

Curso Técnico Superior Profissional em: Tecnologias e Programação de Sistemas de Informação

2º Ano/1º Semestre

Unidade Curricular: Aplicações Centradas em Redes

Docente: Michael Silva / Hugo Perdigão

INTRODUÇÃO ÀS LINGUAGENS PARA O DESENVOLVIMENTO DE APLICAÇÕES CENTRADAS EM REDES

MVC: Model, View, Controller

Model View Controller (MVC) is a **software architecture pattern, commonly used**: it is therefore a **popular choice for architecting web apps**. In general, it **separates out the application logic into three separate parts**, promoting **modularity and ease of collaboration and reuse**. It also makes applications **more flexible and welcoming to iterations**.

The parts of MVC

Model: Model code typically reflects real-world things. This code can hold raw **data**, or it will define the essential components of your app. For instance, if you were building a **To-do app**, the model code would **define what a “task” is and what a “list” is** – since those are the main components of a todo app.

View: View code is made up of all the **functions that directly interact with the user**. This is the code that makes your app **look nice**, and otherwise defines **how your user sees and interacts with it**.

Controller: Controller code acts as a liaison between the Model and the View, receiving user input and **deciding what to do with it**. It's the **brains** of the application, and **ties together the model and the view**.

Cofinanciado por:

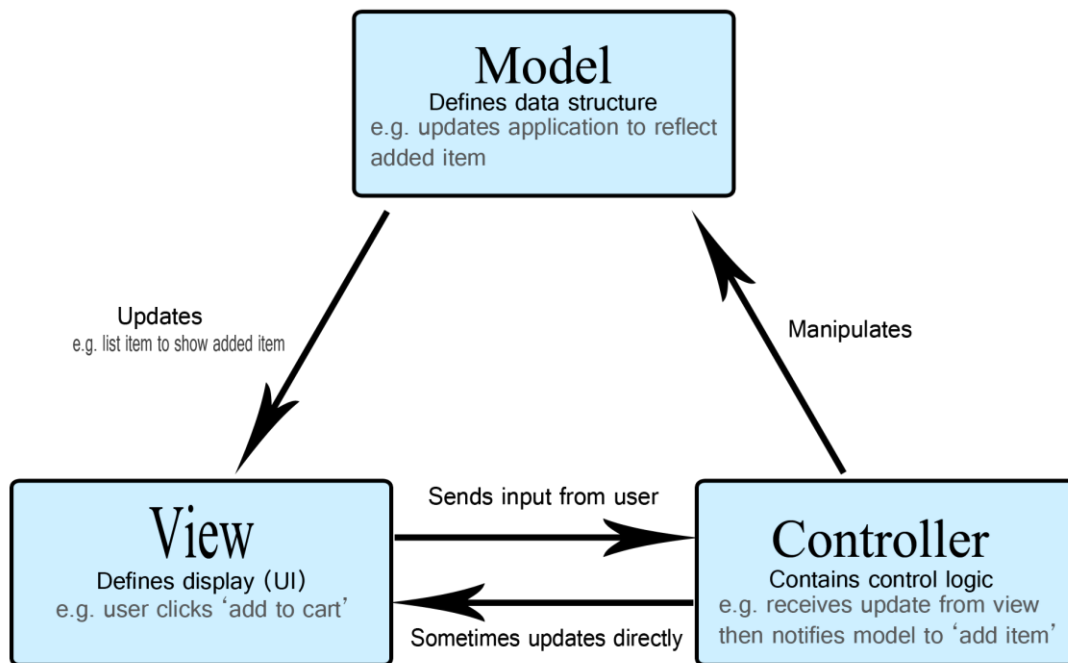
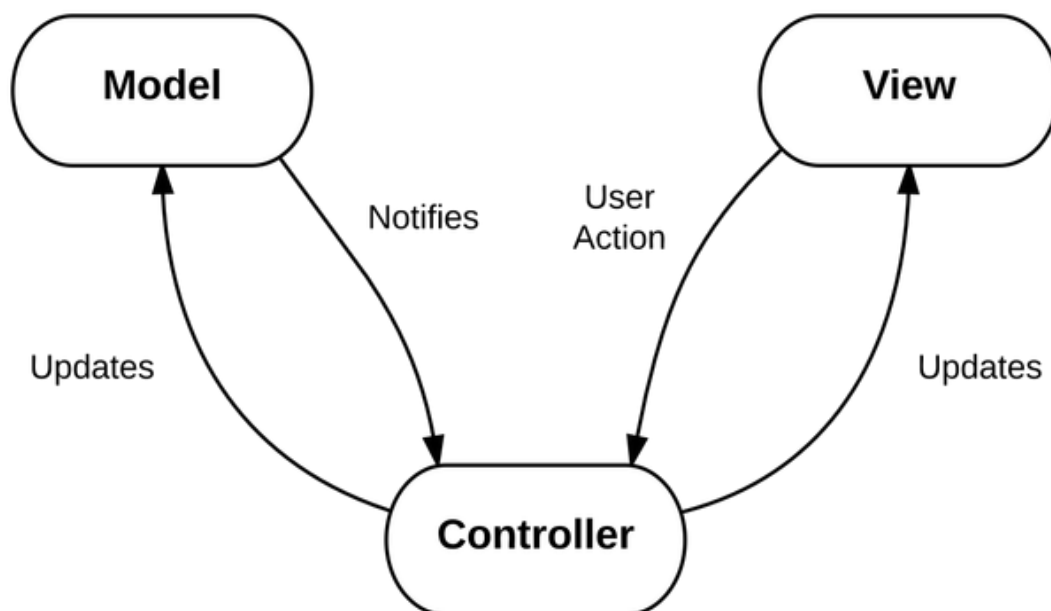


