



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE TECNOLOGÍA EN COMPUTACIÓN

**MONOGRAFÍA: PREVIO A LA OBTENCIÓN DEL
TÍTULO DE TECNÓLOGO EN: COMPUTACIÓN**

**TEMA: “DESARROLLO DE UN SISTEMA DE GEO
LOCALIZACIÓN PARA MONITOREAR LOS VEHÍCULOS DE
TRANSPORTE DE LA COOPERATIVA ANDINA”**

AUTOR: VEGA OBANDO, EDISON FERNANDO

DIRECTOR: ING. SANCHO ARIAS, JOSÉ ALBERTO

SANGOLQUÍ

2017

CERTIFICADO



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE TECNOLOGÍA EN COMPUTACIÓN

CERTIFICADO

Certifico que la Monografía titulada "DESARROLLO DE UN SISTEMA DE GEO LOCALIZACIÓN PARA MONITOREAR LOS VEHÍCULOS DE TRANSPORTE DE LA COOPERATIVA ANDINA" realizado por el Sr. Edison Fernando Vega Obando, está terminado, ha sido revisado periódicamente y analizado por el software anti-plagio, cumple las normas establecidas por la Universidad de Fuerzas Armadas ESPE, por lo tanto, me permito acreditarlo y autorizar al señor Edison Fernando Vega Obando para que lo sustente públicamente.

Sangolqui, 08 de diciembre del 2017

A handwritten signature in blue ink that reads 'José Alberto Sancho Arias'.

Ing. José Alberto Sancho Arias
DIRECTOR

CERTIFICADO DE AUTENTICIDAD



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE TECNOLOGÍA EN COMPUTACIÓN

CERTIFICADO DE AUTENTICIDAD

Yo, Edison Fernando Vega Obando, con cedula de identidad N° 1003360466 declaro que esta monografía titulada "DESARROLLO DE UN SISTEMA DE GEO LOCALIZACIÓN PARA MONITOREAR LOS VEHÍCULOS DE TRANSPORTE DE LA COOPERATIVA ANDINA", ha sido desarrollado en base a una investigación, respetando derechos intelectuales de terceros, cuyas fuentes son citadas e incorporadas en la bibliografía.

En virtud de esta declaración me responsabilizo del contenido, veracidad y alcance científico de esta monografía.

Sangolqui, 08 de diciembre del 2017

Edison Fernando Vega Obando

C.C. 1003360466

AUTORIZACIÓN



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE TECNOLOGÍA EN COMPUTACIÓN

AUTORIZACIÓN

Yo, Edison Fernando Vega Obando, autorizo a la Universidad de la Fuerzas Armadas la publicación en la biblioteca virtual y/o repositorio digital la MONOGRAFÍA de grado titulada "DESARROLLO DE UN SISTEMA DE GEO LOCALIZACIÓN PARA MONITOREAR LOS VEHÍCULOS DE TRANSPORTE DE LA COOPERATIVA ANDINA" cuyo contenido y criterio es de mi exclusiva responsabilidad y autoría.

Sangolqui, 08 de diciembre del 2017

Una firma manuscrita en tinta azul que parece decir 'Edison Vega'.

Edison Fernando Vega Obando

C.C 1003360466

DEDICATORIA

Este trabajo lo dedico a Dios por darme la vida, a mi Madre por ser el pilar fundamental para desarrollar cada uno de mis sueños y por ser la persona a quien debo muchas enseñanzas la cual me ha inculcado muchos valores especialmente la responsabilidad, la persistencia y sobre todo la humildad para lograr los objetivos que me proponga en la vida, a mi hermano por ser un apoyo incondicional en todo momento y darme el mejor regalo que es Danna, a cada una de las personas que de una u otra forma han marcado mi vida para ser día a día un mejor ser humano y un buen profesional.

AGRADECIMIENTO

Quiero agradecer en primer lugar a Dios, a mis familiares por su constante apoyo incondicional, a mis amigos por su ayuda y consejos en la elaboración del presente trabajo y a todos quienes han aportado para la consecución y finalización de este trabajo.

Agradezco al Ing. José Sancho, Director de Tesis, por su apoyo y aportes profesionales, a todo el cuerpo docente que fue parte de mi formación en las aulas.

ÍNDICE DE CONTENIDO

CERTIFICADO	ii.
CERTIFICADO DE AUTENTICIDAD	iii.
AUTORIZACIÓN	iv.
DEDICATORIA	iv
AGRADECIMIENTO	vi
ÍNDICE DE CONTENIDO	vii
RESUMEN	xii
ABSTRACT	xiv
CAPITULO 1	
INTRODUCCIÓN	1
1.1. Antecedentes	1
1.2. Planteamiento del Problema	1
1.3. Justificación e Importancia	2
1.4. Objetivos	2
1.4.1. Objetivo General.....	2
1.4.2. Objetivos Específicos.....	2
1.5. Alcance.....	3
1.6. Factibilidad	3
1.6.1. Factibilidad Operativa	3
1.6.2. Factibilidad Operacional	4
1.6.3. Factibilidad Técnica	4
1.6.4. Factibilidad Tecnológica	4
1.6.5. Factibilidad Económica	4
CAPITULO 2	
MARCO TEÓRICO	6
2.1. Sistemas de Información Geográfica (GIS)	6
2.1.1. Funcionamiento de un Sistema de información Geográfica.....	6
2.1.2. La creación de Datos	6
2.1.3. La Representación de Datos	6
2.1.3.1. Raster	6

2.1.3.2. Vectorial.....	7
2.1.3.3. Puntos.....	7
2.1.3.4. Líneas.....	7
2.1.3.5. Polígonos.....	7
2.1.4. Ventajas y desventajas de los modelos raster y vectorial.....	8
2.1.4.1. Ventajas.....	8
2.1.4.2. Desventajas.....	9
2.1.5. Captura de datos.....	9
2.1.6. Análisis espacial mediante Sistemas de Información Geográfica..	9
2.1.7. Modelo topológico.....	10
2.1.8. Redes.....	10
2.1.9. Superposición de mapas.....	10
2.1.10. Cartografía automatizada.....	10
2.1.11. Geo estadística.....	10
2.1.12. Geo codificación.....	10
2.1.13. El futuro de los Sistema de Información Geográfica (SIG).....	11
2.1.14. Cartografía en entornos web.....	11
2.1.15. La tercera dimensión.....	11
2.2. Sistema de posicionamiento geográfico (GPS).....	11
2.3. Historia.....	12
2.4. Señal GPS.....	13
2.5. Ventajas y desventajas del GPS.....	13
2.5.1. Ventajas del GPS.....	13
2.5.2. Desventajas del GPS.....	14
2.6. Frameworks de desarrollo.....	14
2.6.1. Nodejs.....	14
2.6.2. Express.....	15
2.6.3. Angularjs.....	15
2.6.4. Ionic.....	16
2.6.4.1. Ventajas de las aplicaciones móviles híbridas.....	16
2.6.5. Sublime Text.....	17
2.6.6. Android.....	18
2.7. Metodologías de Desarrollo.....	18
2.7.1. Metodología UWE.....	19
2.7.2. UML.....	22
2.8. Servicios Web.....	25
2.8.1. Servicios Web SOAP (Simple Object Access Protocol).....	26
2.8.2. Servicios Web Rest.....	27
2.9. Tecnologías para Diseño Web.....	28
2.9.1. Html5.....	28
2.9.2. CSS3.....	28
2.9.3. Bootstrap.....	28
2.10. Alojamiento de Aplicaciones Web.....	29
2.10.1. Dominio.....	29
2.10.2. Hosting.....	29

2.10.3. Servidor Privado Virtual (VPS).....	29
---	----

CAPITULO 3

ANÁLISIS, DISEÑO Y DESARROLLO.....	30
3.1. Desarrollo del componente backend.....	30
3.1.1. Análisis	32
3.1.1.1. Obtención de Requerimientos	32
3.1.1.2. Clasificación de los requerimientos	32
3.1.2. Diseño.....	33
3.1.2.1. Definición del escenario.....	33
3.1.2.2. Estructuración del Software	33
3.1.2.3. Definición de tiempos.....	38
3.1.2.4. Asignación de Recursos	38
3.1.3. Desarrollo	39
3.1.3.1. Codificación	40
3.1.3.2. Verificación o comprobación	40
3.1.4. Verificación o comprobación	44
3.1.4.1. Emulación y simulación	44
3.1.5. Entrega	45
3.1.5.1. Distribución	45
3.2. Desarrollo del componente móvil	45
3.2.1. Análisis	46
3.2.1.1. Obtención de los requerimientos.	46
3.2.1.2. Clasificación de los requerimientos	46
3.2.2. Diseño.....	46
3.2.3. Definición del escenario.....	47
3.2.4. Estructuración del Software	47
3.2.5. Definición de tiempos.....	49
3.2.6. Asignación de Recursos	49
3.2.7. Desarrollo.	50
3.2.8. Codificación	50
3.2.9. Verificación o comprobación	51
3.2.9.1. Emulación y Simulación	51
3.2.9.2. Dispositivos Reales.....	52
3.2.10. Entrega	54
3.2.10.1. Distribución	54
3.3. Desarrollo del componente web	54
3.3.1. Modelos de Requerimiento	55
3.3.2. Modelo de Contenido.....	56
3.3.3. Modelo de Navegación	56
3.3.4. Modelo de Presentación	57
3.3.5. Modelo de Procesos	58

CAPITULO 4

IMPLEMENTACIÓN PILOTO.....61

- 4.1. Implementación Backend61
- 4.2. Implementación Componente Móvil61
- 4.3. Implementación Componente Web62
- 4.4. Verificación o comprobación de campo.....63

CAPITULO 5

CONCLUSIONES Y RECOMENDACIONES69

- 5.1. Conclusiones.....69
- 5.2. Recomendaciones.....69

BIBLIOGRAFÍA.....70

ÍNDICE DE FIGURAS

Figura 1 Sistema de Posicionamiento mediante mapas.....	12
Figura 2 Señal GSM.....	13
Figura 3 Arquitectura Node.js.....	15
Figura 4 Aplicación Híbrida.....	16
Figura 5 Señal GSM.....	17
Figura 6 Metodología UWE.....	19
Figura 7 Ejemplo diagrama de casos de uso.....	19
Figura 8 Ejemplo modelo de contenido.....	20
Figura 9 Ejemplo modelo de navegación.....	20
Figura 10 Ejemplo modelo de presentación.....	21
Figura 11 Ejemplo modelo de proceso.....	22
Figura 12 Actor.....	23
Figura 13 Caso de uso.....	23
Figura 14 Ejemplo diagrama de casos de uso.....	24
Figura 15 Clase.....	24
Figura 16 Ejemplo diagrama de clases.....	25
Figura 17 Mensajes SOAP.....	26
Figura 18 Ejemplo mensaje Soap.....	27
Figura 19 Servicios Web Rest.....	27
Figura 20 Componentes Plataforma de Geolocalización.....	30
Figura 21 Componentes Plataforma de Geolocalización.....	31
Figura 22 Metodología de Desarrollo de Aplicaciones Móviles.....	31

Figura 23 Diagrama de Base de Datos.	33
Figura 24 Casos de Uso Interacción Componentes	34
Figura 25 Arquitectura componente Backend.....	39
Figura 26 Estructura del proyecto.....	40
Figura 27 Prueba del servicio de login	41
Figura 28 Prueba del servicio de localización	42
Figura 29 Prueba del servicio de almacenamiento	43
Figura 30 Prueba del servicio registro de usuario	43
Figura 31 Interacción componente Móvil.....	45
Figura 32 Casos de uso aplicación móvil	47
Figura 33 Pantalla de Registro	48
Figura 34 Pantalla de Login.....	48
Figura 35 Pantalla de Inicio de Servicio Tacking	48
Figura 36 Arquitectura componente Móvil	50
Figura 37 Estructura del proyecto móvil	51
Figura 38 Simulador Genymotion	51
Figura 39 Ejecución APP en Emulador	52
Figura 40 Imagen APP Capturada en el móvil	53
Figura 41 Recepción de posición en el backend	53
Figura 42 Interacción componente web.....	55
Figura 43 Diagrama de Casos de Uso Componente Web.....	56
Figura 44 Diagrama de Contenido Componente Web.....	56
Figura 45 Diagrama Navegación Componente Web	57
Figura 46 Modelo de Presentación Componente Web.....	57
Figura 47 Modelo de Procesos Componente Web	58
Figura 48 Mapbox.....	59
Figura 49 Login componente Web.....	59
Figura 50 Pantalla inicial componente Web	60
Figura 51 Consulta vehículo componente Web	60
Figura 52 Ejecución backend	61
Figura 53 Ejecución componente móvil.....	62
Figura 54 Ejecución componente web.....	63
Figura 55 Log Backend Puebas Reales	64
Figura 56 Visualización mapa componente web	64
Figura 57 Visualización mapa componente web real	65
Figura 58 Visualización de la pantalla de horarios y rutas.....	65
Figura 59 Landing Page inicial	66
Figura 60 Landing Page introducción	66
Figura 61 Landing Page características generales.....	67
Figura 62 Landing Page características específicas	67
Figura 63 Landing Page características contacto.....	68

ÍNDICE DE TABLAS

Tabla 1 Presupuesto Referencial	5
Tabla 2 Ventajas modelos raster y vectorial.....	8
Tabla 3 Metodología de Desarrollo de Aplicaciones Móviles	32
Tabla 4 Caso de Uso Registrar	34

Tabla 5 Caso de Uso Ingresar.....	35
Tabla 6 Caso de Uso Guardar.....	35
Tabla 7 Caso de Uso Consultar.....	36
Tabla 8 Estructura Servicios Rest	37
Tabla 9 Tiempos estimados de desarrollo servicios Rest.....	38
Tabla 10 Recursos requeridos para el desarrollo	39
Tabla 11 Características Servidor de Pruebas	44
Tabla 12 Urls Servicio de Pruebas	45
Tabla 13 Clasificación de los requerimientos	46
Tabla 14 Definición de tiempos	49
Tabla 15 Características Ambiente Producción.....	49

RESUMEN

El transporte terrestre interprovincial en el Ecuador es el medio más utilizado por los viajeros para el desplazamiento dentro del país. Actualmente no existe un sistema de bajo costo y fácil implementación que permita la monitorización en tiempo real para conocer la ubicación de las unidades de transporte, además sistemas actuales, para su implementación, requieren la instalación de equipamiento específico en cada unidad. El presente estudio detalla la creación de una plataforma de monitoreo de ubicación en tiempo real aplicado a la Cooperativa de transporte Andina, la plataforma se conformará de tres componentes, una aplicación móvil que podrá ser instalado en un Smartphone y ubicado en el bus de transporte, la aplicación será capaz de capturar y enviar la ubicación basada en el GPS del dispositivo, un sistema de backend que se instalado en un servidor y podrá ser accedido a través de internet, este componente es un API REST capaz de recibir y almacenar los datos enviados desde el dispositivo móvil, además podrá enviar estos datos al componente web, este último componente permitirá la visualización de las unidades de transporte de la Cooperativa Andina en un mapa y permitirá la visualización del desplazamiento en tiempo real así como mostrar información adicional de la unidad de transporte específica.

PALABRAS CLAVE:

- **PLATAFORMA DE MONITOREO DE UBICACIÓN**
- **APLICACIÓN MÓVIL**
- **BACKEND**
- **COMPONENTE WEB**

ABSTRACT

Interprovincial land transport in Ecuador is the most used medium for travelers to travel within the country. Currently there is no low cost and easy implementation system that allows monitoring in real time to know the location of transport units, in addition to current systems, for implementation, require the installation of specific equipment in each unit. The present study details the creation of a real-time location monitoring platform applied to the Andean Transport Cooperative, the platform will consist of three components, a mobile application that can be installed on a smartphone and located on the transport bus, the application will be able to capture and send the location based on the GPS of the device, a backend system that is installed on a server and can be accessed through the internet, this component is a REST API capable of receiving and storing the data sent from the mobile device, you can also send this data to the web component, this last component will allow the visualization of the transport units of the Cooperativa Andina on a map and will allow the visualization of the displacement in real time as well as show additional information of the unit. specific transport.

KEYWORDS:

- **LOCATION MONITORING PLATFORM**
- **MOBILE APP**
- **BACKEND**
- **WEB COMPONENT**

CAPÍTULO 1. INTRODUCCIÓN

1.1. Antecedentes

Un sistema de geo localización Vehicular Automatizada es aplicable a los sistemas de localización remota en tiempo real, basado generalmente en el uso de un GPS, GSM, Bluetooth, WiFi y un sistema de transmisión que es frecuentemente un dispositivo de tecnología inalámbrico. Es importante conocer de datos referenciados de vehículos como posición y ruta de los mismos.

Mediante el sistema se podrá monitorear su ubicación cuando este se encuentra cubriendo una ruta asignada, su desplazamiento, complementado con la visualización en un mapa que muestre la ubicación geográfica, ofreciendo el estado del recorrido por parte de la cooperativa de transporte Andina.

1.2. Planteamiento del Problema

El transporte terrestre interprovincial en el Ecuador es el medio más utilizado por los viajeros para el desplazamiento dentro del país. Es difícil encontrar medios de transporte que cuenten con un sistema que permita el monitoreo de la ubicación, cuando el mismo se encuentra cubriendo una ruta asignada, lo que impide la obtención de información sobre el desplazamiento vehicular. No es posible la visualización en un mapa que muestre la ubicación geográfica y no se puede ofrecer información sobre el estado del recorrido por parte de la cooperativa de transporte Andina; por lo tanto, no se tiene un verdadero control de la flota vehicular de la Cooperativa de transporte Andina.

Actualmente existen sistemas de monitoreo que permiten la obtención de este tipo de información, sin embargo, dichas soluciones son costosas y requieren de la instalación de dispositivos especializados en cada unidad.

1.3. Justificación e Importancia

La falta de un sistema de bajo costo que permita obtener información de la geo localización de las unidades de transporte público interprovincial motivan la creación de una plataforma alternativa a las existentes en el mercado, que permita el monitoreo de las unidades de transporte lo que facilitará el control, supervisión y obtención de información de la ubicación de las diferentes unidades de transporte, las mismas que podrán ser visualizadas en un mapa, generando así un nuevo concepto de control para la cooperativa de transporte Andina.

En términos generales los motivos que justifican la creación de esta plataforma se detallan a continuación:

- Sistema de más bajo costo.
- Sin instalaciones de dispositivos especializados.
- Posibilidad de visualizar la ubicación.
- Falta de información del recorrido del vehículo
- Falta de control de los buses de la Cooperativa Andina
- Servicio adicional que puede brindar la empresa de transporte mejorando la información de los viajes
- Desconocimiento de la existencia de este tipo de soluciones y como es su funcionamiento

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar una aplicación geo referenciada para monitorear las unidades de transporte de la Cooperativa Andina con interacción entre un dispositivo móvil y un portal web.

1.4.2. Objetivos Específicos

- Desarrollar una aplicación con tecnología móvil que permita obtener la ubicación geo referenciada de las unidades de transporte.

- Desarrollar una aplicación web que permita la visualización de las unidades de transporte de la cooperativa Andina.

1.5. Alcance

El presente trabajo de investigación abarca la creación de tres componentes:

- Aplicación móvil cliente que será desarrollada para el sistema Android.
 - Componente móvil de interacción con servicios web para almacenar posiciones de unidades de transporte
 - Rutas
 - Horarios y fechas
 - Conductores / pasajeros
 - Ubicación geográfica
- Aplicación backend para recepción, publicación y almacenamiento de datos de geo localización en una base de datos Open Source Postgres.
- Aplicación web que permita la visualización de los datos de geolocalización en un mapa y publicado en un servidor web
 - Componente de unidades de transporte
 - Rutas
 - Horarios y fechas
 - Conductores / pasajeros
 - Ubicación geográfica
 - Impresión de los datos de las rutas de las unidades de transporte de la cooperativa Andina.

1.6. Factibilidad

1.6.1. Factibilidad Operativa

Para el desarrollo de la plataforma se cuenta con la experiencia de desarrollo en lenguaje Javascript y se investigará los frameworks que se usan para los diferentes componentes que la conforman.

1.6.2. Factibilidad Operacional

Los componentes del sistema no necesitan de equipamiento costoso o especializado lo que hace que la plataforma sea fácilmente implementable, en este sentido es operacionalmente factible el desarrollo como alternativa a sistemas de geo posicionamiento de vehículos de transporte.

1.6.3. Factibilidad Técnica

Para el desarrollo del presente estudio se cuenta con un lugar apropiado equipado con una estación de trabajo, computador portátil y acceso a internet. Para el desarrollo del software se cuenta con acceso a material de investigación para poder culminar con éxito y cumplir los objetivos planteados.

