EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Colima

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

SISTEMA DE GESTIÓN DE ACEITES USADOS

TESIS

QUE PARA OBTENER EL GRADO DE

MAESTRO EN SISTEMAS COMPUTACIONALES

PRESENTA

I.S.C. FRANCISCO DE JESÚS RIFAS MEJÍA

DIRECTOR DE TESIS

M.C. ARIEL LIRA OBANDO

CO-DIRECTOR DE TESIS

M.C. ROSA DE GUADALUPE CANO ANGUIANO

VILLA DE ÁLVAREZ, COLIMA, FEBRERO DE 2021.

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Colima
División de Estudios de Posgrado e Investigación

Villa de Álvarez, Colima, **24/Febrero/2021**
Oficio No. **DEPI 1.2.11/010/2021**

ALUMNO FRANCISCO DE JESÚS RIFAS MEJÍA
PASANTE DE LA MAESTRÍA EN SISTEMAS COMPUTACIONALES
PRESENTE

La División de Estudios de Posgrado e Investigación de acuerdo al procedimiento para la obtención del Título de Maestría de los Institutos Tecnológicos y habiendo cumplido con todas las indicaciones que la comisión revisora hizo a su trabajo profesional denominado "**SISTEMA DE GESTIÓN DE ACEITES USADOS**", por la opción de tesis, que para obtener el grado de Maestro en Sistemas Computacionales será presentado por Usted, tiene a bien concederle la **AUTORIZACIÓN** de impresión de la tesis citada.

Sin otro particular por el momento, aprovecho la ocasión para enviarle un cordial y afectuoso saludo.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
"Estudiar para prever y prever para actuar"

SEP - TecNM
INSTITUTO TECNOLÓGICO
De Colima
DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN

RAMONA EVELIA CHÁVEZ VALDEZ
JEFA DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

RECV/cas

C.p. Archivo.

Av. Tecnológico No. 1, Colonia Liberación. C.P. 28976,
Vila de Álvarez, Colima.
Tel. 312-312-6393, 312-314-0933, 312-312-9920 Ext. 113 y 213
email: posgrado@colima.tecnm.mx
www.colima.tecnm.mx

México 2021 Año de la Independencia

Agradecimientos

Quiero agradecer al Instituto Tecnológico de Colima por darme la oportunidad de cursar la Maestría en Sistema Computacionales y en su momento la ingeniería. A todos los maestros que siempre tuvieron sus puertas abiertas para poder quitarles un poco de su tiempo con mis dudas. Especialmente a todos mis profesores de la maestría que cada uno, a su manera, me dio las herramientas para terminar mi tesis.

También quiero agradecer a todos los involucrados en mi tesis: director, co-director y revisores por su tiempo y dedicación, pero sobre todo por la paciencia brindada para concluir está importante etapa de mi vida. Mi más grande agradecimiento al M.C. Ariel Lira Obando por nunca perder la fe y no abandonarme a mi suerte con tanto obstáculo en el camino.

A mis compañeros de la generación 2017 por siempre haber mantenido un ambiente de respeto y apoyo durante el aprendizaje, a mi amigo Andrés Hernández Jiménez que me ayudó a dar los primeros pasos para mi desarrollo del proyecto de la tesis.

Por último y no menos importante a mi familia, mi hermana y mi madre que siempre me apoyaron de todas las maneras posibles, mis hijos que cuando no estaban presente me mandaban mensajes y venían a visitarme para que no me sintiera solo.

Dedico mi tesis a mi padre que, aunque ya no está conmigo de manera física sé que siempre me acompañas, te prometí que un día seguiría estudiando, espero que donde sea que estés celebres esto conmigo.

“Frecuentemente hay más que aprender de las preguntas inesperadas de un niño que de los discursos de un hombre.”

John Locke

Resumen

En este proyecto se desarrolló un sistema de gestión de aceites usados para solucionar el problema que enfrente la empresa Tecnologías Disruptivas S.A.P.I. de C.V. en su proceso de recolección. Se realizó un análisis de la situación actual y los requerimientos de la empresa con lo que se propuso la implantación de un sistema informático para la gestión de datos, generación de información y administración de la logística. Se analizan las diversas soluciones actuales y sus desventajas antes las necesidades de la empresa recolectora. Definimos una metodología y seleccionamos el algoritmo de Dijkstra para el cálculo de rutas óptimas con lo cual se pretende disminuir los tiempos de recolección y ahorrar en insumos como la gasolina. Como resultado obtuvimos un sistema compuesto de dos partes, una parte en ambiente web para la captura de datos de las empresas y la generación de información mediante reportes solicitados por la empresa recolectora, y también una aplicación móvil que permite al recolector del aceite la captura de la información de las recolecciones. Con los resultados obtenidos la empresa recolectora puede llevar ahora un mejor control de la información y acceder a ella de manera inmediata para la toma de decisiones.

Abstract

In this project a used oil management system was developed to solve the problem faced by the company Tecnologías Disruptivas S.A.P.I. de C.V. in its collection process. An analysis of the current situation and the requirements of the company was carried out with which the implementation of a computer system for data management, information generation and logistics administration was proposed. Various current solutions and their disadvantages are analyzed before the needs of the collecting company. We define a methodology and select the Dijkstra algorithm for the calculation of optimal routes, which is intended to reduce collection times and save on goods such as gasoline. As a result, we obtained a system composed of two parts, one part in a web environment for the capture of data from the companies and the generation of information through reports requested by the collecting company, and also a mobile application that allows the collector to capture the information of the collections. With to the results obtained, the collecting company can now have better control of the information and access it immediately for decision-making.

Índice

Tabla de ilustraciones

1. Introducción

Hoy en día las actividades diarias se viven y desarrollan de manera acelerada; el éxito o fracaso de una empresa depende en gran medida de los tiempos involucrados en cada uno de los procesos involucrados en ellas.

El presente capítulo expone la situación actual que vive una empresa recolectora de aceite de cocina que busca contar con mecanismos que permitan realizar de manera eficiente y oportuna las actividades involucradas en el cumplimiento de sus objetivos.

1.1. La naturaleza del problema

En Colima, México, la empresa **Tecnologías Disruptivas S.A.P.I. de C.V.** se dedica a la producción de biodiesel a partir de aceites de cocina usados (ACU). Las empresas proveedoras del ACU (a quien en lo sucesivo se le nombrará **Empresa Proveedora**) agendan su recolección mediante llamadas, mensajes o correos electrónicos o, en caso contrario, el personal de la empresa Tecnologías Disruptivas S.A.P.I. de C.V. (a quien en lo sucesivo se le nombrará **Empresa Recolectora**) tiene que comunicarse constantemente con dichos empresas proveedoras registradas en su directorio para saber si es posible realizar una recolección. Esto ha traído problemas de logística para reunir la materia prima ya que en más de una ocasión alguna Empresa Proveedora solicita el servicio de recolección cuando el camión recolector acaba de terminar la ruta en la que se ubica el ACU a recolectar; por consiguiente, esto ha impedido el incremento en el volumen de la recolección, disminuyendo las ganancias y aumentando los gastos al tener que realizar, en algunas ocasiones, dobles recorridos.

Muchas veces, el no realizar la recolección en el momento en el que lo solicita la Empresa Proveedora, ha derivado en la pérdida de proveedores y materia prima, debido a que tales empresas proveedoras buscan otra empresa recolectora porque tienen la necesidad de liberar sus almacenes.

También es importante recalcar que la Empresa Recolectora no cuenta con un mecanismo de publicidad que le permita darse a conocer de manera expansiva y que nuevas Empresas Proveedoras puedan tener contacto directo con ella para darse de alta y ser consideradas como candidatas a ser proveedoras de ACU.

Otro tipo de problema que presenta la Empresa Recolectora es la forma como registran sus actividades, ya que su proceso principal, el de recolección de ACU, es registrado en hojas de Excel, encontrándose con muchas limitantes cuando se desea consultar información valiosa e importante que tiene que ver con agrupamientos, condiciones u ordenamientos, por ejemplo:

- Desplegar las mejores Empresas Proveedoras por zona
- Calcular el total de adeudo que se tienen por Empresa Proveedora
- Calcular el precio neto de cada litro de ACU (incluyendo los gastos de operación involucrados), entre otros.

Por último, el que es considera el problema principal y motivo de la presente tesis, la Empresa Recolectora enfrenta un obstáculo para llegar a las Empresas Proveedoras en un mismo recorrido; todo parece indicar que invierte tiempo y dinero innecesario en el cumplimiento de dicha actividad, esto debido a que no cuenta con un recorrido óptimo en la recolección del ACU o, en algunas ocasiones, por falta de comunicación con la Empresa Proveedora, el recorrido es improductivo debido a que el encargado de entregar el ACU no se encuentra en el lugar indicado, el horario o día en el que acudió a hacer la recolección no era el adecuado o no se tiene la cantidad esperada del producto.

Como conclusión de este apartado podemos decir que la Empresa Recolectora necesita mejorar sus procesos debido a que ha tenido un crecimiento significativo en lo que se refiere a superficie geográfica que atiende y número de Empresas Proveedoras a las que le da servicio (incluyendo para ellas toda la documentación que se genera desde su registro hasta los comprobantes de pago de la recolección realizada).

1.2. El contexto del problema

Parte fundamental de la recolección de ACU es el diseño del recorrido diario y su planeación, para lograrlo de la mejor manera es necesario calcular la ruta que el conductor necesitará, de lo contrario se seguirán haciendo gastos innecesarios. El tráfico, las distancias y el costo de la gasolina hacen que sea importante atacar este problema.

De acuerdo a la comparativa realizada por (Luna, 2016), el estado de Colima, México tiene una de las mayores proporciones de autos por persona en México y, según estadísticas del año 2015, se detectó que el estado cuenta con 711 mil 235 habitantes y 298 mil 88 automotores, lo que representa 419 autos motorizados por cada 1 mil habitantes. Este dato es importante ya que la planeación correcta de una ruta de recolección óptima disminuirá los tiempos de traslado y el consumo de combustible requerido que a su vez repercute en la emisión de gases contaminantes al medio ambiente.

En los indicadores de (CME Group & Sistema de Información Energética, 2017), se puede apreciar que a partir del mes de junio de 2014 los precios del petróleo han disminuido, sin embargo el precio de la gasolina han ido a la alza y en el estado de Colima, México, su precio actual es mayor a $19.21 pesos mexicanos según los datos obtenidos al día 18 de Agosto de 2018 en (GasolinaMX.com, 2018).

Sumado a esto el cambio climático y el calentamiento global expuesto por la Organización de las Naciones Unidas (ONU) (United Nations, n.d.) han impulsado la tendencia hacia la producción de productos derivados del reciclaje y la reutilización. La refinación implica la reconversión de miles de litros de residuos peligrosos que en la actualidad tienen un destino incierto y que según (González Canal & González Ubierna, 2015) un litro de ACU desechado de forma incorrecta podría llegar a contaminar 40,000 litros de agua.

1.3. Revisión de la literatura

Uno de los trabajos con mayor similitud en lo que respecta al título de este trabajo es el de la tesis presentada por (Cartas Díaz, 2014) titulada “Sistema de Recolección de Aceite Usado para Conversión de Biodiesel”, en el que se abordan los aspectos relacionados con los gastos y costos generados durante el proceso de recolección de ACU, las ventajas de los biocombustibles y los gases de efecto invernadero, haciendo énfasis en la tasa de recuperación al momento de convertir el ACU en biodiesel. Esta tesis se enfoca en los beneficios de la conversión y no cubre el proceso de la recolección más que del punto de vista relacionado a los costos de acuerdo a las personas involucradas en el proceso.

El artículo publicado por (Berov, 2016) implementa algoritmos genéticos para la distribución de insumos en áreas urbanas usando el servicio de Google Maps. Este trabajo es el que tiene más similitud al propuesto en la presente tesis en lo referente al desarrollo, termina tomando un enfoque diferente para el cálculo del recorrido, plantea la división de una ruta en varias más cortas y optimizadas para disminuir los tiempos de recolección partiendo desde un recolector hasta n recolectores.

En el trabajo desarrollado por (Wang et al., 2015) se enfocan en la utilización de GPS y el uso de servicios de mapas, no utilizan el algoritmo de Dijkstra, empiezan a recolectar las rutas con el apoyo del Google Maps ellos abordan el problema a partir de entradas de información recolectados por el GPS y así poder generar las rutas con base a los puntos de conflictos detectados. El desarrollo propuesto en esta tesis no necesita la recolección de información mediante GPS, se desea tener planeada la ruta antes de realizar la recolección y probablemente en un futuro pueda ser considerado mejora.

(Kedia & Naick, 2017) realiza una revisión de la optimización de rutas de vehículos verdes (vehículos eléctricos) aplicando el algoritmo de Particle swarm optimization y el algoritmo de Dijkstra; considera que los algoritmos meta heurísticos se desarrollaron para optimizar el tiempo de ejecución o la complejidad del tiempo además de la minimización de costos y considera al algoritmo de Dijkstra como un algoritmo rápido para la planeación de rutas.

1.4. Planteamiento de la solución del problema a investigar

Los sistemas de información en la actualidad están transformando en la forma en que trabajan las organizaciones. Utilizándolos se obtienen grandes mejoras, puesto que automatizan los procesos operativos que forman parte de la empresa, proveen información de apoyo al proceso de las tomas de decisiones y facilitan la obtención de ventajas competitivas a través de su implementación dentro de la organización (Hamidian Fernández & Ospino Sumoza, 2015).

Se plantea como solución el diseño y desarrollo de un sistema informático que permita a la Empresa Recolectora el control en la recolección de ACU y los datos generados por dicha actividad. El sistema será formado por dos partes, uno en ambiente web y otro en dispositivos móviles.

Mediante el ambiente web se permitirá a las Empresas Proveedoras registrarse y definir sus fechas con horarios de recolección adecuados a sus necesidades, de igual manera el sistema permitirá a la Empresa Recolectora gestionar las rutas de recolección, obteniendo en cada una de ellas la más óptima a través de la implementación del algoritmo de Dijkstra y apoyándose en las herramientas proporcionadas por Google Maps. La parte implementada en dispositivos móviles será exclusiva de los recolectores y será la encargada de alimentar, con datos al sistema para su posterior transformación en información.

1.5. Justificación

Como mencionamos anteriormente, el presente trabajo de tesis plantea la creación de un sistema de información para la gestión en la recolección de ACU, para que posteriormente sea transformado en biodiesel. La justificación se basa en los siguientes puntos:

Beneficios de impacto:

- **Ambientales**. La refinación implica la reconversión de miles de toneladas de residuos peligrosos que en la actualidad tienen un destino incierto.
- **Económicos**. Beneficios para los sectores Logístico, Energético, Agroindustrial y Apoyo a Negocios al obtener productos más económicos.
- **Sociales**. El manejo integral de los aceites usados permite su reciclado, genera energía, protege nuestra salud y evita la contaminación de nuestro aire, suelo y agua.

Beneficios para los usuarios del sistema:

- Disminuir, por parte de la Empresa Recolectora, los tiempos de recolección y, con ello, dar lugar a una mayor y mejor atención a los usuarios.
- Reducir los gastos en combustible para los vehículos de recolección al hacer más corto el recorrido y por consecuencia disminuir la contaminación al reducir la emisión de gases contaminantes.
- Permitir, a las Empresas Proveedoras, tener una mejor comunicación al momento de requerir una recolección.
- Poder detectar las áreas donde se concentra las grandes cantidades de aceites de cocina usado.
- Ayudar a que nuevas Empresas Proveedoras entren en contacto con la Empresa Recolectora para aumentar la cantidad de materia prima obtenida.
- Facilitar a la Empresa Recolectora el acceso a la información para la toma de decisiones de nivel estratégico.

1.6. Objetivos

1.6.1. General

Desarrollar e implantar un sistema informático para la gestión de datos y generación de información para el control y administración de la logística asociada a la recolección de ACU.

1.6.2. Particulares

- Disminuir el consumo de combustible utilizado por los camiones recolectores que participan en las rutas de recolección.
- Incrementar la recolección de ACU mediante una mejora en la planificación de las rutas.
- Ofrecer informes y datos estadísticos que permitan una toma de decisión a nivel estratégico.
- Garantizar la recolección ACU sin costo alguno para la Empresa Proveedora y sin importar ubicación y cantidad acumulada.
- Proporcionar un mecanismo que permita una comunicación directa entre Empresa Proveedora y Empresa Recolectora, conociendo la situación que guarda cada una de ellas en un momento determinado.

1.7. Alcances y limitaciones

El sistema propuesto comprende desde el momento en que una empresa se registra en el sistema hasta el momento en el que se le liquida el pago correspondiente por el ACU recolectado. Se incluirán catálogos de empresas, puntos de recolección, contactos y recolectores; con base a los puntos de recolección registrados y a los días y horarios de recolección, se generarán rutas óptimas que permitan recorridos eficientes y se mantendrá el conteo sobre los litros de ACU recolectados. Como sugerencia del cliente, se implementó una opción extra que permite dar salida del ACU a la planta de tratamiento, esto con la finalidad de que la existencia registrada en el sistema coincida con lo existente en los depósitos de almacenamiento.

Como una limitación del sistema podemos decir que este no incluye ningún aspecto relacionado al proceso de transformación de ACU en biodiesel, ni administrará ningún proceso relacionado con la planta de tratamiento, solo registrará las cantidades de ACU que se transportan del almacén a dicha planta de tratamiento, haciendo una estimación del total de biodiesel que se obtendrá de dicha recolección, esto con la finalidad de informar a la sociedad sobre los logros y beneficios obtenidos con esta actividad.

1.8. Estudios de factibilidad

1.8.1. Factibilidad económica

Desarrollar el proyecto es viable económicamente puesto que se desarrolla en conjunto entre el Instituto Tecnológico de Colima y la empresa Tecnologías Disruptivas S.A.P.I. de C.V. sin costo alguno para ambas partes. La empresa cuenta con suficientes equipos de cómputo y equipos móviles para desarrollar y probar los sistemas sin realizar ningún gasto extra.

Por otro lado, el proyecto se desarrollará mediante software de licencia libre, el costo de las herramientas de programación tampoco representará gasto alguno puesto que el manejador de bases de datos MySQL como el ambiente de desarrollo para Web y Android son de uso gratuito.

El uso de los servicios de mapas de Google es gratuito y aunque cuentan con su contraparte de paga, no es impedimento para la implementación y desarrollo del sistema.

1.8.2. Factibilidad operativa

Actualmente la empresa realiza la captura de información y recolección de manera manual en formatos pre impresos y hojas de cálculo, haciendo la planeación sobre la marcha e improvisando todo el proceso asociado.

El proyecto no afectará la operación que tiene la empresa hasta el momento, tampoco interrumpirá las actividades comunes. El proyecto podrá ser implementado por fases en la empresa para mejorar el desempeño gradual del proceso de recolección.

En cuanto al proceso de recolección no sufrirá cambios drásticos en la manera en que se ha venido realizando, la diferencia radica en que ahora se implementará logística en la recolección y control de las cantidades recolectadas y los formatos asociados a cada proceso serán llenados de manera electrónica, contando con información en tiempo real.

1.8.3. Factibilidad técnica

Para el desarrollo del proyecto son necesarias 3 herramientas: motor de base de datos, lenguaje web y mapas con distancias y guías de recorrido. Para la primera herramienta existen muchas opciones, optándose por el Sistema Gestor de Base de Datos (SGBD) MySQL por cumplir con la factibilidad económica y operativa. Como segunda herramienta se eligió el lenguaje CodeIgniter por contar con la capacidad que el desarrollo necesita, además de incluir de manera automática el soporte del motor de la base de datos seleccionado. Y por último la herramienta de mapas es proporcionada por Google Maps realizando las tareas necesarias para la programación como es la de brindar distancias e indicaciones de manejo de un punto a otro.

Por tal motivo se considera que el proyecto cuenta con la factibilidad técnica necesaria ya que todas las herramientas se encuentran al alcance y no es necesario desarrollar ninguna para el desarrollo e implementación del mismo.

1.8.4. Factibilidad legal

En el estado de Colima y en toda la República Mexicana no existe ningún impedimento legal para la implantación de un sistema informático enfocado a mejorar la logística y control de la información de una empresa.

En el Artículo 3, fracciones II y VII, y 33 así como en la denominación del capítulo II, del título segundo de la Ley Federal de Transparencia y Acceso a la Información Pública Gubernamental se delimita el uso de la información que se

recolecta de las empresas y establece que se deberá generar un documento por la empresa responsable de la recopilación y tratamiento adecuado de datos personales para ser puesto a disposición del titular de los datos, mismo que será implementado en el sistema siguiendo los lineamientos marcados por la ley.

1.9. Análisis costo-beneficio

Para la determinación de la relación costo-beneficio que tiene el presente proyecto de desarrollo deberá tenerse en cuenta lo siguiente:

- Para el desarrollo se utilizarán herramientas de software gratuito como lo son MySQL para la base de datos, CodeIgniter para el desarrollo web y Google Maps para las medidas de distancias.
- El costo del servidor donde será alojado el proyecto no representa ningún costo porque la empresa tiene donde alojar el sistema. También cuentan con varios dispositivos móviles para la aplicación sin necesidad de adquirir uno de manera específica para el desarrollo y utilización.
- Se disminuye el consumo de combustible en los vehículos recolectores, se mejora la planificación de las rutas y se ofrece información estadística de manera rápida y eficiente

Teniendo en cuenta lo anteriormente expuesto, el presente proyecto no implicará gasto para la empresa beneficiada y desde otra perspectiva podemos señalar que, de no ser realizado, muchas empresas que serían beneficiadas de manera colateral seguirán desechando el ACU de manera incorrecta.

1.10. Hipótesis

Siguiendo la narrativa de la presente tesis se plantea como hipótesis que: "*La implantación de un sistema informático que gestione el proceso de recolección de ACU y genere rutas óptimas para la obtención de información a partir de los datos recabados permitirá una reducción en los costos y tiempos involucrados en dicho proceso".*

1.10.1 Variable independiente

La variable independiente es la implantación del sistema informático para gestionar el proceso de recolección y generación de rutas.

1.10.2. Variable dependiente

La variable dependiente es la información producida a partir de los datos almacenados por las Empresas Proveedoras, las recolecciones y rutas generadas por la aplicación y los datos almacenados por la Empresa Recolectora.

1.11. Métodos y herramientas

Para el desarrollo del proyecto se optó por utilizar la metodología ágil XP mientras que para la generación de la ruta óptima en el proceso de recolección se trabajó con el algoritmo de Dijkstra.

La metodología ágil XP, según (Wells, 2013), es una opción ideal porque se centra en la satisfacción del cliente y ha sido probada por muchas compañías de diferentes tamaños y giros alrededor del mundo.

Como herramientas para los productos de análisis, diseño y desarrollo se utilizó el siguiente software:

- Balsamiq Mockups 3: Software para el diseño de mockups de las interfaces web y móviles.
- Enterprise Architect 12.1: En la documentación de los casos de uso y el diagrama de clases.
- MySQL 5.1: Manejador de base de datos seleccionado por su uso libre y licencia sin costo, amplia documentación y fácil migración a otro manejador en caso de ser necesario.
- Framework de PHP CodeIgniter 3.1.8: Para la programación de la página web, además del uso de JavaScript y HTML 5.
- Android Studio 3.1.2: Se utilizó como el ambiente de desarrollo para la generación de la aplicación en dispositivos móviles.
- API de Google Maps: Para la obtención de información de las distancias y tiempos en la generación de rutas.

1.12. Organización de la tesis

La actual tesis se encuentra estructurada de la siguiente manera:

El primer capítulo plasma la introducción de la investigación científica, la naturaleza del problema, los objetivos del proyecto, su justificación y sienta las bases para la solución del mismo.

En el segundo capítulo se describen los antecedentes y trabajos relacionados o similares al proyecto, explicando la aportación de cada uno de ellos y entendiendo las diferencias y semejanzas de este proyecto.

En el tercer capítulo se describe la metodología utilizada para el análisis y desarrollo de la tesis, explicando claramente las fases que componen tal metodología.

En el cuarto capítulo se aborda el desarrollo de la metodología seleccionada, haciendo énfasis en los productos que se generaron en cada etapa de la misma tanto para la parte web y como la móvil.

En el quinto capítulo analizamos los resultados obtenidos en la implantación de los sistemas desarrollados y la información generada por los mismos.

En el sexto capítulo exponemos las conclusiones a las que se llegaron y proponemos trabajo a futuro para mejorar el proyecto y continuar mejorándolo.

2. Estado del campo del conocimiento

En el siguiente capítulo se detalla el marco histórico, contextual y teórico del proyecto. En el marco histórico se hace mención de los estudios, avances y aportaciones que se han hecho en torno al problema que se desea resolver empezando desde el primer problema del agente viajero hasta su solución mediante soluciones de grafos. En el marco contextual ahondamos en las soluciones actuales con sus ventajas y limitaciones, se nombran los diferentes sistemas móviles y de plataforma con un perfil similar al buscado. Finalmente, en el marco teórico abordamos las herramientas necesarias para lograr el desarrollo del proyecto de manera exitosa.

2.1. Marco histórico

El marco histórico dentro de una investigación científica tiene como propósito mostrar la reseña histórica que permite precisar el contexto. En ese sentido, (Carrasco Díaz, 2009) señala que “es una narración descriptiva de cómo surge, evoluciona y se agudiza el problema de investigación”. De acuerdo a eso, el marco histórico es la limitación de los hechos pasados en la que se establece, cuáles han sido las diferentes etapas por las que ha atravesado el objeto de estudio en el desarrollo, hasta llegar al estado en que se encuentra al someterlo a investigación. También comprende la narración de los estudios que ha tenido el objeto de estudio, nombrando los hallazgos que a cada caso correspondan.

La optimización de las rutas y la necesidad de encontrar el camino más corto ha sido ampliamente ligada al problema del agente viajero o TSP (Travelling Salesman Problem por sus siglas en inglés). Según (Colaboradores de Wikipedia, n.d.-d) en su artículo de “Travelling salesman problem” indica que la primera mención del problema fue en un libro de 1832 pero no contiene ningún tratamiento matemático del mismo o solución planteada.

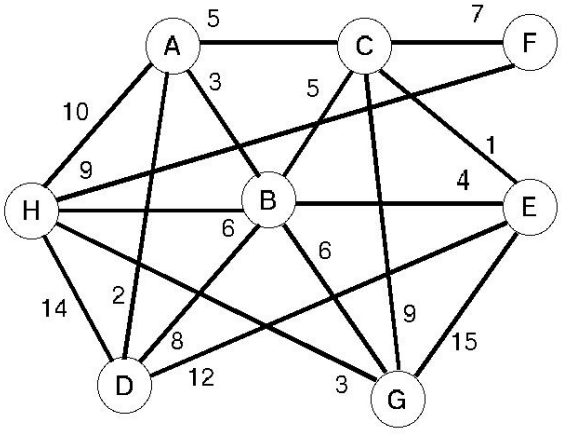


Ilustración 1. Grafo ponderado.

El TSP puede ser modelado como un grafo ponderado (ver ilustración 1) para su representación y tratamiento, por consiguiente, existen diferentes aproximaciones para su solución. Una de las primeras soluciones propuestas fue el algoritmo del vecino más cercano, el cual permite elegir entre los puntos que no han sido visitados en el grafo como próximo movimiento y de acuerdo a (Chen, Han, Wang, & Liu, 2010) se considera de las más importantes para consultas espaciales en sistemas de información avanzada de los años recientes y por consiguiente el algoritmo del vecino más cercano también se extiende a un escenario de red vial utilizando la distancia de red como la métrica de distancia. En la ilustración 2 podemos ver cómo quedaría resuelto un grafo ponderado bajo el tratamiento del algoritmo del vecino más cercano.

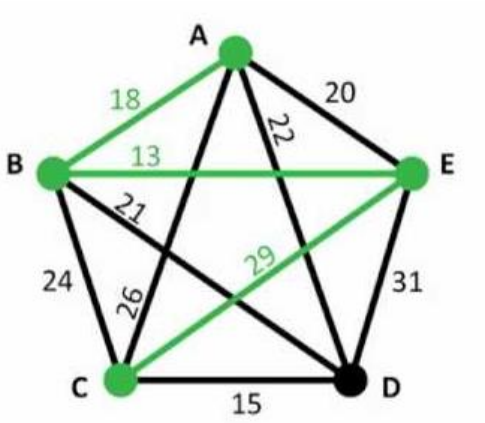


Ilustración 2. Grafo ponderado solucionado con el algoritmo del vecino más cercano.

