



Módulo 8: Drupal

Theming

Aula 6

Tabela de conteúdos

01

Temas

A camada de
apresentação do Drupal

02

Twig

Sistema de templates
do Drupal

01

Temas

Camada de apresentação do Drupal

Temas

Os temas são responsáveis pelo aspecto do site, providenciando o HTML, CSS, JS, tipografia, imagens e experiência interactiva que as pessoas encontram quando visualizam o site.

Existem duas maneiras de adicionar temas a um site:

- Descarregar e instalar um tema de drupal.org (www.drupal.org/project/project_theme)
- Criar um tema personalizado

olivero

Home Products About Blog Contact

Black & White Photos
Stone Statues
Antiques & Artifacts
Replicas

Drupal 9: T Beautiful C...!

by [Dries Buytaert](#), 14 January 2020

SHARE THIS POST

If you thought opening paragraphs were amazing, wait till you begin writing blog posts with leads! They'll blow your mind. They draw your readers into an article.

Once you've got them reading with some nice punchy bold text that grabs their attention, they're like puddin' in your hands. They'll click all your links and read all your comments and probably think you have the best site they've ever been on.

Instalação de temas via Drupal UI

(método séc. XVII)

bootstrap 8.x-3.25

Install

Works with Drupal: ^9.3 || ^10

```
composer require 'drupal/bootstrap:^3.25'
```



[Using Composer to manage Drupal site dependencies](#)

Alternative installation files

[Download tar.gz](#) 251.28 KB

[Download zip](#) 421.88 KB

Downloads are for manual installation, which is not recommended when using Drupal 8 or later.

View file hashes: [MD5](#), [SHA-1](#), [SHA-256](#)

01

Copiar URL do tema

Ir à página do tema e copiar o link "Download tar.gz" sob a secção "Alternative installation files" (clique com o botão direito do rato e escolher "Copiar ligação").

Instalação de temas via Drupal UI (método séc. XVII)

Home > Administration

Add new theme ☆

You can find [modules](#) and [themes](#) on [drupal.org](#). The following file extensions are supported: *zip tar tgz gz bz2*.

Add from a URL

For example: <https://ftp.drupal.org/files/projects/name.tar.gz>

Or

Upload a module or theme archive

Nenhum ficheiro selecionado.

For example: *name.tar.gz* from your local computer

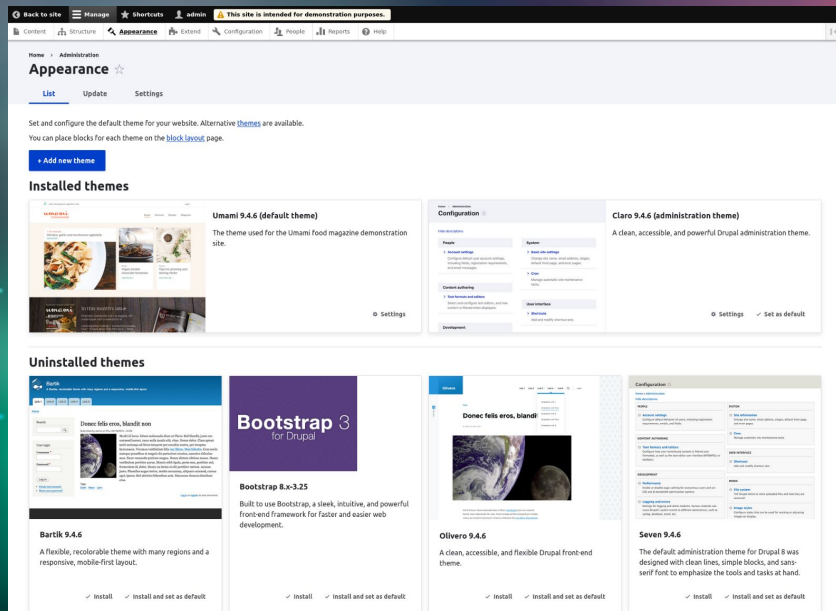
Continue

02

Colar URL do tema

Colar o URL em `/admin/theme/install` no campo “Add from url” e premir o botão “Continue”

Instalação de temas via Drupal UI (método séc. XVII)



03

Ativar tema

Ativar o tema em /admin/appearance e
escolher as suas configurações

Instalação de temas via Composer e Drush

bootstrap 8.x-3.25

Install

Works with Drupal: ^9.3 || ^10

```
composer require 'drupal/bootstrap:^3.25'
```



[Using Composer to manage Drupal site dependencies](#)

Alternative installation files

[Download tar.gz](#) 251.28 KB

[Download zip](#) 421.88 KB

Downloads are for manual installation, which is not recommended when using Drupal 8 or later.

