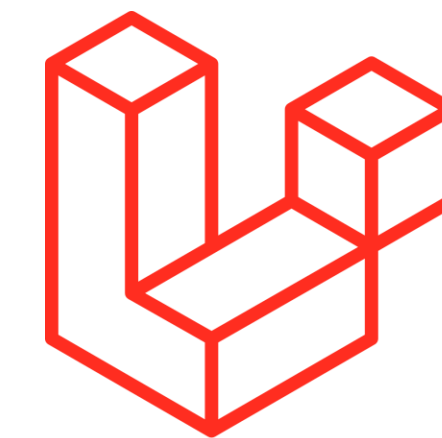


# INTRODUÇÃO AO LARAVEL

**PRIMEIRO PROJETO - CRUD (CREATE)**

PROF. ESP. JEFERSON ROBERTO DE LIMA - [JEFERSON.LIMA17@ETEC.SP.GOV.BR](mailto:JEFERSON.LIMA17@ETEC.SP.GOV.BR)

# INTRODUÇÃO AO LARAVEL



## CRIANDO NOSSO PROJETO LARAVEL

- Hoje vamos iniciar a criação de um projeto Laravel com banco de dados.
- Para iniciarmos um projeto laravel será necessário digitar o seguinte comando no cmd:
  - `composer create-project --prefer-dist laravel/laravel biblioteca_laravel`
- Após a realização desse comando daremos início a configuração do nosso banco de dados na aplicação.

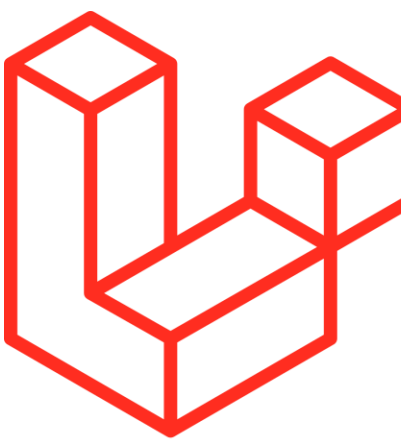


# INTRODUÇÃO AO LARAVEL

## CRIANDO UMA CONEXÃO COM O BANCO DE DADOS

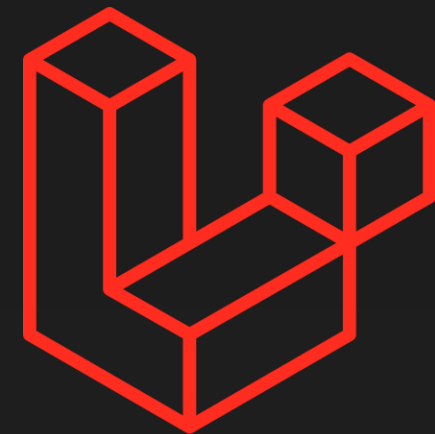
- Acessando o PHPMYADMIN no browser você criará um banco de dados com o nome “biblioteca\_laravel”, na sequência você deverá editar o arquivo .env com as informações do servidor.
  - DB\_CONNECTION=mysql  
DB\_HOST = 127.0.0.1  
DB\_PORT=3306  
DB\_DATABASE=biblioteca\_laravel  
DB\_USERNAME=root  
DB\_PASSWORD=