1.6.4. Factibilidad Tecnológica

Para el desarrollo se cuenta con el siguiente equipamiento:

- Computador Portátil. donde se instalarán las herramientas de desarrollo y cuenta con las siguientes características
 - Procesador: Intel Pentium i7
 - Memoria: 8 GB
 - Disco: 250 GB
- Servidor de pruebas. En este equipo se implementará los componentes que se requieran para que sean accedidos a través de internet, contará con las características mínimas necesarias para soportar la aplicación, estas son:
 - Procesador: Intel Xeon
 - Memoria: 2 GB
 - Disco: 50 GB
 - IP. Publica

En base a las premisas se puede mencionar que tecnológicamente el proyecto es factible.

1.6.5. Factibilidad Económica

El proyecto será financiado en su totalidad por el autor del presente estudio, se adjunta el presupuesto referencial:

Tabla 1
Presupuesto Referencial

ITEM	VALOR REFERENCIAL
Hardware	1800
Software	250
Servidor (Hosting)	190
Material de Oficina	200
TOTAL	\$ 2,440.00

CAPÍTULO 2. MARCO TEÓRICO

2.1. Sistemas de Información Geográfica (GIS)

Los sistemas de información geográfica son herramientas y componentes que ayudan a organizar y gestionar la información espacial o geográfica, permitiendo la integración de diferentes variables económica, ambiental, etc, y apoyando en la toma de decisiones.

2.1.1. Funcionamiento de un Sistema de información Geográfica

Es una base de datos geográfica (alfanumérica) relacionada con identificadores de los objetos gráficos de mapas digitales, de esta forma al señalar un objeto se pueden acceder a los atributos o consultando a la base de datos se puede obtener la información cartográfica.

Un SIG puede desplegar la información en capas temáticas, con almacenamiento independiente y gestión ágil y sencilla.

2.1.2. La creación de Datos

Para la creación de datos se utiliza con frecuencia el método de digitalización, el mismo que consiste en realizar la búsqueda de los datos geográficos desde las imágenes orto-rectificadas (satelitales o áreas) disponibles.

2.1.3. La Representación de Datos

Para representar los datos se utilizan dos formas: raster, vectorial.

2.1.3.1. Raster

En su forma más simple, un ráster consta de una matriz de celdas (o píxeles) organizadas en filas y columnas (o una cuadrícula) en la que cada celda contiene un valor que representa información, como la temperatura. Los rásteres son fotografías aéreas digitales, imágenes de satélite, imágenes digitales o incluso mapas escaneados.

Si bien la estructura de datos ráster es simple, es excepcionalmente útil para una amplia variedad de aplicaciones. En un SIG, los usos de los datos ráster se pueden dividir en cuatro categorías principales:

- Rásteres en forma de mapas base.
- Rásteres en forma de mapas de superficie
- Rásteres en forma de mapas temáticos
- Rásteres en forma de atributos de una entidad

2.1.3.2. Vectorial

El objetivo es representar con precisión la localización de elementos geográficos y fenómenos con límites definidos, vinculando dicha información a un registro de una base de datos, en el que se describen sus atributos. La información se utiliza para construir mapas y describir atributos específicos, por ejemplo, un lago y sus niveles de contaminación con diferentes rangos de colores.

Los elementos vectoriales se construyen a partir de su integridad territorial por medio de la aplicación de normas topológicas, por ejemplo: "los polígonos no deben superponerse". Las redes irregulares de triángulos (TIN) nos permiten visualizar diversos valores como la altitud. Las entidades del mundo real se modelan digitalmente a través de puntos, líneas o polígonos.

2.1.3.3. Puntos

Transmiten poca información y son utilizados para representar lugares en un mapa, pero con una escala pequeña, por ejemplo, ciudades.

2.1.3.4. Líneas

En los elementos lineales se puede medir la distancia y son utilizadas para representar carreteras, ferrocarriles o ríos.

2.1.3.5. Polígonos

Transmiten una cantidad más grande de información que otros elementos geométricos, son utilizados para representar un área específica de la superficie terrestre, por ejemplo, edificios, parques, lagos, etc.; permiten medir el área y perímetro.

2.1.4. Ventajas y desventajas de los modelos raster y vectorial

Existen ventajas y desventajas a la hora de utilizar un modelo de datos raster o vector para representar la realidad.

2.1.4.1. Ventajas

La Tabla 2 muestra las ventajas de cada uno de los modelos de representación de datos

Tabla 2
Ventajas modelos raster y vectorial

Vectorial	Raster	Parámetro de comparación
Estructura compacta de los datos y optimización de uso de memoria.	Estructura muy simple de los datos.	Estructura de la información
Codificación eficiente de operaciones espaciales y topología.	Operaciones de superposición sencillas.	Nivel de las operaciones
Gráficos vectoriales con buena definición a diferentes escalas.	Formato de visualización óptimo.	Formato de visualización
Mejor compatibilidad con bases de datos relacionales.	Almacenamiento adecuado de imágenes digitales.	Compatibilidad con entornos de almacenamiento
Facilidad en la ejecución de operaciones de re-escalado y re-proyección.		Operaciones de re-escalado y re-proyección
Mantenimiento y actualización de datos sencilla.		Administración de datos
Mejor capacidad de análisis, en redes.		Capacidad de análisis

2.1.4.2. Desventajas

La Tabla 3 muestra las desventajas de cada uno de los modelos de representación de datos.

Tabla 3
Desventajas modelos raster y vectorial

Vectorial	Raster	Parámetro de comparación
Estructura de datos compleja.	Utiliza mayor capacidad de memoria para su almacenamiento.	Estructura de la información
Mayor dificultad en implementación y representación de Las operaciones de superposición	Mayor dificultad para generar reglas topológicas.	Implementación de operaciones
Para variaciones altas de datos, la eficacia se reduce.	Salidas gráficas poco estéticas y menos vistosas.	Resolución gráfica
Formato difícil para mantenerlo actualizado.		Gestión del formato

2.1.5. Captura de datos

Los datos pueden ser ingresados directamente en un GIS por medio de instrumentos de captura digital como los Sistema de Posicionamiento Global (GPS). Los mapas en película o impresos pueden ser transformados a datos digitales mediante un escaneo.

2.1.6. Análisis espacial mediante Sistemas de Información Geográfica

Existe un incremento sustancial de nuevas técnicas de análisis que van cambiando y son proporcionadas mediante paquetes de software o kits de desarrollo de software (SDK) y que brindan una infinidad de nuevas herramientas de análisis espacial.

2.1.7. Modelo topológico

Se entiende como topología a la relación espacial entre diferentes elementos gráficos y la correspondiente posición en el mapa, las mismas que deben ser establecidas en un software mediante reglas de geometría matemática.

2.1.8. Redes

Se utiliza para determinar la ruta más óptima entre dos puntos, tomando en cuenta sentidos de circulación y direcciones restringidas. Se pueden incorporar otros elementos como el límite de velocidad o los niveles de servicio, los mismos que nos ayudarán a obtener más precisión en los cálculos.

2.1.9. Superposición de mapas

Se realiza mediante un proceso llamado álgebra de mapas, utilizando métodos matemáticos y combinando valores de cada matriz raster.

2.1.10. Cartografía automatizada

Son procesos automatizados que permiten la construcción de mapas mediante herramientas incorporadas en los sistemas de información geográfica.

2.1.11. Geo estadística

Permite analizar y desplegar propiedades estadísticas mediante el empleo de la teoría de grafos y matrices algebraicas, partiendo de variables espaciales.

2.1.12. Geo codificación

Permite interpretar la entrada de datos de una ubicación, ya sea ingresando una dirección, un código postal, un lugar o coordenadas y transformarla en la ubicación real en la superficie de la tierra y presentarla en un mapa.

2.1.13. El futuro de los Sistema de Información Geográfica (SIG)

Cada vez se masifica el uso de aplicaciones relacionadas con ubicaciones en el mapa en diferentes sistemas de seguridad, defensa, arqueología, transporte y aplicaciones móviles.

El futuro se visualiza como la integración de los SIG en muchas más aplicaciones, perfeccionando la información y estadísticas presentadas para mejorar el servicio y la toma de decisiones.

2.1.14. Cartografía en entornos web

Es común encontrar herramientas, aplicativos o sistemas encaminados a gestionar información cartográfica en entornos web, se han creado API's que permiten a los desarrolladores implementar el tema de la información geográfica en diferentes sistemas web de uso público o privado, con o sin costo.

2.1.15. La tercera dimensión

Los SIG se están perfeccionado para la gestión de datos en tres dimensiones y además agregar el elemento tiempo para acercarse cada vez más a la realidad.

Actualmente existe la tendencia de caminar hacia la tercera dimensión, por lo cual ya se han creado varios sistemas híbridos que son el paso intermedio entre 2D y 3D y que han mejorado especialmente en la visualización.

2.2. Sistema de posicionamiento geográfico (GPS)

Un GPS, ayuda a obtener con gran precisión la ubicación de un objeto, persona o lugar.

Su funcionamiento está sustentado en la trilateración y 24 satélites (3 son repuesto) que orbitan alrededor de la tierra y distribuidos de tal forma que cubren su superficie.

El sistema se conforma de tres elementos:

- Los satélites que forman el elemento espacial y cuya función es transmitir la hora y posición exacta, gracias al reloj atómico ultra preciso que tienen incorporados.

- Monitoreo de la continuidad del servicio de cada satélite, es el elemento de control, actualiza las posiciones y sincroniza los relojes atómicos.
- Receptores GPS, son el elemento usuario y se encargan de tomar las señales de los satélites y determinar la posición o sus coordenadas.

2.3. Historia

El GPS fue creado por el Departamento de Defensa de los Estados Unidos e inaugurado en 1995; nació entre 1964 y 1967 con Transit, al inicio su uso era netamente militar, pero en la actualidad se ha diversificado en aplicaciones civiles para diferentes tipos de dispositivos y ámbitos.

El acceso a los satélites en un inicio tomaba alrededor de 90 minutos, es decir el posicionamiento era discontinuo. Con la incorporación de los relojes atómicos en los satélites, se mejoró el tiempo de acceso a la información emitida por los mismos, permitiendo una sincronización entre todos ellos.

En la actualidad, no es común utilizar mapas en papel y los mapas digitales han masificado su uso en diferentes áreas públicas y privadas.

En los años 70 y 80, se realizaron avances importantes, por ejemplo, el programa de navegación de señal codificada NAVSTAR que unió a la armada y fuerza aérea estadounidense.

Estados Unidos brindó el servicio de posicionamiento en 2009 a la OACI para ayudar en sus necesidades y requerimientos.

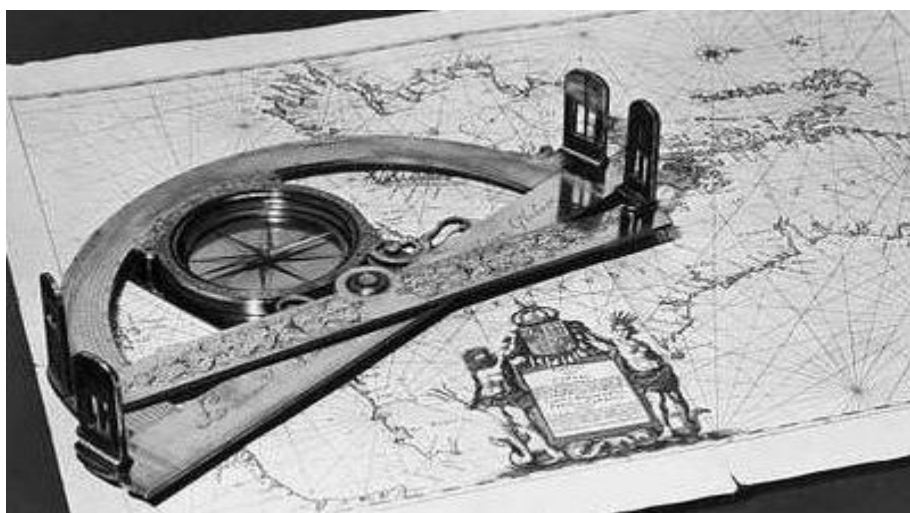


Figura 1 Sistema de Posicionamiento mediante mapas.
Fuente: (SGS, 2016)

2.4. Señal GPS

Brinda datos como la hora, el número de semana e información para determinar el estado de su funcionamiento, los mismos que son enviados de forma precisa cada 30 segundos, transmitiendo 1500 bits codificados pseudoaleatoriamente conocidos como códigos PNR y que son decodificados por los receptores.

La señal está compuesta de dos partes, la primera tiene que ver con la hora de acuerdo al reloj atómico del satélite y la segunda brinda información de la órbita del satélite.



Figura 2 Señal GSM
Fuente: (Egotrack, 2016)

2.5. Ventajas y desventajas del GPS

2.5.1. Ventajas del GPS

- Localizar un lugar determinado y brindar información de la ruta para llegar al mismo, en ciertas aplicaciones se ha integrado ayuda de voz.
- Brinda información sobre sitios de interés próximos al posicionamiento en el que nos encontremos, o al lugar donde estamos queriendo llegar.
- Está integrado en varios dispositivos y en diferentes aplicaciones que nos ayudan en el trabajo como en la vida cotidiana.

- Existen aplicativos que permiten determinar las distancias en millas o kilómetros, el consumo de combustible, la velocidad y el tiempo estimado de llegada al destino.
- Brinda información del posicionamiento de un objeto o dispositivo en caso de una sustracción o pérdida del mismo.

2.5.2. Desventajas del GPS

- Consume muchos recursos ya que está constantemente recibiendo información de los satélites.
- El abuso del uso del GPS podría provocar daños en la salud por la emisión de las ondas radioactivas.
- Para tener una buena experiencia y servicio adecuado, es necesario mantener el dispositivo y software actualizado, eso nos ahorraría muchos problemas en caso de que existan cambio de sentido en vías, mantenimiento de las vías, cambio de direcciones, etc.

2.6. Frameworks de desarrollo

2.6.1. Nodejs

“Es un entorno de ejecución para Javascript construido con el motor [JavaScript V8 de Chrome](#)” (Nodejs, 2017).

Nodejs permite la ejecución de Javascript del lado del servidor, es una tecnología que permite realizar componentes de backend usando lenguaje javascript, desde el año 2009 esta tecnología ha ido creciendo en cuanto a su uso debido a la facilidad de uso, facilidad para implementación y a su desempeño. Nodejs al ser una tecnología orientada en eventos hace que su respuesta sea más rápida en comparación con otras tecnologías que sirven para programación del lado del servidor tales como Java o PHP.

Nodejs ha ido en creciente evolución lo que ha permitido que se creen librerías las mismas que pueden ser usadas dentro de los proyectos y ahorran tiempo en la implementación de funcionalidades complejas.

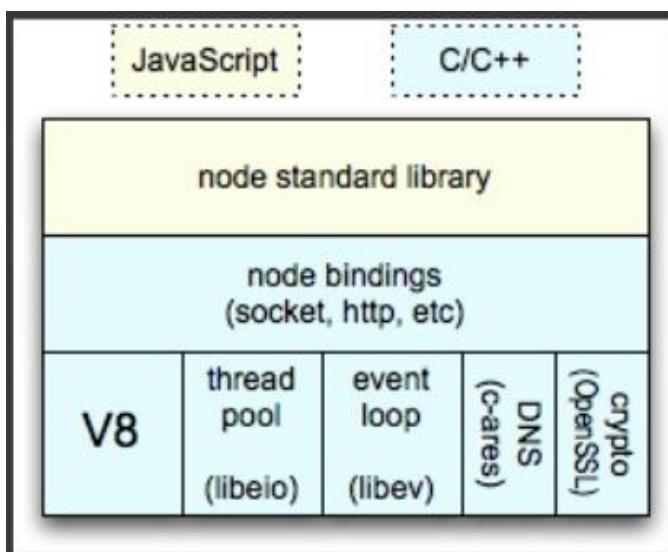


Figura 3 Arquitectura Node.js

Fuente: (www.nodehispano.com, 2017)

2.6.2. Express

Es un framework que permite usar Nodejs para el desarrollo de aplicaciones web su principal uso es para la creación de middlewares que permitan la comunicación de los componentes de backend con los componentes de front end de una aplicación web mediante el uso de API basados en servicios Rest.

2.6.3. Angularjs

Es un framework mantenido por Google, su uso principal es para la creación de aplicaciones Web en su componente de front end, al ser basado en javascript Angularjs permite ampliar las capacidades del lenguaje HTML que se usa para el diseño de páginas web.

Angularjs basa el diseño de aplicaciones en un patrón de diseño MVC (Modelo Vista Controlador) lo que permite que el código sea mantenido de manera más fácil.

Angularjs tiene como objetivo la creación de aplicaciones web tipo SPA (Single Page Application) es decir que exista una sola página y que en la misma se cambien ciertas porciones lo que permite la reutilización de los componentes generales.

2.6.4. Ionic

Es un framework que permite la creación de aplicaciones móviles híbridas. Una aplicación híbrida es un tipo de aplicación móvil que usa un navegador para desplegar la interfaz de usuario. Ionic es una colección de herramientas que permite a los desarrolladores usar la misma tecnología que se usa en sitios web para crear aplicaciones móviles principalmente usa Javascript, CSS, HTML.

Ionic usa Angularjs como frameworks para la parte visual de las aplicaciones y Cordova para crear y empaquetar la aplicación en una nativa sea esta de Android, IOS o Windows.

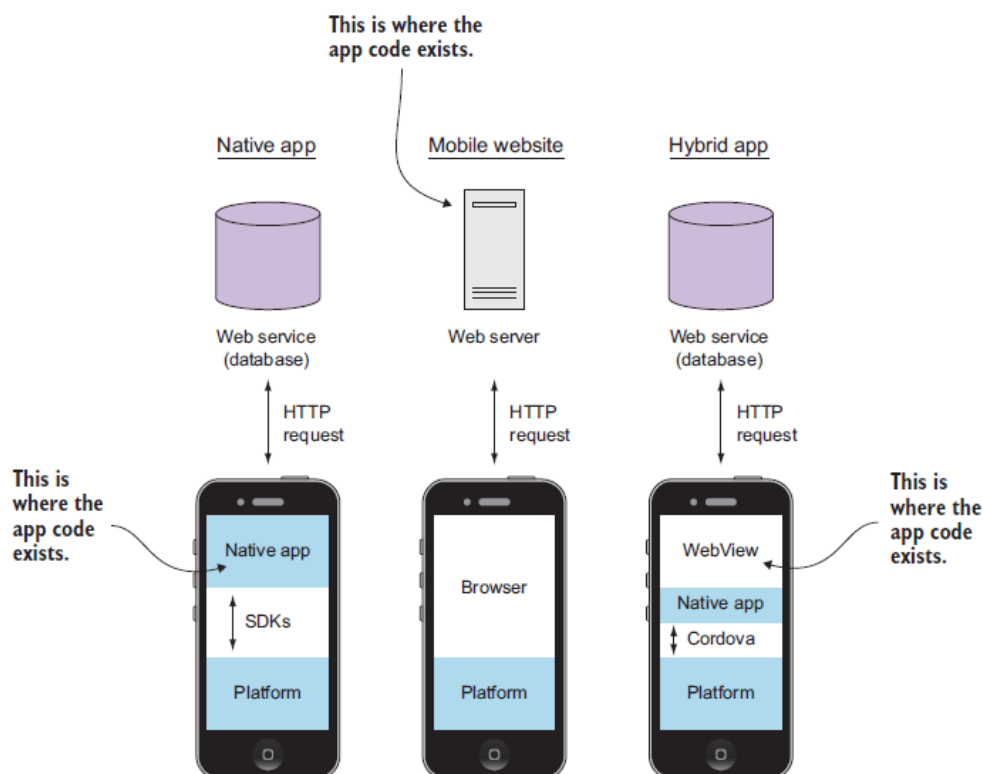


Figura 4 Aplicación Híbrida.

Fuente: (Wilken, 2016)

2.6.4.1. Ventajas de las aplicaciones móviles híbridas

- **Multiplataforma.** Las aplicaciones móviles híbridas que se desarrollan con Ionic pueden ser empaquetadas para diferentes sistemas operativos

móviles, es decir su desarrollo es independiente del sistema móvil en el que se requiere su ejecución.

- Tecnología igual a los sitios web. Desarrollar una aplicación móvil híbrida es similar a desarrollar una página web tradicional, lo que permite que los desarrolladores con experiencia en desarrollo web puedan fácilmente desarrollar aplicaciones móviles.
- Acceso a componentes del móvil. Ionic permite el acceso a componentes del móvil, a través de plugins, siendo fácil su manipulación e inclusión en la aplicación móvil proveyéndoles de funcionalidades que usen por ejemplo: Cámara, GPS, etc.
- Fácil desarrollo. El desarrollo de aplicaciones móviles híbridas es relativamente más fácil debido a que se usa componentes iguales a los que se utilizan en páginas web, la curva de aprendizaje para el desarrollo de aplicaciones híbridas es menor comparándolo con el desarrollo de una aplicación móvil nativa.

