

# Manual Técnico



## 1. Introducción

### Propósito del Manual

El propósito de este manual técnico es proporcionar una guía completa para la implementación, configuración, mantenimiento y soporte del Sistema de Certificados de Residencia y Vecindamiento de la Alcaldía de Acacías. Este documento está destinado a los desarrolladores, administradores y personal técnico encargado de la gestión y evolución del sistema.

### Alcance del Sistema

Este sistema permite a los ciudadanos de Acacías realizar trámites de solicitud de certificados de residencia y vecindamiento a través de una plataforma digital. El sistema incluye funcionalidades que facilitan la gestión de solicitudes, la validación por parte de los validadores y la emisión de certificados en formato PDF con características de seguridad como códigos QR.

Además, el sistema ha sido diseñado para garantizar la interoperabilidad con otros sistemas internos de la Alcaldía, asegurando una gestión eficiente y transparente de los trámites. El sistema de certificados integra la verificación de direcciones, la asignación de roles específicos para validadores, y la creación de un historial de solicitudes accesible para todos los actores involucrados.

### Descripción General del Sistema

La plataforma de Certificados de Residencia y Vecindamiento está construida sobre un entorno tecnológico robusto, utilizando el framework **Laravel** para el back-end y **Livewire** para la creación de interfaces interactivas sin recargar la página.

El sistema está compuesto por varios módulos interconectados que permiten a los usuarios realizar solicitudes, hacer seguimientos de sus trámites, y descargar certificados válidos. Por su parte, los validadores tienen acceso a funcionalidades específicas que les permiten aprobar, rechazar o devolver solicitudes, así como registrar visitas domiciliarias en el caso de los certificados de vecindamiento.

Entre las principales funcionalidades del sistema se incluyen:

- **Formulario de Solicitud:** Los usuarios pueden iniciar una solicitud para un certificado de residencia o vecindamiento.

- **Flujo de Validación:** El sistema gestiona dos etapas de validación: Validador 1 y Validador 2.
- **Generación de Certificados:** Los certificados se emiten con códigos QR para verificar su autenticidad.
- **Interoperabilidad:** El sistema se integra con las bases de datos internas de la Alcaldía para asegurar la consistencia y la validez de los datos.

Este manual está diseñado para describir los aspectos técnicos de la plataforma, proporcionando los detalles necesarios para su implementación y mantenimiento.



## 2. Arquitectura del Sistema

### Descripción General de la Arquitectura

La arquitectura del Sistema de Certificados de Residencia y Vecindamiento está diseñada para ser escalable, modular y segura, garantizando un rendimiento óptimo y una fácil

administración. El sistema sigue un enfoque basado en el patrón de **arquitectura en capas**, donde cada capa tiene responsabilidades específicas que permiten la gestión eficiente de los datos y la interacción del usuario.

La arquitectura se divide en los siguientes componentes principales:

### 1. Front-End (Interfaz de Usuario):

- Desarrollado utilizando **Livewire** y **Tailwind CSS**, lo que permite interfaces interactivas y modernas sin la necesidad de recargar la página.
- Las vistas son adaptables, lo que garantiza una experiencia de usuario fluida tanto en dispositivos móviles como en computadoras de escritorio.
- Los usuarios interactúan con formularios dinámicos, mapas interactivos y la visualización de estados de las solicitudes en tiempo real.

### 2. Back-End (Lógica de Negocio):

- Utiliza el framework **Laravel**, conocido por su robustez y capacidad para manejar tareas complejas, como la validación de datos y la gestión de usuarios.
- La lógica de negocio está estructurada en controladores y servicios que gestionan la interacción entre los formularios, la base de datos y los validadores.
- **Laravel Jetstream** proporciona una solución para la gestión de usuarios, roles y autenticación de manera segura.

### 3. Base de Datos:

- El sistema utiliza una base de datos **MySQL** (o **PostgreSQL** dependiendo de la configuración) para almacenar los registros de las solicitudes, usuarios, validadores, y certificados.
- La base de datos está organizada en varias tablas interrelacionadas, incluyendo:

- **Usuarios:** Información personal y de autenticación de los ciudadanos.
- **Solicitudes:** Detalles sobre cada solicitud de certificado, incluyendo su estado y fecha de creación.
- **Validaciones:** Registros de las acciones realizadas por los validadores.
- **Certificados:** Información sobre los certificados emitidos, incluyendo el estado y el código QR.

#### 4. Integraciones y Comunicación entre Sistemas:

- El sistema está diseñado para integrarse con otros sistemas de la Alcaldía, asegurando que la información de los ciudadanos se sincronice correctamente.
- Se utilizan **APIs RESTful** para la comunicación entre el sistema de certificados y otras plataformas internas, como el registro de identificación nacional y la base de datos centralizada de la Alcaldía.
- El sistema también permite la verificación en tiempo real de los estados de las solicitudes y la validación de documentos a través de APIs externas.

### Componentes Principales de la Arquitectura

#### 1. Módulo de Solicitudes:

- El corazón del sistema, donde los usuarios crean nuevas solicitudes de certificados. El módulo maneja la entrada de datos, las validaciones y el almacenamiento de las solicitudes.
- **Proceso:** El usuario inicia una solicitud, el sistema valida los datos y la envía a la cola de validación para su revisión por parte de los validadores.

#### 2. Módulo de Validadores:

- Consiste en dos roles de validadores, **Validador 1** y **Validador 2**, encargados de revisar y aprobar las solicitudes. Este módulo está integrado con las funciones de validación en tiempo real, lo que facilita la tarea de revisión y optimización del flujo de trabajo.
- **Proceso:** Los validadores reciben las solicitudes para su revisión, pueden agregar notas y emitir certificados, o devolverlas para su corrección.

### 3. Generación de Certificados:

- Una vez aprobada la solicitud, el sistema genera un certificado en formato PDF que incluye un código QR único para verificar la autenticidad del documento.
- El sistema permite que los usuarios descarguen el certificado directamente desde la plataforma.

### 4. Módulo de Consulta:

- Los usuarios pueden consultar el estado de sus solicitudes a través de una interfaz simple y directa. Dependiendo del estado de la solicitud (pendiente, aprobado, rechazado, emitido), se muestran diferentes opciones y mensajes para guiar al usuario en su proceso.