# INTRODUÇÃO AO LARAVEL



## TESTANDO A NOSSA CONEXÃO COM A BASE DE DADOS

- No terminal, digite o seguinte comando:
  - `php artisan tinker`
- No console do PSY SHELL digite o seguinte comando:
  - `DB::connection()->getPdo()`
- Você obterá a seguinte instrução na tela:



```
>>> DB::connection()->getPdo()
```

```
=> PDO {#3566
```

```
    inTransaction: false,
```

```
    attributes: {
```

```
        CASE: NATURAL,
```

```
        ERRMODE: EXCEPTION,
```

```
        AUTOCOMMIT: 1,
```

```
        PERSISTENT: false,
```

```
        DRIVER_NAME: "mysql",
```

```
        SERVER_INFO: "Uptime: 10429  Threads: 6  Questions: 1402  Slow queries: 0  Opens: 62  Flush tables: 1  Open tables: 39  Queries  
r second avg: 0.134",
```

```
        ORACLE_NULLS: NATURAL,
```

```
        CLIENT_VERSION: "mysqlnd 5.0.12-dev - 20150407 - $Id: 7cc7cc96e675f6d72e5cf0f267f48e167c2abb23 $",
```

```
        SERVER_VERSION: "5.5.5-10.4.21-MariaDB",
```

```
        STATEMENT_CLASS: [
```

```
            "PDOStatement",
```

```
        ],
```

```
        EMULATE_PREPARES: 0,
```

```
        CONNECTION_STATUS: "127.0.0.1 via TCP/IP",
```

```
        DEFAULT_FETCH_MODE: BOTH,
```

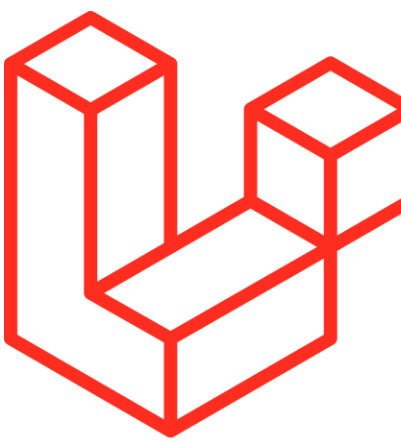
```
    },
```

```
}
```

```
>>> █
```

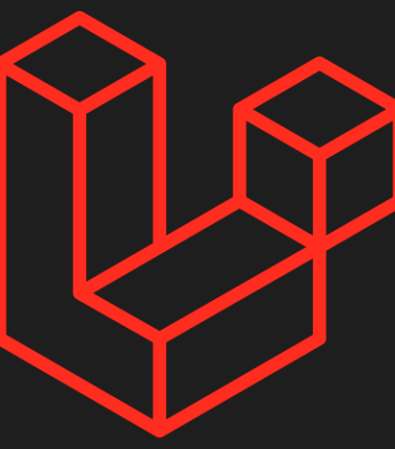


# INTRODUÇÃO AO LARAVEL



## CRIANDO NOSSA PRIMEIRA VIEW

- Para iniciarmos o nosso projeto vamos desenvolver uma interface simples com apenas HTML.
- Vamos criar um arquivo com o nome inicio.blade.php no seguinte repositório:
  - Projeto/resources/views/
- Nesse arquivo vamos criar um formulário com os seguintes campos:
  - Nome
  - Valor
  - Quantidade

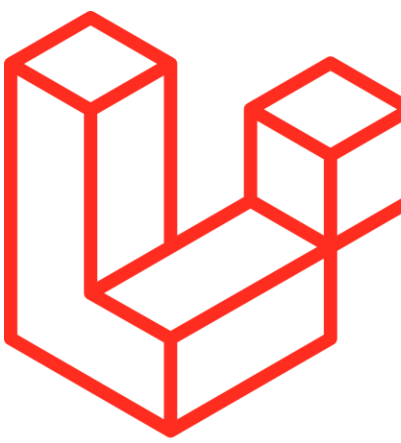


biblioteca\_laravel > resources > views > inicio.blade.php

```
1  <!DOCTYPE html>
2  <html lang="pt">
3      <head>
4          <meta charset="utf-8">
5          <meta name="viewport" content="width=device-width, initial-scale=1">
6          <title>Produtos</title>
7      </head>
8      <body>
9          <h1>Produtos</h1>
10
11          <form action="/cadastrar-produto" method="POST">
12              @csrf
13              <label for="lblNome">Nome:</label>
14              <input type="text" name="nome">
15              <br><br>
16              <label for="lblValor">Valor:</label>
17              <input type="text" name="valor">
18              <br><br>
19              <label for="lblQuantidade">Quantidade:</label>
20              <input type="text" name="estoque">
21              <br><br>
22              <button>Cadastrar</button>
23          </form>
24      </body>
25  </html>
26
```

# INTRODUÇÃO AO LARAVEL

## CONHECENDO O DIRETÓRIO MIGRATIONS



- Após criarmos a nossa tela de cadastro vamos iniciar o processo de criação de uma entidade no MYSQL/Maria DB utilizando apenas o Laravel.
- No projeto Laravel temos o seguinte diretório:
  - Projeto/database/migrations
- Nesse diretório contém exemplos de como podemos criar as nossas entidades utilizando apenas a programação.

