



# **Relatório de criação de repositório local e remoto usando GIT**

Alunos: Diego Chruscinski de Souza e  
Gustavo Moretto Dalla Costa

Professora: Eliane Maria De Bortoli Fávero

Outubro, 2023

# Resumo:

Este relatório técnico fornece um guia passo a passo abrangente para utilizar a ferramenta de controle de versão GIT. Este guia permitirá acompanhar todo o ciclo de vida de um pequeno projeto em Java, desde o início até a conclusão. Além disso, o repositório estará acessível para visualização remota ao término deste documento.

# Descrição de Atividades:

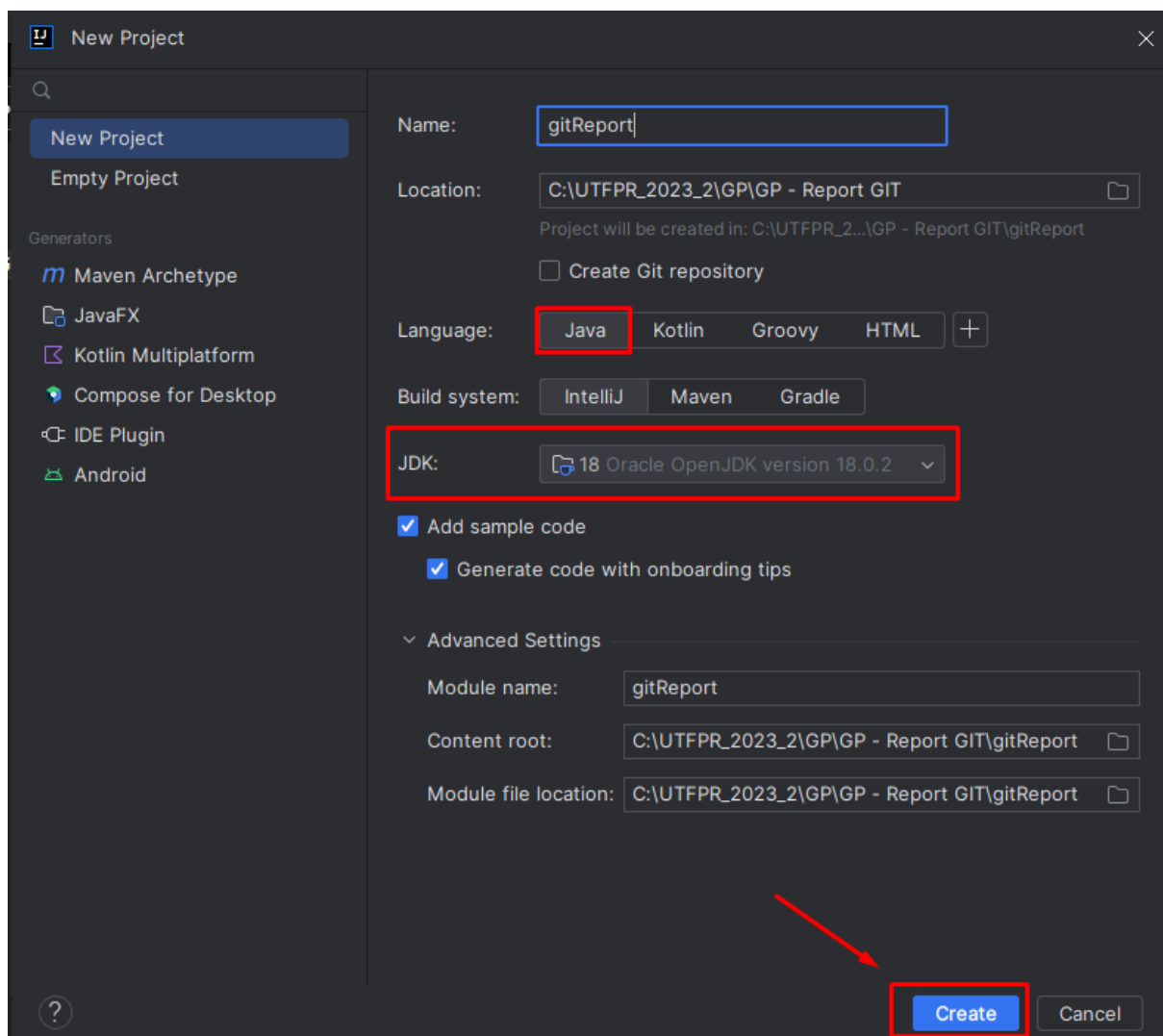
## 1.0 - Configuração do projeto:

### 1.1 - Inicialização da IDE IntelliJ:

Ambiente de desenvolvimento escolhido para criação deste projeto.

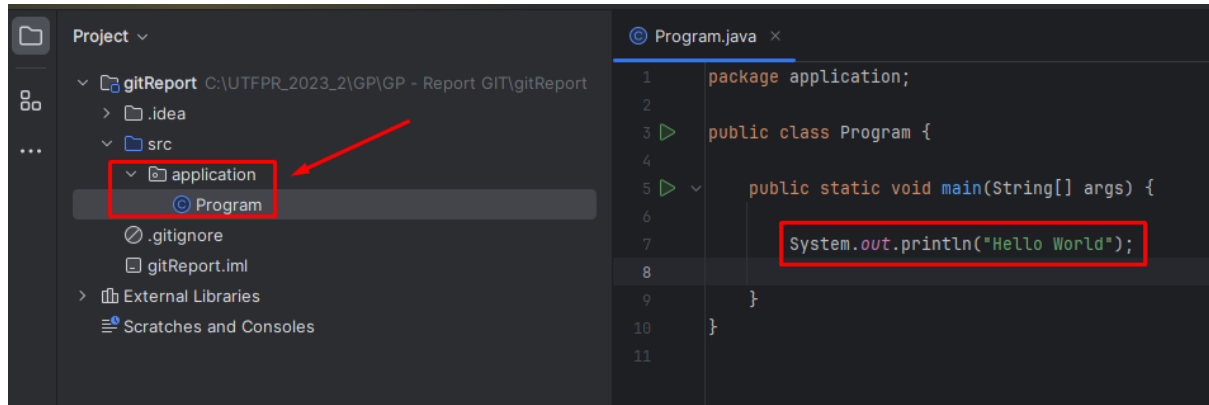
### 1.2 - Criação de novo projeto:

Criação do projeto de nome **gitReport** localizado na pasta **C:\UTFPR\_2023\_2\GP\GP - Report GIT**, qual utiliza a linguagem de programação **Java** no **JDK 18.0.2**.



