

UNIVERSIDAD NACIONAL DE CAJAMARCA
FACULTAD DE INGENIERÍA
ESCUELA ACADÉMICO PROFESIONAL
DE INGENIERÍA DE SISTEMAS



**“DESARROLLO DE UNA APLICACIÓN WEB PARA EL
CONTROL Y MONITOREO EN TIEMPO REAL QUE PERMITA
MEJORAR EL SERVICIO DEL TRANSPORTE INTERNO DE LA
UNIVERSIDAD NACIONAL DE CAJAMARCA”**

TESIS:

PARA OBTENER EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

AUTOR:

Bach. ORDOÑEZ MARÍN CÉSAR AUGUSTO

ASESOR:

Dr. EDWIN VALENCIA CASTILLO

Ingeniero de Sistemas

CAJAMARCA-PERÚ

JULIO 2019

ÍNDICE GENERAL

ÍNDICE GENERAL	ii
ÍNDICE DE TABLAS	vii
ÍNDICE DE FIGURAS	ix
RESUMEN	xii
ABSTRACT	xiii
CAPÍTULO I. INTRODUCCIÓN	1
CAPÍTULO II. MARCO TEÓRICO	4
2.1 ANTECEDENTES TEÓRICOS DE LA INVESTIGACIÓN	4
2.1.1 INTERNACIONALES.....	4
2.1.2 NACIONALES.....	5
2.2 BASES TEÓRICAS	6
2.2.1 SERVICIO DE TRANSPORTE PÚBLICO.....	6
2.2.2 SISTEMA DE POSICIONAMIENTO GLOBAL.....	7
2.2.3 RECEPTOR GPS.....	8
2.2.4 SISTEMA DE MONITOREO VEHICULAR.....	8
2.2.5 COMUNICACIÓN BIDIRECCIONAL EN TIEMPO REAL.....	10
2.2.6 NODE.JS.....	10
2.2.7 SOCKET.IO.....	11
2.2.8 APLICACIÓN WEB.....	11
2.2.9 PROCESO DE DESARROLLO DE SOFTWARE.....	12
2.3 DEFINICIÓN DE TÉRMINOS BÁSICOS	17
CAPÍTULO III. MATERIALES Y MÉTODOS	18
3.1 PROCEDIMIENTO	18
3.1.1 VISIÓN DEL PROYECTO.....	18
3.1.1.1 DESCRIPCIÓN DEL PROYECTO.....	18
3.1.1.1.1 OPORTUNIDAD DE MEJORA.....	18
3.1.1.1.2 PROBLEMAS EN EL PROCESO.....	19
3.1.2 DESCRIPCIÓN DE USUARIOS.....	20
3.1.2.1 DEFINICIÓN DE USUARIOS.....	20
3.1.2.2 PERFILES DE USUARIO.....	20
3.1.2.3 AMBIENTES DE LA APLICACIÓN WEB.....	20
3.1.3 VISTA GENERAL DE LA APLICACIÓN WEB.....	21

3.1.3.1	PERSPECTIVA DE LA APLICACIÓN WEB.....	21
3.1.3.2	BENEFICIOS DE LA APLICACIÓN WEB.....	21
3.1.3.3	DEPENDENCIAS DE LA APLICACIÓN WEB.....	22
3.1.3.4	RECURSOS DEL SOFTWARE Y HARDWARE.....	22
3.1.4	REQUERIMIENTOS DE LA APLICACIÓN WEB.....	22
3.1.4.1	REQUISITOS FUNCIONALES.....	23
3.1.4.2	REQUISITOS NO FUNCIONALES.....	27
3.1.4.2.1	REQUISITOS DE RENDIMIENTO.....	27
3.1.4.2.2	SEGURIDAD.....	27
3.1.4.2.3	MANTENIBILIDAD.....	27
3.1.4.2.4	ARQUITECTURA.....	27
3.1.5	CASOS DE USO.....	27
3.1.5.1	DIAGRAMA DE CASOS DE USO.....	28
3.1.5.1.1	PERFILES DE USUARIO.....	28
3.1.5.1.2	DESCRIPCIÓN DE ACTORES.....	28
3.1.5.2	PAQUETES DE CASOS DE USO.....	29
3.1.5.3	DESCRIPCIÓN DEL CASO DE USO.....	30
3.1.6	DIAGRAMA DE ACTIVIDADES.....	47
3.1.6.1	DIAGRAMA DE ACTIVIDADES DEL PAQUETE FRONT – END.....	47
3.1.6.1.1	INGRESAR A LA APLICACIÓN WEB.....	47
3.1.6.1.2	SELECCIONAR PARADERO.....	47
3.1.6.1.3	CÓMO LLEGAR AL PARADERO.....	48
3.1.6.1.4	CAMBIAR DE PARADERO.....	48
3.1.6.1.5	MONITOREO DEL BUS.....	49
3.1.6.1.6	VER RUTA DEL BUS.....	49
3.1.6.1.7	SELECCIONAR DESTINO.....	50
3.1.6.2	DIAGRAMA DE ACTIVIDADES DEL PAQUETE BACK – END.....	50
3.1.6.2.1	ENVÍO DE COORDENADAS AL SERVIDOR.....	50
3.1.6.2.2	ENVÍO DE COORDENADAS AL USUARIO.....	51
3.1.6.2.3	RECIBIR COORDENADAS.....	51
3.1.6.2.4	CALCULO DE DATOS DE MONITOREO.....	52
3.1.7	MATRIZ DE TRAZABILIDAD.....	53
3.1.8	MODELO CONCEPTUAL.....	54
3.1.8.1	DIAGRAMA DE ANÁLISIS DE CLASES.....	54
3.1.8.2	DIAGRAMA DE CLASES DE DISEÑO.....	55
3.1.8.3	DIAGRAMA DE SECUENCIA DE SUCESOS DEL SISTEMA.....	55
3.1.8.3.1	DIAGRAMA DE SECUENCIA INGRESAR A LA APLICACIÓN.....	56
3.1.8.3.2	DIAGRAMA DE SECUENCIA SELECCIONAR PARADERO.....	57
3.1.8.3.3	DIAGRAMA DE SECUENCIA CÓMO LLEGAR AL PARADERO.....	57
3.1.8.3.4	DIAGRAMA DE SECUENCIA CAMBIAR DE PARADERO.....	57

3.1.8.3.5	DIAGRAMA DE SECUENCIA MONITOREO DE BUS	58
3.1.8.3.6	DIAGRAMA DE SECUENCIA VER RUTA DEL BUS	58
3.1.8.3.7	DIAGRAMA DE SECUENCIA SELECCIONAR DESTINO	59
3.1.8.3.8	DIAGRAMA DE SECUENCIA ENVÍO DE COORDENADAS AL SERVIDOR 59	
3.1.8.3.9	DIAGRAMA DE SECUENCIA RECIBIR COORDENADAS.....	60
3.1.8.3.10	DIAGRAMA DE SECUENCIA CALCULO DE DATOS DE MONITOREO ...	60
3.1.9	ANÁLISIS DE LA SOLUCIÓN	61
3.1.9.1	VIABILIDAD TÉCNICA.....	61
3.1.9.2	VIABILIDAD ECONÓMICA.....	63
3.1.10	ANÁLISIS COSTO – BENEFICIO.....	63
3.1.11	DISEÑO DEL SISTEMA	64
3.1.11.1	ARQUITECTURA DE LA SOLUCIÓN.....	64
3.1.11.2	REPRESENTACIÓN DE LA ARQUITECTURA.....	64
3.1.11.3	ARQUITECTURA DE CAPAS O NIVELES	66
3.1.11.4	DIAGRAMA DE COMPONENTES.....	67
3.1.11.5	DISEÑO DE BASE DE DATOS.....	68
3.1.11.5.1	BASE DE DATOS RELACIONAL.....	68
3.1.12	PROTOTIPOS.....	69
3.1.12.1	PANTALLAS DEL SISTEMA	69
3.1.12.1.1	PANTALLA DE BIENVENIDA A LA APLICACIÓN WEB	69
3.1.12.1.2	PANTALLA PRINCIPAL DE LA APLICACIÓN.....	70
3.1.12.1.3	PANTALLA SELECCIONAR PARADERO	71
3.1.12.1.4	PANTALLA MAPA	72
3.1.12.1.5	PANTALLA VENTANA EMERGENTE OPCIONES DE PARADERO	74
3.1.12.1.6	PANTALLA VENTANA EMERGENTE CAMBIAR DE PARADERO	75
3.1.12.1.7	PANTALLA CÓMO LLEGAR AL PARADERO	77
3.1.12.1.8	PANTALLA VENTANA EMERGENTE MONITOREO DEL BUS	78
3.1.12.1.9	PANTALLA VER RUTA DEL BUS.....	78
3.1.12.1.10	PANTALLA SELECCIONAR DESTINO	80
3.1.13	DESARROLLO DEL SISTEMA.....	81
3.1.13.1	CAPA ACCESO A DATOS	81
3.1.13.2	CAPA DE NEGOCIO O LÓGICA.....	81
3.1.13.3	CAPA DE PRESENTACIÓN.....	84
3.1.14	INSTALACIÓN DE LA APLICACIÓN WEB.....	119
3.1.15	EVALUACIÓN Y PRUEBAS.....	126
3.1.15.1	EVALUACIÓN	126
3.1.15.2	PRUEBAS	127
3.1.15.3	MEDICIÓN DEL TRAFICO.....	131

3.2 TRATAMIENTO Y ANÁLISIS DE DATOS Y PRESENTACIÓN DE RESULTADOS	132
3.2.1 RESULTADO DE LA INVESTIGACIÓN	132
3.2.1.1 RESULTADOS DE LA ENCUESTA PARA EL INDICADOR TIEMPO.....	132
3.2.1.2 RESULTADOS DE LA ENCUESTA PARA EL INDICADOR COSTOS	133
<i>CAPÍTULO IV ANÁLISIS Y DISCUSIÓN DE RESULTADOS.....</i>	135
4.1 ANÁLISIS DE RESULTADOS	135
4.1.1 INDICADOR: TIEMPO PROMEDIO DE ESPERA DE LA COMUNIDAD UNIVERSITARIA POR EL BUS UNIVERSITARIO.....	135
4.1.1.1 CONTRASTACIÓN DE LA HIPÓTESIS	135
4.1.1.2 FORMULACIÓN DE LA HIPÓTESIS ESTADÍSTICA	140
4.1.1.3 NIVEL DE SIGNIFICANCIA.....	140
4.1.1.4 VALORES ESTADÍSTICOS DEL PROCEDIMIENTO	140
4.1.1.5 ESTABLECER REGIÓN CRÍTICA	141
4.1.1.6 TOMA DE DECISIÓN.....	141
4.1.2 INDICADOR: COSTO QUE ASUMEN LA COMUNIDAD UNIVERSITARIA AL OPTAR POR UN TRANSPORTE ALTERNATIVO PAGADO AL BUS UNIVERSITARIO	142
4.1.2.1 FORMULACIÓN DE LA HIPÓTESIS ESTADÍSTICA	146
4.1.2.2 NIVEL DE SIGNIFICANCIA.....	146
4.1.2.3 VALORES ESTADÍSTICOS DEL PROCEDIMIENTO	147
4.1.2.4 ESTABLECER REGIÓN CRÍTICA	147
4.1.2.5 TOMA DE DECISIÓN.....	148
4.2 DISCUSIÓN DE RESULTADOS.....	148
4.2.1 INTERPRETACIÓN DE LA INFORMACIÓN	148
4.2.2 SIMILITUDES ENTRE RESULTADOS OBTENIDOS EN LA INVESTIGACIÓN Y LOS ANTECEDENTES TEÓRICOS	149
<i>CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES.....</i>	152
5.1 CONCLUSIONES	152
5.2 RECOMENDACIONES	154
<i>REFERENCIAS BIBLIOGRÁFICAS</i>	155
<i>ANEXOS.....</i>	158
ANEXO 1: CRONOGRAMA DE TESIS	158
ANEXO 2: ENCUESTA A LOS USUARIOS DE LOS BUSES UNIVERSITARIOS.....	159

ANEXO 3: DATOS RECABADOS PRE TEST.....	161
ANEXO 4: DATOS RECABADOS POST TEST.....	165
ANEXO 5: VALIDACIÓN DEL INSTRUMENTO POR UN ESTADÍSTICO	169
ANEXO 6: SOLICITUD DE VALIDACIÓN DIRIGIDA A EXPERTOS EN DESARROLLO DE SOFTWARE.....	171

ÍNDICE DE TABLAS

<i>Tabla 1: Definición del problema</i>	19
<i>Tabla 2: Perfiles de usuarios</i>	20
<i>Tabla 3: Beneficios de la aplicación web</i>	22
<i>Tabla 4: Ingresar a la aplicación</i>	23
<i>Tabla 5: Seleccionar paradero</i>	23
<i>Tabla 6: Como llegar al paradero</i>	24
<i>Tabla 7: Cambiar de paradero</i>	24
<i>Tabla 8: Monitoreo del bus</i>	24
<i>Tabla 9: Ver ruta del bus</i>	25
<i>Tabla 10: Seleccionar destino</i>	25
<i>Tabla 11: Envío de coordenadas al servidor</i>	25
<i>Tabla 12: Envío de coordenadas a los usuarios</i>	26
<i>Tabla 13: Calculo de datos de monitoreo</i>	26
<i>Tabla 14: Usuario</i>	26
<i>Tabla 15: Actor conductor</i>	28
<i>Tabla 16: Actor usuario</i>	28
<i>Tabla 17: Actor administrador</i>	29
<i>Tabla 18: Caso de uso ingresar a la aplicación web</i>	31
<i>Tabla 19: Caso de uso seleccionar paradero</i>	32
<i>Tabla 20: Caso de uso como llegar al paradero</i>	33
<i>Tabla 21: Caso de uso cambiar paradero</i>	34
<i>Tabla 22: Caso de uso monitoreo de bus</i>	35
<i>Tabla 23: Caso de uso ver ruta del bus</i>	36
<i>Tabla 24: Caso de uso seleccionar paradero</i>	37
<i>Tabla 25: Caso de uso envío de coordenadas al servidor</i>	38
<i>Tabla 26: Caso de uso recibir coordenadas</i>	40
<i>Tabla 27: Caso de uso cálculo de datos de monitoreo</i>	41
<i>Tabla 28: Caso de uso nuevo paradero</i>	42
<i>Tabla 29: Caso de uso eliminar paradero</i>	43
<i>Tabla 30: Caso de uso modificar paradero</i>	44
<i>Tabla 31: Caso de uso buscar paradero</i>	45
<i>Tabla 32: Matriz de trazabilidad casos de uso - requisitos funcionales</i>	53
<i>Tabla 33: Estereotipos de clases utilizados en los diagramas de secuencia</i>	56
<i>Tabla 34: Resumen de capacidades del sistema</i>	61
<i>Tabla 35: Costos de implementación</i>	63
<i>Tabla 36: Descripción de elementos pantalla de inicio vista móvil</i>	70
<i>Tabla 37: Descripción de elementos pantalla seleccionar paradero vista móvil</i>	72

<i>Tabla 38: Descripción de elementos pantalla seleccionar paradero vista web</i>	72
<i>Tabla 39: Descripción de elementos pantalla mapa vista web y móvil</i>	74
<i>Tabla 40: Descripción de elementos pantalla ventana emergente opciones de paradero</i>	75
<i>Tabla 41: Descripción de elementos pantalla ver ruta del bus</i>	78
<i>Tabla 42: Tabla resumen prueba de carga</i>	127
<i>Tabla 43: Tabla resumen prueba de rendimiento 100 usuarios</i>	128
<i>Tabla 44: Tabla resumen prueba de rendimiento 500 usuarios</i>	129
<i>Tabla 45: Tabla resumen prueba de rendimiento 1000 usuarios</i>	130
<i>Tabla 46: Tabla resumen prueba de stress</i>	131
<i>Tabla 47: Resultados de la encuesta indicador tiempo pre - test y post - test</i>	132
<i>Tabla 48: Resultados de la encuesta indicador costos pre - test y post - test</i>	133
<i>Tabla 49: Resumen costos pre test y post test</i>	136
<i>Tabla 50: Resumen tiempos pre test y post test</i>	143

ÍNDICE DE FIGURAS

<i>Fig. 1: Arquitectura de un sistema de monitoreo vehicular</i>	9
<i>Fig. 2: Arquitectura entre un servidor de navegación y el sistema de monitoreo vehicular</i>	10
<i>Fig. 3: La arquitectura de una aplicación web</i>	12
<i>Fig. 4: Disciplinas, fases, iteraciones del RUP [19]</i>	13
<i>Fig. 5: Ciclo de vida del RUP [20]</i>	14
<i>Fig. 6: Actores de la aplicación web</i>	28
<i>Fig. 7: Diagrama paquetes de caso de uso</i>	29
<i>Fig. 8: Diagrama casos de uso front – end</i>	30
<i>Fig. 9: Diagrama casos de uso back – end</i>	30
<i>Fig. 10: Diagrama de actividad ingresar a la aplicación</i>	47
<i>Fig. 11: Diagrama de actividad seleccionar paradero</i>	47
<i>Fig. 12: Diagrama de actividad cómo llegar al paradero</i>	48
<i>Fig. 13: Diagrama de actividad cambiar de paradero</i>	48
<i>Fig. 14: Diagrama de actividad monitoreo del bus</i>	49
<i>Fig. 15: Diagrama de actividad ver ruta del bus</i>	49
<i>Fig. 16: Diagrama de actividad seleccionar destino</i>	50
<i>Fig. 17: Diagrama de actividad envío de coordenadas al servidor</i>	50
<i>Fig. 18: Diagrama de actividad envío de coordenadas al usuario</i>	51
<i>Fig. 19: Diagrama de actividad recibir coordenadas</i>	51
<i>Fig. 20: Diagrama de actividad cálculo de datos de monitoreo</i>	52
<i>Fig. 21: Diagrama de Análisis de clases – Interfaces</i>	54
<i>Fig. 22: Diagrama de Análisis de clases – Controles</i>	54
<i>Fig. 23: Diagrama de clase guardar usuario</i>	55
<i>Fig. 24: Diagrama de clase monitoreo</i>	55
<i>Fig. 25: Diagrama de secuencia ingresar a la aplicación</i>	56
<i>Fig. 26: Diagrama de secuencia seleccionar paradero</i>	57
<i>Fig. 27: Diagrama de secuencia cómo llegar al paradero</i>	57
<i>Fig. 28: Diagrama de secuencia cambiar de paradero</i>	58
<i>Fig. 29: Diagrama de secuencia monitoreo de bus</i>	58
<i>Fig. 30: Diagrama de secuencia ver ruta del bus</i>	58
<i>Fig. 31: Diagrama de secuencia seleccionar destino</i>	59
<i>Fig. 32: Diagrama de secuencia envío de coordenadas al servidor</i>	59
<i>Fig. 33: Diagrama de secuencia recibir coordenadas</i>	60
<i>Fig. 34: Diagrama de secuencias cálculo de datos de monitoreo</i>	60
<i>Fig. 35: Diagrama de arquitectura</i>	65
<i>Fig. 36: Arquitectura de capas o niveles</i>	67
<i>Fig. 37: Diagrama de componentes</i>	67

Fig. 38: Diagrama de base de datos entidad relación	68
Fig. 39: Logo de la aplicación web	69
Fig. 40: Pantalla bienvenida vista móvil.....	69
Fig. 41: Pantalla bienvenida vista web	70
Fig. 42: Pantalla inicio vista móvil	70
Fig. 43: Pantalla seleccionar paradero vista móvil.....	71
Fig. 44: Pantalla seleccionar paradero vista web	72
Fig. 45: Pantalla mapa sin logueo de facebook vista móvil.....	73
Fig. 46: Pantalla mapa con logueo de facebook vista móvil.....	73
Fig. 47: Pantalla mapa sin logueo de facebook vista web	73
Fig. 48: Pantalla mapa con logueo de facebook vista web	74
Fig. 49: Figura pantalla ventana emergente opciones de paradero	75
Fig. 50: Pantalla cambiar de paradero vista móvil	76
Fig. 51: Pantalla cambiar de paradero vista web.....	76
Fig. 52: Pantalla cómo llegar al paradero vista móvil	77
Fig. 53: Pantalla como llegar al paradero vista web.....	77
Fig. 54: Pantalla ventana emergente monitoreo del bus.....	78
Fig. 55: Pantalla ver ruta del bus vista móvil	79
Fig. 56: Pantalla ver ruta del bus vista web	79
Fig. 57: Pantalla seleccionar destino.....	80
Fig. 58: Instalación paso 1.....	119
Fig. 59: Instalación paso 2.....	119
Fig. 60: Instalación paso 3.....	119
Fig. 61: Instalación paso 4.....	119
Fig. 62: Instalación paso 5.....	119
Fig. 63: Instalación paso 6.....	119
Fig. 64: Instalación paso 7.....	120
Fig. 65: Instalación paso 8.....	120
Fig. 66: Instalación paso 9.....	120
Fig. 67: Instalación paso 10.....	120
Fig. 68: Instalación paso 11.....	120
Fig. 69: Instalación paso 12.....	120
Fig. 70: Instalación paso 13.....	121
Fig. 71: Instalación paso 14.....	121
Fig. 72: Instalación paso 15.....	121
Fig. 73: Instalación paso 16.....	121
Fig. 74: Instalación paso 17.....	122
Fig. 75: Instalación paso 18.....	122
Fig. 76: Instalación paso 19.....	122
Fig. 77: Instalación paso 20.....	122

<i>Fig. 78: Instalación paso 21</i>	122
<i>Fig. 79: Instalación paso 22</i>	123
<i>Fig. 80: Instalación paso 23</i>	123
<i>Fig. 81: Instalación paso 24</i>	123
<i>Fig. 82: Instalación paso 25</i>	123
<i>Fig. 83: Instalación paso 26</i>	123
<i>Fig. 84: Instalación paso 27</i>	123
<i>Fig. 85: Instalación paso 28</i>	124
<i>Fig. 86: Instalación paso 29</i>	124
<i>Fig. 87: Instalación paso 30</i>	124
<i>Fig. 88: Instalación paso 31</i>	124
<i>Fig. 89: Instalación paso 32</i>	124
<i>Fig. 90: Instalación paso 33</i>	124
<i>Fig. 91: Instalación paso 34</i>	125
<i>Fig. 92: Instalación paso 35</i>	125
<i>Fig. 93: Instalación paso 36</i>	125
<i>Fig. 94: Instalación paso 37</i>	125
<i>Fig. 95: Instalación paso 38</i>	126
<i>Fig. 96: Instalación paso 39</i>	126
<i>Fig. 97: Grafico resumen prueba de carga</i>	128
<i>Fig. 98: Grafico resumen prueba de rendimiento 100 usuarios</i>	129
<i>Fig. 99: Grafico resumen prueba de rendimiento 500 usuarios</i>	129
<i>Fig. 100: Grafico resumen prueba de rendimiento 1000 usuarios</i>	130
<i>Fig. 101: Grafico resumen prueba de stress</i>	131
<i>Fig. 102: Tiempos de espera de los usuarios por los buses universitarios</i>	133
<i>Fig. 103 Región de Aceptación y Rechazo – Indicador Tiempo</i>	141
<i>Fig. 104 Región de Aceptación y Rechazo – Indicador Costos</i>	147

RESUMEN

La presente tesis tuvo como objetivo desarrollar una aplicación web para el control y monitoreo en tiempo real que permita mejorar el servicio del transporte interno de la Universidad Nacional de Cajamarca.

Esta investigación está basada en la necesidad de los usuarios del transporte público de la Universidad Nacional de Cajamarca, que, al no contar con una plataforma que les informe en tiempo real la ubicación y el tiempo promedio de llegada de los buses de transporte público de la Universidad Nacional de Cajamarca, optan por utilizar otro medio de transporte público (pagado), en lugar de esperar el bus de la Universidad Nacional de Cajamarca (gratuito).

Como solución se planteó el desarrollar una aplicación web para el control y monitoreo en tiempo real de los buses de la Universidad Nacional de Cajamarca, el cual muestra en una interfaz sencilla e intuitiva la ubicación del bus en tiempo real, el tiempo promedio de llegada al paradero seleccionado y la ruta que está siguiendo, para el desarrollo del aplicativo se usó tecnologías orientadas a Web (Node.js, socket.IO, Postgresql), usando la arquitectura Modelo Vista Controlador, para el uso de mapas, el trazo de rutas y mostrar el monitoreo se usó Google Maps y las apis que este provee.

Como resultado se demostró que la aplicación web para el control y monitoreo en tiempo real, mejoró la calidad del servicio del transporte interno de la Universidad Nacional de Cajamarca.

Palabras clave: Transporte público, Aplicación web, Web Socket, Monitoreo.

ABSTRACT

The objective of this thesis was to develop a web application for control and monitoring in real time to improve the internal transport service of the National University of Cajamarca.

This research is based on the need of public transport users of the National University of Cajamarca; that, not having a platform that informs them in real time about the location and average arrival time of the public transport buses of the National University of Cajamarca, they choose to use another means of public transport (paid), instead of wait for the bus of the National University of Cajamarca (free).

As a solution, it was proposed to develop a web application for the control and real-time monitoring of the buses of the National University of Cajamarca, which shows in a simple and intuitive interface the location of the bus in real time, the average time of arrival at the bus station. selected whereabouts, the route that you are following, for the development of the application we used Web-oriented technologies (Node.js, socket.IO, Postgresql), using the Model View Controller architecture, for the use of maps the route of routes and show the monitoring was used Google Maps and the apis it provides.

As a result, it was demonstrated that the web application for control and monitoring in real time, improved the quality of the internal transposer service of the National University of Cajamarca.

KEYWORDS: Public transport, Web application, Web Socket, Monitoring.

CAPÍTULO I. INTRODUCCIÓN

La presente investigación refiere al tema del servicio de transporte público, que se define como los medios de transporte en que los pasajeros no son los propietarios de los mismos, siendo servidos por terceros, que pueden ser suministrados tanto por empresas públicas como privadas, y la posible mejora haciendo uso de la tecnología, es decir con la implementación de una aplicación web de control y monitoreo en tiempo real.

Este estudio aplicativo se realiza en la Universidad Nacional de Cajamarca, dedicada a la formación profesional y a la investigación en el departamento de Cajamarca, que cuenta con sub sedes ubicadas en Jaén, Chota, Bambamarca, Celendín y Cajabamba. Cuenta en la actualidad con 10 facultades que abarcan un total de 24 escuelas profesionales.

Los problemas que se abordan en la Universidad Nacional de Cajamarca respecto al servicio de transporte público interno son: los estudiantes no saben exactamente a qué hora llegarán los buses debido a que no existe comunicación a tiempo real entre los usuarios y los buses, lo que genera una desorganización entre la oferta y la demanda del servicio, no contar con un medio tecnológico que realice un control y monitoreo en tiempo real de los buses de la Universidad Nacional de Cajamarca, generando inconvenientes como: que un estudiante opte por elegir utilizar otro medio de transporte (pagado), en vez de esperar uno de los buses de la Universidad Nacional de Cajamarca (gratuito) que viene con un poco de retraso, que éstos puedan sentirse impacientes, especialmente cuando tienen prisa por algún motivo. Asimismo, por la pérdida de tiempo y dinero que implica esta situación.

Con la implementación de una aplicación web para el control y monitoreo en tiempo real se responderá a la siguiente pregunta que hace referencia a los problemas en el servicio de transporte interno de la Universidad Nacional de Cajamarca: ¿en qué medida el desarrollo de una aplicación web para el control y monitoreo en tiempo real mejora el servicio del transporte interno de la Universidad Nacional de Cajamarca? Para dar respuesta a esta pregunta se demuestra la veracidad de la hipótesis general: el desarrollo de una aplicación web para el control y monitoreo en tiempo real permitirá mejorar

significativamente el servicio del transporte interno de la Universidad Nacional de Cajamarca.

La justificación teórica de esta investigación indica que la tecnología GPS o global positioning system, es un sistema global de navegación satelital que ha facilitado la obtención de datos sobre geolocalización a tiempo real, lo cual ha constituido una herramienta valiosa desde lo personal hasta el terreno empresarial, mostrando particular eficiencia en su aplicación para sistemas de control y monitoreo en servicios de transportes [1]. Esta tecnología les permite conocer y controlar en tiempo real las diferentes variables como posición (latitud, longitud), fecha, hora y velocidad de desplazamiento, que están presentes en un trayecto, las que sin duda se verán reflejadas en el bienestar y la eficiencia en el servicio de transporte. La utilidad metodológica, radica en que puede ser replicado en diferentes campos donde sea necesario la comunicación bidireccional a tiempo real. Las implicaciones prácticas radican en que, con la implementación de un sistema de comunicación bidireccional a tiempo real, se pretende ayudar a la eficiencia del servicio de transporte interno de la Universidad Nacional de Cajamarca. Es por ello que este estudio pretende dar respuesta a los inconvenientes en el transporte publica interno de la Universidad Nacional de Cajamarca, a través de una aplicación web de control y monitoreo a tiempo real.

El alcance de este proyecto de investigación es la implementación y el despliegue de una aplicación web de control y monitoreo en tiempo real, a través de la comunicación bidireccional, con tecnología websocket.

El objetivo general de este estudio es: desarrollar una aplicación web para el control y monitoreo en tiempo real que permita mejorar el servicio del transporte interno de la Universidad Nacional de Cajamarca, para lo cual se tiene que lograr los siguientes objetivos específicos: Analizar los requisitos para el desarrollar de la aplicación web, diseñar la arquitectura de la aplicación web, implementar la aplicación web, realizar pruebas a la aplicación web.

Para la organización de este informe, se ha dividido en cinco capítulos. El primero corresponde a la Introducción, donde se describe el contexto, el problema, la justificación o importancia de la investigación, los alcances de la investigación y los objetivos. En el

segundo capítulo se hace el marco teórico, que hace una revisión de la literatura sobre temas relacionados con estudio de investigación, como conceptos de sistemas de transporte público, sistemas de posicionamiento global, receptor GPS, sistemas de monitoreo vehicular, comunicación bidireccional a tiempo real y aplicación web. El tercer capítulo trata de los materiales y métodos, donde se describe la ubicación geográfica donde se realizó el estudio, el procedimiento que describe paso a paso y con detalle los procesos realizados, también se especifica el tratamiento y análisis de los datos y presentación de resultados. En el cuarto capítulo, análisis y discusión de resultados, se explica y discute los resultados siguiendo la secuencia de los objetivos. Y el último capítulo, conclusiones y recomendaciones, se dan a conocer los resultados finales frente a cada objetivo planteado, y se indica algunas recomendaciones lo cual constituye un aporte de este trabajo de investigación.

CAPÍTULO II. MARCO TEÓRICO

2.1 ANTECEDENTES TEÓRICOS DE LA INVESTIGACIÓN

2.1.1 INTERNACIONALES

El servicio de transporte en autobús es, en todos los sentidos, muy útil e importante, pero sin un buen sistema de administración de autobuses, es posible que el servicio no sea utilizado eficientemente, generando malestares en los usuarios.

Yago [2], en su tesis: “Aplicación web y móvil para el seguimiento de autobuses escolares”. En el cual desarrolló y desplegó una plataforma web, mediante el uso el framework CakePHP¹, el cual le permite gestionar el servicio de geolocalización de flotas por GPS, mediante push notifications o emails, el servicio de alertas sobre la proximidad de buses escolares a una zona definida, con la finalidad de informar a los padres de familia de la llegada de sus hijos a su parada habitual antes de que ésta tenga lugar, en esta tesis el autor llega a la conclusión que el uso del framework CakePHP permite gestionar el servicio de geolocalización por GPS y el framework Phonegap permite compilar la aplicación web como aplicaciones móviles y poder hacer el envío de notificaciones.

Ching [3], en su tesis: “Bus tracking system”, En el cual desarrolla un sistema de seguimiento de autobuses que proporciona a los usuarios de los autobuses una plataforma en tiempo real para verificar la información actualizada del tráfico del autobús usando lenguajes de programación como PHP, JavaScript, AJAX, JavaServlet y Java Server Pages (JSP) para desarrollar el sistema, en esta tesis el autor llega a la conclusión que instalando los receptores GPS en los buses y usando la tecnología GPS de rastreo se consiguió enviar los datos de posición del autobús hacia el servidor central, información de seguimiento que se proporciona a los usuarios en tiempo real.

¹ CakePHP es un framework para el desarrollo aplicaciones web escrito en PHP, creado sobre los conceptos de Ruby on Rails.

2.1.2 NACIONALES

A nivel nacional, se ha encontrado algunos trabajos de investigación similares, como: Conza [4], en su tesis: “Desarrollo de una aplicación web, orientada a servicios para el monitoreo de una flota de vehículos haciendo uso de la tecnología GPS”. En el cual desarrolla un sistema de rastreo geo referencial para el rastreo y monitoreo de una flota vehicular en la ciudad del cusco, mediante el uso de los GPS equipados en los vehículos el cual proporciona la posición, hora y velocidad, estos datos son enviados a una estación central mediante servicios web REST. En la estación central procesan estos datos lo almacenan en una base de datos MySQL; al mismo tiempo envían estos datos a los clientes a través de websockets que en ese preciso momento están conectados rastreando los vehículos en tiempo real, dichos clientes visualizan la ubicación de los vehículos en tiempo real recibida desde la estación central a través de un mapa que provee google maps, en esta tesis el autor llega a la conclusión que se logró desarrollar una aplicación web para la gestión y monitoreo de una flota de vehículos, que hace uso de la tecnología GPS implementando la transmisión de datos bidireccional cliente – servidor, para la comunicación en tiempo real, con la tecnología web socket.

Meza, Leño [5], en su tesis: “Sistema de monitoreo de una red de buses de transporte público e información para usuarios empleando transceptores GPS/GSM”. El cual desarrolló un sistema de monitoreo de una red de buses de transporte público, mediante el empleo de tecnologías de comunicación y posicionamiento. El empleo de transceptores GPS/GSM permitirá obtener datos de posición (latitud, longitud), fecha, hora y velocidad de desplazamiento del móvil monitoreado. Esta información será enviada a una central para poder procesar esta información. Una vez procesada esta información el administrador tomará acciones correctivas como controlar el exceso de velocidad, informar a los choferes de rutas alternativas entre otra que el administrador considere necesarias, proveer la información de tiempo y distancia de llegada a los usuarios a través de mensajes de texto o aplicaciones para smartphones. Este sistema se pretende ser integrado a los sistemas del metropolitano y el metro de Lima y así poder lograr un sistema integrado de transporte, en esta tesis el autor llega a la conclusión que es posible la realización de un sistema de monitoreo, empleando comunicación GSM, posicionamiento satelital (GPS), procesamiento por software, gestión de base de datos y envío de información al usuario a través del paradero.

Ureta [6], en su tesis; “Desarrollo de una aplicación web para el monitoreo de vehículos con dispositivos GPS que comercializa una empresa de telecomunicaciones”. En el cual presenta la metodología de desarrollo comprendida en la realización de una solución de monitoreo para vehículos con dispositivos GPS, el software propuesto forma parte del servicio de monitoreo vehicular que ofrece una empresa de telecomunicaciones, se logró implementar una solución web de monitoreo vehicular a múltiples clientes sin necesidad de instalaciones personales, solo con el uso de un navegador web. Actualmente el sistema cuenta con 95 clientes y 200 unidades transmitiendo todos los días, siendo la principal herramienta de monitoreo de muchas de ellas, en esta tesis el autor llega a la conclusión que se pudo implementar una solución web de monitoreo vehicular que da servicio a múltiples clientes sin necesidad de instalaciones personales, solo con el uso de un navegador web, los usuarios pueden gozar de las más representativas funcionalidades que una solución de monitoreo puede brindar de manera inmediata.

2.2 BASES TEÓRICAS

El monitoreo de vehículos con dispositivos o receptores GPS, abarca en la actualidad un abanico extenso de tecnologías, desde el concepto de sistemas de posicionamiento, sistemas de información geográfica y aplicaciones software. Como es de suponer todas estas tecnologías tienen una profunda base de conocimientos y distintas aplicaciones, debido a esto, nos enfocaremos en explicar de manera objetiva estas y su influencia en el servicio de monitoreo de vehículos.

2.2.1 SERVICIO DE TRANSPORTE PÚBLICO

Escalona [7], el servicio de transporte público comprende los medios de transporte en que los pasajeros no son los propietarios de los mismos, siendo servidos por terceros. Los servicios de transporte público pueden ser suministrados tanto por empresas públicas como privadas, también nos indica que es el medio de traslado de personas o bienes desde un lugar hasta otro. El transporte comercial moderno está al servicio del interés público e incluye todos los medios e infraestructuras implicadas en el movimiento de las personas o bienes, así como los servicios de recepción, entrega y manipulación de tales bienes. El transporte comercial de personas se clasifica como servicio de pasajeros y el de bienes como servicio de mercancías. Como en todo el mundo, el transporte es y ha sido en Latinoamérica un elemento central para el progreso o el atraso de las distintas civilizaciones y culturas.

2.2.2 SISTEMA DE POSICIONAMIENTO GLOBAL

Ramos [7], el Sistema de Posicionamiento Global (GPS) es el más conocido y usado sistema de navegación por satélite. Que consiste en por lo menos una red de 24 satélites orbitando a 20200 km alrededor de la superficie terrestre y que continuamente transmiten señales de frecuencias duales que son procesadas por los receptores GPS. En un principio el GPS fue diseñado para uso militar ya que el 22 de febrero de 1978 fue lanzado el primer satélite y fueron 2 eventos clave que hicieron que el GPS fuera de uso completamente civil. El primero en 1983 cuando fue abierto al público y el 2000 cuando se desactivó la accesibilidad selectiva, que consistía en una degradación intencionada de la transmisión.

Yen [1], menciona que la navegación por satélite comenzó a principios de 1970, ya que tres sistemas de satélites fueron explorados antes que se diera inicio al programa GPS: El sistema de navegación por satélite de la marina de Estados Unidos, conocido como TRANSIT, el sistema TIMATION de la marina de Estados Unidos, su nombre viene del agrupamiento de las palabras (Time Navigation), y el proyecto de la fuerza de Estados Unidos 621B. Estos 3 proyectos proporcionaron diversos alcances técnicos que enriquecieron los inicios del programa GPS. El programa GPS fue aprobado en diciembre de 1973. El primer satélite del programa fue lanzado en 1978. En agosto de 1993, el sistema GPS ya tenía alrededor de 24 satélites en órbita y en diciembre del mismo año la capacidad operativa inicial fue establecida. En febrero de 1994, la agencia de aviación federal (FAA) declaró al GPS listo para el uso de la aviación.

Pozo [8], indica que el principio de un sistema GPS es calcular la posición de un punto en el espacio de coordenadas (x, y, z), basándose en las distancias del punto a un mínimo de tres satélites cuya ubicación es conocida. La distancia del equipo GPS y el satélite se mide calculando el tiempo que demora llegar la señal al punto de referencia. Para poder calcular este tiempo, los relojes de los satélites deberán estar sincronizados. Si bien los relojes de los satélites son muy precisos por lo contrario los relojes de los equipos GPS o receptores no lo son tanto debido a que son fabricados a bajo coste, a fin de corregir cualquier inconveniente, es usado un cuarto satélite para la corrección de los cálculos y así encontrar la posición exacta del receptor o GPS.

2.2.3 RECEPTOR GPS

Bao [1], describe a un receptor GPS como un dispositivo electrónico compuesto por una antena, una cadena de radio frecuencia y convertidor análogo digital, que se comportan de la siguiente manera: La señal transmitida del satélite GPS son recibidas por la antena. A través de la cadena de radio frecuencia (RF) la señal de entrada es amplificada a una adecuada amplitud y la frecuencia es convertida a la frecuencia de salida deseada. Luego la señal de salida es digitalizada por un convertidor análogo digital (ADC). La antena, la cadena RF y el ADC son dispositivos usados en el receptor GPS. Luego que la señal es digitalizada, se puede utilizar un software para localizar las coordenadas del receptor GPS, a la cual se llama fase de transición de la información de navegación. En un receptor convencional esta fase es desempeñada por el hardware. Los pseudo - rangos y la información de efemérides pueden ser obtenidas de la información de la navegación. Con la información de efemérides se puede obtener la posición de los satélites. Por último, la posición del receptor puede ser calculado por la posición de los satélites y el pseudo - rango.

Rodríguez [9], hace una descripción muy acertada de las ventajas que ofrecen los receptores GPS y sus diversos usos, debido a su tamaño reducido es fácil de transportar en diversos tipos de vehículos, como bicicletas, autos, helicópteros, y así cartografiar pistas y carreteras, describiendo a un receptor GPS como una gran herramienta para un sistema de información geográfica.

2.2.4 SISTEMA DE MONITOREO VEHICULAR

Jen [10], define que un sistema de monitoreo vehicular como aquel sistema que se encarga de administrar y analizar la información de la posición de un vehículo proveniente de un dispositivo inalámbrico en tiempo real, así mismo, un sistema de monitoreo vehicular es el centro de cualquier sistema de información de transporte. Este dispositivo inalámbrico puede ser un dispositivo GPS o un transceptor de radio frecuencia o RF instalado en el vehículo, en caso de que el vehículo transite en la cobertura de una red inalámbrica puede usar un transceptor RF a fin de ahorrar costos, debido a que la conexión GPS suele ser más costosa.

Así mismo el autor, detalla la estructura básica de una tabla que maneja información de un sistema de monitoreo vehicular, esta tabla por lo menos deberá poseer 3 campos, los cuales son:

- Grupo: en caso de que la red de sensores o transmisores se dividan en grupos, se puede prevenir transmisión innecesaria.
- Fuente: el número del nodo que directamente transmite el mensaje.
- Tiempo: fecha de la recepción del mensaje.

Kaplan [11], afirma que la función de una aplicación de monitoreo vehicular, es aquel sistema donde la posición del vehículo es determinada y luego enviada de manera inalámbrica a una instalación de control centralizado. Así mismo describe la arquitectura de un sistema de monitoreo vehicular, como la composición de un receptor GPS, sensores auxiliares conectados al receptor GPS y un procesador de computadora para controlar los componentes y calcular la solución de la posición optimizada ver fig.1.

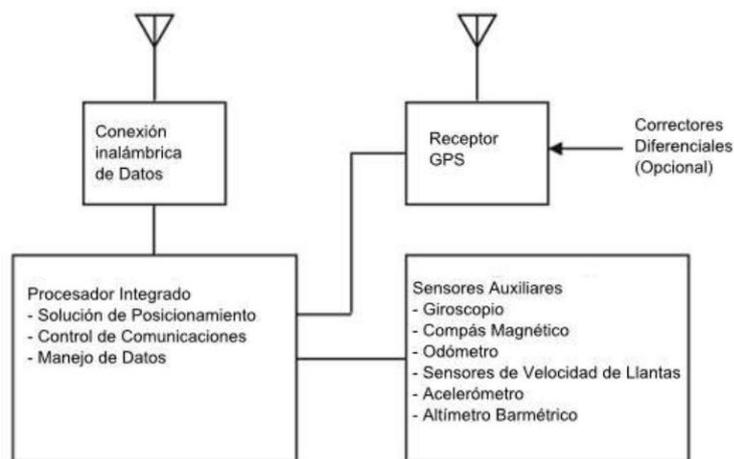


Fig. 1: Arquitectura de un sistema de monitoreo vehicular

Thangaelu [12], afirma que los sistemas de monitoreo vehicular son dispositivos electrónicos en vehículos para permitir a los dueños de los vehículos o a terceros, rastrear la ubicación del vehículo. Los más modernos sistemas de monitoreo vehicular usan equipos GPS para asegurarse la ubicación exacta del vehículo. Así mismo muchos sistemas de monitoreo vehicular combinan características celulares y satelitales para transmitir la posición del vehículo al usuario ver fig. 2.

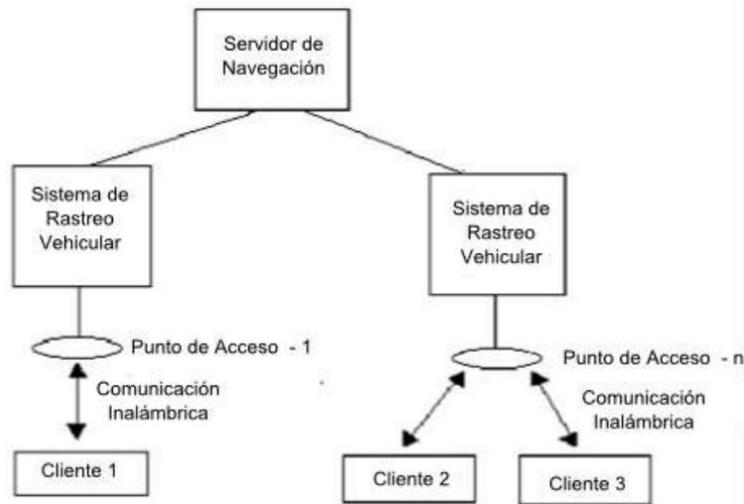


Fig. 2: Arquitectura entre un servidor de navegación y el sistema de monitoreo vehicular

2.2.5 COMUNICACIÓN BIDIRECCIONAL EN TIEMPO REAL

Schramm [13], afirma que el proceso de comunicación tiene dos características fundamentales, no existe un sólo mensaje emitido sino grupos de mensajes y grupos de mensajes emitidos por códigos verbales y no verbales, y la comunicación es bidireccional, en el que emisor y receptor va intercambiando mensajes precisando la información recibida, solicitando aclaraciones y comprobando lo que el oyente ha entendido sus posiciones en el proceso en donde el emisor y el receptor intercambian alternativamente sus papeles es decir el emisor pasa a ser receptor y el receptor pasa a ser emisor. Se caracteriza a grandes rasgos por el hecho de que la respuesta tiene tanta importancia cuantitativa como la emisión y, por consiguiente, emisión y recepción precisan una participación equivalente.

2.2.6 NODE.JS

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript², asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como, por ejemplo, servidores web.

² ECMAScript es una especificación de lenguaje de programación publicada por ECMA International, basado en el popular lenguaje JavaScript

Fue creado por Ryan Dahl en 2009 y su evolución está apadrinada por la empresa Joyent, que además tiene contratado a Dahl en plantilla.

Node.js es similar en su propósito a Twisted o Tornado de Python, Perl Object Environment de Perl, libevent o libev de C, EventMachine de Ruby, vibe.d de D y Java EE de Java existe Apache MINA, Netty, Akka, Vert.x, Grizzly o Xsocket. Al contrario que la mayoría del código JavaScript, no se ejecuta en un navegador, sino en el servidor. Node.js implementa algunas especificaciones de CommonJS.5 Node.js incluye un entorno REPL para depuración interactiva [15].

2.2.7 SOCKET.IO

Socket.io es una librería en JavaScript para Node.js que permite una comunicación bidireccional en tiempo real entre cliente y servidor. Para ello se basa principalmente en websocket, pero también puede usar otras alternativas como sockets de adobe flash, JSONP polling o long polling en AJAX, seleccionando la mejor alternativa para el cliente justo en tiempo de ejecución.

Websockets es una tecnología basada en el protocolo ws, este hace posible establecer una conexión continua full – duplex, entre un cliente y servidor. Un cliente websocket podría ser el navegador del usuario, pero el protocolo es una plataforma independiente [16].

2.2.8 APLICACIÓN WEB

Jablonski [14], afirma que una aplicación web no tiene nada de extraordinario comparado con una aplicación de computadora, ya que no deja de tomar en cuenta las reglas tradicionales del diseño de una aplicación. Una aplicación web es una aplicación que muestra páginas HTML generadas dinámicamente y sus componentes son similares a una aplicación computacional.

Si se analiza la arquitectura conceptual, en una aplicación web se puede reconocer ciertos componentes como el paquete de presentación, lógica del negocio y manejo de datos, en otras palabras, la interfaz de usuario, la funcionalidad de la aplicación y la manipulación de la información.

Si bien una aplicación web es una aplicación computacional estándar, tiene características específicas, por ejemplo, una aplicación web no necesita ser ejecutable, se basa principalmente en el contenido, la interfaz de usuario está basada en páginas HTML generadas dinámicamente y la navegación está basada en hipervínculos ver fig. 3.

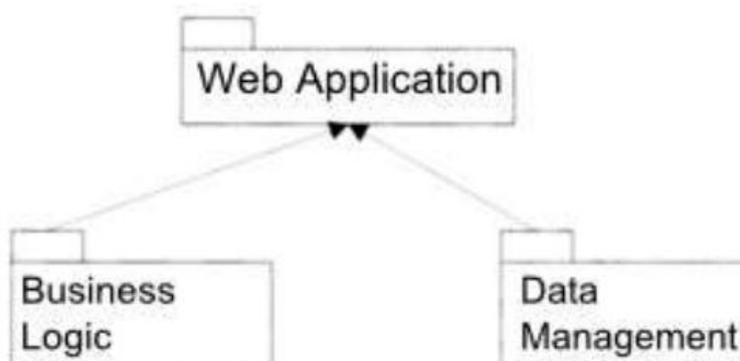


Fig. 3: La arquitectura de una aplicación web

Shklar [15], afirma que una aplicación web es una agrupación de protocolos y lenguajes asociados, que incluye al protocolo HTTP (HyperText Transfer Protocol) y al lenguaje HTML (HyperText Markup Language) que son fundamentales al momento de la creación y de la transmisión de una página web. Así mismo hace mención de otros protocolos de internet que intervienen en una aplicación web, por ejemplo: TELNET, FTP, SMTP e IMAP, así mismo lenguajes avanzados como XML. A esto le añade el conocimiento de bases de datos y presentación multimedia, el autor afirma que la idea de una aplicación web es casi la suma de todos los conceptos, ya que las personas que diseñan aplicaciones web, no sólo deberían conocer de protocolos como HTTP y HTML, ya deberán conocer Javascript, XML, bases de datos relacionales, diseño gráfico y multimedia entre otros conceptos.

