PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN SISTEMAS Y COMPUTACIÓN

"DESARROLLO DE UN SISTEMA PARA EL CONTROL DE VENDEDORES UTILIZANDO GEOLOCALIZACIÓN CON APLICACIONES MÓVILES. CASO DE ESTUDIO: TOPESA"

MARÍA GLORIA MARCILLO DÍAZ

DIRECTOR:

Ing. Damián Nicolalde

QUITO, ENERO 2019

RESUMEN

El presente trabajo de disertación busca desarrollar un sistema que permita dar seguimiento al trabajo de los vendedores de la empresa TOPESA S.A., con el fin de garantizar la eficacia de este. El sistema permite verificar que los vendedores se encuentran en sus rutas, buscando nuevos clientes interesados en los productos de la empresa, y realizando las respectivas visitas a los mismos.

Para la realización de este trabajo se utilizó una metodología ágil denominada como "Metodología XP", la misma que permitió una retroalimentación continua con el cliente, así como buscar soluciones simples para cualquier requerimiento que se pudiera presentar.

Durante la elaboración del presente trabajo se utilizaron herramientas tecnológicas actuales que permiten el desarrollo de aplicaciones móviles y web, además de otras como la geolocalización y bases de datos en tiempo real.

Con el fin de verificar que el sistema realizar todas sus funciones de forma correcta, se realizaron pruebas constantemente. Estas eran normalmente, el ingreso y modificación de datos, considerando todo el tipo de información que el cliente podrían llegar a ingresar.

Este proyecto permitió conocer más sobre los sistemas de geolocalización y sus amplios beneficios y funciones. Además de, nuevas metodologías de desarrollo de software que pueden servir para simplificar y mejorar el trabajo, así como mantener una buena comunicación con el cliente.

DEDICATORIA

Esta disertación va dedicada a todas las personas que estuvieron a mi lado apoyándome durante todos estos años.

A mis padres, por toda su paciencia y apoyo continúo durante toda mi vida. Por brindarme una vida maravillosa y trabajar arduamente todo este tiempo para poder otorgarme una excelente educación. Por estar siempre ahí respondiendo mis preguntas y ayudando a siempre saciar mi curiosidad por el mundo.

A mis hermanas, por cuidarme y enseñarme sobre la vida. Por estar junto a mí en mis travesuras, y ser mis compañeras de vida.

A mis amigos, a los que tengo cerca y a los que están más lejos. A cada uno de ellos, por levantarme cuando lo necesitaba y siempre estar a mi lado.

AGRADECIMIENTOS

A mi familia, a mis padres por ser el pilar fundamental de mi vida y haber luchado durante toda mi vida por darme lo mejor y enseñarme muchas cosas durante el camino, por apoyarme en todo lo que he querido hacer; y a mis hermanas, por estar ahí cuidándome y acompañándome en todos los momentos importantes y en los que no también.

A TOPESA por permitir realizar este proyecto, y ayudarme con todas las herramientas necesarias para hacerlo.

TABLA DE CONTENIDO

RESUMEN 3

DEDICATORIA 4

AGRADECIMIENTOS 5

ÍNDICE DE FIGURAS 10

ÍNDICE DE TABLAS 12

CAPÍTULO 1. INTRODUCCIÓN 15

1.1. Justificación 15

1.2. Planteación del Problema 15

1.3. Objetivo 15

1.3.1. Objetivo General 15

1.3.2. Objetivos Específicos 15

CAPÍTULO 2. FUNDAMENTOS TEÓRICOS 17

2.1. La Empresa 17

2.1.1. Visión 17

2.1.2. Misión 17

2.1.3. Principios y Valores 17

2.1.4. Política de Calidad 18

2.1.5. Organigrama 19

2.1.6. Mapa de Procesos 20

2.2. Ingeniería de Software 21

2.2.1. Metodologías Ágiles 21

2.2.2. Metodología XP 22

2.3. Geolocalización 27

2.4. Aplicaciones Móviles 28

2.4.1. Aplicación Móvil 28
2.4.2. Dispositivos Móviles 28
2.4.3. Sistemas Operativos Móviles 28
2.5. Bases de Datos 29
2.5.1. Sistema Gestor de Base de Datos 30
2.6. Lenguajes de Programación 31
2.7. Aplicaciones Web 34
2.7.1. Servicios Web 34
2.8. Internet 34
2.9. Servidores de Aplicaciones 35
CAPÍTULO 3. ITERACIONES 36
3.1. ITERACIÓN 1 - EMPLEADOS 36
3.1.1. Planeación 36
3.1.2. Diseño 39
3.1.3. Codificación 42
3.1.4. Pruebas 44
3.2. ITERACIÓN 2 - VEHÍCULOS 46
3.2.1. Planeación 46
3.2.2. Diseño 49
3.2.3. Codificación 51
3.2.4. Pruebas 53
3.3. ITERACIÓN 3 - CLIENTES 55
3.3.1. Planeación 55
3.3.2. Diseño 58
3.3.3. Codificación 60

3.3.4. Pruebas 62
3.4. ITERACIÓN 4 – VISITAS 64
3.4.1. Planeación 64
3.4.2. Diseño 67
3.4.3. Codificación 69
3.4.4. Pruebas 71
3.5. ITERACIÓN 5 – LOCALIZACIONES 73
3.5.1. Planeación 73
3.5.2. Diseño 76
3.5.3. Codificación 77
3.5.4. Pruebas 79
CAPÍTULO 4. ITERACIONES ANDROID 81
4.1. ITERACIÓN 1 - EMPLEADOS 81
4.1.1. Planeación 81
4.1.2. Diseño 84
4.1.3. Codificación 85
4.1.4. Pruebas 87
4.2. ITERACIÓN 2 – VEHÍCULOS 89
4.2.1. Planeación 89
4.2.2. Diseño 92
4.2.3. Codificación 93
4.2.4. Pruebas 95
4.3. ITERACIÓN 3 – CLIENTES 97
4.3.1. Planeación 97
4.3.2. Diseño 100

4.3.3. Codificación 102
4.3.4. Pruebas 104
4.4. ITERACIÓN 4 – VISITAS 106
4.4.1. Planeación 106
4.4.2. Diseño 109
4.4.3. Codificación 112
4.4.4. Pruebas 114
4.5. ITERACIÓN 5 – LOCALIZACIONES 116
4.5.1. Planeación 116
4.5.2. Diseño 119
4.5.3. Codificación 125
4.5.4. Pruebas 128
4.6. MODELO RELACIONAL 130
CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES 131
5.1. Conclusiones 131
5.2. Recomendaciones 132
BIBLIOGRAFÍA 134

ÍNDICE DE FIGURAS

Figura 0-1. Organigrama (Marcillo D, 2018) 19

Figura 0-2. Mapa de Procesos (Marcillo D, 2018) 20

Figura 0-3. Capas de la Ingeniería de Software (Pressman, 2010) 21

Figura 1-1. Diseño Simple – Empleados (Marcillo D, 2018) 39

Figura 1-2. Prototipo 1 – WEB – Empleados (Marcillo D, 2018) 41

Figura 1-3. Prototipo 2 – WEB – Empleados (Marcillo D, 2018) 41

Figura 2-1. Diseño Simple – WEB – Vehículos (Marcillo D, 2018) 49

Figura 2-2. Prototipo 1 – WEB – Vehículos (Marcillo D, 2018) 50

Figura 2-3. Prototipo 2 – WEB – Vehículos (Marcillo D, 2018) 51

Figura 3-1. Diseño Simple – WEB – Clientes (Marcillo D, 2018) 58

Figura 3-2. Prototipo 1 – WEB – Clientes (Marcillo D, 2018) 59

Figura 3-3. Prototipo 2 – WEB – Clientes (Marcillo D, 2018) 60

Figura 4-1. Diseño Simple – WEB – Visitas (Marcillo D, 2018) 67

Figura 4-2. Prototipo 1 – WEB – Visitas (Marcillo D, 2018) 68

Figura 4-3. Prototipo 2 – WEB – Visitas (Marcillo D, 2018) 68

Figura 5-1. Diseño Simple – WEB – Localizaciones (Marcillo D, 2018) 76

Figura 5-2. Prototipo – WEB – Localizaciones (Marcillo D, 2018) 77

Figura 6-1. Diseño Simple – Android – Empleados (Marcillo D, 2018) 84

Figura 6-2. Prototipo – Android – Empleados (Marcillo D, 2018) 85

Figura 7-1. Diseño Simple – Android – Vehículos (Marcillo D, 2018) 92

Figura 7-2. Prototipo – Android – Vehículos (Marcillo D, 2018) 93

Figura 8-1. Diseño Simple – Android – Clientes (Marcillo D, 2018) 100

Figura 8-2. Prototipo 1 – Android – Clientes (Marcillo D, 2018) 101

Figura 8-3. Prototipo 2 – Android – Clientes (Marcillo D, 2018) 101

Figura 8-4. Prototipo 3 – Android – Clientes (Marcillo D, 2018) 102

Figura 9-1. Diseño Simple 1 – Android – Visitas (Marcillo D, 2018) 109

Figura 9-2. Diseño Simple 2 – Android – Visitas (Marcillo D, 2018) 109

Figura 9-3. Prototipo 1 – Android – Empleado (Marcillo D, 2018)..... 110

Figura 9-4. Prototipo 2 – Android – Empleado (Marcillo D, 2018)..... 111

Figura 9-5. Prototipo 3 – Android – Empleado (Marcillo D, 2018)..... 111

Figura 10-1. Diseño Simple – Android – Localizaciones (Marcillo D, 2018) 119

Figura 10-2. Prototipo 1 – Android – Localizaciones (Marcillo D, 2018)..... 120

Figura 10-3. Prototipo 2 – Android – Localizaciones (Marcillo D, 2018)..... 121

Figura 10-4. Prototipo 3 – Android – Localizaciones (Marcillo D, 2018)..... 121

Figura 10-5. Prototipo 4 – Android – Localizaciones (Marcillo D, 2018)..... 122

Figura 10-6. Prototipo 5 – Android – Localizaciones (Marcillo D, 2018)..... 123

Figura 10-7. Prototipo 6 – Android – Localizaciones (Marcillo D, 2018)..... 123

Figura 10-8. Prototipo 7 – Android – Localizaciones (Marcillo D, 2018)..... 124

Figura 11-1. Modelo Relacional (Marcillo D, 2019)..... 130

ÍNDICE DE TABLAS

Tabla 0-1. Resumen Lenguajes de Programación (Marcillo D, 2019)..... 33

Tabla 1-1. Historia de Usuario 1 – Administrar Empleados (Marcillo D, 2018).. 37

Tabla 1-2. Plan de Iteraciones – Empleados (Marcillo D, 2018)..... 38

Tabla 1-3. Criterio de Prueba de Aceptación 1 (Marcillo D, 2018)..... 38

Tabla 1-4. Criterio de Prueba de Aceptación 2 (Marcillo D, 2018)..... 39

Tabla 1-5. Tarjeta CRC – Empleados (Marcillo D, 2018)..... 40

Tabla 1-6. Prueba de Aceptación (Marcillo D, 2018)..... 44

Tabla 1-7. Prueba de Caja Negra (Marcillo D, 2018) 45

Tabla 2-1. Historia de Usuario 2 – Administrar Vehículos (Marcillo D, 2018).... 47

Tabla 2-2. Plan de Iteración (Marcillo D, 2018) 48

Tabla 2-3. Criterio de Prueba de Aceptación 1 (Marcillo D, 2018)..... 48

Tabla 2-4. Criterio de Prueba de Aceptación 2 (Marcillo D, 2018)..... 49

Tabla 2-5. Tarjeta CRC – Vehículos (Marcillo D, 2018) 50

Tabla 2-6. Prueba de Aceptación (Marcillo D, 2018)..... 53

Tabla 2-7. Prueba de Caja Negra (Marcillo D, 2018) 54

Tabla 3-1. Historia de Usuario 3 – Administrar Clientes (Marcillo D, 2018)..... 56

Tabla 3-2. Plan de Iteración (Marcillo D, 2018) 57

Tabla 3-3. Criterio de Prueba de Aceptación 1 (Marcillo D, 2018)..... 57

Tabla 3-4. Criterio de Prueba de Aceptación 2 (Marcillo D, 2018)..... 58

Tabla 3-5. Tarjeta CRC – Clientes (Marcillo D, 2018)..... 59

Tabla 3-6. Prueba de Aceptación (Marcillo D, 2018)..... 62

Tabla 3-7. Prueba de Caja Negra (Marcillo D, 2018) 63

Tabla 4-1. Historia de Usuario 4 – Administrar Visitas (Marcillo D, 2018)..... 65

Tabla 4-2. Plan de Iteración (Marcillo D, 2018) 66

Tabla 4-3. Criterio de Prueba de Aceptación (Marcillo D, 2018) 66

Tabla 4-4. Tarjeta CRC – Visitas (Marcillo D, 2018) 67

Tabla 4-5. Prueba de Aceptación (Marcillo D, 2018) 71

Tabla 4-6. Prueba de Caja Negra (Marcillo D, 2018) 72

Tabla 5-1. Historia de Usuario 5 – Administrar Localizaciones (Marcillo D, 2018) 74

Tabla 5-2. Plan de Iteración (Marcillo D, 2018) 75

Tabla 5-3. Criterio de Prueba de Aceptación (Marcillo D, 2018) 75

Tabla 5-4. Tarjeta CRC – Localización (Marcillo D, 2018) 76

Tabla 5-5. Prueba de Aceptación (Marcillo D, 2018) 79

Tabla 5-6. Prueba de Caja Negra (Marcillo D, 2018) 80

Tabla 6-1. Historia de Usuario 6 – Administrar Empleados (Marcillo D, 2018) .. 82

Tabla 6-2. Plan de Iteraciones – Empleados (Marcillo D, 2018) 83

Tabla 6-3. Criterio de Prueba de Aceptación (Marcillo D, 2018) 83

Tabla 6-4. Tarjeta CRC – Empleados (Marcillo D, 2018) 84

Tabla 6-5. Prueba de Aceptación (Marcillo D, 2018) 87

Tabla 6-6. Prueba de Caja Negra (Marcillo D, 2018) 88

Tabla 7-1. Historia de Usuario 7 – Administrar Vehículos (Marcillo D, 2018) 90

Tabla 7-2. Plan de Iteración (Marcillo D, 2018) 91

Tabla 7-3. Criterio de Prueba de Aceptación (Marcillo D, 2018) 91

Tabla 7-4. Tarjeta CRC – Vehículos (Marcillo D, 2018) 92

Tabla 7-5. Prueba de Aceptación (Marcillo D, 2018) 95

Tabla 7-6. Prueba de Caja Negra (Marcillo D, 2018) 96

Tabla 8-1. Historia de Usuario 8 – Administrar Clientes (Marcillo D, 2018) 98

Tabla 8-2. Plan de Iteración (Marcillo D, 2018) 99

Tabla 8-3. Criterio de Prueba de Aceptación (Marcillo D, 2018) 99

Tabla 8-4. Tarjeta CRC – Clientes (Marcillo D, 2018) 100

Tabla 8-5. Prueba de Aceptación (Marcillo D, 2018) 104

Tabla 8-6. Prueba de Caja Negra (Marcillo D, 2018) 105

Tabla 9-1. Historia de Usuario 9 – Administrar Visitas (Marcillo D, 2018) 107

Tabla 9-2. Plan de Iteración (Marcillo D, 2018) 108

Tabla 9-3. Criterio de Prueba de Aceptación (Marcillo D, 2018) 108

Tabla 9-4. Tarjeta CRC – Visitas (Marcillo D, 2018) 110

Tabla 9-5. Prueba de Aceptación (Marcillo D, 2018) 114

Tabla 9-6. Prueba de Caja Negra (Marcillo D, 2018) 115

Tabla 10-1. Historia de Usuario 10 – Administrar Localizaciones (Marcillo D, 2018) 117

Tabla 10-2. Plan de Iteración (Marcillo D, 2018) 118

Tabla 10-3. Criterio de Prueba de Aceptación (Marcillo D, 2018) 118

Tabla 10-4. Tarjeta CRC – Localización (Marcillo D, 2018) 119

Tabla 10-5. Prueba de Aceptación (Marcillo D, 2018) 128

Tabla 10-6. Prueba de Caja Negra (Marcillo D, 2018) 129

CAPÍTULO 1. INTRODUCCIÓN

1.1. JUSTIFICACIÓN

El presente trabajo de disertación pretende mantener un control de las actividades que realizan los vendedores de la empresa por medio de una aplicación móvil para la utilización de estos, una aplicación web para uso de los administradores, y herramientas tecnológicas actuales como la geolocalización; con el fin de garantizar la eficacia del proceso de los mismos.

1.2. PLANTEACIÓN DEL PROBLEMA

Dentro de la empresa TOPESA, no se cuenta con ningún proceso o sistema que permita mantener un control de las actividades de los vendedores, y que verifique que se encuentren realizando su trabajo adecuadamente, así como que se estén movilizando, buscando nuevos clientes y realizando sus respectivas visitas a los mismos.

1.3. OBJETIVO

1.3.1. OBJETIVO GENERAL

Desarrollar un sistema para el control de vendedores utilizando geolocalización con aplicaciones móviles. Caso de estudio: TOPESA.

1.3.2. OBJETIVOS ESPECÍFICOS

- Analizar los requerimientos en base a los parámetros establecidos en una metodología de desarrollo para satisfacer las necesidades del usuario.
- Diseñar el sistema en base a la lista de requerimientos y tomando en cuenta los parámetros que la metodología de desarrollo indique para garantizar la eficiencia de las actividades de los vendedores.
- Codificar la aplicación móvil tomando en cuenta los modelos diseñados para poder monitorear si los vendedores están o no cumpliendo con sus respectivas tareas.

- Evaluar la funcionalidad de la aplicación, a través de las diferentes pruebas de software para determinar si el modelo diseñado verdaderamente satisface las necesidades del cliente.

CAPÍTULO 2. FUNDAMENTOS TEÓRICOS

2.1. LA EMPRESA

Dentro de este capítulo se describirá a la empresa para la cual será desarrollado el sistema, con el fin de conocer mejor a la misma.

2.1.1. VISIÓN

Constituirse en una empresa líder, consolidada en el mercado de elementos de fijación, con productos diferenciados y de reconocimiento nacional por la excelencia en el servicio y atención al cliente.

2.1.2. MISIÓN

Fabricar y comercializar elementos de calidad reconocida en el mercado, caracterizándonos por la provisión oportuna de productos y la búsqueda permanente de la satisfacción de los clientes internos y externos.

2.1.3. PRINCIPIOS Y VALORES

2.1.3.1. PRINCIPIOS

- **Calidad**: Aplicación de procesos y oferta de productos que brinden total confianza al consumidor.
- **Innovación**: Buscando constantemente mejorar los productos y servicios, invirtiendo en proyectos de investigación y desarrollo.
- **Puntualidad**: Respetando los plazos establecidos en la entrega del material solicitado por el cliente mediante la optimización de los tiempos de producción.
- **Excelencia operacional**: Haciendo lo correcto, según los criterios de los clientes, empleados, accionistas y proveedores. Realizando las actividades de la empresa de tal manera que se traduzcan en mejores resultados, siendo la base de la mejora continua en áreas de seguridad, personal, servicios y costos.

- **Responsabilidad social en la seguridad, salud y medio ambiente**: actuando de manera responsable para proteger a los trabajadores de la empresa y cumpliendo con normas establecidas para la protección ambiental.

