

Instituto Federal de Brasília - IFB (Campus Taguatinga)  
Bacharelado em Ciências da Computação  
Paradigmas de Programação  
Estudantes: Marcos B. Campos e Nathália Pinheiro  
Professor: Roberto Fontes

# Plano de Projeto

## Trajetória para ser um C.C.

### - Planejamento

Itens de um Plano de Projeto	Conteúdo
1. Introdução	Contém uma descrição dos objetivos do documento, o público ao qual ele se destina e em linhas gerais o propósito do projeto a ser desenvolvido. Pode adicionalmente conter termos e abreviações usadas, além de informar como o plano deve evoluir.
2. Escopo do projeto	Esta seção descreve em linhas gerais o projeto a ser desenvolvido, comunicando o propósito do mesmo, e a importância do projeto para todas as partes envolvidas. O escopo do projeto que será executado é apresentado com uma descrição dos requisitos técnicos (isto é, os requisitos do produto a ser desenvolvido) que podem ser funcionais, não funcionais (desempenho, usabilidade, portabilidade, confiabilidade, etc.) e tecnológicos (tecnologia a ser utilizada). Também, apresentam-se requisitos não técnicos (como, por exemplo, treinamento) e o escopo não contemplado (que descreve quais funcionalidades não fazem parte do escopo do projeto).
3. Organização do projeto	Apresenta-se uma descrição da estrutura organizacional do projeto, incluindo organograma e a descrição de papéis e responsabilidades.
4. Equipe e infra-estrutura	Contém descrição da equipe e da infra-estrutura utilizada para o desenvolvimento do projeto, incluindo: pessoal, equipamentos, ferramentas, software de apoio, materiais, dentre outros. Isto visa garantir uma estrutura adequada para a execução das atividades previstas no plano. Nesta seção também é apresentada o planejamento da alocação de pessoal no projeto.

<b>Itens de um Plano de Projeto</b>	<b>Conteúdo</b>
5. Acompanhamento do projeto	Esta seção do plano de projeto relaciona os momentos para realização das atividades de verificação do projeto, as quais poderão ser feitas pela equipe técnica das instituições envolvidas (desenvolvedora e cliente), e também a forma como estas atividades serão realizadas. Estas atividades incluem a realização de reuniões e geração de relatórios descrevendo informações sobre o progresso do projeto.
6. Marcos do projeto	Contém uma descrição de marcos (milestones) importantes do projeto (incluindo as datas de início e fim do projeto), bem como os artefatos que serão entregues pela empresa desenvolvedora nestes marcos, quando aplicável. Apenas marcos relevantes devem ser listados, ou seja, aqueles que contribuirão para a medição do desempenho do projeto. Por exemplo: reuniões de revisão, apresentação de protótipos ou realização de testes de aceitação. Note que é possível inserir uma visão do cronograma do projeto neste item, destacando apenas os marcos importantes e suas datas alvo.
7. Gerência de riscos	Os riscos identificados para o projeto estão detalhados e monitorados nos relatórios de progresso. Exemplos de riscos compreendem: risco de pessoal, risco tecnológico e de escopo, dentre outros. Um caso de risco de escopo é a falta de clareza na definição do escopo de projeto, que pode resultar em inúmeras solicitações de mudança de escopo
8. Qualidade do produto (ou sistema)	Informa-se a metodologia de desenvolvimento adotada no projeto. Caso, por exemplo, alguma ferramenta específica de desenvolvimento venha a ser utilizada no projeto, isso deve ser descrito neste item. Adicionalmente, informam-se como os artefatos serão gerados por este projeto, os padrões adotados, formatos dos arquivos e templates a serem empregados. Também, neste item, costuma-se informar os critérios de aceitação do projeto.

Itens de um Plano de Projeto	Conteúdo
9. Testes do produto	Este item apresenta uma descrição do projeto de testes do projeto, incluindo detalhamento da estratégia de implementação dos testes, com estágios e tipos de testes a serem realizados para garantir a conformidade do produto com as especificações de requisitos funcionais, não funcionais e requisitos de aceitação do projeto.
10. Referências	Apresenta-se uma relação dos documentos pertinentes ao projeto.

