



# Github: Branches, Markdown e Funcionalidades de Projetos

Bem-vindos a este guia sobre branches no Github, Markdown e as funcionalidades essenciais para gerenciar projetos com eficiência. Vamos explorar os conceitos, ferramentas e melhores práticas para otimizar seu fluxo de trabalho.



by Victor Santos Rohod



# O que são branches no Github?

Branches são como ramificações no desenvolvimento de um projeto. Elas permitem que você trabalhe em novas funcionalidades ou correções de bugs em separado, sem afetar o código principal.

**1**

## **Isolando Mudanças**

Crie branches para testar ideias, implementar novos recursos ou corrigir erros sem afetar o código principal.

**2**

## **Trabalho Colaborativo**

Permite que vários desenvolvedores trabalhem em diferentes partes do projeto simultaneamente, sem conflitos.



# Importância de um bom fluxo de branches

Um fluxo de branches bem estruturado garante organização, evita conflitos e facilita o gerenciamento de versões do projeto.

1

## Branch Principal

Mantem o código principal do projeto estável, com versões liberadas.

2

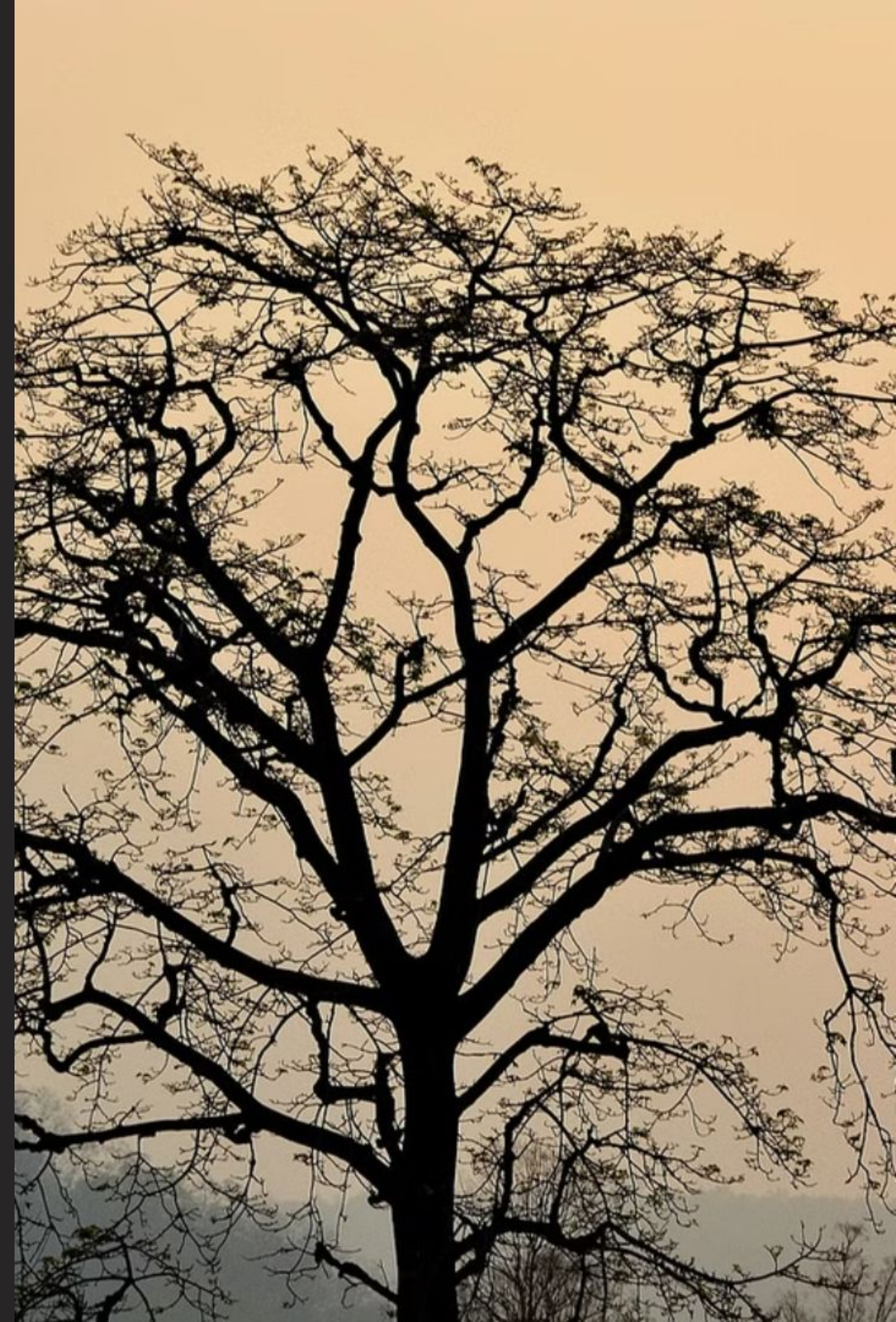
## Branches de Funcionalidades

Desenvolvimento de novas funcionalidades separadamente do código principal.

3

## Branches de Correções

Para solucionar bugs ou problemas específicos no código.



# Melhores práticas para criar e gerenciar branches

A adoção de boas práticas simplifica o trabalho, reduz erros e promove colaboração.

## Nomes Descritivos

Nomes claros que identificam o propósito da branch, como "feature-login" ou "bugfix-issue-123".

## Branches Pequenas

Crie branches focadas em uma tarefa específica, facilitando a revisão e o merge.

## Merge Regular

Integre as branches de forma frequente, garantindo que as mudanças sejam combinadas e testadas.

# Markdown: uma linguagem de marcação simples e poderosa

Markdown é uma linguagem de marcação leve, ideal para escrever conteúdo no Github e em outros ambientes.

## Formatação de Texto

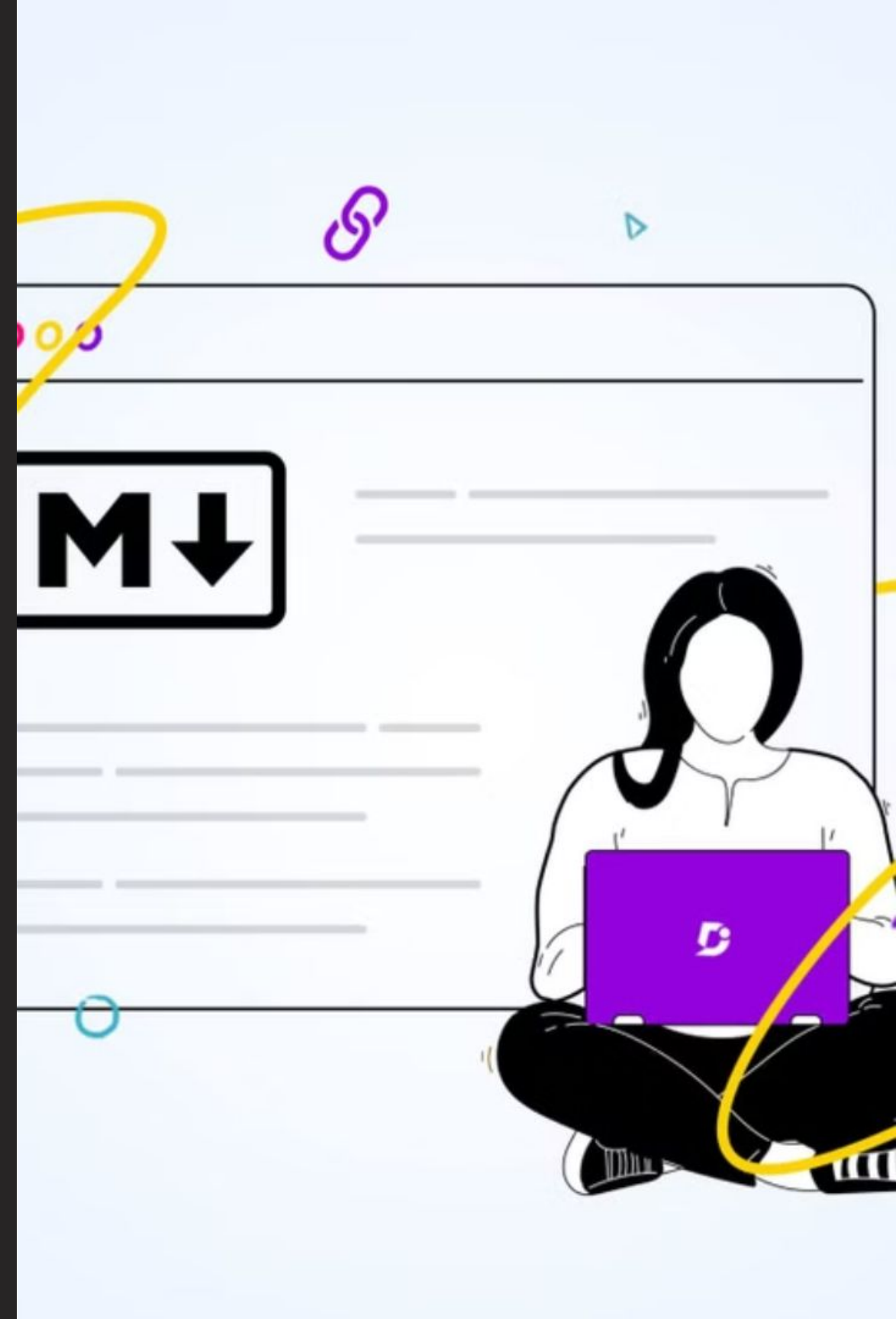
Crie títulos, parágrafos, listas, negrito, itálico e código com uma sintaxe simples.

