Manual del programador

Empleados CamiSmart

Samuel Rodríguez González – 2º Dam

Índice

1. Introducción

Este manual está dirigido a programadores que necesiten mantener, actualizar o ampliar la funcionalidad del sistema **CRUD de empleados** desarrollado en Laravel.

El sistema gestiona tres tablas principales:

- **Tipo de Jornada (`tipo_jornadas`)**
- **Puestos (`puestos`)**
- **Empleados (`datos`)**

Cada tabla tiene sus respectivas **vistas**, **controladores** y **rutas** en ***`web.php`***.

---

2. Configuración del Proyecto

Requisitos Previos

2.1 Composer y Laravel instalados.

2.2 Tener una base de datos MySQL configurada en ***`.env`***:

```
11  DB_CONNECTION=mysql
12  DB_HOST=127.0.0.1
13  DB_PORT=3306
14  DB_DATABASE=empleadoscamissmart
15  DB_USERNAME=root
16  DB_PASSWORD=
```

2.3 Instalar dependencias con:

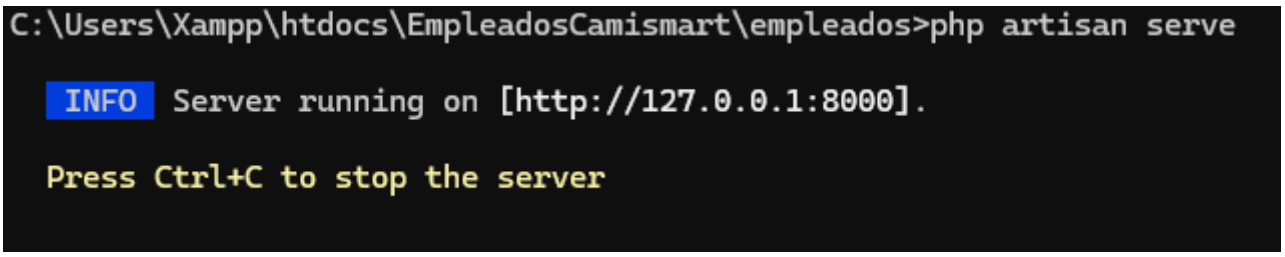
```
- composer install
- npm install && npm run dev
```

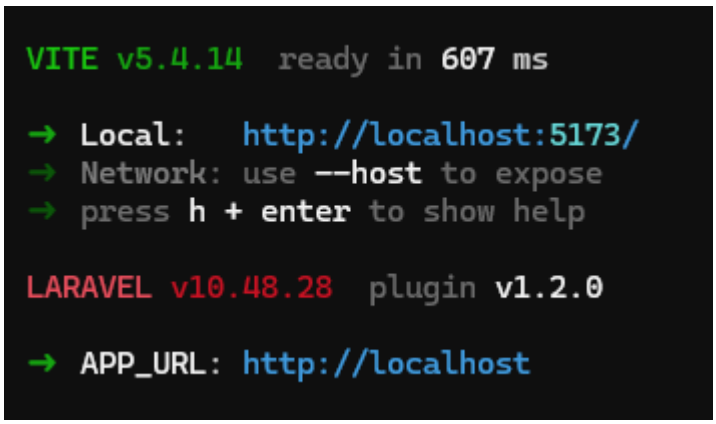
2.4 Ejecutar migraciones:

```
- php artisan migrate
```

2.5 Iniciar el servidor y procesar recursos Frontend:

```
- php artisan serve

- npm run dev
```





3 Rutas (routes/web.php)

El archivo web.php gestiona las rutas de la aplicación.

3.1 Código de web.php:

```php
<?php

use Illuminate\Support\Facades\Route;

Route::get(uri: '/', action: function (): Factory|View {
    return view(view: 'welcome');
});

Auth::routes();
Route::resource(name: 'datos',controller: App\Http\Controllers\DatoController::class)->middleware(middleware: 'auth');
Route::resource(name: 'puestos',controller: App\Http\Controllers\PuestoController::class)->middleware(middleware: 'auth');
Route::resource(name: 'tipo_jornadas',controller: App\Http\Controllers\TipoJornadaController::class)->middleware(middleware: 'auth');
Route::get(uri: '/home', action: [App\Http\Controllers\HomeController::class, 'index'])->name(name: 'home');
```