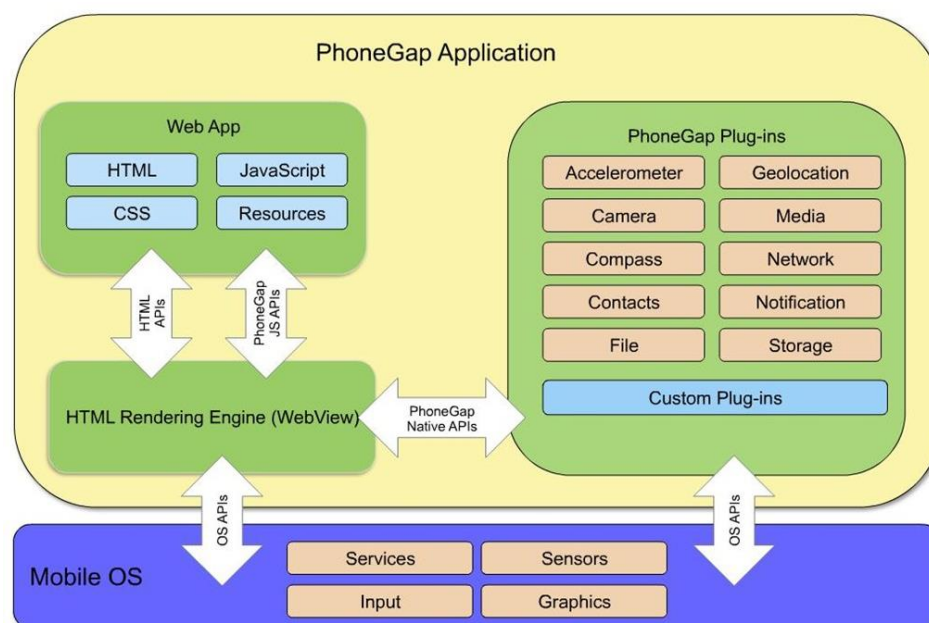


Figura 5 Señal GSM

Fuente: (www.gaurav.mobi, 2017)

2.6.5. Sublime Text

Es un editor de texto y editor de código fuente escrito en C++ y Python. Desarrollado originalmente como una extensión de Vim, con el tiempo fue

creando una identidad propia, por esto aún conserva un modo de edición tipo vi llamado Vintage mode. Se puede descargar y evaluar de forma gratuita.

2.6.6. Android

Es un sistema operativo para dispositivos móviles basado en linux es mantenido por Google ya que lo adquirió en el año 2005. Es un sistema operativo abierto y gratuito, debido a esto ha tenido un crecimiento en popularidad porque no tiene costo de licenciamiento.

Desde su aparecimiento se ha mejorado en muchos aspectos y se han ido liberando varias versiones las mismas que se listan a continuación:

Honeycomb	3.0–3.2.6	22 de febrero de 2011
Ice Cream Sandwich	4.0–4.0.4	18 de octubre 2011
Jelly Bean	4.1–4.3.1	9 de julio de 2012
KitKat	4.4–4.4.4, 4.4W–4.4W.2	31 de octubre de 2013

Las aplicaciones para Android pueden ser instaladas de dos maneras: a través de Google Play Store o de manera manual mediante un archivo apk. En la primera opción se necesita una cuenta de Google para poder acceder a la tienda de aplicaciones la segunda no necesitará una cuenta, pero a su vez representa un riesgo de seguridad debido a que ciertas aplicaciones pueden contener virus.

2.7. Metodologías de Desarrollo

Las metodologías de desarrollo de software son mecanismos que permiten la construcción de aplicaciones garantizando la calidad, en la actualidad existe software orientado a aplicaciones web lo que ha permitido que se construyan aplicaciones complejas y que sean alojadas en Internet y accedidas a través de un browser por tal motivo es indispensable contar con mecanismos o metodologías que ayuden en todas las fases de desarrollo, en este sentido se pueden mencionar varias metodologías:

- UWE. Metodología basada en UML usa exclusivamente notaciones y extensiones de UML.
- WML. Esta metodología permita la representación de especificaciones gráficas, es decir, es orientado al diseño de las interfaces de usuario.

2.7.1. Metodología UWE

La metodología UWE define varias etapas y modelos:

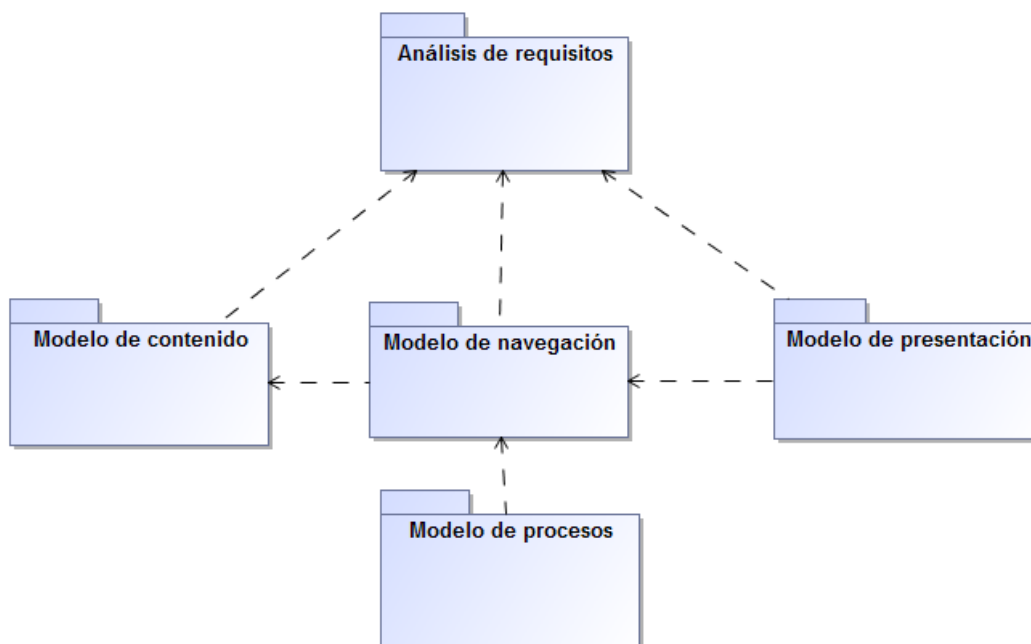


Figura 6 Metodología UWE

Fuente: (Citlali Guadalupe Nieves Guerrero)

Análisis de Requisitos. En esta etapa se muestran los requisitos de la aplicación web, la forma de representación es a través de diagramas de casos de uso.

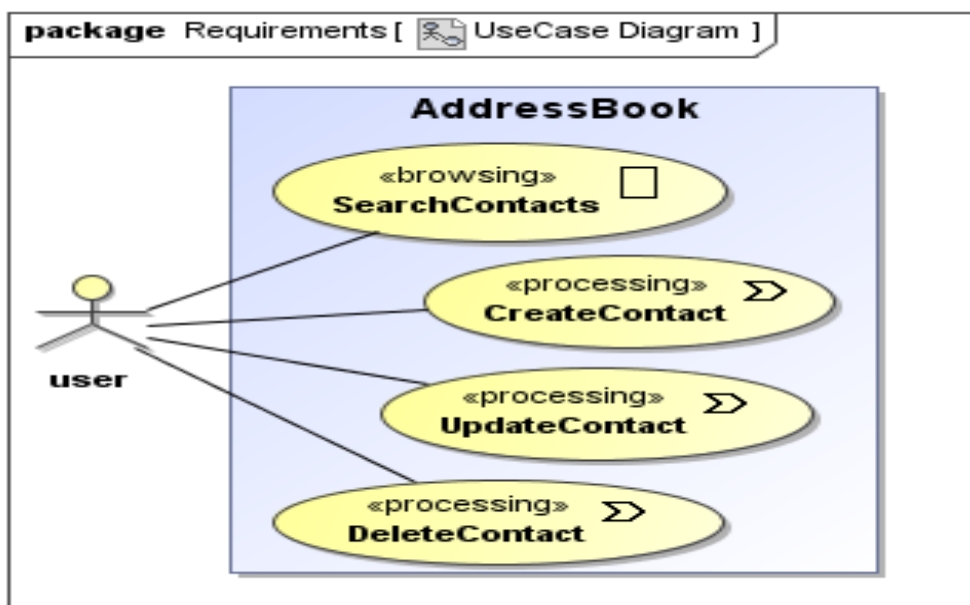


Figura 7 Ejemplo diagrama de casos de uso

Fuente: (uwe.pst.ifi.lmu.de, 2017)

Modelo de Contenido. En este modelo se representan los detalles de los componentes involucrados en la aplicación, la forma de representación es a través de diagramas de clases.

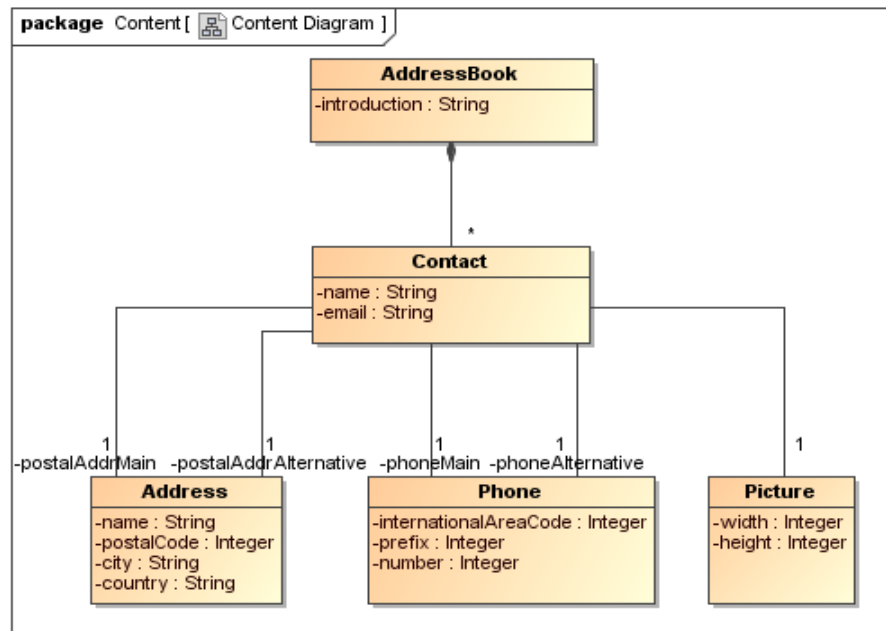


Figura 8 Ejemplo modelo de contenido

Fuente: (uwe.pst.ifi.lmu.de, 2017)

Modelo de Navegación. En este modelo se representa la navegación de los diferentes objetos de la aplicación web se representa las estructuras de los menús, índices y consultas.

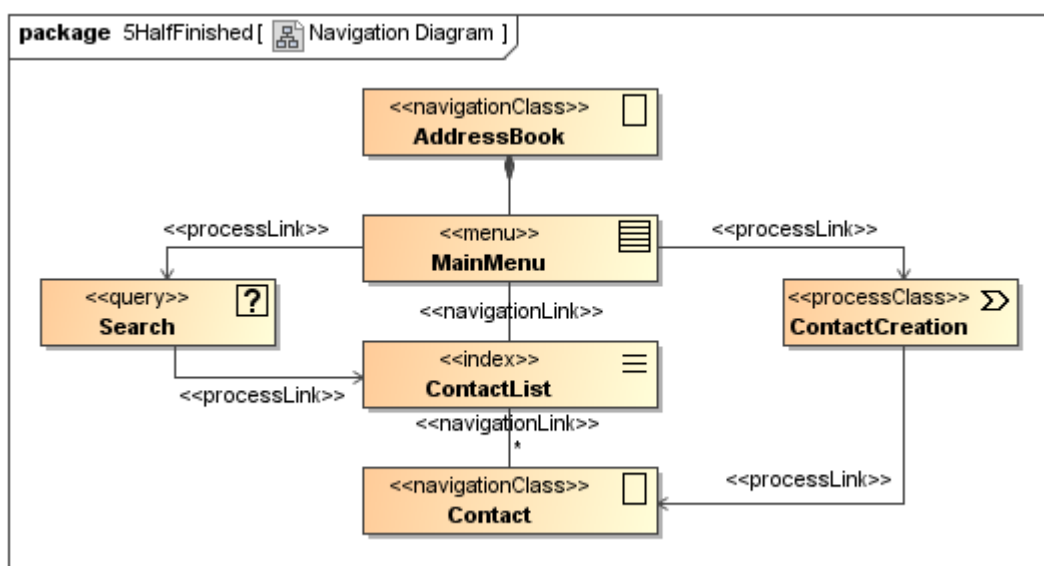


Figura 9 Ejemplo modelo de navegación

Fuente: (uwe.pst.ifi.lmu.de, 2017)

Modelo de Presentación. En este modelo se representa el diseño de las interfaces de usuario o vistas de manera abstracta, es decir, muestra una pantalla básica de lo que sería las interfaces web se basa en el modelo de navegación. Este modelo nos da la idea clara de la parte visual de la aplicación web.

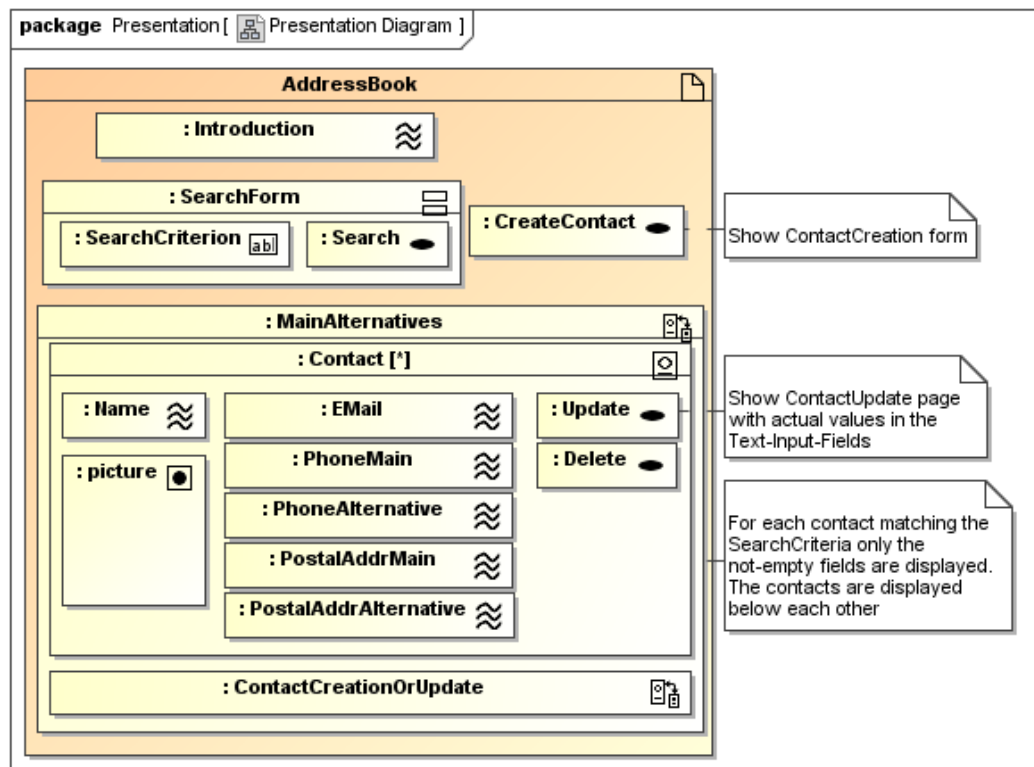


Figura 10 Ejemplo modelo de presentación

Fuente: (uwe.pst.ifi.lmu.de, 2017)

Modelo de Proceso. El objetivo del modelo de procesos es mostrar el flujo de ejecución de la aplicación Web, su forma de representación es a través de diagramas de actividades.

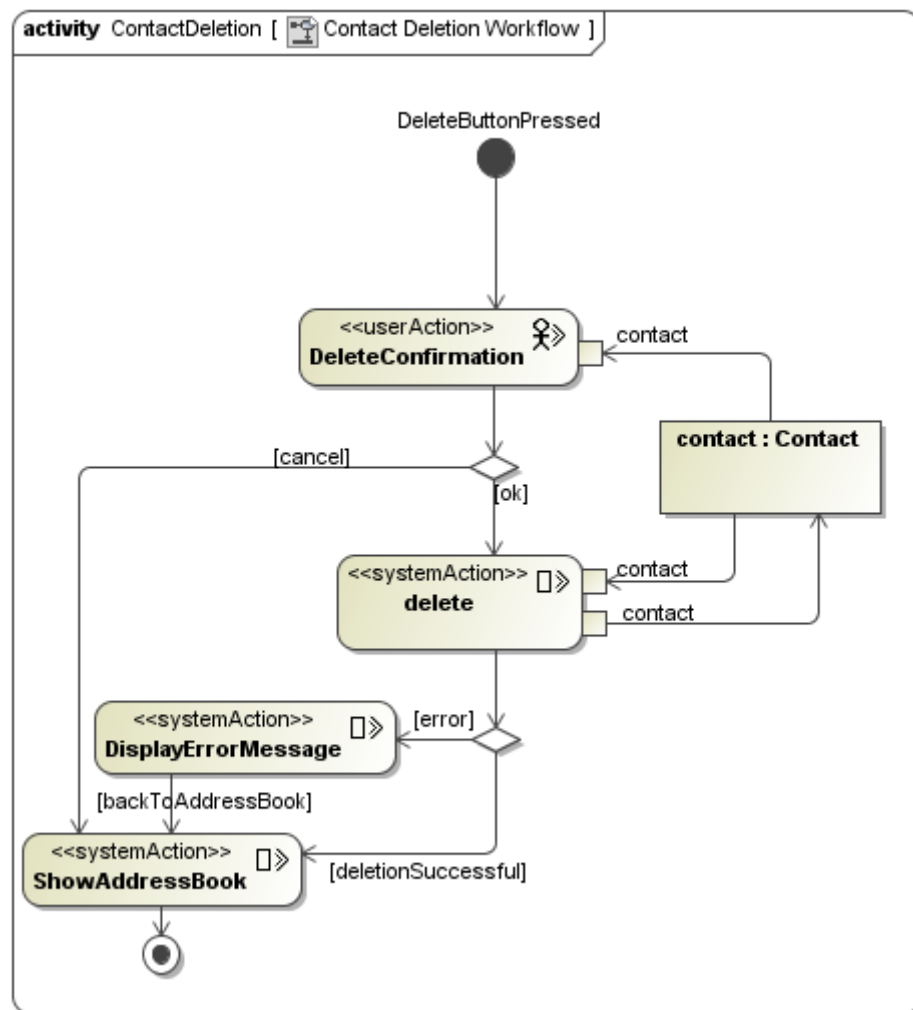


Figura 11 Ejemplo modelo de proceso

Fuente: (uwe.pst.ifi.lmu.de, 2017)

2.7.2. UML

El Lenguaje Unificado de Modelado es un estándar que consta de normas y especificaciones gráficas y sirve para representar diagramas, esquemas y documentación de un software específico en desarrollo.

UML define varios tipos de diagramas para representar la funcionalidad y comportamiento que debe tener determinado software. Los diagramas que se especifican en UML son:

- Diagrama de casos de uso. El diagrama de casos de uso representa las acciones que un usuario (Actor) realiza en el sistema consta de los siguientes elementos:
 - Actor

- Relaciones
- Caso de uso

El Actor especifica el usuario y su rol dentro del sistema.

