cuya función radica en permitir al usuario realizar pedidos de productos a proveedores remotos.
- Aplicaciones de recetas: aplicativos que tienen como funcionalidad ayudar al usuario final a preparar platillos y guiarlos durante dicho proceso teniendo en cuenta los insumos

De acuerdo con la clasificación expuesta anteriormente se encontraron los siguientes aplicativos similares:

- Aplicaciones de control de inventario:
 - Real Inventory: aplicación móvil para sistema operativo iOS, que permite la sincronización con *cashier live*, un programa diseñado para la gestión de inventarios que permite recuento de elementos en almacén, control de existencias y características de éste (hora de recepción, descripción, tipo...) es un software que cuenta con actualizaciones constantes y posee una arquitectura muy sólida y se ejecuta en tiempo real; sin embargo, no cuenta con modalidad gratuita y se requiere pago mensual para aprovechar los beneficios la cuenta para su uso requiere pago, de igual manera la restricción que solamente funciona con IOS limita los usuarios que pueden acceder a la misma.[11]
 - ABC Inventory: es un aplicativo sencillo de control de inventarios cuya principal fortaleza radica en que posee gran usabilidad, permitiendo a casi cualquier usuario manipularla sin mayores complicaciones, permite dividir y personalizar las entidades dentro del sistema según las necesidades del usuario, esto permite controlar imágenes, modificar logos, controlar cantidades y precios y gestionar proveedores, sin embargo está

- diseñada para pequeñas empresas y esto provoca que el espacio de almacenamiento en nube no sea muy amplio, sin embargo solamente se puede usar la licencia en un computador limitando el acceso a solamente un punto[12]
- PartKeeper: aplicativo que permite el manejo de inventario de componentes electrónicos, permite el empaquetamiento y control de características de productos a partir de las necesidades del usuario, categorizando los componentes, manejo de proveedores y reconocimiento de usuarios según perfil dentro del aplicativo, de igual manera se puede conseguir de manera gratuita; no obstante, esa limitado únicamente al sector de la electrónica, el usuario debe poseer una base de datos en MySQL ya que no tiene persistencia propia y debe ser vinculada con un manejador a la misma[13]
 - Openbravo: es una solución poderosa que permite la actualización en tiempo real de movimientos con altos volúmenes de productos, el aplicativo permite desde el manejo de un producto en específico hasta el manejo de relaciones con proveedores, se encuentra desplegado en nube y es open source. A pesar de ello, no cuenta con manejo en dispositivos móviles y no tiene control de pedidos a clientes remotos teniendo en cuenta transporte [14]
 - Aplicaciones de domicilios:
 - Grubhub (IOS/Android): aplicativo de pedido de domicilios, cuenta con disponibilidad en IOS y Android, permite identificar los restaurantes que se encuentren más cerca a la ubicación del móvil y el filtrado de platos según las necesidades del usuario, aunque la aplicación es gratuita los restaurantes pueden cobrar tarifas adicionales que no se ven reflejados en el aplicativo y no permite modificación de insumo en platos, es decir, se ofrecen los productos únicamente como están en los restaurantes, actualmente se encuentra suspendida y no ofrece soporte por parte de los desarrolladores [15]
 - Doordash: aplicativo móvil que se encuentra disponible en IOS y Android, este permite puntuar las entregas realizadas a partir de la interfaz por los usuarios, permitiendo tener en cuenta la calidad de los alimentos, la popularidad de los restaurantes y la misma satisfacción del cliente; sin embargo, maneja una tarifa de cobro variable según impuestos o impuestos y no presta servicios fuera de Estados Unidos de igual manera la interfaz es poco amigable y suele prestarse para inconvenientes con los usuarios finales [16]
 - Postmates (iOS/Android): aplicativo disponible en IOS y Android que permite no solamente el pedido y entrega de comida sino también de cualquier producto, esta plataforma está disponible actualmente en más de 90 ciudades en el mundo sin embargo todas ellas se encuentran en Estados Unidos. Sin embargo, se incluye un pago extra en cada orden y durante horas pico suelen aplicarse tarifas dinámicas, provocando que el precio mostrado en la aplicación no sea necesariamente el real[17]
 - Aplicaciones de recetarios:
 - Nestle Cocina: aplicación de recetario disponible en IOS y Android, A través de una interfaz de usuario intuitiva y sencilla permite mostrar más de 1000 recetas de variada índole, permitiendo búsquedas personalizadas según valor nutricional , tiempo de preparación o categorización, incluye de igual manera una lista de compra para gestionar los insumos en la preparación de los alimentos; sin embargo, las recetas ya se encuentran estandarizadas y no pueden ser modificadas por el usuario, no posee forma de

realimentación y no permite ver precios de los insumos necesarios para la preparación de los platillos[18]

- Recetario Villy: aplicación de cocina disponible en IOS y Android con más de 100 recetas para selección, los usuarios pueden subir sus propias recetas a partir del registro con la plataforma, los recetarios se clasifican según su forma de preparación o por características propia de las recetas, de igual manera se puede compartir actividades a través de vinculación con redes sociales; sin embargo, no permite manejo de precios en cuanto a insumos o pedidos de los mismos a corresponsales de proveedores más cercanos al punto de pedido[19]

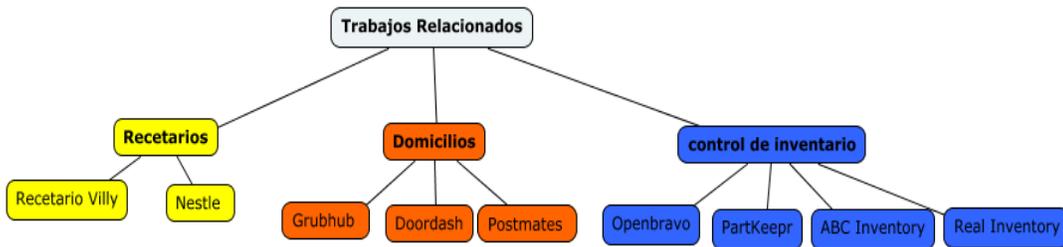


Imagen 2 Trabajos Relacionados

En la tabla 1.0 se aprecian las comparaciones entre cada una de estas aplicaciones teniendo en cuenta los objetivos propuestos y los servicios establecidos anteriormente; cada aplicativo fue separado teniendo en cuenta la distribución y caracterización anteriormente expuesta, en la parte de la cabecera se encuentran los aplicativos similares representados con su cita bibliográfica, en el lateral izquierdo se encuentran los criterios agrupados según sus características. Si el trabajo presenta el criterio, se encuentra en la celda correspondiente una X:

	Característica/aplicativo	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]
Control de inventarios y proveedores	Control de inventarios	X	X	X	X					
	Manejo de proveedores de transporte		X		X					
	Manejo de sugerencias en cuanto a proveedores					X	X	X		
	Sugerencia de preferencias del usuario								X	X
	Manejo de proveedores de alimentos		X		X			X	X	
optimización	Permite cálculo de pedido óptimo									
	Ofrece realimentación de					X	X			

	los domici- lios									
	Cálculos au- tomatizados según canti- dad de perso- nas									
	Notificación de costos	X	X	X	X	X	X	X		
Interfaz	Manejo de múltiple idioma				X				X	
	Orientado a personas de cualquier edad(usabili- dad)		X					X	X	
	Personaliza- ción de inter- faz según perfil			X					X	X
Recetas	Control de recetarios								X	X
	Inserción de modificacio- nes a plantilla de recetarios								X	
	Extracción masiva de da- tos de recetas a fuentes ex- ternas									
tecnolo- gía	Uso de tecno- logías de in- formación	X	X	X	X	X	X	X	X	X
	acceso gratis	X			X			X	X	X
	Acceso en móviles	X	X		X	X	X	X	X	X
	Servicio en nube	X		X		X		X	X	X

Tabla 1 Trabajos relacionados

Como se puede apreciar en la tabla 1.0, no existe un aplicativo que tenga en cuenta todos los criterios integrados necesarios para dar solución a la problemática planteada.

A nivel de control de inventarios y proveedores existen solamente dos aplicaciones que permiten control tanto de proveedores de alimentos como de transporte y de manejo de inventario en bodega. Las sugerencias en cuanto a cuál puede ser el proveedor óptimo para un pedido no se encuentra en ningún aplicativo provocando que el usuario sea el que tenga que realizar el pedido de manera manual y sin tener en cuenta aspectos de eficiencia en cuanto a variables. De igual manera, se aprecia que dos aplicativos son los únicos que tienen en cuenta las preferencias del usuario para iniciar la solicitud de pedido.

En cuanto al segundo grupo “optimización” se aprecia que es donde más falencias tienen los aplicativos demostrando que solamente hay dos de ellos que cumplen con el criterio de ofrecer realimentación acerca de los domicilios pedidos por el usuario y la mayoría cumplen con notificar costos de productos al usuario; sin embargo, al mirar los otros criterios como automatización de selección en cuanto al producto óptimo o cálculo automático del número de productos a pedir según cantidad de usuarios no se poseen, provocando que el cálculo del mejor pedido sea gestionado por el usuario, esto en el contexto de un comedor comunitario o incluso teniendo en cuenta a World Vision se vuelve muy difícil de controlar ya que ellos tienen que agilizar cálculos para permitir cubrir la mayor cantidad de pedidos para los niños de los comedores comunitarios.

Se aprecia que en cuanto a usabilidad (interfaz) los aplicativos manejan interfaces muy poco usables con usuarios de cualquier edad provocando que el nicho de mercado sea más reducido debido a que se necesitan más conocimientos técnicos para el uso adecuado de las mismas.

Prosiguiendo con las recetas, solamente los dos últimos aplicativos permiten el control de ingredientes y formulación como tal de una receta. Sin embargo, ninguno de los aplicativos posee modificación en cuanto a las recetas ofertadas ofreciendo únicamente instrucciones a manera de guía y pasos a seguir, así mismo estas recetas están ligadas a una cantidad de personas delimitando los ingredientes a cumplir con cierto número de porciones. Sería recomendable que para los usuarios del sistema se encuentren recetas a modo de pedidos pre cargados en el aplicativo con el fin de agilizar los domicilios y dar mantenimiento a los mismos a través de modificaciones por interfaz, de igual manera, cómo controlar para cuántas personas en total va a ser elaborada la receta.

Finalmente se observa que no existen aplicativos que manejen de manera integral los criterios necesarios para realizar efectivamente las tareas de los comedores y la administración de Visión Mundial (VM) y evitar con ello la problemática central de este trabajo de grado: el hambre y el desperdicio de alimentos. Por todo lo anterior, se presenta a la solución informática trabajada durante este trabajo de grado denominada “Comedores”, el cual se constituye en un aplicativo que se centra en la corrección de la problemática.

“Comedores” ofrece una aplicación que busca minimizar costos para los comedores y aumentar la velocidad de entrega de los pedidos, esto es permitido a partir del cálculo de un modelo de programación lineal (Simplex LP) que considera variables como la fecha de caducidad, costo de producto, disponibilidad y distancia entre proveedores y comedores según cada pedido. “Comedores” busca mejorar las decisiones logísticas para el cumplimiento de los pedidos permitiendo cumplir cada pedido de forma eficiente en cuanto a costos y distancia, para todo tipo de usuarios con un menú sencillo para acceder a cualquier parte de la misma, ofreciendo ingreso rápido a temas de control de existencias de los proveedores de alimentos, control de vehículos según características para el transporte de los productos, realización de prepedidos para agilizar las peticiones tanto de comedores como de VM, ejecutando de una manera rápida y precisa la gestión de alimentos ya sea de donación o por costos mínimos optimizados por el mismo sistema; dichos costos se obtienen de las variables de fecha de caducidad, distancia, y costo de productos permitiendo presentar a partir de una interfaz sencilla y entendible las mejores opciones al usuario.

De igual manera, “Comedores” ofrece un sistema de optimización basado en un algoritmo de probabilidad de tipo Simplex y en conjunto con una normalización de valores en cuanto a variables de costos que permite ofrecer una opción viable en cuanto a evitar desperdicio de alimentos y mejorar la entrega, considerando tiempos de entrega de pedidos desde un proveedor alimenticio a un comedor.

5. CONTRIBUCIONES

5.0 Fases de desarrollo y descripción de ejecución.

“Comedores” se desarrolló teniendo en cuenta cinco (5) fases, las cuales se pueden observar en el desarrollo de esta sección.

5.1 Contextualización de problemática.

En esta primera fase se identificó la problemática y el contexto del mismo teniendo en cuenta la información suministrada por Visión Mundial, el Banco Mundial de Alimentos y el equipo de trabajo de Ingeniería Industrial; así mismo, se identificó cómo a través del aplicativo se podría solucionar la situación de desperdicio de alimentos y cómo se podría agilizar los procesos de pedidos de VM a proveedores alimentarios y el transporte de los mismos; en última instancia, durante esta etapa, se recibe un algoritmo de optimización teniendo en cuenta los costos para maximizar la eficiencia de la ejecución de los pedidos.

- **5.1.1 Contextualización documentos levantados por el equipo de Ingeniería Industrial.**

A partir de la recopilación de estadísticas y documentos informativos de la situación de hambre y desperdicio a nivel nacional, se recopila información del contexto actual y de cómo se maneja la problemática hoy en día; de igual manera se identifican los actores y cuáles son las funcionalidades de cada uno de ellos, las situaciones del sector, cómo operan los comedores, y la manera en la que se manejan los cargamentos de alimentos. [24]

- **5.1.2 Recepción de modelo de optimización:**

Se recibe el modelo de optimización por parte del equipo de trabajo de Ingeniería Industrial; dicho modelo se desarrolla utilizando método Simplex para hallar la mejor eficiencia en cuanto a proveedores de alimentos. En conjunto con una normalización de datos se procede a estandarizar las medidas y disminuir el margen de error de cálculos; una vez hallada la eficiencia, se procede a calcular el mejor proveedor siguiendo el método de Simplex LP y los criterios de decisión para la resolución de la problemática.

Dichas variables son: el costo del producto, la distancia del comedor a un proveedor de alimento, el costo del transporte y el tipo de transporte.

A continuación, se describe el funcionamiento del método Simplex.[29]

1. Se plantea la función objetivo es decir que función deberá ser minimizada para encontrar la mejor eficiencia: La fórmula en el caso de esta problemática es:

$$((A1x1)^2 + (A2x2)^2 + (A3x3)^2) / NP * 100 = A1E$$

Donde A1 es la representación numérica de las variables de costos, distancia o tipo de transporte, las x serán las variables de optimización del modelo, E es la eficiencia a medir del proveedor actual modelo y NP será el número de proveedores a evaluar.

2. Se plantean las restricciones del modelo donde ningún valor de x puede ser menor a 0 evitando valores negativos dentro de la fórmula:

$$X1 \geq 0;$$

$$X2 \geq 0;$$

$$X3 \geq 0;$$

Y en donde la sumatoria de todas debe dar un resultado =1 para permitir evaluar la probabilidad:

$$X1 + X2 + X3 = 1$$

3. A continuación, se plantea un proveedor ideal para la minimización, en donde todas sus variables de medida serán las mínimas para tener una referencia en cuanto a la minimización de los demás proveedores, es decir:

$$A1 = 1;$$

$$A2 = 1;$$

$$A3 = 1;$$

Con ello se sacará la eficiencia del proveedor ideal y se comparará con los demás para la organización de la lista de los mismos según el cálculo de la fórmula.

4. Se normalizan las cantidades numéricas de las cantidades de los proveedores en cuanto a costo, distancia, costo vehículo y tipo, esto para evitar que haya mucha diferencia entre cálculos de los proveedores permitiendo trabajar con valores relativamente cercanos. La fórmula aplicada en este caso será:

$$\frac{X - \mu}{\sigma}$$

Donde:

X es el valor que se desea normalizar,

Miu es la media aritmética de la distribución

Ro: es la desviación estándar de la distribución

Una vez se tengan obtenidos los valores normalizados se procede a ejecutar el siguiente paso.

5. El algoritmo recibirá a forma de vector las variables numéricas de cada proveedor ubicando los valores en la siguiente matriz:

Se crea la tabla inicial donde se introducen las variables de holgura en un lateral y los coeficientes en el encabezado. La última línea contiene los valores de la función objetivo y los costes reducidos $Z_j - C_j$.

Esta última se calcula:

“ $Z_j = \sum(C_{bi} \cdot P_j)$ para $i = 1..m$, donde si $j = 0$, $P_0 = b_i$ y $C_0 = 0$, y en caso contrario $P_j = a_{ij}$. Aunque al tratarse de la primera tabla del método Simplex y ser todos los C_b nulos se puede simplificar el cálculo, y por esta vez disponer $Z_j = -C_j$.”[25]

Tabla I . Iteración nº 1							
			3	2	0	0	0
Base	C _b	P ₀	P ₁	P ₂	P ₃	P ₄	P ₅
P ₃	0	18	2	1	1	0	0
P ₄	0	42	2	3	0	1	0
P ₅	0	24	3	1	0	0	1
Z		0	-3	-2	0	0	0

Tabla 2 Proceso Simplex

- Se empieza a iterar por la columna donde el valor de la evaluación en z sea la mayor optimizando la eficiencia, siempre se debe tomar el valor menor y negativo de la última hilera, esta columna se denominará columna pivote, una vez tomada la elección se procede a dividir los valores de la columna P₀ entre los valores de la columna pivote, siempre que ambos elementos sean estrictamente positivos, se escoge a continuación la columna cuyo resultado haya sido el mínimo.
- Se continua la iteración de este proceso hasta que no haya valores negativos en la columna Z finalmente se reemplazan los valores obtenidos por cada columna en la función objetivo y se obtiene el valor de la eficiencia para este proveedor.