## Diagrama de Arquitectura del Sistema

A continuación, se presenta un diagrama visual que muestra los componentes del sistema y su interacción:

[ Front-End (Livewire + Tailwind CSS) ]

|

[ Back-End (Laravel + Jetstream) ]

|

[ Base de Datos (MySQL/PostgreSQL) ]

|

[ APIs Externas (Validación, Sistema Central) ]

Este esquema visual representa la comunicación entre los distintos componentes del sistema, donde la interfaz de usuario interactúa con el backend, que a su vez se comunica con la base de datos y otras APIs externas para validar y gestionar los datos.

### 3. Requisitos Técnicos

Los requisitos técnicos describen los elementos necesarios para implementar, configurar y mantener el Sistema de Certificados de Residencia y Vecindamiento de la Alcaldía de Acacías. A continuación, se detallan los aspectos clave que aseguran el correcto funcionamiento de la plataforma, desde la infraestructura hasta las configuraciones específicas de software.

#### 3.1 Hardware y Software Requerido

Para que el sistema funcione de manera eficiente y segura, se requiere una infraestructura adecuada, tanto en términos de hardware como de software.

- **Servidor Web:**
  - **Requisitos Mínimos:**
    - Procesador: Intel Core i5 o equivalente.
    - RAM: 8 GB.
    - Almacenamiento: 50 GB de espacio libre en disco.
    - Sistema Operativo: Linux (preferiblemente Ubuntu 20.04 LTS o

superior).

- Conexión a Internet estable con al menos 10 Mbps de velocidad de subida y bajada.

- **Recomendado:**

- Procesador: Intel Xeon o equivalente.
- RAM: 16 GB o más.
- Almacenamiento SSD: 100 GB o más.
- Sistema Operativo: Ubuntu 20.04 LTS o superior.
- Conexión a Internet con al menos 100 Mbps de velocidad.

- **Base de Datos:**

- **MySQL o PostgreSQL** como base de datos relacional.
- **Requisitos:**
  - Espacio de almacenamiento recomendado: 100 GB para bases de datos iniciales, con posibilidad de expansión.
  - Debe estar configurada para aceptar conexiones remotas y ser respaldada regularmente.

### 3.2 Configuración del Servidor

El servidor debe estar configurado adecuadamente para garantizar el rendimiento y la seguridad del sistema. A continuación se describen las configuraciones esenciales:

- **Instalación de Dependencias:**

- **PHP:** Versión 8.2 o superior.



- **Composer:** Para la gestión de dependencias de Laravel.
- **Nginx o Apache** como servidor web.
- **Certificados SSL** para garantizar conexiones seguras.
- **Configuración de PHP:**
  - `upload_max_filesize = 25M`
  - `post_max_size = 25M`
  - `memory_limit = 256M`
  - `max_execution_time = 300`
- **Configuración de Nginx/Apache:**
  - En Nginx:
    - `client_max_body_size 25M;`
    - `proxy_read_timeout 300;`
    - `proxy_connect_timeout 300;`
    - `proxy_send_timeout 300;`
- **Instalación de Certificados SSL:**
  - Los certificados SSL deben ser instalados en el servidor para garantizar que todas las conexiones entre los usuarios y el sistema sean seguras. Puede utilizarse Let's Encrypt para generar certificados gratuitos y automáticos.

### 3.3 Base de Datos y Estructura de Tablas

El sistema utiliza una base de datos relacional, como **MySQL** o **PostgreSQL**, para almacenar

los datos de los usuarios, las solicitudes, los validadores y los certificados. La estructura de la base de datos debe ser eficiente y normalizada para garantizar un buen rendimiento en las consultas.

- **Tablas Principales:**

- **Usuarios:** Contendrá información sobre los ciudadanos que realizan las solicitudes (nombre, identificación, dirección, etc.).
- **Solicitudes:** Detalles sobre las solicitudes de certificados (estado, tipo de certificado, fecha de creación, etc.).
- **Validaciones:** Información sobre el proceso de validación de cada solicitud (observaciones de los validadores, estado de validación, etc.).
- **Certificados:** Datos sobre los certificados emitidos (número de trámite, fecha de emisión, código QR, etc.).

- **Relaciones:**

- Las tablas estarán interrelacionadas mediante claves foráneas, asegurando la integridad de los datos.
- Ejemplo: La tabla de **Solicitudes** estará vinculada a la tabla de **Usuarios** mediante el `user_id`.

### 3.4 Seguridad y Cifrado

El sistema debe cumplir con los estándares de seguridad más recientes para proteger los datos de los usuarios y garantizar que la información sensible esté cifrada. Las siguientes medidas de seguridad son esenciales:

- **Cifrado de Datos:**

- **Contraseñas:** Las contraseñas deben ser almacenadas usando el algoritmo **bcrypt** o **Argon2**.
- **Datos de Solicitudes:** Todos los datos sensibles deben ser cifrados

utilizando métodos de cifrado robustos (AES-256).

- **Comunicación Segura:** Todas las comunicaciones deben utilizar el protocolo **HTTPS** mediante **TLS/SSL**.
- **Autenticación y Autorización:**
  - El sistema debe usar **Laravel Sanctum** para manejar la autenticación de usuarios mediante tokens.
  - **Control de Roles y Permisos:** Laravel Jetstream debe ser configurado para gestionar los roles de usuario y las autorizaciones para acceder a diferentes módulos (usuarios, validadores, administradores).
- **Protección contra Vulnerabilidades Comunes:**
  - **XSS (Cross-Site Scripting):** Se debe sanitizar la entrada de datos para evitar inyecciones de código malicioso.
  - **SQL Injection:** El uso de ORM (Object-Relational Mapping) como **Eloquent** previene la inyección de SQL.
  - **CSRF (Cross-Site Request Forgery):** El sistema debe generar y validar tokens CSRF para proteger las solicitudes POST.

### 3.5 Respaldo y Recuperación de Datos

Es esencial contar con un sistema de respaldo y recuperación de datos para prevenir la pérdida de información en caso de fallos del sistema.

- **Respaldo de Base de Datos:**
  - Se deben realizar copias de seguridad regulares de la base de datos (al menos una vez al día).
  - Las copias de seguridad deben almacenarse en un servidor seguro y deben ser accesibles solo para el personal autorizado.

- **Plan de Recuperación ante Desastres:**

- En caso de un fallo grave, el sistema debe permitir la restauración de los datos a partir de las copias de seguridad.
- El plan debe incluir un proceso claro para restaurar la base de datos y la plataforma en su totalidad.