## 1. Introdução

O projeto consiste em uma trajetória para as pessoas que almejam ser Cientista da Computação. Esta trajetória é baseada nas grades curriculas de faculdades renomadas como Harvard, Oxford, UnB, IfB, dentre outras. Além disso, há a disponibilização de conteúdos gratuitos para cursos, vídeo-aulas e livros referentes a área da Computação. Têm como objetivo facilitar a auto-aprendizagem de autodidatas ao mesmo tempo em que providencia um conteúdo linear das matérias necessárias para entendimento fundamental da área de interesse.

## 2. Escopo do projeto

Em uma visão técnica, o trabalho envolveu as tecnologias atuais do mercado de trabalho como o framework progressivo de interfaces de usuário **Vue**, a linguagem multi-paradigma **JavaScript** junto ao **Node** que permite a interação em qualquer sistema operacional.

Em uma concepção geral, o projeto é para *Web* e foi planejado para os interessados na área de Ciências da Computação. A construção do site foi produzida considerando a responsabilidade com os diversas janelas de exibição (tanto do *mobile* quanto do dispositivo *desktop*). O gerenciamento de atividades a serem desenvolvidas foram dispostas por meio de um cronograma dividido entre os dois integrantes do grupo: Marcos e Nathália.

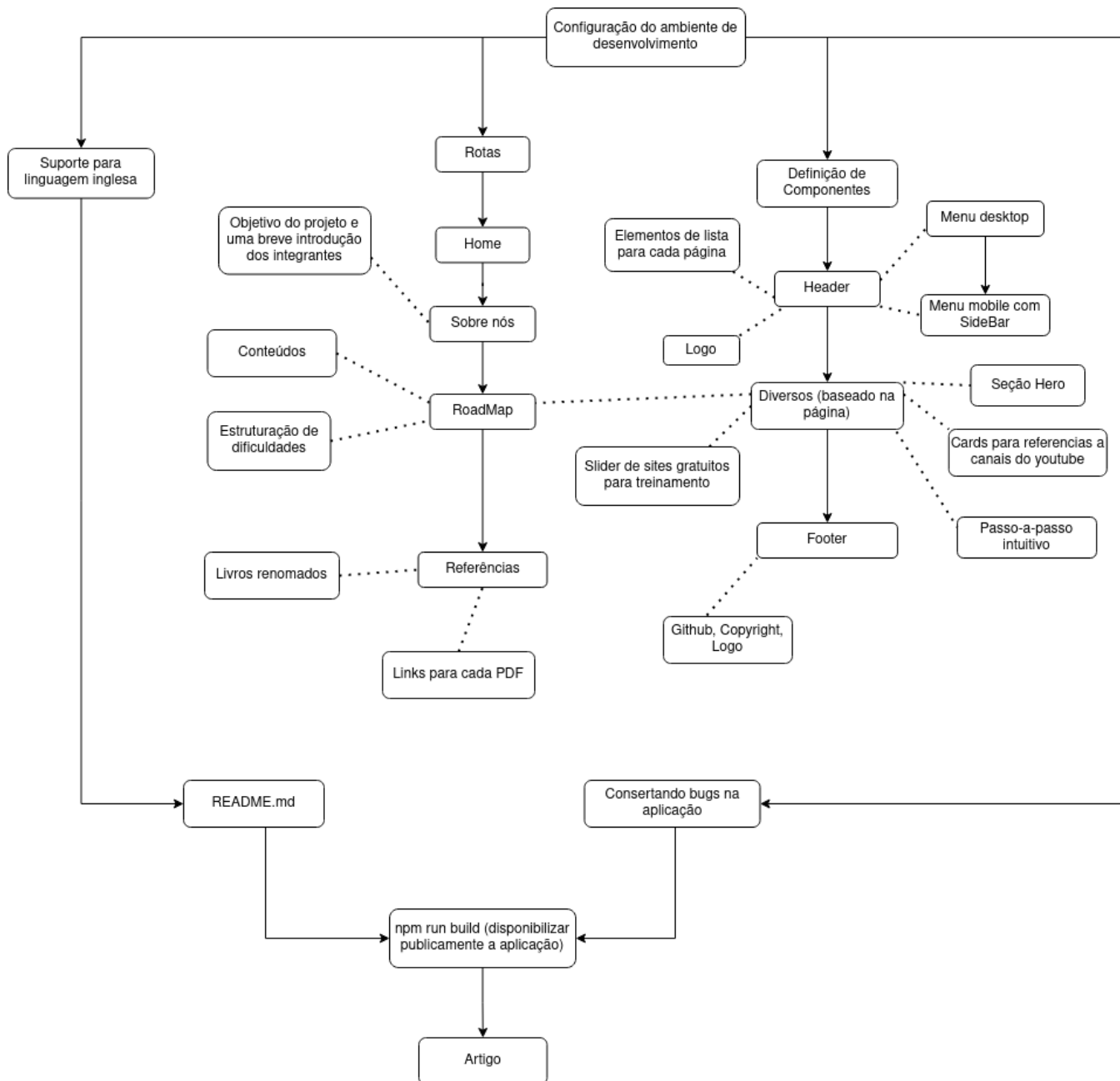
A estrutura do projeto é baseada no padrão de arquitetura MVC e tende a organizar os componentes por páginas envolvidas. Neste projeto utilizou-se 4 páginas, sendo elas, 1 para **página inicial**, 1 para "**sobre nós**", 1 para **referências** e 1 para o **roadmap** .