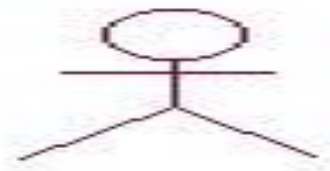


Figura 12 Actor

Fuente: (users.dcc.uchile.cl, 2017)

El caso de uso representa una acción o tarea que es ejecutada por un actor o por otro caso de uso

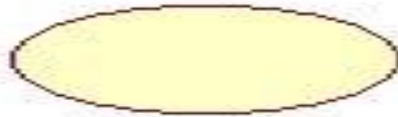


Figura 13 Caso de uso

Fuente: (users.dcc.uchile.cl, 2017)

Las relaciones son representaciones de la interacción de los casos de uso de un sistema, dentro de las relaciones que se pueden presentar tenemos las siguientes:

- Asociación
- Dependencia
- Generalización

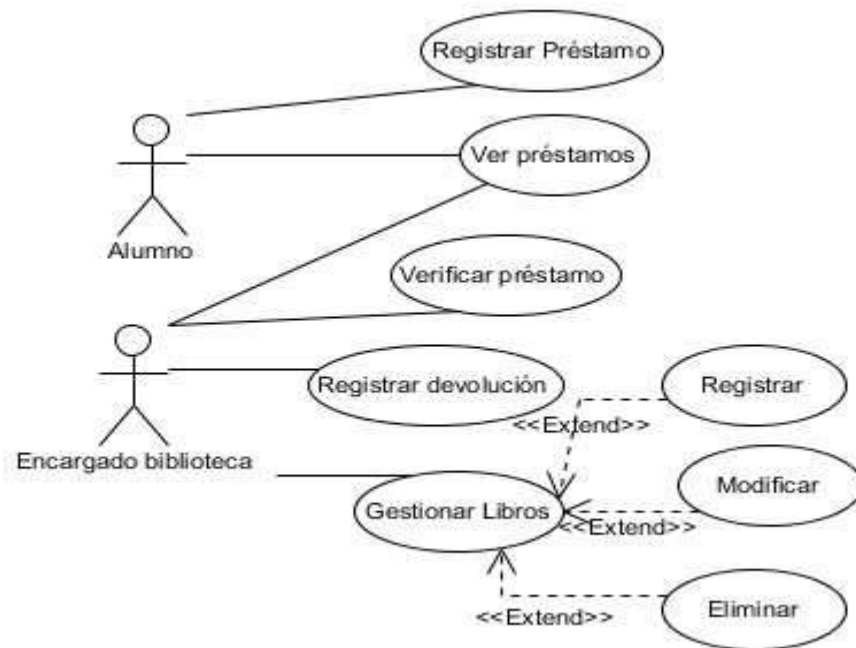


Figura 14 Ejemplo diagrama de casos de uso

Fuente: (www.researchgate.net, 2017)

- Diagrama de clases. Este diagrama muestra las clases de un sistema, las relaciones que existen entre ellas, así como sus métodos y atributos. El diagrama de clases consta de los siguientes elementos:

- Clase
- Relaciones

La clase es una representación de un objeto y muestra toda su información en cuanto a métodos y atributos

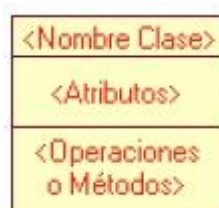


Figura 15 Clase

Fuente: (users.dcc.uchile.cl, 2017)

Las clases a su vez pueden tener relaciones entre ellas las relaciones que se pueden presentar entre las clases son

- Herencia
- Agregación

- Asociación

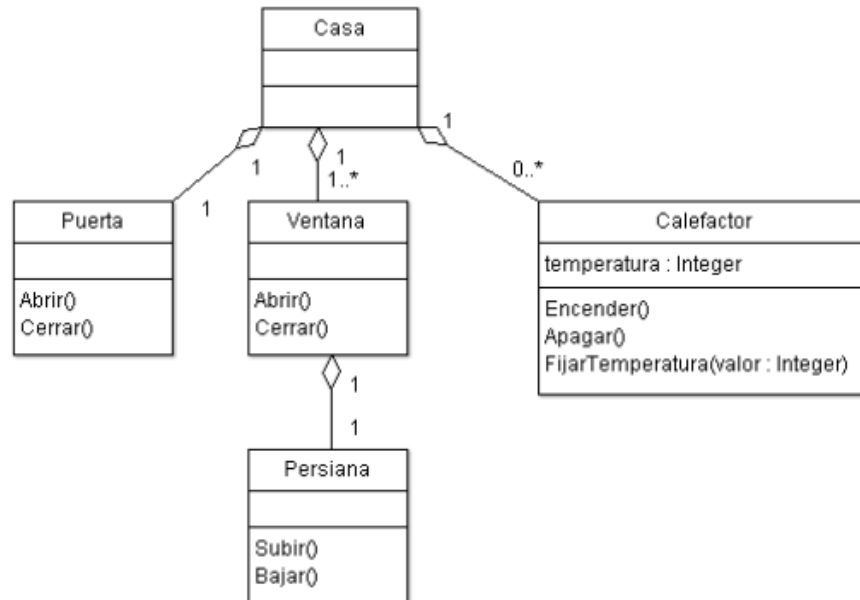


Figura 16 Ejemplo diagrama de clases

Fuente: (users.dcc.uchile.cl, 2017)

- Diagrama de secuencia. Este diagrama muestra los componentes de software y el intercambio de mensajes entre ellos.
- Diagrama de colaboración. Este diagrama muestra como los objetos se comunican y colaboran entre ellos para cumplir un objetivo dentro de sistema.
- Diagrama de estado. Este diagrama muestra el estado del sistema conforme se realicen determinados eventos o acciones dentro del mismo.
- Otro tipo de diagramas. En UML se pueden especificar más diagramas dependiendo de la necesidad de diseño de determinado software entre los que se pueden anotar el diagrama de paquetes, diagrama de arquitectura de software, diagrama de actividad, etc.

2.8. Servicios Web

Los servicios web son tecnologías de computación distribuida que se implementan principalmente para comunicación entre sistemas, los servicios web especifican estándares y protocolos que permite a los sistemas

informáticos intercambiar información independiente de la plataforma y del lenguaje de programación y la comunicación se la realiza a través del protocolo http lo que permite que los sistemas se conecten a través de internet sin importar la ubicación geográfica del mismo.

Existen principalmente dos tipos de Servicios Web, depende de la arquitectura de la aplicación su utilización. En los siguientes enunciados se mencionará las características de SOAP y Rest.

2.8.1. Servicios Web SOAP (Simple Object Access Protocol).

La especificación fue inicialmente desarrollada por Microsoft, el intercambio de información se la realiza exclusivamente a través de XML. SOAP define un documento que especifica los métodos, atributos y tipo de datos que usa el servicio, es decir, se especifica claramente y de forma detallada las funciones del servicio web, dicho documento de denominada WSDL.

La especificación SOAP define varios tipos de mensajes que permiten el acceso o la ejecución de ciertas funciones en un equipo remoto y la obtención o no de cierta información.



Figura 17 Mensajes SOAP.

Fuente: (Universidad de Alicante, 2017)

Dichos mensajes son enviados en formato XML y generalmente usan el protocolo http. Un ejemplo de mensaje SOAP se muestra a continuación:


```

1 | <SOAP-ENV:Envelope
2 |   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
3 |   SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
4 |   <SOAP-ENV:Body>
5 |     <ns:getTemperatura xmlns:ns="http://j2ee.ua.es/ns">
6 |       <area>Alicante</area>
7 |     </ns:getTemperatura>
8 |   </SOAP-ENV:Body>
9 | </SOAP-ENV:Envelope>

```

Figura 18 Ejemplo mensaje Soap.

Fuente: (Universidad de Alicante, 2017)

2.8.2. Servicios Web Rest

Los servicios web Rest se basan en las operaciones definidas en el protocolo http, es decir, pueden usar los verbos GET, POST, PUT, y DELETE para definir las acciones.

Los servicios Rest ofrecen una alternativa liviana a SOAP debido a que no se debe enviar o estructurar mensajes XML simplemente es necesario invocar la URL por la que se expone el servicio.

Los servicios Rest pueden retornar los mensajes en formato XML o JSON, sin embargo es más común el uso de mensajes JSON debido a la facilidad de su interpretación por los diferentes lenguajes de programación y los mismos se pueden implementar usando herramientas básicas.

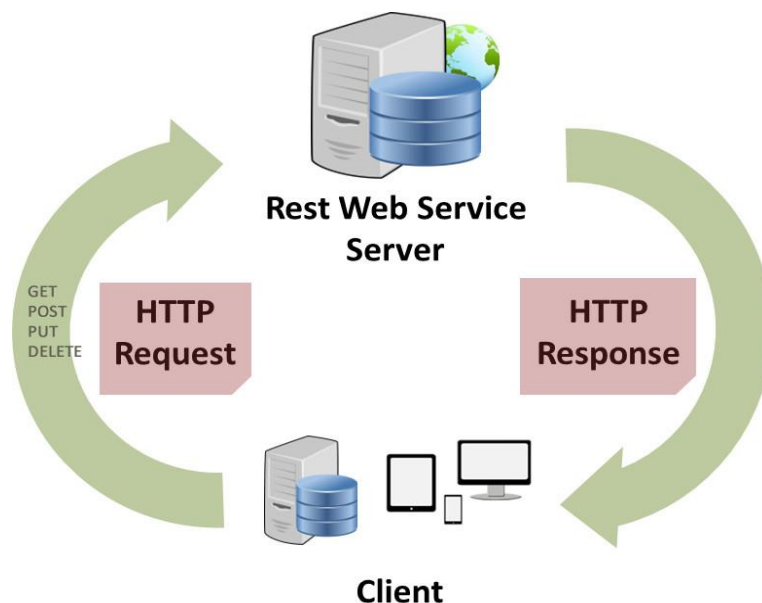


Figura 19 Servicios Web Rest.

Fuente: (restysoap.blogspot.com, 2017)

2.9. Tecnologías para Diseño Web

2.9.1. Html5

Es la última versión del lenguaje HTML el mismo que se usa para diseñar y presentar contenido orientado a la web y visualizado mediante los navegadores. Con el lenguaje HTML se suministra a los navegadores ciertas “instrucciones” para que interpreten y distribuyan los diferentes componentes de la página web, es decir, ubicación de texto, imágenes, multimedia, etc.

2.9.2. CSS3

Es un lenguaje que se utiliza para controlar el aspecto gráfico de los componentes creados con html. Su uso se centra exclusivamente para crear la parte visual de una página web permitiendo separar la estructura html de la presentación visual. CCS3 es la última versión del lenguaje e incorpora varias mejoras entre las que se pueden anotar las siguientes:

- Bordes
- Backgrounds
- Selectores
- Formatos

2.9.3. Bootstrap

Es un framework desarrollado inicialmente por Twitter que permite desarrollar interfaces web basados en CCS3 y javascript. Bootstrap define el comportamiento de las interfaces para que sean adaptables al tamaño del dispositivo, a esto se le denomina diseño responsivo. Al hacer uso del framework esta tarea se facilita debido a que de forma nativa Bootstrap define clases CCS3 para implementar este comportamiento.

Bootstrap es open source y actualmente se encuentra en la versión 3.3.7, a la fecha del presente estudio, y soporta complementos para mejorar la parte visual de las páginas web.

2.10. Alojamiento de Aplicaciones Web.

2.10.1. Dominio

Un dominio de Internet es un nombre único que identifica a un sitio web en Internet (Wikipedia, 2017). El objetivo de un dominio es traducir la dirección IP a un nombre que se más fácilmente memorizable lo que permite la búsqueda del mismo.

Un dominio también facilita el movimiento o cambio de dirección IP física debido a que el nombre se traducirá a una dirección IP especificada y que puede ser cambiada.

2.10.2. Hosting

Es un servicio que permite almacenar contenido multimedia e información en un espacio y que sea accedido a través internet, es decir, alojan las páginas web en un espacio accesible desde Internet.

Un servicio de hosting permite alojar páginas web y sus componentes, estos son bases de datos, dominios, mecanismos de seguridad, etc.

Un servicio de hosting generalmente es accedido a través de un panel de control limitado, es decir permite administrar solo las funcionalidades que fueron contratadas.

2.10.3. Servidor Privado Virtual (VPS)

Un servidor privado virtual es un servidor que es completamente administrado por el usuario que lo contrata, es decir, puede manipular el sistema operativo y las aplicaciones que corren dentro de este servidor.

Este tipo de servidores son usados en aplicaciones en las que se pretende hacer uso de puertos, aplicaciones o funciones que no son las estandarizadas por las empresas de hosting para el alojamiento de páginas web.

CAPÍTULO 3. ANÁLISIS, DISEÑO Y DESARROLLO

En base al fundamento teórico del presente estudio se desarrollarán los diferentes módulos que formarán parte de la aplicación. Estos componentes son:

- Componente Backend
- Componente móvil
- Componente WEB

3.1. Desarrollo del componente backend

El componente de backend permitirá tanto a la aplicación móvil como al componente web comunicarse, para almacenar o recuperar de una base de datos, las diferentes posiciones geo referenciadas de los vehículos que cuenten con dicha aplicación móvil.

El componente de Backend básicamente será un API que publique servicios Rest los mismos que será consumidos desde el dispositivo móvil, esto quiere decir que cierta lógica debe estar en este componente y que permita implementar la funcionalidad requerida por la plataforma propuesta.

En la Figura 20 se muestra la distribución e interacción de los diferentes componentes de la aplicación de manera general.

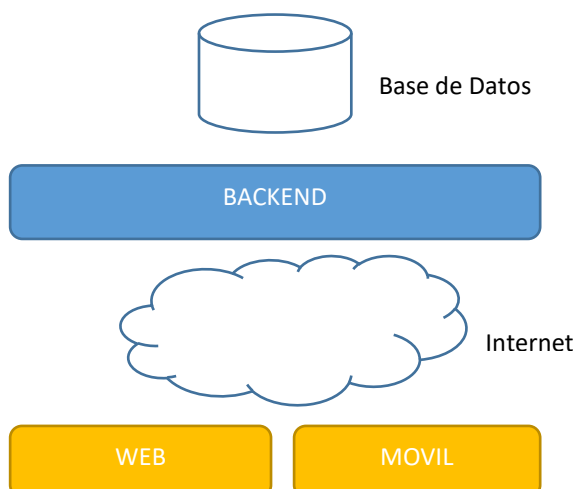


Figura 20 Componentes Plataforma de Geo localización
Fuente: El Autor

Los diferentes componentes se desarrollaron en lenguaje Javascript como base, la arquitectura de la aplicación de las diferentes librerías se muestra en la figura 21:

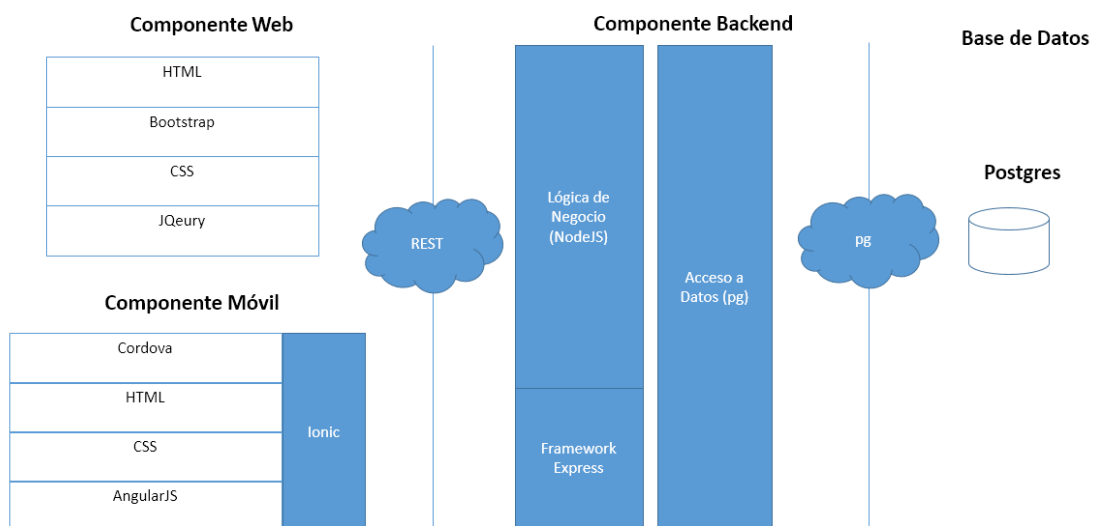


Figura 21 Componentes Plataforma de Geolocalización
Fuente: El Autor

Para el desarrollo de este componente se hará uso de la metodología propuesta por (Maira Cecilia Gasca Mantilla, 2013) y que contempla las siguientes etapas:

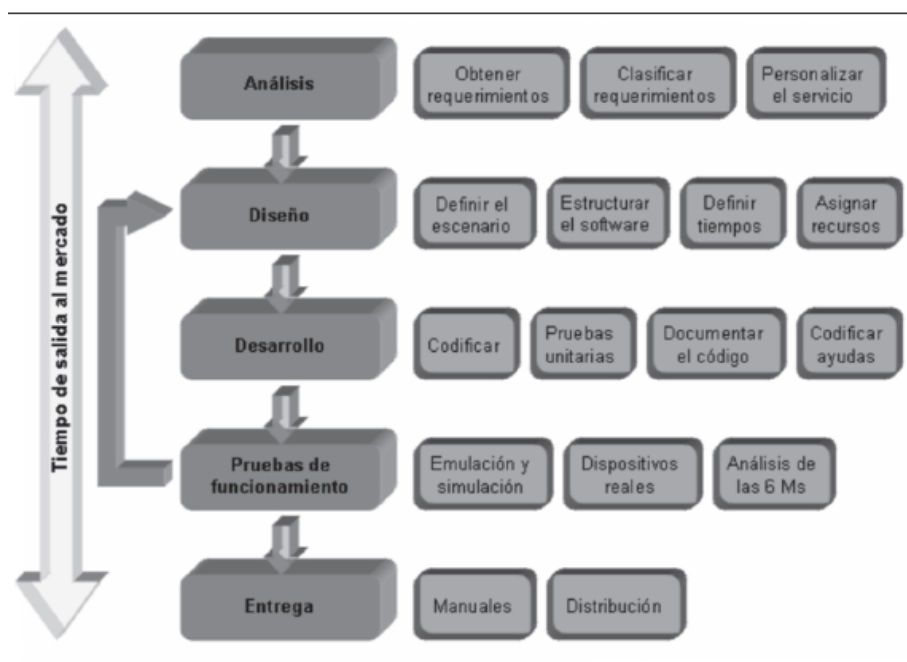


Figura 22 Metodología de Desarrollo de Aplicaciones Móviles
Fuente: (Maira Cecilia Gasca Mantilla, 2013)

3.1.1. Análisis

En esta etapa se plantea los requerimientos que el componente de backend debe cumplir para el desarrollo de la plataforma.

3.1.1.1. Obtención de Requerimientos

En esta fase se identificó que el componente de backend debe permitir las siguientes funcionalidades:

- Recibir las coordenadas del componente móvil y almacenarlos en la base de datos.
- Recibir los datos de registro de un nuevo usuario y almacenarlo en la base de datos.
- Recibir las credenciales de usuario y validar el mismo.
- Entregar la última coordenada o posición de determinado vehículo.

3.1.1.2. Clasificación de los requerimientos

Del levantamiento de requerimientos inicial se los clasificó por categorías y por tipo de funcionalidad, se obtiene la siguiente tabla

Tabla 3
Metodología de Desarrollo de Aplicaciones Móviles

Número	Requerimiento	Componente de Uso
1	Recibir las coordenadas del componente móvil y almacenarlos en la base de datos	MOVIL
2	Recibir los datos de registro de un nuevo usuario y almacenarlo en la base de datos	MOVIL
3	Recibir las credenciales de usuario y validar el mismo	MOVIL Y WEB
4	Entregar la última coordenada o posición de determinado vehículo	WEB

3.1.2. Diseño

En esta etapa se realizarán los esquemas o diagramas que contemplen la mejor solución en cuanto a diseño para el desarrollo de la aplicación.

3.1.2.1. Definición del escenario

El componente de backend recibirá datos de coordenadas provenientes de la aplicación móvil y almacenará en una base de datos para su posterior recuperación y consulta. La manera de conectarse a esta componente es por medio de servicios Rest usando el protocolo HTTP, este servicio deberá estar siempre disponible a través de Internet para su acceso mediante el componente móvil y web.

Se definió además la estructura de la base que almacenará la información. El diseño se muestra en la figura 23.

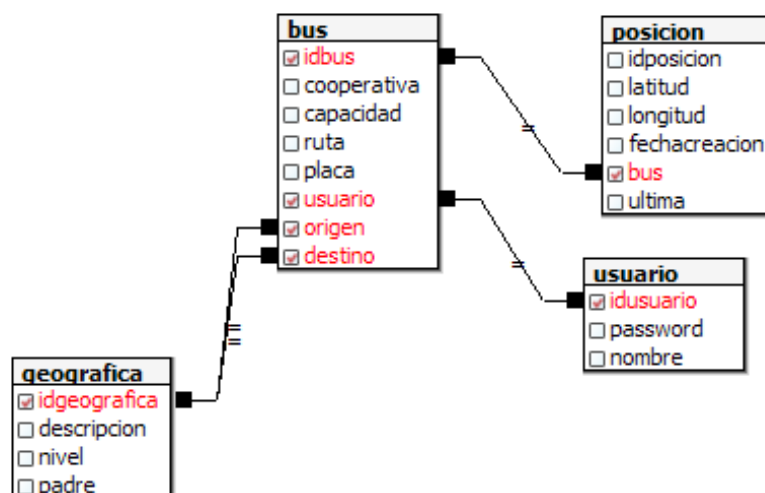


Figura 23 Diagrama de Base de Datos.
Fuente: El Autor

3.1.2.2. Estructuración del Software

En esta fase se identificó como interactúan los componentes móvil y web con el backend con las diferentes funcionalidades que debe exponer el API Rest.

