**DIRECCION NACIONAL DE VIVIENDA SOCIAL RESIDUAL**



GUIA DE USO – MODULO DE ADMINISTRACION DE CARTERA

"Gestión de Cartera - DNVSR" V 1.0

La Paz - 2024

TABLA DE CONTENIDO

[1 Introducción 3](#_Toc183764925)

[2 Requerimientos Tecnicos 3](#_Toc183764926)

[2.1 ENTORNO SERVIDOR 3](#_Toc183764927)

[2.2 BASE DE DATOS 3](#_Toc183764928)

[2.2.1 Modelado de la base de datos 3](#_Toc183764929)

[2.3 CONFIGURACION DEL ENTORNO 4](#_Toc183764930)

[2.4 almacenamiento 4](#_Toc183764931)

[2.5 Caché y sesiones 4](#_Toc183764932)

[2.6 Correo electrónico 5](#_Toc183764933)

[2.7 Cola de trabajos 5](#_Toc183764934)

[2.8 Localización 5](#_Toc183764935)

[2.9 Seguridad 5](#_Toc183764936)

[2.10 Front-end 5](#_Toc183764937)

[3 Requerimientos específicos de Laravel 5](#_Toc183764938)

[3.1 Entorno de desarrollo 5](#_Toc183764939)

[3.2 Deployment / lanzamiento 5](#_Toc183764940)

[4 Estructura del proyecto 5](#_Toc183764941)

[5 Autenticación 6](#_Toc183764942)

[6 Secciones y funcionalidad 6](#_Toc183764943)

[6.1 Inicio 6](#_Toc183764944)

[6.2 Beneficiarios 7](#_Toc183764945)

[6.3 PROYECTOS 9](#_Toc183764946)

[7 LISTA DE LIBRERIAS INDISPENSABLES PARA EL SISTEMA 9](#_Toc183764947)

[8 LISTA DE CDN’S UTILIZADAS POR EL SISTEMA 10](#_Toc183764948)

# Introducción

Esta guía de uso describe la estructura y funcionamiento del sistema "Gestión de Cartera - DNVSR" V 1.0, una aplicación desarrollada en Laravel para gestionar beneficiarios, pagos, planes y vouchers.

# Requerimientos Tecnicos

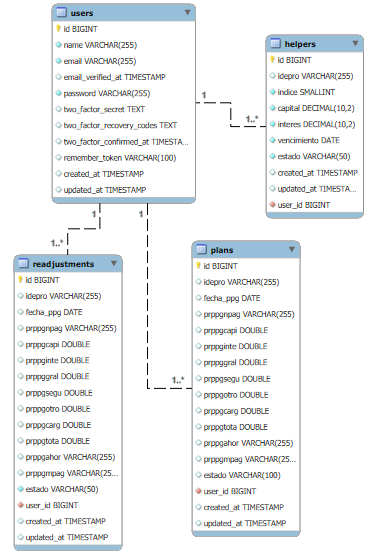
## ENTORNO SERVIDOR

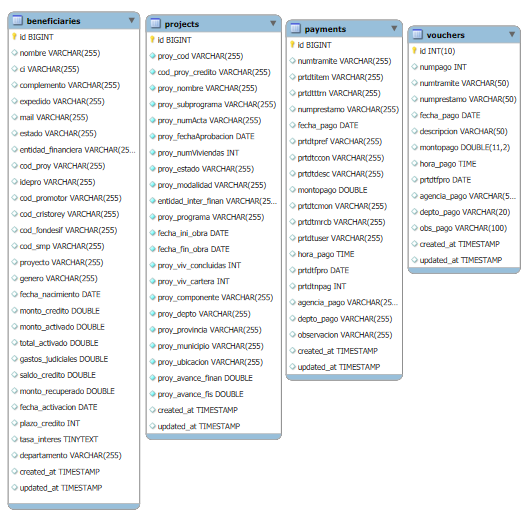
* PHP ^8.3
* Servidor web compatible con PHP (Apache, Nginx, etc.)
* Composer (gestor de dependencias de PHP)
* Node.js y NPM (para compilación de assets frontend)

## BASE DE DATOS

* PostgreSQL ^17.0
* Puerto de base de datos: 2255 (configurable)

### Modelado de la base de datos





## CONFIGURACION DEL ENTORNO

* Archivo “.env” configurado con las variables de entorno necesarias
* APP\_KEY generada (ya está presente en el archivo “.env” proporcionado)
* Zona horaria configurada a UTC (ajustable según necesidades)

## almacenamiento

Sistema de archivos local (configurable para usar otros drivers como S3).

