

Roteiro Aula Prática

PROJETO DE SOFTWARE

ROTEIRO DE AULA PRÁTICA

PROJETO DE SOFTWARE

Unidade: U4_Projeto_de_Software_avançado

Aula: A1_Gerenciamento_de_Configuração

OBJETIVOS

Definição dos objetivos da aula prática:

- Aprender a gestão de artefatos gerados pelo desenvolvimento de software através de ferramenta de gerenciamento de configuração.
- Compreender o funcionamento de um repositório de artefatos compartilhados com todos os colaboradores do time de desenvolvimento.
- Acessar o site GitHub, criar usuário do repositório, efetuar o download dos artefatos do projeto e atualizar artefato após modificação na instalação local.

SOLUÇÃO DIGITAL:

“GitHub e Git”

O GitHub é um conjunto de ferramentas para a gestão de configuração (ou Gerenciamento de Configuração de Software - SCM) tem como principal objetivo a Gerenciamento de Mudanças: controlar a alteração dos artefatos, para a criação de branches para isolar o desenvolvimento, a revisão de código (via Pull Requests) antes da integração, e a aprovação das mudanças. É muito útil para o Controle de Versão e Rastreabilidade.

LINK SOLUÇÃO DIGITAL ONLINE (EXCETO ALGETEC): <https://github.com/> e <https://git-scm.com/downloads>

PROCEDIMENTOS PRÁTICOS E APLICAÇÕES

Procedimento/Atividade nº 1

Inserir o nome do experimento: ***Uso de repositório de artefatos em Gestão de Configuração***

Atividade proposta: Neste exercício prático, você usará recursos da ferramenta para obter artefatos de um projeto de software armazenado em repositório, efetuar modificação em artefatos e atualizar o repositório com as modificações realizadas, deixando disponível para uso do time ou para integração do sistema.

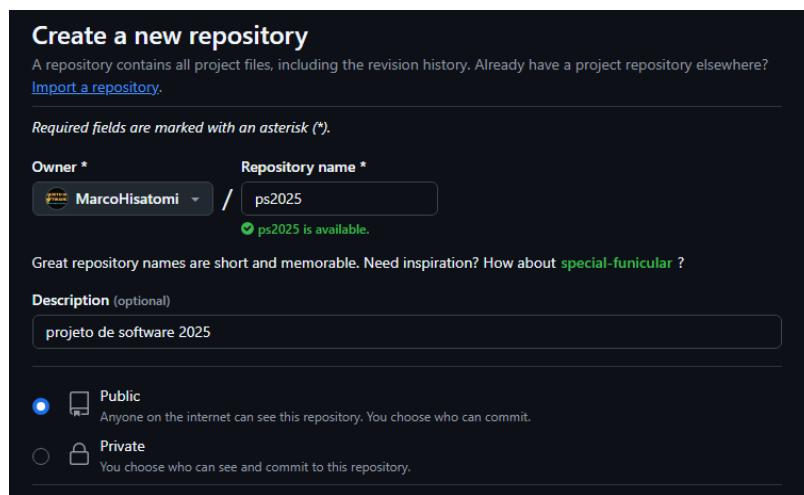
1. Preparar ambiente remoto para criação do repositório GitHub (Pré-requisito):
 - a) Conta GitHub: se ainda não tem, crie uma conta gratuita em github.com.
 - b) Git Instalado: baixe e instale o Git em seu computador: git-scm.com/downloads.