5.2 Proceso de modelado de adaptación:

En este componente se desarrolla el modelo de adaptación teniendo en cuenta el contexto actual de la problemática y de cada uno de los actores a partir de encuestas y documentación de procesos de tareas entregados por el equipo de Ingeniería Industrial. Se construyen los perfiles y servicios necesarios para la correcta ejecución de la aplicación, considerando el proceso que se realiza para llevar a cabo las entregas de los productos; una vez estandarizado el proceso se plantean las funcionalidades individuales de cada actor, subdividiendo responsabilidades y especificando qué va a poder controlar cada uno dentro del programa y limitando las demás funcionalidades:

- Comedor:** los comedores serán los actores que podrán realizar pedidos, solicitar insumos según preferencias de costo o de proveedor de productos, seleccionar el proveedor para el transporte de los insumos; de igual manera, podrá controlar las recetas propias ingresándolas al sistema, modificándolas en cuanto a insumos o descargándolas en un Excel para importarlas o trabajar sobre ellas más cómodamente
- Cliente Visión Mundial:** Estará en capacidad de realizar pedidos, establecer nuevos vehículos o productos, modificar o crear comedores según características, seleccionar proveedores alimenticios y proveedores de transporte
- Proveedor de alimentos:** son aquellas empresas o industrias que ofrecerán los productos o alimentos a los comedores comunitarios; estos podrán modificar productos en cuanto a

características y cantidades, agregar nuevos alimentos y aceptar o rechazar solicitudes de envío

- **Proveedor de transporte:** son aquellas empresas que participarán en el sistema con sus unidades de vehículos para el transporte de productos alimenticios; estos podrán modificar o crear nuevos vehículos dentro de la plataforma, así como aceptar o rechazar peticiones de envío

En este punto es importante identificar que las variables que permitirán a la aplicación presentar una sugerencia de un proveedor están relacionadas a los costos las siguientes variables:

- Cantidad: cantidad de producto en el proveedor alimenticio.
- Costo: costo de producto en proveedor alimenticio.
- Distancia: distancia del comedor al proveedor alimenticio.
- Transporte: vehículo en el que se transportaran los productos.

Estas variables y los costos se incluyen en la función objetivo que maximiza la eficiencia y disminuye el desperdicio de recursos tanto monetarios como de productos; de igual manera, un punto clave es que los pedidos podrán venir a modo de donación o por compra; esto, sin embargo, se verá reflejado en el algoritmo de optimización: cuando se trate de una donación el tiempo de vencimiento de los productos siempre será de tres (3) días. La función de optimización se resuelve aplicando el método Simplex expuesto anteriormente.

- **5.2.1 Recepción de encuestas por parte del equipo de Ingeniería Industrial:**

A partir de las encuestas y modelos planteados por el equipo de Ingeniería Industrial se procede a construir la diferenciación de perfiles, las funcionalidades clave dentro del sistema, y las tareas a realizar por cada uno de ellos; de igual manera, se recibe información estadística del proceso actual y de los índices de hambre y de desperdicio a nivel nacional.

- **5.2.2 Actores:**

A partir del reconocimiento de las funciones y tareas de los stakeholders dentro del proceso, se proponen cuatro (4) tipos de perfiles, cada uno con sus diferentes tareas y funciones dentro del sistema, los cuales son presentados a continuación:

- Comedor: perfil de los comedores, son aquellos que tienen la necesidad de realizar los pedidos para adquirir los productos según las necesidades de los niños que tienen a cargo; de igual manera, es importante destacar que estos necesitan obtener el mejor beneficio teniendo en cuenta costo de producto y transporte, así como tiempos de entrega; de igual manera, estos deben acceder a la disponibilidad de productos dentro del sistema y de rechazar o aceptar a los proveedores que puedan suplir las necesidades; asimismo, estos perfiles pueden crear y modificar recetas dentro del sistema, estableciendo los insumos y su cantidad para la elaboración, se componen de las siguientes características
 - Dirección
 - Datos de contacto (teléfono, correo)
 - Número de niños
 - Código de identificación único

- Proveedor alimenticio: dentro de esta categoría se destacan los proveedores de alimentos, estos tendrán la posibilidad de manejar la disponibilidad de productos propios dentro del sistema, destacando su cantidad, precio y características; de igual manera, podrán aceptar o rechazar un pedido, una vez el comedor comunitario lo solicite. Poseen las siguientes características:
 - Dirección
 - Datos de contacto
 - Datos de contacto de encargado
 - Productos a ofrecer (precio, cantidad, descripción)
 - Proveedor de transporte: perfil que podrá controlar los vehículos que sean de su propiedad, podrá modificar a los mismos en cuanto a datos de contacto, nuevos registros y características en general, así como eliminarlos del sistema. Este perfil también podrá decidir si aceptar o rechazar una solicitud de transporte de productos una vez el comedor haya realizado una petición, a continuación, se describen las características:
 - Transportes disponibles
 - Datos de contacto
 - Datos de contacto de encargado
 - Administrador: perfil del administrador del sistema, quien podrá visualizar y editar toda característica del sistema en cuanto a registros (perfiles, productos, proveedores, comedores y pedidos) y podrá realizar pedidos y modificar recetas dentro del sistema. Tendrá las siguientes características
 - Datos de contacto
 - Acceso Comedores disponibles
 - Acceso a Proveedores disponibles
 - Acceso a productos
 - Ejecución de algoritmos de optimización
- **5.2.3 Lineamientos de contexto:**

Teniendo en cuenta el proceso y la identificación de cada uno de sus componentes se observa que la solución debe adaptarse al proceso que los comedores manejan actualmente y que permita, mediante tecnologías de la información, la automatización de tareas recurrentes; a continuación, se observan las características más importantes del proceso que actualmente se encuentra implementado dentro de los comedores respecto a los pedidos destacando los siguientes aspectos clave:

 - Se tiene en cuenta que cada comedor es independiente de otro y que cada uno tiene un número de personas a las cuales atiende de manera recurrente.
 - Se ofrecen menús diarios, por lo cual se realizan pedidos día a día y no en lote para un lapso mayor de tiempo.
 - Los productos que suelen ser donados por los proveedores alimenticios generalmente tienen fecha de vencimiento muy cercana a los tres días.
 - Actualmente VM y el Banco de Alimentos incurren en elevados costos de productos y transporte al no tener un modelo de información centralizado.
 - Los menús suelen ser modificados por los comedores no siguiendo un modelo estándar de productos necesarios para suplir la necesidad alimenticia.

- Los proveedores de transporte deben manejar distintos tipos de camiones, dependiendo de qué productos se van a trasladar de un proveedor alimenticio a un comedor.
- Los cálculos para realizar una receta dentro de un comedor deben ser hechos a mano ya que se tiene dificultad para calcular el número de personas contra el número de ingredientes necesario por receta.
- Los conductores suelen cobrar por una cierta cantidad de distancia recorrida en metros, esto varía dependiendo del tipo de transporte y su capacidad de carga.
- Las variables que se tienen en cuenta para el cálculo del costo total de un pedido son cinco (5) teniendo en cuenta proveedores y productos: el costo del valor unitario de producto, la disponibilidad de un proveedor alimenticio determinado, el tiempo de transporte, el costo de transporte y la distancia entre el comedor y el proveedor alimenticio

- **5.2.4 Estrategias de adaptación según contexto:**

A partir de las características de contexto identificadas en la sección anterior se plantean las siguientes estrategias para dar solución a la problemática, con sus respectivas razones:

- Recetas completamente modificables en cuanto a ingredientes y cantidades según se requiera.
- Modificación de características de productos por el administrador y los proveedores.
- Ingreso de direcciones para cálculo del vehículo más apto a través de api de Google maps.
- Se tendrán en cuenta variables de costos, distancia, tiempo y disponibilidad de productos y proveedores para ofrecer sugerencias.
- Se ofrecerán sugerencias de productos teniendo en cuenta el número de niños que posea el comedor comunitario y la receta que se elija.
- Centralización de información de productos y proveedores para los administradores del sistema.
- Toda entidad del sistema podrá ser exportada a Excel permitiendo manipular información desde afuera del sistema.
- Se ofrecerán usuarios a los 4 roles de sistema comedores, proveedores alimenticios, proveedores de transporte y administradores.

- **5.2.5 Mockups para pre-visualización, guía del modelo final del sistema y prototipo de sistema:**

Una vez identificadas las tareas de cada rol dentro de la aplicación se procede a desarrollar una aproximación visual mediante el desarrollo de mockups, esto permitirá la previsualización de cada uno de los menús y pantallas ofrecida a los roles dentro del sistema, clarificando el funcionamiento, pasos y restricciones y permitiendo crear una primera aproximación del diseño del front y la interacción del usuario con la misma (UI).

A continuación, se exponen los mockups que representan los servicios expuestos anteriormente:

MockUp: recetas

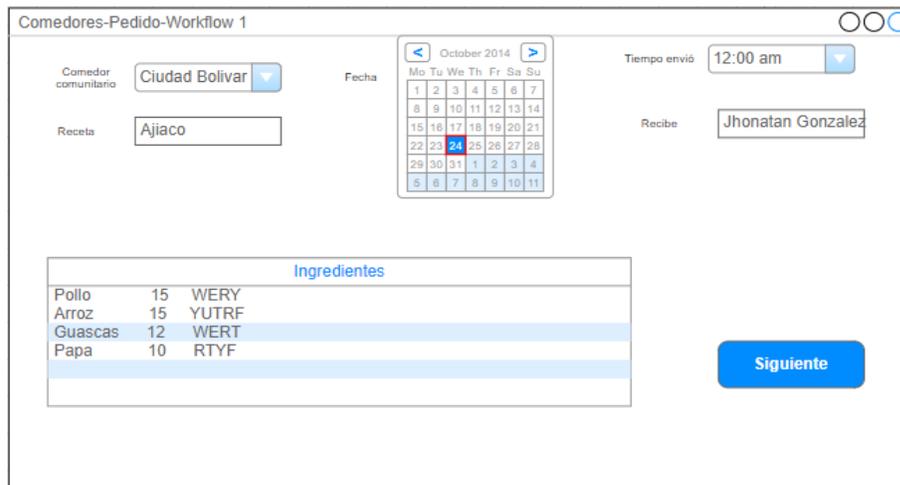


Imagen 3 Mockup recetario

En la imagen 3 se expone el mockup del recetario. Esta ventana permitirá al usuario buscar recetas que se encuentran relacionadas con el comedor, una vez éste la seleccione traerá los ingredientes configurados por este comedor con anterioridad necesarios para realizar el plato, de igual manera se traerán las cantidades necesarias para producir tantos platos de receta como niños se encuentren registrados en el comedor. De igual manera se puede configurar la fecha de entrega del pedido y la persona que recibirá el domicilio.

Mockup recomendación proveedores alimenticios y confirmación

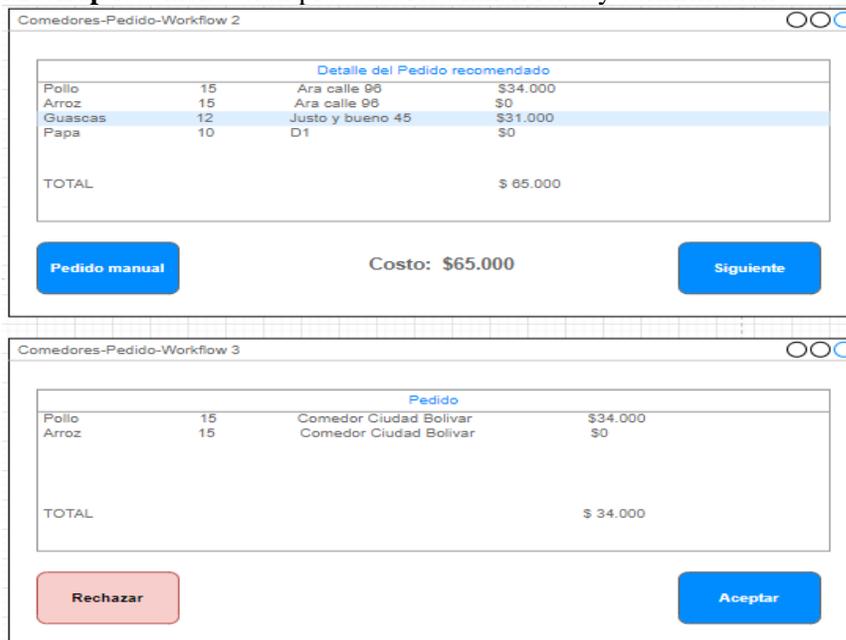


Imagen 4 Mockup Recomendación y aceptación de pedido

En la imagen 4 se aprecia cómo se visualizaría la respuesta del algoritmo de recomendación en la interfaz de usuario, los proveedores que vengan en este campo serán aquellos sugeridos por el algoritmo, permitiendo evidenciar cantidad, precios y nombres de los mismos. En este punto, el usuario podrá aceptar o rechazar el proveedor recomendado, en caso de que se rechace el proveedor se mostrará el segundo más eficiente o en caso de que se requiera se puede realizar el ajuste de manera manual.

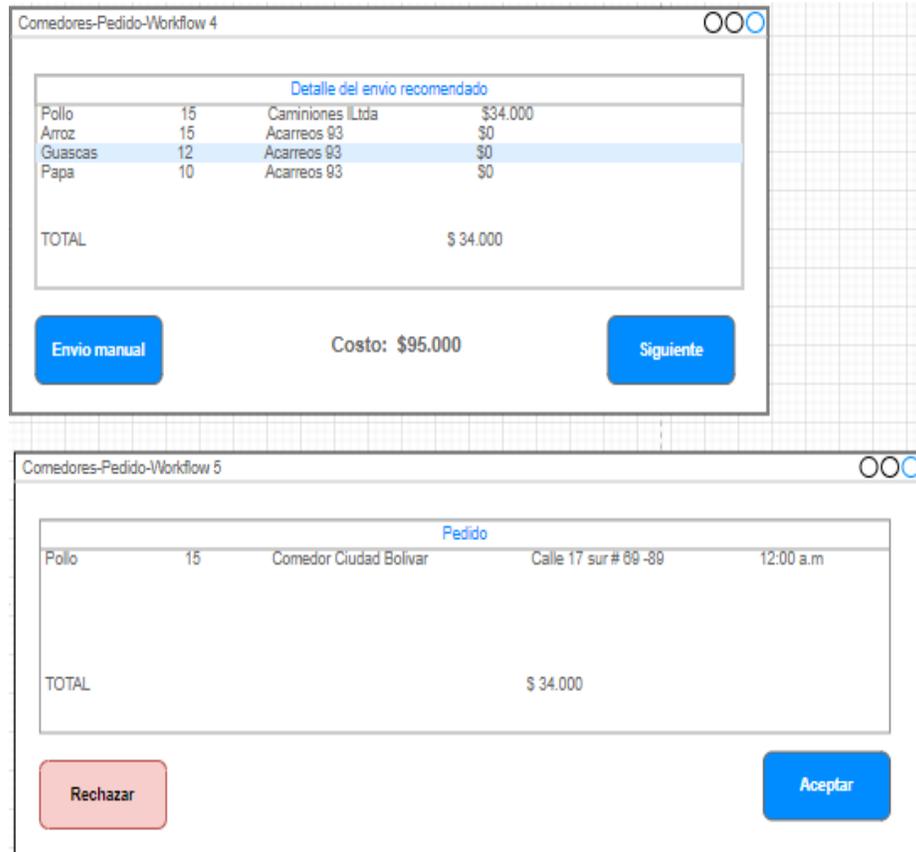


Imagen 5 Proveedores de transporte y confirmación

En la imagen 5 se aprecia cómo se presentaría al usuario los transportes recomendados por el algoritmo, teniendo en cuenta la distancia del pedido desde el comedor comunitario hasta el proveedor de alimentos teniendo en cuenta los costos de transportes por pedido y el tipo de productos que se va a transportar. De igual manera se puede aceptar la sugerencia del sistema o rechazarla y recibir el segundo proveedor más recomendado o realizar la modificación manualmente según conveniencia del usuario.

- **5.2.6 Servicios a ofrecer**

Según la problemática analizada anteriormente y el contexto de la situación se proponen los siguientes servicios para facilitar la tarea y que serán incluidos dentro de la plataforma informática para mejorar la planificación de proceso.

- **Realizar pedido:** este servicio tendrá como funcionalidad la realización de pedidos a solicitud del comedor comunitario; estos pedidos podrán ser personalizados según la necesidad de éste o de las funcionalidades requeridas por el mismo usuario. El servicio estará compuesto de las siguientes funcionalidades:
 - **Control de catálogos:** esta funcionalidad ofrecerá al usuario de revisión de existencias por producto y precio, permitiendo la selección de los grupos de productos que más se ajusten a la necesidad del comedor. Éste también podrá modificar la cantidad o incluso borrar o insertar nuevos productos a la lista.
 - **Sugerir proveedores:** el sistema permitirá a través de un modelo de optimización basado en atributos de eficiencia, la sugerencia al comedor comunitario de proveedores alimenticios según productos o la sugerencia de medios de transporte según los vehículos precargados en sistema, esto permitirá mejorar la eficiencia en cuanto a costos de envíos y de productos, los cuales también podrán ser monitoreados por el administrador del sistema para procesos estadísticos.

