



Introdução ao Laravel

Tomatcher: Gustavo Lemos



A stack abordada:

Windows

Postgresql

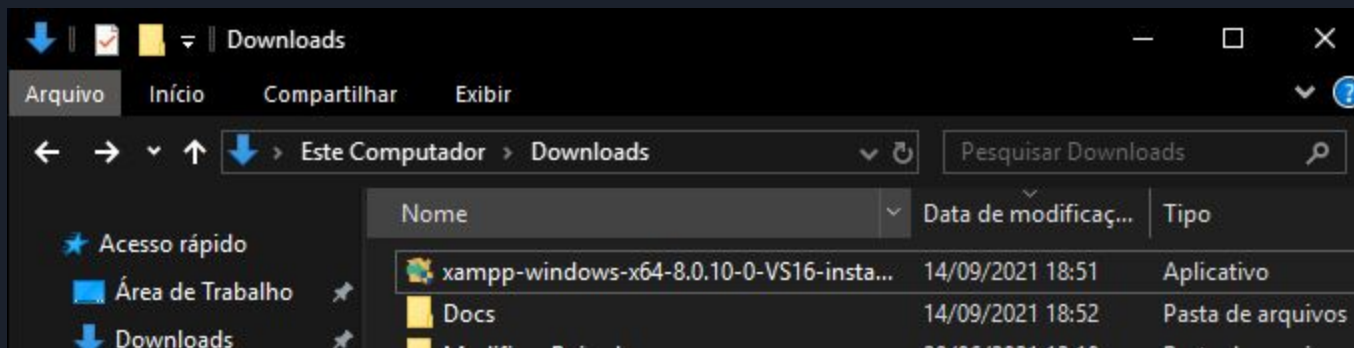
PHP 8 - (XAMPP + Composer)

VSCode

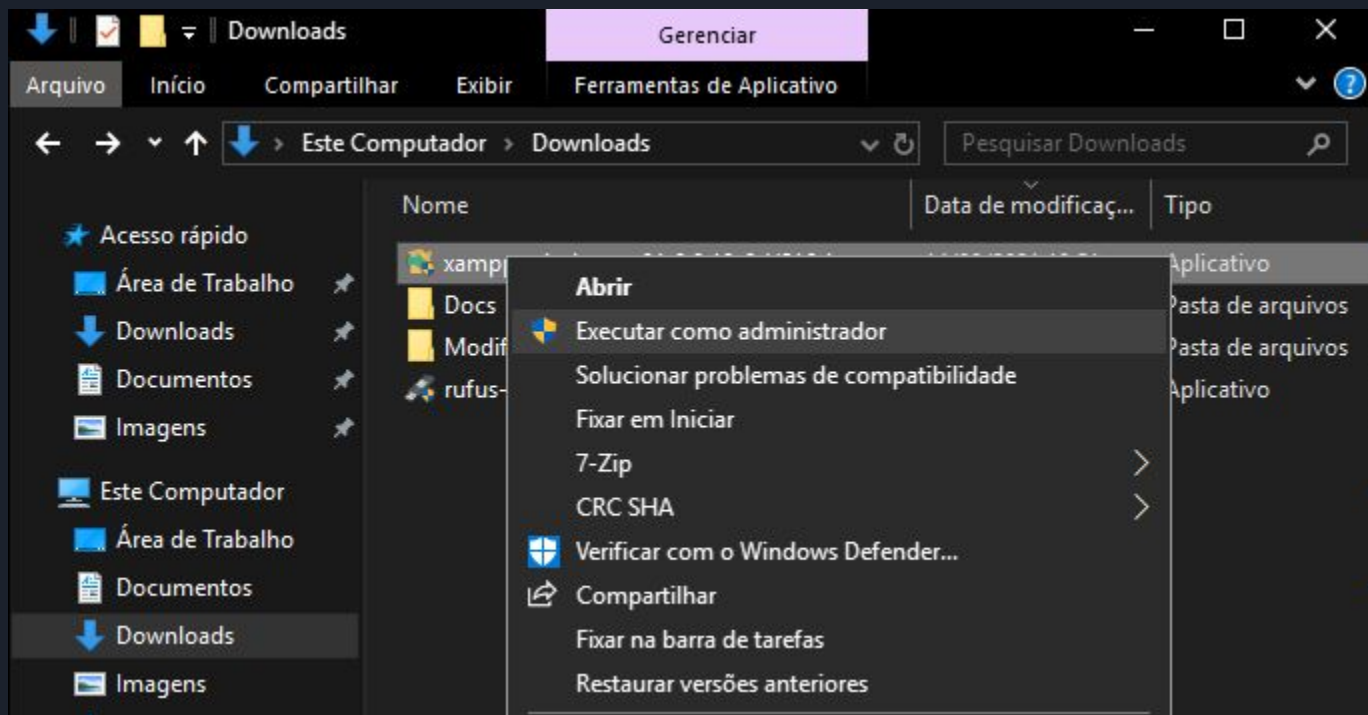
Preparando o ambiente



XAMPP - Instalação



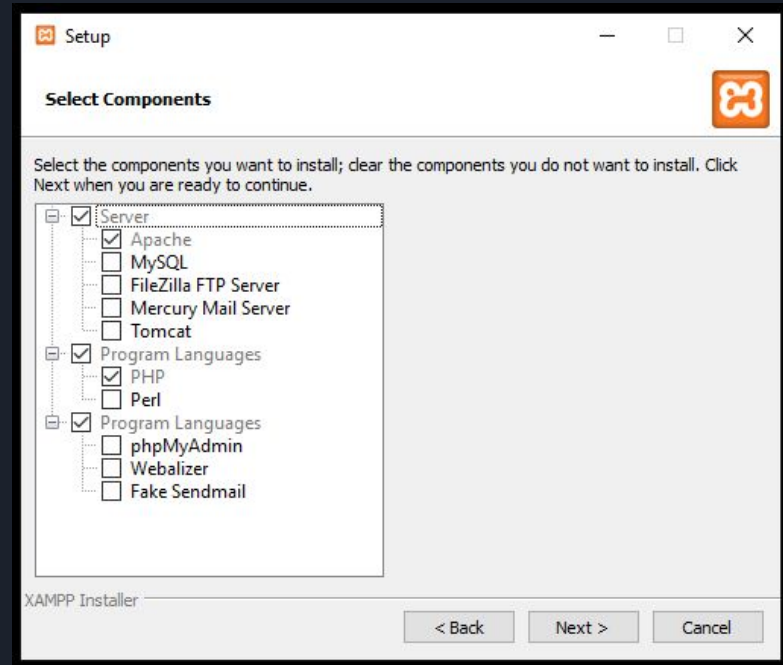
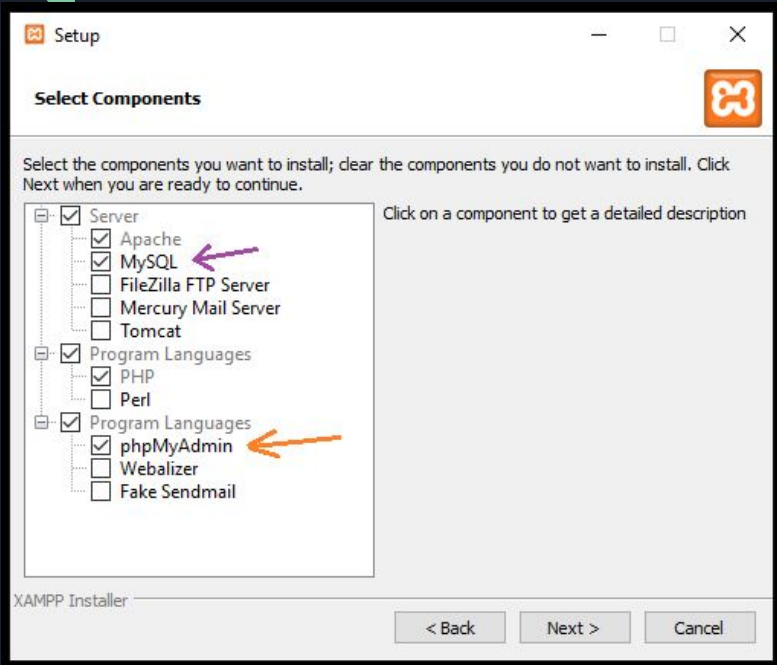
XAMPP - Instalação



