



GDG Lauro de Freitas

# Desenvolvimento de Sites com WordPress

Do zero ao avançado



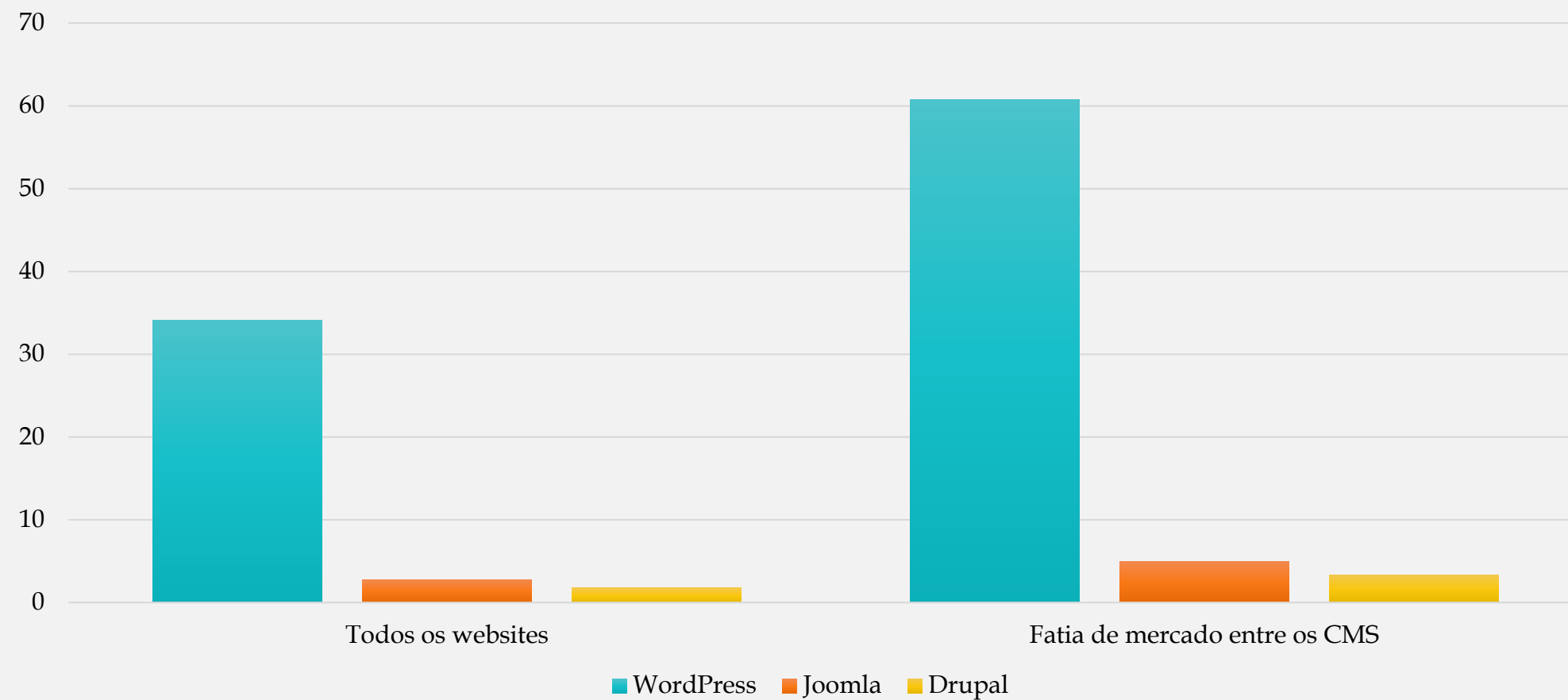
# Apresentação

- Bacharel em Ciência da Computação pela UFBA – 2011
- Desenvolvedor web desde 2008
- Desenvolvedor mobile desde 2012



Flávio Escobar

# Introdução



Fonte: [w3techs.com](http://w3techs.com), 01/07/2019



# Introdução

- Por ser o maior e mais utilizado CMS do mercado, a quantidade de plugins e temas prontos para baixar é gigantesca – bom e ruim
- Pontos positivos
  - (Relativamente) fácil de encontrar funcionalidades prontas que atendem sua demanda
  - Grande variedade de plugins gratuitos
  - Muitos desenvolvedores se preocupam com código bem escrito e atentam para possíveis vulnerabilidades
- Problemas
  - Plugins e temas com código mal escrito e vulnerabilidades de segurança
  - Conflitos entre plugins
  - Plugins que **não fazem tudo** ou que fazem **muito mais** do que você precisa
  - Alguns plugins e temas não são atualizados com uma frequência desejável
  - Muitos plugins oferecem apenas um pequeno conjunto de funcionalidades. A versão completa é paga