2.2.9 PROCESO DE DESARROLLO DE SOFTWARE

Para el proceso de desarrollo de software usaremos RUP (Proceso Unificado de Rational), la cual proporciona disciplinas en las cuales se encuentran artefactos que proveen herramientas para documentar e implementar de una manera fácil y eficiente, todas las guías para un buen desarrollo, dentro de las respectivas fases con las cuales cuenta. Además, se utilizará el Lenguaje Unificado de Modelado (UML) como herramienta para desarrollar los planos del software [19].

➤ Dimensiones de RUP

RUP nos permite manejar dos dimensiones:

- **Horizontal:** La cual representa el tiempo y demuestra los aspectos del ciclo de vida del proceso.
- **Vertical:** Representa las disciplinas, que agrupan actividades definidas.

En la fig. 4 se puede observar como varía el énfasis de cada disciplina en un cierto plazo en el tiempo, y durante cada una de las fases. Por ejemplo, en iteraciones tempranas, pasamos más tiempo en requerimientos, y en las ultimas iteraciones pasamos más tiempo en poner en práctica la realización del proyecto en sí.

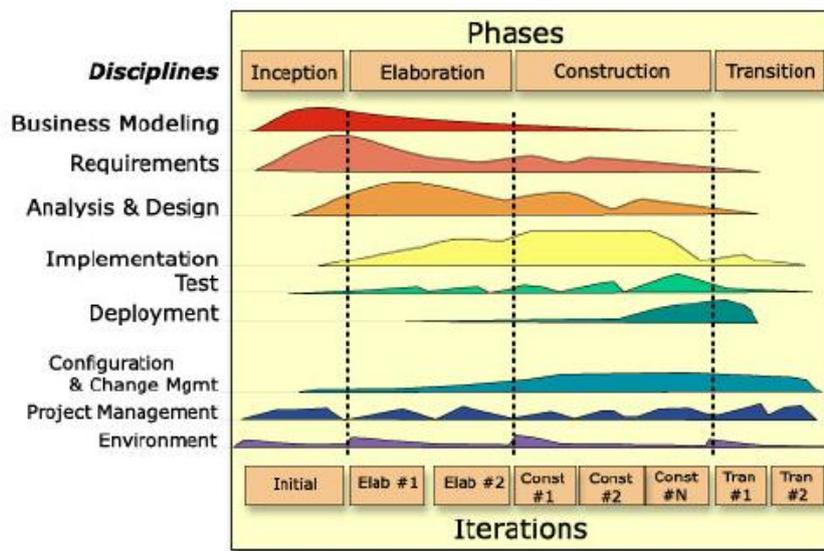


Fig. 4: Disciplinas, fases, iteraciones del RUP [19]

➤ Características esenciales de RUP

- **Proceso Dirigido por los Casos de Uso:** Con esto se refiere a la utilización de los Casos de Uso para el desenvolvimiento y desarrollo de las disciplinas con los artefactos, roles y actividades necesarias. Los casos de uso son la base para la implementación de las fases y disciplinas del RUP. Un caso de uso es una secuencia de pasos a seguir para la realización de un fin o propósito, y se relaciona directamente con los requerimientos, ya que un caso de uso es la secuencia de pasos que conlleva la realización e implementación de un requerimiento planteado por el cliente. Como lenguaje para describir los diagramas que conlleva este proceso se puede

utilizar el mismo que provee RUP o como en nuestro caso ya mencionamos, que utilizaremos el lenguaje de modelado UML, el cual solo difiere de RUP en notación mas no en significado.

- **Proceso Iterativo e Incremental:** Es el modelo utilizado por RUP para el desarrollo de un proyecto de software. Este modelo plantea la implementación del proyecto a realizar en Iteraciones, con lo cual se pueden definir objetivos por cumplir en cada iteración y así poder ir completando todo el proyecto iteración por iteración, con lo cual se tienen varias ventajas, entre ellas se puede mencionar la de tener pequeños avances del proyectos que son entregables al cliente el cual puede probar mientras se está desarrollando otra iteración del proyecto, con lo cual el proyecto va creciendo hasta completarlo en su totalidad.
- **Proceso Centrado en la Arquitectura:** Define la arquitectura de un sistema, y una arquitectura ejecutable construida como un prototipo evolutivo. Arquitectura de un sistema es la organización o estructura de sus partes más relevantes. Una arquitectura ejecutable es una implementación parcial del sistema, construida para demostrar algunas funciones y propiedades. RUP establece refinamientos sucesivos de una arquitectura ejecutable, construida como un prototipo evolutivo.

➤ Fases de RUP

El ciclo de vida del software del RUP se descompone en cuatro fases secuenciales como se muestra en la siguiente figura:

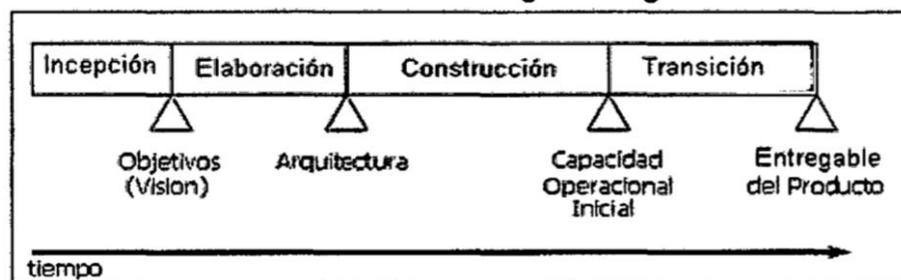


Fig. 5: Ciclo de vida del RUP [20]

En cada extremo de una fase se realiza una evaluación para determinar si los objetivos de la fase se han cumplido. Si es satisfactoria la evaluación, el proyecto se mueve a la próxima fase.

A continuación, describimos brevemente las fases de RUP

- **Fase de inepción:** También conocida por otros nombres como: Concepción, estudio de oportunidad, etc. Define el ámbito y objetivos del proyecto, así como, la funcionalidad y capacidades del producto.
- **Fase de Elaboración:** Se estudia la funcionalidad y el dominio del problema a profundidad, con lo cual se define una arquitectura básica, para luego planificar el proyecto considerando los recursos disponibles.
- **Fase de Construcción:** Consiste en desarrollar el producto a través de iteraciones, donde cada iteración involucra tareas de análisis, diseño e implementación. En esta fase gran parte del trabajo es programación y pruebas, las cuales van junto con su respectiva documentación.
- **Fase de Transición:** Se libera el producto y se entrega al usuario para uso real, se incluye tareas de instalación, capacitación, etc. Estas tareas también se realizan en iteraciones. En esta fase finalmente se completan los manuales de usuario y se refinan con la información anterior.

➤ **Iteraciones de RUP**

Este proceso se refiere a la realización de un ciclo de vida de un proyecto y se basa en la evolución de prototipos ejecutables que se muestran a los usuarios y clientes.

En este ciclo de vida iterativo a cada iteración se reproduce el ciclo de vida en cascada a menor escala, estableciendo los objetivos de una iteración en función de la evaluación de las iteraciones precedentes y las actividades se encadenan en una mini - cascada con un alcance limitado por los objetivos de la iteración.

➤ **Disciplinas de RUP**

Las disciplinas conllevan los flujos de trabajo, los cuales son una secuencia de pasos para la culminación de cada disciplina, estas disciplinas se dividen en dos grupos:

- **Las primarias:** Son las necesarias para la realización de un proyecto de software; entre estas se tienen: Modelado del Negocio, Requerimientos, Análisis y Diseño, Implementación, Pruebas, Despliegue.

- **Las de apoyo:** Sirven de soporte a las primarias y especifican otras características en la realización de un proyecto de software; entre estas se tienen: Entorno, gestión del proyecto, gestión de configuración y cambios.

➤ **Organización y elementos en RUP**

Entre los elementos de RUP tenemos: Flujos de trabajo, detalle de los flujos de trabajo, actores, actividades y artefactos. Entonces el flujo de trabajo de requerimientos conlleva varios pasos, cada uno de estos pasos tiene asociado uno o varios actores, los cuales a su vez son los encargados de la ejecución de varias actividades, las cuales a la vez están definidas en artefactos o guías para su realización. A continuación, describimos brevemente cada uno de estos elementos:

- **Actores o Roles:** Son los personajes encargados de la realización de las actividades definidas dentro de los flujos de trabajo de cada una de las disciplinas del RUP, divididos en varias categorías.
- **Artefactos:** Son las entradas y salidas de las actividades, realizadas por los actores, los cuales utilizan y van produciendo estos artefactos para tener guías. Un artefacto puede ser un documento, un modelo o un elemento de modelo.
 - **Artefactos de Modelado del negocio:** Capturan y presentan el contexto del negocio del sistema. Los artefactos del modelado del negocio sirven como entrada y como referencia para los requisitos del sistema.
 - **Artefactos Requerimientos del sistema:** Capturan y presentan la información usada para definir las capacidades requeridas del sistema.
 - **Artefactos Análisis y diseño del sistema:** Capturan y presenta la información relacionada con la solución a los problemas se presentaron en los requisitos fijados.

2.3 DEFINICIÓN DE TÉRMINOS BÁSICOS

Aplicación web: Se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador. En otras palabras, es un programa que se codifica en un lenguaje interpretable por los navegadores web en la que se confía la ejecución al navegador [16].

Transporte público: Es el término aplicado al transporte colectivo de pasajeros. A diferencia del transporte privado, los viajeros de transporte público tienen que adaptarse a los horarios y a las rutas que ofrezca el operador y dependen en mayor o menor medida de la intervención regulatoria del Gobierno [20].

Web Socket: Es una tecnología que proporciona un canal de comunicación bidireccional y full - duplex sobre un único socket TCP [21].

Monitoreo vehicular: Se aplica a los sistemas de localización remota en tiempo real, basados generalmente en el uso de un GPS, GSM, Bluetooth, WiFi y un sistema de transmisión que es frecuentemente un módem inalámbrico [18].

Comunicación bidireccional: Es en la que emisor y receptor/es intercambian mensajes precisando la información recibida, solicitando aclaraciones y comprobando lo que el oyente ha entendido, resulta más eficaz, especialmente si junto a la comunicación verbal también se utiliza a la no verbal [23].

Receptor GPS: Es un dispositivo de bolsillo que permite saber la posición geográfica longitud y latitud con una precisión de unos metros, usando la tecnología GPS [24].

Sistema de monitoreo vehicular: Es un sistema desarrollado con la finalidad de brindar servicios de control y monitoreo a distintos objetos (vehículos/personas) que tengan incorporado un GPS [19].

CAPÍTULO III. MATERIALES Y MÉTODOS

La presente investigación fue realizada en la Universidad Nacional de Cajamarca; ubicada en la Ciudad Universitaria (Av. Atahualpa No 1050).

El desarrollo de la investigación se desarrolló en un periodo de siete meses, considerados a partir del mes de julio de 2018. En este lapso se realizó la caracterización para el desarrollo del Sistema de Control y Monitoreo en Tiempo Real, a nivel de requerimientos. Así mismo se consideró el análisis, diseño e implementación de la solución.

3.1 PROCEDIMIENTO

Para el proceso de desarrollo de software usaremos RUP (Proceso Unificado de Rational), la cual proporciona disciplinas en las cuales se encuentran artefactos que proveen herramientas para documentar e implementar de una manera fácil y eficiente. Además, se utilizará el Lenguaje Unificado de Modelado (UML) como herramienta para desarrollar los planos del software [19].

3.1.1 VISIÓN DEL PROYECTO

En esta sección se reúne, analiza y define las necesidades y características claves de la aplicación web.

3.1.1.1 DESCRIPCIÓN DEL PROYECTO

3.1.1.1.1 OPORTUNIDAD DE MEJORA

La aplicación web proporcionará una herramienta en la que los usuarios de los buses de la Universidad Nacional de Cajamarca puedan realizar un monitoreo en tiempo real de estos, conociendo la ubicación y tiempo de llegada al paradero de destino.

Adicionalmente, se simplificará el tiempo de espera, los recursos utilizados en el transporte diario de los usuarios de los buses de la Universidad Nacional de Cajamarca disminuirán ya que los usuarios utilizarán los buses de la Universidad Nacional de Cajamarca que es gratuito.

La tecnología websocket, permitirá optimizar el tiempo de comunicación entre los clientes y el servidor, optimizando el envío y recepción de la información en tiempo real.

3.1.1.1.2 PROBLEMAS EN EL PROCESO

La definición del problema se esquematiza de la siguiente manera.

Tabla 1: Definición del problema

Problema general	<ul style="list-style-type: none"> ➤ No se cuenta con una aplicación web en tiempo real que proporcione a los usuarios un monitoreo de los buses de la Universidad Nacional de Cajamarca, lo que genera que los usuarios tengan prolongados tiempos de espera y adicionalmente opten por otros medios de transporte pagados, generando un gasto adicional en ellos.
Problemas específicos	<ul style="list-style-type: none"> ➤ Los usuarios desconocen la ubicación actual de los buses de la Universidad Nacional de Cajamarca.
	<ul style="list-style-type: none"> ➤ Los usuarios desconocen el tiempo faltante para que el bus de la Universidad Nacional de Cajamarca llegue a su paradero.
	<ul style="list-style-type: none"> ➤ Pérdida de tiempo y gastos en otros medios de transporte pagados por parte de los usuarios.
	<ul style="list-style-type: none"> ➤ Viaje de los buses con pocos pasajeros.
Grupo afectado	<ul style="list-style-type: none"> ➤ Todos los usuarios de los buses de la Universidad Nacional de Cajamarca.
Una solución exitosa	<ul style="list-style-type: none"> ➤ Mostrar en tiempo real la ubicación de los buses de la Universidad Nacional de Cajamarca.
	<ul style="list-style-type: none"> ➤ Mostrar en tiempo real el tiempo promedio de llegada de los buses de la Universidad Nacional de Cajamarca a su paradero de destino.
	<ul style="list-style-type: none"> ➤ Mostrar en tiempo real la ruta que viene tomando el bus de la Universidad Nacional de Cajamarca.
	<ul style="list-style-type: none"> ➤ Mostrar el tiempo promedio que demoraría el usuario en llegar desde su ubicación actual hasta su paradero de destino, ya sea a pie o en automóvil.
	<ul style="list-style-type: none"> ➤ Mostrar la ruta que puede seguir el usuario para llegar desde su ubicación actual hasta su paradero de destino, ya sea a pie o en automóvil.
	<ul style="list-style-type: none"> ➤ Ahorrar tiempo y recursos económicos a los usuarios de los buses de la Universidad Nacional de Cajamarca.
	<ul style="list-style-type: none"> ➤ Optimizar la capacidad de los buses de la Universidad Nacional de Cajamarca en cada viaje que estos realicen.

3.1.2 DESCRIPCIÓN DE USUARIOS

3.1.2.1 DEFINICIÓN DE USUARIOS

Los usuarios son los miembros de la comunidad universitaria que hacen uso de los buses de la Universidad Nacional de Cajamarca. Por otro lado, otra parte de los usuarios es el personal a cargo de los buses universitarios, entre ellos los conductores. Todos los usuarios de la organización poseen celulares smartphone o una computadora con acceso a internet. Los usuarios tienen conocimiento sobre el uso de smartphone, computadoras, acceso a internet y la instalación de aplicaciones.

Los conductores de los buses tienen acceso al aplicativo mediante un smartphone en donde podrán seleccionar el paradero de destino del bus que van conduciendo, mientras que los usuarios pasajeros del bus tienen el acceso a la aplicación desde un smartphone o una computadora donde podrán hacer el monitoreo del bus y recibir información en tiempo real, todo esto a través de internet [16].

3.1.2.2 PERFILES DE USUARIO

Los usuarios del sistema se pueden clasificar como:

Tabla 2: Perfiles de usuarios

Usuarios sistema
1. Administrador
2. Conductores
3. Comunidad universitaria

3.1.2.3 AMBIENTES DE LA APLICACIÓN WEB

Los usuarios pueden ubicarse dentro o fuera de la Universidad Nacional de Cajamarca. El único requisito es que cuenten con un smartphone o una computadora que utilicen para acceder a la aplicación web y tener conexión a internet para la obtención de la información enviada en tiempo real.

El envío de los datos en tiempo real se realizará desde los smartphones que se encuentran dentro de los buses hacia el servidor web, y este servidor web a su vez enviará los datos a los usuarios que estén realizando el monitor en tiempo real de los buses.

Para mostrar la ubicación de los usuarios, los buses y los paraderos se está haciendo uso de google maps y las apis que este nos proporciona para calcular y mostrar la información como el tiempo de llegada y distancia de los buses y los usuarios hacia los paraderos.

3.1.3 VISTA GENERAL DE LA APLICACIÓN WEB

En esta sección se describe las capacidades de la aplicación, así como su integración con el servidor web.

El sistema tendrá el nombre: "El carro de la U".

3.1.3.1 PERSPECTIVA DE LA APLICACIÓN WEB

La aplicación web de control y monitoreo tiene una arquitectura cliente - servidor. En el servidor web se deberá tener como sistema operativo base ubuntu server con acceso mediante ssh, como gestor de base de datos postgresQL 9.5 y como servidor web Node JS bajo el framework denominado SAILS JS, las interfaces graficas están bajo HTML5 y el framework materialize. El servicio encargado de la comunicación con los buses de la Universidad Nacional de Cajamarca se denomina sockets, este servicio gestiona conexiones a la base de datos y los paquetes de datos enviados por los buses de la Universidad Nacional de Cajamarca. Los hardware instalados en los buses serán celulares con sistema operativo android, y es a través de este dispositivo que se obtienen los datos generados de los buses y son enviados al servidor por el servicio en modo cliente a través de la red wifi o la red de datos del celular. Todos los usuarios pueden usar su ordenador, tablet, celular, etc. con acceso a internet y/o el aplicativo instalado, estos deberán poder tener acceso a la aplicación web ya sea por la web o por la aplicación móvil (android, iOS).

La aplicación web muestra el monitoreo en tiempo real de los buses de la Universidad Nacional de Cajamarca, esto lo hace a través de google maps y para mostrar las distancias y los tiempos de llegada a los paraderos se hará uso de las apis proporcionadas por google maps.

3.1.3.2 BENEFICIOS DE LA APLICACIÓN WEB

Con los beneficios obtenidos en desarrollo del sistema, la calidad servicio de transporte de la Universidad Nacional de Cajamarca aumento significativamente, generando un uso más óptimo en la capacidad de los buses. Esto conlleva, a un ahorro en los gastos de pasajes de los estudiantes universitarios y optimización de sus tiempos. El resumen de los beneficios del sistema se puede esquematizar de la siguiente manera.

Tabla 3: Beneficios de la aplicación web

Beneficios	Características que lo soportan
Disminución en los tiempos de espera	Gracias al monitoreo en tiempo real proporcionada por la aplicación web, los usuarios podrán saber la hora exacta de llegada de los buses y podrán aprovechar mejor sus tiempos.
Ahorro de gastos de transporte	Al tener la información en tiempo real los usuarios podrán optar por usar el transporte de la Universidad Nacional de Cajamarca (gratuito) y no optar por otro medio de transporte (pagado), lo cual le generará a los usuarios un ahorro económico.
Optimización del uso de los buses de la Universidad Nacional de Cajamarca	Los usuarios al tener la información exacta de la llegada de los buses a sus respectivos paraderos optaran por hacer uso de estos y no de otro medio de transporte lo que generará que los buses viajen a una mayor capacidad.

3.1.3.3 DEPENDENCIAS DE LA APLICACIÓN WEB

- La disponibilidad de la aplicación web dependerá de la disponibilidad del servidor web.
- Los usuarios necesitarán de un navegador web sin importar el sistema operativo de su computadora o un smartphone.
- El tiempo de respuesta dependerá del tipo de conexión (velocidad de acceso) que tenga el usuario a internet y del tráfico existente en la red.

3.1.3.4 RECURSOS DEL SOFTWARE Y HARDWARE

El sistema se ha desarrollado usando los recursos de hardware y software ya existentes en la Universidad Nacional de Cajamarca, los cuales ya estaban considerados dentro del presupuesto para la implantación del sistema administrativo; por lo que no será necesario un presupuesto adicional.

3.1.4 REQUERIMIENTOS DE LA APLICACIÓN WEB

En esta sección se busca identificar y documentar los requerimientos de la aplicación web, de una manera que sea entendible tanto para los usuarios finales como para el equipo de desarrollo.

3.1.4.1 REQUISITOS FUNCIONALES

A continuación, se muestran los requerimientos funcionales de la aplicación web, los cuales indican lo que el producto debe hacer.

Tabla 4: Ingresar a la aplicación

Número de requisito	RF01
Nombre de requisito	Ingresar a la aplicación
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Usuario
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Los usuarios al abrir la aplicación web tendrán dos opciones una de ingresar directamente y poder seleccionar su paradero y otra de ingresar con facebook, al seleccionar el ingreso con facebook quedarán registrados los datos obtenidos de facebook, previa solicitud de autorización por parte de la aplicación web hacia el usuario para obtener sus datos, datos que serán almacenados en la base de datos de la aplicación web.

Tabla 5: Seleccionar paradero

Número de requisito	RF02
Nombre de requisito	Seleccionar paradero
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Usuario
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El usuario una vez dentro de la aplicación se le mostrará la lista de todos los paraderos disponibles, de los cuales podrá seleccionar al que se dirigirá a esperar el bus de la Universidad Nacional de Cajamarca, una vez seleccionado se mostrará su paradero dentro del mapa con un marcador de su ubicación, al presionar sobre el marcador del paradero seleccionado se le mostrará dos opciones para poder elegir (como llegar al paradero y cambiar de paradero), aparte de la información que se le muestra (distancia a la que se encuentra y el tiempo que le tomaría en llegar dirigiéndose a pie o en movilidad motorizada).

Tabla 6: Como llegar al paradero

Número de requisito	RF03
Nombre de requisito	Como llegar al paradero
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Usuario
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El usuario al presionar sobre el marcador de su paradero y luego al seleccionar la opción cómo llegar al paradero se le mostrará la ruta que puede seguir para llegar hacia su paradero seleccionado, también se le mostrará la distancia y el tiempo que le tomará llegar a su paradero, por defecto se le mostrará la ruta que debe seguir a pie, pero el usuario podrá modificar esta ruta seleccionando la opción de ver la ruta en movilidad motorizada y viceversa.

Tabla 7: Cambiar de paradero

Número de requisito	RF04
Nombre de requisito	Cambiar de paradero
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Usuario
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El usuario al presionar el marcador de su paradero seleccionado, inicialmente se le mostrará la opción de poder cambiar de paradero, al presionar en la opción cambiar de paradero, se le mostrará la lista de los paraderos disponibles y podrá seleccionar el nuevo paradero que desee, en el mapa será actualizado el marcador del paradero y podría repetir el mismo proceso.

Tabla 8: Monitoreo del bus

Número de requisito	RF05
Nombre de requisito	Monitoreo del bus
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Usuario
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El usuario una vez seleccionado su paradero se le mostrará en el mapa la ubicación del bus que se dirige hacia su paradero seleccionado y se le mostrará también su desplazamiento en tiempo real, al presionar sobre el icono del bus se mostrará la información del tiempo promedio en que el bus llegará al paradero seleccionado y la

distancia en la que se encuentra, esta información se actualizará en tiempo real conforme el bus se vaya desplazando hacia su destino.

Tabla 9: Ver ruta del bus

Número de requisito	RF06
Nombre de requisito	Ver ruta del bus
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Usuario
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El usuario al presionar sobre el marcador del bus y luego al seleccionar la opción ver ruta se le mostrará la ruta que sigue el bus universitario para llegar hacia su paradero final, esta ruta será actualizada en tiempo real con las coordenadas recibidas desde el bus de la Universidad Nacional de Cajamarca enviadas mediante websocket.

Tabla 10: Seleccionar destino

Número de requisito	RF07
Nombre de requisito	Seleccionar destino
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Conductor
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El conductor una vez dentro del sistema se le mostrará toda la lista de los paraderos disponibles y deberá seleccionar el paradero al cual se dirige, una vez seleccionado el aplicativo empezará a enviar los datos necesarios (coordenadas, distancia, tiempo promedio de llegada).

Tabla 11: Envío de coordenadas al servidor

Número de requisito	RF08
Nombre de requisito	Envío de coordenadas al servidor
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Conductor
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El conductor una vez seleccionado su destino, la aplicación web cada vez que detecte un cambio en su posición (latitud, longitud) la enviará hacia el servidor luego de realizar una validación que el cambio con respecto a su última posición sea mayor a 50 metros.

Tabla 12: Envío de coordenadas a los usuarios

Número de requisito	RF09
Nombre de requisito	Envío de coordenadas a los usuarios
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Servidor
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El servidor una vez que recibe las coordenadas de los buses de la Universidad Nacional de Cajamarca, envía estas coordenadas a todos los usuarios que tengan el aplicativo web activo y estén realizando el monitoreo del bus en tiempo real.

Tabla 13: Calculo de datos de monitoreo

Número de requisito	RF10
Nombre de requisito	Calculo de datos de monitoreo
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Usuario
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

La aplicación web utilizada por los usuarios realizará un cálculo interno por medio de la posición del paradero seleccionado (latitud, longitud) y la última posición (latitud, longitud) enviado por el bus de la Universidad Nacional de Cajamarca que se dirige hacia dicho paradero, con esta información calculará el tiempo y la distancia promedio en la que estará llegando a su paradero y mostrará esta información en tiempo real en la interfaz de la aplicación web a los usuarios.

Tabla 14: Usuario

Número de requisito	RF11
Nombre de requisito	Paraderos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Administrador
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

El administrador podrá buscar un paradero existente poder seleccionarlos y editarlos, eliminar un paradero o crear un nuevo paradero, los administradores tienen una contraseña para poder ingresar a la parte administrativa de la aplicación web.

3.1.4.2 REQUISITOS NO FUNCIONALES

3.1.4.2.1 REQUISITOS DE RENDIMIENTO

- La aplicación web puede ser accedida desde cualquier punto con conexión a internet sin la necesidad de instalar algún software externo.
- La aplicación web deberá ser lo más amigable e intuitiva posible para que los usuarios puedan ser capaces de usar todas las funciones disponibles dentro de la aplicación web.

3.1.4.2.2 SEGURIDAD

- Password de usuarios encriptado en la base de datos.
- Login para ingreso al área administrativa de la aplicación web.
- Asignación de determinadas funciones a determinados módulos para los usuarios.
- Restricción de comunicación entre determinados módulos.

3.1.4.2.3 MANTENIBILIDAD

- Copia de seguridad semanales programados para realizarse automáticamente por la aplicación web.
- La aplicación web deberá permitir y soportar agregar, modificar y eliminar nuevos requerimientos.

3.1.4.2.4 ARQUITECTURA

- La arquitectura aplicada en este proyecto es la de cliente – servidor en el que las tareas se reparten entre los proveedores de servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien da la respuesta.
- Instalar la aplicación web en el servidor web para su comunicación con los clientes de la aplicación web.
- En este caso se está utilizando un servidor web.

3.1.5 CASOS DE USO

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica [16]. En esta sección se presentan los casos de uso del sistema.

3.1.5.1 DIAGRAMA DE CASOS DE USO

Un diagrama de casos de uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción [17].

3.1.5.1.1 PERFILES DE USUARIO

Los actores que hacen uso de la aplicación web han sido catalogados de acuerdo a los roles que cumplen en la aplicación web. Las responsabilidades de estos roles fueron descritas en perfiles de usuarios y su clasificación para esta propuesta se muestra a continuación.

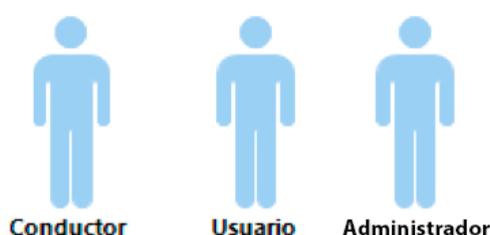


Fig. 6: Actores de la aplicación web

3.1.5.1.2 DESCRIPCIÓN DE ACTORES

Tabla 15: Actor conductor

ACT-0001	Conductor
Versión	1.0 (15/09/2018)
Descripción	Es el encargado de utilizar la aplicación web desarrollada para los conductores, seleccionar el paradero de su destino.
Comentarios	Ninguno

Tabla 16: Actor usuario

ACT-0002	Usuario
Versión	1.0 (15/09/2018)
Descripción	Personas que hacen uso de los buses e la Universidad Nacional de Cajamarca y que realizaran el monitoreo en tiempo real de estos, utilizando la información que se les brinda para tomar las decisiones que crean necesarias.
Comentarios	Ninguno

Tabla 17: Actor administrador

ACT-0003	Administrador
Versión	1.0 (15/09/2018)
Descripción	Este actor representa el administrador del sistema.
Comentarios	Ninguno

3.1.5.2 PAQUETES DE CASOS DE USO

Para una mejor organización y presentación, los casos de uso se han agrupado en dos paquetes (ver fig. 7) de acuerdo a su funcionalidad y propósito en la aplicación web de la siguiente manera:

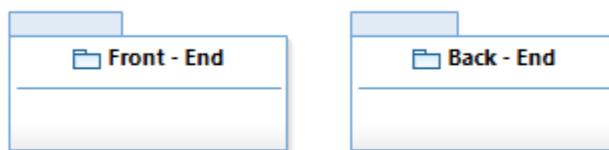


Fig. 7: Diagrama paquetes de caso de uso

A continuación, se lista los casos de uso que integran cada uno de los dos paquetes:

➤ Casos de uso del paquete Front - End

1. Ingresar a la aplicación web.
2. Seleccionar paradero.
3. Como llegar al paradero.
4. Cambiar de paradero.
5. Monitoreo del bus.
6. Ver ruta del bus.
7. Seleccionar destino.

➤ Caso de uso del paquete Back - End

1. Envío de coordenadas al servidor.
2. Recibir coordenadas.
3. Calculo de datos de monitoreo.

3.1.5.3 DESCRIPCIÓN DEL CASO DE USO

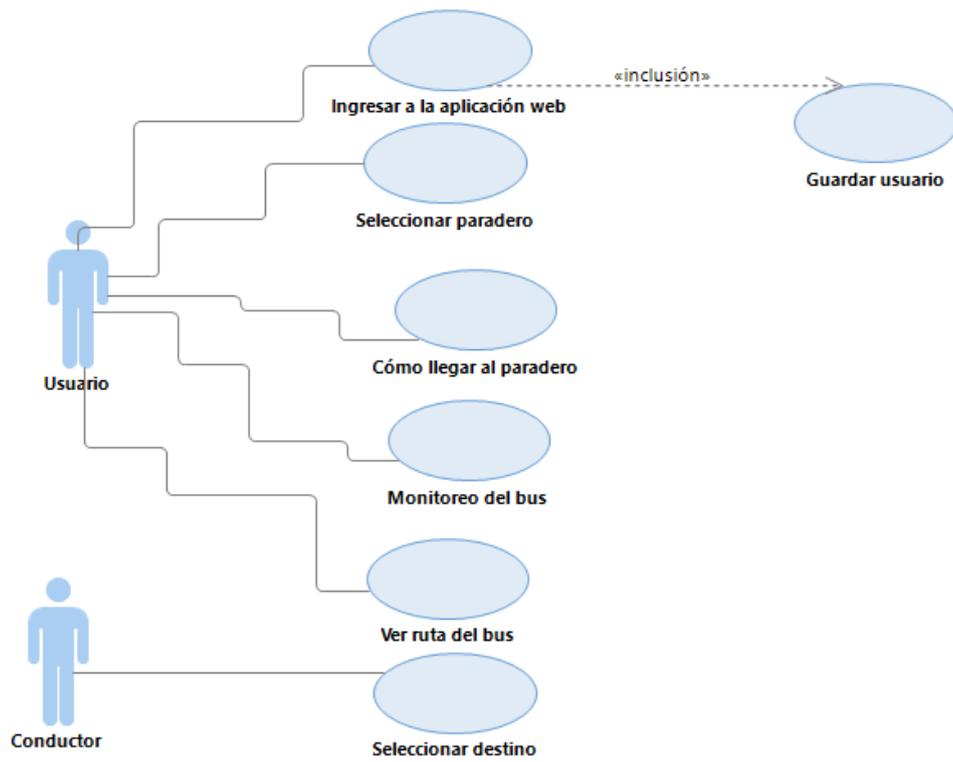


Fig. 8: Diagrama casos de uso front – end

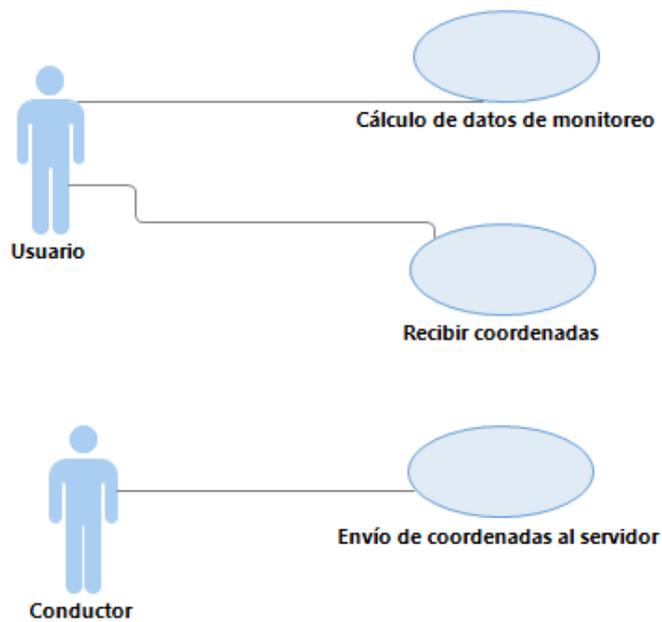


Fig. 9: Diagrama casos de uso back – end

Tabla 18: Caso de uso ingresar a la aplicación web

CU-0001	Ingresar a la aplicación	
Versión	1.0 (15/09/2018)	
Descripción	La aplicación web deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario quiera ingresar a la aplicación web.</i>	
Precondición	El usuario tiene que tener instalado la app en su dispositivo móvil o ingresar a través de un navegador web y contar con acceso a internet.	
Secuencia normal	Paso	Acción
	1	El actor usuario abre la aplicación web por medio de la app instalada en su celular o un navegador web.
	2	La aplicación web <i>le mostrará las opciones de ingresar directamente o ingresar por medio de Facebook.</i>
	3	El actor usuario al <i>selecciona la opción ingresar, la aplicación web salta al paso 7 de lo contrario al seleccionar la opción ingresar con Facebook la aplicación web salta al paso 4.</i>
	4	La aplicación web cargará una ventana emergente de Facebook donde se le mostrará al usuario la opción de aceptar o rechazar el uso de la aplicación de Facebook que obtendrá sus datos
	5	Si el usuario selecciona la opción rechazar la aplicación web saltará al paso 2, de lo contrario si selecciona la opción aceptar seguirá al paso 6.
	6	Se almacenarán en la base de datos, los datos del usuario obtenidos del Facebook con el que se logueo.
	7	La aplicación web <i>muestra la lista de los paraderos disponibles.</i>
Pos condición	El usuario a ingresado correctamente	
Excepciones	Paso	Acción

	5	Si el usuario no desea otorgar los permisos a la aplicación web, la aplicación web, <i>salta al paso 2</i> , a continuación, este caso de uso <i>continúa</i>
Rendimiento	Paso	Tiempo máximo
	2	1 segundo(s)
	4	1 segundo(s)
	6	1 segundo(s)
Frecuencia esperada	200 veces por día	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	En construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 19: Caso de uso seleccionar paradero

CU-0002	Seleccionar Paradero	
Versión	1.0 (15/09/2018)	
Descripción	La aplicación web deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario quiera seleccionar un paradero</i> .	
Precondición	El usuario tiene que tener instalado la app en su dispositivo móvil o ingresar a través de un navegador web, contar con acceso a internet y haber ingresado a la aplicación web.	
Secuencia normal	Paso	Acción
	1	La aplicación web le muestra al usuario la lista de los paraderos disponibles.
	2	El usuario selecciona un paradero de la lista mostrada.
	3	La aplicación web <i>muestra el mapa con la ubicación del usuario y del paradero seleccionado</i> .
Poscondición	El paradero se muestra con un marcador en el mapa	
Excepciones	Paso	Acción

	5	<i>Si el usuario no desea seleccionar ningún paradero, puede cerrar la aplicación</i>
Rendimiento	Paso	Tiempo máximo
	1	1 segundo(s)
	2	10 segundo(s)
	3	1 segundo(s)
Frecuencia esperada	200 veces por día	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	En construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 20: Caso de uso como llegar al paradero

CU-0003	Como Llegar al Paradero	
Versión	1.0 (15/09/2018)	
Descripción	La aplicación web deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario quiera saber cómo llegar a su paradero seleccionado.</i>	
Precondición	El usuario tiene que haber seleccionado un paradero.	
Secuencia normal	Paso	Acción
	1	La aplicación web le muestra al usuario su paradero en el mapa con un marcador.
	2	El usuario al presionar sobre el marcador de su paradero se le mostrará una ventana emergente donde tendrá la opción de elegir “cómo llegar”.
	3	El actor usuario al <i>selecciona la opción cómo llegar, la aplicación web calcula la ruta que debe seguir desde su posición actual hasta el paradero y se la mostrará.</i>
Poscondición	La ruta se muestra en el mapa correctamente.	
Excepciones	Paso	Acción

	4	Si el usuario no desea que se le muestre la ruta de cómo llegar, la aplicación web salta al paso 1, a continuación, este caso de uso <i>continúa</i>
Rendimiento	Paso	Tiempo máximo
	1	3 segundo(s)
	2	10 segundo(s)
	3	15 segundo(s)
Frecuencia esperada	200 veces por día	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	En construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 21: Caso de uso cambiar paradero

CU-0004	Cambiar de Paradero	
Versión	1.0 (15/09/2018)	
Descripción	La aplicación web deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario quiera cambiar de paradero</i> .	
Precondición	El usuario tiene que haber seleccionado un paradero.	
Secuencia normal	Paso	Acción
	1	La aplicación web le muestra al usuario su paradero en el mapa con un marcador.
	2	El usuario al presionar sobre el marcador de su paradero se le mostrará una ventana emergente donde tendrá la opción de elegir “cambiar paradero”.
	3	El actor usuario al <i>selecciona la opción cambiar paradero</i> , la aplicación web le mostrará en una ventana emergente la lista de los paraderos disponibles.
	4	El usuario selecciona el nuevo paradero que desea.

	5	La aplicación web elimina el marcador del antiguo paradero y agrega un nuevo marcador con el nuevo paradero seleccionado.
Poscondición	El paradero se cambió correctamente.	
Excepciones	Paso	Acción
	4	<i>Si el usuario no desea cambiar de paradero, la aplicación web salta al paso 1, a continuación, este caso de uso continúa</i>
Rendimiento	Paso	Tiempo máximo
	1	3 segundo(s)
	2	10 segundo(s)
	3	5 segundo(s)
	4	10 segundo(s)
	5	10 segundo(s)
Frecuencia esperada	100 veces por día	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	En construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 22: Caso de uso monitoreo de bus

CU-0005	Monitoreo de Bus	
Versión	1.0 (15/09/2018)	
Descripción	La aplicación web deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario quiera monitorear un bus</i> .	
Precondición	El usuario tiene que haber seleccionado un paradero.	
Secuencia normal	Paso	Acción
	1	La aplicación web le muestra al usuario el bus que se dirige a su paradero dentro del mapa con un

		marcador y la ruta que está siguiendo en tiempo real.
	2	El usuario al presionar sobre el marcador del bus se le mostrará una ventana emergente donde se podrá observar la información del bus (tiempo y distancia) de llegada al paradero, información que se irá actualizando en tiempo real.
Poscondición	La información se muestra correctamente.	
Excepciones	Paso	Acción
	4	Si el usuario no desea monitorear el bus, la aplicación web salta al paso 1, a continuación, este caso de uso continúa
Rendimiento	Paso	Tiempo máximo
	1	3 segundo(s)
	2	3 segundo(s)
Frecuencia esperada	200 veces por día	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	En construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 23: Caso de uso ver ruta del bus

CU-0006	Ver ruta del bus	
Versión	1.0 (15/09/2018)	
Descripción	La aplicación web deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario quiera saber la ruta que está siguiendo el bus</i> .	
Precondición	El usuario tiene que haber seleccionado un paradero.	
Secuencia normal	Paso	Acción
	1	La aplicación web le muestra al usuario el bus en el mapa con un marcador.

	2	El usuario al presionar sobre el marcador del bus se le mostrará una ventana emergente donde tendrá la opción de elegir “ver ruta”.
	3	El actor usuario al <i>selecciona la opción ver ruta, la aplicación web calcula la ruta que sigue el bus desde su ubicación actual hasta su paradero final.</i>
Poscondición	La ruta se muestra en el mapa correctamente.	
Excepciones	Paso	Acción
	4	Si <i>el usuario no desea que se le muestre la ruta que sigue el bus universitario, la aplicación web salta al paso 1, a continuación, este caso de uso continúa</i>
Rendimiento	Paso	Tiempo máximo
	1	3 segundo(s)
	2	10 segundo(s)
	3	15 segundo(s)
Frecuencia esperada	200 veces por día	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	En construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 24: Caso de uso seleccionar paradero

CU-0007	Seleccionar Destino	
Versión	1.0 (15/09/2018)	
Descripción	La aplicación web deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el conductor empiece su labor.</i>	
Precondición	El conductor tiene que haber ingresado a la aplicación desde el dispositivo móvil implementado en los buses.	
Secuencia normal	Paso	Acción
	1	La aplicación web le muestra al conductor la lista de los paraderos disponibles.

	2	El conductor seleccionará el paradero al cual se dirigirá.
Poscondición	Paradero seleccionado correctamente.	
Excepciones	Paso	Acción
	4	<i>Si el conductor no selecciona un paradero, la aplicación web salta al paso 1, a continuación, este caso de uso continúa</i>
Rendimiento	Paso	Tiempo máximo
	1	3 segundo(s)
	2	10 segundo(s)
Frecuencia esperada	200 veces por día	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	En construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 25: Caso de uso envío de coordenadas al servidor

CU-0008	Envío de Coordenadas al Servidor	
Versión	1.0 (15/09/2018)	
Descripción	La aplicación web deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el conductor envía las coordenadas del bus al servidor.</i>	
Precondición	La aplicación web debe estar conectada a internet.	
Secuencia normal	Paso	Acción
	1	El conductor ingresa a la pantalla de monitoreo donde se le muestra un marcador con su ubicación actual.
	1	La aplicación web detecta un cambio de posición al recibir nuevas coordenadas (latitud, longitud).

	2	La aplicación web verifica que entre las nuevas coordenadas (latitud, longitud) recibidas y las últimas almacenadas hay una diferencia mayor a 50 metros.
	3	La aplicación web almacena internamente las nuevas coordenadas recibidas.
	4	La aplicación web envía las nuevas coordenadas al servidor.
	5	Se actualiza y muestra al conductor su nueva posición.
Poscondición	Coordenadas enviadas correctamente.	
Excepciones	Paso	Acción
	4	Si la aplicación web verifica que la distancia entre las coordenadas (latitud, longitud) recibidas y las últimas almacenadas es menor a 50 metros, <i>la aplicación web salta al paso 1</i> , a continuación, este caso de uso <i>continúa</i>
Rendimiento	Paso	Tiempo máximo
	1	5 segundo(s)
	2	3 segundo(s)
	3	3 segundo(s)
	4	3 segundo(s)
Frecuencia esperada	200 veces por día	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	En construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 26: Caso de uso recibir coordenadas

CU-0009	Recibir Coordenadas	
Versión	1.0 (15/09/2018)	
Descripción	La aplicación web deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario recibe las nuevas coordenadas.	
Precondición	La aplicación web debe estar realizando el monitoreo en tiempo real de los buses.	
Secuencia normal	Paso	Acción
	1	El usuario ingresa a la pantalla de monitoreo del bus.
	2	La aplicación web envía su identificador y este se registra en el servidor.
	3	El socket se activó y se mantiene escuchando a la espera de la recepción de las nuevas coordenadas.
	4	La aplicación web recibe las nuevas coordenadas enviados desde el servidor mediante la comunicación bidireccional websocket
Poscondición	Coordenadas recibidas correctamente.	
Excepciones	Ninguna	
Rendimiento	Paso	Tiempo máximo
	1	1 segundo(s)
	2	3 segundo(s)
	3	3 segundo(s)
Frecuencia esperada	200 veces por día	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	En construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 27: Caso de uso cálculo de datos de monitoreo

CU-0010	Calculo de Datos de Monitoreo	
Versión	1.0 (15/09/2018)	
Descripción	La aplicación web deberá comportarse tal como se describe en el siguiente caso de uso cuando calcula los datos de monitoreo en tiempo real.	
Precondición	La aplicación web debe estar realizando el monitoreo en tiempo real de los buses.	
Secuencia normal	Paso	Acción
	1	El usuario ingresa a la pantalla de monitoreo del bus.
	2	La aplicación web recibe las nuevas coordenadas del servidor en tiempo real mediante la comunicación bidireccional websocket.
	2	La aplicación web calcula el nuevo tiempo promedio de llegada del bus en monitoreo con las nuevas coordenadas obtenidas y las coordenadas del paradero.
	3	La aplicación web calcula la nueva distancia de llegada del bus en monitoreo con las nuevas coordenadas obtenidas y las coordenadas del paradero.
	4	La aplicación muestra en interfaz los nuevos datos actualizados, calculados con las nuevas coordenadas obtenidas.
Poscondición	Mostrar nueva información calculada en tiempo real.	
Excepciones	Ninguna	
Rendimiento	Paso	Tiempo máximo
	1	1 segundo(s)
	2	3 segundo(s)
	3	3 segundo(s)
	4	1 segundo(s)

Frecuencia esperada	200 veces por día
Importancia	Vital
Urgencia	Inmediatamente
Estado	En construcción
Estabilidad	Alta
Comentarios	Ninguno

Tabla 28: Caso de uso nuevo paradero

UC-0011	Nuevo Paradero	
Versión	1.0 (15/09/2018)	
Descripción	La aplicación deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador registre un nuevo paradero.	
Precondición	El usuario tiene que estar identificado como administrador del sistema.	
Secuencia normal	Paso	Acción
	1	El actor Administrador solicita a la aplicación web comenzar el proceso añadir un nuevo paradero.
	2	La aplicación web solicita los datos respectivos al administrador.
	3	El actor Administrador proporciona los datos requeridos.
	4	La aplicación web almacena los datos, introduce un identificador único para el paradero de forma automática y le indica al administrador que la acción se ha llevado con éxito.
Postcondición	El paradero se ha añadido correctamente a la base de datos.	
Excepciones	Paso	Acción
	4	Si comprueba que el paradero ya ha sido introducido, la aplicación web vuelve al paso 2, a continuación, este caso de uso continúa
Rendimiento	Paso	Tiempo máximo
	2	1 segundo(s)
	4	1 segundo(s)

Frecuencia esperada	1 vez anual
Importancia	vital
Urgencia	inmediatamente
Estado	en construcción
Estabilidad	alta
Comentarios	Ninguno

Tabla 29: Caso de uso eliminar paradero

UC-0012	Eliminar Paradero	
Versión	1.0 (15/09/2018)	
Descripción	La aplicación deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador solicite eliminar un paradero.	
Precondición	El usuario tiene que estar identificado como administrador.	
Secuencia normal	Paso	Acción
	1	El actor Administrador solicita a la aplicación web comenzar el proceso de borrar un paradero.
	2	La aplicación web solicita al administrador que seleccione el paradero que quiere borrar.
	3	El actor Administrador selecciona el paradero que quiere borrar de la lista de paradero actualmente introducidos.
	4	El sistema muestra una descripción detallada del paradero a borrar
	5	El actor Administrador verifica la eliminación del paradero.
	6	El sistema elimina el paradero seleccionado y le indica al administrador que la operación ha finalizado con éxito.
Postcondición	El paradero se ha eliminado correctamente.	
Excepciones	Paso	Acción
	5	Si el paradero no es el que quiere borrar el administrador, el actor Administrador selecciona otro paradero diferente de la lista, salta al paso 3, a continuación este caso de uso continúa
Rendimiento	Paso	Tiempo máximo

	2	1 segundo(s)
	4	1 segundo(s)
	6	1 segundo(s)
Frecuencia esperada	1 veces por año(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 30: Caso de uso modificar paradero

UC-0004	Modificar Paradero	
Versión	1.0 (15/09/2018)	
Descripción	La aplicación web deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador quiera modificar algún paradero de los paraderos registrados.	
Precondición	El usuario tiene que estar identificado como administrador.	
Secuencia normal	Paso	Acción
	1	El actor Administrador solicita a la aplicación web comenzar el proceso de modificar los datos de un paradero.
	2	La aplicación web solicita al administrador que seleccione el paradero a modificar.
	3	El actor Administrador selecciona el paradero entre la lista que le muestra la aplicación web.
	4	La aplicación web muestra todas las características del paradero
	5	El actor administrador modifica los valores oportunos cambiando los valores actuales por los nuevos.
	6	La aplicación web modifica el paradero con los nuevos valores y le indica al administrador que la operación ha finalizado con éxito.
Postcondición	El paradero ha sido modificado con los nuevos valores	

Excepciones	Paso	Acción
	5	Si el paradero no es el deseado, el actor administrador buscará un nuevo paradero de la lista que proporciona la aplicación web, salta al paso 3, a continuación este caso de uso continúa
Rendimiento	Paso	Tiempo máximo
	2	1 segundo(s)
	4	1 segundo(s)
	6	1 segundo(s)
Frecuencia esperada	1 veces por año(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	alta	
Comentarios	Ninguno	

Tabla 31: Caso de uso buscar paradero

UC-0005	Buscar Paradero	
Versión	1.0 (15/09/2018)	
Descripción	La aplicación web deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador quiera buscar algún dato de los paraderos.	
Precondición	El usuario tiene que estar identificado como administrador.	
Secuencia normal	Paso	Acción
	1	El actor Administrador solicita a la aplicación web comenzar el proceso de buscar los datos de un paradero.
	2	El sistema solicita al administrador que seleccione el paradero a buscar.
	3	El actor Administrador selecciona el paradero entre la lista que le muestra la aplicación web.
	4	La aplicación web muestra todas las características del paradero.