Figure 1: Exemplo de arquitetura MVC

Exemplo 2 (melhor adaptado ao laravel)



Cofinanciado por:

Laravel

- **Routes:** para definir os endereços no url a usar. Estão ligadas às views e aos controllers, ou seja, as Routes fazem a ponte entre views e controllers.
 - Views – Routes - Controllers
- **Views:** templates
 - Blade templates engine, permite:
 - Usar um layout genérico comum a toda a app
 - Receber, apresentar e gerir dados nos templates de forma fácil
- **Controllers:** Onde são definidas as regras de negócio/business logic da app.

Instalação

Instalar o Composer (dependency manager for PHP)

Windows:

- <https://getcomposer.org/doc/00-intro.md#installation-windows>
- Fazer download e correr o ficheiro .exe para instalar
- Abrir um novo terminal OU fechar o terminal actual e abrir um novo terminal
- Escrever **composer -V** no terminal para verificar a instalação

Mac:

- <https://getcomposer.org/download/>
- Copiar as linhas indicadas e correr no terminal
- Instalar globalmente, no terminal:

```
○ mv composer.phar /usr/local/bin/composer
```

Cofinanciado por:



Instalar o Laravel

No terminal:

```
composer global require "laravel/installer"
```

Windows:

- O Instalador do composer já configurou

Mac:

- Abrir o ficheiro ~/.bash_profile e/ou ~/.bashrc
- Adicionar a linha: export PATH="\$HOME/.composer/vendor/bin:\$PATH"
- Fechar o terminal atual e abrir um novo terminal

Para verificar, no terminal, escrever: laravel

Criar uma nova aplicação/projecto laravel

Na pasta a criar o projecto, correr o seguinte comando:

```
laravel new mysite
```

Entrar no directório do projecto:

```
cd mysite
```

Local Development Server

```
php artisan serve
```

<http://127.0.0.1:8000/>

Explorar a pasta do projecto, tem muitas pastas e ficheiros.
Vamos aprender aos poucos, conforme for preciso.

Cofinanciado por:



Basic Routing

- As routes ficam no ficheiro `projectname/routes/web.php`
- As views (templates) ficam na pasta `/resources/views/`
 - `welcome.blade.php`
 - tem o nome blade que é o templates engine

Blade layout files

- No **VS Code**, instalar a extension: laravel blade snippets

Permite definir num ficheiro o **Layout** comum a toda a app.

- Para definir secções no Ficheiro pai: `layout.blade.php`
 - `@yield('content')`
 - com único valor simples default:
 - `@yield('title', 'ACR')`

- Nos ficheiros filho

- No início do ficheiro, usar:
 - `@extends('layout')`
- para definir conteúdo da secção (definida no pai)

```
@section('content')
<h1>Contact Form</h1>
@endsection
```

- Section com único valor (não é preciso `@endsection` pois não é um bloco)
 - `@section('title', 'value')`

Cofinanciado por:



Sending Data to your views

Na route:

```
return view('welcome', [  
    'text' => 'ACR',  
    'title' => request('title')  
]);
```

Na view:

```
{{ $text }}
```

- Variáveis são escaped por omissão, pode ser retirado caso necessário

Controllers

Ficheiros onde são definidas as regras de negócio/business logic da app.

Ficam na pasta app/Http/Controllers

Comando para criar controllers, **desde a pasta do projecto**:
php artisan make:controller PagesController

Fontes e mais recursos

<https://laravel.com/>

<https://laracasts.com/series/laravel-from-scratch-2018>

<https://laravel.com/docs>

<https://laracasts.com/skills/laravel>

Cofinanciado por:

