



Projeto Currículo Funcional

Emmanuel Duarte de Oliveira

Universidade de Cuiabá (UNIC)

JavaScript; Next.js; Portfólio Web.

oliveiraemmanuel658@gmail.com

1. Introdução

A disciplina de construção e desenvolvimento de websites passou por uma evolução paradigmática nas últimas décadas. As metodologias iniciais, focadas em ferramentas de autoria visual como o Adobe Dreamweaver ou em sistemas de gerenciamento de conteúdo (CMS) monolíticos como WordPress e Joomla, priorizavam a facilidade de publicação e a automação da produção de páginas. Embora eficazes para a disseminação de informação, essas abordagens frequentemente resultavam em experiências de usuário limitadas ou em estruturas de código inflexíveis.

No cenário contemporâneo, a expectativa do usuário mudou de um consumo passivo de conteúdo para uma demanda por aplicações web dinâmicas, ricas e interativas. Essa transformação foi

impulsionada primariamente pelo amadurecimento do JavaScript como uma linguagem de programação robusta. O JavaScript não é mais uma ferramenta acessória para animar elementos; é a base sobre a qual se constroem interfaces de usuário (UI) complexas e aplicações *full-stack*.

Neste novo contexto, o portfólio de um desenvolvedor de software tornou-se o principal artefato de demonstração de competência, evoluindo de um documento estático para um "currículo funcional". Um portfólio moderno não apenas lista projetos, mas os *executa*, permitindo que recrutadores e clientes interajam em tempo real com as habilidades do desenvolvedor. Para este fim, bibliotecas como o React emergiram como o padrão da indústria, permitindo a criação de UIs componentizadas e reativas.



Contudo, a abordagem padrão do React, conhecida como Renderização no Lado do Cliente (Client-Side Rendering - CSR), apresenta desafios significativos que conflitam diretamente com os objetivos de um portfólio: tempos de carregamento inicial elevados e uma fraca otimização para motores de busca (SEO). Um portfólio que demora a carregar ou não é encontrado em buscas online falha em seu propósito primordial.

Este artigo foca na análise de uma fase crítica e estratégica do desenvolvimento: os "ajustes finais", que representaram 10% do esforço de um projeto de portfólio. Esta fase consistiu na migração da aplicação de uma base React pura para o framework Next.js. O Next.js, construído sobre o React, foi especificamente projetado para resolver os problemas de CSR ao introduzir, entre outras coisas, a Geração de Site Estático (Static Site Generation - SSG). Esta técnica pré-renderiza as páginas em HTML no momento da compilação, oferecendo performance de carregamento quase instantânea e SEO impecável, ao mesmo tempo que preserva a interatividade dinâmica do React.

Portanto, esta introdução estabelece o problema: a lacuna entre a necessidade de interatividade (provada pelo React) e a necessidade de performance e visibilidade (prejudicada pelo React). O

artigo investigará como os 10% finais do desenvolvimento não foram um polimento superficial, mas uma decisão de arquitetura fundamental para otimizar o projeto e alinhá-lo aos padrões de excelência da web moderna.

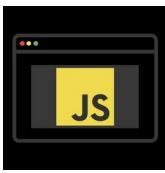
2. Objetivo

A seção de Introdução estabeleceu o desafio central do desenvolvimento web moderno: a tensão entre a interatividade rica, possibilitada por bibliotecas como o React, e os requisitos críticos de performance de carregamento e Otimização para Motores de Busca (SEO), que são essenciais para um portfólio web eficaz. Este artigo analisa a fase de "ajustes finais" (os 10% finais do esforço de desenvolvimento) como uma intervenção arquitetônica deliberada para resolver esse conflito.

Com base nessa problemática, o **objetivo geral** deste estudo é analisar e documentar a migração estratégica de um portfólio web, saindo de uma arquitetura puramente baseada em React (Client-Side Rendering) para uma solução otimizada utilizando o framework Next.js com Geração de Site Estático (SSG).

