

Gerenciamento de Projetos na Pesquisa Acadêmica

Introdução ao GitHub, Git, VSCode, Markdown e Python para o desenvolvimento e compartilhamento de códigos computacionais

- **Repositório remoto de projetos computacionais**
- **Plataforma para compartilhamento de códigos**
- **Plataforma para desenvolvimento colaborativo**

Você já tem uma conta no GitHub?
Acesse através do link <http://github.com/> e crie uma conta!

- **Repositórios públicos podem ser acessados e baixados**

Acesse este repositório <https://github.com/ossu/bioinformatics>
e torne-se um bioinformata!

- **Baixe um arquivo ou pasta usando o DownGit**

<https://downgit.github.io/>

- **Ferramenta para gerenciamento de versão de códigos**
- **Possui interface gráfica e linha de comando**

Verifique se vc já tem o Git instalado. Digite no terminal:
`git --version`

Leia o guia de instalação do Git para seu Sistema Operacional(OS):
<https://git-scm.com/book/pt-br/v2/Começando-Instalando-o-Git>

Especificamente para usuários de Windows, faça a instalação do Git SCM
<https://gitforwindows.org/>

Configurando o Git (através de linha de comando)

- Crie uma pasta vazia no seu computador, ela será seu *repositório local*
- Abra o git bash (Windows) ou um terminal (Linux, MacOS) nesta pasta
- Configure um usuário da seguinte forma:

```
git config --global user.name "seu nome"
```

```
git config --global user.email meu@email.com
```

- Configure o nome do ramo principal do git da seguinte forma:

```
git config --global init.defaultBranch main
```

- Confira suas configurações (e o local do arquivo .gitconfig) da seguinte forma:

```
git config --list --show-origin
```

VSCode

- **Ambiente de Desenvolvimento Integrado (IDE) *open source* generalista, modular e extensível**
- **Excelente para codificar em Python**
- **Integração com Git e GitHub através de interface simples**

Você já tem o VSCode instalado?

Instale o VSCode para o seu Sistema Operacional (OS) através do link

<https://code.visualstudio.com/>

VSCode

- **O VSCode permite instalar diversas extensões que permitem personalizar o IDE e integrá-lo com diversos outros apps**
- **As seguintes extensões serão usadas adiante na apresentação:**
 - ✓Python
 - ✓Jupyter
 - ✓GitLens
 - ✓Materials Icon Theme
 - ✓Markdown All in One
- **Para instalar extensões use o ícone *Extensions* da barra vertical na lateral esquerda, digite o nome da extensão e clique em *install***

Git + VSCode

- **A barra vertical lateral deve mostrar um ícone chamado *Source Control*, esse será o recurso para manusear as funcionalidades do Git através do VSCode**

- **Para criar um repositório local:**

- ✓ Abra uma pasta no VSCode: File > Open Folder
- ✓ Clique no ícone Source Control e escolha iniciar um repositório local nesta pasta

- **Escolha os três pontinhos do Source Control (no topo) e habilite as opções:**

- ✓ Source Control Repositories; Souce Control; Commits; Branches; Remotes

Git + VSCode

•As principais operações realizadas no Git através do Source Control no VSCode são:

- ✓Stage All changes (botão + na seção Changes)
- ✓Commit (botão com forma de "check" na seção Source Control)
- ✓Restore (Checkout) (ao clicar com o botão direito do mouse em um arquivo da seção Commits)
- ✓Publish to GitHub ou push (botão em formato de nuvem ou de setas com movimento circular na seção Source Control Repositories)

Git + VSCode

•O ciclo padrão de ações básicas do Git funciona da seguinte forma:

- ✓Quando algo é modificado (criado, deletado, alterado) na pasta do repositório local, então o Git indica uma mudança de status da pasta. Isso é visto no VSCode através da alteração de cores dos arquivos da pasta e da numeração que aparece no ícone Source Control na barra lateral esquerda
- ✓O usuário coloca as modificações em Stage (clcando em Stage All Chenges)
- ✓O usuário faz o Commit das modificações colocadas em Stage (digitando uma message que descreve o commit e clicando em Commit)
- ✓O usuário publica o commit no GitHub (clcando em Publish to GitHub ou em Push)

•Esse último estágio pode exigir login e senha (para ser digitada no campo que surge automaticamente centralizado no topo do VSCode)

Para mais informações, visite o link [Git + VSCode](#)

Git + GitHub

•Crie um novo repositório no GitHub da seguinte forma:

- ✓No canto superior direito da tela clique no botão + e em New repository
- ✓Escolha um nome para seu repositório (o nome do projeto)
- ✓Descreva brevemente o repositório (projeto)
- ✓Defina se o repositório será público ou privado
- ✓Clique no botão criar repositório
- ✓Observe (mas sem precisar salvar) o link HTTPS criado para o repositório