## Caché y sesiones

* Caché configurada para usar la base de datos
* Sesiones configuradas para usar la base de datos
* Tiempo de vida de la sesión: 480 minutos (configurable)

## Correo electrónico

Configurado para usar el driver de log en desarrollo (configurable para producción).

## Cola de trabajos

Configurada para usar la base de datos.

## Localización

Configurado para español (es) con fallback a español (UTC por defecto del Framework).

## Seguridad

BCRYPT\_ROUNDS configurado a 12 para hashing de contraseñas.

El proyecto implementa múltiples capas de seguridad utilizando los frameworks Laravel Fortify, Jetstream y Sanctum. A continuación, se detallan los principales componentes de seguridad:

### Autenticación y Control de Acceso

#### Sistema de Autenticación

- Implementación de guardias de autenticación web mediante Laravel Fortify

- Gestión de sesiones seguras con middleware personalizado

- Protección de rutas API mediante tokens Sanctum

#### Gestión de Contraseñas

- Validación robusta de contraseñas con reglas predefinidas

- Sistema de restablecimiento de contraseñas seguro

- Almacenamiento encriptado de contraseñas mediante hashing de 12 rondas

#### Características de Seguridad Avanzadas

- Autenticación de dos factores (2FA)

- Verificación de correo electrónico

- Gestión de perfiles de usuario con actualizaciones seguras

#### Protección contra Amenazas

1. Middleware de Seguridad

- Encriptación de cookies

- Validación de tokens CSRF

- Autenticación de sesiones

#### API Security

- Tokens de acceso mediante Sanctum

- Dominios estatales configurables para API

- Gestión de expiración de tokens

#### Características Adicionales

- Sistema de límite de intentos (Rate Limiting) para prevenir ataques de fuerza bruta

- Redirecciones seguras post-autenticación

- Gestión de sesiones múltiples

- Validación de datos de entrada en todas las operaciones de usuario

Esta implementación proporciona una base sólida de seguridad que protege tanto la interfaz web como las APIs del sistema, asegurando la integridad de los datos y la privacidad de los usuarios.

## Front-end

Vite configurado para compilación de assets.

# Requerimientos específicos de Laravel

* Extensiones de PHP requeridas por Laravel (INTL, OpenSSL, PGSQL, PDO\_PGSQL, PDO, Mbstring, Tokenizer, XML, XSL, Zip)
* Permisos de escritura en directorios específicos (storage, bootstrap/cache)

## Entorno de desarrollo

* Git para control de versiones

## Deployment / lanzamiento

* Laravel Valet, Laragon, o similares para desarrollo local
* Git para control de versiones

# Estructura del proyecto

El proyecto sigue la estructura estándar de Laravel:

* app/: Contiene el código principal de la aplicación.
* Http/Controllers/\*: Controladores.
* Models/: Modelos de Eloquent.
* routes/: Definiciones de rutas.
* config/: Archivos de configuración.
* database/: Migraciones y factories.