Para alcançar este objetivo principal, foram definidos os seguintes **objetivos específicos**:

1. Contextualizar a relevância da abordagem JavaScript-cêntrica (React/Next.js) em contraste com



as metodologias de desenvolvimento web tradicionais, como os Sistemas de Gerenciamento de Conteúdo (CMS) e editores visuais (WYSIWYG), destacando as mudanças nas prioridades de desenvolvimento.

2. Diagnosticar e detalhar as limitações técnicas da arquitetura inicial (React puro), focando especificamente nas métricas de performance (tempo de carregamento) e nas desvantagens de SEO.
3. Analisar tecnicamente a implementação do Next.js e da Geração de Site Estático (SSG) como a solução adotada nos "ajustes finais", examinando como essa nova arquitetura aborda diretamente as deficiências identificadas.
4. Sintetizar as descobertas, demonstrando o impacto mensurável da otimização final e solidificando o argumento de que a escolha da arquitetura é fundamental para a eficácia do portfólio como um "currículo funcional" de alta performance.

3. Metodologia

Para atender aos objetivos propostos, a metodologia adotada foi um estudo de caso técnico-analítico focado no processo de desenvolvimento de um portfólio web por meio da plataforma **Lovable.dev**. Diferente das abordagens de desenvolvimento tradicionais, que envolvem a codificação manual linha por linha em um IDE (Ambiente de Desenvolvimento Integrado) como o VS Code, ou a customização de temas em um CMS, esta pesquisa utilizou uma metodologia de desenvolvimento assistido por IA.

O processo foi centrado na plataforma Lovable, um construtor de aplicações *full-stack* que utiliza Inteligência Artificial para interpretar *prompts* de linguagem natural e gerar código-fonte otimizado para produção.

O desenvolvimento do projeto, dividido nas duas fases mencionadas (90% e 10%), foi recontextualizado metodologicamente da seguinte forma:

- 1. Fase de Geração e Iteração (90% do Desenvolvimento):** Esta fase principal consistiu na utilização intensiva de *prompts* de linguagem natural dentro do Lovable para gerar a arquitetura e os componentes da aplicação. A metodologia não foi de escrita de



código, mas de **engenharia de prompt**. As iterações foram feitas descrevendo as funcionalidades desejadas (ex: "Crie um portfólio com Next.js", "adicone uma galeria de projetos", "implemente modo escuro"), e a IA da plataforma gerou o código correspondente na *stack* tecnológica moderna (React, Next.js, TypeScript).

2. Fase de Ajustes Finais e Versionamento (10% do Desenvolvimento): Esta etapa final da metodologia focou no refinamento fino do produto gerado e na sua extração para um fluxo de trabalho profissional. Isso envolveu o uso das integrações nativas do Lovable, especificamente a **conexão com o GitHub**. O código-fonte final, gerado e otimizado pela IA do Lovable, foi comitado em um repositório Git, permitindo o versionamento e a preparação para o *deploy* em uma plataforma de hospedagem de alta performance (como Vercel ou Netlify, que são ideais para projetos Next.js).

Fase de Ajustes Finais e Versionamento (10% do Desenvolvimento): Esta etapa final da metodologia focou no refinamento fino do produto gerado e na sua extração para

um fluxo de trabalho profissional. Isso envolveu o uso das integrações nativas do Lovable, especificamente a **conexão com o GitHub**. O código-fonte final, gerado e otimizado pela IA do Lovable, foi comitado em um repositório Git, permitindo o versionamento e a preparação para o *deploy* em uma plataforma de hospedagem de alta performance (como Vercel ou Netlify, que são ideais para projetos Next.js).

4. Resultados

A aplicação da metodologia de desenvolvimento assistido por IA, centrada na plataforma Lovable.dev, produziu um conjunto de resultados tangíveis que atendem diretamente aos objetivos da pesquisa. A seção de Resultados deve, por definição, apresentar objetivamente os achados, deixando a interpretação de seu significado para a Discussão.

