UNIVERSIDAD AMERICANA

Facultad de Ingeniería y Arquitectura

Importación de Ciudades y Ciudadanos en Laravel

Integrantes del grupo:

- Josué David Cano Medina
- José Daniel Levi Loaysiga
- Armando Nicolás Alguera Arróliga

Docente:

- José Duran García

Asignatura:

- Diseño Web y Comercio Electrónico

Repositorio:

- https://github.com/Josuedcm03/registropersona

Introducción

El sistema desarrollado permite gestionar información de ciudades y ciudadanos en una aplicación web construida con Laravel. Como parte de su funcionalidad, se implementó una herramienta para importar datos desde archivos en formato Excel (XLSX) o CSV, lo cual facilita el ingreso masivo de registros sin necesidad de hacerlo manualmente.
Esta funcionalidad es especialmente útil cuando se trabaja con grandes volúmenes de datos, ya que reduce errores humanos, ahorra tiempo y mejora la eficiencia del sistema.
La solución se diseñó para cubrir la importación de datos en dos modelos clave del sistema: City (ciudad) y Citizen (ciudadano), cada uno con sus respectivos campos y validaciones.

Objetivos

- **Objetivo General**
    - Implementar la funcionalidad de importación de datos para los modelos City y Citizen en Laravel, utilizando archivos CSV o Excel como fuente de entrada.
- **Objetivos Específicos**
    - Diseñar interfaces de usuario para cargar archivos de importación.
    - Definir rutas y métodos en los controladores que gestionen el proceso.
    - Integrar la biblioteca Laravel Excel para el procesamiento de archivos.
    - Validar los datos importados antes de insertarlos en la base de datos.

Herramientas utilizadas

- Laravel 10.x: Framework PHP utilizado para el desarrollo del sistema.
- PHP 8.x: Lenguaje de programación base del proyecto.
- Laravel Excel (maatwebsite/excel): Paquete que permite leer y procesar archivos Excel y CSV de forma eficiente.
- Git y GitHub: Control de versiones y alojamiento del código fuente.
- Visual Studio Code (VSCode): Editor de código utilizado para el desarrollo.
- Navegador Web: Para pruebas y visualización de las vistas del sistema.

Desarrollo de la funcionalidad

1. Instalación del Paquete Laravel Excel

Para implementar la funcionalidad de importación desde archivos .xlsx y .csv, se utilizó el paquete Laravel Excel de la biblioteca maatwebsite/excel.

La instalación se realizó mediante Composer ejecutando el siguiente comando en la terminal del proyecto:

```
composer require maatwebsite/excel
```

2. Vistas de cargas de archivo

a. Modificación de index Cities para importación

Se creó al lado del botón que ya existía para crear una ciudad en el index, un botón para abrir un modal, que contiene un formulario sencillo para subir un archivo .csv o .xlsx con datos de ciudades:

```
        <!-- Botón: Importar -->
        <button onclick="document.getElementById('importModalCity').showModal()"
                class="px-4 py-2 bg-green-600 text-white text-sm rounded-md shadow hover:bg-green-700 transition">
            Importar
        </button>
    </div>
</div>

<!-- Modal de Importación de Ciudades -->
<dialog id="importModalCity" class="rounded-lg p-6 shadow-xl w-full max-w-md backdrop:bg-black/30">
    <form method="POST" action="{{ route('cities.import') }}" enctype="multipart/form-data">
        @csrf
        <h2 class="text-xl font-bold mb-4 text-gray-800 dark:text-black">Importar archivo XLSX/CSV</h2>

        <input type="file" name="file"
            class="mb-4 w-full border px-4 py-2 rounded text-sm" required>
        <div class="flex justify-end gap-4 mt-4">
            <button type="button" onclick="document.getElementById('importModalCity').close()"
                    class="px-4 py-2 bg-gray-300 rounded hover:bg-gray-400">
                Cancelar
            </button>
            <button type="submit"
                    class="px-4 py-2 bg-green-600 text-white rounded hover:bg-green-700">
                Importar
            </button>
```