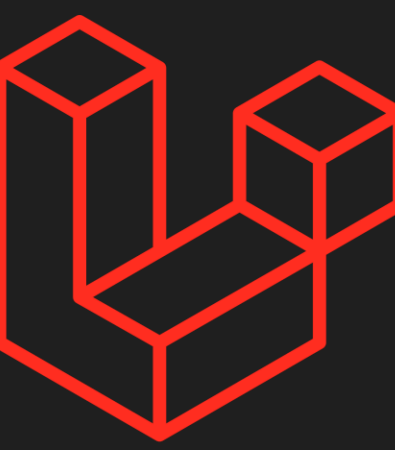




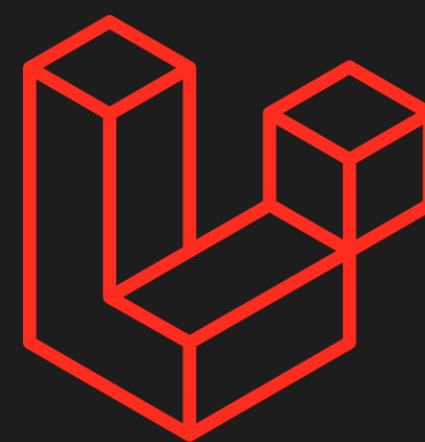
# INTRODUÇÃO AO LARAVEL

## CONHECENDO O DIRETÓRIO MIGRATIONS/MODELS

- No terminal do VSCode digite o seguinte comando:
  - `php artisan make:model Produto -m`
- Repare que ao realizar o comando acima você obteve um novo arquivo no diretório migrations e também no diretório projeto/app/models/ com o nome Produto.php
- Vamos editar os respectivos arquivos.



```
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreateProdutosTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('produtos', function (Blueprint $table) {
17             $table->id();
18             $table->string('nome');
19             $table->decimal('valor');
20             $table->integer('estoque');
21             $table->timestamps();
22         });
23     }
24
25     /**
26      * Reverse the migrations.
27      *
28      * @return void
29      */
30     public function down()
31     {
32         Schema::dropIfExists('produtos');
33     }
34 }
35
```

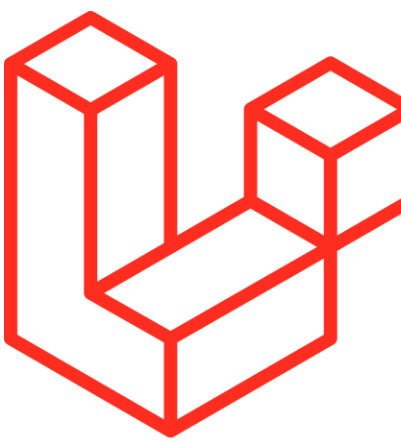


biblioteca\_laravel > app > Models > Produto.php

```
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Produto extends Model
9  {
10     use HasFactory;
11
12     protected $fillable = ['nome', 'valor', 'estoque'];
13 }
14
```



# INTRODUÇÃO AO LARAVEL









































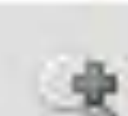
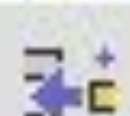


## TESTANDO O DIRETÓRIO MIGRATIONS

- Para realizarmos os testes do repositório migrations e a criação da entidade produto, rode o seguinte comando no terminal do VSCODE:
  - `php artisan migrate`
- A partir de agora você pode visualizar as alterações realizadas no seu banco de dados.