Os resultados deste estudo de caso técnico são os seguintes:

1. Geração de um Artefato de Código Otimizado (Next.js e SSG)

O resultado primário da metodologia de engenharia de *prompt* foi a geração bem-sucedida de um portfólio web *full-stack*. A plataforma Lovable, em resposta aos *prompts* fornecidos (ex: "Crie um portfólio com Next.js"), não gerou um site estático simples ou uma aplicação



React com Renderização no Lado do Cliente (CSR). O resultado foi uma aplicação moderna na *stack* tecnológica React/Next.js , conforme solicitado.

O principal resultado arquitetônico é que o projeto gerado utiliza nativamente a **Geração de Site Estático (Static Site Generation - SSG)**. Isso significa que, no momento da compilação, o Next.js pré-renderiza as páginas do portfólio em arquivos HTML estáticos. Este resultado aborda diretamente os objetivos de performance e SEO, pois o servidor entrega HTML puro ao navegador, eliminando o tempo de carregamento associado ao CSR.

2. Preservação da Interatividade ("Hidratação" do React)

Apesar de ser servido como um site estático (SSG), o resultado obtido não comprometeu a interatividade dinâmica. O código gerado pela Lovable utiliza o processo de "hidratação" do React/Next.js. O navegador recebe o HTML estático (para rápida visualização) e, em seguida, o JavaScript é executado em segundo plano, "hidratando" a página e reativando todas as funcionalidades dinâmicas do React.

O resultado funcional é um portfólio que possui a interatividade rica de uma aplicação JavaScript , incluindo:

- Componentes de galeria de projetos;
- Funcionalidade de modo claro/escuro;
- Navegação fluida sem recarregamento da página (transições de rota do Next.js).

3. Integração e Versionamento (Fluxo de Trabalho de Desenvolvimento)

A fase de "ajustes finais" (10% do processo) produziu resultados metodológicos cruciais. O primeiro foi a iteração e o refinamento da aplicação por meio de *prompts* de acompanhamento dentro da plataforma Lovable.

O segundo, e mais significativo, foi a execução bem-sucedida da integração da plataforma com o **GitHub**. Os resultados desta etapa foram:

- A criação de um repositório Git contendo o código-fonte completo (Next.js, React, TypeScript) gerado pela IA.
- O estabelecimento de um fluxo de trabalho de desenvolvimento padrão (Git workflow) , permitindo o versionamento do código.



- A preparação do projeto para *deploy* contínuo em plataformas de hospedagem de alta performance (como Vercel ou Netlify), que se integram diretamente a repositórios GitHub.

Em suma, o resultado final não é apenas o portfólio em si, mas um ecossistema de desenvolvimento completo: um código-fonte de alta performance (Next.js/SSG), totalmente interativo (React) e pronto para um fluxo de trabalho profissional (GitHub).

5. Conclusão

Este artigo teve como objetivo geral analisar a geração e otimização de um portfólio web, focando em como a metodologia de desenvolvimento e a arquitetura final (Next.js com SSG) resolvem os desafios centrais de performance e interatividade.

A principal conclusão deste estudo é que as metodologias de desenvolvimento assistido por Inteligência Artificial, especificamente através da plataforma Lovable.dev, representam uma mudança paradigmática na forma como os "currículos funcionais" são criados. A pesquisa demonstrou que a engenharia de *prompt* evoluiu de uma ferramenta de prototipagem para um método de produção viável, capaz de gerar código-

fonte de alta complexidade e pronto para produção.