### 1.3 - Alimentação dos arquivos:

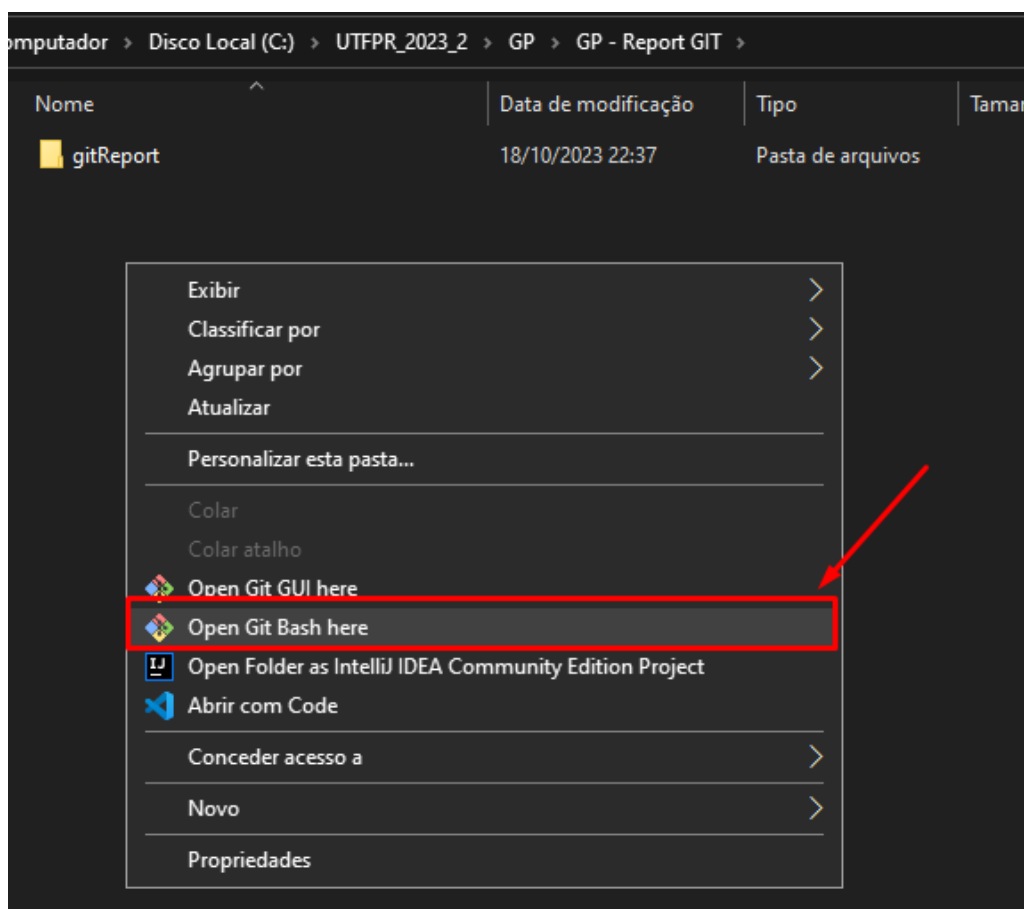
Criação do pacote **application** e da classe **Program**. Inserção do **print** "Hello World" na classe.



## 2.0 - Configuração inicial do GIT:

### 2.1 - GIT BASH (Console):

Abertura do **console GIT** na pasta do projeto.



## 2.2 - GIT INIT:

**Inicialização** do repositório **GIT** local no diretório do projeto.

```
$ git init
Initialized empty Git repository in C:/UTFPR_2023_2/GP/GP - Report GIT/.git/
```

## 2.3 - GIT ADD:

**Adição** de todos os arquivos no diretório ao índice.

```
$ git add .
```

## 2.4 - GIT COMMIT:

**Criação** do primeiro **commit** do projeto, nomeado como "First Commit".

```
$ git commit -m "First Commit"
[master (root-commit) e907a9b] First Commit
6 files changed, 67 insertions(+)
create mode 100644 gitReport/.gitignore
create mode 100644 gitReport/.idea/.gitignore
create mode 100644 gitReport/.idea/misc.xml
create mode 100644 gitReport/.idea/modules.xml
create mode 100644 gitReport/gitReport.iml
create mode 100644 gitReport/src/application/Program.java
```

## 3.0 - Configuração do repositório remoto no GITHUB:

### 3.1 - Criação do repositório remoto:

Link do Repositório:

<https://github.com/chruscinskiDiego/GP-Report-GIT>

**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \* chruscinskiDiego / Repository name \* GP-Report-GIT  
✓ GP-Report-GIT is available.

Great repository names are short and memorable. Need inspiration? How about [solid-guacamole](#) ?

Description (optional)  
Report to the university exemplifying the use of git

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

Initialize this repository with:

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

**Add .gitignore**  
.gitignore template: None  
Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

**Choose a license**  
License: None  
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

ⓘ You are creating a public repository in your personal account.