Otra solución en la búsqueda de la ruta más corta es el algoritmo de Prim (ilustración 3) el cual fue diseñado en 1930 por el matemático Vojtech Jarnik. Para solucionar el problema este encuentra un subconjunto de aristas que forman un árbol con todos los vértices, donde el peso total de todas las aristas en el árbol es el mínimo posible (Colaboradores de Wikipedia, n.d.-b). En la solución hibrida propuesta por (Kheirkhahzadeh & Barforoush, 2009) mezclan el algoritmo de Prim con el de la optimización por colonia de hormigas, el algoritmo cumple la función de reducir el grafo y la optimización por colonia de hormigas el de encontrar la ruta más corta.

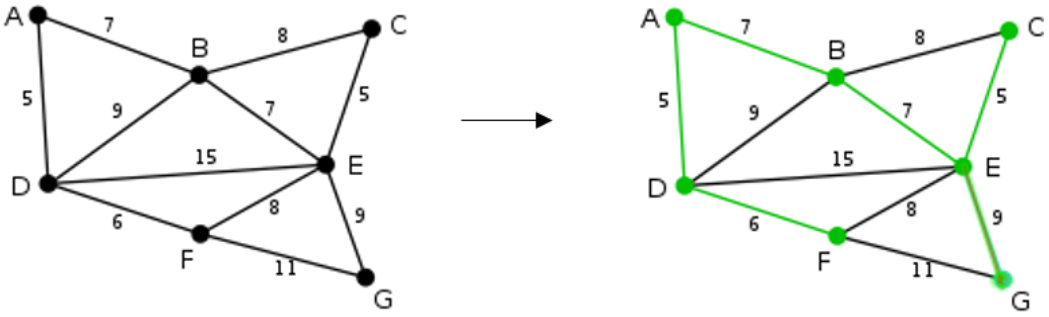


Ilustración 3. Grafo, antes y después del algoritmo de Prim.

Tiempo después se propuso el algoritmo de Dijkstra también llamado algoritmo de caminos mínimos (ilustración 4), es un algoritmo para la solución del camino más corto, partiendo de un vértice origen hacia el resto de los vértices en un grafo que tiene pesos para cada arista. Se nombra así por Edsger Dijkstra, científico de la computación de los Países Bajos que lo reseñó por primera vez en 1959 (Colaboradores de Wikipedia, n.d.-a). En la actualidad existen mejoras realizadas al algoritmo original como el presentado en el artículo (Wei & Meng, 2014) que se enfoca en enfatizar el congestionamiento vial y su repercusión en las redes de caminos al momento de encontrar caminos más cortos.

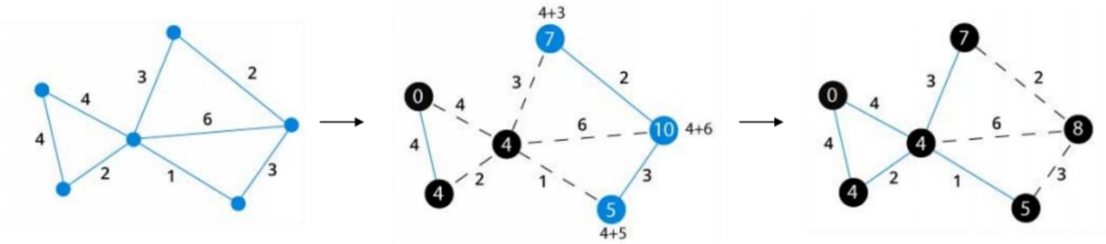


Ilustración 4. Solución de un grafo aplicando el algoritmo de Dijkstra.

Por último, surgió la solución de optimización por colonia de hormigas que imita el comportamiento observado en las hormigas y su forma de encontrar caminos cortos entre las fuentes de comida y su nido. Este algoritmo probabilístico (ilustración 5) es el usado para la optimización de caminos propuesto por Marco Dorigo en su disertación doctoral en 1992 y la idea surgió de las actividades que las hormigas hacen cuando están buscando comida (Pei, Wang, & Zhang, 2012).

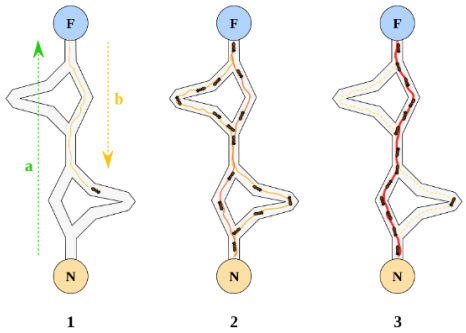


Ilustración 5. Optimización por colonia de hormigas.

2.2. Marco contextual

En primer lugar, el marco contextual delimita el ámbito o el ambiente físico dentro del cual se desarrolla la investigación, el mismo tema de investigación puede devolver diferentes resultados, dependiendo del lugar en el que se aplica.

El marco contextual interviene de manera directa en los objetivos generales y en los específicos, porque nos da las características particulares que se

consideren más apropiadas para la realización del marco teórico. Se puede identificar al sujeto, objeto y medio en el que se desarrolla la investigación mediante el marco contextual.

En segundo lugar, el marco contextual proporciona características y elementos cualitativos y cuantitativos del medio, personas o ambiente en el que se desarrolla la investigación.

En conclusión el marco contextual fija con precisión los límites de la investigación, aporta argumentos únicos y propios, esboza y define el alcance que deberá sobreponerse en el trabajo en coherencia con los objetivos planteados (Hernandez Sampieri, Fernandez Collado, & Baptista Lucio, 2010).

Las características del marco contextual son (Del Cid, Méndez, & Sandoval, 2011):

- Encuadrar un trabajo de investigación es describir dónde (lugar o ambiente) se lleva a cabo el evento o problema de investigación.
- Señalar algunos de los autores que han investigado el tema, los métodos y técnicas que utilizaron para llegar a los resultados obtenidos.
- La realización del marco contextual precisa visitas a bibliotecas, centros de investigación e información, consultas de páginas web y a expertos en la materia.

En la actualidad existen soluciones que abordan el problema del control de las recolecciones o distribuciones sin embargo el principal obstáculo con estas soluciones es que son bastante generales y cuentan con opciones innecesarias para la empresa que al final solo traen complicaciones en el proceso e incrementa el tiempo necesario en las actividades.

Para abordarlas las clasificaremos en tres categorías:

- Sistemas de planificación de recursos empresariales (ERP, por sus siglas en inglés, Enterprise Resource Planning).
- Aplicaciones móviles.
- Sistemas a la medida no comerciales.

2.2.1. Sistemas ERP

Los sistemas ERP son los sistemas de información gerenciales que integran y manejan muchos de los negocios asociados con las operaciones de producción y de los aspectos de distribución de una compañía en la producción de bienes o servicios (Colaboradores de Wikipedia, n.d.-c).

Un sistema ERP es una aplicación informática que permite gestionar todos los procesos de negocio de una compañía en forma integrada, por lo general este tipo de sistemas está compuesto de módulos como Recursos Humanos, Ventas, Contabilidad y Finanzas, Compras, Producción entre otros, brindado información cruzada e integrada de todos los procesos del negocio, la implementación de esta herramienta en una empresa conlleva un proceso de transformación y redefinición de sus procesos (Chiesa, 2004). Algunos de los sistemas EPR en el mercado son los siguientes:

- Zoho One.
- SAP ERP.
- Microsoft Dynamics 365.
- Microsoft Dynamics NAV
- Microsoft Dynamics GP.
- Odoo.
- Oracle ERP Cloud.
- ERPNext.
- Openbravo.
- SAP S/4HANA
- Compiere
- Dolibarr
- Adempiere
- WebERP

Todos estos sistemas ERP están pensados desde empresas medianas hasta grandes y cuentan con desventajas que se presentan al implementarse. Aunque algunos se encuentran registrados bajo licencia GNU y son gratuitos, otros requieren la compra de una licencia o el pago de mensualidades/ anualidades. El sistema desarrollado en esta tesis es completamente gratuito para la empresa Tecnologías Disruptivas S.A.P.I. de C.V. en todas sus etapas incluidas.

Los sistemas ERP tienen un costo de implementación y mantenimiento que incrementan en relación a las áreas que se desea abarcar en la empresa, y su soporte tiene un costo independientemente si el sistema ERP es gratuito o de paga, con el tiempo una solución a la medida puede evolucionar con las nuevas metas de la empresa y evitar tener que volver a caer en todo el proceso requerido para implementar nuevos complementos de un sistema ERP o en el peor de los casos migrar a uno nuevo.

Por otro lado, los tiempos utilizados para personalizar un sistema ERP pueden llegar a ser bastante extensos si no se cuenta con la guía adecuada, siendo en algunas ocasiones imposible de personalizar por completo.

Por último y no menos importante se encuentra la complejidad que conlleva un sistema EPR que está planteado para la mayor cantidad de escenarios posibles y que a consecuencia de esto se vuelve demasiado extenso y difícil de dominar. La

complejidad trae consigo el incremento de los tiempos dedicados a una actividad, lo cual se transforma en recursos materiales y económicos.

2.2.2. Aplicaciones móviles

Una aplicación móvil o app (por sus siglas en inglés) es un programa que se descarga desde plataformas de distribución o bien puede ser instalado directamente en el dispositivo y al que puede acceder directamente desde su teléfono. Más en específico, en la plataforma Android existen soluciones para controlar las rutas, algunos de ellos son los siguientes:

- **Workwave Route Manager.** Recibe rutas de un EPR para su procesamiento.
- **MySmrtRoute Route Planner.** Planeador de rutas, optimizador de viaje, asistente de entregas. El usuario plantea todos los puntos deseados, no es de manera automática.
- **Optimización de Rutas.** Optimiza rutas con múltiples paradas. Acomoda las paradas de acuerdo a las distancias, no toma en cuenta los tiempos.
- **Planificador de Rutas de Reparto.** Ayuda a planear la distribución, entrega y recolección para el negocio. Al igual que las aplicaciones anteriores el usuario define el orden de las paradas.
- **Route Optimizer.** Administración y optimización de rutas para los mensajeros. Cuenta con un sistema de generación de rutas automático, pero no incluye el control de la recolección, es exclusivo para las entregas.
- **Multi Stop Route Planner.** Guarda y administra el tiempo para el ahorro de combustible. Se enfoca en los puntos de entrega o recolección sin tener control sobre los datos que necesitan ser almacenados.

Todas estas aplicaciones se encuentran para descargarse y todas son de paga, con la desventaja de que en todas ellas se tiene que programar las rutas de manera manual, incluso algunas solo permiten programar un recorrido de un punto de partida a un punto de llegada. La solución desarrollada en la presente tesis permite administrar rutas y calcula de manera automática el orden de las paradas con base en los cálculos hechos como resultado de la interacción obtenida desde Google Maps.

En el trabajo publicado por (Mustafa & Žari, 2018) desarrollan una aplicación para móviles en sistema operativo Android, teniendo como la principal característica el ofrecer rutas alternas en caso de que la ruta óptima esté bloqueada, lamentablemente es una aplicación que no permite gestionar múltiples destinos y sobre todo no cuenta con la programación de los horarios que es esencial en nuestro problema planteado.

2.2.3. Sistemas a la medida no comerciales

Existen algunas soluciones similares a la planteada en este trabajo, pero no es posible acceder a ellas ya que se encuentran en posesión de empresas que no tienen intención de hacer público su sistema o comercializarlo. Empresas como Coca-Cola, Estafeta, DHL y muchas otras cuentan con sus propios sistemas, sin embargo, no es posible evaluar las características de los sistemas ya que no se encuentra a nuestro alcance acceder a ellos para analizarlos.

2.3. Marco teórico

Según (Navarro, 2010), el marco teórico es el fundamento de los trabajos científicos y de investigación. Es la agrupación de procedimientos, teorías e ideas que fueran examinadas por un grupo o un autor, valiendo de metodología a un investigador para completar su propia actividad. Es un armónico círculo hacia el aumento del conocimiento, estableciendo el punto de partida desde las cuales se busca validar una cuestión específica. Todos los marcos teóricos suelen dividirse en dos pilares: las herramientas y los conocimientos básicos del tema y contexto, y por el otro lado, mostrando los registros obtenidos, reconociendo problemas e ideando propuestas al respecto, comprobando informaciones o creando nuevos conocimientos.

2.3.1. Patrón de arquitectura MVC

En su artículo “Arquitectura y diseño de sistemas web modernos” (Castejón Garrido, 2004) hace mención de como las aplicaciones web en pocos años se han convertido en intrincados sistemas con interfaces de usuario cada vez más similares a las aplicaciones de escritorio, proporcionando servicio a procesos de negocio de considerable importancia y asentándose sobre las mismas requisitos rigurosos de accesibilidad y respuesta. Esto ha demandado consideraciones sobre la mejor arquitectura y las técnicas de diseño más apropiadas.

El MVC (Modelo-Vista-Controlador) es un patrón de arquitectura de software que, utiliza 3 componentes (Modelos, Vistas y Controladores) diferencia la lógica de la aplicación de la lógica de la vista en una aplicación. Es una arquitectura importante porque se le ha utiliza tanto en componentes gráficos básicos hasta sistemas corporativos. Casi todos los frameworks modernos usan MVC (o alguna adaptación del MVC) para la arquitectura, entre ellos destacan Ruby on Rails, CodeIgniter, AngularJS, Python, Django y muchos otros más (Hernández, 2015).

- Modelo: Tiene como función el manejo de los datos, generalmente (pero no necesariamente) gestiona el acceso a la base de datos. Inserciones, modificaciones, consultas, eliminaciones, etc. todo eso forma parte del modelo.

- Vista: Es la interfaz que interactúa con el usuario, despliega de manera visual los datos. Ni el modelo ni el controlador son responsables de cómo se verán los datos, esa carga recae únicamente en la vista.
- Controlador: Es el intermediario entre los modelos y las vistas, recibe las indicaciones del usuario y se encarga de solicitar los datos al modelo y de proporcionárselos a la vista.

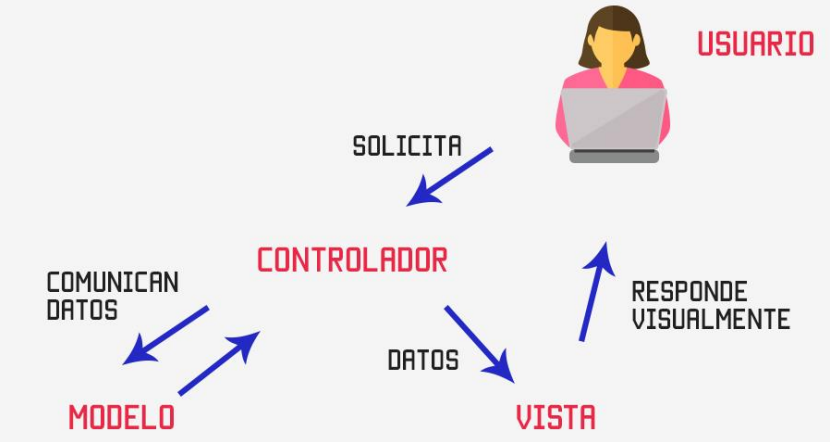


Ilustración 6. Funcionamiento del patrón MVC.

De acuerdo con (Fernández R. Y. & Díaz G. Y., 2012) el patrón MVC nace con el objetivo de disminuir el esfuerzo de programación necesario en la puesta en marcha de sistemas múltiples y sincronizados de los mismos datos, a partir de uniformar el diseño de las aplicaciones. El patrón MVC es un modelo que separa las partes que conforman una aplicación en el modelo, las vistas y los controladores, permitiendo la ejecución por separado de cada elemento, asegurando que un pequeño espacio de tiempo se puede realizar la actualización y mantenimiento del software de forma sencilla. Partiendo del uso de frameworks centrados en el patrón MVC se logra una mayor eficacia en la organización del trabajo y mayor especialización de los programadores (desarrolladores y diseñadores).

2.3.2. Sistemas de bases de datos

El autor (Date, 2001) afirma que un sistema de base de datos es básicamente un sistema computarizado para llevar registros. es posible considerar la propia base de datos como una especia de armario electrónico para archivar, es decir, es un depósito o contenedor de una colección de archivos de datos computarizados.

En el libro de (Osorio Rivera, 2008) definen un sistema de manejo de base de datos (en inglés DBMS Database Management System) como un subconjunto de elementos interrelacionados y una serie de programas que permiten a varios usuarios tener accesos a estos archivos ya sea para consultarlos o actualizarlos. Entre los objetivos de un DBMS está el de proporcionar a los usuarios una visión abstracta de la información, lo cual quiere decir que el sistema oculta ciertos detalles relativos a la forma como los datos se almacenan; esto se debe a la necesidad de

diseñar estructuras complejas de datos como consecuencia de la búsqueda de la eficiencia en el almacenamiento, el acceso y la administración de la información. La abstracción se da en tres niveles:

- Nivel físico: El cual complete a la manera como realmente se almacenan los datos en los medios de almacenamiento.
- Nivel conceptual: En el que se describen los datos que realmente se almacenan y las relaciones que existen entre ellos.
- Nivel de visión: Describe sólo una parte de la base de datos la cual se puede hacer de diferentes maneras para una misma base de datos.

Ilustración 7. Niveles de un DBMS.

2.3.3. Sistemas web

En el artículo publicado por (Baez, 2012) dice que los "sistemas web" o "aplicaciones web" como también se les conoce, son aquellos que están desarrollados y desplegados no sobre un sistema operativo (Windows, Linux, etc.) o plataforma. Estos se alojan en un servidor en la nube (internet) o sobre una intranet (red local). Son muy similares a una página web, pero en realidad los “sistemas web” tienen características y funciones muy fuertes que brindan soluciones a casos particulares. Los sistemas web funcionan con bases de datos que permiten acceder, procesar y desplegar información de manera dinámica para el usuario. Los sistemas programados para plataformas web, tienen diferencias muy marcadas con otros tipos de sistemas, lo que los hace muy convenientes para las empresas que lo usan y también para los usuarios que trabajan en el sistema.

Los sistemas web modernos pueden ser accedidos desde diferentes dispositivos como teléfonos móviles, tabletas, smart tv, etc.

Ilustración 8. Sistemas web.

2.3.4. Aplicación móvil

Según (Martínez, Felipe;Campoy, 2011) un dispositivo móvil es un aparato de pequeño tamaño, con algunas capacidades de procesamiento, alimentación autónoma, con conexión permanente o intermitente a una red, con memoria limitada, diseñados específicamente para una función, pero que pueden llevar a cabo otras funciones más generales.

Las aplicaciones móviles comúnmente conocidas como “apps” por su abreviatura en inglés de “application”, son utilizadas cada vez más en teléfonos inteligentes y tabletas para el acceso a entretenimiento, noticias, juegos, estado del tiempo, etc. Estas aplicaciones de software han estado disponibles desde hace algunos años para los dispositivos móviles. Las aplicaciones móviles son los nuevos reemplazos de los programas para computadoras con fines de aumentar las capacidades informáticas creados por Microsoft, Office Suite, lectores de PDF de Adobe o videojuegos. Apple lanzó en enero de 2007 su tienda de aplicación iTunes, sus clientes empezaron a descargar las aplicaciones móviles provocando con esto que al poco tiempo Android, Rim, Nokia, etc. desarrollaran las tecnologías competidoras. (Florido Benítez, 2016).

Hoy en día una aplicación móvil puede ser no solamente para entretenimiento, sino también para las labores de trabajo, el dispositivo móvil se considera como una herramienta indispensable en el trabajo y puede ser factor clave para hacer más eficiente las actividades diarias.

2.3.5. Algoritmo de Dijkstra

Al algoritmo de Dijkstra se le conoce también como el algoritmo de caminos mínimos, es un algoritmo para encontrar el camino más corto teniendo un vértice

origen al resto de vértices en un grafo (ponderado) con pesos en las aristas. Su nombre hace referencia a Edsger Dijkstra, quien lo describió por primera vez en 1959 (EcuRed contributors, 2015).

Para el actual trabajo de tesis se utilizó el algoritmo de Dijkstra, la dinámica del algoritmo es ir explorando todos los caminos más cortos que salen del vértice origen y nos llevan a todos los vértices restantes; el algoritmo termina cuando se obtiene el camino más corto partiendo del vértice origen, visitando el resto de vértices que componen el grafo (Alcalde, 2017).

Como ya lo habíamos comentado con anterioridad una de sus aplicaciones más importantes es resolver grafos ponderados con una gran cantidad de nodos para encontrar así las rutas más cortas entre un nodo origen y un nodo destino. En la ilustración 9 se ejemplifica como se ejecuta el algoritmo y como se va resolviendo conforme se ejecutan sus diferentes pasos, en 4 imágenes podemos darnos cuenta de cómo funciona y la eficiencia del mismo.

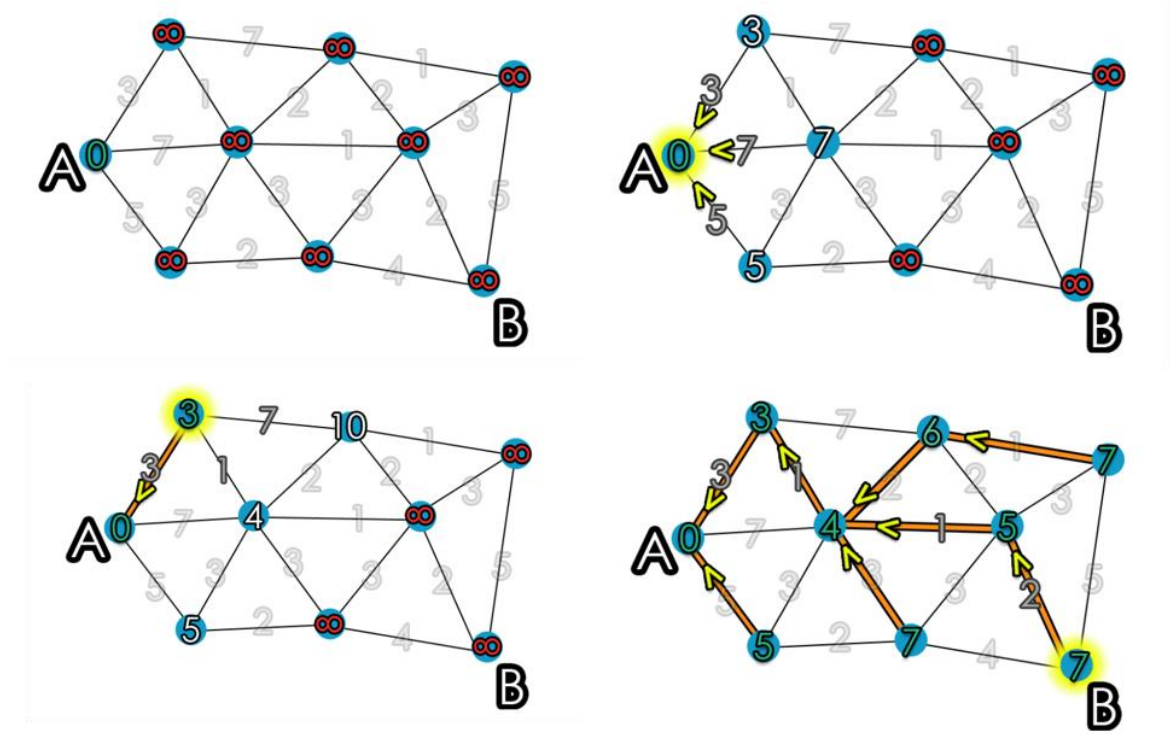


Ilustración 9. Ejemplo de funcionamiento del algoritmo de Dijkstra.

3. Métodos empleados

En este capítulo describimos la metodología utilizada, primero abordamos las dos metodologías que se utilizaron para obtener nuestra metodología híbrida con la que se desarrolló el sistema, la cual consiste de la unión de la metodología en cascada y la metodología de eXtreme Programming (XP).

En un proyecto de desarrollo de software la metodología define “quién debe hacer qué, cuándo y cómo debe hacerlo”, de tal manera que no exista ambigüedad en cuanto a las acciones a desarrollar, permitiendo llevar a buen término la culminación del mismo. No existe una metodología de software universal o general, las características de cada proyecto (equipo de desarrollo, recursos, etc.) demandan que el proceso sea configurable.

Ilustración 10. Proceso de desarrollo de software.

3.1. Metodología en cascada

Hay ocasiones en las que los requerimientos para cierto problema quedan bien comprendidos; por ejemplo, en los trabajos donde desde la comunicación hasta la implementación llevan una forma razonablemente lineal. Esta situación también es aplicable en ocasiones donde se deben hacer ajustes o actualizaciones bien definidas a un sistema previo. También ocurre en un número limitado de casos, en nuevos esfuerzos de desarrollo, con la característica de que los requerimientos están bien definidos y tienen una estabilidad razonable (Pressman, 2002).

El modelo en cascada desarrolla su proceso de manera secuencial, para el desarrollo de software los grupos de etapas deben ejecutarse unas tras otra. Se le conoce así por las posiciones que tiene las diferentes fases que forman el proyecto, ubicadas una encima de otra, y llevando un flujo de ejecución de arriba hacia abajo, como una cascada (Domínguez, 2017).

Al modelo de la cascada, se le conoce como ciclo de vida clásico, insinúa un punto de vista sistemático y secuencial para el desarrollo del software, este inicia con la definición de los requerimientos por parte del cliente y continua a través de

planeación, modelado, construcción y despliegue, para terminar con el apoyo del software terminado (ilustración 11) (Pressman, 2002).

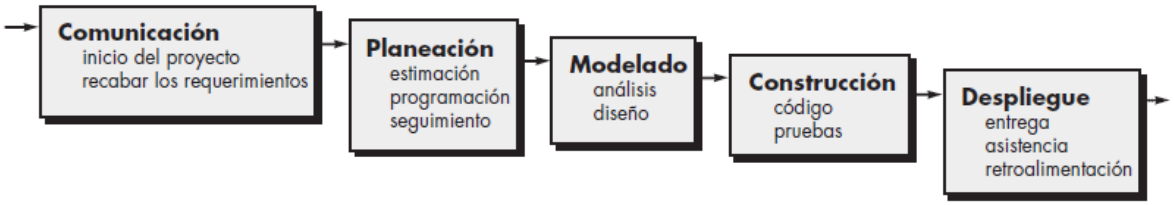


Ilustración 11. Modelo de cascada.

De acuerdo con (Sommerville, 2005) las principales etapas de este modelo se transforman en actividades fundamentales de desarrollo:

1. Análisis y definición de requerimientos. A partir de las consultas con los usuarios se definen los servicios, restricciones y metas del sistema. Después, se precisan en detalle y ayudan como una especificación del sistema.
2. Diseño del sistema y del software. Durante el proceso para el diseño del sistema se separan los requerimientos en sistemas hardware o software. Fija una arquitectura completa del sistema. El diseño del software reconoce y explica las abstracciones esenciales del sistema software y sus relaciones.
3. Implementación y prueba de unidades. A través de esta etapa, el diseño del software se realiza como un conjunto o unidades de programas. Se someten a prueba de unidades para comprobar que cada una cumpla su especificación.
4. Integración y prueba del sistema. Los programas se unen y se prueban como un sistema completo para garantizar que se obedecen los requerimientos del software. Después de terminar las pruebas, el sistema software se debe entregar al cliente.
5. Funcionamiento y mantenimiento. Esta fase casi siempre es la más larga del ciclo de vida. Se realiza la instalación del sistema y se pone en funcionamiento. El mantenimiento conlleva reparar los errores no descubiertos en las etapas previas del ciclo de vida, mejorar la implementación de las partes del sistema y destacar los servicios del sistema una vez que se encuentren nuevos requerimientos.