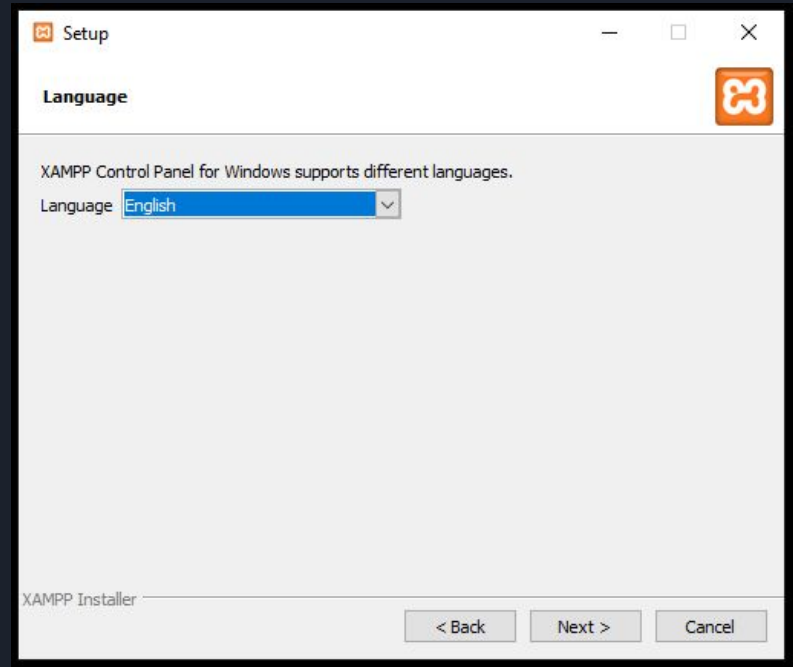
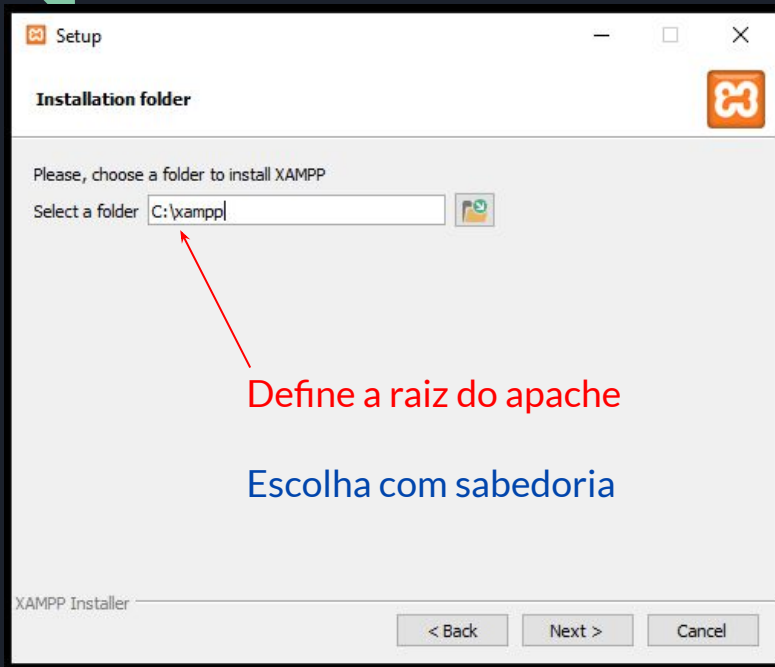
XAMPP - Instalação



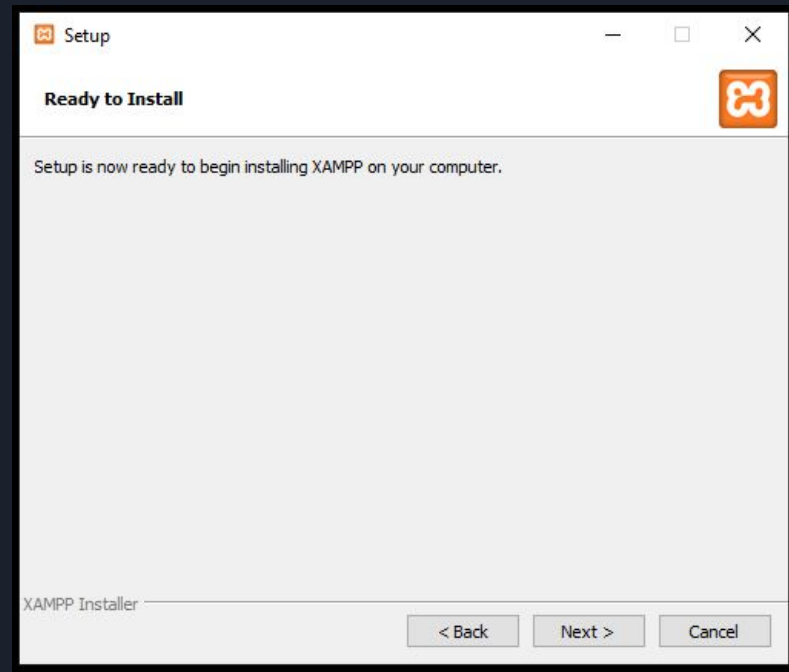
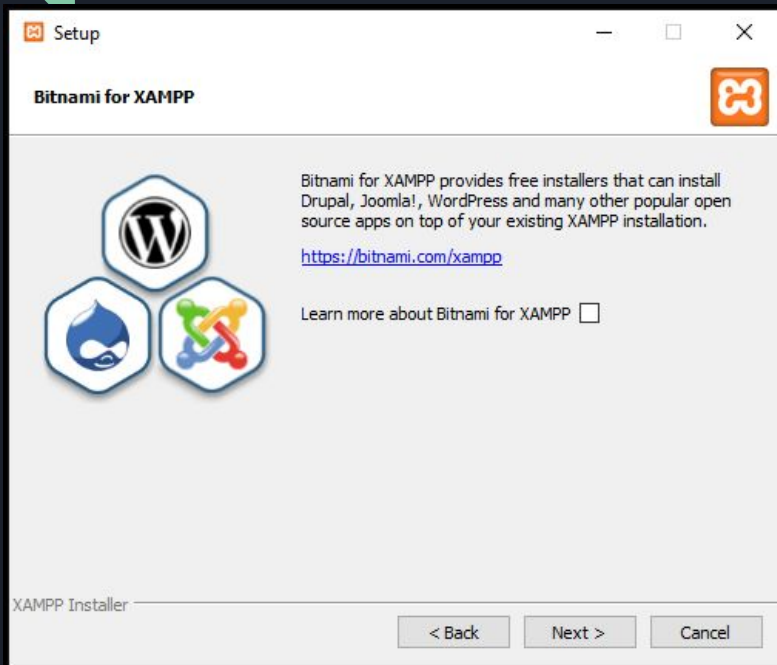
XAMPP - Instalação (pt2)



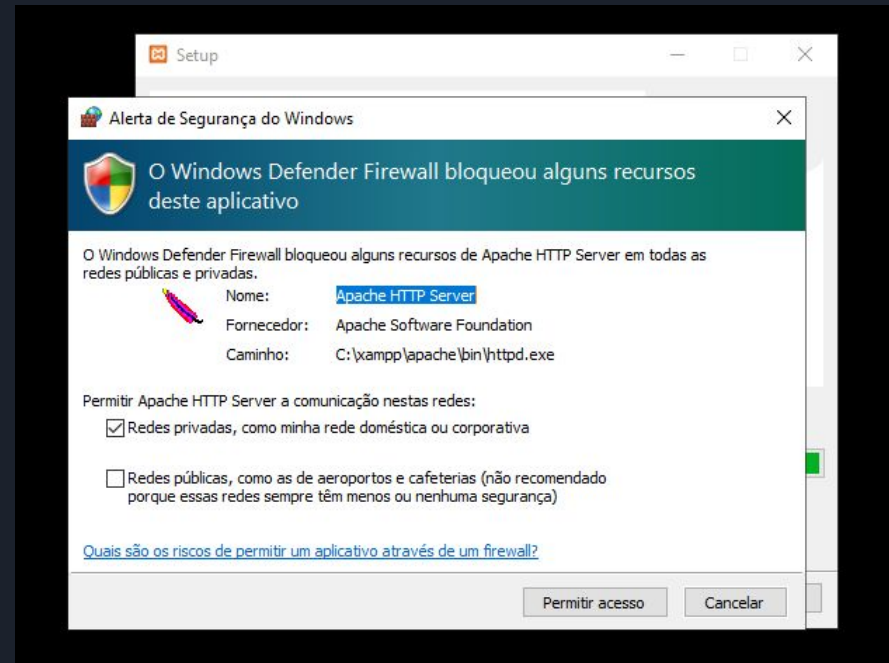
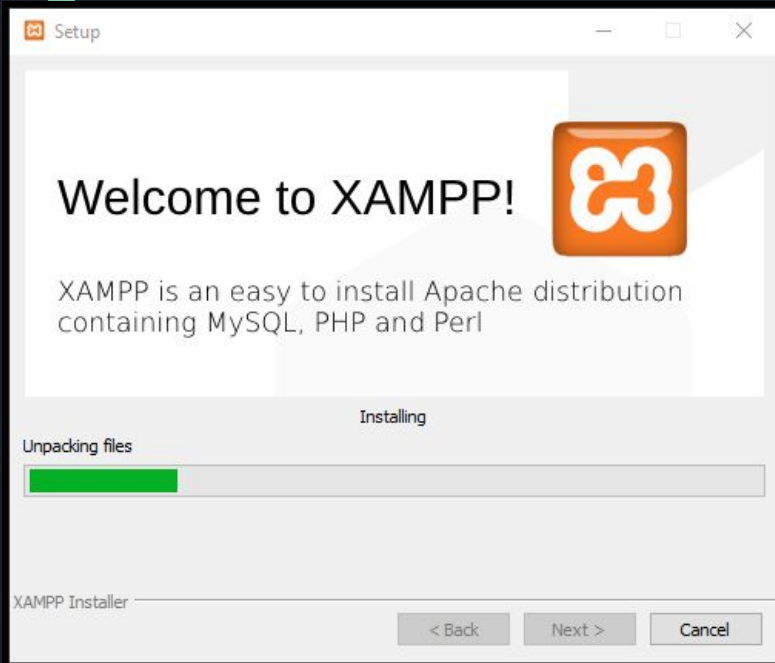
XAMPP - Instalação (pt3)