# Autenticación

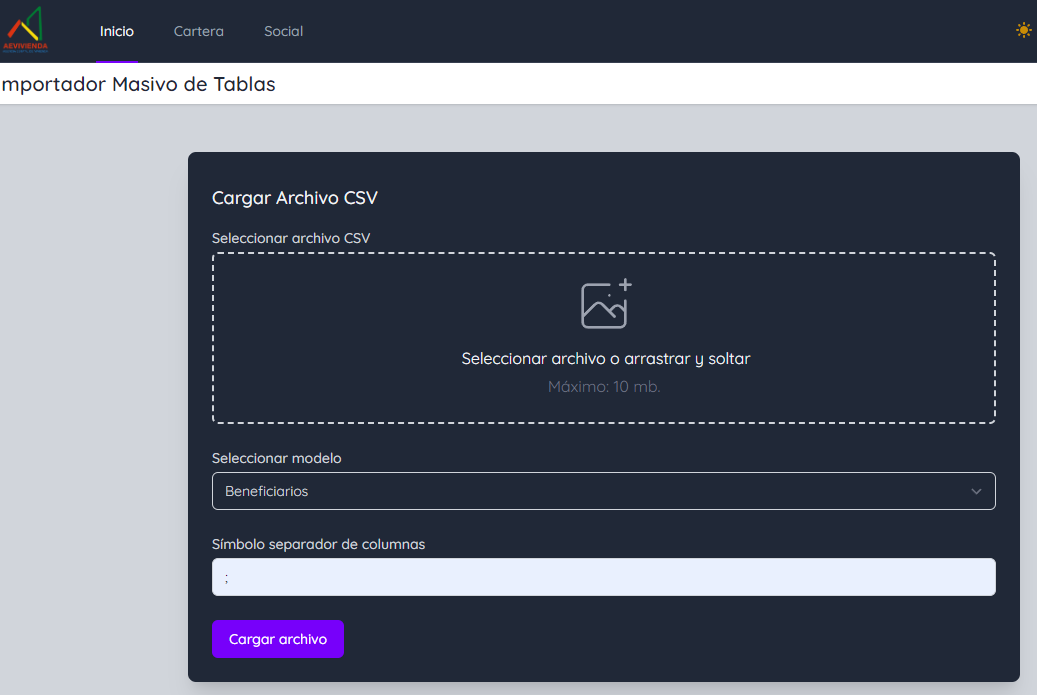
La aplicación utiliza Laravel Fortify mediante Jetstream para la autenticación. Las rutas protegidas están envueltas en el middleware **auth:sanctum**.

# Secciones y funcionalidad

## Inicio

Vista destinada a facilitar la importación masiva de registros de tipo Pago, Planes, Beneficiarios y Vouchers.

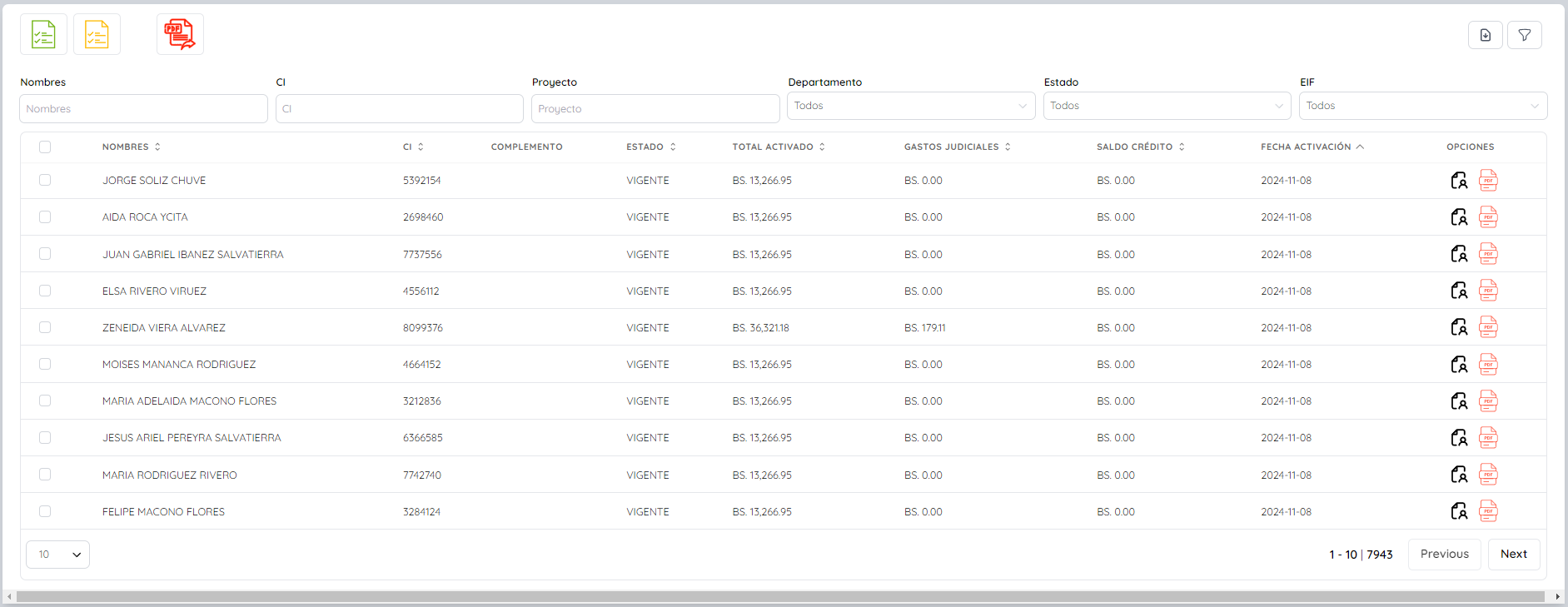
Este solo admite archivos de tipo Comma Separated Values (CSV), además, se deberá especificar el carácter separador de registros (comúnmente usados “,” “;” “|”).