## Imagens e Links

Insira imagens e links de forma fácil, utilizando a sintaxe Markdown apropriada.


## Tabelas

Crie tabelas para organizar informações, usando uma sintaxe intuitiva.





# Utilizando Markdown em projetos do Github



```
Over the last couple of months, I've been writing a lot of [Markdown](http://daringfireball.net/projects/markdown/), primarily for these blog posts. I've tried various text editors, but haven't been entirely happy with the writing experience until last week when I discovered two truly helpful packages for [Sublime Text](http://sublimetext.com). Here's the setup which I used to write this very post and which I'll be using to write Markdown from now on.
```

## Writing Markdown in Sublime Text

I've been a long-time fan of the [Sublime Text](http://sublimetext.com) editor. It's lightweight, fast, and highly customizable. Plus, it simply looks gorgeous on a MacBook with a retina display.

If I need to sit down and focus on my writing, I like to enter Sublime's *distraction free mode*, which then switches to fullscreen and hides everything but the current file's content. That makes it a lot easier to *stay focused* on the task of writing:

You can either click on *View | Enter Distraction Free Mode* in the menu bar or use the (slightly uncomfortable) *CTRL+CMD+SHIFT+F* keyboard shortcut.

Additionally, I usually activate the *Do Not Disturb* mode to prevent Mac OS X from showing Growl notifications in the upper right corner of the screen.



## Formatação de Texto

Crie títulos de nível 1, 2 e 3 com #, ## e ###. Utilize *texto* para itálico e **texto** para negrito. Use `código` para exibir código inline.

## Listas

Crie listas não ordenadas com \*, -, ou + seguido de um espaço e o item da lista. Para listas ordenadas, use números seguidos de um ponto e o item da lista.

## Links e Imagens

Insira links com [texto do link](url do link) e imagens com ![texto alternativo](url da imagem).

## Bloco de Código

Para exibir um bloco de código, use três crases (```) antes e depois do código.

# Funcionalidades avançadas do Github para gerenciar projetos

O Github oferece ferramentas avançadas para auxiliar no gerenciamento de projetos e na colaboração entre equipes.



## Project Boards

Visualização de tarefas, sprints e progressos do projeto.



## Code Review

Revisão do código por pares para garantir a qualidade e melhores práticas.



