

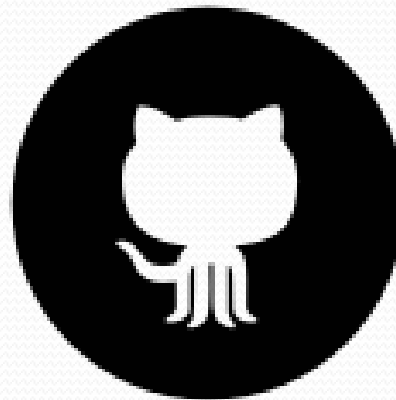
Faculdade Metropolitana da Grande Fortaleza – Fametro –



Desenvolvimento de Aplicações Corporativas

João Júnior

Unidade 4 – GitHub



GitHub

O que é o GitHub?

- É o maior host para repositórios Git e o ponto central de colaboração para milhões de desenvolvedores e projetos no mundo inteiro.
- Apesar de não fazer parte diretamente do projeto Git, caso deseje trabalhar com Git profissionalmente, você vai interagir em algum momento com o GitHub.

GitHub

Conta no GitHub

- Primeiro, precisamos criar uma conta no GitHub
- Para isso, basta acessar <http://github.com>
- Faça o cadastro seguindo as orientações do site



The screenshot shows the GitHub homepage with a focus on the sign-up process. The header includes navigation links: Personal, Open source, Business, Explore, Pricing, Blog, Support, a search bar, and Sign In/Sign up buttons. The main content area features the text "How people build software" and "Millions of developers use GitHub to build personal projects, support their businesses, and work together on open source technologies." To the right, there is a sign-up form with fields for "Pick a username", "Your email address", and "Create a password". Below the password field, a note states: "Use at least one letter, one number, and seven characters." A green "Sign up for GitHub" button is positioned below the form. At the bottom of the form, a small disclaimer reads: "By clicking 'Sign up for GitHub', you agree to our [terms of service](#) and [privacy policy](#). We'll occasionally send you account related emails."

Conectando ao GitHub

- Precisamos ter uma chave SSH para autenticação e, assim, realizarmos nossos commits.
- Vá em "Settings" -> "SSH and GPG keys".
- Clique em "New SSH key".
- No campo "Title", digite um nome para identificar o computador onde foi gerada a chave.
- Abra o arquivo **~/.ssh/id_rsa.pub** com o comando:

```
cd ~/.ssh  
notepad id_rsa.pub
```

- No campo Key, cole todo o conteúdo que está dentro do arquivo que acabamos de abrir.
- Clique em "Add SSH key"