Postcondición	El paradero ha sido encontrado	
Excepciones	Paso	Acción
	5	Si el paradero no es el deseado, el actor Administrador buscará un nuevo paradero de la lista que proporciona la aplicación web, salta al paso 3, a continuación este caso de uso continúa
Rendimiento	Paso	Tiempo máximo
	2	1 segundo(s)
	4	1 segundo(s)
	6	1 segundo(s)
Frecuencia esperada	1 vez al año	
Importancia	vital	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

3.1.6 DIAGRAMA DE ACTIVIDADES

3.1.6.1 DIAGRAMA DE ACTIVIDADES DEL PAQUETE FRONT – END

3.1.6.1.1 INGRESAR A LA APLICACIÓN WEB

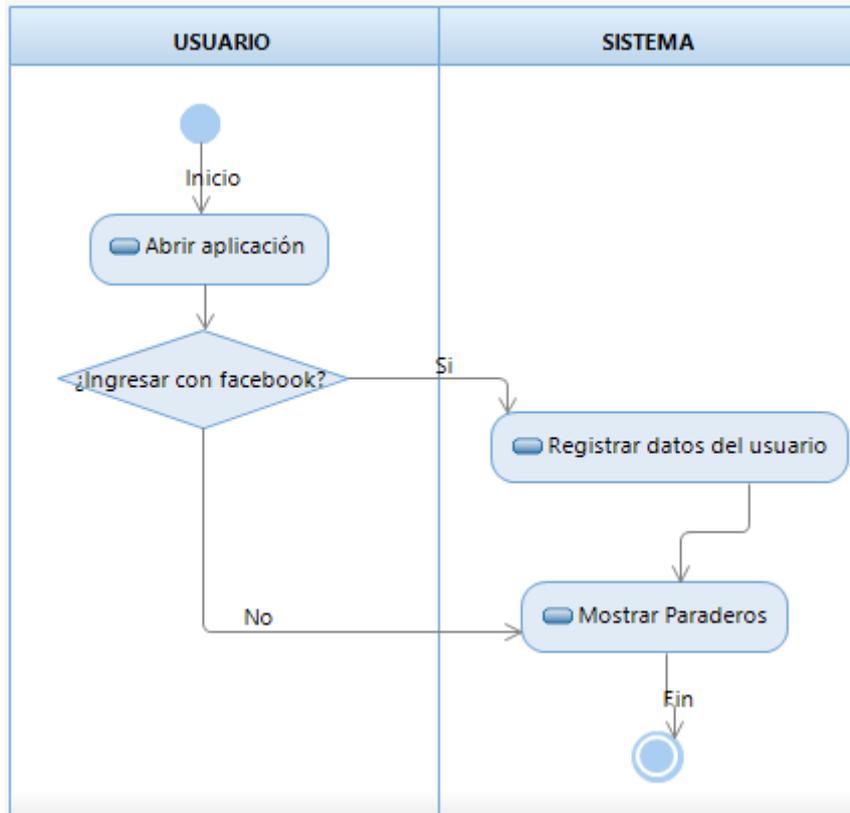


Fig. 10: Diagrama de actividad ingresar a la aplicación

3.1.6.1.2 SELECCIONAR PARADERO

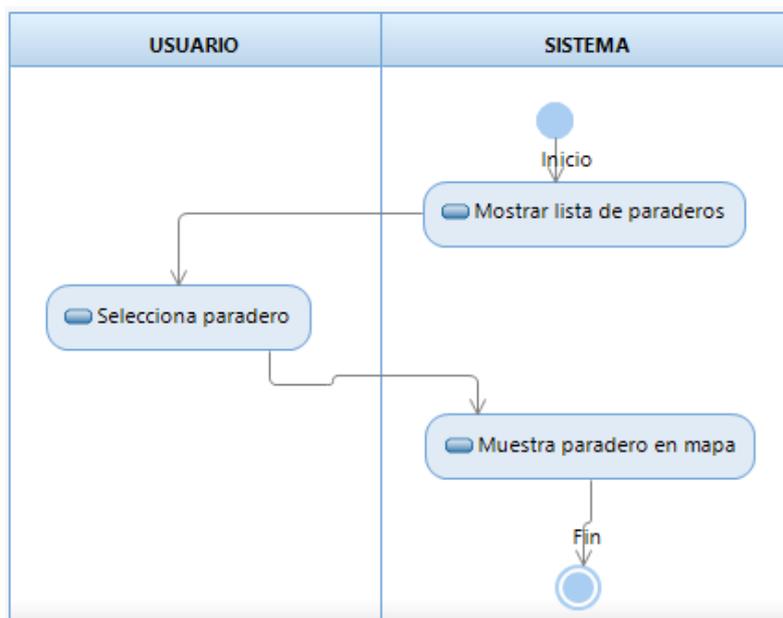


Fig. 11: Diagrama de actividad seleccionar paradero

3.1.6.1.3 CÓMO LLEGAR AL PARADERO

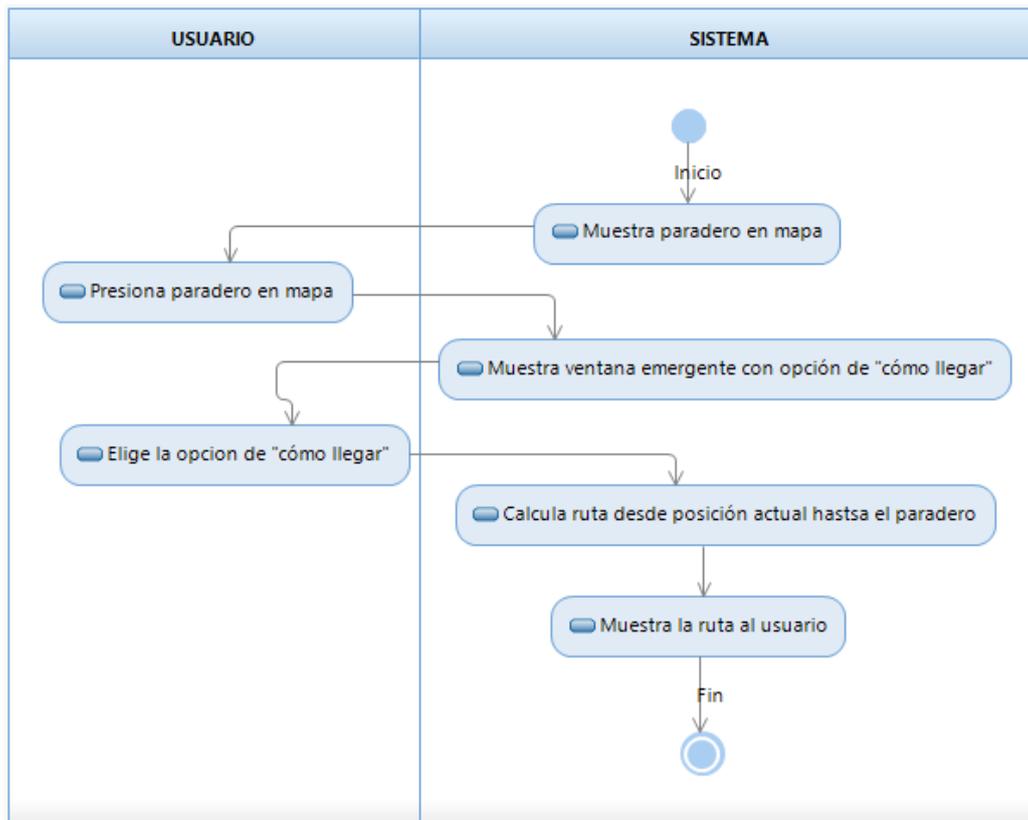


Fig. 12: Diagrama de actividad cómo llegar al paradero

3.1.6.1.4 CAMBIAR DE PARADERO

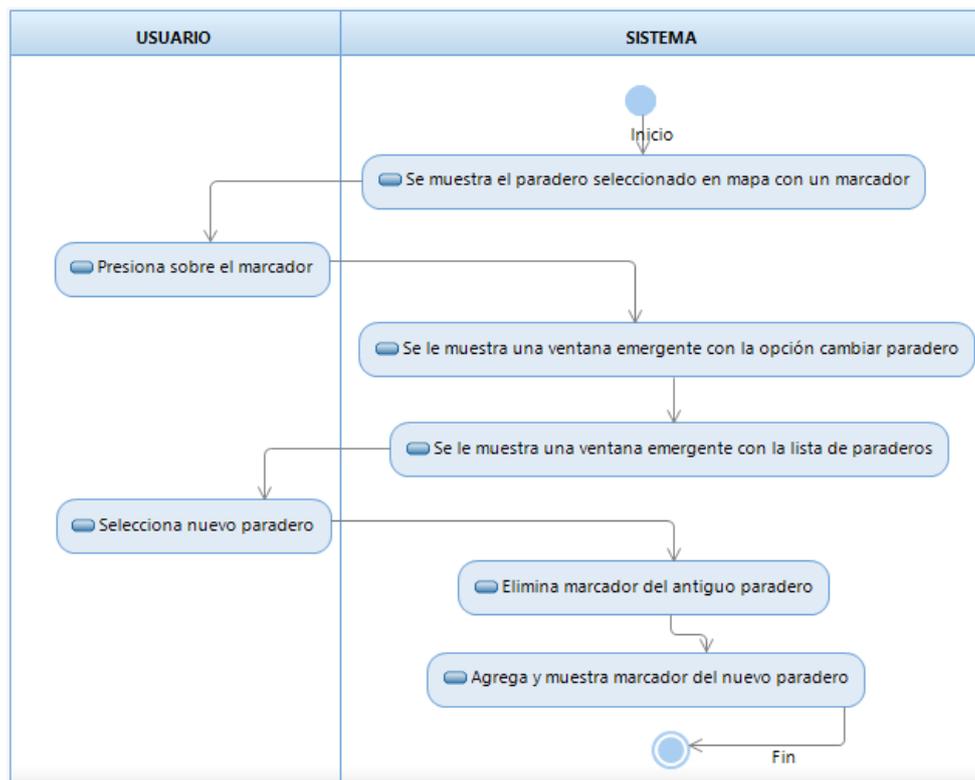


Fig. 13: Diagrama de actividad cambiar de paradero

3.1.6.1.5 MONITOREO DEL BUS

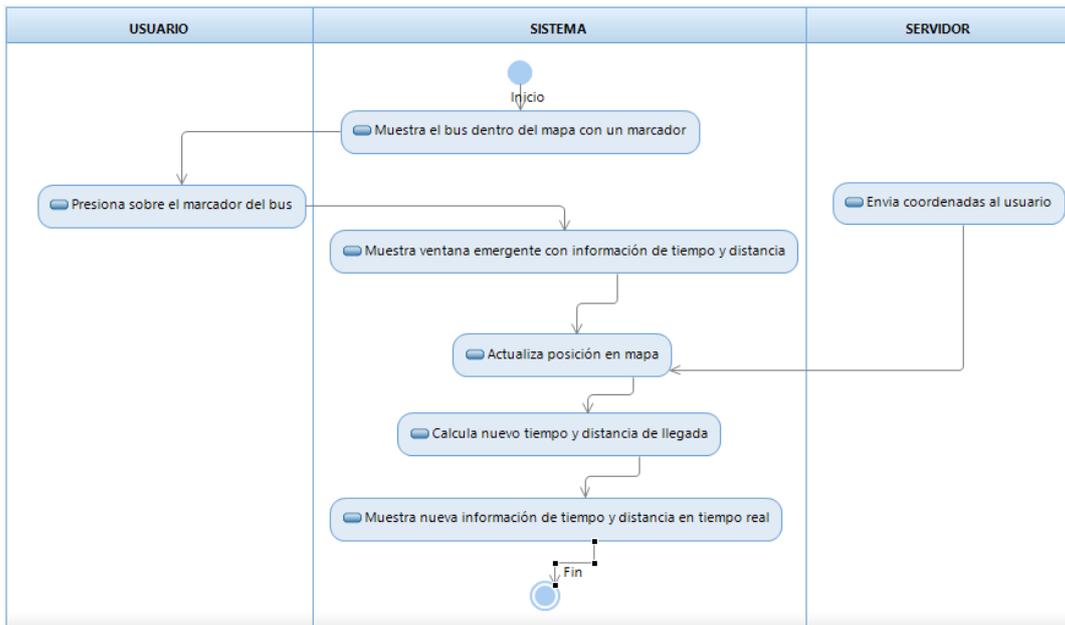


Fig. 14: Diagrama de actividad monitoreo del bus

3.1.6.1.6 VER RUTA DEL BUS

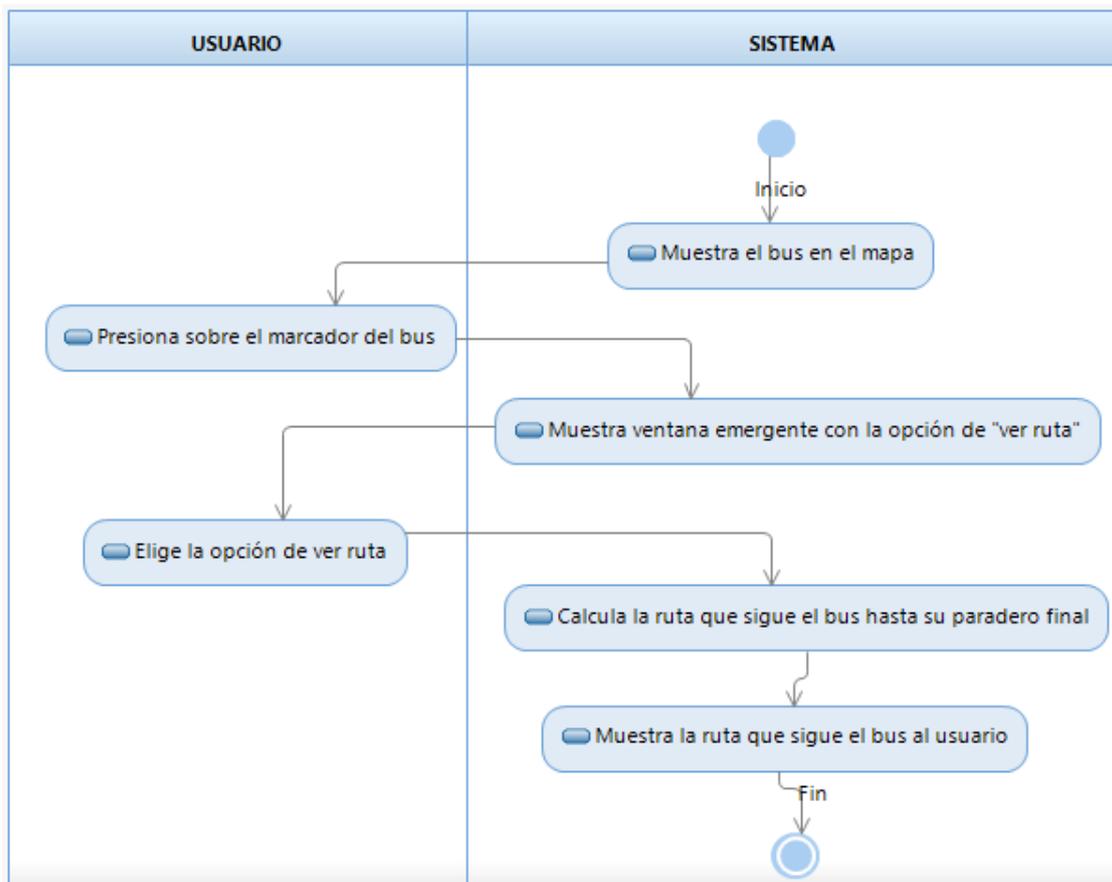


Fig. 15: Diagrama de actividad ver ruta del bus

3.1.6.1.7 SELECCIONAR DESTINO

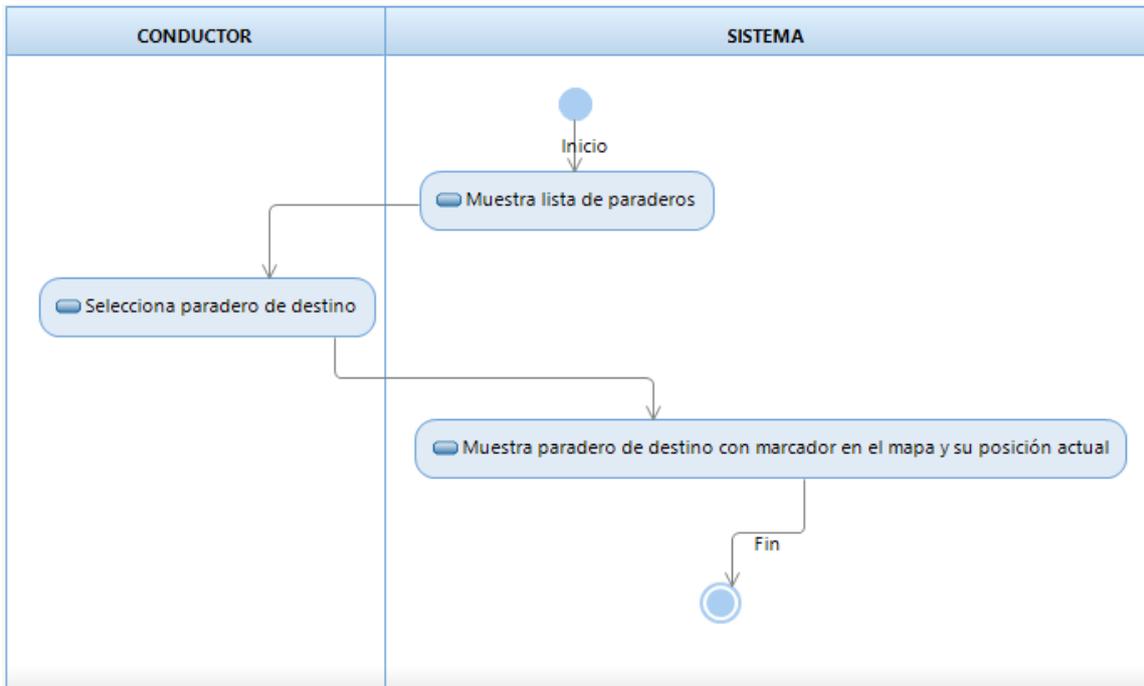


Fig. 16: Diagrama de actividad seleccionar destino

3.1.6.2 DIAGRAMA DE ACTIVIDADES DEL PAQUETE BACK – END

3.1.6.2.1 ENVÍO DE COORDENADAS AL SERVIDOR

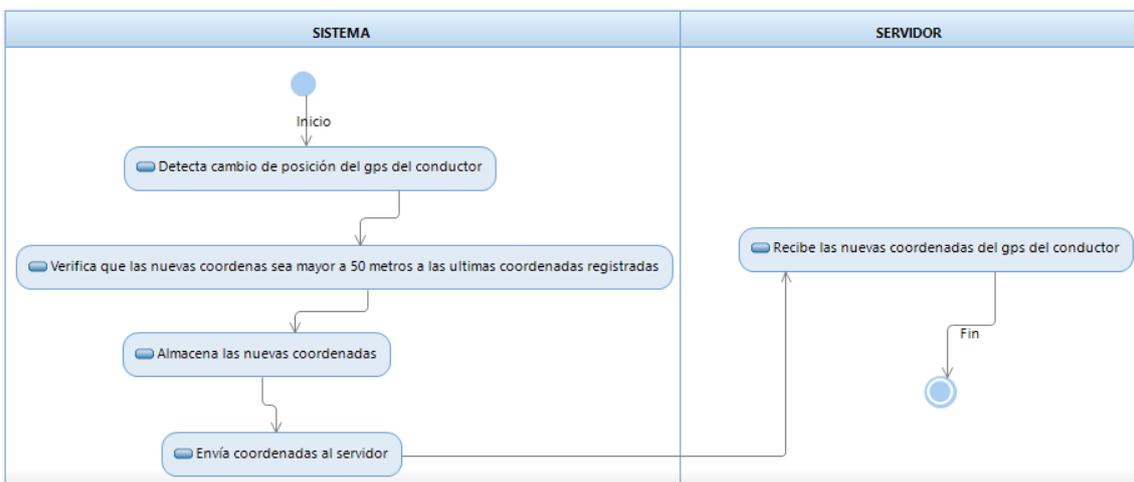


Fig. 17: Diagrama de actividad envío de coordenadas al servidor

3.1.6.2 ENVÍO DE COORDENADAS AL USUARIO

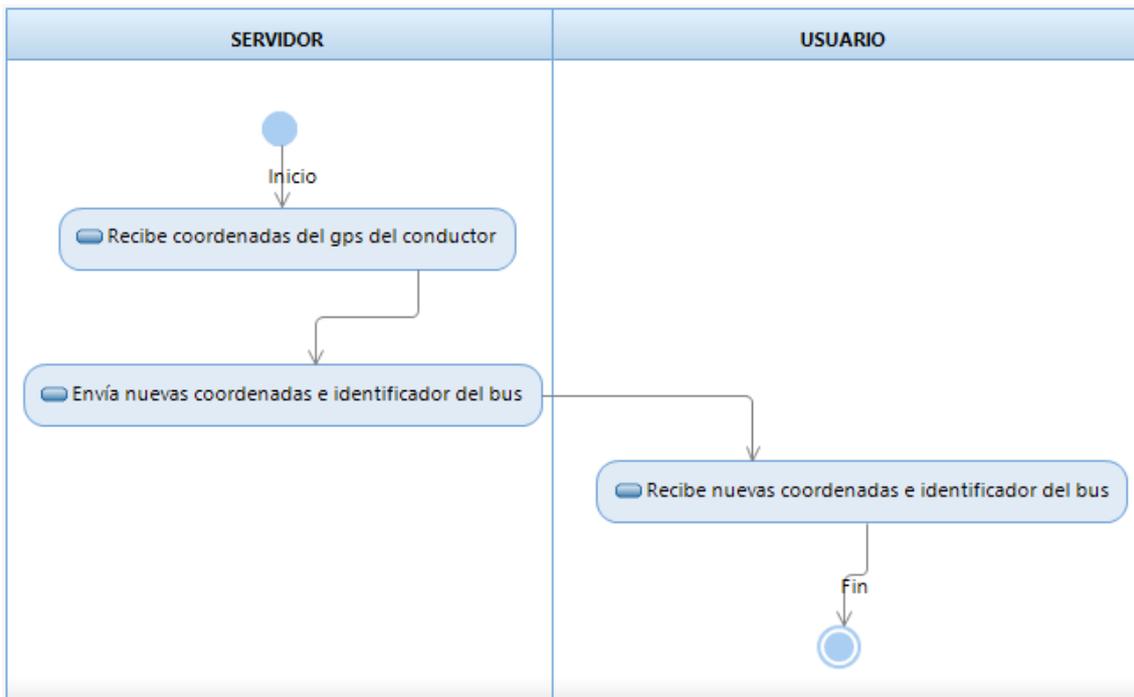


Fig. 18: Diagrama de actividad envío de coordenadas al usuario

3.1.6.2.3 RECIBIR COORDENADAS

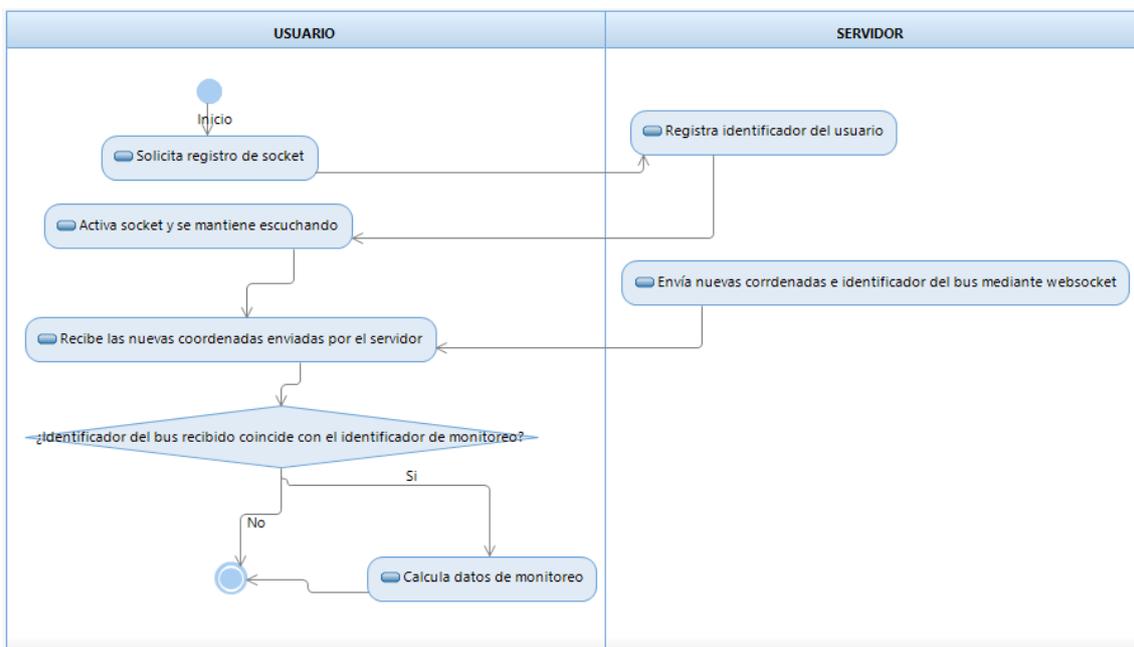


Fig. 19: Diagrama de actividad recibir coordenadas

3.1.6.2.4 CALCULO DE DATOS DE MONITOREO

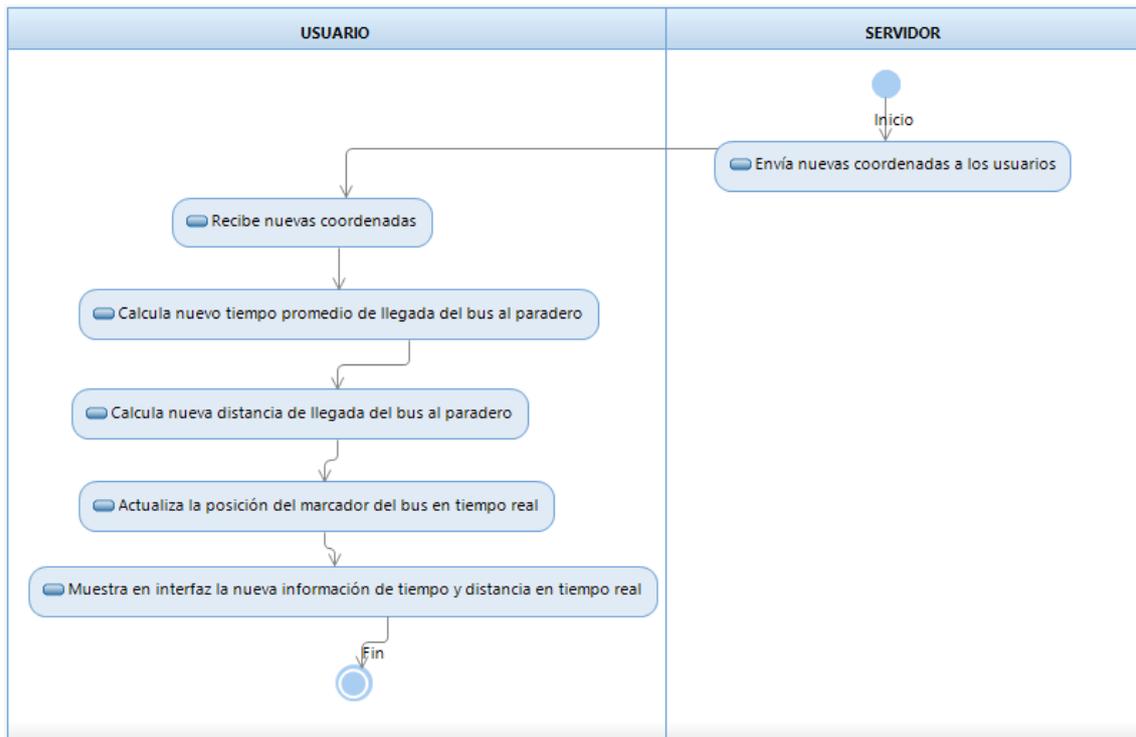


Fig. 20: Diagrama de actividad cálculo de datos de monitoreo

3.1.7 MATRIZ DE TRAZABILIDAD

Tabla 32: Matriz de trazabilidad casos de uso - requisitos funcionales

CASOS DE USO	1. Ingresar a la aplicación	2. Seleccionar Paradero	3. Cómo Llegar al Paradero	4. Cambiar de Paradero	5. Monitoreo de Bus	6. Ver ruta del bus	7. Seleccionar Destino	8. Envío de Coordenadas al Servidor	9. Envío de Coordenadas a los Usuarios	10. Calculo de Datos de Monitoreo	11. Paradero
RF01	X										
RF02		X									
RF03			X								
RF04				X							
RF05					X						
RF06						X					
RF07							X				
RF08								X			
RF09									X		
RF10										X	
RF11											X

3.1.8 MODELO CONCEPTUAL

Un modelo conceptual explica los conceptos significativos en un dominio del problema, es el artefacto más importante a crear durante el análisis orientado a objetos [22].

En esta sección se identifican dichos conceptos.

3.1.8.1 DIAGRAMA DE ANÁLISIS DE CLASES

3.1.8.1.1 INTERFACES



Fig. 21: Diagrama de Análisis de clases – Interfaces

3.1.8.1.2 CONTROLES

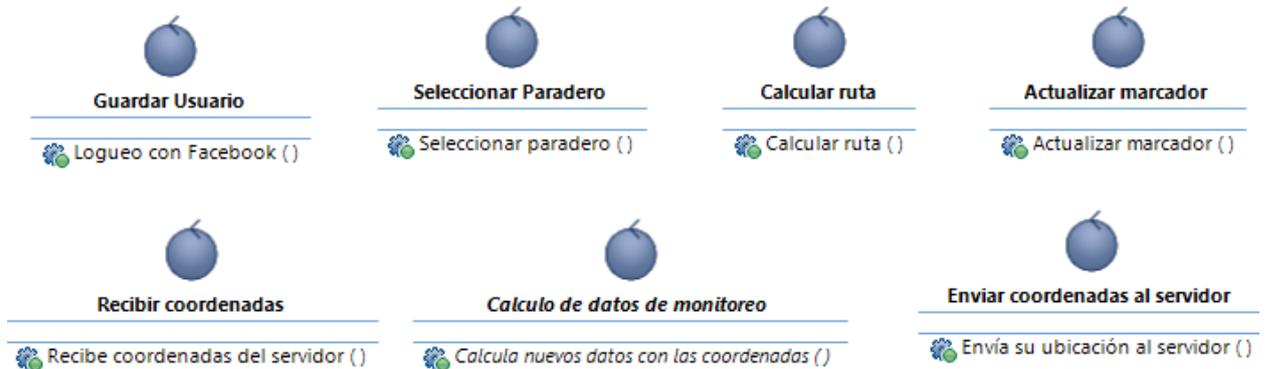


Fig. 22: Diagrama de Análisis de clases – Controles

3.1.8.2 DIAGRAMA DE CLASES DE DISEÑO

El conjunto de todas las clases usadas en el modelo conceptual, junto con sus relaciones, forma el Diagrama de Clases de Diseño. Un Diagrama de Clases de Diseño muestra la especificación para las clases software de una aplicación [23].

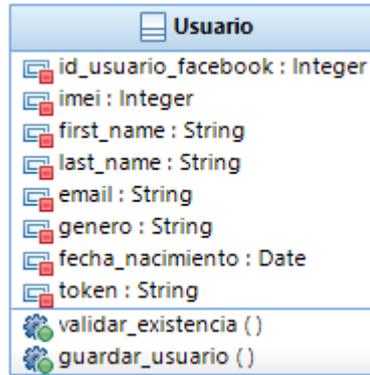


Fig. 23: Diagrama de clase guardar usuario

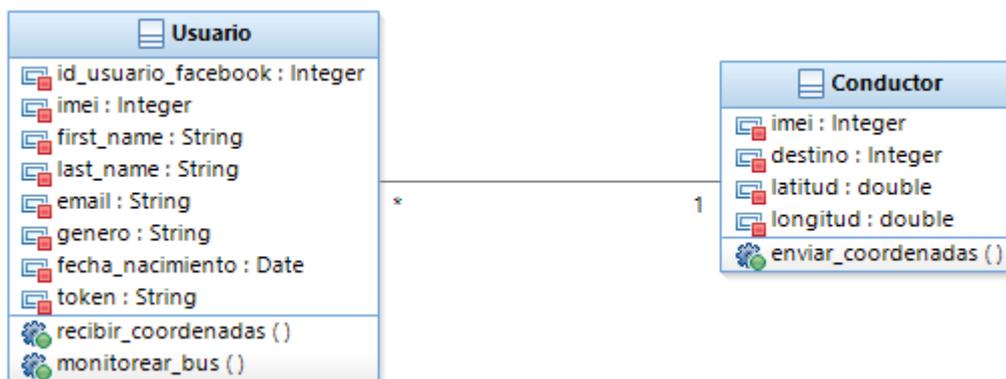


Fig. 24: Diagrama de clase monitoreo

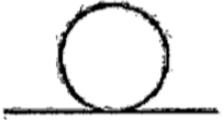
3.1.8.3 DIAGRAMA DE SECUENCIA DE SUCESOS DEL SISTEMA

El diagrama de secuencia es una representación gráfica que muestra, un determinado escenario de un caso de uso, los eventos generados por actores externos, su orden y los eventos internos del sistema.

El diagrama de sucesos pone especial énfasis en el orden y el momento en que se envían los mensajes a los objetos [23].

Los objetos utilizados corresponden a clases que permitirán diferenciar una arquitectura de 3 capas (ver sección 3.1.11 Arquitectura del sistema). La asignación de responsabilidades se realizó siguiendo los lineamientos del patrón Entidad – Control – Frontera (ECB) que es una variación del Patrón MVC y cuya representación se describe en la siguiente tabla [22].

Tabla 33: Estereotipos de clases utilizados en los diagramas de secuencia

Tipo	Descripción	Imagen
Boundary Class (Interfaz)	Describe a objetos que representan a la vista correspondiente a las interfaces y que interactúan con los actores del sistema en el mundo exterior. Representan a las páginas web del sistema.	
Control Class (Control)	Describe a objetos que implementan el comportamiento del sistema, delega el trabajo a otros objetos y administra las interacciones transaccionales.	
Entity Class (Entidad)	Describe a los objetos que son pasivos. Ellos no inician las interacciones, encapsulan la lógica del negocio y muestran una estructura de datos.	

3.1.8.3.1 DIAGRAMA DE SECUENCIA INGRESAR A LA APLICACIÓN

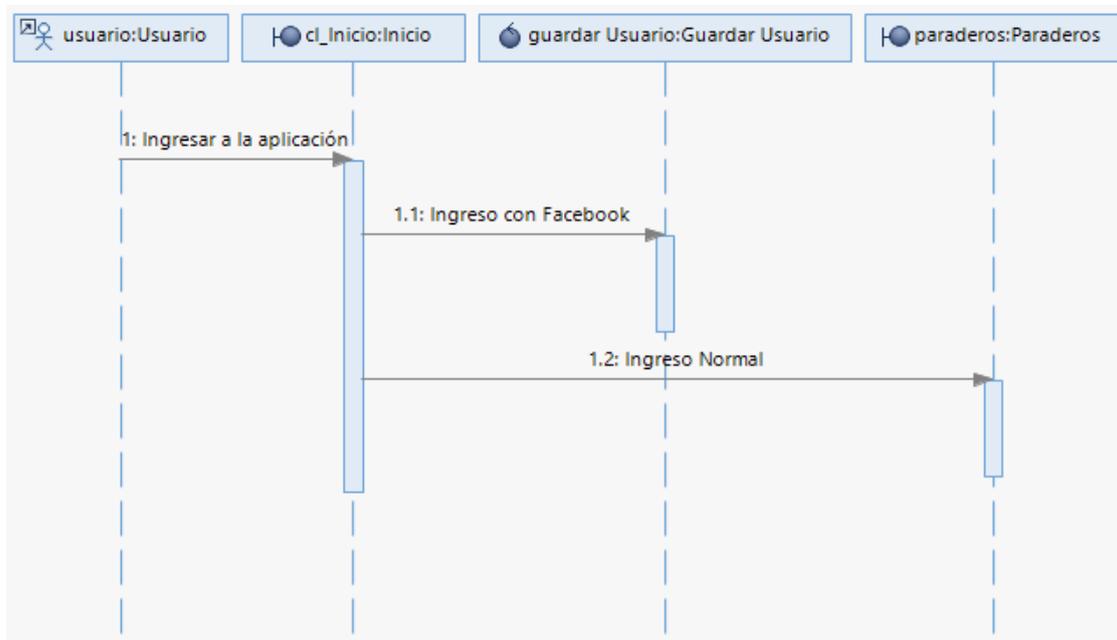


Fig. 25: Diagrama de secuencia ingresar a la aplicación

3.1.8.3.2 DIAGRAMA DE SECUENCIA SELECCIONAR PARADERO

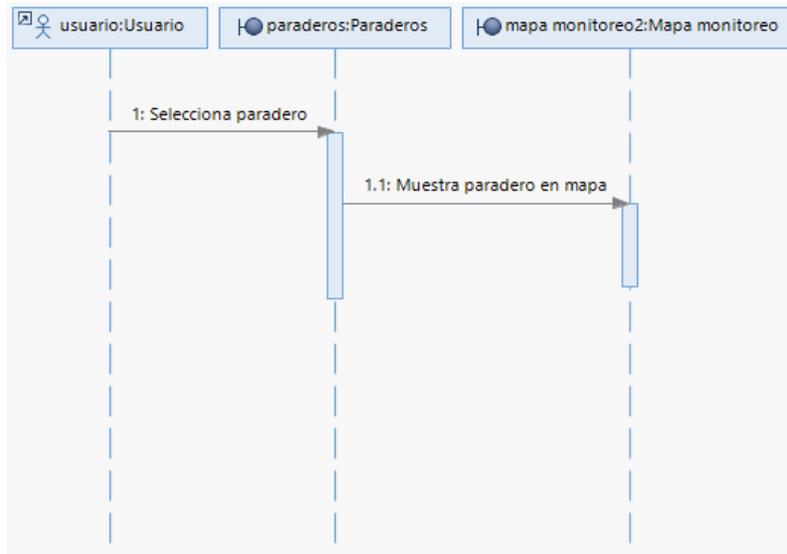


Fig. 26: Diagrama de secuencia seleccionar paradero

3.1.8.3.3 DIAGRAMA DE SECUENCIA CÓMO LLEGAR AL PARADERO

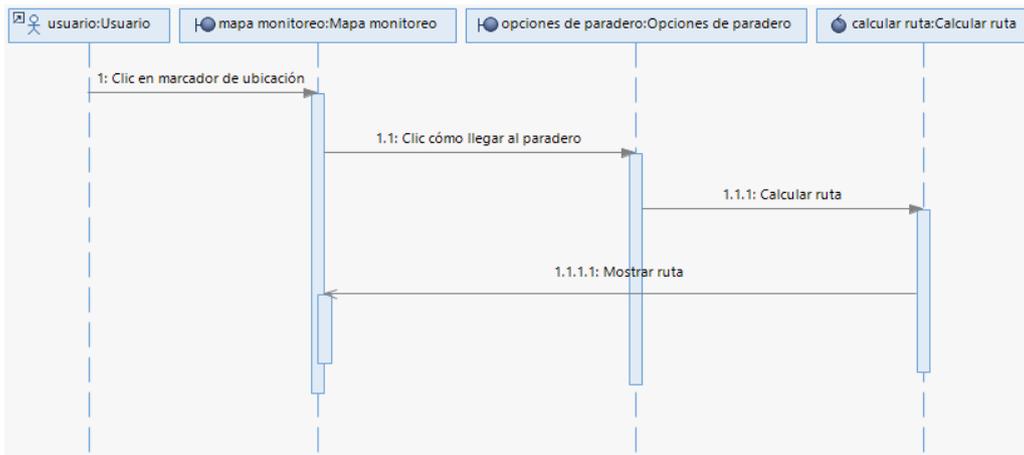


Fig. 27: Diagrama de secuencia cómo llegar al paradero

3.1.8.3.4 DIAGRAMA DE SECUENCIA CAMBIAR DE PARADERO

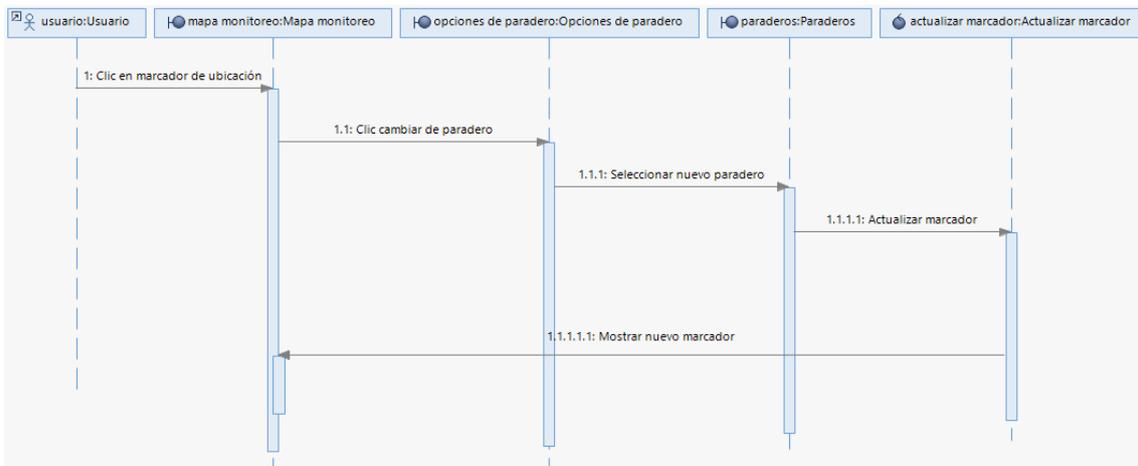


Fig. 28: Diagrama de secuencia cambiar de paradero

3.1.8.3.5 DIAGRAMA DE SECUENCIA MONITOREO DE BUS

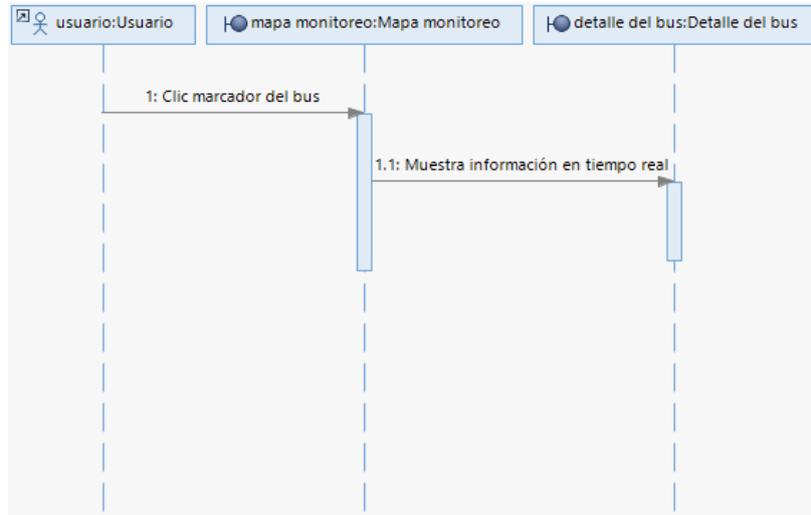


Fig. 29: Diagrama de secuencia monitoreo de bus

3.1.8.3.6 DIAGRAMA DE SECUENCIA VER RUTA DEL BUS

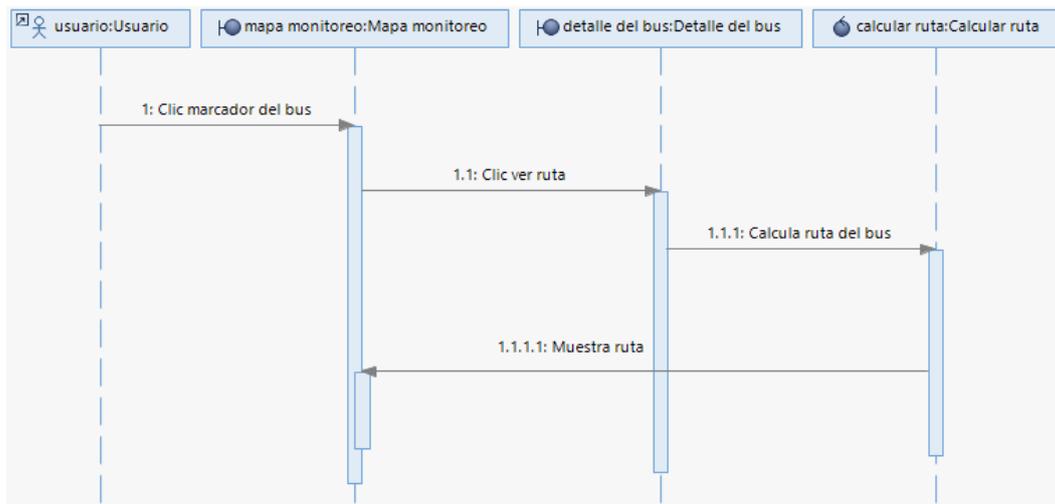


Fig. 30: Diagrama de secuencia ver ruta del bus

3.1.8.3.7 DIAGRAMA DE SECUENCIA SELECCIONAR DESTINO

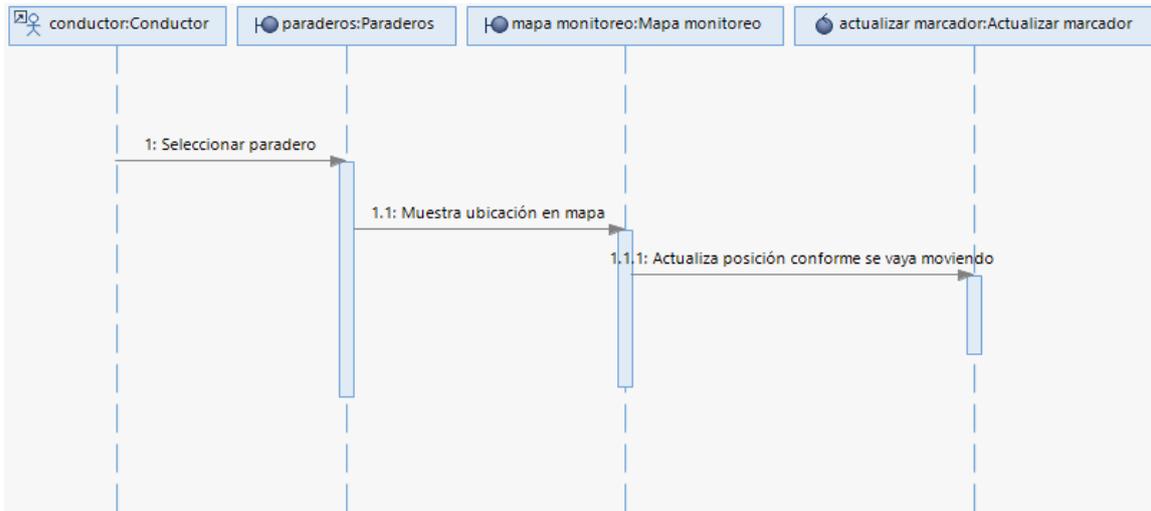


Fig. 31: Diagrama de secuencia seleccionar destino

3.1.8.3.8 DIAGRAMA DE SECUENCIA ENVÍO DE COORDENADAS AL SERVIDOR

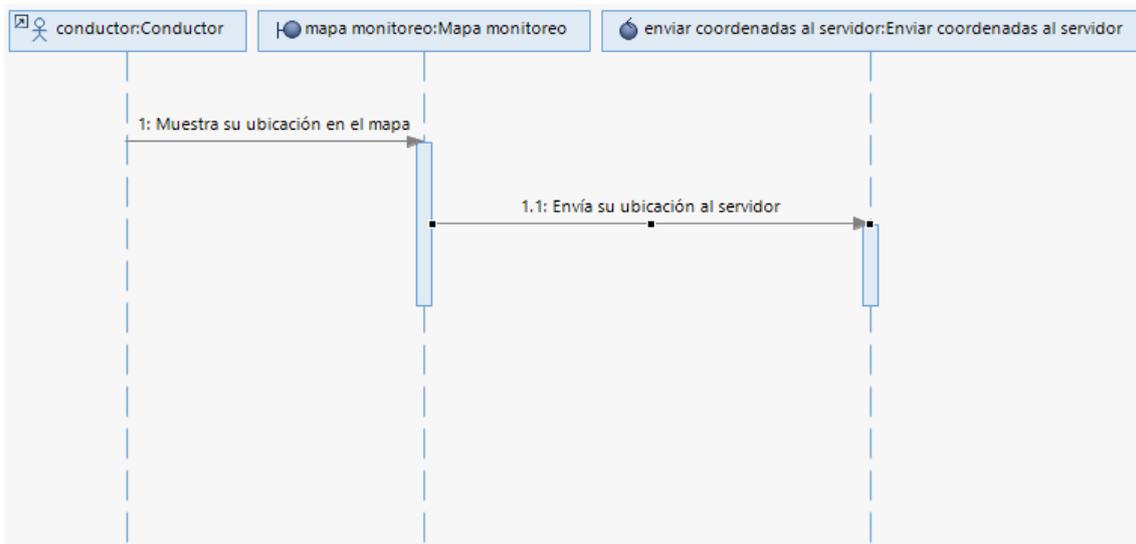


Fig. 32: Diagrama de secuencia envío de coordenadas al servidor

3.1.8.3.9 DIAGRAMA DE SECUENCIA RECIBIR COORDENADAS

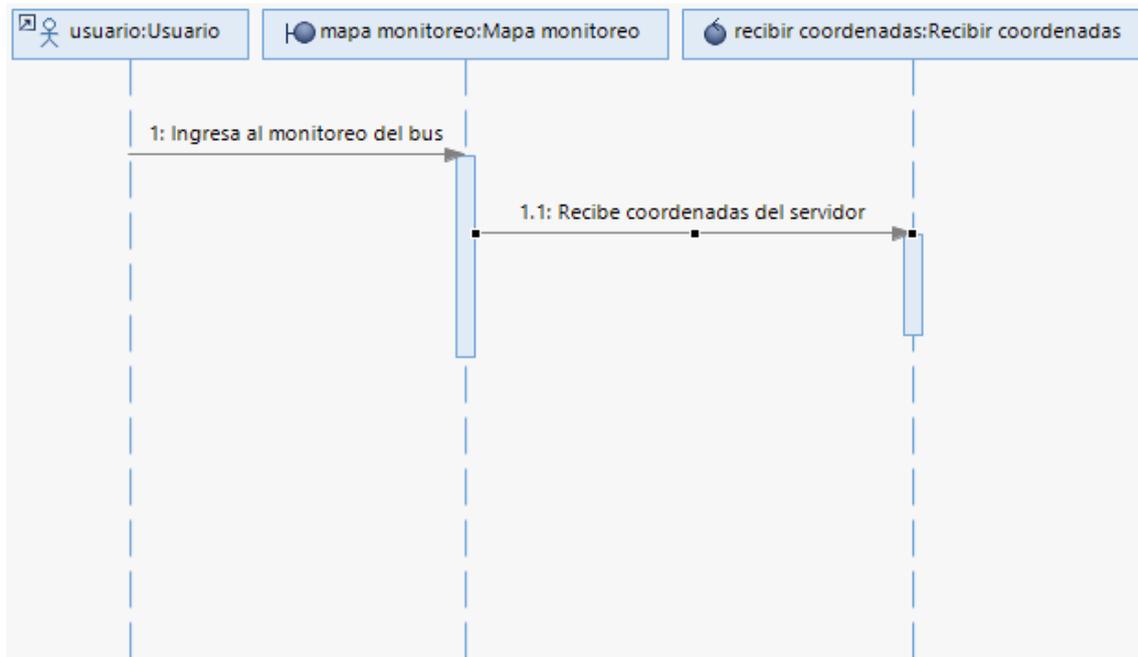


Fig. 33: Diagrama de secuencia recibir coordenadas

3.1.8.3.10 DIAGRAMA DE SECUENCIA CALCULO DE DATOS DE MONITOREO

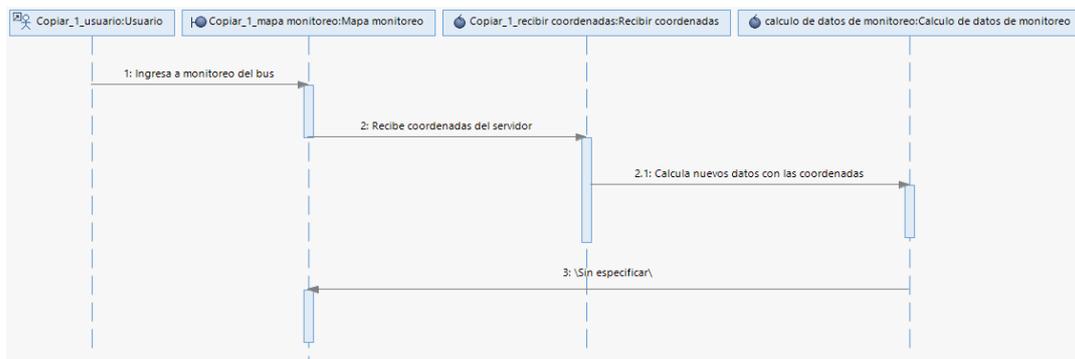


Fig. 34: Diagrama de secuencias cálculo de datos de monitoreo

3.1.9 ANÁLISIS DE LA SOLUCIÓN

En este apartado se evalúan las condiciones técnicas, operativas y económicas que pueden asegurar el cumplimiento de las metas y objetivos del presente proyecto, y así, determinar su viabilidad.