View file hashes: [MD5](#), [SHA-1](#), [SHA-256](#)

01

Copiar o comando composer e correr na linha de comandos

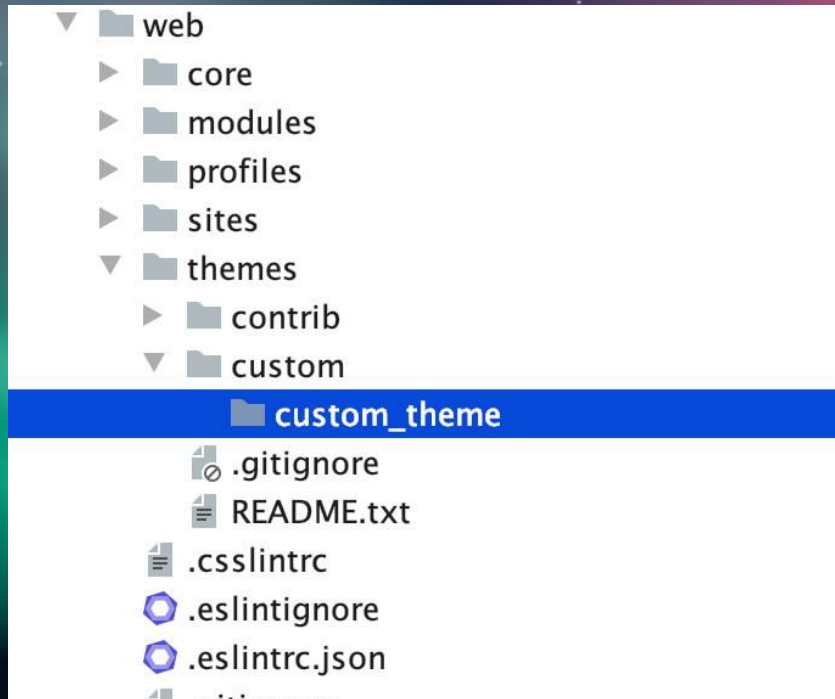
```
composer require 'drupal/bootstrap:^3.25'
```

02

Ativar o tema e definir como padrão com o Drush

```
drush theme:enable bootstrap  
drush config:set system.theme default  
bootstrap --yes
```


Criar tema personalizado



01

Criar pasta “Custom” em themes/

02

Criar pasta “custom_theme”
(nome do tema a criar) em
themes/custom

Criar tema personalizado

03

Criar ficheiro custom_theme.info.yml com a seguinte informação:

```
// web > themes > custom > custom_theme > custom_theme.info.yml
```

```
name: Custom Theme
```

```
type: theme
```

```
description: 'Custom Theme for My Website.'
```

```
package: Other
```

```
core_version_requirement: ^8 || ^9
```

Criar tema personalizado

04

Criar ficheiro `libraries.yml` para especificar as bibliotecas de CSS e JS e as suas pastas correspondentes

```
// web > themes > custom > custom_theme > custom_theme.libraries.yml
```

```
global-styling:
```

```
  version: 1.x
```

```
  js:
```

```
    js/script.js: {}
```

```
  css:
```

```
    theme:
```

```
      css/style.css: {}
```

Criar tema personalizado

05

Ligar o ficheiro libraries.yml file ao tema, adicionando-o no ficheiro info.yml:

```
// web > themes > custom > custom_theme > custom_theme.info.yml
```

```
name: Custom Theme
```

```
type: theme
```

```
description: 'Custom Theme for My Website.'
```

```
package: Other
```

```
core_version_requirement: ^8 || ^9
```

```
libraries:
```

```
- custom_theme/global-styling
```

Criar tema personalizado

06

Herdar o tema base, neste caso o 'classy', que faz parte do Drupal core.

```
// web > themes > custom > custom_theme > custom_theme.info.yml
```

```
name: Custom Theme
```

```
type: theme
```

```
description: 'Custom Theme for My Website.'
```

```
package: Other
```

```
core_version_requirement: ^8 || ^9
```

```
libraries:
```

```
- custom_theme/global-styling
```

```
base theme: classy
```