2.1.3.2. VALORES

- **Honestidad**: El personal de la empresa debe actuar siempre en base a la verdad y auténtica justicia, expresando respeto por uno mismo y por los demás.
- **Responsabilidad**: reflexionando, administrando, orientando y valorando las consecuencias del cumplimiento de las obligaciones en el plano moral.
- **Voluntad**: realizando tareas asignadas con la mejor predisposición, por encima de las dificultades que se puedan presentar.
- **Respeto**: reconociendo, apreciando y valorando las cualidades de los demás y sus derechos; además aceptando los diferentes criterios y actitudes dentro de la filosofía de la organización.
- **Optimismo**: Enfrentando contratiempos con buen ánimo y perseverancia, descubriendo lo positivo de las personas y circunstancias, confiando en nuestras capacidades y posibilidades.
- **Comunicación**: intercambiando de forma efectiva pensamientos, ideas y sentimientos con las personas que nos rodean, en un ambiente de cordialidad y buscando el enriquecimiento personal.

2.1.4. POLÍTICA DE CALIDAD

TOPESA, la primera Industria del país que fabrica y comercializa tornillos, pernos, tuercas y otros elementos de sujeción de alta calidad, cumpliendo con las normas ANSI, ASTM e ISO, satisface los requerimientos de sus clientes gracias a la mejora continua de sus procesos y a su personal calificado.

2.1.5. ORGANIGRAMA

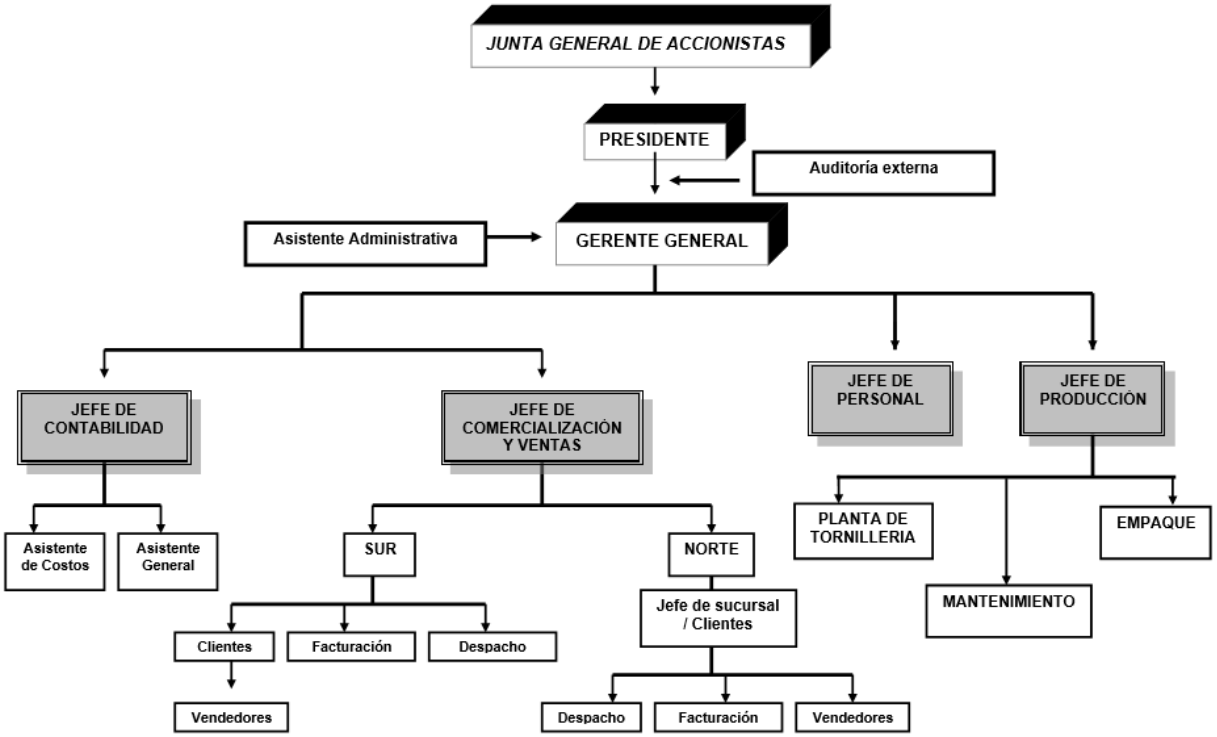


Figura 0-1. Organigrama (Marcillo D, 2018)

2.1.6. MAPA DE PROCESOS

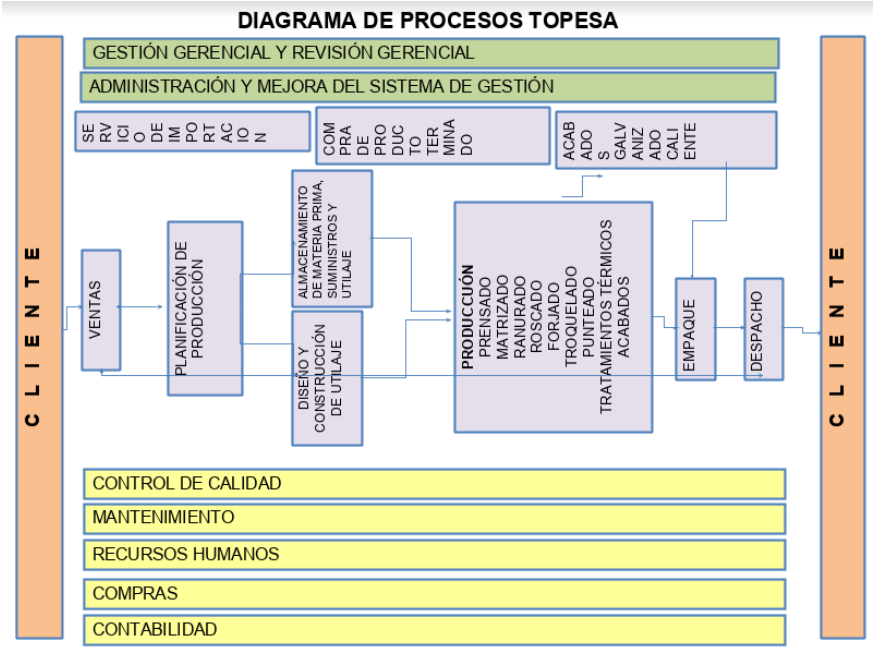


Figura 0-2. Mapa de Procesos (Marcillo D, 2018)

2.2. INGENIERÍA DE SOFTWARE

La IEEE desarrolló una completa definición en el año 1993 donde dice que la Ingeniería de Software es: "La aplicación de un enfoque sistemático, disciplinado, y cuantificable al desarrollo, operación, y mantenimiento del software; es decir la aplicación de Ingeniería del Software." (Pressman, 2010).

La ingeniería de software se encuentra conformada por varias capas.



Figura 0-3. Capas de la Ingeniería de Software (Pressman, 2010)

2.2.1. METODOLOGÍAS ÁGILES

Este tipo de metodologías son mucho más flexibles y permiten la realización de cambios en la marcha para que estos se puedan ajustar a la realidad de los equipos y proyectos, los mismos que llegan a ser subdivididos en proyectos más pequeños y distribuidos para que sean tratados de una forma mucho más independiente. Además, el cliente llega a ser parte del equipo de desarrollo y se busca mantener una comunicación extensa con el mismo, esto debido a que las metodologías ágiles se encuentran orientadas a las personas no a los procesos. También, se busca generar la menor cantidad de documentos posibles a menos que se absolutamente necesario. Inclusive, dentro de estas metodologías, se esperan entregas constantes del software para ir comprobando así su progreso y calidad. (Whitten & Bentley, 2008)

2.2.2. METODOLOGÍA XP

La metodología XP (Extreme Programming) es un tipo de metodología ágil cuyo objetivo es potenciar las distintas relaciones interpersonales para alcanzar el éxito del desarrollo de un software, por medio de la promoción del trabajo en equipo, procurando que los desarrolladores a su vez aprendan mucho más, y buscando mantener un buen clima de trabajo. Esta metodología tiene como base a la realimentación continua entre el equipo de desarrollo y el cliente, así como mantener una excelente comunicación entre todos los involucrados, además de buscar implementar las soluciones de la manera más simple, y estar dispuesto a enfrentar los cambios que se pudiesen presentar. La metodología XP es especialmente utilizada en proyectos cuyos requisitos no son claros y se encuentran cambiando a todo momento, y además existe un alto riesgo técnico. (Whitten & Bentley, 2008)

La metodología XP busca convertir a la programación en un proceso más "humanizado", es decir, que las personas formen relaciones y la comunicación sea amplia entre todos los participantes de la realización del proyecto. Por otro lado, la metodología cuenta con cuatro variables fundamentales a la hora de desarrollar un nuevo proyecto: coste, la inversión económica que se realiza, así como los recursos; el tiempo, el tiempo que se emplea para entregar el proyecto en una fecha determinada; la calidad, tanto del código como de la aplicación ya desarrollada; y el alcance, que es el conjunto de las funcionalidades. De estas variables, tres podrán ser elegidas por el cliente o el jefe de proyectos, y la cuarta será seleccionada por el resto del equipo. (Whitten & Bentley, 2008)

Asimismo, esta metodología busca medir su utilidad por medio de cuatro valores que describen a los aspectos que permitirán alcanzar los objetivos, estos valores son: comunicación, debe ser muy extensa entre todos los que se encuentran involucrados y se busca que la misma sea directa en lugar de escrita; simplicidad, es necesario que la solución sea lo más sencilla posible pues esto nos dará una ventaja al momento de realizar cambios; realimentación, aquí se le entregará al usuario el software desarrollado desde un principio para poder obtener sus

impresiones y las necesidades que no han sido satisfechas; coraje, este valor busca mejorar mucho más lo desarrollado aun si funciona. (Whitten & Bentley, 2008)

DOCE PRÁCTICAS BÁSICAS

Existen doce prácticas que se pueden relacionar entre sí para obtener mejores resultados, y estas son las siguientes:

- **Diseño simple:** en sí se busca hacer uso del diseño más simple que permita que todo funcione para poder a su vez facilitar el mantenimiento y amenorar la cantidad de riesgos a los cambios que se realicen.
- **Refactorización:** se modifica el código para dejarlo en un buen estado, esto se consigue al volver a escribir, desde un punto de vista global, las partes que se necesiten, sin tomar en cuenta los cambios que se realicen.
- **Test:** considerado como la base fundamental de las prácticas de la metodología XP, su objetivo es evitar errores, no de detectarlos, así como prevenirlos en lugar de corregirlos. Pueden ser de tres tipos: de aceptación, se crea junto con el cliente final y refleja las necesidades funcionales del mismo; unitario, el programador es quien los realiza con el objetivo de verificar que todos los métodos funcionen de manera correcta; de integridad, el equipo de desarrollo es quien los crea para poder revisar que todo el conjunto trabaja correctamente después de cambios realizados.
- **Estándares de codificación:** se busca que todo el equipo de desarrollo cuenta con normas de codificación común, es decir, nomenclaturas propias que sean entendibles por todos los miembros.
- **Propiedad colectiva del código:** el código desarrollado es propiedad de todos los miembros del equipo y cualquiera de ellos puede y debe de modificarlo si lo considera necesario, para obtener un código mucho más eficiente y entendible.
- **Programación por parejas:** colocar a dos personas a realizar el trabajo puede resultar una gran ventaja, especialmente cuando uno de ellos piensa en un nivel táctico y el otro en un nivel estratégico para así conseguir que ambos procesos

se encuentren activos todo el tiempo lo que a su vez minimiza el número de errores y mejor la calidad del producto. Y por supuesto, es necesario que estos roles se intercalen continuamente para poder utilizar toda las posibilidades tácticas y estratégicas.

- **Integración continua:** conforme como se van desarrollado las partes estas deben de ser integradas al sistema, con el fin de detectar los errores desde un inicio y no cuando ya este todo completado.
- **40 horas semanales:** trabajar en exceso provoca que el equipo desarrolle un software de baja calidad puesto a que el equipo se encuentra cansado y desmotivado.
- **Metáfora del negocio:** en la metodología se crea una metáfora, una historia en común que describe cómo se comportan las diferentes partes del sistema, y esta debe de ser lo suficientemente bien redactada para que el cliente se sienta cómodo con la misma.
- **Cliente in situ:** es fundamente que el cliente se encuentre en el mismo lugar para poder mantener una comunicación cara a cara con el equipo de desarrollo, inclusive es considerado parte del equipo.
- **Entregas frecuentes:** es necesario desarrollar versiones pequeñas del sistema, sin que sea necesario que otorguen toda la funcionalidad del mismo, pero sí que permitan visionar como va a ser el resultado final y que el cliente se familiarice con el entorno y que agreguen un valor de negocio al cliente.
- **Planificación incremental:** la planificación en esta metodología debe de ser revisada constantemente de manera incremental, tomando en cuenta las necesidades de negocio que aporten más valor a la empresa.

CICLO DE VIDA

La metodología XP cuenta con ciclo de vida divido en 6 partes:

- **Fase de exploración:** dentro de esta fase se desarrollan tres procesos:
    - *Historias de usuario:* es en donde se describen las funcionalidades que se espera tenga el aplicativo, son escritas por el cliente final

y deben de ser descripciones cortas y concisas de lo que se espera que realice el sistema.

- *Spike arquitectónico:* es donde se prueba la tecnología y se conoce más a fondo la metodología para poder familiarizarse con la misma. Además, se realiza un prototipo que permita demostrar que la arquitectura es válida para el proyecto.
- *Metáfora del negocio:* es una historia común compartida por el equipo y el usuario que busca que el ultimo se sienta cómodo refiriéndose al sistema y a los desarrolladores a implementar las clases y objetos del sistema.

- **Fase de planificación:** se busca obtener una planificación después de seguir un procedimiento determinado en donde en primer lugar el cliente entrega las historias de usuario priorizando las funcionalidades de mayor a menor importancia, luego el equipo estudia y estima el coste de implementar estas funcionalidades, posteriormente se desarrolla una reunión para el plan de entregas y se determina el alcance del proyecto.
- **Fase de iteraciones:** cada una de las iteraciones puede tardar de dos a tres semanas, y siguen el siguiente proceso: recopilar las historias de usuario de cada iteración, se describen cada tarea que se ha de realizar por cada una de las historias, se priorizan las tareas que otorgan más valor al negocio y se reparten estas tareas al equipo y las menos importantes se colocan en una cola de tareas sin asignar para seleccionarlas posteriormente.
- **Fase de producción:** dentro de esta fase no se espera tener a todas las funcionalidades implementadas, sino únicamente algunas que permitan obtener un software que el usuario puede utilizar y que además brinde valor real al negocio.
- **Fase de mantenimiento:** después de que el alcance de proyecto ha sido alcanzado, se van verificando las nuevas historias de usuario que surjan después de que el proyecto haya sido colocado en producción, y se añaden las nuevas funcionalidades de acuerdo con el valor de negocio

que aportan, y del presupuesto que se requiera. En esta fase el equipo de desarrollo se reduce y deja únicamente a algunos miembros.

- **Fase de muerte del proyecto:** el proyecto entra en esta fase cuando no existen más historias de usuario o se ha reducido el valor de estas. Con el tiempo se irá desinvirtiendo en el mismo hasta dejarlo completamente cuando ya no aporte ningún valor para el negocio.

ROLES

- **Programador:** es el encargado de desarrollar las pruebas unitarias y realizar el código del programa.
- **Cliente:** es quién escribe las historias de usuario y las prioriza, además de que ofrece la dimensión del negocio, diseña las pruebas de aceptación y está disponible todo el tiempo puesto a que represente a los usuarios finales.
- **Encargado de pruebas o tester:** es quien ayuda al cliente para diseñar las pruebas de aceptación y posteriormente las ejecuta, así como las pruebas de integración para después difundir los resultados entre el cliente y el equipo. Es además el responsable de las herramientas automatizadoras que se utilizan en las pruebas.
- **Encargado de seguimiento o tracker:** es el que va a realimentar el proceso y ofrecer medidas que permitan mejorar las estimaciones siguientes. Además, sigue cada iteración del proceso y también evalúa las posibilidades de eliminar o incorporar nuevas historias de usuario.
- **Entrenador o coach:** es el que se encuentra encargado del proceso general y garantiza que se siga la metodología XP puesto a que conoce a fondo la misma, lo que a su vez le permite ayudar al equipo cuando requieran aplicar las prácticas de XP.
- **Consultor:** no es parte del equipo, posee conocimiento de un área en particular y únicamente ayuda a resolver un problema en específico.
- **Gestor:** es el responsable mayor del proyecto y quien forma un puente entre los clientes y el equipo.

2.3. GEOLOCALIZACIÓN

La geolocalización es mejor descrita como la obtención de la posición geográfica de una persona, lugar u objeto, en tiempo real. En la actualidad, se utilizan dispositivos habilitados para internet, smartphones o sistemas basados en GPS; para obtener esta posición. Esta ubicación en términos generales es únicamente un punto específico dentro de un mapa. Este punto se encuentra formado por dos componentes: latitud y longitud; los mismos que informaran al GPS donde se encuentra. Además, una ubicación puede ser encontrada de dos maneras: directa o referencial. En la primera, la ubicación es determinada por todos los puntos que se han obtenido por parte de un dispositivo GPS. Mientras que, en la manera referencial, se busca calcular algún tipo de co-relación entre dos puntos determinados. (Holdener, 2011)

El GPS (Sistema de Posicionamiento Global por sus siglas en inglés) es un sistema de navegación basado en la utilización de satélites que fue desarrollado por el Departamento de Defensa de Estados Unidos en los años 70's. Este sistema provee de información de posicionamiento continua desde cualquier parte del mundo bajo cualquier tipo de condición climática. Debido a que este sistema sirve a una gran cantidad de usuarios, y así también por cuestiones de seguridad, este sistema es de una sola vía, es decir, los usuarios pueden únicamente recibir las señales de los satélites. Consiste en tres segmentos: el segmento de espacio, el segmento de control y el segmento de usuario. El primero consta de una constelación de 24 satélites operativos. El segundo consiste en una red global de estaciones de rastreo. El tercer segmento incluye a todos los usuarios, tanto militares como civiles, que hacen uso de la información provista por el GPS. (El-Rabbany, 2002)

2.4. APLICACIONES MÓVILES

2.4.1. APLICACIÓN MÓVIL

Una aplicación móvil es un software diseñado específicamente para ser utilizado en un dispositivo móvil. Las aplicaciones son en general descargadas desde una tienda de aplicaciones virtual, e instaladas por el mismo dueño del dispositivo. Una vez que la aplicación haya sido instalada, esta trabaja en conjunto con el sistema operativo del dispositivo, lo que a su vez suele permitir que la aplicación tome ventaja de algunas características propias del dispositivo. (Salz & Moranz, 2013)

2.4.2. DISPOSITIVOS MÓVILES

Un dispositivo móvil es un dispositivo de computación portable, normalmente equipado con una pantalla táctil. Estos dispositivos otorgan a los usuarios de servicios de voz, datos y multimedia en cualquier momento y lugar, así como en cualquier formato. Los dispositivos móviles son capaces de ejecutar cualquier aplicación de software, conocidas como aplicaciones móviles. Además, los dispositivos móviles proveen de una infinita cantidad de aplicaciones para todo tipo de necesidad, desde ofrecer contenido educacional, de entretenimiento, salud e investigación, hasta la utilización de herramientas para simplificar las actividades de las personas. (Collins & Ellis, 2015)

2.4.3. SISTEMAS OPERATIVOS MÓVILES

Un sistema operativo móvil es un sistema desarrollado específicamente para dispositivos móviles, cómo, por ejemplo: teléfonos celulares, tabletas y otros dispositivos portátiles. Estos sistemas mezclan ciertas características de los sistemas operativos de las computadoras con las distintas características propias de los dispositivos móviles como la pantalla táctil, Wi-Fi, sistema de GPS, cámara de video, entre otras.