3.1.9.1 VIABILIDAD TÉCNICA

Se examina a continuación el ajuste entre las necesidades que se pueden identificar según la problemática y la solución a dichas necesidades a través de los indicadores de efectividad, confiabilidad y facilidad de uso.

➤ Efectividad del Sistema

Para materializar de alguna manera la efectividad del sistema, se lista en la siguiente tabla un conjunto de características que resuman la capacidad del mismo y que demuestren de qué manera se consiguen beneficios.

Tabla 34: Resumen de capacidades del sistema

Resumen de capacidades	
Beneficios	Características que lo soportan
Mayor facilidad para la gestión de la información	Envío y recepción de información en tiempo real hacia el servidor.
Misma información para todos los usuarios	Envío de información del servidor y recepción por parte de los usuarios en tiempo real.
Los usuarios pueden realizar seguimiento a los buses universitarios	El usuario desde la aplicación web puede ubicar el bus universitario en el mapa que se le muestra y puede ver su movimiento y ubicación actual en tiempo real.
Mayor demanda de los buses universitarios	Los usuarios a través de la aplicación web tendrán la información de llegada de los buses a sus paraderos en tiempo real con la cual podrán optar por hacer uso de los buses universitarios.
Seguridad de la información	Acceso a la información proporcionada por los usuarios únicamente por el administrador de la aplicación a través de un usuario y una contraseña

<p>Solución para diversos tipos de industria que requieran monitorear sus vehículos</p>	<p>Aplicación web que monitorea vehículos con dispositivos GPS de manera satisfactoria, uso de las funcionalidades del software desde distintos equipos conectados a la red, visualización de la información a través de un navegador web estándar, actualización del software en un solo equipo servidor que es replicado instantáneamente por todos los equipos clientes.</p>
---	---

➤ **Confiabilidad del sistema**

La aplicación web es confiable en cuanto a seguridad ya que la información de carácter confidencial se mantendrá en reserva o será accedida solo por las personas adecuadas. Se trabaja con validación de usuarios, por ende, solo aquellos con los permisos adecuados pueden actualizar o solo visualizar la parte administrativa de la aplicación web.

➤ **Facilidad de uso**

La facilidad de uso de la aplicación web se ve reflejada en que éste es muy intuitivo y se puede adaptar a la forma lógica de actuar de los usuarios. Estos usuarios no requieren una capacitación convencional, es decir, como la que se realizaría al instalar un sistema cualquiera, puesto que para el uso del mismo no se requieren conocimientos básicos de computación. Por otro lado, el entorno grafico simple, sencillo y amigable dará al usuario una sensación de seguridad en el manejo de la aplicación web.

3.1.9.2 VIABILIDAD ECONÓMICA

Para estimar un costo referencial del producto, se toman en cuenta factores laborales y no laborales.

Tabla 35: Costos de implementación

Rubro	Unidad	Cantidad	Costo U. S/.	Costo Total S/.	Observación
Gasto de Personal					
➤ Honorarios del Desarrollador	mes	4	900.00	3600.00	
Materiales y Suministros					
➤ Papel	paq	2	11.00	22.00	
➤ CD	und	5	1.00	5.00	
➤ Lapiceros	und	8	8.00	8.00	
➤ Folder Manila	paq	1	10.00	10.00	
Equipamiento					
➤ Computadora	mes	4	100.00	400.00	Disponible
➤ Tablet	und	1	400.00	400.00	
Licencias de Software					
➤ Hosting	Año	5	232.32	1161.6	
➤ Dominio .com	Año	5	47.20	236	
➤ Desarrollador android	Lic	1	81.87	81.87	
➤ Desarrollador Ios	Lic	1 x 5 años	324	1620	
Pasajes y alimentación					
➤ Pasajes	Ser	1	50.00	50.00	
➤ Refrigerios	Ser	1	50.00	50.00	
Servicios Varios					
➤ Fotocopias e Impresiones	Ser	1	30.00	30.00	
TOTAL				7674.47	

3.1.10 ANÁLISIS COSTO – BENEFICIO

Luego de haber presentado la viabilidad técnica y económica se realizará a continuación un análisis el cual deberá reflejar el porqué del desarrollo del proyecto, es decir, de qué manera contribuye y qué ofrece para que sea justificable su implementación.

La solución que se plantea en el proyecto tiene por finalidad, como ya se ha mencionado, contribuir con la mejora en el servicio el transporte interno de la Universidad Nacional de Cajamarca, acortando los tiempos de espera de los usuarios de los buses de la Universidad Nacional de Cajamarca, implementando un sistema de control y monitoreo en tiempo real.

Dado que la aplicación web se orienta al área de transporte de la Universidad Nacional de Cajamarca, los beneficiarios directos serán todos los usuarios de los buses universitarios, Los cuales tendrán un mejor servicio, así como también pueden reducir costos en pasajes al decidir usar los buses universitarios y no optar por otros medios de transporte pagados y además podrán acortar los tiempos de espera, tiempos que podrá utilizar en otras actividades.

Para el desarrollo del proyecto se emplearán distintos recursos (con su respectivo costo) los cuales se pueden agrupar en: tiempo, personas y tecnología. En función al tiempo, se puede visualizar en detalle las actividades en el cronograma del proyecto en el ANEXO 1. Entonces, el costo relacionado al tiempo se ve justificado: el monto resulta de un costo por hora bastante reducido y una cantidad de horas que son necesarias para llevar a cabo las etapas del proyecto. En relación a las personas, el presente proyecto es desarrollado de manera individual (por ello los costos unitarios del factor tiempo) lo cual también justifica el tiempo que tomará el desarrollo.

Con todo lo expuesto, se puede concluir que el análisis costo – beneficio tiene un saldo positivo hacia el beneficio que traería la solución, no solo por lo expuesto líneas arriba sino porque esta aplicación web constituye una solución única en el área de transporte público y privado en la ciudad de Cajamarca, sería algo innovador, ya que existen soluciones del tipo cliente – servidor pero no a la medida, además la solución del tipo web beneficiará tecnológicamente a la Universidad Nacional de Cajamarca. Queda así entonces, justificada la implementación de este sistema web.

3.1.11 DISEÑO DEL SISTEMA

3.1.11.1 ARQUITECTURA DE LA SOLUCIÓN

En esta sección se detalla la arquitectura que se emplea en la aplicación web para lo cual primero se indica el tipo de arquitectura elegida. Luego, se presenta el diseño de la arquitectura de alto nivel que se utiliza en la solución. Esto implica dividir la aplicación en componentes funcionales posicionados en capas, las cuales también son detalladas.

3.1.11.2 REPRESENTACIÓN DE LA ARQUITECTURA

La arquitectura a utilizar será web. Se distinguen dos secciones, el cliente, donde se encuentra el usuario de la aplicación web, que accederá a esta por medio de un navegador (Internet Explorer o Mozilla Firefox, Google Chrome, otros), un celular android o iOS y la segunda sección la conforma el servidor, en donde residen los datos, las reglas y lógica

de la aplicación web, se deberá tener como sistema operativo base Ubuntu Server 16.04 LTS, como gestor de base de datos PostgreSQL 9.5 y como servidor web Node JS bajo el framework denominado SAILS JS, las interfaces graficas están bajo HTML5 y el framework Materialize.

Uno de los motivos por los que se realiza una aplicación web es porque se sabe que este tipo de aplicaciones emplean "light clients", que son clientes que no ejecutan demasiadas labores de procesamiento para la ejecución de la misma aplicación, lo cual es un punto esencial ya que lo que menos se desea es que en la sección cliente se realicen demasiadas tareas, solo las necesarias para que el usuario final pueda acceder a la aplicación y realizar el trabajo deseado. El auge de las redes locales y la popularidad de internet han posibilitado el acceso a través de computadores y otros dispositivos móviles, ha aumentado y extendido el empleo de las aplicaciones web las cuales pueden ser utilizadas por usuarios ubicados en cualquier lugar del planeta con acceso a internet.

Para este proyecto en particular, es conveniente que la solución sea una aplicación web fundamentalmente por el tema de acceso a la aplicación en cualquier momento (las 24 horas del día, los 7 días de la semana) lo cual constituye una de sus principales atracciones por los beneficios que esto brinda, como es el acceso por parte de los usuarios a la información (tiempo y distancia de llegada) de los buses universitarios desde cualquier lugar con conexión a internet sin importar la instalación de algún software, así como otras ventajas detalladas en los capítulos anteriores y que colaboran a la solución de la problemática planteada en el capítulo 1, sobre todo en el efecto del transporte público.

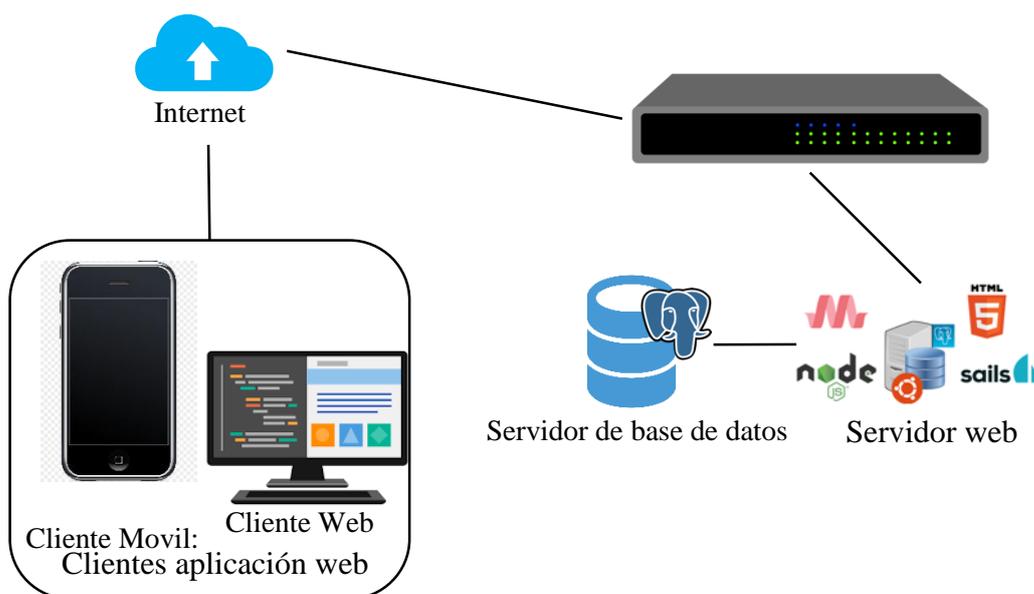


Fig. 35: Diagrama de arquitectura

3.1.11.3 ARQUITECTURA DE CAPAS O NIVELES

También conocida como arquitectura de tres capas, la arquitectura de tres capas, define cómo organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores.

Capa de presentación: Es la que se encarga de que el sistema interactúe con el usuario y viceversa, muestra el sistema al usuario, le presenta la información y obtiene la información del usuario en un mínimo de proceso. En el mundo de la informática es conocida como interfaz gráfica y debe tener la característica de ser amigable, o sea, entendible y fácil de usar para el usuario. Esta capa se comunica únicamente con la capa intermedia o de negocio.

Capa de negocio o lógica: Es donde residen las funciones que se ejecutan, se reciben las peticiones del usuario, se procesa la información y se envían las respuestas tras el proceso. Se denomina capa de negocio o capa de lógica del negocio, porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de acceso a datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

Capa de acceso a datos: Esta capa es la encargada de almacenar los datos del sistema y de los usuarios. Su función es almacenar y devolver datos a la capa de negocio, aunque para esto también es necesario en algunos casos, que tengan procedimientos almacenados y funciones dentro de la capa. En una arquitectura de tres capas, esta capa es la única que puede acceder a los mismos. Está formada por uno o varios sistemas gestores de bases de datos, localizados en un mismo servidor o en varios.

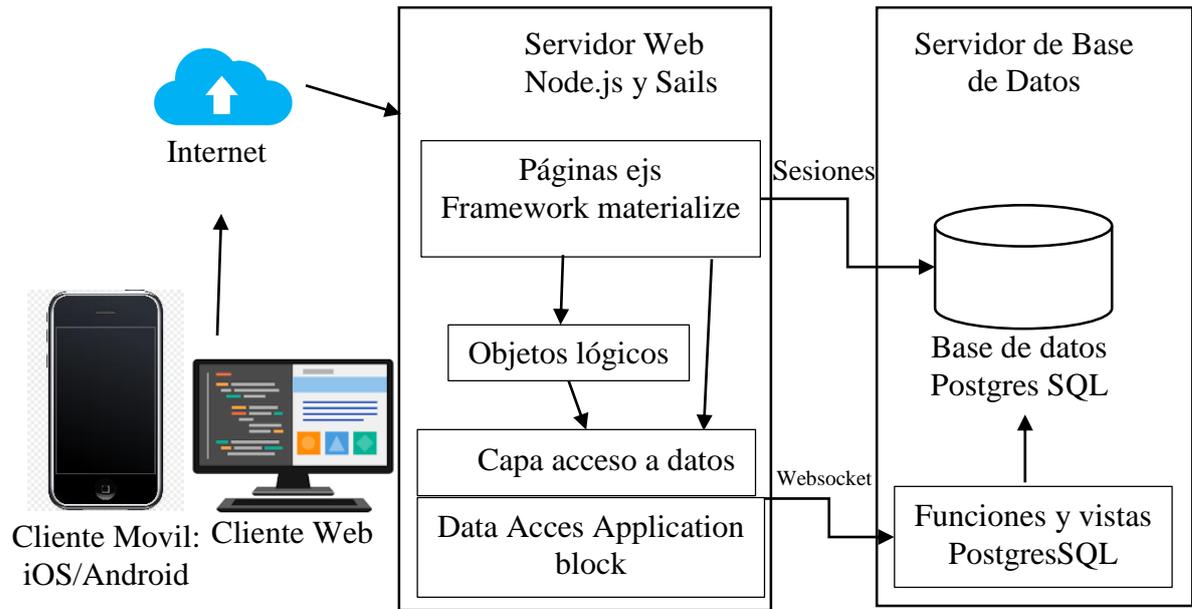


Fig. 36: Arquitectura de capas o niveles

3.1.11.4 DIAGRAMA DE COMPONENTES

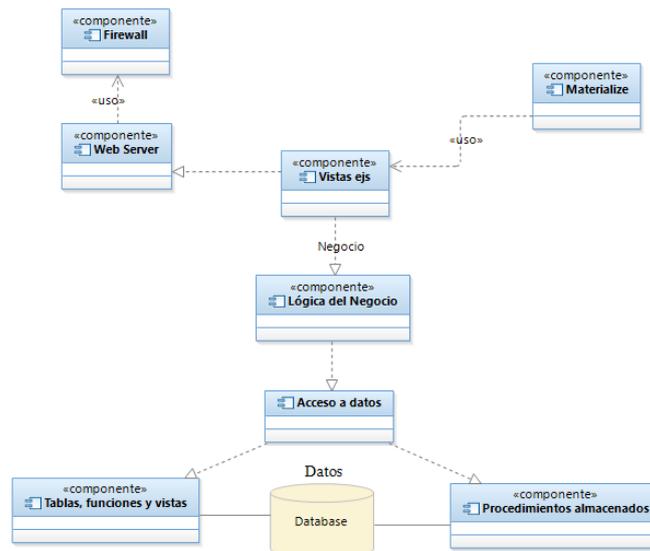


Fig. 37: Diagrama de componentes

3.1.11.5 DISEÑO DE BASE DE DATOS

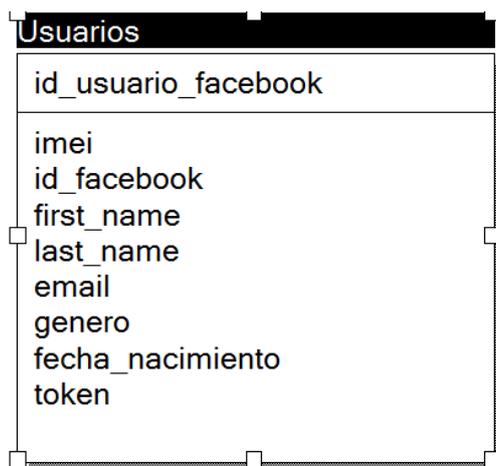


Fig. 38: Diagrama de base de datos entidad relación

3.1.11.5.1 BASE DE DATOS RELACIONAL

Después de haber analizado los requerimientos y definir los casos de uso se definió la base de datos, se eligió un modelo relacional ya que en este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que éstos se almacenen no tiene relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar por un usuario no experto. La información puede ser recuperada o almacenada por medio de consultas que ofrecen una amplia flexibilidad y poder para administrar la información.

Este modelo considera la base de datos como una colección de relaciones. De manera simple, una relación representa una tabla que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real. Cada fila también se puede denominar tupla o registro y a cada columna también se le puede llamar campo o atributo.

En la fig. 26 se muestra la base de datos, en este caso solo se manejará una única tabla de usuarios en donde se registrará la información de los usuarios que deseen ingresar a la aplicación mediante Facebook, previa aceptación por parte del usuario.

3.1.12 PROTOTIPOS

El sistema tendrá el siguiente icono



Fig. 39: Logo de la aplicación web

3.1.12.1 PANTALLAS DEL SISTEMA

Se muestra el prototipo del sistema organizado por casos de uso. Para cada caso de uso se incluyen las pantallas que permiten realizar tanto el camino básico como los caminos alternativos, pero las anotaciones incluidas en ésta se refieren a la ejecución del camino básico. Para una mejor comprensión de la secuencia de las pantallas, se recomienda revisarlas en forma paralela a la especificación del caso de uso.

3.1.12.1.1 PANTALLA DE BIENVENIDA A LA APLICACIÓN WEB

Es la página de bienvenida que se le mostrará al usuario por tres segundos antes de ingresar a la siguiente pantalla automáticamente, el usuario podrá ingresar a esta pantalla desde su celular Android, iOS o un navegador web (Internet Explorer o Mozilla Firefox, Google Chrome, entre otros) en la siguiente dirección web www.elcarrodelau.com.



Fig. 40: Pantalla bienvenida vista móvil

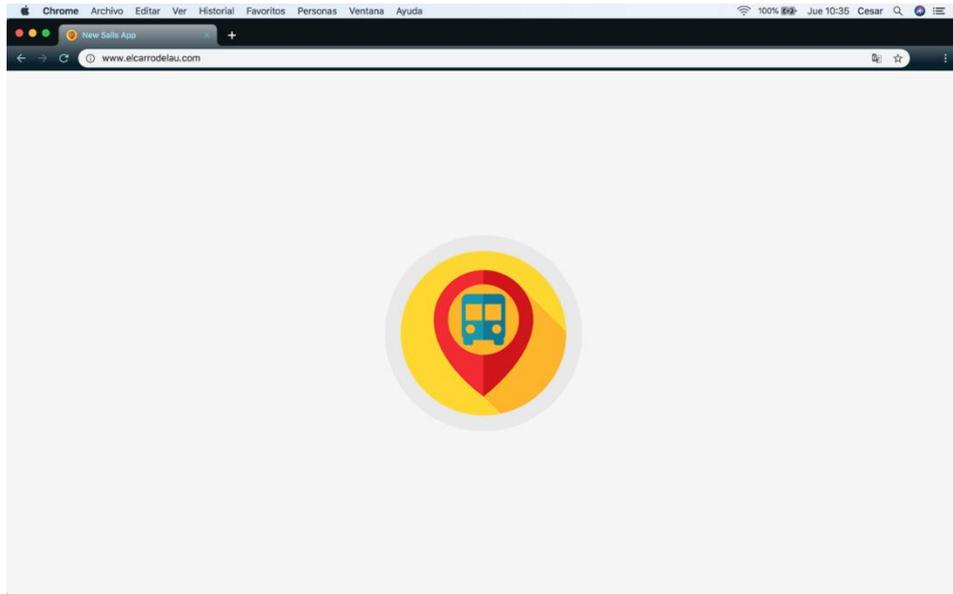


Fig. 41: Pantalla bienvenida vista web

3.1.12.1.2 PANTALLA PRINCIPAL DE LA APLICACIÓN

Luego de mostrar la pantalla de bienvenida al usuario se le mostrará la pantalla principal donde podrá seleccionar la opción de ingresar directamente a la siguiente pantalla o poder continuar logueándose con Facebook.



Fig. 42: Pantalla inicio vista móvil

Tabla 36: Descripción de elementos pantalla de inicio vista móvil

	Botón para que los usuarios inviten a sus contactos de facebook a utilizar la aplicación.
	Botón para compartir la aplicación en las diferentes redes sociales utilizadas por el usuario.
	Botón para ingresar a la pantalla de selección de paradero sin logueo con facebook.
	Botón para loguearse con facebook, almacenar los datos del usuario en base de datos y pasar a la pantalla de selección de paradero.

3.1.12.1.3 PANTALLA SELECCIONAR PARADERO

En esta pantalla se le mostrará al usuario la lista de todos los paraderos disponibles, de los cuales podrá elegir al que desea dirigirse y poder realizar el monitoreo correspondiente del bus universitario.

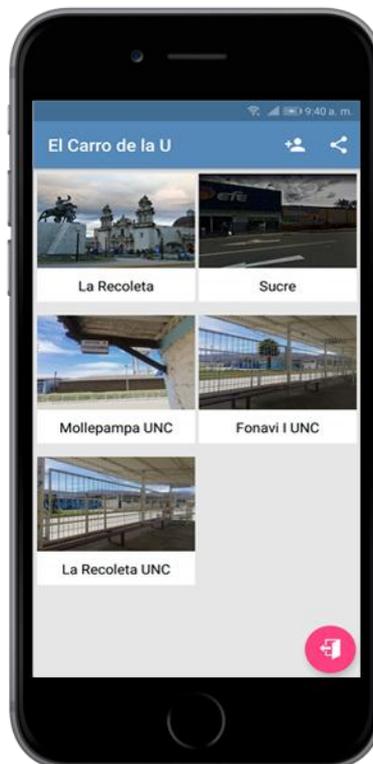


Fig. 43: Pantalla seleccionar paradero vista móvil

Tabla 37: Descripción de elementos pantalla seleccionar paradero vista móvil

	Botón para que los usuarios inviten a sus contactos de facebook a utilizar la aplicación.
	Botón para compartir la aplicación en las diferentes redes sociales utilizadas por el usuario.
	Botón para cerrar sesión de facebook y regresar a la pantalla de inicio.

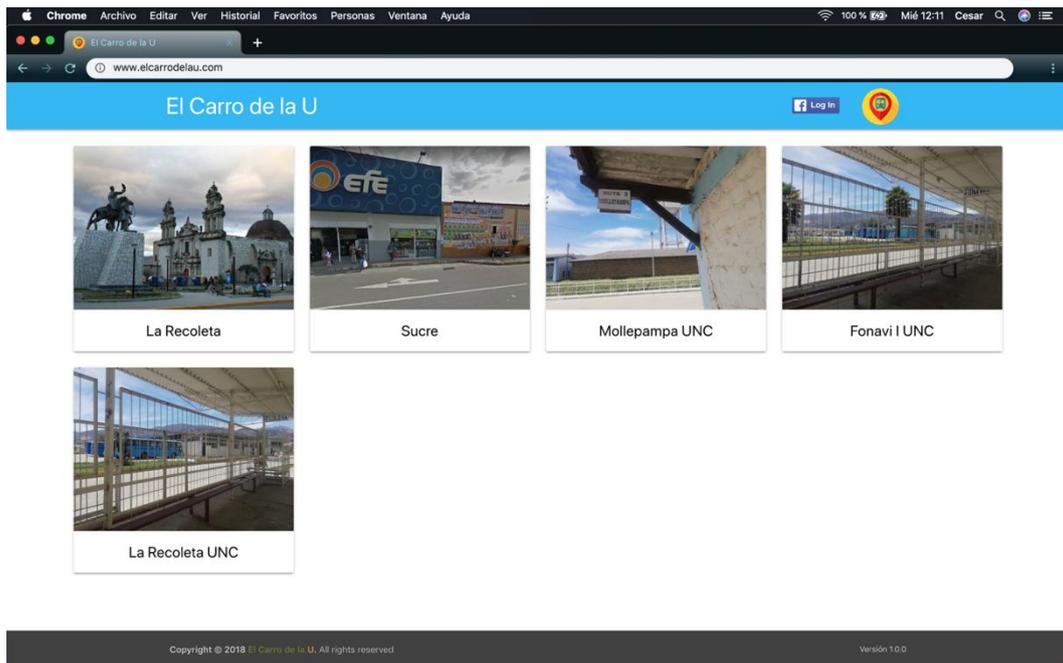


Fig. 44: Pantalla seleccionar paradero vista web

Tabla 38: Descripción de elementos pantalla seleccionar paradero vista web

	Botón para loguearse con facebook almacenar los datos del usuario en base de datos.
---	---

3.1.12.1.4 PANTALLA MAPA

En esta pantalla se mostrará el mapa de google maps, dentro del cual se mostrará un marcador con su paradero seleccionado, otro marcador indicando la ubicación del bus universitario que irá actualizando la posición en el mapa en tiempo real, en caso el usuario haya ingresado con logueo por facebook se le mostrará un marcador con la imagen obtenida de su facebook, de lo contrario únicamente se le mostrará su ubicación en el mapa con un punto azul propio de google maps.

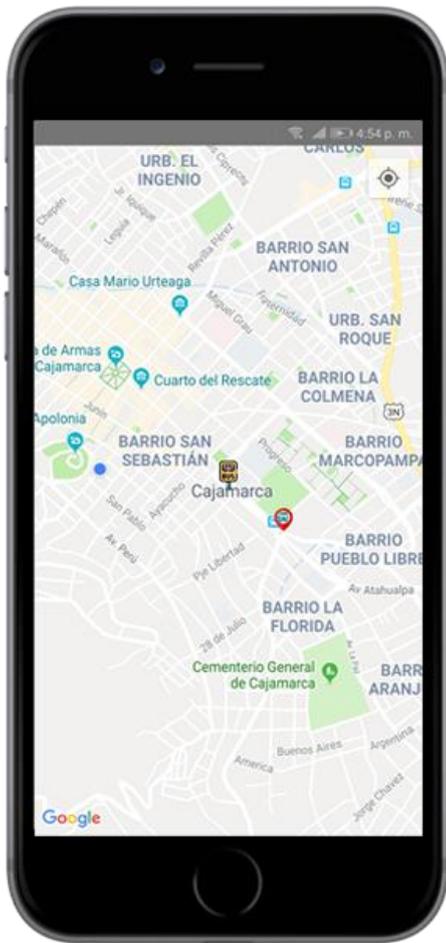


Fig. 45: Pantalla mapa sin logueo de facebook vista móvil

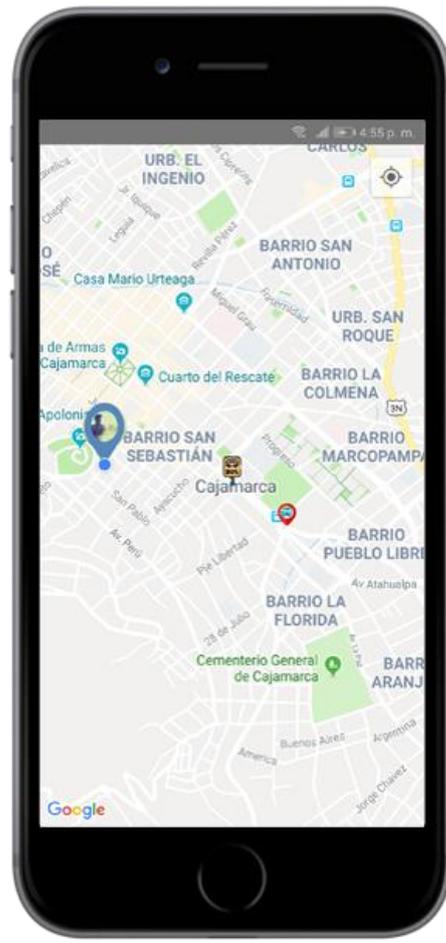


Fig. 46: Pantalla mapa con logueo de facebook vista móvil

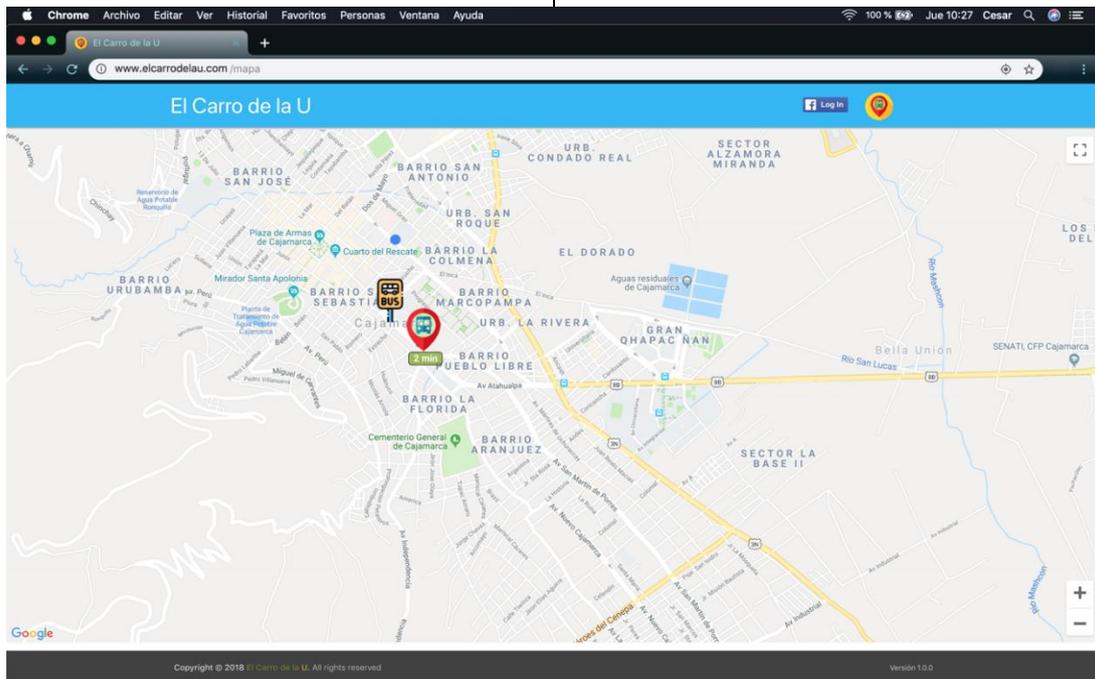


Fig. 47: Pantalla mapa sin logueo de facebook vista web

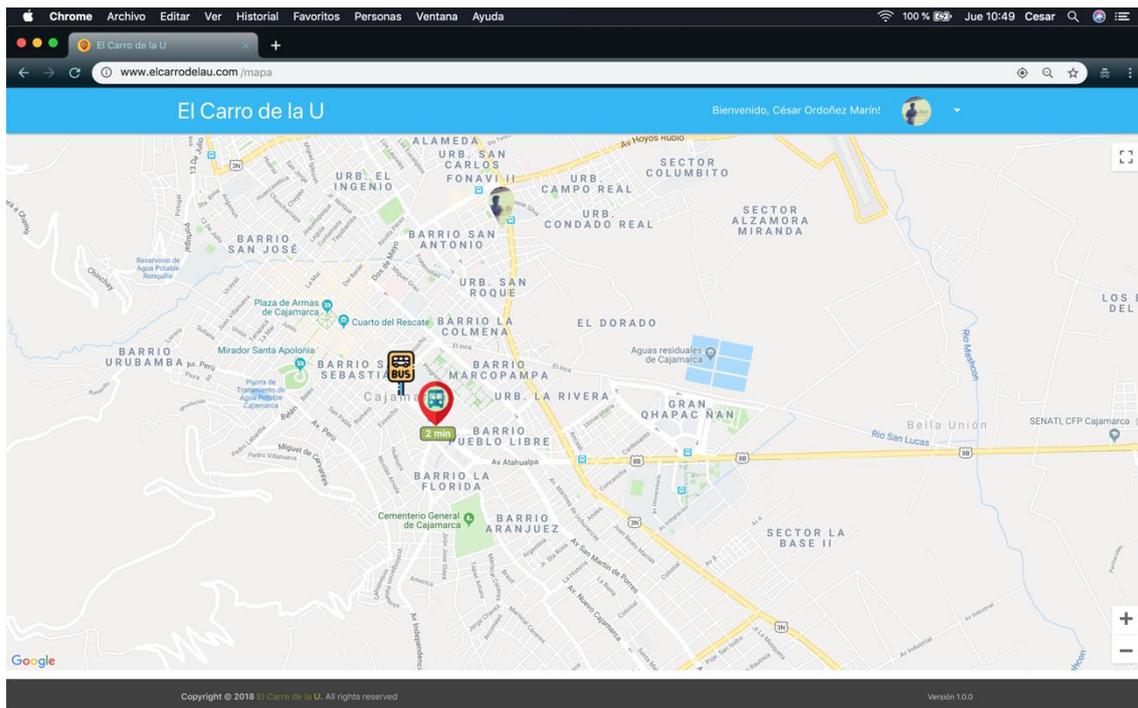


Fig. 48: Pantalla mapa con logueo de facebook vista web

Tabla 39: Descripción de elementos pantalla mapa vista web y móvil

	Ubicación actual del usuario mostrada sin logueo con facebook.
	Ubicación actual del usuario mostrada con logueo con facebook.
	Marcador que indica la ubicación del paradero seleccionado por el usuario.
	Marcador que indica la ubicación actual del bus universitario.

3.1.12.1.5 PANTALLA VENTANA EMERGENTE OPCIONES DE PARADERO

El usuario podrá presionar sobre el marcador del paradero y le aparecerá una ventana emergente donde se le mostrará la imagen de su paradero, el nombre del paradero y se le dará información de distancia desde su posición actual hasta el paradero seleccionado, tiempo que demoraría en llegar caminando o en auto, también dos botones con las opciones de cambiar de paradero y la opción de cómo llegar al paradero.

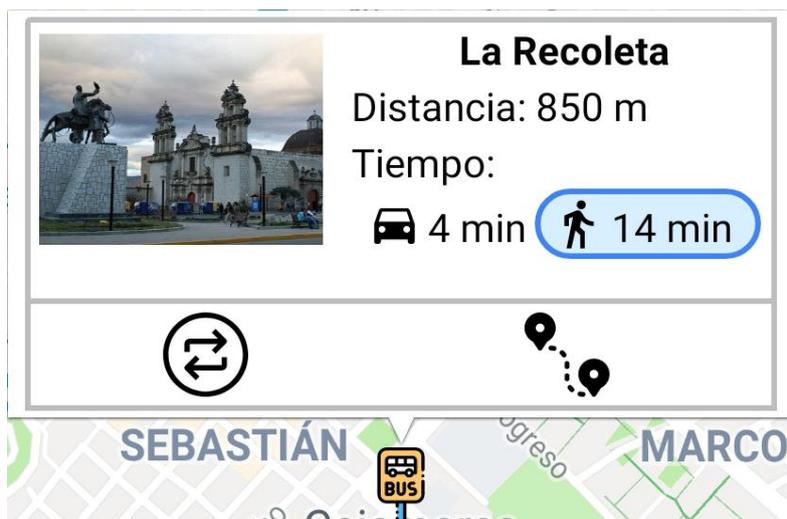


Fig. 49: Figura pantalla ventana emergente opciones de paradero

Tabla 40: Descripción de elementos pantalla ventana emergente opciones de paradero

	<p>Opción de cómo llegar en auto, al presionar sobre este botón se le mostrará la ruta que debería seguir el usuario para llegar desde su posición actual hasta su paradero en auto.</p>
	<p>Opción de cómo llegar caminando, al presionar sobre este botón se le mostrará la ruta que debería seguir el usuario para llegar desde su posición actual hasta su paradero caminando.</p>
	<p>Opción cambiar de paradero, al presionar sobre este botón se le mostrará una ventana emergente con la lista de paraderos disponibles registrados en el sistema para que el usuario pueda elegir un nuevo paradero.</p>
	<p>Opción como llegar al paradero, al presionar sobre este botón se le mostrará la ruta que debe seguir el usuario desde su posición actual hasta el paradero seleccionado, por defecto se le mostrará la ruta que debe seguir caminando, pero la puedo cambiar presionando sobre la opción de cómo llegar en auto o viceversa.</p>

3.1.12.1.6 PANTALLA VENTANA EMERGENTE CAMBIAR DE PARADERO

Luego que el usuario selecciona la opción cambiar de paradero se le mostrará una ventana emergente con la lista de todos los paraderos disponibles registrados en la aplicación web, el usuario puede presionar fuera del modal, la tecla escape (modo web) o botón retroceder

(modo móvil) para que se cierre en caso no quiera cambiar de paradero o puede seleccionar un nuevo paradero para que sea actualizado en el mapa de google maps.

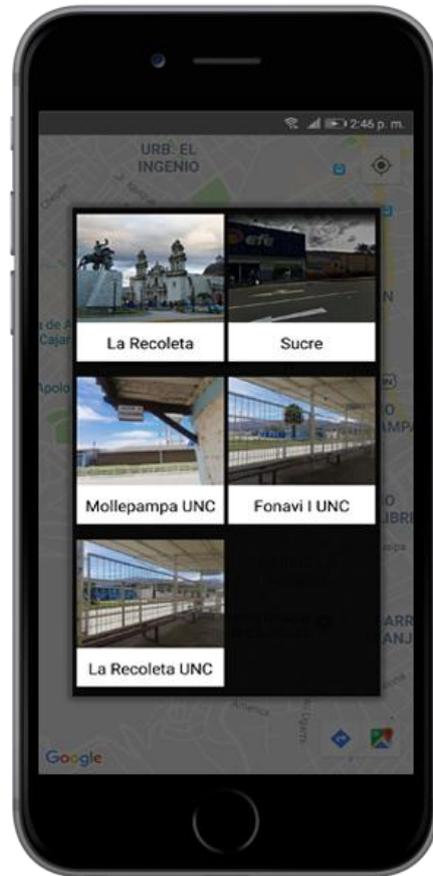


Fig. 50: Pantalla cambiar de paradero vista móvil

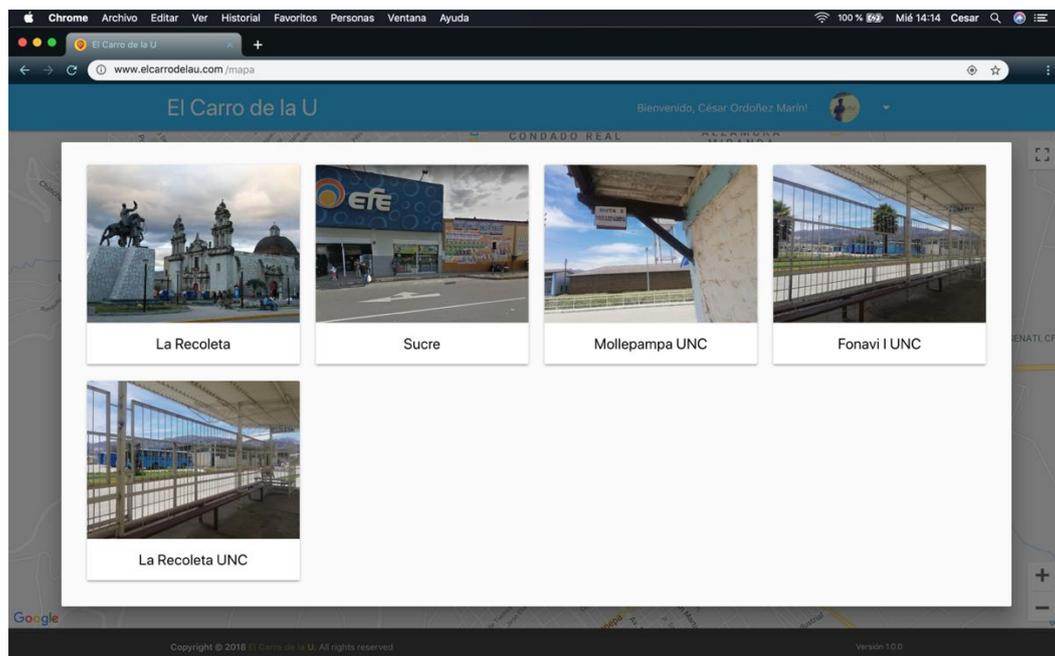


Fig. 51: Pantalla cambiar de paradero vista web

3.1.12.1.7 PANTALLA CÓMO LLEGAR AL PARADERO

El usuario luego de seleccionar la opción de cómo llegar al paradero se le mostrará la ruta que debería seguir en el mapa desde su ubicación actual hasta la ubicación de su paradero seleccionado, el usuario puede modificar la ruta pudiendo seleccionar las opciones de dirigirse en auto o caminando, la ruta será actualizada en el mapa luego de seleccionar una de estas opciones.