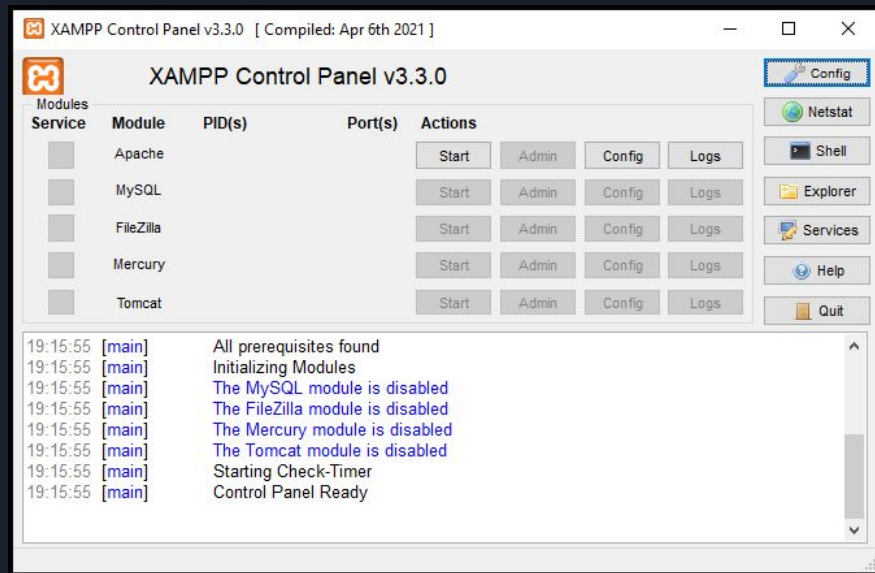
XAMPP - Instalação (pt4)



XAMPP - Instalação (pt5)



XAMPP - Instalação (finalmente!!!) \o/\o/\o/



Instalação do composer

https://getcomposer.org/download/



[Home](#) | [Getting Started](#) | [Download](#) | [Documentation](#) | [Browse Packages](#)

Download Composer Latest: v2.1.7

Windows Installer

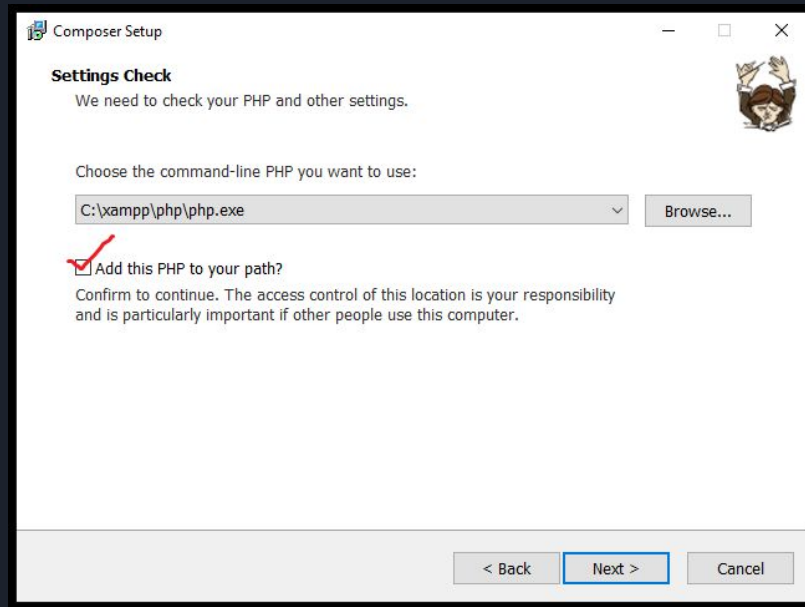
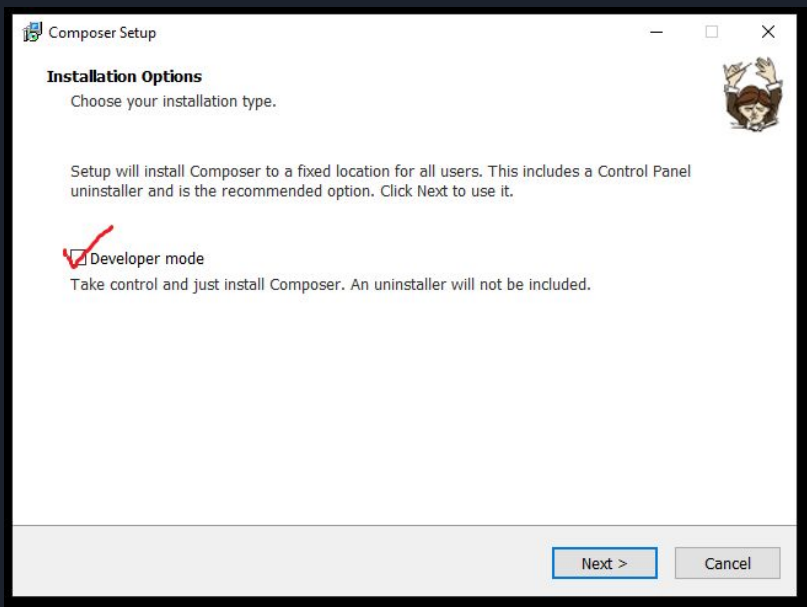
The installer - which requires that you have PHP already installed - will download Composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.

Command-line installation



Instalação do composer





O laravel exige os seguintes módulos:

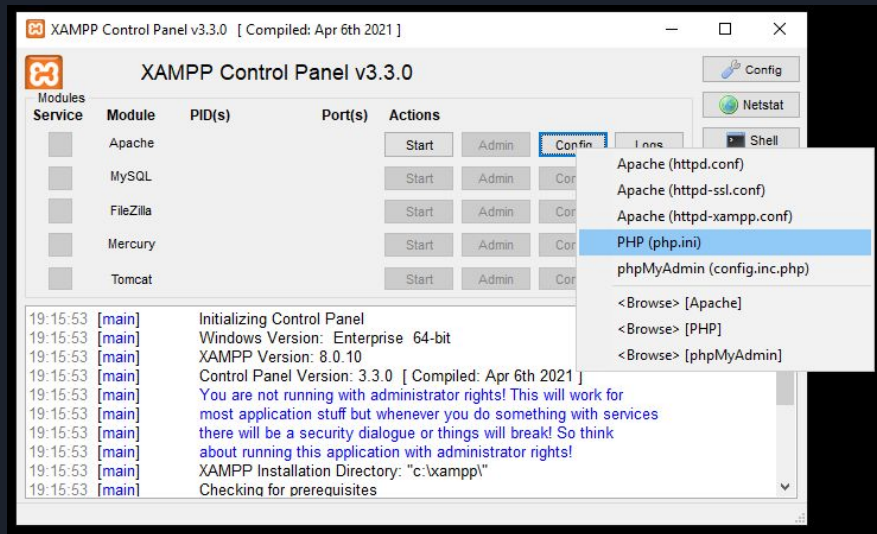
- BCMath PHP Extension -> funções matemáticas
- CType PHP Extension -> manipulação de caracteres
- Fileinfo PHP Extension
- JSON PHP Extension
- Mbstring PHP Extension -> manipulação de strings
- OpenSSL PHP Extension -> funções de criptografia
- PDO PHP Extension -> conexão com o banco de dados
- Tokenizer PHP Extension -> interface para modificar o php
- XML PHP Extension

Obs: Muitas extensões já estão habilitadas no PHP 8 (os casos podem variar)

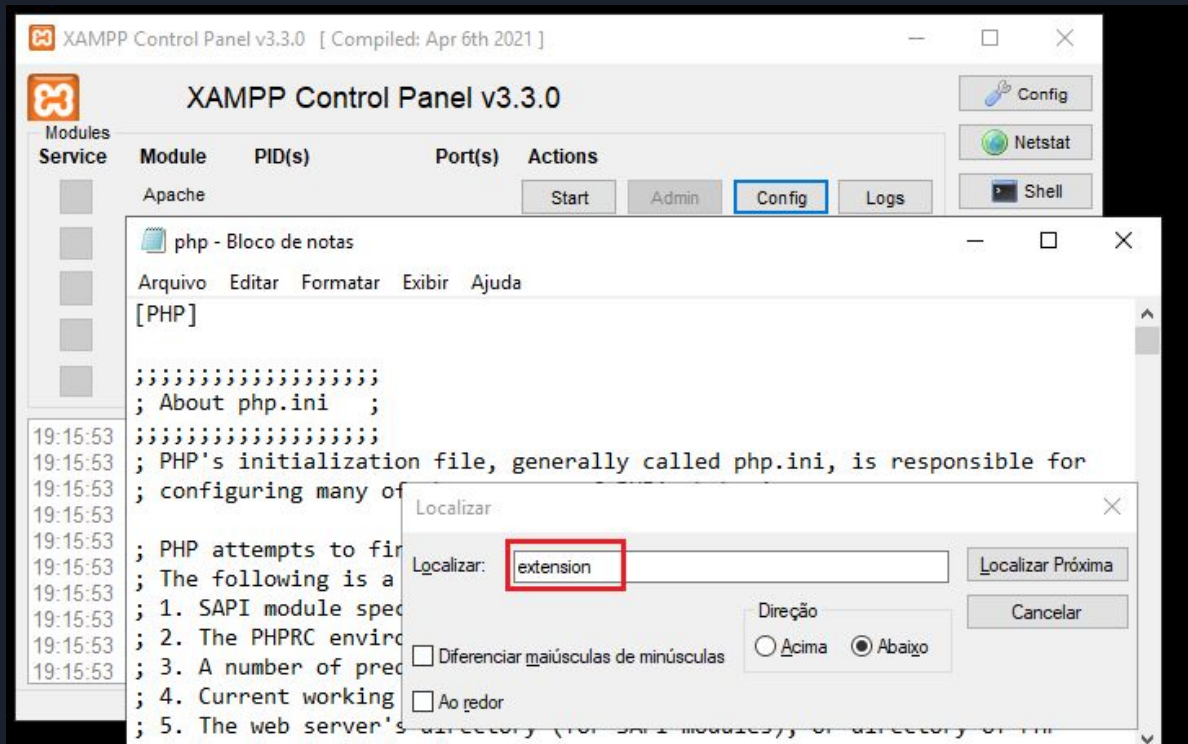
Habilitando os módulos

Arquivo de configuração do PHP (php.ini)