Los sistemas operativos móviles son responsables de identificar y definir cada una de estas distintas características, así como las funciones que cumplen los dispositivos móviles.

Aunque en la actualidad, muchas de las compañías han optado por lanzar al mercado sus propios sistemas operativos, las más comúnmente utilizadas continúan siendo: Android de Google Inc., iPhone OS de Apple, Windows de Microsoft y Bada de Samsung Electronics. (Khurana, 2014)

2.4.3.1. ANDROID

Android es un sistema operativo para teléfonos móviles que busca que estos permitan un funcionamiento parecido a las computadoras, así como una excelente conexión a la internet. Fue desarrollado en un principio por Google, aunque otros también ayudaron, y actualmente se encuentra en constantes actualizaciones y modificando su diseño para obtener un mejor producto. (Purdy, 2011)

Android puede ser adaptable a todo tipo de hardware, desde teléfonos hasta relojes, electrodomésticos y cámaras. Otra característica de Android es la portabilidad con la cuenta, así como la arquitectura de este, que se encuentra basada en componentes inspirados de Internet. Incluso se busca que los dispositivos que lo utilicen se encuentren conectados Internet todo el tiempo, además de contar con una gran variedad de servicios y un nivel de seguridad bastante aceptable junto con una alta calidad en los gráficos y el sonido. (Gironés, 2013)

2.5. BASES DE DATOS

Una base de datos es básicamente una colección de datos que se encuentran integrados controlado su redundancia y cuentan con una estructura que muestra las distintas interrelaciones y restricciones que se pueden encontrar en el mundo real. Los datos deben de ser independientes de las bases de datos, así como la definición y descripción de cada tipo de dato requieren encontrarse almacenadas junto a los datos. (Piñeiro, 2013)

Dentro de las bases de datos, a diferencia de los sistemas de ficheros tradicionales, no existirán datos redundantes a menos que sea

imprescindiblemente necesario. Además, los datos que se encuentran dentro de una base de datos son independientes de los sistemas de software que las utilizan, señalando así que cualquier cambio producido en la aplicación no deberá alterar la base de datos ni viceversa. (Piñeiro, 2013)

Las bases de datos cuentan con diversos tipos de modelos de datos, los mismos que pueden ser clasificados de manera general en conceptuales, lógicos o físicos.

2.5.1. SISTEMA GESTOR DE BASE DE DATOS

Es un sistema de software que permite el almacenamiento y acceso a la información de una base de datos, por medio de una serie de procedimientos. Este sistema nos permite definir a los datos de acuerdo con diferentes niveles de abstracción, así como su manipulación en la base de datos, es decir, se puede insertar, modificar, eliminar o consultar los datos. También, permite mantener la integridad, privacidad y seguridad de los datos y de la base de datos en sí. (Piñeiro, 2013)

2.5.1.1. FIREBASE

Dentro de una de las tecnologías disponibles en el servicio de Firebase, se puede encontrar una base de datos en tiempo real. Esta base de datos es no relacional y se encuentra en la nube. Permite sincronizar los dispositivos que se encuentren conectados e incluso se la puede acceder cuando no se encuentra con servicio de red, debido a su caché local. Esta base de datos es manejada de acuerdo con los eventos que sucedan y es muy diferente a las bases de datos relacionales tradicionales.

Los cambios que se realicen a los datos dentro de la base de datos disparan eventos en el código del cliente por lo que se puede tratar y actualizar el estado de la interfaz de usuario de acuerdo con la respuesta generada. Cuenta con reglas basadas en expresiones, que determinan que datos son estructurados y los permisos que los usuarios tienen para acceder a los mismos. El propósito de

esta base es ser responsiva, permitiendo así crear experiencia en tiempo real. (Moroney, 2017)

2.6. LENGUAJES DE PROGRAMACIÓN

Un lenguaje de programación se puede entender como el conjunto de caracteres y símbolos mezclados entre sí concorde con una sintaxis determina que les permite enviar instrucciones al computador. Se cuentan con dos tipos de lenguajes, de alto y de bajo nivel.

El lenguaje de bajo nivel es aquel que está más cerca de la arquitectura de la misma computadora. Aquí podemos encontrar al lenguaje de máquina, el mismo que se basa en la combinación de ceros y unos; y el lenguaje ensamblador, el mismo que permite desarrollar software que controle dispositivos de E/S, reproduzcan sonido, etc., aunque cuenta con la desventaja de que es el más difícil de escribir y depurar. (Catalinas, 2002)

Por otro lado, existen los lenguajes de alto nivel, los mismos que están más cerca al usuario y a los desarrolladores, y además son totalmente independientes de la arquitectura del computador, es decir, pueden ser ejecutados en cualquier computadora sin importar el microprocesador que posean. Pero, para poder traducir este tipo de lenguaje a uno de bajo nivel, como el lenguaje de máquina, es necesario de un traductor, puede ser un compilador, ensamblador o intérprete, para cada computadora. (Catalinas, 2002)

2.6.1. JAVA

Comentado [A-1]: Vas a usar Java?

Java es un lenguaje de programación orientado a objetos, sencillo, interpretado, distribuido y robusto, que permite la independencia de plataformas.

En primer lugar, todo lo que se encuentre dentro de Java es considerado un objeto, a excepción de los datos primitivos, los mismos que son encapsulados por clases incorporadas. Esta orientación a objetos permite manejar los problemas complejos, corregir problemas con mayor facilidad y una fácil reutilización. (Groussard, 2014)

Java a su vez permite tener una independencia con las plataformas. Es decir, que los programas desarrollados en Java pueden ser ejecutados en cualquier tipo de computadora siempre y cuando tenga instalado el Java Runtime Environment (JRE). Un JRE se encuentra disponible para cualquier computadora sin importar el sistema operativo. Desde computadoras con Windows, pasando por las computadoras Macintosh, computadoras con Unix o Linux e incluso celulares. (Groussard, 2014)

2.6.2. HTML

HTML o Hypertext Markup Language (Lenguaje de Marcas de Hipertexto), es un lenguaje estándar para desarrollar páginas web. El hipertexto es aquel texto desplegado en un dispositivo con computadora, y este contiene enlaces para poder navegar a otros documentos de hipertexto. Un documento de hipertexto describe distintos objetos como imágenes, tablas, formularios o listas.

Actualmente se encuentra en su quinta versión, HTML 5 y se encuentra a cargo de la W3C (World Wide Web Consortium), organización encargada de los estándares de las tecnologías vinculadas con la internet. (Kurniawan, 2015)

2.6.3. JAVASCRIPT

Es un lenguaje de programación interpretado que permite el desarrollo de aplicaciones. Además, tiene la capacidad de ser orientado a objetos. Es usado mayormente en navegadores web y puede ser insertado dentro del mismo código de HTML. JavaScript también permite que las variables no requieran ser de un tipo específico. A pesar de eso, el lenguaje cuenta con números, cadenas, y valores booleanos como tipo de datos primitivos.

JavaScript permite reconocer acciones generadas por el usuario, conocidos como eventos, lo que le permite al sistema ser mucho más interactivo con el usuario.

2.6.4. CSS

Las hojas de estilo en cascada (CSS por sus siglas en inglés) son aquellas que le permiten que las páginas HTML tengan un formato agradable para el usuario. Permite controlar desde los colores utilizados, hasta el estilo de fuente de los textos, márgenes, tamaño de columnas, y entre muchas otras cosas más. CSS puede ser utilizado en muchas páginas, lo que permite cambiar el diseño de todo un sitio web de manera rápida y eficaz. Es por eso por lo que ha sido recomendado por la W3C como el formato para las páginas HTML.

2.6.5. RESUMEN

Tecnología	Característica
Java	Java es un lenguaje de programación orientado a objetos, sencillo, interpretado, distribuido y robusto, que permite la independencia de plataformas.
HTML	Es un lenguaje estándar para desarrollar páginas web.
JavaScript	Es un lenguaje de programación interpretado que permite el desarrollo de aplicaciones. Tiene la capacidad de ser orientado a objetos.
CSS	Las hojas de estilo en cascada son aquellas que le permiten que las páginas HTML tengan un formato agradable para el usuario.

Tabla 0-1. Resumen Lenguajes de Programación (Marcillo D, 2019)

2.7. APLICACIONES WEB

2.7.1. SERVICIOS WEB

Los servicios web son objetos cuyas funciones son accesibles por medio de determinados protocolos de red, sin tomar en consideración las distintas tecnologías o plataformas que se encuentran involucradas en la solicitud. (Lequerica, 2003)

Es considerado como una parte de la lógica del negocio, puesto a que es capaz de procesar determinadas solicitudes, así como tiene la capacidad de ser accesido desde cualquier lugar por personas que cuenten con un determinado permiso por medio de cualquier medio. Es por eso que este se ubicará entre el usuario y el código del programa y estará acargo de abstraer las distintas especificaciones técnicas de la aplicación que atienda al llamado, para que de esta manera cualquier lenguaje de programación pueda acceder a la aplicación. (Lequerica, 2003)

2.8. INTERNET

La internet podría entenderse como el conjunto masivo de redes de computadoras que se encuentra conectadas en todo el mundo por medio de torres de radiocomunicación, satélites y cableado de fibra óptica. Dentro de la internet se puede encontrar una gigantesca cantidad de información, así como programas en las computadoras que se encuentran conectadas. También se puede incluir a contenido multimedia, como música y video. (Cabello, 2014)

También, se debe de tomar en cuenta que la internet es dinámica, nunca se mantiene igual. Constantemente se modifican sus usos, así como la infraestructura mundial. Igualmente, no existe ningún tipo de frontera internacional en cuanto a la internet, no importa el lenguaje o la computadora. (Cabello, 2014)

2.9. SERVIDORES DE APLICACIONES

Los servidores de aplicaciones son los que van a ejecutar algunos programas que una aplicación, o incluso el usuario, solicitan a un servidor determinado. Básicamente es un servidor que está dentro de una red específica y que ejecuta determinadas aplicaciones de software con el fin de ofrecer ciertos servicios a los programas o usuarios que lo requieran. Estos servidores generalmente gestionan gran parte de las funciones de lógica de negocio de una aplicación web, así como el acceso a los datos. (Cabello, 2014)

Cuentan con una Interfaz de Programación de Aplicaciones (API) que de alguna manera permite que los desarrolladores tengan que realizar ciertos códigos. Además de esto, los servidores de aplicaciones a su vez pueden otorgar cierto soporte a los principales estándares soportados en el Intenert y en determinados sitios web. Un servidor de aplicaciones debe de tener:

- **Alta disponibilidad:** los sistemas deben de encontrarse en constante funcionamiento las 24 horas del día, los 365 días del año. Por lo que se deben contar con algun tipo de reemplazo en caso de que el sistema principal presente fallos.
- **Escalabilidad:** en caso de que la carga de trabajo aumente, el servidor a su vez debe de poseer la capacidad de ser ampliado según como vaya creciendo.
- **Mantenimiento:** se necesita que los servidores brinde un mantenimiento sencillo y rápido, por lo que se necesitan implementar módulos para que el servidor trabaje.

CAPÍTULO 3. ITERACIONES

3.1. ITERACIÓN 1 - EMPLEADOS

3.1.1. PLANEACIÓN

3.1.1.1. HISTORIAS DE USUARIO

	Enunciado de la historia				Criterios de aceptación			
Identificador (ID) de la historia	**Rol**	**Característica / Funcionalidad**	**Razón / Resultado**	**Número (#) de escenario**	**Criterio de aceptación (Título)**	**Contexto**	**Evento**	**Resultado / Comportamiento esperado**
HU-E1	Como administrador	Necesito ingresar la información de un nuevo empleado.	Con la finalidad de agregarlo a la base de datos de la empresa.	1	Ingresar nuevo	En caso de que se desee ingresar un nuevo empleado.	Cuando se desplieguen los campos para el ingreso.	A continuación, se mostrará un mensaje indicando que el nuevo registro ha sido almacenado en la base de datos.
HU-E2	Como administrador	Necesito modificar la información de un empleado existente.	Con la finalidad de hacer cambios en su información.	1	Editar empleado existente	En caso de que se requiera modificar la información de un empleado existente.	Cuando se desplieguen los campos para la modificación.	A continuación, se mostrará un mensaje indicando que el registro ha sido modificado.
HU-E3	Como administrador	Necesito eliminar de manera permanente un empleado existente.	Con la finalidad de borrar su registro.	1	Eliminar empleado	En caso de que se busque eliminar un empleado del registro.	Cuando acepte eliminar el empleado.	A continuación, se mostrará un mensaje indicando que el empleado ha sido eliminado de manera exitosa.
HU-E4	Como administrador	Necesito ver un listado de todos los empleados.	Con la finalidad de verificar su información rápidamente.	1	Existen empleados	En caso de que existan empleados registrados.	Cuando se despliegue el listado de empleados.	A continuación, se mostrará un listado con la información de todos los empleados registrados.

Tabla 1-1. Historia de Usuario 1 – Administrar Empleados (Marcillo D, 2018)

3.1.1.2. PLAN DE ITERACIÓN

PLAN DE ITERACIONES					
Número de Iteración	**Fecha Inicio**	**Fecha Fin**	**Número de Historia**	**Nombre de Historia**	**Responsable**
1	01-04-2018	15-04-2018	1	Administrar Empleados	María Gloria Marcillo
2	16-04-2018	05-05-2018	2	Administrar Vehículos	María Gloria Marcillo
3	08-05-2018	30-05-2018	3	Administrar Clientes	María Gloria Marcillo
4	01-06-2018	30-06-2018	4	Administrar Visitas	María Gloria Marcillo
5	01-07-2018	30-07-2018	5	Administrar Localizaciones	María Gloria Marcillo

Tabla 1-2. Plan de Iteraciones – Empleados (Marcillo D, 2018)

3.1.1.3. CRITERIOS DE PRUEBA DE ACEPTACIÓN

VISIÓN DE LA PRUEBA			
Nombre	Información Empleado	**Código**	CPA-E1
N.º Historia	1		
Iteración	1		
Función crítica	Si		
Descripción	Verifica que se pueda acceder a la información de los empleados.		
Categoría	Administración de Empleados		
Prerrequisito			

Tabla 1-3. Criterio de Prueba de Aceptación 1 (Marcillo D, 2018)

VISIÓN DE LA PRUEBA			
Nombre	Manipulación datos Empleado	**Código**	CPA-E2
N.º Historia	1		
Iteración	1		
Función crítica	Si		
Descripción	Verifica que se pueda ingresar, modificar y eliminar los registros de empleados en la base de datos.		
Categoría	Administración de Empleados		
Prerrequisito			

Tabla 1-4. Criterio de Prueba de Aceptación 2 (Marcillo D, 2018)

3.1.2. DISEÑO

3.1.2.1. DISEÑO SIMPLE

WEB



Figura 1-1. Diseño Simple – Empleados (Marcillo D, 2018)

3.1.2.2. TARJETAS CRC

NÚMERO:	T01	**NOMBRE:**	Empleado
DESCRIPCIÓN:		En esta clase se podrá controlar el ingreso, modificación, eliminación y consulta de los datos de los empleados.	
RESPONSABILIDADES		**COLABORADORES**	**MÉTODOS**
Ingreso de Empleados		Administrador	insertEmpleado
Modificación de Empleados		Administrador	updateEmpleado
Eliminación de Empleados		Administrador	deleteEmpleado
Consulta de Empleados		Administrador	getEmpleados
Creación Usuario		Administrador	createUser

Tabla 1-5. Tarjeta CRC – Empleados (Marcillo D, 2018)

3.1.2.3. PROTOTIPOS

WEB

En la siguiente imagen se puede observar una tabla que contiene la información de todos los empleados, así como botones que permiten editar su información, o borrarlos de la base de datos. Además, se observa una sección donde se puede ingresar un nuevo cliente, con los respectivos campos requeridos.

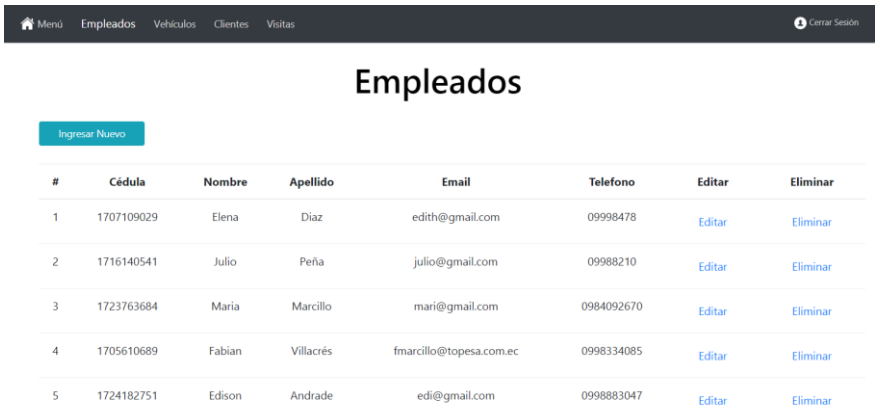


Figura 1-2. Prototipo 1 – WEB – Empleados (Marcillo D, 2018)

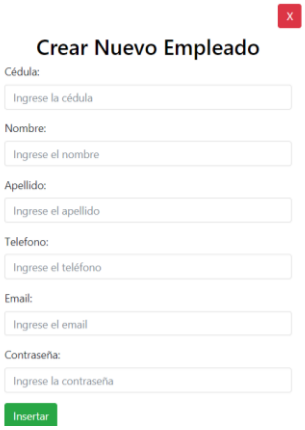


Figura 1-3. Prototipo 2 – WEB – Empleados (Marcillo D, 2018)

3.1.3. CODIFICACIÓN

3.1.3.1. ESTÁNDARES DE CODIFICACIÓN

- **Identificador**: la clave identificadora del empleado tomará la siguiente forma: key_empleado.
- **Variables**: todas las variables empleadas serán nombradas de acuerdo a su valor o característica. La primera letra de cada una de las palabras será mayúscula, con excepción de que la primera letra de la primera palabra es minúscula. Ejemplos: returnArrEmpleados, nombre, telefono.
- **Funciones**: las funciones vendrán a describir su respectiva tarea u operación. La primera palabra será en minúsculas y las otras contarán únicamente con la primera letra mayúscula. Ejemplo: cargarMapa(), verificarIdentificacion(), editar(), eliminar().
- **Bases**: las bases serán nombradas como dbRef, más un guión bajo y el nombre de la clase en minúsculas. Ejemplo: dbRef_empleados, dbRef_vehiculos.
- **Botones**: los botones llevarán los caracteres 'btn' al inicio, más el nombre del botón con la primera letra en mayúscula. Ejemplo: btnEditar, btnNuevo.
- **Divisiones**: las divisiones tendrán al inicio los caracteres 'div', más el nombre de la división con la primera letra en mayúscula. Ejemplo: divNuevo, divMapa.

3.1.3.2. CÓDIGO

CLASE

La clase de Empleado cuenta con 6 campos: cédula, nombre, apellido, teléfono, email, y el identificador. Todos los campos son de tipo string o cadena.