## Beneficiarios

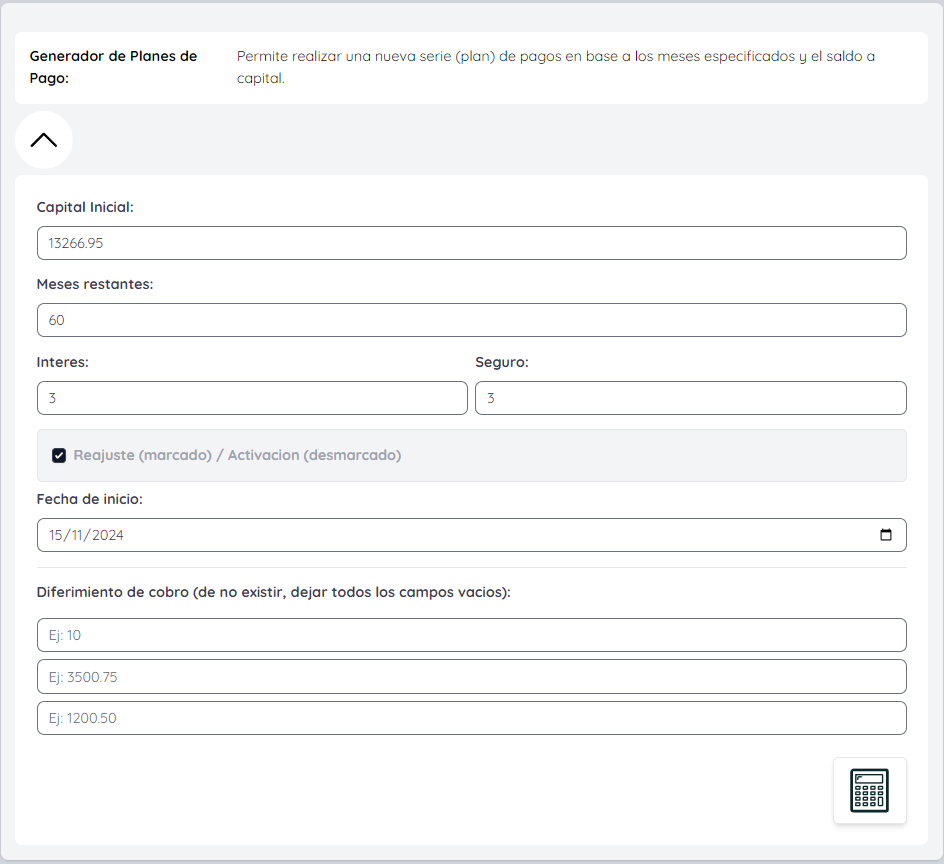
Esta vista permite acceder a la tabla que contiene a todos los beneficiarios existentes en la tabla Beneficiaries, también permite las siguientes operaciones:

* Creación masiva de planes de pagos.
* Reajuste masivo de planes de pagos.
* Exportación masiva / especifica de documentos PDF con los planes de pago del o los beneficiarios.
* Búsqueda avanzada de beneficiarios.
* Exportación masiva / especifica de la lista de beneficiarios, en formato Excel o CSV.