Filtros

Contendo a palavra:

	Tabela	Acções	Registos	Tipos
<input type="checkbox"/>	failed_jobs	  Procurar  Estrutura  Pesquisar  Inserir  Limpar  Eliminar	0	Inno
<input type="checkbox"/>	migrations	  Procurar  Estrutura  Pesquisar  Inserir  Limpar  Eliminar	5	Inno
<input type="checkbox"/>	password_resets	  Procurar  Estrutura  Pesquisar  Inserir  Limpar  Eliminar	0	Inno
<input type="checkbox"/>	personal_access_tokens	  Procurar  Estrutura  Pesquisar  Inserir  Limpar  Eliminar	0	Inno
<input type="checkbox"/>	produtos	  Procurar  Estrutura  Pesquisar  Inserir  Limpar  Eliminar	0	Inno
<input type="checkbox"/>	users	  Procurar  Estrutura  Pesquisar  Inserir  Limpar  Eliminar	0	Inno
6 tabelas		Soma	5	Inno



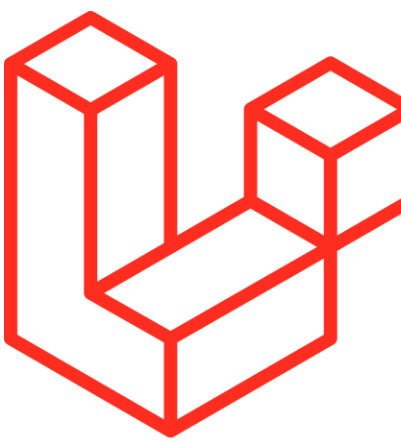
☐ Marcar todos

Com os seleccionados:





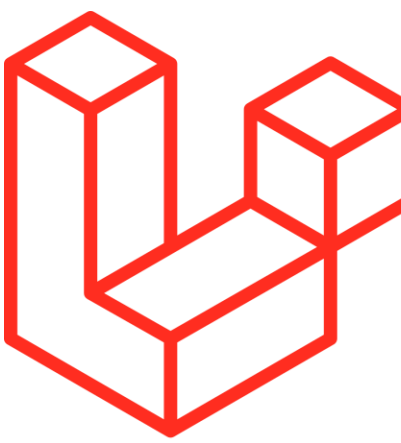
# INTRODUÇÃO AO LARAVEL



## TESTANDO O DIRETÓRIO MIGRATIONS

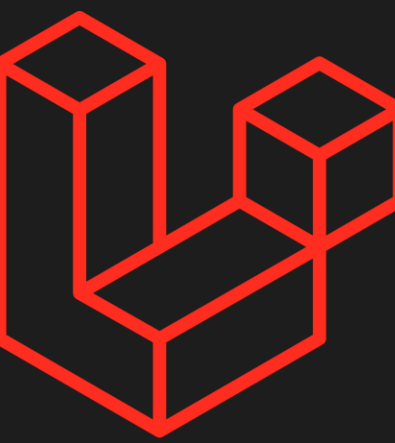
- A partir de agora você criou uma série de entidades em seu respectivo banco de dados.
- Podemos entender que na próxima vez que criarmos um projeto no Laravel podemos deletar os outros arquivos no diretório migrations e deixar apenas o arquivo do nosso interesse.
- Assim será criado apenas a entidade desenvolvida durante a construção do projeto.

# INTRODUÇÃO AO LARAVEL



## DIRECIONAMENTO DAS ROTAS

- Com os seguintes arquivos criados:
  - Projeto/resources/views/inicio.blade.php
  - Projeto/app/models/Produto.php
  - Projeto/database/migrations/2022\_04\_10\_1152238\_create\_produtos\_table.php
- Podemos criar a rota /cadastrar-produto que esta inserida no arquivo inicio.blade.php, utilizando a seguinte programação:



biblioteca\_laravel > routes > web.php

```
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  /*
6  |-----
7  | Web Routes
8  |-----
9  |
10 | Here is where you can register web routes for your application. These
11 | routes are loaded by the RouteServiceProvider within a group which
12 | contains the "web" middleware group. Now create something great!
13 |
14 */
15
16 use App\Models\Produto;
17 use Illuminate\Http\Request;
18
19 Route::get('/', function () {
20     return view('inicio');
21 });
22
23 Route::post('/cadastrar-produto', function(Request $request){
24     //dd($request->all());
25
26     Produto::create([
27         'nome' => $request->nome,
28         'valor' => $request->valor,
29         'estoque' => $request->estoque
30     ]);
31
32     echo "Produto criado com sucesso!";
33 });
34
```