3.2. Metodología eXtreme Programming (XP)

(Kendall & Kendall, 2011) dice que la metodología ágil se basa no solamente en los resultados, además se centra en valores, principios y prácticas. Los principios y valores decretados son indispensables para la programación ágil; estos mismos crean el contexto para la cooperación entre clientes y programadores. Para poder

ser analistas ágiles, hay que apegarse a los siguientes principios y valores desarrollados por (Beck, 2000) en su trabajo sobre el modelado ágil, al cual denominó “programación extrema” o “XP”.

Valores del modelado ágil:

1. **La comunicación.** Existe la probabilidad de una mala comunicación en todo esfuerzo humano. Todos los sistemas que ocupan de una constante actualización y diseño técnico son particularmente propensos a tales errores. Si asociamos a esto los tiempos de entrega rigurosos, jerga especializada y el cliché de que los desarrolladores prefieren hablar con las computadoras en vez de las personas, acabamos con el potencial de encontrarnos con serios problemas de comunicación. Los proyectos pueden sufrir retrasos; se puede resolver el problema de manera incorrecta; los programadores reciben castigo incluso hasta por llevar los problemas a los gerentes; la gente sale o se une a mitad del proyecto sin las actualizaciones apropiadas, y la letanía continúa.
2. **La simpleza.** Cuando laboramos en un proyecto de programación de software, somos propensos a agobiarnos con el tamaño y la complejidad del mismo. Sin embargo, no podemos correr sino hasta aprender a caminar, ni podemos caminar hasta no aprender a erguirnos. La sencillez en el desarrollo de software expresa que debemos comenzar con la cosa más fácil que podamos hacer.
3. **La retroalimentación.** Es el tercer valor fundamental es importante cuando utilizamos una metodología de programación extrema. Al considerar la retroalimentación en este contexto, es bueno tener en cuenta que está implicada con el concepto del tiempo. Una buena respuesta concisa que sea conveniente y apropiado para el cliente, el programador y el analista puede llevarse a cabo en cuestión de segundos, minutos, días, semanas o meses, dependiendo de lo que se necesita, de quién se esté comunicando y de lo que se aspira a hacer con la retroalimentación. Un compañero programador podría pasarle un caso de prueba que rompa las líneas de código que usted acaba de redactar, pero esa ayuda es invaluable en términos de poder solucionar lo que no funciona antes de que se agregue y se encaje en el sistema.
4. **La valentía.** Es el cuarto valor en la programación ágil. La valentía se asocia con el nivel necesario de seguridad y confort que debe existir en el equipo de desarrollo. Especifica que no se debe tener miedo de desperdiciar una tarde o un día de desarrollo y empezar de nuevo si el trabajo no salió de la manera esperada. Significa tener la capacidad de estar en contacto con las corazonadas (y los resultados de prueba) en correspondencia con lo que funciona y lo que no. Valentía también significa contestar a la retroalimentación de forma concisa, actuando con base en los instintos de sus compañeros del equipo cuando los demás piensa que existe una forma mejor y simple para llegar a su objetivo.

La programación extrema mejora un proyecto de software de cinco maneras esenciales: comunicación, simplicidad, retroalimentación, respeto y coraje. Los programadores extremos se comunican constantemente con sus clientes y compañeros programadores (Wells, 2013)

El ciclo de vida idóneo para la metodología eXtreme Programming está formado de seis fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto (Letelier & Penadés, 2006).

1. **Exploración.** En la primera fase, se enfoca en los clientes que reseñan a grandes rasgos las historias del usuario que son de importancia para la primera entrega del producto. A su vez el equipo de programación se adapta con las tecnologías, herramientas y prácticas que se necesitarán en el proyecto. Se experimenta la tecnología y se investigan las posibilidades de la arquitectura del sistema construyendo un prototipo. Esta fase solo necesita de pocas semanas a pocos meses, todo depende del tamaño y familiaridad que tengan los desarrolladores con la tecnología.
2. **Planificación de la entrega.** Para esta etapa, el cliente ordena la prioridad de cada historia de usuario, y de acuerdo a eso, los programadores proyectan una estimación del esfuerzo necesario para cada una de las historias. Se realizan acuerdos sobre lo que deberá de llevar la primera entrega y se genera un cronograma en conjunto con el cliente. Se espera que la entrega deberá obtenerse en un máximo de tres meses. La duración de esta fase es de unos pocos días. Las evaluaciones de esfuerzo asociado a la implementación de las historias son establecidas por los programadores usando como medida el punto. Un punto, es igual a una semana ideal de programación. Las historias por lo general valen de 1 a 3 puntos. De igual manera el equipo de desarrollo mantiene un registro de la "velocidad" de desarrollo, medida en puntos por iteración, teniendo en cuenta principalmente la suma de puntos correspondientes a las historias de usuario que fueron finalizadas en la última iteración.
3. **Iteraciones.** En esta fase se realizan varias versiones sobre el sistema antes de ser sometido a la entrega. Para el plan de entrega las iteraciones no deben de tener más de tres semanas de diferencia entre ellas. Se debe de buscar una arquitectura del sistema que pueda ser usada durante la parte restante del proyecto en la primera. Esto se alcanza seleccionando las historias que obliguen la creación de esta arquitectura, sin embargo, esto no siempre se logra ya que es el cliente quien decide las historias que se incluirán en cada iteración (para optimizar el valor de negocio). En la última iteración el sistema deberá estar listo para ponerse en producción. Varios elementos deben ser tomados en cuenta durante la realización del plan de la iteración los cuales son: historias de usuario no planteadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no

finalizadas en la iteración anterior. Todo el trabajo de la iteración es descrito en tareas de programación, las tareas son repartidas entre los programadores y cada uno es responsable de una tarea, pero son trabajadas en parejas de programadores.

4. **Producción.** Durante la fase de producción se necesitan pruebas complementarias y revisiones de rendimiento antes de poder llevar el sistema al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la implementación de nuevas características a la versión actual, respondiendo a cambios solicitados durante esta fase. En esta fase es probable que los tiempos de entrega entre cada iteración baje a una semana. Todas las propuestas, sugerencias e son documentadas para su posterior incorporación (por ejemplo, durante la fase de mantenimiento).
5. **Mantenimiento.** Una vez que se encuentra en producción la primera versión del producto, el proyecto de eXtreme Programming debe sostener el sistema en funcionamiento al mismo tiempo que genera nuevas iteraciones. Para lograr esto se necesita de tareas de soporte para el cliente. De esta manera, la velocidad de desarrollo puede disminuir después de la puesta en marcha del sistema en producción. Es probable que en esta fase se necesite nuevo personal dentro del equipo y cambios en su estructura.
6. **Muerte del proyecto.** Una vez que el cliente ya no tiene más historias para ser incluidas en el sistema. Para esto es necesario que se hayan satisfecho las necesidades del cliente en otros aspectos como confiabilidad y rendimiento del sistema. Se escribe la documentación final del sistema y no se hacen más cambios en la arquitectura. Un proyecto también llega a su fin cuando el sistema no cumple con los requisitos establecidos, no produce los resultados esperados o cuando no hay presupuesto para mantenerlo.

3.3. Metodología en cascada con desarrollo ágil XP

Para el desarrollo de esta aplicación se utilizó la metodología híbrida entre la metodología en cascada y la metodología XP. La metodología XP es una opción ideal para el desarrollo de software porque se centra en la satisfacción del cliente y ha sido probada por muchas compañías de diferentes tamaños y giros alrededor del mundo. También se puede adaptar a los cambios de requerimientos sin afectar el desarrollo completo, con esta metodología se pudo generar entregas en determinados periodos de tiempo e ir consolidando la aceptación de la implementación de los requerimientos en el sistema (Wells, 2013).

3.3.1. Requisitos/Exploración

En esta fase, los clientes explican a grandes rasgos las historias de usuario que son de importancia para la primera entrega del proyecto. A su vez el equipo de programación se adapta con las tecnologías, herramientas y prácticas que se necesitarán en el proyecto. Se experimenta la tecnología y se investigan las posibilidades de la arquitectura del sistema construyendo un prototipo. Esta fase solo necesita de pocas semanas a pocos meses, todo depende del tamaño y familiaridad que tengan los desarrolladores con la tecnología (Letelier & Penadés, 2006).

En esta fase se llevaron a cabo las primeras reuniones con la empresa, el encargado general nos dio información de las historias de usuario que son de importancia para la primera entrega del producto. Al mismo tiempo se seleccionaron las herramientas, tecnologías y prácticas que se utilizaron en el proyecto. Se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo en base a mock-ups. La fase de exploración tomó pocos meses, debido al tamaño del proyecto y hasta que se lo logró una familiaridad con la tecnología seleccionada para el sistema.

En esta etapa es importante:

- Revisión de documentación de la empresa (manual de organización, de procedimientos, entre otros)
- Observación.

Productos de la etapa:

- **Entrevistas.** Es una herramienta que reúne información de parte de personas o de grupos. Comúnmente los entrevistados son usuarios de los sistemas actuales o usuarios en potencia del sistema propuesto. En algunas ocasiones, los entrevistados son empleados o gerentes que brindan los datos para el nuevo sistema ya que serán afectados por él (Senn, 1996).
- **Requerimientos.** El requerimiento es una propiedad que el sistema debe tener o puede ser una restricción que el sistema debe cumplir para ser aceptada por el cliente. Un requerimiento funcional describe la interacción entre el sistema y su ambiente independientemente de su implementación. Un requerimiento no funcional describe aspectos del sistema que son visibles por el usuario que no incluyen una relación directa con el comportamiento funcional del sistema (Quiroga, n.d.).
- **Cuestionarios.** Esta técnica permite obtener información que guarda relación con varios aspectos de un sistema a partir de un gran grupo de personas. El uso de formatos homogeneizados para las preguntas puede dar como resultado datos más confiables que otras técnicas; su

utilización y al no ser específico para cada usuario asegura el anonimato de los encuestados, lo que nos da como resultado respuestas más honestas (Senn, 1996).

- **Diagrama de flujo.** El diagrama de flujo también conocido como diagrama de actividades es una forma utilizada para ilustrar gráficamente un algoritmo o un proceso de cualquier tipo, a través de una serie de pasos estructurados y vinculados que posibilitan su revisión como un todo. Para representar gráficamente estos procesos en los diagramas de flujo se utiliza, una serie predeterminada de figuras geométricas que representan cada paso puntual del proceso que está siendo evaluado. Estas formas predefinidas se relacionan entre sí a través de flechas y líneas que marcan la dirección del flujo e indican el recorrido del proceso, como en el caso de un mapa (Raffino, 2018).
- **Mock-Ups.** Es una representación a escala o puede ser en tamaño real de un dispositivo o diseño que tiene como fin ser utilizado para la promoción, evaluación del diseño, demostración u otros fines. Un mockup puede ser prototipo si cuenta con al menos una parte de la funcionalidad de un sistema y permite pruebas del diseño. Los mockups se utilizan por los diseñadores generalmente para recabar comentarios por parte de los usuarios (Colaboradores de Wikipedia, 2019).

3.3.2. Planificación de la entrega

En esta fase el cliente estableció la prioridad de cada historia de usuario, y correspondientemente, se realizó una estimación del esfuerzo necesario de cada una de ellas. Se tomó un acuerdo sobre el contenido de la primera entrega y se determinó un cronograma en conjunto con el cliente. La primera entrega se realizó dentro de los primeros tres meses como se proyectó. Esta fase dura unos pocos días. En el documento en el que se indican los requisitos, se detalla una lista de los requerimientos acordados. Los desarrolladores deben entender de forma clara el producto que van a desarrollar. Esto se logra teniendo una lista detallada de los requisitos, y usando una comunicación fluida con el cliente hasta que finalice el tiempo de desarrollo (Domínguez, 2017).

Productos de la etapa:

- **Diagrama de Gantt.** El diagrama o gráfica de Gantt es una herramienta sumamente útil cuando de formulación y gerencia de proyectos se trata, ya que permite definir, de una manera gráfica, práctica y sistemática, la duración de las distintas actividades que deben ejecutarse para completar de forma exitosa un determinado proyecto (TuGimnasiaCerebral, 2019).

3.3.3. Diseño

En esta etapa se detalla la estructura interna del software, y las relaciones entre las partes que lo componen. Descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras. Es conveniente distinguir entre diseño de alto nivel o arquitectónico y diseño detallado. El primero de ellos tiene como objetivo definir la estructura de la solución (una vez que la fase de análisis ha descrito el problema) identificando grandes módulos (conjuntos de funciones que van a estar asociadas) y sus relaciones. Con ello se define la arquitectura de la solución elegida. El segundo define los algoritmos empleados y la organización del código para comenzar la implementación (Domínguez, 2017).

Productos de la etapa:

- **Diagrama arquitectónico.** En esta etapa se define la estructura global del sistema y la interrelación o comunicación de sus grandes componentes. Dichos componentes deben responder claramente a los requerimientos establecidos por el cliente.
- **Casos de uso.** Un caso de uso relata una historia detallada sobre cómo interactúa un usuario final (que tiene cierto número de roles posibles) con el sistema en situaciones específicas. La historia puede ser un lineamiento de tareas, un texto narrativo o interacciones, una descripción fundada en un formato o una representación diagramática. Sin importar su forma, un caso de uso ilustra el software o sistema desde el punto de vista del usuario final (Pressman, 2002).
- **Diagrama relacional.** Diseño de la base de datos basado en el modelo relacional postulado por Edgar Frank Codd en 1970.
- **Diccionario de datos.** Un diccionario de datos es un conjunto de definiciones que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.
- **Algoritmo de secuencia.** Conjunto de reglas que, aplicada sistemáticamente a unos datos de entrada apropiados, resuelven un problema en un numero finito de pasos elementales (Fanjul, 2018).

3.3.4. Iteraciones

Esta fase incluyó varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega estuvo compuesto por iteraciones de no más de tres semanas.

En la primera iteración se estableció una arquitectura del sistema para ser utilizada durante el resto del proyecto. Esto se logró escogiendo las historias que fuercen la creación de esta arquitectura. Al final de la última iteración el sistema estará listo para entrar en producción.

Los elementos que debieron tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior.

Todo el trabajo de la iteración fue expresado en tareas de programación, cada una de ellas es asignada a un tiempo planificado para la entrega y de cada iteración.

Productos de la etapa:

- **Sistema web.** En general, el término también se utiliza para designar aquellos programas informáticos que son ejecutados en el entorno del navegador (por ejemplo, un applet de Java) o codificado con algún lenguaje soportado por el navegador (como JavaScript, combinado con HTML); confiándose en el navegador web para que reproduzca (renderice) la aplicación (Alegsa, 2018).
- **Aplicación móvil.** Es aquella desarrollada especialmente para ser ejecutada en dispositivos móviles como un teléfono celular, tabletas y similares (Alegsa, 2017).

3.3.5. Verificación

Como su propio nombre indica, una vez que termina la fase de implementación se verifica que todos los componentes del sistema funcionen correctamente y cumplen con los requisitos.

El objetivo de las pruebas es el de obtener información de la calidad del software, y sirven para: encontrar defectos o bugs, aumentar la calidad del software, refinar el código previamente escrito sin miedo a romperlo o introducir nuevos bugs, etc. (Domínguez, 2017)

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

Productos de la etapa:

- **Pruebas.** Una vez generado el código fuente, el software debe probarse para descubrir (y corregir) tantos errores como sea posible antes de entregarlo al cliente. La meta es diseñar una serie de casos de prueba que tengan una alta probabilidad de encontrar errores (Pressman, 2002).

4. Desarrollo del sistema

En el capítulo actual se utiliza la metodología propuesta en el capítulo anterior y mostramos los resultados obtenidos durante el seguimiento y la aplicación. En la fase de exploración obtuvimos los requerimientos para el diseño del sistema usando la herramienta de entrevistas y los diagramas de flujo, los cuales nos permitieron dar una propuesta de las vistas del sistema, representadas por mock-ups. En la planificación de la entrega, en conjunto con la empresa recolectora, generamos los diagramas de Gantt. En la etapa de diseño obtuvimos el diagrama arquitectónico, los casos de uso, el diagrama relacional, el diccionario de datos y un algoritmo de secuencia para describir la solución aplicada para la generación de la ruta más corta. En la etapa de programación se obtuvo el sistema web y la aplicación móvil y en la última etapa se realizaron las pruebas de funcionamiento.

4.1. Entrevista

Durante las reuniones llevadas con la Empresa Recolectora se dieron los primeros puntos necesarios para el planeamiento y el desarrollo del sistema, la entrevista que definió las partes más importantes es la siguiente.

Entrevista para el sistema de gestión de aceites usados

1. **¿Qué espera que el sistema realice?**
   Que facilite la operación de la recolección de aceites usados y nos ayude a controlar los gastos involucrados en la operación.
2. **¿El cliente puede registrarse libremente?**
   Si, ellos pueden buscarnos o nosotros acudimos a los negocios que generan el aceite necesitado.
3. **¿Qué proceso realiza la empresa para verificar los datos del cliente e incluirlo en la ruta de recolección?**
   Ninguno. Si durante las primeras recolecciones no generan la cantidad necesaria de ACU y no son redituables dejamos de visitarlos o la recolección la hacemos menos frecuente.
4. **¿El servicio de recolección es realmente gratis o involucra pagos de alguna manera en cualquiera de las dos direcciones?**
   El recolector puede pagar el aceite a la hora de la recolección o posteriormente después de varias recolecciones. A veces se paga el aceite con otros artículos.
5. **¿Existe un mínimo de recolección?**
   20 litros.
6. **¿Se sabe la cantidad de aceite que se ha recolectado a cada cliente?**
   No.
7. **¿Se necesita controlar la venta del aceite reciclado?**
   No es necesario.

8. **Al momento de realizar el proceso con el aceite usado, ¿Se procesa por lotes de recolección o por una cantidad de litros determinados?**
   Eso es control interno de la planta de procesamiento, nosotros solo llevamos el control de la existencia.
9. **¿Necesita llevar el control de las ventas del producto terminado?**
   Sí, de ser posible.

4.2. Requerimientos

Los requerimientos de la empresa fueron muy concisos, siempre enfocados hacia los objetivos definidos en el proyecto. Enseguida nombramos los requisitos funcionales y no funcionales que corresponden al proyecto.

4.2.1. Requerimientos funcionales

En la ilustración 12 se muestran los requerimientos funcionales para el sistema definidos por la Empresa Recolectora.

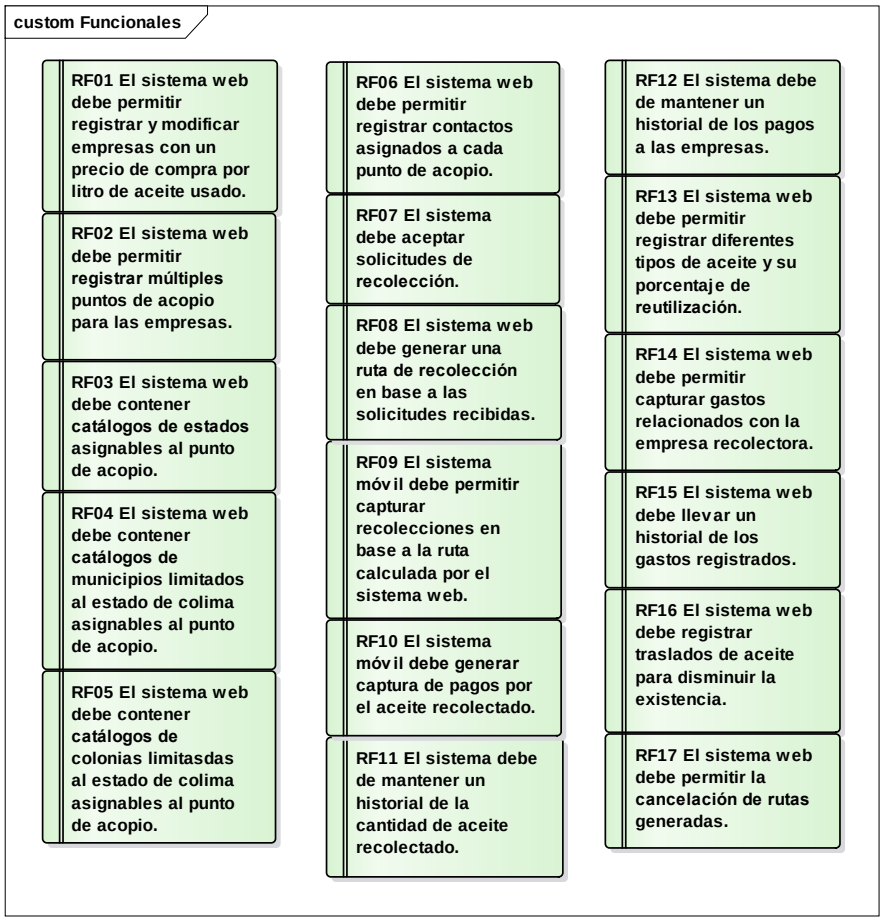


Ilustración 12. Requerimientos funcionales.

4.2.2. Requerimientos no funcionales

En la ilustración 13 vienen algunas características con las que el sistema debe cumplir sin ser obligatorias para el funcionamiento adecuado o que hayan sido requeridos por la Empresa Recolectora.

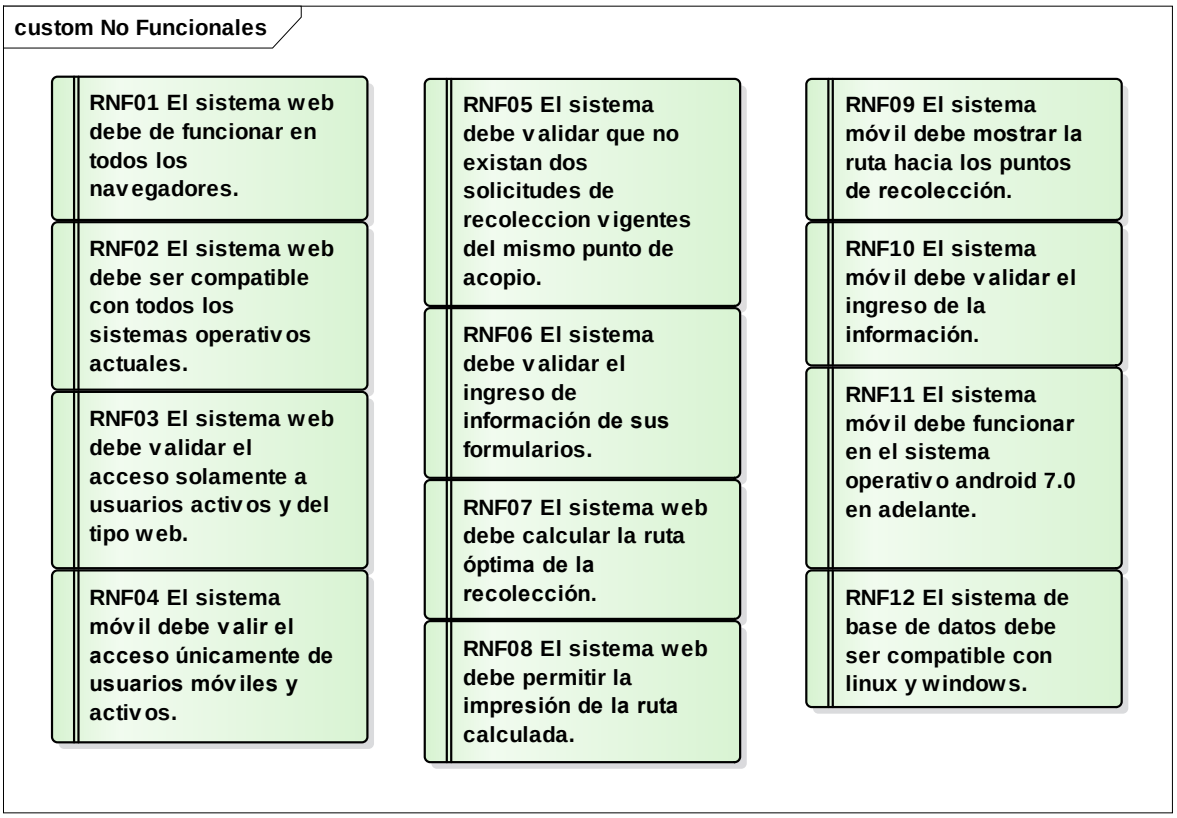


Ilustración 13. Requerimientos no funcionales.

4.3. Diagramas de flujo

Con los diagramas de flujo ilustramos el proceso que usaba la empresa recolectora para llevar a cabo su funcionamiento (ilustración 14), es un proceso que inicia con un primer contacto por parte de ellos para poder saber si es posible realizar la recolección, usualmente se hacía mediante teléfono.

Nuestra propuesta permite que la empresa recolectora haga el primer contacto y cuando se hayan generado la cantidad de solicitudes de recolección necesarias para generar una ruta esta sea planeada y calculada para tener el recorrido óptimo (ilustración 15).

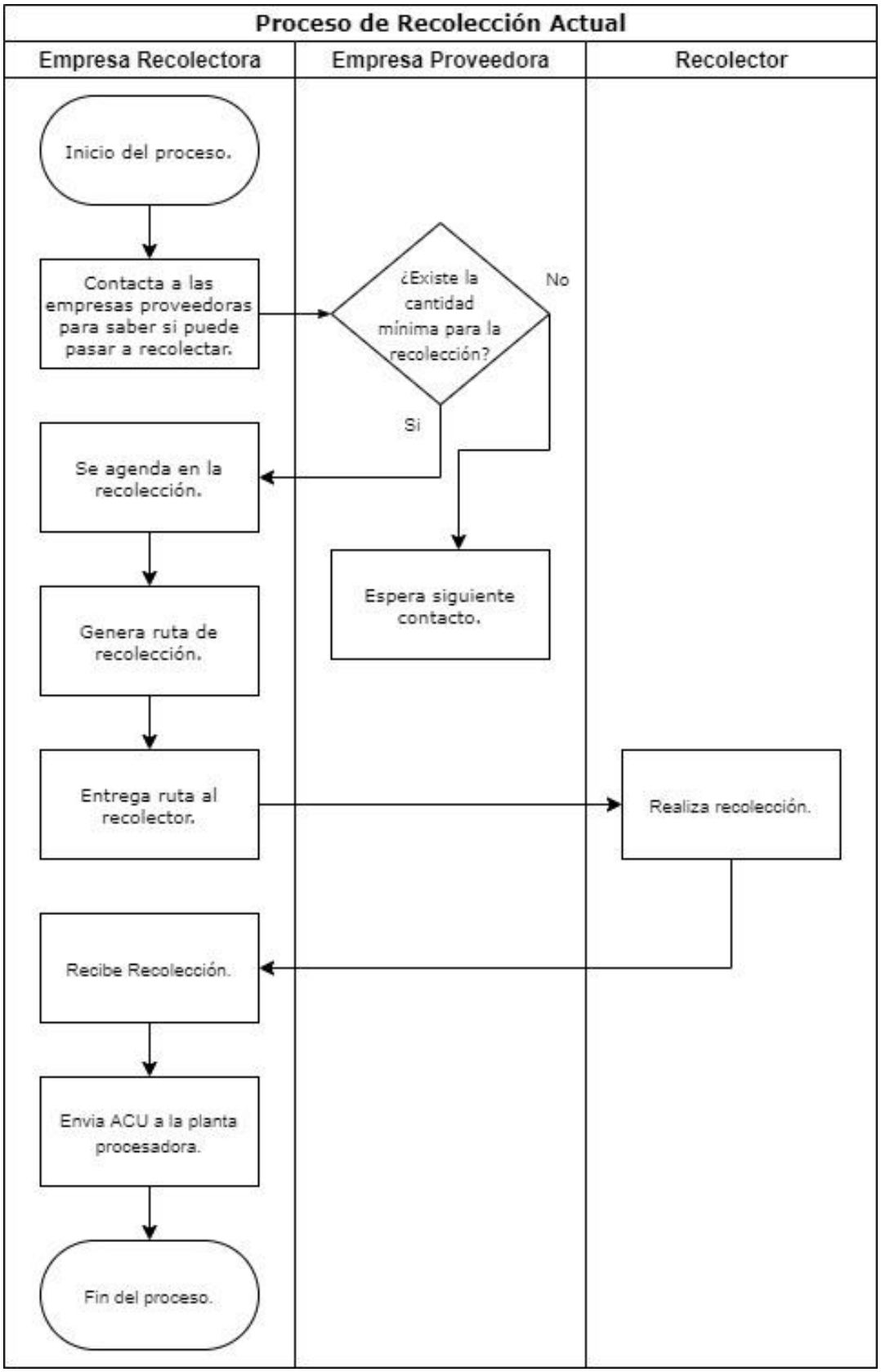


Ilustración 14. Diagrama de flujo del funcionamiento actual.