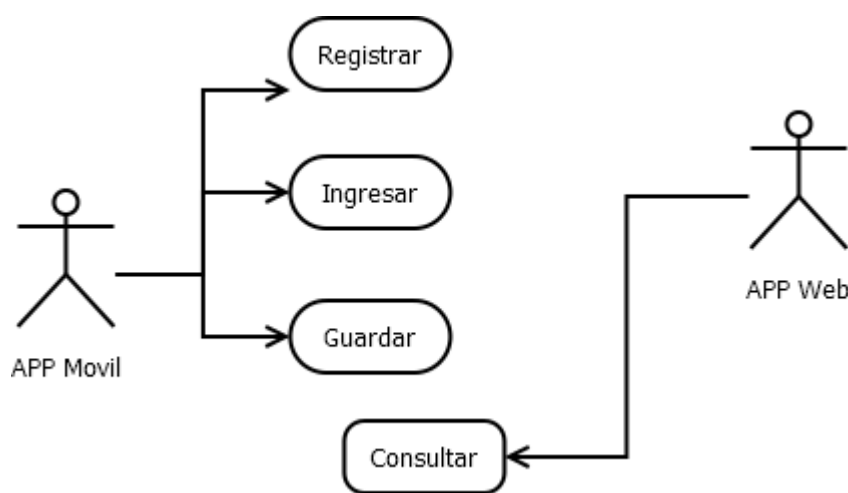


Figura 24 Casos de Uso Interacción Componentes
Fuente: El Autor

Tabla 4
Caso de Uso Registrar

Caso de Uso	Registrar
Actores	Front End APP Móvil
Tipo	Básico
Propósito	Acción para el registro de un nuevo usuario
Resumen	Desde la app móvil se llena un formulario de registro con sus credenciales y envía para la creación de sus credenciales de acceso al sistema a través de la llamada al servicio rest.
Precondiciones	El usuario debe ser un correo electrónico válido, se debe enviar de manera obligatoria el usuario y contraseña.
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario llena e formulario de registro y se envía desde la app móvil a través de servicio rest. 2. El usuario da click en el botón registrar e la app móvil 3. El usuario recibe un mensaje de confirmación.
Subflujos	No existen subflujos para este caso de uso
Excepciones	En caso de tener un usuario con las mismas credenciales se notificará al para que cambio.

Tabla 5
Caso de Uso Ingresar

Caso de Uso	Ingresar
Actores	Front End APP Móvil
Tipo	Básico
Propósito	Acción para la autenticación e ingreso a la aplicación
Resumen	El usuario llena un formulario de login en la app móvil con sus credenciales y envía para la autenticación en el sistema la misma que llama al servicio rest de autenticación.
Precondiciones	El usuario debe estar previamente registrado y activo
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario llena el formulario de login en la app móvil. 2. El usuario da click en el botón ingresar en la app móvil 3. Se valida las credenciales en la base de datos. 4. Si son correctas se envía un mensaje de respuesta por medio del servicio rest.
Subflujos	No existen subflujos para este caso de uso
Excepciones	En caso de tener ser credenciales incorrectas no se envía la información del usuario y desde la app móvil no se podrá acceder al sistema

Tabla 6
Caso de Uso Guardar

Caso de Uso	Guardar
Actores	Front End APP Móvil
Tipo	Básico
Propósito	Acción para el almacenamiento de las posiciones basados en el GPS del móvil
Resumen	El usuario inicia su sesión en la app móvil e inicia el servicio de localización el mismo que envía la

	ubicación basado en el GPS cada 3 segundos, estos datos se enviar al backend a través del servicio rest.
Precondiciones	Se debe ingresar al sistema con credenciales válidas, se debe contar con GPS en el dispositivo y mantenerlo activado.
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario de la app móvil ingresar a la app. 2. El usuario da click en el botón iniciar tracking 3. La app móvil empieza a enviar las diferentes posiciones a través del servicio rest.
Subflujos	No existen subflujos para este caso de uso
Excepciones	En caso de no tener cobertura o señal GPS en el móvil se mantendrá la última posición conocida

Tabla 7
Caso de Uso Consultar

Caso de Uso	Consultar
Actores	APP Web
Tipo	Básico
Propósito	Acción para consultar la posición de determinado bus de transporte
Resumen	Un usuario de la aplicación web ingresa a la misma y visualiza la posición en un mapa en tiempo real de los buses de la cooperativa Andina
Precondiciones	Se debe tener activos los dispositivos móviles de las unidades de transporte y los mismos deben estar en modo de tracking.
Flujo Principal	<ol style="list-style-type: none"> 1. El usuario ingresa a la página web. 2. El usuario ingresa a la sección del mapa. 3. La página web consulta la última posición, a través del servicio rest, cada 3 segundos para refrescar la misma en el mapa.
Subflujos	No existen subflujos para este caso de uso

Excepciones	En caso de no tener registros se mostrará la última posición conocida.
-------------	--

En la Figura 25 podemos observar el diagrama de clases del backend de la aplicación.

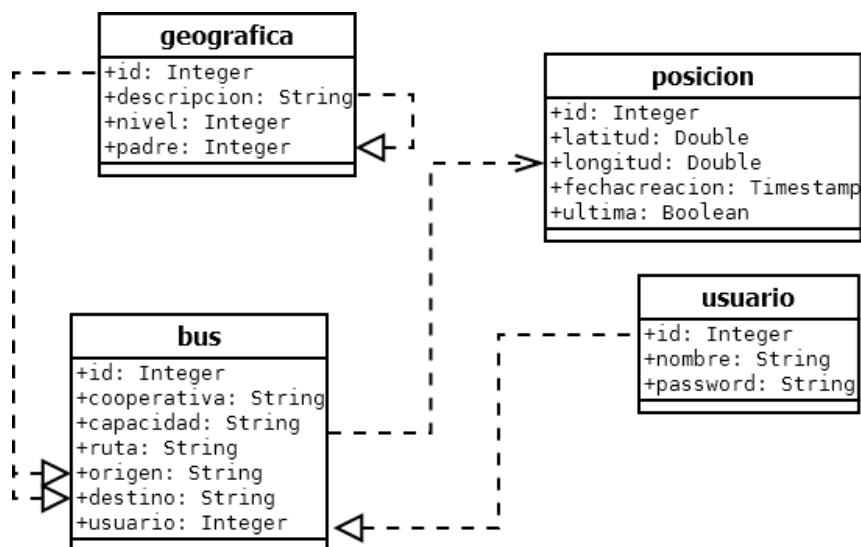


Figura 25 Diagrama de Clases Backend.
Fuente: El Autor

El diagrama de clases representa los recursos que el backend debe publicar en los servicios Rest y el diagrama de casos de uso representa el tipo de operación que se debe realizar sobre la base de datos. Basados en dichos diagramas se estableció la estructura que debe contemplar cada servicio del API Rest, misma que se muestra en la tabla 8.

Tabla 8
Estructura Servicios Rest

URL Servicio	Método HTTP	Parámetros de Entrada	Parámetros de Salida
/api/registro	POST	Nombre, Password	Usuario
/api/login	POST	Nombre, Password	Usuario
/api/localización	POST	Latitud, longitud, fechacreacion	Posición
/api/localización/:bus	GET	Latitud, longitud, fechacreacion, bus	Posición de determinado bus

/api/localización/	GET	Posición de todos los buses
--------------------	-----	-----------------------------

3.1.2.3. Definición de tiempos

En esta fase se estableció el tiempo que tomará desarrollar cada uno de los servicios que debe exponer el backend en base al análisis realizado en la fase de Obtención de Requerimientos. El tiempo estimado de desarrollo por cada servicio se muestra en la tabla 9.

Tabla 9
Tiempos estimados de desarrollo servicios Rest

Número	Requerimiento	Servicio	Tiempo de Desarrollo
1	Recibir las coordenadas del componente móvil y almacenarlos en la base de datos	/api/localización	2 d
2	Recibir los datos de registro de un nuevo usuario y almacenarlo en la base de datos	/api/registro	2 d
3	Recibir las credenciales de usuario y validar el mismo	/api/login	2 d
4	Entregar la última coordenada o posición de determinado vehículo	/api/localización/	2 d

3.1.2.4. Asignación de Recursos

En esta fase se definió los recursos que serán necesarios para el desarrollo del componente de backend. Esto incluye recursos tecnológicos, humanos y financieros.

Para el presente estudio se tomó en consideración los recursos humanos y tecnológicos, no se consideró los recursos financieros debido a que en esta

fase no es necesario aun realizar una inversión para el despliegue de la aplicación.

La tabla 10 muestra el detalle de recursos mínimos necesarios para el desarrollo del componente.

Tabla 10
Recursos requeridos para el desarrollo

Tipo de Recurso	Detalle	Cantidad
Analista Programador	Analista Programador con conocimientos de NodeJS	1
Computador	Memoria: 4GB Disco: 2 GB Libres Procesador: Core i5	1
Servidor, Hosting	Servidor VPS, Hosting con acceso mediante IP Publica	1

3.1.3. Desarrollo

En esta etapa se implementó el componente de backend basados en el diseño que se realizó en la etapa anterior. Para el desarrollo se usó el IDE Sublime Text 3 y la arquitectura sigue un patrón MVC como se muestra en la figura 25.

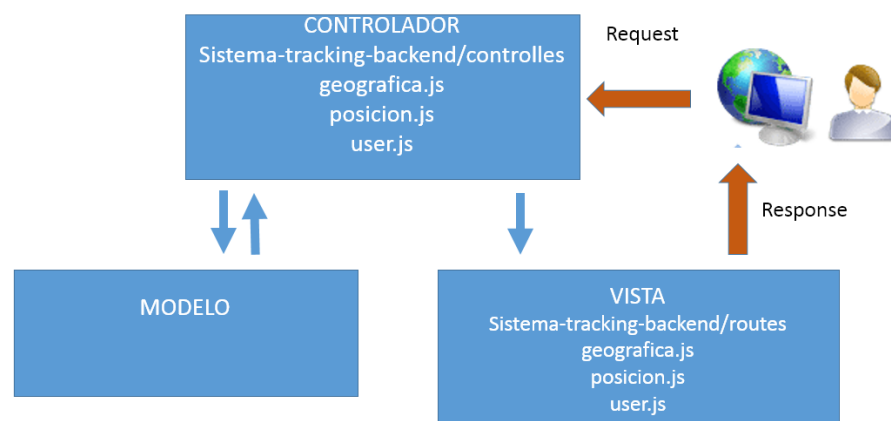


Figura 25 Arquitectura componente Backend
Fuente: El Autor

3.1.3.1. Codificación

Para la codificación del componente se usó el framework Express, dicho framework tiene como base a NodeJs por lo tanto se usó el lenguaje Javascript para la programación de cada servicio. Los servicios que expone el backend son de tipo Rest los mismos que serán llamados desde una URL.

La estructura del proyecto se basa en el estándar que usualmente se usa para los proyectos desarrollados con Express y consta de los siguientes elementos principales:

- **Controllers.**- Carpeta que contiene la lógica de programación, se encargan de la funcionalidad de determinados métodos o funciones.
- **Routes.** Carpeta que contiene las clases que publicarán las rutas del api Rest ya sea con métodos Post o Get.
- **Views.** Para el API Rest no se crearán vista o pantallas de usuario debido a que será un componetes de exposición de servicios web.

Se usó express-generator para la creación de la estructura del proyecto inicial para lo cual se realizó el siguiente proceso.

1. Se instaló express-generator con el comando `npm install express-generator -g`
2. Se creó el proyecto con el comando `express sistema-tracking-backend`

La estructura del proyecto creado se muestra en la figura 26:

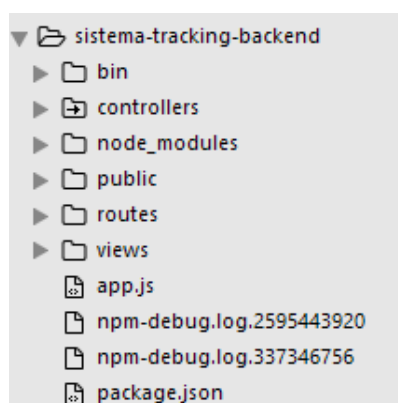


Figura 26 Estructura del proyecto
Fuente: El Autor

3.1.3.2. Verificación o comprobación

Una vez desarrollado el API Rest se realizaron las pruebas para probar la correcta funcionalidad de cada uno de los servicios, para esto se usó la

herramienta Postman, la misma que permite el consumo de los servicios Rest desde una interfaz amigable.

Las pruebas se realizaron configurando las llamadas desde una máquina local, antes de lo cual se ejecutó el servidor de backend programado para recibir estas llamadas a los diferentes servicios Rest, el mismo que escucha las peticiones en el puerto 3000.

Método POST

Acción Recibir las credenciales de usuario y validar el mismo

URL /api/login

Correcta SI

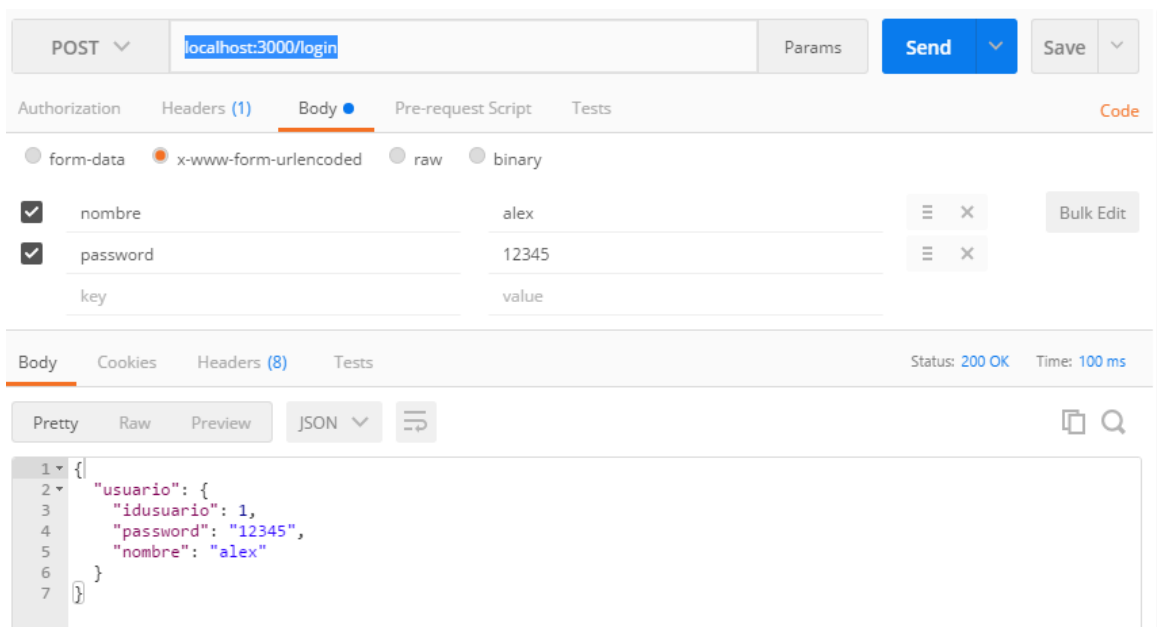


Figura 27 Prueba del servicio de login

Fuente: El Autor

La figura 27 muestra la comprobación del consumo del servicio web para validar el inicio de sesión de usuario, se envía a través de métodos POST y devuelve un objeto con los datos del mismo en caso de que se satisfactorio la validación de credenciales.

Método GET

Acción Entregar la última coordenada o posición de determinado vehículo

URL /api/ localización/

Correcta SI

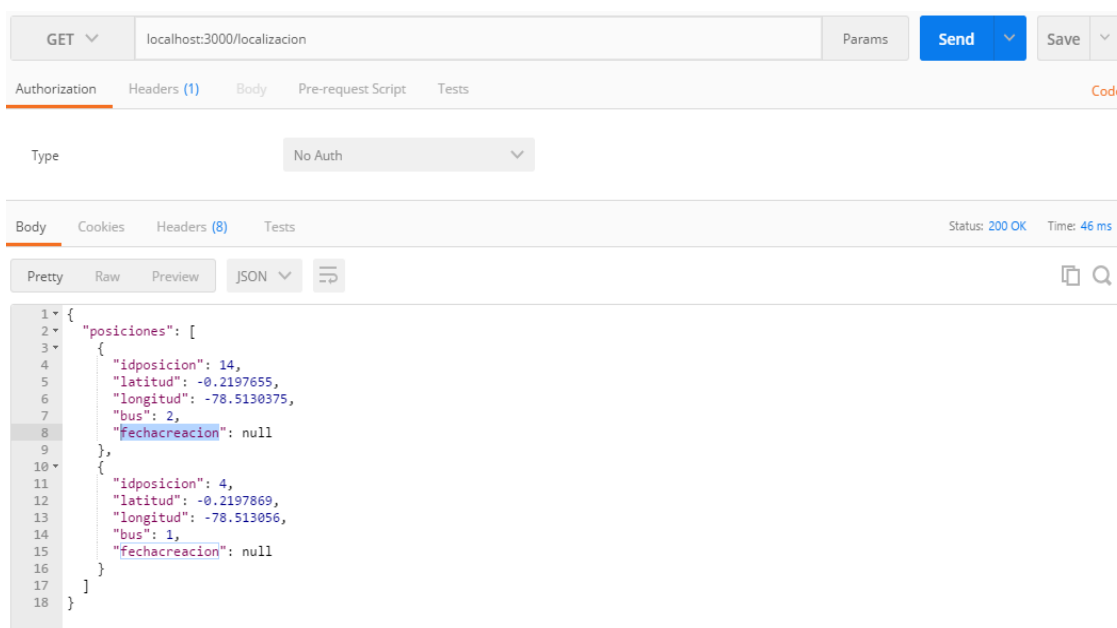


Figura 28 Prueba del servicio de localización

Fuente: El Autor

La figura 28 muestra la comprobación del consumo del servicio web para consulta de la localización de las unidades de transporte de la cooperativa Andina, el consumo para este servicio es a través del método GET.

Método POST

Acción Recibir las coordenadas del componente móvil y almacenarlos en la base de datos

URL /api/ localización/

Correcta SI

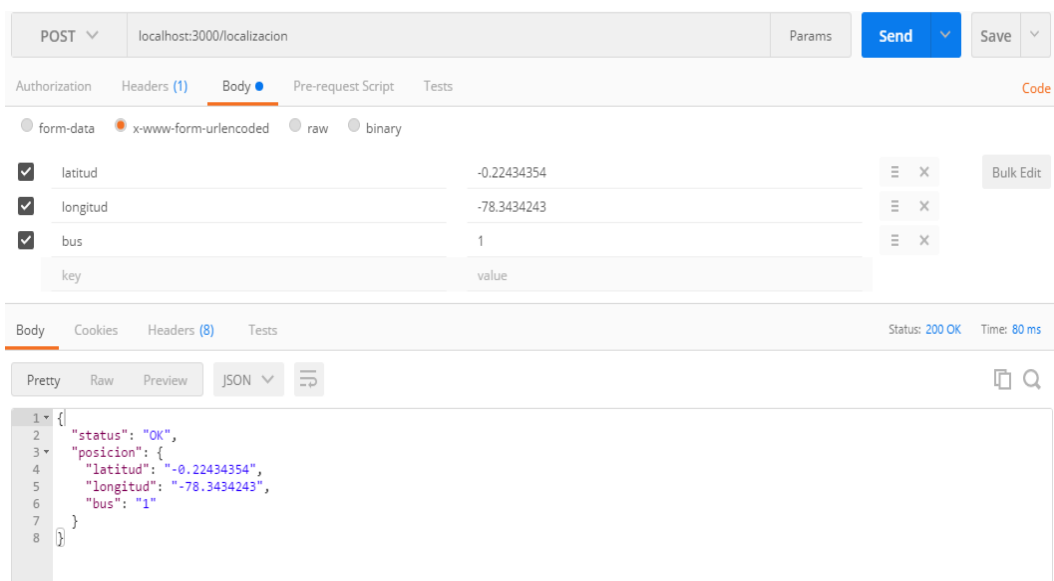


Figura 29 Prueba del servicio de almacenamiento
Fuente: El Autor

La figura 29 muestra la comprobación del servicio para el almacenamiento de la posición de una unidad de transporte, el método se lo consume a través del método POST y se envía como cuerpo la latitud, longitud y el identificador de la unidad de transporte específica de la cooperativa Andina.