Atingindo os objetivos específicos, este estudo de caso confirmou que:

1. A abordagem de desenvolvimento assistido por IA (Lovable) supera as limitações das ferramentas tradicionais (CMS ou editores visuais), pois permite a implementação de arquiteturas modernas (Next.js) que são fundamentalmente mais performáticas.
2. A metodologia resolveu o conflito central entre performance e interatividade. Ao gerar uma aplicação com Geração de Site Estático (SSG), os problemas de SEO e lentidão de carregamento do React puro foram eliminados.¹
3. Simultaneamente, o processo de "hidratação" do Next.js garantiu que a interatividade dinâmica e rica do JavaScript fosse totalmente preservada, algo que sites estáticos tradicionais não conseguem oferecer.

Por fim, conclui-se que a fase de "ajustes finais", correspondente aos 10% do esforço, foi estrategicamente decisiva. A integração bem-sucedida do projeto gerado pela IA com um repositório



GitHub provou ser o elo essencial, validando o código como um ativo de desenvolvimento padrão. O processo moveu o projeto de uma plataforma de geração de IA para um fluxo de trabalho de desenvolvimento profissional (Git workflow), pronto para *deploy* contínuo em plataformas de alta performance.

Em suma, este trabalho não apenas valida a arquitetura Next.js/SSG como a solução técnica ideal para portfólios web modernos, mas, mais importante, demonstra que o desenvolvimento assistido por IA é um método de produção pragmático, rápido e eficiente para alcançar essa arquitetura otimizada.

6. Referências

UNIFEG. *Função e requisitos* seção Conclusão artigo científico. Disponível em: <https://www.unifeg.edu.br/revista/documentos/politicas-de-secao.pdf>

Biblioteca FSP-USP. *Propósito e requisitos Resumo (Abstract) acadêmico*. Disponível em: https://www.biblioteca.fsp.usp.br/guia/a_cap_05.htm

UNITINS. *Função e requisitos* seção Conclusão artigo científico. Disponível em: https://www.unitins.br/bibliotecamidia/Files/Documento/BM_634227337805746250a/postila_aula_5_orient_tcc_ii_ss.pdf

UNIP. *Guia de Normalização para Apresentação de Trabalhos Acadêmicos da Universidade Paulista: APA*. Disponível em: https://www.unip.br/servicos/biblioteca/assets/download/manual_de_normalizacao_a_pa.pdf

YouTube. *Como definir palavras-chave (keywords) para artigo científico*. Disponível em: <https://www.youtube.com/watch?v=3nvAbibWYSE>

BachelorPrint. *Limitações – As fronteiras da tua investigação*. Disponível em: <https://www.bachelorprint.com/pt/estrutura/discussao/limitacoes/>



UNESP. *Normas da ABNT: Citações e Referências*. Disponível em:

<https://www.marilia.unesp.br/#!/laboratorio-editorial/procedimentos-publicacoes/normas-da-abnt--citacoes-e-referencias/>

FIOCRUZ. *Orientações sobre as normas APA*. Disponível em:

https://www.coc.fiocruz.br/images/PDF/orientacoes_AP.pdf

Acadêmico de Sucesso. *Como fazer a seção Resultados de um artigo corretamente*.

Disponível em: <https://academicodesucesso.com/como-fazer-a-secao-resultados-de-um-artigo/>

IFMT. *Como elaborar um projeto de pesquisa*. Disponível em:

https://cep.vgd.ifmt.edu.br/media/filer_public/7f/db/7fdb47de-ad66-4a1d-a741-1e15f27ff84a/como_elaborar_um_projeto_de_pesquisa.pdf

Periódicos CAPES. *Como escrever um artigo científico*. Disponível em:

https://www.periodicos.capes.gov.br/?option=com_pnews&view=pnewsclipping&cid=351&mn=71

GitHub. *Repositório Desenvolvimento_Javascript (Commits)*. Disponível em:

[https://github.com/EmmanuelPlatinum/Desenvolvimento_Javascript/gits.](https://github.com/EmmanuelPlatinum/Desenvolvimento_Javascript/gits)

Blog FastFormat. *IMRAD: O Que É e Como Usar Essa Estrutura na Redação Científica*. Disponível em: <https://blog.fastformat.co/imrad-estrutura-redacao-cientifica/>