CONTROLADOR

Cuenta con 12 métodos que permitirán realizar las distintas acciones que se han enviadas desde la aplicación. Estos son:

- ejecutarInsert (): obtiene la información ingresada en la aplicación y la coloca en distintas variables para su posterior utilización.
- createUser (): cuando un empleado es ingresado por primera vez, es necesario crear un usuario que le permita acceder al sistema móvil. Este método es el que permite crear el mismo a partir de un correo electrónico y una contraseña.
- insertar (): permite ingresar los datos del empleado en la base de datos. Además, posteriormente emite una alerta señalando que el empleado ha sido registrado exitosamente.
- ejecutarEditar (): obtiene los datos del empleado a ser modificado, los coloca en variables y llama al siguiente método para poder actualizar la información en la base de datos.
- editar (): actualiza los datos del empleado dentro de la base de datos.
- ejecutarEliminar (): verifica con el usuario que se encuentre seguro de eliminar el empleado seleccionado.
- eliminar (): elimina al empleado de la base de datos de manera permanente.
- mostrarNuevo (): prepara el formulario para el ingreso de un nuevo empleado, dejando todos los campos en vacío y controlando la visibilidad de los botones.
- cargarEditar (): obtiene los datos del empleado desde la base de datos y los coloca en el formulario para su posterior edición.
- mostrarExistente (): controla los campos y botones que podrán ser visibles cuando se va a modificar un empleado existente.
- cerrarNuevo (): oculta el formulario.
- logOut (): permite cerrar la sesión del usuario actual.

3.1.4. PRUEBAS

3.1.4.1. PRUEBA DE ACEPTACIÓN

PRUEBA DE ACEPTACIÓN	
Nombre:	Información Empleados
Código:	1
Descripción:	Se debe visualizar toda la información de todos los empleados, la misma que debe incluir: cédula de identidad, nombres y apellidos, y un teléfono de contacto, de cada uno.
Condiciones de ejecución:	Los datos deben estar registrados en la base de datos.
Entradas:	Ninguna
Pasos de ejecución:	1. Abrir el menú de Administración de Empleados 2. Verificamos la información desplegada en la tabla.
Resultado esperado:	Dentro de la tabla se debe de visualizar cada uno de los empleados con sus respectivos datos.
Evaluación de la prueba:	La prueba realizada finalizó con un éxito al 100% debido a que se muestra la información de todos los empleados registrados en la base de datos.

Tabla 1-6. Prueba de Aceptación (Marcillo D, 2018)

3.1.4.2. PRUEBA DE CAJA BLANCA

Las pruebas de caja blanca realizadas durante esta iteración permitieron comprobar que tanto el código, como la estructura funcionase de manera adecuada. Esta prueba fue realizada durante la implementación de la iteración, al verificar que no existiesen ningún tipo de problemas en el código, que todos los métodos: de ingreso, modificación, eliminación y consulta de los empleados; trabajen efectiva y conjuntamente; y que el sistema interactúe adecuadamente con la base de datos.

3.1.4.3. PRUEBA DE CAJA NEGRA

ENTRADAS	RESULTADOS ESPERADOS	RESULTADO
Despliegue de pantalla	La pantalla se muestre correctamente, así como todos sus componentes.	OK ✓
Ingreso de empleado	Los empleados son ingresados a la base de datos correctamente.	OK ✓
Modificación de empleado	La información de los empleados puede ser modificada sin ningún problema.	OK ✓
Eliminación de empleado	Los empleados pueden ser eliminados de la base de datos.	OK ✓
Consulta de empleados	La información de todos los empleados es presentada al usuario.	OK ✓

Tabla 1-7. Prueba de Caja Negra (Marcillo D, 2018)

3.2. ITERACIÓN 2 - VEHÍCULOS

3.2.1. PLANEACIÓN

3.2.1.1. HISTORIAS DE USUARIO

	Enunciado de la historia				Criterios de aceptación			
Identificador (ID) de la historia	**Rol**	**Característica / Funcionalidad**	**Razón / Resultado**	**Número (#) de escenario**	**Criterio de aceptación (Título)**	**Contexto**	**Evento**	**Resultado / Comportamiento esperado**
HU-V1	Como un administrador	Necesito ingresar la información de los vehículos de un empleado.	Con la finalidad de agregarlo a la base de datos de la empresa.	1	Ingresar nuevo vehículo	En caso de que se requiera ingresar un nuevo vehículo.	Cuando se desplieguen los campos para el ingreso.	A continuación, se mostrará un mensaje indicando que el vehículo ha sido almacenado correctamente.
HU-V2	Como un administrador	Necesito modificar los datos de un vehículo registrado.	Con la finalidad de modificar su información cuando sea necesario.	1	Modificar un vehículo	En caso de que se necesite modificar la información de un vehículo.	Cuando se desplieguen los campos para la modificación.	A continuación, se mostrará un mensaje indicando que el vehículo ha sido modificado correctamente.
HU-V3	Como un administrador	Necesito eliminar vehículos de la base	Con la finalidad de borrar un registro considerado innecesario.	1	Eliminar vehículo	En caso de que se requiera eliminar un vehículo	Cuando se despliegue la información del vehículo.	A continuación, se mostrará un mensaje indicando que el vehículo ha sido eliminado.
HU-V4	Como un administrador	Necesito ver un listado de todos los vehículos.	Con la finalidad de conocer la información de cada uno.	1	Existen vehículos.	En caso de que existan vehículos registrados.	Cuando se despliegue un listado de los vehículos.	A continuación, se mostrará un listado con la información de todos los vehículos.

Tabla 2-1. Historia de Usuario 2 – Administrar Vehículos (Marcillo D, 2018)

3.2.1.2. PLAN DE ITERACIÓN

PLAN DE ITERACIONES					
Número de Iteración	**Fecha Inicio**	**Fecha Fin**	**Número de Historia**	**Nombre de Historia**	**Responsable**
1	01-04-2018	15-04-2018	1	Administrar Empleados	María Gloria Marcillo
2	16-04-2018	05-05-2018	2	Administrar Vehículos	María Gloria Marcillo
3	08-05-2018	30-05-2018	3	Administrar Clientes	María Gloria Marcillo
4	01-06-2018	30-06-2018	4	Administrar Visitas	María Gloria Marcillo
5	01-07-2018	30-07-2018	5	Administrar Localizaciones	María Gloria Marcillo

Tabla 2-2. Plan de Iteración (Marcillo D, 2018)

3.2.1.3. CRITERIOS DE PRUEBA DE ACEPTACIÓN

VISIÓN DE LA PRUEBA			
Nombre	Manipulación datos Vehículo	**Código**	CPA-V1
N.º Historia	2		
Iteración	2		
Función crítica	Si		
Descripción	Verifica que se pueda ingresar, modificar y eliminar los registros de los vehículos en la base de datos.		
Categoría	Administración de Vehículos		
Prerrequisito			

Tabla 2-3. Criterio de Prueba de Aceptación 1 (Marcillo D, 2018)

VISIÓN DE LA PRUEBA			
Nombre	Información Vehículo	**Código**	CPA-V2
N.º Historia	2		
Iteración	2		
Función crítica	Si		
Descripción	Verifica que se pueda acceder a la información de los vehículos.		
Categoría	Administración de Vehículos		
Prerrequisito			

Tabla 2-4. Criterio de Prueba de Aceptación 2 (Marcillo D, 2018)

3.2.2. DISEÑO

3.2.2.1. DISEÑO SIMPLE



Figura 2-1. Diseño Simple – WEB – Vehículos (Marcillo D, 2018)

3.2.2.2. TARJETAS CRC

NÚMERO:	T02	**NOMBRE:**	Vehículo
DESCRIPCIÓN:		En esta clase se podrá controlar el ingreso, modificación, eliminación y consulta de los datos de los vehículos.	
RESPONSABILIDADES		**COLABORADORES**	**MÉTODOS**
Ingreso de Vehículos		Administrador	insertVehiculo
Modificación de Vehículos		Administrador	updateVehiculo
Eliminación de Vehículos		Administrador	deleteVehiculo
Consulta de Vehículos		Administrador	getVehiculos

Tabla 2-5. Tarjeta CRC – Vehículos (Marcillo D, 2018)

3.2.2.3. PROTOTIPOS

En la siguiente pantalla se puede observar una tabla que contiene la información de todos los vehículos registrados en la base de datos, así como las opciones para crear uno nuevo, editar un vehículo existente, o incluso eliminarlo completamente.

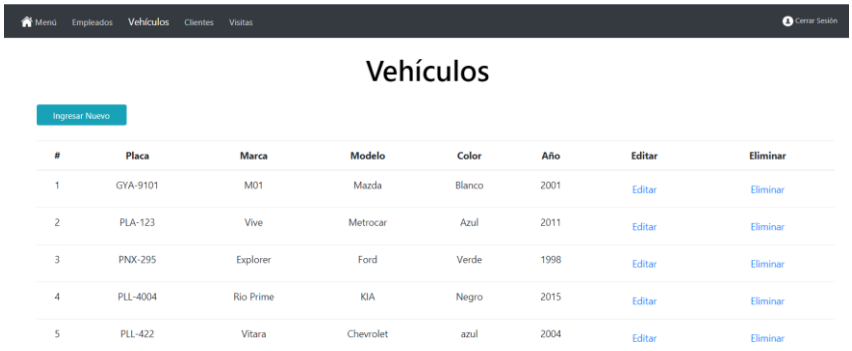


Figura 2-2. Prototipo 1 – WEB – Vehículos (Marcillo D, 2018)

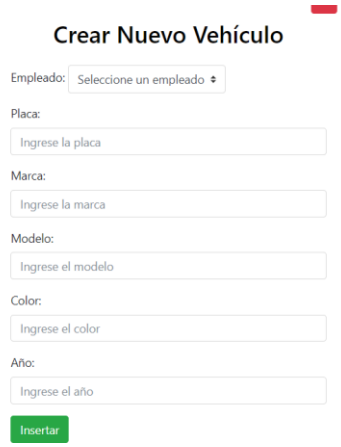


Figura 2-3. Prototipo 2 – WEB – Vehículos (Marcillo D, 2018)

3.2.3. CODIFICACIÓN

3.2.3.1. ESTÁNDARES DE CODIFICACIÓN

- **Identificador**: la clave identificadora del empleado tomará la siguiente forma: key_vehiculo.
- **Variables**: todas las variables empleadas serán nombradas de acuerdo a su valor o característica. La primera letra de cada una de las palabras será mayúscula, con excepción de que la primera letra de la primera palabra es minúscula. Ejemplos: placaVehiculo, marca.
- **Funciones**: las funciones vendrán a describir su respectiva tarea u operación. La primera palabra será en minúsculas y las otras contarán únicamente con la primera letra mayúscula. Ejemplo: cargarMapa(), verificarRepeticion(), editar(), eliminar().
- **Bases**: las bases serán nombradas como dbRef, más un guión bajo y el nombre de la clase en minúsculas. Ejemplo: dbRef_empleados, dbRef_vehiculos.

- **Botones**: los botones llevarán los caracteres 'btn' al inicio, más el nombre del botón con la primera letra en mayúscula. Ejemplo: btnEditar, btnNuevo.
- **Divisiones**: las divisiones tendrán al inicio los caracteres 'div', más el nombre de la división con la primera

3.2.3.2. CÓDIGO

CLASE

La clase de Vehículo cuenta con 7 campos: identificador, empleadoID, placa, marca, modelo, color y año. Todos los campos son de tipo string o cadena.

CONTROLADOR

Cuenta con 12 métodos que permitirán realizar las distintas acciones que se han enviadas desde la aplicación. Estos son:

- ejecutarInsert (): obtiene la información del formulario y la guarda en variables temporales para su posterior utilización.
- insertar (): almacena la información del vehículo en la base de datos y notifica al usuario sobre esta acción.
- ejecutarEditar (): recibe los datos del formulario y los coloca en variables para usarlas en el siguiente método.
- verificarRepeticion (): verifica que las placas de los vehículos no se repitan.
- editar (): actualiza los datos del vehículo dentro de la base de datos.
- ejecutarEliminar (): verifica con el usuario que se encuentre seguro de eliminar el vehículo seleccionado.
- eliminar (): elimina permanentemente al vehículo de la base de datos.
- mostrarNuevo (): limpia el formulario de cualquier texto existente para el ingreso de un nuevo registro.
- cargarEditar (): extrae la información de un vehículo existente en la base de datos, y la coloca en el formulario para su edición.

- mostrarExistente (): controla la visibilidad de ciertos botones y campos durante el proceso de edición.
- cerrarNuevo (): oculta el formulario.
- logOut (): permite cerrar la sesión del usuario actual.

3.2.4. PRUEBAS

3.2.4.1. PRUEBA DE ACEPTACIÓN

PRUEBA DE ACEPTACIÓN	
Nombre:	Información Vehículos
Código:	2
Descripción:	Se debe visualizar toda la información de todos los vehículos, la misma que debe incluir: placa, marca, modelo, color y año.
Condiciones de ejecución:	Los datos deben estar registrados en la base de datos.
Entradas:	Ninguna
Pasos de ejecución:	1. Abrir el menú de Administración de Vehículos 2. Verificamos la información desplegada en la tabla.
Resultado esperado:	Dentro de la tabla se debe de visualizar cada uno de los vehículos con sus respectivos datos.
Evaluación de la prueba:	La prueba realiza finalizó con un éxito al 100% debido a que se muestra la información de todos los vehículos registrados en la base de datos.

Tabla 2-6. Prueba de Aceptación (Marcillo D, 2018)

3.2.4.2. PRUEBA DE CAJA BLANCA

La prueba de caja blanca realizada durante esta iteración permitió comprobar que tanto el código, como la estructura funcionase de manera adecuada. Esta prueba fue realizada durante la implementación de la iteración, al verificar que no existiesen ningún tipo de problemas en el código, que todos los métodos: de ingreso, modificación, eliminación y consulta de los vehículos; trabajen efectiva y conjuntamente; y que el sistema interactúe adecuadamente con la base de datos.

3.2.4.3. PRUEBA DE CAJA NEGRA

ENTRADAS	RESULTADOS ESPERADOS	RESULTADO
Despliegue de pantalla	La pantalla se muestre correctamente, así como todos sus componentes.	OK ✓
Ingreso de vehículo	Los vehículos son ingresados a la base de datos correctamente.	OK ✓
Modificación de vehículo	La información de los vehículos puede ser modificada sin ningún problema.	OK ✓
Eliminación de vehículo	Los vehículos pueden ser eliminados de la base de datos.	OK ✓
Consulta de vehículos	La información de todos los vehículos es presentada al usuario.	OK ✓

Tabla 2-7. Prueba de Caja Negra (Marcillo D, 2018)

3.3. ITERACIÓN 3 - CLIENTES

3.3.1. PLANEACIÓN

3.3.1.1. HISTORIAS DE USUARIO

	Enunciado de la historia				Criterios de aceptación			
Identificador (ID) de la historia	**Rol**	**Característica / Funcionalidad**	**Razón / Resultado**	**Número (#) de escenario**	**Criterio de aceptación (Título)**	**Contexto**	**Evento**	**Resultado / Comportamiento esperado**
HU-C1	Como un administrador	Necesito modificar la información de los clientes de la empresa	Con la finalidad de modificar la información del mismo cuando se necesario.	1	Modificación cliente existente	En caso de que se vaya a modificar la información de un cliente.	Cuando se desplieguen los campos para la modificación.	A continuación, se mostrará un mensaje indicando que el registro ha sido modificado.
HU-C2	Como un administrador	Necesito eliminar de manera permanente un empleado existente	Con la finalidad de borrar su registro.	1	Eliminar empleado	En caso de que se busque eliminar un empleado del cliente.	Cuando acepte eliminar el cliente.	A continuación, se mostrará un mensaje indicando que el cliente ha sido eliminado de manera exitosa.
HU-C3	Como un administrador	Necesito ver un listado de todos los clientes registrados.	Con la finalidad de conocer la información de cada uno.	1	Existen clientes.	En caso de que existan clientes registrados.	Cuando se despliegue un listado de los clientes.	A continuación, se mostrará un listado con la información de todos los clientes registrados.

Tabla 3-1. Historia de Usuario 3 – Administrar Clientes (Marcillo D, 2018)

3.3.1.2. PLAN DE ITERACIÓN

PLAN DE ITERACIONES					
Número de Iteración	**Fecha Inicio**	**Fecha Fin**	**Número de Historia**	**Nombre de Historia**	**Responsable**
1	01-04-2018	15-04-2018	1	Administrar Empleados	María Gloria Marcillo
2	16-04-2018	05-05-2018	2	Administrar Vehículos	María Gloria Marcillo
3	08-05-2018	30-05-2018	3	Administrar Clientes	María Gloria Marcillo
4	01-06-2018	30-06-2018	4	Administrar Visitas	María Gloria Marcillo
5	01-07-2018	30-07-2018	5	Administrar Localizaciones	María Gloria Marcillo

Tabla 3-2. Plan de Iteración (Marcillo D, 2018)

3.3.1.3. CRITERIOS DE PRUEBA DE ACEPTACIÓN

VISIÓN DE LA PRUEBA			
Nombre	Manipulación datos Cliente	**Código**	CPA-C1
N.º Historia	3		
Iteración	3		
Función crítica	Si		
Descripción	Verifica que se pueda ingresar, modificar y eliminar los registros de los clientes en la base de datos.		
Categoría	Administración de Clientes		

Tabla 3-3. Criterio de Prueba de Aceptación 1 (Marcillo D, 2018)

VISIÓN DE LA PRUEBA			
Nombre	Información Cliente	**Código**	CPA-C2
N.º Historia	3		
Iteración	3		
Función crítica	Si		
Descripción	Verifica que se pueda acceder a la información de los clientes.		
Categoría	Administración de Clientes		
Prerrequisito			

Tabla 3-4. Criterio de Prueba de Aceptación 2 (Marcillo D, 2018)

3.3.2. DISEÑO

3.3.2.1. DISEÑO SIMPLE

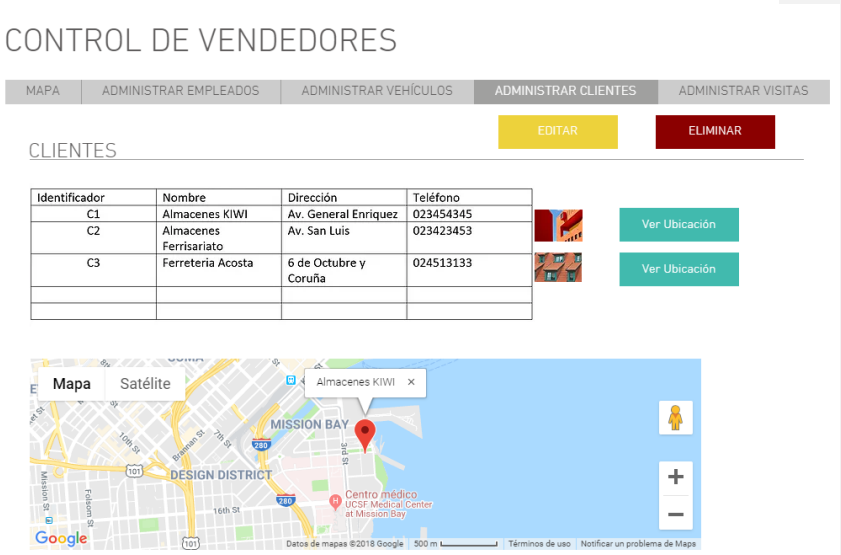


Figura 3-1. Diseño Simple – WEB – Clientes (Marcillo D, 2018)

3.3.2.2. TARJETAS CRC

NÚMERO:	T03	**NOMBRE:**	Cliente
DESCRIPCIÓN:		En esta clase se podrá controlar el ingreso, modificación, eliminación y consulta de los datos de los clientes.	
RESPONSABILIDADES		**COLABORADORES**	**MÉTODOS**
Modificación de Clientes		Administrador	updateCliente
Eliminación de Clientes		Administrador	deleteCliente
Consulta de Clientes		Administrador	getClientes

Tabla 3-5. Tarjeta CRC – Clientes (Marcillo D, 2018)

3.3.2.3. PROTOTIPOS

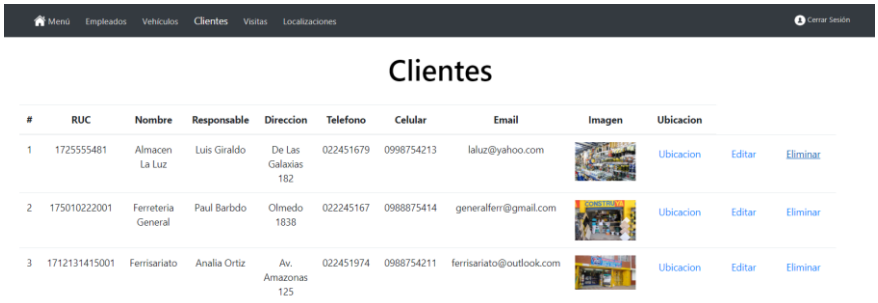


Figura 3-2. Prototipo 1 – WEB – Clientes (Marcillo D, 2018)

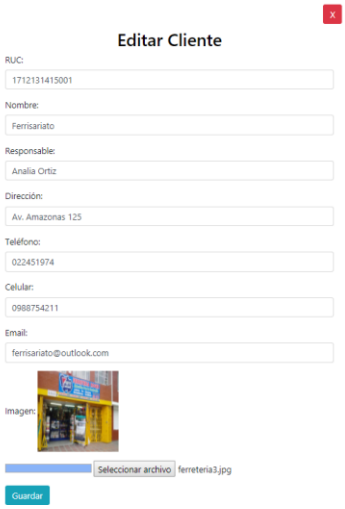


Figura 3-3. Prototipo 2 – WEB – Clientes (Marcillo D, 2018)

3.3.3. CODIFICACIÓN

3.3.3.1. ESTÁNDARES DE CODIFICACIÓN

- **Identificador**: la clave identificadora del empleado tomará la siguiente forma: key_cliente.
- **Variables**: todas las variables empleadas serán nombradas de acuerdo a su valor o característica. La primera letra de cada una de las palabras será mayúscula, con excepción de que la primera letra de la primera palabra es minúscula. Ejemplos: identificacion, imagenFile.
- **Funciones**: las funciones vendrán a describir su respectiva tarea u operación. La primera palabra será en minúsculas y las otras contarán únicamente con la primera letra mayúscula. Ejemplo: cargarMapa(), editar(), eliminar().
- **Bases**: las bases serán nombradas como dbRef, más un guión bajo y el nombre de la clase en minúsculas. Ejemplo: dbRef_localizaciones.