#### **4. Diseño e Implementación de Funcionalidades**

En esta sección, se detallan las principales funcionalidades implementadas en el Sistema de Certificados de Residencia y Avicindamiento, con el objetivo de ofrecer una experiencia de usuario eficiente, segura y fácil de usar. Estas funcionalidades abarcan desde la creación de solicitudes hasta la emisión de certificados, pasando por los procesos de validación y la consulta en tiempo real del estado de las solicitudes.

##### **4.1 Diseño de Interfaces de Usuario**

Las interfaces de usuario han sido diseñadas para ser intuitivas y fáciles de usar. El objetivo es minimizar la complejidad para el usuario final, permitiéndole realizar trámites sin complicaciones. Las interfaces incluyen los siguientes elementos clave:

- **Pantalla de Bienvenida:**

- Una pantalla inicial que explica las funcionalidades principales del sistema. Este diseño asegura que los usuarios puedan familiarizarse rápidamente con la plataforma antes de comenzar el proceso de solicitud.

- **Imagen Relacionada:**

- Pantalla de bienvenida con instrucciones claras para el usuario.

- **Formulario de Solicitud:**

- Un formulario dinámico que permite a los usuarios ingresar los datos necesarios para solicitar un certificado. Se incluyen campos obligatorios y la posibilidad de adjuntar documentos adicionales.
- El formulario incorpora validaciones automáticas que guían al usuario en el proceso, minimizando errores y facilitando la interacción.
- **Imagen Relacionada:**
  - Formulario para la creación de solicitudes, con validación en tiempo real.
- **Formulario de Verificación:**
  - Al registrarse, los usuarios reciben un correo de verificación. El sistema permite al usuario confirmar su correo, asegurando que los datos sean correctos antes de continuar con la solicitud.
  - **Imagen Relacionada:**
    - Pantalla de verificación de correo.

#### 4.2 Funcionalidad de Validadores

El sistema está diseñado con dos roles de validación, **Validador 1** y **Validador 2**, para garantizar un proceso de verificación preciso y eficiente. Ambos roles cuentan con interfaces especializadas que permiten revisar y gestionar las solicitudes de manera rápida.

- **Validador 1:**
  - **Interfaz de Revisión:**
    - Los validadores tienen acceso a una tabla que lista todas las solicitudes pendientes. Pueden revisar los datos de cada solicitud y decidir si aprobarla, rechazarla o devolverla con observaciones para corrección.

- **Imagen Relacionada:**
  - Vista de las solicitudes pendientes para el Validador 1.
- **Validación de Información:**
  - Se incluyen herramientas para verificar la información proporcionada por los usuarios y los documentos adjuntos. Los validadores pueden agregar notas de validación visibles para el siguiente validador.
- **Imagen Relacionada:**
  - Proceso de validación con notas y observaciones.
- **Validador 2:**
  - **Revisión Final:**
    - El Validador 2 tiene la capacidad de revisar las solicitudes aprobadas por el Validador 1. Si todo está en orden, puede emitir el certificado. Si se detectan problemas, puede devolver la solicitud para que se corrija.
  - **Imagen Relacionada:**
    - Pantalla de emisión de certificados.
  - **Validaciones Avanzadas:**
    - Además de la revisión final, el Validador 2 realiza validaciones adicionales, como la verificación de archivos adjuntos y la validación de que los datos cumplen con los requisitos establecidos por la Alcaldía.
  - **Imagen Relacionada:**

- Confirmación para emisión del certificado.

#### 4.3 Integración de Tablas y Vistas por Roles

Cada rol dentro del sistema tiene una vista personalizada que muestra únicamente la información relevante para su tarea. Esto asegura que los usuarios y validadores trabajen con datos claros y organizados:

- **Usuario:**

- El usuario puede consultar el estado de sus solicitudes, ver su historial y crear nuevas solicitudes si la anterior ha sido rechazada o completada.
- **Imagen Relacionada:**

- Tabla de historial de solicitudes del usuario.

- **Validador 1 y 2:**

- Las tablas de los validadores permiten filtrar solicitudes por su estado (pendiente, en revisión, aprobada). Los validadores pueden realizar acciones rápidas como aprobar, rechazar o devolver solicitudes con observaciones.

#### 4.4 Generación de Certificados

Una vez que la solicitud ha sido aprobada por el Validador 2, el sistema genera un certificado en formato PDF que incluye los siguientes elementos:

- **Información del Certificado:**

- El certificado incluye el nombre del solicitante, número de trámite, dirección y otros datos relevantes. Además, se incluye un código QR único para verificar la autenticidad del documento.

- **Imagen Relacionada:**

- Certificado de residencia con código QR y firma digital del validador.

- **Descarga del Certificado:**

- Los usuarios pueden descargar su certificado directamente desde la interfaz, si el estado del trámite es "Emitido".
- **Imagen Relacionada:**
  - Proceso de descarga del certificado en formato PDF.

#### 4.5 Módulo de Consulta

Los usuarios pueden consultar el estado de sus solicitudes en tiempo real a través de la plataforma. Este módulo presenta los siguientes aspectos:

- **Métodos de Consulta:**

- Los usuarios pueden consultar su solicitud mediante el número de documento o el número de solicitud. Esto facilita el acceso al estado del trámite.
- **Imagen Relacionada:**
  - Interfaz de consulta inicial con opciones de búsqueda.

- **Visualización de Resultados:**

- Después de realizar la consulta, el sistema muestra el estado actual del trámite y, si está "Emitido", el usuario puede proceder a la descarga del certificado.
- **Imagen Relacionada:**
  - Resultado de la consulta exitosa con estado "Emitido".

#### Impacto y Beneficios

- **Mejora en la Experiencia del Usuario:**

La interfaz sencilla y las validaciones en tiempo real aseguran que los usuarios puedan completar el proceso de solicitud y consulta sin dificultades.



- **Eficiencia para los Validadores:**

La separación de roles entre Validador 1 y Validador 2 permite una revisión más rápida y precisa, con menos posibilidad de error.

- **Optimización del Proceso:**

La integración de todas las etapas del proceso en un sistema unificado asegura que los datos se gestionen de manera coherente y eficiente.

