

Plataforma Web Para La Gestión De Información De Los Usuarios Beneficiarios De Servicios Sociales Adjudicados Por La Secretaría De Desarrollo Social De La Alcaldía De Bucaramanga

Gabriel de Jesús Álvarez Tejada

Fabio Andrey Ojeda Villamizar

Trabajo de Grado para optar al título de Ingeniero de Sistemas

Director

Jathinson Meneses Mendoza

Magister en Gestión, Aplicación y Desarrollo de Software

Codirector

Henry Andrés Jiménez Herrera

Magíster en Ingeniería de Sistemas e Informática

Universidad Industrial de Santander

Facultad de Ingenierías Físico-mecánicas

Escuela de Ingeniería de Sistemas e Informática

Bucaramanga - Colombia

2023

### **Dedicatoria**

Este proyecto se lo queremos dedicar a cada persona en nuestras vidas que nos ayudaron a mejorar: amigos, compañeros, profesores y familiares, en especial destacamos a:

A nuestras madres Marlene Tejada y Olga Liliana Villamizar quienes nos dieron soporte a lo largos de todos estos años.

A nuestros padres Gabriel Alvarez Mejia Q.E.P.D. y Fabio Ojeda Arias Q.E.P.D. quienes con sus enseñanzas nos formaron y nos hicieron ser las personas que somos hoy en día.

A nuestros amigos: Julian Jerez Nieto, Pablo Rojas Yepes, Daniel Martínez Serrano y Álvaro Ríos Garzón quienes nos brindaron sus conocimientos.

A nuestro director Jathinson Meneses Mendoza y codirector Henry Andres Jimenez Herrera quienes nos guiaron en el desarrollo de este proceso.

A todos ellos agradecemos acompañarnos y ayudarnos en el desarrollo de proyecto.

### **Agradecimientos**

Durante el proceso de desarrollo de esta tesis de grado, enfrentamos desafíos significativos y logramos hitos importantes. Queremos expresar nuestro agradecimiento a varias personas y entidades que fueron fundamentales en este camino de la siguiente manera:

Agradecemos a Dios por su continua dirección y apoyo, que nos inspiraron a perseverar.

A nuestros queridos padres y familias, les agradecemos por su amor, apoyo incondicional y sacrificio a lo largo de nuestra educación, siendo un motor esencial para nuestro logro.

A nuestros amigos, les damos las gracias por su apoyo emocional y aliento en momentos de duda, su amistad ha sido inestimable.

A nuestros profesores y asesores, les agradecemos por su orientación experta, paciencia y compromiso con nuestra formación académica.

A la Secretaría de Desarrollo de Bucaramanga y a todas las personas que colaboraron con nuestro proyecto, les agradecemos por su apoyo y oportunidad de aprendizaje.

A la Universidad Industrial de Santander, agradecemos la oportunidad de crecimiento y aprendizaje en este campus enriquecedor.

Este logro se debe al esfuerzo conjunto de muchas personas e instituciones, y estamos profundamente agradecidos por su contribución. Gracias por ser parte de nuestro viaje y por hacer posible la finalización de esta tesis de grado.

Con gratitud, Gabriel Alvarez Tejada y Fabio Ojeda Villamizar

## Tabla de Contenido

<b>Introducción</b>	<b>16</b>
<b>1. Generalidades del proyecto</b>	<b>17</b>
1.1. Contexto	17
1.2. Descripción de la entidad donde se realizó la práctica	17
1.2.1. Misión	18
1.2.2. Visión	18
<b>2. Planteamiento del problema</b>	<b>20</b>
2.1. Justificación	20
2.1.1. Estado del arte	20
2.1.2. Impacto	21
2.1.3. Viabilidad	21
<b>3. Objetivos</b>	<b>23</b>
3.1. Objetivo general	23
3.2. Objetivos específicos	23
<b>4. Marco de referencia</b>	<b>24</b>
4.1. Arquitectura De Software	24

PLATAFORMA PARA LA GESTIÓN DE INFORMACIÓN EN LOS BENEFICIARIOS	5
4.1.1. Patrones De Arquitectura De Software	25
4.1.2. Aspectos	28
4.1.3. Frameworks	30
4.2. Selección	30
4.2.1. Framework para Back-end	31
4.2.2. Framework para Front-end	31
4.2.3. Arquitectura para Back-end	32
<b>5. Metodología</b>	<b>34</b>
<b>6. Requerimientos funcionales y Requerimientos no funcionales</b>	<b>36</b>
6.1. Requerimientos funcionales	36
6.2. Requerimientos no funcionales	36
<b>7. Casos de Uso</b>	<b>37</b>
7.1. Casos de uso	37
<b>8. Arquitectura</b>	<b>40</b>
8.1. Back-end (ASP.NET Core 6)	40
8.1.1. Arquitectura Monolítica Modular	40
8.2. Front-end (Angular 13)	42
8.3. Base de Datos (MySQL):	44
8.4. Vistas de Usuario:	45

<b>9. Desarrollo</b>	<b>47</b>
9.1. Registro Eficiente de Beneficiarios	48
9.1.1. Aumento en la velocidad de registro	48
9.1.2. Disminución de errores de registro	49
9.2. Centralización de Datos	49
9.2.1. Duplicidad de Registros	49
9.2.2. Generación de informes	51
9.3. Acceso Seguro y Control de Privacidad	52
9.3.1. Medidas de Seguridad	52
9.3.2. Control de Acceso	52
9.4. Rendimiento	53
9.4.1. Pruebas con Postman	53
<b>10. Conclusiones:</b>	<b>55</b>
<b>11. Mejoras</b>	<b>56</b>
11.1. Recomendaciones	56
11.1.1. <b>Seguridad de Datos</b>	56
11.1.2. <b>Escalabilidad</b>	56
11.1.3. <b>Arquitectura de Micro-servicios</b>	56
11.1.4. <b>Monitorización y Registro</b>	56
11.1.5. <b>Integración de Sistemas Externos</b>	56

PLATAFORMA PARA LA GESTIÓN DE INFORMACIÓN EN LOS BENEFICIARIOS	7
11.2. Mejoras a Futuro:	57
11.2.1. <b>Machine Learning y Analítica Avanzada</b>	57
11.2.2. <b>Automatización Inteligente</b>	57
11.2.3. <b>Geo-localización</b>	57
11.2.4. <b>Participación Ciudadana</b>	57
11.2.5. <b>Mejora de la Experiencia del Usuario</b>	57
<b>Referencias Bibliográficas</b>	<b>58</b>
<b>Apéndices</b>	<b>60</b>

### Lista de Figuras

Figura 1.	muestra de observatorio social. Fuente: Referencia (de Desarrollo Social)	18
Figura 2.	Explicación visual del Modelo-Vista-Controlador. Fuente: (16)	27
Figura 3.	Esquema Básico De aplicación Fuente: (Autores)	29
Figura 4.	Tendencias para diferentes framework. Fuente: Referencia (8)	31
Figura 5.	Monolito modular esquema Fuente: Referencia (23)	32
Figura 6.	Tendencias para diferentes Arquitecturas de software. Fuente: Referencia (6)	33
Figura 7.	Acceso a datos en la arquitectura escogida. Fuente: (Autores)	33
Figura 8.	Modelo de ciclo de vida iterativo. Fuente: Referencia (21), Pág. 30	35
Figura 9.	Casos de uso general de la plataforma.	37
Figura 10.	Casos de uso para gestionar perfil de usuarios.	38
Figura 11.	Casos de uso para caracterizaciones.	39
Figura 12.	Esquema estructural nombrado Fuente: (modificado de figura 5)	40
Figura 13.	Muestra de capas en el módulo (librerías). Fuente: (Autores)	41
Figura 14.	Relaciones entre capas de módulos Fuente: (Autores)	43
Figura 15.	Imagen de diferentes configuraciones para bases de datos. Fuente:(Autores)	44
Figura 16.	Imagen de Inicio al aplicativo. Fuente:(Autores)	45
Figura 17.	Imagen de la pagina principal del aplicativo. Fuente:(Autores)	45

Figura 18.	Imagen de vista Inicial del formulario en aplicativo. Fuente:(Autores)	46
Figura 19.	Imagen de la pagina de búsquedas de caracterizaciones. Fuente:(Autores)	46
Figura 20.	Gráfica de caracterizaciones discriminadas por población	51
Figura 21.	Token generado por el API mostrado desde (20)	52
Figura 22.	Tiempo de espera largo en prueba 4.40[s] promedio. Fuente:Autores	54
Figura 23.	Tiempo de espera en pruebas posteriores 118[ms] promedio. Fuente:Autores	54

**Lista de Tablas**

Tabla 1.	Lista de requerimientos funcionales	36
Tabla 2.	Lista de requerimientos no funcionales	36
Tabla 3.	personas que fueron caracterizadas en el transcurso de un mes	47
Tabla 4.	Tiempos en promedio de diligenciamiento caracterizaciones	48
Tabla 5.	Errores en promedio de al diligenciar caracterizaciones	50
Tabla 6.	características generales de las dos formas de recolectar datos	51

### Lista de Apéndices

	<b>pág.</b>
Apéndice A. mockup de vistas inicial	61
Apéndice B. Tabla de presupuesto	62
Apéndice C. Cronograma de Actividades	63
Apéndice D. Diagrama de Entidad-Relación para Base de Datos	64
Apéndice E. Contenido de token Generado Como Respuesta de API validado desde (20)	65

## Glosario

**Plataforma:** en este contexto, se refiere a una infraestructura tecnológica que permite la recopilación, gestión y análisis de datos relacionados con beneficios sociales.

**Gestión de Beneficios Sociales:** el proceso de administrar y supervisar los programas y recursos públicos destinados a brindar asistencia social a los ciudadanos necesitados.

**Administración Municipal:** la gestión y supervisión de asuntos locales y servicios públicos a nivel de una municipalidad o alcaldía.

**Gestión de Datos de Beneficiarios:** la recopilación, almacenamiento, actualización y protección de información personal y económica de los ciudadanos que reciben beneficios sociales.

**Transparencia:** la cualidad de ser claro, abierto y accesible en la gestión de recursos públicos, de modo que los ciudadanos puedan comprender cómo se utilizan los fondos.

**Verificación de Beneficios Sociales:** el proceso de confirmar la elegibilidad de un ciudadano para recibir beneficios sociales según ciertos criterios establecidos.

**Asignación de Beneficios Sociales:** la distribución y entrega de recursos o asistencia social a los beneficiarios aprobados.

**Gestión de Programas Sociales:** la planificación, ejecución y supervisión de programas diseñados para abordar necesidades sociales específicas.

**Optimización:** el proceso de mejorar la eficiencia y el rendimiento de un sistema o proceso existente.

**Calidad de Vida:** el nivel de bienestar y satisfacción experimentado por los ciudadanos en su entorno, que puede mejorarse a través de programas sociales.

**Precisión de la Asignación:** la capacidad de asignar beneficios sociales de manera exacta y sin errores.

**Innovadora:** que introduce nuevas ideas, métodos o tecnologías en la gestión de beneficios sociales.

**Tecnología de la Información (TI):** el conjunto de herramientas, sistemas y recursos informáticos utilizados para gestionar y procesar datos en este contexto.

**Seguimiento:** la monitorización y supervisión continua de los beneficiarios y sus datos para garantizar que se cumplan los requisitos y plazos.

**Recopilación de Datos:** la recolección sistemática de información personal y económica de los ciudadanos para su posterior procesamiento.

**Precedente:** un ejemplo o modelo que establece un estándar para futuras iniciativas similares.

## Resumen

**Título:** Plataforma Web Para La Gestión De Información De Los Usuarios Beneficiarios De Servicios Sociales Adjudicados Por La Secretaría De Desarrollo Social De La Alcaldía De Bucaramanga

**Autores:** Gabriel de Jesús Álvarez Tejada - Fabio Andrey Ojeda Villamizar \*\*

**Palabras Clave:** Plataforma web, software, bases de datos, seguridad, framework, back-end, front-end, Alcaldía de Bucaramanga, Mockup, Servidores, Arquitectura, Sistema de Beneficiarios, Tecnologías.