- c) Configuração do Git: no seu computador local, configure seu nome de usuário no Git (apenas uma vez):

```
git config --global user.name "marcohisatomi"
```

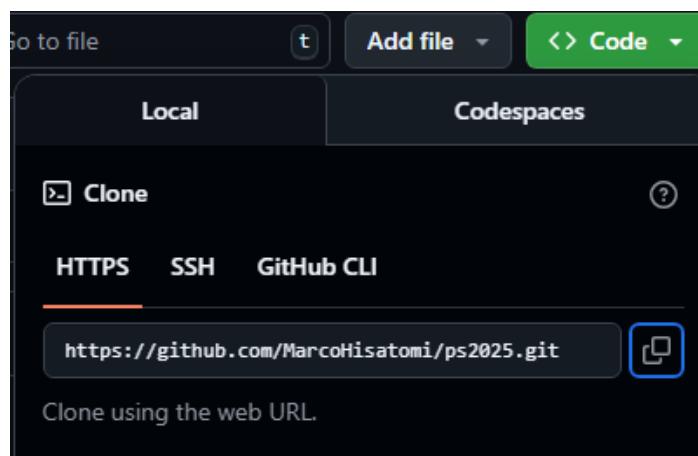
2. Criar um projeto (livre escolha)

Exemplo: “projeto2025_SCM”:



3. No GitHub, site do repositório, obter a URL HTTPS do projeto

Copiar a URL HTTPS do projeto, clicando no botão verde 'Code' (Código).



4. No Git Bash, obter os artefatos do projeto no GitHub, repositório remoto

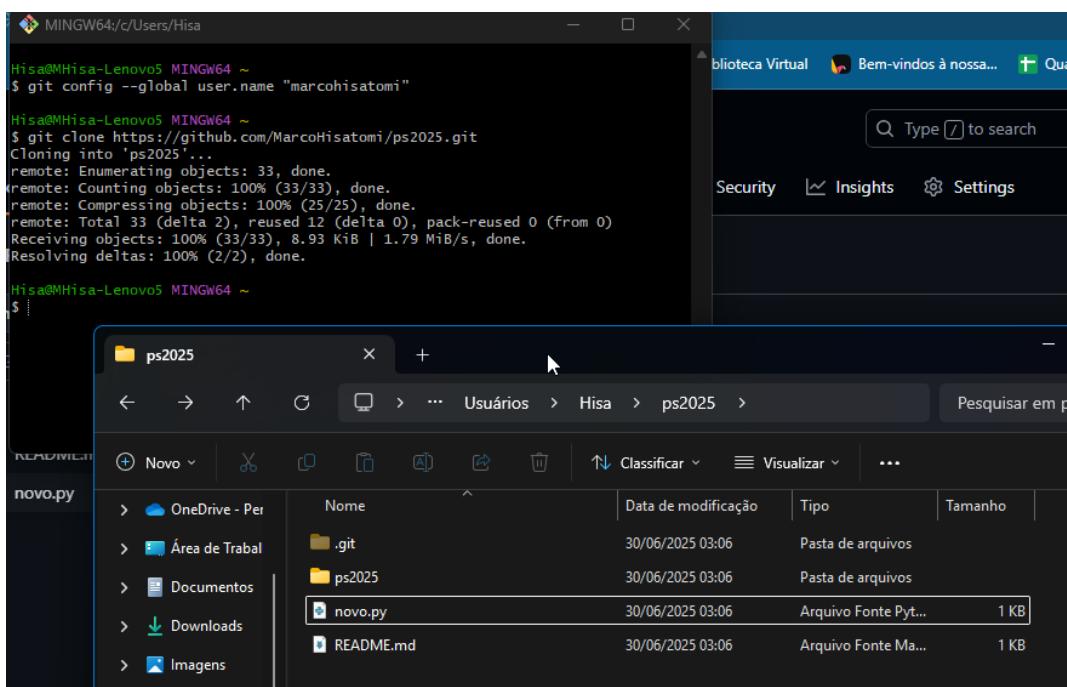
Acesse o Git Bash e cole (botão-direito) a URL copiada, assim os artefatos do projeto serão copiados do GitHub para o Git (no computador local).

```

MINGW64:/c/Users/Hisa
Hisa@MHisa-Lenovo5 MINGW64 ~
$ git clone https://github.com/MarcoHisatomi/ps2025.git
Cloning into 'ps2025'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

```

Perceberá que os artefatos serão baixados (download) numa pasta do seu computador local.