Os componentes fixos, áqueles que aparecem em todas as páginas, são o *Header* e o *Footer*. Enquanto os outros foram feitos especificamente para cada seção.

### 3. Organização do projeto

A estrutura organizacional do projeto é constituída por um organograma geral da aplicação (sem os detalhes internos de como foi produzido em código), o cronograma redigido na aba *Issues* da plataforma de hospedagem **Github**, além da descrição da aplicação e dos seus usos no arquivo *README.md* hospedado no repositório.

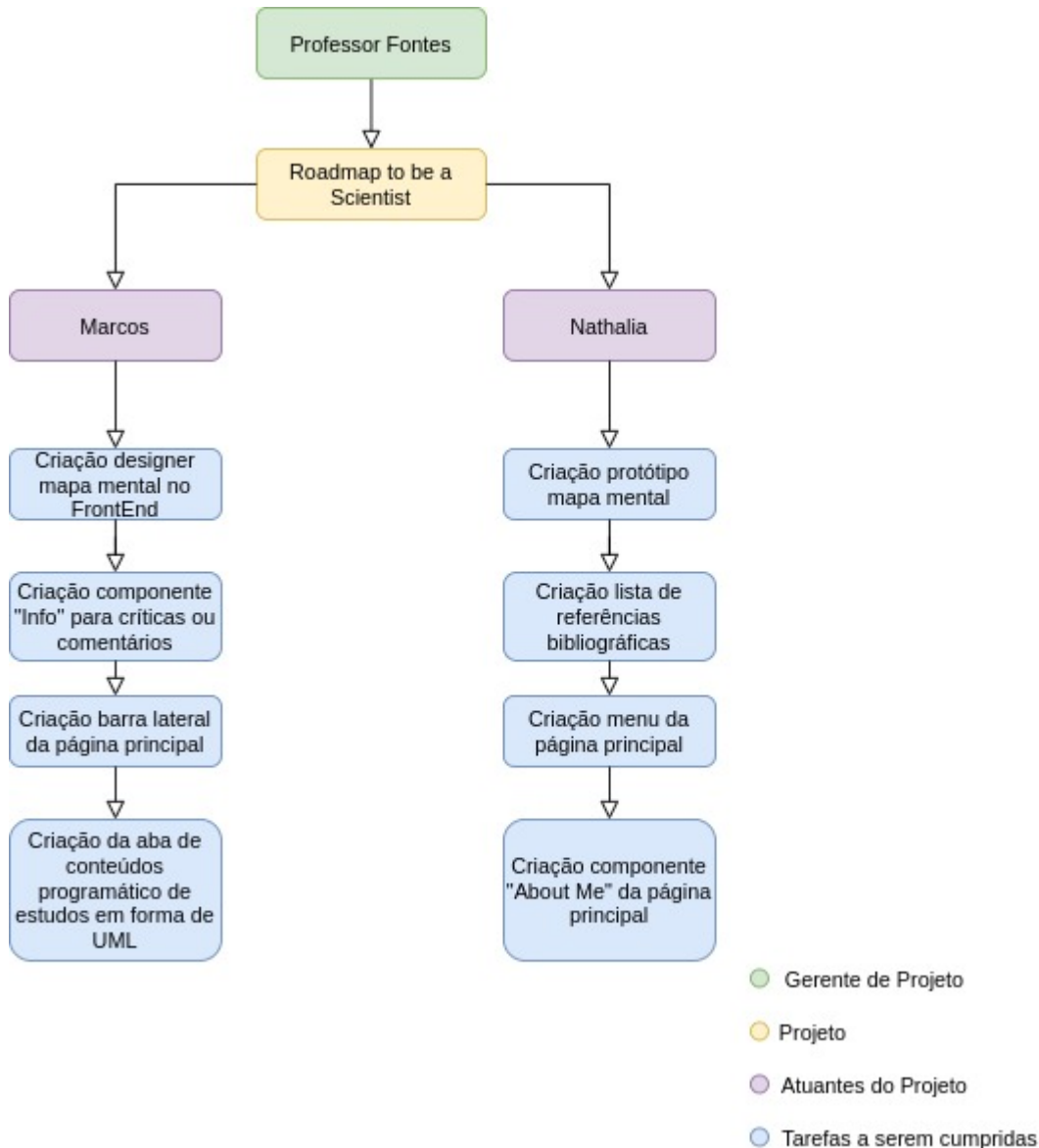
#### - Organograma das atividades



Cada seção que compõem o site é definida na hora da produção, visto que as mudanças são feitas de acordo com a necessidade da sobreposição de informações. Os componentes fixos tiveram prioridade na construção da aplicação, tendo em mente que serão usados em cada rota especificada.

O conjunto de livros da área de conhecimento bem como a disposição de material para providenciar ao público-alvo surgiu a partir da escolha seletiva de autores renomados da área de Computação como Donald E. Knuth, John Von Neumann, Robert Sedgewick, dentre outros.

## - Organograma das responsabilidades



As responsabilidades de cada integrante foram delegadas de forma equivalentes para ambas as partes. Cada atividade têm suas devidas dependências, todavia, podem ser separadas por ordem de importância pois a segregação propõe um modelo de módulos, assim como um **web-service**.

## 4. Equipe e infra-estrutura

A equipe é composta apenas por 2 integrantes: Marcos e Nathália. O equipamento utilizado para criação do site foram os notebooks locais. Ademais, ferramentas como