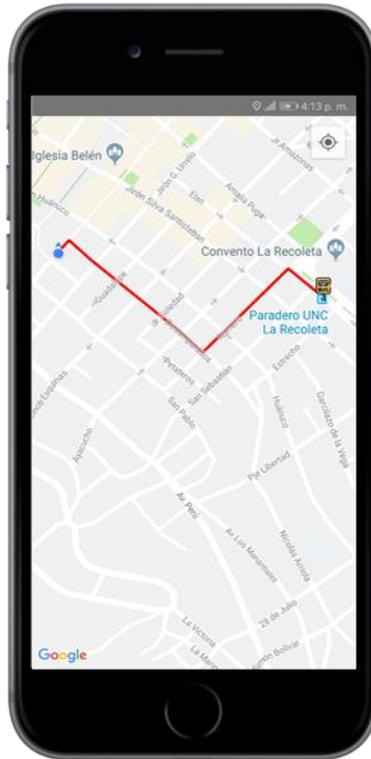


Fig. 52: Pantalla cómo llegar al paradero vista móvil

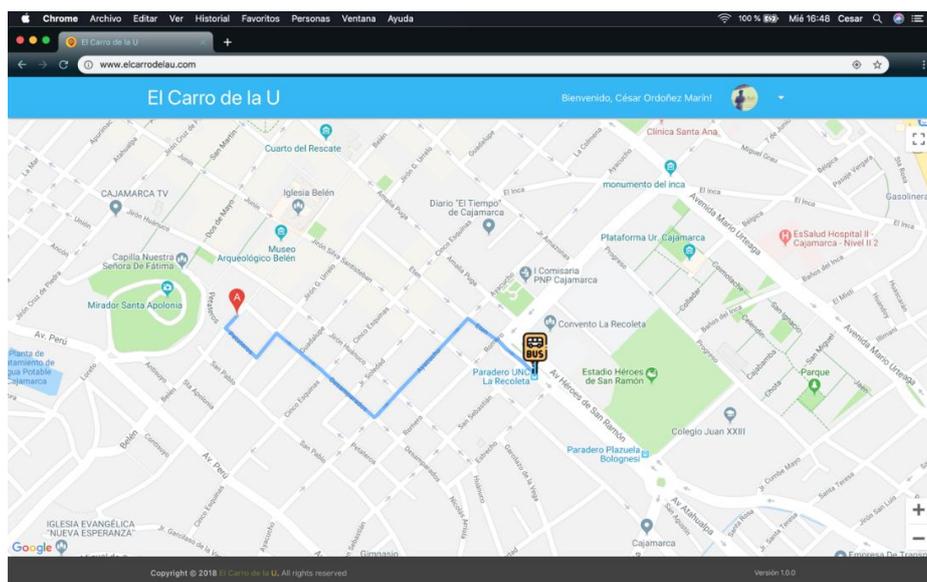


Fig. 53: Pantalla como llegar al paradero vista web

3.1.12.1.8 PANTALLA VENTANA EMERGENTE MONITOREO DEL BUS

El usuario al presionar sobre el marcador del bus se le mostrará una ventana emergente donde tendrá la información del bus en tiempo real que mostrará la distancia y el tiempo que demorará el bus en llegar a su paradero final, también un botón con el cual se podrá ver la ruta que seguirá el bus universitario hasta su paradero final, toda esta información y la posición del marcador será actualizada en tiempo real mediante websocket.

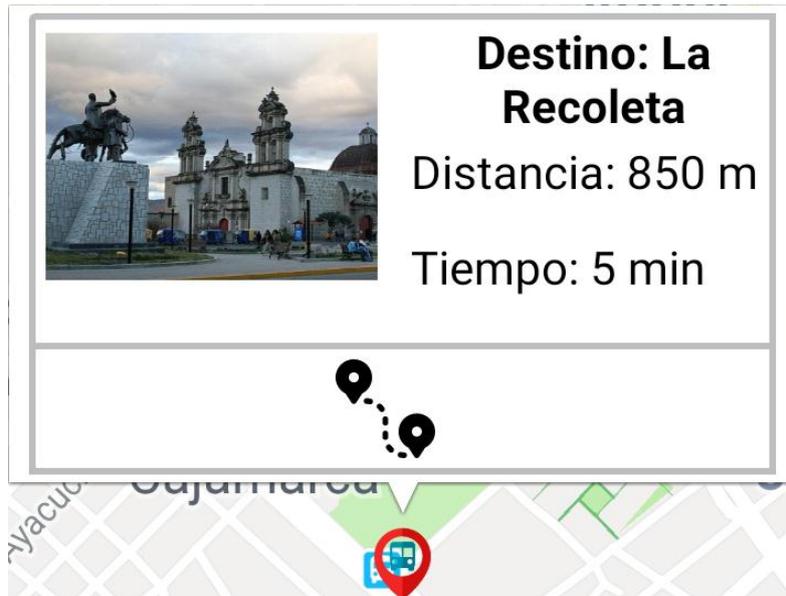


Fig. 54: Pantalla ventana emergente monitoreo del bus

Tabla 41: Descripción de elementos pantalla ver ruta del bus

	Opción ver ruta del bus, al presionar sobre este botón se le mostrará la ruta que sigue el bus hasta llegar a su paradero final, esta ruta se irá actualizando en tiempo real con las coordenadas recibidas mediante websocket.
---	---

3.1.12.1.9 PANTALLA VER RUTA DEL BUS

El usuario luego de seleccionar la opción de ver ruta del bus, se le mostrará la ruta que está siguiendo el bus en el mapa hacia su paradero seleccionado, la ruta se irá actualizando en tiempo real con la llegada de las nuevas coordenadas por websocket.

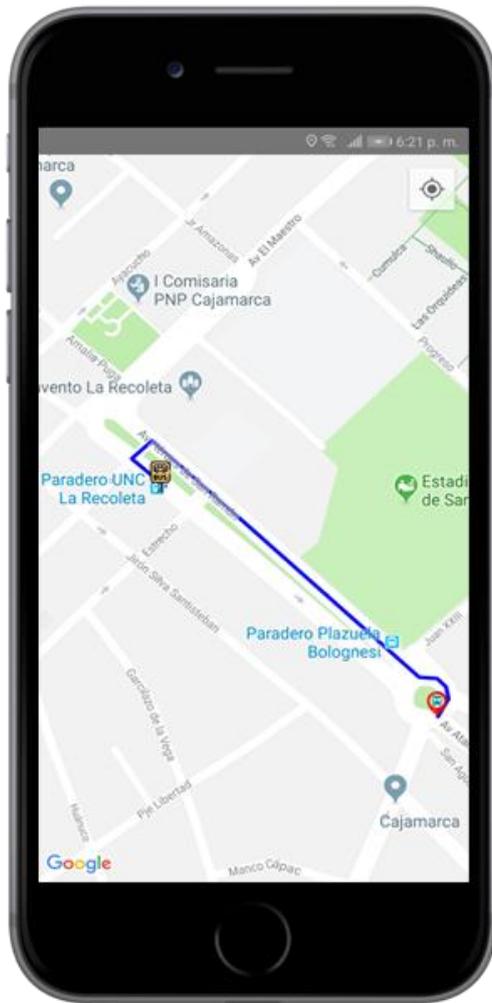


Fig. 55: Pantalla ver ruta del bus vista móvil

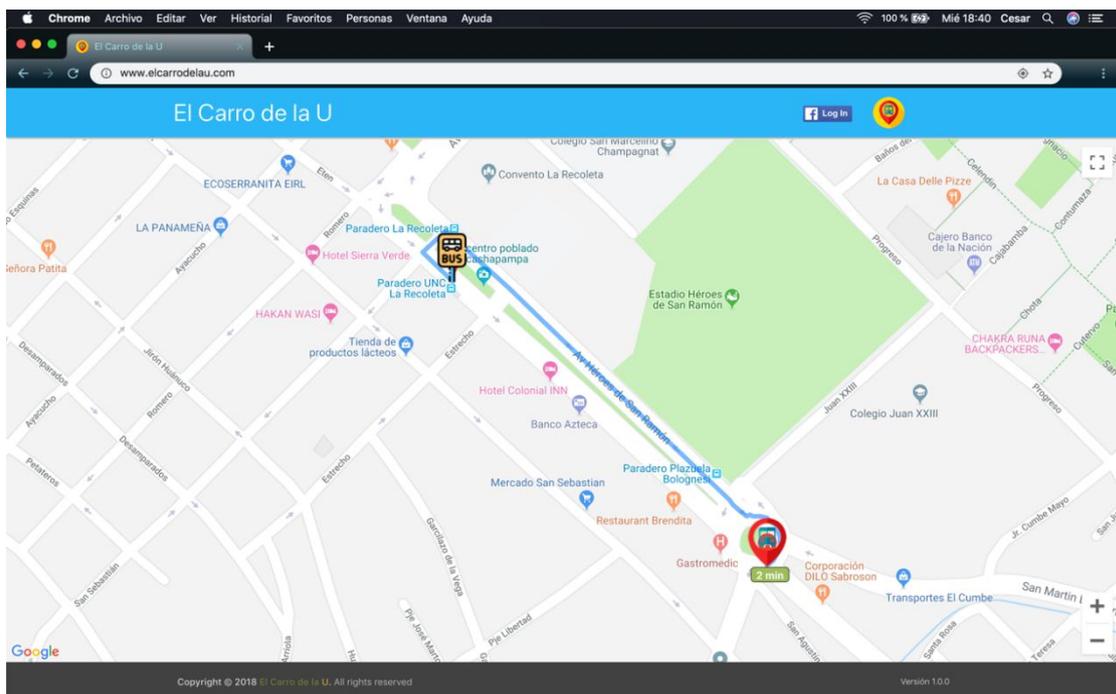


Fig. 56: Pantalla ver ruta del bus vista web

3.1.12.1.10 PANTALLA SELECCIONAR DESTINO

Los conductores de los buses universitarios deberán seleccionar el paradero de destino al que se dirigen, esta opción de selección tendrá únicamente una vista móvil que será desde los celulares implementados en los buses universitarios, una vez seleccionado el paradero el celular comenzará a enviar las coordenadas gps hacia los clientes conectados a la aplicación web mediante la tecnología websocket.

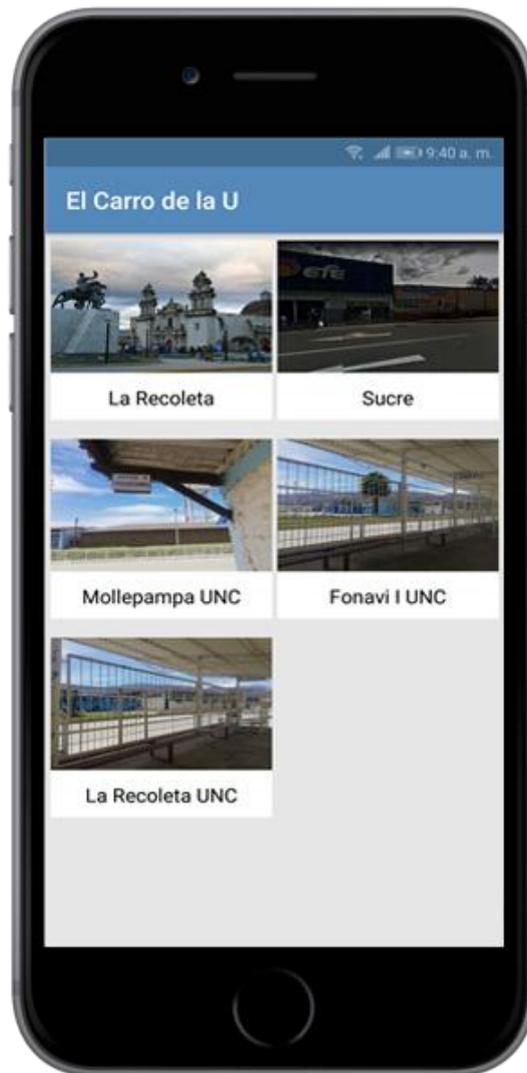


Fig. 57: Pantalla seleccionar destino

3.1.13 DESARROLLO DEL SISTEMA

3.1.13.1 CAPA ACCESO A DATOS

➤ CONEXIÓN CON BASE DE DATOS

➤ CONNECTIONS.JS

A través de este código abriremos una conexión en la base de datos postgresql.

```
PostgresqlServer: {  
  adapter: 'sails-postgresql',  
  host: 'localhost',  
  user: 'elcarrodelau',  
  password: 'P@$w0rd',  
  database: 'ElCarroDeLaUdb'  
}
```

➤ ROUTES.JS

A través de este archivo se configurarán las rutas que llamen a los controladores y estos a su vez llamen a sus métodos.

```
module.exports.routes = {  
  
  '/': {  
    view: 'homepage'  
  },  
  
  '/mapa': 'monitoreo.interface_monitoreo'  
  
};
```

3.1.13.2 CAPA DE NEGOCIO O LÓGICA

➤ CONEXIÓN SOCKET - SERVIDOR

SocketmonitoreoController.js es el archivo ubicado en el servidor que registrará a todos los clientes conectados al sistema web. El archivo es un documento JSON que registra un identificador de cada cliente conectado (conductores y usuarios de los buses), este archivo recibe las coordenadas enviadas por los conductores y las envía a los usuarios conectados.

```
module.exports = {  
  add_socket_id: function (req, res) {  
    if (req.isSocket && req.method === 'POST') {  
      Socketmonitoreo.create()  
        .exec(function (error) {  
          Socketmonitoreo.publishCreate({
```

```

    id: req.socket.id,
    coordenadas: req.param('coordenadas'),
    idDispositivo: req.param('idDispositivo'),
    idParadero: req.param('idParadero')
  });
});

} else if (req.isSocket) {
  // subscribe client to model changes
  Socketmonitoreo.watch(req.socket);
  // console.log('User subscribed to ' + req.socket.id);
  return res.ok('My socket ID is: ' + req.socket.id);
}
}
};

```

➤ CONDUCTOR

El conductor llamara al archivo SocketmonitoreoController.js en el servidor y dentro de este al método add_socket_id para registrar su identificador en el servidor y activar el socket.

```

//SOCKETS
try {
  mSocket = IO.socket("http://ec2-3-17-208-78.us-east-
2.compute.amazonaws.com:1337/?_sails_io_sdk_version=0.13.5");
} catch (URISyntaxException e) {
  throw new RuntimeException(e);
}

JSONObject jsonObject = new JSONObject();

try {
  jsonObject.put("url", "/socketmonitoreo/add_socket_id");
} catch (JSONException e) {
  e.printStackTrace();
}
mSocket.emit("get", jsonObject, new Ack() {
  @Override
  public void call(Object... args) {
    Log.d("MiActividad", "records: " + args[0].toString());
  }
});
mSocket.connect();
mSocket.emit("type", "activity");

```

Luego de registrado el identificador en el servidor los conductores empiezan a enviar sus coordenadas, el identificador de su dispositivo y el identificador de su paradero de destino mediante webSocket.

```

//ENVIO DE COORDENADAS POR SOCKET
JSONObject jsonObject = new JSONObject();

try {
  jsonObject.put("url", "/socketmonitoreo/add_socket_id?coordenadas=" +
String.valueOf(location.getLatitude()) + "," + String.valueOf(location.getLongitude()) + "&idDispositivo=" +
idDispositivo + "&idParadero=" + idParadero);
  jsonObject.put("coordenadas", String.valueOf(location.getLatitude()) + "," +
String.valueOf(location.getLongitude()));
}

```

```

jsonObject.put("idDispositivo", idDispositivo);
jsonObject.put("idParadero", idParadero);

SharedPreferences sharedPreferencesLtlLn = this.getSharedPreferences("com.monitoreo.monitoreo.prefs",
Context.MODE_PRIVATE);
SharedPreferences.Editor editor = sharedPreferencesLtlLn.edit();
editor.putFloat("latitud", (float) location.getLatitude());
editor.putFloat("longitud", (float) location.getLongitude());
editor.apply();

} catch (JSONException e) {
e.printStackTrace();
}
mSocket.emit("post", jsonObject, new Ack() {
@Override
public void call(Object... args) {
Log.d("MiActividad", "records: " + args[0].toString());
}
});

origen = String.valueOf(location.getLatitude()) + "," + String.valueOf(location.getLongitude());
latLng = new LatLng(location.getLatitude(), location.getLongitude());

```

➤ USUARIOS

El usuario llamará al archivo SocketmonitoreoController.js en el servidor y dentro de este al metodo add_socket_id para registrar su identificador en el servidor y activa un socket que recibirá las nuevas coordenadas enviadas por los conductores.

```

//INICIALIZAR SOCKET
try {
mSocket = IO.socket("http://ec2-3-17-208-78.us-east-
2.compute.amazonaws.com:1337/?_sails_io_sdk_version=0.13.5");
} catch (URISyntaxException e) {
throw new RuntimeException(e);
}

//REGISTRAR SOCKET
JSONObject jsonObject = new JSONObject();
try {
jsonObject.put("url", "/socketmonitoreo/add_socket_id");
} catch (JSONException e) {
e.printStackTrace();
}
mSocket.emit("get", jsonObject, new Ack() {
@Override
public void call(Object... args) {
Log.d("MiActividad", "records: " + args[0].toString());
}
});
mSocket.connect();
mSocket.emit("type", "activity");
//SOCKET RECEPCION DE COORDENADAS
mSocket.on("socketmonitoreo", socket_notificacion);

private Emitter.Listener socket_notificacion = new Emitter.Listener() {
@Override
public void call(Object... args) {
final JSONObject data = (JSONObject) args[0];
final String[] coord = {"", "", ""};

MapsActivity.this.runOnUiThread(new Runnable() {

```

```

@Override
public void run() {
    try {
        if (data.getString("verb").equals("created")) {
            JSONObject data_b = new JSONObject(data.getString("data"));
            coord[0] = data_b.getString("coordenadas");
            coord[1] = data_b.getString("idParadero");
            coord[2] = data_b.getString("idDispositivo");
            String[] coord_n = coord[0].split(",");

            if (idParadero.equals(coord[1])) {
                latitudBus = Double.parseDouble(coord_n[0]);
                longitudBus = Double.parseDouble(coord_n[1]);
                googleApi.tiempo(coord_n[0], coord_n[1], latitudParadero, longitudParadero,
getApplicationContext(), "driving", "tiempoLlegada", waypointsRutaBusSend);

                if (markerInfoIsShowBus) {
                    for (final Marker marker : markersBuses) {
                        if (marker.getTitle().equals(coord[2])) {
                            marker.showInfoWindow();
                        }
                    }
                }
                manejador_marcadores_buses(coord_n[0], coord_n[1], coord[2]);
            }
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
}
};

```

3.1.13.3 CAPA DE PRESENTACIÓN

➤ MÓDULOS DE LA APLICACIÓN WEB

➤ SELECCIÓN DE PARADERO

a. Vista Web

```

<link href='https://fonts.googleapis.com/css?family=Open+Sans:400italic,600' rel='stylesheet'
type='text/css'>

```

```

<div id="preloader">
<div class="pre-container">
<div class="logo-circle">

</div>
</div>
</div>

```

```

<ul id="dropdown_sesion" class="dropdown-content">
<li class="divider"></li>
<li>
<a href="#" onclick="logoutFacebook()">Cerrar Sesión</a>
</li>
</ul>

```

```

<nav class="light-blue lighten-1" role="navigation">
<div class="nav-wrapper container">

```

```

<a id="logo-container" href="#" class="brand-logo">El Carro de la U</a>

```

```

<ul class="right hide-on-med-and-down side-nav">
  <li id="li-button-facebook">
    <fb:login-button
      scope="public_profile,email"
      onlogin="checkLoginStateBtn();">
    </fb:login-button>
  </li>
  <li class="li-message-welcome">
    <a href="#" id="welcome-message" class="hiddendiv">
      <i class="mdi-action-account-box left"></i>Account
    </a>
  </li>
  <li class="li-img-logo">
    <a class="dropdown-trigger" href="#" data-target="dropdown_sesion">
      
      <i class="material-icons right icon-logout">arrow_drop_down</i>
    </a>
  </li>
</ul>
</div>
</nav>

<div class="container" style="width: 100% !important;">
  <div class="section">
    <div class="row">
      <div class="col s6 m3">
        <form name="form-coordenadas-recoleta" method="post" accept-charset="utf-8"
          action="/mapa"
          style="text-align: center">
          <div class="card" onClick="document.forms['form-coordenadas-recoleta'].submit();">
            <div class="card-image">
              
            </div>
            <div class="card-action">
              <input style="display: none" name="coordenadas" value="-7.161696, -78.513005">
              <input style="display: none" name="paradero" value="la_recoleta.jpeg">
              <input style="display: none" name="nombre" value="La Recoleta">
              <input style="display: none" name="idparadero" value="1">
              <input style="display: none" name="waypoints[]" value="-7.165597, -78.508069">
              <button type="submit" class="btn-link">La Recoleta</button>
            </div>
          </div>
        </div>
        <div class="col s6 m3">
          <form name="form-coordenadas-sucre" method="post" accept-charset="utf-8"
            action="/mapa"
            style="text-align: center">
            <div class="card" onClick="document.forms['form-coordenadas-sucre'].submit();">
              <div class="card-image">
                
              </div>
              <div class="card-action">
                <input style="display: none" name="coordenadas" value="-7.161696, -78.513005">
                <input style="display: none" name="paradero" value="la_recoleta.jpeg">
                <input style="display: none" name="nombre" value="Sucre">
                <input style="display: none" name="idparadero" value="2">
                <input style="display: none" name="waypoints[]" value="">
                <button type="submit" class="btn-link">Sucre</button>
              </div>
            </div>
          </form>
        </div>
      </div>
    </div>
  </div>

```


b. JavaScript

```
window.fbAsyncInit = function() {
  FB.init({
    appId: '909525559239068',
    cookie: true,
    xfbml: true,
    version: 'v3.1'
  });

  FB.AppEvents.logPageView();

  checkLoginState();
};

(function(d, s, id) {
  var js, fjs = d.getElementsByTagName(s)[0];
  if (d.getElementById(id)) {return;}
  js = d.createElement(s);
  js.id = id;
  js.src = 'https://connect.facebook.net/en_US/sdk.js';
  fjs.parentNode.insertBefore(js, fjs);
}(document, 'script', 'facebook-jssdk'));

function checkLoginState() {
  FB.getLoginStatus(function(response) {
    statusChangeCallback(response);
  });
}

function statusChangeCallback(response) {
  if (response.status === 'connected') {
    FB.api('/me', function (response) {
      document.getElementById('welcome-message').innerHTML = 'Bienvenido, ' + response.name + '!';
      $('#li-button-facebook').hide();
      $('#img-facebook').attr("src", "https://graph.facebook.com/" + response.id +
"/picture?type=small");
    });
  } else {
    document.getElementById('status').innerHTML = 'Please log ' + 'into this app.';
  }
}

$(window).load(function () {
  //preloader
  $('#status').fadeOut();
  $('#preloader').delay(550).fadeOut('slow');
  $('body').delay(550).css({
    'overflow': 'visible'
  });
});

function secondsTimeSpanToHMS(s) {
  var h = Math.floor(s/3600);
  s -= h*3600;
  var m = Math.floor(s/60);
  s -= m*60;
  return h+":"+(m < 10 ? '0'+m : m)+":"+(s < 10 ? '0'+s : s);
}

$(document).ready(function() {
  socket();
});

function socket() {
  function getNumIncDraft() {
    //
```

```

    io.socket.get('/socketmonitoreo/add_socket_id');
    //

};
getNumIncDraft();
}

```

c. Esilos CSS

```

#preloader {
    position: fixed;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    background-color: #f5f5f5;
    z-index: 1000;
}

.pre-container {
    position: absolute;
    left: 50%;
    top: 50%;
    bottom: auto;
    right: auto;
    -webkit-transform: translateX(-50%) translateY(-50%);
    transform: translateX(-50%) translateY(-50%);
    text-align: center;
}

.logo-circle {
    position: relative;
    background: #f5f5f5;
    color: #fff;
    display: block;
    line-height: 70px;
    font-size: 15px;
    -webkit-animation: white-shadow 1s infinite;
    animation: white-shadow 1s infinite;
    height: 250px;
    width: 250px;
    border-radius: 50%;
    text-align: center;
}

@-webkit-keyframes white-shadow {
    0% {
        box-shadow: 0 0 0 rgba(51, 51, 51, 0.3);
        -webkit-transition: box-shadow 0.3s ease-in-out;
    }
    100% {
        box-shadow: 0 0 30px transparent;
        -webkit-transform: translate3d(0, 0, 0);
        -webkit-transition: box-shadow 0.4s ease-in-out;
    }
}

@keyframes white-shadow {
    0% {
        box-shadow: 0 0 0 rgba(51, 51, 51, 0.3);
        -webkit-transition: box-shadow 0.3s ease-in-out;
        transition: box-shadow 0.3s ease-in-out;
    }
    100% {
        box-shadow: 0 0 30px transparent;
    }
}

```

```

    -webkit-transform: translate3d(0, 0, 0);
    transform: translate3d(0, 0, 0);
    -webkit-transition: box-shadow 0.4s ease-in-out;
    transition: box-shadow 0.4s ease-in-out;
  }
}

.btn-link{
  border:none;
  outline:none;
  background:none;
  cursor:pointer;
  color:#000000;
  padding:0;
  /*text-decoration:underline;*/
  font-family:inherit;
  font-size:20px;
}

footer {
  background-color: #424242 !important;
  padding-top: 0 !important;
  position: fixed;
  left: 0;
  bottom: 0;
  width: 100%;
}

footer strong {
  font-size: 1em !important;
}

footer strong a {
  color: rgb(118,125,49) !important;
}

footer strong a label {
  color: rgb(208,170,43) !important;
  font-size: 1em;
}

footer strong a:hover {
  color: rgb(118,125,49) !important;
  text-decoration: underline !important;
}

footer strong a label:hover {
  color: rgb(208,170,43) !important;
  text-decoration: underline !important;
}

footer.page-footer {
  margin-top: 0 !important;
}

footer.page-footer .footer-copyright {
  height: 35px !important;
  line-height: 35px !important;
  font-size: 0.8rem !important;
}

footer.page-footer .footer-copyright {
  overflow: hidden;
  height: 50px;
  line-height: 50px;
  color: rgba(255, 255, 255, 0.8);
}

```

```

background-color: rgba(51, 51, 51, 0.08);
}

.light, .page-footer .footer-copyright {
font-weight: 300;
}

#footer-height {
padding: 0 5px;
}

```

d. Vista Móvil

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.monitoreo.monitoreo.seleccionar_paradero">

<ScrollView
android:layout_width="fill_parent"
android:layout_height="wrap_content">

<RelativeLayout
android:layout_width="match_parent"
android:layout_height="match_parent">

<LinearLayout
android:id="@+id/llFirst"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginBottom="10sp"
android:layout_marginLeft="4dp"
android:layout_marginRight="4dp"
android:gravity="center_vertical"
android:orientation="horizontal"
android:weightSum="2">

<LinearLayout
android:layout_width="50dp"
android:layout_height="wrap_content"
android:layout_marginRight="2dp"
android:layout_marginTop="4dp"
android:layout_weight="1"
android:background="@android:color/white"
android:gravity="center"
android:onClick="cargarMapa"
android:orientation="vertical"
android:tag="-7.1617738,-78.5129839,La_Recoleta,1">

<ImageView
android:id="@+id/ivLaRecoleta"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:adjustViewBounds="false"
android:background="@mipmap/la_recoleta" />

<TextView
android:id="@+id/tvLaRecoleta"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginBottom="10dp"

```

```
        android:layout_marginTop="10dp"
        android:gravity="center"
        android:text="La Recoleta"
        android:textColor="@android:color/black"
        android:textSize="16dp" />
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="50dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="2dp"
    android:layout_marginTop="4dp"
    android:layout_weight="1"
    android:background="@android:color/white"
    android:gravity="center"
    android:onClick="cargarMapa"
    android:orientation="vertical"
    android:tag="-7.1654716,-78.508790,Sucre,2">
```

```
<ImageView
    android:id="@+id/ivSucre"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:adjustViewBounds="false"
    android:background="@mipmap/sucre" />
```

```
<TextView
    android:id="@+id/tvSucre"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:layout_marginTop="10dp"
    android:gravity="center"
    android:text="Sucre"
    android:textColor="@android:color/black"
    android:textSize="16dp" />
```

```
</LinearLayout>
```

```
</LinearLayout>
```

```
<LinearLayout
    android:id="@+id/llSecond"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/llFirst"
    android:layout_marginBottom="10sp"
    android:layout_marginLeft="4dp"
    android:layout_marginRight="4dp"
    android:gravity="center_vertical"
    android:orientation="horizontal"
    android:weightSum="2">
```

```
<LinearLayout
    android:layout_width="50dp"
    android:layout_height="wrap_content"
    android:layout_marginRight="2dp"
    android:layout_marginTop="4dp"
    android:layout_weight="1"
    android:background="@android:color/white"
    android:gravity="center"
    android:onClick="cargarMapa"
    android:orientation="vertical"
    android:tag="-7.1617738,-78.5129839,Mollepampa,3">
```

```
<ImageView
    android:id="@+id/ivMollepampaUnc"
```

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:adjustViewBounds="false"  
android:background="@mipmap/mollepampa_unc" />
```

```
<TextView  
    android:id="@+id/tvMollepampaUnc"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="10dp"  
    android:layout_marginTop="10dp"  
    android:gravity="center"  
    android:text="Mollepampa UNC"  
    android:textColor="@android:color/black"  
    android:textSize="16dp" />
```

```
</LinearLayout>
```

```
<LinearLayout  
    android:layout_width="50dp"  
    android:layout_height="wrap_content"  
    android:layout_marginLeft="2dp"  
    android:layout_marginTop="4dp"  
    android:layout_weight="1"  
    android:background="@android:color/white"  
    android:gravity="center"  
    android:onClick="cargarMapa"  
    android:orientation="vertical"  
    android:tag="-7.1654716,-78.508790,Fonavi,4">
```

```
<ImageView  
    android:id="@+id/ivFonaviUnc"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:adjustViewBounds="false"  
    android:background="@mipmap/fonavi_unc" />
```

```
<TextView  
    android:id="@+id/tvFonaviUnc"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="10dp"  
    android:layout_marginTop="10dp"  
    android:gravity="center"  
    android:text="Fonavi I UNC"  
    android:textColor="@android:color/black"  
    android:textSize="16dp" />
```

```
</LinearLayout>
```

```
</LinearLayout>
```

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_below="@+id/llSecond"  
    android:layout_marginBottom="10sp"  
    android:layout_marginLeft="4dp"  
    android:layout_marginRight="4dp"  
    android:gravity="center_vertical"  
    android:orientation="horizontal"  
    android:weightSum="2">
```

```
<LinearLayout  
    android:layout_width="50dp"  
    android:layout_height="wrap_content"  
    android:layout_marginRight="2dp"  
    android:layout_marginTop="4dp"
```

```

android:layout_weight="1"
android:background="@android:color/white"
android:gravity="center"
android:onClick="cargarMapa"
android:orientation="vertical"
android:tag="-7.1617738,-78.5129839,La_Recoleta_UNC,5">

```

```

<ImageView
    android:id="@+id/ivLaRecoletaUnc"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:adjustViewBounds="false"
    android:background="@mipmap/la_recoleta_unc" />

```

```

<TextView
    android:id="@+id/tvLaRecoletaUnc"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:layout_marginTop="10dp"
    android:gravity="center"
    android:text="La Recoleta UNC"
    android:textColor="@android:color/black"
    android:textSize="16dp" />

```

```
</LinearLayout>
```

```

<LinearLayout
    android:layout_width="50dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="2dp"
    android:layout_marginTop="4dp"
    android:layout_weight="1"
    android:background="@android:color/white"
    android:gravity="center"
    android:onClick="cargarMapa"
    android:orientation="vertical"
    android:tag="-7.1654716,-78.508790,Sucre">

```

```
</LinearLayout>
```

```
</LinearLayout>
```

```
</RelativeLayout>
```

```
</ScrollView>
```

```
</android.support.constraint.ConstraintLayout>
```

e. Clase móvil

```

public class seleccionar_paradero extends AppCompatActivity {

    TelephonyManager manager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_seleccionar_paradero);

        int permissionCheck = ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_PHONE_STATE);
        int permissionCheckLoc = ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION);

        if (permissionCheck != PackageManager.PERMISSION_GRANTED || permissionCheckLoc !=

```

```

PackageManager.PERMISSION_GRANTED) {
    ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_PHONE_STATE,
Manifest.permission.ACCESS_FINE_LOCATION}, 10);
}

    manager = (TelephonyManager) getSystemService(Context.TELEPHONY_SERVICE);

    if (Build.VERSION.SDK_INT >= 23) {
        Log.d("Prueba ", "Prueba Build.VERSION.SDK_INT >= 23 ");
        Intent intent = new Intent();
        String packageName = getApplicationContext().getPackageName();
        PowerManager pm = (PowerManager)
getApplicationContext().getSystemService(Context.POWER_SERVICE);
        if (!pm.isIgnoringBatteryOptimizations(packageName)) {
            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            intent.setAction(Settings.ACTION_REQUEST_IGNORE_BATTERY_OPTIMIZATIONS);
            intent.setData(Uri.parse("package:" + packageName));
            getApplicationContext().startActivity(intent);
        }
    }
}

    public void cargarMapa(View v) {
        Intent i = new Intent(getApplicationContext(), MainActivity.class);
        Bundle bolsa = new Bundle();
        bolsa.putString("paradero", v.getTag().toString());
        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.READ_PHONE_STATE) !=
PackageManager.PERMISSION_GRANTED) {
            return;
        }
        bolsa.putString("idDispositivo", manager.getDeviceId());
        i.putExtras(bolsa);
        startActivity(i);
    }
}

```

➤ MONITOREO

a. Controlador

```

module.exports = {
    interface_monitoreo: function(req, res) {
        res.view('mapa', {coordenadas: req.param('coordenadas'),
            paradero: req.param('paradero'),
            nombre: req.param('nombre')});
    }
};

```

b. Vista

```

<link rel="stylesheet" href="/styles/mapa.css">
<ul id="dropdown_sesion" class="dropdown-content">
  <li class="divider"></li>
  <li>
    <a href="#" onclick="logoutFacebook()">Cerrar Sesión</a>
  </li>
</ul>
<nav class="light-blue lighten-1" role="navigation">
  <div class="nav-wrapper container">
    <a id="logo-container" href="#" class="brand-logo">El Carro de la U</a>

```

```

<ul class="right side-nav">
  <li id="li-button-facebook">
    <fb:login-button
      scope="public_profile,email"
      onlogin="checkLoginState();">
    </fb:login-button>
    <a href="#" id="login" class="brand-logo">
      <div class="fb-login-button" data-width="" data-size="large" data-button-type="login_with"
data-auto-logout-link="false" data-use-continue-as="true"></div>
    </a>
    <a href="#" id="login" class="btn btn-primary">Iniciar sesión</a>
  </li>
  <li>
    <a href="#" id="welcome-message">
      <i class="mdi-action-account-box left"></i>
    </a>
  </li>
  <li>
    <a class="dropdown-trigger" href="#" data-target="dropdown_sesion">
      
      <i class="material-icons right">arrow_drop_down</i>
    </a>
  </li>
</ul>
<a href="#" data-target="nav-mobile" class="sidenav-trigger"><i class="material-
icons">menu</i></a>
</div>
</nav>
<input id="coordenadas" name="coordenadas" value="<%= coordenadas %>" style="display:
none">
<input id="paradero" name="paradero" value="<%= paradero %>" style="display: none">
<input id="nombre" name="nombre" value="<%= nombre %>" style="display: none">
<div id='mapa-form' class="row form-container mapa-form-container">
  <div id="container-mapa" class="container content-mapa">
    <div>
      <form id="form-mapa" class="col s12" style="height: 88vh!important;">
        <div class="row" style="height: 88vh!important;">
          <div id="map" class="pushmenu-push"></div>
        </div>
      </form>
    </div>
  </div>
</div>
<div id="directions-panel"></div>
<div id="modal-cambiar-paradero" class="modal modal-class-cambiar-paradero">
  <div class="modal-content">
    <div class="row">
      <div class="col s4 m3">
        <div class="card">
          <div class="card-image">
            
          </div>
          <div class="card-action">
            <form name="form-coordenadas" method="post" accept-charset="utf-8" action="/mapa"
style="text-align: center">
              <input style="display: none" name="coordenadas" value="-7.161696, -78.513005">
              <input style="display: none" name="paradero" value="la_recoleta.jpeg">
              <input style="display: none" name="nombre" value="La Recoleta">
              <button type="submit" class="btn-link">La Recoleta</button>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

</div>
<div class="col s4 m3">
  <div class="card">
    <div class="card-image">
      
    </div>
    <div class="card-action">
      <form name="form-coordenadas" method="post" accept-charset="utf-8" action="/mapa"
style="text-align: center">
        <input style="display: none" name="coordenadas" value="-7.161696, -78.513005">
        <input style="display: none" name="paradero" value="la_recoleta.jpeg">
        <input style="display: none" name="nombre" value="La Recoleta">
        <button type="submit" class="btn-link">Sucre</button>
      </form>
    </div>
  </div>
</div>
<div class="col s4 m3">
  <div class="card">
    <div class="card-image">
      
    </div>
    <div class="card-action">
      <form name="form-coordenadas" method="post" accept-charset="utf-8" action="/mapa"
style="text-align: center">
        <input style="display: none" name="coordenadas" value="-7.161696, -78.513005">
        <input style="display: none" name="paradero" value="la_recoleta.jpeg">
        <input style="display: none" name="nombre" value="La Recoleta">
        <button type="submit" class="btn-link">Mollepampa UNC</button>
      </form>
    </div>
  </div>
</div>
<div class="col s4 m3">
  <div class="card">
    <div class="card-image">
      
    </div>
    <div class="card-action">
      <form name="form-coordenadas" method="post" accept-charset="utf-8" action="/mapa"
style="text-align: center">
        <input style="display: none" name="coordenadas" value="-7.161696, -78.513005">
        <input style="display: none" name="paradero" value="la_recoleta.jpeg">
        <input style="display: none" name="nombre" value="La Recoleta">
        <button type="submit" class="btn-link">Fonavi I UNC</button>
      </form>
    </div>
  </div>
</div>
<div class="col s4 m3">
  <div class="card">
    <div class="card-image">
      
    </div>
    <div class="card-action">
      <form name="form-coordenadas" method="post" accept-charset="utf-8" action="/mapa"
style="text-align: center">
        <input style="display: none" name="coordenadas" value="-7.161696, -78.513005">
        <input style="display: none" name="paradero" value="la_recoleta.jpeg">
        <input style="display: none" name="nombre" value="La Recoleta">
        <button type="submit" class="btn-link">La Recoleta UNC</button>
      </form>
    </div>
  </div>
</div>

```

```

    </div>
  </div>
</div>
</div>
<div class="modal-footer">
  <a href="#" class="modal-close waves-effect waves-green btn-flat">Agree</a>
</div>
</div>

```

c. JavaScript

```

window.id_facebook = "";

$(function() {

  var app_id = '909525559239068';
  var scopes = 'public_profile, email, user_friends, user_online_presence';

  var btn_login = '<a href="#" id="login" class="brand-logo">'+
    '<div class="fb-login-button" data-width="" data-size="large" data-button-type="login_with" data-auto-logout-link="false" data-use-continue-as="true"></div>' +
    '</a>';

  var div_session = "<div id='facebook-session'>" +
    "<strong></strong>" +
    "<img>" +
    "<a href='#' id='logout' class='btn btn-danger'>Cerrar sesión</a>" +
    "</div>";

  window.fbAsyncInit = function() {

    FB.init({
      appId    : app_id,
      status   : true,
      cookie   : true,
      xfbml    : true,
      version  : 'v2.1'
    });

    FB.getLoginStatus(function(response) {
      statusChangeCallback(response, function() {});
    });
  };

  var statusChangeCallback = function(response, callback) {
    console.log(response);

    if (response.status === 'connected') {
      getFacebookData();
    } else {
      callback(false);
    }
  }

  var checkLoginState = function(callback) {
    FB.getLoginStatus(function(response) {
      callback(response);
    });
  }

  var getFacebookData = function() {
    FB.api('/me', function(response) {
      $('#login').after(div_session);
    });
  }

```

```

    $('#login').remove();
    $('#facebook-session strong').text("Bienvenido: "+response.name);
    $('#facebook-session
img').attr('src', 'http://graph.facebook.com/'+response.id+'/picture?type=large');
    });
}

var facebookLogin = function() {
    checkLoginState(function(data) {
        if (data.status !== 'connected') {
            FB.login(function(response) {
                if (response.status === 'connected')
                    getFacebookData();
            }, {scope: scopes});
        }
    })
}

var facebookLogout = function() {
    checkLoginState(function(data) {
        if (data.status === 'connected') {
            FB.logout(function(response) {
                $('#facebook-session').before(btn_login);
                $('#facebook-session').remove();
            })
        }
    })
}

$(document).on('click', '#login', function(e) {
    e.preventDefault();

    facebookLogin();
})

$(document).on('click', '#logout', function(e) {
    e.preventDefault();

    if (confirm("¿Está seguro?"))
        facebookLogout();
    else
        return false;
})

$(document).ready(function() {
    //DETECTAR SISTEMA OPERATIVO
    var userAgent = navigator.userAgent || navigator.vendor || window.opera;
    if (/iPad|iPhone|iPod/.test(userAgent) && !window.MSStream) {
        $('#container-mapa').perfectScrollbar();
        $('.content-mapa').height(parseInt($(window).height()));
    }

    $(".dropdown-trigger").dropdown();

    $('#modal').modal();
    window.directionsService = new google.maps.DirectionsService;
    window.directionsDisplay = new google.maps.DirectionsRenderer;

    socketMonitoreo();
    load_data_ini_mapa();
});

window.onresize = function() {
    //DETECTAR SISTEMA OPERATIVO
    var userAgent = navigator.userAgent || navigator.vendor || window.opera;

```

```

if (/iPad|iPhone|iPod/.test(userAgent) && !window.MSStream) {
    $('#content-mapa').height(parseInt($(window).height()));
}
};

function load_data_ini_mapa() {
    setTimeout(function() {
        initMap();
    }, 1550);
}

function initMap() {
    window.markers = [];

    var lat_lng_actual = "";

    var myLatLng = new google.maps.LatLng(-7.165715, -78.502748);
    var mapOptions = {
        zoom: 15,
        center: myLatLng,
        disableDefaultUI: true,
        zoomControl: true,
        minZoom: 13,
        fullscreenControl: true
    };

    window.map = new google.maps.Map(document.getElementById('map'), mapOptions);

    directionsDisplay.setMap(map);

    var infoWindow = new google.maps.InfoWindow;
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(function(position) {
            var pos = {
                lat: position.coords.latitude,
                lng: position.coords.longitude
            };

            lat_lng_actual = position.coords.latitude + ',' + position.coords.longitude;

            var icon = {
                url: id_facebook, // url
                scaledSize: new google.maps.Size(60, 60), // scaled size
            };

            var marker_facebook = new MarkerWithLabel({
                position: pos,
                icon: icon,
                id: 'marcador_facebook',
                draggable: false,
                raiseOnDrag: true,
                map: map,
                labelClass: "label_facebook",
                optimized: false
            });
            var tiem_pie = getTimePie($('#coordenadas').val(), lat_lng_actual);
            var tiem_carro = getTimeCarro($('#coordenadas').val(), lat_lng_actual);
            var distancia = getDistancia($('#coordenadas').val(), lat_lng_actual);
        }, function() {
            handleLocationError(true, infoWindow, map.getCenter());
        });
    } else {
        handleLocationError(false, infoWindow, map.getCenter());
    }
    var lat_lon = $('#coordenadas').val().split(',');
    var content = '<div class="col s12">' +

```

```

'<div class="card-panel grey lighten-5 z-depth-1">' +
'<div class="row valign-wrapper">' +
'<div class="col s2">' +
'' +
'</div>' +
'<div class="col s10">' +
'<span class="black-text">' +
'This is a square image. Add the "circle" class to it to make it appear circular.' +
'</span>' +
'</div>' +
'</div>' +
'</div>';
var infowindow = new google.maps.InfoWindow({
  content: content
});

var marker_paradero = new MarkerWithLabel({
  position: new google.maps.LatLng(parseFloat(lat_lon[0]),
    parseFloat(lat_lon[1])),
  icon: '/images/parada-de-autobus.png',
  id: 'marcador_paradero',
  draggable: false,
  raiseOnDrag: true,
  map: map,
  optimized: false
});

markers.push(marker_paradero);
google.maps.event.addListener(marker_paradero, 'click', function() {
  infowindow.open(map,marker_paradero); //map and marker are the variables defined previously
});
google.maps.event.addListener(map, 'click', function() {
  infowindow.close();
});
}

function handleLocationError(browserHasGeolocation, infoWindow, pos) {
  infoWindow.setPosition(pos);
  infoWindow.setContent(browserHasGeolocation ?
    'Error: The Geolocation service failed.' :
    'Error: Your browser doesn\'t support geolocation.');
```

```

  infoWindow.open(map);
}

function getTimePie(cord_paradero, cord_actual) {
  var directionsService = new google.maps.DirectionsService();
  var tiempo_caminando = "";

  var request = {
    origin: cord_actual,
    destination: cord_paradero,
    travelMode: 'WALKING'
  };
  directionsService.route(request, function(result, status) {
    console.log(result)
    if (status == 'OK') {
      $('#distancia_paradero').text(result.routes[0].legs[0].duration.text);
      console.log(result.routes[0].legs[0].duration.text);
    }else{
      return "";
    }
  });
}

function getTimeCarro(cord_paradero, cord_actual) {
  var directionsService = new google.maps.DirectionsService();

```

```

var tiempo_carro = "";

var request = {
  origin: cord_actual,
  destination: cord_paradero,
  travelMode: 'DRIVING'
};
directionsService.route(request, function(result, status) {
  if (status === 'OK') {
    console.log(result.routes[0].legs[0].duration.text);
  }
});

return tiempo_carro;
}

function getDistancia(cord_paradero, cord_actual) {
  var directionsService = new google.maps.DirectionsService();
  var tiempo_carro = "";

  var request = {
    origin: cord_actual,
    destination: cord_paradero,
    travelMode: 'DRIVING'
  };
  directionsService.route(request, function(result, status) {
    if (status === 'OK') {
      console.log(result.routes[0].legs[0].distance.text);
    }
  });

  console.log(tiempo_carro)

  return tiempo_carro;
}

function actualizarBus(coordenadas) {
  var existe_marcador_bus = false;
  var lat_lon = coordenadas.split(',');
  var tiempo_bus = "";
  for (var i = 0; i < markers.length; i++) {
    if(markers[i].id === 'marcador_bus')
      existe_marcador_bus = true;
  }

  var directionsService = new google.maps.DirectionsService();

  var start = coordenadas;
  var end = $('#coordenadas').val();
  var request = {
    origin: start,
    destination: end,
    travelMode: 'DRIVING'
  };
  directionsService.route(request, function(result, status) {
    if (status === 'OK')
      tiempo_bus = result.routes[0].legs[0].duration.text;
  });

  setTimeout(function() {
    if(existe_marcador_bus){
      var newLatLng = new google.maps.LatLng(parseFloat(lat_lon[0]), parseFloat(lat_lon[1]));
      markers[1].animateTo(newLatLng, {easing: 'linear', duration: parseInt(1000)});
      markers[1].setOptions({labelContent: tiempo_bus});
    }else{
      var marker_bus = new MarkerWithLabel({
        position: new google.maps.LatLng(parseFloat(lat_lon[0]), parseFloat(lat_lon[1])),

```

```

        icon: '/images/marcador-bus.png',
        id: 'marcador_bus',
        draggable: false,
        raiseOnDrag: true,
        map: map,
        optimized: false,
        labelContent: tiempo_bus,
        labelAnchor: new google.maps.Point(22, 0),
        labelClass: "label_default",
        labelStyle: {opacity: 0.75, 'background-color': 'rgb(127,169,23)'}
    });

    markers.push(marker_bus);
}
}, 1350);
}

function calculateAndDisplayRoute(coordenadas) {
    var waypts = [];
    directionsService.route({
        origin: coordenadas,
        destination: $('#coordenadas').val(),
        // waypoints: waypts,
        optimizeWaypoints: true,
        travelMode: 'DRIVING'
    }, function(response, status) {
        if (status === 'OK') {
            directionsDisplay.setDirections(response);
            var route = response.routes[0];
            var summaryPanel = document.getElementById('directions-panel');
            summaryPanel.innerHTML = "";
            // For each route, display summary information.
            for (var i = 0; i < route.legs.length; i++) {
                var routeSegment = i + 1;
                summaryPanel.innerHTML += '<b>Route Segment: ' + routeSegment +
                    '</b><br>';
                summaryPanel.innerHTML += route.legs[i].start_address + ' to ';
                summaryPanel.innerHTML += route.legs[i].end_address + '<br>';
                summaryPanel.innerHTML += route.legs[i].distance.text + '<br><br>';
            }
        } else {
            window.alert('Directions request failed due to ' + status);
        }
    });
}

function socketMonitoreo() {
    io.socket.get('/socketmonitoreo/add_socket_id');

    io.socket.on('socketmonitoreo', function(obj) {
        if (obj.verb === 'created') {
            actualizarBus(obj.data.coordenadas)
        }
    });
}

```

d. Estilos CSS

```

div>img[src^='https://graph.facebook.com/10214876563379142/picture?type=small']{
    -webkit-mask-image: url(/images/pin.svg) !important;
    mask-image: url(/images/pin.svg) !important;
    -webkit-mask-position: center center !important;
    mask-position: center center !important;
    -webkit-mask-repeat: no-repeat !important;
    mask-repeat: no-repeat !important;
}

```

```

mask-border-width: auto !important;
mask-border-width: 1rem !important;
mask-border-width: 25% !important;
mask-border-width: 3 !important;
mask-border-width: 2em 3em !important;
mask-border-width: 5% 15% 10% !important;
mask-border-width: 5% 2em 10% auto !important;
}

.content-mapa{
width: 100% !important;
}

.container{
max-width: none!important;
}

.modal-class-cambiar-paradero{
width: 90% !important;
height: 80% !important;
min-height: 80% !important;
}