b. Modificación de index Citizens para importación

Del mismo modo, se creó el mismo botón que abre un modal, que contiene un formulario sencillo para subir un archivo .csv o .xlsx con datos de ciudadanos:

```blade
        <!-- Botón: Importar -->
        <button onclick="document.getElementById('importModal').showModal()"
                class="px-4 py-2 bg-green-600 text-white text-sm rounded-md shadow hover:bg-green-700 transition">
            Importar
        </button>
    </div>
</div>

<!-- Modal: Importación de Ciudadanos -->
<dialog id="importModal" class="rounded-lg p-6 shadow-xl w-full max-w-md backdrop:bg-black/30">
    <form method="POST" action="{{ route('citizens.import') }}" enctype="multipart/form-data">
        @csrf
        <h2 class="text-xl font-bold mb-4 text-gray-800 dark:text-black">Importar archivo XLSX/CSV</h2>

        <input type="file" name="file"
            class="mb-4 w-full border px-4 py-2 rounded text-sm" required>

        <div class="flex justify-end gap-4 mt-4">
            <button type="button" onclick="document.getElementById('importModal').close()"
                    class="px-4 py-2 bg-gray-300 rounded hover:bg-gray-400">
                Cancelar
            </button>
            <button type="submit"
                    class="px-4 py-2 bg-green-600 text-white rounded hover:bg-green-700">
                Importar
            </button>
```

3. Rutas definidas

Se definieron dos rutas de tipo POST en el archivo routes/web.php, cada una encargada de recibir el archivo desde la vista correspondiente y dirigirlo al controlador adecuado.

```php
Route::middleware('auth')->group(function () {
    Route::resource('cities', CityController::class);
    Route::resource('citizens', CitizenController::class);
    Route::post('/report', [ReportCitizenController::class, 'send_report'])->name('report');

    // Importar datos desde archivos Excel para ciudadanos
    Route::post('/citizens/import', [CitizenController::class, 'import'])->name('citizens.import');
    // Importar datos desde archivos Excel para ciudades
    Route::post('/cities/import', [CityController::class, 'import'])->name('cities.import');


});
```

4. Métodos en los controladores

a. CityController

El método import() en CityController recibe el archivo desde el formulario, lo procesa usando Laravel Excel, subiendolo como archivo temporal en el proyecto, dándole un nombre único y eliminandolo luego de importarlo.

```php
public function import(Request $request)
{
    // Validar el archivo de importación
    $request->validate([
        'file' => 'required|file|mimes:xlsx,csv',
    ]);

    // Manejar la importación del archivo
    try {
        // Crear un directorio temporal si no existe
        $tmpPath = public_path('tmp');
        // Verificar si el directorio temporal existe, si no, crearlo
        if (!File::exists($tmpPath)) {
            File::makeDirectory($tmpPath, 0755, true);
        }

        // Mover el archivo a un directorio temporal con un nombre único
        $extension = $request->file('file')->getClientOriginalExtension();
        $uniqueName = Str::uuid() . '.' . $extension;
        $filePath = $request->file('file')->move($tmpPath, $uniqueName);

        // Importar el archivo usando la clase CityImport
        Excel::import(new CityImport, $filePath);

        // Eliminar el archivo temporal después de la importación
        File::delete($filePath);
```

b. CitizenController

El método import() en CitizenController es similar, pero utiliza la clase CitizenImport y maneja los campos del modelo ciudadano.

```php
Excel::import(new CitizenImport, $filePath);
```

5. Clases de importación

a. CityImport

```
php artisan make:import CityImport --model=City
```

```php
class CityImport implements ToModel, WithHeadingRow
{
    public function model(array $row)
    {
        // Validamos datos mínimos
        if (empty($row['name'])) {
            return null;
        }

        // Verificamos si la ciudad ya existe por nombre
        return new City([
            'name' => $row['name'],
            'description' => $row['description'] ?? null,
```

