

Atividades do Módulo 4 - QA NA PRÁTICA



Esse [repositório](#) é dedicado às atividades realizadas durante o Módulo 4 - M4 - LÓGICA DE PROGRAMAÇÃO COM PYTHON do curso de Quality Assurance oferecido pelo [Instituto Jogajunto](#).

📝 Descrição da 4ª Atividade: 🌟

💡 EM SQUADS. Escrevam um PROMPT no chatGPT, em busca da definição dos três conceitos abaixo, relacionados com o uso do Git:

TRACKING - COMMIT - BRANCHES - MERGE

Em seguida, discutam sobre os temas e escolham um integrante do SQUAD que, caso sorteado, deverá explicar para a turma o conceito, dando um exemplo.

- Essa atividade foi feita durante a aula, abaixo segue o que foi realizado em grupo e discutido com toda a turma com algumas considerações minhas:

No Chat GPT inseri as seguintes informações:

Por favor, forneça definições claras e detalhadas dos seguintes conceitos relacionados ao uso do Git:

TRACKING - Descreva o que significa 'tracking' em relação ao Git e como ele é usado para monitorar alterações em repositórios.

COMMIT - Explique o conceito de 'commit' no Git, incluindo sua finalidade e como ele é utilizado para registrar alterações no histórico de um projeto.

BRANCHES - Defina o termo 'branches' no contexto do Git, destacando sua importância na organização do desenvolvimento de software.

MERGE - Descreva o que significa 'merge' em Git, como é realizado e em que situações é útil para combinar diferentes linhas de desenvolvimento.

Desde já agradeço pelas definições para melhor compreensão destes conceitos relacionados ao Git.

Pedi por favor e agradeci no final porque se um dia essa ferramenta virar algo parecido com a [SkyNet](#) vão perceber que eu era um ser humano legal e pode ser que minha vida seja pouparada. 😊

Esse foi o resultado apresentado:

Claro, aqui estão as definições detalhadas dos conceitos relacionados ao Git:

1. **TRACKING:**

No contexto do Git, "tracking" refere-se à capacidade de um repositório local de acompanhar as alterações feitas em um repositório remoto. Quando você clona um repositório remoto para o seu ambiente local, o Git estabelece uma relação de rastreamento entre o seu repositório local e o repositório remoto. Isso permite que você saiba se as alterações foram feitas no repositório remoto desde a última vez que você sincronizou.

O rastreamento é útil para manter seu repositório local atualizado e garantir que você esteja trabalhando com a versão mais recente do código. Para atualizar seu repositório local com as alterações do repositório remoto, você pode usar comandos como `git pull`.

2. **COMMIT:**

No Git, um "commit" é uma operação que registra uma alteração específica no histórico de um projeto. Cada commit é uma captura instantânea do estado do projeto em um determinado momento. Os commits são usados para rastrear o progresso do desenvolvimento, facilitar a colaboração e permitir a reversão a versões anteriores do código.

Cada commit inclui uma mensagem que descreve as alterações feitas. Essas mensagens são cruciais para entender o propósito de cada commit. Os commits também têm identificadores únicos chamados de "hashes" que os tornam identificáveis e recuperáveis.

3. **BRANCHES:**

Em Git, "branches" (ramos) são caminhos de desenvolvimento separados que permitem que você trabalhe em recursos ou correções de bugs independentemente do ramo principal do projeto (normalmente chamado de "branch master" ou "branch principal"). Cada branch representa uma linha de desenvolvimento isolada, onde você pode adicionar, modificar e excluir arquivos sem afetar outros ramos.

Branches são úteis para organizar o desenvolvimento, permitindo que várias equipes ou desenvolvedores individuais trabalhem em paralelo sem interferir no trabalho uns dos outros. Quando o desenvolvimento em um branch é concluído, você pode fundir (merge) as alterações de volta ao branch principal.

4. **MERGE:**

"Merge" em Git se refere ao processo de combinar as alterações de um branch em outro. Isso é comumente usado para incorporar as alterações feitas em um branch de desenvolvimento de recurso ou correção de bug de volta ao branch principal (como o master).

O merge é útil quando você deseja unir diferentes linhas de desenvolvimento e consolidar as alterações. O Git usa algoritmos para combinar automaticamente as alterações, mas podem ocorrer conflitos se duas alterações conflitantes forem feitas na mesma parte de um arquivo. Nesses casos, você precisa resolver os conflitos manualmente.

Integrantes da Squad:

| Beatriz Souza | Bruno Soares | Leanderson Lima | Rebeca Borges | Sara Cruz |

Atividades do Módulo 4 - QA NA PRÁTICA 

Esse [repositório](#) é dedicado às atividades realizadas durante o Módulo 4 - M4 - LÓGICA DE PROGRAMAÇÃO COM PYTHON do curso de Quality Assurance oferecido pelo [Instituto JogaJunto](#).