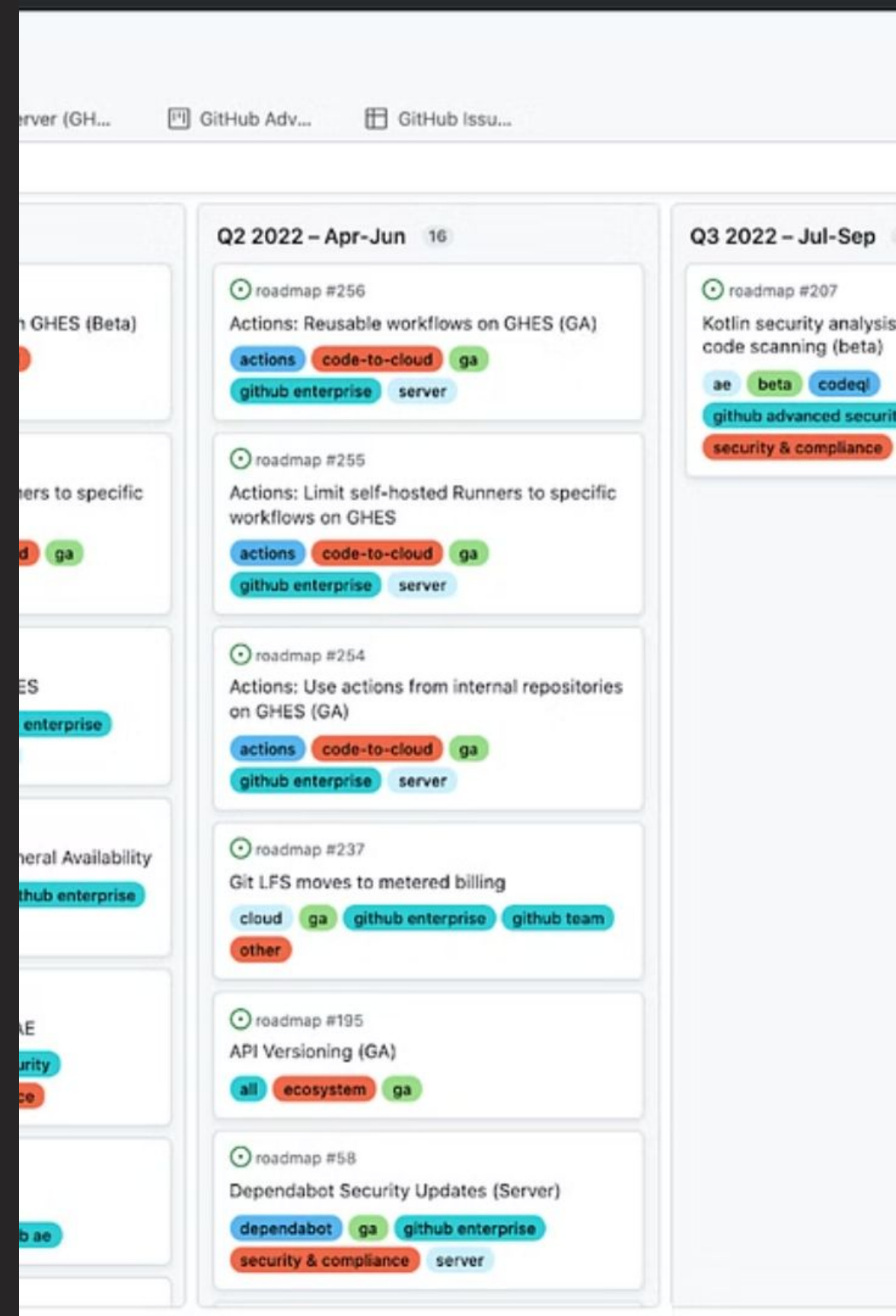
## Milestones

Definição de metas e prazos para o desenvolvimento do projeto.



## Discussions

Fórum para discussões sobre o projeto, decisões e brainstorming.



# Boas práticas de organização e colaboração em projetos

Comunicação clara, organização eficiente e colaboração eficaz são fundamentais para o sucesso de qualquer projeto.

1

## Comunicação

Utilize ferramentas como issues, pull requests e discussions para manter a equipe informada.

2

## Revisão de Código

Avalie e revise o código de outros membros da equipe para garantir qualidade e consistência.

3

## Documentação

Documente o projeto de forma clara e concisa para facilitar a compreensão e a manutenção.