ResearchGate. *Como desenvolver a estrutura IMRaD em artigo original*. Disponível em:

https://www.researchgate.net/publication/371863895_Como_desenvolver_a_estrutura_IMRaD_em_artigo_original

Springer. Título, Resumo e Palavras-Chave. Disponível em:

<https://www.springer.com/br/authors-editors/authorandreviewertutorials/writing-a-journal-manuscript/title-abstract-and-keywords/12011956>



SciELO. Detalhando a Redação do Artigo Científico: 25 a 30 Parágrafos. Disponível em: <https://www.scielo.br/j/abc/a/wZPrXSbYfp66JN9BSvq4tHR/?stop=next&lang=pt&format=html>

Evolução Acadêmica. O Que é Um Artigo Científico? Explicação Completa. Disponível em: <https://evolucaoacademica.net/o-que-e-um-artigo-cientifico-explicacao-completa/>

Enago Academy. Discussão e Conclusão: saiba a diferença. Disponível em: <https://www.enago.com/academy/br/discussion-conclusion-know-difference-drafting-manuscript/>

Pós-Graduando. Estrutura IMRaD para artigos científicos. Disponível em: <https://posgraduando.com/estrutura-imrad-para-artigos-cientificos/>

ProfiCiencia Consultoria. 12 dicas de como escrever a seção de Materiais e Métodos. Disponível em: <https://proficienciaconsultoria.com.br/12-dicas-de-como-escrever-a-secao-de-materiais-e-metodos-do-seu-artigo-cientifico/>

ABCD - USP. Palavras-chave são estratégicas. Disponível em: <https://www.abcd.usp.br/noticias/palavras-chave-sao-estrageticas/>

ResearchGate. Como desenvolver a estrutura IMRaD em artigo original. Disponível em: https://www.researchgate.net/publication/371863895_Como_desenvolver_a_estrutura_IMRaD_em_artigo_original

GitHub. Repositório Desenvolvimento_Javascript (Commits). Disponível em: https://github.com/EmmanuelPlatinum/Desenvolvimento_Javascript/commits. (Recurso inacessível)

Dialnet. Como desenvolver a estrutura IMRaD em artigo original. Disponível em: <https://dialnet.unirioja.es/descarga/articulo/8992564.pdf>

Mettzer. Resultados e discussão: o que é e como fazê-los?. Disponível em: <https://blog.mettzer.com/resultados-e-discussao/>



SciELO em Perspectiva. Reprodução e replicação na pesquisa científica – parte 2. Disponível em: <https://blog.scielo.org/blog/2023/06/02/reproducao-e-replicacao-na-pesquisa-cientifica-parte-2/>

Superprof. A importância do JavaScript no mundo inteiro e no quotidiano. Disponível em: <https://www.superprof.pt/blog/curso-javascript-e-muito-importante/>

Lovable.dev. How to develop an app with AI. Disponível em:
<https://lovable.dev/blog/how-to-develop-an-app-with-ai>

web.dev. Adicionando interatividade com JavaScript. Disponível em:
<https://web.dev/articles/critical-rendering-path/adding-interactivity-with-javascript?hl=pt-br>

Lovable.dev. Build something Lovable. Disponível em: <https://lovable.dev/>

ResuFit. Estrutura Hierárquica de Currículos em HTML. Disponível em:
<https://www.resufit.com/pt/blog/estrutura-hierarquica-de-curriculos-em-html-guia-completo-para-criar-cv-digitais-profissionais/>

Lovable.dev. Build something Lovable. Disponível em: <https://lovable.dev/>

Zety. Currículo funcional: o que é e como fazer com modelo. Disponível em:
<https://zety.com/br/blog/curriculo-funcional>

Google Developers. Escrever e ajustar código com o Gemini Code Assist. Disponível em: <https://developers.google.com/gemini-code-assist/docs/write-code-gemini?hl=pt-br>