Criar tema personalizado

07

Definir as regiões do tema

```
// web > themes > custom > custom_theme > custom_theme.info.yml
```

```
name: Custom Theme
type: theme
description: 'Custom Theme for My Website.'
package: Other
core_version_requirement: ^8 || ^9
libraries:
- custom_theme/global-styling
base theme: classy
regions:
  branding: Branding
  navigation: Main navigation
  search: Search
  content: Content
  right_sidebar: Right sidebar
  footer: Footer
```

Criar tema personalizado

08

Criar o ficheiro page.html.twig para ativar as nossas regiões em vez das presentes no tema de base (classy) e estamos prontos para ativar o tema:

```
// web > themes > custom > custom_theme > templates > page.html.twig
```

```
<header aria-label="Site header" class="header" id="header" role="banner">
  {{ page.branding }}
  {{ page.navigation }}
  {{ page.search }}
</header>
<section class="main" id="main">
  <main aria-label="Site main content" class="content" id="content" role="main">
    {{ page.content }}
  </main>
  <aside class="right--sidebar" id="sidebar" role="complementary">
    {{ page.right_sidebar }}
  </aside>
</section>
<footer aria-label="Site footer" class="footer" id="footer" role="contentinfo">
  {{ page.footer }}
</footer>
```

02

Twig

Sistema de templates do Drupal

Twig em Drupal

O Twig é um template engine para PHP e faz parte da framework Symfony2.

Podemos sobrepor os templates Drupal core adicionando templates ao nosso tema nomeados de acordo com uma nomenclatura específica (naming convention).

Para sobrepor os templates, é necessário:

- Localizar o template que deseja substituir
- Copiar o ficheiro do template para a pasta do nosso tema
- (opcionalmente) Renomear o template de acordo com a nomenclatura
- Modificar o template

Depois de adicionar um template, temos de reconstruir (rebuild) a cache para que o Drupal possa descobrir o novo template.



Nomenclatura para os templates Twig

Todas as convenções em

<https://www.drupal.org/docs/theming-drupal/twig-in-drupal/twig-template-naming-conventions>

HTML (<head> template)

O template que providencia o código para a estrutura básica da página de HTML. Inclui as tags <head>, <title> e <body>.

Base template

html.html.twig

Localização

core/modules/system/templates/html.html.twig

Exemplos de nomenclatura:

html--[internalviewpath].html.twig

html--node--[nodeid].html.twig

html.html.twig

Nomenclatura para os templates Twig

Page template

O template que providencia o código para as páginas.

Base template

page.html.twig

Localização

core/modules/system/templates/page.html.twig

Exemplos de nomenclatura:

page--front.html.twig (homepage)

page--node--edit.html.twig

page--node--1.html.twig

page--node.html.twig

page.html.twig

Nomenclatura para os templates Twig

Block template

O template que providencia o código para os blocos.

Base template

block.html.twig

Localização

core/modules/block/templates/block.html.twig

Exemplos de nomenclatura:

block--[module]--[delta].html.twig

block--[module].html.twig

block.html.twig

Nomenclatura para os templates Twig

Node template

O template que providencia o código para os nodes.

Base template

node.html.twig

Localização

core/modules/node/templates/node.html.twig

Exemplos de nomenclatura:

node--[nodeid]--[viewmode].html.twig

node--[nodeid].html.twig

node--[content-type]--[viewmode].html.twig

node--[content-type].html.twig

node--[viewmode].html.twig

node.html.twig

Nomenclatura para os templates Twig

Field template

O template que providencia o código para os campos.

Base template

field.html.twig

Localização

core/modules/system/templates/field.html.twig

Exemplos de nomenclatura:

field--node--[field-name]--[content-type].html.twig

field--node--[field-name].html.twig

field--node--[content-type].html.twig

field--[field-name].html.twig

field--[field-type].html.twig

field.html.twig

Sintaxe Twig

Variáveis (adicionar . para aceder às propriedades)

```
{{ foo.bar }}
```

Criar variáveis

```
{% set custom_var = content.comments %}
```

Criar arrays

```
{% set args = {'author': author, 'date': created} %}
```

Condicionais

```
{% if content.comments %} {% endif %}
```

```
{% if content.comments is not empty %} {% endif %}
```

```
{% if content.comments is defined %} {% endif %}
```

```
{% if count > 0 %} ... {% elseif count > 1 %} ... {% endif %}
```

Estruturas de controlo

```
{% for user in users %} {{ user.name }} {% endfor %}
```


Sintaxe Twig

Filtros

Valores raw

```
{{ content|raw }}
```

Tradução

```
{{ 'Expressão a traduzir'|t }}
```

Data

```
{{ created|date('Y') }}
```

```
{{ created|format_date('short') }}
```