Painel XAMPP -> Apache -> Botão “Cofig” -> Opção php.in



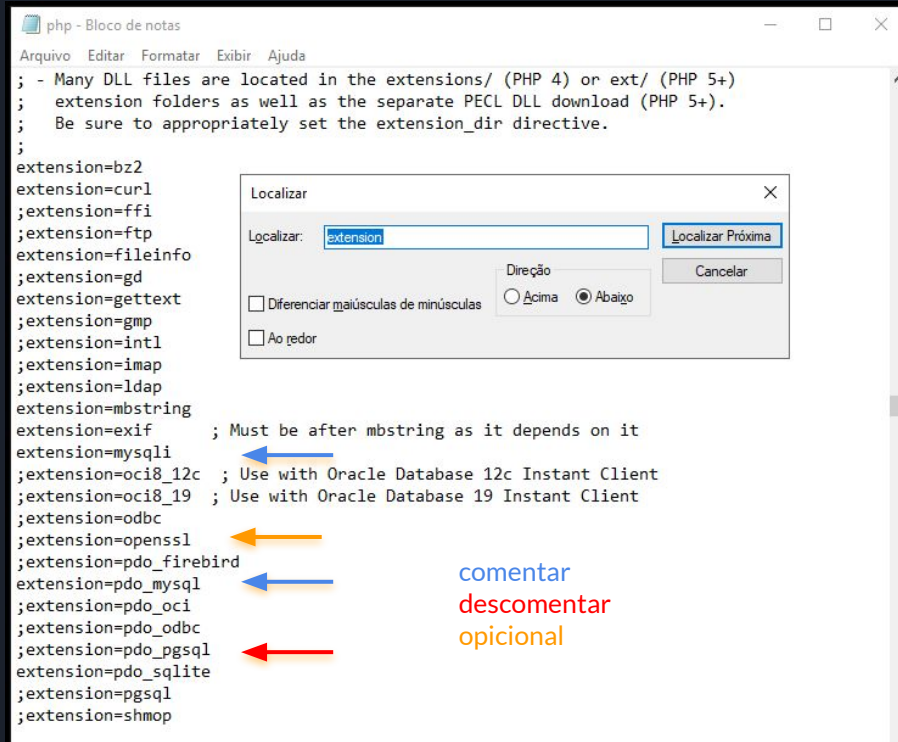
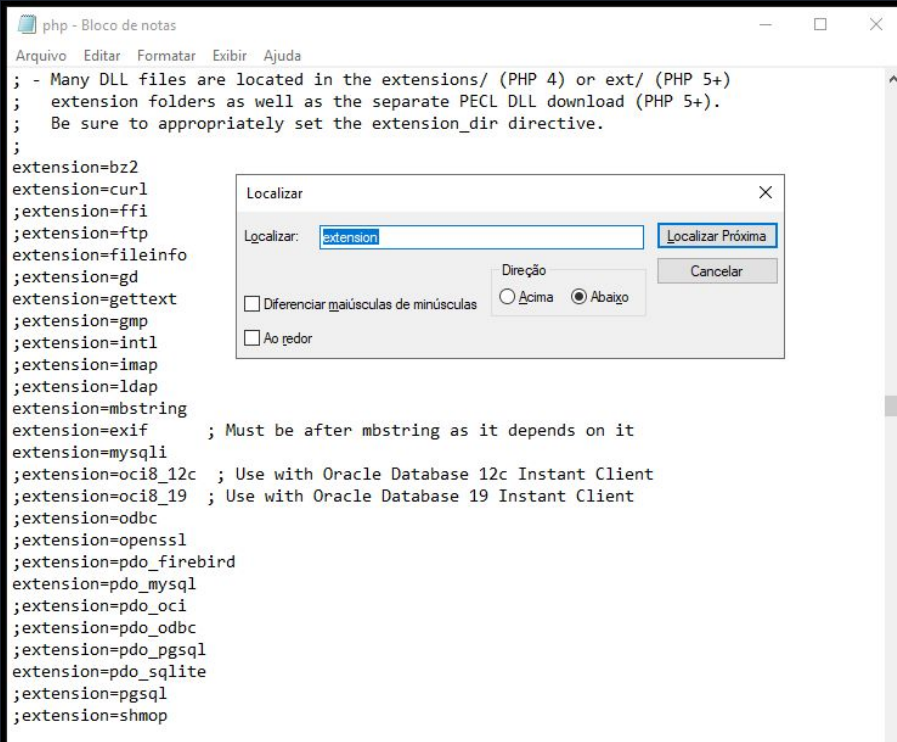
Edição de módulos (php.ini)



Com o php.ini aberto,
pressione Ctrl + f
e busque por:

extension

Edição de módulos (php.ini)



Start no Apache

XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	MySQL			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	FileZilla			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	Mercury			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	Tomcat			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>

19:15:55 [main] All prerequisites found
19:15:55 [main] Initializing Modules
19:15:55 [main] The MySQL module is disabled
19:15:55 [main] The FileZilla module is disabled
19:15:55 [main] The Mercury module is disabled
19:15:55 [main] The Tomcat module is disabled
19:15:55 [main] Starting Check-Timer
19:15:55 [main] Control Panel Ready

Config Netstat Shell Explorer Services Help Quit

XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	8720 8164	80, 443	<input type="button" value="Stop"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	MySQL			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	FileZilla			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	Mercury			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	Tomcat			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>

19:15:55 [main] The MySQL module is disabled
19:15:55 [main] The FileZilla module is disabled
19:15:55 [main] The Mercury module is disabled
19:15:55 [main] The Tomcat module is disabled
19:15:55 [main] Starting Check-Timer
19:15:55 [main] Control Panel Ready

Config Netstat Shell Explorer Services Help Quit

Instalação do composer

https://getcomposer.org/download/



[Home](#) | [Getting Started](#) | [Download](#) | [Documentation](#) | [Browse Packages](#)

Download Composer Latest: v2.1.7

Windows Installer

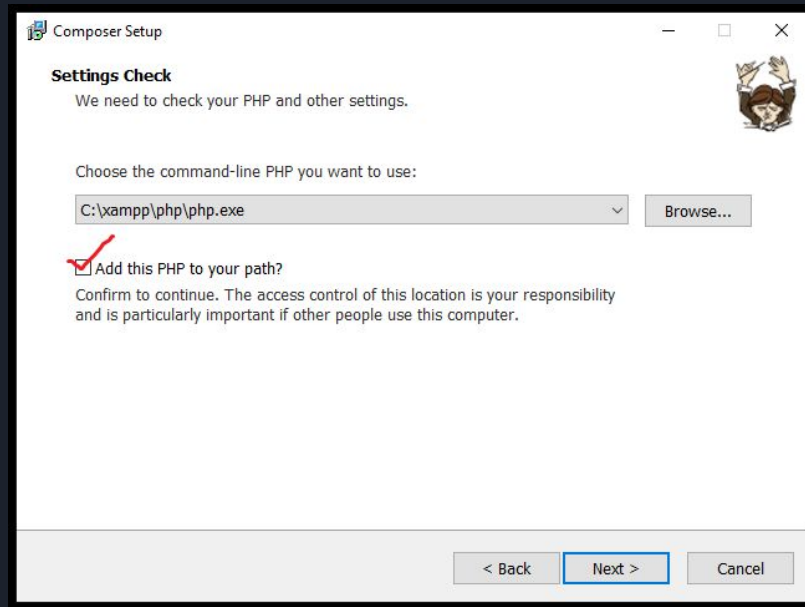
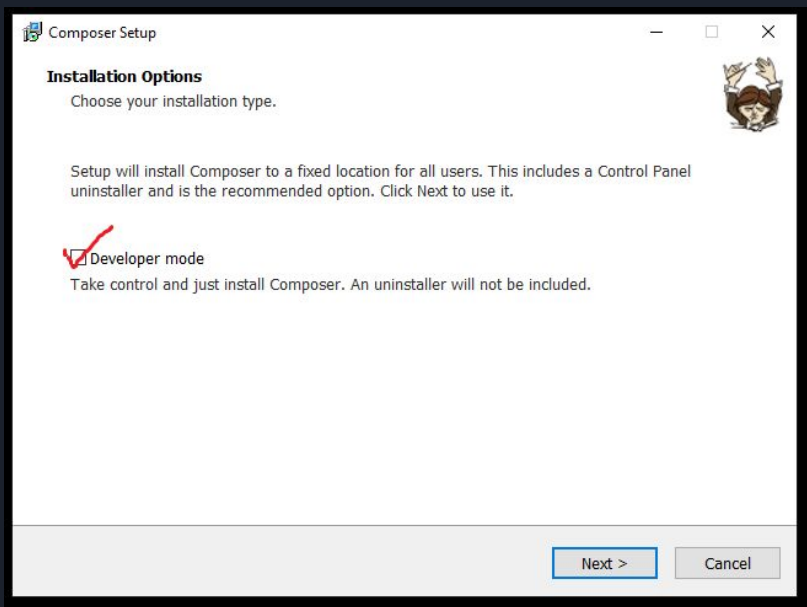
The installer - which requires that you have PHP already installed - will download Composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.