## 5. Integraciones entre Sistemas

En el contexto del Sistema de Certificados de Residencia y Vecindamiento de la Alcaldía de Acacías, las integraciones entre plataformas externas no son automáticas ni en tiempo real. Actualmente, los validadores son los responsables de verificar manualmente la información proporcionada por los usuarios a través de otras plataformas externas. El sistema no realiza validaciones automáticas de datos, sino que proporciona una interfaz que facilita el trabajo de los validadores para realizar estas verificaciones de manera eficiente.

### 5.1 Validación de Información por los Validadores

El proceso de validación se lleva a cabo de manera manual por parte de los validadores, quienes deben acceder a otras plataformas externas para verificar la información proporcionada por los ciudadanos. Esta validación se refiere principalmente a los siguientes datos:

- **Verificación de la Identidad del Ciudadano:**

- Los validadores consultan bases de datos externas para confirmar la identidad del solicitante, como los registros de identificación nacional, para garantizar que la información personal proporcionada coincida con los datos oficiales.

- **Verificación de la Dirección de Residencia:**

- Para el certificado de residencia o vecindamiento, los validadores deben consultar plataformas externas o realizar visitas domiciliarias para verificar la dirección proporcionada por el ciudadano. Este paso es crucial para la emisión de los certificados.

- **Verificación de la Autenticidad de los Documentos:**

- En caso de que el ciudadano haya subido documentos para su solicitud, como el certificado de residencia o identificaciones adicionales, el validador es responsable de comprobar la autenticidad de estos documentos mediante plataformas externas de validación.

## 5.2 Flujo de Trabajo de Validación

El sistema no automatiza la validación de estos datos, sino que proporciona herramientas para que los validadores gestionen de manera más eficiente las solicitudes. A continuación, se describe el flujo de trabajo de validación:

- **Etapas 1: Creación de la Solicitud por el Ciudadano**

- El ciudadano completa el formulario con la información requerida, incluyendo su identificación y dirección.
- Esta solicitud se almacena en el sistema y se envía a los validadores para su revisión.

- **Etapas 2: Verificación por los Validadores**

- El **Validador 1** recibe la solicitud y accede a plataformas externas para validar la identidad y dirección del solicitante. Además, si es necesario, realiza una visita domiciliaria para corroborar los datos.
- El validador puede registrar sus observaciones y, si la solicitud es válida, la envía al **Validador 2** para una revisión final.
- El **Validador 2** revisa la solicitud y la aprobación del Validador 1, asegurándose de que todos los requisitos estén cumplidos antes de emitir el certificado.

- **Etapas 3: Emisión del Certificado**

- Una vez que el **Validador 2** ha aprobado la solicitud, el sistema genera un certificado en formato PDF, el cual está disponible para ser descargado por

el ciudadano.

### 5.3 Herramientas de Soporte para los Validadores

Aunque el proceso de validación es manual, el sistema proporciona diversas herramientas para facilitar la tarea de los validadores:

- **Tablas de Solicitudes:**
  - Los validadores acceden a una tabla dinámica que lista todas las solicitudes pendientes. Estas solicitudes se pueden filtrar por estado, lo que permite a los validadores priorizar las solicitudes que necesitan ser verificadas.
  - **Imagen Relacionada:**
    - Tabla de solicitudes filtrada por estado para facilitar la gestión de los validadores.
- **Formulario de Validación:**
  - Los validadores tienen acceso a un formulario donde pueden registrar sus observaciones sobre la validación de la solicitud. Este formulario también incluye un espacio para agregar notas relacionadas con la verificación de la identidad, la dirección y los documentos adjuntos.
  - **Imagen Relacionada:**
    - Formulario de validación con campos para agregar notas y observaciones.

### 5.4 Registro de Verificación Manual

El sistema permite que los validadores registren de manera detallada el proceso de verificación manual, lo que asegura la trazabilidad y transparencia de cada trámite:

- **Registros de Verificación:**
  - Los validadores pueden registrar la información sobre cada verificación

realizada, incluyendo las plataformas o servicios consultados, los resultados obtenidos y cualquier observación relevante para cada solicitud.

- Esto garantiza que cada solicitud tenga un historial claro y accesible para futuros seguimientos o auditorías.

- **Imagen Relacionada:**

- Historial de registros de verificación de solicitudes.

## 5.5 Impacto y Beneficios

- **Mejora en la Eficiencia del Proceso:**

- Aunque la validación no es automática, las herramientas del sistema ayudan a los validadores a gestionar las solicitudes de manera más eficiente y a reducir los errores durante la verificación.

- **Mayor Transparencia:**

- El registro detallado de las verificaciones manuales garantiza que haya un historial claro de cada solicitud, lo que facilita el seguimiento y las auditorías.

- **Flexibilidad en la Validación:**

- Dado que el proceso depende de la validación manual de los datos, los validadores tienen la flexibilidad para elegir las fuentes externas que consideren adecuadas para verificar la información de manera confiable.

## 6. Soporte y Mantenimiento Evolutivo

El soporte y mantenimiento del Sistema de Certificados de Residencia y Avecindamiento son fundamentales para asegurar que la plataforma siga funcionando de manera eficiente, segura y actualizada con el paso del tiempo. Esta sección detalla las actividades relacionadas con el mantenimiento preventivo, la corrección de errores, las mejoras continuas y las actualizaciones periódicas del sistema.

## 6.1 Mantenimiento Preventivo

El mantenimiento preventivo asegura que el sistema siga funcionando sin interrupciones y que los posibles problemas se detecten y solucionen antes de que afecten el rendimiento o la experiencia del usuario.

- **Actividades Realizadas:**

- **Revisión de la Infraestructura:** Se realiza una auditoría periódica de los recursos del servidor (almacenamiento, RAM, CPU, etc.) para garantizar que el sistema tenga suficiente capacidad para manejar el tráfico y las solicitudes.
- **Monitoreo de la Base de Datos:** Se lleva a cabo un análisis regular de las bases de datos para optimizar las consultas, índices y almacenamiento, lo que reduce el riesgo de cuellos de botella o problemas de rendimiento.
- **Actualización de Dependencias:** Se mantienen actualizadas las bibliotecas y dependencias del proyecto, como **Laravel** y **Livewire**, para aprovechar nuevas características y correcciones de seguridad.
- **Limpieza de Datos:** Se realiza la limpieza periódica de datos redundantes o innecesarios, asegurando que solo los registros relevantes se mantengan en la base de datos.

- **Beneficio:** Este mantenimiento proactivo ayuda a evitar problemas graves, mejora el rendimiento general del sistema y garantiza que el software esté alineado con las mejores prácticas de la industria.