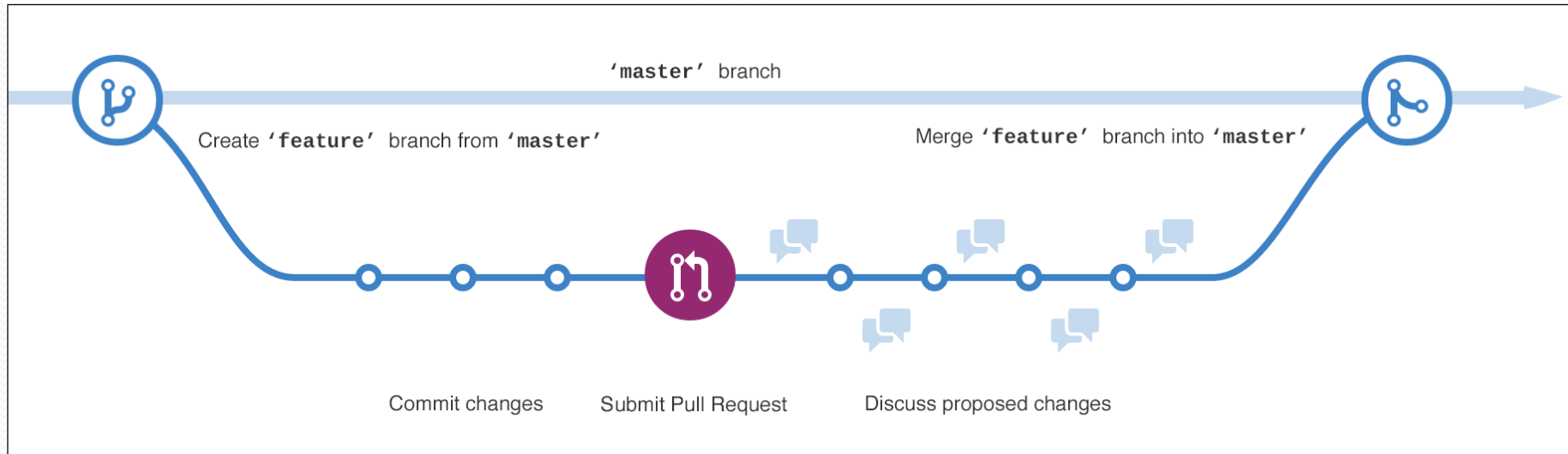
O fluxo do GitHub

- As pessoas podem fazer **fork**, dar **push** e, assim, contribuir com o repositório original.
- Isso chama-se **Pull Request**.
- O fluxo do GitHub basicamente é:
 - 1. Crie um **branch** do **master**;
 - 2. Faça alguns **commits** para melhorar o projeto;
 - 3. Dê o **push** dessa **branch** para o seu projeto no GitHub;
 - 4. Dê um **Pull Request** no GitHub;
 - 5. Discuta e continue dando **commit**, se desejar;
 - 6. O dono do projeto faz o **merge** ou fecha o **Pull Request**;

O que é uma branch?

- Criar **branch** é a forma pela qual trabalhamos em diferentes versões de um repositório ao mesmo tempo.
- Por padrão, todo repositório possui uma branch chamada **master** que é considerada a branch definitiva.
- Ao criar uma branch de master, você está basicamente fazendo uma cópia (ou um snapshot) de master como ele estava naquele momento.
- Se alguém fizer mudanças no master enquanto você trabalha na sua branch, você poderá dar um **pull** nessa atualizações.
- Veja o diagrama para entender um pouco melhor:

O que é uma branch?



- Aqui vemos:
 - O master (que é a branch padrão);
 - A branch criada chamada **"feature"**
 - O caminho que a branch **"feature"** faz até ser **mergiada** para **master**.

O que é *fork*?

- Caso não tenhamos acesso de **push**, podemos dar um **fork** em um projeto.
- Historicamente, o termo **fork** tem sido usado de forma negativa parecendo que alguém tomou um projeto open source em uma direção diferente e, algumas vezes, criar um projeto concorrente.
- No GitHub, um **fork** é simplesmente o mesmo projeto em seu próprio local de trabalho, o que lhe permite fazer alterações em um projeto público para contribuir de uma forma mais aberta.
- Dessa forma, os projetos não precisam se preocupar em ficar adicionando usuários como colaboradores para dar acesso de **push** a eles.
- Para isso, basta clicar em no botão “Fork” em cima do projeto.



Realizando *fork* nos projetos

- Clone o **fork** do projeto para o seu ambiente local:

```
git clone https://github.com/joaobrljunior/dac-2017.2
```

- Crie uma **branch** com um nome simples para lembrar:

```
cd dac-2017.2  
git checkout -b branch-fulano
```

- Abra o arquivo "alunos.txt" e coloque seu nome e sua matrícula:

```
notepad alunos.txt
```

- Cheque se as alterações estão corretas:

```
git diff --word-diff
```

- Faça o **commit** as alterações para a branch:

```
git commit -a -m 'Adicionei meu nome e matrícula.'
```

- Faça o **push** da **branch** para o **fork** do GitHub:

```
git push origin branch-fulano
```

Realizando *fork* nos projetos

- Volte para a página GitHub onde você fez o **fork**
- Perceba que o GitHub detectou que você deu um **push** na **branch** recém criada e exibe o botão verde "Compare & Pull Request".
- Como o nome sugere, ele serve para fazer o **checkout** e solicitar um **Pull Request** para o **master**.
- Clicando no botão, será exibida uma tela para você adicionar um título e uma descrição.
- Essas informações ajudarão o proprietário do **master** a entender a sua contribuição e, assim, realizar o **merge** do seu código com o código dele.
- Claro que ele pode interagir com você solicitando algumas alterações antes de fazer o **merge**.