**Descripción:** Este proyecto tiene como objetivo principal la modernización de la gestión de beneficios sociales a través del desarrollo de una plataforma diseñada específicamente para simplificar la recolección y actualización de datos personales y económicos de los beneficiarios, al mismo tiempo que automatiza los procedimientos de verificación y asignación de estos beneficios.

En un contexto donde la sociedad se vuelve cada vez más digital, la administración eficiente de los beneficios sociales se convierte en una prioridad estratégica. La iniciativa se enfoca en la eliminación de obstáculos burocráticos y en la agilización del acceso a los beneficios, con el propósito de brindar una experiencia más fluida tanto para los ciudadanos como para las entidades encargadas de la administración de estos programas. La plataforma en desarrollo permitirá a los beneficiarios proporcionar y actualizar sus datos de manera sencilla y segura, reduciendo significativamente la carga administrativa y asegurando la precisión de la información. Además, la automatización de los procesos de verificación y asignación acelerará los tiempos de respuesta, garantizando la entrega oportuna de beneficios a aquellos que cumplan con los requisitos correspondientes. En resumen, este proyecto representa un avance significativo en la administración de beneficios sociales al simplificar la gestión de datos y optimizar los procedimientos de asignación, en beneficio tanto de los ciudadanos como de las instituciones responsables de brindar estos servicios.

---

\*\* Facultad de Ingenierías Físico-Mecánicas. Ingeniería de Sistemas. Director: Jathinson Meneses Mendoza, Magister Aplicación y Desarrollo de Software - Co-director: Henry Andrés Jiménez Herrera, Magíster en Sistemas e Informática.

## Abstract

**Title:** Web Platform for the Management of Information of Users Beneficiaries of Social Services Awarded by the Department of Social Development of the Municipality of Bucaramanga

**Authors:** Gabriel de Jesús Álvarez Tejada - Fabio Andrey Ojeda Villamizar \*\*

**Keywords:** Web platform, software, databases, security, framework, back-end, front-end, Mayor's Office of Bucaramanga, Architecture, Beneficiary System, Technologies.

**Description:** This project aims to modernize the management of social benefits through the development of a platform designed specifically to simplify the collection and update of personal and economic data of beneficiaries while automating the procedures for verification and allocation of these benefits.

In a context where society is becoming increasingly digital, the efficient administration of social benefits becomes a strategic priority. The initiative focuses on removing bureaucratic obstacles and streamlining access to benefits with the purpose of providing a smoother experience for both citizens and the entities responsible for administering these programs. The platform in development will allow beneficiaries to provide and update their data in a simple and secure manner, significantly reducing administrative burdens and ensuring data accuracy. Furthermore, the automation of verification and allocation processes will expedite response times, ensuring the timely delivery of benefits to those who meet the respective requirements.

In summary, this project represents a significant advancement in the administration of social benefits by simplifying data management and optimizing allocation procedures, benefiting both citizens and the institutions responsible for providing these services.

---

\*\* Faculty of Physical-Mechanical Engineering. School of Systems Engineering. Director: Jathinson Meneses Mendoza, Master's in Software Management, Application, and Development - Co-director: Henry Andrés Jiménez Herrera, Master's in Systems Engineering and Computer Science

## **Introducción**

En un esfuerzo por abordar desafíos en la gestión de beneficios sociales y ofrecer un apoyo adicional a los ciudadanos, se presenta un proyecto en la administración municipal. Hasta la fecha, no se ha implementado una plataforma similar en la alcaldía. Esta iniciativa surge de la necesidad de contar con una solución integral para la gestión de datos de beneficiarios, ya que mantener un seguimiento de quienes requieren asistencia social se ha vuelto cada vez más complejo.

La plataforma se diseñó para simplificar tanto la recopilación como la actualización de información personal y económica relevante de los beneficiarios, sino que también revolucionará la forma en que se verifican y asignan los beneficios sociales. Este avance no solo promete transparencia y eficacia en la distribución de recursos sociales, sino que también establece un precedente en la alcaldía como una solución pionera en su género. Al hacerlo, se sientan las bases para una gestión más eficiente y equitativa de los programas sociales en beneficio de toda la comunidad.

En resumen, el proyecto no solo se centra en acelerar la toma de decisiones en la asignación de beneficios sociales, sino que también representa un avance hacia una gestión más eficiente y equitativa de los recursos públicos. Esta plataforma innovadora tiene como objetivo mejorar la calidad de vida de los ciudadanos, garantizando la entrega precisa y oportuna de los recursos a los que tienen derecho.

## **1. Generalidades del proyecto**

### **1.1. Contexto**

Este proyecto se lleva a cabo en un contexto en el que la gestión de beneficios sociales se ha vuelto cada vez más compleja, lo que ha generado la necesidad de una solución innovadora. La creciente complejidad en el seguimiento preciso de quienes requieren asistencia social ha resalta-do la importancia de una plataforma integral para la gestión de datos de beneficiarios. Es relevante destacar que, hasta la fecha, en nuestra alcaldía no se ha implementado una plataforma similar, lo que hace que esta iniciativa sea aún más pertinente y necesaria.

El proyecto que se está presentando se lleva a cabo dentro del marco del Observatorio Municipal de Bienestar Social, que es una iniciativa adelantada por la secretaria de desarrollo económico y social de Bucaramanga como se muestra en la figura 1, en su fase de “sistema de identificación de beneficiarios”.

### **1.2. Descripción de la entidad donde se realizó la práctica**

Aunque el trabajo se realiza en virtud de un acuerdo entre la alcaldía y la universidad, se lleva a cabo de manera específica en la Secretaría de Desarrollo Económico y Social.

El siguiente es un fragmento de texto citado de la página web oficial:

*“La secretaría de desarrollo social tiene a su cargo la atención de población vulnerable de la ciudad de Bucaramanga. Esto lo realiza a través de la planificación*



Figura 1. muestra de observatorio social. Fuente: Referencia (de Desarrollo Social)

*de actividades por grupos poblacionales desde donde focaliza las diferentes actividades y servicios para promover la igualdad de derechos y oportunidades.” Fuente: (de Desarrollo Social)*

### 1.2.1. Misión.

*“ Somos una entidad territorial al servicio de los ciudadanos que garantiza el ejercicio de sus derechos y el cumplimiento de sus deberes, promueve el bienestar y desarrollo humano, con oportunidad, equidad, transparencia, responsabilidad social, económica y ambiental, soportada en una gestión institucional innovadora y colaborativa, con un talento humano comprometido con su labor como Servidor Público. ”*

Fuente: (Alcaldía)

### 1.2.2. Visión.

*“ Bucaramanga ciudad global para la gente, comprometida con el bienestar, la protección del medio ambiente, los ecosistemas estratégicos y la seguridad de sus habitantes; de economía pujante y oportunidades para todos. Promueve ciudadanos responsables, activos y comprometidos con su ciudad. Innovadora, que goza de infraestructura tecnológica facilitando la toma de decisiones y brindando acceso a sistemas de educación, salud y movilidad de calidad 24 x 7. ” Fuente: (Alcaldía)*

## 2. Planteamiento del problema

### 2.1. Justificación

En la actualidad muchas personas no tienen la oportunidad de formar parte del servicio de beneficiarios debido a un proceso lento agravado por la gran cantidad de datos que en su mayoría se procesan de manera manual.

Con la creación de la herramienta que compone el diseño, implementación y montaje de una plataforma web se busca mejorar la agilización en el registro de nuevos usuarios que formen parte del sistema de beneficiarios de la Secretaría de Desarrollo Social, así como operadores autorizados por la misma que lleven un control de la plataforma web, permitiendo el registro de usuarios, así como la verificación de los datos de los beneficiarios.

Además, con este proyecto se busca contribuir en el proceso de desarrollo tecnológico que desde la administración se ha venido impulsando en pos de la mejora en la prestación y atención de los servicios por parte de las instituciones regionales y nacionales aportando nuevas herramientas tecnológicas que buscan mejorar el bienestar del ciudadano(a).

**2.1.1. Estado del arte.** En respuesta a los desafíos persistentes en la administración de beneficios sociales y con el objetivo de brindar un apoyo más eficiente a los ciudadanos, se ha concebido un proyecto en el que se apoya la gestión municipal. La falta de una plataforma similar en nuestra alcaldía hasta la fecha ha motivado la creación de esta iniciativa.

Esta respuesta a una necesidad imperante busca simplificar la gestión de datos de beneficiarios y contribuir a la modernización de los procesos de verificación y asignación de beneficios sociales.

**2.1.2. Impacto.** La iniciativa propuesta tiene un impacto significativo en varios aspectos:

- **Simplificación en el Manejo al Recopilar y Actualizar Datos:** La plataforma propuesta propone simplificar considerablemente la recopilación y actualización de información personal y económica de los beneficiarios. Esto reducirá la carga administrativa y mejorará la precisión de los registros. Esto aliviará la complejidad que supone el seguimiento preciso de aquellos que requieren asistencia social.
- **Mayor Transparencia y Eficacia:** La implementación de esta plataforma se traducirá en una asignación más transparente y eficaz de los beneficios sociales. Esto garantizará que los recursos se distribuyan de manera equitativa y justa entre los ciudadanos que los necesitan. Esto con el fin de facilitar una mayor transparencia y eficacia en la distribución de recursos sociales.
- **Precedente en Innovación Municipal:** El proyecto establece un precedente en nuestra alcaldía como una solución pionera en su género. Esto demuestra nuestro compromiso con la innovación en la administración pública y posiciona a nuestra comunidad como líder en la gestión de beneficios sociales.

**2.1.3. Viabilidad.** La viabilidad de este proyecto se basa en su capacidad para abordar los desafíos reales que enfrenta la gestión de beneficios sociales.

Al simplificar la gestión y actualización de datos, así como al mejorar la verificación y asignación de beneficios, se espera que se reduzcan los costos administrativos y se optimice el uso de los recursos públicos.

La plataforma propuesta también puede aumentar la eficiencia en la toma de decisiones al proporcionar datos actualizados. Además, al tratarse de una iniciativa sin precedentes en la alcaldía, la viabilidad se respalda en la voluntad de innovar y mejorar la gestión de programas sociales.

- **Tecnología y Recursos:** La viabilidad tecnológica es esencial para el éxito del proyecto. Esto incluye la disponibilidad de recursos técnicos y financieros, así como la capacitación del personal para utilizar la plataforma.
- **Legalidad y Cumplimiento:** Se deben abordar cuestiones legales y de cumplimiento para garantizar que la plataforma cumpla con las regulaciones y proteja la privacidad de los beneficiarios.

### **3. Objetivos**

#### **3.1. Objetivo general**

Diseñar e Implementar una plataforma web de gestión de información, enfocada a usuarios y/o aspirantes a servicios sociales adjudicados por la secretaría de desarrollo social de la alcaldía de Bucaramanga

#### **3.2. Objetivos específicos**

Identificar los requerimientos necesarios relacionados con el software y el hardware disponibles para el desarrollo del proyecto teniendo en cuenta la infraestructura de la secretaría de desarrollo de la alcaldía de Bucaramanga.

Diseñar una arquitectura para el software web basada en los requerimientos y software identificados para la plataforma de la secretaría de desarrollo de la alcaldía de Bucaramanga.

Implementar un prototipo para el software web de acuerdo a la Arquitectura y Requerimientos identificados para la gestión del proceso de la secretaría de desarrollo de la alcaldía de Bucaramanga.

Validación de la plataforma web por medio de una serie de pruebas funcionales con el apoyo de los usuarios del proceso de la secretaría de desarrollo de la alcaldía de Bucaramanga.

## 4. Marco de referencia

### 4.1. Arquitectura De Software

Actualmente no hay una sola definición aceptada que explique la arquitectura de software. Sin embargo, hay ideas que permiten acercarse al concepto y puede ser representado por un diagrama: círculos o cuadros unidos por alguna línea, alguna identificación de estos elementos y poca cosa más (12). Toda definición textual toma como base este consenso dando una interpretación a los diagramas para dar coherencia.