## 6.2 Actualización de Funcionalidades y Corrección de Errores

Las actualizaciones de funcionalidades y la corrección de errores son parte del proceso continuo para mejorar el sistema y solucionar cualquier problema que surja durante su uso.

- **Actualización de Funcionalidades:**

- **Nuevas Características:** Se añaden funcionalidades adicionales según las necesidades de los usuarios y las peticiones del cliente, como la

optimización de flujos, nuevas opciones de personalización o la mejora de la interfaz.

- **Ajustes en los Flujos de Trabajo:** Se realizan cambios en los flujos de trabajo de los validadores o usuarios para optimizar el proceso y reducir los tiempos de validación.
- **Mejoras en la Usabilidad:** Con base en la retroalimentación recibida de los usuarios y validadores, se ajustan las interfaces para hacerlas más intuitivas y fáciles de usar.
- **Corrección de Errores:**
  - **Errores Detectados por los Usuarios:** Cuando se reportan errores por parte de los usuarios (por ejemplo, problemas con la carga de fotos o fallos en la visualización de los certificados), se investiga la causa y se soluciona de inmediato.
  - **Errores Internos:** El equipo de soporte también realiza auditorías regulares para detectar y corregir errores que podrían no ser evidentes para los usuarios, como fallos en la sincronización de datos o problemas con la generación de certificados.
- **Beneficio:** La corrección rápida de errores y la incorporación de nuevas funcionalidades aseguran que el sistema siga siendo eficiente, confiable y adaptable a las necesidades cambiantes.

### 6.3 Monitoreo del Sistema

El monitoreo continuo del sistema es esencial para detectar posibles problemas antes de que se conviertan en fallos críticos.

- **Herramientas de Monitoreo:**
  - **Monitoreo de Desempeño:** Se implementan herramientas como **Grafana** o **Prometheus** para monitorear el rendimiento del servidor, las consultas a la base de datos y los tiempos de respuesta del sistema.

- **Alertas en Tiempo Real:** Se configuran alertas para notificar al equipo de soporte si se producen errores, caídas del sistema o problemas de rendimiento.
- **Actividades de Monitoreo:**
  - **Revisión de Logs:** Se revisan regularmente los logs del sistema para identificar patrones de error o anomalías que puedan requerir atención.
  - **Monitoreo de Seguridad:** Se realiza un seguimiento continuo de las vulnerabilidades y amenazas de seguridad, implementando medidas para mitigar posibles riesgos.
- **Beneficio:** El monitoreo activo permite detectar y solucionar problemas rápidamente, asegurando que el sistema funcione de manera eficiente y sin interrupciones.

#### 6.4 Pruebas de Calidad y Control de Versión

Las pruebas de calidad son una parte esencial del mantenimiento para garantizar que las nuevas actualizaciones o características no afecten negativamente la funcionalidad existente.

- **Pruebas Realizadas:**
  - **Pruebas de Regresión:** Se ejecutan pruebas de regresión después de cada actualización para asegurarse de que las nuevas características no rompan la funcionalidad existente.
  - **Pruebas de Estrés:** Se simulan cargas altas de usuarios para evaluar el rendimiento del sistema bajo condiciones extremas y asegurar que pueda manejar múltiples solicitudes simultáneas sin fallos.
  - **Pruebas de Seguridad:** Se realizan pruebas periódicas de seguridad para identificar y corregir vulnerabilidades en el sistema, como posibles inyecciones de SQL o ataques de cross-site scripting (XSS).
- **Control de Versiones:**

- **Uso de Git:** El código fuente se mantiene bajo control de versiones utilizando **Git**, permitiendo un seguimiento claro de los cambios realizados y facilitando la gestión de actualizaciones y correcciones.
- **Beneficio:** Las pruebas de calidad y el control de versiones garantizan que el sistema se mantenga estable, seguro y que los nuevos desarrollos no afecten negativamente el rendimiento.

## 6.5 Plan de Soporte Técnico

El soporte técnico es esencial para resolver cualquier problema que pueda surgir durante el uso del sistema.

- **Soporte de Nivel 1:** Se ofrece soporte básico a los usuarios para ayudar con problemas comunes, como el acceso a la plataforma o la recuperación de contraseñas.
- **Soporte de Nivel 2:** El equipo de desarrollo se encarga de resolver problemas más complejos relacionados con la funcionalidad del sistema, como errores en la validación o la emisión de certificados.
- **Soporte de Nivel 3:** En caso de problemas críticos o de seguridad, el equipo de infraestructura y seguridad interviene para realizar auditorías completas y resolver los problemas de manera urgente.

## 7. Documentación y Registros

La documentación es un aspecto crucial para garantizar que el sistema pueda ser comprendido, mantenido y evolucionado por otros equipos en el futuro. Esta sección cubre tanto la documentación técnica como la funcional que es esencial para el soporte y la capacitación de los usuarios y desarrolladores.

### 7.1 Manual de Usuario

El **Manual de Usuario** está diseñado para proporcionar a los usuarios finales una guía paso a paso sobre cómo interactuar con la plataforma. Este manual está dirigido principalmente a los ciudadanos que solicitan certificados y a los validadores responsables de revisar las



solicitudes.

- **Contenido del Manual:**
  - **Proceso de Registro:**
    - Guía para registrarse en el sistema, incluyendo la verificación de correo y la recuperación de contraseñas.
  - **Creación de Solicitudes:**
    - Instrucciones sobre cómo completar el formulario de solicitud de certificado, incluyendo los campos obligatorios y opcionales.
  - **Consulta del Estado de las Solicitudes:**
    - Cómo consultar el estado de las solicitudes (pendiente, aprobado, rechazado, emitido) y cómo interpretar los mensajes del sistema.
  - **Descarga de Certificados:**
    - Instrucciones para descargar el certificado emitido una vez que la solicitud ha sido aprobada.
  - **Resolución de Problemas Comunes:**
    - Guía para resolver problemas frecuentes, como no recibir el correo de verificación o problemas con la contraseña.
- **Beneficio:** El manual asegura que los usuarios puedan utilizar la plataforma de manera autónoma, reduciendo la necesidad de soporte técnico constante.

## 7.2 Manual Técnico para Desarrolladores

El **Manual Técnico para Desarrolladores** está dirigido a los equipos de desarrollo y soporte técnico. Este manual proporciona detalles sobre la estructura del sistema, las tecnologías utilizadas y las prácticas recomendadas para trabajar en el proyecto.

