



Universidade Federal do Ceará  
Centro de Ciências/Departamento de Computação  
Código da Disciplina: CK0084  
Professor: Ismayle de Sousa Santos

**Aula  
02**

# **Sistemas de Informações e Banco de Dados**

**Ambiente de Desenvolvimento Integrado  
Introdução ao Git**

---



**rfbrkh3**



**ismaylesantos@great.ufc.br**



**@IsmayleSantos**

# Agenda

- **IDE**
    - O que é?
    - Vantagens
    - Exemplos
    - Como escolher?
  - **Versionamento de Código**
    - O que é?
    - Versionamento Semântico
    - Git
    - GitFlow
-

# Vamos começar falando sobre ...

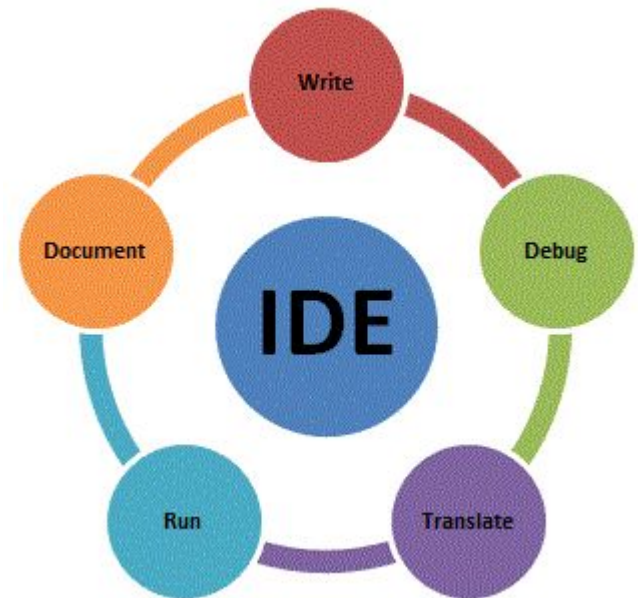
**Integrated  
Development  
Environment**

Alguém sabe o que é?



# IDE

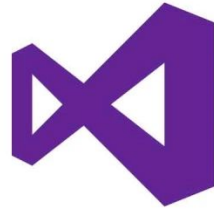
- Ambiente de desenvolvimento integrado
  - **I**ntegrated **D**evelopment **E**nvironment (IDE)
- Uma IDE contém
  - Editor de código fonte
  - Compilador/interpretador
  - Depurador
  - Outros
    - Sistema de Versionamento
    - Widgets



# Exemplos de IDE



eclipse



Visual  
Studio



IntelliJ IDEA



Apache

**NetBeans** IDE



bluej.org

# Benefícios de usar uma IDE

- Aumenta produtividade
- Facilidades
  - Compilação, Deploy, Depuração
- Feedback
  - Erros de codificação
  - Erros de compilação



# Principais IDEs - Top Index

Worldwide, Nov 2020 compared to a year ago:

Rank	Change	IDE	Share	Trend
1	↑	Visual Studio	25.31 %	+3.6 %
2	↑	Eclipse	16.31 %	-1.1 %
3	↓↓	Android Studio	11.39 %	-11.3 %
4	↑	Visual Studio Code	8.74 %	+3.4 %
5	↑	pyCharm	7.73 %	+2.5 %
6	↑	IntelliJ	6.13 %	+1.3 %
7	↓↓↓	NetBeans	5.3 %	-0.2 %