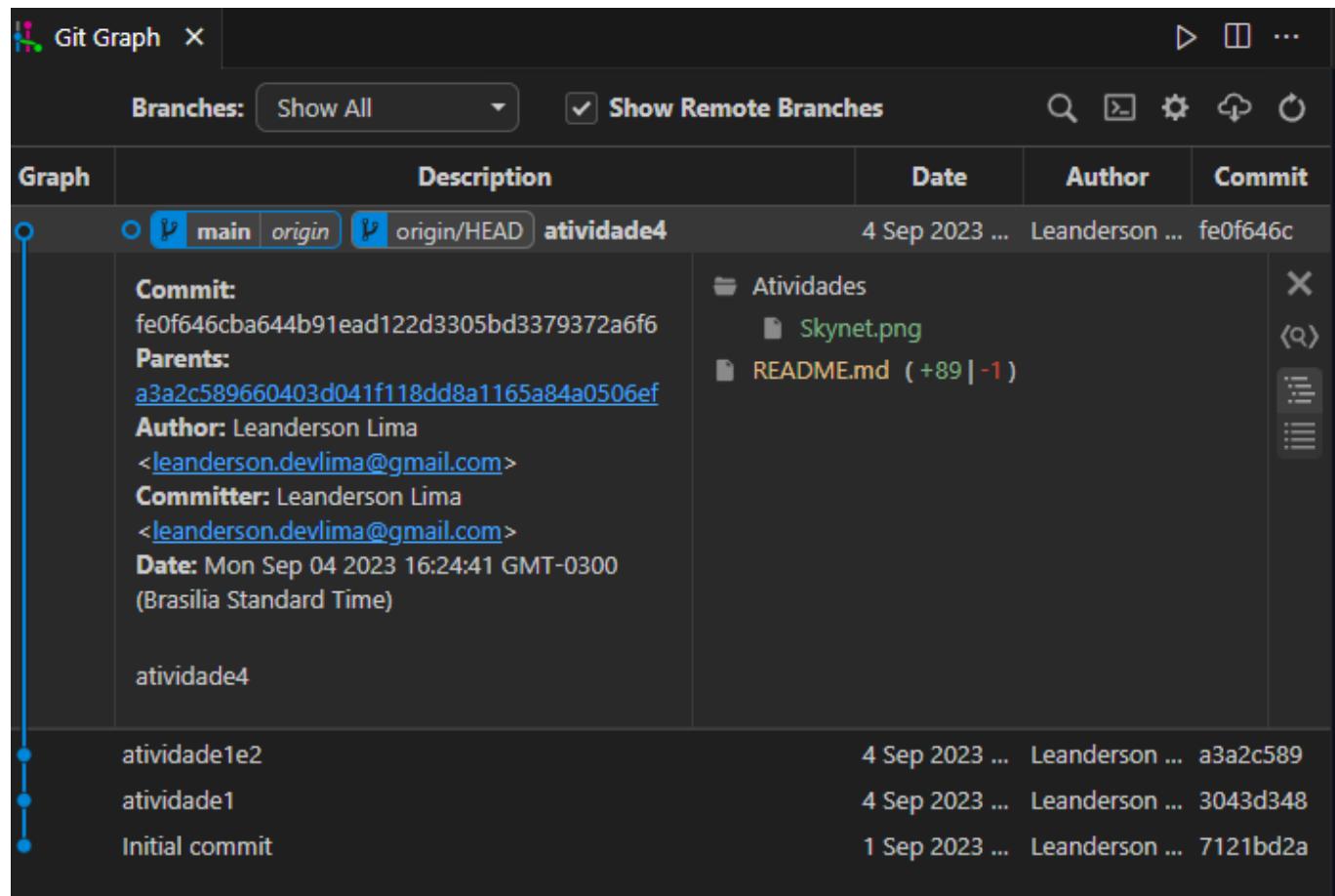
Descrição da 4^a Atividade:

🔍 EM SQUADS. Utilizando seus conhecimentos, crie seu repositório (local e na nuvem) e faça seus primeiros commit e push! Discussão após atividade.

Como foi a experiência? | Houve alguma dificuldade? | Como os integrantes do SQUAD se relacionaram? | Fariam algo diferente? O que? | Como se sentiram nesse processo?

- Para essa atividade vou considerar esse mesmo [repositório](#).

Fiz uso da extensão [Git Graph](#) para demonstrar a execução dessa atividade.



No que diz respeito à experiência resultante desta atividade, observamos que alguns membros da turma demonstraram um maior domínio do conceito de Git, enquanto outros estavam menos familiarizados. Trabalhamos em conjunto, auxiliando-nos mutuamente, para garantir que todos pudessem concluir com sucesso a atividade.

Integrantes da Squad:

| Beatriz Souza | Bruno Soares | Leanderson Lima | Rebeca Borges | Sara Cruz |