- **Contenido del Manual:**
  - **Descripción de la Arquitectura del Sistema:**
    - Detalles sobre la arquitectura del sistema, incluidos los componentes de front-end y back-end, la base de datos y las interacciones entre ellos.
  - **Estructura del Código:**
    - Descripción de la estructura de directorios y archivos del proyecto, con ejemplos de cómo agregar nuevas funcionalidades o modificar las existentes.
  - **Instalación y Configuración:**
    - Guía paso a paso para la instalación y configuración del sistema en entornos de desarrollo, pruebas y producción.
  - **Manejo de Dependencias:**
    - Instrucciones sobre cómo gestionar las dependencias del sistema utilizando **Composer** y cómo mantener el proyecto actualizado.
  - **Procedimientos de Migración de Base de Datos:**
    - Explicación de cómo realizar migraciones de la base de datos y cómo mantener la integridad de los datos a medida que se implementan nuevas características.
- **Beneficio:** Proporciona a los desarrolladores la información necesaria para comprender y trabajar con el sistema de manera eficiente, permitiendo una fácil actualización y mantenimiento.

### 7.3 Registro de Cambios y Mejoras

El **Registro de Cambios** (changelog) es una herramienta vital para rastrear todas las modificaciones realizadas en el sistema. Este registro proporciona una cronología detallada de todas las actualizaciones, mejoras y correcciones realizadas en la plataforma.

- **Contenido del Registro de Cambios:**
  - **Versiones del Sistema:**
    - Cada nueva versión del sistema se documenta con su número de versión, fecha de lanzamiento y una lista de cambios importantes.
  - **Mejoras Implementadas:**
    - Detalles sobre nuevas características o mejoras en las funcionalidades existentes, como la incorporación de nuevas opciones de personalización o la optimización de la experiencia del usuario.
  - **Corrección de Errores:**
    - Descripción de los errores que se han identificado y corregido, incluyendo problemas de validación de formularios, errores en la generación de certificados, entre otros.
  - **Pruebas Realizadas:**
    - Un resumen de las pruebas de calidad realizadas para garantizar que las nuevas características y correcciones no afecten la funcionalidad existente.
- **Beneficio:** Este registro facilita el seguimiento de los cambios y proporciona a los desarrolladores un historial claro de las mejoras realizadas en el sistema.

#### 7.4 Centralización de la Documentación

Toda la documentación técnica y funcional debe estar centralizada en un repositorio accesible, donde los miembros del equipo puedan consultar la información relevante de manera rápida y eficiente.

- **Características del Repositorio:**
  - **Organización por Secciones:**

- La documentación se organiza en secciones claras: Manual de Usuario, Manual Técnico, Registro de Cambios, Procedimientos de Mantenimiento, entre otros.
- **Acceso Controlado:**
  - El acceso a la documentación debe ser controlado según el rol del usuario. Los desarrolladores y administradores pueden tener acceso completo, mientras que los usuarios pueden tener acceso solo a los manuales relevantes para su uso.
- **Actualización Continua:**
  - La documentación debe mantenerse actualizada con cada nueva versión del sistema. Cada vez que se lanza una nueva funcionalidad o se corrige un error, la documentación debe ser revisada y ajustada en consecuencia.

## 8. Soporte y Capacitación

El soporte técnico y la capacitación son elementos clave para garantizar que tanto los usuarios como el equipo de desarrollo puedan aprovechar al máximo las funcionalidades de la plataforma. Esta sección describe los planes de soporte y las estrategias de capacitación que aseguran la operación continua del sistema y la adopción efectiva por parte de los usuarios y validadores.

### 8.1 Soporte Técnico

El soporte técnico es fundamental para resolver problemas que puedan surgir durante el uso de la plataforma. El equipo de soporte debe estar preparado para gestionar cualquier incidencia técnica que los usuarios o validadores puedan enfrentar, asegurando que el sistema se mantenga operativo y eficiente.

- **Niveles de Soporte:**

- **Nivel 1 - Soporte Básico:**

- **Descripción:** El soporte básico está destinado a resolver problemas comunes, como la recuperación de contraseñas, el acceso al sistema

o problemas con la interfaz de usuario.

- **Canales:** El soporte básico se puede proporcionar a través de una plataforma de tickets, correo electrónico o chat en línea.

- **Tiempo de Respuesta:** Respuesta dentro de las 24 horas hábiles.

- **Nivel 2 - Soporte Intermedio:**

- **Descripción:** Este nivel cubre problemas más complejos, como errores en la validación de datos, problemas de sincronización de solicitudes o fallos en la generación de certificados.

- **Canales:** Correo electrónico y soporte telefónico.

- **Tiempo de Respuesta:** Respuesta dentro de las 48 horas hábiles.

- **Nivel 3 - Soporte Avanzado:**

- **Descripción:** El soporte avanzado está destinado a abordar problemas críticos, como fallos en la infraestructura del sistema o problemas graves de seguridad.

- **Canales:** Soporte telefónico directo o reuniones remotas.

- **Tiempo de Respuesta:** Respuesta inmediata en situaciones críticas.

- **Proceso de Resolución de Incidencias:**

- **Recepción de Incidencias:** Las incidencias son registradas a través del sistema de soporte o por contacto directo con el equipo de soporte.
- **Diagnóstico:** El equipo de soporte evalúa el problema y determina si es necesario escalar la incidencia a un nivel superior.
- **Solución:** El equipo de soporte resuelve el problema y, si es necesario, realiza pruebas para verificar que la solución ha sido efectiva.

- **Cierre de Incidencia:** Una vez resuelto el problema, se cierra la incidencia y se informa al usuario de la solución implementada.
- **Beneficio:** Este enfoque estructurado de soporte asegura que los problemas sean gestionados rápidamente y de manera efectiva, minimizando el tiempo de inactividad del sistema.

