

h) Crear un controlador, comenzamos con la casilla

```
/var/www/html/proy_lara # php artisan make:controller CasillaController --resource
```

la sentencia anterior genera un archivo en la subcarpeta **app/Http/Controllers/**

con el siguiente contenido:

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\Casilla;
class CasillaController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $casillas = Casilla::all();
        return view('casilla/list', compact('casillas'));
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view('casilla/create');
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $validacion = $request->validate([
            'ubicacion' => 'required|max:100',
        ]);

        $casilla = Casilla::create($validacion);
    }
}
```

```

    return redirect('casilla')->with('success',
    $casilla->ubicacion . ' guardado satisfactoriamente ...');
}

```

```

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */

```

```

public function show($id)
{
    //
}

```

```

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */

```

```

public function edit($id)
{
    $casilla = Casilla::find($id);
    return view('casilla/edit',
        compact('casilla'));
}

```

```

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */

```

```

public function update(Request $request, $id)
{
    $validacion = $request->validate([
        'ubicacion' => 'required|max:100',
    ]);

    Casilla::whereId($id)->update($validacion);
    return redirect('casilla')
        ->with('success', 'Actualizado correctamente...');
}

```

```

/**
 * Remove the specified resource from storage.
 *

```

```

* @param int $id
* @return \Illuminate\Http\Response
*/
public function destroy($id)
{
    $casilla = Casilla::find($id);
    $casilla->delete();
    return redirect('casilla');
}
} //--- end class

```

Se pueden agregar o quitar métodos a la clase anterior y en cada uno de ellos se indicará la lógica de la aplicación

i) En la carpeta resources/views crear un archivo con el nombre de

plantilla.blade.php y pegar el siguiente código:

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Elecciones</title>
    <link href="{{ asset('css/app.css') }}" rel="stylesheet" type="text/css" />
</head>
<body>
    <div class="container">
        @yield('content')
    </div>
    <script src="{{ asset('js/app.js') }}" type="text/js"></script>
</body>
</html>

```

j) Crear una carpeta **casilla** dentro de **resources/views/** y dentro de ella crear tres archivos: **create.blade.php**, **edit.blade.php** e **list.blade.php**, a continuación se pone el contenido de cada uno:

create.blade.php

```

@extends('plantilla')
@section('content')
<style>
    .uper {

```

```

    margin-top: 40px;
}
</style>
<div class="card uper">
  <div class="card-header">
    Agregar Casillas
  </div>
  <div class="card-body">
    @if ($errors->any())
      <div class="alert alert-danger">
        <ul>
          @foreach ($errors->all() as $error)
            <li>{{ $error }}</li>
          @endforeach
        </ul>
      </div><br />
    @endif
    <form method="post" action="{{ route('casilla.store') }}" enctype="multipart/form-data">
      {{ csrf_field() }}
      <div class="form-group">
        @csrf
        <label for="ubicacion">Ubicación:</label>
        <input type="text" class="form-control" name="ubicacion"/>
      </div>

      <button type="submit" class="btn btn-primary">Guardar</button>
    </form>
  </div>
</div>
@endsection

```

edit.blade.php

```

@extends('plantilla')
@section('content')
<style>
  .uper {
    margin-top: 40px;
  }
</style>
<div class="card uper">
  <div class="card-header">
    Editar casilla
  </div>
  <div class="card-body">
    @if ($errors->any())

```

```

<div class="alert alert-danger">
  <ul>
    @foreach ($errors->all() as $error)
      <li>{{ $error }}</li>
    @endforeach
  </ul>
</div><br />
@endif
<form method="POST"
action="{{ route('casilla.update', $casilla->id) }}"
enctype="multipart/form-data">
  {{ csrf_field() }}
  @method('PUT')
  <div class="form-group">
    @csrf
    <label for="id">ID:</label>
    <input type="text"
class="form-control"
readonly="true"
value="{{ $casilla->id }}"
name="id"/>
  </div>

  <div class="form-group">
    @csrf
    <label for="descripcion">Ubicación:</label>
    <input type="text"
value="{{ $casilla->ubicacion }}"
class="form-control"
name="ubicacion"/>
  </div>

  <button type="submit" class="btn btn-primary">Guardar</button>
</form>
</div>
</div>
@endsection

```

list.blade.php

```
@extends('plantilla')

@section('content')
<style>
.uper {
    margin-top: 40px;
}
</style>
<div class="uper">
    @if(session()->get('success'))
        <div class="alert alert-success">
            {{ session()->get('success') }}
        </div><br />
    @endif
    <table class="table table-striped">
        <thead>
            <tr>
                <td>ID</td>
                <td>UBICACION</td>
                <td colspan="2">Action</td>
            </tr>
        </thead>
        <tbody>
            @foreach($casillas as $casilla)
                <tr>
                    <td>{{ $casilla->id }}</td>
                    <td>{{ $casilla->ubicacion }}</td>
                    <td><a href="{{ route('casilla.edit', $casilla->id) }}"
                        class="btn btn-primary">Edit</a></td>
                    <td>
                        <form action="{{ route('casilla.destroy', $casilla->id) }}"
                            method="post">
                            @csrf
                            @method('DELETE')
                            <button class="btn btn-danger" type="submit"
                                onclick="return confirm('Esta seguro de borrar {{ $casilla->ubicacion }}')">Del</button>
                        </form>
                    </td>
                </tr>
            @endforeach
        </tbody>
    </table>
</div>
@endsection
```

k) Crear las rutas para poder acceder a los recursos, para ello editamos el archivo `routes/web.php` y agregar al final del mismo la siguiente instrucción **`Route::resource("candidato", CandidatoController::class);`** y al principio referenciar la clase con