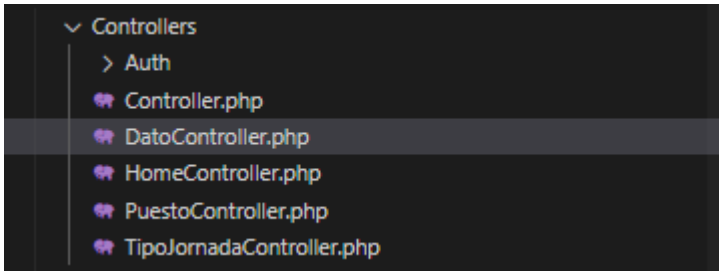
Explicación

- **Auth::routes();** → Genera las rutas para autenticación (login, register, etc.).
- **Route::resource('datos', DatoController::class)->middleware('auth');** → Protege las rutas de datos, permitiendo solo a usuarios autenticados acceder.
- **Se repite para puestos y tipo_jornadas.**
- **Cada Route::resource crea automáticamente rutas para:**
  - index() → Mostrar todos los registros
  - create() → Mostrar formulario de creación
  - store() → Guardar nuevo registro
  - show() → Mostrar un registro específico
  - edit() → Mostrar formulario de edición
  - update() → Guardar cambios en un registro
  - destroy() → Eliminar un registro

---

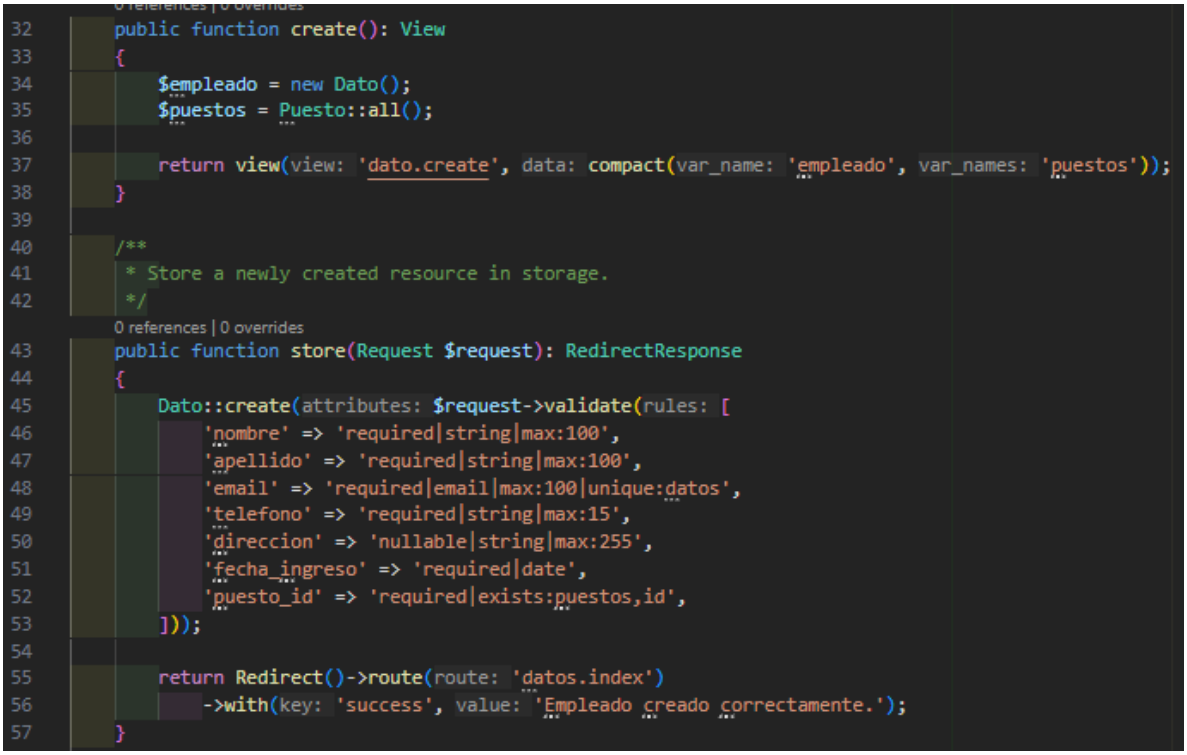
4. Controladores

Los controladores manejan la **lógica del CRUD** para cada modelo.