## 8.2 Capacitación para Administradores y Validadores

La capacitación es clave para asegurar que los administradores y validadores puedan gestionar y utilizar la plataforma de manera eficiente. A continuación, se detallan los planes de capacitación para estos dos roles fundamentales:

- **Capacitación para Administradores:**
  - **Objetivos:**
    - Asegurar que los administradores comprendan todos los aspectos técnicos del sistema, incluida la gestión de usuarios, la configuración del sistema y el manejo de incidencias.
    - Capacitar a los administradores en el proceso de actualización y mantenimiento del sistema.
  - **Contenido de la Capacitación:**
    - **Introducción al Sistema:** Visión general del sistema y sus módulos principales.
    - **Gestión de Usuarios:** Cómo agregar, editar y eliminar usuarios, y cómo asignar roles y permisos.
    - **Monitoreo del Sistema:** Uso de herramientas de monitoreo y revisión de logs para identificar problemas.
    - **Actualización del Sistema:** Procedimientos para actualizar el sistema y realizar copias de seguridad de la base de datos.

- **Métodos de Capacitación:**
  - Talleres presenciales y virtuales.
  - Materiales de referencia (manuales, videos tutoriales).
  - Sesiones prácticas donde los administradores puedan realizar tareas específicas en un entorno de prueba.
- **Capacitación para Validadores:**
  - **Objetivos:**
    - Asegurar que los validadores comprendan el proceso de validación, desde la revisión de solicitudes hasta la emisión de certificados.
    - Capacitar a los validadores en el uso de la plataforma para registrar observaciones, verificar la autenticidad de los documentos y gestionar los estados de las solicitudes.
  - **Contenido de la Capacitación:**
    - **Flujos de Validación:** Cómo revisar las solicitudes, las acciones a realizar y cómo proceder con la validación de documentos.
    - **Uso de Formularios:** Cómo completar y enviar formularios de validación, y cómo registrar notas y observaciones.
    - **Gestión de Estados de Solicitudes:** Cómo cambiar los estados de las solicitudes, desde "Pendiente" hasta "Emitido".
    - **Visitas Domiciliarias (para Certificados de Avecindamiento):** Cómo realizar y registrar visitas domiciliarias, incluyendo la toma y carga de fotos.
  - **Métodos de Capacitación:**
    - Sesiones prácticas, donde los validadores simulan el proceso de

validación en un entorno de prueba.

- Materiales visuales, como infografías y guías rápidas, que faciliten el aprendizaje.
- **Beneficio:** La capacitación garantiza que tanto los administradores como los validadores utilicen la plataforma de manera efectiva, lo que mejora la eficiencia operativa y reduce los errores durante el proceso de validación.

### 8.3 Capacitación para Usuarios Finales

Aunque los usuarios finales generalmente no necesitan soporte técnico directo, es importante proporcionar capacitación sobre cómo usar la plataforma de manera autónoma. El objetivo es minimizar la necesidad de asistencia y mejorar la experiencia del usuario.

- **Contenido de la Capacitación:**
  - **Registro y Creación de Solicitudes:** Instrucciones claras sobre cómo registrarse en la plataforma y cómo completar una solicitud de certificado.
  - **Consulta del Estado de Solicitudes:** Cómo consultar el estado de las solicitudes y cómo interpretar los mensajes del sistema.
  - **Descarga de Certificados:** Guía paso a paso para descargar los certificados emitidos.
  - **Resolución de Problemas Comunes:** Instrucciones para resolver problemas frecuentes, como olvidos de contraseñas o problemas con la verificación de correo.
- **Métodos de Capacitación:**
  - Videos tutoriales accesibles desde la misma plataforma.
  - Manuales en PDF y guías paso a paso.
  - Soporte en línea, a través de una sección de preguntas frecuentes (FAQ) y



un sistema de tickets para problemas más complejos.

## 9. Anexos

Los anexos proporcionan información adicional, diagramas, ejemplos de código, y otros materiales complementarios que son útiles para comprender completamente el sistema y su implementación. Esta sección incluye recursos que pueden ser consultados por el equipo de desarrollo, administradores, y otros actores relevantes para facilitar el uso y mantenimiento de la plataforma.

### 9.1 Diagramas de Arquitectura

Los diagramas de arquitectura ayudan a comprender la estructura del sistema y cómo interactúan sus componentes principales. A continuación se incluye un diagrama representativo de la arquitectura general del sistema.

- **Diagrama de Arquitectura General:**
  - Este diagrama muestra cómo los componentes del sistema (front-end, back-end, base de datos, etc.) se comunican entre sí y cómo los usuarios y validadores interactúan con el sistema.

[ Front-End (Livewire + Tailwind CSS) ]

|

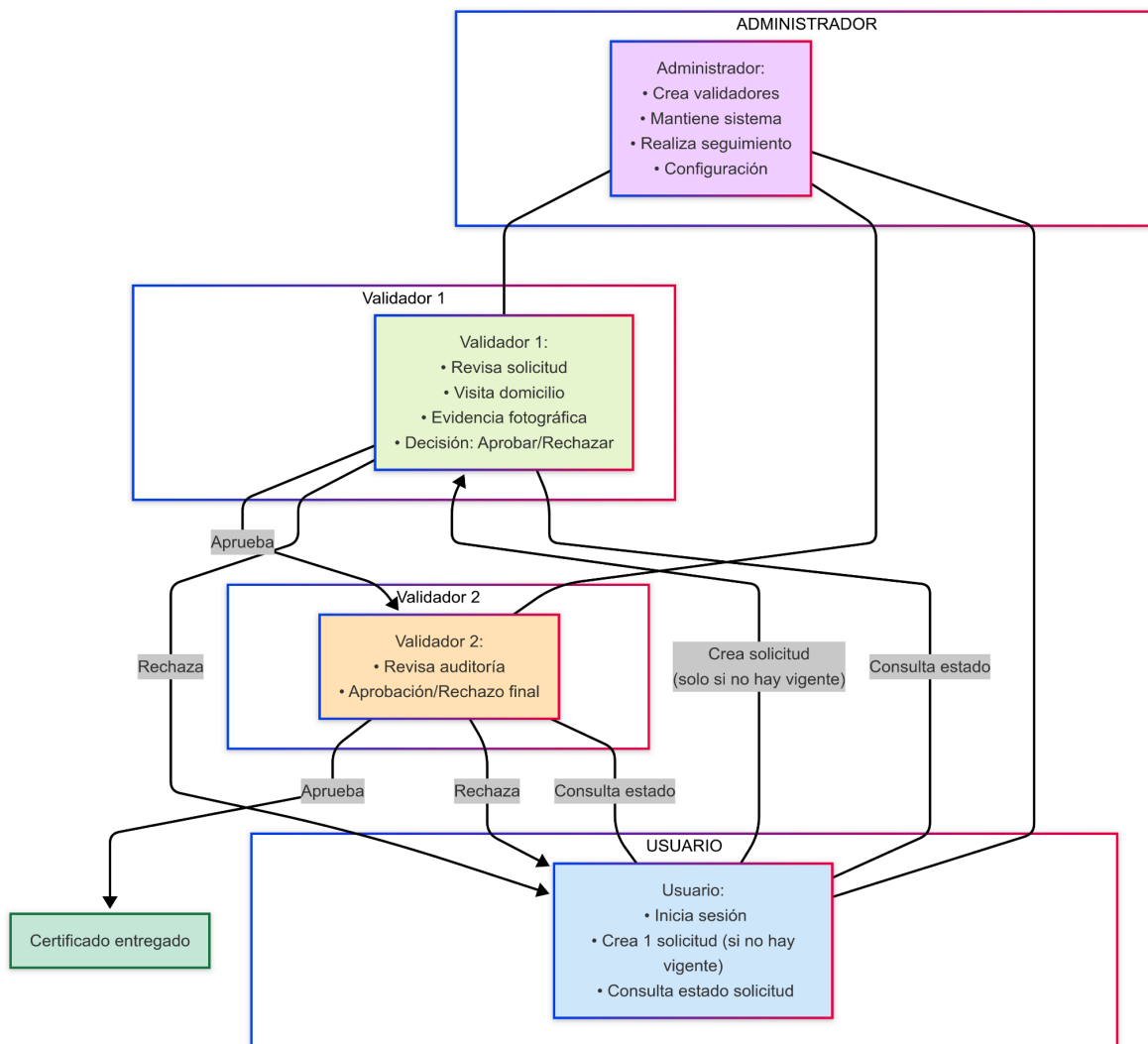
[ Back-End (Laravel + Jetstream) ]