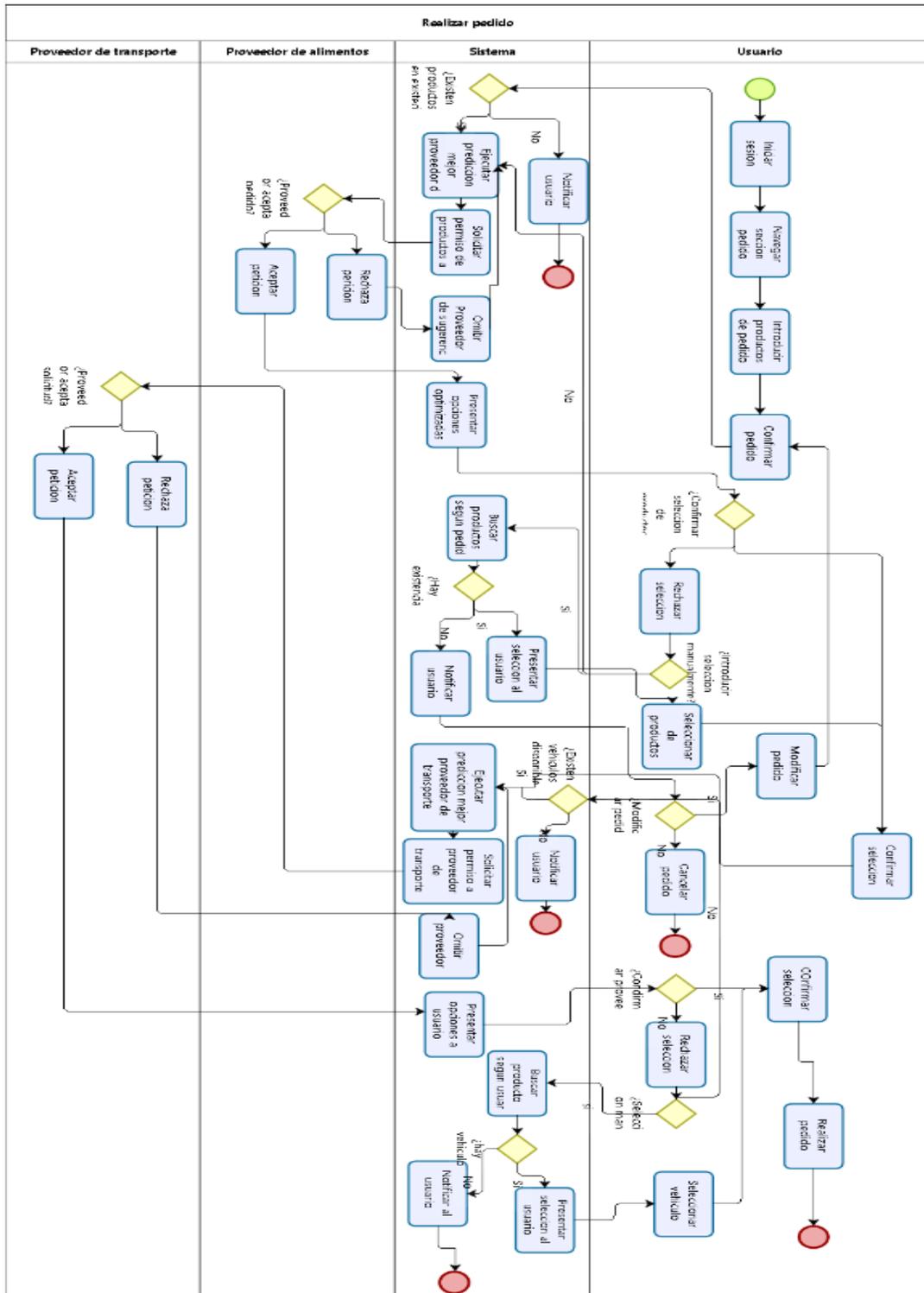


Figura 1 BPMN Realizar pedido

- **Manejo de recetario:** el sistema permitirá la estandarización de recetas o pedidos, estos se guardarán a modo de plantilla y permitirán la precarga de los insumos necesarios para la elaboración del platillo teniendo en cuenta el número de personas que acuden al comedor determinado. Este servicio se compone de dos funcionalidades, que trabajan simultáneamente:
 - **Control de receta:** esta funcionalidad permitirá cargar la receta en el sistema a modo de plantilla permitiendo estandarizar el plato y que en pedidos futuros el pedido se realice de manera más automática permitiendo organizar lotes de insumos necesarios.
 - **Realizar Prepedido de recomendación:** esta característica tiene como funcionalidad presentar al usuario la carga automática de un plato teniendo en cuenta los insumos necesarios para el plato; este pedido permitirá retirar o agregar insumos según el comedor lo requiera; igualmente, se realizará un cálculo automático para que el sistema realice el pedido teniendo en cuenta el número de personas en total que acuden al comedor comunitario, permitiendo que la cantidad se maneje de forma automática y calculada.

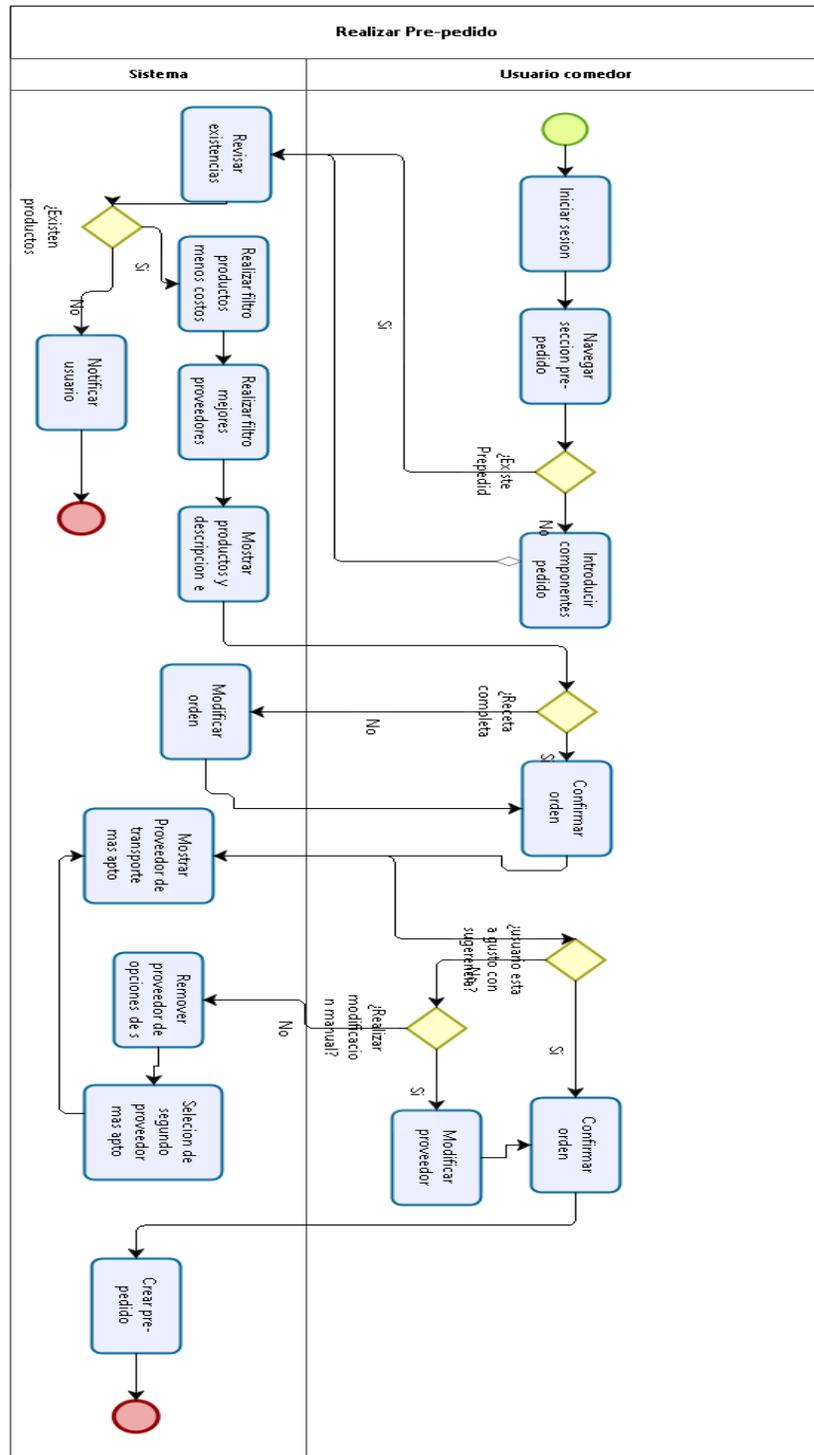


Figura 2 BPMN Realizar prepedido

5.3 Proceso de modelado logístico:

En esta fase se consolida la unión entre los procesos de optimización entregados por el equipo de Ingeniería Industrial con los modelos de adaptación planteados; de esta forma, se puede integrar la lógica de optimización de sugerencias mediante la visualización acorde a cada perfil de usuario y sus funcionalidades propias dentro de la aplicación, permitiendo al sistema realizar sugerencias al comedor o al cliente (VM)

- **5.3.1 Seudocódigo de modelos de optimización planteados por el equipo de Ingeniería Industrial:**

A través del conocimiento adquirido de los modelos de optimización surge la opción de elegir el modelo de optimización de Simplex como factor de éxito más probable, ya que permite consolidar las variables más importantes para optimizar la eficiencia del proceso, a continuación, se explica el proceso:

1. Se construye la función objetivo donde:
$$((A1 \times 1)^2 + (A2 \times 2)^2 + (A3 \times 3)^2) / NP * 100 = A1E$$
 - A1.. AN son las métricas de cada proveedor (Costo del producto, cantidad del producto, distancia de comedor a proveedor y vehículo)
 - Np es el número de proveedores
 - X1...XN son las variables que serán multiplicadas por los coeficientes de A1.. AN
2. Se normalizan las métricas de los proveedores con la formula expuesta anteriormente
3. Se crea el proveedor ideal donde todas sus variables serán las Mínimas
4. Se utiliza la clase de SolverContext que permite emular un proceso Simplex similar al de Excel y que sigue los mismos pasos expuestos con anterioridad para la resolución del proceso.
5. Se utiliza esta misma clase para introducir restricciones de formula Simplex
6. Se ejecuta el proceso de solver por cada proveedor que exista en la base de datos con sus respectivas variables
7. Se organizan según nivel de eficiencia en una lista
8. Se envía la lista organizada al front para que sea visible por el usuario final

- **5.3.2 Seudocódigo de implementación con las integraciones de proveedores:**

A partir de la información de los proveedores suministrada en cuanto a existencias, precios y Productos, se procedió a crear el modelo de la base de datos permitiendo identificar el modelado de las clases; de igual manera, se procede a recopilar información de interconexiones tablas intermedias, elementos necesarios para los informes y la forma de acceder a los mismos, se identifican las siguientes entidades que permitirán almacenamiento:

- **Dinning room:** tabla de comedores que permitirán la descripción de cada uno de ellos teniendo en cuenta, datos de contacto como teléfono, contacto de emergencia, ciudad y departamento; sin embargo, el más importante será la dirección ya que a

- partir de ella se identificarán los proveedores alimenticios más cercanos y la trayectoria que deberá hacer un vehículo para transportar los insumos
- **Disponibility:** tabla que permitirá identificar que disponibilidades tienen cada uno de los proveedores alimenticios; en esta tabla se centrará la cantidad por proveedor y se proveerá la fecha de expiración de los alimentos, centralizando de esta manera todos los productos de todos los proveedores existentes en sistema; de igual manera se estandarizará la unidad de medida de cada producto, para facilitar los pedidos en la plataforma
 - **Ingredient:** tabla que contendrá la referencia a la receta y a los productos, al igual que la cantidad necesaria de productos para elaborar la receta.
 - **Order:** tabla que permitirá identificar la fecha de pedido y el id de una orden creada por el cliente
 - **Order Item:** tabla que permitirá reconocer los elementos de un pedido, es decir, qué productos pidió el cliente, el proveedor alimenticio, de transporte, la aceptación de esta por parte de los proveedores y la fecha de pedido
 - **Pre order:** tabla que permitirá al comedor realizar pedidos de recetas o pedidos personalizados, permite almacenar recetas precargadas y ser presentadas en la visualización de interfaz, también contiene la fecha de la orden
 - **PreOrderItem:** tabla que permitirá definir qué productos pertenecen a una preorden (ver descripción anterior) destacando por producto, el costo, el costo multiplicado por el número de productos de receta, el transporte a emplear, el costo del mismo según trayecto, fecha de vencimiento de productos y aprobaciones de proveedor de transporte y alimento
 - **Product:** tabla que permitirá definir que insumos ofrecen los proveedores de alimentos, conteniendo datos de este como tipo de medida, tipo de producto, modo de conservación, nombre y descripción
 - **Provider:** proveedor dentro del sistema que puede referenciar a uno de tipo alimenticio o de transporte, contendrá datos de contacto como el nombre, la dirección, el tipo, el contacto principal, la ciudad y departamento donde opera
 - **Recipe:** receta alojada dentro del sistema contendrá el nombre y un código de referenciación dentro del sistema.
 - **System User:** tabla que permitirá caracterizar los roles dentro del aplicativo diferenciando a los clientes, de proveedores y de los comedores, contendrá la descripción y datos de contacto.
 - **Transport:** transporte de carga para cargar los insumos de un proveedor alimenticio a un comedor, tendrá información descriptiva del vehículo, como tamaño, tipo de carga, placa, fecha de ensamblado, modo de pago, disponibilidad y los datos del conductor como número de contacto o email

Una vez identificadas las tablas y relaciones se procede a construir las funcionalidades básicas de los servicios ofrecidos, teniendo en cuenta la configuración del ambiente y la visibilidad de la misma interfaz para los tres (3) tipos de usuarios que alberga.

La receta permitirá el alojamiento y precarga de platos dentro del sistema, es decir, se estandariza un plato y sus insumos; sin embargo, se adiciona la capacidad de modificar dicha receta por los comedores ya sea añadiendo más ingredientes o removiéndolos, quedando a consideración la elección del cliente ampliando el modelo personalizable en cuanto a preferencias, de

igual manera, las recetas podrán ser cargadas o descargadas a un Excel por comedor para facilitar la modificación o carga de datos

En cuanto a los pedidos se le permitirá al usuario personalizar los insumos completamente según las existencias en disponibilidad, se sugerirá la mejor opción en cuanto a transporte y producto y se permitirá la elección del comedor por parte de World Vision para realizar la entrega, las selecciones que el sistema presentará ya vendrán optimizadas en cuanto a costo de insumos, distancia y vehículo más apropiado teniendo en cuenta la totalidad de proveedores de ambos tipos dentro del sistema

- **5.3.3 Diagrama de logística:**

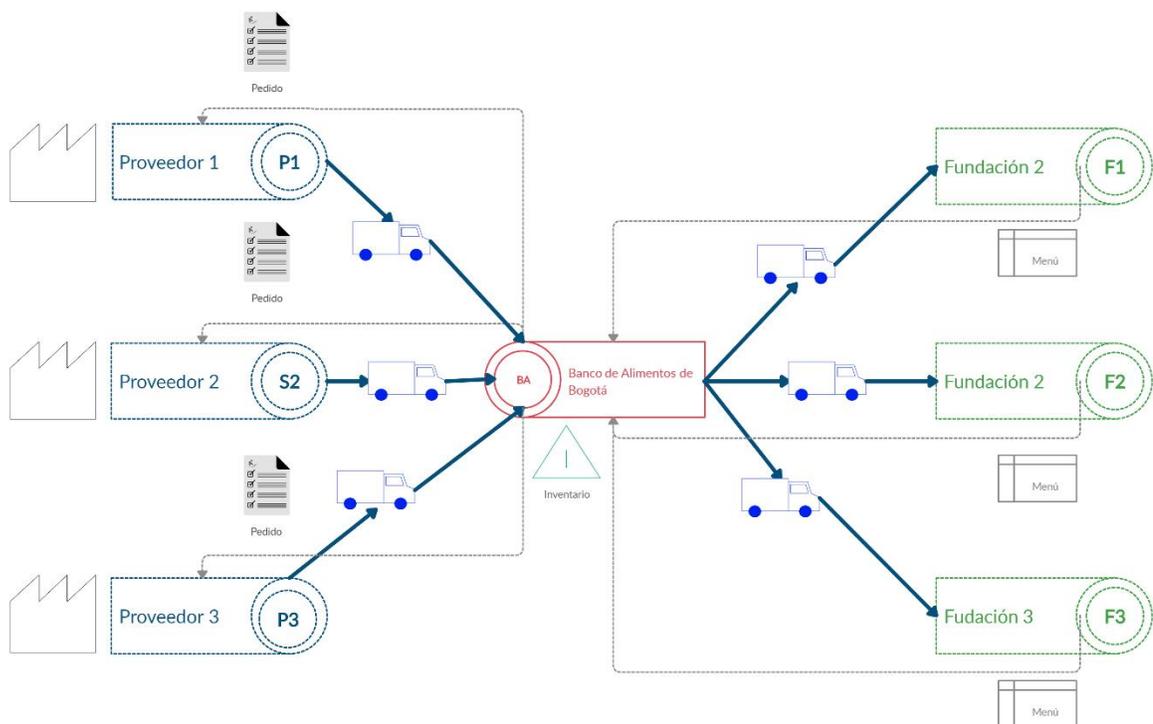


Diagrama 1 Modelo de proceso

En el diagrama 1 Modelo de proceso (Agradecimientos a Alejandro Giron y Clara Mabel Solano) se puede apreciar el proceso general que conforma la cadena de distribución; en primer lugar, se pueden apreciar varios proveedores alimenticios, los cuales serán los que ofrecerán los productos dentro del sistema. Cabe destacar que cada proveedor maneja las características propias de sus productos en lo que se resume marcas, precios y fechas de vencimiento.

En el punto central se puede apreciar al Banco de Alimentos/Visión Mundial quienes serán los administradores del proceso permitiendo tramitar los productos y su respectiva adquisición desde un proveedor a un comedor determinado. Este actor será el que esté al tanto de cada paso

dentro del proceso de negocio cubriendo las necesidades de los comedores y comprando insumos de los oferentes. Para este rol se destacan las variables de costo en cuanto a productos y transporte, compra según disponibilidad y tiempos de entrega a los comedores comunitarios.

Finalmente se destaca a los comedores dentro del modelo quienes serán aquellos que reciban los insumos comprados por el Banco de Alimentos y VM, permitiendo de esta manera cubrir las necesidades de los niños de bajos recursos que asisten a dichas instituciones.

A partir de dicho modelo se busca facilitar el entendimiento del proceso general actual que ejecutan VM y el Banco de Alimentos para atender las necesidades de los comedores comunitarios. Cabe destacar la importancia de la administración de la gestión logística que realizan estas entidades, por ende, es indispensable facilitar el manejo de proveedores, productos y comedores, para soportar el proceso de toma de decisiones relacionadas con la cadena de distribución para así reducir costos y tiempos.

- **5.3.4 Características del proceso:**

Para optimizar el proceso se procede a indicar las siguientes características del modelo de Logística

- Los comedores podrán solicitar pedidos a modo de recetas los cuales serán ampliamente modificables permitiendo elegir los productos que se necesiten, al igual que los proveedores que los ofrezcan.
- Los productos ofrecidos por los proveedores, en caso de donación se entregarán con 3 días mínimo-faltantes para cumplir la fecha de vencimiento.
- Las peticiones podrán ser aceptadas o rechazadas por el proveedor de alimento o de transporte.
- Los transportes actualmente son de dos tipos: de refrigeración o de carga normal según el pedido que se solicite.
- La ruta óptima será necesaria para minimizar distancia de un proveedor de alimento a un comedor, permitiendo de manera indirecta minimizar costos de transporte a la vez.
- Los administradores requerirán toda la información relevante a costos, productos y envíos de una manera centralizada para agilizar proceso de negocio.