VSCode para edição de código e o gerenciador de pacote do Node (NPM) providenciaram os softwares fundamentais para execução da aplicação. A preferência pelo framework Vue para Front-End especificamente, foi para fins educativos já que era um dos frameworks que menos tivemos contato.

Softwares de gerenciamento de projeto como o Trello e Obsidian foram utilizados para definir as tarefas e estabelecer uma data limite para cada. Também, para descrever o planejamento interno e propostas para a aplicação enquanto surgiam no desenvolvimento da aplicação.

A integração do sistema de hospedagem chamado Github veio a calhar pela comunicação entre cada *Branch* produzido por ambos desenvolvedores. As atualizações proporcionadas pelo *Pull Request*, uma das funcionalidades do Git de poder subir uma alteração a partir de um *Commit* tornou a transmissão de atividades mais simplória e direta, tornando-se um diferencial tremendo para a construção do projeto.

Desejamos por um time pequeno devido a incomplexidade do projeto, não havia uma exigência maior do que esta. As atividades poderiam ser executadas tranquilamente por apenas dois integrantes, em um curto espaço de tempo.

## 5. Acompanhamento do projeto

As reuniões para acompanhamento do projeto são feitas às quartas-feiras, geralmente na parte da noite. Dependendo da urgência das atividades, pode ser feita aos fins de semana, dias os quais são exclusivos para desenvolver no turno matutino ou vespertino.