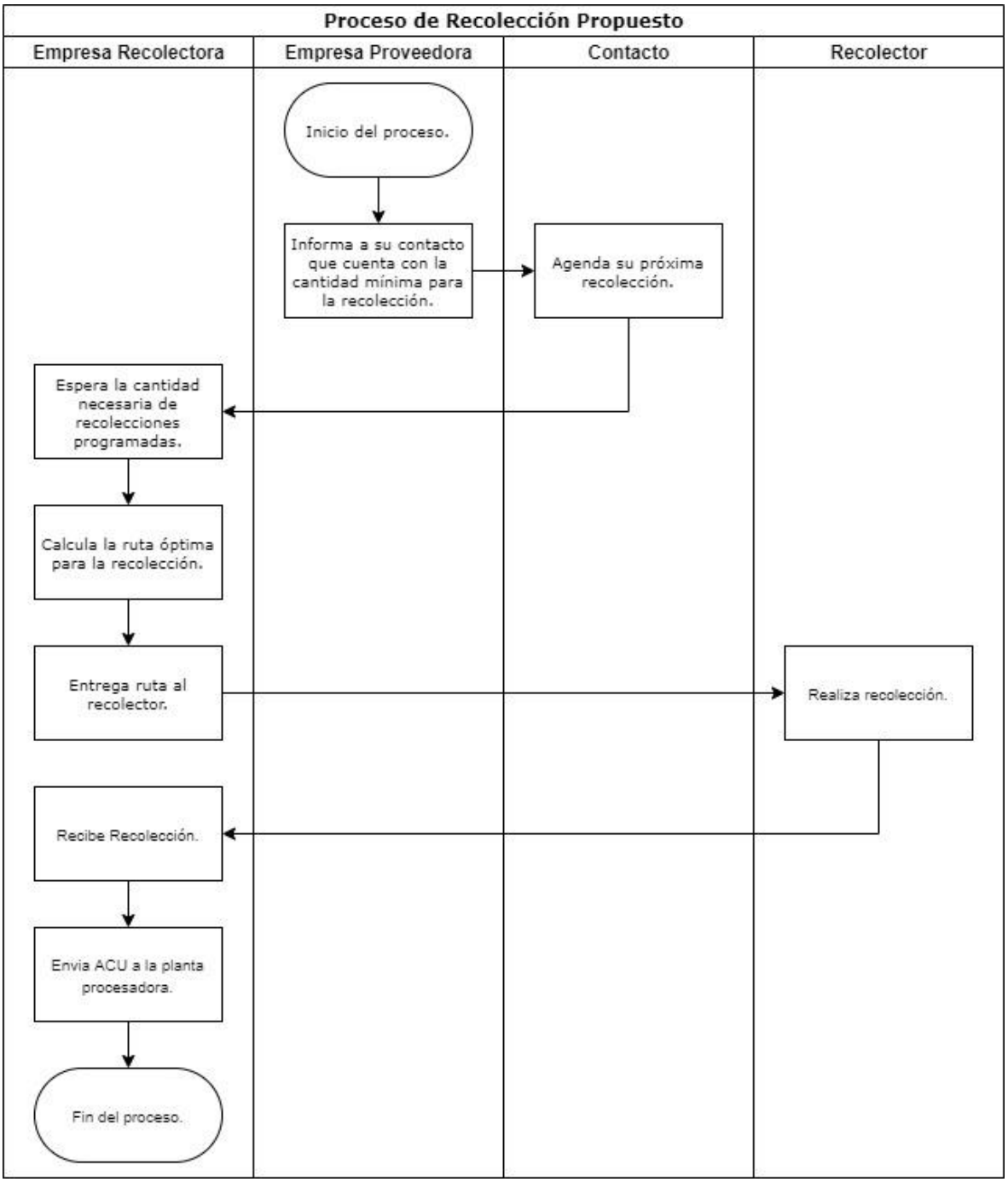


Ilustración 15. Diagrama de flujo del funcionamiento propuesto.

4.4. Mockups

Para poder mostrar a la Empresa Recolectora los posibles resultados del desarrollo del sistema se generaron mockups. En la ilustración 16 y 17 se muestran como quedarían las páginas para acceder al sistema y su página principal.

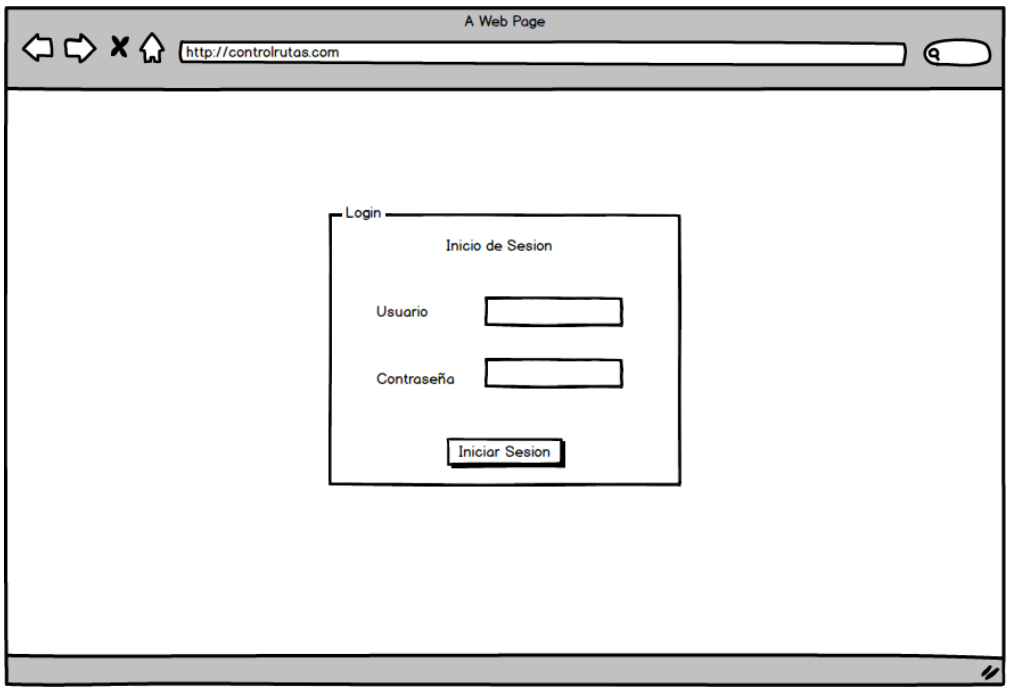


Ilustración 16. Mockup de Login.

Ilustración 17. Mockup de la página principal.

A continuación, se muestra unos ejemplos de catálogos donde se pueden dar de alta un usuario para el ingreso al sistema (ilustración 18) y como capturar un recolector que se utiliza para iniciar sesión en la aplicación móvil (ilustración 19). El

segundo hace uso del primer catálogo. Cabe mencionar que no se representaron otras interfaces como el de tipos de aceite y colonias por tener una finalidad similar.

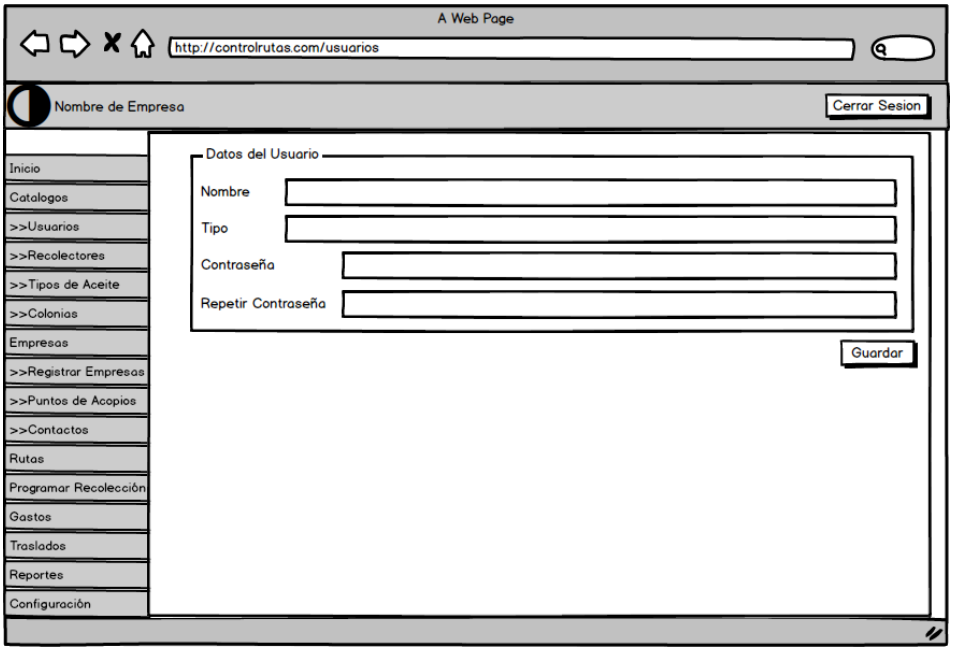


Ilustración 18. Mockup de alta de usuario.

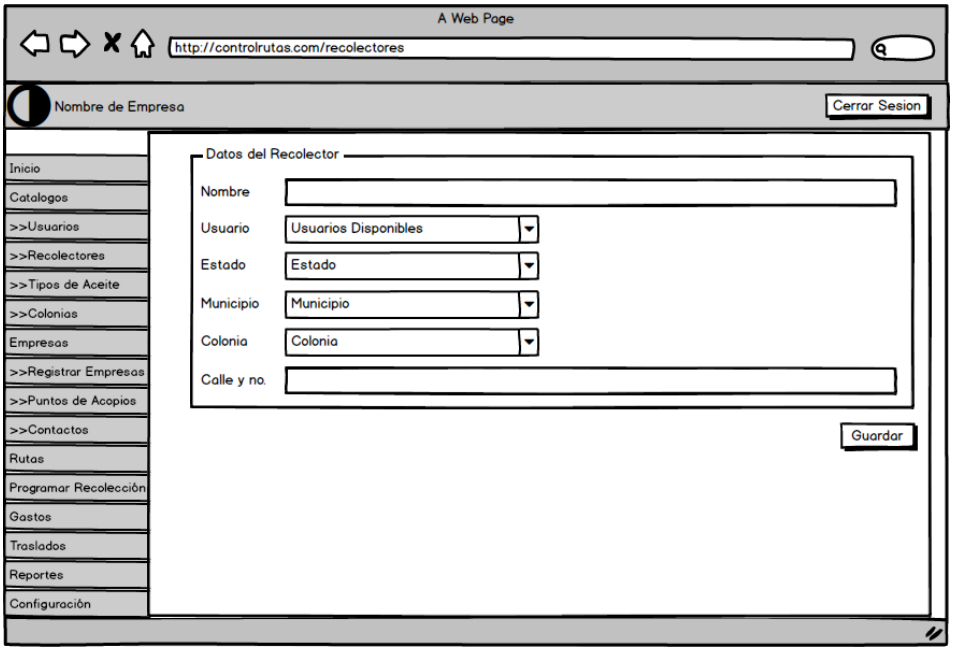


Ilustración 19. Mockup de alta de recolector.

Para el registro de empresas se representaron las tres fases, el registro de la empresa (ilustración 20), sus puntos de acopio (ilustración 21) y sus contactos (ilustración 22). Todas las empresas pueden manejar diferentes puntos de acopio

(sucursales) donde se recoja el ACU, cada punto de acopio puede contar con un contacto que haga la solicitud de recolección.

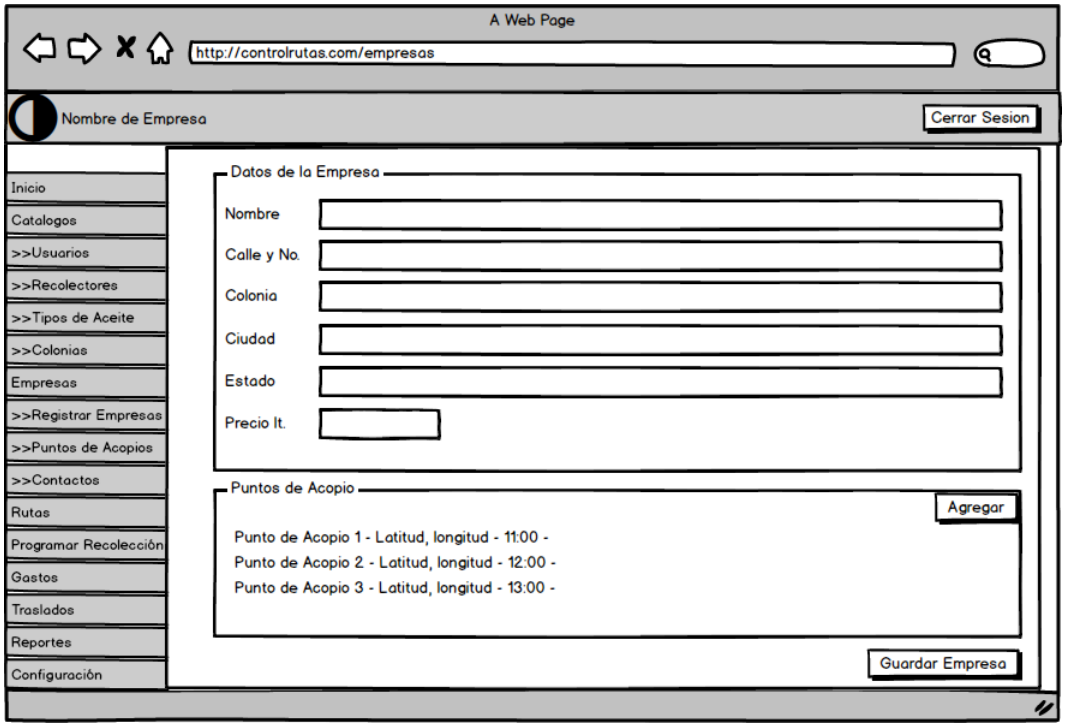


Ilustración 20. Mockup de registro de empresa.

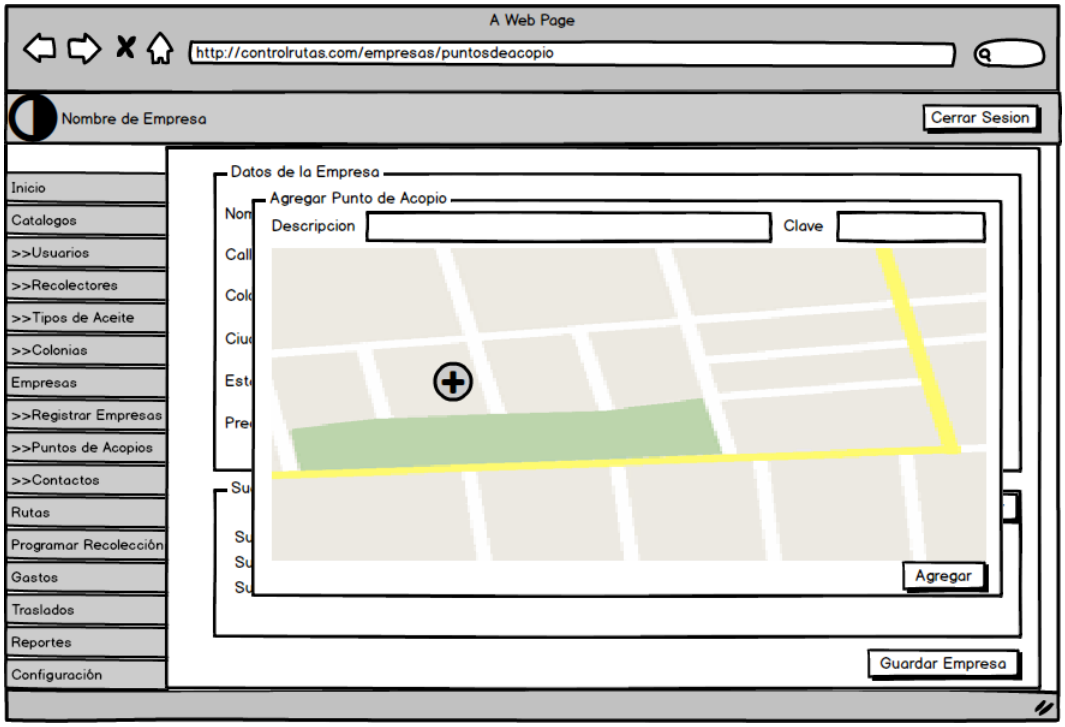


Ilustración 21. Mockup de alta de puntos de acopio.

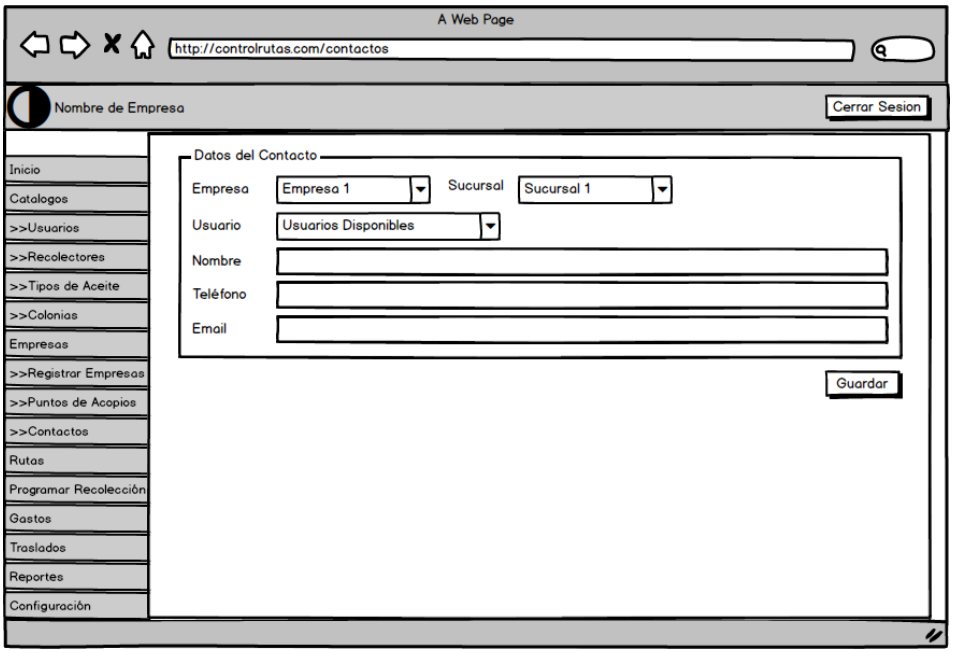


Ilustración 22. Mockup del registro de contactos.

Otro de los requisitos solicitados por la empresa recolectora fue la capacidad de poder registrar sus gastos para llevar el control de egresos, para eso se diseñó una interfaz con los campos básicos para llevar el histórico (ilustración 23).

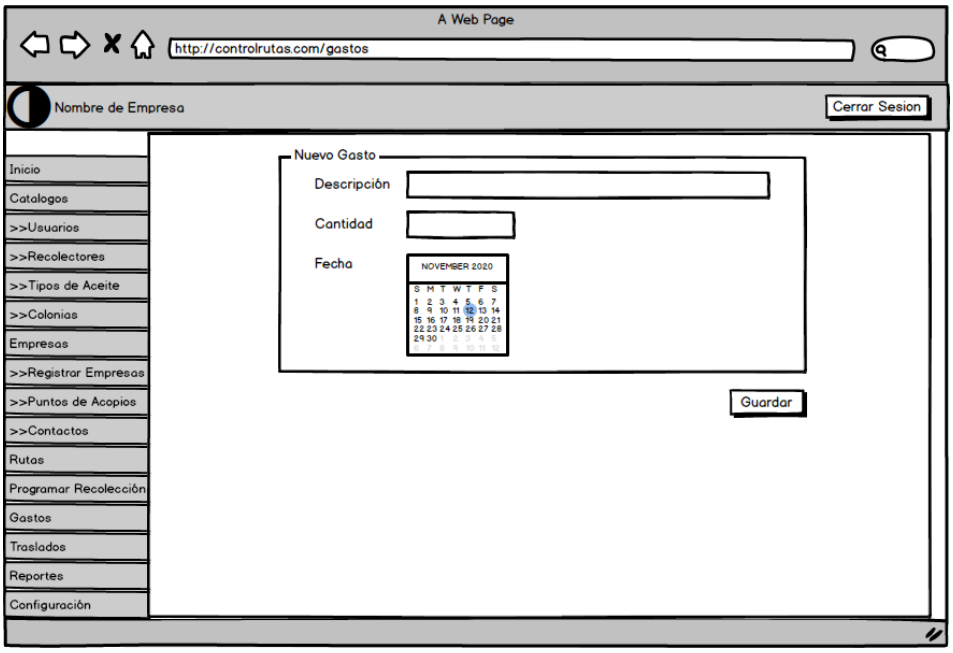


Ilustración 23. Mockup de captura de gastos.

El mock-up de la ilustración 24 ejemplifica como se agendarán las solicitudes de recolección.

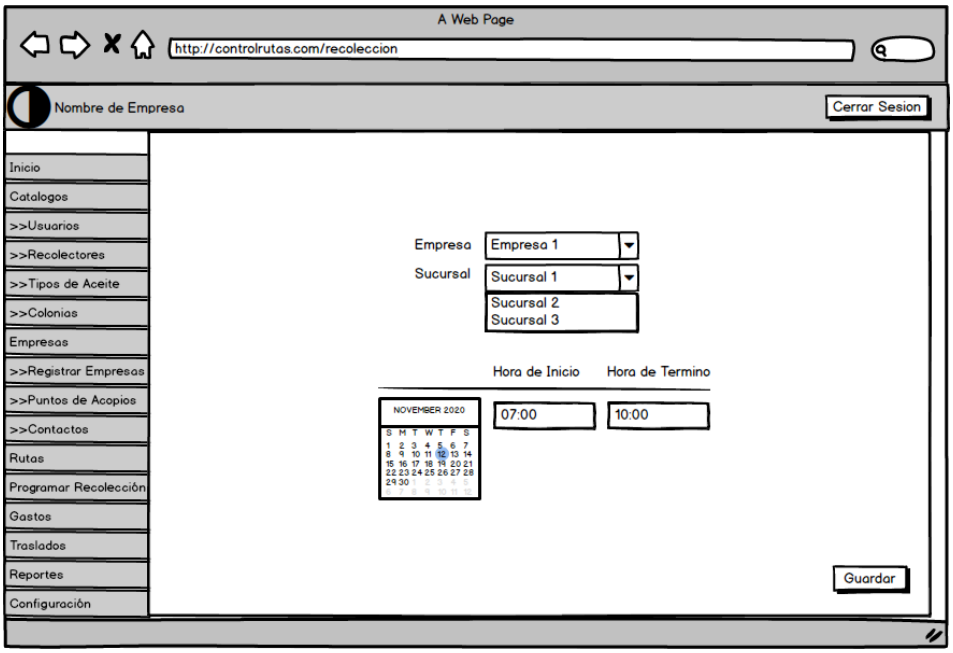


Ilustración 24. Mockup para el registro de recolecciones.

Cuando se tengan las solicitudes necesarias nos movemos al paso donde calculamos la ruta óptima (ilustración 25).

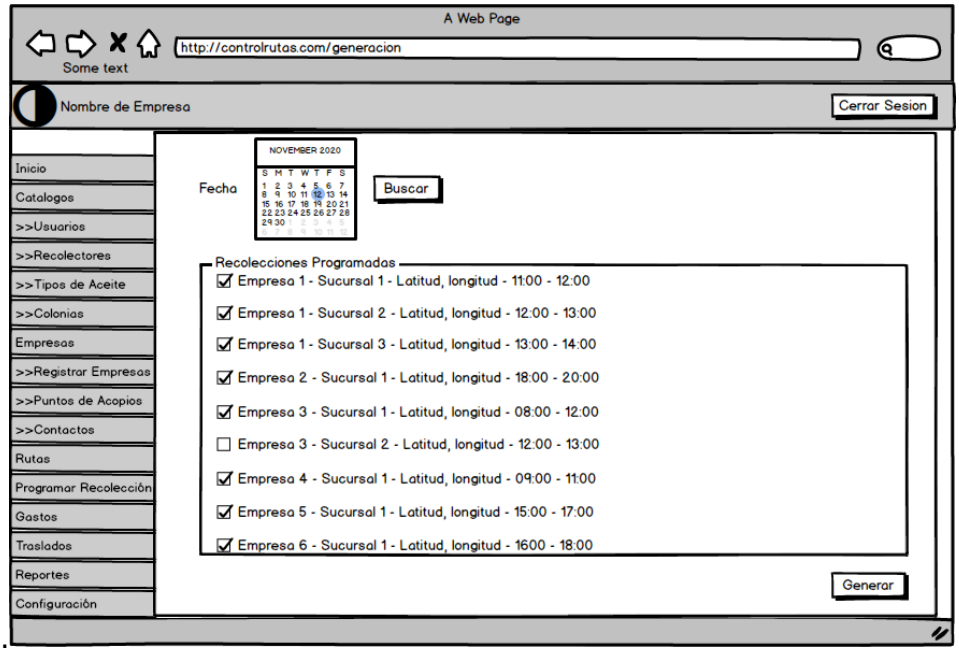


Ilustración 25. Mockup para la generación de la ruta óptima.

El mockup de la ilustración 26 nos muestra una propuesta de la ruta ya calculada.

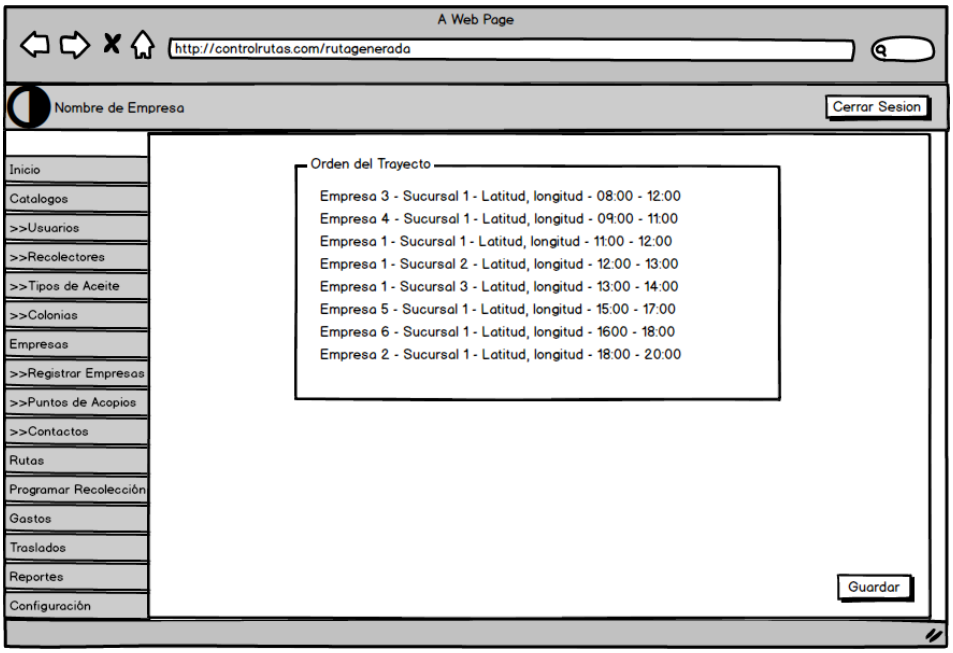


Ilustración 26. Mockup de una ruta calculada.

Una vez realizadas las recolecciones podremos filtrar los datos capturados en su debido reporte (ilustración 27).