5.4 Implementación del Sistema:

En esta etapa se hace referencia a la construcción e implementación del sistema siguiendo las pautas planteadas en los seudocódigos, los procesos de modelado del proceso de optimización, la maquetación de servicios y tareas dentro del sistema según el usuario y la solución a la problemática planteada a nivel social como objetivo principal. Se especifican en mayor grado los requerimientos y se relacionan con los servicios planteados permitiendo priorizarlos según funcionalidad, igualmente se analizan posibles patrones de desarrollo que permitan optimizar la funcionalidad de la plataforma, dando como resultado una arquitectura bien estructurada y que permita un mantenimiento sencillo y fácil de comprender, permitiendo diferenciar, lógica de negocio, acceso a datos y presentación a usuario. Una vez evidenciados los puntos clave a

tratar dentro del diseño se inició la construcción de la plataforma comedores siguiendo metodología Scrum, realizando dailys entre el equipo de trabajo y stand ups con el cliente indirecto que en este caso es la Doctora Clara Mabel Solano del departamento de Ingeniería Industrial.

- **5.4.1 Definición de metodología :** una vez conociendo el contexto de la situación y las necesidades específicas del cliente se procede a definir la metodología a emplear para la resolución del proyecto, dado el lapso de tiempo y la urgencia de un sistema de información por parte del cliente, se elige Scrum[35] como mejor metodología de trabajo para lograr cumplir los objetivos, esto basándose en que ambos desarrolladores provienen del uso de metodologías ágiles, se conocen los procesos y reuniones importantes en torno al levantamiento de requerimientos y funcionalidades y por la premura de la situación, se procede a la creación de calendario en conjunto con Clara Mabel, se organizan dailys y stand ups para revisar la implementación del proyecto tanto interno al grupo como externa con clientes.
- **5.4.2 Definición de requerimientos:** a partir de las necesidades del negocio, de la información obtenida por el equipo de Ingeniería industrial y requerimientos a nivel funcional planteados por los desarrolladores se plantean los requerimientos necesarios para suplir las necesidades y como tal, dar una solución a la problemática, se reconocen requerimientos funcionales y no funcionales permitiendo reconocer los más necesarios e indispensables para el funcionamiento del sistema, los requerimientos se especificaban con cada daily dentro del grupo durante las primeras dos semanas del periodo, los requerimientos se pueden ver en el documento adjunto a este trabajo de grado (Software requirement specification)
- **5.4.3 Refinamiento de requerimientos:** una vez planteados los requerimientos se definía al final de cada semana cuáles eran los más críticos para el sistema y cuáles afectaban más los servicios ofrecidos por el equipo de comedores, dicha priorización se realizó siguiendo el modelado de MoSCoW, donde cada requerimiento se evalúa según importancia teniendo en cuenta los siguientes indicadores:
 - **M:** must have
 - **S:** Should have
 - **C:** could have
 - **W:** Would and would have.

De dicho refinamiento se procede a estipular los tiempos de duración, según nomenclatura M-> 3-4 días, S-> 5-7 días, C 7-9 días, W-> 10 días

- **5.4.4 Planteamiento de atributos de calidad:** la siguiente etapa a desarrollar fue el planteamiento de los atributos de calidad, paso indispensable durante la construcción de una arquitectura de software y que permite identificar que modelos, esquemas o patrones serán los más aptos para la implementación contemplando la problemática y el contexto; sin embargo, se especificó el más importante para dirigir el modelo a conseguir dicha característica.

A continuación, se describen los atributos de calidad seleccionados, una breve definición según citación y el por qué se eligieron dichos atributos:

 - **Funcionabilidad:** “Habilidad de un producto de software de realizar las funciones para las que fue creado” [27].

- **Por qué se eligió:** se eligió porque el producto debe permitir a los usuarios realizar las tareas que se deben hacer a nivel de proceso de negocio, ya que es necesario para la solución de la problemática; de igual manera, este atributo significa el desarrollo como tal del proyecto y especifica los mínimos viables para que pueda ser usado por un usuario.
- **Cómo se aplica:** se aplicará dentro del prototipo mediante la resolución de la problemática general, modelando las tablas que contendrán información, la construcción del backend con la lógica de sugerencias en cuanto a pedidos y recetas y con la construcción del front para que el usuario pueda ejecutar los casos de uso dentro del aplicativo
- **Mantenibilidad:** *“Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas” [26]*
 - **Por qué se eligió:** se eligió este atributo debido a que es un producto que va a quedar a disposición de una causa social, por ello se construyó dentro del equipo de trabajo un software que pueda ser modificado en cuanto a componentes visuales, de lógica y de acceso a datos, se comentó el código y se modularizó según funcionalidades a partir de controladoras para que sea entendible para los demás desarrolladores
 - **Cómo se aplica:** se realizó el desarrollo siguiendo el patrón de Multinivel y capas, el primero porque permite separar el modelo de la vista y del controlador separando las tres lógicas y permitiendo modificar cada uno de estos componentes de forma separada según la necesidad del programador. El patrón de capas permite separar la lógica en el backend orientando cada capa a cumplir un objetivo en específico, como puede ser el acceso a datos, el ofrecimiento de servicios, los wrappers para envío o recepción de Json o los algoritmos de sugerencia, esto permite organizar el código de manera entendible y eficiente para posibles correcciones o modificaciones a futuro.
- **Seguridad:** *“Algunas visiones de la seguridad son:*
 - *Comprobar la identidad de las personas que intentan acceder al sistema.*
 - *Garantizar que sólo las personas específicamente autorizadas pueden ver determinada porción de la información del sistema*
 - *Garantizar que sólo las personas específicamente autorizadas pueden modificar determinada porción de la información del sistema o bien realizar determinadas acciones.”[28]*
 - **Por qué se eligió:** es necesario proteger la información de los actores dentro del sistema ya que se va a trabajar con información sensible como datos de contacto y datos de niños dentro del sistema, por lo cual es indispensable que se manejen buenas medidas de seguridad.
 - **Cómo se aplica:** se aplicó mediante el desarrollo de un Rest api que usa protocolo https para seguridad en cuanto a canal de comunicación; de igual manera se empleó JWT(Json Web Tokens) para el manejo de peticiones; así, solamente los usuarios que se encuentren registrados en la plataforma podrán usar los servicios ofrecidos por la misma, esto debido a que permite el uso de un token con caducidad determinada para evitar problemas de seguridad[30]

- **5.4.5 Desarrollo de casos de uso:** a partir de los actores y funcionalidades según necesidades se construyen casos de uso mediante el agrupamiento de requerimientos refinados en la etapa anterior, a continuación, se puede apreciar el diagrama de casos de uso:

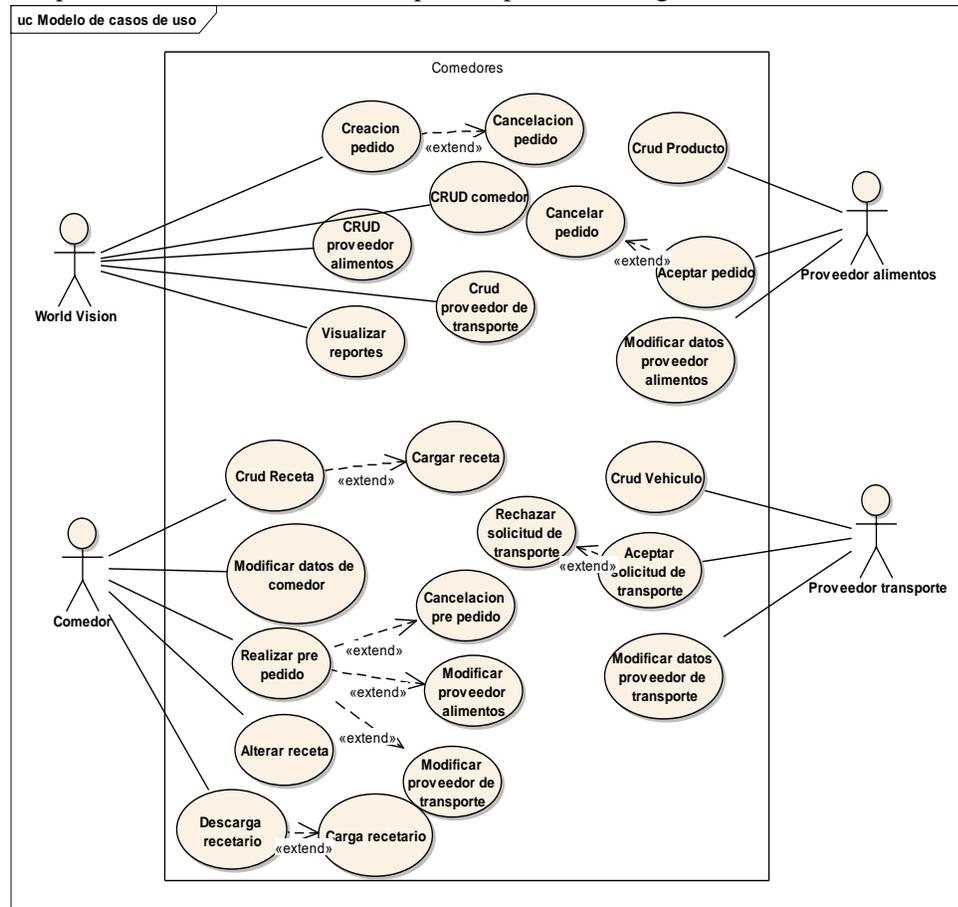


Imagen 6 Diagrama de casos de uso

A continuación, se describen cada uno de los casos de uso presentados en la Imagen 6:

Actor: Visión Mundial (VM)

- **Creación Pedido:** VM tendrá la opción de crear pedidos teniendo en cuenta los insumos que se encuentren en la tabla de disponibilidad según la cantidad de estos, VM podrá realizar dichos pedidos a los comedores comunitarios permitiendo la modificación de productos en cuanto a proveedores alimenticios o transporte en cuanto a proveedores de transporte, de igual manera, podrá anular una orden de pedido si es necesario
- **Cancelación de pedido:** si VM no está conforme con el pedido creado con anterioridad podrá eliminarlo y cancelar la orden hacia los proveedores.

- **CRUD comedores:** VM tiene permisos como administrador del sistema de crear, modificar, actualizar o visualizar comedores almacenados dentro del sistema según la operación lo requiera
- **CRUD Proveedor de alimentos:** VM tiene permisos como administrador del sistema de crear, modificar, actualizar o visualizar Proveedores de alimentos almacenados dentro del sistema según la operación lo requiera.
- **Visualizar reportes:** VM como administrador del sistema podrá ver reportes a nivel de interfaz gráfica que permita visualizar los pedidos realizados, los proveedores alimenticios, proveedores de transporte, productos y comedores y características propias de los mismos.

Actor: Comedor

- **CRUD Receta:** los comedores podrán modificar, crear, ver o borrar recetas del sistema según lo requieran
- **Carga receta:** los comedores podrán cargar recetarios almacenados en la plataforma con anterioridad a modo de plantilla, especificando los productos, cantidades, y procesos necesarios para la elaboración del platillo, esto para agilizar el proceso de petición de prepedidos.
- **Modificar datos de comedor:** el comedor podrá modificar sus datos de contacto alojados dentro del sistema según lo requiera.
- **Realizar prepedido:** el comedor podrá realizar peticiones de insumos a proveedores alimenticios según lo requiera, el sistema utilizara el algoritmo de optimización sugiriendo los mejores proveedores de alimentos y el mejor transporte para realizar él envío, sin embargo, el usuario puede modificar las sugerencias presentadas por pantalla.
- **cancelación de prepedido:** el comedor podrá anular una petición anterior de prepedido anulando la transacción a partir de la interfaz.
- **Modificar proveedor de alimentos:** el sistema por naturaleza sugerirá el mejor proveedor de alimentos al usuario a través de la interfaz, sin embargo, el usuario puede cambiarlo durante la creación del prepedido.
- **Modificar proveedor de transporte:** el sistema por naturaleza sugerirá el mejor proveedor de transporte al usuario a través de la interfaz sin embargo el usuario puede cambiarlo durante la creación del prepedido.
- **Alterar receta:** el comedor podrá alterar la receta que viene por defecto en el sistema permitiendo introducir nuevos insumos a la misma o retirarlos para personalizar el prepedido.
- **Descarga de recetario:** los comedores podrán descargar los recetarios que posean a modo de Excel para facilitar la modificación de los mismos o para tenerlos como referencia fuera del sistema.
- **Carga de recetario:** el sistema admite la carga de recetas a partir de un documento Excel con la misma estructura de la descarga, esto permite facilitar la modificación de recetas por fuera del sistema.

Actor: Proveedor de alimentos

- **Crud producto:** el proveedor podrá modificar, crear, borrar o visualizar los productos que se encuentren en sistema y le pertenezcan.
- **Aceptar pedido:** el proveedor podrá decidir aceptar un pedido una vez se haya solicitado ya sea por un comedor o por VM.
- **Rechazar pedido:** el proveedor podrá decidir rechazar un pedido una vez se haya solicitado ya sea por un comedor o por VM.
- **Modificar datos proveedor de alimentos:** el proveedor de alimentos podrá modificar sus datos de contacto según necesidad.

Actor: Proveedor de transporte

- **CRUD Vehículo:** el proveedor podrá modificar, crear, borrar o visualizar los vehículos que se encuentren en sistema y le pertenezcan.
- **Aceptar pedido:** el proveedor podrá decidir aceptar una solicitud de envío una vez se haya pedido ya sea por un comedor o por VM.
- **Rechazar pedido:** el proveedor podrá decidir rechazar una solicitud de envío una vez se haya pedido ya sea por un comedor o por VM.
- **Modificar datos proveedor de Transporte:** el proveedor de transporte podrá modificar sus datos de contacto según necesidad.

- **5.4.6 Patrones de diseño:**

A continuación, se explican los patrones de diseño utilizados en “Comedores”.

5.4.6.1 Patrón Multinivel: estilo de arquitectura que se encarga de separar los datos de una aplicación, la interfaz de usuario y la lógica en tres componentes independientes.[31]

- **Modelo:** posee la representación de los datos del sistema, la lógica de negocio y el manejo de la persistencia.
- **Vista:** representa los mecanismos de interacción del usuario a partir de la interfaz
- **Data:** es el intermediario entre el modelo y las tablas de base de datos gestionando la data que se transporta entre ellos.

Dentro de la aplicación “Comedores” se tendrá el algoritmo de optimización en el modelo en conjunto con la persistencia dada a partir del uso de *Entity Framework* y la lógica de negocios en general. En la vista se ubicarán los componentes de interacción del usuario empleando tecnologías de angular y que a través de *Promises* se conectarán con el controlador del *backend* para el tratamiento de datos

5.4.6.2 Patrón Layers: es un estilo de arquitectura en donde se trabaja a modo de dependencias por niveles donde los niveles superiores dependen de los niveles inferiores; de igual manera, cada nivel posee una lógica propia encapsulada dirigida al cumplimiento de una función en específico.

Dentro de la arquitectura de comedores el modelo de capas será alojado dentro del *backend* en donde cada capa realizará modificaciones sobre la información o cumplirá tareas específicas

para dividir la lógica, las capas de las cuales se compone el prototipo de comedores son las siguientes:

1. **Controllers:** capa que permitirá la comunicación con el *front* y que permitirá exponer los servicios al exterior del *backend*, vienen divididos según los elementos en la base de datos donde cada elemento tiene su controlador
2. **Logic:** lógica general del negocio aquí se realizarán todas las operaciones necesarias sobre datos para el retorno de una respuesta al usuario, en esta capa también se encuentran los modelos de normalización y de optimización del negocio.
3. **BLL:** (Busines logic Layer) lógica individual por objetos, cada objeto tiene su BLL y cumplirá funcionalidades de negocio propias de éste, permitiendo creación, eliminación, modificación y lectura de cada objeto al igual que construcción de *wrappers* para el retorno del objeto a modo de *Json* o la deserialización del mismo para tratar datos dentro de la lógica
4. **DAL:** (Data Access Layer) capa de acceso a datos, ésta estará presente en cada objeto del modelo permitiendo a través del manejo del *Entity Framework* manejar la persistencia entre base de datos y el *backend*.
5. **Entities:** capa de entidades que serán transversales al modelo en general permitiendo transportar los objetos a través de cada una de las capas para realizar tareas específicas
Se planteó la distribución de capas de esta manera estandarizada ya que permite mayor control sobre las funcionalidades de cada capa y permite al programador identificar fácilmente donde puede ubicarse la funcionalidad.
6. **Integración:** capa que permitirá a través de la construcción de *wrappers* realizar peticiones Rest al Api de Google y obtener los valores de Distancia en cuanto a la ruta óptima de un comedor a un proveedor de alimentos en tiempo real.