Las primeras definiciones hacían referencia a componentes o elementos relacionados entre sí, tiempo después las propiedades de los elementos fueron agregados como parte de la arquitectura, de ahí en adelante cada autor da una interpretación de lo que considera arquitectura de software, una de las definiciones más aceptada fue la propuesta por Bass: “Una arquitectura de software de un programa o un sistema computacional es la estructura del sistema, la cual comprende elementos del software, las propiedades externamente de esos elementos y las relaciones entre ellos (4).

En lo que respecta al mundo del desarrollo web, se considera a la arquitectura de software una disciplina de vital importancia, una de las razones es el constante cambio (mejoras) día a día del internet y la necesidad de las aplicaciones de adaptarse fácilmente a todos estos cambios. A razón de esto, los arquitectos de software deben escoger el patrón y herramientas adecuadas para la aplicación web a desarrollar (25).

**4.1.1. Patrones De Arquitectura De Software.** Un patrón de software es la forma de reutilizar soluciones a problemas recurrentes. Es decir, un patrón de arquitectura de software puede definirse como la forma de estructuras de diseño probadas para solucionar problemas en el desarrollo de aplicaciones (24).

*“El arquitecto de software debe decidir cómo reutilizar un patrón, hacer los ajustes necesarios al contexto específico, a las restricciones del problema, y a la solución que está diseñando.”* (9)

Patrones de arquitectura de software más comunes en el desarrollo de software:

- Patrón de capas
- Patrón Clientes-Servidor
- Patrón Maestro-Esclavo
- Patrón de filtro de tubería
- Patrón de intermediario
- Patrón de igual a igual
- Patrón de bus de evento
- Modelo-Vista-Controlador
- Patrón de pizarra
- Patrón de intérprete

Cada patrón tiene sus pros y contras (17). Sin embargo, para el desarrollo de este proyecto se escogió el patrón modelo-vista-controlador al ser el mejor que se adecúa a las necesidades requeridas para la realización del mismo.

### **Modelo-Vista-Controlador(MVC)**

El modelo-vista-controlador es un patrón de arquitectura de software basado en la programación orientada a objetos cuya característica principal radica en que el Modelo, la Vista y el Controlador se consideran entidades separadas. Por lo tanto, cada componente del patrón se puede programar de manera separada y posteriormente agruparlos en tiempo de ejecución, sí uno de los componentes llega a presentar fallas puede ser reemplazado sin afectar al resto de las partes.

El M.V.C.2 cuenta con tres entidades que son (14):

- **Modelo:** Es el objeto que se encarga de representar los datos del programa. Su función es la de manejar los datos, así como controlar todas las transformaciones que puedan presentar.
- **Vista:** Es el objeto que se encarga de manejar una presentación visual de los datos representados por el Modelo. Su función es la de mostrar en pantalla a los usuarios los datos que maneja el Modelo.
- **Controlador:** Es el objeto que se encarga de proporcionar lógica a las acciones del usuario. Su función es la de actuar sobre los datos representados por el Modelo, centrando toda la interacción entre el Modelo y la vista.

En el ámbito del desarrollo web el patrón MVC es el más óptimo puesto que permite separar

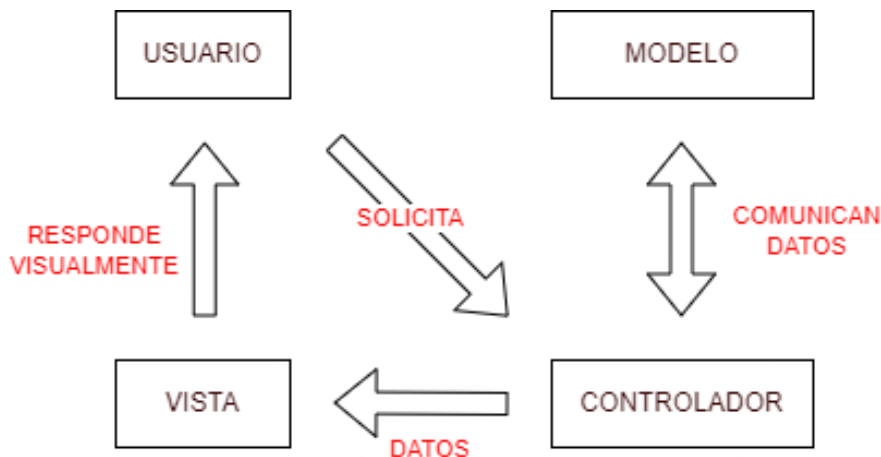


Figura 2. Explicación visual del Modelo-Vista-Controlador. Fuente: (16)

en capas las partes de una aplicación: “Este patrón soluciona los inconvenientes debidos al uso de interfaces gráficas en aplicaciones, pero adicionalmente permite definir claramente las capas que se aplican en la arquitectura Cliente-Servidor, muy común en Internet” (25)

**Desarrollo De Aplicaciones Web** El desarrollo de aplicaciones web es esencial para la creación de software que trabaje por medio del navegador web. El desarrollo de aplicaciones web va de la mano con el desarrollo de aplicaciones móviles para brindar a los usuarios la compatibilidad multiplataforma para un acceso desde computadores, teléfonos y tabletas (27).

El proceso de desarrollo de aplicaciones web es un proceso complejo que se divide en varias secciones como la planificación y el diseño más adecuado para la aplicación web. Un equipo de desarrollo de aplicaciones web se encarga de implementar características vitales con el objetivo de mejorar el rendimiento, funcionalidad y seguridad de la aplicación web para brindar una experiencia amigable, intuitiva y segura a los usuarios (15).

Para la infraestructura digital moderna el desarrollo de aplicaciones web se ha vuelto una parte

importante. Los desarrolladores web por medio de lenguajes de programación, marcos de aplicaciones web, aplicaciones móviles y técnicas de desarrollo web se encargan de la creación de aplicaciones web seguras, funcionales, accesibles y que se adapten fácilmente a varios sectores, así como dar una experiencia agradable al usuario (27).

Una aplicación web es diferente de un sitio web: un sitio web está diseñado para ser estático con el propósito de brindar información a los usuarios; una aplicación web se diseña para ser dinámica y permitir a los usuarios realizar acciones en ellas (22).

**4.1.2. Aspectos.** para el desarrollo de aplicaciones web el frontend y el backend son aspectos vitales para su implementación, por eso es importante que los desarrolladores tengan un buen conocimiento sobre lo que representan estos términos y de las tecnologías para su uso (18).

Si bien el esquema básico de una aplicación cercano a la necesidad (fig. 3) sin embargo la rama en la que se usa un servidor SMTP no forma parte del proyecto pero se dejan facilidades para ello.

**Frontend.** El frontend es la capa de presentación de las aplicaciones web, es decir, es la interfaz visual que le permite al usuario interactuar con la aplicación (7). El encargado del frontend tiene la responsabilidad de crear una experiencia visual que resulte agradable a la vista y a su vez crear un diseño que sea fácil de entender mejorando la interacción para las personas que la vayan a usar.

Para lograr estos objetivos el desarrollador del frontend cuenta con todas las tecnologías que trabajan del lado del navegador web, que se generalizan en tres lenguajes: HTML, CSS y JavaScript

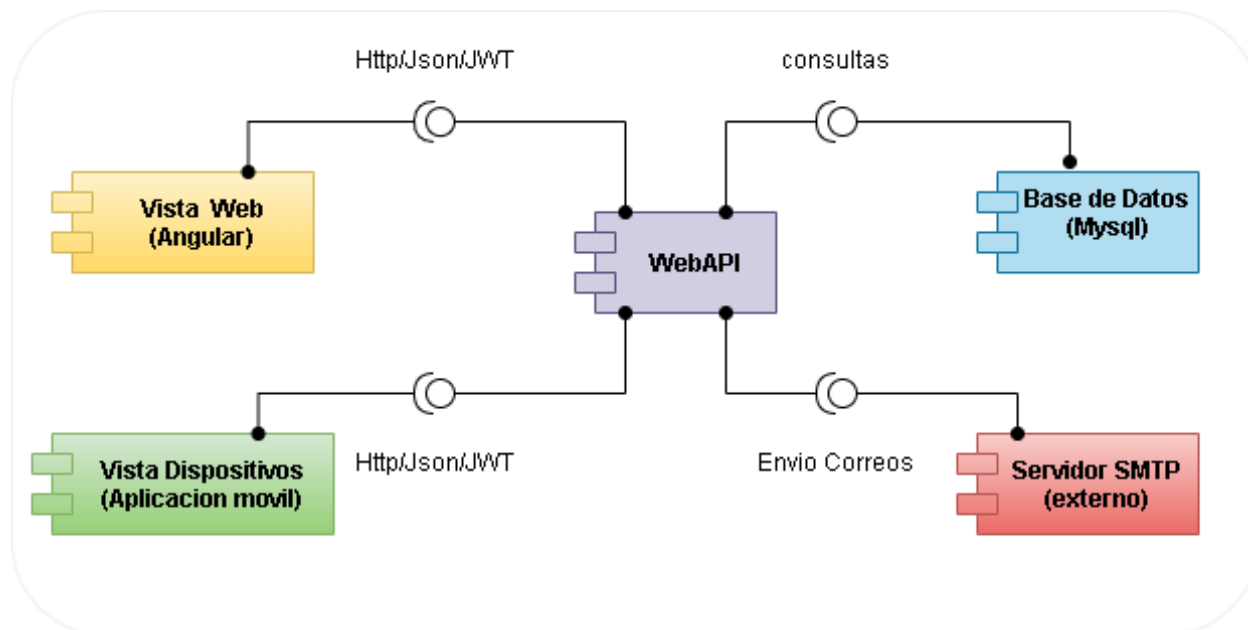


Figura 3. Esquema Básico De aplicación Fuente: (Autores)

(7). Estos lenguajes son los que escogimos para la realización del proyecto, la parte estructural se realizará en HTML y para el apartado gráfico se utiliza CSS.

*“HTML es un lenguaje de marcas... que se encarga en exclusiva de los contenidos de las páginas y aplicaciones Web, así como de su distribución... CSS se encarga de la interfaz gráfica a través de sus selectores y propiedades.”* (1).

**Backend.** El backend es la capa de datos de las aplicaciones web, es decir, es la parte que le da sentido a las acciones de los usuarios en el frontend; donde se manejan los datos de la aplicación. El encargado del backend tiene la responsabilidad de hacer que todo funcione: montando la base de datos, asegurando el procesamiento de datos, gestionando la seguridad y dando acceso al servidor para las solicitudes del navegador web (5).

Para lograr estos objetivos el desarrollador del backend se especializa en lenguajes que trabajan del lado del servidor como son: Java y PHP por nombrar unos; así mismo, debe conocer los lenguajes que interactúan con la base de datos, en nuestro proyecto nos decantamos por usar MySQL para administrar la base de datos.

*“MySQL es un sistema gestor de bases de datos muy conocido y ampliamente usado por su simplicidad y notable rendimiento... (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo” (26).*

**4.1.3. Frameworks.** En el desarrollo de aplicaciones web hay herramientas que facilitan el trabajo, entre ellos tenemos a los frameworks caracterizados por darle consistencia a la escritura de código y evitar mandar líneas de programación repetidamente por sus librerías previamente creadas, así como asegurar las buenas prácticas en el trabajo (13). Los frameworks contienen librerías, extensiones y plugins que facilitan el proceso de creación de aplicaciones web, es decir, son estructuras que simplifican la resolución de una tarea al brindar un conjunto de herramientas y módulos que agilizan el desarrollo al dar solución a funcionalidades requeridas en anteriores aplicaciones (13).

## **4.2. Selección**

Teniendo en cuenta que la arquitectura es la columna vertebral de cualquier proyecto de software, y su éxito depende en gran medida de la elección adecuada de patrones de arquitectura y frameworks, proporcionan soluciones probadas para desafíos comunes de diseño, mientras que

los frameworks, como Angular, ofrecen herramientas predefinidas para agilizar el desarrollo. Estos tres elementos se entrelazan de manera crítica en el mundo de la arquitectura, permitiendo a los desarrolladores crear estructuras sólidas y eficientes.