**Create repository**

### 3.2 - GIT REMOTE ADD:

Conexão do repositório local com o remoto.

```
$ git remote add origin https://github.com/chruscinskiDiego/GP-Report-GIT.git
```

### 3.3 - GIT PUSH:

Envio das alterações do branch principal (master) para o repositório remoto.

```
$ git push -u origin master  
Enumerating objects: 12, done.  
Counting objects: 100% (12/12), done.  
Delta compression using up to 4 threads  
Compressing objects: 100% (8/8), done.  
Writing objects: 100% (12/12), 1.57 KiB | 321.00 KiB/s, done.  
Total 12 (delta 0), reused 0 (delta 0), pack-reused 0  
To https://github.com/chruscinskiDiego/GP-Report-GIT.git  
* [new branch]      master -> master  
branch 'master' set up to track 'origin/master'.
```

## 4.0 - Desenvolvimento de Recursos/Alterações:

### 4.1 - GIT BRANCH DIEGO:

Criação da branch “**diego**” para alterações do usuário.


```
$ git branch diego
```

### 4.2 - GIT CHECKOUT DIEGO

Alteração da branch atual (master) para a “**diego**”.

```
$ git checkout diego  
Switched to branch 'diego'
```

### 4.3 - Alimentação da BRANCH no projeto:



```
© Program.java x  
1 package application;  
2  
3 public class Program {  
4  
5     public static void main(String[] args) {  
6  
7         //commit 1  
8         System.out.println("Hello World");  
9  
10        //commit 2  
11        System.out.println("Commit BRANCH Diego");  
12    }  
13 }  
14 }  
15
```

### 4.4 - GIT ADD:

Adição das novas alterações para a branch “**diego**”.

```
$ git add .
```

### 4.5 - GIT COMMIT:

Commit das alterações adicionadas à branch “**diego**”.

```
$ git commit -m "diego"
[diego d418ef3] diego
1 file changed, 5 insertions(+), 1 deletion(-)
```

#### 4.6 - GIT CHECKOUT MASTER:

Alternação para a branch principal.

```
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

#### 4.7 - GIT MERGE:

Mesclagem das alterações dos branches **diego** com o **master**.

```
$ git merge diego
Updating e907a9b..d418ef3
Fast-forward
 gitReport/src/application/Program.java | 6 +++++-
1 file changed, 5 insertions(+), 1 deletion(-)
```

#### 4.8 - Realização do processo anterior para a BRANCH “gustavo”:

(IDE)

```
//commit 3
System.out.println("Commit BRANCH Gustavo");
```

(BASH)

```
$ git merge gustavo
Updating d418ef3..8d35e14
Fast-forward
 gitReport/src/application/Program.java | 5 +++++
1 file changed, 5 insertions(+)
```

#### 5.0 - GIT PUSH:

Envio de todas as alterações para o repositório remoto

```
$ git push -u origin master
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (18/18), 1.33 KiB | 455.00 KiB/s, done.
Total 18 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 1 local object.
To https://github.com/chruscinskiDiego/GP-Report-GIT.git
 e907a9b..8d35e14 master -> master
branch 'master' set up to track 'origin/master'.
```



## 6.0 - Finalização do projeto:

### 6.1 - GIT TAG:

Criação da tag para a versão do projeto.

```
$ git tag 1.0
```

### 6.2 - GIT PUSH --TAGS:

Envio das tags para o repositório remoto.

```
$ git push --tags
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/chruscinskiDiego/GP-Report-GIT.git
* [new tag]          1.0 -> 1.0
```

### 6.3 - GIT PUSH:

Finalização com envio dos arquivos ao repositório remoto.

```
$ git push origin master
```

# CONCLUSÃO

Neste relatório, detalhamos o ciclo de vida do projeto, desde a sua criação até a conclusão, usando o Git como sistema de controle de versão. Os passos descritos acima representam uma estrutura geral que pode ser adaptada e estendida para atender às necessidades específicas de um projeto.

**\*\*Fim do Relatório\*\***