5. Modifique/crie artefatos em seu computador para implementar (as solicitações dos clientes)

Hora de trabalhar nos artefatos que estão no computador local. Após modificar ou criar itens de software, deve proceder a atualização no repositório do GitHub (no site)

6. Atualize suas modificações, efetuando a gravação (commit) dos artefatos

Faça uma sequência de passos que garantirá que o artefato criado/modificado no seu computador (Git) ficará atualizado no repositório remoto (GitHub)

- Verificar status dos itens, usando o comando “git status”, mostrará qual item-de-configuração está alterado ou é novo.
- Adicionar os arquivos, usando o comando “git add .”, modificados ou novos para o "staging area" (área de preparação)
- Execute um **commit**, usando o comando “git commit -m “mensagem de documentação” ”, para salvar as alterações/criações no repositório remoto.

- d) Por fim, publique o que salvou, usando o comando “git push”, persistindo os artefatos no repositório remoto GitHub (site)

```
Hisa@MHisa-Lenovo5 MINGW64 ~/ps2025 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    nova_funcionalidade.py

nothing added to commit but untracked files present (use "git add" to track)

Hisa@MHisa-Lenovo5 MINGW64 ~/ps2025 (main)
$ git add .

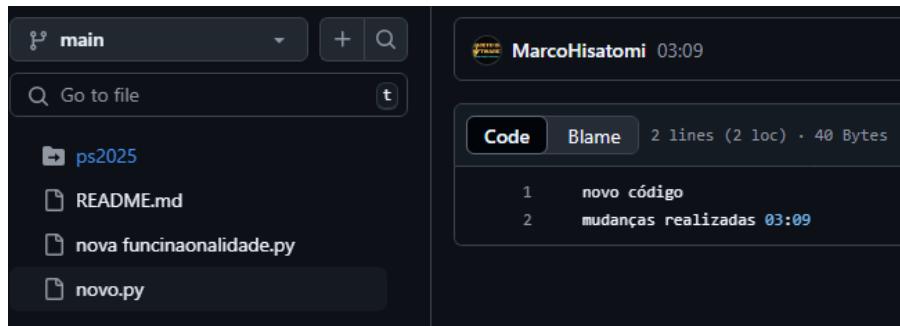
Hisa@MHisa-Lenovo5 MINGW64 ~/ps2025 (main)
$ git commit -m "nova funcionalidade"
[main 7553fd9] nova funcionalidade
 1 file changed, 1 insertion(+)
 create mode 100644 nova_funcionalidade.py

Hisa@MHisa-Lenovo5 MINGW64 ~/ps2025 (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 402 bytes | 402.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/MarcoHisatomi/ps2025.git
  fb333f0..7553fd9  main -> main

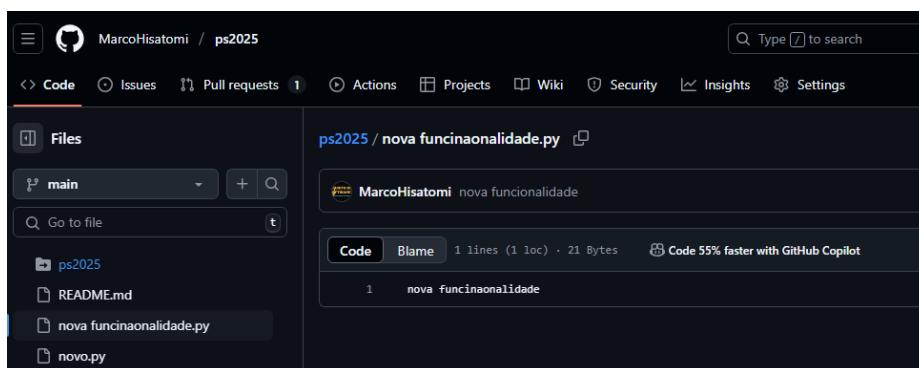
Hisa@MHisa-Lenovo5 MINGW64 ~/ps2025 (main)
$
```

7. Verifique suas modificações/criações, diretamente no repositório remoto GitHub

O artefato “novo.py” foi alterado com adição do conteúdo “mudanças realizadas 03:09”



O artefato “nova funcionalidade.py” foi criada e inserida no projeto remoto (GitHub)



Procedimentos para a realização da atividade: Siga o enunciado para desenvolver essa aula prática.