`use App\Http\Controllers\CasillaController;`

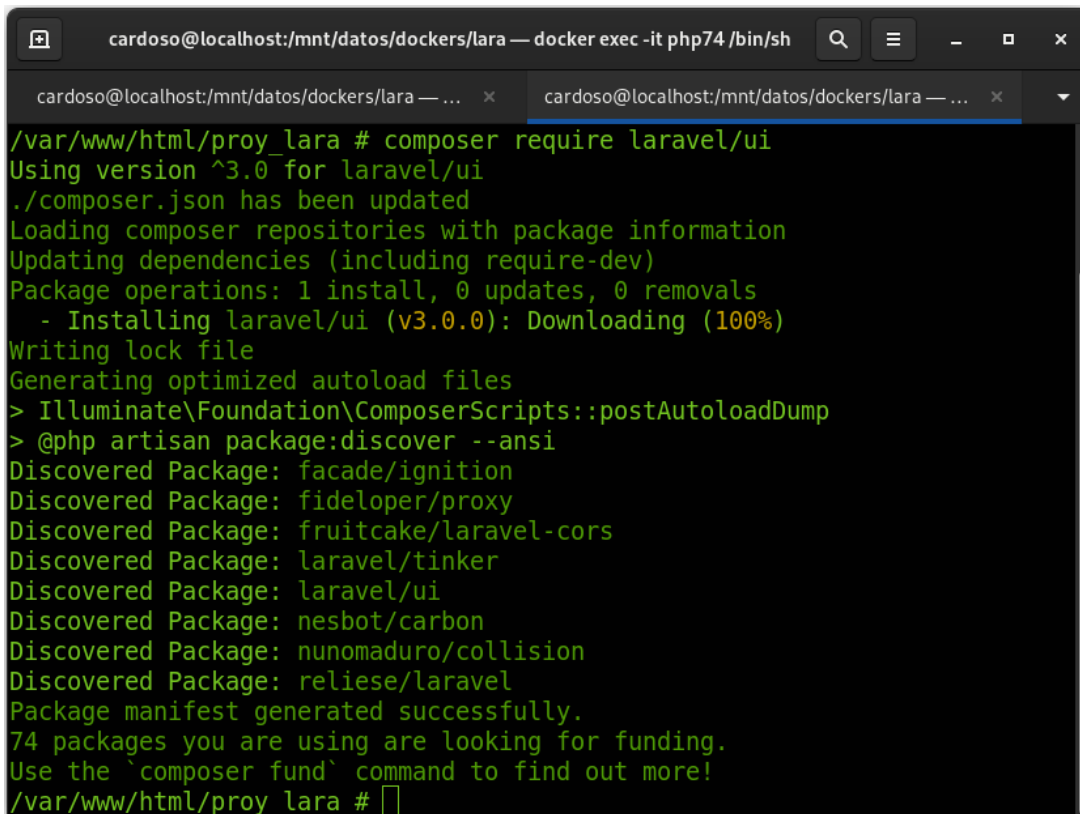
```
<?php
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\CasillaController;

Route::get('/', function () {
    return view('welcome');
});

Route::resource('casilla', CasillaController::class);
```

l) Para dar una apariencia bonita a las vista podemos usar bootstrap, instalando los paquetes correspondientes ejecutando la siguiente sentencia:

`/var/www/html/proy_lara # composer require laravel/ui`



```
cardoso@localhost:/mnt/datos/dockers/lara — docker exec -it php74 /bin/sh
cardoso@localhost:/mnt/datos/dockers/lara — ... x  cardoso@localhost:/mnt/datos/dockers/lara — ... x
/var/www/html/proy_lara # composer require laravel/ui
Using version ^3.0 for laravel/ui
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
 - Installing laravel/ui (v3.0.0): Downloading (100%)
Writing lock file
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: fruitcake/laravel-cors
Discovered Package: laravel/tinker
Discovered Package: laravel/ui
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Discovered Package: reliese/laravel
Package manifest generated successfully.
74 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
/var/www/html/proy_lara #
```

Figura 5. Error al instalar laravel/ui

Para terminar de configurar ejecute

```
/var/www/html/proy_lara # php artisan ui vue
```

```
/var/www/html/proy_lara # php artisan ui bootstrap
```

```
/var/www/html/proy_lara # npm install && npm run dev
```

Si esta última sentencia marca error de npm no disponible, realice los siguientes pasos:

Windows

<https://bertofern.wordpress.com/2019/01/08/solucion-node-js-npm-no-reconocido-como-comando-interno-o-externo/>

Linux

<https://comoinstalar.info/nodejs-en-linux/>

Prueba desde el navegador <http://localhost:8888/casilla/create>

