

PROTOTIPO SOFTWARE PARA LA CONSULTA DE RUTAS DE BUSES Y DE TRANSPORTE MASIVO EN EL ÁREA DE INFLUENCIA DE LAS UNIDADES TECNOLÓGICAS DE SANTANDER, BUCARAMANGA.

MODALIDAD: DESARROLLO TECNOLÓGICO

LISBETH PRADA NIÑO

CC 1.004.966.624

JUAN JOSÉ GONZÁLEZ CAPACHO

CC 1.005.260.507

**UNIDADES TECNOLÓGICAS DE SANTANDER**

**FACULTAD DE CIENCIAS NATURALES E INGENIERIAS**

**TECNOLOGÍA EN DESARROLLO DE SISTEMAS INFORMÁTICOS**

**BUCARAMANGA, 15-DICIEMBRE-2021**



PROTOTIPO SOFTWARE PARA LA CONSULTA DE RUTAS DE BUSES Y DE TRANSPORTE MASIVO EN EL ÁREA DE INFLUENCIA DE LAS UNIDADES TECNOLÓGICAS DE SANTANDER, BUCARAMANGA.

MODALIDAD: DESARROLLO TECNOLÓGICO

LISBETH PRADA NIÑO

CC 1.004.966.624

JUAN JOSÉ GONZÁLEZ CAPACHO

CC 1.005.260.507

Trabajo de Grado para optar al título de

Tecnólogo en Desarrollo de Sistemas Informáticos

**DIRECTOR**

Ing. Roberto Carvajal Salamanca

GRUPO DE INVESTIGACIÓN EN INGENIERÍA DE SOFTWARE (GRIIS)

**UNIDADES TECNOLÓGICAS DE SANTANDER**

**FACULTAD DE CIENCIAS NATURALES E INGENIERIAS**

**TECNOLOGÍA EN DESARROLLO DE SISTEMAS INFORMÁTICOS**

**BUCARAMANGA, 15-DICIEMBRE-2021**

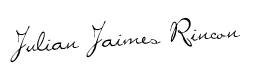
Nota de Aceptación

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Firma del Evaluador



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Firma del Director

**AGRADECIMIENTOS**

Primeramente, agradecemos a Dios por permitirnos culminar nuestro ciclo tecnológico, darnos la fortaleza para no rendirnos en el camino, también agradecemos a las Unidades Tecnológicas de Santander, a la coordinación de sistemas y especialmente a nuestro director de proyecto Roberto Carvajal Salamanca por brindarnos su apoyo en este proceso.

TABLA DE CONTENIDO

[RESUMEN EJECUTIVO 10](#_Toc90492903)

[INTRODUCCIÓN 12](#_Toc90492904)

[1. DESCRIPCIÓN DEL TRABAJO DE INVESTIGACIÓN 13](#_Toc90492905)

[1.1 PLANTEAMIENTO DEL PROBLEMA 13](#_Toc90492906)

[1.2 JUSTIFICACIÓN 14](#_Toc90492907)

[1.3 OBJETIVOS 15](#_Toc90492908)

[1.3.1 OBJETIVO GENERAL 15](#_Toc90492909)

[1.3.2 OBJETIVOS ESPECÍFICOS 15](#_Toc90492910)

[1.4 ESTADO DEL ARTE 16](#_Toc90492911)

[2 MARCO REFERENCIAL 20](#_Toc90492912)

[2.1 MARCO TEÓRICO 20](#_Toc90492913)

[2.1.1 Aplicación web. 20](#_Toc90492914)

[2.1.2 Territorios inteligentes. 21](#_Toc90492915)

[2.1.3 Base de datos. 22](#_Toc90492916)

[2.1.4 Lenguaje de programación orientado a la web. 24](#_Toc90492917)

[2.1.5 Framework. 26](#_Toc90492918)

[2.2 MARCO CONCEPTUAL 28](#_Toc90492919)

[2.2.1 Modelo vista controlador (MVC) 28](#_Toc90492920)

[2.2.2 CRUD 28](#_Toc90492921)

[2.2.3 Programación orientada a objetos 28](#_Toc90492922)

[2.2.4 Java 29](#_Toc90492923)

[2.2.5 Spring 29](#_Toc90492924)

[2.2.6 HTML 29](#_Toc90492925)

[2.3 MARCO LEGAL 29](#_Toc90492926)

[2.3.1 Ley 23 de 1982 Sobre derechos de autor 29](#_Toc90492927)

[3 DISEÑO DE LA INVESTIGACION 31](#_Toc90492928)

[3.1 DESCRIPCIÓN DE LA METOLOGÍA 31](#_Toc90492929)

[3.1.1 Fases del modelo en cascada 32](#_Toc90492930)

[4 DESARROLLO DEL TRABAJO DE GRADO 33](#_Toc90492931)

[4.1 FASE 1. ANÁLISIS. 33](#_Toc90492932)

[4.1.1 Análisis de requerimientos generales. 33](#_Toc90492933)

[4.1.2 Alcances de la aplicación 33](#_Toc90492934)

[4.1.3 Requerimiento funcionales y no funcionales. 34](#_Toc90492935)

[4.1.4 Roles del software 35](#_Toc90492936)

[4.1.5 Funcionalidad de cada rol. 36](#_Toc90492937)

[4.1.6 Selección de herramientas web. 37](#_Toc90492938)

[4.2 FASE 2. DISEÑO DEL SISTEMA. 38](#_Toc90492939)

[4.2.1 Modelo relacional de la base de datos. 38](#_Toc90492940)

[4.2.2 Arquitectura de la aplicación web. 41](#_Toc90492941)

[4.2.3 Estructura del proyecto. 42](#_Toc90492942)

[4.2.4 Descripción de los casos de uso 45](#_Toc90492943)

[4.2.5 Diagrama de secuencia. 67](#_Toc90492944)

[4.3 FASE 3. DESARROLLO DE LA APLICACIÓN 75](#_Toc90492945)

[4.3.1. Módulo de la interfaz para la visualización de contenidos 75](#_Toc90492946)

[4.4 FASE 4. PRUEBAS 86](#_Toc90492947)

[4.4.1. Prueba de funcionalidad 86](#_Toc90492948)

[4.4.2. Casos de prueba 86](#_Toc90492949)

[4.4.3. Prueba de carga 98](#_Toc90492950)

[5 RESULTADOS 100](#_Toc90492951)

[6 CONCLUSIONES 102](#_Toc90492952)

[7 RECOMENDACIONES 103](#_Toc90492953)

[8 REFERENCIAS BIBLIOGRÁFICAS 104](#_Toc90492954)

LISTA DE FIGURAS

[***Figura 1.*** Metodología en cascada 31](#_Toc90493475)

[***Figura 2*** *Modelo relacional.* 39](#_Toc90493476)

[***Figura 3.*** *Modelo Vista Controlador* 41](#_Toc90493477)

[***Figura 4.*** *Estructura del Framework Spring Tool Suite.* 42](#_Toc90493478)

[***Figura 5.*** *Estructura de los controladores* 43](#_Toc90493479)

[***Figura 6.*** *Estructura de los modelos.* 43](#_Toc90493480)

[***Figura 7.*** *Estructuras de las vistas.* 44](#_Toc90493481)

[***Figura 8.*** *Casos de uso usuario administrador.* 45](#_Toc90493482)

[***Figura 9.*** *Formulario inicio de sesión* 47](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493483)

[***Figura 10.*** *Formulario registrar frecuencia.* 49](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493484)

[***Figura 11.*** *Formulario modificar frecuencia* 49](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493485)

[***Figura 12.*** *Visualizar y eliminar frecuencia.* 50](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493486)

[***Figura 13.*** *Formulario registrar punto de recarga* 52](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493487)

[***Figura 14.*** *Formulario modificar punto de recarga* 52](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493488)

[***Figura 15.*** *Visualizar y eliminar punto de recarga* 53](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493489)

[***Figura 16.*** *Formulario registrar parada* 55](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493490)

[***Figura 17.*** *Formulario modificar parada* 55](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493491)

[***Figura 18.*** *Visualizar y eliminar parada* 56](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493492)

[***Figura 19.*** *Formulario registrar ruta* 58](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493493)

[***Figura 20.*** *Formulario modificar ruta* 58](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493494)

[***Figura 21.*** *Visualizar y eliminar ruta* 59](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493495)

[***Figura 22.*** *Formulario registrar parada de la ruta* 59](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493496)

[***Figura 23.*** *Formulario modificar parada de la ruta* 60](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493497)

[***Figura 24.*** *Visualizar y eliminar parada de la ruta* 60](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493498)

[***Figura 25.*** *Formulario registrar frecuencia de la ruta* 61](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493499)

[***Figura 26.*** *Formulario modificar frecuencia de la ruta* 61](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493500)

[***Figura 27.*** *Visualizar y eliminar frecuencia de la ruta* 62](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493501)

[***Figura 28.*** *Casos de uso usuario consultante* 63](#_Toc90493502)

[***Figura 29.*** *Consultar rutas* 64](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493503)

[***Figura 30.*** *Visualizar puntos de recarga* 65](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493504)

[***Figura 31.*** *Visualizar rutas* 66](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493505)

[***Figura 32.*** *Diagrama de secuencia iniciar sesión* 67](#_Toc90493506)

[***Figura 33.*** *Diagrama de secuencia gestionar frecuencias* 68](#_Toc90493507)

[***Figura 34.*** *Diagrama de secuencia gestionar puntos de recarga* 69](#_Toc90493508)

[***Figura 35.*** *Diagrama de secuencia gestionar paradas* 70](#_Toc90493509)

[***Figura 36.*** *Diagrama de secuencia gestionar rutas* 71](#_Toc90493510)

[***Figura 37.*** *Diagrama de secuencia consultar rutas* 72](file:///C:\Users\lisbe\Downloads\F-DC-125%20V1%20(5)%20(1).docx#_Toc90493511)

[***Figura 38.*** *Diagrama de secuencia visualizar puntos de recarga* 73](#_Toc90493512)

[***Figura 39.*** *Diagrama de secuencia visualizar rutas* 74](#_Toc90493513)

[***Figura 40.*** *Menú de navegación de la interfaz visual de contenidos* 75](#_Toc90493514)

[***Figura 41.*** *Módulo inicio de la interfaz visual* 76](#_Toc90493515)

[***Figura 42.*** *Módulo listado de rutas de la interfaz visual* 76](#_Toc90493516)

[***Figura 43.*** *Módulo listado puntos de recarga de la interfaz visual* 77](#_Toc90493517)

[***Figura 44.*** *Modulo inicio de sesión* 78](#_Toc90493518)

[***Figura 45.*** *Menú de navegación de la interfaz gráfica administrativa* 78](#_Toc90493519)

[***Figura 46.*** *Módulo inicio administrativo de la interfaz gráfica administrativa* 80](#_Toc90493520)

[***Figura 47.*** *Módulo usuarios de la interfaz gráfica administrativa* 81](#_Toc90493521)

[***Figura 48.*** *Módulo crear usuario de la interfaz gráfica administrativa* 81](#_Toc90493522)

[***Figura 49.*** *Módulo puntos de recarga de la interfaz gráfica administrativa* 82](#_Toc90493523)

[***Figura 50****. Módulo crear punto de recarga de la interfaz gráfica administrativa* 82](#_Toc90493524)

[***Figura 51.*** *Módulo paradas de la interfaz gráfica administrativa* 83](#_Toc90493525)

[***Figura 52.*** *Módulo crear parada de la interfaz gráfica administrativa* 83](#_Toc90493526)

[***Figura 53.*** *Módulo frecuencias de la interfaz gráfica administrativa* 84](#_Toc90493527)

[***Figura 54.*** *Módulo crear frecuencia de la interfaz gráfica administrativa* 84](#_Toc90493528)

[***Figura 55.*** *Módulo rutas de la interfaz gráfica administrativa* 85](#_Toc90493529)

[***Figura 56.*** *Módulo crear ruta de la interfaz gráfica administrativa* 85](#_Toc90493530)

LISTA DE TABLAS

[***Tabla 1.*** *Descripción de los tipos de la base de datos* 24](#_Toc89981245)

[***Tabla 2.*** *Descripción de los diferentes tipos de lenguajes de programación.* 27](#_Toc89981246)

[**Tabla 3.** *Requerimientos generales.* 35](#_Toc89981247)

[***Tabla 4.*** *Requerimientos funcionales.* 36](#_Toc89981248)

[***Tabla 5.*** *Requerimientos no funcionales.* 37](#_Toc89981249)

[***Tabla 6.*** *Roles del sistema* 38](#_Toc89981250)

[***Tabla 7.*** *Funcionalidad de cada rol del software* 38](#_Toc89981251)

[***Tabla 8.*** *Descripción del modelo relacional de la base de datos.* 41](#_Toc89981252)

[***Tabla 9.*** *Descripción de caso de uso: Iniciar sesión.* 51](#_Toc89981253)

[***Tabla 10.*** *Descripción caso de uso: Gestionar frecuencia* 52](#_Toc89981254)

[***Tabla 11.*** *Descripción caso de uso: Gestionar puntos de recarga* 56](#_Toc89981255)

[***Tabla 12.*** *Descripción caso de uso: Gestionar paradas* 59](#_Toc89981256)

[***Tabla 13.*** *Descripción caso de uso: Gestionar rutas* 63](#_Toc89981257)

[***Tabla 14.*** *Descripción caso de uso: Consultar rutas* 70](#_Toc89981258)

[***Tabla 15.*** *Descripción caso de uso: Visualizar puntos de recarga* 71](#_Toc89981259)

[***Tabla 16.*** *Descripción caso de uso: Visualizar rutas* 72](#_Toc89981260)

[***Tabla 17.*** *Descripción de la interfaz de visualización de contenidos* 85](#_Toc89981261)

[***Tabla 18.*** *Requerimientos generales - resultados* 98](#_Toc89981262)

[***Tabla 19.*** *Casos de prueba, Gestionar usuario* 99](#_Toc89981263)

[***Tabla 20.*** *Casos de prueba, Gestionar punto de recarga* 101](#_Toc89981264)

[***Tabla 21.*** *Casos de prueba, Gestionar parada* 103](#_Toc89981265)

[***Tabla 22.*** *Casos de prueba, Gestionar frecuencia* 105](#_Toc89981266)

[***Tabla 23.*** *Casos de prueba, Gestionar ruta* 107](#_Toc89981267)

[***Tabla 24.*** *Casos de prueba, Inicio de sesión* 109](#_Toc89981268)

[***Tabla 25.*** *Pruebas de carga* 111](#_Toc89981269)

[***Tabla 26.*** *Estado de los resultados esperados.* 113](#_Toc89981270)

# RESUMEN EJECUTIVO

Los territorios inteligentes priorizan la creación de sectores distintivos y sostenibles, en los cuales sobresale el uso de las tecnologías de la información y la comunicación (TIC), de esta manera mejorando la calidad de vida de los habitantes del sector.

Los softwares web son una herramienta que en los últimos años han traído beneficios para las personas en su vida cotidiana, ya que han permitido la automatización de procesos, además de compartir contenido de interés público, ya sea a nivel de un pequeño sector o de grandes ciudades, aportando así al surgimiento de los territorios inteligentes. Ubicándonos en el sector de las Unidades Tecnológicas de Santander (UTS), se puede observar que el número de estudiantes foráneos aumenta semestralmente, personas que se ven afectadas por la movilidad de la ciudad llegando al punto de tener inconvenientes al no saber el recorrido, las paradas y frecuencias de las diferentes rutas que se pueden encontrar en el sector de esta institución.