**4.2.1. Framework para Back-end.** El Framework En el que se realizó este proyecto para el aspecto del backend es ASP.NET 6 el cual representa la evolución de Microsoft en el desarrollo web, destacándose por su mayor rendimiento, modularidad, compatibilidad multiplataforma y mejoras en seguridad. Esta versión permite a los desarrolladores construir aplicaciones web modernas y escalables con facilidad, ofreciendo una mayor flexibilidad y rendimiento, además de una integración más estrecha con el ecosistema de .NET 6, lo que la convierte en una opción poderosa y versátil para proyectos web de vanguardia.



Figura 4. Tendencias para diferentes framework. Fuente: Referencia (8)

**4.2.2. Framework para Front-end.** Un framework bastante usado en el lenguaje de programación de JavaScript desarrollado en TypeScript por desarrolladores de google, es Angular; que por sus características para el montaje del frontend lo escogimos para el desarrollo del

proyecto.

“Angular es un framework basado en el lenguaje Javascript... usa el patrón MVC con un enlazamiento bidireccional de datos entre HTML y JavaScript” (10).

**4.2.3. Arquitectura para Back-end.** La arquitectura escogida es “monolitos modulares” (fig. 5) la cual es un enfoque de diseño de software que combina la simplicidad de un monolito con la capacidad de modularidad. En este enfoque, una aplicación monolítica se divide en módulos independientes y bien definidos, lo que permite el desarrollo y mantenimiento por separado de cada uno de ellos.

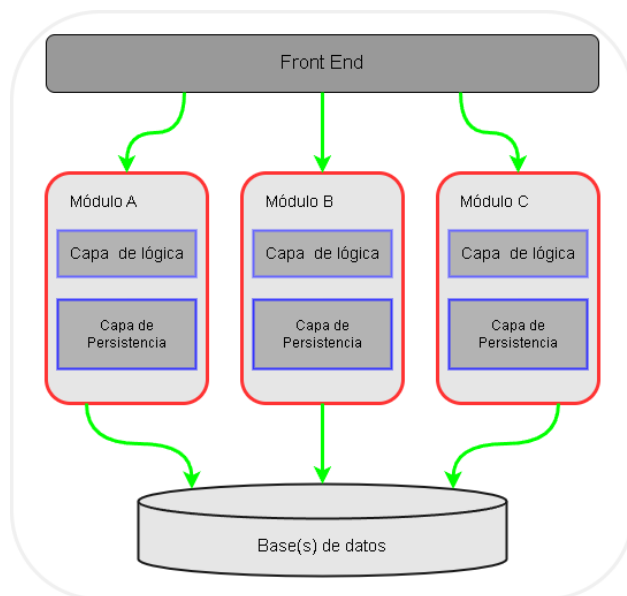


Figura 5. Monolito modular esquema Fuente: Referencia (23)

Estos módulos pueden comunicarse entre sí a través de interfaces específicas o acceso público (Fig. 7), lo que facilita la escalabilidad y la evolución de la aplicación sin sacrificar la cohesión

del código. Esto brinda a los equipos de desarrollo la flexibilidad de trabajar en partes específicas de la aplicación de manera aislada, al tiempo que conserva la eficiencia y la gestión centralizada del monolito.

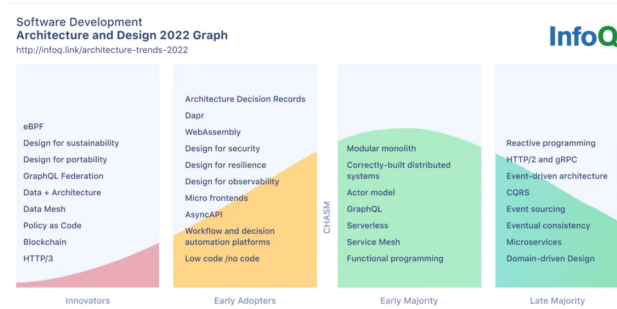


Figura 6. Tendencias para diferentes Arquitecturas de software. Fuente: Referencia (6)

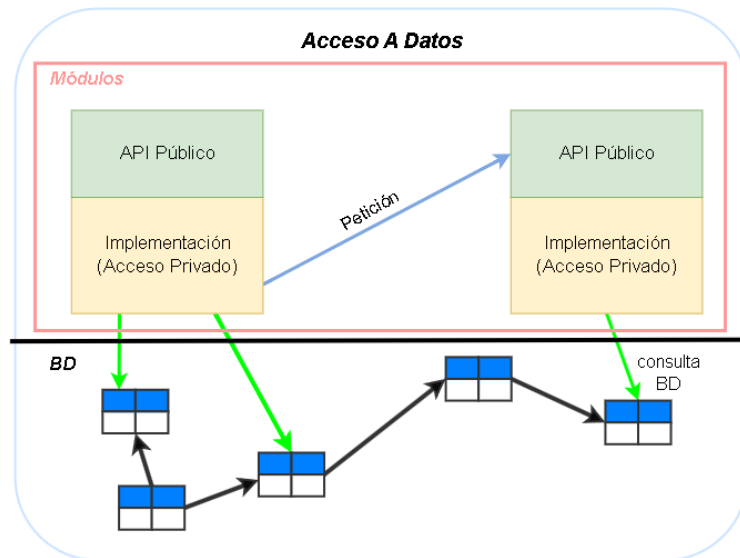


Figura 7. Acceso a datos en la arquitectura escogida. Fuente: (Autores)

## 5. Metodología

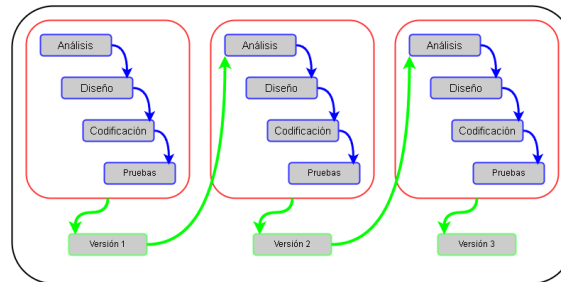
Para el desarrollo del aplicativo web, se optó por utilizar la metodología de desarrollo iterativa debido a su gestión ágil para un rápido avance y su adaptabilidad para agregar cambios en fases finales en la creación de software. Esta metodología surge como mejora de los problemas presentes en la metodología de desarrollo en cascada (19).

La metodología de desarrollo iterativa se deriva de la metodología en cascada y se caracteriza por dividir los requisitos de un proyecto en pequeñas tareas llamadas iteraciones que se desarrollan en plazos cortos de tiempo, que no superan las cuatro semanas. Para que el equipo pueda cumplir con los objetivos, es necesario garantizar que en cada iteración se realicen pruebas y una documentación. De esta forma, se corrigen y previenen errores en las etapas finales del desarrollo. \*

“El tiempo y dinero gastados en la implementación de un diseño fallido, son no recuperables.” Fuente: (28)

---

\* sumado al hecho que los requisitos no están completamente claros por parte del usuario



*Figura 8.* Modelo de ciclo de vida iterativo. Fuente: Referencia (21), Pág. 30

## 6. Requerimientos funcionales y Requerimientos no funcionales

### 6.1. Requerimientos funcionales

En la Tabla 1, se listan los requerimientos funcionales del sistema.

Requerimientos funcionales	Descripción
Registrar usuario	Se podrán registrar usuarios ingresando los datos requeridos.
Iniciar Sección	Los usuarios podrán acceder a la plataforma ingresando las credenciales de acceso registradas.
Cerrar Sección	Los usuarios podrán cerrar sección de la plataforma web
Actualizar Datos	Los usuarios podrán actualizar los datos de su cuenta.
Cambiar contraseña	Los usuarios podrán cambiar la contraseña con la cual ingresan a su cuenta. No se puede repetir contraseñas
Eliminar Usuario	Los usuarios podrán eliminar sus cuentas.
Dar de alta	Los usuarios Root podrán validar las peticiones para dar ingreso a los nuevos usuarios.
Caracterizaciones	Los usuarios podrán realizar las caracterizaciones de las poblaciones.
Búsqueda Registros	Los usuarios podrán buscar las caracterizaciones realizadas y revisar su estado.
Editar registros	Los usuarios podrán realizar ediciones a las caracterizaciones de las poblaciones.
Eliminar registros	Los usuarios Root podrán eliminar registros de caracterizaciones.
Descargar informe	Los usuarios Admin y Root podrán descargar registros de caracterizaciones.

Tabla 1

*Lista de requerimientos funcionales*

### 6.2. Requerimientos no funcionales

A continuación, se presenta la Tabla 2 correspondiente a los requerimientos no funcionales del sistema, a diferencia de los requerimientos funcionales, los no funcionales representan características y/o restricciones del sistema.

Requerimientos no funcionales	Descripción
Acceso simultaneo	La plataforma podrá permitir el acceso a varios usuarios sin provocar colisiones.
Interfaz gráfica intuitiva	Se diseñara una interfaz gráfica fácil de entender para los usuarios.
Dominio privado	El dominio privado solo permitirá acceder a la plataforma web a usuarios de la secretaria de desarrollo
Certificados SSL	El certificado SSL hace que el dominio obtenga una comunicación segura por la red. Las paginas que tienen este certificado se reconocen por que inician con "https".
Publicacion en servidor por medio de FTP	El protocolo FTP es un servicio que posibilita la transferencia de archivos entre el servidor y un dispositivo, desempeñando un papel fundamental para publicar las dos aplicaciones necesarias para el desarrollo del proyecto (front-end, back-end)
Utilizar el protocolo SSH	El protocolo SSH se encarga de proporcionar una conexión segura al servidor a través de Internet. Esta seguridad se logra mediante la encriptación de datos, lo que previene que terceros puedan acceder a la información.
Compatibilidad con navegadores	La página web debe ser especialmente compatible con los siguientes navegadores: Google Chrome, Microsoft Edge, Opera, Firefox, y cualquier otro navegador basado en Chromium.

Tabla 2

*Lista de requerimientos no funcionales*

## 7. Casos de Uso

### 7.1. Casos de uso

Los casos de uso representan las acciones llevadas a cabo por los distintos tipos de usuarios que utilizan la aplicación. En este contexto, se aborda un único actor, que es el usuario principal. Esto significa que no se contemplan roles adicionales, como el de administrador, con funciones adicionales. Los casos de uso se presentan en diversos diagramas con el fin de proporcionar una mayor claridad.

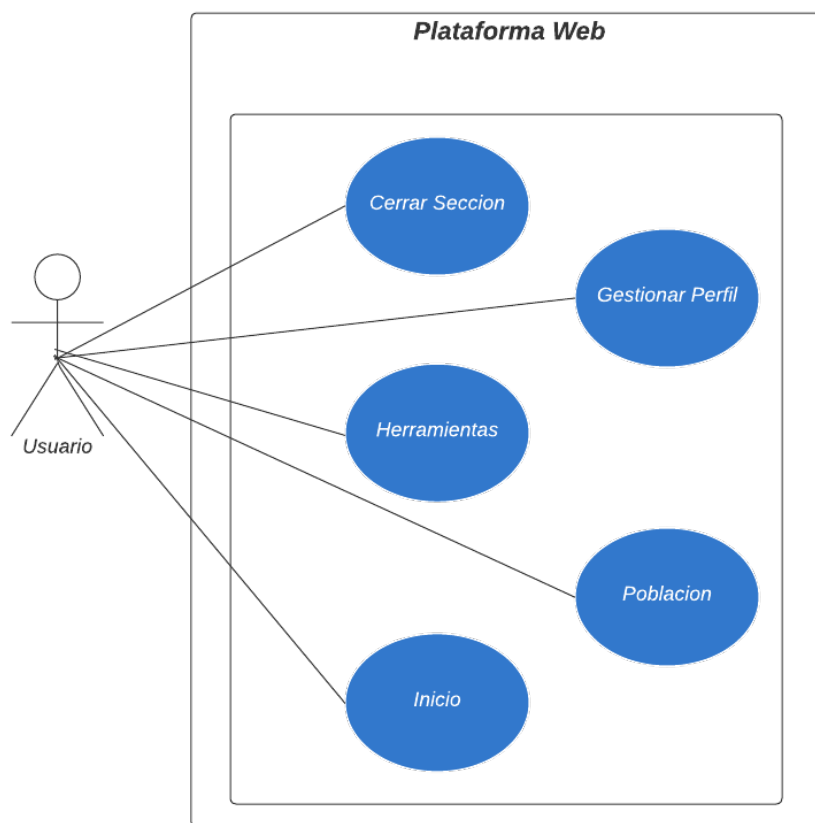


Figura 9. Casos de uso general de la plataforma.