# INTRODUÇÃO AO LARAVEL

## TESTANDO A APLICAÇÃO



- Após finalizar a rota do nosso formulário de cadastro, podemos realizar os testes na aplicação aplicando o seguinte comando o terminal do VSCODE:
  - `php artisan serve`
- Em seguida realizaremos os testes de acordo com a sequência de telas.



PROBLEMAS

SAÍDA

CONSOLE DE DEPURAÇÃO

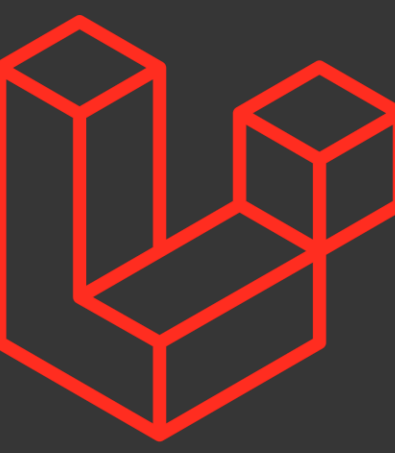
TERMINAL

Windows PowerShell

Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell <https://aka.ms/pscore6>

PS C:\xampp\htdocs\biblioteca\_laravel> **php** artisan serve



← → ↻ ⓘ 127.0.0.1:8000

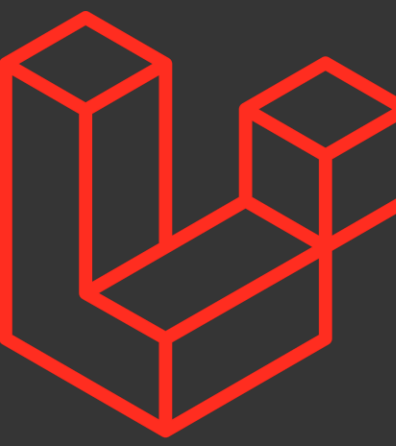
# Produtos

Nome:

Valor:

Quantidade:

Cadastrar



127.0.0.1:8000/cadastrar-produto

Produto criado com sucesso!



 Procurar
  Estrutura
  SQL
  Pesquisar
  Insere
  Exportar





✓ A mostrar registos de 0 - 0 (1 total, A consulta demorou 0,0012 segundos.)

SELECT \* FROM `produtos`

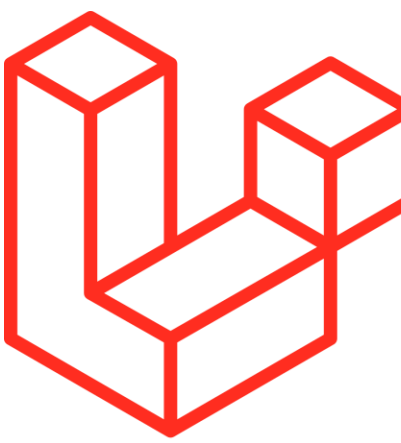
☐ Perfil [
 [Editar em linha](#)] [
 [Edita](#)] [
 [Explicar SQL](#)] [
 [Criar código PHP](#)] [
 [Actualizar](#)]

☐ Mostrar tudo |
 Número de registos: 25 ▼
 Filtrar registos:

+ Opções

		▼		id	nome	valor	estoque	created
<input type="checkbox"/>	 Editar	 Copiar	 Apagar	3	Programando com Kotlin	68.50	40	2022-04

# INTRODUÇÃO AO LARAVEL



## FUNÇÃO CREATE FINALIZADA

- Depois de realizarmos os testes é possível verificar o funcionamento da nossa aplicação web em conjunto com o nosso banco de dados MYSQL/Maria DB.