

Grupo: Samuel Carvalho, Hamilton Gomes, Luiza Graciano, Thayná Isabelly, Carla Rebeca e Carlos Eduardo.

1. Quem criou o Git e o GitHub (Biografia completa)? E a diferença entre eles?

- Quem criou o GitHub?

Chris Wanstrath, J. Hyett, Tom Preston-Werner e Scott Chacon.

A empresa criadora do GitHub foi a GitHub Inc., criada em 2007 em São Francisco nos EUA.

- Quem criou o Git?

Linus Torvalds - Engenheiro de Software, formado em Ciência da Computação, nascido na Finlândia, desenvolvedor do núcleo Linux, utilizado nos SOs Linux, Android e Chrome.

Além de criar o Git, criou o aplicativo para planejamento e registro de mergulho Subsurface.

- Diferença entre Git e GitHub:

O Git é um sistema de controle de versões distribuído, usado para desenvolvimentos não lineares e colaborativos.

O GitHub é uma plataforma de hospedagem de código-fonte e arquivos versionados usando Git.

Os usuários podem compartilhar projetos com outros desenvolvedores que podem também fazer alterações e melhorias.

2. Para que serve? Quem utiliza, o que faz, de onde vem, do que se alimenta?

O Git serve para registrar quaisquer alterações realizadas no código, funciona como um sistema de versionamento de códigos, possibilitando que o programador possa voltar a versões anteriores do código.

O GitHub serve como uma rede social de programadores, na qual é possível interagir, seguir, visualizar código de outros programadores e compartilhar os próprios códigos.

O Git e o GitHub são utilizados por programadores do mundo todo. Essas ferramentas são responsáveis por possibilitar um gerenciamento contínuo e pontual das versões de códigos de um determinado projeto.

3. Qual o principal objetivo da plataforma:

a. Git

Registrar o histórico de modificações de qualquer tipo de arquivo, como um sistema de camadas, onde a camada 0 seria o arquivo base e as demais camadas seriam o mesmo arquivo com modificações. Contudo esse não é o único objetivo do Git, podendo ser usado principalmente no desenvolvimento de software por se tratar de um mecanismo que permite trabalhar de forma remota ou sem conexão, assim os desenvolvedores podem sincronizar suas cópias com modificações junto ao arquivo inicial de cada projeto. Além de que o Git possui diversos mecanismos que ajudam no desenvolvimento e junção de cada projeto, com isso a popularidade do Git cresce cada vez mais no meio tecnológico.