|

[ Base de Datos (MySQL/PostgreSQL) ]

|

[ APIs Externas (Validación, Sistema Central) ]



### Descripción:

- **Front-End:** Interfaz de usuario desarrollada con Livewire y Tailwind CSS, encargada de mostrar los formularios y los estados de las solicitudes.
- **Back-End:** Lógica de negocio gestionada con Laravel, encargado de procesar las solicitudes, la validación de datos y la emisión de certificados.
- **Base de Datos:** Almacena la información de los usuarios, solicitudes, validadores y certificados.

- **APIs Externas:** Servicios que se conectan con el sistema para realizar validaciones de datos, aunque en este sistema las verificaciones se hacen manualmente por los validadores.

## 9.2 Ejemplos de Código

A continuación se proporcionan algunos ejemplos de código que ilustran cómo se manejan ciertas funcionalidades clave dentro del sistema:

- **Generación de Certificados en PDF:**

```
// Código para generar certificado PDF con datos del usuario
```

```
$pdf = PDF::loadView('certificado', ['usuario' => $usuario]);
```

```
return $pdf->download('certificado_'.$usuario->id.'.pdf');
```

**Descripción:** Este fragmento de código genera el certificado en formato PDF utilizando los datos del usuario. La librería **PDF** utilizada se integra con Laravel para facilitar la generación del archivo.

- **Validación de Datos de Usuario:**

```
// Validación de datos del usuario
```

```
$validatedData = $request->validate([
```

```
'nombre' => 'required|string|max:255',
```

```
'direccion' => 'required|string|max:255',
```

```
'documento_identidad' => 'required|numeric',
```

```
]);
```

**Descripción:** Este código valida los datos del formulario de solicitud del usuario antes de ser procesados. Si los datos no cumplen con los requisitos, se genera un error y se le muestra al usuario un mensaje de validación.

## 9.3 Procedimientos de Instalación

Los procedimientos de instalación detallan los pasos necesarios para desplegar el sistema en un nuevo entorno. Esto es esencial para garantizar que el sistema se configure correctamente en servidores de desarrollo, pruebas y producción.

- **Requisitos del Servidor:**

1. Sistema operativo: Ubuntu 20.04 LTS o superior.
2. PHP 8.2 o superior.
3. Base de datos MySQL o PostgreSQL.
4. Servidor web: Nginx o Apache.

**Pasos de Instalación:**

1. **Clonar el Repositorio:**

```
git clone https://github.com/jhonrymat/acacias.git
```

2. **Instalar Dependencias:**

```
cd sistema-certificados
```

```
composer install
```

3. **Configurar el Archivo .env:**

- Copiar el archivo **.env.example** a **.env** y configurar las variables necesarias para la base de datos, correo electrónico, etc.

```
cp .env.example .env
```

4. Generar Clave de Aplicación:

```
php artisan key:generate
```

5. Migrar Base de Datos:

```
php artisan migrate
```

6. Iniciar el Servidor:

```
php artisan serve
```

**Descripción:** Estos pasos cubren desde la clonación del repositorio hasta la configuración de la base de datos y la ejecución del servidor local.

#### 9.4 Registro de Cambios (Changelog)

El registro de cambios documenta todas las modificaciones realizadas en el sistema, incluidas las actualizaciones de funcionalidades, corrección de errores y mejoras en la seguridad.

- **Ejemplo de Registro:**
  - **Versión 1.0.0:**
    - Lanzamiento inicial del sistema con funcionalidades básicas de registro de usuarios y solicitud de certificados.
  - **Versión 1.1.0:**
    - Implementación de la funcionalidad de validadores (Validador 1 y Validador 2).
  - **Versión 1.2.0:**
    - Mejora en la interfaz de usuario para la consulta de solicitudes y

generación de certificados.

- **Versión 1.3.0:**

- Implementación de validación manual de documentos y verificación de dirección.
- Mejora en la experiencia del usuario en dispositivos móviles.

## 9.5 Glosario de Términos Técnicos

En esta sección se definen algunos de los términos técnicos utilizados en el manual para asegurar que todos los usuarios comprendan claramente el funcionamiento del sistema.

- **API (Interfaz de Programación de Aplicaciones):** Un conjunto de funciones y procedimientos que permiten que las aplicaciones interactúen entre sí.
- **PDF (Formato de Documento Portátil):** Formato de archivo utilizado para la generación de certificados.
- **ORM (Mapeo de Objetos Relacionales):** Técnica utilizada en Laravel para interactuar con bases de datos de manera sencilla y eficiente.
- **Livewire:** Librería de Laravel que permite la creación de interfaces dinámicas sin necesidad de recargar la página.