

01

## Refatorando function.php

### Transcrição

Recentemente aprendemos a colocar as *thumbnails* de imagens para ilustrar cada um dos imóveis do site, melhoramos o código HTML e incluímos classes para acessarmos e estilizarmos o CSS facilmente. Também incluímos e isolamos o suporte para novas funcionalidades em `function.php`, e trouxemos o CSS por meio de `get_template_directory_uri()` em `header.php`.

Até o momento, registramos os nossos imóveis em forma de postagens, porém, imagine que queremos vender o nosso tema, eventualmente, para alguém que quer montar a sua própria imobiliária, ou algo similar. Não chega a ser muito intuitivo se a pessoa tiver que cadastrar um post toda vez que quiser acrescentar um imóvel ao site, certo?

Como dito anteriormente, o Wordpress é basicamente um gerenciador de conteúdos, e apesar do post ser considerado um conteúdo, não se trata de um gerenciador de posts. Já sabemos que iremos incluir novas funcionalidades em `functions.php`, assim, vamos acrescentar a função `register_post_type()`, que serve justamente para registrar conteúdo novo, customizado, sendo o primeiro parâmetro `imovel`, um *slug*, ou apelido. O segundo parâmetro será uma série de configurações, `$args`, um *array*.

O código de `functions.php` ficará da seguinte forma:

```
<?php  
  
add_theme_support('post-thumbnails');  
  
$args = array(  
);  
  
register_post_type('imovel', $args );
```

Se atualizarmos o dashboard, não teremos nenhuma mudança visível. Precisaremos tornar esta função pública, fazendo a seguinte alteração no código que acabamos de implementar:

```
$args = array(  
    'public' => true  
);
```

Desta vez, ao atualizarmos o dashboard, verificaremos que houve a adição do link "Posts", que nos oferece as opções "Todos os posts" e "Adicionar novo". Esta configuração também recebe `labels`, que servirá para etiquetar, e também receberá um array.

```
add_theme_support('post-thumbnails');  
  
$labels = array(  
    'name' => 'Imóveis',  
    'name_singular' => 'Imóvel'  
);
```

```
$args = array(
    'labels' => $labels,
    'public' => true
);
```

Voltando ao navegador com o dashboard do Wordpress, o que antes era "Posts" está como "Imóveis", como gostaríamos! Vamos clicar no único imóvel cadastrado até então, "Casa da praia". No topo da página, se lê "Editar post", que é algo que precisaremos alterar também. São muitas modificações em vários arrays, e dificilmente decoraremos tudo isso, sendo assim, consultaremos [a documentação do Wordpress](#)

([http://codex.wordpress.org/Function\\_Reference/register\\_post\\_type](http://codex.wordpress.org/Function_Reference/register_post_type)) em relação a `register_post_type()`.

Como parâmetros, esta função recebe `$args`, argumentos que são as *labels*, cuja lista é grande. Para testar, vamos copiar e colar outra delas em nosso código:

```
$labels = array(
    'name' => 'Imóveis',
    'name_singular' => 'Imóvel',
    'add_new_item' => 'Adicionar novo imóvel'
);
```

Isto fará com que, no dashboard, tenhamos "Adicionar novo imóvel" em vez de "Adicionar novo post". Ou seja, estas etiquetas nos ajudam a nomear e customizar elementos; poderemos trocar "Editar post" por "Editar imóvel", por meio de `edit_item`, e assim por diante. Continuando, e se futuramente quisermos trocar a *label* "Imóveis" por "Casas", por exemplo. Teremos que fazer esta modificação manualmente, porém seria mais interessante isolar o código, como já fizemos anteriormente, assim:

```
add_theme_support('post-thumbnails');

$nomeSingular = 'Imóvel';
$nomePlural = 'Imóveis';

$labels = array(
    'name' => $nomePlural,
    'name_singular' => $nomeSingular,
    'add_new_item' => 'Adicionar novo ' . $nomeSingular,
    'edit_item' => 'Editar ' . $nomeSingular
);
```

Ao testarmos, veremos que tudo continua funcionando conforme esperado. De acordo com a documentação do Wordpress, além das etiquetas, é possível incluirmos uma descrição, que precisará conter dentre os argumentos. Ao lado do link de "Posts" no dashboard, há um ícone de pin, e o mesmo aparece ao lado de "Imóveis", o que não faz muito sentido. Na documentação, buscaremos por algo que nos ajude a alterar isto: `menu_icon`, para o qual poderemos passar uma *string* opcional com o ícone que queremos utilizar, por exemplo.

As opções para ícone podem ser buscadas na internet como "dashicon wordpress", inclusive existe [uma documentação para elas](#) (<https://developer.wordpress.org/resource/dashicons/#editor-ol>), com diversas opções disponíveis, como o `dashicons-admin-multisite`, que incluiremos no código:

```

$nomeSingular = 'Imóvel';
$nomePlural = 'Imóveis';
$description = 'Imóveis da imobiliária Malura';

$labels = array(
    /* código omitido */
);

$args = array(
    'labels' => $labels,
    'public' => true,
    'description' => $description,
    'menu_icon' => 'dashicons-admin-multisite'
);

```

Com isso, o ícone que aparece ao lado do link "Imóveis" passará a ser outro, bem mais coerente.

Há mais uma questão: nosso código em `functions.php` está um tanto desorganizado, portanto usaremos a função `cadastrando_post_type_imoveis()` para isolar o cadastro de imóveis, ou seja, todo o conteúdo após a linha `add_theme_support('post-thumbnails')` ficará dentro dela.

Porém, esta função não se chamará sozinha, precisaremos indicar ao Wordpress para que ela seja chamada. Para isto, utilizaremos a função `add_action()`, que executa tarefas e receberá dois parâmetros, sendo o segundo aquilo que queremos que ela execute, no caso, `cadastrando_post_type_imoveis()`, e o primeiro, o momento para que isto aconteça, `init`, que é logo no início da execução.

```

<?php

add_theme_support('post-thumbnails');

function cadastrando_post_type_imoveis() {
    /* código omitido */
}

add_action('init', 'cadastrando_post_type_imoveis');

```

Ao recarregarmos a página do dashboard, aparentemente tudo estará funcionando conforme esperado; vamos testar incluindo um novo imóvel, "Casa de madeira", por exemplo, com a descrição "A casa de madeira é uma casa muito bonita e bem quente no inverno.". Entretanto, reparem que a página não nos fornece a opção de incluirmos a `thumbnail`, e para isto será necessário incluir outro argumento, `support`, a ser passado dentro de `$args`.

Ele suportará um título, um editor e uma `thumbnail`:

```

$supports = array(
    'title',
    'editor',
    'thumbnail'
);

$args = array(
    'labels' => $labels,
    'public' => true,

```

```
'description' => $description,  
'menu_icon' => 'dashicons-admin-home',  
'supports' => $supports  
);
```

Salvaremos o arquivo e retornaremos ao dashboard, à página de edição de imóvel. Agora, sim, haverá a opção "Imagen destacada" no canto inferior direito. Acrescentaremos uma imagem e clicaremos no botão "Atualizar". Fica a recomendação para que não deixem de consultar a documentação do Wordpress, pois são muitas informações que não precisarão ser memorizadas.