Teniendo en cuenta lo anterior este proyecto de grado proporciona un prototipo de software web que para todos aquellos que lo requieran y se encuentren en el área de influencia de las UTS, tengan la posibilidad de realizar consultas sobre las diferentes rutas de buses y transporte masivo, adicionalmente visualizar los puntos de recarga del transporte masivo.

Para realizar los objetivos planteados se utilizó la metodología tradicional de desarrollo en cascada y UML para el modelado del sistema, de forma que siguiendo cada una de sus fases se logra de manera efectiva los resultados esperados como los requerimientos funcionales y no funcionales, la elaboración del diseño de software y el debido desarrollo del software; logrando un aporte tecnológico al sector del área de influencia de las UTS que por medio de este software logra mejorar la movilidad.

**PALABRAS CLAVE**. Software, rutas, movilidad, UTS, territorio inteligente.

# INTRODUCCIÓN

El presente proyecto pretende compartir contenido relacionado con las rutas de los buses y transporte masivo del área de influencia de las Unidades Tecnológicas de Santander, y los puntos de recarga del transporte masivo, permitiendo también gestionar esta información por medio de un administrador, todo esto a través del desarrollo de un prototipo de software web que tiene como objetivo principal mejorar la movilidad en el sector, disminuyendo el tiempo de traslado de un lugar a otro, y mitigando el sentimiento de angustia de aquellos estudiantes foráneos que no conocen los recorridos que realizan cada uno de los buses o transporte masivo.

Cabe mencionar que la implementación de este tipo de software web es de bajo costo y las herramientas para su desarrollo no son de difícil adquisición, además de permitirnos publicar contenido que ayuda a diferentes personas en actividades de su diario vivir.

Para el desarrollo de este prototipo de software web se utilizaron herramientas como el framework Spring Tool Suite, el gestor de base de datos MySQL, y los lenguajes de programación JAVA y HTML, adicionalmente, la metodología de desarrollo de software que se utilizó fue la metodología tradicional de desarrollo en cascada, logrando así un entrono amigable, intuitivo y fácil de usar.

# 

# DESCRIPCIÓN DEL TRABAJO DE INVESTIGACIÓN

## PLANTEAMIENTO DEL PROBLEMA

En las UTS (Unidades tecnológicas de Santander) se encuentra en ejecución la propuesta de un macroproyecto con el fin de implementar un plan de desarrollo estratégico para iniciar lo que se conoce como un territorio inteligente en el área de influencia de esta institución; de este macroproyecto se deriva un enfoque hacia la movilidad puesto que ésta se ve afectada por el desconocimiento por parte de los ciudadanos sobre las rutas, paradas y horarios de los buses y transporte masivo, adicionalmente la mala gestión por parte del gobierno incrementa dicha problemática. (Carvajal Salamanca, 2021)

Semestralmente una cantidad considerable de personas que originalmente no viven en el sector de la sede principal de las UTS (Bucaramanga) se incorporan a la comunidad uteista, dando paso a que esta parte de la población se vea afectada por la problemática de la movilidad, además, la disminución de la frecuencia del transporte en ciertas horas del día también afecta de manera directa a estas personas, puesto que el transporte masivo o de buses es normalmente el más utilizado por ellas.

Según el análisis planteado, la pregunta de investigación que se formula es la siguiente:

¿El desarrollo de un prototipo de software web para la consulta de rutas de buses y transporte masivo mejorará la movilidad de las personas en el área de influencia de las Unidades tecnológicas de Santander, Bucaramanga?

## JUSTIFICACIÓN

En la actualidad la implementación de tecnologías de la información como aplicaciones móviles o software web son una de las principales herramientas para el mejoramiento de la calidad de vida de las personas puesto que facilitan los procesos que se llevan a cabo en el diario vivir, además el desarrollo de estas aplicaciones no se considera de mayor costo.

Uno de los aspectos a resolver con la realización de este proyecto es el de la movilidad de las personas en el área de influencia de las UTS, puesto que con el uso de esta aplicación se obtendrían beneficios como:

Disminución del tiempo de traslado para conveniencia de los habitantes en el sector mencionado debido a que podrán consultar las rutas y definir cuál de ellas se ajusta más a sus necesidades.

Prevenir el sentimiento de angustia en los estudiantes de nuevo ingreso en las UTS y habitantes en general de la zona, por no conocer las rutas de transporte público masivo en la zona, ya que, en el caso de los estudiantes, periódicamente esta situación se ve a inicios de cada semestre especialmente afectando a los estudiantes foráneos.

Con el desarrollo de este proyecto se genera un aporte al desarrollo de la investigación en las UTS puesto que este trabajo hace parte del macroproyecto “Implementación de las TIC en un modelo de gestión de cultura ciudadana en el área de influencia de las unidades tecnológicas de Santander” del Grupo de Investigación en Ingeniería del Software – GRIIS. Los productos que aportaría este proyecto son: formación de recurso humano (2 estudiantes de tecnología), desarrollo tecnológico (un prototipo de software) y apropiación social del conocimiento (artículo científico).

Las anteriores razones justifican el desarrollo de un prototipo de software web, el cual permita la consulta de rutas de buses y de transporte masivo en el área de influencia de las Unidades Tecnológicas de Santander, más puntualmente en la sede de Bucaramanga para el mejoramiento de la movilidad.

## OBJETIVOS

### OBJETIVO GENERAL

Realizar un prototipo de aplicación orientada a la web para la consulta de rutas de buses y de transporte masivo en el área de influencia de las Unidades Tecnológicas de Santander en Bucaramanga utilizando el lenguaje de programación HTML, la base de datos MySQL, el framework Spring Tool Suite y la metodología tradicional de desarrollo de software.

### OBJETIVOS ESPECÍFICOS

1. Determinar los requerimientos funcionales y no funcionales para la creación del prototipo de aplicación orientada a la web utilizando para ello información brindada por el director del proyecto y referencias de internet.
2. Elaborar el diseño del prototipo de aplicación orientada a la web para la consulta de rutas, paradas y horarios de buses y de transporte masivo en el área de influencia de las Unidades Tecnológicas de Santander en Bucaramanga.
3. Elaborar el prototipo de aplicación orientada a la web para para la consulta de rutas, paradas y horarios de buses y de transporte masivo en el área de influencia de las Unidades Tecnológicas de Santander en Bucaramanga utilizando para ello el lenguaje de programación HTML, el motor de base de datos MySQL, el framework Spring Tool Suite y la metodología tradicional de desarrollo de software.
4. Realizar pruebas de funcionamiento del prototipo de aplicación web - sitio web a fin de verificar su correcto funcionamiento y cumplimiento de los requisitos establecidos.

## ESTADO DEL ARTE

**¿La nueva tecnología en el sector transporte obstaculiza o fortalece el servicio público?**

De acuerdo con un artículo publicado en Verba luris, en el que se siguió una metodología cualitativa de carácter descriptivo, se determinó que el uso de aplicaciones digitales con relación al transporte público masivo produce un efecto positivo de cara al usuario puesto que este al tener la posibilidad de consultar las rutas y definir cuál de ellas será la más óptima para llegar a su destino le genera una sensación de tranquilidad, de la misma manera esta sensación la proveen herramientas de estas aplicaciones las cuales localizan y mitigan incidentes; Este conjunto de herramientas tecnológicas trabajando simultáneamente en la aplicación hacen posible que la movilidad en el uso del transporte público se realice de una manera efectiva y sostenible sin aumentar los impactos que esto conlleva. (Botero Sánchez & Alegría Paredes, 2017).

**Sistema inteligente para el monitoreo automatizado del transporte público en tiempo real.**

En 2019, Yadira Quiñonez, Carmen Lizarraga, Juan Peraza y Oscar Zatarain propusieron un desarrollo tecnológico enfocado en el transporte público de México con la idea que es el medio de transporte más usado en el mundo en los últimos años, es por esto que mediante la implementación de la tecnología se espera poder mejorar muchos aspectos del servicio para que este se adapte a las ultimas necesidades del consumidor actual.

Dicho desarrollo tecnológico tiene como propósito el monitoreo del transporte público en tiempo real por medio de GPS, servicios web y la implementación de BD (base de datos), con el objetivo de manejar un registro en el cual se puede encontrar información como la localización del vehículo, la velocidad y el cumplimiento de las rutas. (Quiñonez, Yadira; Lizarraga Carmen;Peraza, 2019).

**Sistema de localización en tiempo real mediante un servidor web y aplicaciones móviles.**

En la universidad de Guanajuato, México, Ignacio Contreras Contreras y Geovanni Hernández Gómez en el desarrollo de su proyecto de grado mediante un análisis e investigación dedujeron que los estudiantes de esta institución presentaban problemas en el tiempo de llegada para la realización de sus deberes académicos, problema que se generaba a partir de que no se conocía un horario fijo en las rutas del transporte público, por tal motivo ofrecen la implementación de una solución tecnológica basada en un aplicativo para la localización de los autobuses, cuyo objetivo general es que pueda ser utilizado por los habitantes de la zona y los estudiantes de la institución. (Contreras Contreras, Ignacio ; Hernández Gómez, 2017).

**Desarrollo de una aplicación móvil Android para la consulta de rutas de una línea de buses urbanos que circulan por la ciudad de Quito referenciando los puntos de partida y destino del usuario.**

En la sede de Quito, Ecuador de la universidad Politécnica Salesiana, se llevó a cabo como trabajo de titulación la ejecución de una app móvil a través de la metodología Scrum, app que consiste en poder buscar las rutas de manera online, mejorando la movilidad que se veía afectada a causa del desconocimiento de las rutas generado por el hecho de que hay más de 200 existentes y que la cantidad de turistas en esa zona es elevada; Finalmente los autores concluyen que la integración de esta aplicación es beneficiosa para el municipio de Quito en aspectos como: el manejo del tiempo puesto que con la existencia de la aplicación se puede llevar una mejor planificación, en relación a los gastos mensuales de pasajes, estos disminuirían ya que al elegir la ruta más efectiva usaran menos buses y una predicción positiva en la erradicación completa del problema si se gestionan todos las rutas del municipio.

(Montagut Martínez & Vargas Rodríguez, 2019).

**Transporte público inteligente al alcance de sus manos**

Juan Pablo Rojas en compañía de Julio César Bustos y Diego Ordóñez Camacho en un artículo demuestran la importancia de que las personas tengan a la mano la información precisa acerca de las rutas que el servicio de transporte público ofrece, puesto que este se encuentra a disposición de cualquiera incluso para los foráneos. Determinando que estos últimos no están familiarizados con la localización de las paradas, rutas y horarios de este servicio, propusieron un sistema para reconocer las rutas y mejorar la ubicación de los usuarios con el fin de eliminar dicha problemática. (Rojas, Juan Pablo;Bustos, Julio César;Ordóñez Camacho, 2017)

**Desarrollo de una aplicación web para consulta y visualización de rutas intermunicipales de los Municipios de la Provincia “Sabana de Occidente” del Departamento de Cundinamarca.**

El trabajo de grado basado en que la migración de los habitantes de Bogotá (Colombia) a los municipios cercanos incrementó el uso del transporte público intermunicipal causando que la movilidad se viera afectada y en consecuencia el tiempo de traslado de los municipios vecinos a Bogotá aumentara, de manera que decidieron implementar una aplicación web debido al alcance que estas proporcionan y a la vez por su velocidad de respuesta ante el usuario.

La aplicación que desarrollaron cuenta con funcionalidades como la consulta de las rutas, el registro de paraderos según la localización del usuario y la posibilidad de planear un viaje brindándole distintas posibilidades para su realización. (Montagut Martínez & Vargas Rodríguez, 2019)

# 

# MARCO REFERENCIAL

## MARCO TEÓRICO

### Aplicación web.

Se describe como un software enfocado a la web, el cual se caracteriza principalmente por permitir trabajar en línea a través del protocolo HTTP (Hypertext Transfer Protocol) quien hace las veces de intermediario entre el servidor web y el usuario, esto sin necesidad de hacer uso de un programa de escritorio, eliminando procesos de instalación debido a que para hacer uso de esta tecnología solo es necesario tener a disposición un navegador web. Además, estas aplicaciones están principalmente elaboradas con HTML (HyperText Markup Language) y tienen ventajas como fácil acceso a los datos, recopilación y almacenamiento de estos mismos. (Valarezo Pardo, Milton Rafael; Honores Tapia, Joofre Antonio; Gómez Moreno, Antonio Steeven;Vinces Sánchez, 2018).

Existen diferentes tipos de aplicación web, de las cuales podemos mencionar:

* **Aplicación web estática:** Se caracteriza principalmente por su contenido ya que este no suele ser modificado con mucha regularidad porque este proceso no es nada sencillo a causa de que para realizar una actualización se debe descargar el código fuente, realizar las modificaciones y volverlo a subir, lo cual implica que este tipo de aplicaciones no sean viables a largo plazo.
* **Aplicación web dinámica**: Su actualización es más sencilla en comparación con las aplicaciones web de tipo estática ya que estas permiten el manejo de bases de datos para alojar la información, además cuentan con la implementación de un panel administrador para realizar las modificaciones o crear nuevos contenidos. Esto permite que cada vez que los usuarios ingresan a la aplicación visualicen e interactúen con nuevos contenidos.
* **Aplicación web animada:** Este tipo de aplicación se caracteriza por diseñar e implementar contenidos con diferentes efectos animados desarrollados a través de la tecnología flash, la cual actualmente se encuentra en desuso.
* **Aplicación web con "Gestor de contenidos":** Cuenta con un CMS (sistema de gestión de contenidos) el cual permite que el manejo de los datos sea mucho más eficaz, ya que los usuarios administradores tienen la posibilidad de agregar datos y realizar modificaciones con mayor rapidez, siendo esta la razón principal por la que muchas empresas hoy en día han optado por su implementación.(Educació, 2017).

### Territorios inteligentes.

A partir del año 2000 surgió el término “territorios inteligentes” y un año más tarde este término fue resaltado por Sergio Boisier debido a la importancia que adquirió el manejo de la información a lo largo de los últimos años, esto como medida en respuesta a los nuevos desafíos que contrae la globalización junto con las dificultades que se presentan en relación con la economía, sostenibilidad y competitividad de un territorio. Estos territorios tienen el propósito de incentivar a los habitantes a trabajar en beneficio de un bien común a través del desarrollo de distintos proyectos que buscan mejorar la calidad de vida. (Quintero Pérez & Gómez Suárez, 2021).

De los territorios inteligentes o Smart Places comúnmente tienden a pensar que se trata solo de ciudades, pero en realidad se definen como cualquier sector geográfico pudiendo ser un municipio, un área de influencia, una urbanización, entre otras, que interactúa con sus habitantes en base a las necesidades que surgen de estos, almacenando y analizando la información que proviene de la interacción entre el territorio inteligente y sus habitantes.

Estos territorios se caracterizan por consolidar un sector destacado y sustentable, en el cual sobre sale el uso de las TIC como herramienta de innovación, todo esto a causa de que actualmente las personas hacen uso diario de dispositivos móviles, páginas web, redes sociales, tecnología IOT, generando así grandes volúmenes de datos que dan información, como el aumento en el consumo de estos dispositivos o herramientas, los modelos más utilizados y mejores análisis de oferta.(Carvajal Salamanca, 2021).

### Base de datos.

Se define como un conjunto de datos recopilados que se almacenan y organizan de manera lógica por medio de la utilización de programas diseñados específicamente para la manipulación y organización de estos datos, estos programas reciben el nombre de motor de base de datos que en combinación con otras aplicaciones crean lo que se denomina como un modelo de base de datos, el cual se resguarda en los equipos de forma digital. (Oracle, 2018).

