

# Planejamento do Minicurso De Git e GitHub

Diogo L.C. Felipe

## Introdução

O contexto das empresas de tecnologia ao redor do mundo hoje tem tendido cada vez mais para distribuição de seus profissionais e bases de código. Trabalho remoto em grandes equipes é uma prática que se torna cada dia mais comum nas principais empresas ditadoras de tendências no mercado, como é o caso do Facebook, Amazon, Spotify, Uber, Google etc. Tal crescimento deve-se muito ao fato de em 2005, para facilitar o desenvolvimento em código aberto do Kernel do Linux, Linus Torvalds ter criado o sistema de controle de versão distribuído denominado Git.

O sistema ganhou força entre seus concorrentes por alguns motivos, dentre eles o fato de ser mais leve, simples de usar e de código fonte aberto. Desde então cada vez mais empresas ao redor do mundo adotam o sistema em seus projetos, tornando tal conhecimento mandatório para qualquer profissional que deseja adentrar o mercado de trabalho ou ascender em carreira.

É importante notar que mesmo os profissionais autônomos e que não possuem uma equipe robusta de desenvolvedores se beneficiam enormemente do git. A portabilidade e controle de versionamento proposto pela tecnologia torna os profissionais consideravelmente mais produtivos, consequentemente aumentando seu valor no mercado e sua capacidade de trabalhabilidade.

Existem ainda várias plataformas que utilizam o git, através de interfaces mais convidativas, para facilitar o versionamento e armazenamento de código. Dentre elas destaca-se o GitHub. O GitHub é muitas vezes usado como parte fundamental do currículo de qualquer programador. Nele é possível armazenar e contribuir com códigos abertos, além de repositórios privados, gerando assim um histórico profissional que serve como portfólio para programadores ao redor do mundo. Também não é preciso dizer que as maiores empresas do mundo matem partes cruciais de seus códigos na plataforma, tornando-a essencial para os desenvolvedores que almejam emprego entre elas.

Dito isso, fica evidente a necessidade de aprender git e GitHub o mais rápido possível na vida acadêmica e profissional de um desenvolvedor. Com esses conceitos é possível retirar maiores benefícios das aulas e adentrar com maior facilidade em projetos acadêmicos ou profissionais ainda na universidade. O maior problema enfrentado nesse ponto é a falta de cursos específicos para a tecnologia nos primeiros semestres das universidades. Dessa forma, o atual planejamento vem como tentativa de resolução do problema, fornecendo um curso sólido e prático que venha ensinar não somente git e Github, como também mostrar o mercado de trabalho e introduzir estudantes na comunidade de desenvolvedores mundial.

## Objetivos Gerais

O presente minicurso não se atém somente a conceitos técnicos e tem por objetivos:

- Ensinar de forma prática e sólida os principais conceitos de versionamento com git;
- Ensinar os principais comando em linha de comando do git;
- Mostrar as principais ferramentas da interface web do GitHub;
- Introduzir o aluno na comunidade de desenvolvedores internacional;
- Mostrar os conceitos de código aberto.

## Objetivos Específicos

Como objetivos específicos, no decorrer das aulas será buscado:

1. Falar sobre a motivação de usar o git e GitHub;
2. Determinar a diferença entre git e GitHub;
3. Explicar a importância e os conceitos de distribuição e versionamento de código;
4. Explicar o conceito de repositório;
5. Explicar o conceito de branch;
6. Explicar o que são repositórios locais e remotos;
7. Mostrar o processo de instalação do git;
8. Mostrar o processo de criação de um perfil no GitHub, bem como a criação de um repositório;
9. Ensinar a criar um repositório local e remoto;
10. Ensinar a clonar e adicionar origem a um repositório;
11. Ensinar a buscar alterações de terceiros de um repositório remoto para um local;
12. Explicar como funcionam os commits;
13. Ensinar a enviar alterações de um repositório local para um remoto;
14. Mostrar, via GitHub, o funcionamento de pull requests;
15. Ensinar o processo de resolução de conflitos;
16. Mostrar o processo de visualização do histórico de contribuições;
17. Códigos úteis para utilização como indicadores do crescimento e organização do código.

## Estrutura, Metodologia e Recursos

O minicurso será dividido em três aulas, que serão descritas abaixo, com duração de 3 horas e um intervalo de 15 minutos entre após a primeira 1 hora e meia. Além disso, irão ocorrer pausas controladas para discussão, esclarecimento de dúvidas e captação de feedback.

Além disso, será confeccionado uma apostila com todo conteúdo das aulas. A distribuição da mesma será feita através do GitHub e ela servirá de apoio às aulas, sendo distribuída sempre um dia antes da aula a qual a mesma se refere.

### Aula 1

Objetivos: A primeira aula abordará os tópicos de objetivos específicos listados acima de 1 ao 8. Tais tópicos servem como excelente introdução e base para os conceitos mais avançados que serão abordados nas próximas aulas;

Metodologia: Aula predominantemente expositiva com momentos de prática de média frequência na segunda metade da aula;

Recursos: Slide, Quadro Branco, Piloto, Apagador, Computadores;

Referências:

1. Danic, M., Aljord, P. and Chacon, S. (2009). *Pro Git*. 2nd ed. Berkeley: Apress L.P.
2. WHY GIT FOR YOUR ORGANIZATION. Disponível em: <https://www.atlassian.com/git/tutorials/why-git>. Acesso em 26 Fev. 2020.

### Aula 2

Objetivos: A segunda aula irá abordar os objetivos específicos de 9 a 14.

Metodologia: A aula será predominantemente prática. Os alunos irão aprender na prática todos os conceitos envolvidos e desenvolver repositórios com alguns exercícios relacionados.

Recursos: Slide, Quadro Branco, Piloto, Apagador, Computadores;

Referências:

1. Danic, M., Aljord, P. and Chacon, S. (2009). *Pro Git*. 2nd ed. Berkeley: Apress L.P..

### **Aula 3**

Objetivos: A segunda aula irá abordar os objetivos específicos de 15, 16 e 17.

Metodologia: Na aula será desenvolvido um projeto colaborativo para colocar em prática todos os conceitos estudados nas aulas anteriores. O projeto será hospedado no GitHub e terá colaborações dos alunos. Para evitar que algum aluno não consiga acompanhar a aula, não será desenvolvido nenhum código real, ao invés disso, será feito um repositório para armazenamento e atualização de receitas culinárias colaborativas;

Recursos: Slide, Quadro Branco, Piloto, Apagador, Computadores;

Referências:

1. Danic, M., Aljord, P. and Chacon, S. (2009). *Pro Git*. 2nd ed. Berkeley: Apress L.P.;