b. CitizenImport: Se añaden validaciones respecto al formato del archivo que se subirá y se mantienen las del controlador.

```
php artisan make:import CitizenImport --model=Citizen
```

```php
class CitizenImport implements ToModel, WithHeadingRow, WithMapping
{

    public function map($row): array
    {
        // Evita errores si birth_date viene vacío o como string ya formateado
        // o como número de Excel.
        $birthDate = null;
        if (!empty($row['birth_date'])) {
            if (is_numeric($row['birth_date'])) {
                $birthDate = Date::excelToDateTimeObject($row['birth_date'])->format('Y-m-d');
            } else {
                $birthDate = $row['birth_date'];
            }
        }
        // Retorna un array con los datos del ciudadano, asegurando que cada campo sea opcional
        return [
            'first_name' => $row['first_name'] ?? null,
            'last_name'  => $row['last_name'] ?? null,
            'birth_date' => $birthDate,
            'city_id'    => $row['city_id'] ?? null,
            'address'    => $row['address'] ?? null,
            'phone'      => $row['phone'] ?? null,
        ];
    }
    public function model(array $row)
    {
        // Ignorar filas totalmente vacías

        if (empty($row['first_name']) && empty($row['last_name'])) {
            return null;
        }

        // Validar ciudad existente
        if (!City::where('id', $row['city_id'])->exists()) {
            throw new \Exception("La ciudad con ID {$row['city_id']} no existe.");
        }

        return new Citizen($row);
    }
}
```