Existen diferentes tipos de base de datos, en las cuales están:

***Tabla 1.*** *Descripción de los tipos de la base de datos*

|  |  |
| --- | --- |
| Tipo | Descripción |
| Bases de datos relacionales | Fueron las más utilizadas en la época de los años 80 y se caracteriza por la división de sus tablas que consiste en tener columnas y filas, además de ser eficiente al permitir un fácil acceso a los datos almacenados. |
| Base de datos orientada a objetos | Este tipo de base de datos está inspirada en el modelo de programación orientada a objetos, ya que utiliza como recurso los objetos para representar los datos almacenados en ella. |
| Base de datos distribuidas | Se caracteriza principalmente por su flexibilidad a la hora de almacenar la información debido a que los archivos que la componen pueden estar ubicados en distintos ordenadores, distinta ubicación física o diferentes redes. |
| Almacenes de datos | Su principal ventaja es la rapidez con la que permite la realización de consultas y analizar la información que se almacena. |
| Base de datos NoSQL | Estos últimos años este tipo de base de datos se han vuelto más populares puesto que en comparación con las bases de datos relacionales, este tipo difiere en la organización de los datos ya que esta si permite el ingreso de datos no estructurados o semiestructurados con la posibilidad de modificar y analizar estos mismos. |
| Base de datos orientada a grafos | Consiste en alojar los datos que mencionan la relación que se tiene entre las entidades y los datos que se relacionan con las entidades. |
| Base de datos OLTP | OLTP (On Line Transaction Processing) se caracteriza por permitir transacciones provenientes de un gran número de usuarios con una mayor rapidez. |

Fuente: (Oracle, 2018).

### Lenguaje de programación orientado a la web.

Es un grupo de normas y sintaxis que se denomina como el medio de comunicación entre la máquina y el programador actuando como un intermediario en el desarrollo de aplicaciones web, esto a través de la especificación de datos y acciones ingresadas por el programador haciendo uso de un lenguaje entendible por el mismo, el cual para poder ser leído por el equipo pasa por un proceso de transformación a lenguaje máquina con la ayuda de un compilador de datos.

Los lenguajes de programación se dividen en tres tipos:

* **Lenguaje de máquina:** Es complejo ya que necesita de otro para ser entendible por el humano, pero para los equipos son órdenes directas.
* **Lenguaje de bajo nivel:** Hace el trabajo de traductor entre el lenguaje de máquina y el de alto nivel.
* **Lenguaje de alto nivel:** Generalmente está escrito en palabras comunes al lenguaje humano el cual se encarga de dar instrucciones a la máquina.

(Roman Arenaza, 2019).

Actualmente existen diversos lenguajes de programación orientados a la web, de los cuales se puede mencionar:

***Tabla 2.*** *Descripción de los diferentes tipos de lenguajes de programación.*

|  |  |
| --- | --- |
| Lenguaje | Descripción |
| Python | Es un lenguaje orientado a objetos fácil de aprender que se caracteriza por su sintaxis, la cual permite el desarrollo de aplicaciones con un menor número de líneas de código en comparación con otros lenguajes de programación, ahorrando tiempo en el desarrollo de las aplicaciones. |
| PHP | Hypertext Preprocessor por sus siglas en inglés es un lenguaje de programación que se enfoca en el desarrollo de aplicaciones web con distintos contenidos dinámicos, el cual fue adecuado para trabajar en conjunto con HTML. |
| JavaScript | Es uno de los lenguajes de interpretación más utilizados debido a que posee una sintaxis parecida a los lenguajes de programación Java y C, del cual se especifica que no requiere de compilar su código para su ejecución ya que se puede realizar desde un navegador, es decir, del lado del cliente. |
| C# | Es un lenguaje relativamente nuevo que reúne la mejor parte de algunos lenguajes de programación como lo son Visual Basic, Java y C++ para el desarrollo de proyectos. |

Fuente: (Valarezo Pardo, Milton Rafael; Honores Tapia, Joofre Antonio; Gómez Moreno, Antonio Steeven;Vinces Sánchez, 2018)

### Framework.

Framework se define como la integración de diversas tecnologías que trabajan en conjunto para la creación de un entorno de desarrollo, el cual brinda una modalidad de trabajo ágil, ya que facilita diversas actividades a través de la reutilización del código, haciendo uso de la arquitectura modelo vista controlador. (Constanzo, Marcela A; Casas, 2018).

Algunos de los Framework más usados actualmente son:

**Django:** Es un Framework diseñado para el desarrollo de aplicaciones web, el cual se fundamenta en el lenguaje de programación Python,además este lo caracteriza la rapidez en la que se logra culminar el desarrollo de proyectos debido a que facilita código, por otro lado, la seguridad proporciona información para que los desarrolladores eviten los errores comunes y por último la escalabilidad que permite alto flujo de movimientos en las aplicaciones.(Foundation, 2021).

**Angular:** Es un Framework diseñado e implementado por Google dirigido al desarrollo con JavaScript que utiliza la arquitectura Modelo Vista Controlador. Sus principales características son:

**Velocidad y rendimiento**

* Genera código de forma automática en base de los requerimientos del programador.
* Permite ejecutar la vista en diferentes servidores como lo pueden ser node.js, .Net, PHP, etc.
* Su enrutador de componentes le permite cargar los proyectos de forma muy rápida, debido a que solo carga lo necesario en el momento.

**Productividad**

* Dispone gran cantidad de plantillas para los desarrolladores.
* Mediante líneas de comandos permite realizar pruebas, previsualizar el desarrollo y agregar complementos.
* Proporciona un Feedback al desarrollador, mostrando información de interés a cerca del estado del código.

**Historia completa del desarrollo**

* Permite realizar testing de la aplicación mediante Karma.
* Da la posibilidad de diseñar e implementar animaciones tan complejas como se desee.
* Permite la creación de proyectos web de forma accesible.

(Google, 2021).

**Zend:** Esta enfocado en PHP, por lo tanto, es flexible en cuanto a las versiones de este lenguaje que se desee utilizar, además en cuanto a seguridad, actualiza continuamente para traer los respectivos parches de seguridad, también proporciona información detallada a cerca de los errores encontrados en el código para beneficiar a los programadores en la corrección de los mismos y finalmente brinda soporte a largo plazo. (Perforce Software, 2021).

## MARCO CONCEPTUAL

### Modelo vista controlador (MVC)

Se define como una arquitectura de desarrollo de software enfocada en la web la cual se encarga de separar los diferentes elementos de una aplicación para tratarlos de forma independientes, tales elementos son el modelo que se encarga de todo el manejo de datos, es decir, de obtener y actualizar los mismos, la vista es la encargada de producir la visualización hacia el usuario, generalmente solo encontramos código HTML en estas y los controladores es el encargado de gestionar todos los métodos que la aplicación necesita para funcionar. (Orozco Aguirre, 2019).

### CRUD

Por sus siglas en inglés que hacen referencia a Create, Read, Update, Delete respectivamente, son las acciones que se pueden realizar sobre la información almacenada en una base de datos, más concretamente las acciones de crear nuevos registros, visualizar los registros, modificar dichos registros y eliminarlos complemente de la base de datos. (Contributors, 2021a).

### Programación orientada a objetos

Conocida como POO por sus siglas, la programación orientada a objetos es un paradigma de programación el cual muchos de los lenguajes que actualmente están en auge adoptaron debido a que se basa en unificar los atributos y métodos que se ejecutan en los datos de un programa, al realizar dicha unificación este conjunto de información se define como objeto, en el cual los atributos son las características del objeto y los métodos son las acciones que este puede realizar. (Martínez Canelo, 2020).

### Java

Considerado como el estándar en los lenguajes de programación por varios años, Java se caracteriza por compilar el código mediante el uso de una máquina virtual y por su sintaxis ya que es simular a la de C y C++. (Valarezo Pardo, Milton Rafael; Honores Tapia, Joofre Antonio; Gómez Moreno, Antonio Steeven;Vinces Sánchez, 2018).

### Spring

Es un Framework enfocado en el lenguaje de programación Java que se caracteriza por su manejo de bibliotecas flexibles ya sean internas o externas con el propósito de ofrecer una buena experiencia para el usuario final, también implementa la inyección de dependencias y la inversión de control para proporcionar distintas funcionalidades, su trabajo lo hace a través de un servidor web integrado y tiene gran seguridad para corregir y encontrar cualquiera de las vulnerabilidades del sistema que se reporten. (SpringSource, 2021).

### HTML

Se define como un lenguaje de marcas de Hipertexto razón por la cual no es considerado un lenguaje de programación, además está desarrollado exclusivamente para la creación de contenido web a través del uso de etiquetas que categorizan cada uno de los atributos que existen, brindando la posibilidad de visualizarlo haciendo uso únicamente de un navegador web. (Contributors, 2021b).

## MARCO LEGAL

### Ley 23 de 1982 Sobre derechos de autor

Ley de la cual es preciso mencionar los siguientes artículos:

**Artículo 1°** Los autores de obras literarias, científicas y artísticas gozarán de protección para sus obras en la forma prescrita por la presente Ley y, en cuanto fuere compatible con ella, por el derecho común. También protege esta Ley a los intérpretes o ejecutantes, a los productores de programas y a los organismos de radiodifusión, en sus derechos conexos a los del autor.

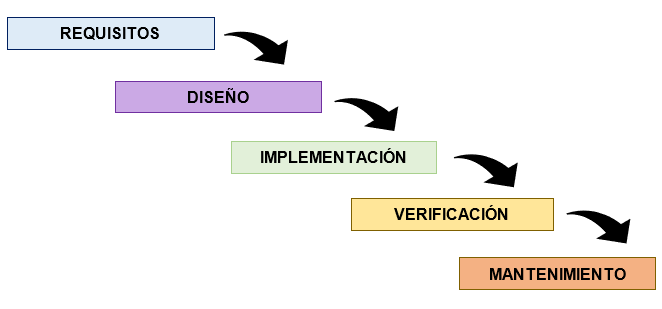
**Artículo 15º** El que con permiso expreso del autor o de sus causahabientes adapta, transporta, modifica, extracta, compendia o parodia, una obra del dominio privado, es titular del derecho de autor sobre su adaptación, transporte, modificación, extracto, compendio o parodia, pero salvo convención en contrario, no podrá darle publicidad sin mencionar el título de la obra originaria y su autor. (REPÚBLICA, 1982).

# DISEÑO DE LA INVESTIGACION

En base a los objetivos planteados para el desarrollo del proyecto, se seleccionó la metodología tradicional de desarrollo de software conocida como metodología en cascada la cual se caracteriza por estar dividida en fases secuenciales, es decir, no se empieza una fase sin haber terminado la anterior.

## DESCRIPCIÓN DE LA METOLOGÍA

La siguiente figura evidencia el proceso que sigue la metodología de desarrollo en cascada:

***Figura 1.*** Metodología en cascada

Fuente: Autores del proyecto

En el año 1970 Winston W. Royce diseñó la metodología en cascada que hoy en día es conocida como la metodología tradicional de desarrollo de software y se convirtió en una de las más utilizadas en este ámbito debido a su enfoque secuencial y estructural, el cual consta de un grupo de fases de desarrollo, donde se debe culminar completamente una fase para poder avanzar a la siguiente. (Zumba Gamboa, 2018).

### Fases del modelo en cascada

A medida que pasaban los años, el modelo en cascada sufrió algunos cambios en función a las necesidades del usuario, por esta esta razón se pueden encontrar diferentes fases dependiendo del autor, pero existen 5 fases que se encuentran en la mayoría de los casos, las cuales se explicarán a continuación:

* **Requisitos:** Se centra en definir los servicios, limitaciones y propósito general del software a construir, con el fin de establecer de forma detallada, las funciones principales y secundarias que tendrán cada tipo de usuario. (Delgado Olivera & Diaz Alonso, 2021).
* **Diseño:** Busca estructurar el software de manera que se pueda definir la arquitectura, junto con la organización de los datos y detalles de la interfaz. (González González & Calero Castañeda, 2019).
* **Implementación:** Se pone en marcha lo establecido en fases anteriores, ya que se construyen los requerimientos de manera lógica, en función de la arquitectura seleccionada (Delgado Olivera & Diaz Alonso, 2021).
* **Verificación:** Su objetivo principal es comprobar que todos los requerimientos construidos en la fase anterior cumplan correctamente con el propósito establecido en la fase de requerimientos. (González González & Calero Castañeda, 2019).
* **Mantenimiento:** Es la encargada de llegado el caso, modificar el producto final, ya que se pone a prueba el software en busca de errores para su posterior corrección, además, se trata de identificar posibles vacíos en el software, los cuales se tratarán de solucionar por medio de nuevos requisitos.

(Delgado Olivera & Diaz Alonso, 2021)**.**

# DESARROLLO DEL TRABAJO DE GRADO

## FASE 1. ANÁLISIS.

### Análisis de requerimientos generales.

En el desarrollo de esta fase, se determinaron los requerimientos principales con los que contará el prototipo de software con el fin de dar una vista general del funcionamiento, además de esto se definirán los requerimientos funcionales y no funcionales.

**Tabla 3.** *Requerimientos generales.*

|  |
| --- |
| Requerimientos generales |
| Ingresar al prototipo de software |
| Gestionar información |
| Búsqueda de rutas |
| Visualizar contenido de las rutas |

Fuente: Autores del proyecto.

### Alcances de la aplicación

A continuación, se describirán los alcances que tendrá el prototipo de software:

* El prototipo de software debe permitir la visualización de las rutas en el área de influencia de las Unidades Tecnológicas de Santander, Bucaramanga.
* El prototipo de software debe permitir la visualización de las paradas de las diferentes rutas de transporte masivo en el área de influencia de las Unidades Tecnológicas de Santander, Bucaramanga.
* El prototipo de software debe permitir la visualización de los puntos de recarga de transporte masivo en el área de influencia de las Unidades Tecnológicas de Santander, Bucaramanga.
* El prototipo de software debe contar con una sección exclusiva para el administrador que le permita gestionar toda la información relacionada con las rutas, las paradas, la frecuencia y puntos de recarga de los buses y transporte masivo en el área de influencia de las Unidades Tecnológicas de Santander, Bucaramanga, además de esto el administrador podrá gestionar la información con respecto a los usuarios.

### Requerimiento funcionales y no funcionales.

Los requerimientos funcionales que se determinaron para el prototipo de software son:

***Tabla 4.*** *Requerimientos funcionales.*

|  |  |
| --- | --- |
| Requerimiento | Descripción |
| Inicio de sesión | La aplicación web debe permitir el acceso al administrador, mediante una autenticación usuario y contraseña. |
| Gestión de rutas | La aplicación web debe permitir el registro, visualización, modificación o eliminación de rutas de buses y Metrolínea. |
| Gestión de paradas | La aplicación web debe permitir el registro, visualización, modificación o eliminación de paradas de buses y Metrolínea. |
| Gestión de frecuencias | La aplicación web debe permitir el registro, visualización, modificación o eliminación de frecuencias de rutas de buses y de Metrolínea |
| Gestión de Puntos de recarga | La aplicación web debe permitir el registro, visualización, modificación o eliminación de puntos de recarga de Metrolínea |
| Visualización de Rutas | La aplicación web debe visualizar rutas con su frecuencia, paradas y recorrido. |
| Consulta de rutas por palabra clave | La aplicación web debe permitir buscar las rutas por medio de palabras clave. |
| Visualización puntos de recarga Metrolínea | La aplicación web debe permitir visualizar los puntos de recarga. |

Fuente: Autores del proyecto.