#map {
width: 100% !important;
height: 97% !important;
margin: 0 !important;
padding: 0 !important;
font-family: "Lucida Grande", Helvetica, Arial, sans-serif !important;
font-size: 12px !important;
}

.gmnoprint a, .gmnoprint span, .gm-style-cc {
display:none;
}
img[src*="https://maps.gstatic.com/mapfiles/api-3/images/google_white5_hdpi.png"] {
display: none;
}
@media (max-device-width: 500px) {
#map {
width: 100%;
height: 440px;
margin: 0;
padding: 0;
font-family: "Lucida Grande", Helvetica, Arial, sans-serif;
font-size: 12px;
}
}

.label_default {
width: 50px;
height: 20px;
border: 1px solid #000000;
border-radius: 5px;
text-align: center;
line-height: 20px;
font-weight: bold;
font-size: 14px;
color: #FFFFFF;
}

```

e. Vista Móvil

```

<?xml version="1.0" encoding="utf-8"?>
<com.monitoreo.elcarrodelau.ayudas.mapa.MapWrapperLayout

```

```

xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/map_relative_layout"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.monitoreo.elcarrodelau.MapsActivity">

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <fragment xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:id="@+id/map"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context="com.developers.uberanimation.MapsActivity" />

</RelativeLayout>
</com.monitoreo.elcarrodelau.ayudas.mapa.MapWrapperLayout>

```

f. Clase Móvil

public class MapsActivity **extends** FragmentActivity **implements** OnMapReadyCallback, GoogleApiClient.ConnectionCallbacks, GoogleApiClient.OnConnectionFailedListener, ResultCallback<Status>, ActivityCompat.OnRequestPermissionsResultCallback, LocationListener {

```

    private static final String TAG = MapsActivity.class.getSimpleName();
    SupportMapFragment mapMonitoreo;
    private GoogleMap mMapMonitoreo;

```

```

    private float v;
    private double lat, lng;
    private LatLng startPosition, endPosition;
    private LatLng actualPosicionBus, posicionParadero;

```

```

    Bundle bolsa;

```

```

    public Socket mSocket;
    String idParadero;

```

```

    double latitudBus, longitudBus;
    String coordenadasParadero, modoUsuario = "walking";

```

```

    private Marker markerParadero;
    private View popup_paradero = null;
    TextView tvNombreParadero, tvDistancia, tvTiempoAuto, tvTiempoCaminando;
    ImageView ivLogoParaderoMapa;
    private OnInfoWindowElemTouchListener infoButtonListenerRuta,
    infoButtonListenerCambiarParadero, infoRICarro, infoRICaminando;
    Button btnRutaMapa, btnCambiarParadero;
    RelativeLayout rlTiempoCarroSeleccionado, rlTiempoCarro, rlTiempoCaminandoSeleccionado,
    rlTiempoCaminando;
    private List<LatLng> polyLineList;
    private PolylineOptions polylineOptions, blackPolylineOptions;
    private Polyline blackPolyline, greyPolyline;

    private Marker markerBus;
    private View popup_bus = null;
    Button btnRutaBus;
    private OnInfoWindowElemTouchListener infoButtonListenerRutaBus;
    Boolean markerInfoIsShowBus = false, isFirstView = true, isLastPolyline = false, isVerRutaBus =
    false;

```

```

TextView tvNombreParaderoBus, tvDistanciaBus, tvTiempoBus;
ImageView ivLogoParaderoBus;
private List<LatLng> polyLineListBus;
private PolylineOptions polylineOptionsBus, blackPolylineOptionsBus;
private Polyline blackPolylineBus, blackPolylineBusLast, greyPolyLineBus;

public static final int REQUEST_LOCATION = 1;
private GoogleApiClient mGoogleApiClient;
private LocationRequest mLocationRequest;
private LocationSettingsRequest mLocationSettingsRequest;
protected static final int REQUEST_CHECK_SETTINGS = 0x1;
Marker myLocationMarker = null;

String latitudActual, longitudActual, latitudParadero, longitudParadero, nombreParadero;
Boolean markerInfoIsShow = false, verRutaUsuario = false;

String ultimo_tiempo = "";
String waypointsRutaBus;

@SuppressLint("WrongViewCast")
MapViewWrapperLayout mapViewWrapperLayout;

TextView tvLabel;

GoogleApi googleApi;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maps);

    bolsa = getIntent().getExtras();
    coordenadasParadero = bolsa.getString("paradero");
    waypointsRutaBus = bolsa.getString("waypoints");
    String[] separatedParadero = coordenadasParadero.split(",");
    latitudParadero = separatedParadero[0];
    longitudParadero = separatedParadero[1];
    nombreParadero = separatedParadero[2].replaceAll("_", " ");
    idParadero = separatedParadero[3];

    mapMonitoreo = (SupportMapFragment)
    getSupportFragmentManager().findFragmentById(R.id.map);
    mapMonitoreo.getMapAsync(MapsActivity.this);

    googleApi = new GoogleApi();

    try {
        mSocket = IO.socket("http://192.168.1.84:1337/?__sails_io_sdk_version=0.13.5");
    } catch (URISyntaxException e) {
        throw new RuntimeException(e);
    }

    JSONObject jsonObject = new JSONObject();
    try {
        jsonObject.put("url", "/socketmonitoreo/add_socket_id");
    } catch (JSONException e) {
        e.printStackTrace();
    }
    mSocket.emit("get", jsonObject, new Ack() {
        @Override
        public void call(Object... args) {
            Log.d("MiActividad", "records: " + args[0].toString());
        }
    });
    mSocket.connect();
    mSocket.emit("type", "activity");
    mSocket.on("socketmonitoreo", socket_notificacion);

```

```

    polyLineList = new ArrayList<>();

    createLocationRequest();
    buildLocationSettingsRequest();
    buildGoogleApiClient();
}

private void createLocationRequest() {
    mLocationRequest = new LocationRequest()
        .setInterval(Constants.UPDATE_INTERVAL)
        .setFastestInterval(Constants.UPDATE_FASTEST_INTERVAL)
        .setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
}

private void buildLocationSettingsRequest() {
    LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder();
    builder.addLocationRequest(mLocationRequest)
        .setAlwaysShow(true);
    mLocationSettingsRequest = builder.build();
}

private void buildGoogleApiClient() {
    mGoogleApiClient = new GoogleApiClient.Builder(this)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .addApi(LocationServices.API)
        .addApi(ActivityRecognition.API)
        .enableAutoManage(this, this)
        .build();
    mGoogleApiClient.connect();
    settingsrequest();
}

public void settingsrequest() {
    LocationRequest locationRequest = LocationRequest.create();
    locationRequest.setPriority(LocationRequest.PRIORITY_BALANCED_POWER_ACCURACY);
    locationRequest.setInterval(30 * 1000);
    locationRequest.setFastestInterval(5 * 1000);
    LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder()
        .addLocationRequest(locationRequest);
    builder.setAlwaysShow(true); //this is the key ingredient

    PendingResult<LocationSettingsResult> result =
        LocationServices.SettingsApi.checkLocationSettings(mGoogleApiClient, builder.build());
    result.setResultCallback(new ResultCallback<LocationSettingsResult>() {
        @Override
        public void onResult(LocationSettingsResult result) {
            final Status status = result.getStatus();
            final LocationSettingsStates state = result.getLocationSettingsStates();
            switch (status.getStatusCode()) {
                case LocationSettingsStatusCodes.SUCCESS:
                    break;
                case LocationSettingsStatusCodes.RESOLUTION_REQUIRED:
                    try {
                        status.startResolutionForResult(MapsActivity.this, REQUEST_CHECK_SETTINGS);
                    } catch (IntentSender.SendIntentException e) {
                    }
                    break;
                case LocationSettingsStatusCodes.SETTINGS_CHANGE_UNAVAILABLE:
                    break;
            }
        }
    });
}

```

```

private Emitter.Listener socket_notificacion = new Emitter.Listener() {
    @Override
    public void call(Object... args) {
        final JSONObject data = (JSONObject) args[0];
        final String[] coord = { "", "" };

        MapsActivity.this.runOnUiThread(new Runnable() {
            @Override
            public void run() {
                try {
                    if (data.getString("verb").equals("created")) {
                        JSONObject data_b = new JSONObject(data.getString("data"));
                        coord[0] = data_b.getString("coordenadas");
                        coord[1] = data_b.getString("idParadero");
                        String[] coord_n = coord[0].split(",");

                        if (idParadero.equals(coord[1])) {
                            latitudBus = Double.parseDouble(coord_n[0]);
                            longitudBus = Double.parseDouble(coord_n[1]);
                            googleApi.tiempo(coord_n[0], coord_n[1], latitudParadero, longitudParadero,
                                getApplicationContext(), "driving", "tiempoLlegada", waypointsRutaBus);

                            if (markerInfoIsShowBus)
                                markerBus.showInfoWindow();

                            calcular_ruta();
                        }
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        });
    }
};

@Override
public void onMapReady(GoogleMap googleMap) {
    mMapMonitoreo = googleMap;
    mMapMonitoreo.setMapType(GoogleMap.MAP_TYPE_NORMAL);
    mMapMonitoreo.setTrafficEnabled(false);
    mMapMonitoreo.setIndoorEnabled(false);
    mMapMonitoreo.setBuildingsEnabled(false);
    mMapMonitoreo.getUiSettings().setZoomControlsEnabled(true);
    if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) !=
        PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
    {
        return;
    }
    mMapMonitoreo.setMyLocationEnabled(true);
    mMapMonitoreo.clear();

    posicionParadero = new LatLng(Double.parseDouble(latitudParadero),
        Double.parseDouble(longitudParadero));

    markerParadero = mMapMonitoreo.addMarker(new MarkerOptions()
        .title("Marker in Home")
        .icon(bitmapDescriptorFromVector(this, R.drawable.ic_parada_de_autobus))
        .position(posicionParadero)
        .snippet("Paradero"));

    mMapMonitoreo.moveCamera(CameraUpdateFactory.newLatLng(posicionParadero));
    mMapMonitoreo.moveCamera(CameraUpdateFactory.newCameraPosition(new
        CameraPosition.Builder()
            .target(googleMap.getCameraPosition().target)

```

```

        .zoom(17)
        .bearing(0)
        .build());

mapWrapperLayout = (MapWrapperLayout) findViewById(R.id.map_relative_layout);
mapWrapperLayout.init(mMapMonitoreo, getPixelsFromDp(this, 39 + 20));

if (popup_paradero == null) {
    popup_paradero = getLayoutInflater().inflate(R.layout.popup_paradero, null);
}

btnRutaMapa = (Button) popup_paradero.findViewById(R.id.btnRutaParadero);
infoButtonListenerRuta = new OnInfoWindowElemTouchListener(btnRutaMapa) {
    @Override
    protected void onClickConfirmed(View v, Marker marker) {
        verRutaUsuario = true;
        verRuta();
    }
};
btnRutaMapa.setOnTouchListener(infoButtonListenerRuta);

btnCambiarParadero = (Button) popup_paradero.findViewById(R.id.btnCambiarParadero);
infoButtonListenerCambiarParadero = new
OnInfoWindowElemTouchListener(btnCambiarParadero) {
    @Override
    protected void onClickConfirmed(View v, Marker marker) {
        Intent i = new Intent(getApplicationContext(), PopupParaderos.class);
        i.putExtras(bolsa);
        startActivity(i);
    }
};
btnCambiarParadero.setOnTouchListener(infoButtonListenerCambiarParadero);

rlTiempoCarroSeleccionado = (RelativeLayout)
popup_paradero.findViewById(R.id.rlTiempoCarroSeleccionado);
rlTiempoCaminandoSeleccionado = (RelativeLayout)
popup_paradero.findViewById(R.id.rlTiempoCaminandoSeleccionado);
rlTiempoCaminando = (RelativeLayout)
popup_paradero.findViewById(R.id.rlTiempoCaminando);

rlTiempoCarro = (RelativeLayout) popup_paradero.findViewById(R.id.rlTiempoCarro);
infoRlCarro = new OnInfoWindowElemTouchListener(rlTiempoCarro) {
    @Override
    protected void onClickConfirmed(View v, Marker marker) {
        rlTiempoCarro.setVisibility(View.GONE);
        rlTiempoCarroSeleccionado.setVisibility(View.VISIBLE);
        rlTiempoCaminandoSeleccionado.setVisibility(View.GONE);
        rlTiempoCaminando.setVisibility(View.VISIBLE);
        modoUsuario = "driving";
        googleApi.distancia(latitudActual, longitudActual, latitudParadero, longitudParadero,
getApplicationContext(), modoUsuario, "distancia", "");
        markerParadero.showInfoWindow();
        verRuta();
    }
};
rlTiempoCarro.setOnTouchListener(infoRlCarro);

infoRlCaminando = new OnInfoWindowElemTouchListener(rlTiempoCaminando) {
    @Override
    protected void onClickConfirmed(View v, Marker marker) {
        rlTiempoCarro.setVisibility(View.VISIBLE);
        rlTiempoCarroSeleccionado.setVisibility(View.GONE);
        rlTiempoCaminandoSeleccionado.setVisibility(View.VISIBLE);
        rlTiempoCaminando.setVisibility(View.GONE);
        modoUsuario = "walking";
        googleApi.distancia(latitudActual, longitudActual, latitudParadero, longitudParadero,
getApplicationContext(), modoUsuario, "distancia", "");
    }
};

```

```

        markerParadero.showInfoWindow();
        verRuta();
    }
};
rITiempoCaminando.setOnTouchListener(infoRICaminando);

if (popup_bus == null) {
    popup_bus = getLayoutInflater().inflate(R.layout.popup_bus, null);
}
btnRutaBus = (Button) popup_bus.findViewById(R.id.btnRutaBus);
infoButtonListenerRutaBus = new OnInfoWindowElemTouchListener(btnRutaBus) {
    @Override
    protected void onClickConfirmed(View v, Marker marker) {
        isVerRutaBus = true;
        Log.d("elimina", "onclickconfirmed");
        verRutaBus();
    }
};
btnRutaBus.setOnTouchListener(infoButtonListenerRutaBus);

mMapMonitoreo.setOnMapClickListener(new GoogleMap.OnMapClickListener() {
    @Override
    public void onMapClick(LatLng latLng) {
        if (markerInfoIsShow)
            markerInfoIsShow = false;

        if (markerInfoIsShowBus)
            markerInfoIsShowBus = false;
    }
});

mMapMonitoreo.setInfoWindowAdapter(new GoogleMap.InfoWindowAdapter() {
    @Override
    public View getInfoWindow(Marker marker) {

        return null;
    }

    @Override
    public View getInfoContents(Marker marker) {
        if (marker.getSnippet().equals("Paradero")) {
            markerInfoIsShow = true;
            markerInfoIsShowBus = false;
            if (popup_paradero == null) {
                popup_paradero = getLayoutInflater().inflate(R.layout.popup_paradero, null);
            }

            tvNombreParadero = (TextView) popup_paradero.findViewById(R.id.tvNombreParadero);
            tvDistancia = (TextView) popup_paradero.findViewById(R.id.tvDistancia);
            tvTiempoAuto = (TextView) popup_paradero.findViewById(R.id.tvTiempoAuto);
            tvTiempoCaminando = (TextView)
popup_paradero.findViewById(R.id.tvTiempoCaminando);
            ivLogoParaderoMapa = (ImageView)
popup_paradero.findViewById(R.id.ivLogoNegocioMapa);

            switch (nombreParadero) {
                case "La Recoleta":

ivLogoParaderoMapa.setBackgroundDrawable(getResources().getDrawable(R.mipmap.la_recoleta));
                break;
                case "Sucre":

ivLogoParaderoMapa.setBackgroundDrawable(getResources().getDrawable(R.mipmap.sucre));
                break;
            }
        }
    }
});

```

```

        infoButtonListenerRuta.setMarker(marker);
        infoButtonListenerCambiarParadero.setMarker(marker);
        infoRICarro.setMarker(marker);
        infoRICaminando.setMarker(marker);

        tvNombreParadero.setText("Destino: " + nombreParadero);

        googleApi.tiempo(latitudActual, longitudActual, latitudParadero, longitudParadero,
getApplicationContext(), "walking", "tiempoPie", "");
        googleApi.tiempo(latitudActual, longitudActual, latitudParadero, longitudParadero,
getApplicationContext(), "driving", "tiempoCarro", "");
        googleApi.distancia(latitudActual, longitudActual, latitudParadero, longitudParadero,
getApplicationContext(), modoUsuario, "distancia", "");

        SharedPreferences sharedPreferences =
getSharedPreferences("com.monitoreo.elcarrodelau.prefs", Context.MODE_PRIVATE);
        String tiempo_pie = sharedPreferences.getString("tiempoPie", "");
        String tiempo_carro = sharedPreferences.getString("tiempoCarro", "");
        String distancia = sharedPreferences.getString("distancia", "");

        tvDistancia.setText("Distancia: " + distancia);
        tvTiempoCaminando.setText(tiempo_pie);
        tvTiempoAuto.setText(tiempo_carro);

        mapWrapperLayout.setMarkerWithInfoWindow(marker, popup_paradero);
        return (popup_paradero);
    } else if (marker.getSnippet().equals("Bus")) {
        markerInfoIsShowBus = true;
        markerInfoIsShow = false;

        if (popup_bus == null) {
            popup_bus = getLayoutInflater().inflate(R.layout.popup_bus, null);
        }

        tvNombreParaderoBus = (TextView) popup_bus.findViewById(R.id.tvNombreParaderoBus);
        tvDistanciaBus = (TextView) popup_bus.findViewById(R.id.tvDistanciaBus);
        tvTiempoBus = (TextView) popup_bus.findViewById(R.id.tvTiempoBus);
        ivLogoParaderoBus = (ImageView) popup_bus.findViewById(R.id.ivLogoParaderoBus);

        switch (nombreParadero) {
            case "La Recoleta":

ivLogoParaderoBus.setBackgroundDrawable(getResources().getDrawable(R.mipmap.la_recoleta));
                break;
            case "Sucre":

ivLogoParaderoBus.setBackgroundDrawable(getResources().getDrawable(R.mipmap.sucre));
                break;
        }

        infoButtonListenerRutaBus.setMarker(marker);

        googleApi.tiempo(String.valueOf(latitudBus), String.valueOf(longitudBus), latitudParadero,
longitudParadero, getApplicationContext(), "driving", "tiempoBus", waypointsRutaBus);
        googleApi.distancia(String.valueOf(latitudBus), String.valueOf(longitudBus), latitudParadero,
longitudParadero, getApplicationContext(), modoUsuario, "distanciaBus", waypointsRutaBus);

        SharedPreferences sharedPreferences =
getSharedPreferences("com.monitoreo.elcarrodelau.prefs", Context.MODE_PRIVATE);
        String tiempo_bus = sharedPreferences.getString("tiempoBus", "");
        String distancia_bus = sharedPreferences.getString("distanciaBus", "");

        tvNombreParaderoBus.setText("Destino: " + nombreParadero);
        tvDistanciaBus.setText("Distancia: " + distancia_bus);
        tvTiempoBus.setText("Tiempo: " + tiempo_bus);

        mapWrapperLayout.setMarkerWithInfoWindow(marker, popup_bus);

```

```

        return (popup_bus);
    } else
        return null;
    }
});

}

public static int getPixelsFromDp(Context context, float dp) {
    final float scale = context.getResources().getDisplayMetrics().density;
    return (int) (dp * scale + 0.5f);
}

public void calcular_ruta() {
    actualPosicionBus = new LatLng(latitudBus, longitudBus);
    SharedPreferences sharedPreferences = getSharedPreferences("com.monitoreo.elcarrodelau.prefs",
Context.MODE_PRIVATE);
    final String tiempo_llegada = sharedPreferences.getString("tiempoLlegada", "");
    if (markerBus == null) {
        ultimo_tiempo = tiempo_llegada;

        markerBus = mMapMonitoreo.addMarker(new MarkerOptions()
            .position(actualPosicionBus)
            .flat(true)
            .snippet("Bus")
            .icon(BitmapDescriptorFactory.fromBitmap(getMarkerIconWithLabel(tiempo_llegada, (float)
0, R.mipmap.ic_bus)))
        );

        mMapMonitoreo.moveCamera(CameraUpdateFactory
            .newCameraPosition
                (new CameraPosition.Builder()
                    .target(actualPosicionBus)
                    .zoom(15.5f)
                    .build()));
    } else {
        ValueAnimator valueAnimator = ValueAnimator.ofFloat(0, 1);
        valueAnimator.setDuration(3000);
        valueAnimator.setInterpolator(new LinearInterpolator());
        valueAnimator.addUpdateListener(new ValueAnimator.AnimatorUpdateListener() {
            @Override
            public void onAnimationUpdate(ValueAnimator valueAnimator) {

                startPosition = new LatLng(markerBus.getPosition().latitude,
markerBus.getPosition().longitude);
                endPosition = new LatLng(latitudBus, longitudBus);

                Location loc1 = new Location("");
                loc1.setLatitude(markerBus.getPosition().latitude);
                loc1.setLongitude(markerBus.getPosition().longitude);

                Location loc2 = new Location("");
                loc2.setLatitude(latitudBus);
                loc2.setLongitude(longitudBus);

                float distanceInMeters = loc1.distanceTo(loc2);

                if (distanceInMeters > 5) {
                    v = valueAnimator.getAnimatedFraction();
                    lng = v * endPosition.longitude + (1 - v)
                        * startPosition.longitude;
                    lat = v * endPosition.latitude + (1 - v)
                        * startPosition.latitude;
                    LatLng newPos = new LatLng(lat, lng);
                    markerBus.setPosition(newPos);
                    markerBus.setAnchor(0.5f, 0.5f);
                    markerBus.setRotation(getBearing(startPosition, newPos));
                }
            }
        });
    }
}

```

```

        if (!ultimo_tiempo.equals(tiempo_llegada))

markerBus.setIcon(BitmapDescriptorFactory.fromBitmap(getMarkerIconWithLabel(tiempo_llegada,
(float) getBearing(startPosition, newPos), R.mipmap.ic_bus)));

        if (isVerRutaBus) {
            Log.d("elimina", "calcula ruta");
            verRutaBus();
        }
    } else if (ultimo_tiempo.equals("")) {
        if (!ultimo_tiempo.equals(tiempo_llegada))

markerBus.setIcon(BitmapDescriptorFactory.fromBitmap(getMarkerIconWithLabel(tiempo_llegada,
(float) getBearing(startPosition, actualPosicionBus), R.mipmap.ic_bus)));

    }
}
});
valueAnimator.start();
}
}

public Bitmap getMarkerIconWithLabel(String label, float angle, int ic_bus) {
    IconGenerator iconGenerator = new IconGenerator(getApplicationContext());
    View markerView = LayoutInflater.from(getApplicationContext()).inflate(R.layout.lay_marker, null);
    ImageView imgMarker = markerView.findViewById(R.id.img_marker);
    tvLabel = markerView.findViewById(R.id.tv_label);
    imgMarker.setImageResource(ic_bus);
    imgMarker.setRotation(angle);
    tvLabel.setText(label);
    iconGenerator.setContentView(markerView);
    iconGenerator.setBackground(null);
    return iconGenerator.makeIcon(label);
}

private List<LatLng> decodePoly(String encoded) {
    List<LatLng> poly = new ArrayList<>();
    int index = 0, len = encoded.length();
    int lat = 0, lng = 0;

    while (index < len) {
        int b, shift = 0, result = 0;
        do {
            b = encoded.charAt(index++) - 63;
            result |= (b & 0x1f) << shift;
            shift += 5;
        } while (b >= 0x20);
        int dlat = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
        lat += dlat;

        shift = 0;
        result = 0;
        do {
            b = encoded.charAt(index++) - 63;
            result |= (b & 0x1f) << shift;
            shift += 5;
        } while (b >= 0x20);
        int dlng = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
        lng += dlng;

        LatLng p = new LatLng((((double) lat / 1E5)),
            (((double) lng / 1E5)));
        poly.add(p);
    }

    return poly;
}

```

```

}

private float getBearing(LatLng begin, LatLng end) {
    double lat = Math.abs(begin.latitude - end.latitude);
    double lng = Math.abs(begin.longitude - end.longitude);

    if (begin.latitude < end.latitude && begin.longitude < end.longitude)
        return (float) (Math.toDegrees(Math.atan(lng / lat)));
    else if (begin.latitude >= end.latitude && begin.longitude < end.longitude)
        return (float) ((90 - Math.toDegrees(Math.atan(lng / lat))) + 90);
    else if (begin.latitude >= end.latitude && begin.longitude >= end.longitude)
        return (float) (Math.toDegrees(Math.atan(lng / lat)) + 180);
    else if (begin.latitude < end.latitude && begin.longitude >= end.longitude)
        return (float) ((90 - Math.toDegrees(Math.atan(lng / lat))) + 270);
    return -1;
}

private BitmapDescriptor bitmapDescriptorFromVector(Context context, @DrawableRes int
vectorDrawableResourceId) {
    Drawable background = ContextCompat.getDrawable(context, vectorDrawableResourceId);
    background.setBounds(0, 0, 85, 85);
    Drawable vectorDrawable = ContextCompat.getDrawable(context, vectorDrawableResourceId);
    Bitmap bitmap = Bitmap.createBitmap(85, 85, Bitmap.Config.ARGB_8888);
    Canvas canvas = new Canvas(bitmap);
    background.draw(canvas);
    vectorDrawable.draw(canvas);
    return BitmapDescriptorFactory.fromBitmap(bitmap);
}

@Override
public void onConnected(@Nullable Bundle bundle) {
    if (isLocationPermissionGranted()) {
        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION
!= PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
{
            return;
        } else {
            settingsrequest();
        }
        Location mLastLocation =
LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient);
        if (mLastLocation != null) {
            //FALTA
            latitudActual = String.valueOf(mLastLocation.getLatitude());
            longitudActual = String.valueOf(mLastLocation.getLongitude());

            if (AccessToken.getCurrentAccessToken() != null) {
                String url = "https://graph.facebook.com/" +
AccessToken.getCurrentAccessToken().getUserId() + "/picture?type=small";
                new AsyncTaskLoadImage().execute(url);
            }
        }

        mLocationRequest = new LocationRequest();
        mLocationRequest.setInterval(5000); //5 seconds
        mLocationRequest.setFastestInterval(3000); //3 seconds

        mLocationRequest.setPriority(LocationRequest.PRIORITY_BALANCED_POWER_ACCURACY);

        LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient,
mLocationRequest, this);

    } else {
        manageDeniedPermission();
    }
}

```

```

@Override
public void onConnectionSuspended(int i) {
    Log.d(TAG, "Conexión suspendida");
    mGoogleApiClient.connect();
}

private boolean isLocationPermissionGranted() {
    int permission = ActivityCompat.checkSelfPermission(
        this,
        Manifest.permission.ACCESS_FINE_LOCATION);
    return permission == PackageManager.PERMISSION_GRANTED;
}

private void manageDeniedPermission() {
    if (ActivityCompat.shouldShowRequestPermissionRationale(this,
        Manifest.permission.ACCESS_FINE_LOCATION)) {
        Toast.makeText(this, "Conceder permisos y reiniciar aplicación",
            Toast.LENGTH_LONG).show();
    } else {
        ActivityCompat.requestPermissions(this, new
            String[]{Manifest.permission.ACCESS_FINE_LOCATION}, REQUEST_LOCATION);
    }
}

@Override
public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
    Toast.makeText(this, "Error de conexión con el código:" + connectionResult.getErrorCode(),
        Toast.LENGTH_LONG).show();
}

@Override
public void onResult(@NonNull Status status) {
    if (status.isSuccess()) {
        Log.d(TAG, "Detección de actividad iniciada");
    } else {
        Log.e(TAG, "Error al iniciar/remover la detección de actividad: " + status.getStatusMessage());
    }
}

@Override
public void onLocationChanged(Location location) {
    if (markerInfoIsShow)
        markerParadero.showInfoWindow();

    if (myLocationMarker != null) {
        LatLng myLatLngUpdate = new LatLng(location.getLatitude(), location.getLongitude());

        myLocationMarker.setPosition(myLatLngUpdate);
    } else if (AccessToken.getCurrentAccessToken() != null) {
        String url = "https://graph.facebook.com/" + AccessToken.getCurrentAccessToken().getUserId()
            + "/picture?type=small";
        new AsyncTaskLoadImage().execute(url);
    }

    latitudActual = String.valueOf(location.getLatitude());
    longitudActual = String.valueOf(location.getLongitude());
}

public class AsyncTaskLoadImage extends AsyncTask<String, String, Bitmap> {
    private final static String TAG = "AsyncTaskLoadImage";

    @Override
    protected Bitmap doInBackground(String... params) {
        Bitmap bitmap = null;
        try {

```

```

        URL url = new URL(params[0]);
        bitmap = BitmapFactory.decodeStream((InputStream) url.getContent());
    } catch (IOException e) {
        Log.e(TAG, e.getMessage());
    }
    return bitmap;
}

@Override
protected void onPostExecute(Bitmap bitmap) {
    LatLng latLngOrigen = new LatLng(Double.parseDouble(latitudActual),
    Double.parseDouble(longitudActual));
    myLocationMarker = mMapMonitoreo.addMarker(new MarkerOptions()
        .title("Ubicación")
        .snippet("Usuario")
        .icon(BitmapDescriptorFactory.fromBitmap(createCustomMarker(MapsActivity.this, bitmap,
    "Posición"))))
        .position(latLngOrigen));
}
}

public static Bitmap createCustomMarker(Context context, Bitmap resource, String _name) {

    View marker = ((LayoutInflater)
    context.getSystemService(Context.LAYOUT_INFLATER_SERVICE)).inflate(R.layout.custom_marker_la
    yout, null);

    CircleImageView markerImage = (CircleImageView) marker.findViewById(R.id.user_dp);
    markerImage.setImageBitmap(resource);
    TextView txt_name = (TextView) marker.findViewById(R.id.name);
    txt_name.setText(_name);

    DisplayMetrics displayMetrics = new DisplayMetrics();
    ((Activity) context).getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);
    marker.setLayoutParams(new ViewGroup.LayoutParams(52,
    ViewGroup.LayoutParams.WRAP_CONTENT));
    marker.measure(displayMetrics.widthPixels, displayMetrics.heightPixels);
    marker.layout(0, 0, displayMetrics.widthPixels, displayMetrics.heightPixels);
    marker.buildDrawingCache();
    Bitmap bitmap = Bitmap.createBitmap(marker.getMeasuredWidth(), marker.getMeasuredHeight(),
    Bitmap.Config.ARGB_8888);
    Canvas canvas = new Canvas(bitmap);
    marker.draw(canvas);

    return bitmap;
}

private void verRuta() {
    if (greyPolyLine != null)
        greyPolyLine.remove();
    if (blackPolyline != null)
        blackPolyline.remove();
    String requestUrl = null;
    try {
        requestUrl = "https://maps.googleapis.com/maps/api/directions/json?" +
            "mode=" + modoUsuario + "&"
            + "transit_routing_preference=less_driving&"
            + "origin=" + latitudActual + "," + longitudActual + "&"
            + "destination=" + latitudParadero + "," + longitudParadero + "&"
            + "key=" + getResources().getString(R.string.google_directions_key);
        JSONObjectRequest jsonObjectRequest = new JSONObjectRequest(Request.Method.GET,
        requestUrl, null,
        new Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {
                try {
                    JSONArray jsonArray = response.getJSONArray("routes");

```

```

    for (int i = 0; i < jsonArray.length(); i++) {
        JSONObject route = jsonArray.getJSONObject(i);
        JSONObject poly = route.getJSONObject("overview_polyline");
        String polyline = poly.getString("points");
        polyLineList = decodePoly(polyline);
    }

    polylineOptions = new PolylineOptions();
    polylineOptions.color(Color.GRAY);
    polylineOptions.width(10);
    polylineOptions.startCap(new SquareCap());
    polylineOptions.endCap(new SquareCap());
    polylineOptions.jointType(ROUND);
    polylineOptions.addAll(polyLineList);
    polylineOptions.geodesic(true);
    greyPolyLine = mMapMonitoreo.addPolyline(polylineOptions);

    blackPolylineOptions = new PolylineOptions();
    blackPolylineOptions.width(10);
    if (modoUsuario.equals("driving"))
        blackPolylineOptions.color(Color.BLUE);
    else if (modoUsuario.equals("walking"))
        blackPolylineOptions.color(Color.RED);

    blackPolylineOptions.startCap(new SquareCap());
    blackPolylineOptions.endCap(new SquareCap());
    blackPolylineOptions.jointType(ROUND);
    blackPolylineOptions.geodesic(true);
    blackPolyline = mMapMonitoreo.addPolyline(blackPolylineOptions);

    ValueAnimator polylineAnimator = ValueAnimator.ofInt(0, 100);
    polylineAnimator.setDuration(2000);
    polylineAnimator.setInterpolator(new LinearInterpolator());
    polylineAnimator.addUpdateListener(new ValueAnimator.AnimatorUpdateListener() {
        @Override
        public void onAnimationUpdate(ValueAnimator valueAnimator) {
            List<LatLng> points = greyPolyLine.getPoints();
            int percentValue = (int) valueAnimator.getAnimatedValue();
            int size = points.size();
            int newPoints = (int) (size * (percentValue / 100.0f));
            List<LatLng> p = points.subList(0, newPoints);
            blackPolyline.setPoints(p);
        }
    });
    polylineAnimator.start();
} catch (Exception e) {
    e.printStackTrace();
}

}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
    }
});
RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
requestQueue.add(jsonObjectRequest);
} catch (Exception e) {
    e.printStackTrace();
}
}

private void verRutaBus() {
    Log.d("elimina", "rutabus");
    String requestUrl = null;
    try {
        requestUrl = "https://maps.googleapis.com/maps/api/directions/json?" +

```

```

"mode=driving&"
+ "transit_routing_preference=less_driving&"
+ "origin=" + longitudBus + "," + longitudBus + "&"
+ "destination=" + longitudParadero + "," + longitudParadero + "&"
+ "waypoints=" + waypointsRutaBus + "&"
+ "key=" + getResources().getString(R.string.google_directions_key);
JsonObjectRequest jsonObjectRequest = new JsonObjectRequest(Request.Method.GET,
    requestUrl, null,
    new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {
            try {
                JSONArray jsonArray = response.getJSONArray("routes");
                for (int i = 0; i < jsonArray.length(); i++) {
                    JSONObject route = jsonArray.getJSONObject(i);
                    JSONObject poly = route.getJSONObject("overview_polyline");
                    String polyline = poly.getString("points");
                    polyLineListBus = decodePoly(polyline);
                }

                if (isFirstView) {
                    polylineOptionsBus = new PolylineOptions();
                    polylineOptionsBus.color(Color.GRAY);
                    polylineOptionsBus.width(10);
                    polylineOptionsBus.startCap(new SquareCap());
                    polylineOptionsBus.endCap(new SquareCap());
                    polylineOptionsBus.jointType(ROUND);
                    polylineOptionsBus.addAll(polyLineListBus);
                    polylineOptionsBus.geodesic(true);
                    greyPolyLineBus = mMapMonitoreo.addPolyline(polylineOptionsBus);

                    blackPolylineOptionsBus = new PolylineOptions();
                    blackPolylineOptionsBus.width(10);
                    blackPolylineOptionsBus.color(Color.BLACK);
                    blackPolylineOptionsBus.startCap(new SquareCap());
                    blackPolylineOptionsBus.endCap(new SquareCap());
                    blackPolylineOptionsBus.jointType(ROUND);
                    blackPolylineOptionsBus.geodesic(true);
                    blackPolylineBusLast = mMapMonitoreo.addPolyline(blackPolylineOptionsBus);
                    isLastPolyline = true;

                    ValueAnimator polylineAnimator = ValueAnimator.ofInt(0, 100);
                    polylineAnimator.setDuration(2000);
                    polylineAnimator.setInterpolator(new LinearInterpolator());
                    polylineAnimator.addUpdateListener(new
ValueAnimator.AnimatorUpdateListener() {
                        @Override
                        public void onAnimationUpdate(ValueAnimator valueAnimator) {
                            List<LatLng> points = greyPolyLineBus.getPoints();
                            int percentValue = (int) valueAnimator.getAnimatedValue();
                            int size = points.size();
                            int newPoints = (int) (size * (percentValue / 100.0f));
                            List<LatLng> p = points.subList(0, newPoints);
                            blackPolylineBusLast.setPoints(p);
                        }
                    });
                    polylineAnimator.start();
                    isFirstView = false;
                } else {
                    if (isLastPolyline) {
                        blackPolylineOptionsBus = new PolylineOptions();
                        blackPolylineOptionsBus.width(10);
                        blackPolylineOptionsBus.color(Color.BLACK);
                        blackPolylineOptionsBus.startCap(new SquareCap());
                        blackPolylineOptionsBus.endCap(new SquareCap());
                        blackPolylineOptionsBus.jointType(ROUND);
                        blackPolylineOptionsBus.geodesic(true);
                    }
                }
            }
        }
    });

```

```

        blackPolylineBus = mMapMonitoreo.addPolyline(blackPolylineOptionsBus);
        blackPolylineBus.setPoints(polyLineListBus);
        isLastPolyline = false;
    } else {
        blackPolylineOptionsBus = new PolylineOptions();
        blackPolylineOptionsBus.width(10);
        blackPolylineOptionsBus.color(Color.BLACK);
        blackPolylineOptionsBus.startCap(new SquareCap());
        blackPolylineOptionsBus.endCap(new SquareCap());
        blackPolylineOptionsBus.jointType(ROUND);
        blackPolylineOptionsBus.geodesic(true);
        blackPolylineBusLast =
mMapMonitoreo.addPolyline(blackPolylineOptionsBus);
        blackPolylineBusLast.setPoints(polyLineListBus);
        isLastPolyline = true;
    }

    if (greyPolyLineBus != null)
        greyPolyLineBus.remove();

    }
    } catch (Exception e) {
        e.printStackTrace();
    }
    }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
        }
    });
    RequestQueue requestQueue = Volley.newRequestQueue(getApplicationContext());
    requestQueue.add(jsonObjectRequest);
} catch (Exception e) {
    e.printStackTrace();
}

if (!isLastPolyline && blackPolylineBusLast != null) {
    Log.d("elimina", "elimina aaaa");
    blackPolylineBusLast.remove();
} else if (isLastPolyline && blackPolylineBus != null) {
    Log.d("elimina", "elimina bbbb");
    blackPolylineBus.remove();
}
}
}
}

```

3.1.14 INSTALACIÓN DE LA APLICACIÓN WEB

a. Instalar postgres

Nos conectamos al servidor web vía SSH.

```
[la-MacBook-Pro-de-Cesar:~ cesar$ ssh elcarrodelau@192.168.1.88
elcarrodelau@192.168.1.88's password: 
```

Fig. 58: Instalación paso 1

Iniciamos como usuario root con: *sudo su -*

```
[elcarrodelau@elcarrodelau:~$ sudo su -
[sudo] password for elcarrodelau: 
```

Fig. 59: Instalación paso 2

Instalamos postgres con el comando: *apt-get install postgresql postgresql-contrib*

```
root@elcarrodelau:~# apt-get install postgresql postgresql-contrib
```

Fig. 60: Instalación paso 3

```
root@elcarrodelau:~# apt-get install postgresql postgresql-contrib
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
 libpq5 libsensors4 postgresql-10 postgresql-client-10 postgresql-client-common postgresql-common ssl-cert sysstat
Paquetes sugeridos:
 lm-sensors postgresql-doc locales-all postgresql-doc-10 libjson-perl openssl-blacklist isag
Se instalarán los siguientes paquetes NUEVOS:
 libpq5 libsensors4 postgresql postgresql-10 postgresql-client-10 postgresql-client-common postgresql-common postgresql-contrib
 ssl-cert sysstat
0 actualizados, 10 nuevos se instalarán, 0 para eliminar y 158 no actualizados.
Se necesita descargar 5.496 kB de archivos.
Se utilizarán 21,8 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
```

Fig. 61: Instalación paso 4

Configurar postgres para su inicio en el servidor con el comando: *update-rc.d postgresql enable*

```
root@elcarrodelau:~# update-rc.d postgresql enable
```

Fig. 62: Instalación paso 5

Iniciar postgres con el comando: *service postgresql start*

```
root@elcarrodelau:~# service postgresql start
```

Fig. 63: Instalación paso 6

Verificamos estado de postgres con el comando: *sudo service postgresql status*

```

root@elcarrodelau:~# sudo service postgresql status
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Mon 2019-05-27 20:21:33 UTC; 2min 23s ago
   Main PID: 2948 (code=exited, status=0/SUCCESS)
     Tasks: 0 (limit: 1080)
    Memory: 0B
   CGroup: /system.slice/postgresql.service

may 27 20:21:33 elcarrodelau systemd[1]: Starting PostgreSQL RDBMS...
may 27 20:21:33 elcarrodelau systemd[1]: Started PostgreSQL RDBMS.
root@elcarrodelau:~# █

```

Fig. 64: Instalación paso 7

Salimos del modo root con el comando: *exit*

```

root@elcarrodelau:~# exit

```

Fig. 65: Instalación paso 8

Cambio de contraseña del entorno postgres con el comando: *sudo passwd postgres*

```

elcarrodelau@elcarrodelau:~$ sudo passwd postgres
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
elcarrodelau@elcarrodelau:~$ █

```

Fig. 66: Instalación paso 9

Acceso al entorno postgres con el comando: *su - postgres*

```

elcarrodelau@elcarrodelau:~$ su - postgres
Password:
postgres@elcarrodelau:~$ █

```

Fig. 67: Instalación paso 10

Ingresamos a la base de datos PostgreSQL con el comando: *psql*

```

postgres@elcarrodelau:~$ psql
psql (10.8 (Ubuntu 10.8-0ubuntu0.18.10.1))
Type "help" for help.

postgres=# █

```

Fig. 68: Instalación paso 11

Cambio contraseña usuario postgres con el comando: *ALTER USER postgres WITH PASSWORD 'postgres';*

```

postgres=# ALTER USER postgres WITH PASSWORD 'postgres';
ALTER ROLE
postgres=# █

```

Fig. 69: Instalación paso 12

Salimos de la base de datos PostgreSQL con el comando: `\q`

```
[postgres=# \q
postgres@elcarrodelau:~$
```

Fig. 70: Instalación paso 13

Salimos del entorno postgres: *exit*

```
[postgres@elcarrodelau:~$ exit
logout
elcarrodelau@elcarrodelau:~$
```

Fig. 71: Instalación paso 14

Acceso al directorio de los archivos de configuración del motor de base de datos con el comando: `cd /etc/postgresql/10/main/`

```
[elcarrodelau@elcarrodelau:~$ cd /etc/postgresql/10/main/
elcarrodelau@elcarrodelau:/etc/postgresql/10/main$
```

Fig. 72: Instalación paso 15

Modificamos el archivo `postgresql.conf` modificación del parámetro `listen_addresses = '*'`; con el comando: *sudo nano postgresql.conf*

```
elcarrodelau@elcarrodelau:/etc/postgresql/10/main$ sudo nano postgresql.conf
```

Fig. 73: Instalación paso 16

```
GNU nano 2.9.8 postgresql.conf Modified
# Any parameter can also be given as a command-line option to the server, e.g.,
# "postgres -c log_connections=on". Some parameters can be changed at run time
# with the "SET" SQL command.
#
# Memory units:  kB = kilobytes      Time units:  ms = milliseconds
#                MB = megabytes      s          = seconds
#                GB = gigabytes      min        = minutes
#                TB = terabytes      h          = hours
#                                   d          = days
#
#-----
# FILE LOCATIONS
#-----
# The default values of these variables are driven from the -D command-line
# option or PGDATA environment variable, represented here as ConfigDir.
data_directory = '/var/lib/postgresql/10/main'      # use data in another directory
                                                    # (change requires restart)
hba_file = '/etc/postgresql/10/main/pg_hba.conf'   # host-based authentication file
                                                    # (change requires restart)
ident_file = '/etc/postgresql/10/main/pg_ident.conf' # ident configuration file
                                                    # (change requires restart)
# If external_pid_file is not explicitly set, no extra PID file is written.
external_pid_file = '/var/run/postgresql/10-main.pid' # write an extra PID file
                                                    # (change requires restart)
#-----
# CONNECTIONS AND AUTHENTICATION
#-----
# - Connection Settings -
listen_addresses = '*' # what IP address(es) to listen on;
                        # comma-separated list of addresses;
                        # defaults to 'localhost'; use '*' for all
                        # (change requires restart)
port = 5432            # (change requires restart)
max_connections = 100 # (change requires restart)
#superuser_reserved_connections = 3 # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text      ^J Justify      ^C Cur Pos      M-U Undo        M-A Mark Text
^X Exit          ^R Read File    ^_ Replace      ^U Uncut Text  ^T To Spell    ^G Go To Line   M-E Redo        M-6 Copy Text
```

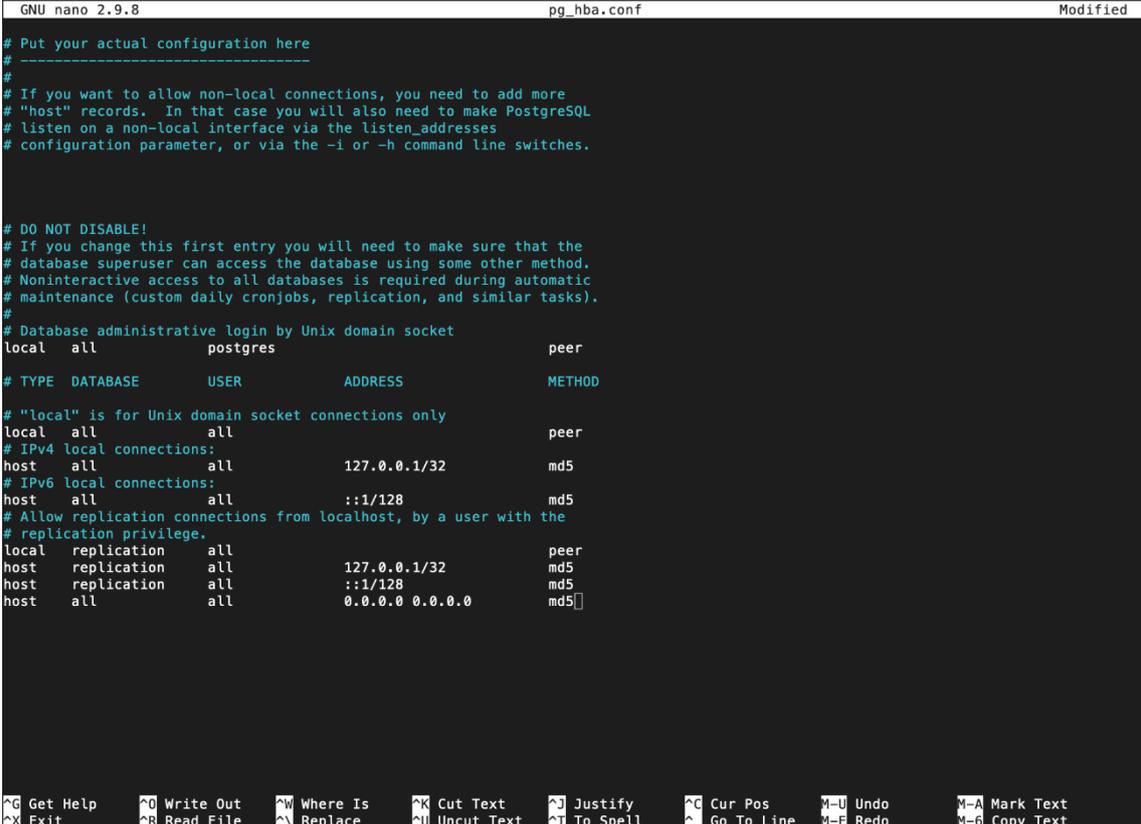
Fig. 74: Instalación paso 17

Modificamos el archivo `pg_hba.conf` con agregar línea de acceso con autenticación md5. Con el comando: `sudo nano pg_hba.conf`

```
elcarrodelau@elcarrodelau:/etc/postgresql/10/main$ sudo nano pg_hba.conf
```

Fig. 75: Instalación paso 18

Agregar línea de acceso con autenticación md5:



```
GNU nano 2.9.8 pg_hba.conf Modified
# Put your actual configuration here
# -----
#
# If you want to allow non-local connections, you need to add more
# "host" records. In that case you will also need to make PostgreSQL
# listen on a non-local interface via the listen_addresses
# configuration parameter, or via the -i or -h command line switches.
#
# DO NOT DISABLE!
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local all postgres peer
# TYPE DATABASE USER ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all peer
# IPv4 local connections:
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all peer
host replication all 127.0.0.1/32 md5
host replication all ::1/128 md5
host all all 0.0.0.0 0.0.0.0 md5
```

Fig. 76: Instalación paso 19

Regresamos a la carpeta raíz con el comando: `cd`

```
elcarrodelau@elcarrodelau:/etc/postgresql/10/main$ cd
```

Fig. 77: Instalación paso 20

Reiniciamos el servicio postgres con el comando: `sudo service postgresql restart`

```
elcarrodelau@elcarrodelau:~$ sudo service postgresql restart
```

Fig. 78: Instalación paso 21

Instalar el servicio del administrador de versiones de Node JS. Con el comando: `sudo curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.31.7/install.sh | bash`

```

elcarrodelau@elcarrodelau:~$ sudo curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.31.7/install.sh | bash
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 10007  100 10007    0     0  11081    0 --:--:-- --:--:-- 11081
=> Downloading nvm from git to '/home/elcarrodelau/.nvm'
=> Cloning into '/home/elcarrodelau/.nvm'...
remote: Enumerating objects: 2, done.
remote: Counting objects: 100% (2/2), done.
remote: Compressing objects: 100% (2/2), done.
Receiving objects: 100% (7495/7495), 2.41 MiB | 8.00 KiB/s, done.
remote: Total 7495 (delta 0), reused 1 (delta 0), pack-reused 7493
Resolving deltas: 100% (4737/4737), done.
* (HEAD detached at v0.31.7)
  master

=> Appending source string to /home/elcarrodelau/.bashrc
=> Close and reopen your terminal to start using nvm or run the following to use it now:

export NVM_DIR="/home/elcarrodelau/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && . "$NVM_DIR/nvm.sh" # This loads nvm

```

Fig. 79: Instalación paso 22

Verificamos la correcta instalación del manager de versiones de Node JS.: ***nvm --version***

```

elcarrodelau@elcarrodelau:~$ nvm --version
0.31.7

```

Fig. 80: Instalación paso 23

Instalar Node JS v8.9.1: ***nvm install v8.9.1***

```

elcarrodelau@elcarrodelau:~$ nvm install v8.9.1
Downloading https://nodejs.org/dist/v8.9.1/node-v8.9.1-linux-x64.tar.xz...
##### 100.0%
Now using node v8.9.1 (npm v5.5.1)
Creating default alias: default -> v8.9.1

```

Fig. 81: Instalación paso 24

Usar la versión instalada de Node JS: ***nvm use v8.9.1***

```

elcarrodelau@elcarrodelau:~$ nvm use v8.9.1
Now using node v8.9.1 (npm v5.5.1)

```

Fig. 82: Instalación paso 25

Instalar framework SailsJS: ***npm install sails***

```

elcarrodelau@elcarrodelau:~$ npm install sails
npm WARN saveError ENOENT: no such file or directory, open '/home/elcarrodelau/package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open '/home/elcarrodelau/package.json'
npm WARN elcarrodelau No description
npm WARN elcarrodelau No repository field.
npm WARN elcarrodelau No README data
npm WARN elcarrodelau No license field.