# Introdução

---

- Muitas pessoas criam sites com WordPress, porém grande parte deles é um amontoado de plugins e temas de terceiros
- Quando um plugin não faz exatamente o que você precisa, caso você o altere e depois uma atualização seja lançada, você pode perder todas as suas alterações. Mesmo que não perca, elas podem não funcionar mais na nova versão
- Se o tema/plugin não foi escrito por você ou sua equipe, você não tem controle total sobre o código

# Premissas

- O código HTML+CSS já está pronto, feito por alguém do front-end
  - Para a apresentação, usarei uma landing page do site [onepagelove.com](https://onepagelove.com)
- Ambiente de desenvolvimento: XAMPP (Apache + MySQL + PHP)
- MySQL Workbench para gerenciar os bancos de dados
- Ambiente de produção: cPanel, acesso SSH

# Versionamento - Git



GitHub



GitLab

# Versionamento - Git



GitHub Desktop



Git for Windows



# Download e instalação do WordPress

[br.wordpress.org](https://br.wordpress.org)



- Criar banco de dados
- Instalar WordPress
- Renomear pasta wp-content
- Ativar modo debug no ambiente de desenvolvimento
- Não commitar wp-config.php

# Criando um tema WordPress – primeiros passos

- Fazer login no WordPress e navegar para **Aparência > Temas**
- Criar nova pasta: `app/themes/gdg-lauro`
- Obrigatório
  - **style.css** - É neste arquivo que inserimos as informações do nosso tema (nome, autor, link, ...)
  - **index.php** - Inicialmente pode ser um arquivo PHP vazio
- Opcional
  - **screenshot.png**
    - JPG, JPEG e GIF também são aceitos, mas não são recomendados pela documentação oficial
    - Dimensões recomendadas: 1200x900
- Ativar!

# Criando um tema WordPress – próximos passos

- Prevenir acesso direto aos arquivos do tema
- header.php
  - Se seu site tiver algum componente de **topo** que aparece em todas as páginas (ex.: menu principal, logo, etc.), mantenha-o neste arquivo
  - `wp_head()`
  - `body_class()`
- footer.php
  - Se seu site tiver algum componente de **rodapé** que aparece em todas as páginas (ex.: menu de rodapé, copyright, etc.), mantenha-o neste arquivo
  - `wp_footer()`
- home.php
  - `get_header()`
  - `get_footer()`

# Criando um tema WordPress – próximos passos

- functions.php
  - `THEME_PATH` e `IMAGES_PATH`
  - Definir `gdg_get_page_title()`
- header.php
  - Adicionar CSS e fontes
  - Utilizar `gdg_get_page_title()`
  - Copiar `<header>` do HTML original
- home.php
  - Copiar tudo entre `</header>` e `<footer>` do HTML original
- footer.php
  - Copiar tudo do `<footer>` em diante, incluindo tags de fechamento do body e do HTML
- Todos os arquivos: corrigir imagens usando `IMAGES_PATH`

# Integrando o layout com o WordPress

- Agora podemos dizer que já temos um tema WordPress, porém ele é estático – seu conteúdo só pode ser alterado via código
- Para tornar o conteúdo dinâmico, utilizaremos alguns recursos, sendo que os principais são:
  - Custom Post Types
  - Theme Customization API (ou “Customizer”)

# Custom Post Types

- [register\\_post\\_type\(\\$post\\_type, \\$args\)](#)
  - labels – plural (name), singular, novo, editar, ver, nome de menu, item pai, etc...
  - slug – o nome que aparecerá na URL
  - dashboard-icon – ícone que será mostrado no menu
    - <http://calebserna.com/dashicons-cheatsheet/>
  - supports – funcionalidades suportadas pelo CPT
    - title, editor, comments, revisions, trackbacks, author, excerpt, thumbnail, ...
  - taxonomies – array de identificadores de taxonomias
    - É necessário registrar essas taxonomias usando **register\_taxonomy()**
  - has\_archive – se o CPT poderá ter uma página de arquivo (listagem)
  - ... São 29 diferentes opções, algumas delas com mais de uma configuração
- Código pode se tornar bastante repetitivo, então criaremos um arquivo **inc/custom-post-types.php** para reutilizar