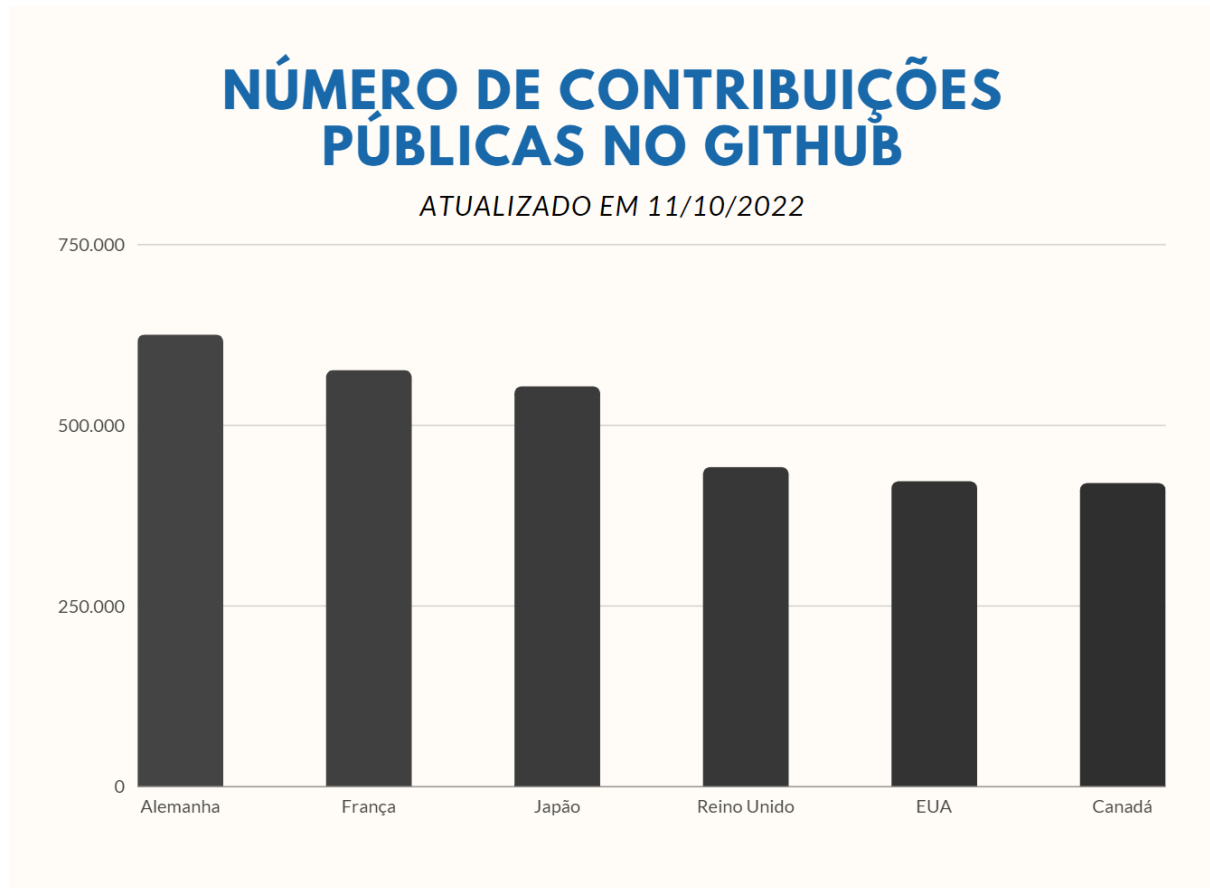
b. Github

Por ser uma plataforma de armazenamento em nuvem que possui um sistema de controle de versões, o Git, permite que os desenvolvedores colaborem em realizar mudanças em projetos compartilhados uma vez que possuem um armazenamento detalhado do progresso. A interface do Github é mais fácil de ser usada e entendida, por isso atrai mais pessoas que estão iniciando na área e não possui muito conhecimento técnico do que o Git. O Github não é uma plataforma unicamente para desenvolvedores ou engenheiros do software, por ter um sistema de armazenamento e flexibilidade é usado por diferentes indústrias para facilitar na criação de projetos que requerem um desenvolvimento em formato de arquivos.

4. Quantos usuários e empresas utilizam no mundo?

A plataforma Github conta com mais de 83 milhões de desenvolvedores e 4 milhões de empresas.

4.1 Quais países que mais utilizam no mundo? (classificação, números, gráficos)



- Alemanha: 624,543
- França: 575,165
- Japão: 552,879
- Reino Unido: 441,055
- EUA: 421,453
- Canadá: 418,958

Explicar para que servem os seguintes comandos:

- O **Git init** cria um repositório vazio.

- O **Git clone** duplica o repositório.
- O **Git add** adiciona novos arquivos ou modifica existentes no index, antes de fazer o commit.
- O **Git status** checa se os arquivos estão no repositório.
- O **Git diff** mostra diferenças entre uma árvore e o index, entre árvores, entre blobs ou entre arquivos
- O **Git branch** é uma ramificação/bifurcação realizada no projeto principal, ou seja, é a criação de um novo caminho/branch para o código visando não alterar o caminho/branch principal.
- O **Git merge** serve para unificar ramos de códigos de desenvolvimentos distintos criados pelo Git branch.
- O **Git push** é usado para enviar o conteúdo de um repositório local para um repositório remoto.
- O **Git pull** é utilizado para buscar e baixar conteúdo de repositórios remotos e fazer a atualização imediata no repositório local para que os conteúdos sejam iguais e atualizados.
- O **Git fetch** é o oposto do git push, visto que esse comando é responsável por realizar o download dos commits, arquivos e referências de um repositório remoto para um repositório local.
- O **Git remote** possibilita criar, visualizar e deletar conexões com outros repositórios, listando os nomes de cada repositório remoto manipulado especificado pelo usuário.
- O **Git commit** é o comando utilizado para registrar as alterações realizadas nos arquivos de um determinado repositório.
- O **Git show** é utilizado para exibir informações acerca de objetos do git, como blobs, trees, marcações e commits.
- O **Git log** permite visualizar as alterações realizadas em um determinado repositório.

- O **Git config** é a função que possibilita definir valores de configuração para o git em nível global ou local e, ao executá-lo, o arquivo de texto de configurações localizado no computador local é alterado.

5. Quais os principais comandos utilizados e quais os termos utilizados no github?

O git é uma parte importante da programação no dia a dia dos que trabalham em equipe principalmente em setores de software.

Há muitos comandos no github por isso que não é tão fácil dominá-los com rapidez. Demanda tempo. Há, no entanto, alguns que são utilizados com frequência no dia a dia.

Principais comandos do GitHub e mais utilizados:

Git clone: por exemplo, é um dos principais comandos e o seu objetivo é baixar o código fonte de um repositório remoto, como por exemplo, o github.

Git branch: é o comando para se trabalhar paralelamente e simultaneamente. Este comando cria, lista e exclui as branches.

Outros principais comandos:

Git Checkout;

Git status;

Git add;

Git commit;

Git push;

Git pull.