- **Botones**: los botones llevarán los caracteres 'btn' al inicio, más el nombre del botón con la primera letra en mayúscula. Ejemplo: btnVer.
- **Divisiones**: las divisiones tendrán al inicio los caracteres 'div', más el nombre de la división con la primera letra en mayúscula. Ejemplo: divNuevo, divMapa.

3.3.3.2. CÓDIGO

CLASE

La clase de Cliente cuenta con 10 campos: identificación, nombre, responsable, dirección, teléfono, celular, email, imagen, ubicación y clienteID.

CONTROLADOR

La información de esta clase será tratada únicamente en modo lectura, dentro de la aplicación web. Cuenta con 2 métodos. Estos son:

- cargarDatos (): obtiene la información todos los clientes que se encuentran dentro de la base de datos y la coloca en la tabla.
- cargarEditar (): extrae la información del cliente de la base de datos, y la coloca en el formulario para su edición.
- ejecutarEditar (): obtiene los datos del formulario y los coloca en variables para utilizarlas en el siguiente método.
- editar (): actualiza los datos del cliente dentro de la base de datos.
- ejecutarEliminar (): verifica que el cliente desee eliminar el cliente seleccionado.
- eliminar (): elimina permanentemente el cliente de la base de datos.
- subirImagen (): permite seleccionar la imagen y despliega la misma en el formulario.
- verUbicacionMapa (): muestra la ubicación del cliente dentro del mapa.
- verTodosMapa (): muestra la ubicación de todos los clientes en el mapa.
- cerrar (): oculta el formulario.
- cerrarMapa (): oculta el mapa.

- logOut (): permite cerrar la sesión del usuario actual.

3.3.4. PRUEBAS

3.3.4.1. PRUEBA DE ACEPTACIÓN

PRUEBA DE ACEPTACIÓN	
Nombre:	Modificar Clientes
Código:	3
Descripción:	Debe modificar los datos del cliente, excepto la imagen y ubicación.
Condiciones de ejecución:	Todos los datos deben estar completos.
Entradas:	Datos del cliente.
Pasos de ejecución:	1. Seleccionar el cliente a modificar. 2. Modificar los campos que se requiera. 3. Guardar la información.
Resultado esperado:	Todos los datos del cliente son modificados correctamente dentro de la base de datos.
Evaluación de la prueba:	La prueba realiza finalizó con un éxito al 100% debido a que se modificó la información del cliente correctamente.

Tabla 3-6. Prueba de Aceptación (Marcillo D, 2018)

3.2.4.2. PRUEBA DE CAJA BLANCA

La prueba de caja blanca realizada durante esta iteración permitió comprobar que el código realiza todas las funciones de manera correcta. La prueba fue realizada durante el proceso de desarrollo, verificando que no existieran errores en el código y que los métodos cumplieran sus funciones.

3.2.4.3. PRUEBA DE CAJA NEGRA

ENTRADAS	RESULTADOS ESPERADOS	RESULTADO
Despliegue de pantalla	La pantalla se muestre correctamente, así como todos sus componentes.	OK ✓
Modificación de cliente	Los clientes son modificados dentro de la base de manera adecuada.	OK ✓
Eliminación del cliente	Los clientes pueden ser eliminados de la base de datos.	OK ✓
Consulta de clientes	La información de todos los clientes es presentada al usuario.	OK ✓

Tabla 3-7. Prueba de Caja Negra (Marcillo D, 2018)

3.4. ITERACIÓN 4 – VISITAS

3.4.1. PLANEACIÓN

3.4.1.1. HISTORIAS DE USUARIO

	Enunciado de la historia				Criterios de aceptación			
Identificador (ID) de la historia	**Rol**	**Característica / Funcionalidad**	**Razón / Resultado**	**Número (#) de escenario**	**Criterio de aceptación (Título)**	**Contexto**	**Evento**	**Resultado / Comportamiento esperado**
HU-Vi1	Como un administrador	Necesito ver un listado de todas las visitas registradas en la base, de acuerdo con una fecha determinada.	Con la finalidad de conocer si los empleados están realizando su trabajo adecuadamente.	1	Existen visitas	En caso de que existan visitas registradas.	Cuando se despliegue un listado de todas las visitas a una fecha.	A continuación, se mostrará un listado con la información de todas las visitas realizadas en una determinada fecha.

Tabla 4-1. Historia de Usuario 4 – Administrar Visitas (Marcillo D, 2018)

3.4.1.2. PLAN DE ITERACIÓN

PLAN DE ITERACIONES					
Número de Iteración	**Fecha Inicio**	**Fecha Fin**	**Número de Historia**	**Nombre de Historia**	**Responsable**
1	01-04-2018	15-04-2018	1	Administrar Empleados	María Gloria Marcillo
2	16-04-2018	05-05-2018	2	Administrar Vehículos	María Gloria Marcillo
3	08-05-2018	30-05-2018	3	Administrar Clientes	María Gloria Marcillo
4	01-06-2018	30-06-2018	4	Administrar Visitas	María Gloria Marcillo
5	01-07-2018	30-07-2018	5	Administrar Localizaciones	María Gloria Marcillo

Tabla 4-2. Plan de Iteración (Marcillo D, 2018)

3.4.1.3. CRITERIOS DE PRUEBA DE ACEPTACIÓN

VISIÓN DE LA PRUEBA			
Nombre	Información Visita	**Código**	CPA-Vi1
N.º Historia	4		
Iteración	4		
Función crítica	Si		
Descripción	Verifica que se pueda acceder a la información de las visitas.		
Categoría	Administración de Visitas		
Prerrequisito			

Tabla 4-3. Criterio de Prueba de Aceptación (Marcillo D, 2018)

3.4.2. DISEÑO

3.4.2.1. DISEÑO SIMPLE

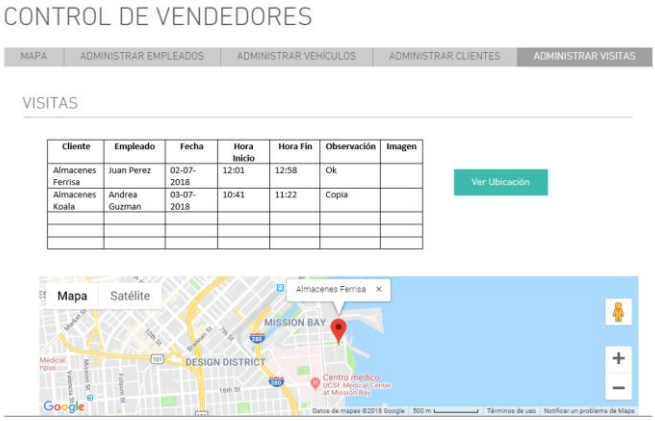


Figura 4-1. Diseño Simple – WEB – Visitas (Marcillo D, 2018)

3.4.2.2. TARJETAS CRC

NÚMERO:	T04	**NOMBRE:**	Visita
DESCRIPCIÓN:		En esta clase se podrá controlar el ingreso y consulta de los datos de las visitas.	
RESPONSABILIDADES		**COLABORADORES**	**MÉTODOS**
Consulta de Visitas		Administrador	getVisitas
Imprimir Reporte		Administrador	printVisitas

Tabla 4-4. Tarjeta CRC – Visitas (Marcillo D, 2018)

3.4.2.3. PROTOTIPOS



Visitas

#	Empleado	Cliente	Fecha	Inicio	Fin	Observación	Imagen	Ubicación
1	Maria Marcillo	Almacen La Luz	02-12-2018	22:08:15	22:11:19	Ok		Ver en mapa
2	Maria Marcillo	Ferrisariato	02-12-2018	22:08:15	22:10:20	Necesita copia factura		Ver en mapa
3	Maria Marcillo	Ferreteria Las Palmas	02-12-2018	22:06:35	22:06:55	Sin novedades		Ver en mapa

Figura 4-2. Prototipo 1 – WEB – Visitas (Marcillo D, 2018)



Figura 4-3. Prototipo 2 – WEB – Visitas (Marcillo D, 2018)

3.4.3. CODIFICACIÓN

3.4.3.1. ESTÁNDARES DE CODIFICACIÓN

- **Variables**: todas las variables empleadas serán nombradas de acuerdo a su valor o característica. La primera letra de cada una de las palabras será mayúscula, con excepción de que la primera letra de la primera palabra es minúscula. Ejemplos: fechaInicio, empleadoID.
- **Funciones**: las funciones vendrán a describir su respectiva tarea u operación. La primera palabra será en minúsculas y las otras contarán únicamente con la primera letra mayúscula. Ejemplo: buscarVisitas().
- **Bases**: las bases serán nombradas como dbRef, más un guión bajo y el nombre de la clase en minúsculas. Ejemplo: dbRef_visitas.
- **Botones**: los botones llevarán los caracteres 'btn' al inicio, más el nombre del botón con la primera letra en mayúscula. Ejemplo: btnEditar, btnNuevo.
- **Divisiones**: las divisiones tendrán al inicio los caracteres 'div', más el nombre de la división con la primera letra en mayúscula. Ejemplo: divMapa.

3.4.3.2. CÓDIGO

CLASE

La clase de Visita cuenta con 12 campos: empleadoID, clienteID, fecha, hora inicio, hora fin, observación, imagen, coordenadas inicio, coordenadas fin.

CONTROLADOR

Cuenta con 12 métodos que permitirán realizar las distintas acciones que se han enviadas desde la aplicación. Estos son:

- mostrarDatos (): obtiene la información de todas las visitas realizadas por los empleados y coloca la misma dentro de la tabla para que puedan ser visualizadas por el usuario.

- buscarVisitas (): busca todas las visitas del empleado seleccionado, en las fechas escogidas.
- verUbicacionMapa (): utiliza las ubicaciones de la visita y las coloca dentro de un mapa para que puedan ser visualizadas.
- cerrarMapa (): oculta el mapa.
- logOut (): permite cerrar la sesión del usuario actual.

3.4.4. PRUEBAS

3.4.4.1. PRUEBA DE ACEPTACIÓN

PRUEBA DE ACEPTACIÓN	
Nombre:	Consulta de visitas
Código:	1
Descripción:	Se debe visualizar toda la información de todas las visitas realizadas por los empleados.
Condiciones de ejecución:	Los datos deben estar registrados en la base de datos.
Entradas:	Ninguna
Pasos de ejecución:	1. Abrir el menú de Administración de Vehículos. 2. Verificamos la información desplegada en la tabla.
Resultado esperado:	Se mostrará una lista con la información de todas las visitas registradas.
Evaluación de la prueba:	La prueba realiza finalizó con un éxito al 100% puesto a que se muestra la información de todas las visitas de los empleados, registradas en la base de datos.

Tabla 4-5. Prueba de Aceptación (Marcillo D, 2018)

3.4.4.2. PRUEBA DE CAJA BLANCA

La prueba de caja blanca que se realizó durante esta iteración permitió verificar que el código ejecutará la funciones y métodos de manera adecuada. La prueba fue realizada en el transcurso de la implementación de la iteración, mientras se comprobaba que no existiera ningún problema dentro del código, así como los métodos para consultar las visitas. Además, se comprobó que la base de datos trabajaba adecuadamente con el sistema.

3.4.4.3. PRUEBA DE CAJA NEGRA

ENTRADAS	RESULTADOS ESPERADOS	RESULTADO
Despliegue de pantalla	La pantalla se muestre correctamente, así como todos sus componentes.	OK ✓
Consultar visitas	Se muestra la información de todas las visitas.	OK ✓

Tabla 4-6. Prueba de Caja Negra (Marcillo D, 2018)

3.5. ITERACIÓN 5 – LOCALIZACIONES

3.5.1. PLANEACIÓN

3.5.1.1. HISTORIAS DE USUARIO

	Enunciado de la historia				Criterios de aceptación			
Identificador (ID) de la historia	**Rol**	**Característica / Funcionalidad**	**Razón / Resultado**	**Número (#) de escenario**	**Criterio de aceptación (Título)**	**Contexto**	**Evento**	**Resultado / Comportamiento esperado**
HU-L1	Como administrador	Necesito ver un mapa con la geolocalización de todos los vehículos.	Con la finalidad de conocer donde se encuentran en ese momento.	1	Existe una o más localizaciones.	En caso de que exista al menos una localización.	Cuando se despliegue el mapa.	A continuación, se mostrará en un mapa la geolocalización de todos los empleados y sus vehículos.
				2	No existen localizaciones	En caso de que no exista ninguna localización.	Cuando se despliegue el mapa.	El sistema mostrará un mapa vacío.

Tabla 5-1. Historia de Usuario 5 – Administrar Localizaciones (Marcillo D, 2018)

3.5.1.2. PLAN DE ITERACIÓN

PLAN DE ITERACIONES					
Número de Iteración	**Fecha Inicio**	**Fecha Fin**	**Número de Historia**	**Nombre de Historia**	**Responsable**
1	01-04-2018	15-04-2018	1	Administrar Empleados	María Gloria Marcillo
2	16-04-2018	05-05-2018	2	Administrar Vehículos	María Gloria Marcillo
3	08-05-2018	30-05-2018	3	Administrar Clientes	María Gloria Marcillo
4	01-06-2018	30-06-2018	4	Administrar Visitas	María Gloria Marcillo
5	01-07-2018	30-07-2018	5	Administrar Localizaciones	María Gloria Marcillo

Tabla 5-2. Plan de Iteración (Marcillo D, 2018)

3.5.1.3. CRITERIOS DE PRUEBA DE ACEPTACIÓN

VISIÓN DE LA PRUEBA			
Nombre	Información Localización	**Código**	CPA-L1
N.º Historia	5		
Iteración	5		
Función crítica	Si		
Descripción	Verifica que se pueda acceder a la información de las geolocalizaciones, y que las mismas puedan ser visualizadas dentro de un mapa.		
Categoría	Administración de Localizaciones		
Prerrequisito			

Tabla 5-3. Criterio de Prueba de Aceptación (Marcillo D, 2018)

3.5.2. DISEÑO

3.5.2.1. DISEÑO SIMPLE



Figura 5-1. Diseño Simple – WEB – Localizaciones (Marcillo D, 2018)

3.5.2.2. TARJETAS CRC

NÚMERO:	T05	**NOMBRE:**	Localización
DESCRIPCIÓN:		En esta clase se podrá controlar el ingreso de las geolocalizaciones de los vendedores, así como la consulta de las mismas.	
RESPONSABILIDADES		**COLABORADORES**	**MÉTODOS**
Consulta de Localizaciones		Administrador	getLocalizaciones

Tabla 5-4. Tarjeta CRC – Localización (Marcillo D, 2018)

3.5.2.3. PROTOTIPOS



Figura 5-2. Prototipo – WEB – Localizaciones (Marcillo D, 2018)

3.5.3. CODIFICACIÓN

3.5.3.1. ESTÁNDARES DE CODIFICACIÓN

- **Variables**: todas las variables empleadas serán nombradas de acuerdo a su valor o característica. La primera letra de cada una de las palabras será mayúscula, con excepción de que la primera letra de la primera palabra es minúscula. Ejemplos: returnArrLocalizaciones, latitud, longitud.

- **Funciones**: las funciones vendrán a describir su respectiva tarea u operación. La primera palabra será en minúsculas y las otras contarán únicamente con la primera letra mayúscula. Ejemplo: actualizarMapa().
- **Bases**: las bases serán nombradas como dbRef, más un guión bajo y el nombre de la clase en minúsculas. Ejemplo: dbRef_localizaciones.
- **Divisiones**: las divisiones tendrán al inicio los caracteres 'div', más el nombre de la división con la primera letra en mayúscula. Ejemplo: divMapa.

3.5.3.2. CÓDIGO

CLASE

La clase de Localización cuenta con 4 campos: latitud, longitud, batería y tiempo.

CONTROLADOR

- obtenerDatos (): obtiene las últimas localizaciones de los vehículos desde la base de datos. Además, coloca toda la información dentro de una tabla para que el usuario pueda elegir a que vehículo quiere rastrear.
- localizarEnMapa (): ubica dentro del mapa un marcador con la ubicación del vehículo.
- actualizarMarker (): actualiza el marcador de cada localización para poder visualizar la ruta que está tomando el empleado. Esta actualización sucede cada vez que se agrega una nueva ubicación del empleado, cada 5 segundos.
- verTodosMapa(): permite mostrar todos los vehículos dentro del mapa.
- logOut (): permite cerrar la sesión del usuario actual.

VISTA

La pantalla contiene una tabla con la información de cada vehículo activo, así como un botón para poder seguir su rastro. En la parte inferior existe un mapa donde se puede visualizar la ruta que mantiene el empleado. Y un menú en la parte superior para desplazarse entre las otras pantallas de la aplicación.