A verificação do cronograma e os programas de gerenciamento de projeto (Trello e Obsidian) deixa os desenvolvedores em alerta, enviando mensagens sobre as atividades em andamento.

Relatórios não são gerados, porém, a apuração das alterações em cada *Commit* no **Github** é essencial para que não haja conflitos ou discrepância para com o objetivo do aplicativo.

O controle de testes foram executados nas rotinas, dependências e nos módulos independentes de toda a aplicação. Cada um deles deve respeitar o princípio de integridade e consistência do projeto, isto é, permitir que o usuário tenha uma experiência agradável ao mesmo tempo em que os módulos atende o seu pré-requisito.

## 6. Marcos do projeto

Os marcos do projeto são realizados após o término de criação de um componente e/ou rota. A inspeção unitária de funcionamento e as suas relações de dependência caso haja

são testadas em tempo de produção, previamente, para que seja considerada finalizada a etapa.

A persistência da aplicação provém desta prevenção, é claro que poderá haver bugs mesmo tendo passado por esse processo e, portanto, o teste do produto torna-se preponderante para examinar todas as rotinas, módulos e/ou componentes do programa.

A apresentação de protótipos precede a inspeção, já que para termos um componente é indispensável a conceituação do elemento a ser inserido. O UML pode ser uma ferramenta poderosa nesse estágio, principalmente para esboçar o passo-a-passo do módulo, expressando as correlações entre cada elemento processual.

## **7. Gerência de riscos**

Nessa parte, a auditoria minuciosa de cada pedaço de código em busca de uma falha de segurança ou de más práticas, até mesmo a investigação em blocos de código que possam tornar o desempenho indesejável serão efetuadas.

Caso haja alguma falha relatada nos testes de sistema, em uma visão técnica, os desenvolvedores encontrará uma solução para tal problema.

Em um meio geral, os riscos também são de pessoal ou escopo que consistem respectivamente na inaptidão técnica dos responsáveis pelo projeto e na falta de clareza do escopo do projeto.

A construção de um escopo bem definido é extremamente importante e que, inclusive, tomou a maior parte do tempo. Prender-se ao que foi solicitado como pré-requisito tornou a construção da aplicação direta.

## **8. Qualidade do produto (ou sistema)**

A metodologia adotada foi a *Agile*, que é um método designado para a produtividade, ou seja, é focada na criação do projeto gradualmente, com adaptação ou possibilidades de mudanças construídas direto no processo. Ela é ideal para grandes projetos que não têm um objetivo claro no começo e são sujeitas a mudanças ao longo do processo.

A ideia para escolha de metodologia deve-se ao fato de 95% das organizações empresariais mundiais praticarem o *Agile* no desenvolvimento de softwares. E, para fins educativos, tentamos ao máximo implementar o modelo *Agile* ao decorrer das atividades.

Assegurar que a adaptação e flexibilidade da aplicação era uma das motivações para escolher tal metodologia. Considerando os riscos de que a aplicação não caia na tentação de constantes mudanças, perdendo o foco do principal objetivo.

A cada *sprint* ou fim de módulo, há o tempo de análise e avaliação, aonde mudanças podem ser feitas e novas ideias serem introduzidas para o próximo *sprint*. Um exemplo em nossa aplicação foi a inserção da linguagem estrangeira ao fim da criação do sistema.

## 9. Testes do produto

Os testes são realizados ao fim de cada *sprint* da metodologia *Agile*. A verificação é praticada no módulo para garantir a conformidade do produto na etapa especificada. Como não há a demanda de testes unitários, testes de integração são efetuados para examinar a comunicação entre módulos dependentes, todavia, estes exclusivamente têm de ser preparados no fim de uma rota.

Testes de sistema são exercidos constantemente para evitar que a nova inserção ou alteração de elemento não quebre a aplicação inteira, logo, pode-se dizer que são incluídos ao início de um novo *sprint*.

Testes no ambiente de aceitação também são acometidos em nossa aplicação, garantindo a sustentabilidade e a harmônia dos elementos que compõem o site, em um ambiente o mais próximo possível de produção.

