

Documentación del Módulo de Consulta Sisbén

Manual técnico del módulo responsable de las consultas de información de hogares del Sisbén.

Índice

1. [Introducción](#)
2. [Arquitectura General](#)
3. [Controlador \(Controller\)](#)
4. [Modelo \(Model\)](#)
5. [Vista \(View\)](#)
6. [Flujo de Datos](#)
7. [Seguridad y Buenas Prácticas](#)
8. [Autor\(es\)](#)
9. [Fecha de Actualización](#)

Introducción

Propósito del Módulo

El Módulo de Consulta Sisbén permite a los usuarios buscar y visualizar la información detallada de los hogares registrados en la base de datos del Sisbén a partir del número de documento de uno de sus integrantes.

Alcance y Funcionalidades Principales

- **Consulta por Documento:** Busca todos los hogares asociados a un número de cédula.
- **Visualización de Hogares:** Muestra la información de cada hogar encontrado, incluyendo:
 - Datos de la vivienda (dirección, comuna, barrio).
 - Información del jefe de hogar.
 - Lista completa de los integrantes del hogar.
- **Estadísticas Globales:** Presenta un resumen con la fecha de corte de los datos, el total de hogares y el total de integrantes en el sistema.

Contexto en el Sistema General

Este módulo es una herramienta clave para acceder a la información consolidada del Sisbén, actuando como una interfaz de solo lectura

para los datos almacenados en la tabla `t_hogares_sisben`.

Arquitectura General

El módulo sigue el patrón **Modelo-Vista-Controlador (MVC)** de Laravel para separar la lógica de negocio, la manipulación de datos y la presentación.

- **Controlador** (`SisbenController`): Orquesta el flujo. Recibe las peticiones del usuario, solicita los datos al modelo y los pasa a la vista.
- **Modelo** (`SisbenData`): Interactúa directamente con la base de datos. Representa la tabla `t_hogares_sisben` y define sus relaciones.
- **Vista** (`consulta.blade.php`): Renderiza la interfaz de usuario. Muestra el formulario de búsqueda y los resultados de la consulta.

El flujo de información es unidireccional: la petición del usuario llega al controlador, este consulta al modelo y el resultado se muestra en la vista.

Controlador:

`SisbenController`

- **Ubicación:** `app/Http/Controllers/SisbenController.php`

Responsabilidades

- Gestionar las rutas para mostrar el formulario de consulta y para procesar las búsquedas.
- Validar los datos de entrada (el número de documento).
- Consultar los hogares relacionados con el documento, procesar la información y agruparla por hogar.
- Enviar todos los datos procesados a la vista para su renderización.

Métodos Clave

- `index($documento_profesional = null)` : Muestra la vista inicial del formulario de consulta.
- `search($cedula, $documento_profesional = null)` :
 1. Valida la `cedula` .
 2. Busca en `SisbenData` los `identificacion_tentativa_num_hogares` (IDs de hogar) que coincidan con la `cedula` .
 3. Para cada ID de hogar, obtiene todos sus miembros con las relaciones `barrio` y `comuna` .
 4. Identifica al jefe de hogar.
 5. Agrupa los datos por hogar.
 6. Calcula el total de hogares, total de integrantes y la fecha del último corte.
 7. Retorna la vista `sisben.consulta` con los resultados.

Validaciones

- Se utiliza el `Validator` de Laravel para asegurar que la `cedula` sea un campo requerido, de tipo `string` y con una longitud máxima de 25 caracteres.
-

Modelo: `SisbenData`

- **Ubicación:** `app/Models/SisbenData.php`

Responsabilidades

- Representar la tabla `t_hogares_sisben` de la base de datos.
- Definir los campos que pueden ser asignados masivamente (`$fillable`).
- Especificar el tipo de dato de ciertos atributos para su correcta conversión (`$casts`).
- Establecer las relaciones con otros modelos (`Barrio` y `Comuna`).

Métodos Clave (Relaciones)

- `barrio()` : Define una relación `belongsTo` con el modelo `Barrio` a través de la clave foránea `cod_barrio` .
- `comuna()` : Define una relación `belongsTo` con el modelo `Comuna` a través de la clave foránea `cod_comuna` .

Seguridad

- El uso del array `$fillable` protege contra vulnerabilidades de asignación masiva, permitiendo que solo los campos especificados puedan ser insertados o actualizados a través de los métodos de Eloquent.
-

Vista: `consulta.blade.php`

- **Ubicación:** `resources/views/sisben/consulta.blade.php`

Responsabilidades

- Presentar el formulario de búsqueda al usuario.
- Renderizar los resultados de la búsqueda, incluyendo:
 - Un resumen estadístico.
 - Tarjetas de información para cada hogar encontrado.
 - Tablas con los detalles de los integrantes de cada hogar.
- Mostrar mensajes de estado (inicial, sin resultados).
- Gestionar la interacción del usuario para iniciar una nueva búsqueda.

Características Técnicas

- **Frameworks/Librerías:**
 - **Tailwind CSS:** Para el diseño y los estilos de la interfaz.
 - **jQuery:** Para la gestión de eventos del formulario.
 - **Font Awesome:** Para la iconografía.

- **Motor de Plantillas:** Blade de Laravel.

Interacción AJAX

- El formulario de búsqueda no utiliza AJAX directamente para enviar los datos. En su lugar, un script de jQuery captura el evento `submit`, previene el comportamiento por defecto y redirige al usuario a una URL limpia y semántica (`/sisben/consulta/{cedula}`). Esta acción desencadena una nueva petición GET que es manejada por el método `search` del controlador.

```
// Fragmento del script en la vista
$('#search-form').on('submit', function(e) {
    e.preventDefault();
    var cedula = $('#cedula').val().trim();
    if (cedula) {
        let url = `{{ url('sisben/consulta') }}/${cedula}`;
        // ... Lógica para añadir documento_profesional
        window.location.href = url;
    }
});
```

Flujo de Datos

1. **Petición Inicial:** El usuario navega a la URL `/sisben/consulta`. El método `index` del `SisbenController` retorna la vista `consulta.blade.php`.
2. **Búsqueda:** El usuario introduce un número de documento en el formulario y hace clic en "Buscar".

3. **Redirección:** El script de jQuery intercepta el envío, construye la URL `/sisben/consulta/{cedula_ingresada}` y redirige al navegador.
 4. **Procesamiento:** La nueva petición GET llega al método `search` del `SisbenController`.
 5. **Consulta a BD:** El controlador utiliza el modelo `SisbenData` para buscar todos los hogares asociados a esa cédula.
 6. **Respuesta:** El controlador procesa los datos, calcula las estadísticas y los pasa a la vista `consulta.blade.php`.
 7. **Renderizado:** La vista recibe los datos y los muestra de forma estructurada: estadísticas, tarjetas de hogar y tablas de integrantes.
-

Seguridad y Buenas Prácticas

- **Validación de Entrada:** Todas las entradas del usuario son validadas en el controlador para prevenir datos malformados.
 - **Protección contra Asignación Masiva:** El modelo `SisbenData` utiliza la propiedad `$fillable`.
 - **Prevención de XSS:** Laravel Blade escapa automáticamente las variables mostradas con `{{ $variable }}`, lo que previene ataques de Cross-Site Scripting (XSS).
 - **Relaciones de Eloquent:** Se utilizan las relaciones de Eloquent (`belongsTo`) para cargar datos relacionados de forma segura y eficiente.
-

Autor(es)

- *Equipo de Desarrollo UFM*
-

Fecha de Actualización

- 27 de agosto de 2025