3.5.4. PRUEBAS

3.5.4.1. PRUEBA DE ACEPTACIÓN

PRUEBA DE ACEPTACIÓN	
Nombre:	Despliega ubicación de vendedores en tiempo real
Código:	1
Descripción:	Se debe visualizar la ubicación en tiempo real de los vendedores dentro de un mapa, tanto en la aplicación web, como en la aplicación móvil.
Condiciones de ejecución:	Las localizaciones de cada vendedor deben estar registrados en la base de datos.
Entradas:	Ninguna
Pasos de ejecución:	1. Abrir el menú de Administración de Localizaciones. 2. Visualizar las ubicaciones.
Resultado esperado:	Se podrá observar la ubicación en tiempo real de cada vendedor, dentro de un mapa.
Evaluación de la prueba:	La prueba realiza finalizó con un éxito al 100% debido a que se muestran todas las ubicaciones de los vendedores dentro del mapa.

Tabla 5-5. Prueba de Aceptación (Marcillo D, 2018)

3.5.4.2. PRUEBA DE CAJA BLANCA

La prueba de caja blanca realizada en esta iteración permitió verificar que el código funcionase de manera correcta. La prueba fue realizada durante el transcurso de la implementación de la iteración, y se verificó que no existiesen ningún tipo de problemas dentro de código, así como que los métodos para obtener las ubicaciones funcionasen de manera adecuada; y además que el sistema interactuará efectivamente con la base de datos.

3.5.4.3. PRUEBA DE CAJA NEGRA

ENTRADAS	RESULTADOS ESPERADOS	RESULTADO
Despliegue de pantalla	La pantalla se muestre correctamente, así como todos sus componentes.	OK ✓
Inicio de Sesión (administrador y empleado)	Los usuarios pueden iniciar sesión adecuadamente.	OK ✓
Envió de ubicación de manera automática	Los dispositivos móviles de los empleados deben de enviar la ubicación de manera automática una vez iniciado el servicio.	OK ✓
Visualización de ubicaciones	Los administradores pueden visualizar la ubicación de los empleados en un mapa.	OK ✓

Tabla 5-6. Prueba de Caja Negra (Marcillo D, 2018)

CAPÍTULO 4. ITERACIONES ANDROID

4.1. ITERACIÓN 1 - EMPLEADOS

4.1.1. PLANEACIÓN

4.1.1.1. HISTORIAS DE USUARIO

	Enunciado de la historia				Criterios de aceptación			
Identificador (ID) de la historia	**Rol**	**Característica / Funcionalidad**	**Razón / Resultado**	**Número (#) de escenario**	**Criterio de aceptación (Título)**	**Contexto**	**Evento**	**Resultado / Comportamiento esperado**
HU-EA1	Como empleado	Necesita acceder a la aplicación.	Con la finalidad de conocer que usuarios están utilizando la misma.	1	Credenciales correctas	En caso de que las credenciales sean correctas.	Cuando la aplicación le solicite al empleado iniciar sesión.	A continuación, el empleado podrá iniciar sesión y utilizar las demás funciones de la aplicación.
				2	Credenciales Incorrectas	En caso de que el correo electrónico o la contraseña sean incorrectas.	Cuando la aplicación le solicite al empleado iniciar sesión.	A continuación, el sistema le pedirá al usuario volver a ingresar sus credenciales.

Tabla 6-1. Historia de Usuario 6 – Administrar Empleados (Marcillo D, 2018)

4.1.1.2. PLAN DE ITERACIÓN

PLAN DE ITERACIONES					
Número de Iteración	**Fecha Inicio**	**Fecha Fin**	**Número de Historia**	**Nombre de Historia**	**Responsable**
1	01-08-2018	15-08-2018	1	Administrar Empleados	María Gloria Marcillo
2	16-08-2018	05-09-2018	2	Administrar Vehículos	María Gloria Marcillo
3	10-09-2018	30-09-2018	3	Administrar Clientes	María Gloria Marcillo
4	01-10-2018	30-10-2018	4	Administrar Visitas	María Gloria Marcillo
5	04-11-2018	30-11-2018	5	Administrar Localizaciones	María Gloria Marcillo

Tabla 6-2. Plan de Iteraciones – Empleados (Marcillo D, 2018)

4.1.1.3. CRITERIOS DE PRUEBA DE ACEPTACIÓN

VISIÓN DE LA PRUEBA			
Nombre	Inicia Sesión	**Código**	CPA-EA1
N.º Historia	1		
Iteración	1		
Función crítica	Si		
Descripción	Verifica que el empleado pueda iniciar sesión en la aplicación móvil.		
Categoría	Administración de Empleados		
Prerrequisito			

Tabla 6-3. Criterio de Prueba de Aceptación (Marcillo D, 2018)

4.1.2. DISEÑO

4.1.2.1. DISEÑO SIMPLE

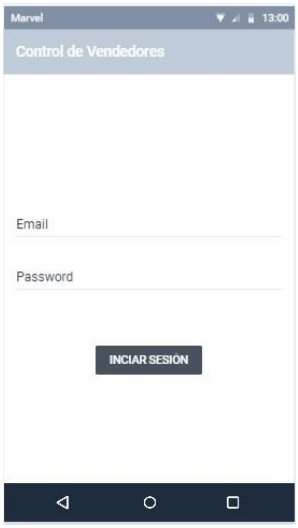


Figura 6-1. Diseño Simple – Android – Empleados (Marcillo D, 2018)

4.1.2.2. TARJETAS CRC

NÚMERO:	T06	**NOMBRE:**	Empleado
DESCRIPCIÓN:		En esta clase se podrá controlar el inicio de sesión del empleado.	
RESPONSABILIDADES		**COLABORADORES**	**MÉTODOS**
Autenticación de Empleados		Administrador	authenticate

Tabla 6-4. Tarjeta CRC – Empleados (Marcillo D, 2018)

4.1.2.3. PROTOTIPOS

Esta pantalla permite el inicio de sesión de todos los usuarios. Tanto como para vendedores como para los administradores. Únicamente se requiere completar las credenciales correctamente para iniciar sesión.

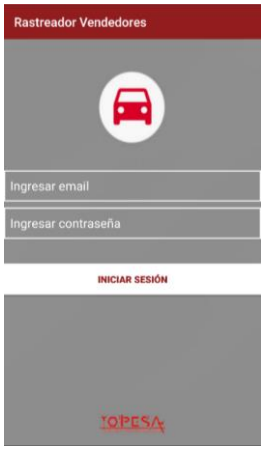


Figura 6-2. Prototipo – Android – Empleados (Marcillo D, 2018)

4.1.3. CODIFICACIÓN

4.1.3.1. ESTÁNDARES DE CODIFICACIÓN

- **Variables**: las variables serán nombradas como su respectiva característica. La primera letra de cada una de las palabras empleadas será mayúscula, con excepción de que la primera letra de la primera palabra es minúscula. Ejemplo: nombre, empleadoID.
- **Métodos**: los métodos utilizados describirán su respectiva tarea o actividad, la primera palabra será en minúsculas y las otras contarán únicamente con la primera letra mayúscula. Ejemplo: cargarDatos().
- **Botones**: los botones llevarán los caracteres 'btn' al inicio, más el nombre del botón con la primera letra en mayúscula. Ejemplo: btnIngresar.

- **Cuadros de Texto**: los cuadros de texto tendrán un 'txt' al inicio, seguido del nombre que lo identifique con la primera letra en mayúscula. Ejemplo: txtUsuario.

4.1.3.2. CÓDIGO

CLASE

La clase de empleado consta de 5 campos: cédula, nombre, apellido, teléfono y correo electrónico o email. Todos los campos son de tipo string o cadena.

CONTROLADOR

En el caso del empleado, la aplicación requiere únicamente las credenciales de este para poder acceder a la misma. A partir de estas se podrán obtener todos los datos del empleado para su uso posterior. El controlador cuenta con los siguientes métodos:

- authenticate (): requiere del email y la contraseña. Este método permite al sistema verificar que las credenciales sean las correctas después de comprobar con la información en la base de datos. Si la respuesta es positiva, el sistema permitirá al usuario acceder a todas las funciones de la aplicación; caso contrario, enviará un mensaje alertando que las credenciales son incorrectas.
- onStart () y onStop (): son métodos propios de la autenticación de Firebase.
- onBackPressed (): restringe al usuario de salir de la aplicación no sin antes presionar el botón de ATRÁS dos veces.

4.1.4. PRUEBAS

4.1.4.1. PRUEBA DE ACEPTACIÓN

PRUEBA DE ACEPTACIÓN	
Nombre:	Autenticación Empleados
Código:	6
Descripción:	El sistema debe verificar las credenciales ingresadas por el empleado y autenticarlas.
Condiciones de ejecución:	El empleado debe contar con un usuario y contraseña.
Entradas:	Correo electrónico y contraseña.
Pasos de ejecución:	1. El empleado inicia la aplicación. 2. Ingresa las credenciales. 3. El sistema autentifica las mismas. 4. Permite iniciar sesión.
Resultado esperado:	El empleado puede iniciar sesión en la aplicación.
Evaluación de la prueba:	La prueba realizada finalizó con un éxito al 100% debido a que el empleado pudo iniciar sesión adecuadamente.

Tabla 6-5. Prueba de Aceptación (Marcillo D, 2018)

4.1.4.2. PRUEBA DE CAJA BLANCA

Las pruebas de caja blanca realizadas durante esta iteración permitieron comprobar que tanto el código, como la estructura funcionase de manera adecuada. Esta prueba fue realizada durante la implementación de la iteración, al verificar que no existiesen ningún tipo de problemas en el código, que el método de autenticación trabajase efectivamente, y que el sistema trabaje adecuadamente con el sistema de autenticación de Firebase.

4.1.4.3. PRUEBA DE CAJA NEGRA

ENTRADAS	RESULTADOS ESPERADOS	RESULTADO
Despliegue de pantalla	La pantalla se muestre correctamente, así como todos sus componentes.	OK ✓
Autenticación de empleado	Los empleados pueden iniciar sesión correctamente.	OK ✓

Tabla 6-6. Prueba de Caja Negra (Marcillo D, 2018)

4.2. ITERACIÓN 2 – VEHÍCULOS

4.2.1. PLANEACIÓN

4.2.1.1. HISTORIAS DE USUARIO

	Enunciado de la historia				Criterios de aceptación			
Identificador (ID) de la historia	**Rol**	**Característica / Funcionalidad**	**Razón / Resultado**	**Número (#) de escenario**	**Criterio de aceptación (Título)**	**Contexto**	**Evento**	**Resultado / Comportamiento esperado**
HU-V4	Como un administrador	Necesito ver un listado de todos los vehículos del usuario.	Con la finalidad de seleccionar el mismo en el que el empleado se desplazará durante su trayecto.	1	Existen vehículos.	En caso de que existan vehículos registrados para el empleado.	Cuando se despliegue la pantalla principal.	A continuación, se mostrará un listado con las placas de los vehículos.
				2	No existen vehículos.	En caso de que no existan vehículos registrados para el empleado.	Cuando se despliegue la pantalla principal.	A continuación, se presentará un listado vacío.

Tabla 7-1. Historia de Usuario 7 – Administrar Vehículos (Marcillo D, 2018)

4.2.1.2. PLAN DE ITERACIÓN

PLAN DE ITERACIONES					
Número de Iteración	**Fecha Inicio**	**Fecha Fin**	**Número de Historia**	**Nombre de Historia**	**Responsable**
1	01-08-2018	15-08-2018	1	Administrar Empleados	María Gloria Marcillo
2	16-08-2018	05-09-2018	2	Administrar Vehículos	María Gloria Marcillo
3	10-09-2018	30-09-2018	3	Administrar Clientes	María Gloria Marcillo
4	01-10-2018	30-10-2018	4	Administrar Visitas	María Gloria Marcillo
5	04-11-2018	30-11-2018	5	Administrar Localizaciones	María Gloria Marcillo

Tabla 7-2. Plan de Iteración (Marcillo D, 2018)

4.2.1.3. CRITERIOS DE PRUEBA DE ACEPTACIÓN

VISIÓN DE LA PRUEBA			
Nombre	Información Vehículo	**Código**	CPA-VA2
N.º Historia	2		
Iteración	2		
Función crítica	Si		
Descripción	Despliega los vehículos asignados al empleado.		
Categoría	Administración de Vehículos		
Prerrequisito			

Tabla 7-3. Criterio de Prueba de Aceptación (Marcillo D, 2018)

4.2.2. DISEÑO

4.2.2.1. DISEÑO SIMPLE

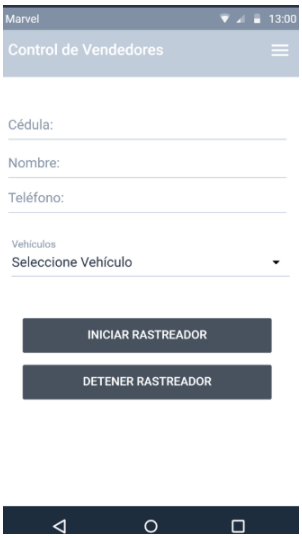


Figura 7-1. Diseño Simple – Android – Vehículos (Marcillo D, 2018)

4.2.2.2. TARJETAS CRC

NÚMERO:	T07	**NOMBRE:**	Vehículo
DESCRIPCIÓN:		En esta clase se podrá obtener la información de los vehículos que pertenecen al empleado.	
RESPONSABILIDADES		**COLABORADORES**	**MÉTODOS**
Consulta de Vehículos		Empleado	getVehiculos
Muestra Vehículos		Empleado	showData

Tabla 7-4. Tarjeta CRC – Vehículos (Marcillo D, 2018)

4.2.2.3. PROTOTIPOS

Antes de que el servicio de rastreador pueda ser inicializado, es necesario seleccionar el vehículo que el empleado utilizará para movilizare. Es posible escogerlo a partir de un combo box como el que se muestra a continuación.



Figura 7-2. Prototipo – Android – Vehículos (Marcillo D, 2018)

4.2.3. CODIFICACIÓN

4.2.3.1. ESTÁNDARES DE CODIFICACIÓN

- **Variables**: las variables serán nombradas como su respectiva característica. La primera letra de cada una de las palabras empleadas será mayúscula, con excepción de que la primera letra de la primera palabra es minúscula. Ejemplo: vehiculoID.
- **Métodos**: los métodos utilizados describirán su respectiva tarea o actividad, la primera palabra será en minúsculas y las otras contarán únicamente con la primera letra mayúscula. Ejemplo: cargarVehiculos().
- **Spinner:** el spinner o listado despegable se nombrará como 'spn' más el nombre del mismo, con la primera letra en mayúscula. Ejemplo: spnVehiculos.

4.2.3.2. CÓDIGO

CLASE

La clase de Vehículo cuenta con 7 campos: identificador, empleadoID, placa, marca, modelo, color y año. Todos los campos son de tipo string o cadena.

CONTROLADOR

Los vehículos que se encuentran almacenados en la base de datos pueden ser visualizados en la aplicación móvil dentro de un spinner o una lista. Esto permitirá controlar a los administradores los vehículos que se encuentran en marcha. Los mismos serán disponibles únicamente para los empleados a los que han sido asignados. Para esto se utiliza el siguiente método:

- showData (): obtiene el identificador del empleado asignado y prosigue a obtener los vehículos asignados al mismo y extraer las placas de cada vehículo, para colocarlas dentro de la lista.

VISTA

Dentro de la pantalla principal, posterior al inicio de sesión, se puede observar un spinner o lista que contiene las placas de los vehículos que el usuario tiene a su disposición.

4.2.4. PRUEBAS

4.2.4.1. PRUEBA DE ACEPTACIÓN

PRUEBA DE ACEPTACIÓN	
Nombre:	Información Vehículos
Código:	7
Descripción:	Se debe visualizar todos los vehículos que se encuentran asignados al usuario.
Condiciones de ejecución:	Los datos deben estar registrados en la base de datos.
Entradas:	Ninguna
Pasos de ejecución:	1. Desplegar la pantalla principal. 2. Obtener los vehículos pertenecientes al usuario. 3. Desplegar los vehículos.
Resultado esperado:	Dentro de la lista se debe de visualizar las placas de los vehículos.
Evaluación de la prueba:	La prueba realiza finalizó con un éxito al 100% debido a que se muestra la información de los vehículos del empleado registrados en la base de datos.

Tabla 7-5. Prueba de Aceptación (Marcillo D, 2018)

4.2.4.2. PRUEBA DE CAJA BLANCA

La prueba de caja blanca realizada durante esta iteración permitió comprobar que tanto el código, como la estructura funcionase de manera adecuada. Esta prueba fue realizada durante la implementación de la iteración, al verificar que no existiesen ningún tipo de problemas en el código, que todos los métodos: de ingreso, modificación, eliminación y consulta de los vehículos; trabajen efectiva y conjuntamente; y que el sistema interactúe adecuadamente con la base de datos.

4.2.4.3. PRUEBA DE CAJA NEGRA

ENTRADAS	RESULTADOS ESPERADOS	RESULTADO
Despliegue de pantalla	La pantalla se muestre correctamente, así como todos sus componentes.	OK ✓
Ingreso de vehículo	Los vehículos son ingresados a la base de datos correctamente.	OK ✓
Modificación de vehículo	La información de los vehículos puede ser modificada sin ningún problema.	OK ✓
Eliminación de vehículo	Los vehículos pueden ser eliminados de la base de datos.	OK ✓
Consulta de vehículos	La información de todos los vehículos es presentada al usuario.	OK ✓

Tabla 7-6. Prueba de Caja Negra (Marcillo D, 2018)

4.3. ITERACIÓN 3 – CLIENTES

4.3.1. PLANEACIÓN

4.3.1.1. HISTORIAS DE USUARIO

	Enunciado de la historia				Criterios de aceptación			
Identificador (ID) de la historia	**Rol**	**Característica / Funcionalidad**	**Razón / Resultado**	**Número (#) de escenario**	**Criterio de aceptación (Título)**	**Contexto**	**Evento**	**Resultado / Comportamiento esperado**
HU-C1	Como un administrador	Necesito ingresar la información de los clientes de la empresa	Con la finalidad de agregarlo a la base de datos de la empresa y poder usar la información en el programa.	1	Datos completos	En caso de que ingrese los datos completos del cliente.	Cuando se desplieguen los campos para el ingreso.	A continuación, se mostrará un mensaje indicando que el cliente ha sido almacenado correctamente en la base.
				2	Datos incompletos	En caso de que falte información del cliente.	Cuando se desplieguen los campos para el ingreso.	A continuación, se mostrará un mensaje solicitando que se complete todos los datos del cliente.
HU-C2	Como un administrador	Necesito ver un listado de todos los clientes registrados.	Con la finalidad de conocer la información de cada uno.	1	Existen clientes.	En caso de que existan clientes registrados.	Cuando se despliegue un listado de los clientes.	A continuación, se mostrará un listado con la información de todos los clientes registrados.