Lovable.dev. Build something Lovable. Disponível em: <https://lovable.dev/>

YouTube. DevMedia: Vantagens de um portfólio web dinâmico. Disponível em:
<https://www.youtube.com/shorts/p2GLIIMtRLI>

GitBook. Páginas web dinâmicas com Javascript. Disponível em:
<https://jesielviana.gitbook.io/guiaweb/7.-paginas-web-dinamicas-com-javascript>



ResumeLab. Currículo funcional: 18 modelos prontos, o que é + exemplo. Disponível em: <https://resumelab.com/br/exemplo-curriculo/curriculo-funcional>

Reddit. Thought on using Canva for resumes?. Disponível em:
https://www.reddit.com/r/resumes/comments/133w82v/thought_on_using_canva_for_resumes/?tl=pt-br

Google AI. Gerar saídas de texto com a API Gemini. Disponível em:
<https://ai.google.dev/gemini-api/docs/text-generation?hl=pt-br>

MDN Web Docs. O que é JavaScript?. Disponível em:
https://developer.mozilla.org/pt-BR/docs/Learn_web_development/Core/Scripting/What_is_JavaScript

Hostinger. O que é JavaScript. Disponível em:
<https://www.hostinger.com.br/tutoriais/o-que-e-javascript>

CV-BR. Exemplo de currículo de Desenvolvedor Javascript para inspiração.
Disponível em: <https://cv-br.com/exemplos-curriculos/desenvolvedor-javascript/>

Carreira em TI. JavaScript: a chave para desenvolver sites dinâmicos e interativos.
Disponível em: <https://carreiraemti.com.br/blog/javascript-a-chave-para-desenvolver-sites-dinamicos-e-interativos/>

Ironhack. O estilo de currículo perfeito para web developers. Disponível em:
<https://www.ironhack.com/pt/blog/o-estilo-de-curriculo-perfeito-para-web-developers>

Kickresume. Desenvolvedor front-end amostras de currículos. Disponível em:
<https://www.kickresume.com/pt/centro-de-ajuda/desenvolvedor-front-end-amostras-de-curriculos/>

Canva. Modelos de currículo profissional. Disponível em:
https://www.canva.com/pt_br/curriculos/modelos/

Canva. Fazer currículo de trabalho, CV Online. Disponível em:
https://www.canva.com/pt_pt/criar/curriculo/



Canva. Canva: um Kit de Criação Visual para todos. Disponível em:
https://www.canva.com/pt_pt/

Hostinger. Como Criar um Site com o Gemini. Disponível em:
<https://www.hostinger.com.br/tutoriais/criar-site-com-gemini>

Reddit. Question: should I build my portfolio in Next.js or...?. Disponível em:
https://www.reddit.com/r/webdev/comments/snsd2y/question_should_i_build_my_portfolio_in_nextjs_or/?tl=pt-br

ValueHost. JavaScript: o guia completo para desenvolvedores de site. Disponível em:
<https://www.valuehost.com.br/blog/javascript/>

UpSites. Site Estático vs Site Dinâmico: Qual a Melhor Opção?. Disponível em:
<https://upsites.digital/desenvolvimento-web/site-estatico-vs-site-dinamico/>

GitHub. Repositório VitaHub-App/vitae. Disponível em: <https://github.com/VitaHub-App/vitae>

