

**Saulo Maciel Rocha**

## **Relatório Científico: Funcionalidades do GitHub**

Lançado em 2008 e é usado desde então para que desenvolvedores possam hospedar seus projetos. Para controle de versão é usado o Git. O GitHub costuma ser o preferido entre os seus utilizadores por oferecer também alguns recursos de redes sociais, já que é possível seguir projetos de outros desenvolvedores e ainda comentar sobre todos eles. O GitHub, além de tudo, possui um recurso bastante interessante, o Git, sendo possível compartilhar um bloco de código. Além disso, também é possível trocar ideias, comentar os demais projetos e ainda pegar o código de alguém para modificar. Ele está disponível gratuitamente, com limite de armazenamento de 300MB.

Para quem busca mais privacidade, o serviço oferece ainda planos pagos, com isso, os desenvolvedores podem ter um maior controle sobre o código fonte, bem como adicionar desenvolvedores fixos e esconder os códigos dos demais membros. O GitHub funciona basicamente na nuvem, por isso é possível armazenar todo o material. Assim sendo, o projeto pode ser acessado de qualquer local.

## **Vantagens de usar o GitHub**

**Compartilhamento de projetos:** Um dos principais motivos pelo grande sucesso do GitHub é a possibilidade de compartilhar projetos. Desde o início do chamado **open-source**, o **GitHub** ganhou bastante destaque e é a rede mais usada atualmente;

**Pode ser usado como portfólio:** Possuir uma conta e projetos no GitHub é um grande diferencial, já que ele está se tornando um grande apoio extracurricular. Com isso, clientes e também empresas poderão conhecer melhor o trabalho do usuário,

**Integração com Git:** O GitHub, como vimos acima, possui total integração com o Git. Assim, o Git permite publicar e ainda gerenciar de forma fácil todos os projetos. E o melhor, de forma rápida e segura;

**Funciona como rede social:** O GitHub, além de várias vantagens, como as mencionadas acima, ainda funciona como uma rede social, o que possibilita que outro desenvolvedor siga o perfil do usuário, receba notificações e ainda é possível enviar mensagens diretas para um determinado perfil, o que facilita a troca de experiências, além claro, de fazer novas amizades;

**Possibilidade de aprender ainda mais:** Através do GitHub os usuários ainda podem aprender sempre mais. Tudo isso através da observação dos demais projetos, bem como a contribuição dos demais participantes.

Por esses e outros motivos, o GitHub é a melhor ferramenta para os desenvolvedores na atualidade. Se você ainda não conhece, vale a pena conferir as funcionalidades

## Geral

As configurações do GIT são armazenadas no arquivo **.gitconfig** localizado dentro do diretório do usuário do Sistema Operacional (Ex.: Windows: C:\Users\Documents and Settings\Leonardo ou \*nix /home/leonardo).

As configurações realizadas através dos comandos abaixo serão incluídas no arquivo citado acima.

Setar usuário

```
git config --global user.name "Leonardo Comelli"
```

Setar email

```
git config --global user.email leonardo@software-ltda.com.br
```

Setar editor

```
git config --global core.editor vim
```

Setar ferramenta de merge

```
git config --global merge.tool vimdiff
```

Setar arquivos a serem ignorados

```
git config --global core.excludesfile ~/.gitignore
```

Listar configurações

```
git config --list
```

## Ignorar Arquivos

Os nomes de arquivos/diretórios ou extensões de arquivos listados no arquivo **.gitignore** não serão adicionados em um repositório. Existem dois arquivos .gitignore, são eles:

- Geral: Normalmente armazenado no diretório do usuário do Sistema Operacional. O arquivo que possui a lista dos arquivos/diretórios a serem ignorados por **todos os repositórios** deverá ser declarado conforme citado acima. O arquivo não precisa ter o nome de **.gitignore**.
- Por repositório: Deve ser armazenado no diretório do repositório e deve conter a lista dos arquivos/diretórios que devem ser ignorados apenas para o repositório específico.

## Repositório Local

---

### Criar novo repositório

```
git init
```

### Verificar estado dos arquivos/diretórios

```
git status
```

## **Adicionar arquivo/diretório (staged area)**

Adicionar um arquivo em específico

```
git add meu_arquivo.txt
```

Adicionar um diretório em específico

```
git add meu_diretorio
```

Adicionar todos os arquivos/diretórios

```
git add .
```

## **Comitar arquivo/diretório**

Comitar um arquivo

```
git commit meu_arquivo.txt
```

Comitar vários arquivos

```
git commit meu_arquivo.txt meu_outro_arquivo.txt
```

Comitar informando mensagem

```
git commit meuarquivo.txt -m "minha mensagem de commit"
```

## **Remover arquivo/diretório**

Remover arquivo

```
git rm meu_arquivo.txt
```

Remover diretório

```
git rm -r diretorio
```

## Visualizar histórico



## Visualizar histórico

Exibir histórico

```
git log
```

Exibir histórico com diff das duas últimas alterações

```
git log -p -2
```

Exibir resumo do histórico (hash completa, autor, data, comentário e qtde de alterações (+/-))

```
git log --stat
```

Exibir informações resumidas em uma linha (hash completa e comentário)

```
git log --pretty=oneline
```

## Referências

<https://www.oficinadanet.com.br/post/14791-o-que-github>

<https://gist.github.com/leocomelli/2545add34e4fec21ec16>