Tabla 8-1. Historia de Usuario 8 – Administrar Clientes (Marcillo D, 2018)

4.3.1.2. PLAN DE ITERACIÓN

PLAN DE ITERACIONES					
Número de Iteración	**Fecha Inicio**	**Fecha Fin**	**Número de Historia**	**Nombre de Historia**	**Responsable**
1	01-08-2018	15-08-2018	1	Administrar Empleados	María Gloria Marcillo
2	16-08-2018	05-09-2018	2	Administrar Vehículos	María Gloria Marcillo
3	10-09-2018	30-09-2018	3	Administrar Clientes	María Gloria Marcillo
4	01-10-2018	30-10-2018	4	Administrar Visitas	María Gloria Marcillo
5	04-11-2018	30-11-2018	5	Administrar Localizaciones	María Gloria Marcillo

Tabla 8-2. Plan de Iteración (Marcillo D, 2018)

4.3.1.3. CRITERIOS DE PRUEBA DE ACEPTACIÓN

VISIÓN DE LA PRUEBA			
Nombre	Manipulación datos Cliente	**Código**	CPA-CA1
N.º Historia	8		
Iteración	3		
Función crítica	Si		
Descripción	Verifica que se pueda ingresar los registros de los clientes en la base de datos.		
Categoría	Administración de Clientes		

Tabla 8-3. Criterio de Prueba de Aceptación (Marcillo D, 2018)

4.3.2. DISEÑO

4.3.2.1. DISEÑO SIMPLE

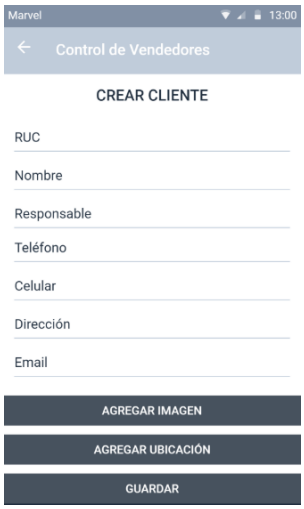


Figura 8-1. Diseño Simple – Android – Clientes (Marcillo D, 2018)

4.3.2.2. TARJETAS CRC

NÚMERO:	T03	**NOMBRE:**	Cliente
DESCRIPCIÓN:		En esta clase se podrá controlar el ingreso y consulta de los datos de los clientes.	
RESPONSABILIDADES		**COLABORADORES**	**MÉTODOS**
Ingreso de Clientes		Empleado	insertCliente
Consulta de Clientes		Empleado	getClientes

Tabla 8-4. Tarjeta CRC – Clientes (Marcillo D, 2018)

4.3.2.3. PROTOTIPOS

En la parte superior de la pantalla de VISITAS se puede observar un pequeño menú, que contará con la opción de AGREGAR CLIENTE, en el caso de que el empleado desee agregar un nuevo cliente.



Figura 8-2. Prototipo 1 – Android – Clientes (Marcillo D, 2018)

Después de seleccionar la opción de AGREGAR CLIENTE, el sistema despliega una nueva pantalla, donde el usuario necesitará completar todos los datos mostrados para proceder a registrar el cliente dentro de la base de datos.



Figura 8-3. Prototipo 2 – Android – Clientes (Marcillo D, 2018)



Figura 8-4. Prototipo 3 – Android – Clientes (Marcillo D, 2018)

4.3.3. CODIFICACIÓN

4.3.3.1. ESTÁNDARES DE CODIFICACIÓN

- **Variables**: las variables serán nombradas como su respectiva característica. La primera letra de cada una de las palabras empleadas será mayúscula, con excepción de que la primera letra de la primera palabra es minúscula. Ejemplo: identificación, imagenURI.
- **Métodos**: los métodos utilizados describirán su respectiva tarea o actividad, la primera palabra será en minúsculas y las otras contarán únicamente con la primera letra mayúscula. Ejemplo: verificarCedula().
- **Botones**: los botones llevarán los caracteres 'btn' al inicio, más el nombre del botón con la primera letra en mayúscula. Ejemplo: btnGuardar.
- **Cuadros de Texto**: los cuadros de texto tendrán un 'txt' al inicio, seguido del nombre que lo identifique con la primera letra en mayúscula. Ejemplo: txtIdentificacion, txtNombre.
- **Imagen**: las imágenes serán denominadas con los caracteres 'img' al inicio, más el nombre de la misma. Ejemplo: imgCliente.

4.3.3.2. CÓDIGO

CLASE

La clase de Cliente cuenta con 10 campos: identificación, nombre, responsable, dirección, teléfono, celular, email, imagen, ubicación y clienteID.

CONTROLADOR

Dentro de la aplicación móvil es donde se puede ingresar la información de los nuevos clientes. Para eso se requiere de los siguientes métodos:

- abrirCamara (): permite al usuario utilizar la cámara de su dispositivo móvil para obtener la imagen del cliente.
- buscarArchivo (): si el usuario lo desea, también puede obtener una imagen que se encuentre en la galería del dispositivo; este método es aquel que permite que eso suceda.
- getLocation (): obtiene la ubicación del cliente y guarda las coordenadas de esta.
- verificarIdentificacion (): verifica que la identificación del cliente sea auténtica.
- verificarCedula (): verifica que la identificación sea una cédula válida.
- verificarRUCNatural (): verifica que la identificación ingresada sea un RUC de persona natural válido.
- verificarRUCPublico (): verifica que la identificación del cliente sea un RUC público válido.
- verificarRUCSociedades (): verifica que la identificación sea un RUC jurídico o de sociedades, válido.
- verificarDatos (): comprueba que el usuario haya llenado todos los campos y contenga todos los atributos del cliente para continuar.
- subirArchivo (): sube la imagen del cliente al almacenamiento de media de la base de datos.

- guardarCliente (): guarda la información del cliente dentro de la base de datos.
- showData (): muestra un listado de los clientes.

4.3.4. PRUEBAS

4.3.4.1. PRUEBA DE ACEPTACIÓN

PRUEBA DE ACEPTACIÓN	
Nombre:	Guardar Clientes
Código:	8
Descripción:	Debe ingresar la información del cliente de manera adecuada. Todos los datos de estos, así como la imagen y coordenadas de la ubicación.
Condiciones de ejecución:	Todos los datos deben estar completos.
Entradas:	Datos del cliente.
Pasos de ejecución:	1. Ingresar los datos del cliente. 2. Agregar una imagen. 3. Obtener la ubicación de este. 4. Verificar que todos los datos estén correctos. 5. Guardar la información en la base.
Resultado esperado:	Todos los datos del cliente son ingresados correctamente dentro de la base de datos.
Evaluación de la prueba:	La prueba realiza finalizó con un éxito al 100% debido a que se almaceno la información del cliente correctamente.

Tabla 8-5. Prueba de Aceptación (Marcillo D, 2018)

4.2.4.2. PRUEBA DE CAJA BLANCA

La prueba de caja blanca realizada durante esta iteración permitió comprobar que el código realiza todas las funciones de manera correcta. La prueba fue realizada durante el proceso de desarrollo, verificando que no existieran errores en el código y que los métodos cumplieran sus funciones.

4.2.4.3. PRUEBA DE CAJA NEGRA

ENTRADAS	RESULTADOS ESPERADOS	RESULTADO
Despliegue de pantalla	La pantalla se muestre correctamente, así como todos sus componentes.	OK ✓
Ingreso de cliente	Los clientes son ingresados dentro de la base de manera adecuada.	OK ✓
Consulta de clientes	La información de todos los clientes es presentada al usuario.	OK ✓

Tabla 8-6. Prueba de Caja Negra (Marcillo D, 2018)

4.4. ITERACIÓN 4 – VISITAS

4.4.1. PLANEACIÓN

4.4.1.1. HISTORIAS DE USUARIO

	Enunciado de la historia				Criterios de aceptación			
Identificador (ID) de la historia	**Rol**	**Característica / Funcionalidad**	**Razón / Resultado**	**Número (#) de escenario**	**Criterio de aceptación (Título)**	**Contexto**	**Evento**	**Resultado / Comportamiento esperado**
HU-Vi1	Como empleado	Necesito ingresar los datos de una visita.	Con la finalidad de indicar que estoy realizando mi trabajo.	1	Ingresar visita	En caso de que ingrese una visita.	Cuando se desplieguen los campos para el ingreso.	A continuación, se mostrará un mensaje indicando que la visita ha sido almacenada correctamente en la base.

Tabla 9-1. Historia de Usuario 9 – Administrar Visitas (Marcillo D, 2018)

4.4.1.2. PLAN DE ITERACIÓN

PLAN DE ITERACIONES					
Número de Iteración	**Fecha Inicio**	**Fecha Fin**	**Número de Historia**	**Nombre de Historia**	**Responsable**
1	01-08-2018	15-08-2018	1	Administrar Empleados	María Gloria Marcillo
2	16-08-2018	05-09-2018	2	Administrar Vehículos	María Gloria Marcillo
3	10-09-2018	30-09-2018	3	Administrar Clientes	María Gloria Marcillo
4	01-10-2018	30-10-2018	4	Administrar Visitas	María Gloria Marcillo
5	04-11-2018	30-11-2018	5	Administrar Localizaciones	María Gloria Marcillo

Tabla 9-2. Plan de Iteración (Marcillo D, 2018)

4.4.1.3. CRITERIOS DE PRUEBA DE ACEPTACIÓN

VISIÓN DE LA PRUEBA			
Nombre	Manipulación datos Visita	**Código**	CPA-ViA1
N.º Historia	9		
Iteración	4		
Función crítica	Si		
Descripción	Verifica que se pueda ingresar los registros de las visitas en la base de datos.		
Categoría	Administración de Visitas		

Tabla 9-3. Criterio de Prueba de Aceptación (Marcillo D, 2018)

4.4.2. DISEÑO

4.4.2.1. DISEÑO SIMPLE

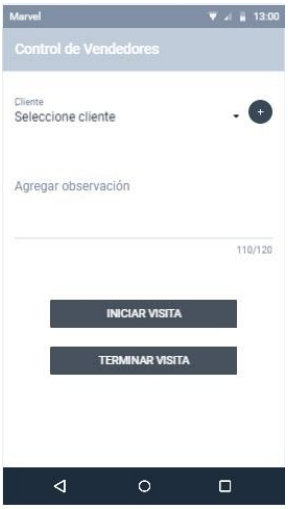


Figura 9-1. Diseño Simple 1 – Android – Visitas (Marcillo D, 2018)

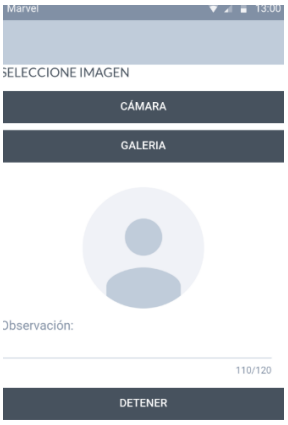


Figura 9-2. Diseño Simple 2 – Android – Visitas (Marcillo D, 2018)

4.4.2.2. TARJETAS CRC

NÚMERO:	T09	**NOMBRE:**	Visita
DESCRIPCIÓN:		En esta clase se podrá controlar el ingreso de los datos de las visitas.	
RESPONSABILIDADES		**COLABORADORES**	**MÉTODOS**
Ingreso de Visitas		Empleado	insertVisita

Tabla 9-4. Tarjeta CRC – Visitas (Marcillo D, 2018)

4.4.2.3. PROTOTIPOS

Al inicializar la pantalla de VISITA, se pueden observar: un combo box que enlista todos los clientes de la empresa, así como un botón INICIAR que indicará el momento y la ubicación exacta del vendedor cuando comience a realizar la visita.

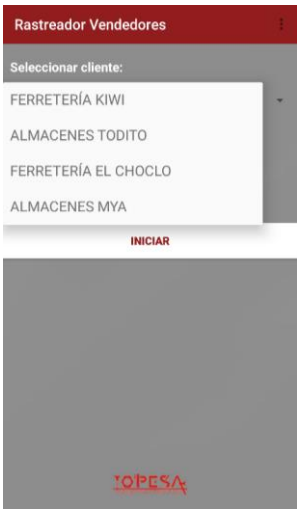


Figura 9-3. Prototipo 1 – Android – Empleado (Marcillo D, 2018)

Para poder iniciar una visita, el usuario necesita seleccionar un cliente, o bien agregar uno nuevo si no existe, y proseguir a seleccionar INICIAR. Una vez iniciada la visita, el usuario podrá ingresar una observación si así lo requiere. Cuando la visita se dé por terminada, el usuario necesita presionar el botón DETENER. Automáticamente, el sistema almacenará la hora y ubicación del usuario en ese momento, como forma de controlar el tiempo que tardó realizando la visita, y que se haya mantenido alrededor del mismo lugar.

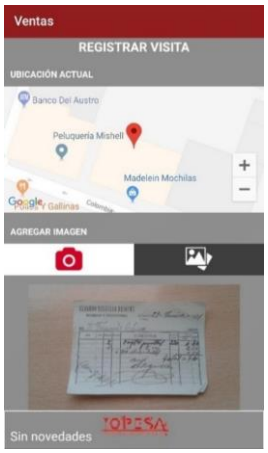


Figura 9-4. Prototipo 2 – Android – Empleado (Marcillo D, 2018)

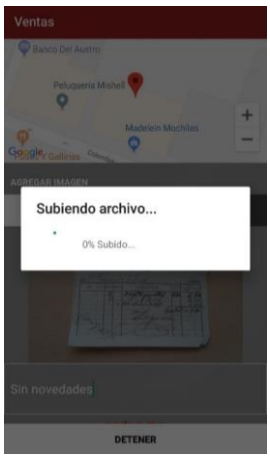


Figura 9-5. Prototipo 3 – Android – Empleado (Marcillo D, 2018)

4.4.3. CODIFICACIÓN

4.4.3.1. ESTÁNDARES DE CODIFICACIÓN

- **Variables**: las variables serán nombradas como su respectiva característica. La primera letra de cada una de las palabras empleadas será mayúscula, con excepción de que la primera letra de la primera palabra es minúscula. Ejemplo: observación, latitudInicio.
- **Métodos**: los métodos utilizados describirán su respectiva tarea o actividad, la primera palabra será en minúsculas y las otras contarán únicamente con la primera letra mayúscula. Ejemplo: subirArchivo().
- **Botones**: los botones llevarán los caracteres 'btn' al inicio, más el nombre del botón con la primera letra en mayúscula. Ejemplo: btnCamara.
- **Cuadros de Texto**: los cuadros de texto tendrán un 'txt' al inicio, seguido del nombre que lo identifique con la primera letra en mayúscula. Ejemplo: txtObservacion.
- **Imagen**: las imágenes serán denominadas con los caracteres 'img' al inicio, más el nombre de la misma. Ejemplo: imgVisita.

4.4.3.2. CÓDIGO

CLASE

La clase de Visita cuenta con 12 campos: empleadoID, clienteID, fecha, hora inicio, hora fin, observación, imagen, coordenadas inicio, coordenadas fin.

CONTROLADOR

Las visitas de los empleados serán registradas a partir de la aplicación móvil. Para esto se requieren los siguientes métodos:

- iniciarVisita (): inicia el proceso de la visita, obteniendo así la hora de inicio de la misma y dando paso a la siguiente pantalla que contendrá más información de esta.

- getLocation (): obtiene la ubicación en tiempo real de donde se está llevando a cabo la visita.
- abrirCamara (): permite abrir la cámara propia del dispositivo para obtener imágenes.
- buscarArchivo (): permite buscar una imagen dentro de la galería del dispositivo y seleccionarla.
- subirArchivo (): sube la imagen al almacenamiento de media de la base de datos.
- guardarVisita (): guarda la información de la visita dentro de la base de datos.

4.4.4. PRUEBAS

4.4.4.1. PRUEBA DE ACEPTACIÓN

PRUEBA DE ACEPTACIÓN	
Nombre:	Registro de visitas
Código:	1
Descripción:	Los empleados deben poder registrar las visitas que realicen.
Condiciones de ejecución:	Los datos deben estar registrados en la base de datos.
Entradas:	Seleccionar cliente y agregar observación
Pasos de ejecución:	1. Cuando el servicio de rastreo este corriendo, el empleado podrá seleccionar "REALIZAR VISITA". 2. El empleado selecciona el cliente de la lista, o bien puede ingresar uno nuevo. 3. El usuario presiona el botón "INICIAR". 4. Posteriormente, ingresa los datos de la visita, y presiona el botón "DETENER" para así concluir con la visita y almacenar la información.
Resultado esperado:	La información respectiva de la visita se almacenará en la base de datos.
Evaluación de la prueba:	La prueba realiza finalizó con un éxito al 100% puesto a que la información de cada visita se almacenó de manera correcta.

Tabla 9-5. Prueba de Aceptación (Marcillo D, 2018)

4.4.4.2. PRUEBA DE CAJA BLANCA

La prueba de caja blanca que se realizó durante esta iteración permitió verificar que el código ejecutará la funciones y métodos de manera adecuada. La prueba fue realizada en el transcurso de la implementación de la iteración, mientras se comprobaba que no existiera ningún problema dentro del código, así como los métodos para ingresar las visitas. Además, se comprobó que la base de datos trabajaba adecuadamente con el sistema.

4.4.4.3. PRUEBA DE CAJA NEGRA

ENTRADAS	RESULTADOS ESPERADOS	RESULTADO
Despliegue de pantalla	La pantalla se muestre correctamente, así como todos sus componentes.	OK ✓
Iniciar visita	Los empleados pueden iniciar una visita después de seleccionar el cliente.	OK ✓
Terminar visita	Los empleados pueden dar por concluida una visita.	OK ✓

Tabla 9-6. Prueba de Caja Negra (Marcillo D, 2018)

4.5. ITERACIÓN 5 – LOCALIZACIONES

4.5.1. PLANEACIÓN

4.5.1.1. HISTORIAS DE USUARIO

	Enunciado de la historia				Criterios de aceptación			
Identificador (ID) de la historia	**Rol**	**Característica / Funcionalidad**	**Razón / Resultado**	**Número (#) de escenario**	**Criterio de aceptación (Título)**	**Contexto**	**Evento**	**Resultado / Comportamiento esperado**
HU-L1	Como empleado	Necesito ingresar la información una nueva localización	Con la finalidad de agregarlo a la base de datos de la empresa y poder ubicarlo	1	GPS encendido	En caso de que tenga el GPS encendido	Cuando se abra la aplicación móvil.	A continuación, se enviarán las coordenadas a la base de manera automática después de un periodo de tiempo determinado.
				2	GPS apagado	En caso de que el GPS se encuentre apagado	Cuando se abra la aplicación móvil.	A continuación, se enviará una alerta indicado que se requiere encender el GPS del dispositivo.
HU-L2	Como administrador	Necesito ver un listado de la geolocalización de todos los vehículos.	Con la finalidad de conocer donde se encuentran en ese momento.	1	Existe una o más localizaciones.	En caso de que exista al menos una localización.	Cuando se despliegue el mapa.	A continuación, se mostrará en un mapa la geolocalización de todos los empleados y sus vehículos.
				2	No existen localizaciones	En caso de que no exista ninguna localización.	Cuando se despliegue el mapa.	El sistema mostrará un mapa vacío.