5.4.7 Arquitectura:

Una vez planteados los atributos de calidad y los patrones, se expone la definición de ésta a continuación:

5.4.7.1 Arquitectura Lógica:

La arquitectura lógica de la plataforma de comedores se plantea a continuación, teniendo en cuenta los componentes, patrones y paquetes de información, agrupando los servicios del sistema según concordancia con las entidades que conforman el mismo

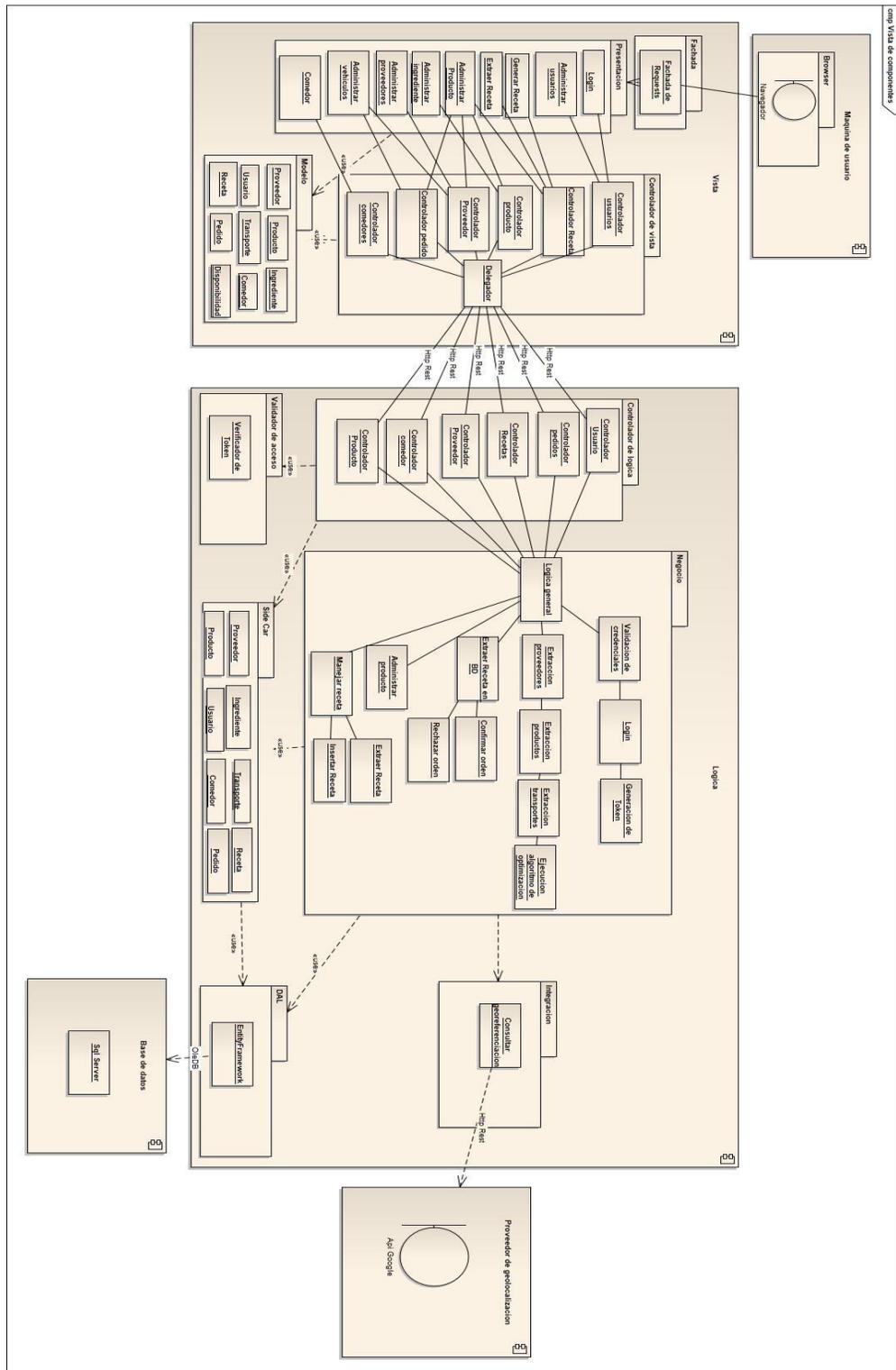


Imagen 7 Arquitectura comedores

La arquitectura se expone en la imagen 7. Se muestra cada uno de los componentes que permitirán realizar las funcionalidades y ejecutar los servicios expuestos con anterioridad. Se puede apreciar una arquitectura multinivel o N-Tier que permite dividir la lógica por componentes físicos, permitiendo lograr mayor escalabilidad, simplicidad y hacer entendible el Código; de esta forma, se compone de módulos la solución, separando la presentación, de la lógica y de la persistencia, a continuación, se expone cada uno de los niveles de la arquitectura:

- **Nivel Vista:** nivel que permite mostrar gráficamente al usuario las tareas ejecutadas por la plataforma, se compone a partir de componentes que permiten al usuario interactuar con el sistema y realizar peticiones; este nivel se comunica a través de una fachada con el browser del usuario, permitiendo abrir la aplicación independientemente del dispositivo en uso, este nivel corresponde al *frontend* y se conectará al *backend* a partir de controladores dedicados, según el tipo de petición y el objeto que controle el otro extremo, se aprecia de igual manera que existe un subcomponente dentro de la vista que permite utilizar las entidades a modo de espejo con las del *backend*, esto es necesario para la construcción de *wrappers* y, el envío posterior del Json al *backend*.
De igual manera, se aprecia que la conexión con el *backend* se hará a través de protocolo Http de tipo Rest, esto debido a que es una manera estandarizada de realizar peticiones y permite una Simplicidad mayor a Soap.
Las funcionalidades y servicios de comedores se expondrán a partir de este componente de arquitectura permitiendo al usuario conectarse al mismo a través de un browser. La visualización de comedores es responsive, permitiendo que sea visualizada en un móvil o en un ordenador.
De igual manera este componente será el encargado de validar las credenciales del usuario en una pantalla de login, a través de la cual se realizará un request y se obtendrá un Bearer Token del backend, el cual es necesario para ejecutar cualquier otro request de la plataforma una vez se haya logueado el usuario.
- **Nivel Lógica:** será el nivel encargado de ejecutar las tareas del *frontend* y de realizar las tareas de comedores, es en este nivel donde se ubicará el algoritmo de optimización (Simplex) y la normalización de datos de los proveedores para la ejecución del cálculo, de igual manera en este nivel se realizará la creación del **Bearer token** que permitirá acceso a los servicios únicamente a los usuarios registrados en plataforma, añadiendo una nueva capa de seguridad. Este nivel también posee una capa transversal a las demás: un sidecar que permitirá el uso de las entidades base a lo largo del *backend*. Adicionalmente el *backend* se conectará con un Api de Google externo empleando Rest para permitir obtener la geolocalización y distancia entre dos puntos (Comedor y Proveedor de alimentos).
Este nivel permitirá la ejecución de la lógica de negocio contenida en comedores a través de las requests provenientes del Frontend, en este nivel se recibirán los requests a partir de controladores según las entidades de comedores (comedores, proveedores, productos).
- **Nivel de datos:** corresponde al manejo de la base de datos a través de *Entity Framework* permitiendo recopilar información que será utilizada en el *backend*, En este nivel se con-

tendrán las tablas definidas según el negocio de comedores conteniendo data como proveedores, productos, comedores y cada una de sus características correspondientes para la ejecución de procesos a nivel de negocio.

El segundo patrón empleado fue el de capas; este fue aplicado en el *backend* y se ubica a lo largo del mismo y permitirá ejecutar las funcionalidades del negocio de manera separada y dividida para aumentar la seguridad y precisión del proceso de ejecución, las capas desarrolladas son representadas a continuación:

- **Capa Controlador de lógica:** contiene a todos los controladores con sus firmas de métodos para ofrecer servicios al exterior del *backend*, se encuentran realizados con un Rest API y asegurados con un JWT (Json Web Tokens) de esta manera únicamente los usuarios que se encuentren registrados en plataforma podrán consumir funcionalidades, todos los controladores tienen habilitado el *Cross origin* para que puedan ser consumidos desde cualquier petición externa. Cada entidad dentro del negocio de comedores tiene su propio controlador permitiendo dividir lógica.
- **Capa de Lógica:** capa de lógica general, permite la transmisión de datos desde la capa de controladores a la capa de BLL o capa de business logic, la capa de lógica se encarga de redireccionar el contenido de las peticiones a la lógica en capas correspondiente, esto permite centralizar lógica general, es en esta capa donde se ejecutan los procesos de optimización y normalización de los datos provenientes de *frontend*, el proceso de normalización recibirá los parámetros de optimización de la base de datos y ejecutara los procesos necesarios para estandarizar la unidad de medida, a continuación con dichos datos normalizados se procede a ejecutar los procesos de Simplex y DEA para hallar la mejor efectividad en cuanto a proveedor alimenticio y de transporte, las variables a calcular con estos procesos son: costo de producto , costo de envío, distancia y días de vencimiento.
- **Capa de negocio o BLL:** capa especializada de lógica de negocio se separa por componentes de entidades permitiendo que la capa asigne al componente correspondiente la ejecución de la petición externa, igualmente permite la conexión con la capa DAL o de acceso a datos separando lógica de persistencia. Esta capa dividirá la lógica general del negocio de comedores permitiendo que cada entidad de negocio posea su propio CRUD y sus funciones especializadas ejecutadas de una manera independiente, por ejemplo: la capa BLL de transporte permitirá buscar el transporte más eficiente según la categoría de alimentos que se requiera transportar minimizando de esta forma costos.
Es importante destacar que esta capa permitirá a través de la minimización de costo y de distancia y la maximización de fecha de vencimiento y disponibilidad la elección del mejor producto y vehículo, de igual manera cuando sea un pedido “partido” es decir que el proveedor es ideal pero no se posee la cantidad acorde al número de niños se ejecutará nuevamente el algoritmo de optimización y se elegirá al siguiente proveedor de alimentos óptimo hasta suplir la necesidad en su totalidad.
- **Capa de acceso a datos o DAL:** la capa de acceso a datos permite a través del uso de un *Entity Framework* el acceso al modelo y datos de la base de datos, se empleó dicha tecnología ya que permite modificar en sentido bidireccional, ya sea desde el modelo en *backend* o desde la misma base de datos, permitiendo realizar consultas rápidas y sin necesidad de

modificar mucho las *queries*. De igual manera cada entidad permitirá ser accesada por una única clase especializada que si bien usa la misma conexión a base de datos permite efectuar cambios o modificaciones específicas de cada entidad dentro del negocio

- **Capa de entidades:** capa de datos que es transversal a todas las capas de lógica, esto porque permite consistencia en el transporte de datos entre las mismas. Esta capa permite la simulación de las clases en base de datos a manera de espejo simulando con exactitud los atributos de cada tabla, su funcionalidad es transportar los datos entre las demás capas. Todas las entidades de negocio se encontraran en esta capa permitiendo identificar desde los comedores hasta los proveedores y sus características propias de objeto, Es importante destacar que esta capa está condicionada por el Entity Framework , es decir todo cambio que se realice en base de datos se verá afectado en el modelo .Finalmente en esta capa también se configuran clases tipo Wrapper que permitirán la modificación del Json de Response según el request del front lo requiera
- **Capa de integración:** capa que tendrá la función de integrar la información de la lógica de negocio con la lógica externa del api de Google, se encargará de realizar peticiones de tipo Rest a este sistema remoto y permitirá obtener datos de geolocalización y distancia de un punto A a un punto B, es decir, entre un comedor y un proveedor de alimentos, esto con el objetivo de que el algoritmo de eficiencia necesita obtener la distancia en tiempo real de las dos entidades para obtener a mejor opción y que la misma sea entregada al usuario.
- **Capa de seguridad:** esta capa posicionada al mismo nivel que los controladores, recibirá las credenciales de usuario introducidas en la página de login y verificara que el mismo si exista en el sistema con las credenciales obtenidas, en caso positivo se le enviará un Bearer Token al front que tendrá vigencia de 24 horas y permitirá acceder a este usuario a las características del sistema de “Comedores”.

Finalmente se plantea la parte de persistencia, este componente permitirá almacenar los datos de las entidades que conforman el sistema y permitirá al nivel de Lógica el acceso a los mismos para la ejecución de funcionalidades y servicios. Este funcionara en conjunto con el Backend y EntityFramework para extraer de manera eficiente los datos de base de datos y utilizarlos en los procesos de lógica de negocio.

● 5.4.7.2 Arquitectura De despliegue:

Se desarrollará el servidor de lógica empleando un servidor de Azure remoto, que a través de configuración, ofrecerá una capa de servicios para el acceso a la funcionalidad interna, esto será realizado mediante un IIS, el servidor permitirá la subdivisión y encapsulamiento de requests según la clasificación de los mismos subdividiendo las tareas y aliviando carga del mismo dicha división vendrá identificada por las entidades encargadas del correcto funcionamiento de la lógica de negocio(comedores, proveedores, productos).

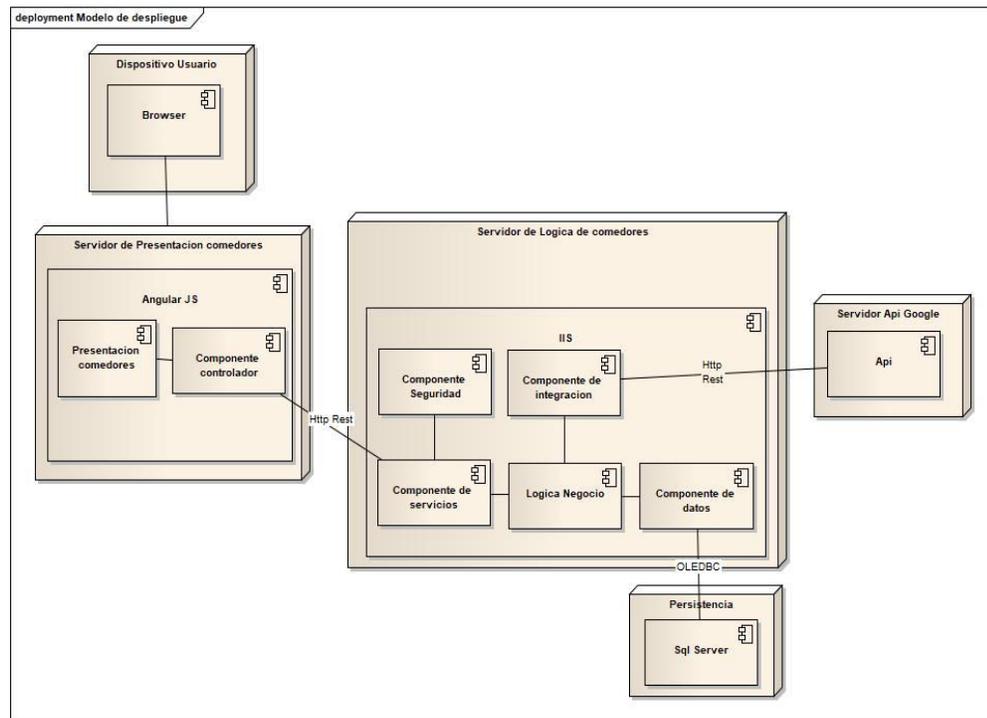


Imagen 8 Arquitectura de despliegue

La mayor ventaja de usar un server de Azure es que permiten escalabilidad mediante el uso de múltiples servidores espejo dependiendo de la demanda de requests realizadas por los usuarios, incrementando el número de máquinas según el número de usuarios, esto será eficiente para el manejo de la parte de lógica del sistema y permitirá la asignación dinámica de requests según la carga de los servidores. El servidor contará un sistema operativo de Windows server 2012 ya que permite una configuración sencilla tipo Plug and play que permite la rápida configuración de servicios que hayan sido desarrollados en tecnología Microsoft.

En la parte del frontend será realizada mediante un aplicativo web y será posible ejecutarla en cualquier dispositivo de usuario esto debido a que se empleará framework Angular JS que es altamente adaptativo a dispositivos, las peticiones se realizarán desde el Browser de sus dispositivos, el browser enviará peticiones según los usuarios a este componente y serán ejecutadas mediante peticiones Promises a través de HTTP/Rest al servidor lógico remoto de IIS.

Adicionalmente se presenta la conexión con un sistema externo, de geolocalización de Google que permitirá identificar la distancia entre el comedor y el proveedor de alimentos, Se realizarán Request de tipo Http-Rest para la conexión y respuesta con el api público de este componente.

finalmente, la persistencia será manejada mediante una base relacional Windows server SQL que permitirá la conexión mediante OLEDB al servidor del IIS para obtener la data necesaria para las ejecuciones de procesos de negocio, esta conexión se realizará de manera automática empleando EntityFramework y Linq.

- **5.4.8 Tecnologías usadas:**

A partir de la arquitectura planteada con anterioridad a continuación se explican las tecnologías empleadas para la resolución de la problemática:

- **Componente Visual:** la parte visual fue desarrollada con Framework de javascript Angular JS ya que dicho *framework* usa lenguaje *Typescript*, facilitando la legibilidad de Código y permitiendo simular el desarrollo a una implementación orientada a objetos, estandarizando la lectura de y entendimiento del código empleado para la construcción del aplicativo WEB, de la misma manera facilita modificaciones o cambios dentro de la implementación de una manera ágil y eficaz. Dicho framework permite convertir los componentes en componentes web nativos permitiendo utilizarlos de nuevo en otras aplicaciones, de esta manera permite crear etiquetas HTML autocontenidas, reutilizables y personalizables. Estos componentes personalizados funcionarán en navegadores modernos y con cualquier biblioteca o *framework* de JavaScript que funcione con HTML[32].

A nivel de la plataforma “Comedores” se utilizaron patrones observer para que el usuario envíe peticiones al Front desde el browser de preferencia y ejecute las acciones necesarias para interactuar con la plataforma, El front se comunica de igual manera con el backend a partir de controladores que simularan las entidades dentro del negocio y separara carga de trabajo de las mismas según el request enviado, estos se comunicaran a partir de protocolo HTTP y empleando REST.

- **Componente De lógica:** la parte lógica se implementó utilizando .NET *framework*, más concretamente C# esto debido a que permite codificación orientada a objetos, dejando que cada objeto sea representado con sus atributos y métodos propios además del uso del encapsulamiento para restringir visibilidad de estos a lo largo del proyecto. C# permite también mantener múltiples versiones de clases almacenadas en forma binaria, de esta manera, permite la ejecución de versiones nuevas y viejas de manera simultánea. De igual manera se eligió debido a que las demás tecnologías tanto Front como base de datos y despliegue en nube se eligieron teniendo en cuenta tecnologías de Microsoft.

En conjunto con C# se emplearon otras tecnologías que son parte del *framework* que permiten mejorar el desarrollo, la estandarización del mismo y la velocidad de implementación, a continuación, se describe cada uno:

- **Entity Framework:** es una tecnología probada y certificada de Microsoft, en conjunto con C# y MS SQL permiten que este ORM se ejecute de manera adecuada y supla las necesidades de los desarrolladores, Hoy se mantiene en constante soporte y mejora, permite la transmisión de datos de la base de datos al modelo lógico en el *backend* de una manera sencilla optimizando líneas de Código y representando el objeto a modo de espejo en ambos extremos.[33]

Se elige EntityFramework debido a que la plataforma se está desarrollando en su mayoría utilizando tecnologías Microsoft a demás al emplear el lenguaje C# y base de datos SQL server permite mejor adopción de esta tecnología como base para el transporte de datos, además permite un rápido reflejo a modo de espejo de las entidades de comedores almacenadas en base de datos al componente de lógica de negocio permitiendo realizar consultas en una sola línea de código.

- **Linq:** es una herramienta de consulta de extensiones integradas en lenguaje C# que permite trabajar de manera ágil y cómoda a través de colecciones de datos, emulando una base de datos, esta misma herramienta permite ejecutar inserciones, modificaciones, eliminaciones y lecturas en una sola línea de Código, ahorrando tiempo en desarrollo [34].

Se emplea Linq dentro de comedores como tecnología ya que se relaciona fácilmente con otras tecnologías de Microsoft y permite agilizar las consultas sobre objetos de la base de datos como si se trataran de listas, agilizando el transporte a datos y a objetos con grandes cantidades de campos como lo son la disponibilidad o los comedores dentro del modelo de negocio de la plataforma.