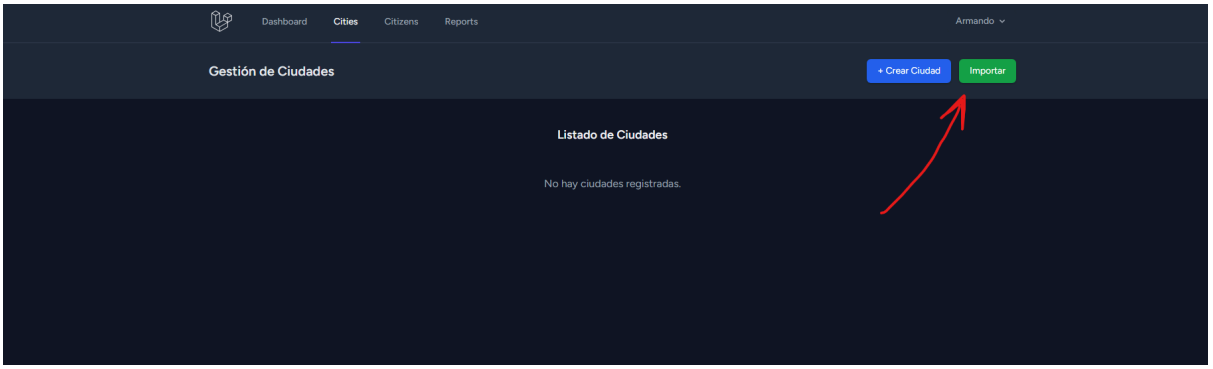
6. Resultados obtenidos

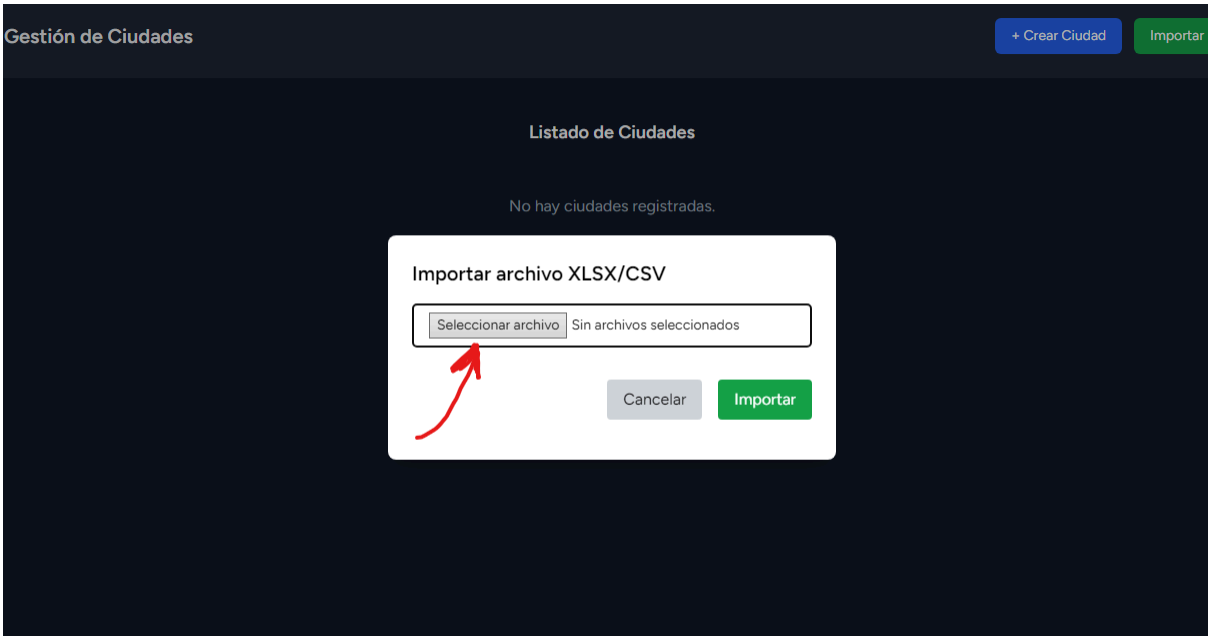
- Ciudades

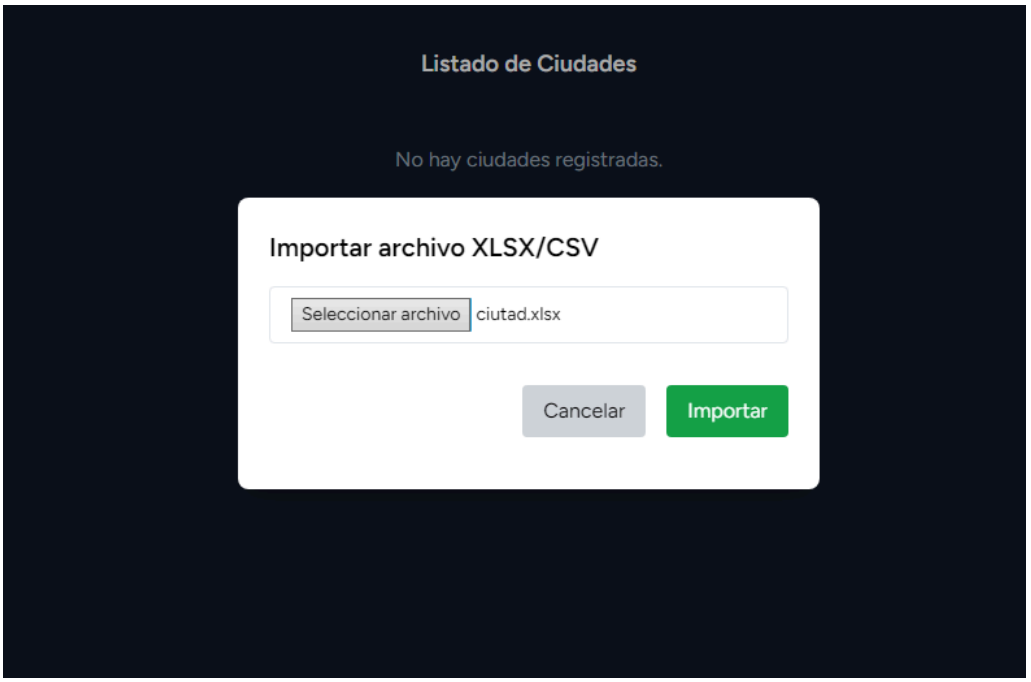
- Archivo a importar

	A	B
1	**name**	**description**
2	Managua	Capital de Nicaragua
3	León	Ciudad colonial
4	Granada	Ciudad turística
5	Estelí	Región norte montañosa