En la Figura 9, muestra el diagrama general de casos de uso de la plataforma web. De este modo, las acciones que el usuario lleva a cabo en la plataforma se detallan en los casos de uso posteriores, lo que facilita la comprensión del funcionamiento global del sistema.

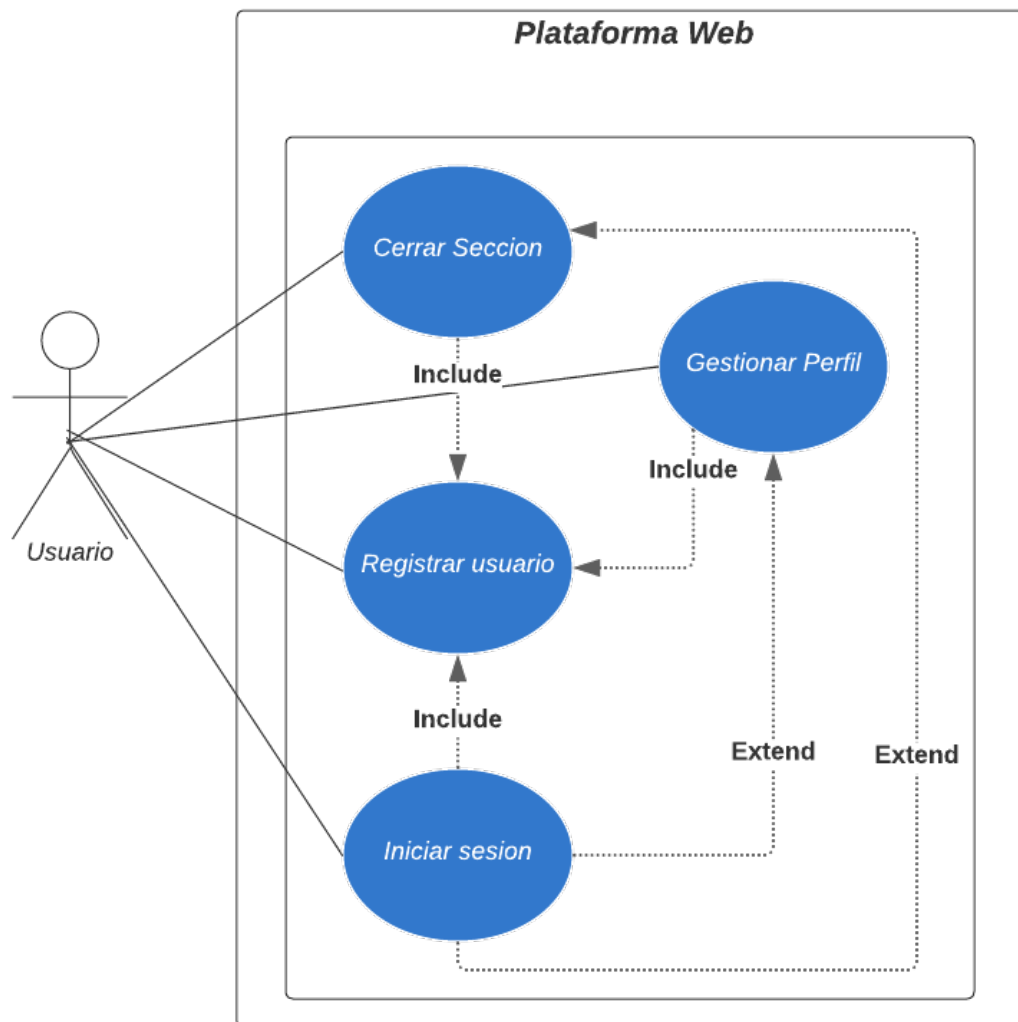


Figura 10. Casos de uso para gestionar perfil de usuarios.

La Figura 10 ilustra el esquema de casos de uso destinado a los usuarios, donde todas las funciones disponibles para los usuarios requieren la creación de una cuenta de usuario. En el caso de gestionar el perfil del usuario y cerrar sesión, implica haber iniciado sesión previamente.

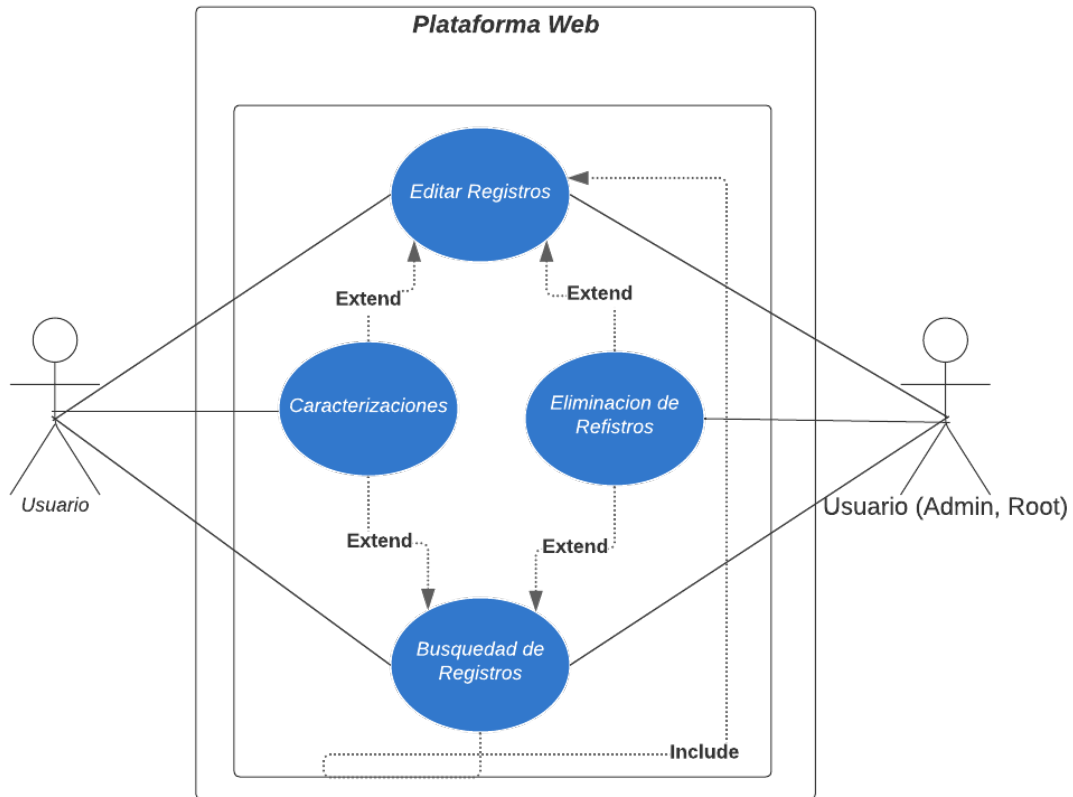


Figura 11. Casos de uso para caracterizaciones.

La Figura 11 muestra los casos de uso que abordan las caracterizaciones. A diferencia de la figura anterior, se presentan dos actores: usuarios y los usuarios admin para mejorar la claridad y la simplicidad del diagrama.

## 8. Arquitectura

### 8.1. Back-end (ASP.NET Core 6)

El back-end de la aplicación se basa en ASP.NET Core 6 y sigue una arquitectura de monolito modular. Se ha organizado en módulos lógicos que abordan diversas funcionalidades.

**8.1.1. Arquitectura Monolítica Modular.** La arquitectura monolítica modular se ha implementado para dividir la aplicación en módulos que manejan funciones específicas, en este caso se hace mención de 3 módulos que son los que se implementaron como se muestra en la figura 12.

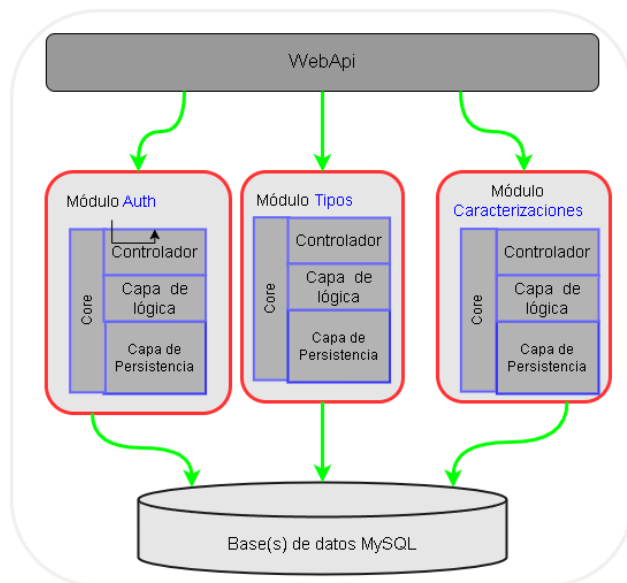


Figura 12. Esquema estructural nombrado Fuente: (modificado de figura 5)

Los módulos que se implementaron se nombraron como:

- Auth: se encarga de la gestión de usuarios incluyendo acceso, permisos y autorizaciones.

- Tipos: se encarga de las listas y menús desplegables, es un modulo parametrizable
- Caracterización: es el módulo que se encarga de la gestión y la manipulación de la información.

Internamente se optó por utilizar la misma estructura en los módulos para hacer el desarrollo más ágil.

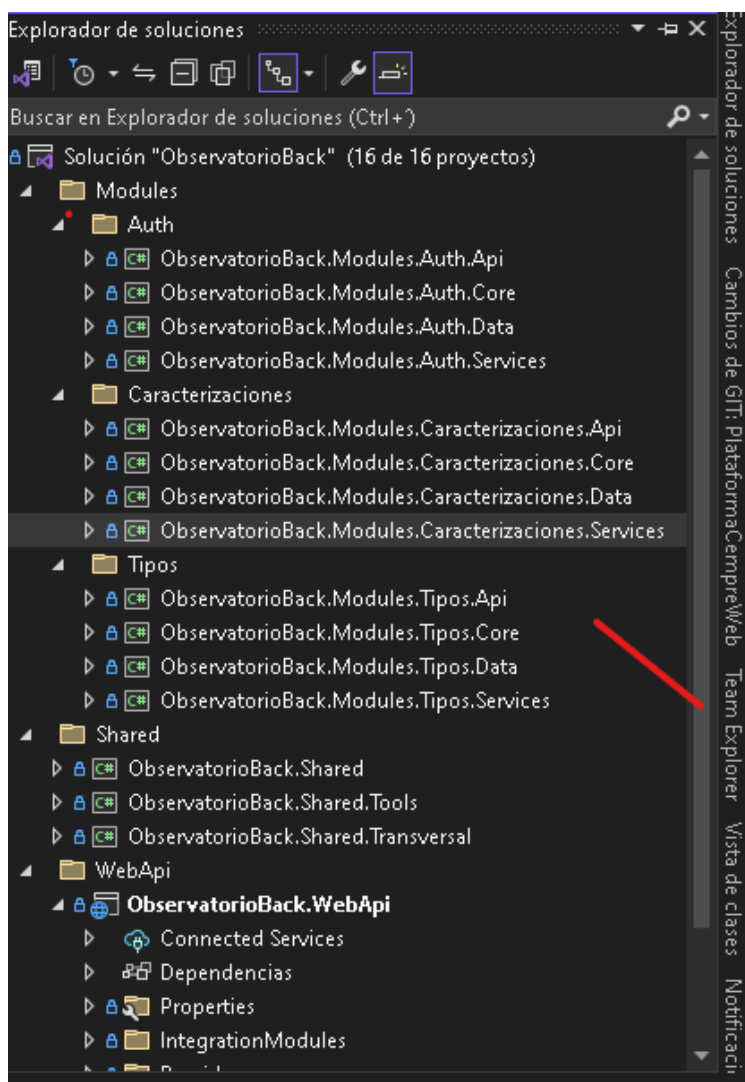


Figura 13. Muestra de capas en el módulo (librerías). Fuente: (Autores)

En la figura 13 se muestran los módulos mencionados previamente, además de las capas internas.