- **Componente de persistencia:** en cuanto a la tecnología usada para persistir la información se elige Microsoft SQL, esto debido a que las tecnologías que se están empleando para todo el desarrollo son de Microsoft por ende se logra mejor concordancia entre componentes y entre transmisión de datos así mismo como la incorporación de *Entity Framework* y *Linq*, de igual manera se elige una base de datos estructurada por la experticia de los desarrolladores con este tipo de bases de datos.

Cabe destacarse que se planteó el desarrollo usando tecnologías de Microsoft porque se plantea realizar el despliegue a tecnologías Cloud de Azure, de esta manera por conveniencia de la exportación de data y de tecnologías se plantea esta solución.

5.5 Pruebas:

En esta sección se presentarán las pruebas efectuadas al sistema durante el desarrollo y después del mismo.

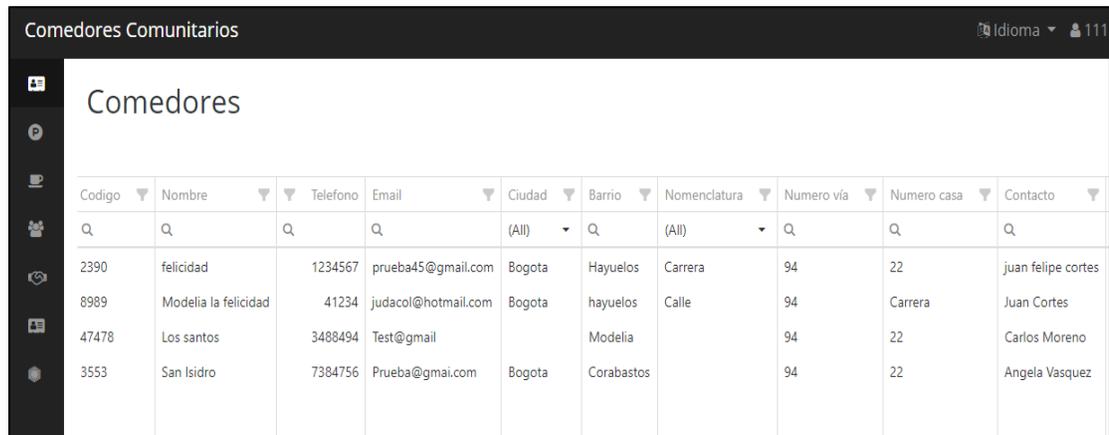
5.5.1 Pruebas funcionales:

Para las pruebas funcionales se contemplaron priorización de requerimientos, servicios, atributos de calidad, casos de uso y procesos reales que deberán efectuar cada uno de los usuarios.

5.5.1.1 Visualización del sistema:

A continuación, se presentan las pantallas de la aplicación final permitiendo identificar las funcionalidades, características y procesos de negocio que representan dentro del modelo macro y que son necesarias para el mínimo viable de la aplicación según necesidades del cliente:

- **Visualización de comedores:**

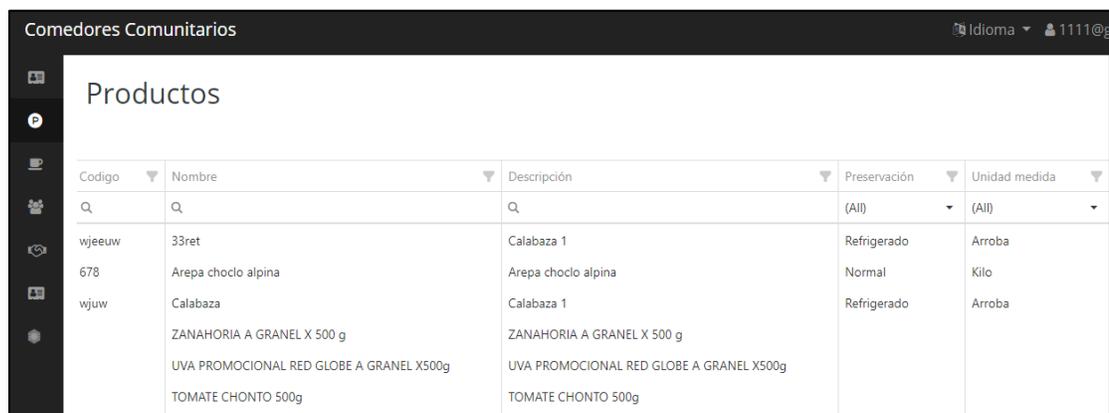


Codigo	Nombre	Telefono	Email	Ciudad	Barrio	Nomenclatura	Numero vía	Numero casa	Contacto
2390	felicidad	1234567	prueba45@gmail.com	Bogota	Hayuelos	Carrera	94	22	juan felipe cortes
8989	Modelia la felicidad	41234	judacol@hotmail.com	Bogota	hayuelos	Calle	94	Carrera	Juan Cortes
47478	Los santos	3488494	Test@gmail		Modelia		94	22	Carlos Moreno
3553	San Isidro	7384756	Prueba@gmai.com	Bogota	Corabastos		94	22	Angela Vasquez

Imagen 9 Comedores

En esta vista se puede apreciar todos los comedores que hacen parte de la plataforma de información, se destaca un código de identificación, un nombre, la ciudad donde se ubica el comedor y datos de contacto de un referente que haga parte del comedor

- **Visualización de Productos:**



Codigo	Nombre	Descripción	Preservación	Unidad medida
wjeeuw	33ret	Calabaza 1	Refrigerado	Arroba
678	Arepa choclo alpina	Arepa choclo alpina	Normal	Kilo
wjuw	Calabaza	Calabaza 1	Refrigerado	Arroba
	ZANAHORIA A GRANEL X 500 g	ZANAHORIA A GRANEL X 500 g		
	UVA PROMOCIONAL RED GLOBE A GRANEL X500g	UVA PROMOCIONAL RED GLOBE A GRANEL X500g		
	TOMATE CHONTO 500g	TOMATE CHONTO 500g		

Imagen 10 Productos

En esta vista se presentan al usuario los productos que se encuentren en sistema se dividen según el proveedor que los oferte variando de esta manera sus precios, pero pudiendo ser el mismo producto. Se presenta el nombre del producto, una descripción breve, el tipo de conservación (Lácteo, Cárnico, entre otros) y la unidad de medida del mismo que permitirá la posterior configuración de los ingredientes para las recetas posteriormente.

- **Visualización Recetas:**

Codigo	Nombre	Unidad medida	Cantidad
23	papa	Arroba	3
32	agua	Arroba	5

Imagen 11 Recetas

En la vista de recetas se podrán visualizar las recetas que se encuentren ligadas a un comedor específico, una vez se digite el comedor y el nombre de la receta aparecerán las recetas que se encuentren almacenadas en base de datos y que pertenezcan a ese comedor, de igual manera será posible visualizar los ingredientes necesarios para elaborar un plato teniendo en cuenta la cantidad en cuanto a unidad de medida del producto.

- **Visualización de proveedores:**

Codigo	NIT	Nombre	Telefono	Email empresarial	Ciudad	Barrio	Nomenclatura	Numero vía	Numero casa
4567		Transporte EL paseo	737373		(All)	1	(All)	15	L
1234		Transporte La economica	366363			1	Carrera	123b	71a-56
0483		BOGOTA LA CANDELARIA					Calle	75b	71a-56
0387		BOGOTA SUR MARICHUELA						14	L
0364		BOGOTA LA ESTANCIA CIUDAD BOLI						75	57R-08
0279		BOGOTA LUCERO BAJO					Calle	75b	71

Imagen 12 Proveedores

En esta vista se podrá visualizar los proveedores alimenticios que se encuentren en la plataforma, se destaca su nombre, código y datos de contacto del mismo como dirección, teléfono y correo

- **Visualización Disponibilidad:**

Proveedores	Productos	Cantidad	Valor unitario	Fecha vencimiento
(All)	(All)	Q	Q	Q
BOGOTA LA CANDELARIA	AREQUIPE ALQUERIA TRADIC + OBLEAS X 320G	4	\$1.234	
BOGOTA LA CANDELARIA	AREQUIPE LIGHT X200g	23	\$946	
BOGOTA LA CANDELARIA	AROMATICA SABORES SURTIDOS x25g	1	\$6.196	
BOGOTA LA CANDELARIA	ARROZ EL COSTAL 3000g	11		
BOGOTA LA CANDELARIA	ARROZ EL COSTAL BOLSA 500g	1		
BOGOTA LA CANDELARIA	ARROZ BLANCO SONORA 3000g	10		
BOGOTA LA CANDELARIA	ARROZ DEL COSTAL PREMIUM 1000g	6		
BOGOTA LA CANDELARIA	ARROZ DIANA BOLSA 3000g	1		

Imagen 13 Disponibilidad

En esta ventana se podrá evidenciar los productos registrados en la plataforma por cada uno de los proveedores, se puede apreciar el nombre, el proveedor que los ofrece, la cantidad a ofertar por parte del proveedor, el valor unitario y la fecha de vencimiento de los productos.

- **Visualización vehículos:**

Proveedores	Codigo	Placa	Marca	Año	Tipo transporte	Conductor	Telefono	Email	Medida pago
(All)	Q	Q	Q	Q	(All)	Q	Q	Q	(All)
Ara	21333	sat789	Audi	1	Cooled	Fernando Ruiz	738384	Abc@gmail.com	
Transporte La economica	3554	jef870	Audi	1		Fernando alonso	87837273	jhfff@gmail.com	Kilometer
Transporte La economica	4343	Bhu789	Nissan	1	Cooled	Carlos Eurobio	63769222	medkfm@gmail.com	Kilometer
Transporte EL paseo	22244	Bgy753	Renault	2		Alejandro Gomez	2345678	whiudh@gmail.com	Kilometer
Transporte EL paseo	22434	Bnu756	Mazda	1	Cooled	Carlos Moreno	12345678	qyet123@gmail.com	Kilometer

Imagen 14 Vehículos

En esta vista se aprecian los vehículos disponibles para el transporte de los productos, todos los vehículos vendrán con el número de placa, el año de producción, los datos de contacto, la marca, el tipo de transporte que puede acarrear, La unidad de cobro (Km) y cuánto cobra por unidad de cobro (ejemplo: 10.000 pesos cada 2 km).

- **Visualización prepedido:**

Comedores Comunitarios

Pre-pedido | Histórico pre-pedido

Comedores: Modela la felicidad

Recetas: Plato Mexicano

Enviar

Ingredientes

Codigo	Nombre	Unidad medida	Cantidad	
Q	Q	(All)	Q	
24	yuca	Arroba	2	Eliminar
3393	Pollo	Arroba	3	Eliminar
23	papa	Arroba	3	Eliminar

Imagen 15 Configuración prepedido

Comedores Comunitarios

Pre-pedido | Histórico pre-pedido

Comedores	Recetas	Fecha	Aceptado	Observación
(All)	(All)	Q	(All)	Q
Modela la felicidad	Plato Mexicano	2020-05-21	<input checked="" type="checkbox"/>	Administrador sin confirmar
Modela la felicidad	Plato Mexicano	2020-05-19	<input type="checkbox"/>	Administrador ha rechazado
Modela la felicidad	Plato Mexicano	2020-05-19	<input checked="" type="checkbox"/>	Administrador confirmardo
Modela la felicidad	Plato Mexicano	2020-05-19	<input checked="" type="checkbox"/>	Administrador confirmardo
Modela la felicidad	Plato Mexicano	2020-05-19	<input type="checkbox"/>	Administrador ha rechazado

Imagen 16 Resumen Prepedido

En esta vista se introduce el comedor y la receta que se desee pedir, en la parte baja se enumerarán los ingredientes y sus cantidades necesarias para el plato, adicionalmente se podrán seleccionar nuevos ingredientes externos al plato que serán presentados en la misma pantalla y podrán ser agregados al pedido total. Cabe destacarse que si se solicita una cantidad de un ingrediente superior a la que se encuentre en existencias el sistema no permitirá realizar el pedido.

Una vez se hayan seleccionado los ingredientes requeridos, el usuario podrá realizar la confirmación del pedido permitiendo presentarlos a modo de histórico en una pestaña al lado de los pedidos, se registrará en ella el nombre del plato, los ingredientes y la fecha de envío.

- **Visualización pedido:**

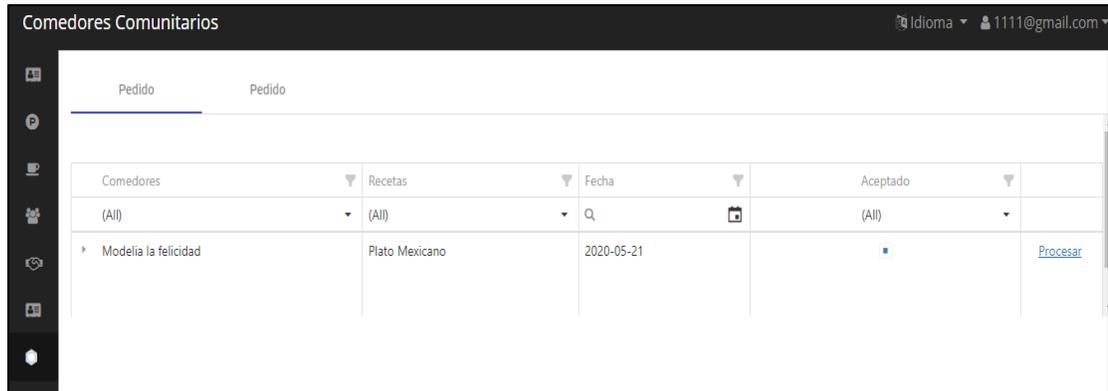


Imagen 17 Configuración Pedido

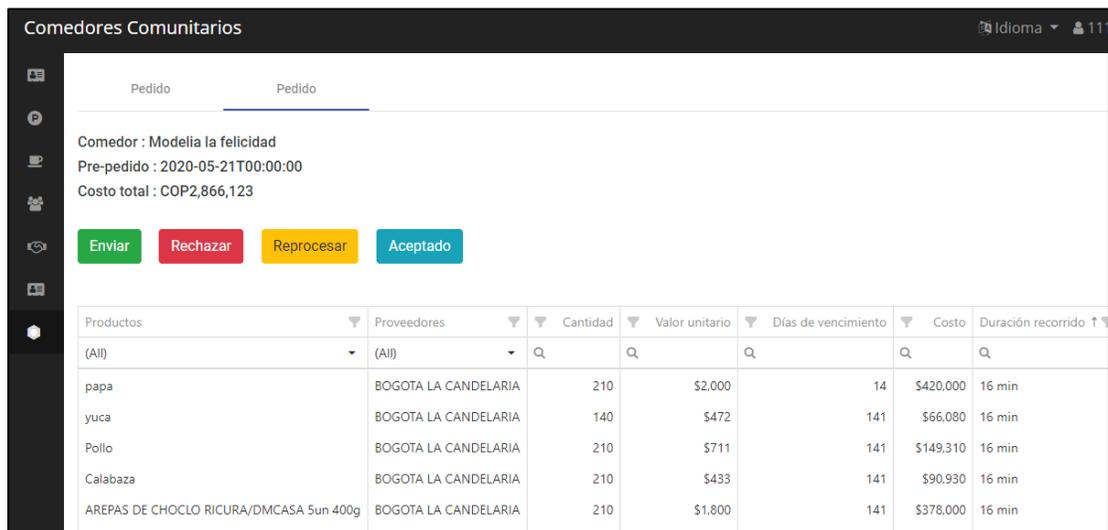


Imagen 18 Resumen Pedido

En la visualización de pedido se destacan los pedidos que fueron solicitados por parte del comedor en la sección de prepedidos, en esta nueva ventana se podrá realizar una confirmación de pedido sobre cada una de las órdenes. Una vez confirmada la orden se procederá a ejecutar el algoritmo de normalización con los datos en base de datos y a continuación se enviarán los datos al algoritmo de Simplex-DEA permitiendo presentar al usuario los productos con sus respectivos proveedores ya optimizados en temas de costo de orden, de igual manera, el sistema realizará el cálculo necesario para elaborar el número de platos para todos los niños ligados a un comedor, agilizando de esta manera, el cálculo manual que se realizaba.

En la pantalla final se puede observar los productos, proveedores, cantidad final, costo de producto por proveedor, transporte más eficiente que pueda transportar dicho producto, la distancia y el costo final del pedido totalizado.

Cada componente y vista fueron probados siguiendo los casos de uso estipulados anteriormente en este documento teniendo en cuenta las entradas y salidas del sistema y revisando la consistencia de los datos de éste.

Es importante destacar que el sistema presentara los productos optimizados en el DEA de acuerdo con lo siguiente: Se maximizara fecha de vencimiento y cantidad y se minimizara costo y distancia, de igual manera si el proveedor óptimo no posee la disponibilidad completa para suplir la necesidad en el pedido, se ejecutara otro DEA y se presentara el siguiente proveedor óptimo hasta suplir la necesidad del pedido.

En caso de cancelaciones por parte del proveedor de alimentos o de transporte, se procede a realizar otra ejecución del algoritmo de DEA presentando las siguientes opciones óptimas en cuanto a transporte y producto marcando las opciones de proveedores que rechazaron como no utilizables dentro de esta ejecución.

5.5.2 Pruebas unitarias:

Para la ejecución de las pruebas unitarias se procede a la identificación de un componente individual específico dentro del sistema, el cual será sometido a requests enviando parámetros a los controladores y obteniendo respuestas, revisando la consistencia de éstas y revisando que los procesos den como resultado una modificación al interior de la base de datos después de la ejecución.

Para evaluar dichos procesos por componente se procede a utilizar la herramienta Postman[36] que permite realizar requests Rest a las controladoras del backend, permitiendo modificar de igual manera los tipos de petición.

Finalmente se cotejaban los datos enviados por petición con los resultados en la base de datos para descartar errores de consistencia o de error en data.

Es importante recalcar que todas las controladoras están protegidas por JWT por motivos de seguridad es por ello que es indispensable enviar peticiones enviando el Bearer Token que se obtiene cuando se realiza el proceso de Login a nivel de usuario, en caso contrario los controladores no responderán a la petición del usuario.

Adicionalmente se realizaron tareas de depuración puntuales a nivel de ejecución de controlador permitiendo revisar línea a línea el código que ha sido implementado y si los cálculos o procesos son acordes a lo esperado.

5.5.3 Pruebas de usabilidad:

Las pruebas de usabilidad se efectuaron una vez el sistema estuvo completo en un 95% del producto mínimo viable y se ejecutaron con los Alejandro Girón, miembro del equipo de ingeniería industrial y la doctora Clara Mabel Solano.