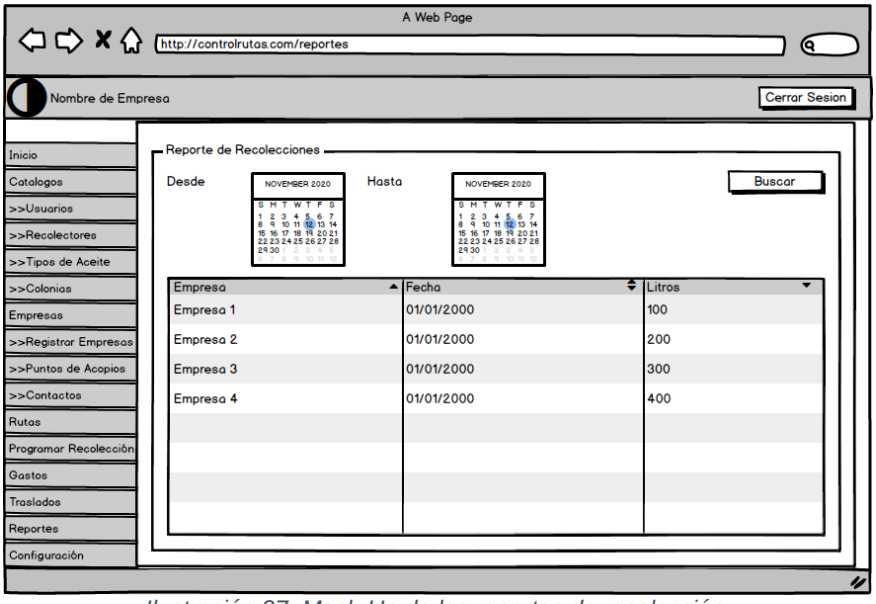


Ilustración 27. Mock-Up de los reportes de recolección.

Como última parte del proceso de la recolección de ACU tenemos la salida del aceite a la empresa procesadora. Se necesita realizar un traslado de ACU para disminuir las existencias en almacén (ilustración 28).

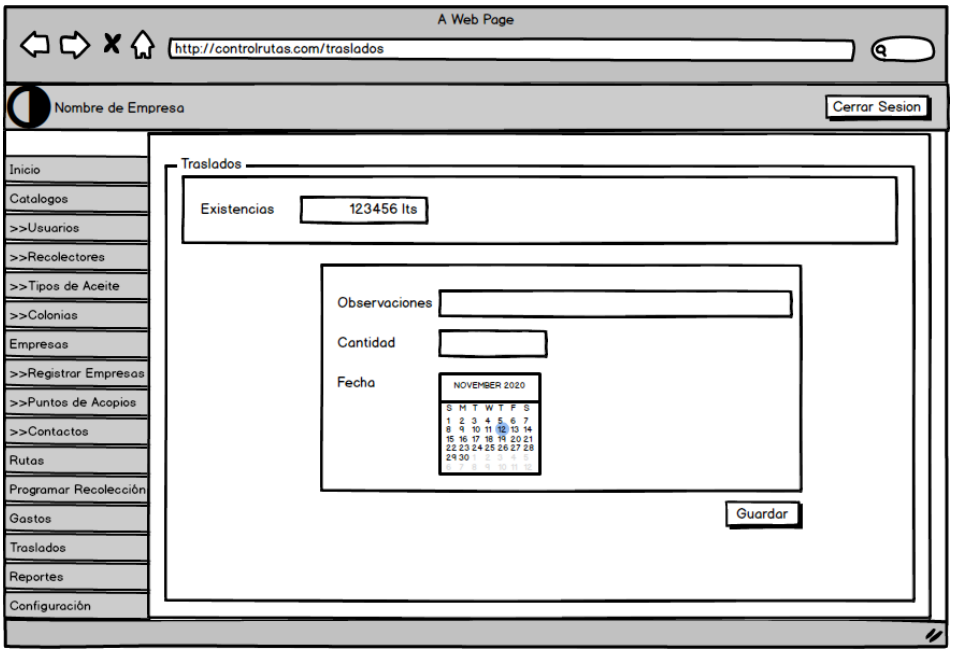


Ilustración 28. Mockup de salida del aceite del almacen.

La parte móvil representada en la ilustración 29 nos ayudará a validar el usuario recolector, selección de rutas, seguir la ruta calculada, captura de recolecciones y registrar pagos en el sistema.

Ilustración 29. Mockups de la aplicación móvil.

4.5. Diagrama de Gantt

En el diagrama de Gantt se establecen las etapas involucradas en el desarrollo del proyecto, las cuales fueron planeadas de acuerdo a los tiempos de la empresa y los cálculos estimados de entrega. El análisis duró un poco menos del tiempo planeado ya que se tuvo la completa disposición de la empresa recolectora para brindar la información necesaria. En el diseño se alargaron los tiempos debido a diversos enfoques que influyeron en el tiempo estimado para concluir dicha etapa.

La programación fue la etapa más desfasada debido a los diversos factores técnicos que alargaron la fase de escritura del código en las diferentes plataformas. Debido a esto las fases de pruebas, documentación, instalación y capacitación sufrieron retrasos como podemos observar en la ilustración 30.

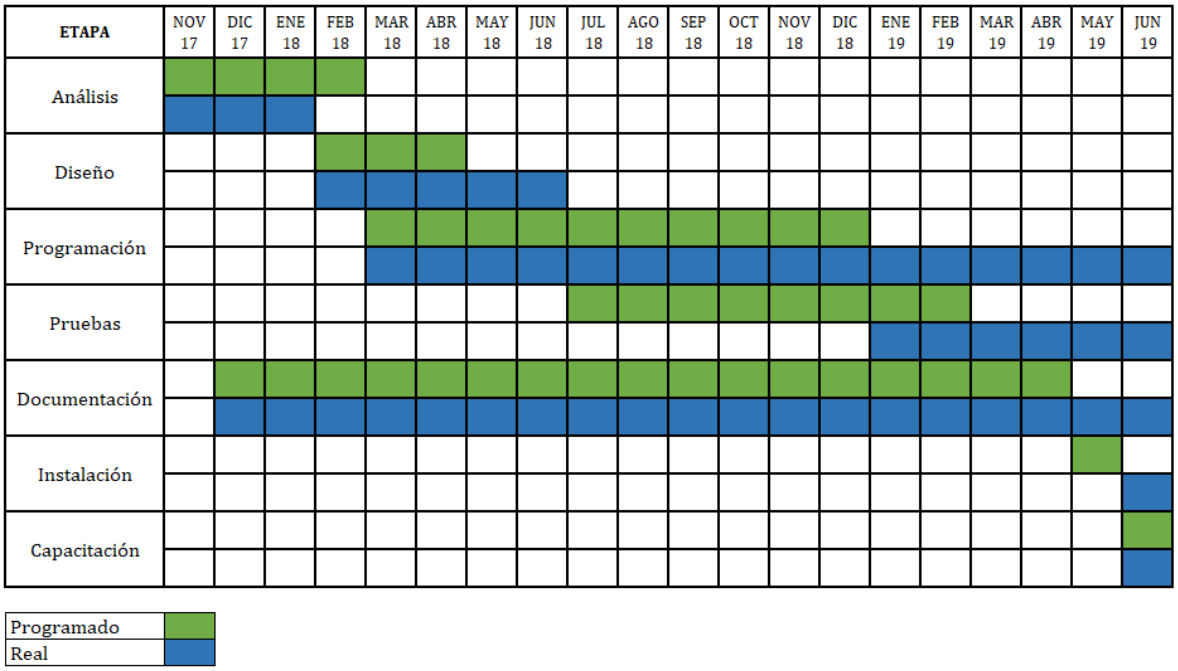


Ilustración 30. Diagrama de Gantt.

4.6. Diagrama arquitectónico

El diseño arquitectónico muestra cómo es la interacción de los 3 actores con el sistema. La empresa proveedora puede solicitar su alta en el sistema, puede solicitar su recolección a las cuales responde el administrador del sistema aceptando o denegando las solicitudes, de igual manera el administrador puede dar de alta empresas proveedoras, agendar recolecciones y generar una ruta de recolección a la cual el recolector puede acceder mediante el dispositivo móvil para capturar la información de las recolecciones de ACU.

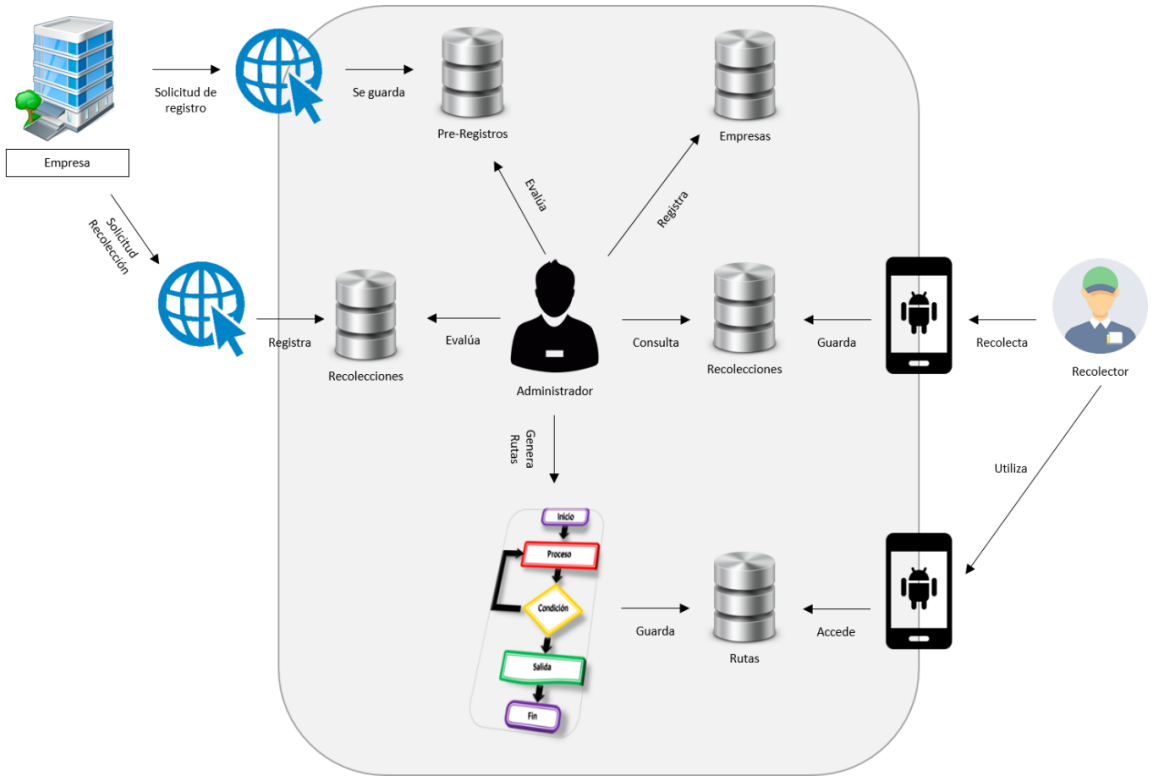


Ilustración 31. Diseño arquitectónico.

4.7. Casos de uso

Con los casos de uso se logra determinar las funciones de cada actor que interactúa con el sistema en las dos plataformas que lo componen. Al igual que en el diagrama arquitectónico (ilustración 31), la empresa proveedora puede interactuar con el sistema web solicitando su registro, puede solicitar su recolección, cancelar la misma o consultar el historial de recolecciones. Por otro lado, el recolector interactúa con la aplicación móvil recorriendo las rutas programadas, realiza pagos y captura la información de las recolecciones de ACU.

Por último, el actor principal que es el administrador del sistema puede realizar todas las funciones antes mencionadas, además de tener el control de las rutas y los catálogos del sistema.

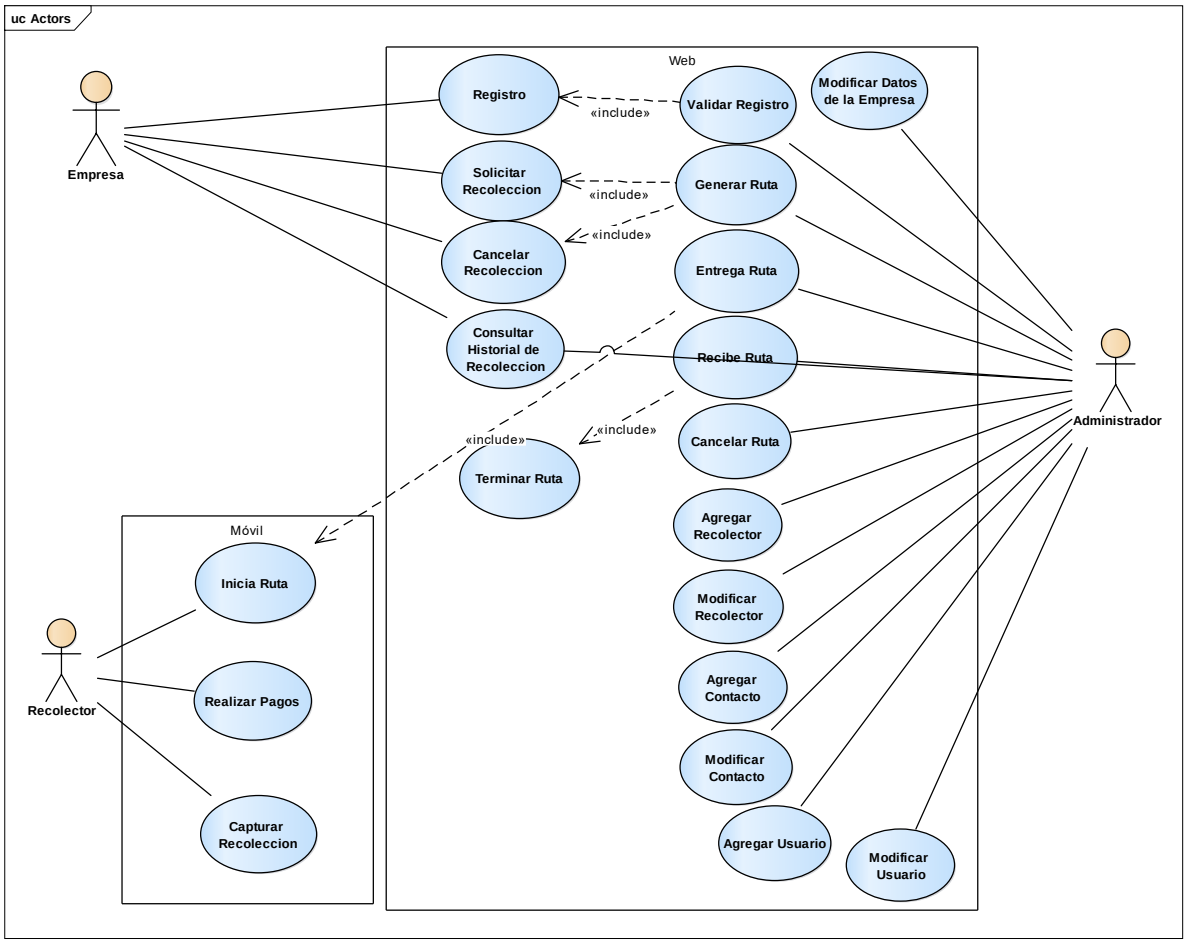


Ilustración 32. Casos de uso.

4.8. Diagrama relacional

Tras realizar el análisis y procesamiento de toda la información obtenida en las etapas anteriores, generamos el diseño de la base de datos representado por el siguiente diagrama relacional (ilustración 33) donde destacan las tablas principales como lo es los puntos de recolección, las rutas y su detalle siendo la parte principal de ambos sistemas.

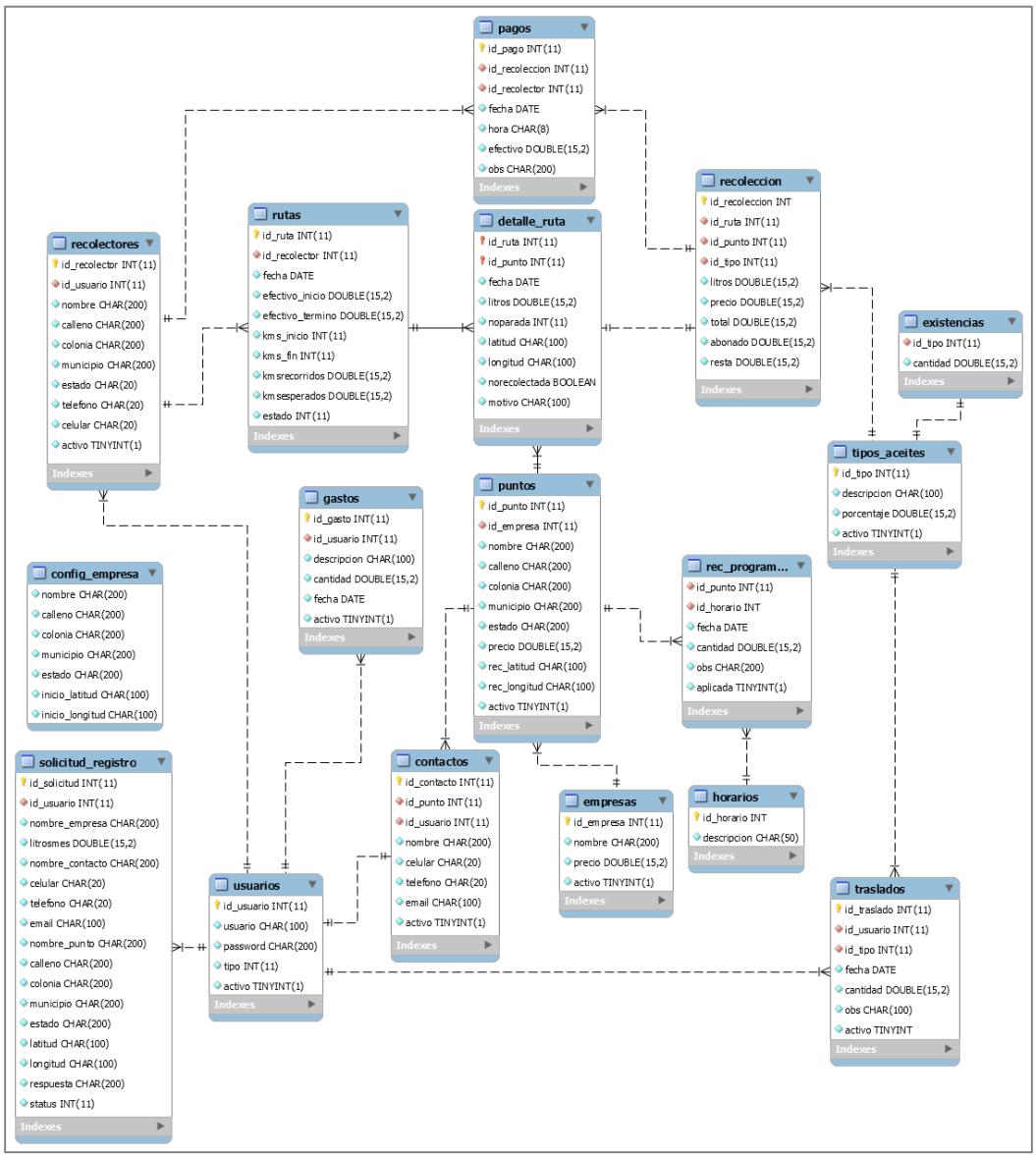


Ilustración 33. Diagrama relacional.

4.9. Diccionario de datos

El diccionario de datos contiene todas las tablas que componen el diagrama relacional especificando sus campos y las características que los identifican como lo es el nombre del campo, su descripción, su tipo de dato, formato y restricciones que tienen. También determina las llaves primarias o foráneas de cada tabla señaladas en el diagrama relacional.

TABLA CONFIG_EMPRESA
Contiene los datos relacionados con la configuración de la empresa y datos en general.

Nombre de la Propiedad	*Descripción*	*Tipo de Dato*	*Valor Nulo (S/N)*	*Restricción*	*PK, CK o FK*	*Clave Externa*
nombre	Nombre de la empresa.	char(200)	N			
calleno	Calle y número de la empresa.	char(200)	N			
colonia	Colonia de la empresa.	char(200)	N			
municipio	Municipio de la empresa.	char(200)	N			
estado	Estado de la empresa.	char(200)	N			
inicio_latitud	Latitud del punto de partida de las recolecciones.	char(100)	N			
inicio_longitud	Longitud del punto de partida de las recolecciones.	char(100)	N			

TABLA USUARIOS
Contiene los datos de los usuarios del sistema.

Nombre de la Propiedad	*Descripción*	*Tipo de Dato*	*Valor Nulo (S/N)*	*Restricción*	*PK, CK o FK*	*Clave Externa*
id_usuario	Identificador numérico del usuario.	int	N		PK	
usuario	Nombre del usuario.	char(100)	N			
password	Contraseña del usuario.	char(200)	N			
tipo	Tipo de usuario.	int	N	1 = usuario web. 2 = usuario móvil.		
activo	Estado del usuario.	tinyint	N	0 = inactivo. 1 = activo.		

TABLA SOLICITUD_REGISTRO
Contiene los datos de las empresas que llenan la solicitud de registro.

Nombre de la Propiedad	*Descripción*	*Tipo de Dato*	*Valor Nulo (S/N)*	*Restricción*	*PK, CK o FK*	*Clave Externa*
id_solicitud	Identificador numérico de la solicitud.	int	N		PK	
id_usuario	Identificador del usuario que la revisó.	int	N		FK	usuarios (id_usuario)
nombre_empresa	Nombre de la empresa.	char(200)	N			
litrosmes	Litros de ACU generados al mes.	double(15,2)	N			
nombre_contacto	Nombre del contacto.	char(200)	N			
celular	Celular del contacto.	char(20)	N			
telefono	Telefono del contacto.	char(20)	N			
email	Email del contacto..	char(100)	N			
nombre_punto	Nombre del punto de acopio.	char(200)	N			
calleno	Calle y no. del punto de acopio.	char(200)	N			
colonia	Colonia del punto de acopio.	char(200)	N			
municipio	Municipio del punto de acopio.	char(200)	N			
estado	Estado del punto de acopio.	char(200)	N			
latitud	Latitud del punto de acopio.	char(100)	N			
longitud	Longitud del punto de acopio.	char(100)	N			
respuesta	Respuesta de la solicitud.	char(200)	N			
status	Estado de la solicitud.	int	N	0 = Generada. 1 = Aceptada. 2 = Rechazada.		

TABLA EMPRESAS
Contiene los datos de las empresas registradas en el sistema.

Nombre de la Propiedad	*Descripción*	*Tipo de Dato*	*Valor Nulo (S/N)*	*Restricción*	*PK, CK o FK*	*Clave Externa*
id_empresa	Identificador numérico de la empresa.	int	N		PK	
nombre	Nombre de la empresa.	char(100)	N			
precio	Precio de compra por litro.	double(15,2)	N			
activo	Estado de la empresa.	tinyint	N	0 = inactiva. 1 = activa.		

TABLA PUNTOS
Contiene los datos de los puntos de recolección.

Nombre de la Propiedad	*Descripción*	*Tipo de Dato*	*Valor Nulo (S/N)*	*Restricción*	*PK, CK o FK*	*Clave Externa*
id_punto	Identificador numérico del punto.	int	N		PK	
id_empresa	Identificador numérico de la empresa a la que corresponde.	int	N		FK	empresas (id_empresa)
calleno	Calle y número del punto de acopio.	char(200)	N			
colonia	Colonia del punto de acopio.	char(200)	N			
municipio	Municipio del punto de acopio.	char(200)	N			
estado	Estado del punto de acopio.	char(200)	N			
precio	Precio de compra por litro.	double(15,2)	N			
inicio_latitud	Latitud del punto de acopio.	char(100)	N			
inicio_longitud	Longitud del punto de acopio.	char(100)	N			
activo	Estado del punto de acopio.	tinyint	N	0 = inactivo. 1 = activo.		

TABLA GASTOS
Contiene los datos de los gastos registrados en el sistema.

Nombre de la Propiedad	*Descripción*	*Tipo de Dato*	*Valor Nulo (S/N)*	*Restricción*	*PK, CK o FK*	*Clave Externa*
id_gasto	Identificador numérico del gasto.	int	N		PK	
id_usuario	Identificador numérico del usuario que registró el gasto.	int	N		FK	usuarios (id_usuario)
descripción	Descripción del gasto.	char(100)	N			
cantidad	Cantidad del gasto.	double(15,2)	N			
fecha	Fecha del gasto.	date	N			
activo	Estado del gasto.	tinyint	N	0 = inactivo. 1 = activo.		

TABLA CONTACTOS
Contiene los datos de los contactos de las empresas.

Nombre de la Propiedad	*Descripción*	*Tipo de Dato*	*Valor Nulo (S/N)*	*Restricción*	*PK, CK o FK*	*Clave Externa*
id_contacto	Identificador numérico del contacto.	int	N		PK	
id_punto	Identificador numérico del punto al que pertenece.	int	N		FK	puntos (id_punto)
id_usuario	Identificador numérico del usuario para iniciar sesion.	int	N		FK	usuarios (id_usuario)
nombre	Nombre del contacto.	char(200)	N			
celular	Celular del contacto.	char(20)	N			

telefono	Telefono del contacto.	char(20)	N			
email	Email del contacto.	char(100)	N			
activo	Estado del contacto.	tinyint	N	0 = inactivo. 1 = activo.		

TABLA HORARIOS
Contiene los datos de los horarios de recolección del sistema.

Nombre de la Propiedad	*Descripción*	*Tipo de Dato*	*Valor Nulo (S/N)*	*Restricción*	*PK, CK o FK*	*Clave Externa*
id_horario	Identificador horario.	int	N		PK	
descripcion	Descripción del horario.	char(50)	N			

TABLA REC_PROGRAMADA
Contiene los datos de las recolecciones programadas.

Nombre de la Propiedad	*Descripción*	*Tipo de Dato*	*Valor Nulo (S/N)*	*Restricción*	*PK, CK o FK*	*Clave Externa*
id_punto	Identificador numérico del punto de recolección.	int	N		PK	
id_horario	Identificador numérico del horario.	int	N		FK	horarios (id_ horario)
fecha	Fecha de la recolección.	date	N			
cantidad	Cantidad reportada.	double(15,2)	N			
obs	Observaciones.	char(200)	N			
aplicada	Estado de la recolección.	tinyint	N	0 = activa. 1 = aplicada.		

TABLA RECOLECTORES
Contiene los datos de los recolectores.

Nombre de la Propiedad	*Descripción*	*Tipo de Dato*	*Valor Nulo (S/N)*	*Restricción*	*PK, CK o FK*	*Clave Externa*
id_recolector	Identificador numérico del recolector.	int	N		PK	
id_usuario	Identificador numérico del usuario asignado al recolector.	int	N		FK	usuarios (id_usuario)
nombre	Nombre del recolector.	char(200)	N			
calleno	Calle y número del recolector.	char(200)	N			
colonia	Colonia del recolector.	char(200)	N			
municipio	Municipio del recolector.	char(200)	N			
estado	Estado del recolector.	char(20)	N			
telefono	Telefono del recolector.	char(20)	N			
email	Email del recolector.	char(20)	N			
activo	Estado del recolector.	tinyint	N			

TABLA TIPOS_ACEITES
Contiene los datos de los tipos de aceite.

Nombre de la Propiedad	*Descripción*	*Tipo de Dato*	*Valor Nulo (S/N)*	*Restricción*	*PK, CK o FK*	*Clave Externa*
id_tipo	Identificador numérico del tipo de aceite.	int	N		PK	
descripcion	Descripción del tipo de aceite.	char(200)	N			
porcentaje	Porcentaje del factor de conversión.	double(15,2)	N			
activo	Estado del recolector.	tinyint	N	0 = inactivo. 1 = activo.		

TABLA EXISTENCIAS
Contiene los datos de las existencias de los tipos de aceite.

Nombre de la Propiedad	*Descripción*	*Tipo de Dato*	*Valor Nulo (S/N)*	*Restricción*	*PK, CK o FK*	*Clave Externa*
id_tipo	Identificador numérico del tipo de aceite.	int	N		PK	
cantidad	Cantidad de existencia.	double(15,2)	N			

TABLA TRASLADOS
Contiene los datos de los traslados realizados.