Tabla 10-1. Historia de Usuario 10 – Administrar Localizaciones (Marcillo D, 2018)

4.5.1.2. PLAN DE ITERACIÓN

PLAN DE ITERACIONES					
Número de Iteración	**Fecha Inicio**	**Fecha Fin**	**Número de Historia**	**Nombre de Historia**	**Responsable**
1	01-08-2018	15-08-2018	1	Administrar Empleados	María Gloria Marcillo
2	16-08-2018	05-09-2018	2	Administrar Vehículos	María Gloria Marcillo
3	10-09-2018	30-09-2018	3	Administrar Clientes	María Gloria Marcillo
4	01-10-2018	30-10-2018	4	Administrar Visitas	María Gloria Marcillo
5	04-11-2018	30-11-2018	5	Administrar Localizaciones	María Gloria Marcillo

Tabla 10-2. Plan de Iteración (Marcillo D, 2018)

4.5.1.3. CRITERIOS DE PRUEBA DE ACEPTACIÓN

VISIÓN DE LA PRUEBA			
Nombre	Manipulación datos Localización	**Código**	CPA-LA1
N.º Historia	10		
Iteración	5		
Función crítica	Si		
Descripción	Verifica que se pueda ingresar la geolocalización de los vendedores en tiempo real.		
Categoría	Administración de Localizaciones		

Tabla 10-3. Criterio de Prueba de Aceptación (Marcillo D, 2018)

4.5.2. DISEÑO

4.5.2.1. DISEÑO SIMPLE

ADMINISTRADOR

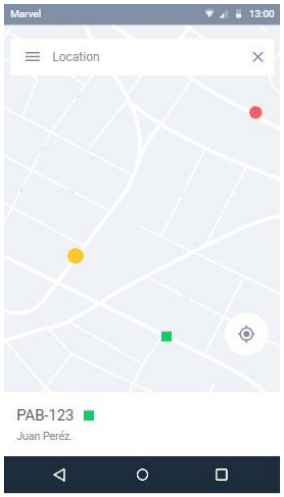


Figura 10-1. Diseño Simple – Android – Localizaciones (Marcillo D, 2018)

4.5.2.2. TARJETAS CRC

NÚMERO:	T05	**NOMBRE:**	Localización
DESCRIPCIÓN:		En esta clase se podrá controlar el ingreso de las geolocalizaciones de los vendedores, así como la consulta de las mismas.	
RESPONSABILIDADES		**COLABORADORES**	**MÉTODOS**
Ingreso de Localizaciones Consulta de Localizaciones		Empleado Administrador	insertLocalizacion getLocalizaciones

Tabla 10-4. Tarjeta CRC – Localización (Marcillo D, 2018)

4.5.2.3. PROTOTIPOS

ADMINISTRADOR

La aplicación móvil con la cual contaran los administradores, es decir, jefes, supervisores, gerentes y demás personas a cargo, requiere igualmente una autenticación del usuario. El administrador ingresa sus respectivas credenciales, y presiona el botón INICIAR SESIÓN.

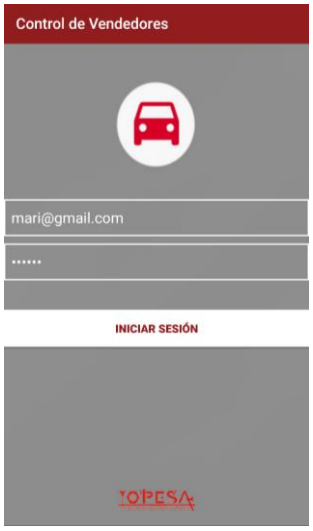


Figura 10-2. Prototipo 1 – Android – Localizaciones (Marcillo D, 2018)

Una vez inicializada la sesión, se despliega automáticamente un mapa que muestra: la posición actual del usuario (marcador de circulo azul), así como marcadores que indican la posición de los vendedores en tiempo real (marcadores con forma de automóvil).

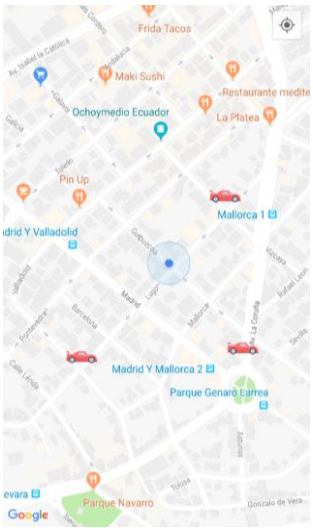


Figura 10-3. Prototipo 2 – Android – Localizaciones (Marcillo D, 2018)

Además, el usuario puede conocer la información de cada vehículo cuando lo selecciona. Se despliega brevemente un rotulo con el número de placa y el nombre del empleado.

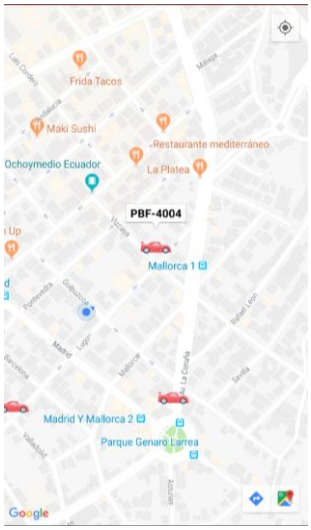


Figura 10-4. Prototipo 3 – Android – Localizaciones (Marcillo D, 2018)

EMPLEADO

En la siguiente pantalla, el sistema despliega la información básica del empleado usuario, así como dos botones que le permitirán al mismo inicializar el servicio de rastreo, y también detenerlo. Además, existe un combo box que despliega una lista con todos los vehículos que se encuentran registrados a nombre del empleado.

La pantalla a su vez cuenta con un botón CERRAR SESIÓN que permitirá al usuario cerrar su sesión únicamente después de haber detenido el servicio.



Figura 10-5. Prototipo 4 – Android – Localizaciones (Marcillo D, 2018)



Figura 10-6. Prototipo 5 – Android – Localizaciones (Marcillo D, 2018)

Una vez seleccionado el vehículo, se procede a inicializar el servicio de rastreo. Adicionalmente se crea una notificación que indica al usuario que su dispositivo móvil está siendo rastreado en ese momento. Así también, al ser un servicio, este continuará trabajando aun cuando el usuario cierre la aplicación para usar otras aplicaciones.

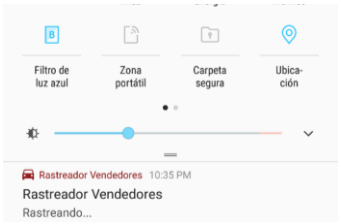


Figura 10-7. Prototipo 6 – Android – Localizaciones (Marcillo D, 2018)

El sistema automáticamente deshabilitará el combo con el fin de evitar cualquier dato erróneo, y a su vez desaparecerá el botón de INICIAR. También, mostrará el botón de REALIZAR VISITA, el mismo que permitirá al empleado registrar todas las visitas que se vayan realizando en el transcurso del día.



Figura 10-8. Prototipo 7 – Android – Localizaciones (Marcillo D, 2018)

4.5.3. CODIFICACIÓN

4.5.3.1. ESTÁNDARES DE CODIFICACIÓN

- **Variables**: las variables serán nombradas como su respectiva característica. La primera letra de cada una de las palabras empleadas será mayúscula, con excepción de que la primera letra de la primera palabra es minúscula. Ejemplo: latitud, vehiculoID.
- **Métodos**: los métodos utilizados describirán su respectiva tarea o actividad, la primera palabra será en minúsculas y las otras contarán únicamente con la primera letra mayúscula. Ejemplo: addMarker().
- **Mapa:** el mapa utilizado será denominado únicamente como map.

4.5.3.2. CÓDIGO

CLASE

La clase de Localización cuenta con 4 campos: latitud, longitud, batería y tiempo.

CONTROLADOR

EMPLEADO

La ubicación de cada empleado es ingresada de manera automática por medio de un servicio en segundo plano, el mismo que cuenta con los siguientes métodos:

- onConnected (): cuando el servicio es conectado, verifica que la aplicación tenga los permisos necesarios para utilizar el GPS, en caso de que no los tenga envía una solicitud al usuario para activarlos.
- cargarEstadosPrevios (): obtiene los estados anteriores por los que ha pasado el vehículo.
- locationIsAtStatus (): verifica si la ubicación actual del empleado se encuentra relativamente en el mismo estado anterior, de este modo se

sabe si es necesario agregar una nueva localización o mantenerlas en estado estacionario.

- startTracking (): inicia el servicio de rastreo y solicitando actualizaciones de la ubicación
- onLocationChanged (): agrega una nueva localización en la base de datos cada vez que esta cambia.
- obtenerBateria (): obtiene el nivel de batería del dispositivo móvil para que el administrador pueda verificar que el mismo pueda mantenerse encendido.

ADMINISTRADOR

La aplicación del administrador le permite visualizar en tiempo real la posición de los vehículos. Los métodos son:

- onConnected (): cuando se conecta, se encarga de verificar que el dispositivo cuente con los permisos para acceder al GPS.
- onLocationChanged (): cada vez que la ubicación del administrador cambie, el método lo mostrará en el mapa.
- obtenerVehiculos (): obtiene las placas de todos los vehículos que se hallan registrados en la base de datos.
- referencia (): utiliza la placa para obtener la referencia que le permitirá obtener la ubicación.
- obtenerDatos (): obtiene la última ubicación del dispositivo del empleado.
- addMarker (): agrega la ubicación dentro del mapa y es visualizada por medio de un marcador.

VISTA

EMPLEADO

El dispositivo de los empleados no cuenta con ninguna pantalla puesto que el ingreso de datos se realiza por medio de un servicio en segundo plano.

ADMINISTRADOR

Después de ingresar sus credenciales, el administrador podrá observar un mapa con distintos marcadores en movimiento que representan el movimiento que los empleados están tomando, así como un marcador para mostrar la ubicación propia del administrador y determinar si se halla cerca o lejos de los empleados.

4.5.4. PRUEBAS

4.5.4.1. PRUEBA DE ACEPTACIÓN

PRUEBA DE ACEPTACIÓN	
Nombre:	Enviar ubicación
Código:	10
Descripción:	Los vendedores deben de tener la posibilidad de enviar su ubicación actual, de manera indirecta, por medio de su dispositivo móvil.
Condiciones de ejecución:	El vendedor debe de inicializar el servicio de rastreo.
Entradas:	Seleccionar un vehículo.
Pasos de ejecución:	1. Iniciar sesión. 2. Seleccionar un vehículo de la lista. 3. Presionar "INICIAR"
Resultado esperado:	La ubicación de cada vendedor será enviada automáticamente a la base de datos cada cierto intervalo de tiempo.
Evaluación de la prueba:	La prueba realiza finalizó con un éxito al 100% debido a que se pudo enviar la ubicación de todos los vendedores de manera automática.

Tabla 10-5. Prueba de Aceptación (Marcillo D, 2018)

4.5.4.2. PRUEBA DE CAJA BLANCA

La prueba de caja blanca realizada en esta iteración permitió verificar que el código funcionase de manera correcta. La prueba fue realizada durante el transcurso de la implementación de la iteración, y se verificó que no existiesen ningún tipo de problemas dentro de código, así como que los métodos para obtener y enviar las ubicaciones funcionasen de manera adecuada; y además que el sistema interactuará efectivamente con la base de datos.

4.5.4.3. PRUEBA DE CAJA NEGRA

ENTRADAS	RESULTADOS ESPERADOS	RESULTADO
Despliegue de pantalla	La pantalla se muestre correctamente, así como todos sus componentes.	OK ✓
Envió de ubicación de manera automática	Los dispositivos móviles de los empleados, deben de enviar la ubicación de manera automática una vez iniciado el servicio.	OK ✓
Visualización de ubicaciones	Los administradores pueden visualizar la ubicación de los empleados en un mapa.	OK ✓

Tabla 10-6. Prueba de Caja Negra (Marcillo D, 2018)

4.6. MODELO RELACIONAL

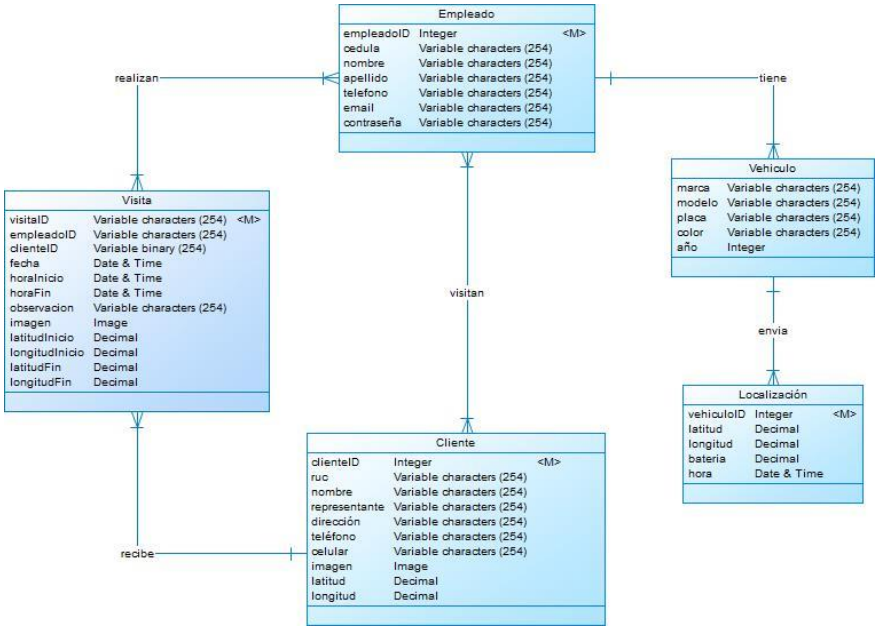


Figura 11-1. Modelo Relacional (Marcillo D, 2019)

CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Se logró desarrollar un sistema que permitiera a la empresa poder mantener un adecuado control de las actividades designadas a los vendedores de esta, de manera eficiente y adecuada, lo cual permitirá a la empresa mejorar la calidad de sus servicios.
- Por medio de una buena comunicación con el usuario, de acuerdo con la metodología aplicada, se puedo determinar correctamente todos los requerimientos que el usuario necesitaba.
- El desarrollo del sistema fue realizado acorde a la metodología seleccionada, lo que permitió que el mismo garantice un correcto funcionamiento.
- Se realizó una planificación adecuada para la realización del sistema y fue seguida de manera apropiada, lo que permitió reducir tiempos durante el desarrollo del sistema.
- Con el fin de brindar una interacción sencilla y rápida, fue desarrollada una interfaz amigable con el usuario que permitan al mismo manipular al sistema de manera intuitiva y sin problemas.
- Se desarrollaron prototipos de diseño de las distintas clases que se utilizaron para junto con el usuario determinar los elementos esenciales dentro del sistema.
- Se establecieron estándares de codificación con el fin de mantener un orden y claridad al código y así poder cumplir con las buenas prácticas dentro de la programación para obtener una buena calidad en el código.
- Las pruebas de aceptación permitieron evaluar el correcto funcionamiento de las diferentes actividades del sistema, con el fin de garantizar la calidad de estas.
- Las pruebas de caja blanca verificaron el funcionamiento del código, lo mismo que permitió corregir cualquier tipo de errores que se pudiesen presentar durante la realización de este.

- Las pruebas de caja negra permitieron comprobar que se cumpliesen correctamente con todos los requerimientos del usuario.
- Este sistema ofrece al usuario una manera fácil de registrar las visitas de los vendedores, la adición de nuevos clientes para la empresa, así como un control sobre el trabajo de los vendedores.
- El sistema les permitirá a los distintos usuarios hacer mucho más eficiente su trabajo, de una manera sencilla y rápida para poder garantizar una mejor calidad de este.

5.2. RECOMENDACIONES

- El sistema puede ser mejorado con el tiempo, y se recomienda la generación de un reporte de las visitas realizadas por los vendedores con el fin de mantener un mejor control sobre las tareas de estos.
- Mantener una buena comunicación con el cliente durante el desarrollo del sistema con el objetivo de obtener los requerimientos necesarios y precisos que el mismo solicite, y así realizar un buen trabajo.
- Realizar la planificación tomando en cuenta situaciones y tiempos reales, así como contemplando la dificultad e importancia de los requerimientos necesarios por realizar.
- Mantener siempre presentes los estándares de codificación con el fin de mantener una buena práctica y calidad del código.
- Diseñar interfaces que sean amigables con el usuario, y cuya utilización sea fácil, intuitiva y cómoda para el mismo.
- Realizar las respectivas pruebas al finalizar cada una de las iteraciones para poder verificar el correcto funcionamiento de cada uno, y si es necesario corregir cualquier tipo de error que se haya presentado.
- Seguir la metodología al pie de la letra para así poder conseguir resultados satisfactorios y de calidad que otorguen al usuario una herramienta efectiva para la realización de su trabajo.

- Con el fin de mantener al sistema actualizado a nuevos cambios que podrían surgir, algunas de las cosas que aumentaría en una nueva versión sería la generación de un reporte con la información de las visitas, así como un control que permita verificar la ruta que tuvo el vehículo durante todo su trayecto.
- En la Facultad de Ingeniería, especialmente en la Escuela de Sistemas, se debería dictar cursos adicionales optativos para quienes deseen aprender con mayor detalle un tema relacionado con la carrera en específico, para así conseguir mayor conocimiento y mejores oportunidades en el futuro.

BIBLIOGRAFÍA

Comentado [A-2]: Ninguna de la referencias bibliográficas cumplen las normas APA 6th, por favor debes colocarlas en este formato

Allen, G., & Owens, M. (2011). *The Definitive Guide to SQLite* (2 ed.). Nueva York, Estados Unidos: Apress. Recuperado el Febrero de 2018

Cabello, A. L. (2014). *Implantación de aplicaciones web en entornos internet, intranet y extranet.* (Vol. 1). Málaga, España: IC Editorial. Recuperado el Febrero de 2018

Catalinas, E. Q. (2002). *Sistemas operativos y lenguajes de programación.* Madrid, España: Editorial Paraninfo. Recuperado el Febrero de 2018

Collins, L., & Ellis, S. R. (2015). *Mobile Devices: Tools and Technologies.* Boca Raton, Estados Unidos: CRC Press. Recuperado el Marzo de 2018

El-Rabbany, A. (2002). *Introduction to GPS: The Global Positioning System* (2 ed.). Londres, Inglaterra: Artech House. Recuperado el Abril de 2018

Flanagan, D. (2006). *JavaScript: The Definitive Guide* (5th ed.). California, EEUU: O'Reilly Media, Inc.

Gironés, J. T. (2013). *El Gran Libro de Android* (5 ed.). Barcelona, España: S.A. Marcombo. Recuperado el Marzo de 2018

Groussard, T. (2014). *JAVA 8: Los fundamentos del lenguaje Java (con ejercicios prácticos corregidos).* Barcelona: Ediciones ENI.

Holdener, A. T. (2011). *HTML5 Geolocation.* California: O'Reilly Media, Inc.

Khurana, R. (2014). *Operating System, 2nd Edition.* India: Vikas Publishing House.

Kurniawan, B. (2015). *HTML : A Beginner's Tutorial.* Brainy Software Inc.

Lequerica, J. R. (2003). *Web Services (edición especial).* Joan Ribas Lequerica.

Maza, M. Á. (2012). *JavaScript.* Málaga, España: Innovación Y Cualificación.

Moroney, L. (2017). *The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform.* Nueva York: Apress.

Piñeiro, J. M. (2013). *Bases de datos relacionales y modelado de datos.* España: Ediciones Paraninfo, S.A.

Pouncey, I., & York, R. (2011). *Beginning CSS: Cascading Style Sheets for Web Design.* Nueva Jersey, EEUU: John Wiley & Sons.

Pressman, R. (2010). *Ingeniería del Software: un enfoque práctico (7a. ed).* México D.F., MX: McGraw-Hill Interamericana.

Purdy, K. (2011). *The Complete Android Guide.* San Diego: 3ones Inc.

Salz, P. A., & Moranz, J. (2013). *The Everything Guide to Mobile Apps.* Massachusetts: Everything Books.

Tanenbaum, A. S. (2003). *Redes de computadoras.* México: PEARSON EDUCACIÓN.

Westermann, E. (2002). *Learn XML In a Weekend.* USA: Course PTR.

Whitten, J. L., & Bentley, L. D. (2008). *Análisis de sistemas: diseño y métodos (7a. ed.).* México D.F., MX: McGraw-Hill Interamericana.