El procedimiento se ejecutó desde el ingreso al sistema pasando por la creación de cada una de las entidades de sistema, desde productos hasta órdenes de pedidos, se explicó cómo se desarrolló el algoritmo de eficiencia y las variables de maximización (disponibilidad, fecha de vencimiento) y las variables de minimización (costo, distancia y tiempo) y se realizó el flujo completo de proceso incluyendo la formulación de recetas y la aceptación y cancelación de pedido.

Finalmente se exportó la data obtenida de la orden más eficiente en cuanto a costos en un Excel y se modificó dicho Excel para que simulara un reporte de consumo.

Se aprecia que el usuario puede acceder a cualquier servicio ofrecido por el sistema en un mínimo de 5 clics desde el menú de opciones.

Los detalles completos de la prueba de usabilidad con los stakeholders se encuentran disponibles en el documento de pruebas adjunto a esta memoria.

5.5.4 Pruebas de tiempo de respuesta

Una vez creadas las clases controladoras se procede a realizar peticiones empleando la interfaz de usuario del frontend con la finalidad de calcular tiempos de respuesta y latencia en el sistema de comedores, cada request se ejecutó 5 veces y con distintos datos, en la tabla de abajo se observan los promedios de las mediciones en cuanto a tiempos de respuesta.

Entidad / acción	Cantidad de campos	Operación	Tiempo	Respuesta	Descripción	Cantidad de registros
Comedor	12	Insert	176 ms	Insertado	Inserción de registro	1
		update	186 ms	Actualizado	actualización de registro	1

		Get	56 ms	Obtenido	obtención de registro	1
		GetAll	80 ms	Obtenidos	Obtención de registros	12
		Delete	138 ms	Eliminado	Eliminación de registro	1
Producto	6	Insert	168 ms	Insertado	Inserción de registro	1
		update	83 ms	Actualizado	actualización de registro	1
		Get	124 ms	Obtenido	obtención de registro	1
		GetAll	136 ms	Obtenidos	Obtención de registros	2018
		Delete	26 ms	Eliminado	Eliminación de registro	1
Provider	12	Insert	109 ms	Insertado	Inserción de registro	1
		update	130 ms	Actualizado	actualización de registro	1
		Get	25 ms	Obtenido	obtención de registro	1
		GetAll	125 ms	Obtenidos	Obtención de registros	15
		Delete	25 ms	Eliminado	Eliminación de registro	1
Transport	13	Insert	223 ms	Insertado	Inserción de registro	1
		update	130 ms	Actualizado	actualización de registro	1
		Get	27 ms	Obtenido	obtención de registro	1
		GetAll	113 ms	Obtenidos	Obtención de registros	8
		Delete	25 ms	Eliminado	Eliminación de registro	1
Recipe	3	Insert	217 ms	Insertado	Inserción de registro	1
		update	101 ms	Actualizado	actualización de registro	1
		Get	15 ms	Obtenido	obtención de registro	1
		GetAll	96 ms	Obtenidos	Obtención de registros	8
		GetRecomendedRecipes	108 ms	Obtenidos	Obtención de recetas recomendadas	8
		GetAllIngredientsFromRecipe	105 ms	Obtenidos	Obtención de ingredientes por receta	10
		Delete	19 ms	Eliminado	Eliminación de registro	1

Ingredient	4	Insert	215 ms	Insertado	Inserción de registro	1
		update	120 ms	Actualizado	actualización de registro	1
		Get	21 ms	Obtenido	obtención de registro	1
		GetAll	99 ms	Obtenidos	Obtención de registros	2050
		Delete	26 ms	Eliminado	Eliminación de registro	1
System User	7	Insert	190 ms	Insertado	Inserción de registro	1
		update	99 ms	Actualizado	actualización de registro	1
		Get	26 ms	Obtenido	obtención de registro	1
		GetAll	105 ms	Obtenidos	Obtención de registros	6
		Delete	32 ms	Eliminado	Eliminación de registro	1
Disponibility	6	Insert	183 ms	Insertado	Inserción de registro	1
		update	112 ms	Actualizado	actualización de registro	1
		Get	28 ms	Obtenido	obtención de registro	1
		GetAll	116 ms	Obtenidos	Obtención de registros	2058
		Delete	26 ms	Eliminado	Eliminación de registro	1
Order	16	Insert	200 ms	Insertado	Inserción de registro	1
		update	150 ms	Actualizado	actualización de registro	1
		Get	42 ms	Obtenido	obtención de registro	1
		GetAll	400 ms	Obtenidos	Obtención de registros	35
		Delete	39 ms	Eliminado	Eliminación de registro	1
		Process	3 seg	Procesada	procesar orden	1
		Process rejected	4 seg	Procesada	reprocesar orden	1
		Matriz Distance	1 seg	Distancia	Distancia Obtenida	1
		Accepted	235 ms	Aceptada	orden aceptada	1
		Canceled	201 ms	Rechazada	orden rechazada	1
order item	7	Insert	220 ms	Insertado	Inserción de registro	1

		update	115 ms	Actualizado	actualización de registro	1
		Get	32 ms	Obtenido	obtención de registro	1
		GetAll	120 ms	Obtenidos	Obtención de registros	569
		Delete	31 ms	Eliminado	Eliminación de registro	1
		GetOrderItemByTransport	45 ms	Obtenido	Obtención de order item por transporte	12
		GetOrderItemByProvider	51 ms	Obtenido	Obtención de order item por proveedor	12
		GetOrderItemByPreorder	42 ms	obtenido	Obtención de order por preorder	16
		ResponseProvider	36 ms	Obtenido	Modificar respuesta de proveedor	1
		ResponseTransport	32 ms		Modificar respuesta de Transporte	1
Pre order	7	Insert	186 ms	Insertado	Inserción de registro	1
		update	109 ms	Actualizado	actualización de registro	1
		Get	32 ms	Obtenido	obtención de registro	1
		GetAll	102 ms	Obtenidos	Obtención de registros	452
		Delete	34 ms	Eliminado	Eliminación de registro	1
		GetPreorderByDinningRoom	102 ms	Obtenido	Obtenido por comedor	1
Token	3	Authenticate	97 ms	Autenticado	Usuario autenticado	1

Tabla 2 medida de respuesta de requests

Se puede apreciar que ningún request supera los 4 minutos de latencia desde un request del frontend hacia el backend y su posterior response, esto concuerda con los atributos de calidad en cuanto a tiempos de respuesta y requerimientos no funcionales, estos últimos detallados en el SRS y especificados en el SAD.

Finalmente se concluye que el tiempo de respuesta es óptimo y no afectara la experiencia de usuario del cliente permitiendo una respuesta rápida y eficaz con cada uno de los request mostrados en la tabla de arriba.

6.0 CONCLUSIONES

6.1 Lecciones aprendidas:

En primer lugar, debido al objetivo general planteado en este proyecto se hace indispensable tener contacto constante con los stakeholders y los intermediarios de estos que corresponde al equipo de Ingeniería industrial y la doctora Clara Mabel Solano, esto para tener realimentación constante acerca de los avances del proyecto y de las entregas parciales las cuales se fueran adaptando a las necesidades del usuario final y, en primera instancia, a la resolución de la problemática expuesta.

La metodología Scrum permitió de una manera paralela el avance tanto con la parte de desarrollo de la plataforma como con las entregas parciales y final hacia los stakeholders; dicha metodología ágil permitió subdividir responsabilidades para los desarrolladores y orientar el esfuerzo hacia el cumplimiento de objetivos durante cada Sprint. De igual manera, permitió analizar los puntos más críticos a nivel de desarrollo siendo en el caso de comedores la normalización de variables para disminuir diferencias de cálculos y el algoritmo de optimización ya que se tenían que plasmar modelos matemáticos a nivel de programación cumpliendo con las métricas y estándares entregados por el equipo de Ingeniería Industrial.

La realimentación inicial acerca de la problemática expuesta fue indispensable para comprender qué procesos y modelos de negocios emplean Vision Mundial y el Banco de Alimentos para tratar de automatizar y simplificar procesos recurrentes y que demandaran mucho tiempo a los funcionarios, de la manera en que la logística sea desarrollada por el empleado en cuanto a pedidos y productos necesarios para el domicilio, pero que los cálculos y opciones óptimas fueran ejecutadas según estándares y lineamientos dados en el modelo de optimización por el sistema de información.

Es indispensable recalcar que la interfaz se planteó lo más sencilla posible para facilitar el manejo de la herramienta a cualquier usuario y que no requiera mucho entrenamiento para usarla; por ello, se estandarizó la visualización como un panel de menú que en menos de tres clics puede ofrecer al usuario todos los servicios implementados y que los mismos sean presentados de una manera atractiva y entendible.

Un factor esencial fue seguir el cronograma según lo propuesto y que, de esta manera, se lograra ejecutar el proyecto con los servicios planteados en los objetivos en el tiempo propuesto.

Adicionalmente cabe destacar que el equipo de trabajo aprendió el uso de nuevas herramientas de .Net en cuanto a la capa de extracción de datos con el respectivo EntityFramework y Linq agilizando el manejo de los datos y del modelo de negocio así como del módulo de presentación en cuanto al control de componentes de manera individual a través de la tecnología del framework de JavaScript Angular JS, permitiendo que el aplicativo fuera *responsive* independientemente del dispositivo donde se ejecute.

Finalmente, cabe destacarse que el modelo de base de datos que se hizo en las primeras instancias de la fase de planeación permitió estructurar la data de una manera eficiente y que no

generara muchos cambios durante la fase de planeación. Así mismo, el uso de tecnologías en nube específicamente Azure permite que aspectos como escalabilidad y disponibilidad se mantengan de una manera eficiente para el uso del sistema de información de comedores.

6.2 Impacto

A continuación, se muestra como impactos comedores, teniendo en cuenta la problemática expuesta de manera tecnológica y social:

6.2.1 Impacto social

Según el objetivo general planteado, el Core del proyecto “Comedores” es permitir agilizar procesos de pedidos a los comedores a través de una herramienta tecnológica que, de manera indirecta, ayude a mitigar las condiciones de hambre que, muchas personas que acuden a estas instituciones, padecen, permitiendo a los niños obtener una alimentación balanceada planteada por los mismos comedores y que, de igual manera, ayude a minimizar costos en cuanto a procesos de los stakeholders de Vision Mundial y el Banco de Alimentos de manera que el dinero que se ahorre permita contribuir de otra manera a estas personas.

“Comedores” está diseñado para contribuir a mitigar el índice de desperdicios de alimentos que se sufre año tras año por parte de los proveedores, permitiendo que a través de la plataforma registren sus productos que no hayan podido ser vendidos y que, en lugar de desecharlos, ayuden a alimentar a personas con bajos recursos.

Finalmente se espera que esta herramienta pueda servir de base para agregar nuevas contribuciones a futuro y que permita seguir ayudando a las personas más necesitadas a través de una herramienta tecnológica que queda a disposición de los stakeholders para dar apoyo y ayuda a quien lo necesite.

6.2.2 Impacto Tecnológico

A partir de las pruebas ejecutadas tanto internamente como externamente se concluye que “Comedores” ofrece a los stakeholders las opciones óptimas en cuanto a pedidos de alimentos y sus respectivos domicilios a través del apoyo de tecnologías de la información y gestión, con la finalidad de que el proceso que se realiza a día de hoy por los interesados se optimice y se reduzca el esfuerzo de cálculo y estimación según la necesidad de proveer productos alimenticios a las personas necesitadas.

La plataforma ha sido desarrollada de igual manera usando tecnologías modernas y entendibles para la mayoría de los programadores para que pueda ser revisada, modificada y mejorada a futuro, empleando tecnologías como .Net y angular que actualmente se encuentran en auge y que corresponden al modelo de desarrollo actual de orientación a objetos. Adicionalmente, la arquitectura empleada para la construcción de “Comedores” permitió subdividir la lógica de negocio y estandarizar clases de desarrollo a través de capas lo cual generaba delegación de responsabilidades en los componentes de la arquitectura, cada capa es un componente único y especializado dentro de la arquitectura que se especializa en ejecutar acciones sobre una clase en específico; es por ello, que si se quiere añadir nueva funcionalidad o nuevos objetos basta

con crear una controladora nueva o modificar la capa de negocio de algún objeto. Adicionalmente, una de las ventajas de emplear Rest es que el desarrollo del front y backend se pueden realizar de manera independientemente porque el protocolo de comunicación ya está estandarizado y basta con sólo ingresar nuevas firmas de los métodos a desarrollar, es por ello, que será fácilmente mantenible y modificable.

Es importante mencionar que la herramienta utiliza tecnologías en la nube de Microsoft de Azure con lo cual se plantea que pueda ser accedida desde cualquier parte y que problemáticas como la seguridad, la disponibilidad y la mantenibilidad sean en parte cubiertas por la infraestructura de uno de los gigantes de la informática hoy en día.

Adicionalmente, cabe destacar que el modelo de optimización planteado en conjunto con el equipo de ingeniería Industrial liderado por Alejandro Girón y Clara Mabel Solano fue traducido a nivel de implementación a través de la construcción de pseudocódigo y posteriormente su traducción a C# , lo anterior teniendo en cuenta las variables más significativas para el negocio y su funcionalidad dentro del proceso (Costo, Distancia, Tiempo, Disponibilidad). Sin embargo, para estandarizar el proceso de optimización y contemplando que las variables podían ser muy desiguales entre ellas, se realizó una normalización de medidas para estandarizar la escala de medida y permitir la ejecución de un algoritmo Simplex-DEA de una forma más precisa.

Posteriormente el resultado obtenido de la ejecución del algoritmo se mostraba al usuario en una interfaz amigable y entendible con el usuario evidenciando costos y tiempos del producto y del transporte, demostrando al usuario el resultado del proceso del algoritmo de eficiencia.

Todo el código y la misma herramienta queda a disposición de los stakeholders y de la Universidad Javeriana con la intención de que, en el futuro, sea mejorada y permita suplir mayor número de necesidades de los comedores comunitarios.

6.3 Conclusiones

En esta sección se exponen las conclusiones acerca de los objetivos planteados para el trabajo de grado:

6.3.1 Conclusión General:

La ejecución del trabajo de grado y su conclusión fueron posibles gracias a las estrategias y metodologías plateadas durante cada fase del proceso y la realimentación constante del cliente y de cada uno de los stakeholders, Lo anterior finalizó con el resultado de la plataforma “Comedores”, una herramienta que otorga a los usuarios la posibilidad de realizar pedidos óptimos en cuanto a costo de productos y de envío a los comedores teniendo en cuenta costo, fecha de vencimiento y disponibilidad, permitiendo paralelamente agilizar procesos de cálculo y minimizando costos y esfuerzo.

La elección de herramientas fue la adecuada ya que permitió el desarrollo del producto en el tiempo establecido y, según los resultados en pruebas, la elección correcta de los productos y proveedores tanto alimenticios como de transporte óptimos. En cuanto a implementación se

puede apreciar que el despliegue de la misma fue realizado de una manera eficiente y sencilla dadas las medidas de desarrollo tomadas, teniendo en cuenta patrones, diseños y modelos arquitectónicos, dividiendo los componentes de visualización, lógica de negocio y acceso a datos permite que los componentes sean de cierta forma auto-sostenibles y que puedan destinarse a futuro, servidores dedicados a cada uno de estos niveles con la finalidad de otorgar mayor disponibilidad al sistema; adicionalmente, la comunicación entre los mismos fue eficaz a través de Rest permitiendo estandarizar intercambio de data.

El desarrollo del algoritmo de eficiencia fue el más complicado de lograr debido a que existían gran cantidad de variables a tener en cuenta y unas debían minimizarse y otras maximizarse según el contexto del pedido; sin embargo, dicha problemática fue solucionada siguiendo los modelos matemáticos planteados y las fórmulas establecidas. De igual manera, el margen de error fue disminuido con la aplicación del algoritmo de normalización permitiendo posicionar las métricas de estas variables en una escala común.

Finalmente, a través de las pruebas con el equipo de Ingeniería Industrial se observó que el sistema si presenta las elecciones en cuanto a producto y transporte óptimos, permitiendo el uso de la herramienta de una manera confiable y segura y que permitirá maximizar eficiencia en el proceso de pedido y minimizar esfuerzo por parte del usuario final.

6.3.2 Conclusiones específicas

Se obtuvo información acerca de la problemática actual, destacando cifras y modelos de proceso que se trata actualmente a nivel de pedidos por parte de los comedores; de igual manera, se obtuvo información del modelo de optimización por parte del equipo de Ingeniería Industrial, obteniendo una primera aproximación a lo que se esperaba que se obtuviera por parte del algoritmo. Una vez teniendo claro el proceso de ejecución de dicho algoritmo se procede a establecer la metodología que se seguirá para adaptarlo a nivel de código teniendo en cuenta características y restricciones matemáticas.

Posteriormente, se planteó un algoritmo que suple dichas necesidades, aunque sufre de errores debido a la escala que maneja el modelo de Simplex. Dentro de las mismas se destaca que al poseer medidas de valor 0 o 1 el algoritmo mostraba siempre dichos datos independientemente de las demás variables, así mismo al realizar un proceso entre variables tan dispares en cuanto a medidas el proceso de cálculo dentro de la maquina se demoraba demasiado tiempo en cálculo de eficiencia. Por ello, se contempló desarrollar un algoritmo de normalización y establecer las variables en una sola escala de medición, permitiendo suplir las deficiencias del algoritmo.

De igual manera, se plantearon los servicios, requerimientos, patrones y arquitectura a seguir según el modelo. De lo anterior, se establecieron los patrones de multinivel y capas ya que permiten el desarrollo independiente de la parte visual y la parte de lógica de negocio, facilitando el desarrollo y permitiendo implementar de manera modular.

Una vez se empiezan a ejecutar pruebas se destaca que se debía tener en cuenta un método de control cuando el pedido del producto fuera parcializado y no completo, es decir, que un proveedor no tenga la cantidad necesaria para suplir la necesidad pero a la vez, es el óptimo; por ello, se contempló elegir este proveedor y relanzar el algoritmo de eficiencia para obtener el

siguiente óptimo y suplir la necesidad en su totalidad siempre teniendo en cuenta los valores ordenados del resultado del algoritmo de optimización que maximiza la fecha de vencimiento y la cantidad, minimizando el costo y la distancia del proveedor al comedor.