Nombre de la Propiedad	*Descripción*	*Tipo de Dato*	*Valor Nulo (S/N)*	*Restricción*	*PK, CK o FK*	*Clave Externa*
id_traslado	Identificador numérico del traslado.	int	N		PK	
id_usuario	Identificador numérico del usuario.	int	N		FK	usuarios (id_usuario)
id_tipo	Identificador numérico del tipo de aceite.	int	N		FK	tipos_aceites (id_tipo)
fecha	Fecha del traslado.	date	N			
cantidad	Cantidad de traslado.	double(15,2)	N			
obs	Obs.	char (200)	N			
activo	Estado del traslado.	tinyint	N			

TABLA RUTAS
Contiene los datos de los rutas generadas.

Nombre de la Propiedad	*Descripción*	*Tipo de Dato*	*Valor Nulo (S/N)*	*Restricción*	*PK, CK o FK*	*Clave Externa*
id_ruta	Identificador numérico de la ruta.	int	N		PK	
id_recolector	Identificador numérico del recolector.	int	N		FK	recolectores (id_recolector)
fecha	Fecha de la ruta.	date	N			
efectivo_inicio	Efectivo de inicio.	double(15,2)	N			
efectivo_termino	Efectivo final.	double(15,2)	N			
kms_inicio	Km. inicial de la ruta.	int	N			
kms_fin	Km. final de la ruta.	int	N			
kmsrecorridos	Kms. recorridos.	int	N			
kmsesperados	Kms. calculados.	int	N			
estado	Estado de la ruta.	tinyint	N	0 = cancelada. 1 = activa. 2 = terminada.		

TABLA DETALLE_RUTA
Contiene los datos de los detalles de las rutas generadas.

Nombre de la Propiedad	*Descripción*	*Tipo de Dato*	*Valor Nulo (S/N)*	*Restricción*	*PK, CK o FK*	*Clave Externa*
id_ruta	Identificador numérico de la ruta.	int	N		FK	rutas (id_ruta)
id_punto	Identificador numérico del punto.	int	N		FK	puntos (id_punto)
fecha	Fecha de la ruta.	date	N			
litros	Litros reportados.	double(15,2)	N			
no_parada	Número de parada.	int	N			
latitud	Latitud del punto de acopio.		N			
longitud	Longitud del punto de acopio.	int	N			
norecolectada	Indicador de recolección no realizada.	tinyint	N			
motivo	Motivo por el cual no se hizo la recolección.	char(100)	N			

TABLA RECOLECCION
Contiene los datos de las recolecciones realizadas con el sistema.

Nombre de la Propiedad	*Descripción*	*Tipo de Dato*	*Valor Nulo (S/N)*	*Restricción*	*PK, CK o FK*	*Clave Externa*
id_recoleccion	Identificador numérico de la recolección.	int	N		PK	
id_ruta	Identificador numérico de la ruta.	int	N		FK	rutas (id_ruta)
id_punto	Identificador numérico del punto.	int	N		FK	puntos (id_punto)
id_tipo	Identificador numérico del tipo de ACU.	int	N		FK	tipos_aceite (id_tipo)
litros	Litros recolectados.	double(15,2)	N			
precio	Precio por litro.	double(15,2)	N			
total	Total de la compra.	double(15,2)	N			
abonado	Cantidad abonada.	double(15,2)	N			
resta	Cantidad restante de la compra.	double(15,2)	N			

TABLA PAGOS
Contiene los datos de las pagos hechos a las recolecciones.

Nombre de la Propiedad	*Descripción*	*Tipo de Dato*	*Valor Nulo (S/N)*	*Restricción*	*PK, CK o FK*	*Clave Externa*
id_pago	Identificador numérico del pago.	int	N		PK	
id_recoleccion	Identificador numérico de la recolección.	int	N		FK	recolecciones (id_recoleccion)
id_recolector	Identificador numérico del recolector.	int	N		FK	recolectores (id_recolector)
fecha	Fecha del pago.	date	N			
hora	Hora del pago.	char(8)	N			
efectivo	Dinero pagado.	double(15,2)	N			
obs	Obs.	char(200)	N			
id_pago	Identificador numérico del pago.	int	N			

4.10. Sistema web

Para lograr el objetivo general del proyecto, se desarrolló un sistema web el cual fue programado con la herramienta CodeIgniter versión 3.1.11 que es un framework que se ejecuta sobre PHP versión 5.6 o posterior. Para el manejo de la base de datos se utilizó MySQL en su versión gratuita versión 5.5. Estas herramientas nos dieron la facilidad de desarrollo y flexibilidad para poder acceder desde cualquier plataforma o sistema operativo.

En la pantalla inicial (ilustración 34) del sistema tenemos la opción de iniciar sesión; para esta pantalla existen la opción de acceder al sistema como recolector, administrador o contacto de una empresa. El botón “Solicitar Registro” permite registrar una nueva empresa por parte de la empresa generadora de ACU.

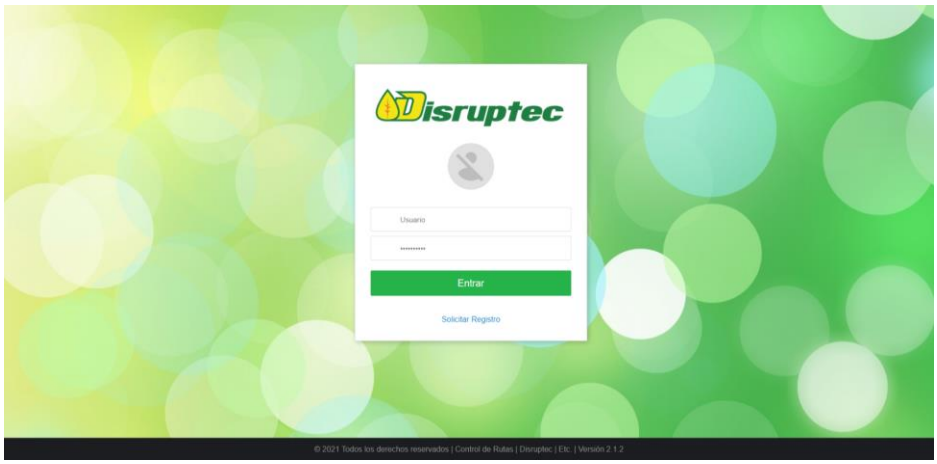


Ilustración 34. Interfaz de inicio.

Cuando una empresa desea solicitar el servicio de recolección, ingresa al enlace ubicado en “Solicitar Registro”, a continuación, debe llenar un formulario el cual consta de 4 partes: datos de la empresa, datos del contacto, datos del punto de acopio y, por último, ubicación en un mapa (ilustración 35). Esta información es revisada posteriormente por el administrador del sistema, quien se encargará de aceptar o rechazar tal solicitud.

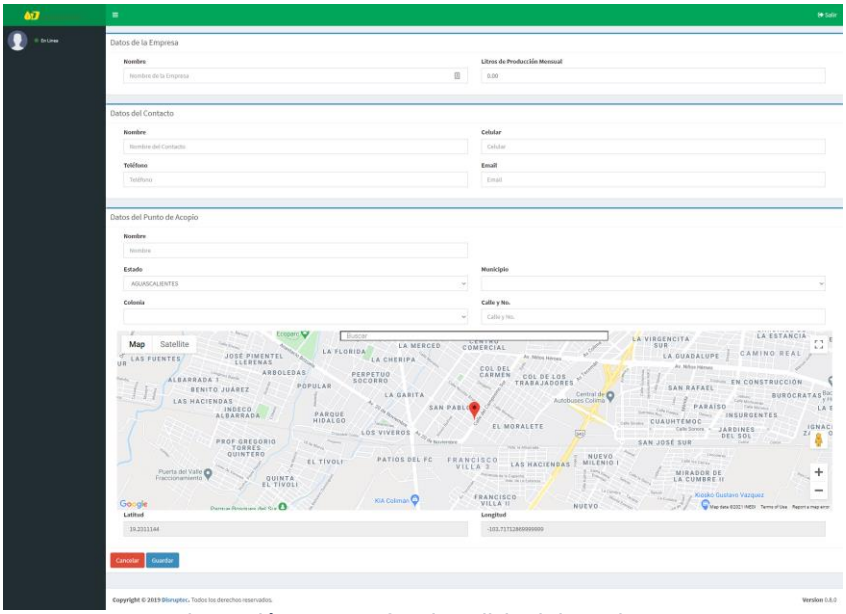


Ilustración 35. Interfaz de solicitud de registro.

Si una empresa es aceptada, se genera un nombre de usuario y contraseña de ingreso para el contacto asignado a ella y al entrar nos lleva a la interfaz de la Empresa Recolectora (ilustración 36) donde tienen la opción de agendar una recolección, cancelar la misma, observar los últimos pagos recibidos y también las últimas recolecciones.

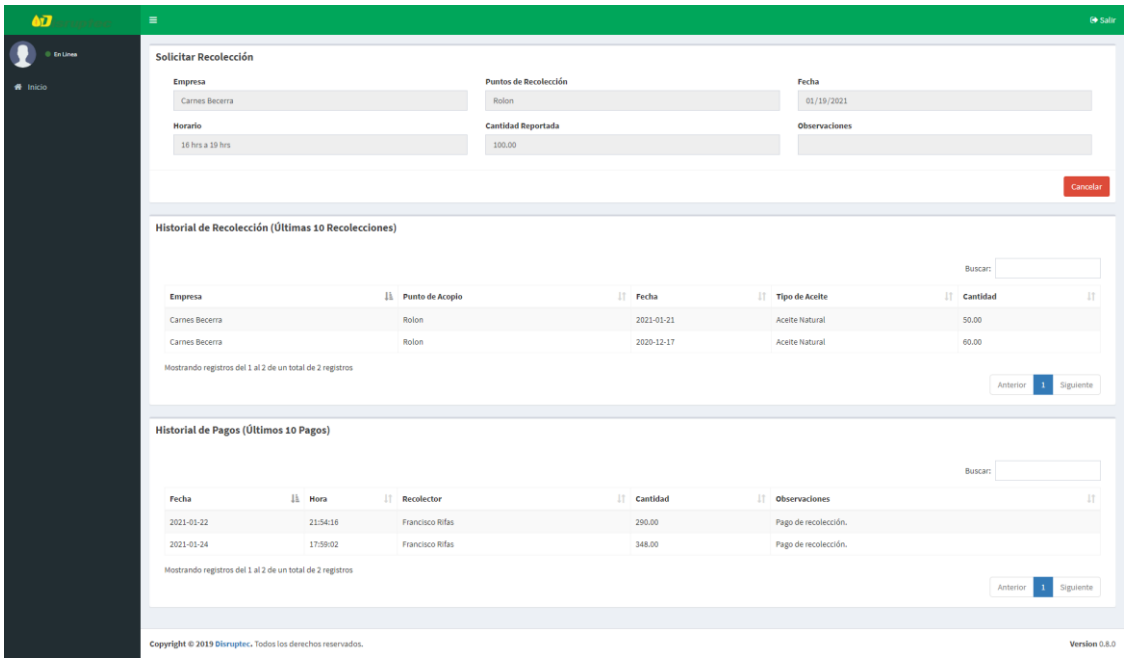


Ilustración 36. Interfaz de la Empresa Recolectora.

Al iniciar como usuario administrador del sistema nos lleva a la interfaz de inicio (ilustración 37) donde podemos ver las estadísticas de las empresas con mayor producción de ACU, también podemos ver las recolecciones agendadas en el sistema con la opción de cancelarlas y las solicitudes de registro de nuevas empresas.

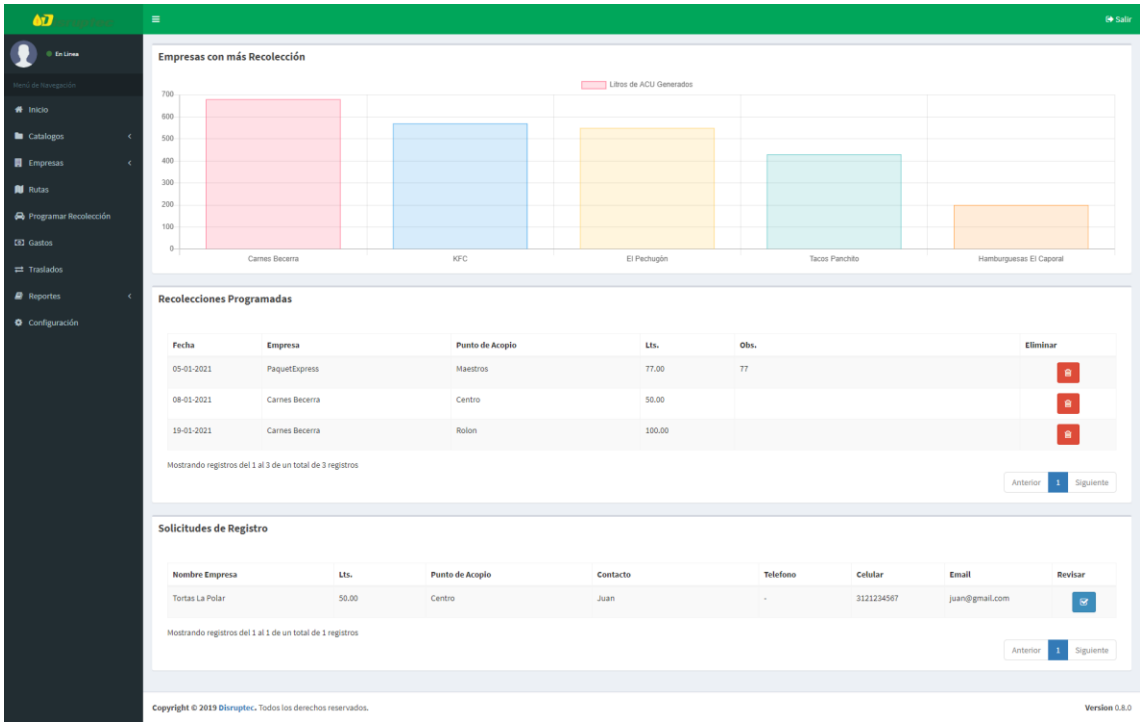


Ilustración 37. Interfaz de inicio.

Al revisar una solicitud de empresa (ilustración 38) debemos validar la información llenada para evitar errores, de ser necesario se corrigen los datos de entrada, la ubicación del punto de acopio y se llenan los campos faltantes. Los campos faltantes son aquellos que solo pueden ser llenados por la empresa recolectora como lo son, el precio de compra del litro de ACU, el usuario y la contraseña asignado al contacto de la empresa generadora para el acceso al portal.

Al final de la página (ilustración 38) tenemos dos opciones: una para rechazar y la otra para aceptar la solicitud y con esto se generan correos electrónicos para informar la decisión, en caso de ser aceptada se envía el usuario y contraseña capturados, en caso contrario se envía el motivo del rechazo.

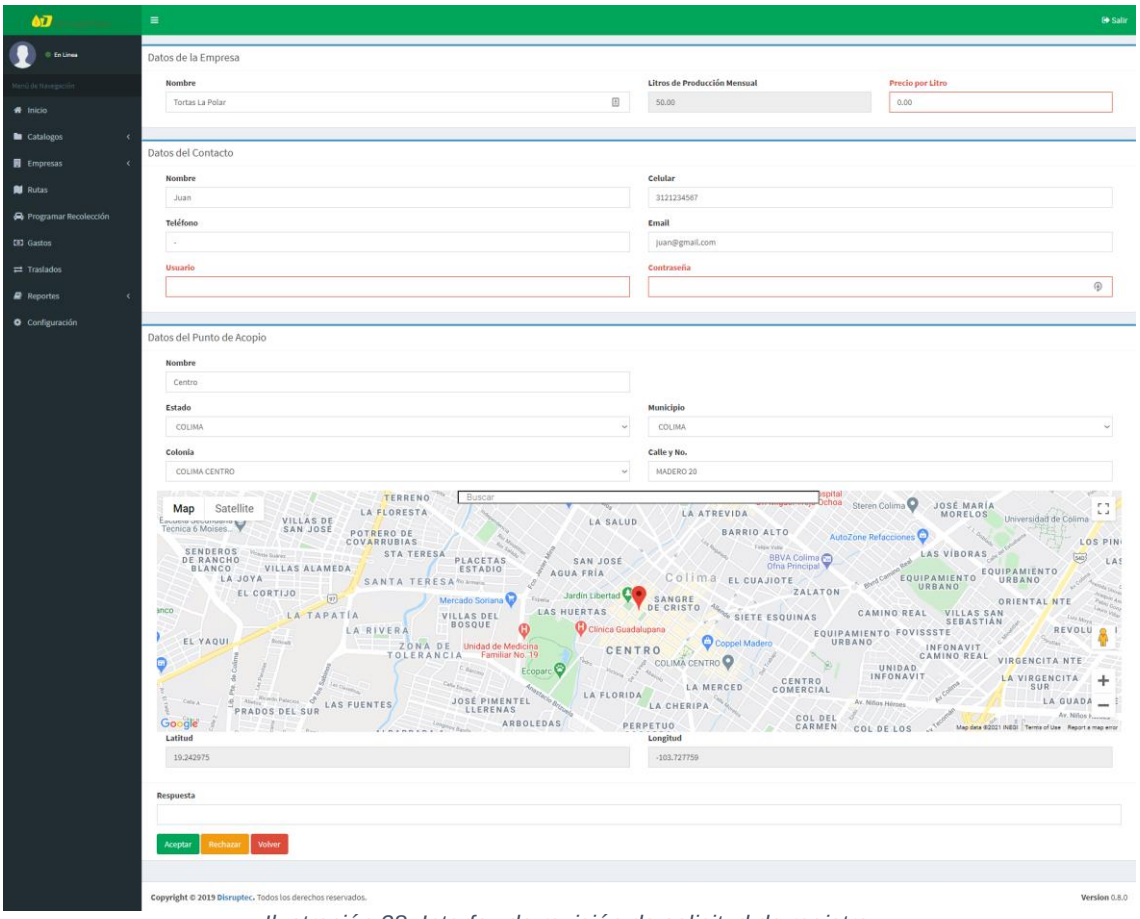


Ilustración 38. Interfaz de revisión de solicitud de registro.

Una parte básica de todo sistema web es la implementación de los listados o catálogos, donde se lleva el control de los datos capturados. El actual cuenta con la capacidad de registrar:

- Usuarios.
- Recolectores.
- Tipos de aceites.

- Colonias.
- Empresas.
- Puntos de acopio
- Contactos.

Para la generación de las rutas es necesario capturar empresas, puntos de acopio y programar recolecciones. En la interfaz de la ilustración 39 guardamos los campos básicos de una empresa como lo es su nombre y el precio sugerido al que le compraremos el litro de ACU a sus puntos de acopio. En esta misma interfaz podemos eliminar o modificar una empresa. Si una empresa es eliminada, todos sus puntos de acopio, así como sus contactos son dados de baja automáticamente.

Como parte del diseño se incluyó un mapa donde mostramos todos los puntos de acopios de las empresas para tener una pequeña noción del impacto que puede causar el sistema en el funcionamiento de la empresa recolectora.

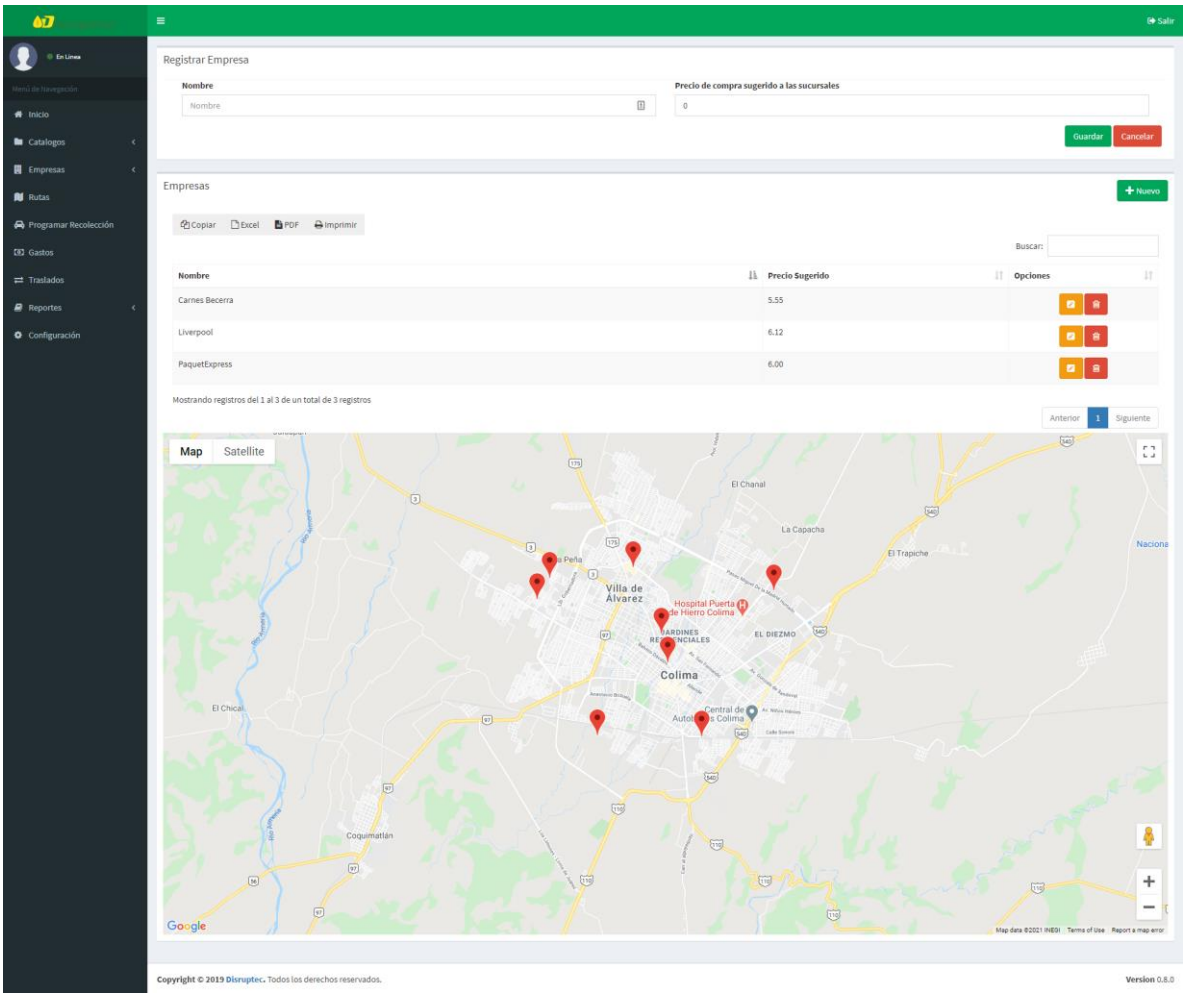


Ilustración 39. Interfaz de captura de empresa.

Como dijimos en el párrafo anterior, después de capturar los datos de una nueva empresa, podemos capturar sus puntos de acopio o dicho de una manera similar, las sucursales donde se recolectará el ACU. En la imagen 40 tenemos un ejemplo de los datos necesarios para el registro de un punto de acopio, en este proceso la parte más importante es capturar correctamente la empresa a la que corresponde el punto de acopio.

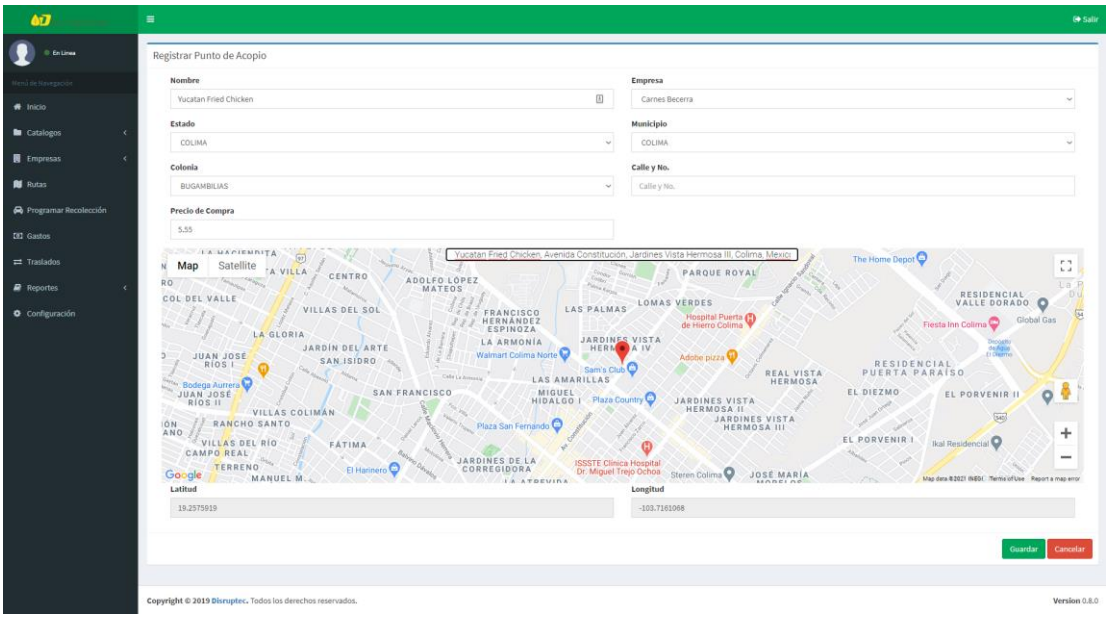


Ilustración 40 Interfaz de captura de punto de acopio.

Una vez capturados tanto los datos de la empresa como de su punto de acopio podemos capturar un contacto para cada punto de acopio y así tener acceso al panel de control antes mencionado.

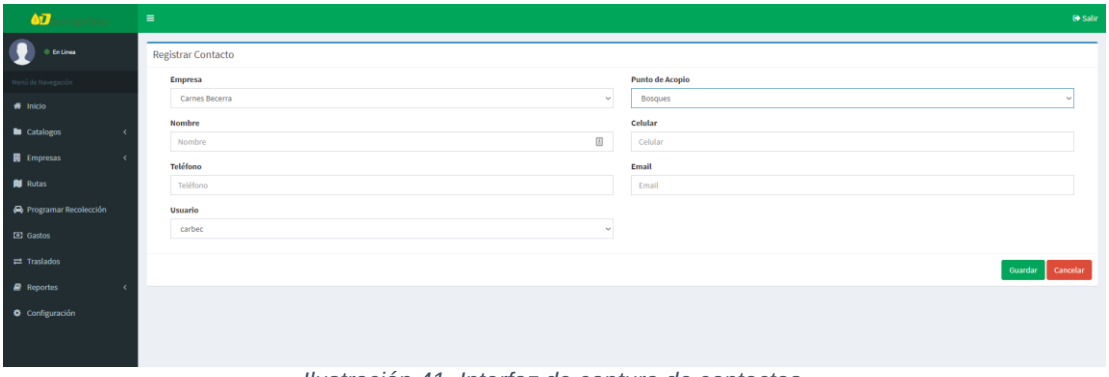


Ilustración 41. Interfaz de captura de contactos.

Antes de llegar al cálculo de una ruta debemos agendar recolecciones, las cuales alimentan el proceso de la generación de las rutas, para esto es necesario seleccionar la empresa, su punto de acopio, la fecha a partir de la cual puede realizarse la recolección, el horario deseado y algunas observaciones de ser necesarias.

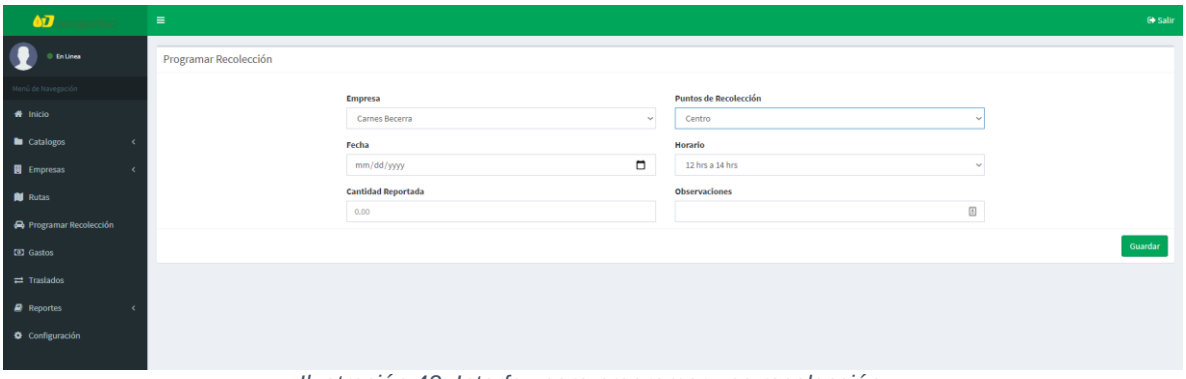


Ilustración 42. Interfaz para programar una recolección.