+ sails@1.2.2
added 205 packages in 51.401s

```

Fig. 83: Instalación paso 26

Verificamos instalación de sails: ***sails -v***

```

elcarrodelau@elcarrodelau:~$ sails -v
1.2.2

```

Fig. 84: Instalación paso 27

Instalar pm2 (Production process manager for Node.js): ***npm install pm2***

```

elcarrodelau@elcarrodelau:~$ npm install pm2
npm WARN saveError ENOENT: no such file or directory, open '/home/elcarrodelau/package.json'
npm WARN enoent ENOENT: no such file or directory, open '/home/elcarrodelau/package.json'
npm WARN elcarrodelau No description
npm WARN elcarrodelau No repository field.
npm WARN elcarrodelau No README data
npm WARN elcarrodelau No license field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.9 (node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.9: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
+ pm2@3.5.1
added 358 packages in 54.355s

```

Fig. 85: Instalación paso 28

Copiamos el Proyecto y la base de datos con los datos inicial al servidor por medio de los comandos:

```
scp /Users/cesar/Desktop/elcarrodelau.sql elcarrodelau@192.168.1.88:~
```

```
scp /Users/cesar/Desktop/elcarrodelau.zip elcarrodelau@192.168.1.88:~
```

```

la-MacBook-Pro-de-Cesar:~ cesar$ scp /Users/cesar/Desktop/elcarrodelau.sql elcarrodelau@192.168.1.88:~
elcarrodelau@192.168.1.88's password:
elcarrodelau.sql                                100% 2118KB  77.3MB/s   00:00
la-MacBook-Pro-de-Cesar:~ cesar$ scp /Users/cesar/Desktop/elcarrodelau.zip elcarrodelau@192.168.1.88:~
elcarrodelau@192.168.1.88's password:
elcarrodelau.zip                                100%  28MB  41.3MB/s   00:00
la-MacBook-Pro-de-Cesar:~ cesar$

```

Fig. 86: Instalación paso 29

Crear usuario para la base de datos

Ingresamos como usuario root con el comando: `su - postgres`

```

elcarrodelau@elcarrodelau:~$ su - postgres
Password:
postgres@elcarrodelau:~$

```

Fig. 87: Instalación paso 30

Ingresamos a la base de datos PostgreSQL con el comando: `psql`

```

postgres@elcarrodelau:~$ psql
psql (10.8 (Ubuntu 10.8-0ubuntu0.18.10.1))
Type "help" for help.

postgres=#

```

Fig. 88: Instalación paso 31

Creamos el usuario con el comando: **`CREATE ROLE elcarrodelau LOGIN ENCRYPTED PASSWORD 'md51f9603726d7830f091f8575f74782e6b' NOSUPERUSER INHERIT NOCREATEDB NOCREATEROLE NOREPLICATION VALID UNTIL '2100-12-31 00:00:00';`**

```

postgres=# CREATE ROLE elcarrodelau LOGIN ENCRYPTED PASSWORD 'md51f9603726d7830f091f8575f74782e6b'
NOSUPERUSER INHERIT NOCREATEDB NOCREATEROLE NOREPLICATION VALID UNTIL '2100-12-31 00:00:00';
CREATE ROLE

```

Fig. 89: Instalación paso 32

Creamos la base de datos con el comando: `createdb -O elcarrodelau ElCarroDeLaUdb`

```

postgres@elcarrodelau:~$ createdb -O elcarrodelau ElCarroDeLaUdb

```

Fig. 90: Instalación paso 33

Ingresamos a la base de datos PostgreSQL con el comando: *psql*

```
postgres@elcarrodelau:~$ psql
psql (10.8 (Ubuntu 10.8-0ubuntu0.18.10.1))
Type "help" for help.

postgres=#
```

Fig. 91: Instalación paso 34

Verificamos la creación de la base de datos con los comandos: *\l*

```
postgres=# \l
          List of databases
-----+-----+-----+-----+-----+-----+
 Name      | Owner   | Encoding | Collate | Ctype   | Access privileges
-----+-----+-----+-----+-----+-----+
 ElCarroDeLaUdb | elcarrodelau | UTF8     | es_ES.UTF-8 | es_ES.UTF-8 |
 postgres     | postgres   | UTF8     | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres      +
 template0    | postgres   | UTF8     | es_ES.UTF-8 | es_ES.UTF-8 | postgres=CTc/postgres +
 template1    | postgres   | UTF8     | es_ES.UTF-8 | es_ES.UTF-8 | =c/postgres      +
             |           |           |           |           | postgres=CTc/postgres
(4 rows)
```

Fig. 92: Instalación paso 35

Restauramos la base de datos con la data inicial del proyecto

Ingresamos a la instancia postgres como usuario root con el comando: *su - postgres*

```
elcarrodelau@elcarrodelau:~$ su - postgres
Password:
postgres@elcarrodelau:~$
```

Fig. 93: Instalación paso 36

Restauramos la base de datos con el comando: *psql -f /home/elcarrodelau/elcarrodelau.sql ElCarroDeLaUdb*

```
postgres@elcarrodelau:~$ psql -f /home/elcarrodelau/elcarrodelau.sql ElCarroDeLaUdb
SET
SET
SET
SET
set_config
-----
(1 row)

SET
SET
SET
CREATE EXTENSION
COMMENT
CREATE EXTENSION
COMMENT
CREATE FUNCTION
ALTER FUNCTION
CREATE FUNCTION
ALTER FUNCTION
CREATE FUNCTION
ALTER FUNCTION
```

Fig. 94: Instalación paso 37

Descomprimos el proyecto con el comando: *unzip elcarrodelau.zip*

```

elcarrodelau@elcarrodelau:~$ unzip elcarrodelau.zip
Archive:  elcarrodelau.zip
  creating:  monitoreo/
  inflating: monitoreo/.DS_Store
  inflating: monitoreo/.editorconfig
  creating:  monitoreo/.git/
  extracting: monitoreo/.git/COMMIT_EDITMSG
  inflating: monitoreo/.git/FETCH_HEAD
  extracting: monitoreo/.git/HEAD
  extracting: monitoreo/.git/ORIG_HEAD
  creating:  monitoreo/.git/branches/
  inflating: monitoreo/.git/config
  inflating: monitoreo/.git/description
  creating:  monitoreo/.git/hooks/
  inflating: monitoreo/.git/hooks/applypatch-msg.sample
  inflating: monitoreo/.git/hooks/commit-msg.sample
  inflating: monitoreo/.git/hooks/fsmonitor-watchman.sample
  inflating: monitoreo/.git/hooks/post-update.sample
  inflating: monitoreo/.git/hooks/pre-applypatch.sample
  inflating: monitoreo/.git/hooks/pre-commit.sample
  inflating: monitoreo/.git/hooks/pre-push.sample
  inflating: monitoreo/.git/hooks/pre-rebase.sample
  inflating: monitoreo/.git/hooks/pre-receive.sample

```

Fig. 95: Instalación paso 38

Navegamos dentro de la carpeta del proyecto e iniciamos el proyecto con el comando:

pm2 start app.js