Command-line installation

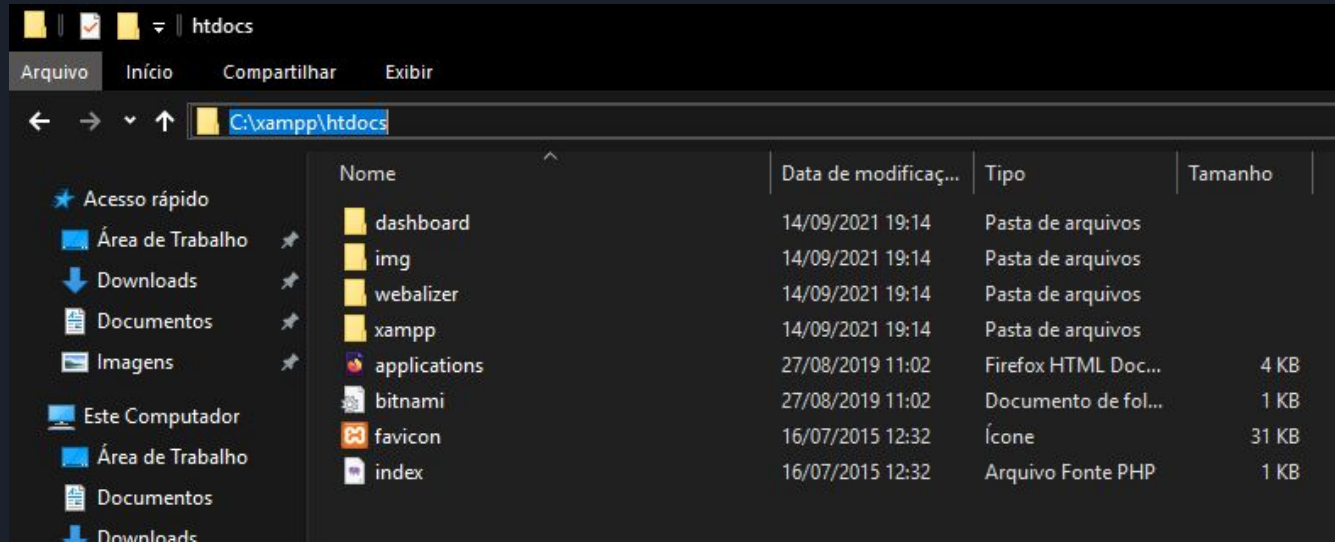


Instalação do composer



Pasta raiz dos projetos no servidor Apache

Todos os projetos do php devem ser armazenados nessa pasta
Obs: o caminho da pasta depende de onde se escolher instalar o XAMPP



Dica: crie um atalho da raiz dos projetos para um local mais conveniente

Criando o database

```
SQL Shell (psql)

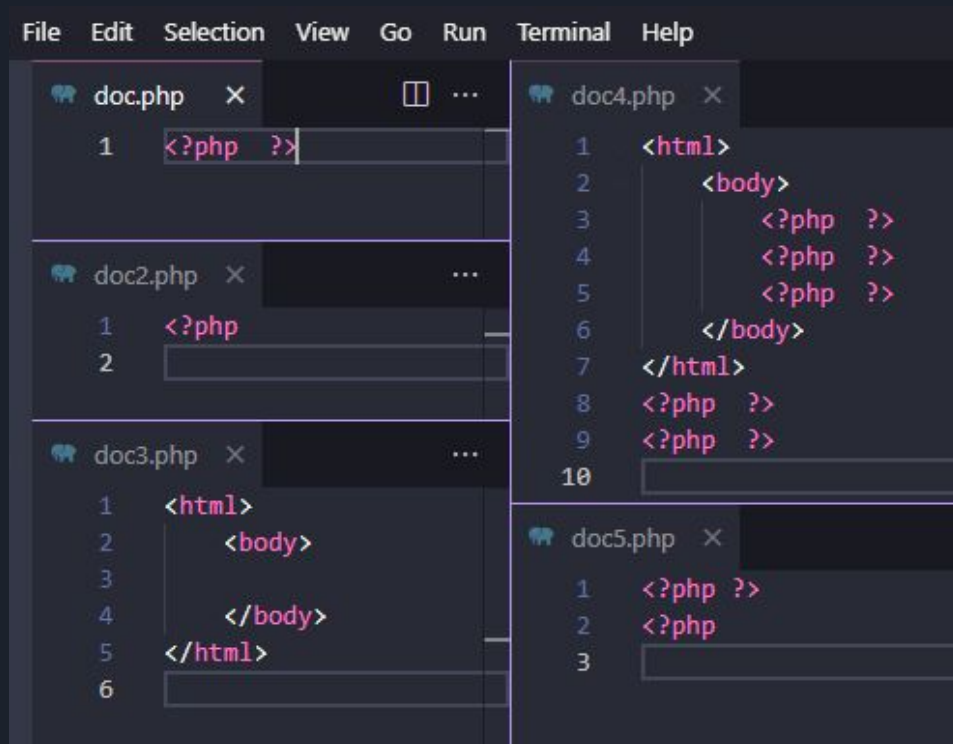
postgres=# \l
fiscaliza_db | postgres | UTF8 | Portuguese_Brazil.1252 | Portuguese_Brazil.1252 |
postgres     | postgres | UTF8 | Portuguese_Brazil.1252 | Portuguese_Brazil.1252 |
template0    | postgres | UTF8 | Portuguese_Brazil.1252 | Portuguese_Brazil.1252 | =c/postgres +
template1    | postgres | UTF8 | Portuguese_Brazil.1252 | Portuguese_Brazil.1252 | postgres=CTc/postgres +
teste_db     | postgres | UTF8 | Portuguese_Brazil.1252 | Portuguese_Brazil.1252 |
users        | postgres | UTF8 | Portuguese_Brazil.1252 | Portuguese_Brazil.1252 |
```

```
postgres=# create database pokemon;
CREATE DATABASE
postgres=# \l
fiscaliza_db | postgres | UTF8 | Portuguese_Brazil.1252 | Portuguese_Brazil.1252 |
pokemon      | postgres | UTF8 | Portuguese_Brazil.1252 | Portuguese_Brazil.1252 |
postgres     | postgres | UTF8 | Portuguese_Brazil.1252 | Portuguese_Brazil.1252 |
template0    | postgres | UTF8 | Portuguese_Brazil.1252 | Portuguese_Brazil.1252 | =c/postgres +
template1    | postgres | UTF8 | Portuguese_Brazil.1252 | Portuguese_Brazil.1252 | postgres=CTc/postgres +
teste_db     | postgres | UTF8 | Portuguese_Brazil.1252 | Portuguese_Brazil.1252 |
postgres     | postgres | UTF8 | Portuguese_Brazil.1252 | Portuguese_Brazil.1252 | =c/postgres +
postgres     | postgres | UTF8 | Portuguese_Brazil.1252 | Portuguese_Brazil.1252 | postgres=CTc/postgres +
```

Mini-tour no mundo do PHP



Estrutura básica de um arquivo php



The screenshot shows a code editor with five open files: doc.php, doc2.php, doc3.php, doc4.php, and doc5.php. The editor has a menu bar with File, Edit, Selection, View, Go, Run, Terminal, and Help. The files contain the following code:

```
doc.php
1 <?php ?>

doc2.php
1 <?php
2

doc3.php
1 <html>
2     <body>
3
4     </body>
5 </html>
6

doc4.php
1 <html>
2     <body>
3         <?php ?>
4         <?php ?>
5         <?php ?>
6     </body>
7 </html>
8 <?php ?>
9 <?php ?>
10

doc5.php
1 <?php ?>
2 <?php
3
```


As variáveis do php são gananciosas

doc.php X

```
1  <?php
2
3  $x = 'cachorro';
4  $$x = 'gato';           // equivale a $cachorro = 'gato';
5  $$$x = 'rato';          // equivale a $gato = 'rato'; ou $$cachorro = 'rato';
6
7  echo $cachorro . "\n";
8  echo $gato . ' \n ';
9  echo $$cachorro . "\n";
10
```

