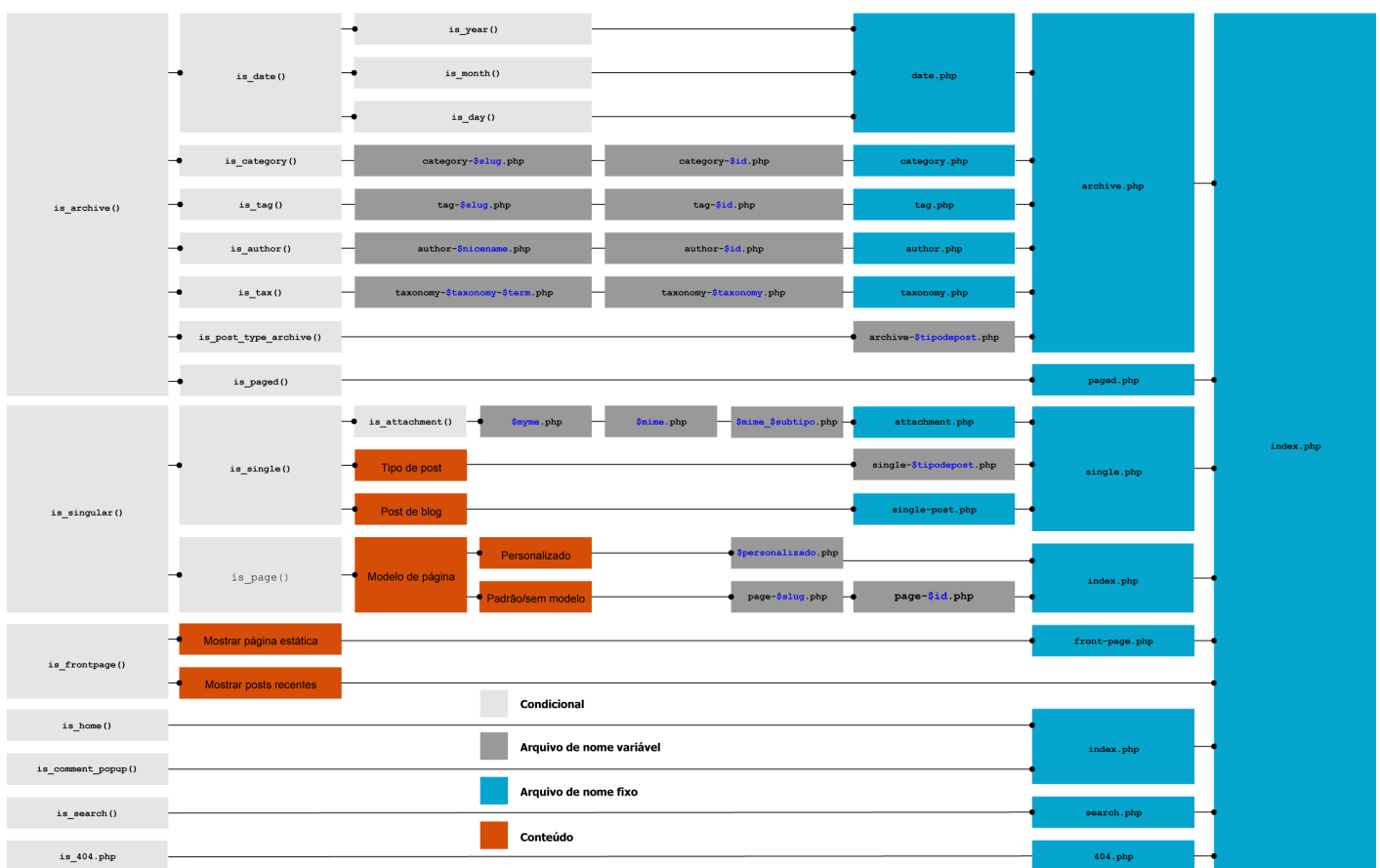


# Como funciona a hierarquia de tema Wordpress?

## Hierarquia de modelos Wordpress

### HIERARQUIA DE MODELOS WORDPRESS



O diagrama acima pode até parecer assustador, mas não é. Funciona assim:





**OBS:** Atualizado para a versão 6.0 em diante (10/06/2022)

Na primeira coluna a esquerda, temos basicamente os tipos de conteúdo de um tema Wordpress que são:

- **Arquivo** - `is_archive()`
- **Página simples** - `is_singular()`
- **Página principal** - `is_frontpage()`

- `is_home()`
- `is_comment_popup()`
- **Página de busca** - `is_search()`
- **Página de erro** - `is_404()`

Para entender o **esquema de cores** deste diagrama leve em conta esta parte da tabela:

	Condicional
	Arquivo de nome variável
	Arquivo de nome fixo
	Conteúdo

Agora vamos a explicação...

## Como funciona?

Apenas para exemplificar o básico, vamos mencionar as situações principais que podem ser percebidas neste diagrama:

- **Página principal** - quem já é usuário Wordpress sabe que existe uma configuração de como será a leitura do site, se será uma exibição de posts recentes ou uma página estática. Bom, na sua hierarquia será:
  1. Se ele estiver configurado para exibir uma página estática. irá buscar por **front-page.php**. Se ele não encontrar este arquivo, ou se o seu tema não o tiver, ele irá exibir o arquivo **index.php** da pasta raiz.
  2. Se ele estiver configurado para exibir os últimos posts, ele irá direto para o arquivo **index.php**.
- **Página de pesquisa** - buscará pelo arquivo **search.php**. Se não o encontrar, buscará por **index.php**.
- **Página de erro** - buscará pelo arquivo **404.php**. Se não o encontrar, buscará por **index.php**.
- **Arquivo** - se for um arquivo, ele verificará que tipo de arquivo está sendo consultado (Ex: `is_date()`, `is_category()`, `is_tag()`, `is_author()`, `is_tax()`, `is_post_type_archive()`, etc...) E para cada caso, o tema pode fornecer uma exibição diferente por tipo de conteúdo, bastando neste caso que tenha um arquivo para cada tipo.
  1. `is_date()` - **date.php**.
  2. `is_category()` - **category.php**
  3. `is_tag()` - **tag.php**
  4. `is_author()` - **author.php**
  5. `is_tax()` (Taxonomia personalizada) - **taxonomy.php**

6. `is_post_type_archive()` - `archive-{tipo_de_post}.php` (**Ex.** se eu tenho um Custom Post Type do tipo **produto**, o nome do arquivo será `archive-produto.php`)
7. Caso o seu tema não tenha um modelo de arquivo para cada tipo de conteúdo, a exibição de arquivo irá buscar diretamente o `archive.php`. Caso não haja também, ele irá exibir o `index.php`.
8. **Agora, imagine o seguinte cenário:**
  - O seu tema tem vários tipo de categoria, mas o arquivo de uma das categorias deve ser exibido de forma diferente das demais. Por exemplo: o arquivo das categorias **Notícias** e **Novidades** terão um layout e o arquivo da categoria **Artigos** terá outro formato. Você pode criar o modelo `category-{slug}.php` que neste caso provavelmente seria `category-artigos.php`.

Seguindo estes exemplos acima, é possível entender toda a hierarquia de modelos nos temas WordPress.

Documentação oficial: [Hierarquia de Arquivos de Tema WordPress](#)

---

Revision #3

Created 10 June 2022 21:15:21 by Patrick Castheliano

Updated 10 June 2022 22:01:47 by Patrick Castheliano