Una vez agendadas varias solicitudes de recolección procedemos a generar la ruta que tiene como objetivo agrupar las solicitudes en horarios similares y ejecutar el algoritmo de Dijkstra para encontrar la ruta óptima. En la ilustración 43 podemos ver las diferentes partes del proceso, las cuales se describen a continuación:

Para generar una ruta se hace el siguiente proceso:

1.- Seleccionamos una fecha y buscamos las solicitudes de recolección.
2.- Marcamos las solicitudes que serán atendidas en esa ruta.
3.- Calculamos la ruta.
4.- Indicamos el recolector asignado a esa ruta, el kilometraje con el que el vehículo iniciará la ruta y el efectivo que se le entrega al recolector.
5.- Guardamos la ruta y si es necesario la imprimimos.

Una vez calculada la ruta obtenemos un mensaje (ilustración 44) indicando que el proceso se ha ejecutado de manera correcta y se dibuja la ruta sobre el mapa de google (ilustración 45).

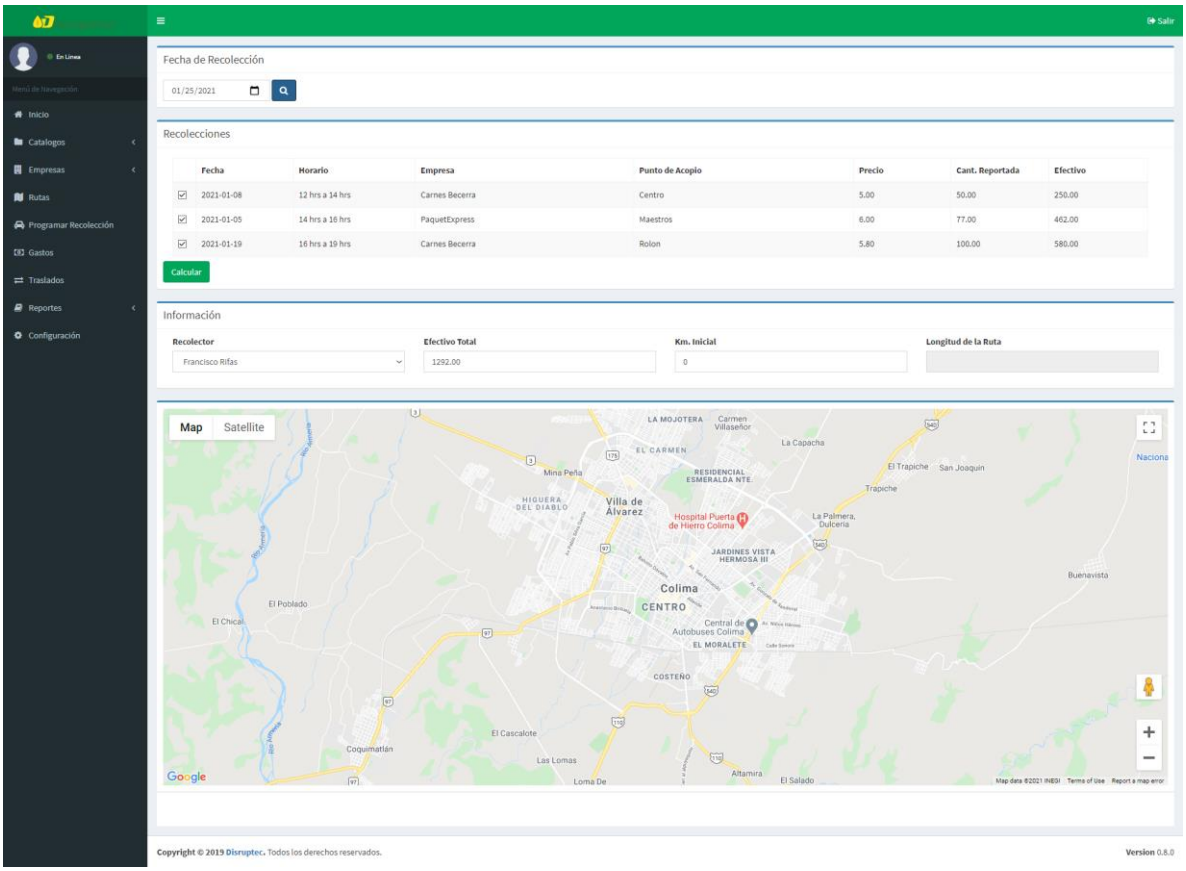


Ilustración 43. Interfaz de generación de rutas.

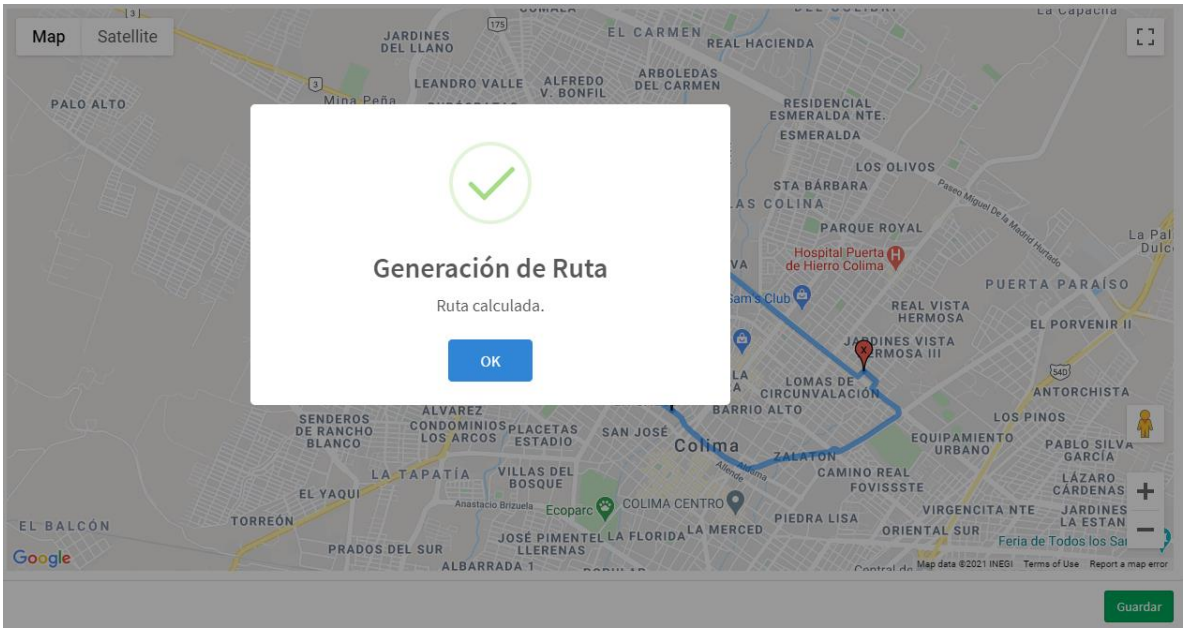


Ilustración 44. Resultado del cálculo.

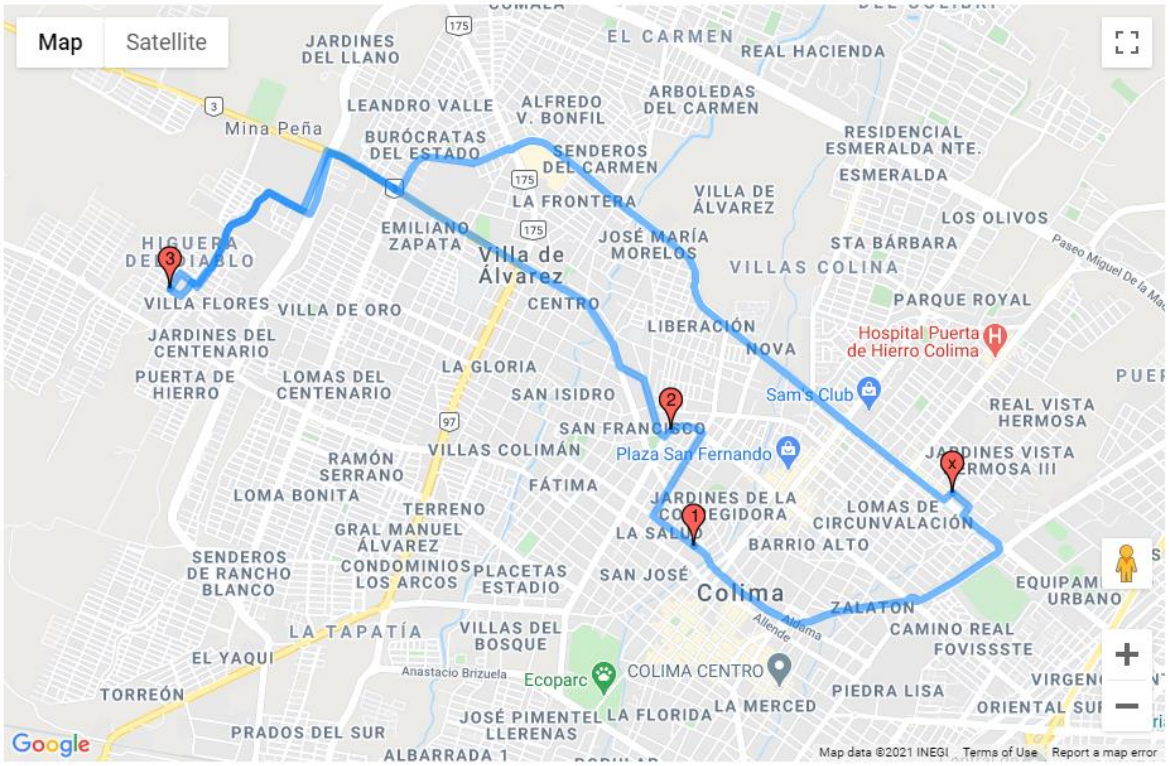


Ilustración 45. Ruta dibujada de google maps.

Una vez guardada la ruta, el proceso continúa en la aplicación móvil, donde el recolector se encarga de alimentar el sistema recorriendo las paradas que corresponden a cada punto de acopio, la información es introducida al sistema y al final del recorrido se entrega la ruta para hacer cuentas con el administrador de la Empresa Recolectora.

En esta parte (ilustración 46) el administrador del sistema valida la información de las recolecciones y contrasta el dinero pagado por los litros de ACU contra lo entregado al inicio de la recolección; además de que toma medida del kilometraje final del vehículo para verificar que los kilómetros recorridos sean similares a los esperados por la ruta calculada.

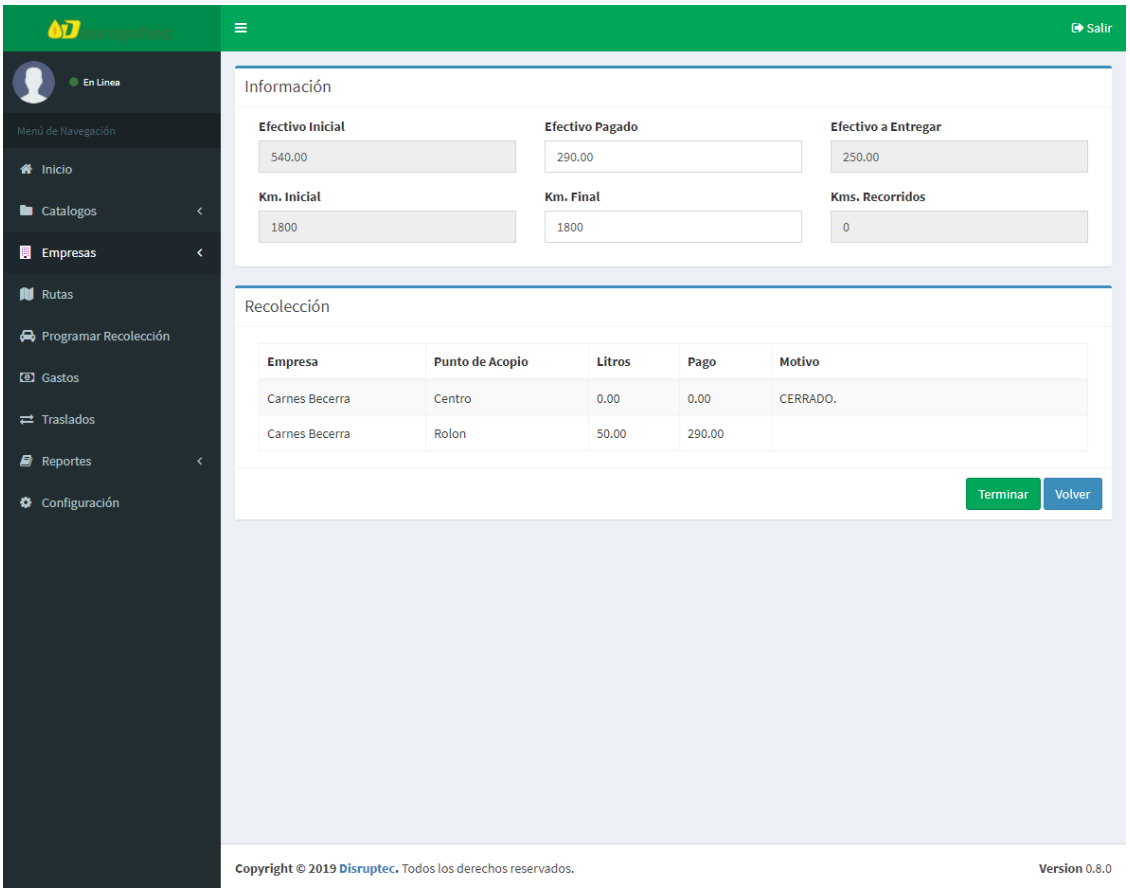


Ilustración 46. Recibimiento de ruta.

Como parte final del proceso es poder acceder al histórico de las recolecciones, podemos configurar los filtros utilizando los parámetros de recolector, empresa, punto de acopio y un rango de fecha determinado.

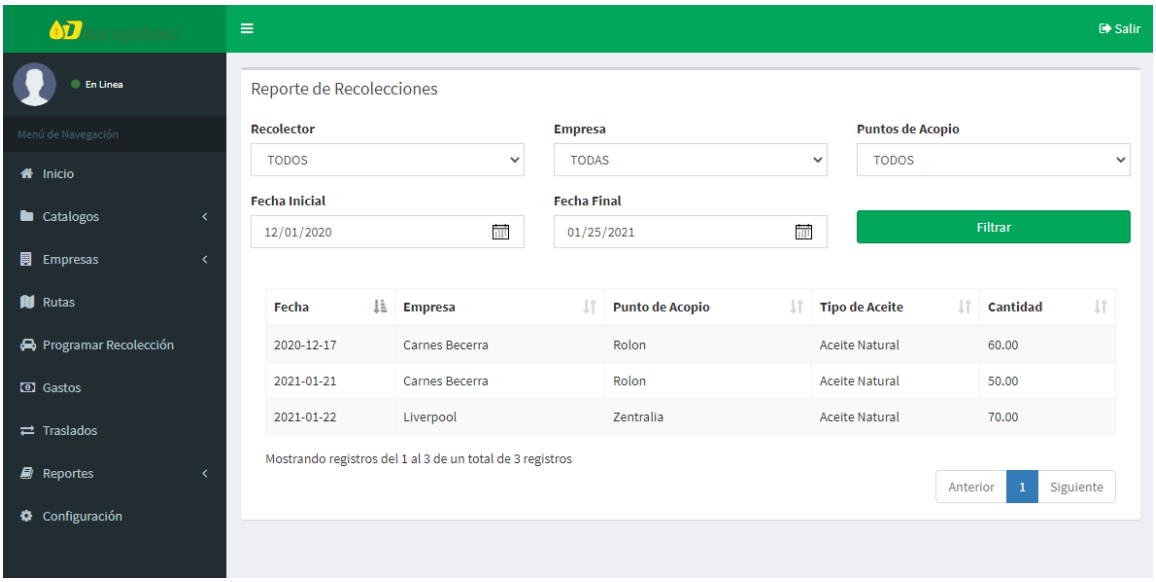


Ilustración 47. Interfaz de reporte de recolecciones.

4.11. Aplicación móvil

La aplicación móvil está compuesta por una serie de interfaces dedicadas a la captura de la información generada por las recolecciones. El primer paso es iniciar sesión en el sistema, en la pantalla principal es necesario proporcionar el usuario y contraseña en los campos indicados (ver ilustración 48) y a continuación presionar el botón de iniciar sesión para hacer el login.

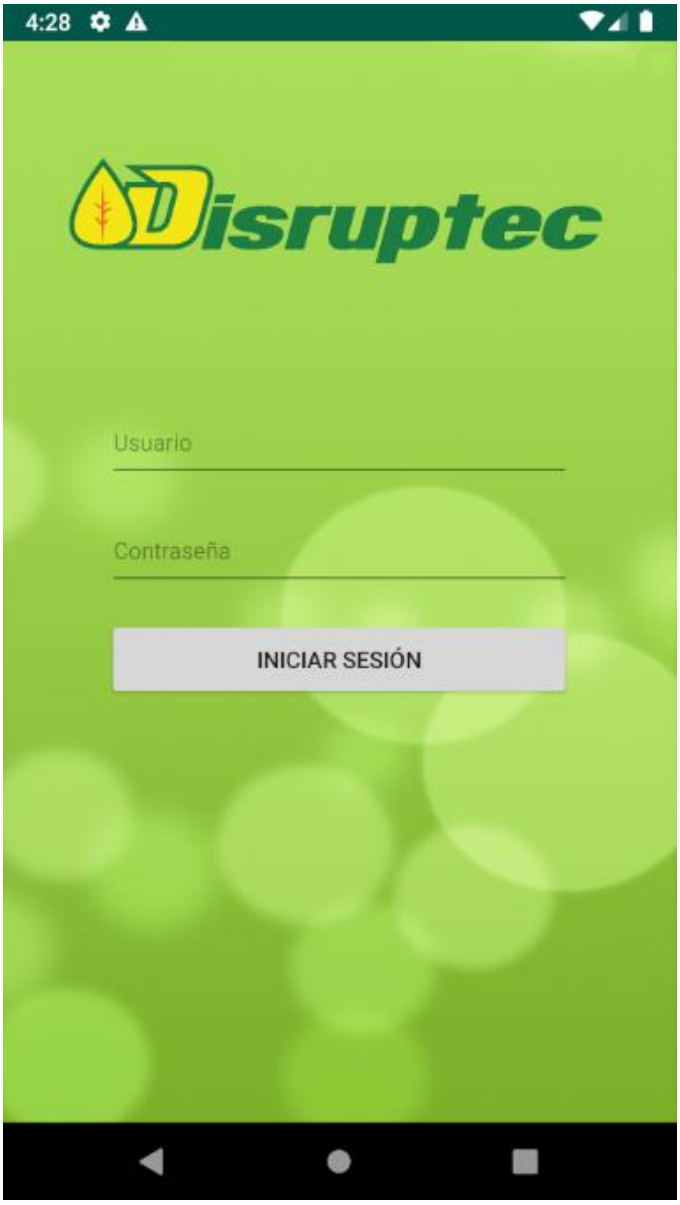


Ilustración 48. Interfaz de login.

La siguiente pantalla que se nos presenta corresponde a nuestra pantalla principal (ver ilustración 49) donde se despliega el nombre del recolector y las rutas activas que tiene en ese momento, si se presiona el botón de CERRAR SESION

volveremos a la pantalla de login de la ilustración 48. Al presionar una de las rutas activas nos envía a la siguiente interfaz de la ilustración 50 donde podemos ver las paradas de la ruta.

Ilustración 49. Interfaz de rutas activas.

Una vez en la interfaz de las paradas de la ruta tenemos 2 opciones: una es activar el mapa para que nos despliegue la ruta óptima hacia nuestra siguiente ubicación (ilustración 51) y la otra consiste en realizar la parada presionando el ícono de la libretita la cual nos mandará a la parte que consiste en la captura de los datos en el momento en el que se efectúa la recolección del aceite en el punto de acopio de la Empresa Proveedora.

Ilustración 50. Interfaz de paradas en la ruta.

La ilustración 51 muestra la interacción de nuestro sistema con la función de Google Maps integrado en el sistema operativo Android, las indicaciones son desplegadas automáticamente por la aplicación.

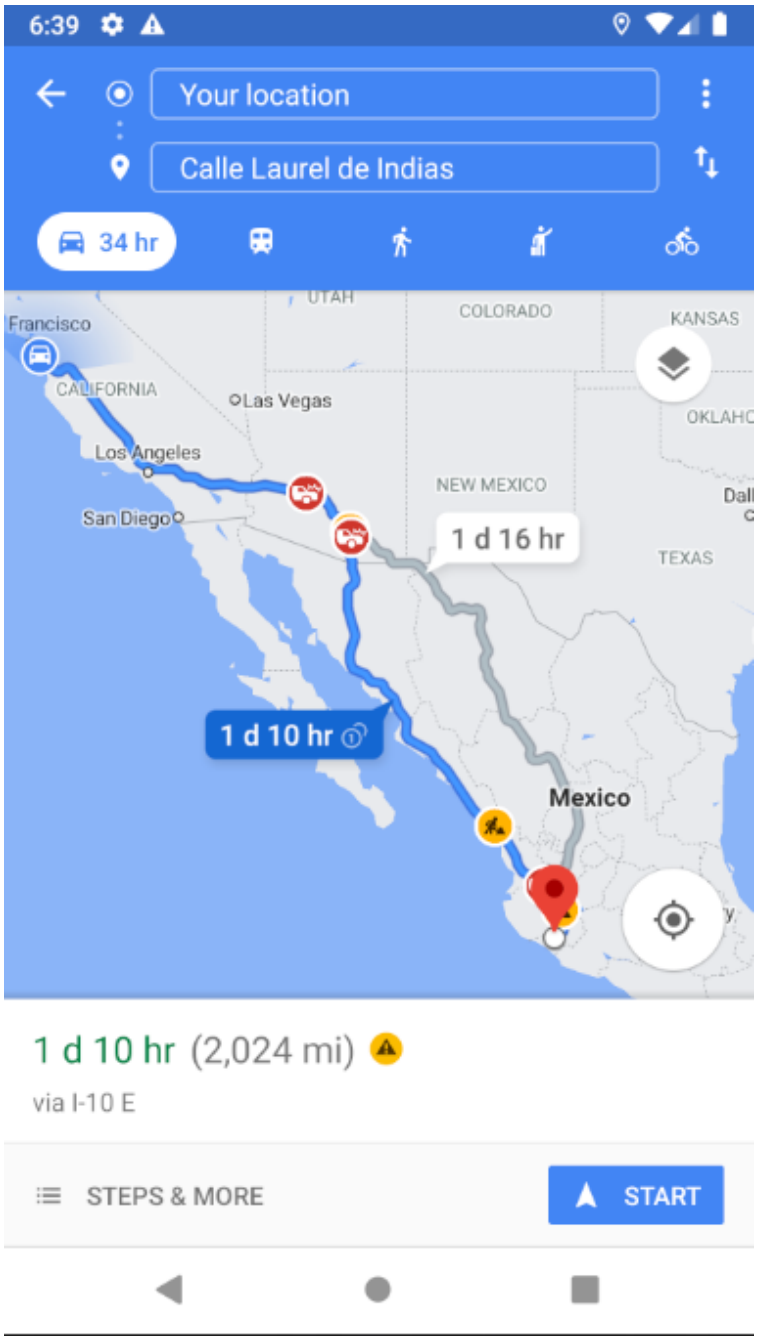


Ilustración 51 Interfaz del mapa.

Por último, y no menos importante, tenemos la interfaz de captura de información donde procedemos a realizar el llenado de la cantidad de litros y el tipo de aceite que se recolectó, el total es calculado con base al precio por litro que la empresa tiene asignado en el sistema y el pago debe ser acorde al total calculado (precio * litros). Al presionar el botón guardar se registra una recolección y un pago de ACU para volver a la pantalla de selección de paradas.

Ilustración 52. Interfaz de captura de recolección.

Con esto terminamos con el proceso llevado a cabo en la aplicación móvil que posteriormente es procesado mediante la aplicación web para el llenado de reportes.

4.12. Pruebas

Durante el desarrollo del proyecto se fueron integrando diferentes pruebas para verificar el correcto funcionamiento de ambas partes del sistema, las pruebas realizadas fueron:

- Pruebas de integridad en la base de datos.
- Pruebas en las interfaces móviles.
- Pruebas en las interfaces web.

Las pruebas de integridad se realizaron validando que la información introducida en las bases de datos fuera correcta, esta función recae sobre el manejador de bases de datos MySQL ya que las restricciones de las llaves primarias y foráneas fueron incluidas en el diseño relacional. En la ilustración 53 podemos ver el error devuelve por el manejador de la base de datos cuando se intenta hacer una inserción con llaves inexistentes.

Puesto que el diseño respeta las formas normales básicas, podemos afirmar que no tenemos errores de redundancia de datos innecesaria, incapacidad para registrar cierta información ni anomalías en el borrado o la modificación de datos.

Ilustración 53 Prueba de integridad de contactos.

Para las pruebas de la interfaz móvil la atención principal fue el filtrado de las rutas; este proceso, aunque secundario, es donde mayormente puede afectar el funcionamiento del sistema ya que al permitir que un recolector capture información de una ruta cancelada, terminada o de otro recolector la consistencia de la información puede verse afectada.

En los datos mostrados en la ilustración 54 podemos observar 4 rutas generadas de las cuales 2 están activas, una cancelada y una terminada. El recolector con identificador numérico 1 tiene una activa y una terminada, por lo tanto, al ingresar a la aplicación móvil solo se podrá tener acceso a la ruta activa como se muestra en la ilustración 55.

id_ruta	id_recolector	fecha	efectivo_inicio	efectivo_termino	kms_inicio	kms_fin	kmsrecorridos	kmsesperados	estado
1	1	17/12/2020	1734.5	0	1500	0	0	23176	1
2	2	21/01/2021	540	0	1800	0	0	15665	0
3	2	21/01/2021	540	0	1800	0	0	15665	1
4	1	22/01/2021	600	180	2000	2010	10	7281	2

Ilustración 54. Tabla de rutas.

Ilustración 55 Rutas activas.

Estas pruebas fueron realizadas con todas las variantes posibles para asegurar el resultado de salida correcto. Los filtros incorporados en las consultas nos dan la certeza de que la información no se volverá inconsistente.

Las pruebas en el sistema web se enfocaron en validar los datos de captura en los listados, validar la disponibilidad de llaves foráneas y la duplicidad de campos secundarios como los nombres repetidos que, aunque no son llaves secundarias se hizo para evitar la confusión entre los usuarios.

Las validaciones en los listados se realizaron en los usuarios, recolectores, tipos de aceite, empresas, puntos de acopio y contactos. En el caso de los usuarios, cuando intentamos agregar un usuario repetido o modificar uno existente para que tenga las mismas características de otro nos manda un mensaje avisando que el usuario ya existe (ilustración 56). Una validación adicional fue evitar que el usuario “admin” pudiera ser eliminado, ya que es el usuario default del sistema para iniciar su funcionamiento posterior a la instalación inicial.