DEBUG CONSOLE

OUTPUT

PROBLEMS

TERMINAL

Souza@DESKTOP-N3VUF0I MINGW64 /c:/xampp/htdocs

\$ php doc.php

gato

rato \n rato



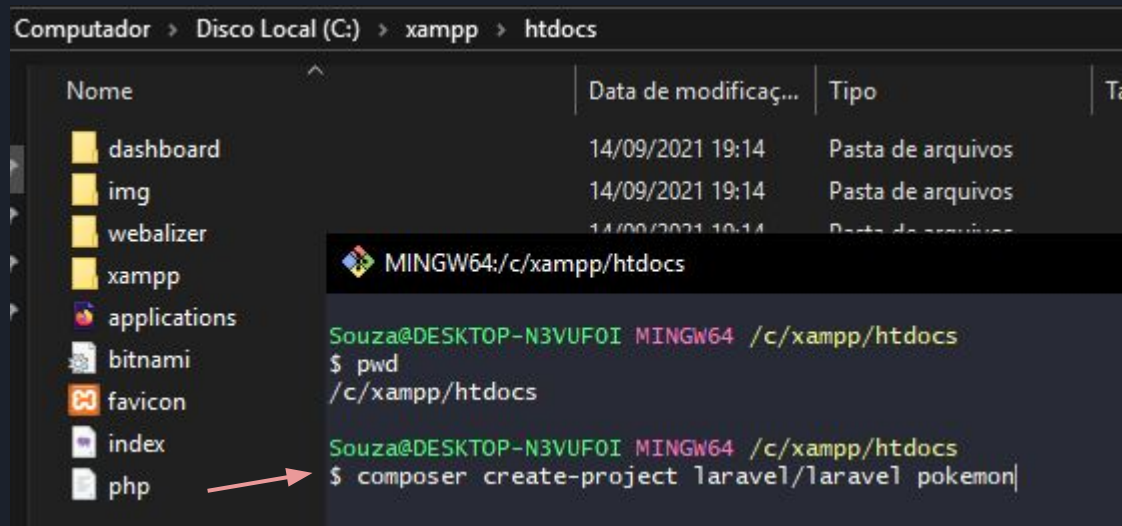
Constantes mágicas

<code>__LINE__</code>	-> retorna o número da linha que foi utilizada
<code>__FILE__</code>	-> caminho absoluto do arquivo que foi utilizada
<code>__DIR__</code>	-> caminho absoluto da pasta
<code>__FUNCTION__</code>	-> nome da função quando está dentro de uma
<code>__CLASS__</code>	-> nome da classe quando está dentro de uma
<code>__TRAIT__</code>	-> nome da trait quando está dentro de uma
<code>__METHOD__</code>	-> nome da método quando está dentro de um
<code>__NAMESPACE__</code>	-> nome do namespace o qual se insere

Instalando o Laravel

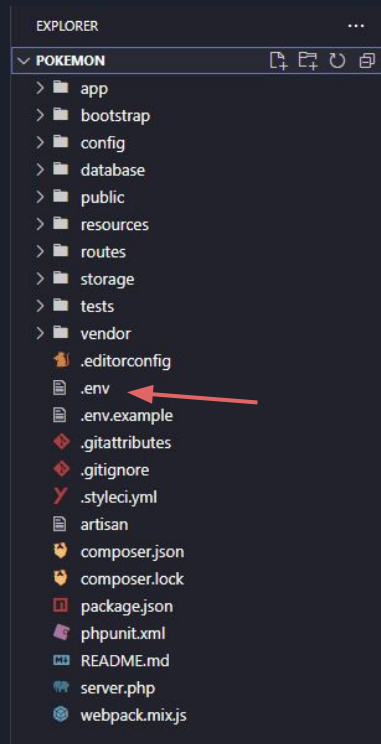


Criação do projeto



Abrindo o projeto

```
Souza@DESKTOP-N3VUF0I MINGW64 /c/xampp/htdocs  
$ cd pokemon/  
  
Souza@DESKTOP-N3VUF0I MINGW64 /c/xampp/htdocs/pokemon  
$ codium .|
```



Configurando o DB no projeto

ANTES

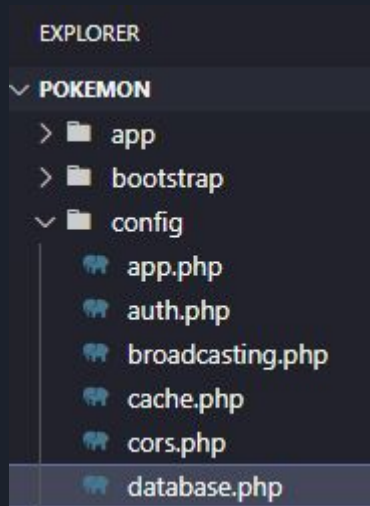
```
.env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:pQXrbYkFt/C0rkPdcMgXsb0/UbG4FSrEp5/WmgeFU8=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_LEVEL=debug
9
10 DB_CONNECTION=mysql
11 DB_HOST=127.0.0.1
12 DB_PORT=3306
13 DB_DATABASE=laravel
14 DB_USERNAME=root
15 DB_PASSWORD=
```

DEPOIS

```
.env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:pQXrbYkFt/C0rkPdcMgXsb0/UbG4FSrEp5/WmgeFU8=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_LEVEL=debug
9
10 DB_CONNECTION=pgsql
11 DB_HOST=127.0.0.1
12 DB_PORT=5432
13 DB_DATABASE=pokemon
14 DB_USERNAME=postgres
15 DB_PASSWORD=1234
```

Configurando o driver padrão de conexão com o DB

ANTES



```
database.php x
2
3 use Illuminate\Support\Str;
4
5 return [
6
7     /*
8      |-----
9      | Default Database Connection Name
10     |-----
11
12     | Here you may specify which of the database
13     | to use as your default connection for all d
14     | you may use many connections at once using
15     |
16     */
17
18     'default' => env('DB_CONNECTION', 'mysql'),
19
```

DEPOIS

```
database.php x .env
2
3 use Illuminate\Support\Str;
4
5 return [
6
7     /*
8      |-----
9      | Default Database Connection Name
10     |-----
11
12     | Here you may specify which of the database
13     | to use as your default connection for all d
14     | you may use many connections at once using
15     |
16     */
17
18     'default' => env('DB_CONNECTION', 'pgsql'),
19
```

Verificando o script de Construção da tabela Users



Obs:

Qualquer alteração, adição ou exclusão de colunas também deverá ser explicita em app/Models/Users.php

```
2014_10_12_000000_create_users_table.php x
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreateUsersTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('users', function (Blueprint $table) {
17             $table->id();
18             $table->string('name');
19             $table->string('email')->unique();
20             $table->timestamp('email_verified_at')->nullable();
21             $table->string('password');
22             $table->rememberToken();
23             $table->timestamps();
24         });
25     }
26 }
```



Configurando Serviço de Autenticação (Passport)





Acrescentando o Passport como dependência do projeto

```
DEBUG CONSOLE  OUTPUT  PROBLEMS  TERMINAL  
  
Souza@DESKTOP-N3VUF0I MINGW64 /c/xampp/htdocs/pokemon  
$ composer require laravel/passport
```



Gerando as chaves de Criptografia dos tokens

```
Souza@DESKTOP-N3VUF0I MINGW64 /c/xampp/htdocs/pokemon
$ php artisan passport:install
Personal access client created successfully.
Client ID: 1
Client secret: JSPUamB16V4cotn2sllaBvXGay8x3C7GnvcZTg5G
Password grant client created successfully.
Client ID: 2
Client secret: n4KHPw1twTbXYvLTMXqytsqeIhLiu814MtC67AXO
```

Ajustar Passport como provedor de serviços

```
app.php x
126  /*
127  |-----
128  | Autoloaded Service Providers
129  |-----
130  |
131  | The service providers listed here will be automatically loaded on the
132  | request to your application. Feel free to add your own services to
133  | this array to grant expanded functionality to your applications.
134  |
135  */
136
137  'providers' => [
138
139      /*
140      * Laravel Framework Service Providers...
141      */
142      Illuminate\Auth\AuthServiceProvider::class,
143      Illuminate\Broadcasting\BroadcastServiceProvider::class,
144      Illuminate\Bus\BusServiceProvider::class,
145      Illuminate\Cache\CacheServiceProvider::class,
146      Illuminate\Foundation\Providers\ConsoleSupportServiceProvider::class,
```

POKEMON

- > app
- > bootstrap
- > config
- app.php

```
127  |-----
128  | Autoloaded Service Providers
129  |-----
130  |
131  | The service providers listed here will be automatically loaded on the
132  | request to your application. Feel free to add your own services to
133  | this array to grant expanded functionality to your applications.
134  |
135  */
136
137  'providers' => [
138
139      /*
140      * Laravel Framework Service Providers...
141      */
142      Laravel\Passport\PassportServiceProvider::class,
143      Illuminate\Auth\AuthServiceProvider::class,
144      Illuminate\Broadcasting\BroadcastServiceProvider::class,
145      Illuminate\Bus\BusServiceProvider::class,
146      Illuminate\Cache\CacheServiceProvider::class,
147      Illuminate\Foundation\Providers\ConsoleSupportServiceProvider::class,
```

Configurando o driver do Passport

```
22 | -----
23 | Authentication Guards
24 | -----
25 |
26 | Next, you may define every authentication guard for your application.
27 | Of course, a great default configuration has been defined for you
28 | here which uses session storage and the Eloquent user provider.
29 |
30 | All authentication drivers have a user provider. This defines how the
31 | users are actually retrieved out of your database or other storage
32 | mechanisms used by this application to persist your user's data.
33 |
34 | Supported: "session"
35 |
36 | */
37 |
38 | 'guards' => [
39 |     'web' => [
40 |         'driver' => 'session',
41 |         'provider' => 'users',
42 |     ],
43 | ],
44 |
```

