



GIT e GITHUB:

Desenvolvendo Habilidades Essenciais para
Colaboração e Controle de Versões



CONTEÚDO



01

O que é GIT e
GITHUB?

02

A importância do
controle de versão e
da colaboração

03

Entendendo o
GIT

04

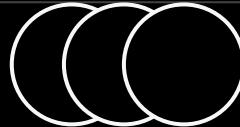
GitHub e
Colaboração em
Equipe





01

O que é GIT e
GITHUB?





GIT

!=



GITHUB



Software de
Controle de
Versão

“Facebook” para
desenvolvedores



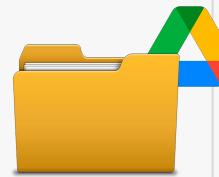
O QUE É GIT?



Projeto HTML



Projeto HTML
V2.0

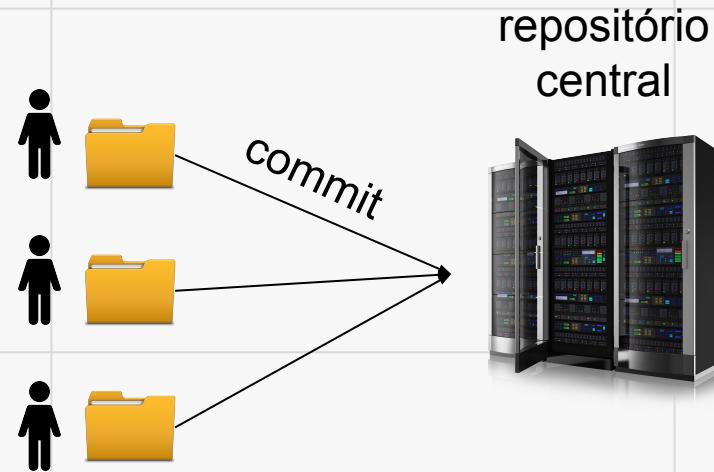


Projeto HTML
V2.0
(MODIFICADO)



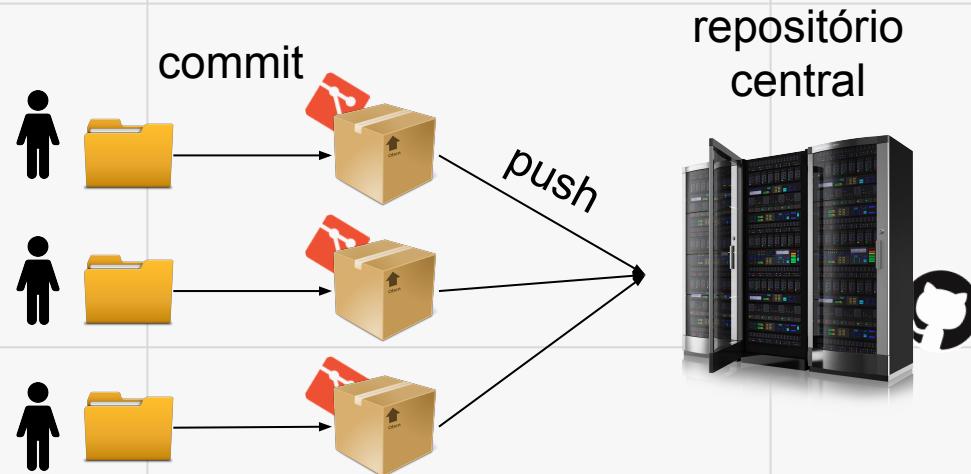
Projeto HTML
V2.0
(FINAL)

VCS CENTRALIZADO



Exemplo modelo centralizado / linear

VCS DISTRIBUIDO

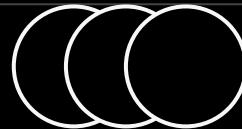


Exemplo modelo distribuído



02

A importância do controle de versões e da colaboração

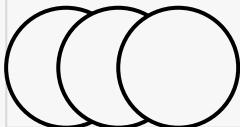


A importância de um controle de versão

Salvar versões de um software permite que os desenvolvedores acompanhem todas as mudanças feitas no código...

A importância de um controle de versão

É como possuir uma “máquina do tempo” e poder viajar para qualquer momento / versão;



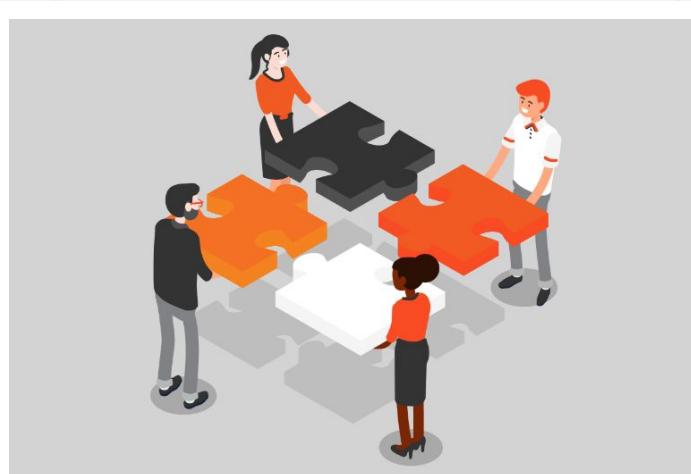
A importância de um controle de versão

- Fundamental para Trabalho em Equipe;
- Facilita Testes Fora do Ambiente de Produção;
- Rastreabilidade e Histórico;
- Aprimora a Comunicação;



A importância da colaboração (Github)

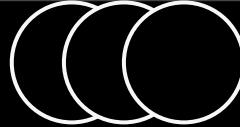
- Gerenciamento de Projetos(Github Insights);
- Visibilidade e Portfólio;
- Colaboração;
- Comunidade e Networking;





03

Entendendo o GIT

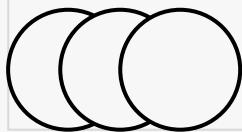


EVOLUÇÃO DO GIT

1985

CVS

- Centralizado;
- Open Source;
- Mais popular;
- Alguns
problemas;



EVOLUÇÃO DO GIT

1985

CVS

2000

SVN

- Centralizado;
- Open Source;
- **Mais popular;**
- Alguns
problemas;

- Centralizado;
- Open Source;
- Ativo até hoje;
- Parecido com o CVS;
- Corrigiu problemas do CVS



EVOLUÇÃO DO GIT

1985

CVS

2000

SVN

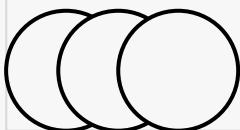
2000

BitKeeper

- Centralizado;
- Open Source;
- **Mais popular;**
- Alguns **problemas;**

- Centralizado;
- Open Source;
- Ativo até hoje;
- Parecido com o CVS;
- Corrigiu problemas do CVS

- Distribuído;
- **Proprietário;**
- Versão comunidade;
- Totalmente **diferente** do CVS;



EVOLUÇÃO DO GIT

Começo da “Treta”

- Questão filosófica

Richard Stallman



Linus Torvalds



EVOLUÇÃO DO GIT

“Treta” Andrew Tridgell X BitKeeper



EVOLUÇÃO DO GIT

“Treta” Linus Torvalds X BitKeeper



EVOLUÇÃO DO GIT

“Treta” Linus Torvalds X BitKeeper



2005

GIT

- Distribuído;
- Open Source;
- Focado em **performance**;
- Feito em 10 Dias;

INICIANDO COM O GIT

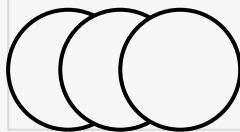
Por meio do site oficial o Git pode ser instalado em ambientes Windows, Linux e MacOS, seguindo as instruções fornecidas pelo assistente de instalação.

<https://git-scm.com/downloads>

INICIANDO COM O GIT

Ambiente Windows:

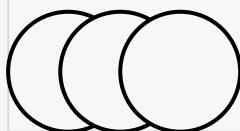
Após baixar e executar o **arquivo .exe** no site oficial, o usuário pode optar por manter as configurações padrão e concluir a instalação.



INICIANDO COM O GIT

Ambiente Linux:

- Abrir o Terminal (**CTRL ALT T**);
- Atualizar o Sistema (**sudo apt update**);
- Instalar o Git (**sudo apt install git**);
- Verificar a instalação (**git –version**);



CRIANDO UM REPOSITÓRIO

- O que é um repositório?
 - Local para armazenar e gerenciar um projeto;
 - Públicos;
 - Privados;

CRIANDO UM REPOSITÓRIO

```
● ● ●  
// Criando um repositório local com o Git  
  
// Após iniciar o terminal e navegar até o diretório desejado, execute:  
  
git init
```

STAGING AREA



VERIFICANDO STATUS



```
// Utilize o comando abaixo para  
// verificar o status atual de seu repositório  
  
git status
```

ADICIONANDO ALTERAÇÕES

```
// Adicionar todos os arquivos (Área de preparação)  
git add .  
  
// Adicionar um arquivo específico  
git add <nomeDoArquivo>
```

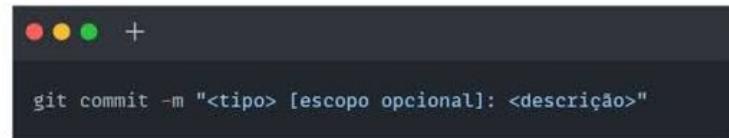
COMITANDO ALTERAÇÕES

```
// Enviar novas alterações ao repositório local  
// Recomendado para cada nova mudança  
git commit -m "Mensagem do commit"  
// Se somente um arquivo específico:  
git commit <caminhoDoArquivo> -m "Mensagem do commit"
```

COMMITS SEMÂNTICOS

Publicação de João Antônio Coelho, no Linkedin.

Commits Semânticos



Feat: Implementação de nova funcionalidade

Fix: Resolução de bugs ou problemas no código

Refactor: Refatoração de código

Style: Ajustes na formatação do código

Chore: Mudanças em ferramentas, configurações e bibliotecas,

Test: Mudança em testes da aplicação

Doc: Alterações na documentação do projeto

</>

VOLTANDO NO “TEMPO”

```
● ● ●  
// Verificar o histórico de commits  
  
git log  
  
// Copie o hash do commit desejado, e em seguida:  
// Voltar para a versão anterior  
  
git checkout <hashDoCommit>
```

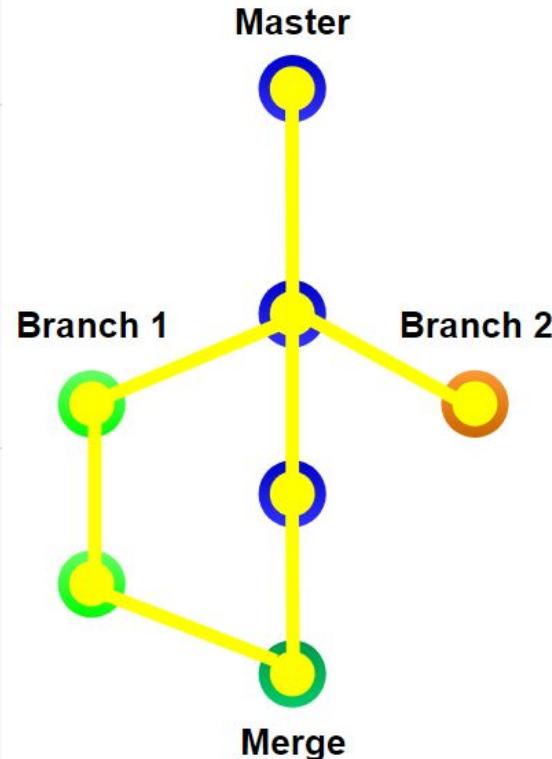
O QUE SÃO AS BRANCHES?

- Ramificações / “galhos” do código principal;



O QUE SÃO AS BRANCHES?

- **Master**: Linha principal de desenvolvimento;
- **Branch 1, 2**: Variações da linha principal;
- **Merge**: Momento onde há a junção de modificações;



COMANDOS BÁSICOS (BRANCHES)

```
// Exibe as branches:  
git branch  
  
// Cria uma nova branch:  
git checkout -b nomeDaBranch  
  
// Muda para uma outra branch:  
git checkout nomeDaBranch  
  
// EXCLUI uma branch existente:  
git branch -D nomeDaBranch
```

UNINDO MODIFICAÇÕES



// Realiza a união de uma branch

```
git merge nomeDaBranch
```

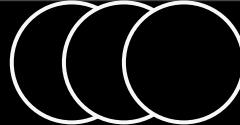
RESOLUÇÃO DE CONFLITOS

- Inconsistências entre diferentes versões do código;
- Ocorrem quando há alterações em partes do código modificadas por outros colaboradores ou em branches distintas;



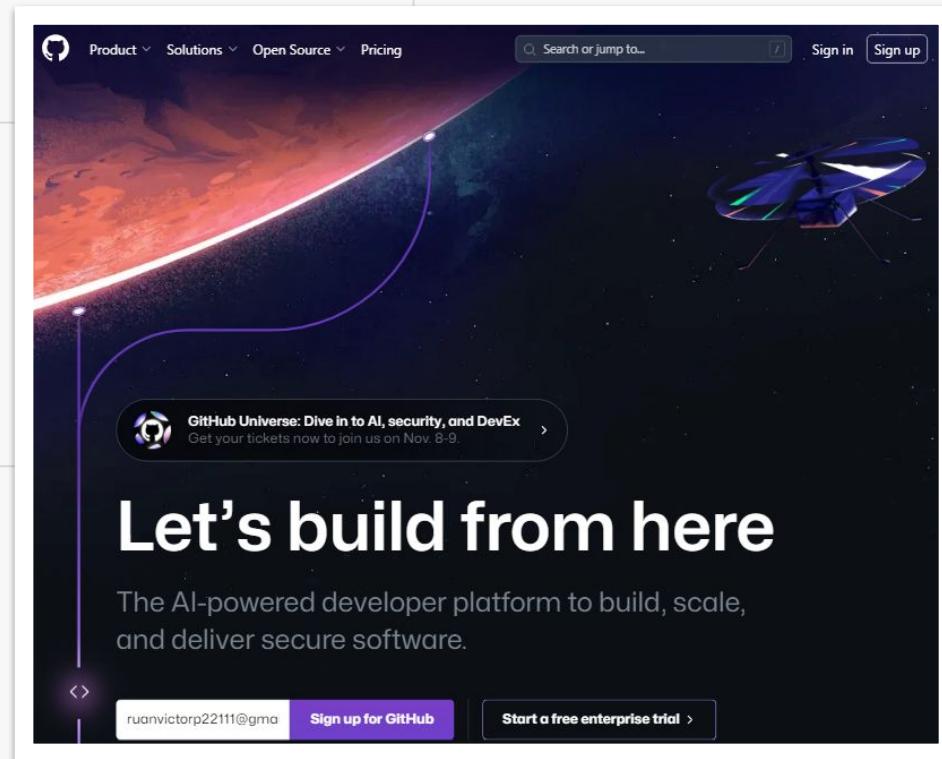
04

GitHub e Colaboração em Equipe

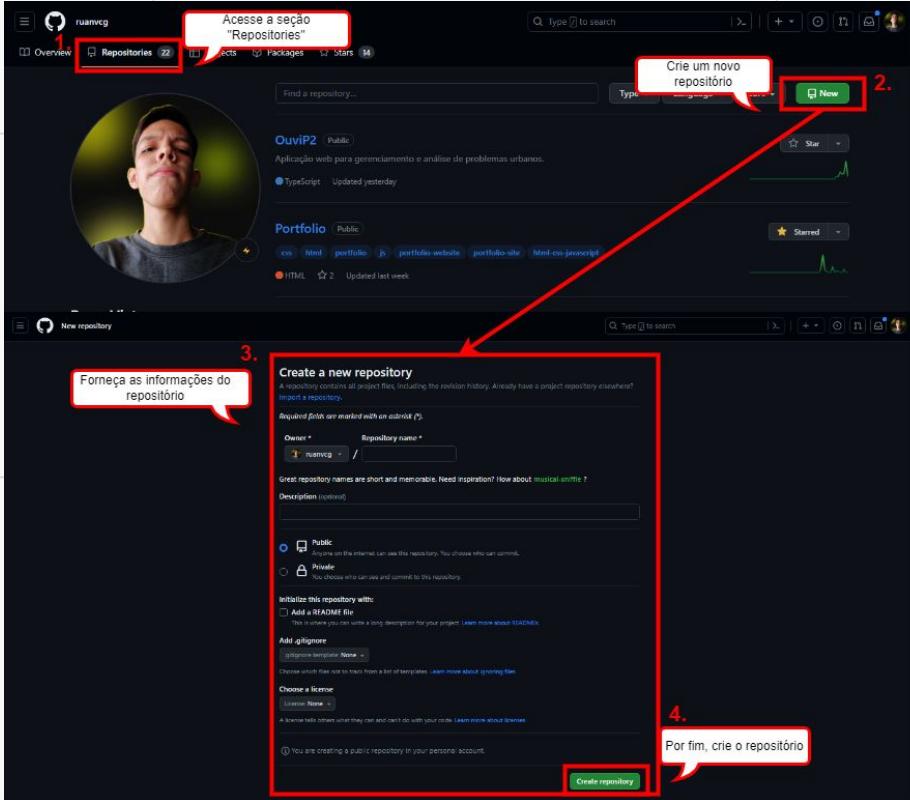


ENTRANDO NO GITHUB

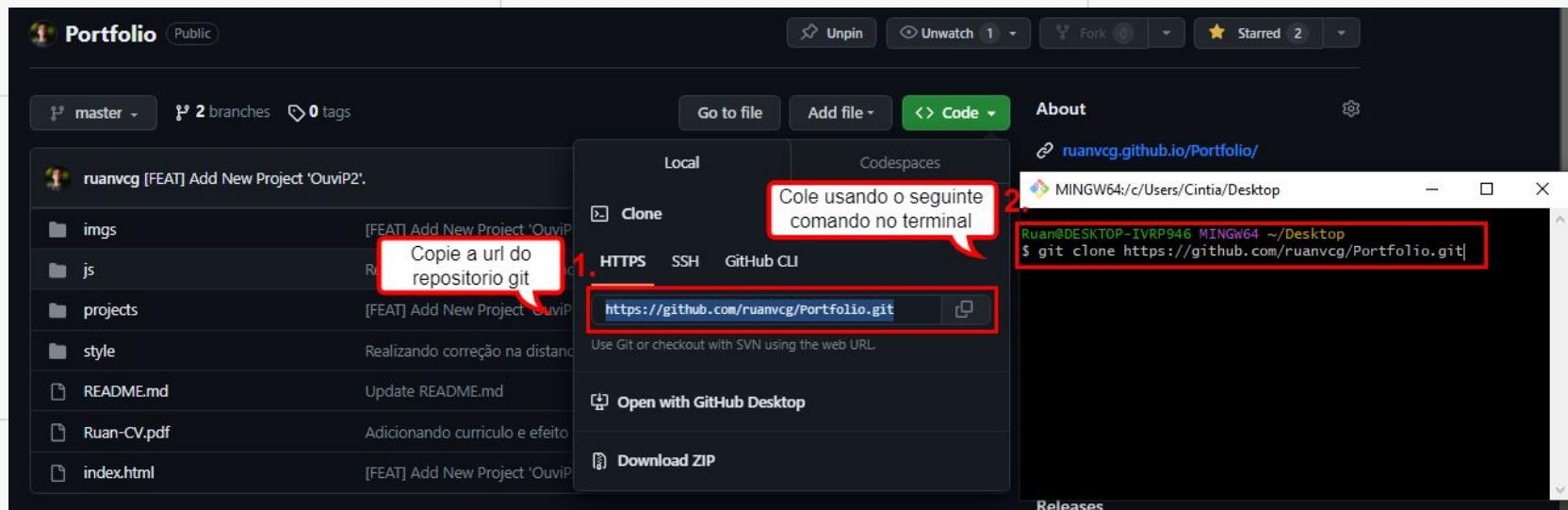
- Acesse:
<https://github.com/> ;
- Crie uma conta gratuitamente;
- Realize seu Login;



CRIANDO REPOSITÓRIOS



CLONANDO REPOSITÓRIOS



COMANDOS BÁSICOS

- Conectar “repo Local” com o “repo Remoto”:

git remote add origin <url-repo>

- Enviar modificações para o “repo Remoto”:

git push -u origin main

COMANDOS BÁSICOS

- Atualizar “repo Local”:

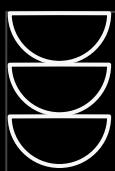
git pull

PULL REQUEST

- Solicitação;
- Revisão de Código;
- Verificar o funcionamento da implementação;

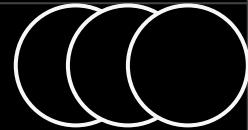
CONTRIBUINDO COM REPOSITÓRIOS

The screenshot shows a GitHub repository page for the user 'ruanvcg' named 'game'. The page includes a navigation bar with 'Code' and 'Pull requests' tabs, and sections for 'Security', 'Insights', and 'Settings'. A search bar at the top right has the placeholder 'Type ⌘ to search'. Below the search bar are buttons for 'Pin', 'Watch 0', and 'Fork'. The 'Fork' button is highlighted with a red box and a callout bubble pointing to it, labeled 'Botão para "copiar" o repositório para seu perfil'. Another callout bubble points to the 'Contribute' and 'Sync fork' buttons at the bottom right, labeled 'Botões para enviar contribuições ou atualizar o repositório para a versão mais recente'. The repository details show it's public, forked from 'barrosar2022/games', and has 1 branch and 0 tags. It also indicates 'This branch is 1 commit ahead of barrosar2022:master'. On the right side, there's an 'About' section for 'Jogos PyGame' with links to 'Readme', 'Activity', '0 stars', '0 watching', and '5 forks'. The footer of the slide features three overlapping circles.



MATERIAIS EXTRAS:

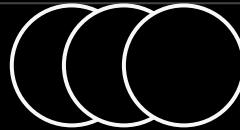
<https://github.com/ruanvcg/Enucompi-Minicurso-Git-Github>



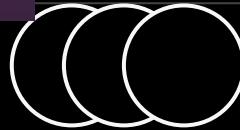


MATERIAIS EXTRAS:

<https://git-scm.com/doc>



MATERIAIS EXTRAS:



REFERÊNCIAS

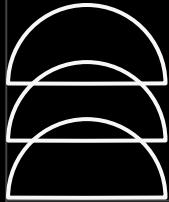
- [Blauw 2018] Blauw, F. F. (2018). The use of git as version control in the south african software engineering classroom. In 2018 IST-Africa Week Conference (IST-Africa), pages Page 1 of 8–Page 8 of 8.
- [de Alwis and Sillito 2009] de Alwis, B. and Sillito, J. (2009). Why are software projects moving from centralized to decentralized version control systems? In 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering, pages 36–39.
- [Decan et al. 2022] Decan, A., Mens, T., Mazrae, P. R., and Golzadeh, M. (2022). On the use of github actions in software development repositories. In 2022 IEEE International Conference on Software Maintenance and Evolution (ICSME), pages 235–245.
- [Elsen 2013] Elsen, S. (2013). Visgi: Visualizing git branches. In 2013 First IEEE Working Conference on Software Visualization (VISSOFT), pages 1–4.
- [Zhao et al. 2018] Zhao, Y., Zhou, Y., Hu, C., Zhang, X., and Zhang, Z. (2018). Git-based version control for beamline control system at the shanghai synchrotron radiation facility. In 2018 5th International Conference on Systems and Informatics (ICSAI), pages 134–138.



REFERÊNCIAS

- [Liberty and Galloway 2021] Liberty, J. and Galloway, J. (2021).
- [Parizi et al. 2018] Parizi, R. M., Spoletini, P., and Singh, A. (2018). Measuring team members' contributions in software engineering projects using git-driven technology. In 2018 IEEE Frontiers in Education Conference (FIE), pages 1–5.
- [Singh et al. 2021] Singh, V., Singh, A., Aggarwal, A., and Aggarwal, S. (2021). Devops based migration aspects from legacy version control system to advanced distributed vcs for deploying micro-services. In 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), pages 1–5.
- [Spinellis 2012] Spinellis, D. (2012). Git. IEEE Software, 29(3):100–101.
- [Xu et al. 2019] Xu, X., Cai, Q., Lin, J., Pan, S., and Ren, L. (2019). Enforcing access control in distributed version control systems. In 2019 IEEE International Conference on Multimedia and Expo (ICME), pages 772–777.





OBRIGADO!

Contatos:

caped.20211p2ads0193@aluno.ifpi.edu.br (Ruan),
caped.20211p2ads0045@aluno.ifpi.edu.br (Bruno),
caped.20211p2ads0355@aluno.ifpi.edu.br (Jociel),
caped.20211p2ads0126@aluno.ifpi.edu.br (João),
anderson.barros@ifpi.edu.br

Credits: This presentation template was created by **Slidesgo**,
including icons by **Flaticon**, infographics & images by **Freepik**

Please keep this slide for attribution