Los requerimientos no funcionales se describen de la siguiente manera:

***Tabla 5.*** *Requerimientos no funcionales.*

|  |  |
| --- | --- |
| Requerimiento | Descripción |
| Seguridad | El acceso del usuario administrativo a la aplicación web estará sujeta a un usuario y una contraseña cifrada. |
| Usabilidad | La aplicación web estará diseñada con interfaz amigable y facilidad de uso. |
| Rendimiento | El análisis y proceso de los datos y la información será rápido. |
| Adaptación | La aplicación web se adaptará a un sistema operativo Windows 8 en adelante. |
| Adquisición | Para adquirir la aplicación web, no necesita ninguna licencia. |

Fuente: Autores del proyecto.

### Roles del software

El prototipo de software web tiene dos diferentes usuarios, cada uno definido en base a los requerimientos previamente establecidos, estos usuarios son:

***Tabla 6.*** *Roles del sistema*

|  |  |  |
| --- | --- | --- |
| Código | Rol | Descripción |
| 1 | Administrador | Debe ingresar con su respectivo usuario y contraseña que una vez verificados le permitirán registrar, visualizar, modificar y eliminar cada uno de las secciones del software. |
| 2 | Consultante | No necesitan de una autenticación de credenciales, ya que solo puede visualizar la información relacionada con las rutas y los puntos de recarga. |

Fuente: Autores del proyecto.

### Funcionalidad de cada rol.

A continuación, se mostrarán las funcionalidades que tiene cada rol del sistema:

***Tabla 7.*** *Funcionalidad de cada rol del software*

|  |  |
| --- | --- |
| Rol | Funcionalidad |
| Administrador | Crear:   * Rutas * Frecuencia * Paradas * Puntos de recarga * Usuarios   Visualizar:   * Rutas * Frecuencia * Paradas * Puntos de recarga * Usuarios   Modificar:   * Rutas * Frecuencia * Paradas * Puntos de recarga * Usuarios   Eliminar:   * Rutas * Frecuencia * Paradas * Puntos de recarga * Usuarios |
| Consultante | Consultar:   * Rutas   Visualizar:   * Rutas * Puntos de recarga |

Fuente: Autores del proyecto.

### Selección de herramientas web.

Para el desarrollo del prototipo de software web se escogió en base a la fase de análisis y el tipo de licencia que manejan. A continuación, se dará a conocer las razones por las que se escogieron dichas herramientas:

* **Framework Spring Tool Suite:** Esta herramienta se seleccionó debido a quetrabaja con el lenguaje de programación Java, por otro lado, permite la inyección de dependencias lo cual brinda gran variedad de complementos a la hora de trabajar, como lo puede ser Thymeleaf y el entorno de desarrollo Spring Boot MVC, además de esto, posee un servidor web integrado y tiene gran seguridad para corregir y encontrar cualquiera de las vulnerabilidades del sistema que se reporten.
* **Lenguaje de programación HTML:** HyperText Markup Language se escogió debido a que es considerado una de las bases fundamentales para el desarrollo web y además se pueden incorporar diferentes tecnologías de desarrollo que trabajen en conjunto con este, como lo son CSS y JavaScript.
* **Base de datos MySQL:** Se escogió este motor de base de datos ya que trabaja base de datos relacionales, además de lo anterior MySQL se enfoca en el desarrollo de aplicaciones web y permite la visualización de información en tiempo real.

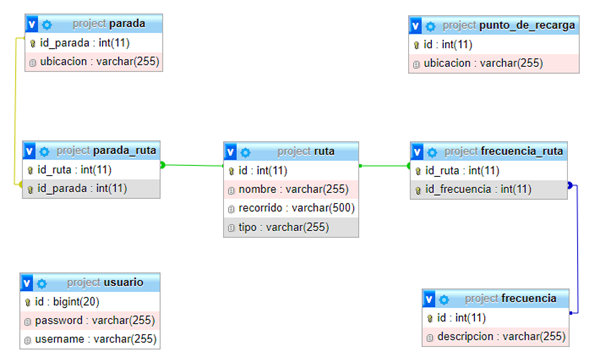
## FASE 2. DISEÑO DEL SISTEMA.

A continuación, se describen los aspectos relacionados con el diseño del prototipo de software, tales aspectos son el modelo entidad relación, los casos de uso, los diagramas de secuencia del mismo.

### Modelo relacional de la base de datos.

Un modelo relacional de una base de datos es aquel que permite visualizar de forma gráfica la estructura de sus tablas y de ellas sus columnas y filas.

El modelo relacional que se elaboró para este prototipo de software se compone de siete (7) tablas, en las cuales se asignó su respectiva llave primaria como identificador único de cada una de ellas, además se asignaron llaves foráneas en los casos que se consideraron pertinentes.

***Figura 2*** *Modelo relacional.*

Fuente: Autores del proyecto.

***Tabla 8.*** *Descripción del modelo relacional de la base de datos.*

|  |  |  |  |
| --- | --- | --- | --- |
| Tabla | Descripción | Campos | Llave  primaria |
| parada | Se almacena la información relacionada con las paradas. | * id\_parada   int (11)   * ubicación   varchar (255) | * id\_parada |
| ruta | Se almacena la información relacionada con las rutas. | * id   int (11)   * nombre   varchar (255)   * recorrido   varchar (500)   * tipo   varchar (255) | * id |
| frecuencia | Se almacena una breve descripción de las diferentes frecuencias que tiene una ruta | * id   int (11)   * descripción varchar (255) | * id |
| parada\_ruta | Se almacena el registro que se genera entre una ruta y sus diferentes paradas | * id\_ruta   int (11)   * id\_parada   int (11) | * id\_parada * id\_ruta |
| frecuencia\_ruta | Se almacena el registro que se genera entre una ruta y sus diferentes frecuencias | * id\_ruta   int (11)   * id\_frecuencia   int (11) | * id\_ruta * id\_frecuencia |
| usuario | Se almacena las credenciales del usuario administrador | * id   bigint (20)   * username   varchar (255)   * password   varchar (255) | * id |
| punto\_de\_recarga | Se almacena la información necesaria para gestionar los puntos de recarga del transporte masivo. | * id   int (11)   * ubicación   varchar (255) | * id |

Fuente: Autores del proyecto.

### Arquitectura de la aplicación web.

La arquitectura del prototipo de software web está dada por el Framework Spring Tool Suite el cual permite la implementación de la arquitectura MVC (Modelo Vista Controlador). Esta arquitectura estructura los componentes de un proyecto para manejarlos de manera individual, estos componentes son:

* **El modelo:** Gestiona todo lo relacionados con los datos, permitiendo la modificación u obtención de los mismos.
* **La vista:** Se traduce como la interfaz que puede visualizar el usuario.
* **El controlador:** Ejecuta las acciones requeridas para el buen funcionamiento del software web. (Orozco Aguirre, 2019).

Teniendo en cuenta lo anterior, la arquitectura MVC se visualiza de la siguiente forma:

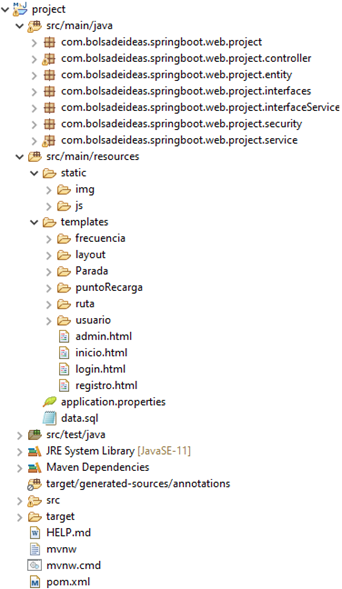
***Figura 3.*** *Modelo Vista Controlador*

Fuente: (Smith, 2021)

### Estructura del proyecto.

La siguiente figura muestra la estructura completa de los paquetes y carpetas del prototipo en el framework Spring Tool Suite:

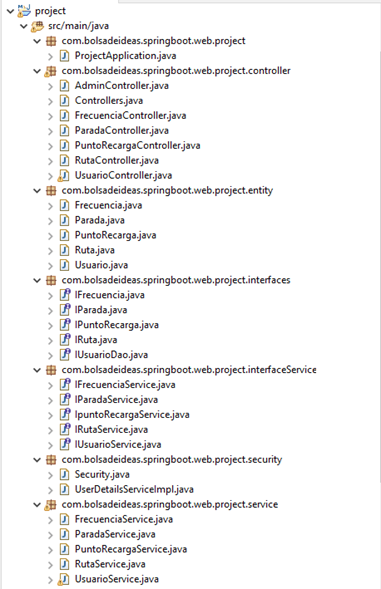
***Figura 4.*** *Estructura del Framework Spring Tool Suite.*



Fuente: Autores del proyecto

La siguiente figura muestra la estructura de los controladores del prototipo en el framework Spring Tool Suite:

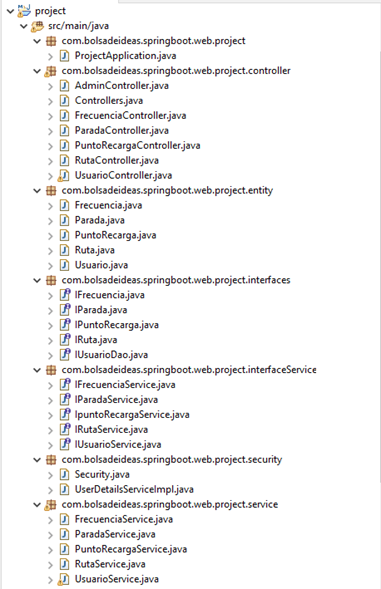
***Figura 5.*** *Estructura de los controladores*



Fuente: Autores del proyecto

La siguiente figura muestra la estructura de los modelos del prototipo en el framework Spring Tool Suite:

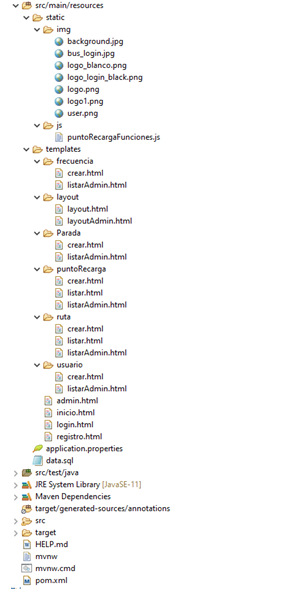
***Figura 6.*** *Estructura de los modelos.*



Fuente: Autores del proyecto

La siguiente figura muestra la estructura de las vistas del prototipo en el framework Spring Tool Suite:

***Figura 7.*** *Estructuras de las vistas.*

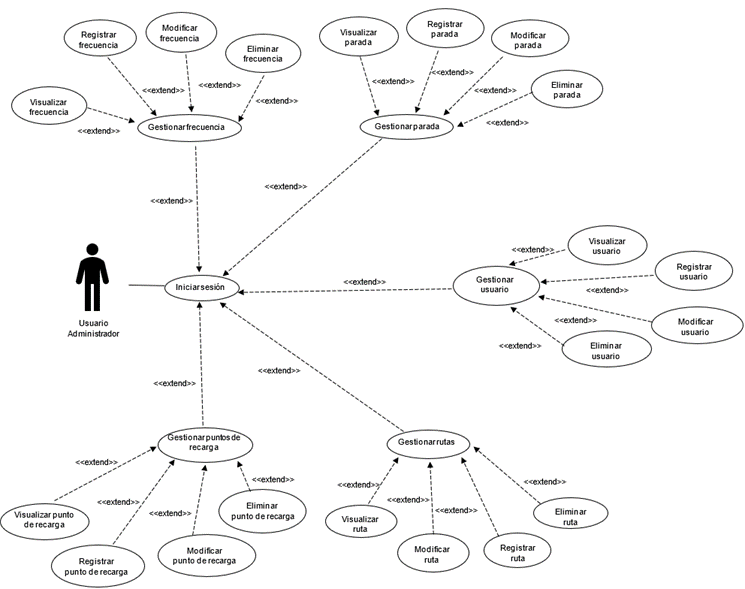


Fuente: Autores del proyecto

### Descripción de los casos de uso

#### Descripción de los casos de uso para rol administrador.

La siguiente figura evidencia todos los casos de uso posibles en el prototipo para el rol de administrador:

***Figura 8.*** *Casos de uso usuario administrador.*

Fuente: Autores del proyecto

***Tabla 9.*** *Descripción de caso de uso: Iniciar sesión.*

|  |  |
| --- | --- |
| Iniciar sesión | |
| Ítem | **Definición** |
| Casos de uso | Iniciar sesión |
| Descripción | El usuario administrador puede ingresar al sistema con un usuario y contraseña previamente establecido, los cuales respectivamente serán verificados por el sistema para permitir su acceso |
| Actor | Usuario administrador |
| Precondición | El usuario y contraseña deben estar creados previamente en la base de datos |
| Acciones | **Secuencia:**  **1.** Ingreso al software  **2.** Ingreso del usuario y contraseña solicitados por el software  **3.** Validación de los datos  **4.** Acceso al software administrativo |
| Postcondición | Validación de los datos de acceso correctamente |
| Excepciones | Si los datos de acceso no se encuentran registrados en la base de datos, el software no permitirá el acceso |
| Ejemplo de interfaz | |
| La siguiente figura muestra una posible interfaz para el módulo de iniciar sesión:  ***Figura 9.*** *Formulario inicio de sesión* | |

Fuente: Autores del proyecto

***Tabla 10.*** *Descripción caso de uso: Gestionar frecuencia*

|  |  |
| --- | --- |
| Gestionar frecuencia | |
| Ítem | **Definición** |
| Casos de uso | Gestionar frecuencia |
| Descripción | El usuario administrador tendrá permisos para registrar, modificar, eliminar y visualizar la frecuencia que puede tener una ruta |
| Actor | Usuario administrador |
| Precondición | Debe haber ingresado al software |
| Acciones | **Secuencia:**   1. Añadir frecuencia    1. Digitar los datos requeridos    2. Guardar la frecuencia 2. Buscar frecuencia 3. Visualizar frecuencias 4. Modificar frecuencia    1. Seleccionar la frecuencia requerida    2. Modificar los datos de la frecuencia    3. Guardar los datos modificados 5. Eliminar frecuencia    1. Seleccionar la frecuencia requerida    2. Eliminar los datos de la frecuencia |
| Postcondición | En limitación con la acción solicitada:   1. Ver todas las frecuencias 2. Modificar las frecuencias 3. Eliminar las frecuencias 4. Guardar los datos de las nuevas frecuencias |
| Excepciones | 1. Si el administrador no tiene sus credenciales correctas, no podrá realizar cambios. 2. Si la frecuencia que desea borrar está actualmente en uso por una ruta, esta no se podrá eliminar. |
| Ejemplo de interfaz | |
| Las siguientes figuras muestran una posible interfaz para el módulo de frecuencias:  ***Figura 10.*** *Formulario registrar frecuencia.*  Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamente  ***Figura 11.*** *Formulario modificar frecuencia*  Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamente  ***Figura 12.*** *Visualizar y eliminar frecuencia.*  Interfaz de usuario gráfica, Aplicación, Sitio web  Descripción generada automáticamente | |