Git + GitHub

•Gere um token para sua conta GitHub da seguinte forma:

- ✓Abra o GitHub e entre na sua conta
- ✓No canto superior direito da tela, clique na sua foto de perfil e inicie o seguinte caminho: Settings > Developer Settings > Personal access tokens > Generate new token
- ✓Crie uma Nota para descrever seu token
- ✓Escolha apenas o seguinte escopo: repo
- ✓Gere um token, copie e cole em um arquivo de texto (se fechar sua janela do browser o token será perdido!)
- ✓Salve no seu computador o arquivo com o token

Git + GitHub

O token será sua senha para interagir com o GitHub através do Git. Após fazer o primeiro *commit* e o *push* para o repositório remoto, você poderá configurar o Git para salvar e resgatar o token automaticamente

•Adicione seu token às configurações do git da seguinte forma

✓Após fazer o *push* use os comandos:

```
git config --global credential.helper store
```

O token deverá ser salvo pelo git em um arquivo chamado .git-credentials no diretório de trabalho do usuário (veja o link [git credential helper](#) para detalhes do seu OS)

Windows > C:\Users\<username>\.git-credentials

Mac > /Users/<username>/.git-credentials

Linux > ~/.git-credentials

Markdown

- **Linguagem de marcação usada para formatar textos (plain text)**
- **Integrada e amplamente usada no GitHub**
- **Integrada e amplamente usada em Python (através do Jupyter Notebook)**
- **Para criar um arquivo Markdown, use a extensão .md**
- **Você pode criar arquivos .md (e qualquer outro) diretamente no VSCode**

Markdown

•O Markdown possui diversos elementos que podem ser usados para formatar textos:

- ✓O símbolo # é usado para criar títulos (assim como ## e ###, etc)
- ✓O símbolo * é usado para criar listas de itens não numerados (ex: * meu item)
- ✓Para listas numeradas se usa itens precedidos de 1. 2. 3., etc

Veja o link <https://www.markdownguide.org/cheat-sheet/> para uma lista com os principais recursos que podem ser usados com Markdown

Arquivos README.md são sempre usados no GitHub para informar aos leitores sobre especificidades do projeto

Python

- **Linguagem de programação generalista**
- **Curva de aprendizagem suave**
- **Amplamente usada para análise de dados**
- **Excelentes bibliotecas de funções para Machine Learning e IA**

Você já tem o Python 3 instalado? Digite no seu terminal:
`python3 --version`

Baixe a versão mais recente do Python3 através do link

<https://www.python.org/>

ATENÇÃO! No Windows, ao instalar o Python3 através do wizard de instalação, escolha a opção de incluir Python3 no *path*

Python + VSCode

- **Arquivos Python têm a terminação .py**
- **As principais extensões do VSCode para Python são:**
 - ✓ Python
 - ✓ Jupyter
- **Ao criar um arquivo .py, o VSCode pode sugerir instalar as extensões acima, caso você ainda não as tenha instalado**

Python + VSCode + Jupyter

- O Jupyter Notebook oferece uma forma dinâmica para o desenvolvimento em Python, sendo muito usado para análise de dados.
- Arquivos Jupyter Notebook para Python possuem terminação .ipynb
- Arquivos .ipynb são amplamente usados no GitHub
- A extensão Jupyter do VSCode habilita o desenvolvimento de arquivos .ipynb no próprio IDE
- Após instalar a extensão Jupyter, pressione Ctrl+Shift+P e pesquise no VSCode por *jupyter: create new blank notebook*. O VSCode pode pedir para selecionar o kernel para o Jupyter: escolha Qualquer versão do Python3 que esteja disponível.
- Saiba mais através do link <https://jupyter.org/>

Python + VSCode + Jupyter

- O Jupyter Notebook usa um esquema de células para desenvolvimento do código
- Células delimitam blocos de códigos interdependentes
- As células podem ser para código Python ou para Markdown, permitindo construir documentos que mesclam códigos com texto.
- Criar, deletar e mover células são ações muito intuitivas no Jupyter Notebook

Conclusão

- **Desenvolva seu código de forma completa, construindo arquivos instrutivos que mesclam Python e Markdown, através de documentos Jupyter Notebook**
- **Gerencie seus arquivos através das ferramentas de versionamento do Git integrado ao VSCode por meio da extensão GitLens**
- **Publique seus resultados do VSCode diretamente para o GitHub**
- **Envie o link do repositório e código SHA do commit mais recente para a revista científica onde você pretende publicar seus resultados**
- **Torne seus códigos e dados públicos e ajude a democratizar a Ciência!**