Tamanho (arrays ou strings)

```
{{ item.files|length }}
```

Mais filtros em

<https://www.drupal.org/docs/theming-drupal/twig-in-drupal/filters-modifying-variables-in-twig-templates>

<https://twig.symfony.com/doc/2.x/filters/index.html>

Sintaxe Twig

Comentário

{# O código abaixo não irá estar ativo

{% for user in users %} ... {% endfor %}

#}

Incluir outros templates

{{ include('sidebar.html') }}

Sintaxe Twig

Template inheritance

```
// base.html.twig
```

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    {% block head %}
```

```
      <link rel="stylesheet" href="style.css"/>
```

```
      <title>{% block title %}{% endblock %} - My Webpage</title>
```

```
    {% endblock %}
```

```
  </head>
```

```
  <body>
```

```
    <div id="content">{% block content %}{% endblock %}</div>
```

```
    <div id="footer">
```

```
      {% block footer %}
```

```
        &copy; Copyright 2011 by <a href="http://domain.invalid/">you</a>.
```

```
      {% endblock %}
```

```
    </div>
```

```
  </body>
```

```
</html>
```

Sintaxe Twig

Template inheritance

```
// extends_base.html.twig

{% extends "base.html" %}

{% block title %}Index{% endblock %}

{% block head %}

    <style type="text/css">
        .important { color: #336699; }
    </style>

{% endblock %}

{% block content %}

    <h1>Index</h1>

    <p class="important">Welcome to my awesome homepage. </p>

{% endblock %}
```

Exercício 1: módulo Twig Tweak

- **Instalar e ativar o módulo Twig Tweak**

https://www.drupal.org/project/twig_tweak

- **Ler documentação do módulo**

https://git.drupalcode.org/project/twig_tweak/-/blob/3.x/docs/cheat-sheet.md

<https://www.drupal.org/docs/contributed-modules/twig-tweak-2x/cheat-sheet>

Exercício 2: ativar modo debug

- **Navegar para a pasta /sites**
- **Copiar o ficheiro `example.settings.local.php`, renomear para `settings.local.php` e mover este ficheiro para a pasta `/sites/default`**
- **Descomentar as seguintes linhas no ficheiro `settings.local.php` (retirar “#”)**
`$settings['cache']['bins']['render'] = 'cache.backend.null';`
`$settings['cache']['bins']['dynamic_page_cache'] = 'cache.backend.null';`
`$settings['cache']['bins']['page'] = 'cache.backend.null';`

Exercício 2: ativar modo debug

- **Abrir o ficheiro `development.services.yml` na pasta `/sites` e adicionar (cuidado com a indentação) e adicionar:**

```
parameters:
  http.response.debug_cacheability_headers: true
twig.config:
  debug: true
  auto_reload: true
  cache: false
services:
  cache.backend.null:
    class: Drupal\Core\Cache\NullBackendFactory
```

Exercício 2: ativar modo debug

- **Navegar para a pasta /sites/default, abrir o ficheiro settings.php e descomentar o seguinte código, caso esteja comentado:**

```
$local_settings = __DIR__ . "/settings.local.php";  
if (file_exists($local_settings)) {  
    include $local_settings;  
}
```

- **Limpar a cache**
ddev drush cr

Exercício 3: criar view Eventos

- **Importar base de dados do TPC da aula 2 (quem não tiver o TPC da aula 2 feito)**
- **Criar view do tipo de conteúdos Eventos**
 - View name: Eventos
 - Machine name: events
 - View settings: type Evento
 - Page settings: Create a page
 - Menu: Normal entry menu

Exercício 4: ativar tema

- Fazer download do ficheiro `my_umami.zip` do Teams
- Descompactar para a pasta `themes/custom`
- Ativar tema **My Umami** em **Appearance**
- Descomentar a linha 23 do ficheiro `views-view--events.html.twig` em `my_umami/templates`
`{{ dd(rows.0['#rows'].0['#node']) }}`

TPC: criar e modificar view Restaurantes

- **Criar view do tipo de conteúdos Restaurante à semelhança da view Eventos**
- **Criar novo template para a view Restaurantes de modo a ter um slideshow de Restaurantes**
Consultar documentação do splide se existirem dúvidas
<https://splidejs.com/guides/getting-started/>
- **Sincronizar ambiente local com o Pantheon. Quem não conseguir sincronizar e só tiver o trabalho no ambiente de desenvolvimento local, tirar captura de ecrã e submeter na tarefa.**