Fuente: Autores del proyecto

***Tabla 11.*** *Descripción caso de uso: Gestionar puntos de recarga*

|  |  |
| --- | --- |
| Gestionar puntos de recarga | |
| Ítem | **Definición** |
| Casos de uso | Gestionar puntos de recarga |
| Descripción | El usuario administrador tendrá permisos para registrar, modificar, eliminar y visualizar los puntos de recarga. |
| Actor | Usuario administrador |
| Precondición | 1. Debe haber ingresado al software 2. Para la acción de modificar, eliminar o visualizar debe existir al menos un registro de punto de recarga. |
| Acciones | **Secuencia:**   1. Añadir puntos de recarga    1. Digitar los datos requeridos    2. Guardar los datos 2. Buscar puntos de recarga 3. Visualizar puntos de recarga 4. Modificar puntos de recarga    1. Seleccionar el punto de recarga requerido    2. Modificar los datos    3. Guardar los datos modificados 5. Eliminar puntos de recarga    1. Seleccionar el punto de recarga requerido    2. Eliminar los datos |
| Postcondición | En limitación con la acción solicitada:   1. Ver todos los puntos de recarga 2. Modificar los puntos de recarga 3. Eliminar los puntos de recarga 4. Guardar los datos de los nuevos puntos de recarga |
| Excepciones | 1. Si el administrador no tiene sus credenciales correctas, no podrá realizar cambios o agregar nuevos registros. |
| Ejemplo de interfaz | |
| Las siguientes figuras muestran una posible interfaz para el módulo de puntos de recarga:    ***Figura 13.*** *Formulario registrar punto de recarga*  ***Figura 14.*** *Formulario modificar punto de recarga*    ***Figura 15.*** *Visualizar y eliminar punto de recarga* | |

Fuente: Autores del proyecto

***Tabla 12.*** *Descripción caso de uso: Gestionar paradas*

|  |  |
| --- | --- |
| Gestionar paradas | |
| Ítem | **Definición** |
| Casos de uso | Gestionar paradas |
| Descripción | El usuario administrador tendrá permisos para registrar, modificar, eliminar y visualizar las paradas que pueden tener las rutas. |
| Actor | Usuario administrador |
| Precondición | 1. Debe haber ingresado al software 2. Para la acción de modificar, eliminar o visualizar debe existir al menos un registro de parada |
| Acciones | **Secuencia:**   1. Añadir parada    1. Digitar los datos requeridos    2. Guardar los datos de la parada 2. Buscar parada 3. Visualizar parada 4. Modificar parada    1. Seleccionar la parada requerida    2. Modificar los datos de la parada    3. Guardar los datos modificados 5. Eliminar parada    1. Seleccionar la parada requerida    2. Eliminar los datos de la parada |
| Postcondición | En limitación con la acción solicitada:   1. Ver todas las paradas 2. Modificar las paradas 3. Eliminar las paradas 4. Guardar los datos de las nuevas paradas |
| Excepciones | 1. Si el administrador no tiene sus credenciales correctas, no podrá realizar cambios. 2. Si la parada que desea borrar está actualmente en uso por una ruta, esta no se podrá eliminar. |
| Ejemplo de interfaz | |
| Las siguientes figuras muestran una posible interfaz para el módulo de paradas:  ***Figura 16.*** *Formulario registrar parada*  Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamente  ***Figura 17.*** *Formulario modificar parada*  Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamente  ***Figura 18.*** *Visualizar y eliminar parada*  Interfaz de usuario gráfica, Aplicación, Sitio web  Descripción generada automáticamente | |

Fuente: Autores del proyecto

***Tabla 13.*** *Descripción caso de uso: Gestionar rutas*

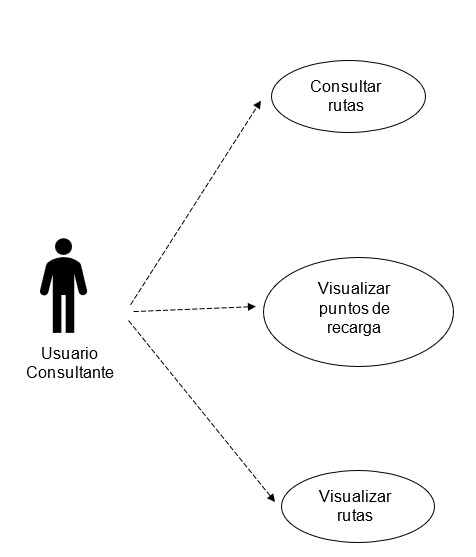
|  |  |
| --- | --- |
| Gestionar rutas | |
| Ítem | **Definición** |
| Casos de uso | Gestionar rutas |
| Descripción | El usuario administrador tendrá permisos para registrar, modificar, eliminar y visualizar las rutas de los buses y transporte masivo, además, podrá añadir frecuencias y paradas a las mismas. |
| Actor | Usuario administrador |
| Precondición | 1. Debe haber ingresado al software 2. Para añadir una parada a la ruta, debe existir al menos un registro en la tabla parada. 3. Para añadir una frecuencia a la ruta, debe existir al menos un registro en la tabla frecuencia. |
| Acciones | **Secuencia:**   1. Añadir ruta    1. Digitar los datos requeridos    2. Guardar los datos de la ruta 2. Buscar ruta 3. Visualizar ruta 4. Modificar ruta    1. Seleccionar la ruta requerida    2. Modificar los datos de la ruta    3. Guardar los datos modificados 5. Eliminar ruta    1. Seleccionar la ruta requerida    2. Eliminar los datos de la ruta |
| Postcondición | En limitación con la acción solicitada:   1. Ver todas las rutas 2. Modificar las rutas 3. Eliminar las rutas 4. Guardar los datos de las nuevas rutas |
| Excepciones | 1. Si el administrador no tiene sus credenciales correctas, no podrá realizar cambios. |
| Ejemplo de interfaz | |
| Las siguientes figuras muestran una posible interfaz para el módulo de rutas:  ***Figura 19.*** *Formulario registrar ruta*  Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamente  ***Figura 20.*** *Formulario modificar ruta*  Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamente  ***Figura 21.*** *Visualizar y eliminar ruta*  Interfaz de usuario gráfica, Texto, Aplicación  Descripción generada automáticamente  ***Figura 22.*** *Formulario registrar parada de la ruta*  Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamente  ***Figura 23.*** *Formulario modificar parada de la ruta*  Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamente  ***Figura 24.*** *Visualizar y eliminar parada de la ruta*  Interfaz de usuario gráfica  Descripción generada automáticamente con confianza baja  ***Figura 25.*** *Formulario registrar frecuencia de la ruta*  Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamente  ***Figura 26.*** *Formulario modificar frecuencia de la ruta*  Interfaz de usuario gráfica, Aplicación  Descripción generada automáticamente  ***Figura 27.*** *Visualizar y eliminar frecuencia de la ruta*  Captura de pantalla de un celular  Descripción generada automáticamente | |

Fuente: Autores del proyecto

#### Descripción de los casos de uso para rol consultante.

La siguiente figura evidencia todos los casos de uso posibles en el prototipo para el rol de consultante:

***Figura 28.*** *Casos de uso usuario consultante*



Fuente: Autores del proyecto

***Tabla 14.*** *Descripción caso de uso: Consultar rutas*

|  |  |
| --- | --- |
| Consultar rutas | |
| Ítem | **Definición** |
| Casos de uso | Consultar rutas |
| Descripción | El usuario consultante podrá buscar las rutas de los buses o transporte masivo por medio de palabras claves |
| Actor | Usuario consultante |
| Precondición | 1. Debe digitar las palabras claves para realizar la consulta de búsqueda 2. Debe existir al menos un registro de ruta, parada y frecuencia. |
| Acciones | **Secuencia:**  Buscar rutas por medio de palabras claves |
| Postcondición | Ver todas las rutas con sus respectivas paradas y frecuencias |
| Ejemplo de interfaz | |
| La siguiente figura muestra una posible interfaz para el buscador:  ***Figura 29.*** *Consultar rutas*  Captura de pantalla de un celular  Descripción generada automáticamente | |

Fuente: Autores del proyecto

***Tabla 15.*** *Descripción caso de uso: Visualizar puntos de recarga*

|  |  |
| --- | --- |
| Visualizar puntos de recarga | |
| Ítem | **Definición** |
| Casos de uso | Visualizar puntos de recarga |
| Descripción | El usuario consultante podrá visualizar los puntos de recarga del transporte masivo |
| Actor | Usuario consultante |
| Precondición | Debe existir al menos un registro de punto de recarga |
| Acciones | **Secuencia:**  Visualizar puntos de recarga |
| Postcondición | Ver todos los puntos de recarga |
| Ejemplo de interfaz | |
| La siguiente figura muestra una posible interfaz para el módulo de listado puntos de recarga:  Interfaz de usuario gráfica, Texto, Aplicación  Descripción generada automáticamente  ***Figura 30.*** *Visualizar puntos de recarga* | |

Fuente: Autores del proyecto

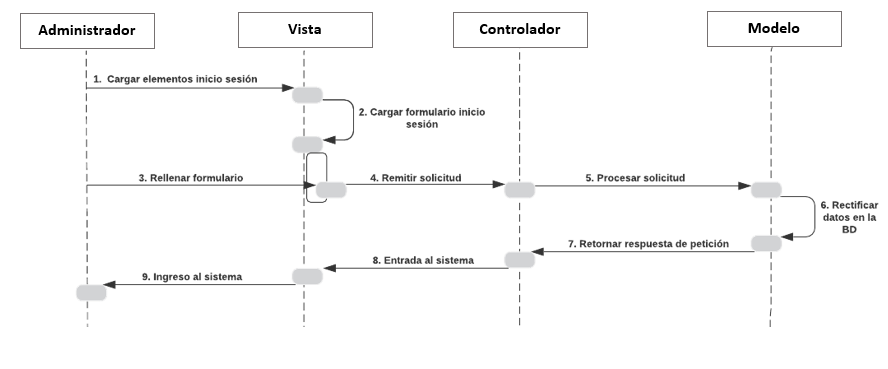
***Tabla 16.*** *Descripción caso de uso: Visualizar rutas*

|  |  |
| --- | --- |
| Visualizar rutas | |
| Ítem | **Definición** |
| Casos de uso | Visualizar rutas |
| Descripción | El usuario consultante podrá visualizar las rutas de los buses y transporte masivo |
| Actor | Usuario consultante |
| Precondición | Debe existir al menos un registro ruta, parada y frecuencia |
| Acciones | **Secuencia:**  Visualizar rutas |
| Postcondición | Ver todas las rutas |
| Ejemplo de interfaz | |
| La siguiente figura muestra una posible interfaz para el módulo de listado de rutas:  Interfaz de usuario gráfica  Descripción generada automáticamente  ***Figura 31.*** *Visualizar rutas* | |

Fuente: Autores del proyecto

### Diagrama de secuencia.

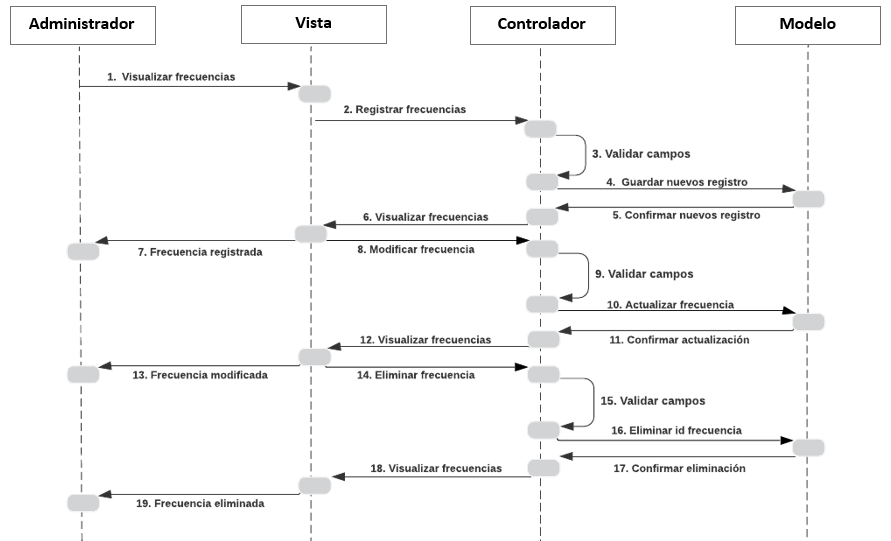
La siguiente figura evidencia el proceso que se ejecuta al iniciar sesión:

***Figura 32.*** *Diagrama de secuencia iniciar sesión*

Fuente: Autores del proyecto

La siguiente figura evidencia el proceso que se ejecuta al gestionar frecuencias:

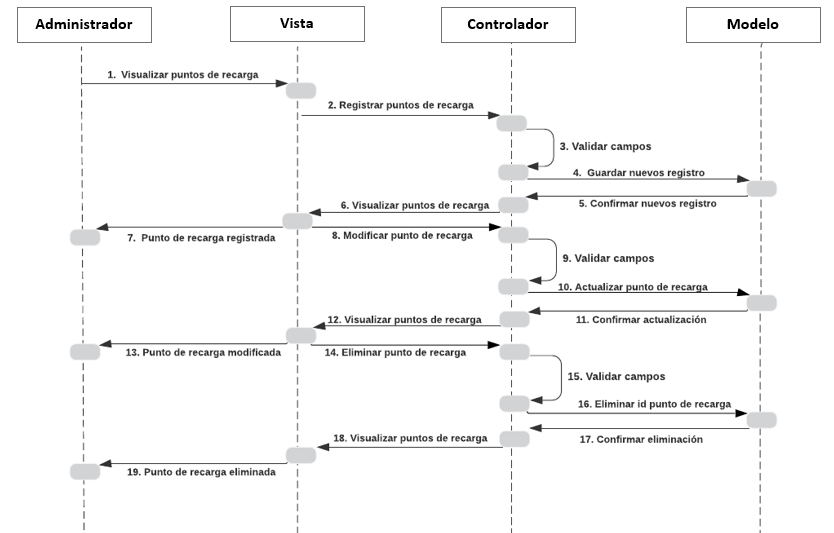
***Figura 33.*** *Diagrama de secuencia gestionar frecuencias*



Fuente: Autores del proyecto

La siguiente figura evidencia el proceso que se ejecuta al gestionar puntos de recarga:

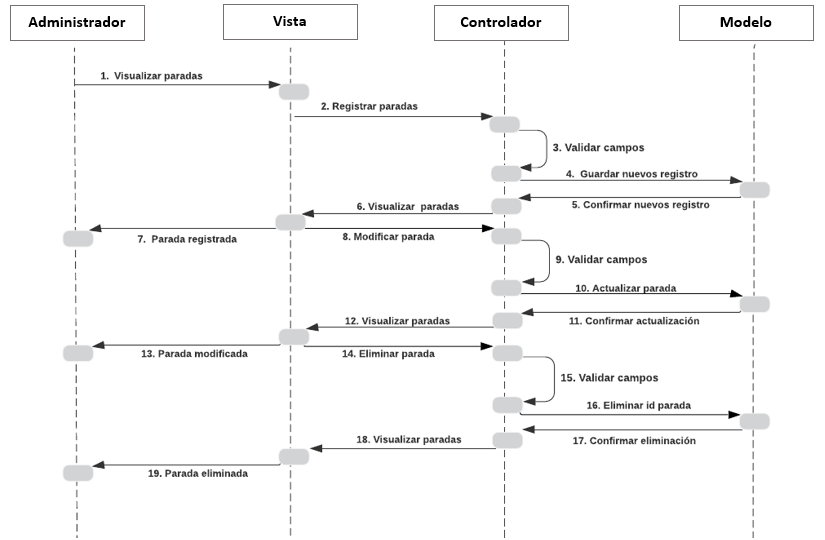
***Figura 34.*** *Diagrama de secuencia gestionar puntos de recarga*



Fuente: Autores del proyecto

La siguiente figura evidencia el proceso que se ejecuta al gestionar paradas:

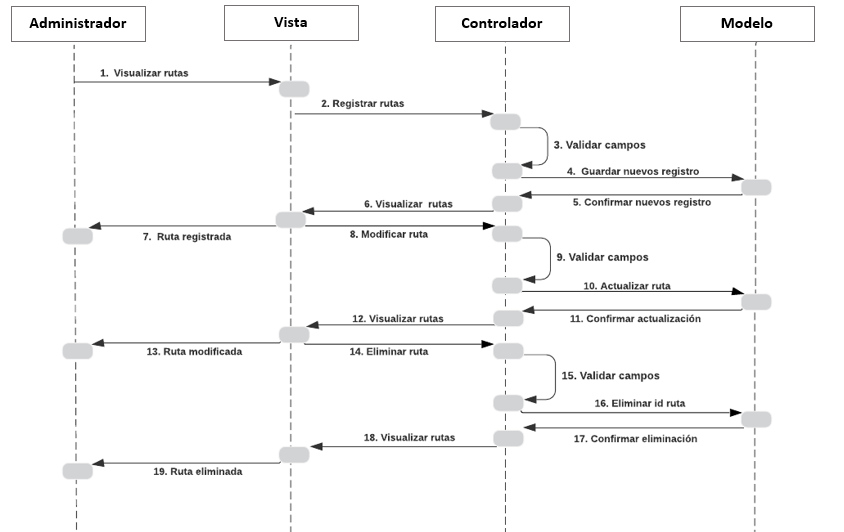
***Figura 35.*** *Diagrama de secuencia gestionar paradas*



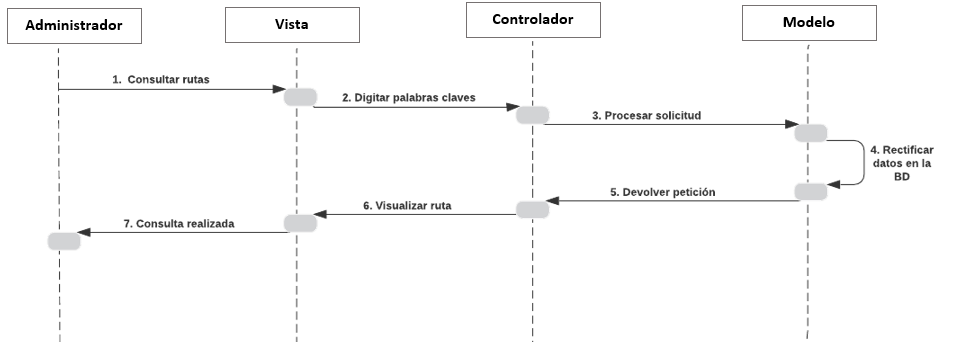
Fuente: Autores del proyecto

La siguiente figura evidencia el proceso que se ejecuta al gestionar rutas:

***Figura 36.*** *Diagrama de secuencia gestionar rutas*



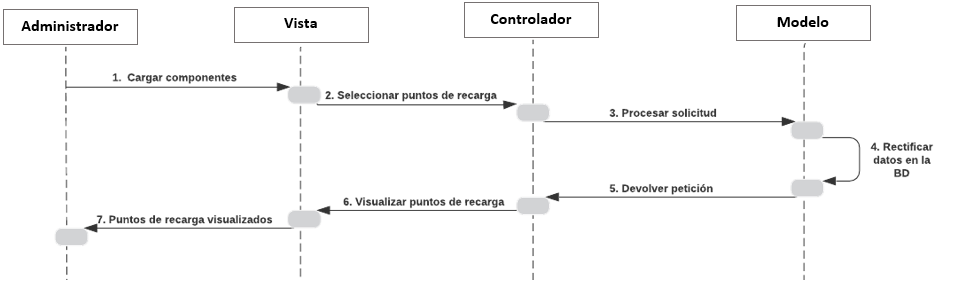
Fuente: Autores del proyecto

La siguiente figura evidencia el proceso que se ejecuta al consultar rutas: ******

***Figura 37.*** *Diagrama de secuencia consultar rutas*

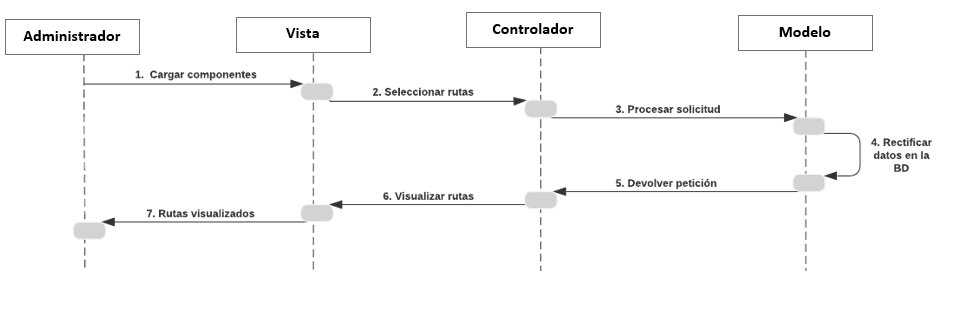
Fuente: Autores del proyecto

La siguiente figura evidencia el proceso que se ejecuta al visualizar puntos de recarga:

***Figura 38.*** *Diagrama de secuencia visualizar puntos de recarga*

Fuente: Autores del proyecto

La siguiente figura evidencia el proceso que se ejecuta al visualizar rutas:

***Figura 39.*** *Diagrama de secuencia visualizar rutas*

Fuente: Autores del proyecto

## FASE 3. DESARROLLO DE LA APLICACIÓN

* + 1. Módulo de la interfaz para la visualización de contenidos

A continuación, se observarán diversas figuras las cuales representan la interfaz gráfica para el usuario consultante del prototipo:

Interfaz de usuario gráfica

Descripción generada automáticamente***Figura 40.*** *Menú de navegación de la interfaz visual de contenidos*

Fuente: Autores del proyecto

***Tabla 17.*** *Descripción de la interfaz de visualización de contenidos*

|  |  |
| --- | --- |
| Categoría | Descripción |
| Inicio | Permite consultar la información de los servicios actuales en el prototipo de software web |
| Listado de rutas | Permite visualizar todo el contenido relacionado con las rutas, además, otorga la posibilidad de buscar por medio de palabras clave |
| Listado de puntos de Recarga | Permite visualizar todo el contenido relacionado con los puntos de recarga |

Fuente: Autores del proyecto

***Figura 41.*** *Módulo inicio de la interfaz visual*

Interfaz de usuario gráfica

Descripción generada automáticamente

Fuente: Autores del proyecto

***Figura 42.*** *Módulo listado de rutas de la interfaz visual*

Imagen que contiene Tabla

Descripción generada automáticamente

Fuente: Autores del proyecto

***Figura 43.*** *Módulo listado puntos de recarga de la interfaz visual*

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Fuente: Autores del proyecto

* + 1. Módulo de la interfaz gráfica administrativa

A continuación, se observarán diversas figuras las cuales representan la interfaz gráfica para el usuario administrador del prototipo:

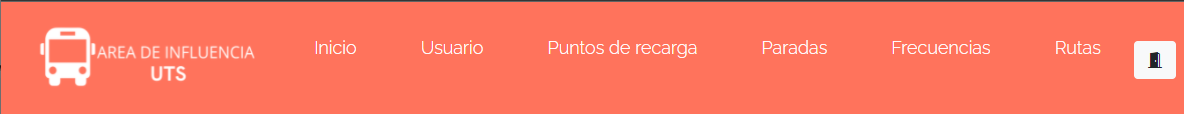
***Figura 44.*** *Modulo inicio de sesión*

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Fuente: Autores del proyecto

***Figura 45.*** *Menú de navegación de la interfaz gráfica administrativa*



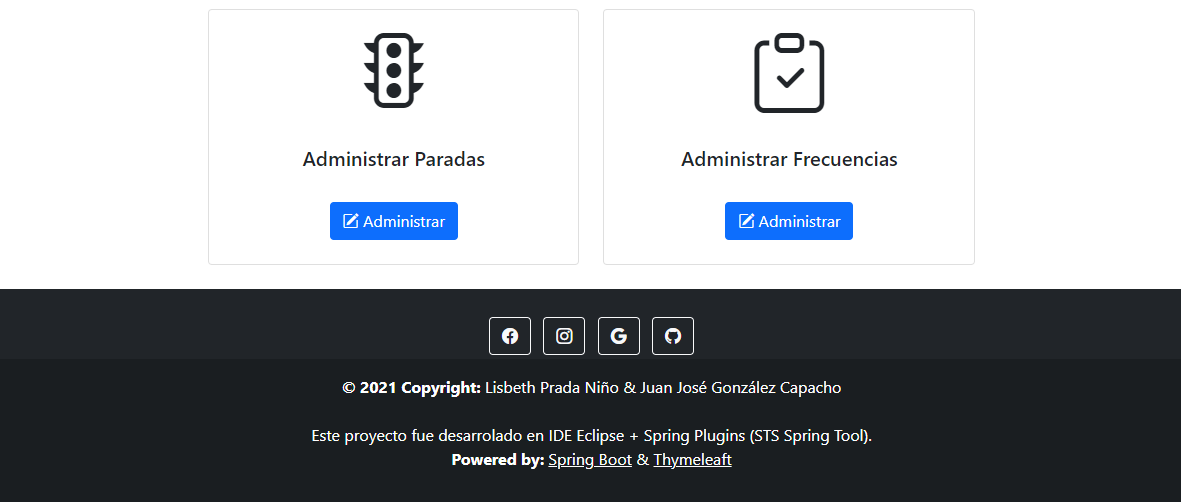
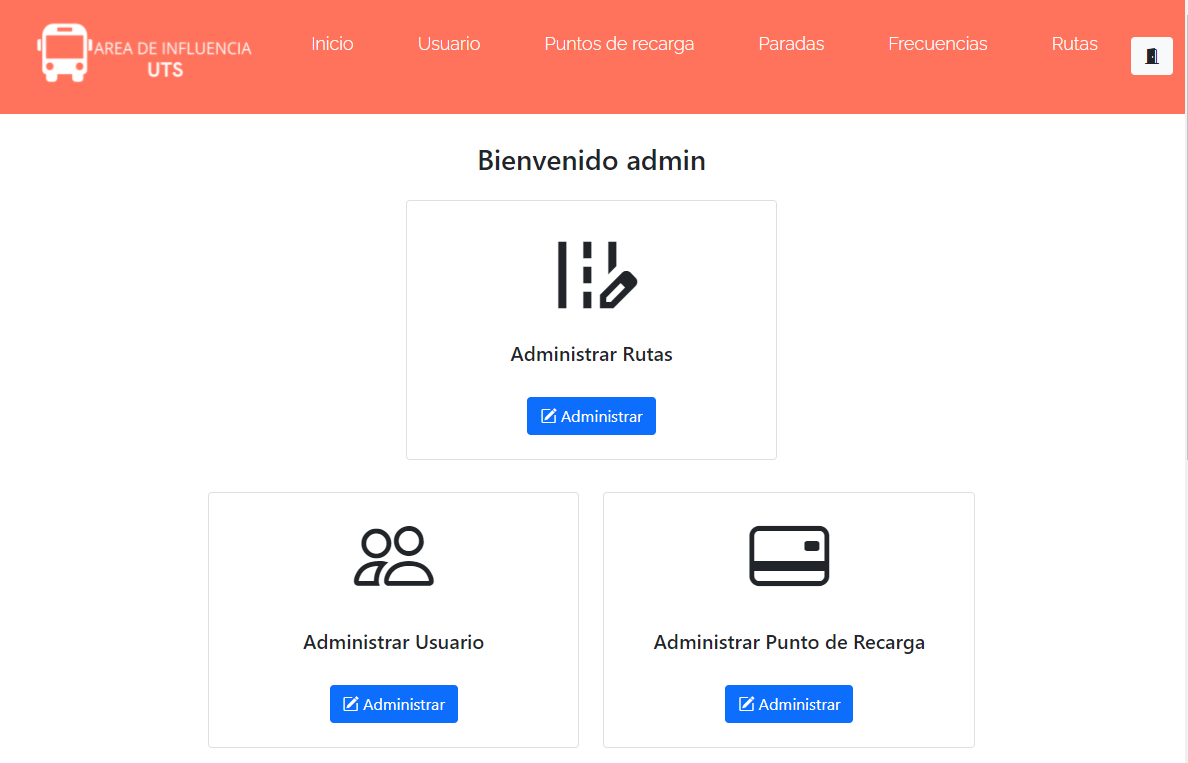
Fuente: Autores del proyecto

***Tabla 18.*** *Descripción de la interfaz gráfica administrativa*

|  |  |
| --- | --- |
| Categoría | Descripción |
| Iniciar sesión | Permite visualizar el formulario diseñado para iniciar sesión como administrador. |
| Inicio administrativo | Muestra las categorías que se podrán gestionar por el administrador. |
| Usuarios | Permite gestionar todos los datos de los usuarios del sistema. |
| Crear Usuario | Permite añadir un nuevo usuario al sistema. |
| Puntos de recarga | Permite gestionar todo el contenido relacionado con los puntos de recarga. |
| Crear punto de recarga | Permite añadir un nuevo punto de recarga. |
| Paradas | Permite gestionar todo el contenido relacionado con las paradas de los buses o transporte masivo. |
| Crear parada | Permite añadir una nueva parada. |
| Frecuencias | Permite gestionar todo el contenido relacionado con las frecuencias. |
| Crear frecuencia | Permite añadir una nueva frecuencia. |
| Rutas | Permite gestionar todo el contenido relacionado con las rutas junto con sus diversas paradas y diferentes frecuencias. |
| Crear ruta | Permite añadir una nueva ruta. |

Fuente: Autores del proyecto

***Figura 46.*** *Módulo inicio administrativo de la interfaz gráfica administrativa*



Fuente: Autores del proyecto

***Figura 47.*** *Módulo usuarios de la interfaz gráfica administrativa*

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Fuente: Autores del proyecto

***Figura 48.*** *Módulo crear usuario de la interfaz gráfica administrativa*

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Fuente: Autores del proyecto

***Figura 49.*** *Módulo puntos de recarga de la interfaz gráfica administrativa*

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Fuente: Autores del proyecto

***Figura 50****. Módulo crear punto de recarga de la interfaz gráfica administrativa*

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Fuente: Autores del proyecto

***Figura 51.*** *Módulo paradas de la interfaz gráfica administrativa*

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Fuente: Autores del proyecto

***Figura 52.*** *Módulo crear parada de la interfaz gráfica administrativa*

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Fuente: Autores del proyecto

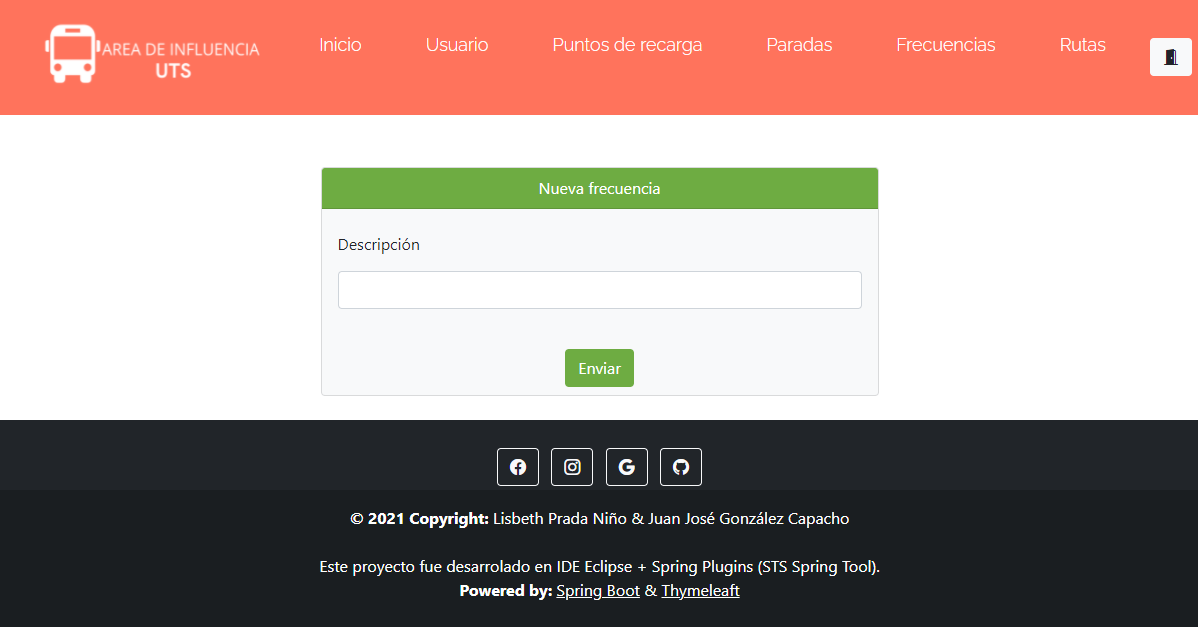
***Figura 53.*** *Módulo frecuencias de la interfaz gráfica administrativa*

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Fuente: Autores del proyecto

***Figura 54.*** *Módulo crear frecuencia de la interfaz gráfica administrativa*



Fuente: Autores del proyecto

***Figura 55.*** *Módulo rutas de la interfaz gráfica administrativa*

Imagen que contiene Sitio web

Descripción generada automáticamente

Fuente: Autores del proyecto

***Figura 56.*** *Módulo crear ruta de la interfaz gráfica administrativa*

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Fuente: Autores del proyecto

## FASE 4. PRUEBAS

* + 1. Prueba de funcionalidad

En la siguiente tabla se muestra los resultados obtenidos al realizar pruebas de funcionamiento con respecto a los requerimientos generales.

***Tabla 19.*** *Requerimientos generales - resultados*

|  |  |
| --- | --- |
| Requerimientos generales | Resultado |
| Ingresar al prototipo de software | Exitoso |
| Gestionar información | Exitoso |
| Búsqueda de rutas | Exitoso |
| Visualizar contenido de las rutas | Exitoso |

Fuente: Autores del proyecto

* + 1. Casos de prueba

En las siguientes tablas se encontrarán los resultados obtenidos en los casos de pruebas que se desarrollaron para validar el correcto funcionamiento del software.

***Tabla 20.*** *Casos de prueba, Gestionar usuario*

|  |  |
| --- | --- |
| Casos de pruebas – Gestionar usuario | |
| No. | **1** |
| Nombre identificador | CP\_AddUsuario1 |
| Descripción | Adicionar usuario a la base de datos |
| Precondición | Iniciar sesión |
| Entradas | Datos del nuevo usuario |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo usuarios 3. Presionar el icono añadir 4. Diligenciar los datos del nuevo usuario 5. Presionar en “Enviar” |
| Resultados esperados | Registrará el username y la contraseña cifrada en la tabla usuario de la base de datos |
| Postcondición | Redireccionará al módulo usuarios |
| Estado | Correcto |
| Prioridad | Alta |
| No. | **2** |
| Nombre identificador | CP\_UpdateUsuario1 |
| Descripción | Modificar los datos de un usuario |
| Precondición | 1. Iniciar sesión 2. Debe existir al menos un registro en la tabla usuario |
| Entradas | Nuevos datos del usuario |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo usuarios 3. Presionar el icono editar del usuario 4. Diligenciar los nuevos datos del usuario 5. Presionar en “Enviar” |
| Resultados esperados | Modificar los datos en la tabla usuario del usuario seleccionado |
| Postcondición | Redireccionará al módulo usuarios |
| Estado | Correcto |
| Prioridad | Alta |
| No. | **3** |
| Nombre identificador | CP\_DeleteUsuario1 |
| Descripción | Eliminar usuario de la base de datos |
| Precondición | 1. Iniciar sesión 2. Debe existir al menos un registro en la tabla usuario |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo usuarios 3. Presionar el icono eliminar del usuario |
| Resultados esperados | Eliminar los datos en la tabla usuario del usuario seleccionado |
| Postcondición | Mantenerse en el módulo usuarios |
| Estado | Correcto |
| Prioridad | Alta |
| No. | **4** |
| Nombre identificador | CP\_ReadUsuario1 |
| Descripción | Listar todos los usuarios creados en una sola tabla |
| Precondición | 1. Iniciar sesión |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo usuarios |
| Resultados esperados | Mostrar todos los datos de los usuarios en una sola tabla de manera correcta. |
| Postcondición | Mantenerse en el módulo usuarios |
| Estado | Correcto |
| Prioridad | Alta |

Fuente: Autores del proyecto

***Tabla 21.*** *Casos de prueba, Gestionar punto de recarga*

|  |  |
| --- | --- |
| Casos de pruebas – Gestionar punto de recarga | |
| No. | **1** |
| Nombre identificador | CP\_AddPuntoDeRecarga1 |
| Descripción | Adicionar punto de recarga a la base de datos |
| Precondición | Iniciar sesión |
| Entradas | Datos del nuevo punto de recarga |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo puntos de recarga 3. Presionar el icono añadir 4. Diligenciar los datos del nuevo punto de recarga 5. Presionar en “Enviar” |
| Resultados esperados | Registrará los datos diligenciados en la tabla punto\_de\_recarga de la base de datos |
| Postcondición | Redireccionará al módulo puntos de recarga |
| Estado | Correcto |
| Prioridad | Alta |
| No. | **2** |
| Nombre identificador | CP\_UpdatePuntoDeRecarga1 |
| Descripción | Modificar los datos de un punto de recarga |
| Precondición | 1. Iniciar sesión 2. Debe existir al menos un registro en la tabla punto\_de\_recarga |
| Entradas | Nuevos datos del punto de recarga |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo puntos de recarga 3. Presionar el icono editar del punto de recarga 4. Diligenciar los nuevos datos del punto de recarga 5. Presionar en “Enviar” |
| Resultados esperados | Modificar los datos en la tabla punto\_de\_recarga del punto de recarga seleccionado |
| Postcondición | Redireccionará al módulo puntos de recarga |
| Estado | Correcto |
| Prioridad | Alta |
| No. | **3** |
| Nombre identificador | CP\_DeletePuntoDeRecarga1 |
| Descripción | Eliminar punto de recarga de la base de datos |
| Precondición | 1. Iniciar sesión 2. Debe existir al menos un registro en la tabla punto\_de\_recarga |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo puntos de recarga 3. Presionar el icono eliminar del punto de recarga |
| Resultados esperados | Eliminar los datos en la tabla punto\_de\_recarga del punto de recarga seleccionado |
| Postcondición | Mantenerse en el módulo puntos de recarga |
| Estado | Correcto |
| Prioridad | Alta |
| No. | **4** |
| Nombre identificador | CP\_ReadPuntoDeRecarga1 |
| Descripción | Listar todos los puntos de recarga creados en una sola tabla |
| Precondición | 1. Iniciar sesión |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo puntos de recarga |
| Resultados esperados | Mostrar todos los datos de los puntos de recarga en una sola tabla de manera correcta. |
| Postcondición | Mantenerse en el módulo puntos de recarga |
| Estado | Correcto |
| Prioridad | Alta |

Fuente: Autores del proyecto

***Tabla 22.*** *Casos de prueba, Gestionar parada*

|  |  |
| --- | --- |
| Casos de pruebas – parada | |
| No. | **1** |
| Nombre identificador | CP\_AddParada1 |
| Descripción | Adicionar parada a la base de datos |
| Precondición | Iniciar sesión |
| Entradas | Datos de la nueva parada |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo paradas 3. Presionar el icono añadir 4. Diligenciar los datos de la nueva parada 5. Presionar en “Enviar” |
| Resultados esperados | Registrará los datos diligenciados en la tabla parada de la base de datos |
| Postcondición | Redireccionará al módulo paradas |
| Estado | Correcto |
| Prioridad | Alta |
| No. | **2** |
| Nombre identificador | CP\_UpdateParada1 |
| Descripción | Modificar los datos de una parada |
| Precondición | 1. Iniciar sesión 2. Debe existir al menos un registro en la tabla parada |
| Entradas | Nuevos datos de la parada |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo paradas 3. Presionar el icono editar de la parada 4. Diligenciar los nuevos datos de la parada 5. Presionar en “Enviar” |
| Resultados esperados | Modificar los datos en la tabla parada de la parada seleccionada |
| Postcondición | Redireccionará al módulo paradas |
| Estado | Correcto |
| Prioridad | Alta |
| No. | **3** |
| Nombre identificador | CP\_DeleteParada1 |
| Descripción | Eliminar parada de la base de datos |
| Precondición | 1. Iniciar sesión 2. Debe existir al menos un registro en la tabla parada |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo paradas 3. Presionar el icono eliminar de la parada |
| Resultados esperados | Eliminar los datos en la tabla parada de la parada seleccionada |
| Postcondición | Mantenerse en el módulo paradas |
| Estado | Correcto |
| Prioridad | Alta |
| No. | **4** |
| Nombre identificador | CP\_ReadParada1 |
| Descripción | Listar todas las paradas creadas en una sola tabla |
| Precondición | 1. Iniciar sesión |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo paradas |
| Resultados esperados | Mostrar todos los datos de las paradas en una sola tabla de manera correcta. |
| Postcondición | Mantenerse en el módulo paradas |
| Estado | Correcto |
| Prioridad | Alta |

Fuente: Autores del proyecto

***Tabla 23.*** *Casos de prueba, Gestionar frecuencia*

|  |  |
| --- | --- |
| Casos de pruebas – frecuencia | |
| No. | **1** |
| Nombre identificador | CP\_AddFrecuencia1 |
| Descripción | Adicionar frecuencia a la base de datos |
| Precondición | Iniciar sesión |
| Entradas | Datos de la nueva frecuencia |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo frecuencias 3. Presionar el icono añadir 4. Diligenciar los datos de la nueva frecuencia 5. Presionar en “Enviar” |
| Resultados esperados | Registrará los datos diligenciados en la tabla frecuencia de la base de datos |
| Postcondición | Redireccionará al módulo frecuencias |
| Estado | Correcto |
| Prioridad | Alta |
| No. | **2** |
| Nombre identificador | CP\_UpdateFrecuencia1 |
| Descripción | Modificar los datos de una frecuencia |
| Precondición | 1. Iniciar sesión 2. Debe existir al menos un registro en la tabla frecuencia |
| Entradas | Nuevos datos de la frecuencia |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo frecuencias 3. Presionar el icono editar de la frecuencia 4. Diligenciar los nuevos datos de la frecuencia 5. Presionar en “Enviar” |
| Resultados esperados | Modificar los datos en la tabla frecuencia de la frecuencia seleccionada |
| Postcondición | Redireccionará al módulo frecuencias |
| Estado | Correcto |
| Prioridad | Alta |
| No. | **3** |
| Nombre identificador | CP\_DeleteFrecuencia1 |
| Descripción | Eliminar frecuencia de la base de datos |
| Precondición | 1. Iniciar sesión 2. Debe existir al menos un registro en la tabla frecuencia |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo frecuencias 3. Presionar el icono eliminar de la frecuencia |
| Resultados esperados | Eliminar los datos en la tabla frecuencia de la frecuencia seleccionada |
| Postcondición | Mantenerse en el módulo frecuencias |
| Estado | Correcto |
| Prioridad | Alta |
| No. | **4** |
| Nombre identificador | CP\_ReadFrecuencia1 |
| Descripción | Listar todas las frecuencias creadas en una sola tabla |
| Precondición | 1. Iniciar sesión |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo frecuencias |
| Resultados esperados | Mostrar todos los datos de las frecuencias en una sola tabla de manera correcta. |
| Postcondición | Mantenerse en el módulo frecuencias |
| Estado | Correcto |
| Prioridad | Alta |

Fuente: Autores del proyecto

***Tabla 24.*** *Casos de prueba, Gestionar ruta*

|  |  |
| --- | --- |
| Casos de pruebas – ruta | |
| No. | **1** |
| Nombre identificador | CP\_AddRuta1 |
| Descripción | Adicionar ruta a la base de datos |
| Precondición | Iniciar sesión |
| Entradas | Datos de la nueva ruta |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo rutas 3. Presionar el icono añadir 4. Diligenciar los datos de la nueva ruta 5. Presionar en “Enviar” |
| Resultados esperados | Registrará los datos diligenciados en la tabla ruta, parada\_ruta, frecuencia\_ruta de la base de datos |
| Postcondición | Redireccionará al módulo rutas |
| Estado | Correcto |
| Prioridad | Alta |
| No. | **2** |
| Nombre identificador | CP\_UpdateRuta1 |
| Descripción | Modificar los datos de una ruta |
| Precondición | 1. Iniciar sesión 2. Debe existir al menos un registro en la tabla ruta |
| Entradas | Nuevos datos de la ruta |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo rutas 3. Presionar el icono editar de la ruta 4. Diligenciar los nuevos datos de la ruta 5. Presionar en “Enviar” |
| Resultados esperados | Modificar los datos en la tabla ruta, parada\_ruta, frecuencia\_ruta de la ruta seleccionada |
| Postcondición | Redireccionará al módulo rutas |
| Estado | Correcto |
| Prioridad | Alta |
| No. | **3** |
| Nombre identificador | CP\_DeleteRuta1 |
| Descripción | Eliminar ruta de la base de datos |
| Precondición | 1. Iniciar sesión 2. Debe existir al menos un registro en la tabla ruta |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo rutas 3. Presionar el icono eliminar de la ruta |
| Resultados esperados | Eliminar los datos en la tabla ruta, parada\_ruta, frecuencia\_ruta de la ruta seleccionada |
| Postcondición | Mantenerse en el módulo rutas |
| Estado | Correcto |
| Prioridad | Alta |
| No. | **4** |
| Nombre identificador | CP\_ReadRuta1 |
| Descripción | Listar todas las rutas creadas en una sola tabla |
| Precondición | 1. Iniciar sesión |
| Pasos | 1. Iniciar sesión 2. Dirigirse al módulo rutas |
| Resultados esperados | Mostrar todos los datos de las rutas en una sola tabla de manera correcta. |
| Postcondición | Mantenerse en el módulo rutas |
| Estado | Correcto |
| Prioridad | Alta |

Fuente: Autores del proyecto

***Tabla 25.*** *Casos de prueba, Inicio de sesión*

|  |  |
| --- | --- |
| Casos de pruebas – inicio de sesión | |
| No. | **1** |
| Nombre identificador | CP\_Login1 |
| Descripción | Iniciar sesión de manera correcta |
| Precondición | Debe existir al menos un usuario |
| Entradas | 1. Usuario 2. Contraseña |
| Pasos | 1. Ingresar al software 2. Dirigirse al módulo inicio de sesión 3. Diligenciar los datos requeridos 4. Presionar el botón “INICIAR SESIÓN” |
| Resultados esperados | Otorgar acceso al módulo administrativo |
| Postcondición | Redireccionará módulo inicio de la interfaz administrativa |
| Estado | Correcto |
| Prioridad | Alta |
| No. | **2** |
| Nombre identificador | CP\_Login2 |
| Descripción | Iniciar sesión con datos erróneos |
| Precondición | Debe existir al menos un usuario |
| Entradas | 1. Usuario 2. Contraseña |
| Pasos | 1. Ingresar al software 2. Dirigirse al módulo inicio de sesión 3. Diligenciar los datos requeridos de manera errónea 4. Presionar el botón “INICIAR SESIÓN” |
| Resultados esperados | Denegar el acceso al módulo administrativo |
| Postcondición | Se mantendrá en el módulo inicio de sesión, y el software le mostrará un mensaje en rojo que le advertirá que ingresó mal los datos. |
| Estado | Correcto |
| Prioridad | Alta |
| No. | **3** |
| Nombre identificador | CP\_Login3 |
| Descripción | Iniciar sesión con datos incompletos |
| Precondición | Debe existir al menos un usuario |
| Entradas | 1. Usuario 2. Contraseña |
| Pasos | 1. Ingresar al software 2. Dirigirse al módulo inicio de sesión 3. Presionar el botón “INICIAR SESIÓN” sin haber rellanado los datos requeridos |
| Resultados esperados | Denegar el acceso al módulo administrativo |
| Postcondición | Se mantendrá en el módulo inicio de sesión, y el software le mostrará un mensaje en rojo que le advertirá que ingresó mal los datos. |
| Estado | Correcto |
| Prioridad | Alta |

Fuente: Autores del proyecto

* + 1. Prueba de carga

En la siguiente tabla se muestra los resultados obtenidos con respecto el tiempo de carga de cada módulo del software.