## 10. Referências

A relação de referências bibliográficas documentadas que estão disponíveis para a aplicação:

Alexandrescu, Andrei. Modern C++ Design: Generic Programming and Design Patterns Applied. 2001. Disponível em: [https://github.com/manjunath5496/Computer-Science-Reference-Books/blob/master/comp\(45\).pdf](https://github.com/manjunath5496/Computer-Science-Reference-Books/blob/master/comp(45).pdf). Acesso em: 24 de Novembro de 2021.

Martin, C Robert. Clean Code: A Handbook of Agile Software Craftsmanship. Boston: Alta Books, 2008. Disponível em: [https://github.com/manjunath5496/Computer-Science-Reference-Books/blob/master/comp\(105\).pdf](https://github.com/manjunath5496/Computer-Science-Reference-Books/blob/master/comp(105).pdf). Acesso em: 24 de Novembro de 2021.

Timothy G. Mattson. Patterns for Parallel Programming. Addison Wesley Professional, 2004. Disponível em: [https://github.com/manjunath5496/Computer-Science-Reference-Books/blob/master/comp\(59\).pdf](https://github.com/manjunath5496/Computer-Science-Reference-Books/blob/master/comp(59).pdf). Acesso em: 24 de Novembro de 2021.

Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. P Compilers. Pearson, 1986. Disponível em: [https://github.com/manjunath5496/Computer-Science-Reference-Books/blob/master/comp\(12\).pdf](https://github.com/manjunath5496/Computer-Science-Reference-Books/blob/master/comp(12).pdf). Acesso em: 24 de Novembro de 2021.

Van D. Peter. Expert C Programming. Prentice Hall, 1994. Disponível em: [https://github.com/manjunath5496/Computer-Science-Reference-](https://github.com/manjunath5496/Computer-Science-Reference-Books/blob/master/comp(12).pdf)



[Books/blob/master/comp\(18\).pdf](#).

Acesso em: 24 de Novembro de 2021.

A. Lew, H. Mauch. Dynamic Programming: A Computational Tool. Springer, 2010.

Disponível em: [https://github.com/manjunath5496/Computer-Science-Reference-Books/blob/master/comp\(186\).pdf](https://github.com/manjunath5496/Computer-Science-Reference-Books/blob/master/comp(186).pdf). Acesso em: 24 de Novembro de 2021.

Michael A. Band, Bradley C. Kuszmaul. Data Structures and Algorithms for Big Databases. Stony Brook University, Tokutet, 2019. Disponível em:

[https://github.com/manjunath5496/Computer-Science-Reference-Books/blob/master/comp\(310\).pdf](https://github.com/manjunath5496/Computer-Science-Reference-Books/blob/master/comp(310).pdf). Acesso em: 24 de Novembro de 2021.

Jean H. Galier. Logic for Computer Science. Dover Publications, 2015. Disponível em:

[https://github.com/manjunath5496/Computer-Science-Reference-Books/blob/master/comp\(440\).pdf](https://github.com/manjunath5496/Computer-Science-Reference-Books/blob/master/comp(440).pdf). Acesso em: 24 de Novembro de 2021.

Donald E. Knuth. The Art of Computer Programming. Addison Wesley Professional, 2011. Disponível em:

[http://broiler.astrometry.net/~kilian/The\\_Art\\_of\\_Computer\\_Programming%20-%20Vol%201.pdf](http://broiler.astrometry.net/~kilian/The_Art_of_Computer_Programming%20-%20Vol%201.pdf). Acesso em 24 de Novembro de 2021

Hanchett Erick, Listwon Benjamim. Vue.js in action. Manning, 2018. Disponível em:

<https://github.com/liyajunforweb/book/blob/master/Vue.js%20in%20Action.Pdf>. Acesso em 24 de Novembro de 2021

Crockford Douglas, O Melhor do JavaScript. Alta Books, 2008. Disponível em:

<https://www.skoob.com.br/livro/pdf/o-melhor-do-javascript/livro:64755/edicao:71483>.

Acesso em 24 de Novembro de 2021