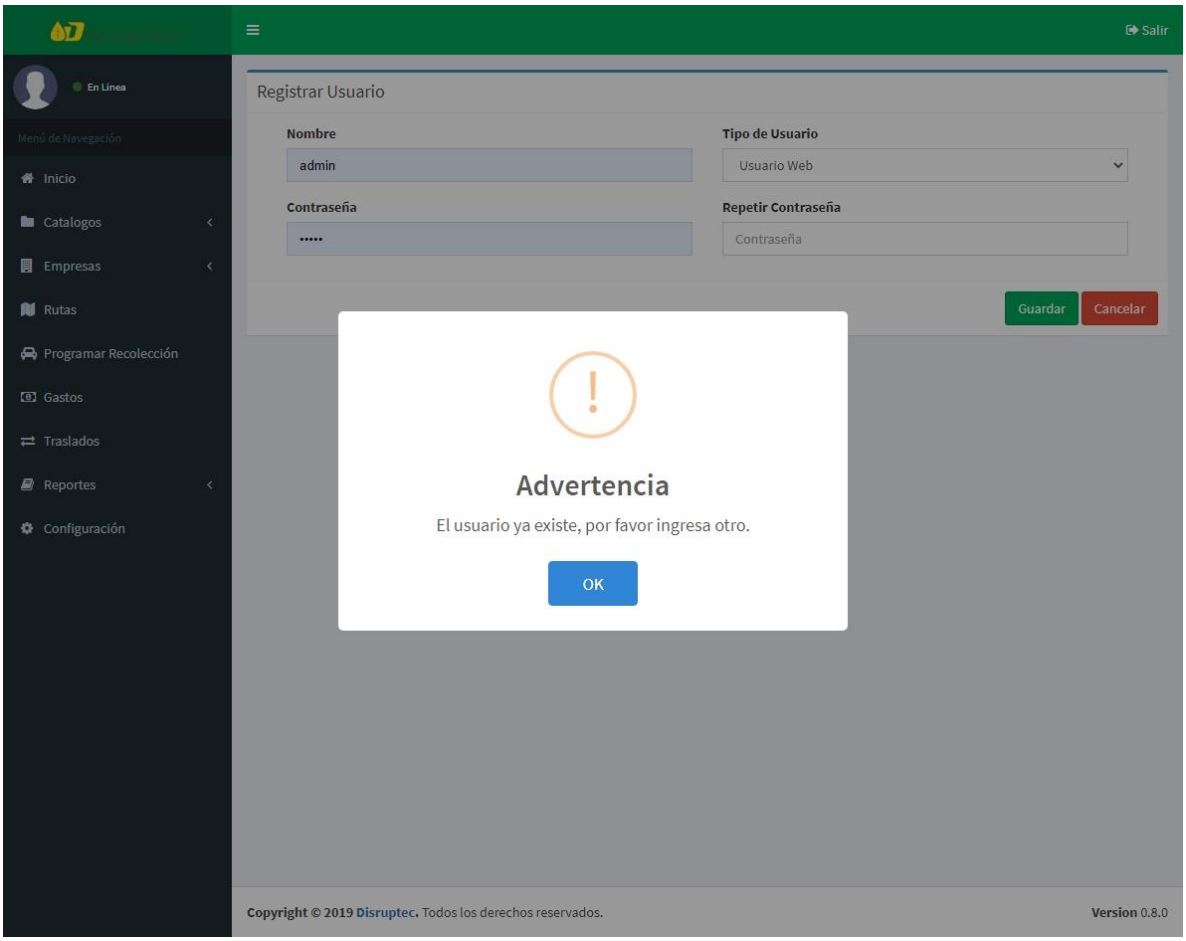


Ilustración 56. Validación de usuario repetido.

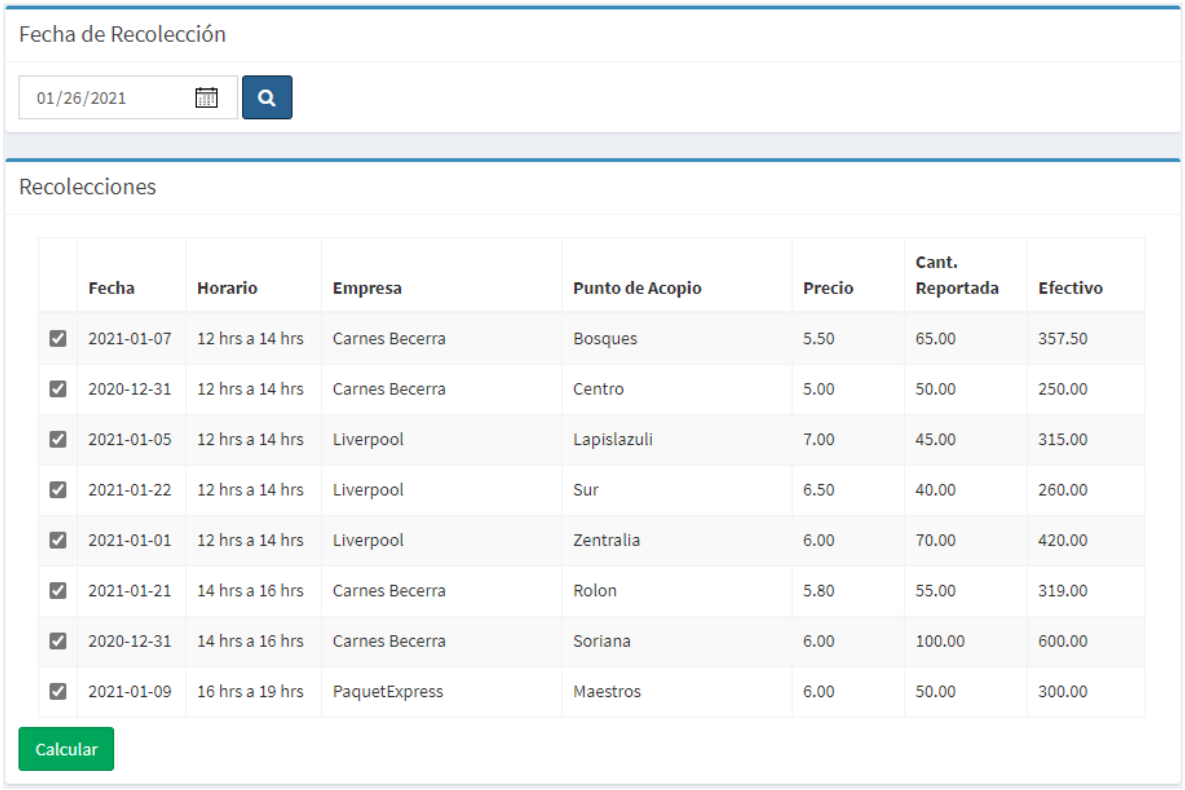
Durante el desarrollo del sistema web se realizaron las siguientes pruebas mostradas en la ilustración 57. Estas pruebas ayudaron a que todas las interfaces cumplieran con un ambiente libre de errores o posibles entradas incorrectas de parte del usuario web.

PRUEBAS SISTEMA WEB			
Prueba	**Interfaz**	**Descripción**	**Resultados**
Validar usuario repetido.	Usuarios	Validar que no existan dos usuarios con el mismo nombre.	Mensaje de error al registrar usuarios con el mismo nombre.
Evitar eliminación de usuario administrador.	Usuarios	Prevenir la eliminación del usuario admin.	Mensaje de error al intentar eliminar el usuario admin.
Validar recolector repetido.	Recolectores	Validar que no existan dos recolectores con el mismo nombre.	Mensaje de error al registrar un recolector repetido.
Usuarios disponibles para el recolector.	Recolectores	Validar que existan usuarios no asignados y del tipo móvil para el recolector.	Mensaje de advertencia al no existir usuarios disponibles.
Validar tipo de aceite repetido.	Tipos de Aceite	Validar que no existan dos tipos de aceite con el mismo nombre.	Mensaje de error al registrar un tipo de aceite repetido.
Validar empresa repetida.	Empresas	Validar que no existan dos empresas con el mismo nombre.	Mensaje de error al registrar una empresa repetida.
Validar punto de acopio repetido.	Puntos de Acopio	Validar que no existan dos puntos de acopio con el mismo nombre.	Mensaje de error al registrar un punto de acopio repetido.
Validar contacto repetido.	Contactos	Validar que no existan dos tipos de aceite con el mismo nombre.	Mensaje de error al registrar un tipo de aceite repetido.
Usuarios disponibles para el contacto.	Contactos	Validar que existan usuarios no asignados y del tipo web para el contacto.	Mensaje de advertencia al no existir usuarios disponibles.
Empresas disponibles para el contacto.	Contactos	Validar que existan empresas registradas y activas.	Mensaje de advertencia al no existir empresas activas.
Validar recolecciones programadas.	Rutas	Validar que existan recolecciones programadas al calcular una ruta.	Mensaje de advertencia al no existir recolecciones programadas para la fecha especificada.
Validar recoleccion programada.	Programar Recolección	Validar que no exista una recolección programada vigente para un punto de acopio.	Mensaje de que ya existe una recoleccion programada para el punto de acopio.

Ilustración 57. Pruebas al sistema web.

5. Resultados obtenidos

Como resultado del presente proyecto obtuvimos dos sistemas, uno en ambiente móvil y otro en web, como parte principal del proyecto fue mejorar los tiempos de recolección y para eso se utilizó el algoritmo de Dijkstra. En el ejemplo de la imagen 58 podemos ver 8 puntos de acopio los cuales fueron programados para una ruta.



	Fecha	Horario	Empresa	Punto de Acopio	Precio	Cant. Reportada	Efectivo
☑	2021-01-07	12 hrs a 14 hrs	Carnes Becerra	Bosques	5.50	65.00	357.50
☑	2020-12-31	12 hrs a 14 hrs	Carnes Becerra	Centro	5.00	50.00	250.00
☑	2021-01-05	12 hrs a 14 hrs	Liverpool	Lapislazuli	7.00	45.00	315.00
☑	2021-01-22	12 hrs a 14 hrs	Liverpool	Sur	6.50	40.00	260.00
☑	2021-01-01	12 hrs a 14 hrs	Liverpool	Zentralia	6.00	70.00	420.00
☑	2021-01-21	14 hrs a 16 hrs	Carnes Becerra	Rolon	5.80	55.00	319.00
☑	2020-12-31	14 hrs a 16 hrs	Carnes Becerra	Soriana	6.00	100.00	600.00
☑	2021-01-09	16 hrs a 19 hrs	PaquetExpress	Maestros	6.00	50.00	300.00

Ilustración 58. Puntos de acopio programados.

Durante la generación de la ruta nuestro navegador en la sección de la consola nos muestra información importante paso a paso; en la ilustración 59 podemos ver dos conjuntos de datos, el primero son las recolecciones solicitadas y el segundo son las recolecciones que se marcaron para ser atendidas. Una vez obtenidos los puntos marcados procedemos a calcular la ruta la cual es dibujada como lo muestra la ilustración 60 que nos detalla el recorrido a seguir por el recolector para lograr el recorrido óptimo el cual nos ayudará a disminuir el consumo de gasolina, mejorar los tiempos y visitar todos los lugares necesarios para cubrir todos los puntos de acopio incluidos en la recolección.

```
▼(8) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}]
  ▶0: {fecha: "2021-01-07", id_horario: "1", horarioc: "12 hrs a 14 hrs", id_punto: "4", punto: "Bos…
  ▶1: {fecha: "2020-12-31", id_horario: "1", horarioc: "12 hrs a 14 hrs", id_punto: "1", punto: "Cen…
  ▶2: {fecha: "2021-01-05", id_horario: "1", horarioc: "12 hrs a 14 hrs", id_punto: "7", punto: "Lap…
  ▶3: {fecha: "2021-01-22", id_horario: "1", horarioc: "12 hrs a 14 hrs", id_punto: "5", punto: "Sur…
  ▶4: {fecha: "2021-01-01", id_horario: "1", horarioc: "12 hrs a 14 hrs", id_punto: "2", punto: "Zen…
  ▶5: {fecha: "2021-01-21", id_horario: "2", horarioc: "14 hrs a 16 hrs", id_punto: "6", punto: "Rol…
  ▶6: {fecha: "2020-12-31", id_horario: "2", horarioc: "14 hrs a 16 hrs", id_punto: "3", punto: "Sor…
  ▶7: {fecha: "2021-01-09", id_horario: "3", horarioc: "16 hrs a 19 hrs", id_punto: "8", punto: "Mae…
    length: 8
  ▶__proto__: Array(0)
                                                                          generacion:467
▼(8) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}]
  ▶0: {id_punto: "4", id_horario: "1", fecha: "2021-01-07", horario: "12 hrs a 14 hrs", empresa: "Ca…
  ▶1: {id_punto: "1", id_horario: "1", fecha: "2020-12-31", horario: "12 hrs a 14 hrs", empresa: "Ca…
  ▶2: {id_punto: "7", id_horario: "1", fecha: "2021-01-05", horario: "12 hrs a 14 hrs", empresa: "Li…
  ▶3: {id_punto: "5", id_horario: "1", fecha: "2021-01-22", horario: "12 hrs a 14 hrs", empresa: "Li…
  ▶4: {id_punto: "2", id_horario: "1", fecha: "2021-01-01", horario: "12 hrs a 14 hrs", empresa: "Li…
  ▶5: {id_punto: "6", id_horario: "2", fecha: "2021-01-21", horario: "14 hrs a 16 hrs", empresa: "Ca…
  ▶6: {id_punto: "3", id_horario: "2", fecha: "2020-12-31", horario: "14 hrs a 16 hrs", empresa: "Ca…
  ▶7: {id_punto: "8", id_horario: "3", fecha: "2021-01-09", horario: "16 hrs a 19 hrs", empresa: "Pa…
    length: 8
  ▶__proto__: Array(0)
```


Ilustración 59. Recolecciones programadas y recolecciones utilizadas.

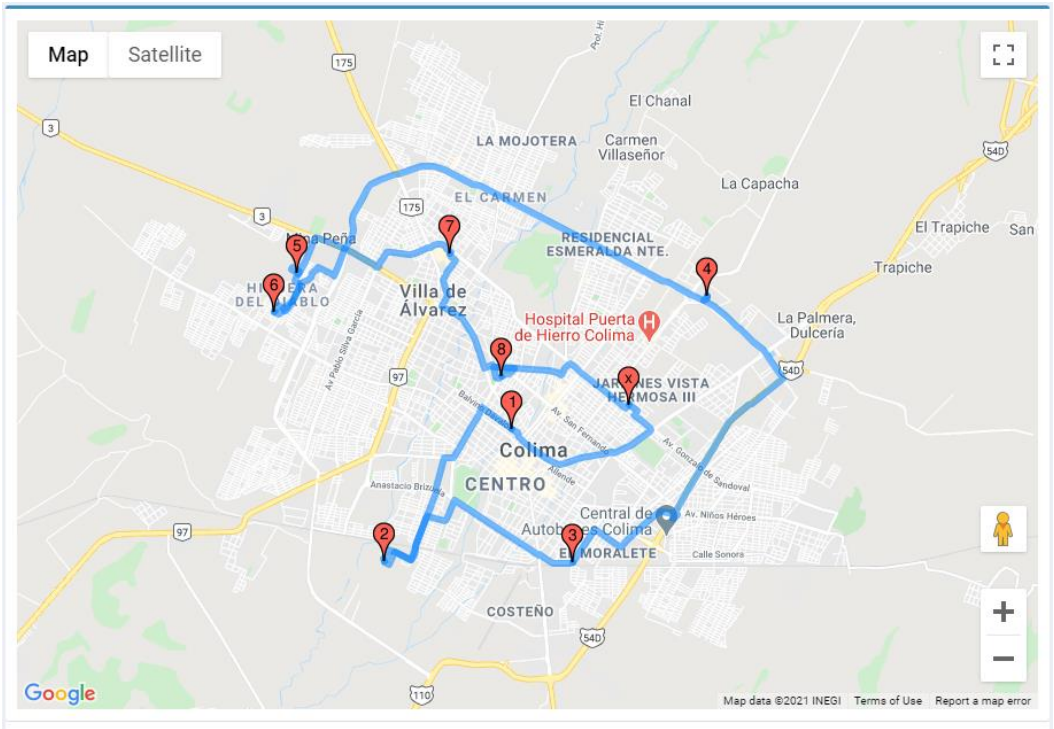


Ilustración 60. Ruta resultada de un cálculo.

Ya calculada la ruta nos arroja más información producto del procesamiento de las coordenadas con el uso del Google Maps y el algoritmo de Dijkstra. Como punto 1 nos muestra todos los caminos posibles para cubrir un recorrido de un conjunto de paradas con el mismo horario, de estos caminos selecciona el que tiene la menor distancia (punto 2), punto 3 ilustra la ruta que se utiliza para el dibujado de google maps y el punto 4 nos da la misma ruta, pero ahora con todos los campos necesarios para poder ser impresa en formato PDF.

Ilustración 61. Resultados del cálculo de una ruta.

Con este resultado podemos cubrir el principio básico del cálculo de la ruta óptima la cual sirve como eje principal en el proceso de la programación, recolección y control de la información proporcionada al recolector.

6. Conclusiones y recomendaciones

Cuando se inició el proyecto en la empresa recolectora todo el proceso se llevaba de manera manual, no tenían manera de saber todas las empresas generadoras de ACU a las cuales atendían ya que incluso no existía un documento de Excel para llevar el control. Las recolecciones eran agendadas mediante llamadas cada cierto periodo con el fin de tener una respuesta positiva. El recolector tenía un formato impreso el cual llenaba cuando recolectaba y los precios de compra eran cambiados durante recolecciones por errores o malas interpretaciones, las empresas no tenían manera de solicitar una recolección hasta recibir la llamada.

Con el proyecto terminado ahora la empresa recolectora lleva un control de todas las empresas generadoras, puede acceder a la base de datos para contactarlas o bien esperar las solicitudes mediante el portal de los contactos, el recolector puede capturar una recolección en segundos mediante la aplicación móvil sin depender de papeles que pueden extraviarse y los precios se cargan automáticamente para llevar un control de pagos más eficiente. La empresa recolectora ahora puede generar información a partir de los datos introducidos por el recolector con las herramientas de reporte del sistema y tomar decisiones oportunas.

Se cumple el objetivo principal que era el de desarrollar e implantar un sistema que lleva el control y administración de la logística asociada a la recolección de ACU. Algunos objetivos secundarios no pudieron ser evaluados por motivos de tiempo como lo son comparativas del consumo de combustible de antes y después del sistema y el incremento de recolección de ACU que solamente con el tiempo se podrá determinar.

Como mejoras futuras se recomienda lo siguiente:

- Implementar modificación de rutas sin cancelar y calcular de nuevo.
- Añadir un segundo servicio de distancias además de Google Maps para prevenir errores del servicio principal.
- Poder cancelar una parada durante el transcurso de la ruta.
- Permitir la comunicación entre Empresa Recolectora y Empresa Proveedora de ACU mediante la misma plataforma con un chat.
- Mejora en el algoritmo para permitir una cantidad ilimitada de horarios de recolección.
- Mejora en la interfaz gráfica web.

Finalmente, en respuesta a la hipótesis planteada en este proyecto se obtuvo un resultado incompleto puesto que no se pudo comprobar la reducción de costos y tiempos involucrados en la recolección por falta de tiempo para realizar comparaciones significativas.

7. Referencias bibliográficas

Alcalde, A. (2017). Algoritmos de caminos cortos. Retrieved October 14, 2018, from https://elbauldelprogramador.com/algoritmos-de-caminos-cortos/

Alegsa, L. (2017). Definición de aplicación móvil (app). Retrieved November 25, 2019, from http://www.alegsa.com.ar/Dic/aplicacion_movil.php

Alegsa, L. (2018). Definición de aplicación web. Retrieved November 25, 2019, from http://www.alegsa.com.ar/Dic/aplicacion_web.php

Baez, S. (2012). Sistemas Web. Retrieved November 15, 2018, from http://www.knowdo.org/knowledge/39-sistemas-web

Berov, T. D. (2016). A Vehicle Routing Planning System for Goods Distribution in Urban Areas Using Google Maps and Genetic Algorithm. *International Journal for Traffic & Transport Engineering*, *6*(2), 159–167.

Cartas Díaz, O. G. (2014). *Sistema de recolección de aceite usado para conversión de biodiesel*.

Castejón Garrido, J. S. (2004). Arquitectura y diseño de sistemas web modernos. *Revista de Ingeniería Informáctica Del CIIRM*, 1–6.

Chen, Z., Han, J., Wang, B., & Liu, W. (2010). Voronoi-Based k-Path Nearest Neighbor Query in Road Networks. *International Conference on Computer and Information Application*, 52–55. https://doi.org/10.1109/ICCIA.2010.6141535

Chiesa, F. (2004). Metodología para selección de sistemas ERP. *Reportes Tecnicos En Ingenieria de Software*, *6*(1), 17–37. https://doi.org/10.1016/j.procs.2015.01.056

CME Group, & Sistema de Información Energética. (2017). Históricos precios diarios petróleo WTI, Brent y MME. Retrieved January 13, 2018, from Sistema Integral sobre Economía Minera (SINEM) website: http://www.sgm.gob.mx/Web/SINEM/energeticos/wti_brent_mme.html

Colaboradores de Wikipedia. (n.d.-a). Algoritmo de Dijkstra. Retrieved October 5, 2018, from https://es.wikipedia.org/wiki/Algoritmo_de_Dijkstra

Colaboradores de Wikipedia. (n.d.-b). Algoritmo de Prim. Retrieved October 28, 2018, from https://es.wikipedia.org/wiki/Algoritmo_de_Prim

Colaboradores de Wikipedia. (n.d.-c). Sistema de planificación de recursos empresariales. Retrieved from https://es.wikipedia.org/wiki/Sistema_de_planificación_de_recursos_empresariales

Colaboradores de Wikipedia. (n.d.-d). Travelling salesman problem. Retrieved September 12, 2018, from https://en.wikipedia.org/wiki/Travelling_salesman_problem

Colaboradores de Wikipedia. (2019). Mockup. Retrieved November 25, 2019, from https://es.wikipedia.org/wiki/Mockup

Date, C. J. (2001). Introducción a los sistemas de bases de datos. In J. L. Vázquez (Ed.), *Biblioteca da Universidade da Coruña* (7ma.). México: Person Educación de México, S.A. de C.V.

Del Cid, A., Méndez, R., & Sandoval, F. (2011). *Investigación. Fundamentos y metodología*. Retrieved from

https://josedominguezblog.files.wordpress.com/2015/06/investigacion-fundamentos-y-metodologia.pdf

Domínguez, P. (2017). En qué consiste el modelo en cascada. Retrieved August 10, 2019, from https://openclassrooms.com/en/courses/4309151-gestiona-tu-proyecto-de-desarrollo/4538221-en-que-consiste-el-modelo-en-cascada

EcuRed contributors. (2015). Algoritmo de Dijkstra. Retrieved November 15, 2018, from https://www.ecured.cu/index.php?title=Algoritmo_de_Dijkstra&oldid=2558694

Fanjul, S. C. (2018). En realidad, ¿qué es exactamente un algoritmo? Retrieved November 25, 2019, from https://retina.elpais.com/retina/2018/03/22/tendencias/1521745909_941081.html

Fernández R. Y., & Díaz G. Y. (2012). Patrón Modelo-Vista-Controlador. *Revista Telem@tica*, *11*(1), 11. Retrieved from http://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15/10

Florido Benítez, L. (2016). La implementación del mobile marketing como herramienta multidisciplinaria en el sector turístico y aeroportuario. Retrieved November 12, 2018, from http://www.eumed.net/libros-gratis/2016/1539/aplicacion.htm

GasolinaMX.com. (2018). Precio Gasolina en Colima. Retrieved August 18, 2018, from http://www.gasolinamx.com/estado/colima

González Canal, I., & González Ubierna, J. A. (2015). Aceites usados de cocina. Problemática ambiental, incidencias en redes de saneamiento y coste del tratamiento en depuradoras. In *Aguasresiduales.Info*. Retrieved from http://www.aguasresiduales.info/revista/articulos/problematica-ambiental-incidencias-en-redes-de-saneamiento-y-coste-del-tratamiento-en-depuradoras-de-los-aceites-usados-en-cocina

Hamidian Fernández, B. F., & Ospino Sumoza, G. R. (2015). ¿Por qué los sistemas de información son esenciales? *Anuario*, *38*(2011), 161–183. Retrieved from http://servicio.bc.uc.edu.ve/derecho/revista/idc38/art07.pdf

Hernandez Sampieri, R., Fernandez Collado, C., & Baptista Lucio, M. del P. (2010). Metodologia de la investigacion. In *Metodología de la investigación*. Retrieved from http://www.casadellibro.com/libro-metodologia-de-la-investigacion-5-ed-incluye-cd-rom/9786071502919/1960006

Hernández, U. (2015). MVC (Model, View, Controller) Explicado. Retrieved November 10, 2018, from https://codigofacilito.com/articulos/mvc-model-view-controller-explicado

Kedia, R. K., & Naick, B. K. (2017). Review of vehicle route optimisation. *2017 2nd IEEE International Conference on Intelligent Transportation Engineering, ICITE 2017*, 57–61. https://doi.org/10.1109/ICITE.2017.8056881

Kendall, K., & Kendall, J. (2011). *Análisis y Diseños de Sistemas*. Retrieved from http://cotana.informatica.edu.bo/downloads/ld-Analisis y Diseno de Sistemas_Kendall-8va.pdf

Kheirkhahzadeh, M., & Barforoush, A. A. (2009). A hybrid algorithm for the Vehicle Routing Problem. *IEEE Congress on Evolutionary Computation*, *46*(8).

Letelier, P., & Penadés, M. C. (2006). Métodologías ágiles para el desarrollo de

software: eXtreme Programming (XP). *Técnica Administrativa, Buenos Aires*, *05*(26). Retrieved from http://www.cyta.com.ar/ta0502/v5n2a1.htm

Luna, O. A. (2016, September 20). Colima, 5° lugar con más autos per cápita. *Perriodismo*. Retrieved from https://www.perriodismo.com.mx/2016/09/20/colima-5-lugar-con-mas-autos-per-capita/

Martínez, Felipe;Campoy, A. (2011). *Aplicaciones para dispositivos móviles*. 1.

Mustafa, S. Al, & Žari, N. (2018). *MasterRoute : Android Application for Finding the Optimal Traffic Route*.

Navarro, J. (2010). Definición de Marco Teórico. Retrieved from https://www.definicionabc.com/ciencia/marco-teorico.php

Osorio Rivera, F. L. (2008). *Bases de datos relacionales: Teoría y práctica* (1a.; F. E. ITM, Ed.). Medellin, Colombia: Textos Académicos.

Pei, Y., Wang, W., & Zhang, S. (2012). Basic ant colony optimization. *Proceedings - 2012 International Conference on Computer Science and Electronics Engineering, ICCSEE 2012*, *1*, 665–667. https://doi.org/10.1109/ICCSEE.2012.178

Pressman, R. (2002). *Ingeniería del Software. Un enfoque práctico.*

Quiroga, J. P. (n.d.). Requerimientos Funcionales y No Funcionales. Retrieved November 25, 2019, from http://www.electrohuila.com.co/Portals/0/UpDocuments/0b530417-2986-450e-bd92-34928a11e2f5.pdf

Raffino, M. E. (2018). Diagrama de Flujo. Retrieved November 25, 2019, from https://concepto.de/diagrama-de-flujo/

Senn, J. A. (1996). *Análisis y Diseño de Sistemas de Información*. *2*.

Sommerville, I. (2005). *Ingeniaria del Software*.

TuGimnasiaCerebral. (2019). ¿Qué es Gráfica de Gantt? Cómo Crearla y Ejemplos. Retrieved November 25, 2019, from http://tugimnasiacerebral.com/herramientas-de-estudio/que-es-un-diagrama-o-grafica-de-gantt

United Nations. (n.d.). Climate Change. Retrieved June 9, 2018, from http://www.un.org/en/sections/issues-depth/climate-change/

Wang, J., Rui, X., Song, X., Tan, X., Wang, C., & Raghavan, V. (2015). A novel approach for generating routable road maps from vehicle GPS traces. *International Journal of Geographical Information Science*, *29*(1), 69–91. https://doi.org/10.1080/13658816.2014.944527

Wei, M., & Meng, Y. (2014). Research on the optimal route choice based on improved Dijkstra. *Proceedings - 2014 IEEE Workshop on Advanced Research and Technology in Industry Applications, WARTIA 2014*, 303–306. https://doi.org/10.1109/WARTIA.2014.6976257

Wells, D. (2013). Extreme Programming: A gentle introduction. Retrieved May 17, 2018, from http://www.extremeprogramming.org