# Qual IDE escolher?

- Linguagem de Programação
- Integrações com outras ferramentas e plugins



**NetBeans**



**IntelliJIDEA**





# Eclipse IDE

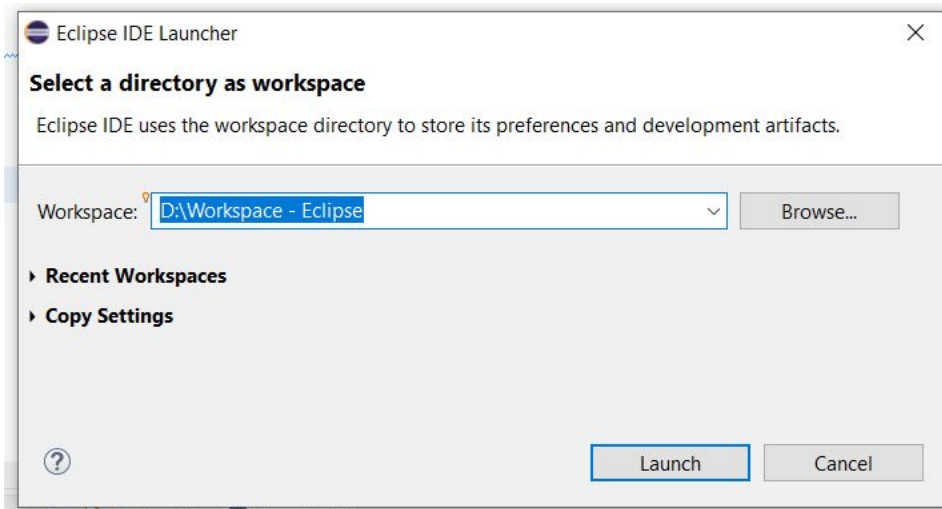
- Ambiente de desenvolvimento popular
- IDE de código aberto
- Multilinguagem
- Multiplataforma
- Forte orientação ao desenvolvimento baseado em **plug-ins**