4.1 DatoController.php

```php
<?php

namespace App\Http\Controllers;

use App\Models\Dato;
use App\Models\Puesto;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Redirect;
use Illuminate\View\View;

/**
 * Class DatoController
 * @package App\Http\Controllers
 */
1 reference | 0 implementations
class DatoController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    0 references | 0 overrides
    public function index(Request $request): View
    {
        $datos = Dato::paginate();

        return view(view: 'dato.index', data: compact(var_name: 'datos'))
            ->with(key: 'i', value: ($request->input(key: 'page', default: 1) - 1) * $datos->perPage());
    }
```



```php
    public function create(): View
    {
        $empleado = new Dato();
        $puestos = Puesto::all();

        return view(view: 'dato.create', data: compact(var_name: 'empleado', var_names: 'puestos'));
    }

    /**
     * Store a newly created resource in storage.
     */
    0 references | 0 overrides
    public function store(Request $request): RedirectResponse
    {
        Dato::create(attributes: $request->validate(rules: [
            'nombre' => 'required|string|max:100',
            'apellido' => 'required|string|max:100',
            'email' => 'required|email|max:100|unique:datos',
            'telefono' => 'required|string|max:15',
            'direccion' => 'nullable|string|max:255',
            'fecha_ingreso' => 'required|date',
            'puesto_id' => 'required|exists:puestos,id',
        ]));

        return Redirect()->route(route: 'datos.index')
            ->with(key: 'success', value: 'Empleado creado correctamente.');
    }
```

Explicación

- index() → Muestra la lista de empleados con paginación.
- create() → Carga la vista del formulario y obtiene los puestos disponibles. - store() → Valida y guarda un nuevo empleado en la base de datos.
- **Los demás métodos (**show**,** edit**,** update**,** destroy**) manejan edición y eliminación.**

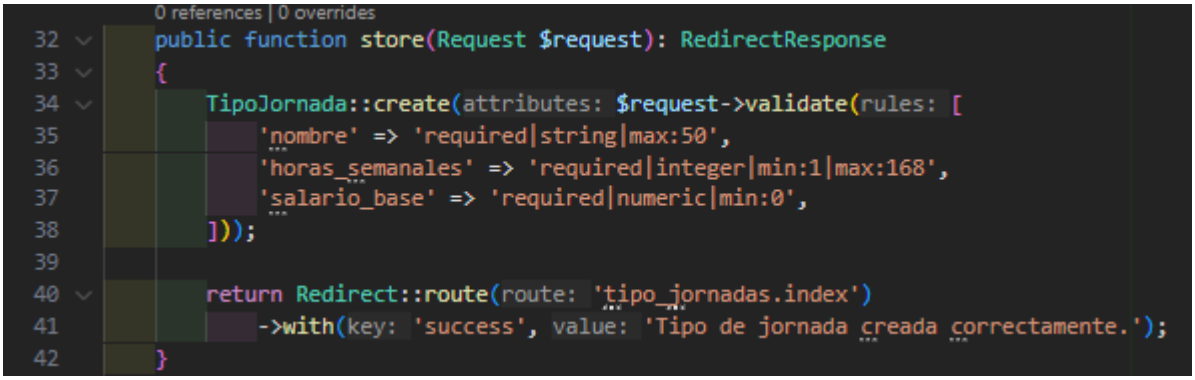
4.2 PuestoController.php

```
0 references | 0 overrides
public function create(): View
{
    $puesto = new Puesto();
    $tipoJornadas = TipoJornada::all();

    return view(view: 'puestos.create', data: compact(var_name: 'puesto', var_names: 'tipoJornadas'));
}

0 references | 0 overrides
public function store(Request $request): RedirectResponse
{
    Puesto::create(attributes: $request->validate(rules: [
        'nombre' => 'required|string|max:100',
        'descripcion' => 'nullable|string',
        'tipo_jornada_id' => 'required|exists:tipo_jornadas,id',
    ]));

    return Redirect::route(route: 'puestos.index')
        ->with(key: 'success', value: 'Puesto creado correctamente.');
}
```

Explicación

- Se usa tipo_jornadas.id como clave foránea en puestos.