Método POST

Acción Recibir los datos de registro de un nuevo usuario y almacenarlo en la base de datos

URL /api/registro/

Correcta SI

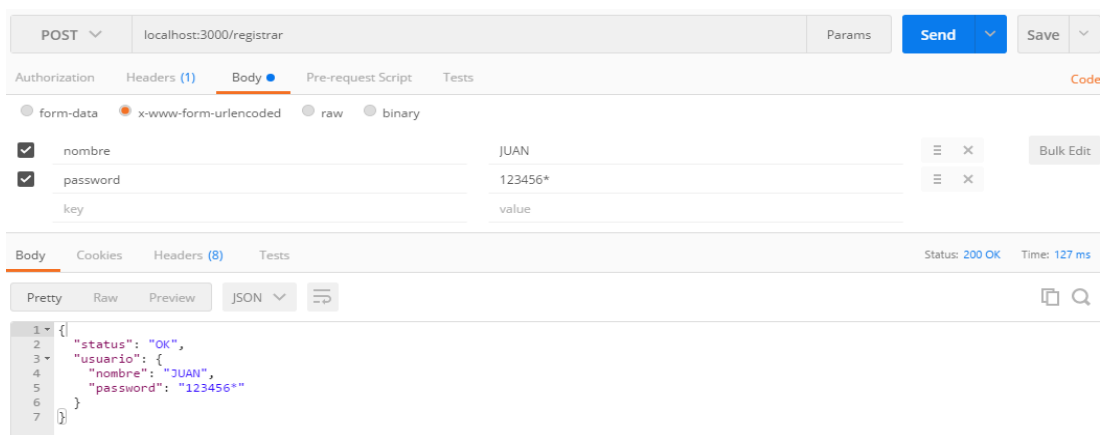


Figura 30 Prueba del servicio registro de usuario
Fuente: El Autor

La figura 30 muestra la comprobación del para el proceso de registro de un usuario nuevo, el consumo se lo realiza a través del método POST y como cuerpo del mensaje se envían un nombre de usuario y la clave.

3.1.4. Verificación o comprobación

En esta etapa se probará el sistema desde un servidor al que se pueda acceder desde el Internet, es decir, debe ser publicado para simular el ambiente real de la aplicación. Para estos se usó un servicio de Hosting con un servidor VPS dicho servidor cuenta con una IP pública que permite el acceso desde Internet y facilita la publicación de los servicios web contenidos en el componente backend. El servidor VPS cuenta con las características que se muestran en la tabla 11:

Tabla 11
Características Servidor de Pruebas

Componente	Detalle
Sistema Operativo	Centos 6
Nodejs	7.0.0
Memoria Ram	4 GB
Disco Duro	20 GB
Procesador	

3.1.4.1. Emulación y simulación

En esta fase de las pruebas pretende validar si la solución funciona de manera correcta en un ambiente casi real para lo cual se subió el proyecto al servidor VPS y se ejecutó el mismo se encuentra publicado en la IP <http://31.220.57.146> y el puerto 3000. En este servidor se alojará el backend que permitirá la recepción de las ubicaciones de las unidades de transporte.

Los servicios expuestos se los volvió a evaluar con la herramienta Postman para verificar su correcto funcionamiento. En la siguiente tabla se adjunta las URL que deben ser consumidas por el componente móvil.

Tabla 12
Urls Servicio de Pruebas

Petición	Servicio	URL Pública
POST	/api/localización	http://31.220.57.146:3000/localización
POST	/api/registro	http://31.220.57.146:3000/registro
POST	/api/login	http://31.220.57.146:3000/login

3.1.5. Entrega

3.1.5.1. Distribución

La aplicación por ser desarrollada en Nodejs será distribuida en carpeta sin compilación y subida al servidor respectivo.

3.2. Desarrollo del componente móvil

Para el componente móvil se diseñará una aplicación con el framework Ionic 1, es decir, será una aplicación híbrida lo que permitirá que la misma sea empaquetada para cualquier sistema operativo móvil. Para el presente estudio y basado en el alcance planteado se realizarán las pruebas sobre el sistema operativo Android.

El componente móvil se encargará de enviar las posiciones capturadas con el GPS y las enviará a la base de datos por medio del API Rest como vemos en la figura 31.

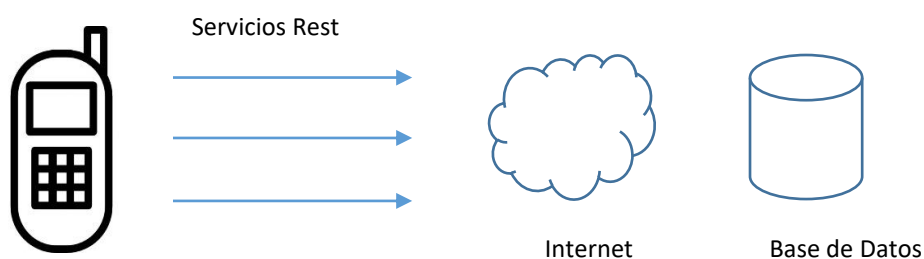


Figura 31 Interacción componente Móvil
Fuente: El Autor

3.2.1. Análisis

En esta etapa se plantea los requerimientos que el componente móvil debe cumplir para el desarrollo de la plataforma.

3.2.1.1. Obtención de los requerimientos.

En esta fase se identificaron las funcionalidades que debe cumplir el componente móvil

- Ingresar al sistema.
- Registrar un nuevo usuario
- Capturar la coordenada del GPS y enviarla al API.
- Realizar capturar periódicas de las coordenadas.

3.2.1.2. Clasificación de los requerimientos

En esta fase se clasificó los requerimientos, y se identificó cuál de ellos deben tener interfaz de usuario y cuales son de funcionalidad sin necesidad de interfaz de usuario.

Tabla 13
Clasificación de los requerimientos

Número	Requerimiento	GUI
1	Ingresar al sistema	SI
2	Registrar un nuevo usuario	SI
3	Capturar la coordenada del GPS y enviarla al API	SI
4	Realizar capturar periódicas de las coordenadas	NO

3.2.2. Diseño

En esta etapa se diseñarán los diagramas o esquemas de la aplicación y a diferencia del componente anterior en esta fase se incluirán los diseños de las pantallas de usuario que serán incluidas en el componente móvil.

3.2.3. Definición del escenario

En esta fase se identificó los requerimientos que debe mantener la aplicación para poder conectarse al API Rest, debido a que el API va a ser accedido a través del internet es necesario que el móvil cuente con acceso a datos o WIFI, es decir, debe considerarse el funcionamiento en un entorno **Conectado**.

3.2.4. Estructuración del Software

En esta fase se diseñó los diagramas de casos de uso y diseño preliminar de pantallas que formarán parte de la aplicación móvil.

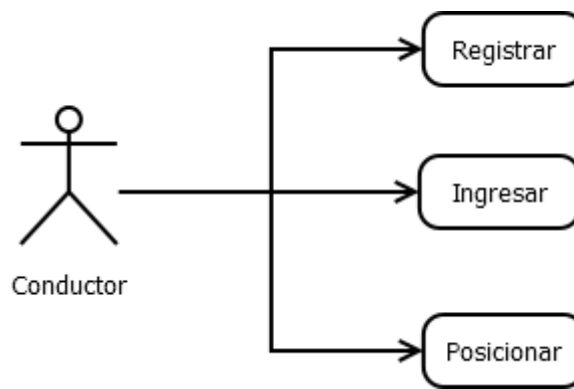


Figura 32 Casos de uso aplicación móvil
Fuente: El Autor

Para el desarrollo de la aplicación móvil es necesaria diseñar las interfaces que verá el usuario. Para esto se diseñaron las pantallas a manera de diseños de prototipo basados en el levantamiento de requerimientos en donde se identificó cuáles de los mismos deben contar con una interfaz de usuario.

Los diseños preliminares fueron maquetados en la herramienta Balsamiq Mockups los mismos que se muestran en la figura 33, 34 y 35:

The image shows a smartphone screen with a registration form. The title is "Registro". Below the title are four input fields: "Placa de Bus", "Cooperativa", "Capacidad Bus", and "Ruta". At the bottom of the form is a blue button labeled "Registrar".

Figura 33 Pantalla de Registro
Fuente: El Autor

The image shows a smartphone screen with a login form. It has two input fields: "Usuario" and "Clave". Below the fields are two buttons: a blue "Login" button and a red "Registrarse" button.

Figura 34 Pantalla de Login
Fuente: El Autor

The image shows a smartphone screen with a service start form titled "Posicionamiento". It contains five input fields: "Conductor", "Fecha y Hora Salida (Automático)" (with a calendar icon), "Origen", and "Destino". At the bottom is a red button labeled "Iniciar Servicio".

Figura 35 Pantalla de Inicio de Servicio Tacking
Fuente: El Autor

3.2.5. Definición de tiempos

En esta fase se estableció el tiempo que tomará desarrollar cada una de las interfaces de usuario y la comunicación con el backend. El tiempo de desarrollo por cada servicio se muestra en la tabla 14.

Tabla 14
Definición de tiempos

Número	Requerimiento	Tiempo
1	Ingresar al sistema	3 d
2	Registrar un nuevo usuario	2 d
3	Capturar la coordenada del GPS y enviarla al API	2 d
4	Realizar capturar periódicas de las coordenadas	2 d

3.2.6. Asignación de Recursos

En esta fase se determinó que recursos son necesarios para el desarrollo del proyecto, contemplo recursos tanto tecnológicos como recursos humanos. La tabla 15 muestra lo requerido para el desarrollo:

Tabla 15
Características Ambiente Producción

Tipo de Recurso	Detalle	Cantidad
Analista Programador	Analista Programador con conocimientos de NODEjs, IONIC, Angularjs	1
Computador	Memoria: 4GB Disco: 2 GB Libres Procesador: Core i5	1
Servidor, Hosting	Servidor VPS, Hosting con acceso mediante IP	1

3.2.7. Desarrollo.

En esta etapa se implementó el componente móvil basado en el diseño que se realizó en la etapa anterior. Para el desarrollo se usó el IDE Sublime Text 3.

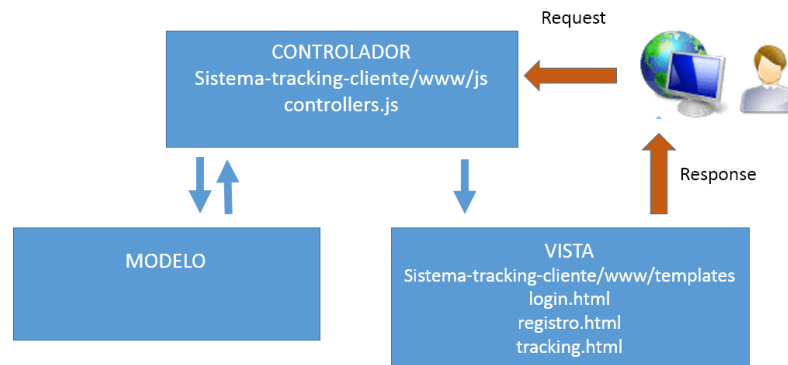


Figura 36 Arquitectura componente Móvil
Fuente: El Autor

3.2.8. Codificación

Para el desarrollo del componente móvil se usó el framework IONIC, el lenguaje de programación usado fue javascript y el IDE Sublime Text.

Para la generación del proyecto se usó el comando:

- ionic start sistema-tracking-backend blank

Adicionalmente en este componente se incluyeron librerías que permitan acceder a los dispositivos GPS de los teléfonos móviles. El plugin que permite acceder a estas funcionalidades es cordova-geolocation y se instala con el siguiente comando:

- cordova plugin add cordova-plugin-geolocation

La estructura del proyecto se muestra en la figura 37, una vez creado el proyecto mediante consola:

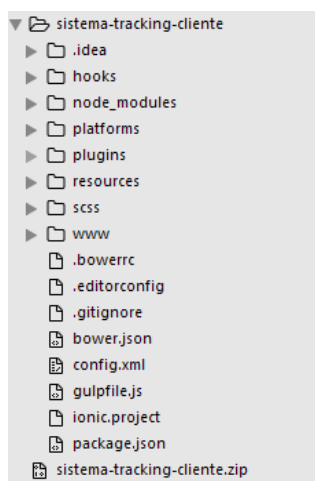


Figura 37 Estructura del proyecto móvil
Fuente: El Autor

3.2.9. Verificación o comprobación

3.2.9.1. Emulación y Simulación

En esta etapa se realizó la verificación de la funcionalidad del componente móvil en un simulador de dispositivos Android. El simulador que se usó para verificar el funcionamiento es Genymotion debido a que esta herramienta es ligera, permite la instalación de varios modelos de dispositivos y se la puede lanzar desde el framework Ionic.

Las pruebas de funcionamiento de las realizó para dos modelos específicos de smartphones, como se muestra en la figura 38:

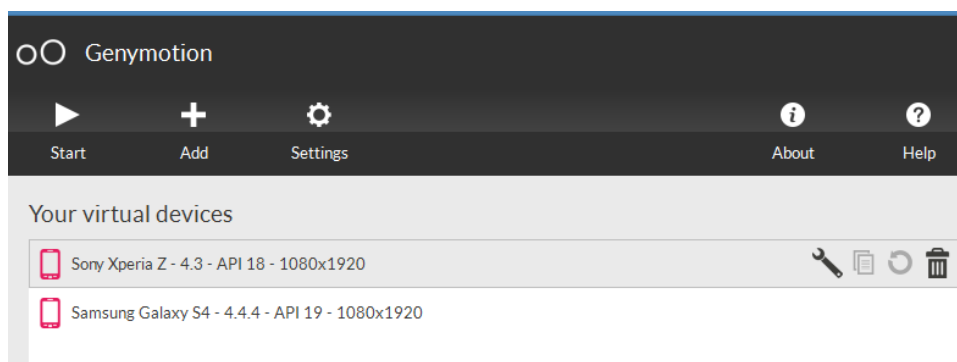


Figura 38 Simulador Genymotion
Fuente: El Autor

Para lanzar la aplicación en el emulador se usó el comando:

- ionic run android

Una vez que la aplicación se ejecutó en el emulador se validó su funcionalidad. Los resultados de las pruebas fueron satisfactorios se lanzó el emulador y se la aplicación cliente se comunicó con el backend correctamente

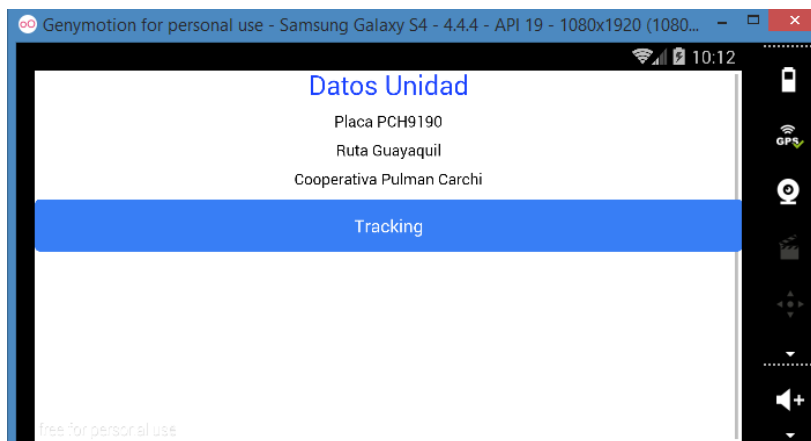


Figura 39 Ejecución APP en Emulador
Fuente: El Autor

3.2.9.2. Dispositivos Reales

Para la prueba con dispositivos reales se compiló la aplicación para los sistemas Android y se instaló en un dispositivo real. Para el empaquetamiento de la aplicación móvil se usó el comando:

- ionic build android

Las pruebas se las realizó en un smarphone Sony Xperia Z, se instaló la app, se ejecutó y se comprobó que se comunicue con el backend y envié la posición del GPS.

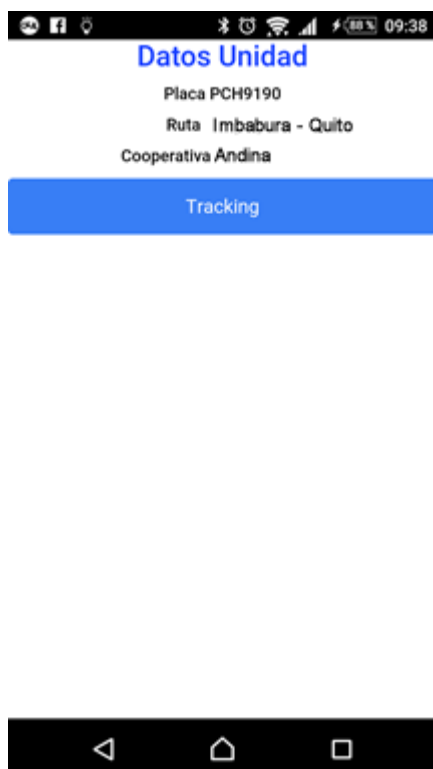


Figura 40 Imagen APP Capturada en el móvil
Fuente: El Autor

```
(node:7906) DeprecationWarning: Using Buffer without `new` will soon st
alloc() ` instead.
OPTIONS /localizacion 200 5.986 ms - 13
Entra a post
{ latitud: -0.2197662, longitud: -78.5131773, bus: 2 }
POST /localizacion 200 40.255 ms - 80
OPTIONS /localizacion 200 1.587 ms - 13
Entra a post
{ latitud: -0.2197662, longitud: -78.5131773, bus: 2 }
POST /localizacion 200 3.179 ms - 80
Entra a post
{ latitud: -0.21967497, longitud: -78.5130436, bus: 2 }
POST /localizacion 200 1.053 ms - 81
Entra a post
{ latitud: -0.21967497, longitud: -78.5130436, bus: 2 }
POST /localizacion 200 0.891 ms - 81
Entra a post
{ latitud: -0.21967497, longitud: -78.5130436, bus: 2 }
POST /localizacion 200 1.171 ms - 81
Entra a post
{ latitud: -0.21967636, longitud: -78.51304877, bus: 2 }
POST /localizacion 200 2.253 ms - 82
Entra a post
{ latitud: -0.21967636, longitud: -78.51304877, bus: 2 }
POST /localizacion 200 1.055 ms - 82
```

Figura 41 Recepción de posición en el backend
Fuente: El Autor

3.2.10. Entrega

3.2.10.1. Distribución

La aplicación se distribuirá compilada como un APK para que pueda ser instalada en los Smartphone con sistema Android.

3.3. Desarrollo del componente web

El componente web servirá para mostrar a los usuarios la ubicación de los buses de transporte, este componente se conectará al API Rest, obtendrá las ubicaciones finales de los buses y las dibujará en un mapa.

Al conectarse al mismo API Rest que se conecta la aplicación móvil la ubicación será en tiempo real y se programará para que cada cierto tiempo se actualice con la última coordenada geográfica y se pueda evidenciar el movimiento de cada unidad de transporte.

Adicionalmente al hacer click en cada icono que representa la unidad de transporte el componente web permitirá visualizar datos informativos complementarios:

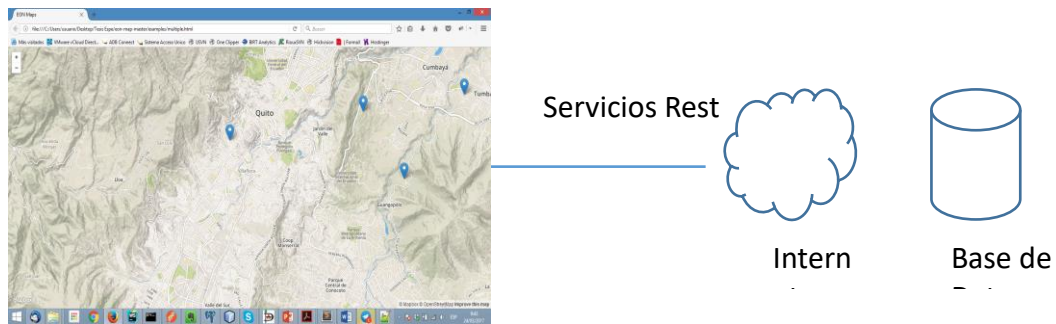


Figura 42 Interacción componente web
Fuente: El Autor

Para el desarrollo del componente WEB se hará uso de la metodología UWE que es un estándar UML que se orienta al desarrollo WEB los diagramas requeridos por la metodología se diseñarán en herramienta MagicUWE en su versión trial (Nomagic, 2017) .