<https://www.eclipse.org/downloads/>

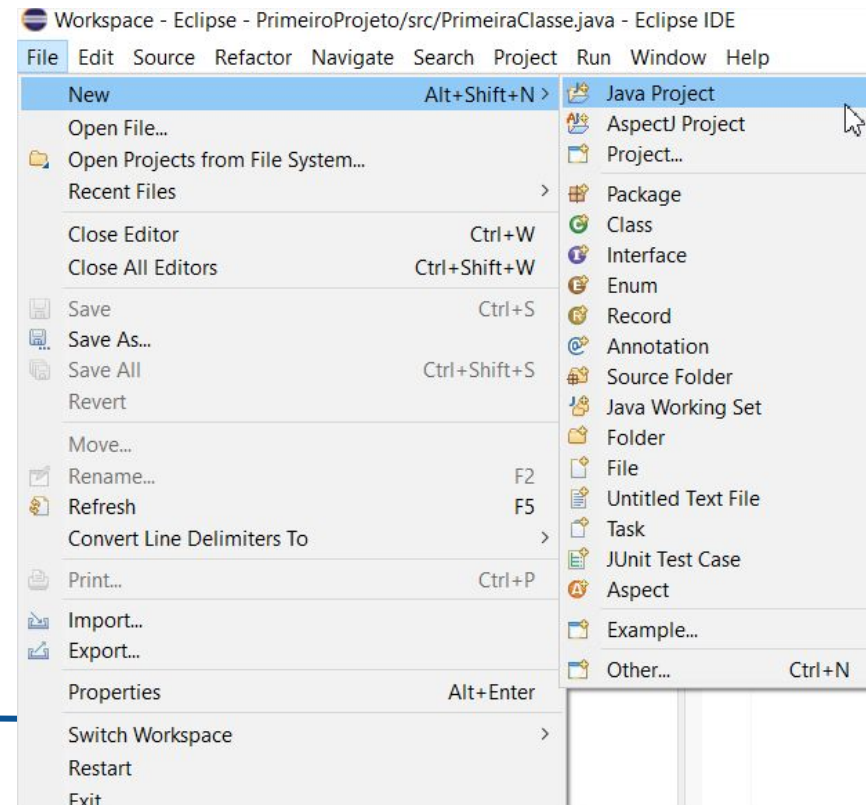
---

# Eclipse IDE - Conceitos Básicos



**Workspace** = espaço em disco onde vão ser salvos os projetos

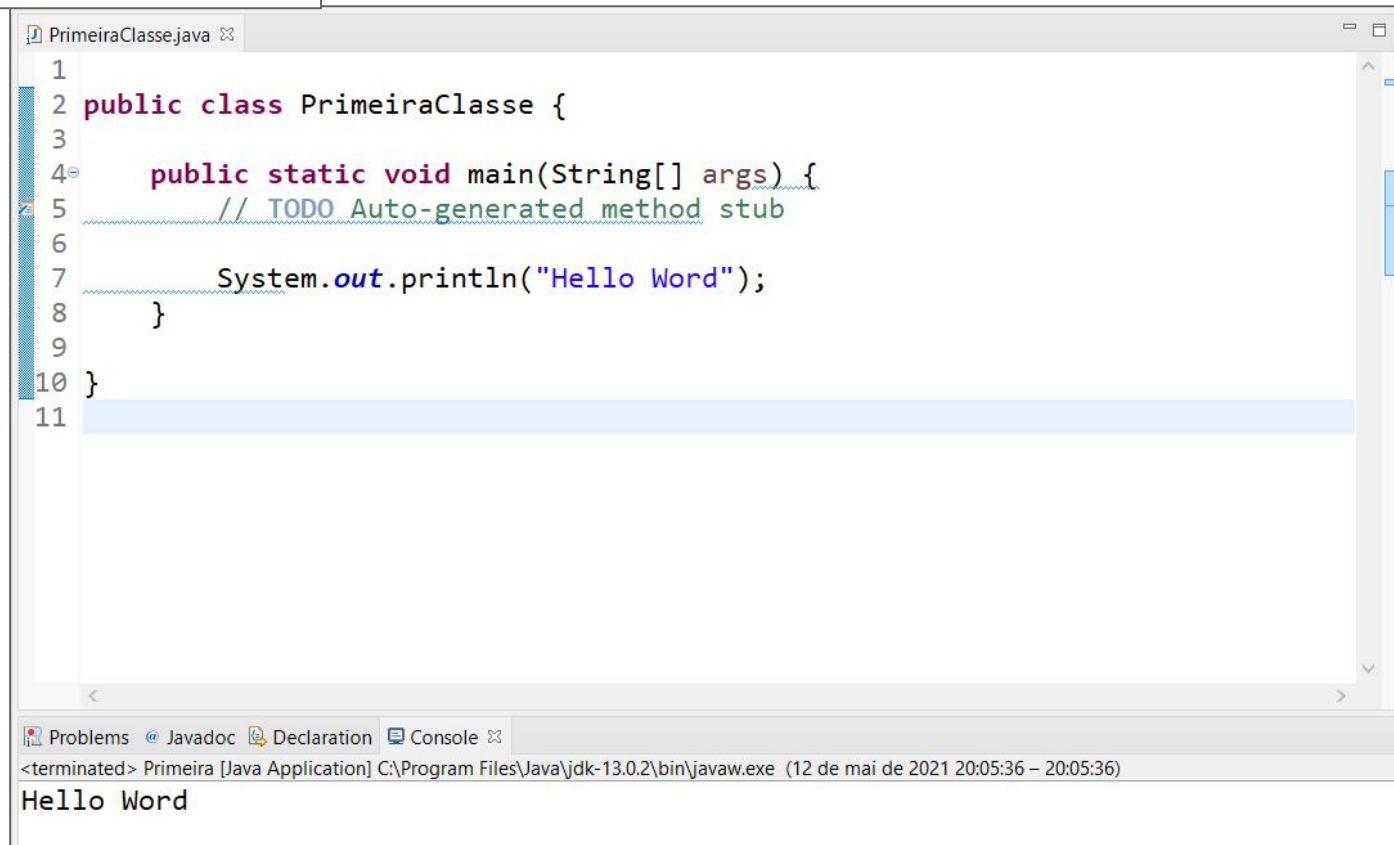
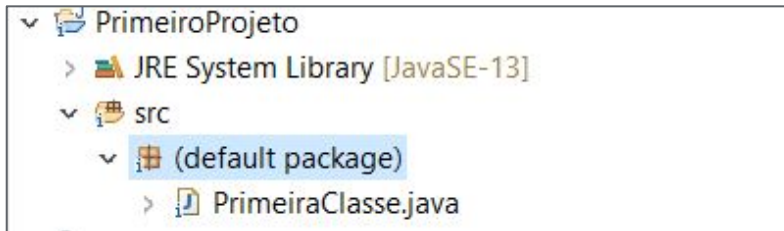
**Java Project** = Projeto com código-fonte na linguagem Java



# Eclipse IDE - Conceitos Básicos

- Precisa do Java instalado
    - JDK = Java Development Kit
    - <https://www.oracle.com/java/technologies/javase-jdk16-downloads.html>
  - Perspectivas
    - Visões diferentes do Projeto
  - Para Executar
    - Run As -> Java Application
-