1. Enunciado para um projeto em desenvolvimento de software sob a perspectiva do estudante (ou criatividade num cenário hipotético)

Com muitas alterações solicitadas por vários stakeholders (do time de operações), praticamente ao mesmo tempo, exige um controle efetivo nos artefatos em criação/modificação.

Cada colaborador é responsável por parte dos artefatos que estão sendo desenvolvido, até mesmo itens de documentação ou de design, portanto, muitas pessoas dependem entre si dos artefatos que já foram implementados.

Portanto, um método eficiente é o uso de um repositório único e remoto em que todos podem acessar a última versão do artefato, efetuando atualizações quando efetuar o retorno do artefato modificado e, quando necessário, executar o merge das linhas de códigos correspondentes às modificações em andamento.

Crie um projeto novo, na ferramenta (GitHub) de gestão de configuração e controle de artefatos, com características colaborativa, que possibilite a modificação pelos membros de um time de desenvolvimento de software. Obtenha os artefatos do projeto em seu computador local a partir do repositório remoto único. Lembrando que esse repositório é o principal lugar que estará armazenado todos os artefatos atualizados em sua última versão ou em mudanças controladas. Efetue modificações em um dos artefatos e/ou crie um artefato, em seguida, atualize o repositório com a versão mais atual do seu computador, diretamente no repositório remoto.

Demonstre que consegue controlar a obtenção do artefato diretamente do repositório remoto e que, também, consegue efetuar o upload dos artefatos modificados/criados em seu computador.

2. Questão Reflexiva: ao final descreva o resultado que obteve com a prática desenvolvida

Descreva um resumo das habilidades que desenvolveu com o exercício do problema proposto, comente o que aprendeu e/ou aperfeiçoou através do conhecimento obtido com essa aula prática.

Avaliando os resultados:

Responda a seguinte questão de acordo com o cenário a seguir:

A atividade propõe a criação de um projeto em ferramenta de gestão de configuração que possibilita o gerenciamento dos artefatos de um projeto de software, bem como disponibilizar os artefatos modificados/criados por um membro do time, com as respectivas documentações.

1. Explique como as competências desenvolvidas contribuem para a confiabilidade na execução das tarefas profissionais de um time de desenvolvimento de software.
2. Por que é importante utilizar ferramenta de gestão de configuração em ambientes reais?

Checklist:

- Assegurar que o projeto foi criado no ambiente remoto com sucesso;
- Assegurar que os artefatos do repositório do projeto remoto foi atualizado no computador local;
- Assegurar que os artefatos modificados/criados foram inseridos/atualizados no repositório remoto;
- Revisar todas as solicitações deste Roteiro de Aula Prática.
- Submeter o resultado conforme as instruções recebidas.

RESULTADOS

Resultados do experimento:

O estudante deve entregar um arquivo em PDF contendo evidências dos passos solicitado para realizar o exercício.

O arquivo deverá conter:

- Capa;
- Folha de rosto com os dados da disciplina e do aluno;
- Evidência (prints) completa do exercício;
- Referências bibliográficas (quando houver).

Resultados de Aprendizagem:

Ao final da atividade, espera-se que o estudante:

- O estudante seja capaz de criar um projeto no GitHub.
- O estudante seja capaz de obter os artefatos do projeto remoto (do GitHub) para o computador local (no Git);
- O estudante possa perceber o potencial uso da ferramenta de gerenciamento de configuração em sua atividade profissional após a realização dessa aula prática.