A continuación, se describen cada una de las etapas y modelos de la metodología UWE orientadas al presente estudio:

3.3.1. Modelos de Requerimiento

En esta etapa se identificó el rol del usuario y las acciones que puede ejecutar dentro de la aplicación web. Al ser un visor en el que se muestra un mapa básicamente tendría dos acciones principales:

- Ingreso al Sistema
- Visualización de Mapa

La figura 43 muestra el diagrama de casos de uso del componente web:

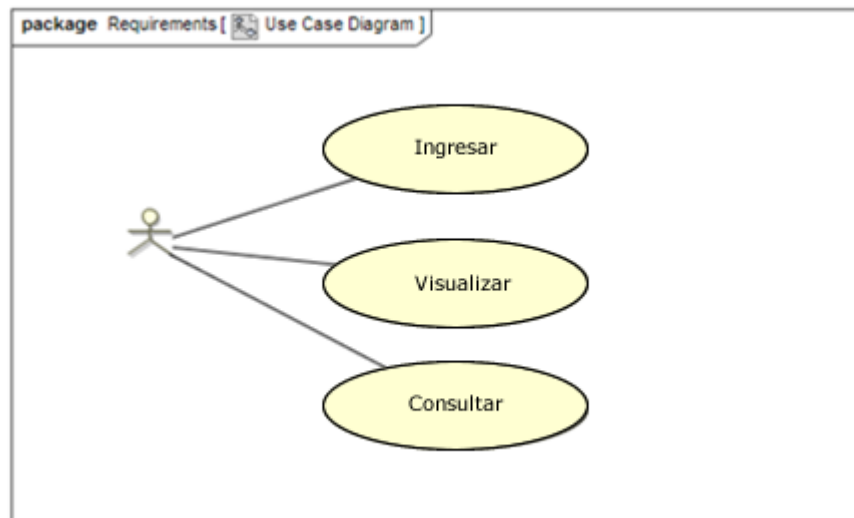


Figura 43 Diagrama de Casos de Uso Componente Web
Fuente: El Autor

3.3.2. Modelo de Contenido

En esta etapa del modelo se representa la información del dominio mediante diagrama de clases de UML. La Figura 44 muestra el diagrama de clases del componente Web basados en el diagrama de casos de uso:

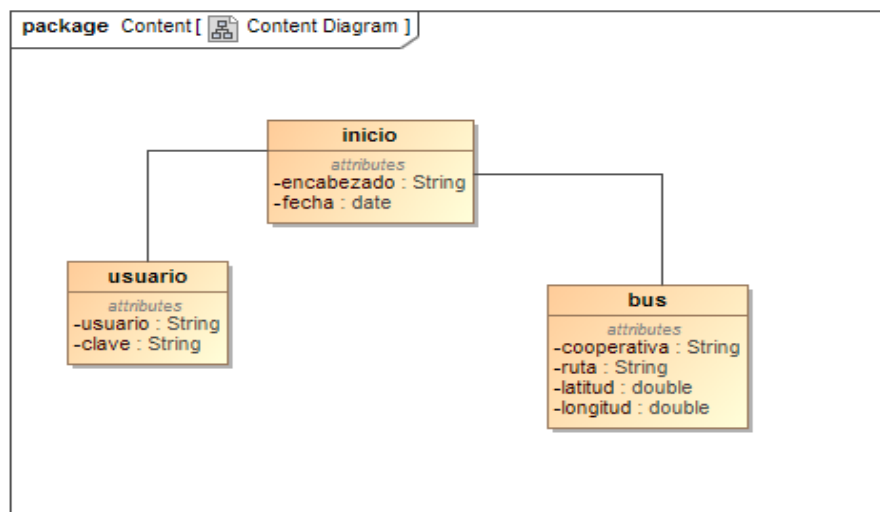


Figura 44 Diagrama de Contenido Componente Web
Fuente: El Autor

3.3.3. Modelo de Navegación

En esta fase se representa las acciones y rutas de navegación de cada nodo del sistema, en el presente estudio el diagrama representa las rutas que

sigue el sistema en las acciones de Ingreso y Consulta de Detalle de las unidades de transporte.

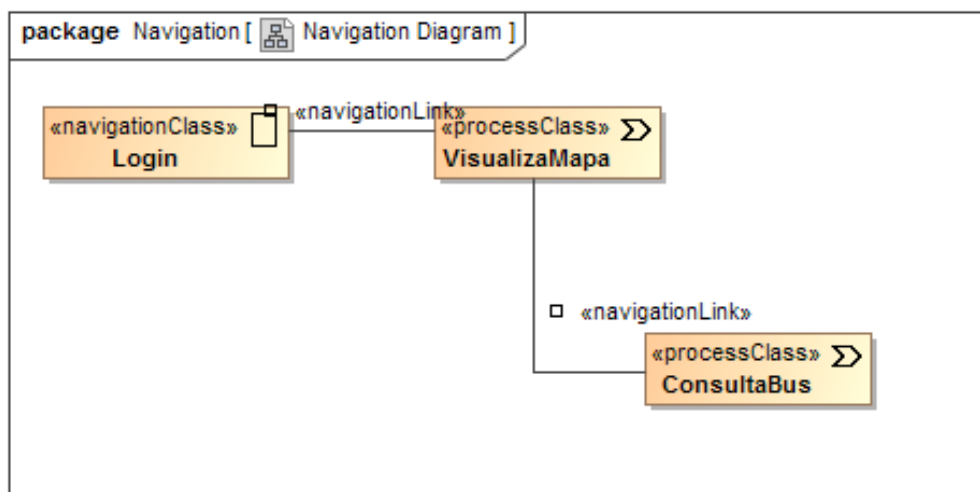


Figura 45 Diagrama Navegación Componente Web
Fuente: El Autor

3.3.4. Modelo de Presentación

En esta fase se diseñó el diagrama de navegación que mostrará una descripción básica de las interfaces de usuario del componente web y de los mensajes q mostrar en cada una de ellas.

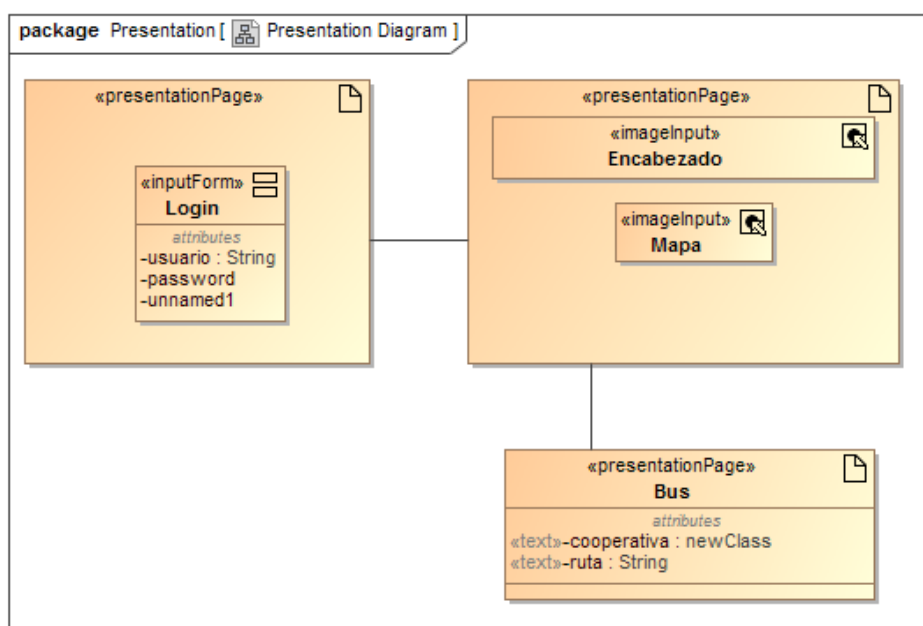


Figura 46 Modelo de Presentación Componente Web
Fuente: El Autor

3.3.5. Modelo de Procesos

Para el presente estudio el modelo de procesos identifica como sería el flujo de la aplicación y que visualizará el usuario al ejecutar las acciones que estarán disponibles en la página Web.

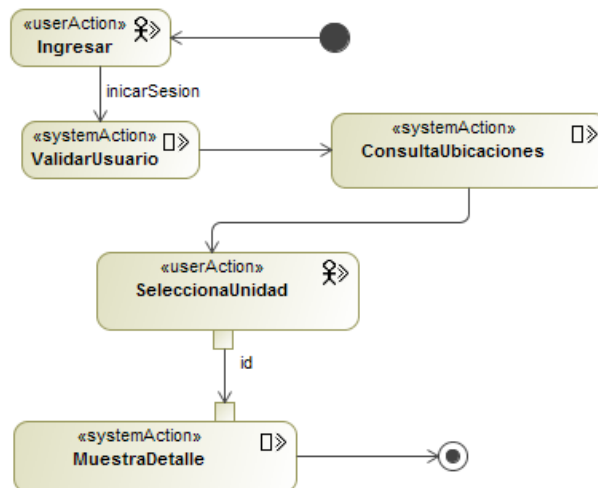


Figura 47 Modelo de Procesos Componente Web
Fuente: El Autor

Con los modelos realizados se procedió a la codificación de la aplicación Web, es decir, a diseñar las interfaces, las acciones y la navegación que debe permitir la página. En la aplicación web uno de los componentes principales es el Mapa, para el presente estudio se usará Mapbox.

Para usar Mapbox es necesario registrarse y obtener un token que permita el uso del API del mapa, eso se lo realizó en la página web <https://www.mapbox.com>. Una vez obtenido el token se verificó la funcionalidad y el despliegue correcto del mapa mediante el código de ejemplo que se muestra en la página.

El código de ejemplo se lo modificó para que el API del mapa se conecte al web service del componente de backend y muestre la posición del componente móvil en tiempo real.



Figura 48 Mapbox
Fuente: El Autor

Una vez que se realizó la funcionalidad del mapa se implementó los procesos de login y consultar individual. El diseño final del componente web se muestra en la figura 49:

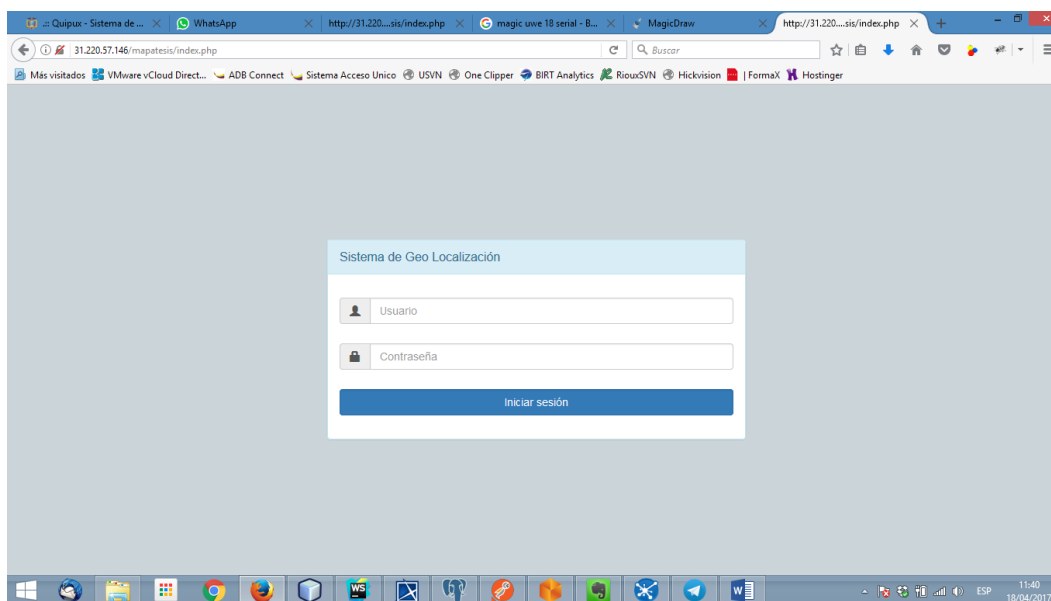


Figura 49 Login componente Web
Fuente: El Autor

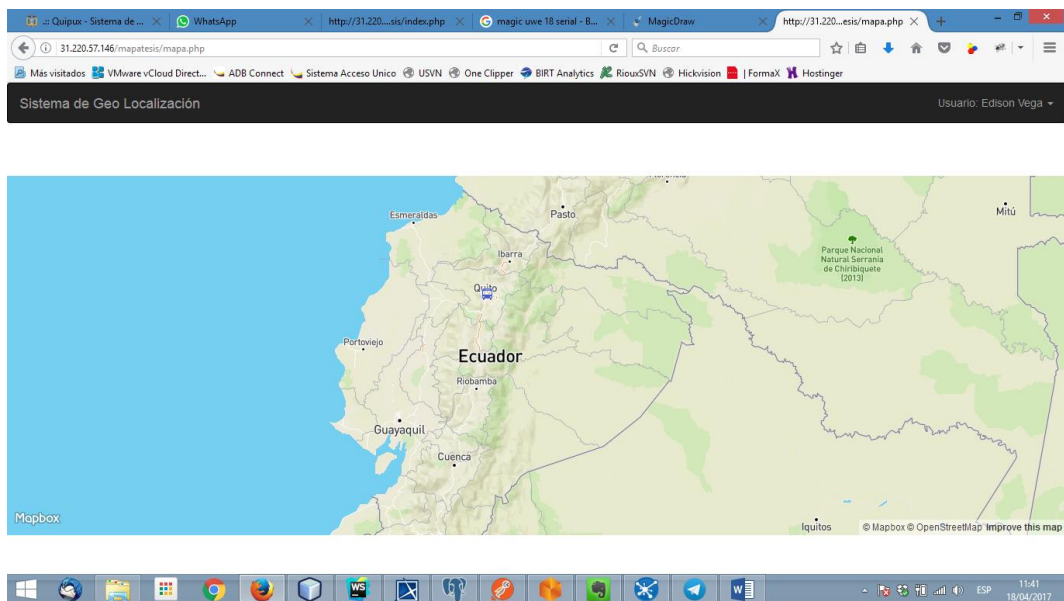


Figura 50 Pantalla inicial componente Web
Fuente: El Autor

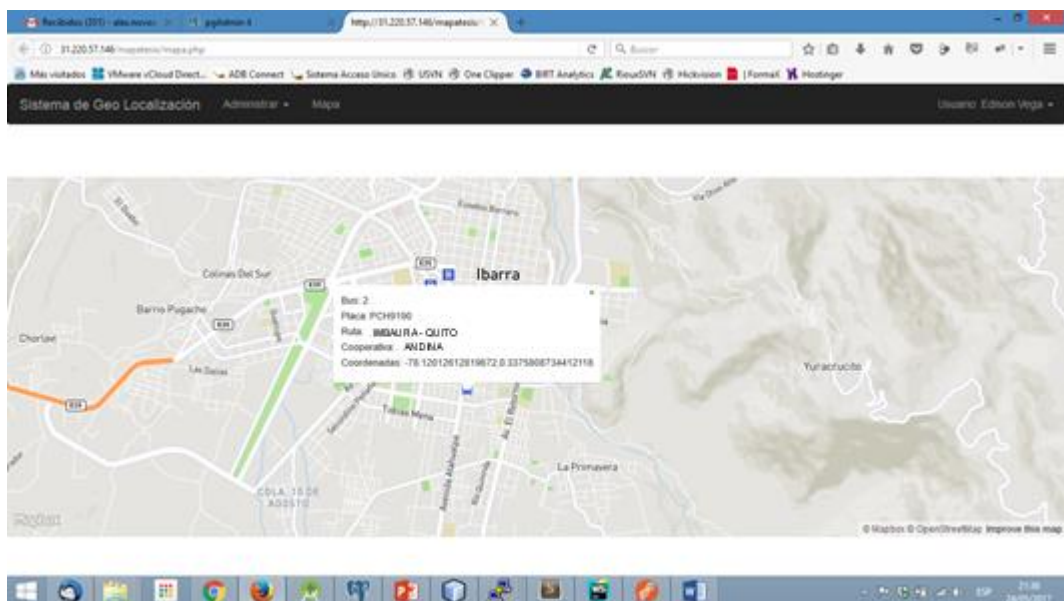


Figura 51 Consulta vehículo componente Web
Fuente: El Autor

CAPÍTULO 4. IMPLEMENTACIÓN PILOTO

4.1. Implementación Backend

El componente backend fue desarrollado con el framework Express por lo que su ejecución depende de Nodejs. Para la implementación de este componente se instaló un servidor privado virtual en un servicio de hosting con las siguientes herramientas de software:

- Sistema Operativo Linux Centos 6.5
- NodeJs versión 7
- Base de datos Postgres 7.2

El código fuente de la aplicación se lo ubicó en la siguiente ruta:

- /opt/nodeapp/sistema-tracking-backend/

Para la configuración en producción fue necesaria la configuración de la aplicación como servicio de Linux para evitar caídas al momento de reiniciar el servidor Linux, esto se logró con el paquete PM2.

Para la ejecución del componente de backend se realizó con el siguiente comando:

- pm2 ./bin/www

```
Use `pm2 show <id|name>` to get more details about an app
[root@pruebas sistema-tracking-backend]# pm2 list
```

App name	id	mode	pid	status	restart	uptime	cpu	mem	watching
www	0	fork	9127	online	0	3D	0%	25.2 MB	disabled

```
Use `pm2 show <id|name>` to get more details about an app
[root@pruebas sistema-tracking-backend]#
```

Figura 52 Ejecución backend
Fuente: El Autor

4.2. Implementación Componente Móvil

El componente móvil se instaló en un smarphone con sistema operativo Android, la aplicación previamente probada en los simuladores se empaquetó haciendo uso del framwork Ionic y se la instaló en el dispositivo móvil. El comando usado para el empaquetamiento es:

- ionic build android

Debido a que no es una fuente que se encuentra en la tienda de aplicaciones oficiales de Google se debió activar en el teléfono la opción de **Confiar en Fuentes Externas**.

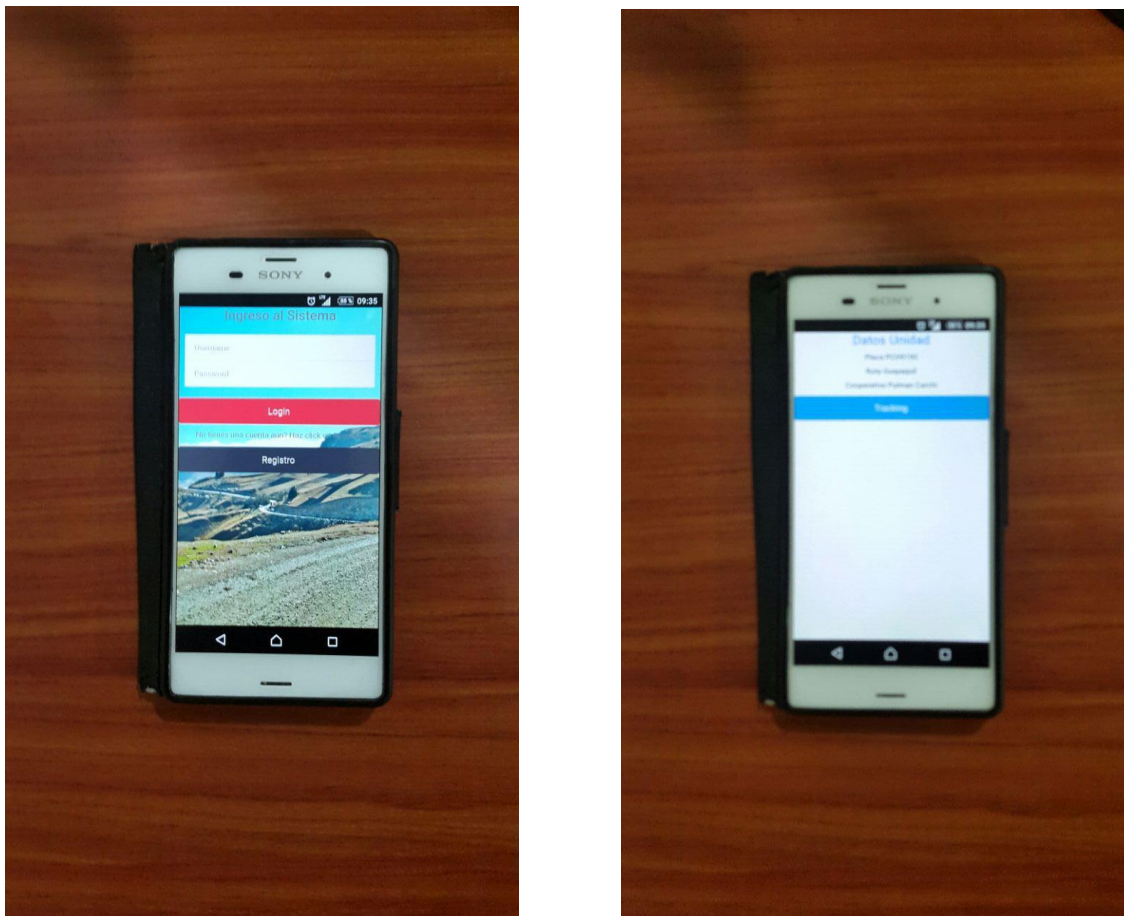


Figura 53 Ejecución componente móvil
Fuente: El Autor

4.3. Implementación Componente Web

El componente web se instaló en un servidor privado virtual, dicho componente se programó en el lenguaje PHP por lo que fue necesaria la instalación de un servidor web apache. Las herramientas usadas se mencionan a continuación.

