

Passo 1: Configuração do Projeto Laravel

1. Abra o Terminal (ou Prompt de Comando)

Navegue até o diretório onde você extraiu o projeto Laravel. Por exemplo:

```
cd C:\Users\SeuUsuario\Documents\hello-world
```

2. Instale as Dependências do Projeto

Execute o comando para instalar as dependências do Laravel:

```
composer install
```

3. Configure o Ambiente

Renomeie o arquivo `.env.example` para `.env`:

```
ren .env.example .env
```

Abra o arquivo `.env` com um editor de texto e configure a conexão com o banco de dados SQLite.

Passo 2: Configuração do Banco de Dados SQLite

1. Crie o Banco de Dados SQLite

- No diretório do projeto, crie um arquivo de banco de dados SQLite. Por exemplo, `database/database.sqlite`.

2. Configure o SQLite no Laravel

No arquivo `.env`, configure as variáveis para usar o banco de dados SQLite:

```
DB_CONNECTION=sqlite
```

```
DB_DATABASE=/absolute/path/to/database/database.sqlite
```

Nota: Substitua `/absolute/path/to/database/database.sqlite` pelo caminho absoluto para o arquivo SQLite criado.

3. Crie as Tabelas

Crie uma migration para as tabelas `professores` e `disciplinas`:

```
php artisan make:migration create_professores_table
```

```
php artisan make:migration create_disciplinas_table
```

Abra as migrations geradas em `database/migrations/` e defina as tabelas:

Migration para professores:

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateProfessoresTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('professores', function (Blueprint $table) {
            $table->id();
            $table->string('nome');
            $table->string('email')->unique();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('professores');
    }
}
```

Código:

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateProfessoresTable extends Migration
{
```

```

/**
 * Run the migrations.
 *
 * @return void
 */
public function up()
{
    Schema::create('professores', function (Blueprint $table) {
        $table->id();
        $table->string('nome');
        $table->string('email')->unique();
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('professores');
}
}

```

Migration para disciplinas:

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateDisciplinasTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('disciplinas', function (Blueprint $table) {
            $table->id();
            $table->string('nome');
            $table->text('descricao')->nullable();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('disciplinas');
    }
}
```

Código:

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateDisciplinasTable extends Migration
{
    /**
     * Run the migrations.
```

```

    *
    * @return void
    */
    public function up()
    {
        Schema::create('disciplinas', function (Blueprint $table) {
            $table->id();
            $table->string('nome');
            $table->text('descricao')->nullable();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('disciplinas');
    }
}

```

Execute as migrations para criar as tabelas:

```
php artisan migrate
```

Passo 3: Implementação do CRUD

1. Crie Modelos e Controladores

Crie os modelos e controladores para **Professores** e **Disciplinas**:

```

php artisan make:model Professor -m
php artisan make:controller ProfessorController
php artisan make:model Disciplina -m
php artisan make:controller DisciplinaController

```

2. Defina as Rotas

No arquivo `routes/web.php`, adicione as rotas para os controladores:

```
use App\Http\Controllers\ProfessorController;  
use App\Http\Controllers\DisciplinaController;  
  
Route::resource('professores', ProfessorController::class);  
Route::resource('disciplinas', DisciplinaController::class);
```

3. Implemente os Métodos dos Controladores

Edite os controladores gerados (`ProfessorController` e `DisciplinaController`) para implementar o CRUD.

Exemplo de `ProfessorController`:

```
<?php  
  
namespace App\Http\Controllers;  
  
use App\Models\Professor;  
use Illuminate\Http\Request;  
  
class ProfessorController extends Controller  
{  
    public function index()  
    {  
        $professores = Professor::all();  
        return view('professores.index', compact('professores'));  
    }  
  
    public function create()  
    {  
        return view('professores.create');  
    }  
  
    public function store(Request $request)  
    {  
        $request->validate([  
            'nome' => 'required',  
            'email' => 'required|email|unique:professores',  
        ]);  
        Professor::create($request->all());  
        return redirect()->route('professores.index');  
    }  
}
```

```

public function edit(Professor $professor)
{
    return view('professores.edit', compact('professor'));
}

public function update(Request $request, Professor $professor)
{
    $request->validate([
        'nome' => 'required',
        'email' => 'required|email|unique:professores,email,' .
$professor->id,
    ]);
    $professor->update($request->all());
    return redirect()->route('professores.index');
}

public function destroy(Professor $professor)
{
    $professor->delete();
    return redirect()->route('professores.index');
}
}

```

Faça o mesmo para `DisciplinaController`.

4. Crie as Views

Crie as views em `resources/views/professores` e `resources/views/disciplinas` para exibir formulários e listas.

Exemplo para `professores/index.blade.php`:

```

<!DOCTYPE html>
<html>
<head>
  <title>Professores</title>
</head>
<body>
  <h1>Professores</h1>
  <a href="{{ route('professores.create') }}">Adicionar Novo Professor</a>
  <table border="1">
    <thead>
      <tr>
        <th>ID</th>
        <th>Nome</th>
        <th>Email</th>
        <th>Ações</th>
      </tr>
    </thead>
    <tbody>
      @foreach ($professores as $professor)
        <tr>
          <td>{{ $professor->id }}</td>
          <td>{{ $professor->nome }}</td>
          <td>{{ $professor->email }}</td>
          <td>
            <a href="{{ route('professores.edit', $professor->id) }}">Editar</a>
            <form action="{{ route('professores.destroy', $professor->id) }}" method="POST" style="display:inline;">
              @csrf
              @method('DELETE')
              <button type="submit">Deletar</button>
            </form>
          </td>
        </tr>
      @endforeach
    </tbody>
  </table>
</body>
</html>

```

Código:

```

<!DOCTYPE html>
<html>
<head>
  <title>Professores</title>
</head>
<body>
  <h1>Professores</h1>
  <a href="{{ route('professores.create') }}">Adicionar Novo
Professor</a>
  <table border="1">
    <thead>
      <tr>
        <th>ID</th>
        <th>Nome</th>
        <th>Email</th>
        <th>Ações</th>
      </tr>
    </thead>
    <tbody>
      @foreach ($professores as $professor)

```



```

        <tr>
            <td>{{ $professor->id }}</td>
            <td>{{ $professor->nome }}</td>
            <td>{{ $professor->email }}</td>
            <td>
                <a href="{{ route('professores.edit',
$professor->id) }}">Editar</a>
                <form action="{{
route('professores.destroy', $professor->id) }}" method="POST"
style="display:inline;">
                    @csrf
                    @method('DELETE')
                    <button type="submit">Deletar</button>
                </form>
            </td>
        </tr>
    @endforeach
</tbody>
</table>
</body>

</html>

```