- **.Api** es la capa mas externa del módulo, y su función es recibir las peticiones ya sea desde el cliente (front) o desde otro módulo
- **.Services** es la capa de lógica de negocios
- **.Core** es la capa de acceso a datos que se comunica con la base de datos
- **.Data** es la capa que contiene el contexto

En este caso cada módulo del back tiene la estructura que se muestra a continuación (fig. 14), en cada una de las relaciones se utiliza la inyección de dependencias por medio de contratos (interfaces)

## 8.2. Front-end (Angular 13)

El front-end, desarrollado en Angular 13 utilizando la librería visual “material”, proporciona una interfaz de usuario que interactúa con el back-end:

1. **Interfaz de Usuario Responsiva:** Se ha creado una interfaz de usuario que se adapta a diferentes dispositivos, como móviles y computadoras de escritorio. 16
2. **Formularios y Validaciones:** Se utilizan formularios para recopilar datos de usuarios y se aplican validaciones para asegurar la precisión de los datos. 18
3. **Acceso a Vistas:** Se utiliza la carga perezosa(lazy loading), es una técnica de carga diferida que permite cargar módulos y componentes solo cuando son necesarios, mejorando la

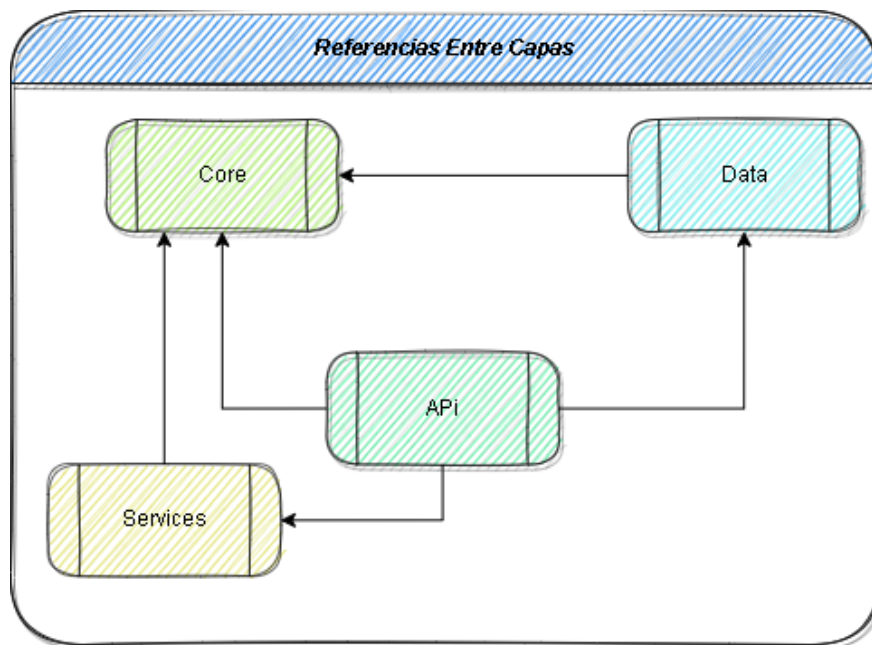


Figura 14. Relaciones entre capas de módulos Fuente: (Autores)

velocidad de carga inicial de la aplicación al dividirla en segmentos que se descargan bajo demanda.

4. **Seguridad del Front-end:** Se han implementado medidas de seguridad en el front-end para proteger contra amenazas como la inyección de scripts maliciosos.
5. **Rutas Amigables:** en Angular, las rutas amigables se utilizan para definir URL legibles y significativas en lugar de URLs con parámetros complejos. Estas rutas se configuran mediante el módulo “RouterModule” y permiten una navegación más intuitiva y una mejor experiencia de usuario en una aplicación web. Referencia:(3)

### 8.3. Base de Datos (MySQL):

La base de datos MySQL sirve como el repositorio de datos para la aplicación:

1. **Optimización de Consultas:** Se han implementado índices y optimizado las consultas para mejorar la velocidad de lectura y escritura.
2. **Seguridad de la Base de Datos:** Se han configurado permisos y roles en la base de datos para garantizar la seguridad y restringir el acceso no autorizado.

La forma en la que se puede organizar la base de datos varia, y se puede resumir en la figura 15.

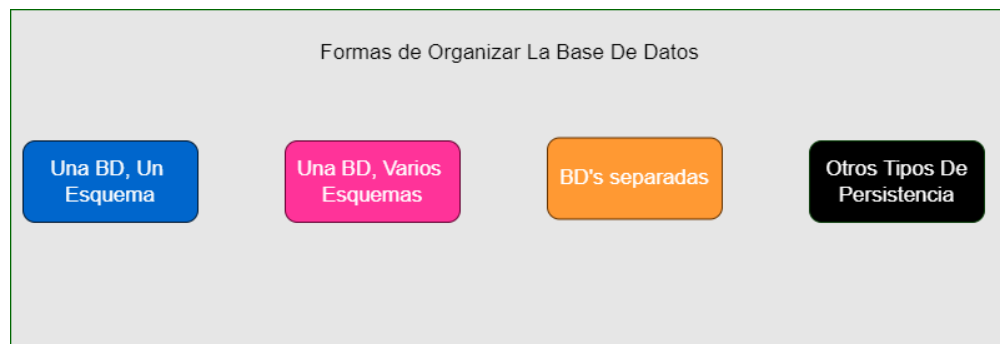


Figura 15. Imagen de diferentes configuraciones para bases de datos. Fuente:(Autores)

8.4. Vistas de Usuario:



Figura 16. Imagen de Inicio al aplicativo. Fuente:(Autores)

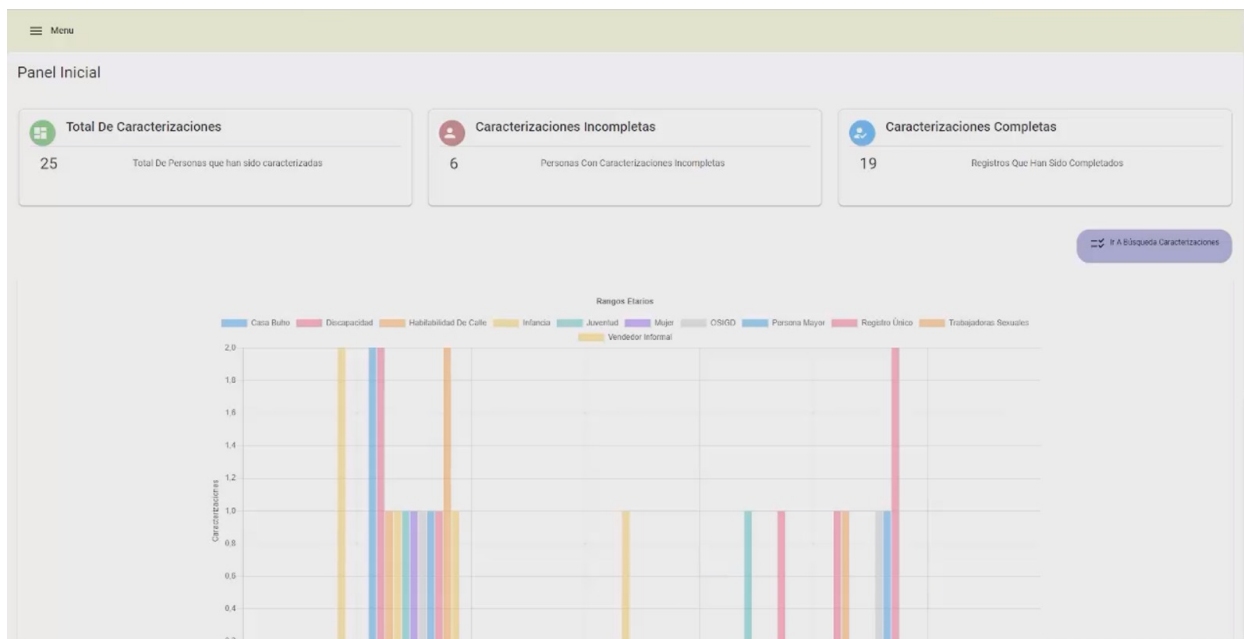


Figura 17. Imagen de la pagina principal del aplicativo. Fuente:(Autores)

Figura 18. Imagen de vista Inicial del formulario en aplicativo. Fuente:(Autores)

Población Caracterizada	Primer Nombre	Segundo Nombre	Primer Apellido	Segundo Apellido	Tipo De Documento	Numero De Documento	Estado	Fecha de Registro	Acción
Vendedor Informal					DNI (Fuera De Colombia)	545852	Incompleto	19/09/2023	[Edit] [Delete]
Vendedor Informal							Completo	19/09/2023	[Edit] [Delete]
Vendedor Informal					DNI (Fuera De Colombia)	25123123	Incompleto	19/09/2023	[Edit] [Delete]
Casa Buho	Gabriel		Alvarez		Cédula De Ciudadanía	147852369	Completo	18/09/2023	[Edit] [Delete]
Discapacidad	gt		as		Cédula De Extranjería	5556	Incompleto	19/09/2023	[Edit] [Delete]
OSIGD	Nanmdo		bam		Cédula De Ciudadanía	444	Completo	21/09/2023	[Edit] [Delete]
Habitabilidad De Calle	Luz		Frank		En Trámite	8768465	Completo	18/09/2023	[Edit] [Delete]
Trabajadoras Sexuales	Giorno		Giovania		Cédula De Extranjería	83314	Completo	19/09/2023	[Edit] [Delete]
Infancia	in		grid		Cédula De Extranjería	21	Completo	21/09/2023	[Edit] [Delete]
Casa Buho	Nicolas		Jaimes		Cédula De Ciudadanía	1234567890	Completo	20/09/2023	[Edit] [Delete]

Figura 19. Imagen de la pagina de búsquedas de caracterizaciones. Fuente:(Autores)

## 9. Desarrollo

En este capítulo se exponen los resultados clave de la implementación de la plataforma de registro y gestión de beneficiarios de programas sociales de la Secretaría de Desarrollo en un período de muestreo de un mes. Como se muestra en la tabla 3, la cantidad de personas que tuvieron una caracterización efectiva de manera manual \* fue de 431, mientras que el número de caracterizaciones efectivas mediante la plataforma fue de 631 (esto teniendo en cuenta que se digitalizaron las 431 anteriores). Es importante señalar que el valor de 431 se obtuvo después de depurar las inconsistencias presentes en los registros previos.

Población	Cantidad de Personas	
	Manual	Plataforma
Casa Búho	56	79
Discapacidad	23	38
Habitabilidad De Calle	37	77
Mujer	18	28
Juventud	45	59
Infancia	12	20
Trabajadoras Sexuales	72	97
O.S.I.G.D.	34	39
Persona Mayor	60	90
Vendedor Informal	74	104
Registro Único	0	0
<b>Totales :</b>	<b>431</b>	<b>631</b>

Tabla 3

*personas que fueron caracterizadas en el transcurso de un mes*

---

\* se refiere a la realizada por medio del formato impreso

## 9.1. Registro Eficiente de Beneficiarios

**9.1.1. Aumento en la velocidad de registro.** La implementación de la plataforma ha resultado en un aumento significativo en la velocidad de registro de beneficiarios.

Poblacion	Tiempo de Registro	
	Manual	Plataforma
Casa Búho	7.50	5.20
Discapacidad	6.00	4.00
Habitabilidad De Calle	5.70	4.60
Mujer	6.80	4.8
Juventud	7.20	3.70
Infancia	6.10	5.10
Trabajadoras Sexuales	5.90	4.30
O.S.I.G.D.	6.40	4.90
Persona Mayor	6.60	4.40
Vendedor Informal	6.30	4.50
Registro Único	0.00	0.00
<b>Promedio:</b>	<b>6.45</b>	<b>4.55</b>
<b>Porcentaje de variación</b>	<b>≈ 29.46 %</b>	

Tabla 4

*Tiempos en promedio de diligencia miento caracterizaciones*

teniendo en cuenta la siguiente ecuación:

$$\text{Porcentaje de Variación} = \left( \frac{V_{inicial} - V_{final}}{V_{inicial}} \right) \times 100 \quad (1)$$

Como se evidencia en la tabla 4, se observa un valor de mejora <sup>1</sup> en los tiempos necesarios

---

<sup>1</sup> disminución en el tiempo requerido para llevar a cabo el proceso de caracterización

para el registro, aproximadamente del 29.46% <sup>2</sup>, resultado que se obtuvo con por medio de la ecuación (1).