***Tabla 26.*** *Pruebas de carga*

|  |  |  |
| --- | --- | --- |
| Módulo | Tiempos de carga (segundos) | |
| Chrome | Firefox |
| Inicio | 0,53 | 0,36 |
| Listado de rutas | 0,42 | 0,28 |
| Listado de puntos de Recarga | 0,62 | 0,44 |
| Iniciar sesión | 0,66 | 0,57 |
| Inicio administrativo | 0.96 | 0,53 |
| Usuario | 0,59 | 0,47 |
| Crear Usuario | 0,41 | 0,4 |
| Puntos de recarga | 0,58 | 0,32 |
| Crear punto de recarga | 0.5 | 0,31 |
| Paradas | 0,47 | 0,31 |
| Crear parada | 0,51 | 0,37 |
| Frecuencias | 0,58 | 0,4 |
| Crear frecuencia | 0,43 | 0,38 |
| Rutas | 0,63 | 0,46 |
| Crear ruta | 0,47 | 0,42 |

Fuente: Autores del proyecto

# RESULTADOS

Los resultados que se obtuvieron en la elaboración de este proyecto de grado se mencionan a continuación:

***Tabla 27.*** *Estado de los resultados esperados.*

|  |  |  |
| --- | --- | --- |
| Resultado | Estado | Observaciones |
| Las especificaciones de los requerimientos funcionales y no funcionales del prototipo de software. | Completado | Se encuentra en la sección 4.1.3. del documento |
| Artículo científico que evidencie los resultados del proyecto. | Completado | Ninguna |
| La elaboración del diseño del software. | Completado | Se encuentra en la sección 4.2. del documento |
| Prototipo de software web para la consulta de rutas en el área de influencia de las UTS, desarrollado con HTML, Framework Spring y gestor de base de datos MySQL. | Completado | Ninguna |
| Manual de instalación | Completado | Ninguna |
| Manual de usuario. | Completado | Ninguna |
| URL con direccionamiento a GitHub con toda la evidencia del desarrollo del proyecto de grado (código fuente, script de la base de datos, manuales de uso e instalación). | Completado | Ninguna |

Fuente: Autores del proyecto

# CONCLUSIONES

* Los requerimientos funcionales y no funcionales para la elaboración del prototipo de software web, ayudaron en gran medida ya que estos definieron las funcionalidades principales que se necesitaban para que el producto final cumpliera con todas las especificaciones planteadas en el principio del proyecto. Cabe mencionar que dichos requerimientos se definieron en base a diversos softwares planteamientos similares.
* MVC fue la arquitectura que se utilizó para el correcto desarrollo de este proyecto ya que es de gran importancia en proyectos tecnológicos, la implementación de una arquitectura de software. MVC se escogió debido a que separa el contenido del proyecto en el modelo, la vista y los controladores, brindando así ventajas como la legibilidad del código.
* Para el desarrollo y ejecución del prototipo software para la consulta de rutas de buses y de transporte masivo en el área de influencia de las unidades tecnológicas de Santander, Bucaramanga, esto haciendo uso de las funcionalidades que brinda el framework Spring Tool Suite, tendrá beneficios tales como, mejorar la movilidad de los habitantes de la zona, disminuir el tiempo de traslado de un lugar a otro, y beneficiar en gran parte a los estudiantes foráneos.
* Comprobar el correcto funcionamiento y el cumplimiento de los requisitos establecidos, a través de diferentes pruebas realizadas al software, permitiéndonos analizar y corregir los errores, mediante lo establecido en la norma ISO 29119 (Software Testing Standard). Todas las pruebas realizadas arrojaron resultados positivos, demostrando así, que el software cumple con todos los requerimientos establecidos inicialmente.

# RECOMENDACIONES

* Para la versión final del software se recomienda la implementación de un mapa interactivo que permita ver la ubicación de cada una de las paradas de las rutas, y adicionalmente, la ubicación de los puntos de recarga de la tarjeta de transporte masivo.
* Se recomienda que, para la versión final de software se incluya un nuevo módulo de contáctenos con el objetivo de permitir al usuario consultante contactar con el administrador o administradores.
* A la hora de ingresar nuevos registros, se recomienda seguir con el mismo formato que se utilizó en un inició, esto para mantener una uniformidad en la vista del consultante.

# REFERENCIAS BIBLIOGRÁFICAS

Botero Sánchez, M. E., & Alegría Paredes, D. P. (2017). ¿La nueva tecnología en el sector transporte obstaculiza o fortalece el servicio público? *Verba Luris*, *37*. https://doi.org/10.18041/0121-3474/verbaiuris.0.1022

Carvajal Salamanca, R. (2021). *Implementación de las TIC en un modelo de gestión de cultura ciudadana en el área de influencia de las unidades tecnológicas de Santander*. Unidades Tecnológicas de Santander.

Constanzo, Marcela A; Casas, S. I. (2018). *Usabilidad de framework web: identificación de problemas y propuesta de evaluación*. http://sedici.unlp.edu.ar/bitstream/handle/10915/73289/Documento\_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y

Contreras Contreras, Ignacio ; Hernández Gómez, G. (2017). Sistema de localización en tiempo real mediante un servidor web y aplicaciones móviles. *Pistas Educativas*. http://www.itc.mx/ojs/index.php/pistas/article/view/1073/884

Contributors, M. (2021a). *CRUD*. MDN Web Docs. https://developer.mozilla.org/es/docs/Glossary/CRUD

Contributors, M. (2021b). *HTML: Lenguaje de etiquetas de hipertexto*. https://developer.mozilla.org/es/docs/Web/HTML

Delgado Olivera, L. de la claridad;, & Diaz Alonso, L. M. (2021). Modelos de Desarrollo de Software. *Revista Cubana de Ciencias Informáticas*. http://scielo.sld.cu/scielo.php?pid=S2227-18992021000100037&script=sci\_arttext&tlng=en

Educació, J. (2017). *¿Qué tipos de aplicaciones web existen?* https://fp.uoc.fje.edu/blog/que-tipos-de-aplicaciones-web-existen/

Foundation, D. S. (2021). *¿Por qué Django?* Django. https://www.djangoproject.com/start/overview/

González González, F., & Calero Castañeda, S. L. (2019). *Comparación de las metodologías cascada y ágil para el aumento de la productividad en el desarrollo de software* [Universidad Santiago de Cali]. https://repository.usc.edu.co/handle/20.500.12421/1208

Google. (2021). *Características y beneficios*. Angular. https://angular.io/features

Martínez Canelo, M. (2020). *¿Qué es la Programación Orientada a Objetos?* Profile. https://profile.es/blog/que-es-la-programacion-orientada-a-objetos/

Montagut Martínez, C., & Vargas Rodríguez, J. G. (2019). *Desarrollo de una aplicación web para consulta y visualización de rutas intermunicipales de los Municipios de la Provincia “Sabana de Occidente” del Departamento de Cundinamarca.* http://hdl.handle.net/11349/16291

Oracle. (2018). *¿Qué es una base de datos?* https://www.oracle.com/py/database/what-is-database/

Orozco Aguirre, H. R. (2019). *MODELO VISTA CONTROLADOR (MVC) Y WEB ARCHIVES (WARS)*. http://ri.uaemex.mx/bitstream/handle/20.500.11799/108712/secme-35486\_1.pdf?sequence=1

Perforce Software, I. (2021). *¿Por qué elegir Zend Services?* Zend. https://www.zend.com/

Quiñonez, Yadira; Lizarraga Carmen;Peraza, J. Z. O. (2019). Sistema inteligente para el monitoreo automatizado del transporte público en tiempo real. *Revista Ibérica de Sistemas e Tecnologias de Informação*, 94–105. https://doi.org/10.17013/risti.31.94–105

Quintero Pérez, G. I., & Gómez Suárez, M. A. (2021). De las Smart Cities a los territorios inteligentes: semejanzas, diferencias y trascendencias. *Revistarquis*, *10*(1), 23–33. https://doi.org/10.15517/ra.v10i1.45257

Ley 23 de 1982 Sobre derechos de autor, (1982). https://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=3431

Rojas, Juan Pablo;Bustos, Julio César;Ordóñez Camacho, D. (2017). Transporte público inteligente al alcance de sus manos. *Enfoque UTE*, *8*(1), 122–134. https://doi.org/10.29019/enfoqueute.v8n1.143

Roman Arenaza, R. E. (2019). *Lenguajes de programación Javascript Java y Javascript. Características. Norma de escritura. Variables y operadores lógicos. Mensajes. Ejercicios. Estructuras condicionales. Funciones y objetos. Aplicaciones*. http://repositorio.une.edu.pe/handle/UNE/3026

Smith, S. (2021). *Información general de ASP.NET*. Microsoft. https://docs.microsoft.com/es-es/aspnet/core/mvc/overview?view=aspnetcore-5.0

SpringSource. (2021). *Spring*. https://spring.io/

Valarezo Pardo, Milton Rafael; Honores Tapia, Joofre Antonio; Gómez Moreno, Antonio Steeven;Vinces Sánchez, L. F. (2018). *Comparación de tendencias tecnológicas en aplicaciones web* (27th ed., p. 6). https://doi.org/https://dx.doi.org/10.17993/3ctec.2018.070133.1-11

Zumba Gamboa, J. (2018). Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software. *INNOVA Research Journal*, *3*, 20–33. https://dialnet.unirioja.es/servlet/articulo?codigo=6777227

Botero Sánchez, M. E., & Alegría Paredes, D. P. (2017). ¿La nueva tecnología en el sector transporte obstaculiza o fortalece el servicio público? *Verba Luris*, *37*. https://doi.org/10.18041/0121-3474/verbaiuris.0.1022

Carvajal Salamanca, R. (2021). *Implementación de las TIC en un modelo de gestión de cultura ciudadana en el área de influencia de las unidades tecnológicas de Santander*. Unidades Tecnológicas de Santander.

Constanzo, Marcela A; Casas, S. I. (2018). *Usabilidad de framework web: identificación de problemas y propuesta de evaluación*. http://sedici.unlp.edu.ar/bitstream/handle/10915/73289/Documento\_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y

Contreras Contreras, Ignacio ; Hernández Gómez, G. (2017). Sistema de localización en tiempo real mediante un servidor web y aplicaciones móviles. *Pistas Educativas*. http://www.itc.mx/ojs/index.php/pistas/article/view/1073/884

Contributors, M. (2021a). *CRUD*. MDN Web Docs. https://developer.mozilla.org/es/docs/Glossary/CRUD

Contributors, M. (2021b). *HTML: Lenguaje de etiquetas de hipertexto*. https://developer.mozilla.org/es/docs/Web/HTML

Delgado Olivera, L. de la claridad;, & Diaz Alonso, L. M. (2021). Modelos de Desarrollo de Software. *Revista Cubana de Ciencias Informáticas*. http://scielo.sld.cu/scielo.php?pid=S2227-18992021000100037&script=sci\_arttext&tlng=en

Educació, J. (2017). *¿Qué tipos de aplicaciones web existen?* https://fp.uoc.fje.edu/blog/que-tipos-de-aplicaciones-web-existen/

Foundation, D. S. (2021). *¿Por qué Django?* Django. https://www.djangoproject.com/start/overview/

González González, F., & Calero Castañeda, S. L. (2019). *Comparación de las metodologías cascada y ágil para el aumento de la productividad en el desarrollo de software* [Universidad Santiago de Cali]. https://repository.usc.edu.co/handle/20.500.12421/1208

Google. (2021). *Características y beneficios*. Angular. https://angular.io/features

Martínez Canelo, M. (2020). *¿Qué es la Programación Orientada a Objetos?* Profile. https://profile.es/blog/que-es-la-programacion-orientada-a-objetos/

Montagut Martínez, C., & Vargas Rodríguez, J. G. (2019). *Desarrollo de una aplicación web para consulta y visualización de rutas intermunicipales de los Municipios de la Provincia “Sabana de Occidente” del Departamento de Cundinamarca.* http://hdl.handle.net/11349/16291

Oracle. (2018). *¿Qué es una base de datos?* https://www.oracle.com/py/database/what-is-database/

Orozco Aguirre, H. R. (2019). *MODELO VISTA CONTROLADOR (MVC) Y WEB ARCHIVES (WARS)*. http://ri.uaemex.mx/bitstream/handle/20.500.11799/108712/secme-35486\_1.pdf?sequence=1

Perforce Software, I. (2021). *¿Por qué elegir Zend Services?* Zend. https://www.zend.com/

Quiñonez, Yadira; Lizarraga Carmen;Peraza, J. Z. O. (2019). Sistema inteligente para el monitoreo automatizado del transporte público en tiempo real. *Revista Ibérica de Sistemas e Tecnologias de Informação*, 94–105. https://doi.org/10.17013/risti.31.94–105

Quintero Pérez, G. I., & Gómez Suárez, M. A. (2021). De las Smart Cities a los territorios inteligentes: semejanzas, diferencias y trascendencias. *Revistarquis*, *10*(1), 23–33. https://doi.org/10.15517/ra.v10i1.45257

Ley 23 de 1982 Sobre derechos de autor, (1982). https://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=3431

Rojas, Juan Pablo;Bustos, Julio César;Ordóñez Camacho, D. (2017). Transporte público inteligente al alcance de sus manos. *Enfoque UTE*, *8*(1), 122–134. https://doi.org/10.29019/enfoqueute.v8n1.143

Roman Arenaza, R. E. (2019). *Lenguajes de programación Javascript Java y Javascript. Características. Norma de escritura. Variables y operadores lógicos. Mensajes. Ejercicios. Estructuras condicionales. Funciones y objetos. Aplicaciones*. http://repositorio.une.edu.pe/handle/UNE/3026

Smith, S. (2021). *Información general de ASP.NET*. Microsoft. https://docs.microsoft.com/es-es/aspnet/core/mvc/overview?view=aspnetcore-5.0

SpringSource. (2021). *Spring*. https://spring.io/

Valarezo Pardo, Milton Rafael; Honores Tapia, Joofre Antonio; Gómez Moreno, Antonio Steeven;Vinces Sánchez, L. F. (2018). *Comparación de tendencias tecnológicas en aplicaciones web* (27th ed., p. 6). https://doi.org/https://dx.doi.org/10.17993/3ctec.2018.070133.1-11

Zumba Gamboa, J. (2018). Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software. *INNOVA Research Journal*, *3*, 20–33. https://dialnet.unirioja.es/servlet/articulo?codigo=6777227