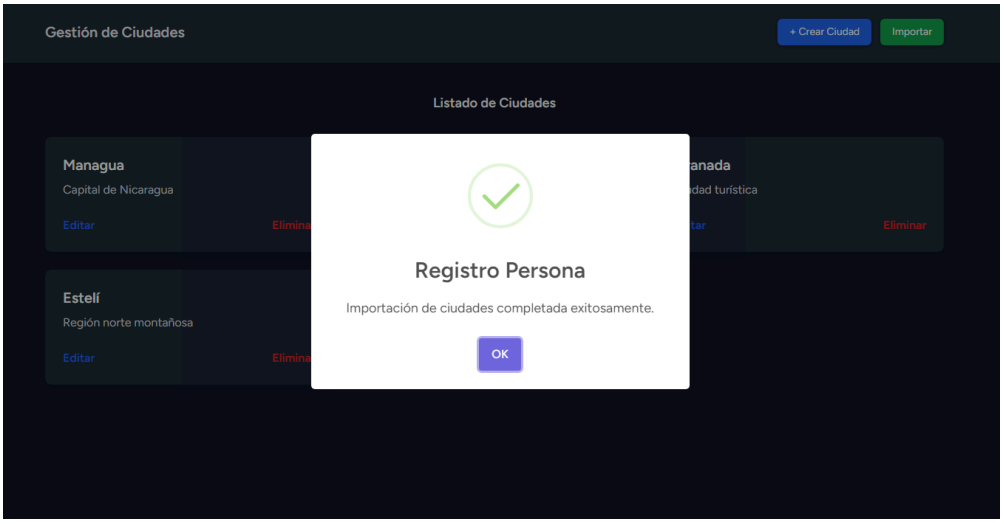
- Proceso de importación

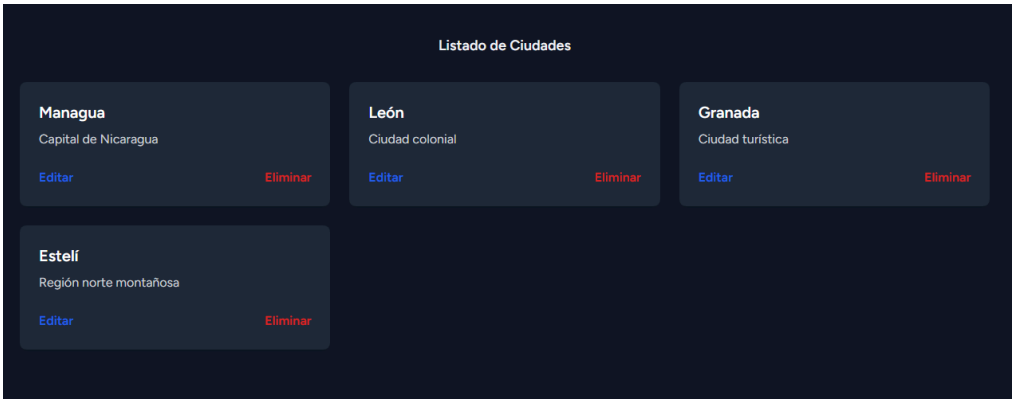





Listado de Ciudades
No hay ciudades registradas.
Importar archivo XLSX/CSV
Seleccionar archivo
ciutad.xlsx
Cancelar
Importar

- **Resultado**

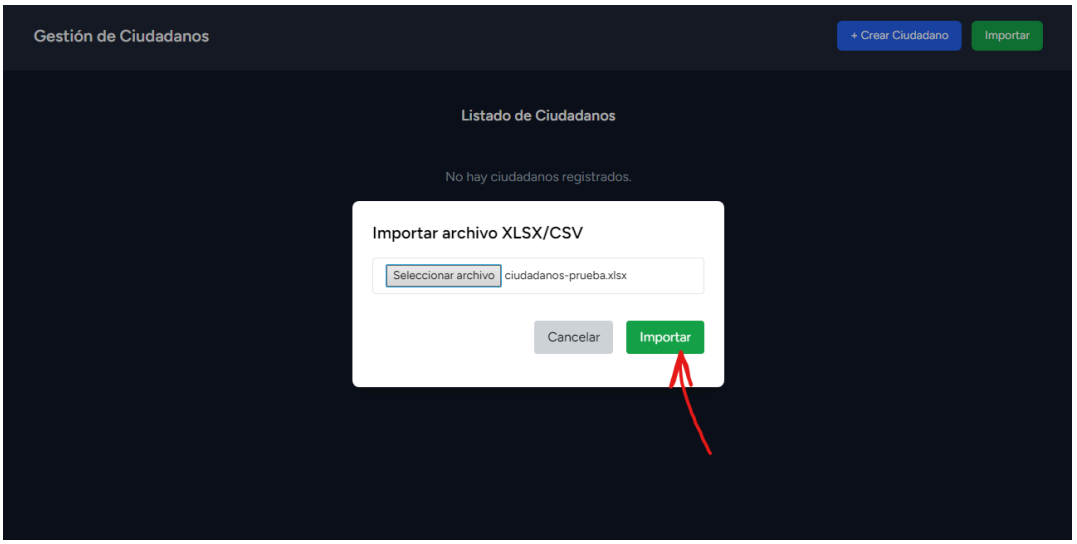

Gestión de Ciudades
+ Crear Ciudad
Importar
Listado de Ciudades
Managua
Capital de Nicaragua
Editar
Eliminar
Estelí
Región norte montañosa
Editar
Eliminar
Registro Persona
Importación de ciudades completada exitosamente.
OK