**9.1.2. Disminución de errores de registro.** La implementación de la plataforma ha resultado en una disminución significativa en errores de registro de beneficiarios. Se tomaron en cuenta estas características para considerarlos como un error:

- Rayones o manchones de tinta.
- textos ilegibles.
- Datos donde no corresponden.

Como se evidencia en la tabla 5, se observa una disminución en la aparición de errores al realizar los registros, aproximadamente del 60.44%<sup>3</sup>.

## 9.2. Centralización de Datos

**9.2.1. Duplicidad de Registros.** Antes de continuar con los resultados, es necesario hacer las siguientes aclaraciones:

- Para considerar la duplicidad, se debe tener en cuenta que el número de documento y la población sean los mismos.

---

<sup>2</sup> esto se debe al redondeo de cifras

<sup>3</sup> Este resultado se obtuvo al usar la formula 1

Población	Errores de Registro	
	Manual	Plataforma
Casa Búho	18	6
Discapacidad	32	15
Habitabilidad De Calle	23	12
Mujer	13	3
Juventud	16	9
Infancia	8	6
Trabajadoras Sexuales	22	9
O.S.I.G.D.	5	1
Persona Mayor	29	9
Vendedor Informal	16	2
Registro Único	0	0
<b>Promedio:</b>	<b>18</b>	<b>7</b>
<b>Porcentaje de variación</b>	<b>≈ 60.44 %</b>	

Tabla 5

*Errores en promedio de al diligenciar caracterizaciones*

- Puede darse la situación en la que el mismo número de documento esté presente en varias poblaciones.

El proceso de reducción de duplicados se llevó a cabo utilizando un enfoque que se centró en la revisión y estandarización de los datos, permitiendo que la plataforma identificara y eliminara duplicados de manera automática (tratándolos como ediciones). Esto garantizó que los datos fueran más coherentes y confiables, lo que, a su vez, mejoró la calidad de la información disponible.

La adopción de esta metodología no solo optimizó la integridad de la base de datos, sino que también redujo el riesgo de errores humanos en el proceso de identificación de duplicados. En última instancia, esta mejora resultó en una base de datos más eficiente y efectiva para su uso en análisis.

Descripción	Manual	Plataforma
Total de Registros	431	631
Duplicados Detectados	47	0
Criterios de Duplicidad	Revisión manual	Lógica Implementada
Método de Eliminación	Eliminación manual	Edición
Características	Reducción de duplicados manuales	Eliminación de duplicados de manera eficiente
	Mayor riesgo de errores humanos	Base de datos más eficiente
	Mayores tiempos para almacenamiento de datos	Datos más coherentes y precisos

Tabla 6  
*características generales de las dos formas de recolectar datos*

**9.2.2. Generación de informes.** Para plataforma se implemento la función de ge-

neración de informes, esto consiste en los siguientes aspectos:

- Generación de gráficas usando los datos de las caracterizaciones realizadas para llevar un control de los datos registrados.
- Descargas en documento tipo Excel de los informes totalizados para revisión y control.

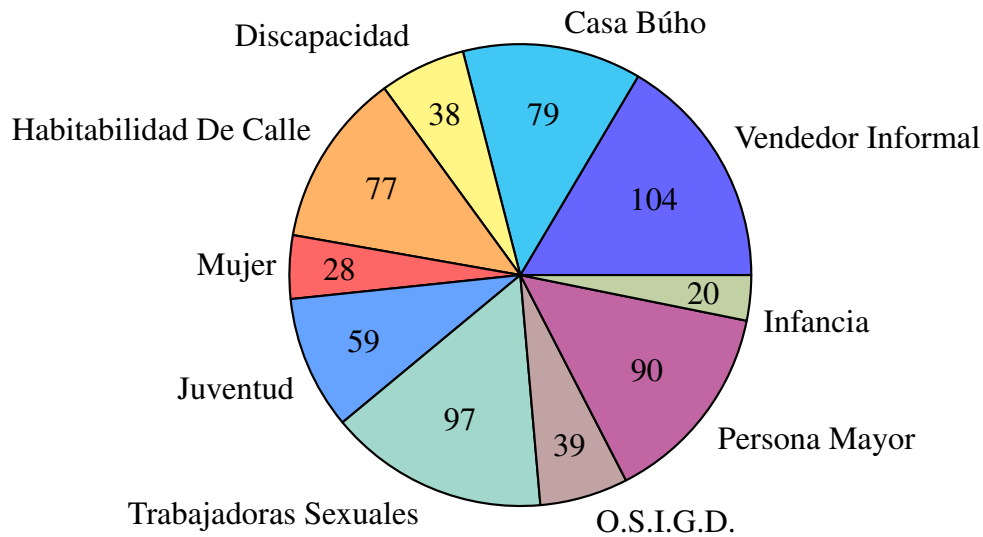


Figura 20. Gráfica de caracterizaciones discriminadas por población

### 9.3. Acceso Seguro y Control de Privacidad

**9.3.1. Medidas de Seguridad.** En la plataforma, se han implementado medidas de seguridad que han resultado en un historial impecable, sin ningún incidente de filtración de datos o acceso no autorizado. Estas medidas están centradas en el control de flujos para la gestión de accesos y autorizaciones, abarcando no solo la capa frontal, sino también la parte posterior del sistema. Además de adherirse a las mejores prácticas de seguridad, el sistema utiliza el estándar JWT (JSON Web Tokens) para validar y garantizar la autenticidad de las solicitudes y la integridad de los datos transmitidos, lo que refuerza aún más la seguridad y la confiabilidad de la plataforma en las transacciones un ejemplo se muestra en la figura 21.