EXPLORER

POKEMON

app

bootstrap

config

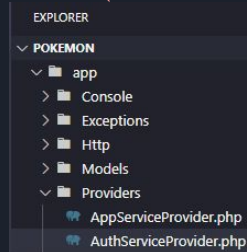
app.php

auth.php

```
35 |
36 | */
37 |
38 | 'guards' => [
39 |     'web' => [
40 |         'driver' => 'session',
41 |         'provider' => 'users',
42 |     ],
43 |     'api' => [
44 |         'driver' => 'passport',
45 |         'provider' => 'users',
46 |     ],
47 | ],
```

Configurar provedor de serviço de autenticação

```
AuthServiceProvider.php X
1 <?php
2
3 namespace App\Providers;
4
5 use Illuminate\Foundation\Support\Providers\AuthServiceProvider as ServiceProvider;
6 use Illuminate\Support\Facades\Gate;
7
8 class AuthServiceProvider extends ServiceProvider
9 {
10     /**
11      * The policy mappings for the application.
12      *
13      * @var array
14      */
15     protected $policies = [
16         // 'App\Models\Model' => 'App\Policies\ModelPolicy',
17     ];
18
19     /**
20      * Register any authentication / authorization services.
21      *
22      * @return void
23      */
24     public function boot()
25     {
26         $this->registerPolicies();
27
28         //
29     }
30 }
```



```
1 <?php
2
3 namespace App\Providers;
4
5 use Illuminate\Foundation\Support\Providers\AuthServiceProvider as ServiceProvider;
6 use Illuminate\Support\Facades\Gate;
7 use Laravel\Passport\Passport;
8
9 class AuthServiceProvider extends ServiceProvider
10 {
11     /**
12      * The policy mappings for the application.
13      *
14      * @var array
15      */
16     protected $policies = [
17         'App\Models\Model' => 'App\Policies\ModelPolicy',
18     ];
19
20     /**
21      * Register any authentication / authorization services.
22      *
23      * @return void
24      */
25     public function boot()
26     {
27         $this->registerPolicies();
28         Passport::Routes();
29
30         //
31     }
32 }
```

Configurar serviço de token na classe User

User.php X

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Database\Eloquent\Factories\HasFactory;
7 use Illuminate\Foundation\Auth\User as Authenticatable;
8 use Illuminate\Notifications\Notifiable;
9 use Laravel\Sacntum\HasApiTokens;
10
11 class User extends Authenticatable
12 {
13     use HasApiTokens, HasFactory, Notifiable;
14 }
```

substituir

EXPLORER

- POKEMON
 - app
 - Console
 - Exceptions
 - Http
 - Models
 - User.php

User.php X

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Database\Eloquent\Factories\HasFactory;
7 use Illuminate\Foundation\Auth\User as Authenticatable;
8 use Illuminate\Notifications\Notifiable;
9 use Laravel\Passport\HasApiTokens;
10
11 class User extends Authenticatable
12 {
13     use HasApiTokens, HasFactory, Notifiable;
14 }
```

Gerando o database

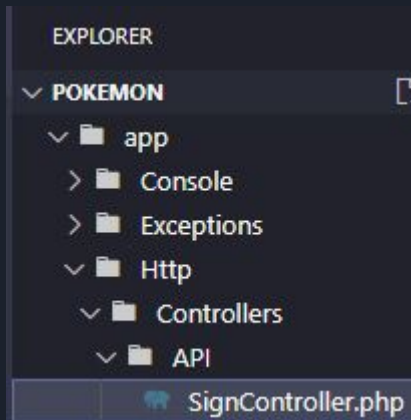
```
Souza@DESKTOP-N3VUF0I MINGW64 /c:/xampp/htdocs/pokemon
$ php artisan migrate ←
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (9.58ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (5.56ms)
Migrating: 2016_06_01_000001_create_oauth_auth_codes_table
Migrated: 2016_06_01_000001_create_oauth_auth_codes_table (7.15ms)
Migrating: 2016_06_01_000002_create_oauth_access_tokens_table
Migrated: 2016_06_01_000002_create_oauth_access_tokens_table (7.32ms)
Migrating: 2016_06_01_000003_create_oauth_refresh_tokens_table
Migrated: 2016_06_01_000003_create_oauth_refresh_tokens_table (5.29ms)
Migrating: 2016_06_01_000004_create_oauth_clients_table
```


Implementando autenticação e autorização



Criar métodos de autenticação

```
Souza@DESKTOP-N3VUF0I MINGW64 /c/xampp/htdocs/pokemon  
$ php artisan make:controller API/SignController  
Controller created successfully.
```



```
SignController.php X  
1  <?php  
2  
3  namespace App\Http\Controllers\API;  
4  
5  use App\Http\Controllers\Controller;  
6  use Illuminate\Http\Request;  
7  
8  class SignController extends Controller  
9  {  
10     //  
11 }  
12
```

Inserir métodos aqui

Método de cadastro (signUp)

```
use App\Models\User;  
use Illuminate\Support\Facades\Hash;
```

← Acima da classe SignController

```
public function signUp( Request $request ){  
  
    $user = User::create( [  
        'name'      => $request->name,  
        'email'     => $request->email,  
        'password' => Hash::make( $request->password )  
    ] );  
  
    return response()->json( [ 'mensagem' => 'Cadastrado com sucesso' ] );  
}
```

← Dentro da classe SignController

Método de login (signIn)

`use Auth;`

← Acima da classe SignController

```
public function signIn( Request $request ){  
  
    $credenciais = $request->validate( [ 'email' => 'email|required', 'password' => 'required' ] );  
    $autenticado = Auth::attempt( $credenciais );  
  
    if( !$autenticado )  
        return response()->json( [ 'mensagem' => 'Erro, verifique suas credenciais' ] );  
  
    $user = Auth::user();  
    $token = $user->createToken( "pokeToken" )->accessToken;  
  
    return response()  
        ->json( [ 'nome' => $user->name, 'email' => $user->email, 'access_token' => $token ] );  
}
```

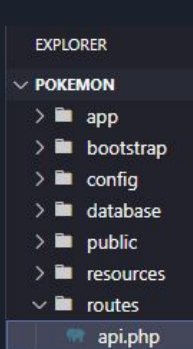
← Dentro da classe SignController

Método de logout (signOut)

```
public function signOut( Request $request ){  
    $exito = $request->user()->token()->delete();  
  
    if( !$exito ) return response()->json( [ 'mensagem' => 'Erro, falha na autenticação' ] );  
  
    return response()->json( [ 'mensagem' => 'Deslogado com sucesso' ] );  
}
```

Dentro da classe SignController

Configurando as rotas da api



```
api.php x
1 <?php
2
3 use Illuminate\Http\Request;
4 use Illuminate\Support\Facades\Route;
5
6 /*
7  |-----
8  | API Routes
9  |-----
10 |
11 | Here is where you can register API routes for your application. These
12 | routes are loaded by the RouteServiceProvider within a group which
13 | is assigned the "api" middleware group. Enjoy building your API!
14 |
15 |*/
16
17 Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
18     return $request->user();
19 });
20
```



```
api.php x
1 <?php
2
3 namespace App\Http\Controllers\API;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\Route;
7
8 /*
9  |-----
10 | API Routes
11 |-----
12 |
13 | Here is where you can register API routes for your application. These
14 | routes are loaded by the RouteServiceProvider within a group which
15 | is assigned the "api" middleware group. Enjoy building your API!
16 |
17 |*/
18
19 Route::post('cadastro', [ SignController::class, 'signUp' ] );
20 Route::post('login', [ SignController::class, 'signIn' ] );
21
22 Route::middleware('auth:api')->group( function () {
23     Route::post('logout', [ SignController::class, 'signOut' ] );
24 });
25
```

CRUD





Crud de uma classe chamada Pokemon

Atributos:

- Nome;
- Tipo;
- Peso;

Um usuário pode ter vários pokemon e um pokemom pertence a um usuário

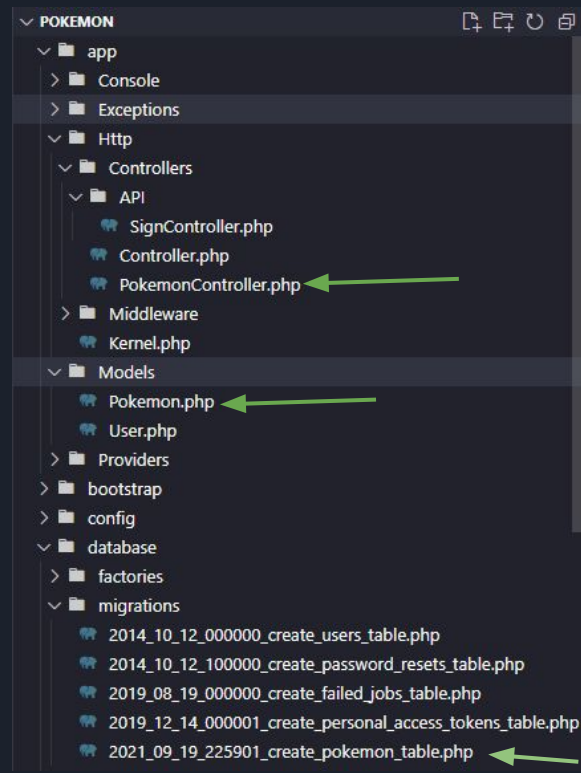
Criando os arquivos para realizar o CRUD de Pokemons

```
Souza@DESKTOP-N3VUF0I MINGW64 /c:/xampp/htdocs/pokemon
$ php artisan make:model Pokemon -mc
Model created successfully.
Created Migration: 2021_09_19_225901_create_pokemon_table
Controller created successfully.
```