Posteriormente se unificaron la parte visual y la parte lógica; esto fue posible gracias al modelo arquitectónico establecido en la planeación y que permitió desarrollar ambos componentes de una manera independiente basándose únicamente en un archivo Json para la información. Adicionalmente, el componente del front desarrolla la subdivisión de roles dentro de la aplicación permitiendo acceder a funcionalidades únicas dentro de la misma según los siguientes perfiles: comedor, administrador, proveedor de transporte y proveedor de alimentos. A estos se les presenta la información de manera diferente a través de un identificadorr único de perfil y de la misma interfaz de visualización.

A continuación, se realizaron pruebas con usuarios finales, en este caso, el equipo de Ingeniería Industrial, se obtuvo observaciones y se corrigieron detalles de implementación y visualización. Sin embargo, el proceso necesario para suplir y mitigar la problemática se establece como el correcto y se logra el producto mínimo viable.

6.3 Trabajo futuro

Comedores deja a disposición una versión 1.0 que puede ser ampliada y que deja a disposición de los dueños posibles modificaciones o mejoras que requieran ser desarrolladas a futuro.

En primer lugar, se pueden agregar nuevos proveedores tanto alimenticios como de transporte a la base de datos, esto debido a que hoy en día sólo se tiene a ARA y sus distintas seccionales a nivel Bogotá. De esta manera, el algoritmo de optimización podrá elegir entre mayor cantidad de opciones, mejorando de esta forma la variedad y posibles resultados a futuro. De igual manera, debe permitir la diversificación de productos y de vehículos. Comedores permite esta mejora desde su interfaz de usuario sin necesidad de modificar lógica de negocio o base de datos.

Otro aspecto que podría ser mejorado es el algoritmo de optimización; si bien hoy en día los resultados son los esperados se pueden codificar otras opciones diferentes a Simplex-DEA y que permita un cálculo más preciso y eficiente a medida que el número de proveedores, productos y transportes aumente.

Finalmente, se plantea la opción de desarrollar un módulo de geolocalización que permita al usuario consultar en qué parte de la ciudad va su pedido en tiempo real, permitiendo monitorear el proceso de domicilio.

REFERENCIAS

- [1] C. E. E. Tiempo, «Desperdicio de 34 % de alimentos mientras hay 27 % de gente en pobreza», El Tiempo, oct. 11, 2019. <https://www.eltiempo.com/economia/sectores/descomunal-desperdicio-de-alimentos-en-colombia-pais-con-27-de-gente-en-la-pobreza-422048> (accedido may 10, 2020).
- [2] C. Radio, «En más del 60% aumentan los casos de desnutrición en niños de Bogotá», Caracol Radio, jun. 11, 2019. https://caracol.com.co/emisora/2019/06/11/bogota/1560258569_574673.html (accedido may 10, 2020).
- [3] Marketing4Food, «Cadena de distribución», Marketing y Publicidad Alimentos, Bebidas y Gran Consumo. <https://marketing4food.com/glosario/cadena-de-distribucion/> (accedido may 10, 2020).
- [4] Autor desconocido, Modelos de optimización «Matemática y negocios». http://gc.inite-labs.com/recursos/files/r157r/w13107w/MateNegocios_unidad%202.pdf (accedido may 10, 2020).
- [5] Pedro Halcarteray «Qué es un modelo de optimización?», Blog - Notus, oct. 02, 2018. [/que-es-un-modelo-de-optimizacion/](http://que-es-un-modelo-de-optimizacion/) (accedido may 10, 2020).
- [6] Autor desconocido «¿Qué es Conservación de Alimentos? » Su Definición y Significado [2020]», Concepto de - Definición de. <https://conceptodefinicion.de/conservacion-alimentos/> (accedido may 10, 2020).
- [7] Accion contra el hambre «¿Qué es el hambre?» <https://www.accioncontraelhambre.org/es/te-contamos/blog-testimonios/que-es-el-hambre> (accedido may 10, 2020).
- [8] María Estela Raffino «Sistema de Información: Concepto, Tipos, Elementos y Ejemplos». <https://concepto.de/sistema-de-informacion/> (accedido may 10, 2020).
- [9] Humberto Cervantes «Arquitectura de Software», SG Buzz. <https://sg.com.mx/revista/27/arquitectura-software> (accedido may 10, 2020).
- [10] Nicole Malde «Qué es Frontend y Backend». <https://platzi.com/blog/que-es-frontend-y-backend/> (accedido may 10, 2020).

- [11] Retail Inventory «Retail Inventory», App Store. <https://apps.apple.com/us/app/retail-inventory/id440020325> (accedido may 10, 2020).
- [12] ABC Inventory «Free ABC Inventory Software Features». http://almyta.com/abc_inventory_software.asp (accedido may 10, 2020).
- [13] PartKeepr «PartKeepr - Open Source Inventory Management». <https://partkeepr.org/> (accedido may 10, 2020).
- [14] Openbravo «Accelerate End-to-End Omnichannel Execution and Innovation in the Cloud», Openbravo. <https://www.openbravo.com/> (accedido may 10, 2020).
- [15] Grubhub «Grubhub food delivery ». <https://grubhub.com> (accedido may 10, 2020).
- [16] DoorDash «DoorDash Food Delivery & Takeout - From Restaurants Near You». <https://www.doordash.com/en-US> (accedido may 10, 2020).
- [17] «Postmates: Food Delivery, Groceries, Alcohol - Anything from Anywhere». <https://postmates.com/> (accedido may 10, 2020).
- [18] Nestle «Recetas de cocina, trucos y consejos | Nestlé Cocina». <https://www.nestleco-cina.es> (accedido may 10, 2020).
- [19] Autor Desconocido «Recetario, recetas de cocina - Aplicaciones en Google Play». <https://play.google.com/store/apps/details?id=com.villy.recetario&hl=es> (accedido may 10, 2020).
- [20] Banco de Alimentos. (2018). Banco de Alimentos. Obtenido de <http://www.bancodealimentos.org.co/como-vincular-tu-organizacion/>(accedido may 10, 2020).
- [21] Benavente, R., & Cornejo, C.. Plan de ruteo para la distribución de ayuda humanitaria no alimentaria ante un terremoto de gran magnitud en Lima Metropolitana y Callao. LACCEI. (2017) (accedido may 10, 2020).
- [22] DANE.DANE. «censo nacional de población y vivienda 2018» <https://www.dane.gov.co/index.php/estadisticas-por-tema/demografia-y-poblacion/censo-nacional-de-poblacion-y-vivenda-2018/cuantos-somos>(accedido may 10, 2020).

- [23] DAPRE. Presidencia de la República. «Ley1990» <https://dapre.presidencia.gov.co/normativa/normativa/LEY%201990%20DEL%2002%20DE%20AGOSTO%20DE%202019.pdf> (accedido may 10, 2020).
- [24] DNP. Departamento Nacional de Planeación. «Interés Perdida y desperdicio de alimentos» https://mrv.dnp.gov.co/Documentos%20de%20Interes/Perdida_y_Desperdicio_de_Alimentos_en_colombia.pdf (accedido may 10, 2020).
- [25] Autor Desconocido «Programación Lineal: ejemplo del método Simplex». http://www.phpsimplex.com/ejemplo_metodo_simplex.htm (accedido may 10, 2020).
- [26] Autor Desconocido «Mantenibilidad». <https://iso25000.com/index.php/normas-iso-25000/iso-25010/26-mantenibilidad> (accedido may 10, 2020).
- [27] Gustavo Cuen, «Atributos de calidad en el desarrollo de software», 15:22:46 UTC, Accedido: may 10, 2020. [En línea]. Disponible en: <https://es.slideshare.net/gustavo-cuen77/atributos-de-calidad-en-el-desarrollo-de-software-30272613>.
- [28] Autor Desconocido «Atributos de calidad», uqbar-wiki. wiki.uqbar.org/wiki/articulos/atributos-de-calidad.html (accedido may 10, 2020).
- [29] Autor Desconocido «Teoría del Método Simplex». http://www.phpsimplex.com/teoria_metodo_simplex.htm (accedido may 10, 2020).
- [30] Sergio Daniel «Introducción a JSON Web Tokens (JWT)», Platzi. <https://platzi.com/blog/introduccion-json-web-tokens/> (accedido may 10, 2020).
- [31] Autor Desconocido «Modelo vista controlador (MVC)». <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html> (accedido may 10, 2020).
- [32] Autor Desconocido «Las 5 principales ventajas de usar Angular para crear aplicaciones web | campusMVP.es». <https://www.campusmvp.es/recursos/post/las-5-principales-ventajas-de-usar-angular-para-crear-aplicaciones-web.aspx> (accedido may 10, 2020).
- [33] Autor Desconocido campus MVP, «Acceso a datos con .NET: Diferencias entre Entity Framework 6 “tradicional” y Entity Framework Core», campusMVP.es.

<https://www.campusmvp.es/recursos/post/que-diferencias-hay-entre-entity-framework-6-tradicional-y-entity-framework-core.aspx> (accedido may 10, 2020).

[34] J. Turrado, «Introducción rápida a LINQ con C#: manejar información en memoria nunca fue tan sencillo», campusMVP.es. <https://www.campusmvp.es/recursos/post/introduccion-rapida-a-linq-con-c-sharp.aspx> (accedido may 10, 2020).

[35] Autor desconocido «Qué es SCRUM», Proyectos Ágiles, ago. 04, 2008. <https://proyectosagiles.org/que-es-scrum/> (accedido may 24, 2020).

[36] Postman«Postman | The Collaboration Platform for API Development», Postman. <https://www.postman.com/> (accedido may 24, 2020).

ANEXO 2

CARTA DE AUTORIZACIÓN DE LOS AUTORES
(Licencia de uso)Bogotá, D.C., 17 de junio de 2020Señores
Biblioteca Alfonso Borrero Cabal S.J.
Pontificia Universidad Javeriana
Cuidad

Los suscritos:

Jhonatan Stiven Gonzalez Pelufo, con C.C No 1022388263

Juan David Cortes Olaya, con C.C No 1016076734

En mi (nuestra) calidad de autor (es) exclusivo (s) de la obra titulada:
Comedores - Aplicación adaptativa de apoyo a procesos de distribución de alimentos

(por favor señale con una "x" las opciones que apliquen)

Tesis doctoral Trabajo de grado Premio o distinción: Sí No

cual:

presentado y aprobado en el año 2020, por medio del presente escrito autorizo (autorizamos) a la Pontificia Universidad Javeriana para que, en desarrollo de la presente licencia de uso parcial, pueda ejercer sobre mi (nuestra) obra las atribuciones que se indican a continuación, teniendo en cuenta que en cualquier caso, la finalidad perseguida será facilitar, difundir y promover el aprendizaje, la enseñanza y la investigación.

En consecuencia, las atribuciones de usos temporales y parciales que por virtud de la presente licencia se autorizan a la Pontificia Universidad Javeriana, a los usuarios de la Biblioteca Alfonso Borrero Cabal S.J., así como a los usuarios de las redes, bases de datos y demás sitios web con los que la Universidad tenga perfeccionado un convenio, son:

AUTORIZO (AUTORIZAMOS)	SI	NO
1. La conservación de los ejemplares necesarios en la sala de tesis y trabajos de grado de la Biblioteca.	X	
2. La consulta física (sólo en las instalaciones de la Biblioteca)	X	
3. La consulta electrónica - on line (a través del catálogo Biblos y el Repositorio Institucional)	X	
4. La reproducción por cualquier formato conocido o por conocer	X	
5. La comunicación pública por cualquier procedimiento o medio físico o electrónico, así como su puesta a disposición en Internet	X	
6. La inclusión en bases de datos y en sitios web sean éstos onerosos o gratuitos, existiendo con ellos previo convenio perfeccionado con la Pontificia Universidad Javeriana para efectos de satisfacer los fines previstos. En este evento, tales sitios y sus usuarios tendrán las mismas facultades que las aquí concedidas con las mismas limitaciones y condiciones	X	

De acuerdo con la naturaleza del uso concedido, la presente licencia parcial se otorga a título gratuito por el máximo tiempo legal colombiano, con el propósito de que en dicho lapso mi (nuestra) obra sea explotada en las condiciones aquí estipuladas y para los fines indicados, respetando siempre la titularidad de los derechos patrimoniales y morales correspondientes, de

acuerdo con los usos honrados, de manera proporcional y justificada a la finalidad perseguida, sin ánimo de lucro ni de comercialización.

De manera complementaria, garantizo (garantizamos) en mi (nuestra) calidad de estudiante (s) y por ende autor (es) exclusivo (s), que la Tesis o Trabajo de Grado en cuestión, es producto de mi (nuestra) plena autoría, de mi (nuestro) esfuerzo personal intelectual, como consecuencia de mi (nuestra) creación original particular y, por tanto, soy (somos) el (los) único (s) titular (es) de la misma. Además, aseguro (aseguramos) que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el derecho a la imagen, intimidad, buen nombre y demás derechos constitucionales. Adicionalmente, manifiesto (manifestamos) que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la elaboración, presentación, investigación y, en general, contenidos de la Tesis o Trabajo de Grado es de mí (nuestro) competencia exclusiva, eximiendo de toda responsabilidad a la Pontificia Universidad Javeriana por tales aspectos.

Sin perjuicio de los usos y atribuciones otorgadas en virtud de este documento, continuaré (continuaremos) conservando los correspondientes derechos patrimoniales sin modificación o restricción alguna, puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación de los derechos patrimoniales derivados del régimen del Derecho de Autor.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, “*Los derechos morales sobre el trabajo son propiedad de los autores*”, los cuales son irrenunciables, imprescriptibles, inembargables e inalienables. En consecuencia, la Pontificia Universidad Javeriana está en la obligación de RESPETARLOS Y HACERLOS RESPETAR, para lo cual tomará las medidas correspondientes para garantizar su observancia.

NOTA: Información Confidencial:

Esta Tesis o Trabajo de Grado contiene información privilegiada, estratégica, secreta, confidencial y demás similar, o hace parte de una investigación que se adelanta y cuyos resultados finales no se han publicado. Si No

En caso afirmativo expresamente indicaré (indicaremos), en carta adjunta, tal situación con el fin de que se mantenga la restricción de acceso.

NOMBRE COMPLETO	No. del documento de identidad	FIRMA
Juan David Cortes Olaya	1016076734	
Jhonatan Stiven Gonzales Pelufo	1022388263	Jhonatan Stiven Gonzalez Pelufo

FACULTAD: Ingeniería

PROGRAMA ACADÉMICO: Maestría en ingeniería de sistemas y computación

Anexo 3
BIBLIOTECA ALFONSO BORRERO CABAL, S.J.
DESCRIPCIÓN DE LA TESIS O DEL TRABAJO DE GRADO
FORMULARIO

TÍTULO COMPLETO DE LA TESIS DOCTORAL O TRABAJO DE GRADO						
Comedores - Aplicación adaptativa de apoyo a procesos de distribución de alimentos						
SUBTÍTULO, SI LO TIENE						
AUTOR O AUTORES						
Apellidos Completos		Nombres Completos				
Cortes Olaya		Juan David				
Gonzales Pelufo		Jhonatan Stiven				
DIRECTOR (ES) TESIS O DEL TRABAJO DE GRADO						
Apellidos Completos		Nombres Completos				
Carrillo Ramos		Angela Cristina				
FACULTAD						
Ingeniería de sistemas						
PROGRAMA ACADÉMICO						
Tipo de programa (seleccione con "x")						
Pregrado	Especialización	Maestría	Doctorado			
		X				
Nombre del programa académico						
Maestría en ingeniería de sistemas y computación						
Nombres y apellidos del director del programa académico						
Angela Cristina Carrillo Ramos						
TRABAJO PARA OPTAR AL TÍTULO DE:						
Magíster en Ingeniería de Sistemas y Computación						
PREMIO O DISTINCIÓN (En caso de ser LAUREADAS o tener una mención especial):						
CIUDAD	AÑO DE PRESENTACIÓN DE LA TESIS O DEL TRABAJO DE GRADO	NÚMERO DE PÁGINAS				
Bogotá	2020	59				
TIPO DE ILUSTRACIONES (seleccione con "x")						
Dibujos	Pinturas	Tablas, gráficos y diagramas	Planos	Mapas	Fotografías	Partituras
X		X				
SOFTWARE REQUERIDO O ESPECIALIZADO PARA LA LECTURA DEL DOCUMENTO						
Nota: En caso de que el software (programa especializado requerido) no se encuentre licenciado por la Universidad a través de la Biblioteca (previa consulta al estudiante), el texto de la Tesis o Trabajo de Grado quedará solamente en formato PDF.						
Ninguno						

TIPO	DURACIÓN (minutos)	MATERIAL ACOMPAÑANTE			
		CANTIDAD	FORMATO		
			CD	DVD	Otro ¿Cuál?
Vídeo	10	1			Mp4
Audio					
Multimedia					
Producción electrónica					
Otro Cuál?					
DESCRIPTORES O PALABRAS CLAVE EN ESPAÑOL E INGLÉS					
Son los términos que definen los temas que identifican el contenido. <i>(En caso de duda para designar estos descriptores, se recomienda consultar con la Sección de Desarrollo de Colecciones de Biblioteca Alfonso Borrero Cabal S.J en el correo biblioteca@javeriana.edu.co, donde se orientará).</i>					
ESPAÑOL			INGLÉS		
Algoritmo DEA			DEA Algorithm		
Proveedores			Providers		
Solución IT			IT Solution		
Comedores COMunitarios			Community Dinners		
Optimizacion			Optimizacion		
RESUMEN DEL CONTENIDO EN ESPAÑOL E INGLÉS (Máximo 250 palabras - 1530 caracteres)					
<p>Aplicativo de TI que permite a través de una ejecución de algoritmo DEA ofrecer a comedores comunitarios la mejor selección de proveedores alimenticios y de transporte que permita minimizar costos y beneficiar a la mayor parte de personas que dependen de estos comedores, el aplicativo fue desarrollado en C# para el backend y Angular para el frontend, en cuanto a la persistencia se manejó base de datos SQL Microsoft server, de igual manera se aprovecharon patrones de Multinivel y capas para proveer una arquitectura escalable y estandarizable</p>					
<p>IT application that allows through a DEA algorithm execution to offer community kitchens the best selection of food and transportation providers that minimize costs and beneficiaries for the majority people who depend on these kitchens, the application was developed in C # For the backend and Angular for the frontend, in terms of persistence, the Microsoft SQL server database was managed, in the same way, Multilevel patterns and layers were used to provide a scalable and standardizable architecture</p>					