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzbnZmLm9yZy93cy8yMDAxLzA1LzI1IiwiaWF0IjoiYm5hbWU10iJnYwJyaWVsIiwiaHR0cDovL3NjaGVtYXMuZG1sc29hcC5vcmcvd3MvMjAwNS8wNS9pZGVudG10eS9jbGFpbXMvZW1haWxhZGRyZXRzIjoic2hhZG93XzgzT0B0b3RtYW1sLmNvbSI6Imh0dHA6Ly9yZ2h1bWVzLnhtbHNvYXAub3JnL3dzLzIwMDUvMDUvaWR1bnRpdHkvY2xhaW1zL3N1cm5hbWU10iJBBHJhcmV6IiwiaHR0cDovL3NjaGVtYXMuZG1sc29hcC5vcmcvd3MvMjAwNS8wNS9pZGVudG10eS9jbGFpbXMvZW1haWxhZGRyZXRzIjoic2hhZG93XzgzT0B0b3RtYW1sLmNvbSI6Imh0dHA6Ly9yZ2h1bWVzLnhtbHNvYXAub3JnL3dzLzIwMDUvMDUvaWR1bnRpdHkvY2xhaW1zL3N1cm5hbWU10iJodHRwczovL2xvY2FsaG9zdDo0MjAwLnkv3Hsppk1D-uCNrEA1qSsQCsTax3a1JRgTms0X4QA
```

Figura 21. Token generado por el API mostrado desde (20)

**9.3.2. Control de Acceso.** El sistema de control de acceso basado en roles ha sido fundamental para garantizar la privacidad de los datos sensibles. Este enfoque técnico permite

restringir el acceso a la información confidencial únicamente a aquellos individuos que poseen alguno de los siguientes roles

- Root
- Admin
- Usuario
- Beneficiario (aunque este último rol se encuentra en desuso).

De esta manera, se asegura que solo el personal autorizado tenga la capacidad de acceder a dicha información, proporcionando una capa adicional de seguridad y confidencialidad.

## 9.4. Rendimiento

**9.4.1. Pruebas con Postman.** Se realizaron pruebas de rendimiento para garantizar que la plataforma sea ágil y responda eficientemente, dichas pruebas se realizaron Mediante la herramienta “postman” en el caso del back se realizaron pruebas en endpoints tomando los tiempos de ejecución y respuesta como referencia (figura 22), para realizar correcciones las respectivas y mejorar la respuesta por parte del A.P.I. (figura 23).

El criterio para la selección de los endpoints depende de dos numerales:

- su posición crítica de seguridad
- su potencial tráfico alto



### **10. Conclusiones:**

- Se logró diseñar un prototipo operativo para caracterización de beneficiarios, identificando los requisitos de software necesarios.
- Se logró el despliegue en un servidor en la red local de la secretaría, permitiendo a los funcionarios registrar y acceder a la plataforma.
- A través de pruebas, se confirmó una mejora en la velocidad de registros de caracterizaciones y una disminución de errores en las mismas, lo que indica un aumento en la eficiencia de la plataforma.
- La importancia de la validación participativa con los operarios quedó en evidencia, esto permitió realizar ajustes y mejoras continuas en la plataforma, asegurando que se ajuste a las necesidades y expectativas de los operadores.

## 11. Mejoras

### 11.1. Recomendaciones

**11.1.1. Seguridad de Datos.** Mejorar y mantener una sólida seguridad de datos es fundamental para proteger la información personal y económica de los beneficiarios. Esto implica la implementación de medidas como la encriptación de datos en reposo y en tránsito, así como la gestión adecuada de contraseñas y permisos de acceso.

**11.1.2. Escalabilidad.** Asegurar que la plataforma sea escalable para manejar un aumento en el número de beneficiarios y programas sociales sin degradación del rendimiento. Esto puede incluir la optimización del código y la infraestructura subyacente.

**11.1.3. Arquitectura de Micro-servicios.** Evaluar la posibilidad de migrar hacia una arquitectura de micro-servicios, dado el crecimiento previsto. Los micro-servicios pueden facilitar la gestión de componentes individuales y permitir actualizaciones sin afectar todo el sistema.

**11.1.4. Monitorización y Registro.** Implementar un sistema de monitorización constante para detectar problemas en tiempo real y llevar un registro de eventos y errores para facilitar la depuración y el análisis de problemas.

**11.1.5. Integración de Sistemas Externos.** Trabajar en la integración fluida con sistemas externos, como bases de datos gubernamentales o de otras organizaciones relacionadas con programas sociales, para obtener y compartir datos de manera eficiente.

## **11.2. Mejoras a Futuro:**

**11.2.1. Machine Learning y Analítica Avanzada.** Considerar la implementación de técnicas de aprendizaje automático para mejorar la identificación y asignación de beneficios, así como la detección de posibles fraudes o inexactitudes en los datos.

**11.2.2. Automatización Inteligente.** Explorar la automatización inteligente mediante el uso de chatbots o asistentes virtuales para brindar asistencia a los beneficiarios y responder a preguntas frecuentes.

**11.2.3. Geo-localización.** Integrar la geo-localización para optimizar la asignación de recursos en función de la ubicación y las necesidades específicas de las comunidades.

**11.2.4. Participación Ciudadana.** Fomentar la participación de los ciudadanos a través de la retroalimentación y la implementación de funcionalidades que permitan a los beneficiarios expresar sus necesidades y sugerencias.

**11.2.5. Mejora de la Experiencia del Usuario.** Realizar pruebas periódicas de usabilidad y recopilar comentarios de los usuarios para realizar mejoras en la experiencia del usuario y la interfaz de usuario.

### Referencias Bibliográficas

- [1] Alarcón, J. (2022). ¿qué necesitas saber en 2020 para ser un buen desarrollador web front-end?  
Fecha de creación: 19 de septiembre 2022 7:15:00 a.m., Fecha de actualización: 12 de abril de 2023.
- [Alcaldía] Alcaldía, N. Nuestra alcaldía.
- [3] Alvarez, M. A. (2018). Parámetros en las rutas angular. 05 de junio de 2018.
- [4] Bass, L., Clements, P., and Kazman, R. (2003). *Software Architecture in Practice*, volume 3. Addison-Wesley Professional.
- [5] Bautista, I. (2021). Backend y frontend, ¿qué es y cómo funcionan en la programación?
- [6] Betts, T. (2022). Software architecture and design infoq trends report—april 2022. Fecha de creación: 4 de abril de 2022.
- [7] Cappola, M. (Fecha de publicación no disponible). Frontend y backend: qué son, en qué se diferencian y ejemplos.
- [8] Casals, A. (2022). Infoq .net trends report 2022. Fecha de acceso: 26 de septiembre de 2022.
- [9] Castro, E. (2023). Patrones en la arquitectura de software.
- [10] Coppola, M. (2022). Los 9 mejores frameworks para desarrollo web en 2022. Fecha de creación: 20 de junio de 2022 7:00:00 a.m., Fecha de actualización: 20 de enero de 2023.

[de Desarrollo Social] de Desarrollo Social, S. Secretaría de desarrollo social.

[12] Fernández, L. F. (2006). Arquitectura de software. *Software Guru*, 2(3):40–45.

[13] Gabriela, M. (2020). Guía completa del framework: qué es, cuáles tipos existen y por qué es importante en internet.

[14] González, Y. D. and Romero, Y. F. (2012). Patrón modelo-vista-controlador. *Telemática*, 11(1):47–57.

[15] Gómez, J. L. B. (2015). *Desarrollo de aplicaciones web en el entorno servidor*. Ediciones Paraninfo, SA.

[16] Hernandez, U. (2015). Mvc (model, view, controller) explicado.

[17] Huaman, W. (2018). Los 10 patrones comunes de arquitectura de software.

[18] Ibarra, S. G. P., Quispe, J. R., Mullicundo, F. F., and Lamas, D. A. (2021). Herramientas y tecnologías para el desarrollo web desde el frontend al backend. In *XXIII Workshop de Investigadores en Ciencias de la Computación (WICC 2021, Chilecito, La Rioja)*.

[19] Javier, J. (2013). Ciclos de vida para gestionar (no sólo software): cascada, espiral, iterativo, incremental o ágil.

[20] JWT.io (Sin fecha). Jwt.io - json web tokens.

[21] Laboratorio Nacional de Calidad del Software de INTECO, España (2009). *Ingeniería del*

*Software: Metodologías y Ciclos de Vida.* Ministerio de Industria, Turismo y Comercio de España. Primera versión editada en marzo de 2009.

[22] Maluenda, R. (2020). Tipos de desarrollo de aplicaciones web: ejemplos y características.

[23] NetMentor (2023). ¿qué es un monolito modular? comparación de arquitecturas.

[24] Novoseltseva, E. (2020). Los 5 principales patrones de arquitectura de software.

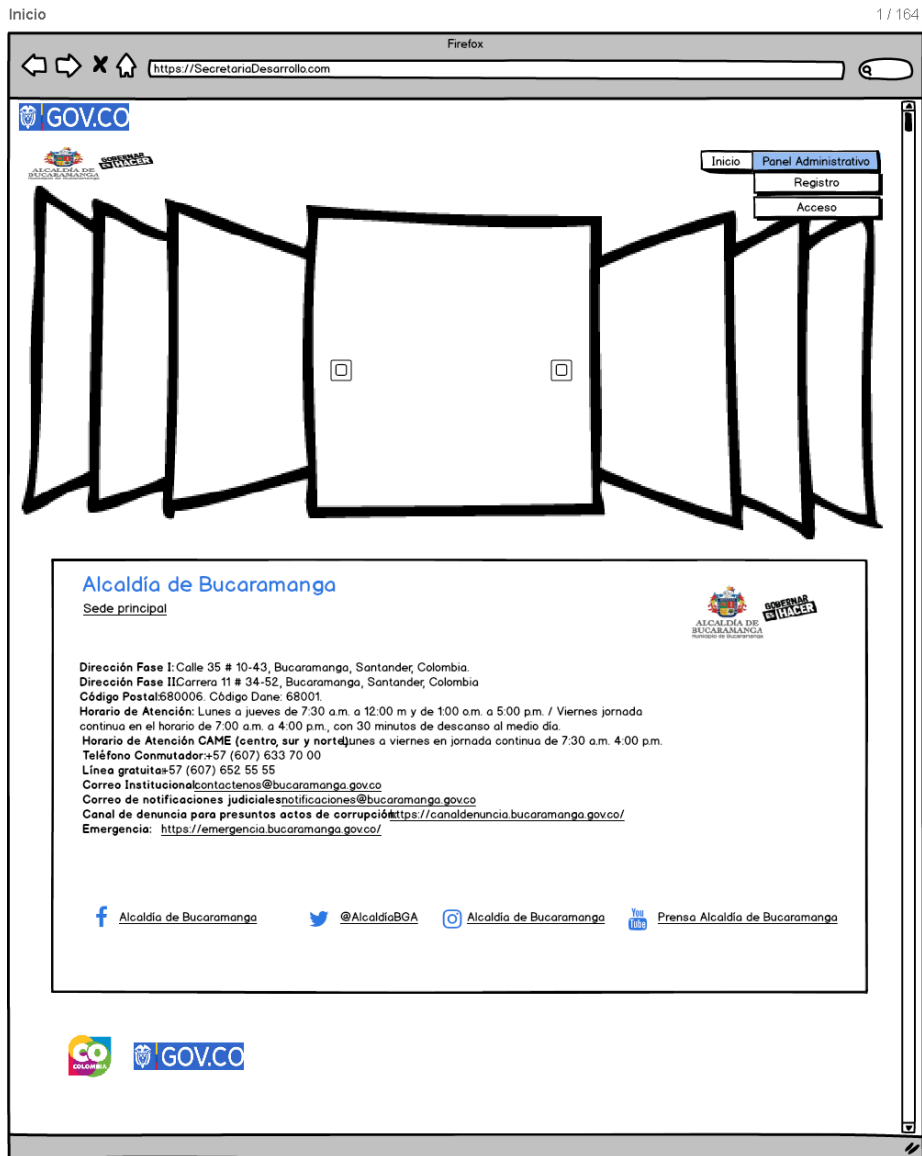
[25] S., C. A. L. (2009). Cómo mantener el patrón modelo vista controlador en una aplicación orientada a la web. *I*, 4(7):72–78.

[26] Santillán, L. A. C., Ginestá, M. G., and Mora, P. (2014). *Bases de datos en MySQL*. Universitat oberta de Catalunya.

[27] Schmidt, R. (2023). Desarrollo de aplicaciones web en 2023.

[28] Tia (2015). Desarrollo iterativo y creciente. Fecha de creación: 16 de julio de 2015, el nombre del autor es el usuario de la plataforma.

Apéndice A. mockup de vistas inicial

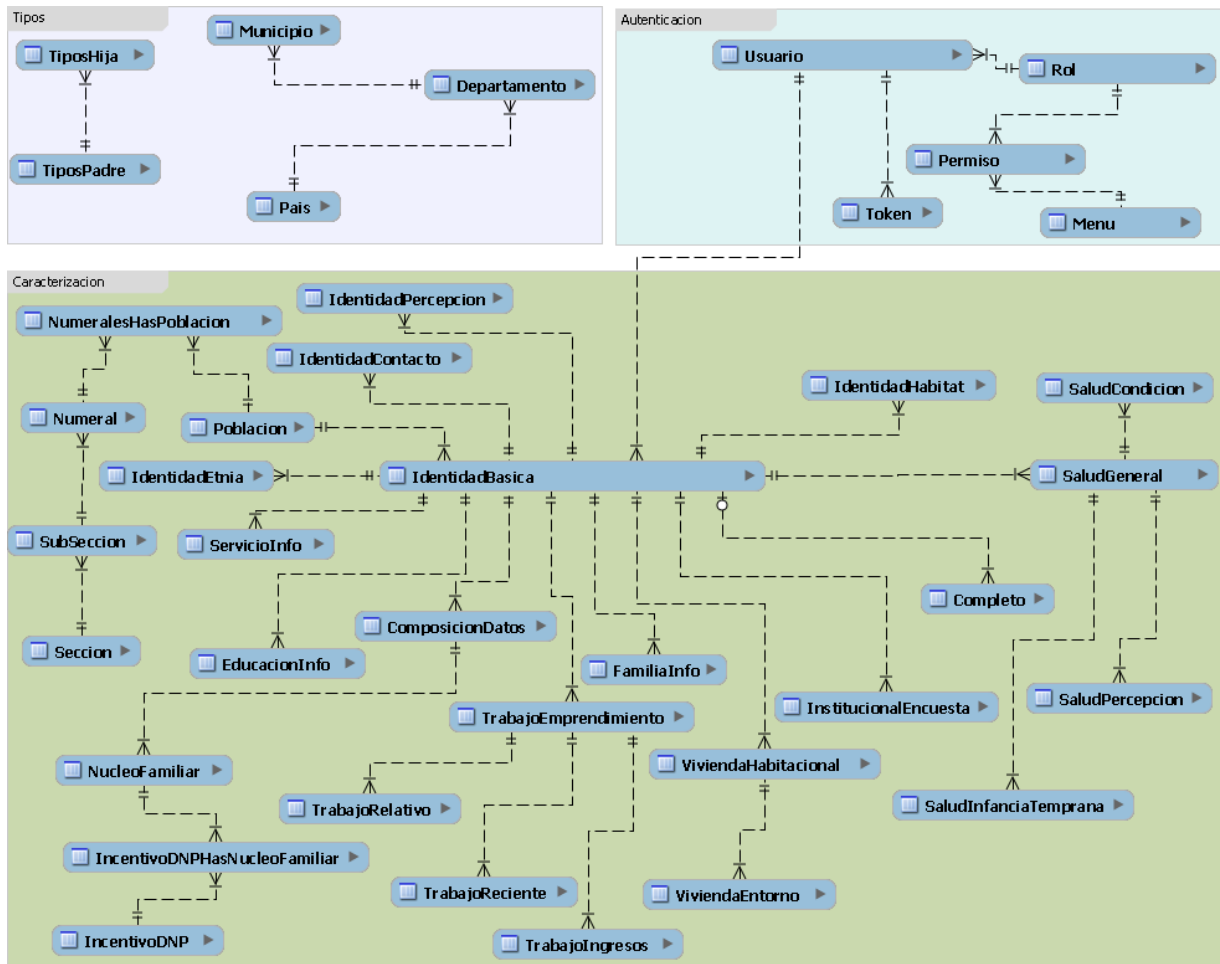


**Apéndice B. Tabla de presupuesto**

<b>PRESUPUESTO</b>						
Nombres y Apellidos	Formación Básica	Función	Horas por semana	Valor hora Pesos (COP)	Dedicación (Semanas)	Total Pesos (COP)
Gabriel de Jesús Alvarez Tejada	Estudiante de Ingeniería de sistemas	Desarrollo de software	30	7,000	720	5,040,000
Fabio Andrey Ojeda Villamizar	Estudiante de Ingeniería de sistemas	Desarrollo de software	30	7,000	720	5,040,000
Jathinson Meneses Mendoza	Ingeniero	Director	3	152,500	72	10,980,000
Henry Andrés Jiménez Herrera	Ingeniero	Codirector	3	152,500	72	10,980,000
<b>Subtotal</b>						<b>32,040,000</b>



**Apéndice D. Diagrama de Entidad-Relación para Base de Datos**



**Apéndice E. Contenido de token Generado Como Respuesta de API validado desde (20)**

## Decoded EDIT THE PAYLOAD AND SECRET

**HEADER: ALGORITHM & TOKEN TYPE**

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

**PAYLOAD: DATA**

```
{
  "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname": "gabriel",
  "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress": "shadow_857@hotmail.com",
  "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname": "Alvarez",
  "http://schemas.microsoft.com/ws/2008/06/identity/claims/version": "1",
  "http://schemas.microsoft.com/ws/2008/06/identity/claims/role": "Root",
  "exp": 1696573235,
  "iss": "https://localhost:4200",
  "aud": "https://localhost:4200"
}
```