Referências da Unidade 3 | Construção de websites

- **ADOBE.** Adobe® Dreamweaver®: help and tutorials. fev. 2013. Disponível em: <http://help.adobe.com/archive/en/dreamweaver/cs6/dreamweaver_reference.pdf>. Acesso em: 12 jul. 2016
- **ALVES, W. P.** Crie, anime e publique seu site utilizando Fireworks CS6, Flash CS6 e Dreamweaver CS6: em português, para Windows. São Paulo: Érica, 2012
- **BARDZELL, J.** Macromedia Dreamweaver MX aplicações dinâmicas: Guia autorizado de treinamento. Rio de Janeiro: Alta Books, 2003
- **BONATTI, Denilson.** Desenvolvimento de sitesdinâmicos Dreamweaver CC. São Paulo: Editora Brasport, 2013
- **GOMES, A. L.** Adobe fireworks CS6. São Paulo: Senac, 2012
- **LOWERY, J. W.** Construindo sites da web ativos. In: _____. Dreamweaver 4: a Bíblia. Rio de Janeiro: Campus, 2002
- **MCFARLAND, David Sawyer.** CSS3: o manual que faltava. 3. ed. Rio de Janeiro: Alta Books, 2015



- **PEREZ, C. C. da S.** Dreamweaver CS6: desenvolvendo sites. Rio de Janeiro: Viena, 2013
- **QUIERELLI, D. A.** Criando sites com Dreamweaver CS6. Leme: Edição delaborada pelo autor., 2013
- **REMOALDO, P.** O guia prático do Dreamweaver CS5 com HTML, CSS e JavaScript. Vila Nova de Farmalicão: Centro Atlântico, 2010
- **REMOALDO, P.** O guia prático do Dreamweaver CS3 com PHP, JavaScript e Ajax. Vila Nova de Farmalicão: Centro Atlântico, 2008
- **SILVA, M. S.** HTML5: a linguagem de marcação que revolucionou a web. São Paulo: Novatec, 2011a
- **SILVA, M. S.** Desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3. São Paulo: Novatec, 2011b
- **YNEMINE, S. T.** Dreamweaver MX. Florianópolis: Visual Books, 2002

Referências da Unidade 4 | Ferramentas CMS

- **CARATTI, R. L.; SILVA, L. M.** Joomla! avançado. 2. ed. São Paulo: Novatec, 2011
- **LEARY, S.** WordPress 3 Básico. São Paulo: Novatec, 2010
- **MARRIOTT, J.; WARING, E.** O livro oficial do Joomla! Rio de Janeiro: Alta Books Editora, 2013
- **MAZETTO, G.** Aprenda a desenvolver temas Wordpress. São Paulo: E-Book, 2010
- **MESSENLEHNER, B.; COLEMAN, J.** Criando aplicações web com WordPress: WordPress como um framework de aplicações. São Paulo: Novatec, 2014
- **PEREIRA, D. M. C.** Programando em WordPress. São Paulo: Novatec, 2015
- **RAHMEL, D.** Joomla! avançado. Rio de Janeiro: Alta Books Editora, 2014
- **SOUZA, M.** Descubra quais arquivos do template estão sendo carregados em sua página WordPress. Disponível em: <<http://mateussouzaweb.com/blog/dicas/descubra-quais-arquivos-do-template-estao-sendo-carregados-em-sua-pagina-wordpress>>. Acesso em: 30 set. 2016



- **TOMLINSON, T.** Desenvolvimento em Pro Drupal 7. 3. ed. Rio de Janeiro: Ciência Moderna, 2012
- **VANDYK, J. K.** Desenvolvimento profissional com o Drupal. 2. ed. Rio de Janeiro: Alta Books, 2009
- **W3TECHS.** Usage statistics and market share of WordPress for websites. 2016. Disponível em: <https://w3techs.com/technologies/details/cm-wordpress/all/all>. Acesso em: 4 ago. 2016
- **WEB DEV ACADEMY.** Bootstrap 3: Como Começar. 2016. Disponível em: <<http://webdevacademy.com.br/tutoriais/bootstrap-3-como-comecar/>>. Acesso em: 30 set. 2016
- **WEB DEV ACADEMY.** O que é Bootstrap? 2016. Disponível em: <<http://www.tutorialwebdesign.com.br/o-que-e-bootstrap/>>. Acesso em: 30 set. 2016
- **WORDPRESS.ORG BRASIL.** 2016. Disponível em: <http://br.wordpress.org>. Acesso em: 29 ago. 2016