

Vantagens: GIT x GITHUB

Git

Imagine que, há alguns meses, você desenvolveu um site que possuía determinada função. Com o passar do tempo, essa função perdeu sua utilidade e você foi instruído a retirá-la do código fonte deste site. Após seis meses, por uma questão de estratégia de negócio, essa função deve ser novamente implementada. Para programadores(as) que utilizam o GIT, reincluir a funcionalidade é uma tarefa simples: basta buscar, em seu repositório, a versão que a contempla e retomá-la. Para aqueles(as) que não utilizam, talvez seja necessário reescrever o código desta função, consumindo novamente o tempo gasto para desenvolvê-la da primeira vez. Diante desta situação, podemos concluir que uma das maiores vantagens de usar o GIT é a economia de tempo e recursos, uma vez que a consulta de diferentes versões de uma mesma aplicação é muito recorrente no trabalho do(a) programador(a).

Outro grande benefício do GIT é justamente o fato de ele ser um sistema **distribuído**. Isso significa que, diferentemente de outros sistemas de controle de versionamento populares na época em que foi lançado, o GIT possui repositórios, e não somente um único local com o histórico de seu trabalho.

Para auxiliar o trabalho em equipe, recursos como o fluxo de desenvolvimento do **Gitflow** também trazem muitos benefícios, inclusive porque pode ser acessado por qualquer membro do time em qualquer lugar. Ou seja, por ser o mais rápido controle de versionamentos existente hoje no mercado, o GIT vale a pena porque proporciona a seus usuários e estudantes de programação grande otimização de tempo e recursos.

GitHub

Para os profissionais da área de tecnologia, as vantagens de utilizar o GitHub são inúmeras.

A maior delas, talvez, seja a oportunidade de aprender com programadores e programadoras mais experientes que você, especializado nas mesmas ou em diferentes áreas. Estas conexões são riquíssimas para os estudantes.

Outros benefícios são:

- Possibilidade de acompanhar e colaborar com projetos de diferentes equipes;
- Aprender programação na prática ao observar o avanço do desenvolvimento de aplicações de terceiros;
- Participar de discussões a respeito de novas tecnologias;
- Obter auxílio de outros programadores para resolver problemas relacionados a seus projetos;
- Controlar as diferentes versões de um código com armazenamento em nuvem;
- Registrar ações e projetos desenvolvidos por você em uma espécie de portfólio online, etc.