Listado de Ciudades
Managua
Capital de Nicaragua
Editar
Eliminar
León
Ciudad colonial
Editar
Eliminar
Granada
Ciudad turística
Editar
Eliminar
Estelí
Región norte montañosa
Editar
Eliminar

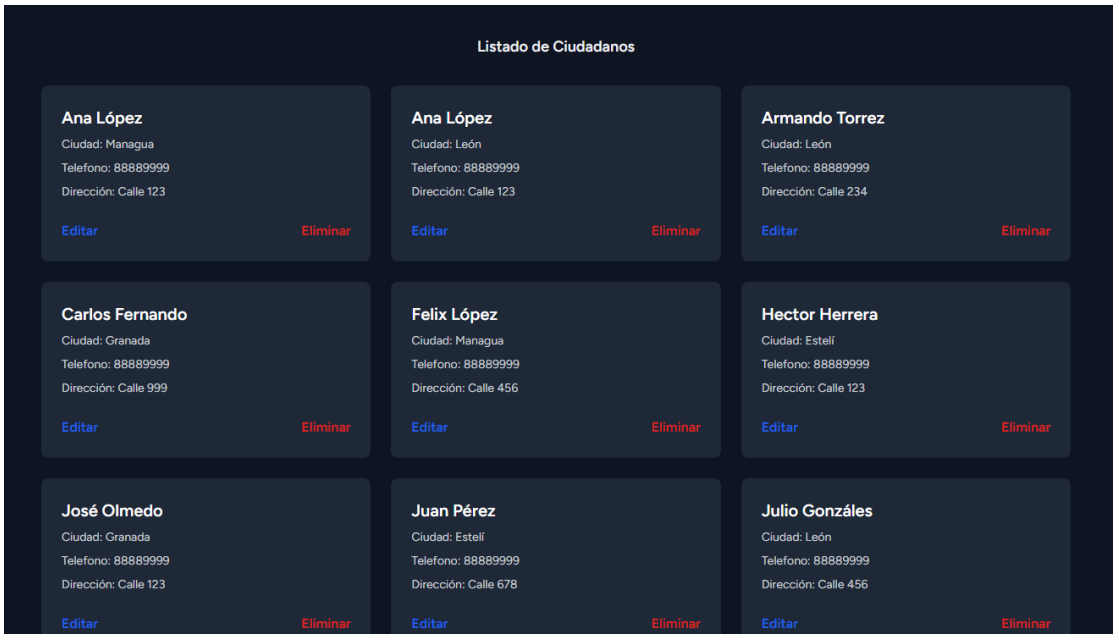
- **Ciudadanos**
  - **Archivo a importar**

	A	B	C	D	E	F
1	first_name	last_name	birth_date	city_id	address	phone
2	Ana	López	14/3/2001	1	Calle 123	88889999
3	Armando	Torrez	4/3/2001	2	Calle 234	88889999
4	Felix	López	24/3/2001	1	Calle 456	88889999
5	Juan	Pérez	14/3/2001	4	Calle 678	88889999
6	Ana	López	14/2/2000	2	Calle 123	88889999
7	Julio	Gonzáles	14/3/2001	2	Calle 456	88889999
8	José	Olmedo	14/3/2001	3	Calle 123	88889999
9	Carlos	Fernando	22/3/1999	3	Calle 999	88889999
10	Hector	Herrera	14/3/2001	4	Calle 123	88889999

  - **Proceso de importación**



  - **Resultado**



7. Conclusiones

La implementación del sistema de importación para los modelos City y Citizen permitió automatizar el ingreso de datos de forma masiva, lo cual representa una mejora significativa en eficiencia y precisión frente al registro manual.
El uso del paquete Laravel Excel facilitó el proceso de lectura de archivos, mientras que SweetAlert2 brindó una experiencia visual clara para los usuarios.

Posibles mejoras futuras:

- Vista previa de los datos antes de guardarlos.
- Reporte más detallado de errores por fila tras la importación.