Obs:

php artisan make:model <nome-da-model> (cria a classe model)
os parâmetros -mc criam, respectivamente:

- a migration (script de criação da tabela)
- o controller



Editando a migration da classe Pokemon

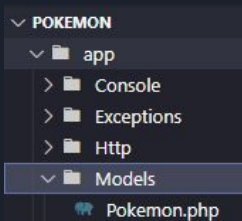
```
2021_09_19_225901_create_pokemon_table.php X
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreatePokemonTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14      public function up()
15      {
16          Schema::create('pokemon', function (Blueprint $table) {
17              $table->id();
18              $table->timestamps();
19          });
20      }
```

```
> app
> bootstrap
> config
> database
> factories
> migrations
  2014_10_12_000000_create_users_table.php
  2014_10_12_100000_create_password_resets_table.php
  2019_08_19_000000_create_failed_jobs_table.php
  2019_12_14_000001_create_personal_access_tokens_table.php
  2021_09_19_225901_create_pokemon_table.php
```

```
class CreatePokemonTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('pokemon', function (Blueprint $table) {
            $table->id();
            $table->string('nome')->nullable(false);
            $table->enum('tipo', [ 'agua', 'grama', 'fogo' ]);
            $table->float('peso');
            $table->timestamps();

            $table->bigInteger('user_id');
            $table->foreign('user_id')->references('id')->on('users');
        });
    }
}
```

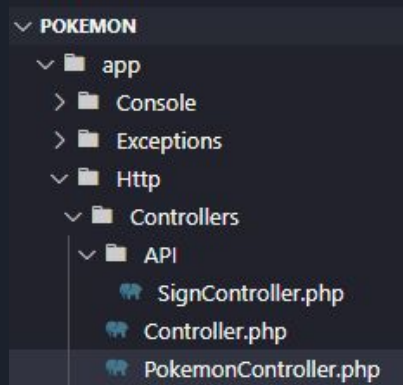
Editando a model da classe Pokemon



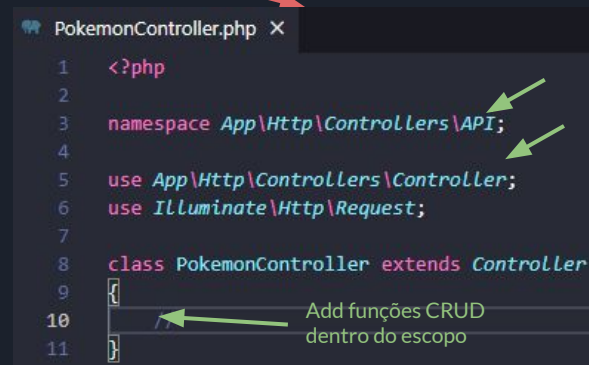
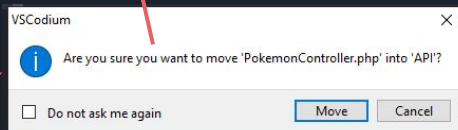
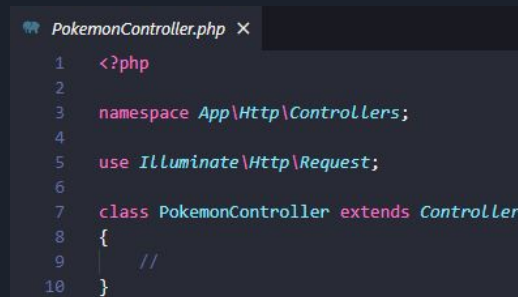
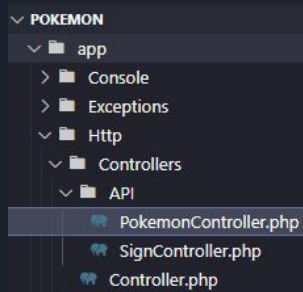
```
Pokemon.php X
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Pokemon extends Model
9  {
10     use HasFactory;
11 }
12
```

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

Editando o controller da classe Pokemon



Movendo o arquivo
PokemonController.php
para
app/Http/Controllers/API



Editando o controller da classe Pokemon (create)

```
use App\Models\Pokemon;  
use Auth;
```

Acima da classe PokemonController

```
public function create( Request $request ){  
  
    $user = Auth::user();  
  
    Pokemon::create( [  
        'nome'      => $request->nome,  
        'tipo'      => $request->tipo,  
        'peso'      => $request->peso,  
        'user_id'   => $user->id  
    ] );  
  
    return response()->json( [ 'mensagem' => 'pokemon cadastrado com sucesso' ] );  
}
```

Dentro da classe PokemonController

Editando o controller da classe Pokemon (retrieve)

```
public function retrieve( Request $request ){  
  
    $user = Auth::user();  
  
    $listaPokemons = Pokemon::where( 'user_id', $user->id )  
                                ->get();  
  
    return response()  
        ->json( [  
            'mensagem' => 'Lista de pokemons de ' . $user->name,  
            'pokemons' => $listaPokemons  
        ] );  
}
```

Dentro da classe PokemonController

Editando o controller da classe Pokemon (update)

```
public function update( Request $request ){

    $user = Auth::user();

    $resultado = Pokemon::where( 'user_id', $user->id )
        ->update( [
            'nome' => $request->nome,
            'tipo' => $request->tipo,
            'peso' => $request->peso
        ] );

    if( $resultado )
        return response()->json( [ 'mensagem' => 'Alterado com sucesso' ] );
    else
        return response()->json( [ 'mensagem' => 'Erro' ] );
}
```

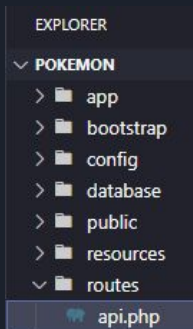
Dentro da classe PokemonController

Editando o controller da classe Pokemon (delete)

```
public function delete( Request $request ){  
  
    $user = Auth::user();  
  
    $resultado = Pokemon::where( 'user_id', $user->id )  
                        ->where( 'id', $request->id )  
                        ->delete();  
  
    if( $resultado )  
        return response()->json( [ 'mensagem' => 'Deletado com sucesso' ] );  
    else  
        return response()->json( [ 'mensagem' => 'Erro, elemento não encontrado' ] );  
}
```


Dentro da classe PokemonController

Criando rotas para os novos controllers



```
api.php x
1 <?php
2
3 namespace App\Http\Controllers\API;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\Route;
7
8 /*
9 |-----
10 | API Routes
11 |-----
12 |
13 | Here is where you can register API routes for your application. These
14 | routes are loaded by the RouteServiceProvider within a group which
15 | is assigned the "api" middleware group. Enjoy building your API!
16 |
17 |*/
18
19 Route::post( 'cadastro', [ SignController::class, 'signUp' ] );
20 Route::post( 'login', [ SignController::class, 'signIn' ] );
21
22 Route::middleware('auth:api')->group( function () {
23     Route::post( 'logout', [ SignController::class, 'signOut' ] );
24 });
25
```

```
1 <?php
2
3 namespace App\Http\Controllers\API;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\Route;
7
8 /*
9 |-----
10 | API Routes
11 |-----
12 |
13 | Here is where you can register API routes for your application. These
14 | routes are loaded by the RouteServiceProvider within a group which
15 | is assigned the "api" middleware group. Enjoy building your API!
16 |
17 |*/
18 Route::post( 'cadastro', [ SignController::class, 'signUp' ] );
19 Route::post( 'login', [ SignController::class, 'signIn' ] );
20
21 Route::middleware('auth:api')->group( function () {
22     Route::post( 'logout', [ SignController::class, 'signOut' ] );
23
24     Route::post( 'novo', [ PokemonController::class, 'create' ] );
25     Route::post( 'lista', [ PokemonController::class, 'retrieve' ] );
26     Route::post( 'alteracao', [ PokemonController::class, 'update' ] );
27     Route::post( 'exclusao', [ PokemonController::class, 'delete' ] );
28 });
```



Reconstruindo o banco de dados & executando a aplicação

```
Souza@DESKTOP-N3VUF0I MINGW64 /c/xampp/htdocs/pokemon  
$ php artisan migrate  
Migrating: 2021_09_19_225901_create_pokemon_table  
Migrated: 2021_09_19_225901_create_pokemon_table (26.99ms)
```

```
Souza@DESKTOP-N3VUF0I MINGW64 /c/xampp/htdocs/pokemon  
$ php artisan serve  
Starting Laravel development server: http://127.0.0.1:8000
```



O que não foi abordado

- Criação de uuids;
- Validação dos requests;
- Front-end;
- Schedules para a limpeza da tabela de tokens;
- Formas de logout;
- DatabaseSeeders & DataFactory;
- Detalhes sobre o Eloquent;
- Detalhes sobre o QueryBuilder;

Visto que existe diversos tipos de pokemon, definí-los como enum não é uma boa ideia, o ideal é criar uma tabela de tipos, porém como o objetivo é apenas ilustrar uma introdução optou-se pelo enum

OBRIGADO