- Sistema Operativo Linux Centos 6.5
- Apache 2.4

Para el inicio del servidor se ejecutó el comando:

- `service httpd start`

Una vez probado el servidor, se instaló las fuentes en la carpeta definidas por defecto por el sistema operativo centos 6.5, la ubicación de la carpeta se menciona a continuación:

- var /www/html

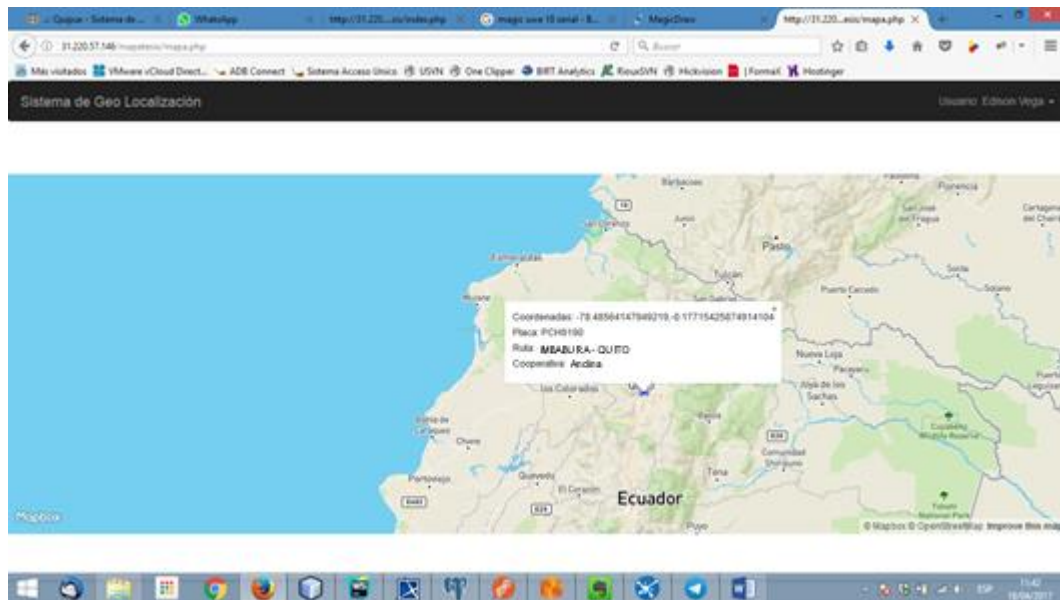


Figura 54 Ejecución componente web
Fuente: El Autor

4.4. Verificación o comprobación de campo

Las pruebas de campo son validaciones que se realizó sobre todos los componentes de la aplicación en un ambiente real, para esto una vez implementado todos los componentes se realizaron las siguientes acciones:

- Registro de nuevo usuario
- Login en la aplicación
- Tracking de ubicación

Estas acciones fueron realizadas desde un dispositivo móvil con la aplicación instalada y el componente web y backend funcionando en un servidor con IP pública. La prueba principal consistió en la ejecución de la aplicación desde un Smartphone ubicado en un vehículo y simular un viaje o movilización del mismo. Una vez iniciado el servicio de tracking se validó el correcto envío de posiciones al backend para lo cual se habilitó el log en el servidor para poder visualizar los mensajes recibidos y enviados.

```

0 | www | Cooperativa: 'PULMAN',
0 | www | placa: 'PCH9190' } ]
0 | www | GET /localizacion/ 304 6.581 ms - -
0 | www | Entra a post
0 | www | { latitud: -0.3435623, longitud: -78.4385578, bus: 2 }
0 | www | POST /localizacion 200 0.915 ms - 80
0 | www | Entra a post
0 | www | { latitud: -0.3435623, longitud: -78.4385578, bus: 2 }
0 | www | POST /localizacion 200 0.685 ms - 80
0 | www | Entra a post
0 | www | { latitud: -0.3435623, longitud: -78.4385578, bus: 2 }
0 | www | POST /localizacion 200 0.932 ms - 80
0 | www | [ anonymous {
0 | www | idposicion: 4833,
0 | www | latitud: -0.3435623,
0 | www | longitud: -78.4385578,
0 | www | bus: 2,
0 | www | fechacreacion: null,
0 | www | ruta: 'GUAYAQUIL - QUITO',
0 | www | cooperativa: 'PULMAN',
0 | www | placa: 'PCH9190' } ]
0 | www | GET /localizacion/ 304 6.419 ms - -
0 | www | [ anonymous {
0 | www | idposicion: 4833,
0 | www | latitud: -0.3435623,
0 | www | longitud: -78.4385578,
0 | www | bus: 2,
0 | www | fechacreacion: null,
0 | www | ruta: 'GUAYAQUIL - QUITO',
0 | www | cooperativa: 'PULMAN',
0 | www | placa: 'PCH9190' } ]
0 | www | GET /localizacion/ 304 6.312 ms - -

```

Figura 55 Log Backend Puebas Reales
Fuente: El Autor

De manera simultánea se validó que las posiciones sean recogidas por el componente web y visualizadas en el mapa, para esto se verificó que se presente el icono correctamente y el mismo muestre la última posición en tiempo real.

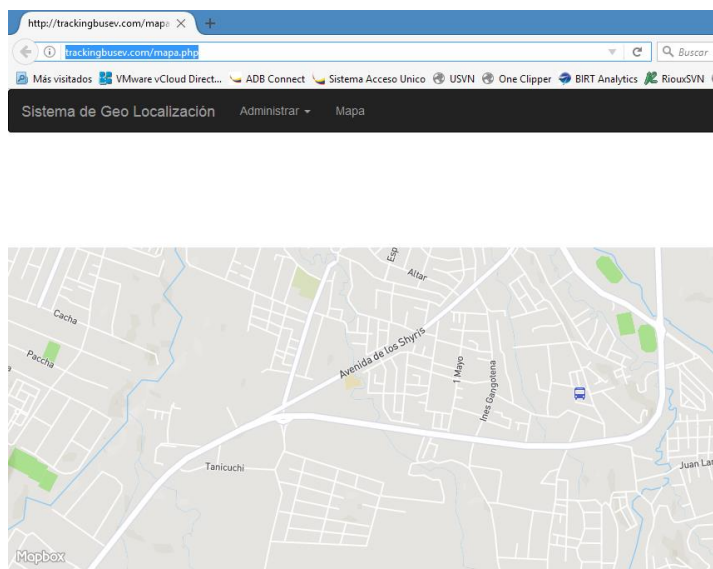


Figura 56 Visualización mapa componente web
Fuente: El Autor

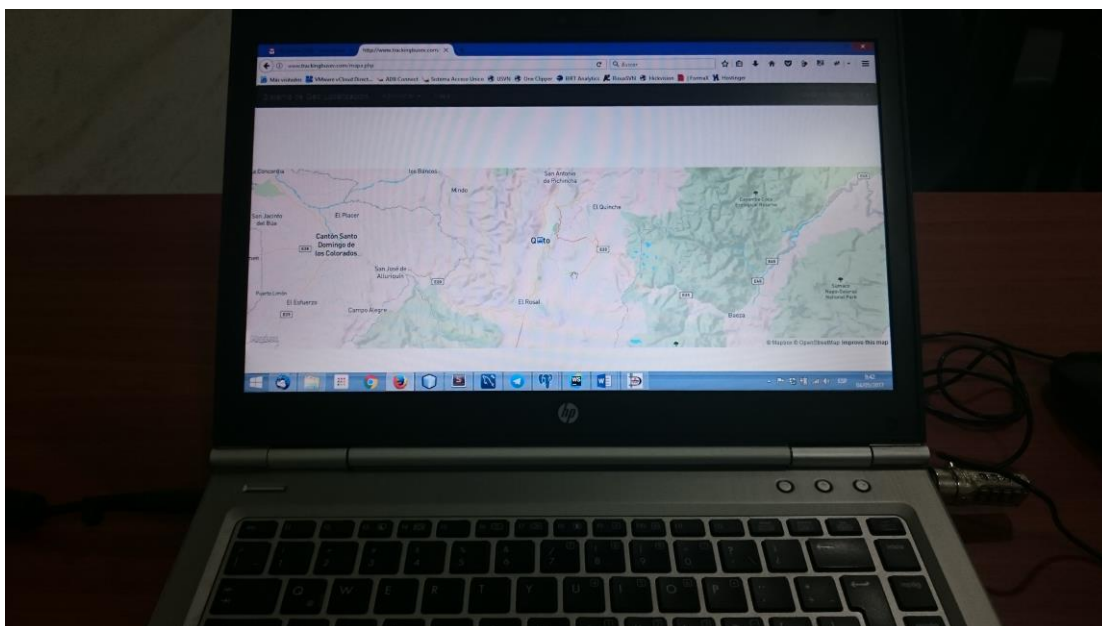


Figura 57 Visualización mapa componente web real
Fuente: El Autor

Sistema de Geo Localización				
Administrar	Mapa	Horarios	Usuario: Edison Vega	
Horarios y rutas				
Conductor	Ruta	Hora de salida	Hora de llegada	# Pasajeros
Andrés Pérez	Quito - Ibarra	13:00	15:30	30
Santiago Bustamante	Quito - Otavalo	10:00	12:00	23
Pedro Cevallos	Ibarra - Quito	18:00	20:30	40
Rodrigo Ortiz	Otavalo - Quito	16:00	18:00	34
				Imprimir

Figura 58 Visualización de la pantalla de horarios y rutas
Fuente: El Autor

El componente web se alojó en un servicio de hosting que incluyó un dominio para que sea más fácil acceder al mismo al tener que recordar no una dirección IP sino un nombre, para el presente estudio se usó el nombre de dominio www.trackingbusev.com.

El proceso de configuración y adquisición del servicio de hosting se muestra en el anexo 2.

Adicionalmente se vio la necesidad de especificar las características del servicio y de los diferentes componentes para lo cual se hizo uso de una página web en la que se especificó esta información esta página es la página inicial o Landing Page la misma que se muestra a continuación:



Figura 59 Landing Page inicial
Fuente: El Autor

La Landing Page cuenta con diferentes secciones en cada una de ellas se especificó las características y ciertas secciones promocionales del servicio. Las diferentes secciones se muestran en las siguientes ilustraciones:

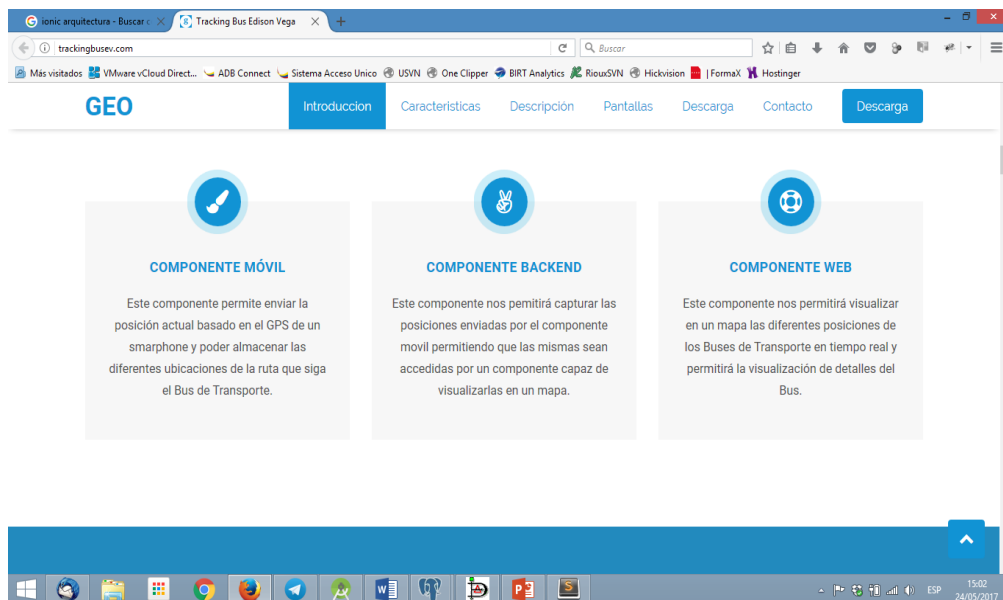


Figura 60 Landing Page introducción
Fuente: El Autor



Figura 61 Landing Page características generales
Fuente: El Autor

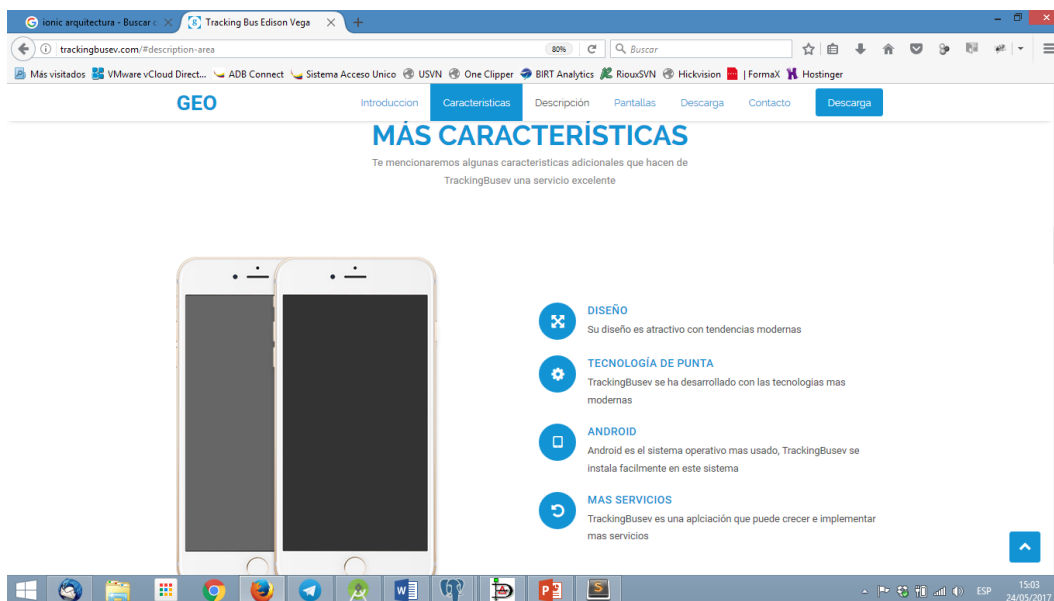


Figura 62 Landing Page características específicas
Fuente: El Autor

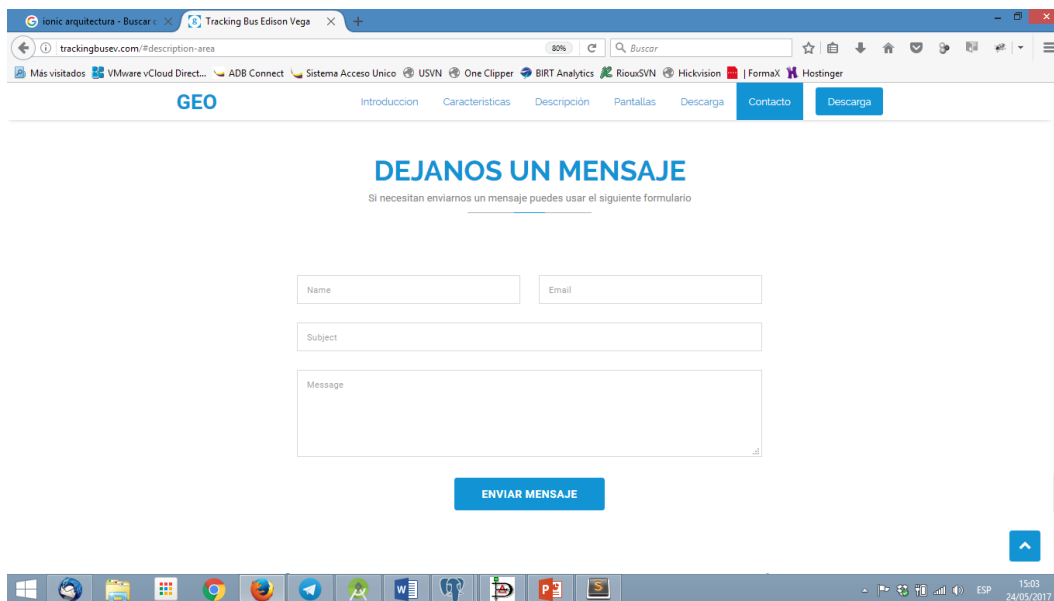


Figura 63 Landing Page características contacto
Fuente: El Autor

CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- La plataforma propuesta en el presente estudio es una alternativa de solución de bajo costo y fácil implementación para la ubicación de las unidades de transporte interprovincial.
- Se desarrolló un componente de backend que permite la recepción y difusión de las coordenadas de ubicación tomados de la aplicación móvil.
- Se desarrolló la aplicación móvil que permite la captura de las coordenadas de ubicación del GPS de un smarphone y las envía al backend para su posterior almacenamiento.
- Se desarrolló la aplicación web que recupera la última coordenada y dibuja la posición en un mapa.
- Las metodologías de desarrollo permiten construir plataformas de software de una manera ordenada e incluyen todas las fases para garantizar el éxito del proyecto.

5.2. Recomendaciones

- Continuar el desarrollo de la plataforma y poder incluir mayores funcionalidades a cada componente de la plataforma desarrollada en el presente estudio, así como también poder sincronizar la visualización de más de un vehículo al mismo tiempo.
- Ampliar el soporte para otro tipo de dispositivos móviles con diferente sistema operativo para futuros estudios y pruebas.
- La aplicación móvil depende en gran medida de la cobertura y señal de las operadoras de telefonía móvil sea esta movistar, claro o cnt, por lo que cuando no se cuente con cobertura no se podrá recibir los datos de ubicación, se recomienda identificar las zonas en las que no hay dicha cobertura para establecer futuros mecanismos de identificación de la posición.

BIBLIOGRAFÍA

- Arcgis. (2016, 10 19). Retrieved from <http://desktop.arcgis.com/es/arcmap/10.3/manage-data/raster-and-images/what-is-raster-data.htm>
- Citlali Guadalupe Nieves Guerrero, J. P. (n.d.). *UWE en Sistema de Recomendación de Objetos de Aprendizaje. Aplicando Ingeniería Web: Un Método en Caso de Estudio*.
- Egotrack. (2016, 10 18). Retrieved from <http://www.egotrack.net/es/teoria.html>
- Institute for Informatics. (2017, 04 10). *UWE Tutorial*. Retrieved from <http://uwe.pst.ifi.lmu.de/teachingTutorialSpanish.html>
- Leonard Richardson, S. R. (2007). *RESTful Web Services*. O'Reilly.
- Maira Cecilia Gasca Mantilla, L. L. (2013, 08 27). Metodologías para el desarrollo de aplicaciones Mviles. Colombia.
- Nodejs. (2017, 03 23). *Nodejs*. Retrieved from Nodejs: <https://nodejs.org/es/>
- Nomagic. (2017, 04 18). <https://www.nomagic.com/products/magicdraw>. Retrieved from <https://www.nomagic.com/products/magicdraw>
- restysoap.blogspot.com. (2017, 04 27). Retrieved from <http://restysoap.blogspot.com/2015/10/normal-0-21-false-false-false-es-pe-x.html>
- SGS. (2016, 10 25). Retrieved from <http://www.sgs.edu.sk/HTML/gps.htm>
- Universidad de Alicante. (2017, 04 27). *Introducción a los Servicios Web*. Retrieved from <http://www.jtech.ua.es/j2ee/publico/servc-web-2012-13/sesion01-apuntes.html>
- users.dcc.uchile.cl*. (2017, 11 11). Retrieved from <https://users.dcc.uchile.cl/~psalinas/uml/casosuso.html>
- uwe.pst.ifi.lmu.de*. (2017, 11 9). Retrieved from [uwe.pst.ifi.lmu.de: http://uwe.pst.ifi.lmu.de/teachingTutorialProcessSpanish.html](http://uwe.pst.ifi.lmu.de/teachingTutorialProcessSpanish.html)
- Ventajasdesventajas.com. (2016, 10 21). Retrieved from <http://www.ventajasdesventajas.com/gps>
- Wikineos. (2016, 10 25). Retrieved from http://es.wikineos.com/wiki/Manual:Mapas_digitales
- Wikipedia. (2016, 10 25). Retrieved from https://es.wikipedia.org/wiki/Topolog%C3%ADa_geoespacial
- Wikipedia. (2016, Octubre 15). Retrieved from https://es.wikipedia.org/wiki/Sistema_de_informaci%C3%B3n_geogr%C3%A1fica
- Wikipedia. (2016, 10 25). Retrieved from <https://es.wikipedia.org/wiki/Cartograf%C3%ADa>
- Wikipedia. (2016, 10 25). Retrieved from <https://es.wikipedia.org/wiki/Ret%C3%ADcula>
- Wikipedia. (2017, 04 27). Retrieved from https://es.wikipedia.org/wiki/Dominio_de_Internet
- Wilken, J. (2016). *Ionic in action*. NY: Manning.
- www.gaurav.mobi*. (2017, 11 9). Retrieved from [www.gaurav.mobi: https://www.gaurav.mobi/blog/hybrid-vs-native-apps](http://www.gaurav.mobi)

www.nodehispano.com. (2017, 11 9). Retrieved from *www.nodehispano.com*:
<http://www.nodehispano.com/page/9/>

www.researchgate.net. (2017, 11 11). Retrieved from
https://www.researchgate.net/figure/282613841_fig14_Fig-2-Diagrama-de-casos-de-uso-del-sistema-BibliotecaNFC

(2017, 05 02). Retrieved from <https://diccionarioactual.com/sistema-de-posicionamiento-global/>

(2017, 05 02). Retrieved from <https://apoyovial.net/2016/09/07/ventajas-y-desventajas-del-uso-del-gps/>