4.3 TipoJornadaController.php



```
0 references | 0 overrides
public function store(Request $request): RedirectResponse
{
    TipoJornada::create(attributes: $request->validate(rules: [
        'nombre' => 'required|string|max:50',
        'horas_semanales' => 'required|integer|min:1|max:168',
        'salario_base' => 'required|numeric|min:0',
    ]));

    return Redirect::route(route: 'tipo_jornadas.index')
        ->with(key: 'success', value: 'Tipo de jornada creada correctamente.');
}
```

Explicación

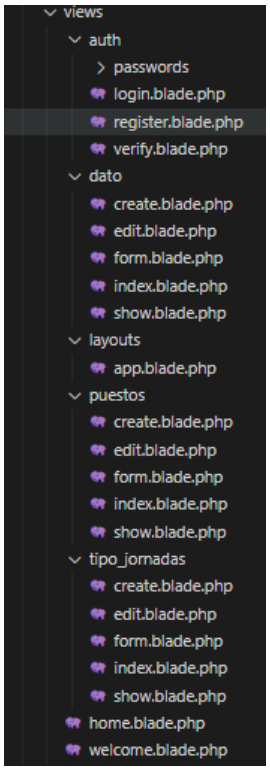
-Se encarga de gestionar los tipos de jornada (horas y salario base).

---

5. Vistas (resources/views/)

Las vistas están organizadas en carpetas:

- views/dato/ → CRUD de empleados
- views/puesto/ → CRUD de puestos
- views/tipo-jornada/ → CRUD de tipo de jornada



Ejemplo de index.blade.php en views/tipo_jornadas/:

```blade
@extends(view: 'layouts.app')

@section(section: 'template_title')
    Tipo Jornadas
@endsection

@section(section: 'content')
    <div class="container-fluid">
        <div class="row">
            <div class="col-sm-12">
                <div class="card">
                    <div class="card-header">
                        <div style="display: flex; justify-content: space-between; align-items: center;">
                            <span id="card_title">
                                {{ __(key: 'Tipo Jornadas') }}
                            </span>

                            <div class="float-right">
                                <a href="{{ route(name: 'tipo_jornadas.create') }}" class="btn btn-primary btn-sm float-right" data-placement="left">
                                  {{ __(key: 'Create New') }}
                                </a>
                            </div>
                        </div>
                    </div>

                    @if ($message = Session::get(key: 'success'))
                        <div class="alert alert-success m-4">
                            <p>{{ $message }}</p>
                        </div>
                    @endif
```

```blade
                    <div class="card-body bg-white">
                        <div class="table-responsive">
                            <table class="table table-striped table-hover">
                                <thead class="thead">
                                    <tr>
                                        <th>Nº</th>
                                        <th>Nombre</th>
                                        <th>Horas Semanales</th>
                                        <th>Salario Base</th>
                                        <th>Acciones</th>
                                    </tr>
                                </thead>
                                <tbody>
                                    @php $i = 0; @endphp <!-- Inicializamos la variable $i para numerar los registros -->
                                    @foreach ($tipoJornadas as $tipoJornada)
                                        <tr>
                                            <td>{{ ++$i }}</td>
                                            <td>{{ $tipoJornada->nombre }}</td>
                                            <td>{{ $tipoJornada->horas_semanales }}</td>
                                            <td>{{ $tipoJornada->salario_base }}</td>
                                            <td>
                                                <form action="{{ route(name: 'tipo_jornadas.destroy', parameters: $tipoJornada->id) }}" method="POST">
                                                    <a class="btn btn-sm btn-primary" href="{{ route(name: 'tipo_jornadas.show', parameters: $tipoJornada->id) }}">
                                                        <i class="fa fa-fw fa-eye"></i> {{ __(key: 'Show') }}
                                                    </a>
                                                    <a class="btn btn-sm btn-success" href="{{ route(name: 'tipo_jornadas.edit', parameters: $tipoJornada->id) }}">
                                                        <i class="fa fa-fw fa-edit"></i> {{ __(key: 'Edit') }}
                                                    </a>
                                                    @csrf
                                                    @method('DELETE')
```

```blade
                                                    @method('DELETE')
                                                    <button type="submit" class="btn btn-danger btn-sm" onclick="event.preventDefault(); confirm('¿Estás seguro de eliminar este tipo de jornada?')
                                                        <i class="fa fa-fw fa-trash"></i> {{ __(key: 'Delete') }}
                                                    </button>
                                                </form>
                                            </td>
                                        </tr>
                                    @endforeach
                                </tbody>
                            </table>
                        </div>
                    </div>
                </div>

                {!! $tipoJornadas->withQueryString()->links() !!}
            </div>
        </div>
    </div>
@endsection
```

Fecha: *21/02/2025* - **Versión: 1.0 - Laravel 10**