# Custom Post Types – integrando os palestrantes

- `get_posts()`
  - `post_type`
  - `numberofposts`
  - `orderby` – o default é 'date'
  - `order` – o default é 'desc'
- `get_permalink()`
- `get_the_post_thumbnail_url()`
  - Necessário ativar o suporte a thumbnails: `add_theme_support( 'post-thumbnails' );`
- `wp_reset_postdata()`

# Dever de casa #1

---

- Integrar patrocinadores e depoimentos com os respectivos CPTs
- Criar um novo CPT chamado “Destaques” para as chamadas abaixo da foto principal
  - Integrar este novo CPT ao tema





# Theme Customization API (ou “Customizer”)

- Poderoso, intuitivo para o usuário final, porém **subutilizado**
- Menu do WP > Aparência > Personalizar
- Permite que o usuário altere aspectos do layout imediatamente
- 4 componentes
  - Settings – são as opções de configuração em si, onde os dados ficam salvos
  - Sections – são grupos de settings
  - Controls – são componentes HTML que serão renderizados pelo Customizer para nos permitir alterar as settings
  - Partial – são botões de atalho para tornar a vida do usuário mais simples

# Theme Customization API (ou “Customizer”)

- `$wp_customize->add_setting($id, $args)`
  - Em `$args` normalmente definimos um valor default para a setting
- `$wp_customize->add_section($id, $args)`
  - `$title` – nome a ser exibido na seção
  - `$priority` – ordem de exibição
- `$wp_customize->add_control($id, $args)`
  - `$args` não é utilizado quando `$id` é um objeto de controle ao invés de string
  - Objetos de controle
    - `WP_Customize_Control` – deve-se especificar um tipo (`type`) – text, checkbox, radio, ...
    - `WP_Customize_Image_Control`
    - Outros: `WP_Customize_Color_Control`, `WP_Customize_Upload_Control`, etc...

# Theme Customization API (ou “Customizer”)

- `$wp_customize->selective_refresh->add_partial($id, $args)`
  - O principal argumento a se preocupar é o ‘selector’, um seletor CSS onde o botão de atalho deverá ser exibido
- `get_theme_mod( $setting )`
  - A função que “faz a coisa acontecer” – exibe o conteúdo salvo na setting
- `is_customize_preview()` – evitar conflito com outros JS
- Assim como nos CPTs, não queremos duplicar um monte de código, então criaremos um arquivo `inc/customizer.php` e o reutilizaremos

# Dever de casa #2

- Criar as seguintes sections
  - Foto do Evento
    - Criar setting para alterar foto do evento
  - Ingressos
    - Criar settings para personalizar título, botões e rodapé
  - Redes Sociais
    - Criar uma setting para personalizar o link de cada uma das redes sociais
  - Rodapé
    - Criar uma setting para editar o texto de copyright

# Criando um formulário de contato

- Existem centenas de plugins de formulário de contato
  - Fazem exatamente o que você quer (nem mais, nem menos)?
  - Fazem exatamente **do jeito que você quer?**
  - São bem escritos?
  - Possuem manutenção periódica?
  - São gratuitos?
  - Possuem tradução para os idiomas que você precisa?
  - ... Entre outras perguntas
- Por que não fazemos nosso próprio plugin?

# Criando um plugin

- Criar pasta **gdg-contato** em `app/content/plugins`
- Criar arquivo com mesmo nome dentro da pasta: **gdg-contato.php**
- Escrever comentário com, pelo menos, o nome do plugin
  - **Plugin Name: GDG Contato**
- `add_action` – associa uma função a uma determinada ação do WP
- `add_shortcode` & `do_shortcode`
- `wp_enqueue_scripts`
- `add_menu_page`

# Criando um plugin

- `add_settings_error, settings_errors`
- `settings_fields, do_settings_sections`
- `register_setting`
- `add_settings_section, add_settings_field`

# Escolhendo um plugin para instalar

- O que verificar ao escolher um plugin:
  - Atualizações recentes
  - Quantidade de usuários
  - Compatibilidade com a versão do WP que você está utilizando
  - Atividade nos fóruns (respostas, bugs resolvidos, etc.)



# Avançando ainda mais

---

- Usar SASS (Grunt, Node)
- Criar um script de deploy

# Dicas adicionais

- Mantenha o WordPress sempre atualizado, incluindo seus temas e plugins
- Se usar um script de deploy, estenda-o para gerar o arquivo wp-config.php
- Sempre que possível, escreva seu código dentro dos padrões do WordPress:  
[https://codex.wordpress.org/WordPress\\_Coding\\_Standards](https://codex.wordpress.org/WordPress_Coding_Standards)

# Dúvidas?



Obrigado!

Flávio Escobar

GDG Lauro de Freitas – 06/07/2019