```

elcarrodelau@elcarrodelau:~/monitoreo$ pm2 start app.js
[PM2] Starting /home/elcarrodelau/monitoreo/app.js in fork_mode (1 instance)
[PM2] Done.

```

Name	id	mode	status	u	cpu	memory
app	0	fork	online	0	0%	29.3 MB

Use `pm2 show <id|name>` to get more details about an app

Fig. 96: Instalación paso 39

3.1.15 EVALUACIÓN Y PRUEBAS

3.1.15.1 EVALUACIÓN

En esta fase de desarrollo de la aplicación web se va a evaluar tanto el funcionamiento como la usabilidad, pasando por la compatibilidad con los distintos navegadores.

Existen una serie de pautas que se deben cumplir para que el nivel de usabilidad de la aplicación sea alto:

- Debemos ponernos en lugar del usuario y tener en cuenta en qué modo deseáramos encontrar la información.
- Pensar en las posibles limitaciones que el usuario a nivel de procesamiento, ancho de banda o resolución pudiera tener.
- Diseñar el grado de profundidad que daremos a nuestros contenidos y definir un árbol de navegación atractivo sin excesivos enlaces.
- Obtener un diseño visual con importante carga semántica, que transmita lo que uno propone y que cree un impacto visual sin que produzca confusión.

Con estas recomendaciones se ha tratado de realizar la aplicación web, existen muchas recomendaciones para crear un nivel alto de usabilidad, pero a mi parecer, estas son las más importantes.

Al seguir estas recomendaciones desde el principio del proyecto la usabilidad se ha ido incorporando de un modo interactivo, iterativo e incremental. En el resultado final se nota que se ha buscado una interface sencilla, agradable y cómoda para que el usuario final se desenvuelva con soltura intuitivamente.

3.1.15.2 PRUEBAS

A medida que he ido creando la aplicación web se ha ido realizando pruebas para comprobar el rendimiento, para ello se utilizó un programa llamado Apache JMeter, que permite realizar pruebas de carga, rendimiento y estrés.

3.1.15.2.1 PRUEBAS DE CARGA

Una prueba de carga se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada. Esta carga puede ser el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dura la carga. Esta prueba puede mostrar los tiempos de respuesta de todas las transacciones importantes de la aplicación. Si la base de datos, el servidor de aplicaciones, etc.. también se monitorizan, entonces esta prueba puede mostrar el cuello de botella en la aplicación.

La prueba realizada en este caso es que la aplicación web pueda responder a 1000 usuarios conectados al mismo tiempo sin errores, obteniendo el siguiente resultado.

Tabla 42: Tabla resumen prueba de carga

Etiqueta	# Muestras	Media	Min	Max	Desv. Estandar	% Error	Rendimiento	Kb/sec	Media de Bytes
HTTP Request	20000	6694	0	71915	4439.74	11.81%	41.3	671.83	16664.3
Total	20000	6694	0	71915	4439.74	11.81%	41.3	671.83	16664.3

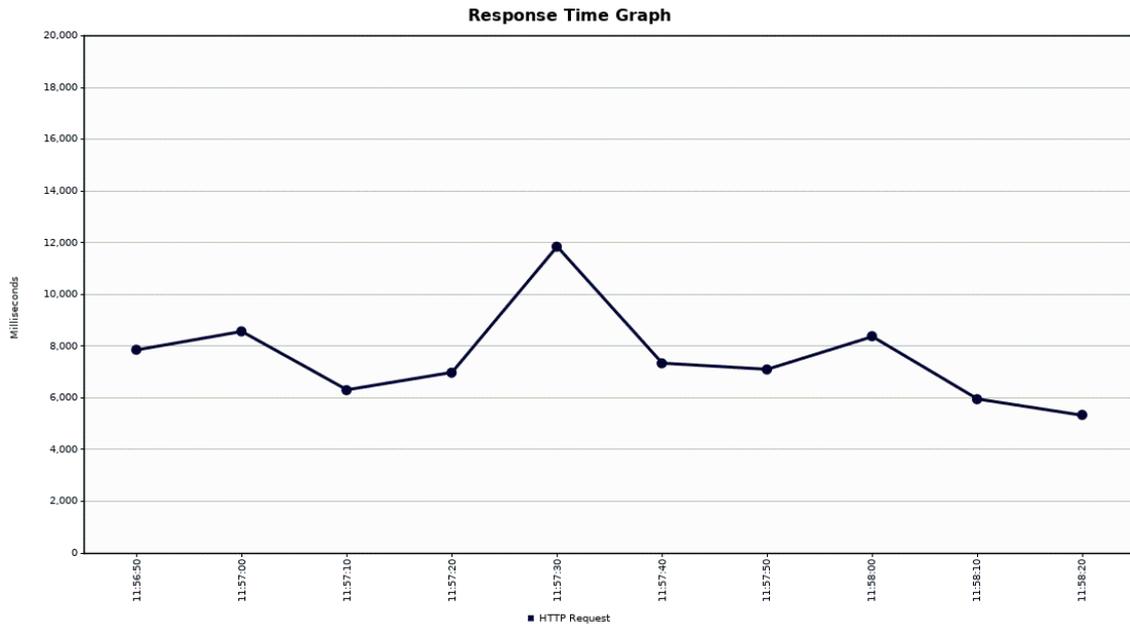


Fig. 97: Grafico resumen prueba de carga

3.1.15.2.2 PRUEBAS DE RENDIMIENTO

Son las pruebas que se realizan, desde una perspectiva, para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo.

La prueba realizada en este caso es para medir la diferencia en la latencia media entre 100 500 y 1000 usuarios conectados al mismo tiempo, obteniendo los siguientes resultados:

➤ 100 USUARIOS

Tabla 43: Tabla resumen prueba de rendimiento 100 usuarios

Etiqueta	# Muestras	Media	Min	Max	Desv. Estandar	% Error	Rendimiento	Kb/sec	Media de Bytes
HTTP Request	1000	725	258	1126	126.19	0.00%	119	2158.7	18570.5
Total	1000	725	258	1126	126.19	0.00%	119	2158.7	18570.5

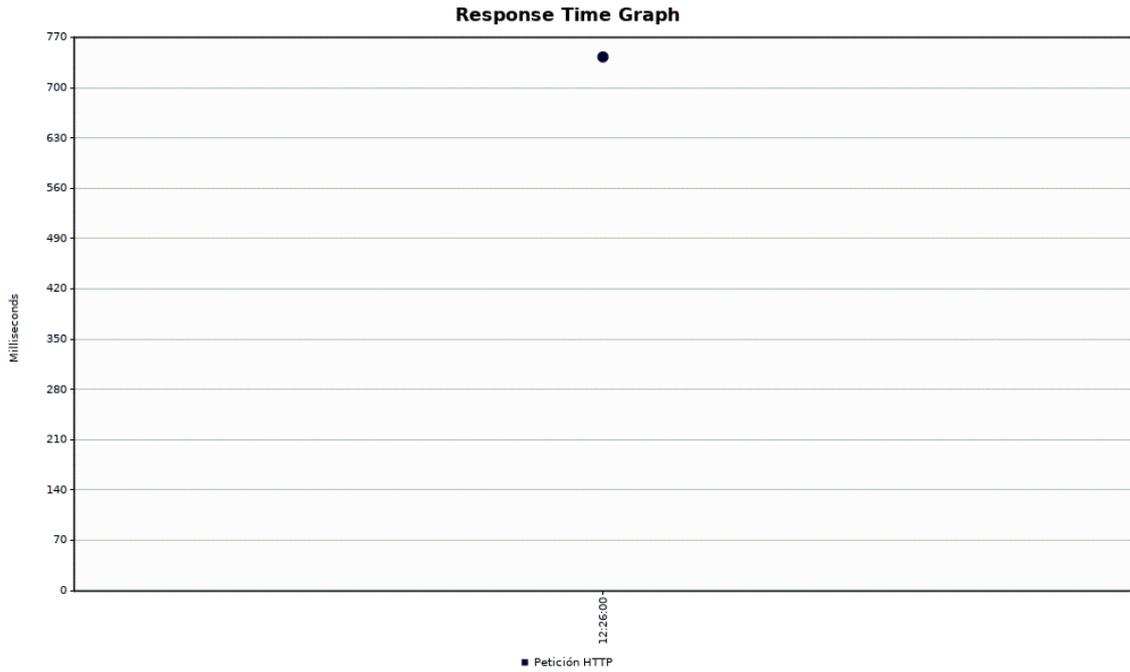


Fig. 98: Grafico resumen prueba de rendimiento 100 usuarios

➤ 500 USUARIOS

Tabla 44: Tabla resumen prueba de rendimiento 500 usuarios

Etiqueta	# Muestras	Media	Min	Max	Desv. Estandar	% Error	Rendimiento	Kb/sec	Media de Bytes
HTTP Request	5000	3359	12	5584	1135.12	0.00%	120.3	2181.42	18570.7
Total	5000	3359	12	5584	1135.12	0.00%	120.3	2181.42	18570.7

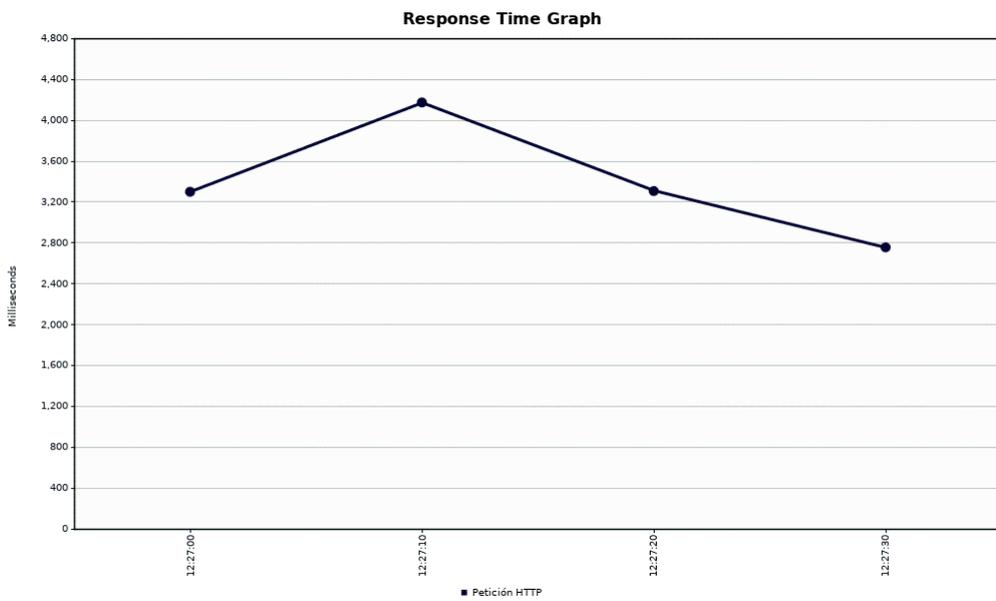


Fig. 99: Grafico resumen prueba de rendimiento 500 usuarios

➤ 1000 USUARIOS

Tabla 45: Tabla resumen prueba de rendimiento 1000 usuarios

Etiqueta	# Muestras	Media	Min	Max	Desv. Estandar	% Error	Rendimiento	Kb/sec	Media de Bytes
HTTP Request	20000	6694	0	71915	4439.74	11.81%	41.3	671.83	16664.3
Total	20000	6694	0	71915	4439.74	11.81%	41.3	671.83	16664.3

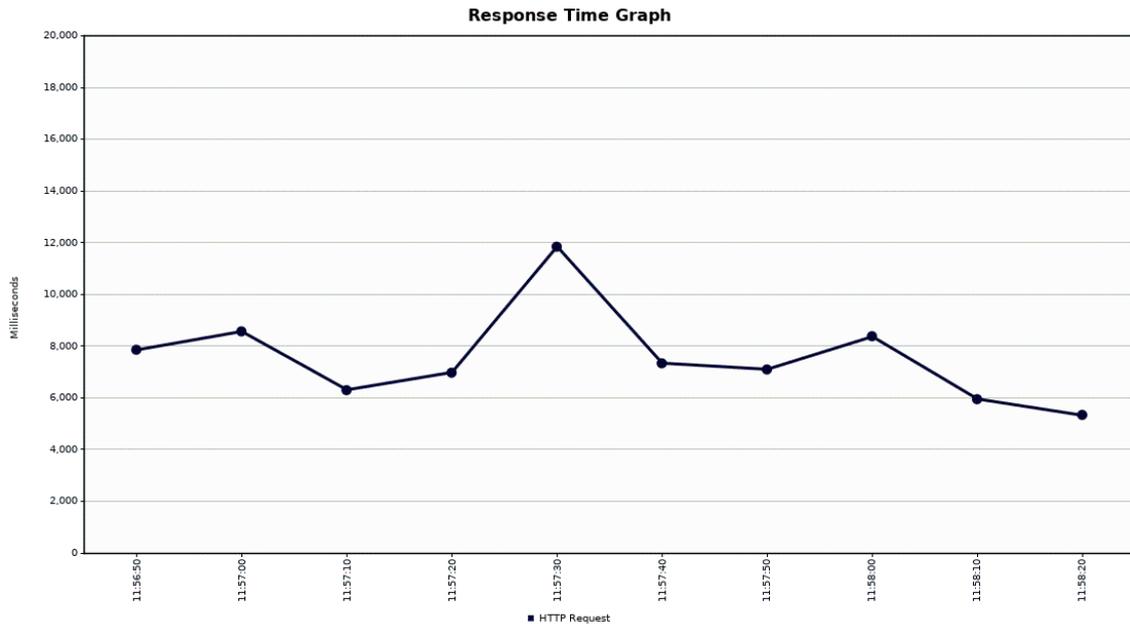


Fig. 100: Grafico resumen prueba de rendimiento 1000 usuarios

3.1.15.2.3 PRUEBAS DE ESTRÉS

Esta prueba se utiliza normalmente para romper la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

La prueba realizada en este caso es para medir cuantos usuarios soporta la aplicación web hasta que ya empieza a devolver errores, obteniendo como resultado que a partir de 6000 usuarios conectados en intervalo de 1 minutos la aplicación empieza a ralentizarse provocando que el tiempo de respuesta por parte del servidor incremente, como se muestra en los siguientes gráficos:

Tabla 46: Tabla resumen prueba de stress

Etiqueta	# Muestras	Media	Min	Max	Desv. Estandar	% Error	Rendimiento	Kb/sec	Media de Bytes
HTTP Request	19824	17970	0	132557	30007.11	49.93%	92.9	942.17	10383
Total	19824	17970	0	132557	30007.11	49.93%	92.9	942.17	10383

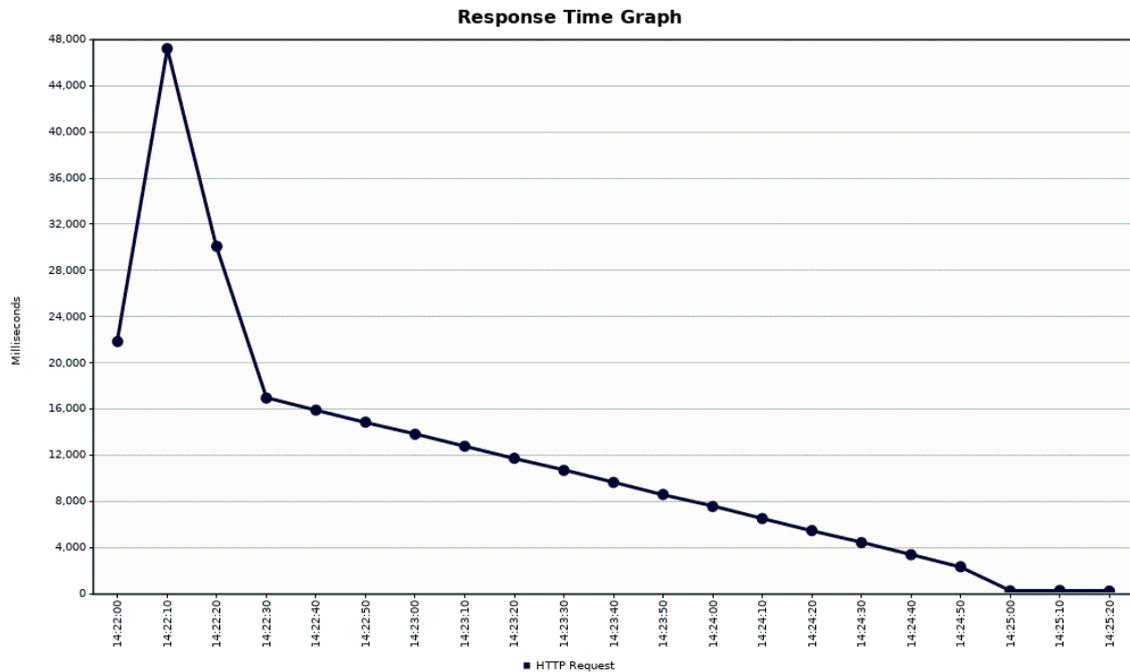


Fig. 101: Grafico resumen prueba de stress

3.1.15.3 MEDICIÓN DEL TRAFICO

Google Maps recopila datos acerca de sus usuarios, incluida su localización en tiempo real. Lo hace incluso cuando no has activado esta opción, la aplicación práctica es, entre otras, la de crear un modelo dinámico y en tiempo real del tráfico en Google Maps. La aplicación cruza los datos de ubicación de los usuarios de Google con la velocidad a la que se desplazan y el mapa de carreteras. Con estos datos consigue saber cómo de intenso es el tráfico en una zona.

No sólo los dispositivos Android ayudan a Google Maps a saber dónde hay un atasco, también los iPhone, que comparten su ubicación con Google desde esta aplicación. Actualmente, Maps es quizás la forma más precisa de saber el estado del tráfico en las carreteras, pues su casi monopolio del sector hace que casi todo el mundo la use para viajar.

3.2 TRATAMIENTO Y ANÁLISIS DE DATOS Y PRESENTACIÓN DE RESULTADOS

3.2.1 RESULTADO DE LA INVESTIGACIÓN

A continuación, se presentan los resultados de la investigación en base a la información recogida mediante las técnicas e instrumentos de recolección de datos, las que se muestran mediante cuadros y gráficos estadísticos de acuerdo a la hipótesis de trabajo. Para evaluar la variable independiente se aplicó dos encuestas a cada uno de los usuarios de los buses universitarios, en total fueron 147 los encuestados, obteniendo el siguiente resultado:

3.2.1.1 RESULTADOS DE LA ENCUESTA PARA EL INDICADOR TIEMPO

Con las encuestas aplicadas también se ha realizado un resumen de tiempo separado por pregunta, estos tiempos son del pre-test (Sin la utilización de la aplicación web de control y monitoreo en tiempo real) y post-test (utilizando de la aplicación web de control y monitoreo en tiempo real), como podemos observar en los siguientes cuadros y gráficos:

En el ANEXO 2 podemos ver el modelo de encuesta y en el ANEXO 3 se encuentran los valores del PRE TEST y en el ANEXO 4 los valores del POST TEST.

Tabla 47: Resultados de la encuesta indicador tiempo pre - test y post - test

RESULTADOS DE LA ENCUESTA			
PREGUNTA		Minutos	
		PRE - TEST	POST - TEST
3	¿Usted cuántos minutos espera normalmente para abordar el bus?	4.761	3.775
4	¿Usted cuánto tiempo tarda normalmente en llegar a su destino final?	1.809	1.312

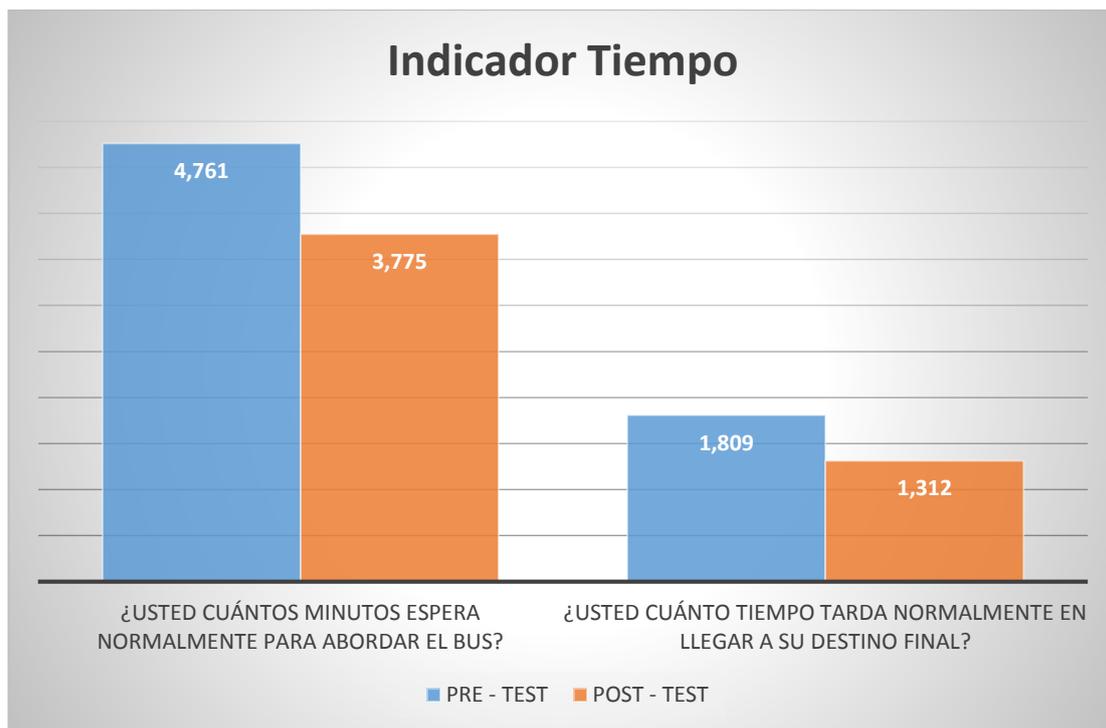


Fig. 102: Tiempos de espera de los usuarios por los buses universitarios

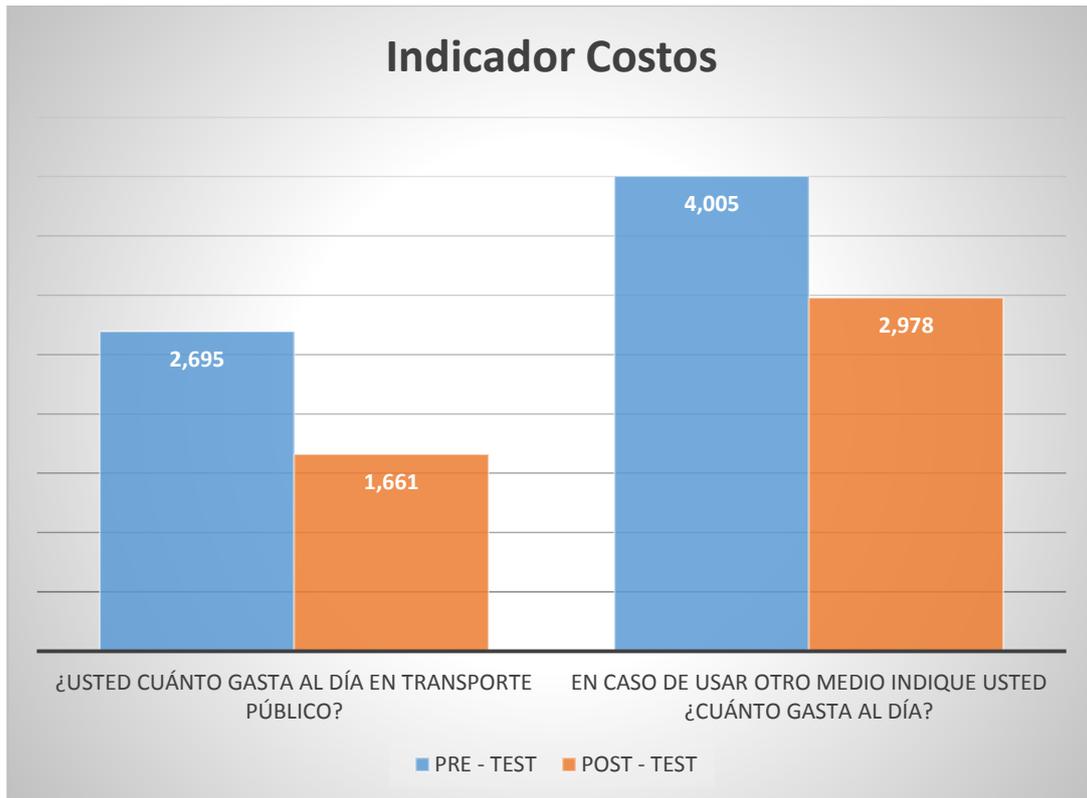
Como podemos observar se ha logrado optimizar los tiempos de espera de los usuarios por los buses universitarios, esto beneficiará enormemente a los usuarios de los buses universitarios de la Universidad Nacional de Cajamarca.

3.2.1.2 RESULTADOS DE LA ENCUESTA PARA EL INDICADOR COSTOS

En el ANEXO 2 podemos ver el modelo de encuesta y en el ANEXO 3 se encuentran los valores del PRE TEST y en el ANEXO 4 los valores del POST TEST tomados en la encuesta para identificar la mejora en los gastos de los usuarios de los buses de la Universidad Nacional de Cajamarca.

Tabla 48: Resultados de la encuesta indicador costos pre - test y post - test

RESULTADOS DE LA ENCUESTA			
PREGUNTA		Soles	
		PRE - TEST	POST - TEST
5	¿Usted cuánto gasta al día en transporte público?	2.695	1.661
6	En caso de usar otro medio indique usted ¿Cuánto gasta al día?	4.005	2.978



Como podemos observar se ha logrado reducir los gastos en servicios de transporte públicos alternos (pagados) a los buses de la Universidad Nacional de Cajamarca (gratuitos) por parte de los usuarios de los buses universitarios, esto beneficiará enormemente a los usuarios de los buses universitarios de la Universidad Nacional de Cajamarca.

CAPÍTULO IV ANÁLISIS Y DISCUSIÓN DE RESULTADOS

4.1 ANÁLISIS DE RESULTADOS

4.1.1 INDICADOR: TIEMPO PROMEDIO DE ESPERA DE LA COMUNIDAD UNIVERSITARIA POR EL BUS UNIVERSITARIO

4.1.1.1 CONTRASTACIÓN DE LA HIPÓTESIS

Para la contrastación de la hipótesis planteada se va a utilizar la prueba T De Student para datos pareados o muestras relacionadas [31], el cual corresponde al diseño de investigación realizado, es decir, esta prueba estadística exige dependencias entre ambas muestras, en las que hay dos momentos uno antes y otro después. Con el ello se da a entender que, en el primer periodo, las observaciones servirán de control o testigo, para conocer los cambios que se suscitan después de aplicar la variable independiente. Con la prueba t se comparan las medias y las desviaciones estándar de grupo de datos y se determina si entre esos parámetros las diferencias son estadísticamente significativas o si sólo son diferencias aleatorias.

En probabilidad y estadística, la distribución t (de Student) es una distribución de probabilidad que surge del problema de estimar la media de una población normalmente distribuida cuando el tamaño de la muestra es pequeño. Aparece de manera natural al realizar la prueba t de Student para la determinación de las diferencias entre dos medias muestrales y para la construcción del intervalo de confianza para la diferencia entre las medias de dos poblaciones cuando se desconoce la desviación típica de una población y ésta debe ser estimada a partir de los datos de una muestra.

Utilizaremos la siguiente fórmula para muestras relacionadas:

$$t = \frac{\bar{d}}{\frac{\sigma_d}{\sqrt{N}}}$$

Donde:

t: Valor estadístico del procedimiento.

\bar{d} : Valor promedio o media aritmética de las diferencias entre los momentos después y antes.

$\sigma \partial$: Desviación estándar de las diferencias entre los momentos después y antes.

N: Tamaño de la muestra.

La media aritmética de las diferencias se obtiene de la siguiente manera:

$$\bar{\partial} = \frac{\sum \partial}{N}$$

La desviación estándar de las diferencias se obtiene como sigue:

$$\sigma \partial = \sqrt{\frac{\sum(\partial - \bar{\partial})^2}{N - 1}}$$

Trabajaremos con la tabla resumen de los datos obtenidos de la evaluación: En el ANEXO 3 se encuentran los valores del PRE TEST y en el ANEXO 4 los valores del POST TEST tomados como prueba.

Tabla 49: Resumen costos pre test y post test

TABLA RESUMEN EN SOLES						
Nº	Persona	PRE-TEST	POST-TEST	$\partial = (X_D - X_A)$	$(\partial - \bar{\partial})$	$(\partial - \bar{\partial})^2$
		X_A	X_D			
1	P1	3	1	-2	0.06	0.00
2	P2	2	0.5	-1.5	0.56	0.31
3	P3	4	1.5	-2.5	-0.44	0.19
4	P4	3	1	-2	0.06	0.00
5	P5	10.5	7.5	-3	-0.94	0.88
6	P6	8	6	-2	0.06	0.00
7	P7	8	5.5	-2.5	-0.44	0.19
8	P8	7	4	-3	-0.94	0.88
9	P9	3	0.5	-2.5	-0.44	0.19
10	P10	2	1	-1	1.06	1.13
11	P11	10	8.5	-1.5	0.56	0.31
12	P12	14	12.5	-1.5	0.56	0.31
13	P13	11	7	-4	-1.94	3.76
14	P14	30	28.5	-1.5	0.56	0.31
15	P15	5	3.5	-1.5	0.56	0.31
16	P16	10	7.5	-2.5	-0.44	0.19
17	P17	2	0.5	-1.5	0.56	0.31
18	P18	12	10.5	-1.5	0.56	0.31
19	P19	4	2.5	-1.5	0.56	0.31

20	P20	1	0	-1	1.06	1.13
21	P21	4	2	-2	0.06	0.00
22	P22	13	9	-4	-1.94	3.76
23	P23	8	5.5	-2.5	-0.44	0.19
24	P24	6	5	-1	1.06	1.13
25	P25	4	1.5	-2.5	-0.44	0.19
26	P26	5.5	3	-2.5	-0.44	0.19
27	P27	8	5	-3	-0.94	0.88
28	P28	4	1.5	-2.5	-0.44	0.19
29	P29	3	0.5	-2.5	-0.44	0.19
30	P30	5	2	-3	-0.94	0.88
31	P31	6	3	-3	-0.94	0.88
32	P32	6	5	-1	1.06	1.13
33	P33	7	4.5	-2.5	-0.44	0.19
34	P34	7	6	-1	1.06	1.13
35	P35	12	9	-3	-0.94	0.88
36	P36	9	6	-3	-0.94	0.88
37	P37	4.7	3.5	-1.2	0.86	0.74
38	P38	3	2	-1	1.06	1.13
39	P39	10	9	-1	1.06	1.13
40	P40	12	10	-2	0.06	0.00
41	P41	4	1	-3	-0.94	0.88
42	P42	9	7	-2	0.06	0.00
43	P43	5	3	-2	0.06	0.00
44	P44	5	2.5	-2.5	-0.44	0.19
45	P45	8	5	-3	-0.94	0.88
46	P46	2.8	1.4	-1.4	0.66	0.44
47	P47	10	7	-3	-0.94	0.88
48	P48	8	5	-3	-0.94	0.88
49	P49	4	2.5	-1.5	0.56	0.31
50	P50	2	0.5	-1.5	0.56	0.31
51	P51	6.8	5.3	-1.5	0.56	0.31
52	P52	4	1.5	-2.5	-0.44	0.19
53	P53	3	1.5	-1.5	0.56	0.31
54	P54	3.5	2	-1.5	0.56	0.31
55	P55	2.5	0.5	-2	0.06	0.00
56	P56	1	0	-1	1.06	1.13
57	P57	8	5	-3	-0.94	0.88
58	P58	6	3.5	-2.5	-0.44	0.19
59	P59	8	5	-3	-0.94	0.88
60	P60	4	1.5	-2.5	-0.44	0.19
61	P61	4	1	-3	-0.94	0.88
62	P62	2	0	-2	0.06	0.00
63	P63	6	4.5	-1.5	0.56	0.31
64	P64	4	3	-1	1.06	1.13
65	P65	3	1.5	-1.5	0.56	0.31
66	P66	16	13	-3	-0.94	0.88
67	P67	4	1.5	-2.5	-0.44	0.19

68	P68	12	10.5	-1.5	0.56	0.31
69	P69	6	3.5	-2.5	-0.44	0.19
70	P70	8	7	-1	1.06	1.13
71	P71	2	1	-1	1.06	1.13
72	P72	2.8	1.4	-1.4	0.66	0.44
73	P73	8	5	-3	-0.94	0.88
74	P74	7	4	-3	-0.94	0.88
75	P75	4.5	2	-2.5	-0.44	0.19
76	P76	7	5.5	-1.5	0.56	0.31
77	P77	0	0	0	2.06	4.25
78	P78	3	0	-3	-0.94	0.88
79	P79	8	5.5	-2.5	-0.44	0.19
80	P80	15	12	-3	-0.94	0.88
81	P81	6	4.5	-1.5	0.56	0.31
82	P82	0	0	0	2.06	4.25
83	P83	9	6.5	-2.5	-0.44	0.19
84	P84	9	6	-3	-0.94	0.88
85	P85	1.5	0.5	-1	1.06	1.13
86	P86	8	6.5	-1.5	0.56	0.31
87	P87	8	6.5	-1.5	0.56	0.31
88	P88	7	5.5	-1.5	0.56	0.31
89	P89	7	5.5	-1.5	0.56	0.31
90	P90	2	0.5	-1.5	0.56	0.31
91	P91	8	6	-2	0.06	0.00
92	P92	6	4.5	-1.5	0.56	0.31
93	P93	4	2.5	-1.5	0.56	0.31
94	P94	8	5.5	-2.5	-0.44	0.19
95	P95	0	0	0	2.06	4.25
96	P96	7	4	-3	-0.94	0.88
97	P97	1	0	-1	1.06	1.13
98	P98	2	0	-2	0.06	0.00
99	P99	7	4.5	-2.5	-0.44	0.19
100	P100	12.5	11	-1.5	0.56	0.31
101	P101	5	2	-3	-0.94	0.88
102	P102	2.5	1.5	-1	1.06	1.13
103	P103	2	0	-2	0.06	0.00
104	P104	3	1.5	-1.5	0.56	0.31
105	P105	2.5	1	-1.5	0.56	0.31
106	P106	4	1	-3	-0.94	0.88
107	P107	12	8	-4	-1.94	3.76
108	P108	21	17	-4	-1.94	3.76
109	P109	11	9.5	-1.5	0.56	0.31
110	P110	30	27.5	-2.5	-0.44	0.19
111	P111	3	1.5	-1.5	0.56	0.31
112	P112	3	1	-2	0.06	0.00
113	P113	11	8	-3	-0.94	0.88
114	P114	10	7	-3	-0.94	0.88
115	P115	4	1.5	-2.5	-0.44	0.19

116	P116	10	8.5	-1.5	0.56	0.31
117	P117	5	2	-3	-0.94	0.88
118	P118	8	5.5	-2.5	-0.44	0.19
119	P119	8	6.5	-1.5	0.56	0.31
120	P120	12	10.5	-1.5	0.56	0.31
121	P121	3.5	2	-1.5	0.56	0.31
122	P122	12	10	-2	0.06	0.00
123	P123	10	7.5	-2.5	-0.44	0.19
124	P124	20	17.5	-2.5	-0.44	0.19
125	P125	4	2.5	-1.5	0.56	0.31
126	P126	3	1.5	-1.5	0.56	0.31
127	P127	3	1.5	-1.5	0.56	0.31
128	P128	2	0.5	-1.5	0.56	0.31
129	P129	3	1	-2	0.06	0.00
130	P130	4	2.5	-1.5	0.56	0.31
131	P131	6	3.5	-2.5	-0.44	0.19
132	P132	10	8.5	-1.5	0.56	0.31
133	P133	1	0	-1	1.06	1.13
134	P134	4	1.5	-2.5	-0.44	0.19
135	P135	3	1.5	-1.5	0.56	0.31
136	P136	4	2.5	-1.5	0.56	0.31
137	P137	6	3	-3	-0.94	0.88
138	P138	7	3	-4	-1.94	3.76
139	P139	4	2.5	-1.5	0.56	0.31
140	P140	7	4	-3	-0.94	0.88
141	P141	20	17	-3	-0.94	0.88
142	P142	3	1	-2	0.06	0.00
143	P143	5	3.5	-1.5	0.56	0.31
144	P144	15	14	-1	1.06	1.13
145	P145	7	4.5	-2.5	-0.44	0.19
146	P146	14	11	-3	-0.94	0.88
147	P147	11	8.5	-2.5	-0.44	0.19

Dónde:

X_A : Representa la evaluación de la línea base, es decir los costos del pre-test.

X_D : Representa la evaluación de los resultados obtenidos, es decir los costos del post test.

$\hat{\delta} = (X_D - X_A)$: Diferencias entre los momentos después y antes, es decir post- test menos pre-test.

4.1.1.2 FORMULACIÓN DE LA HIPÓTESIS ESTADÍSTICA

➤ **Hipótesis Nula (H₀):**

La utilización de la aplicación web no disminuye los costos que asumen la comunidad universitaria al optar por un transporte alternativo pagado al bus universitario. No disminuyendo los costos.

$$H_0: \mu_D \geq \mu_A; \mu_D - \mu_A \geq 0$$

➤ **Hipótesis alternativa (H_a):**

La utilización de la aplicación web disminuye los costos que asumen la comunidad universitaria al optar por un transporte alternativo pagado al bus universitario. Disminuyendo significativamente los costos.

$$H_a: \mu_D < \mu_A; \mu_D - \mu_A < 0$$

4.1.1.3 NIVEL DE SIGNIFICANCIA

Para esta investigación utilizaremos 5% de nivel de significancia, debido que se trata de un proyecto de investigación, por lo cual se utilizará ese. nivel de significancia.

$$\alpha = 0.05$$

4.1.1.4 VALORES ESTADÍSTICOS DEL PROCEDIMIENTO

$$\bar{d} = \frac{-303}{147} = -2.06$$

$$\sigma_d = \sqrt{\frac{97.31}{146}} = 0.82$$

$$t = \frac{-2.06}{\frac{0.82}{\sqrt{147}}} = -30.46$$

De acuerdo a la tabla t-student el valor de la probabilidad p del valor estadístico del procedimiento es el siguiente:

$$\rho(t < -30.46) = 0.0005$$

4.1.1.5 ESTABLECER REGIÓN CRÍTICA

gl: Grados de libertad (N - 1)

tt: Distribución t de student

$$gl = 146$$

$$tt_{0.05;146} = -1.645$$

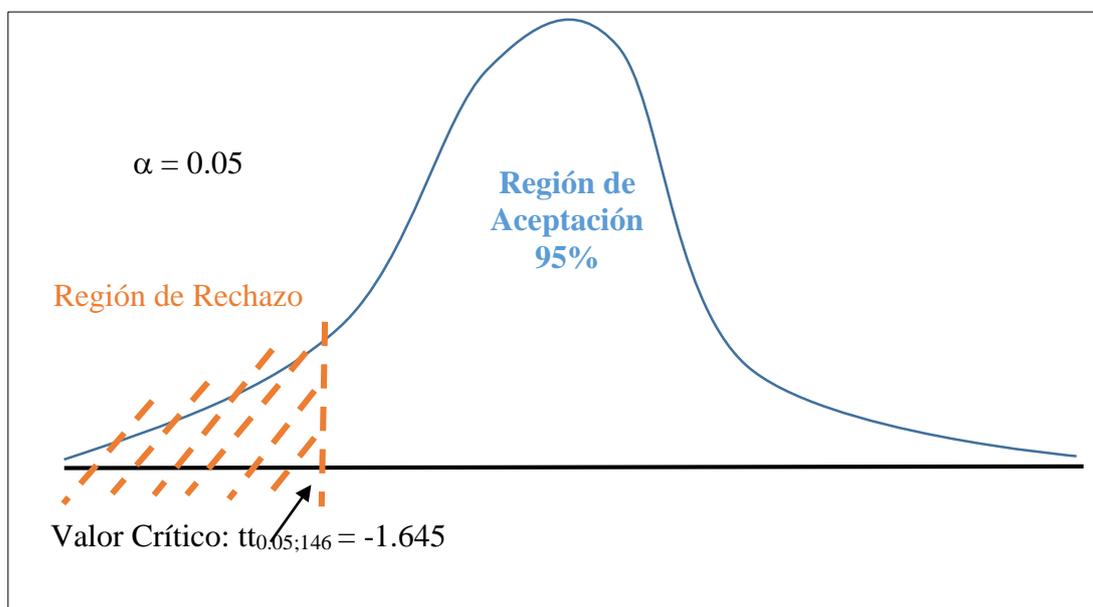


Fig. 103 Región de Aceptación y Rechazo – Indicador Tiempo

De acuerdo a lo obtenido:

$$t < tt = -30.46 < -1.645$$

$$p < a = 0.0005 < 0.05$$

Como son verdaderas las dos proposiciones, entonces rechazamos la hipótesis H_0 , y aceptamos H_a .

4.1.1.6 TOMA DE DECISIÓN

Aceptamos la hipótesis alternativa y diremos que se puede ratificar que la implementación de la aplicación web tiene un efecto positivo en la disminución de los costos que asumen los usuarios de los buses universitario, con un nivel de confianza del 95% y un nivel de significancia del 5%. Los datos confirman la hipótesis parcialmente porque se ha inicializado un cambio y una mejora en cuanto al servicio del transporte público interno de la Universidad Nacional de Cajamarca; sin embargo, depende de la Universidad Nacional de Cajamarca en que este cambio sea continuo y permanente y que se respete el proceso establecido.

4.1.2 INDICADOR: COSTO QUE ASUMEN LA COMUNIDAD UNIVERSITARIA AL OPTAR POR UN TRANSPORTE ALTERNATIVO PAGADO AL BUS UNIVERSITARIO

Utilizaremos la siguiente fórmula para muestras relacionadas:

$$t = \frac{\bar{\partial}}{\frac{\sigma \partial}{\sqrt{N}}}$$

Donde:

t: Valor estadístico del procedimiento.

$\bar{\partial}$: Valor promedio o media aritmética de las diferencias entre los momentos después y antes.

$\sigma \partial$: Desviación estándar de las diferencias entre los momentos después y antes.

N: Tamaño de la muestra.

La media aritmética de las diferencias se obtiene de la siguiente manera:

$$\bar{\partial} = \frac{\sum \partial}{N}$$

La desviación estándar de las diferencias se obtiene como sigue:

$$\sigma \partial = \sqrt{\frac{\sum(\partial - \bar{\partial})^2}{N - 1}}$$

Trabajaremos con la tabla resumen de los datos obtenidos de la evaluación: En el ANEXO 3 se encuentran los valores del PRE TEST y en el ANEXO 4 los valores del POST TEST tomados como prueba.

Tabla 50: Resumen tiempos pre test y post test

TABLA RESUMEN EN MINUTOS						
Nº	Persona	PRE-TEST	POST-TEST	$\partial = (X_D - X_A)$	$(\partial - \bar{\partial})$	$(\partial - \bar{\partial})^2$
		X_A	X_D			
1	P1	6	6	0	1.48	2.20
2	P2	7	6	-1	0.48	0.23
3	P3	7	7	0	1.48	2.20
4	P4	8	7	-1	0.48	0.23
5	P5	8	5	-3	-1.52	2.30
6	P6	7	5	-2	-0.52	0.27
7	P7	8	7	-1	0.48	0.23
8	P8	7	5	-2	-0.52	0.27
9	P9	6	3	-3	-1.52	2.30
10	P10	7	6	-1	0.48	0.23
11	P11	7	5	-2	-0.52	0.27
12	P12	7	6	-1	0.48	0.23
13	P13	5	3	-2	-0.52	0.27
14	P14	9	7	-2	-0.52	0.27
15	P15	7	5	-2	-0.52	0.27
16	P16	7	6	-1	0.48	0.23
17	P17	7	5	-2	-0.52	0.27
18	P18	6	4	-2	-0.52	0.27
19	P19	7	6	-1	0.48	0.23
20	P20	5	2	-3	-1.52	2.30
21	P21	9	7	-2	-0.52	0.27
22	P22	5	3	-2	-0.52	0.27
23	P23	6	3	-3	-1.52	2.30
24	P24	6	6	0	1.48	2.20
25	P25	8	8	0	1.48	2.20
26	P26	8	7	-1	0.48	0.23
27	P27	5	3	-2	-0.52	0.27
28	P28	7	4	-3	-1.52	2.30
29	P29	7	7	0	1.48	2.20
30	P30	6	4	-2	-0.52	0.27
31	P31	6	4	-2	-0.52	0.27
32	P32	5	5	0	1.48	2.20
33	P33	6	3	-3	-1.52	2.30
34	P34	5	5	0	1.48	2.20
35	P35	6	4	-2	-0.52	0.27
36	P36	7	6	-1	0.48	0.23
37	P37	8	6	-2	-0.52	0.27
38	P38	6	4	-2	-0.52	0.27
39	P39	7	6	-1	0.48	0.23
40	P40	8	8	0	1.48	2.20
41	P41	6	6	0	1.48	2.20
42	P42	7	6	-1	0.48	0.23
43	P43	7	4	-3	-1.52	2.30
44	P44	8	7	-1	0.48	0.23

45	P45	5	5	0	1.48	2.20
46	P46	8	8	0	1.48	2.20
47	P47	9	7	-2	-0.52	0.27
48	P48	7	4	-3	-1.52	2.30
49	P49	6	4	-2	-0.52	0.27
50	P50	6	5	-1	0.48	0.23
51	P51	5	3	-2	-0.52	0.27
52	P52	10	8	-2	-0.52	0.27
53	P53	6	3	-3	-1.52	2.30
54	P54	7	4	-3	-1.52	2.30
55	P55	6	5	-1	0.48	0.23
56	P56	6	6	0	1.48	2.20
57	P57	8	7	-1	0.48	0.23
58	P58	8	5	-3	-1.52	2.30
59	P59	7	6	-1	0.48	0.23
60	P60	7	6	-1	0.48	0.23
61	P61	7	5	-2	-0.52	0.27
62	P62	8	5	-3	-1.52	2.30
63	P63	7	5	-2	-0.52	0.27
64	P64	8	7	-1	0.48	0.23
65	P65	7	6	-1	0.48	0.23
66	P66	5	2	-3	-1.52	2.30
67	P67	5	4	-1	0.48	0.23
68	P68	8	5	-3	-1.52	2.30
69	P69	6	4	-2	-0.52	0.27
70	P70	5	3	-2	-0.52	0.27
71	P71	5	2	-3	-1.52	2.30
72	P72	6	5	-1	0.48	0.23
73	P73	7	7	0	1.48	2.20
74	P74	7	6	-1	0.48	0.23
75	P75	7	6	-1	0.48	0.23
76	P76	7	7	0	1.48	2.20
77	P77	6	4	-2	-0.52	0.27
78	P78	4	3	-1	0.48	0.23
79	P79	7	6	-1	0.48	0.23
80	P80	8	6	-2	-0.52	0.27
81	P81	8	6	-2	-0.52	0.27
82	P82	6	5	-1	0.48	0.23
83	P83	7	6	-1	0.48	0.23
84	P84	8	5	-3	-1.52	2.30
85	P85	5	3	-2	-0.52	0.27
86	P86	7	6	-1	0.48	0.23
87	P87	6	3	-3	-1.52	2.30
88	P88	9	7	-2	-0.52	0.27
89	P89	8	6	-2	-0.52	0.27
90	P90	7	6	-1	0.48	0.23
91	P91	8	6	-2	-0.52	0.27
92	P92	4	2	-2	-0.52	0.27

93	P93	7	7	0	1.48	2.20
94	P94	8	7	-1	0.48	0.23
95	P95	8	6	-2	-0.52	0.27
96	P96	9	9	0	1.48	2.20
97	P97	4	3	-1	0.48	0.23
98	P98	5	4	-1	0.48	0.23
99	P99	6	4	-2	-0.52	0.27
100	P100	6	4	-2	-0.52	0.27
101	P101	5	2	-3	-1.52	2.30
102	P102	8	7	-1	0.48	0.23
103	P103	6	4	-2	-0.52	0.27
104	P104	6	5	-1	0.48	0.23
105	P105	7	5	-2	-0.52	0.27
106	P106	4	2	-2	-0.52	0.27
107	P107	7	7	0	1.48	2.20
108	P108	6	4	-2	-0.52	0.27
109	P109	8	8	0	1.48	2.20
110	P110	5	2	-3	-1.52	2.30
111	P111	2	0	-2	-0.52	0.27
112	P112	4	3	-1	0.48	0.23
113	P113	8	7	-1	0.48	0.23
114	P114	6	3	-3	-1.52	2.30
115	P115	4	2	-2	-0.52	0.27
116	P116	5	4	-1	0.48	0.23
117	P117	10	10	0	1.48	2.20
118	P118	6	5	-1	0.48	0.23
119	P119	7	5	-2	-0.52	0.27
120	P120	5	5	0	1.48	2.20
121	P121	5	3	-2	-0.52	0.27
122	P122	7	4	-3	-1.52	2.30
123	P123	9	9	0	1.48	2.20
124	P124	7	4	-3	-1.52	2.30
125	P125	3	0	-3	-1.52	2.30
126	P126	11	11	0	1.48	2.20
127	P127	6	5	-1	0.48	0.23
128	P128	4	3	-1	0.48	0.23
129	P129	6	3	-3	-1.52	2.30
130	P130	5	4	-1	0.48	0.23
131	P131	7	6	-1	0.48	0.23
132	P132	8	7	-1	0.48	0.23
133	P133	8	6	-2	-0.52	0.27
134	P134	7	6	-1	0.48	0.23
135	P135	8	7	-1	0.48	0.23
136	P136	5	4	-1	0.48	0.23
137	P137	4	1	-3	-1.52	2.30
138	P138	4	3	-1	0.48	0.23
139	P139	3	2	-1	0.48	0.23
140	P140	8	6	-2	-0.52	0.27

141	P141	6	6	0	1.48	2.20
142	P142	6	4	-2	-0.52	0.27
143	P143	7	6	-1	0.48	0.23
144	P144	8	8	0	1.48	2.20
145	P145	7	6	-1	0.48	0.23
146	P146	7	7	0	1.48	2.20
147	P147	7	6	-1	0.48	0.23

Dónde:

X_A : Representa la evaluación de la línea base, es decir los tiempos del pre-test.

X_D : Representa la evaluación de los resultados obtenidos, es decir los tiempos del post test.

$\delta = (X_D - X_A)$: Diferencias entre los momentos después y antes, es decir post- test menos pre-test.

4.1.2.1 FORMULACIÓN DE LA HIPÓTESIS ESTADÍSTICA

➤ **Hipótesis Nula (H_0):**

La utilización de la aplicación web no disminuye los tiempos de espera de la comunidad universitaria por los buses. No disminuyendo los tiempos.

$$H_0: \mu_D \geq \mu_A; \mu_D - \mu_A \geq 0$$

➤ **Hipótesis alternativa (H_a):**

La utilización de la aplicación web disminuye los tiempos de espera de la comunidad universitaria por los buses universitarios. Disminuyendo significativamente los tiempos.

$$H_a: \mu_D < \mu_A; \mu_D - \mu_A < 0$$

4.1.2.2 NIVEL DE SIGNIFICANCIA

Para esta investigación utilizaremos 5% de nivel de significancia, debido que se trata de un proyecto de investigación, por lo cual se utilizará ese. nivel de significancia.

$$\alpha = 0.05$$

4.1.2.3 VALORES ESTADÍSTICOS DEL PROCEDIMIENTO

$$\bar{d} = \frac{-218}{147} = -1.48$$

$$\sigma d = \sqrt{\frac{136.71}{146}} = 0.97$$

$$t = \frac{-1.48}{\frac{0.97}{\sqrt{147}}} = -18.58$$

De acuerdo a la tabla t-student el valor de la probabilidad p del valor estadístico del procedimiento es el siguiente:

$$\rho(t < -18.58) = 0.0005$$

4.1.2.4 ESTABLECER REGIÓN CRÍTICA

gl: Grados de libertad (N - 1)

tt: Distribución t de student

$$gl = 146$$

$$tt_{0.05;146} = -1.645$$

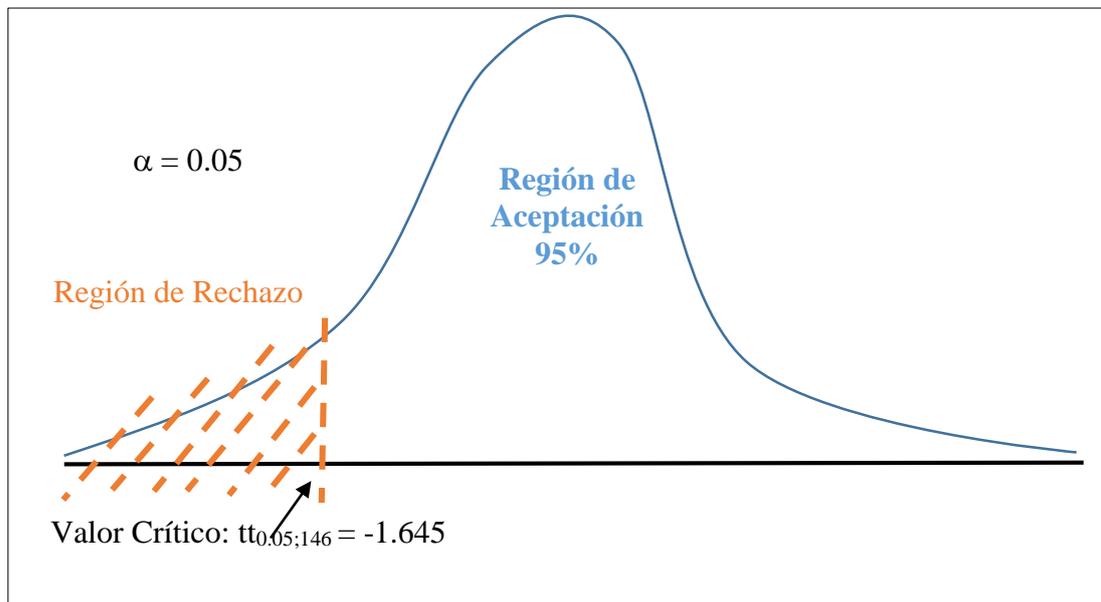


Fig. 104 Región de Aceptación y Rechazo – Indicador Costos

De acuerdo a lo obtenido:

$$t < tt = -18.58 < -1.645$$

$$p < \alpha = 0.0005 < 0.05$$

Como son verdaderas las dos proposiciones, entonces rechazamos la hipótesis H_0 , y aceptamos H_a .

4.1.2.5 TOMA DE DECISIÓN

Aceptamos la hipótesis alternativa y diremos que se puede ratificar que la implementación de la aplicación web tiene un efecto positivo en la disminución de los tiempos de espera de los usuarios de los buses universitarios, con un nivel de confianza del 95% y un nivel de significancia del 5%. Los datos confirman la hipótesis parcialmente porque se ha inicializado un cambio y una mejora en cuanto al servicio del transporte público interno de la Universidad Nacional de Cajamarca; sin embargo, depende de la Universidad Nacional de Cajamarca en que este cambio sea continuo y permanente y que se respete el proceso establecido.

4.2 DISCUSIÓN DE RESULTADOS

4.2.1 INTERPRETACIÓN DE LA INFORMACIÓN

Con los resultados obtenidos podemos afirmar que efectivamente el desarrollo de una aplicación web para el control y monitoreo en tiempo real de los buses de la Universidad Nacional de Cajamarca produce resultados favorables y ayuda a mejorar el servicio del transporte interno de la Universidad Nacional de Cajamarca, reduciendo los tiempos de espera de los usuarios y ayudándoles a reducir los gastos usados en otros medios de transportes pagados.

Cuando se decidió optar por el desarrollo de una aplicación web para el control y monitoreo en tiempo real de los buses de la Universidad Nacional de Cajamarca, de antemano se había observado que tenía deficiencias en el servicio del transporte interno, los prolongados lapsos de tiempo hasta la llegada del autobús universitario, falta de control en la programación de las salidas, suboptimización en el uso del servicio por parte de la comunidad universitaria, esto se debe a que, no existe comunicación a tiempo real entre los usuarios y los autobuses, lo que genera una desorganización entre la oferta y la demanda del servicio (Asimetría de Información). Y con ello, la ineficiencia del servicio, una vez implementado la aplicación web se vieron mejorados los lapsos de tiempo dando la información de los buses en tiempo real a los usuarios, lo cual trajo consigo una mejor calidad en el servicio.

En la primera parte del despliegue de la aplicación web los conductores de los buses universitarios tuvieron muchos problemas para adaptarse al proceso, al igual que la

aplicación web presentaba fallas, y los usuarios no estaban totalmente satisfechos con la implementación, además de la resistencia al cambio, debido a que una aplicación web siempre pasa por una fase prueba hasta que se corrijan todos los errores y se llegue a una estabilidad.

En la segunda parte cuando ya se estabilizó la aplicación web y se habían cumplido con los requerimientos en su totalidad se observó que los conductores de los buses se acostumbraron, por lo que se pudo dar como satisfactoria la implementación realizada en los buses de la Universidad Nacional de Cajamarca.

Al comparar un antes y un después de la investigación se mejoró en cuanto a:

➤ **Tiempo de espera**

Se ha disminuido el tiempo de espera de los usuarios por los buses universitarios, se cuenta con información a tiempo real proporcionada por los buses, a la cual pueden acceder los usuarios únicamente con conexión a internet, lo que da como resultado que los usuarios optimicen sus tiempos y un mejor servicio del transporte interno de la Universidad Nacional de Cajamarca.

➤ **Costos**

Se han disminuido los costos asumidos por los usuarios de los buses universitarios que optan por un transporte alternativo pagado, se cuenta con la información a tiempo real de la llegada de los buses lo cual les permite a los usuarios optar por tomar los buses universitarios en lugar de otro medio de transporte pagado, lo que da como resultado un ahorro económico para los usuarios de los buses universitarios y un mejor servicio del transporte interno de la Universidad Nacional de Cajamarca.

4.2.2 SIMILITUDES ENTRE RESULTADOS OBTENIDOS EN LA INVESTIGACIÓN Y LOS ANTECEDENTES TEÓRICOS

Con respecto a la comparación de los resultados obtenidos en el presente trabajo de investigación con los resultados de algunos de los Antecedentes Teóricos indicados anteriormente en la literatura, se encuentran las siguientes semejanzas y diferencias:

Yago [2], en su tesis: “Aplicación Web y Móvil para el seguimiento de autobuses escolares”

SEMEJANZAS: En ambas tesis se desarrolló y desplego una plataforma web, para el seguimiento de buses a una zona definida, utilizando geolocalización por GPS, y compilando la aplicación web como aplicaciones móviles.

DIFERENCIAS: En la tesis se utiliza push notifications o emails que son medios de comunicación en tiempo real más lentos, en la aplicación web para los buses de la Universidad Nacional de Cajamarca se usa websockets es una tecnología basada en el protocolo ws, este hace posible establecer una conexión continua full - duplex, entre un cliente y servidor.

Ching [3], en su tesis “Bus tracking system”

SEMEJANZAS: En ambas tesis desarrolló un sistema un sistema de seguimiento de autobuses que proporciona una plataforma en tiempo real para verificar la información actualizada del tráfico del autobús.

DIFERENCIAS: En la tesis se utiliza receptores GPS en los buses buses y usando la tecnología GPS de rastreo se consiguió enviar los datos de posición del autobús hacia el servidor central, y en el caso de la presente tesis se utiliza un dispositivo móvil (celular) que se encarga de enviar las coordenadas obtenidas por el celular, el paradero de destino y el código del dispositivo hacia el servidor web.

Conza [4], en su tesis: “Desarrollo de una aplicación web, orientada a servicios para el monitoreo de una flota de vehículos haciendo uso de la tecnología GPS”

SEMEJANZAS: En ambas tesis se desarrolla un sistema de rastreo georeferencial para el rastreo y monitoreo de una flota vehicular, donde se proporciona la posición hacia un servidor central, el cual al mismo tiempo envían estos datos a los usuarios a través de websockets que en ese preciso momento están conectados rastreando los vehículos en tiempo real, dichos usuarios visualizan la ubicación de los vehículos en tiempo real recibida desde la estación central a través de un mapa que provee Google Maps.

DIFERENCIAS: En la tesis utiliza GPS equipados en los vehículos, los cuales envían la información al servidor central mediante web REST, y en el caso de la presente tesis se utilizan dispositivo móvil (celular) que envía la información al servidor web mediante websockets.

Meza, Leño [5], en su tesis: “Sistema de monitoreo de una red de buses de transporte público e información para usuarios empleando transeptores GPS/GSM”

SEMEJANZAS: En ambas tesis se desarrolló un sistema de monitoreo de una red de buses, mediante el empleo de tecnologías de comunicación y posicionamiento, enviando la información de posición (latitud, longitud) hacia un servidor.

DIFERENCIAS: En la tesis se utilizan transeptores GPS/GSM que permite obtener datos de posición (latitud, longitud), fecha, hora y velocidad de desplazamiento del móvil monitoreado, la información de tiempo y distancia de llegada a los usuarios a través de mensajes de texto o aplicaciones para smartphones, y en el caso de la presente tesis utilizan dispositivo móvil (celular) y la información es enviado a los clientes que estén conectados a la aplicación web.

Ureta [6], en su tesis; “Desarrollo de una aplicación web para el monitoreo de vehículos con dispositivos GPS que comercializa una empresa de telecomunicaciones”

SEMEJANZAS: En ambas tesis se logró implementar una solución web de monitoreo vehicular a múltiples clientes sin necesidad de instalaciones personales, solo con el uso de un navegador web

DIFERENCIAS: En la tesis software forma parte del servicio de monitoreo vehicular que ofrece una empresa de telecomunicaciones, y en el caso de la presente tesis no se utiliza ningún servicio de monitoreo vehicular externo, es todo desarrollado por el mismo programador.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Se cumplió con el objetivo de desarrollar una aplicación web para el control y monitoreo en tiempo real que mejoró el servicio del transporte interno de la Universidad Nacional de Cajamarca, que le brinda a los usuarios de los buses universitarios información en tiempo real, con la finalidad de disminuir los tiempos de espera y gastos en transportes alternativos pagados por parte de los usuarios de los buses de la Universidad Nacional de Cajamarca, obteniendo un efecto positivo en la implementación de la aplicación web, mejorando los tiempos de espera y disminuyendo los gastos en transportes alternativos.

Se logró describir y analizar cada proceso que se lleva a cabo dentro de la aplicación web, los cuales fueron descritos en los casos de uso, y en los diagramas de actividad. Para la aplicación web presentada se eligió el Proceso Unificado de Desarrollo de Software, metodología de desarrollo de software que gracias a su flexibilidad permitió crear un marco de trabajo especializado que sirve de guía durante el proceso de desarrollo.

Se logró diseñar la arquitectura de la aplicación web, lo cual fue diseñado con la ayuda del lenguaje unificado de modelado (UML) como herramienta para desarrollar los planos del software.

La implementación de la aplicación web con las características descritas en los capítulos anteriores se desarrolló en el lenguaje de programación Node.js, instalándolo correctamente y se realizó las pruebas necesarias para corregir los errores que pudieron presentarse al desplegarse al inicio de la implementación. Se logró la capacitación a los conductores de los buses universitarios, pudiendo resolver sus dudas y tomando sus recomendaciones para realizar cambios que se ajusten a sus sugerencias.

Se logró realizar las pruebas a la aplicación web, lo cual fue realizado por medio de pruebas de carga, de rendimiento y de estrés las cuales se realizaron para observar el comportamiento de la aplicación bajo una cantidad de peticiones esperadas, monitoreando los tiempos de respuesta de la aplicación web.

La aplicación web optimizó los tiempos de espera y disminuyó los gastos en medios de transporte alternativos pagados, para evaluar el efecto se realizó un análisis estadístico,

con PRE TEST y POST TEST, además se realizó una encuesta de satisfacción a los usuarios de los buses universitarios. La aplicación web desarrollada y desplegada agrega valor procedimental y eficiencia al servicio del transporte interno de la Universidad Nacional de Cajamarca pues:

- Reduce los costos en otros medios de transportes pagados por parte de los usuarios de los buses universitarios, generándoles un ahorro económico a dichos usuarios.
- Minimiza los tiempos de espera para abordar los buses universitarios, debido a que se puede tener acceso en tiempo real a la información de los buses universitarios.
- Existe una mayor seguridad y el respaldo de la información en el sistema, esto se debe a que los datos están protegidos de personas ajenas respetando la privacidad de los usuarios.

Los costos del sistema se calculan en base al costo de los recursos que se estiman emplear para su construcción:

- Recursos de software. Las cuentas de desarrollador de android, iOS y el servidor web, licencias de las aplicaciones de análisis, diseño y construcción están incluidas el plan del presupuesto.
- Recurso de hardware. Los dispositivos móviles (Celulares) usados para el envío de la información de los buses pertenecen al área de transporte de la Universidad Nacional de Cajamarca; así como los ambientes de desarrollo, preproducción y producción con una arquitectura similar a la usada para este proyecto.

Fue necesario abarcar las tareas administrativas que tienen que ver con el transporte interno de la Universidad Nacional de Cajamarca ya que, todas las tareas mencionadas en los capítulos anteriores, permiten que el lector de la tesis tenga un enfoque completo de lo que esta tesis quiere mostrar.

5.2 RECOMENDACIONES

Es recomendable que se tenga en cuenta para la implementación de futuras versiones de la aplicación web y móvil, las siguientes funcionalidades:

- Se debe desarrollar la aplicación web de forma nativa para el sistema operativo iOS.
- Para los posteriores desarrollos nativos móviles, también se debe utilizar arquitectura de capas por motivo de seguridad.
- Se debe agregar un manual de uso en la aplicación web que se le mostrará a los usuarios únicamente la primera vez que ingresen a la aplicación web y también que pueda visualizarse a través de un botón.
- La utilización correcta del manual de uso permite que cada módulo se pueda manejar correctamente todos los módulos implementados dentro de la aplicación web, así como también en el caso de que se quiera hacer algún cambio de las características o funcionalidad de alguno de los módulos dentro de la aplicación web, en caso de existir algún requerimiento futuro.
- Se debe realizar los backup de la información, ante probables contingencias, estos backup sirven para mayor seguridad; deben estar en un contenedor web, al igual que una copia de la aplicación web.
- El éxito en la implantación de la aplicación web se dará cuando la aplicación web se instale en todos los buses universitarios. Para ello se recomienda capacitar constantemente a los conductores de los buses, solucionando sus dudas o algún inconveniente que puedan tener ya que ellos serán los que usen la aplicación web para el envío de la información de los buses en tiempo real. También el administrador de la aplicación web deberá realizar ocasionalmente las pruebas necesarias para garantizar la estabilidad de la aplicación web.
- Se puede tomar como base la implementación de la aplicación, para realizar una réplica en otros campos donde sea necesario la comunicación bidireccional en tiempo real, por ejemplo, en el servicio de transporte público, transporte privado, taxis, moto taxis, entre otros.
- La seguridad de la aplicación web, al estar en internet, aun es vulnerable ya que se utiliza el protocolo HTTP (Hyper Text Transport Protocol), se recomienda mejorar la seguridad utilizando el protocolo HTTPS (Hypertext Transfer Protocol Secure).

REFERENCIAS BIBLIOGRÁFICAS

- [1] T. J. Cham, *Fundamentals of Global Positioning System Receivers A Software Approach* John Wiley & Sons, 2000.
- [2] E. Y. Marco, *Aplicación Web y Móvil para el seguimiento de autobuses escolares*, Valencia, 2015.
- [3] C. C. Loong, *Bus Tracking System*, Perak, 2013.
- [4] M. H. C. Berrocal, *Desarrollo de una Aplicación Web Orientada a Servicios Para el Monitoreo de una Flota de Vehículos Haciendo uso de la Tecnología GPS*, Cusco, 2013.
- [5] J. C. M. Romer y V. G. L. Pariona, *Sistema de monitoreo de una red de buses de transporte público e información para usuarios empleando transceptores GPS/GSM*, Lima, 2017.
- [6] R. G. B. Ureta, *Desarrollo de una aplicación web para el monitoreo de vehículos con dispositivos gps que comercializa una empresa de telecomunicaciones*, Lima, 2012.
- [7] A. Escalona, «Monografias,» [En línea]. Available: <https://www.monografias.com/trabajos81/transporte-publico/transporte-publico2.shtml>. [Último acceso: 2018 06 26].
- [8] P. R. Bosch, *Improvements in autonomous GPS navigation of Low Earth Orbit Satellites*, Catalunya, 2008.
- [9] A. P. Ruz, *Sistema de Posicionamiento Global (GPS): Descripción, Análisis de errores, aplicaciones y futuro*, Málaga, 2000.
- [10] R. R. Solano, *Las Técnicas GPS como herramienta en la gestión ambiental*, 1998.
- [11] T. Jen, *Advances in multimedia modeling*, Singapur, 2007.
- [12] E. D. Kaplan, *Understanding GPS: Principles and applications*, 2006.
- [13] A. Thangavelu, *Location and Identification and Vehicle Tracking using VANET (VETRAC) IEEE – ICSCN- MIT*, Chennai, 2007.
- [14] W. Schramm, «Emersonbarradas,» 26 Octubre 2008. [En línea]. Available: <http://emersonbarradas.blogspot.com/2008/10/teoras-y-modelos-de-la-comunicacin.html>. [Último acceso: 26 Junio 2018].

- [15] J. Tonina, «Folderit,» 04 01 2016. [En línea]. Available: <https://folderit.net/itech/es/nodejs-parte-i-es/>. [Último acceso: 26 06 2018].
- [16] A. Arrs, «Hipertextual,» 08 08 2018. [En línea]. Available: <https://hipertextual.com/archivo/2014/08/socketio-javascript/>. [Último acceso: 26 06 2018].
- [17] S. Jablonski, Guide to web application and platform architectures, Springer, 2004.
- [18] S. Leon, Web Application architecture: principles, protocols and practices, 2003.
- [19] Krebs y Jochen, The value of RUP certification, IBM. Retrieved 2008- 05-13, 2007.
- [20] I. Jacobson, G. BOOCH y J. RUMBAUGH, The Unified Software Development Process. Addison-Wesley, 1999.
- [21] S. L. Mora, Programación en Internet: Clientes Web, 2001.
- [22] P. Transportation.org, «Public Transportation.org,» 30 04 2011. [En línea]. Available: <https://www.publictransportation.org/Pages/default.aspx>. [Último acceso: 26 06 2018].
- [23] Wikipedia, «Wikipedia,» [En línea]. Available: <https://es.wikipedia.org/wiki/WebSocket>. [Último acceso: 27 06 2018].
- [24] G. Gomez, Manual pruebas de funcionamiento para monitoreo, México, 2016.
- [25] L. Flores, «Brainly.lat,» [En línea]. Available: <https://brainly.lat/tarea/6831791>. [Último acceso: 27 06 2018].
- [26] Wikipedia, «Wikipedia,» 04 10 2017. [En línea]. Available: https://es.wikipedia.org/wiki/Receptor_GPS. [Último acceso: 27 06 2018].
- [27] J. A. M. Jaimes, Sistemas de control y monitoreo vehicular, 2014.
- [28] Lorenzi NM, Kouroubali A, Detmer DE y Bloomrosen M, How to successfully select and implement electronic health records (EHR) in small ambulatory practice settings. BMC medical informatics and decision making, 2009.
- [29] I. Jacobson, G. BOOCH y J. RUMBAUGH, The Unified Software Development Process. Addison-Wesley, 1999.
- [30] C. Larman, UML y patrones. Prentice Hall, 2002.
- [31] J. M. Gil, Procedimientos estadísticos, Madrid: ESIC, 1998.

- [32] Microsoft Corporation , Microsoft SQL Server 2012, California: O'Reilly Media, 2013.
- [33] Devexpress, «Devexpress,» Developer Express Inc, 1998 - 2016. [En línea]. Available: <https://www.devexpress.com/>.
- [34] D. propio, Escritor, *Diseño Propio*. [Performance]. 2018.
- [35] S. L. Mora, Programación en Internet: Clientes We, 2001.
- [36] R. A. E. y. A. d. A. d. l. L. Española, Diccionario de la lengua española, Madrid, 2014.

ANEXOS

ANEXO 1: CRONOGRAMA DE TESIS

Actividad		Julio	Agosto	Setiembre	Octubre	Noviembre	Diciembre	Enero
PROYECTO	Selección y delimitación de Tema	X						
	Problematización	X						
	Justificación	X						
	Marco Teórico	X						
	Hipótesis		X					
	Metodología		X					
INVESTIGACIÓN	Revisión de fuentes	X	X					
	Revisión de fuentes Primarias							
	Búsqueda de información complementaria		X					
ELABORACIÓN	Primer Capítulo			X				
	Segundo Capítulo			X	X			
	Tercer Capítulo				X	X		
	Cuarto Capítulo					X	X	
	Introducción a Conclusiones			X	X	X	X	
Lectura de los						X		
Correcciones y						X	X	
Elaboración de copia							X	
Defensa final.							X	

ANEXO 2: ENCUESTA A LOS USUARIOS DE LOS BUSES UNIVERSITARIOS

Cuestionario de satisfacción del servicio del transporte interno de la Universidad Nacional de Cajamarca para el desarrollo de una aplicación web de control y monitoreo.

ENCUESTA

La finalidad de esta encuesta es para conocer la opinión de los usuarios de los buses universitarios con respecto a la calidad del servicio del transporte interno de la Universidad Nacional de Cajamarca. Por favor marque con una X su respuesta a cada una de las preguntas.

DATOS GENERALES

-
1. Sexo Hombre Mujer
2. Edad 12 – 17 18 – 24 25 – 34
- 35 – 44 45 – 54 45 – 54
- 55 – 64 65 y +

TIEMPO

-
3. ¿Usted cuántos minutos espera normalmente para abordar el bus?
- Menos de 1 minuto De 1 a 3 minutos De 3 a 5 minutos
- De 3 a 10 minutos De 10 a 15 minutos De 15 a 30 minutos
- De 30 a 60 minutos Más de 1 hora
4. ¿Usted cuánto tiempo tarda normalmente en llegar a su destino final?
- Menos de 15 min Entre 15 y 30 min Entre 30 y 1 hora
- Entre 1 y 1 ½ hora Más de 1 ½ hora

COSTOS

5. ¿Usted cuánto gasta al día en transporte público?

_____ soles

6. En caso de usar otro medio indique usted ¿Cuánto gasta al día?

_____ Soles

ANEXO 3: DATOS RECABADOS PRE TEST

PERSONA	PREGUNTA			
	P3	P4	P5	P6
E1	5	1	1	2
E2	5	2	1	1
E3	5	2	3	1
E4	6	2	1	2
E5	5	3	3.5	7
E6	5	2	4	4
E7	6	2	4	4
E8	5	2	2	5
E9	4	2	1	2
E10	5	2	1	1
E11	6	1	4	6
E12	6	1	7	7
E13	4	1	5	6
E14	6	3	10	20
E15	6	1	2	3
E16	6	1	5	5
E17	6	1	1	1
E18	4	2	5	7
E19	5	2	2	2
E20	4	1	0.5	0.5
E21	7	2	2	2
E22	3	2	3	10
E23	4	2	4	4
E24	5	1	2	4
E25	6	2	2	2
E26	6	2	0.5	5
E27	3	2	3	5
E28	4	3	2	2
E29	5	2	1.5	1.5
E30	4	2	2	3
E31	4	2	2	4
E32	4	1	3	3
E33	4	2	2	5
E34	3	2	3	4
E35	4	2	2	10
E36	5	2	3	6
E37	6	2	0.7	4
E38	4	2	2	1

E39	5	2	4	6
E40	6	2	4	8
E41	5	1	2	2
E42	5	2	4	5
E43	5	2	2	3
E44	6	2	1	4
E45	4	1	2	6
E46	5	3	1.4	1.4
E47	6	3	5	5
E48	5	2	1	7
E49	5	1	2	2
E50	4	2	1	1
E51	3	2	2.8	4
E52	7	3	2	2
E53	5	1	1	2
E54	5	2	0.5	3
E55	5	1	1	1.5
E56	5	1	0.5	0.5
E57	6	2	4	4
E58	6	2	3	3
E59	5	2	4	4
E60	5	2	2	2
E61	5	2	1	3
E62	7	1	1	1
E63	5	2	5	1
E64	5	3	1	3
E65	5	2	1.5	1.5
E66	4	1	8	8
E67	3	2	2	2
E68	5	3	6	6
E69	5	1	4	2
E70	4	1	4	4
E71	4	1	1	1
E72	4	2	1.4	1.4
E73	4	3	4	4
E74	6	1	3	4
E75	6	1	1.5	3
E76	5	2	3	4
E77	5	1	0	0
E78	1	3	1	2
E79	5	2	4	4
E80	7	1	5	10

E81	6	2	2	4
E82	5	1	0	0
E83	5	2	1	8
E84	6	2	4	5
E85	4	1	0.5	1
E86	5	2	3	5
E87	5	1	1	7
E88	7	2	2	5
E89	6	2	2	5
E90	5	2	1	1
E91	6	2	4	4
E92	3	1	2	4
E93	5	2	2	2
E94	5	3	4	4
E95	6	2	0	0
E96	7	2	4	3
E97	3	1	0.5	0.5
E98	4	1	1	1
E99	5	1	2	5
E100	5	1	4.5	8
E101	4	1	2	3
E102	5	3	2	0.5
E103	5	1	1	1
E104	5	1	1	2
E105	5	2	0.5	2
E106	3	1	2	2
E107	5	2	6	6
E108	5	1	6	15
E109	6	2	5	6
E110	3	2	18	12
E111	1	1	2	1
E112	2	2	1.5	1.5
E113	6	2	1	10
E114	5	1	4	6
E115	3	1	2	2
E116	3	2	5	5
E117	7	3	2	3
E118	5	1	4	4
E119	6	1	3	5
E120	3	2	2	10
E121	3	2	1	2.5
E122	5	2	2	10

E123	6	3	5	5
E124	6	1	10	10
E125	2	1	1	3
E126	6	5	1	2
E127	4	2	1.5	1.5
E128	3	1	1	1
E129	4	2	2	1
E130	3	2	2	2
E131	5	2	3	3
E132	3	5	5	5
E133	5	3	0.5	0.5
E134	5	2	1	3
E135	7	1	2	1
E136	3	2	2	2
E137	3	1	2	4
E138	3	1	2	5
E139	2	1	1	3
E140	5	3	2	5
E141	4	2	10	10
E142	5	1	1.5	1.5
E143	5	2	2	3
E144	5	3	5	10
E145	5	2	2	5
E146	5	2	4	10
E147	5	2	4	7

ANEXO 4: DATOS RECABADOS POST TEST

PERSONA	PREGUNTA			
	P3	P4	P5	P6
E1	5	1	0	1
E2	4	2	0.5	0
E3	5	2	1	0.5
E4	6	1	0	1
E5	3	2	1.5	6
E6	3	2	3	3
E7	6	1	2	3.5
E8	3	2	1	3
E9	2	1	0.5	0
E10	4	2	1	0
E11	5	0	3	5.5
E12	5	1	6.5	6
E13	2	1	3	4
E14	5	2	9	19.5
E15	4	1	1.5	2
E16	6	0	4.5	3
E17	4	1	0	0.5
E18	2	2	4.5	6
E19	5	1	1	1.5
E20	2	0	0	0
E21	5	2	1	1
E22	2	1	1	8
E23	2	1	2	3.5
E24	5	1	1.5	3.5
E25	6	2	0	1.5
E26	6	1	0	3
E27	2	1	2	3
E28	2	2	1.5	0
E29	5	2	0	0.5
E30	2	2	0	2
E31	3	1	0	3
E32	4	1	2.5	2.5
E33	2	1	1.5	3
E34	3	2	2.5	3.5
E35	2	2	0	9
E36	4	2	1	5
E37	4	2	0	3.5
E38	3	1	1.5	0.5

E39	5	1	3.5	5.5
E40	6	2	3	7
E41	5	1	1	0
E42	4	2	3	4
E43	3	1	1	2
E44	6	1	0.5	2
E45	4	1	1	4
E46	5	3	0.7	0.7
E47	5	2	4	3
E48	3	1	0	5
E49	4	0	1.5	1
E50	4	1	0.5	0
E51	2	1	2.3	3
E52	6	2	0	1.5
E53	3	0	0	1.5
E54	3	1	0	2
E55	5	0	0.5	0
E56	5	1	0	0
E57	5	2	2	3
E58	4	1	1	2.5
E59	5	1	2	3
E60	4	2	1.5	0
E61	4	1	0	1
E62	5	0	0	0
E63	4	1	4.5	0
E64	4	3	0.5	2.5
E65	4	2	1	0.5
E66	2	0	6	7
E67	2	2	0	1.5
E68	3	2	5.5	5
E69	3	1	2	1.5
E70	3	0	3.5	3.5
E71	2	0	0.5	0.5
E72	4	1	0.7	0.7
E73	4	3	3	2
E74	5	1	1	3
E75	6	0	0	2
E76	5	2	2	3.5
E77	4	0	0	0
E78	0	3	0	0
E79	5	1	2	3.5
E80	6	0	4	8

E81	5	1	1.5	3
E82	5	0	0	0
E83	5	1	0.5	6
E84	4	1	2	4
E85	3	0	0	0.5
E86	4	2	2	4.5
E87	3	0	0	6.5
E88	6	1	1.5	4
E89	5	1	1.5	4
E90	5	1	0.5	0
E91	5	1	3	3
E92	2	0	1.5	3
E93	5	2	1	1.5
E94	4	3	2	3.5
E95	4	2	0	0
E96	7	2	2	2
E97	2	1	0	0
E98	3	1	0	0
E99	3	1	1.5	3
E100	3	1	3.5	7.5
E101	2	0	1	1
E102	4	3	1.5	0
E103	3	1	0	0
E104	4	1	0.5	1
E105	3	2	0	1
E106	1	1	0	1
E107	5	2	4	4
E108	3	1	4	13
E109	6	2	4	5.5
E110	1	1	17.5	10
E111	0	0	1.5	0
E112	1	2	0.5	0.5
E113	6	1	0	8
E114	3	0	2	5
E115	2	0	1.5	0
E116	2	2	4	4.5
E117	7	3	1	1
E118	4	1	2	3.5
E119	4	1	2	4.5
E120	3	2	1	9.5
E121	2	1	0	2
E122	3	1	1	9

E123	6	3	3	4.5
E124	4	0	8	9.5
E125	0	0	0.5	2
E126	6	5	0	1.5
E127	4	1	0.5	1
E128	2	1	0	0.5
E129	2	1	1	0
E130	3	1	1	1.5
E131	4	2	2.5	1
E132	2	5	4.5	4
E133	3	3	0	0
E134	5	1	0.5	1
E135	6	1	1	0.5
E136	2	2	1	1.5
E137	1	0	0	3
E138	3	0	0	3
E139	2	0	0	2.5
E140	3	3	1	3
E141	4	2	8	9
E142	4	0	0.5	0.5
E143	4	2	1.5	2
E144	5	3	4.5	9.5
E145	4	2	0	4.5
E146	5	2	2	9
E147	5	1	3.5	5

ANEXO 5: VALIDACIÓN DEL INSTRUMENTO POR UN ESTADÍSTICO

VALIDACION DEL INSTRUMENTO

1. IDENTIFICACIÓN DEL EXPERTO

Nombre y Apellidos : Miguel Ángel Macetas Hernández
Profesión : Estadístico
Grado Académico : Doctor
Institución donde labora: Universidad Nacional de Cajamarca
Cargo : Docente
DNI : 17435544

2. RESPONSABLE DE LA INVESTIGACIÓN

César Augusto Ordoñez Marín

3. IDENTIFICACIÓN DE LA INVESTIGACIÓN

“DESARROLLO DE UNA APLICACIÓN WEB PARA EL CONTROL Y MONITOREO EN TIEMPO REAL QUE PERMITA MEJORAR EL SERVICIO DEL TRANSPORTE INTERNO DE LA UNIVERSIDAD NACIONAL DE CAJAMARCA”

4. OBJETIVOS DE LA INVESTIGACIÓN

Desarrollar una aplicación web para el control y monitoreo en tiempo real que permita mejorar el servicio del transporte interno de la Universidad Nacional de Cajamarca

5. JUICIO DEL EXPERTO RESPECTO AL CUESTIONARIO DE LA INVESTIGACIÓN

a. Pertinencia de las preguntas con los objetivos:

Suficiente: X
Medianamente Suficiente:
Insuficiente:

b. Pertinencia de las preguntas con las Variables:

Suficiente:
Medianamente Suficiente:
Insuficiente:

c. Pertinencia de las preguntas con las dimensiones:

Suficiente:
Medianamente Suficiente:
Insuficiente:

d. Pertinencia de las preguntas con los indicadores:

Suficiente:
Medianamente Suficiente:
Insuficiente:

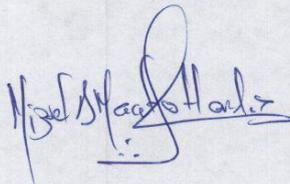
e. Redacción de las preguntas:

Adecuada:
Inadecuada:

f. El instrumento diseñado a su juicio es:

Valido:
No Valido:

Observaciones:



ANEXO 6: SOLICITUD DE VALIDACIÓN DIRIGIDA A EXPERTOS EN DESARROLLO DE SOFTWARE

Cajamarca, julio 2019

Señor:

Ing. Villar Zamora Lester Juan De Dios

Presente

Reciba un cordial saludo

Motivado por su reconocida formación y experiencia en el **desarrollo de software**, me complace dirigirme a usted para solicitar su valiosa colaboración en la validación de la aplicación web “El carro de la U”, el mismo que servirá para recolectar información relativa a la investigación denominada: **“DESARROLLO DE UNA APLICACIÓN WEB PARA EL CONTROL Y MONITOREO EN TIEMPO REAL QUE PERMITA MEJORAR EL SERVICIO DEL TRANSPORTE INTERNO DE LA UNIVERSIDAD NACIONAL DE CAJAMARCA”**, la cual estoy realizando.

Asimismo, anexo el cuadro de variables e indicadores para una rápida comprensión y validación de estos.

Agradeciendo anticipadamente su valiosa colaboración en el desarrollo e impulso de la investigación.

Muy cordialmente,

César Augusto Ordoñez Marín

INSTRUMENTO PARA LA VALIDACIÓN

Fecha: Junio – 2019

Apreciación cualitativa

CRITERIOS	Excelente	Bueno	Regular	Deficiente
Usabilidad (Aplicación de fácil uso e intuitivos)	X			
Fiabilidad (Aplicación no falla mientras esta en ejecución)	X			

Observaciones:

.....

.....

.....

.....

Validado por: Lester Joon De Dios Villar Zamora

Profesión: Ingeniero de Sistemas

Lugar de Trabajo: MINE SENSE SOLUTIONS SAC

Cargo: Jefe de Investigación y Desarrollo

Firma:



Cajamarca, julio 2019

Señor:

Ing. López Mego Henry

Presente

Reciba un cordial saludo

Motivado por su reconocida formación y experiencia en el **desarrollo de software**, me complace dirigirme a usted para solicitar su valiosa colaboración en la validación de la aplicación web “El carro de la U”, el mismo que servirá para recolectar información relativa a la investigación denominada: **“DESARROLLO DE UNA APLICACIÓN WEB PARA EL CONTROL Y MONITOREO EN TIEMPO REAL QUE PERMITA MEJORAR EL SERVICIO DEL TRANSPORTE INTERNO DE LA UNIVERSIDAD NACIONAL DE CAJAMARCA”**, la cual estoy realizando.

Asimismo, anexo el cuadro de variables e indicadores para una rápida comprensión y validación de estos.

Agradeciendo anticipadamente su valiosa colaboración en el desarrollo e impulso de la investigación.

Muy cordialmente,

César Augusto Ordoñez Marín

INSTRUMENTO PARA LA VALIDACIÓN

Fecha: Junio – 2019

Apreciación cualitativa

CRITERIOS	Excelente	Bueno	Regular	Deficiente
Usabilidad (Aplicación de fácil uso e intuitivos)	X			
Fiabilidad (Aplicación no falla mientras esta en ejecución)	X			

Observaciones:

.....

.....

.....

.....

Validado por: Henry López Mego

Profesión: Ingeniero de Sistemas

Lugar de Trabajo: Mine Sense Solutions

Cargo: Jefe de Investigación y Desarrollo

Firma: 