* Revisión de perfil del beneficiario:
  + Datos personales y crediticios del beneficiario
  + Ver lista de planes de pago
  + Ver historial de pagos
  + Generación detallada de:
    - Reajuste de plan de pagos
    - Activación de cartera
    - Diferimiento de cobro (se anexa al final del plan de pagos)



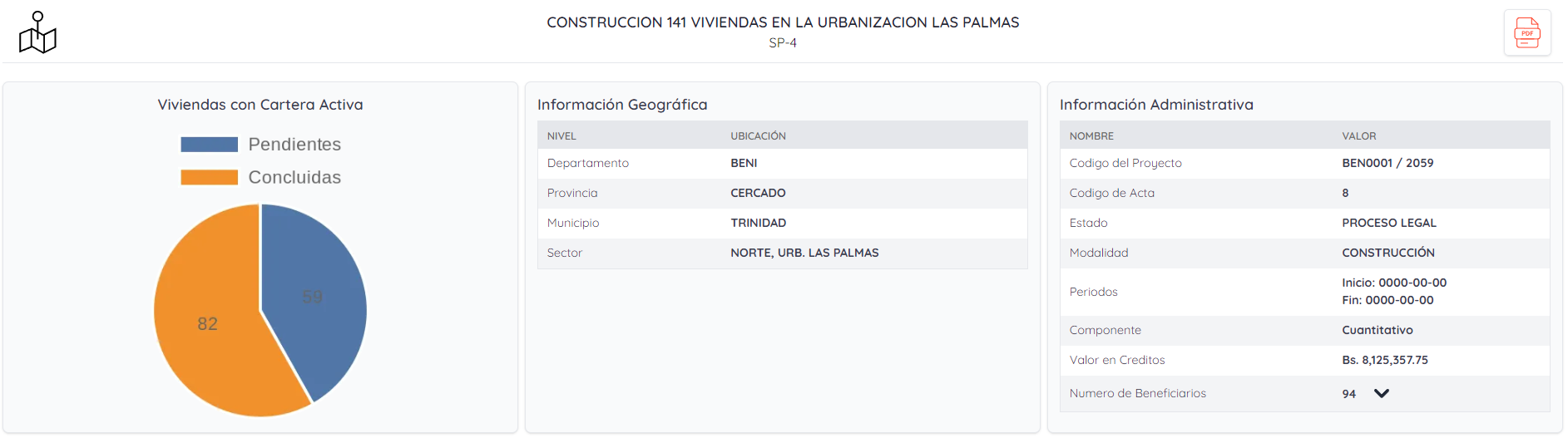


## PROYECTOS

Esta vista permite acceder a la tabla que contiene a todos los proyectos existentes en la tabla Projects, también permite las siguientes operaciones:

* Revisión de datos de proyectos.





# LISTA DE LIBRERIAS INDISPENSABLES PARA EL SISTEMA

La siguiente lista está contenida dentro el archivo composer.json del sistema.

* "php": "^8.3",
* "bacon/bacon-qr-code": "\*",
* "barryvdh/laravel-dompdf": "^3.0",
* "laravel-lang/common": "^6.4",
* "laravel/framework": "^11.9",
* "laravel/jetstream": "^5.2",
* "laravel/sanctum": "^4.0",
* "laravel/tinker": "^2.9",
* "livewire/livewire": "^3.5",
* "openspout/openspout": "^4.26",
* "phpoffice/phpspreadsheet": "^3.3",
* "power-components/livewire-powergrid": "^6.0"

# LISTA DE CDN’S UTILIZADAS POR EL SISTEMA

* <script src="https://code.highcharts.com/highcharts.js"></script>
* <script src="https://code.highcharts.com/highcharts-more.js"></script>
* <script src="https://code.highcharts.com/modules/exporting.js"></script>
* <script src="https://code.highcharts.com/modules/export-data.js"></script>
* <script src="https://code.highcharts.com/modules/accessibility.js"></script>
* <script src="https://code.highcharts.com/highcharts-more.js"></script>
* <script src="https://code.highcharts.com/modules/dumbbell.js"></script>
* <script src="https://code.highcharts.com/modules/lollipop.js"></script>
* <link rel="preconnect" href="https://fonts.googleapis.com">
* <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
* <link ref="https://fonts.googleapis.com/css2?family=Quicksand:wght@300..700&display=swap" rel="stylesheet">
* <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
* <script src="https://cdn.tailwindcss.com"></script>

REPOSITORIO DE TRABAJO

Con el fin de centralizar el desarrollo del sistema, se cuenta con el siguiente repositorio privado:

* https://github.com/DnvsrDesarrollo1/dnvsr-cartera