# Eclipse IDE





# BlueJ



## Porque O BlueJ?

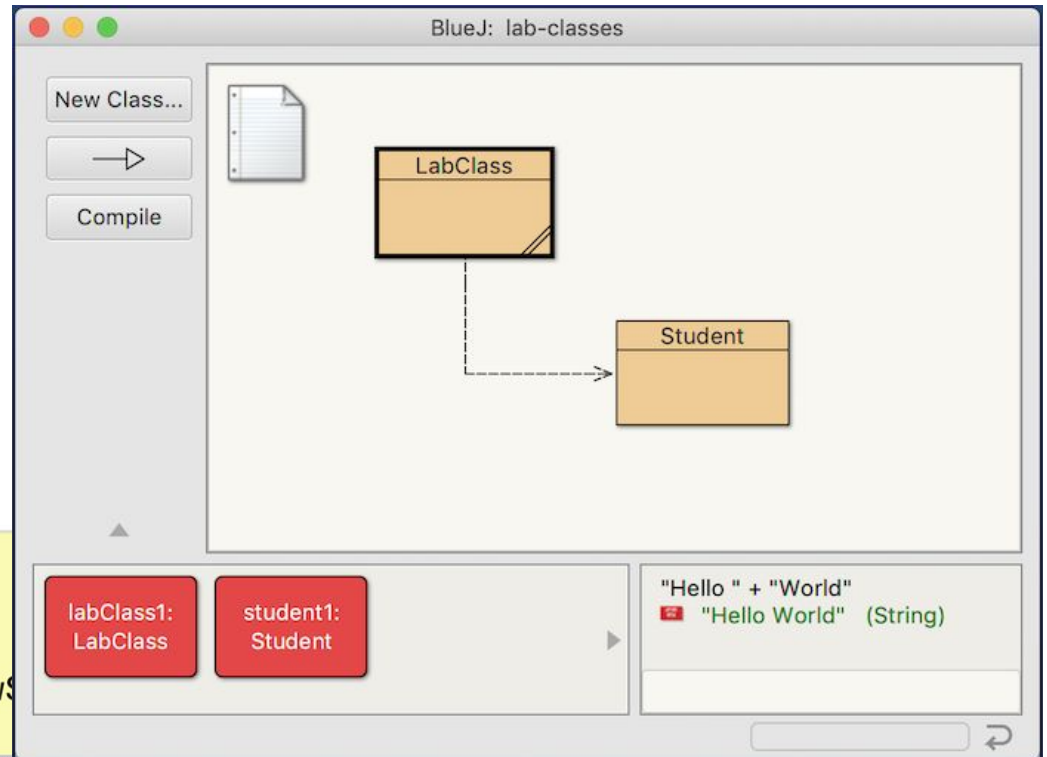
- BlueJ é uma IDE para desenvolvimento Java
- Desenvolvido principalmente para iniciação na programação orientada a objetos
- Estrutura do programa em desenvolvimento como um diagrama UML
- Visualização da modelagem das classes

## Download

- <https://www.bluej.org/>
-



# BlueJ



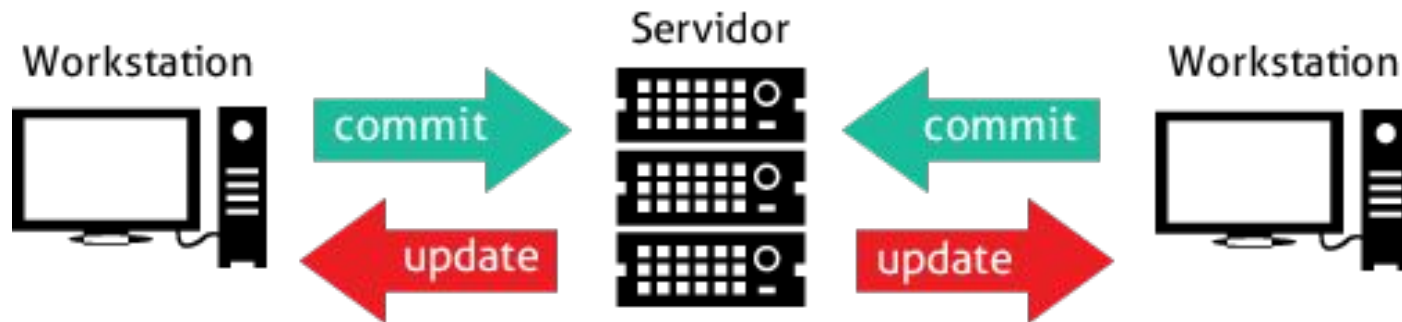
```
/**
 * Add a student to this LabClass.
 */
public void enrollStudent(Student newStudent) {
    if (students.size() == capacity) {
        System.out.println("The class is full.");
    }
    else {
        students.add(newStudent);
    }
}
```

***Agora vamos tratar de Versionamento de  
Código...***

---

# Versionamento de Código

- O que é Controle de versão?
  - É o registro de alterações em um arquivo ou conjunto de arquivos ao longo do tempo, para que você possa recuperar versões específicas mais tarde





# Versionamento de Código

- Sistemas de gerenciamento de Versões
  - Git
  - Subversion (SVN)
  - SourceSafe



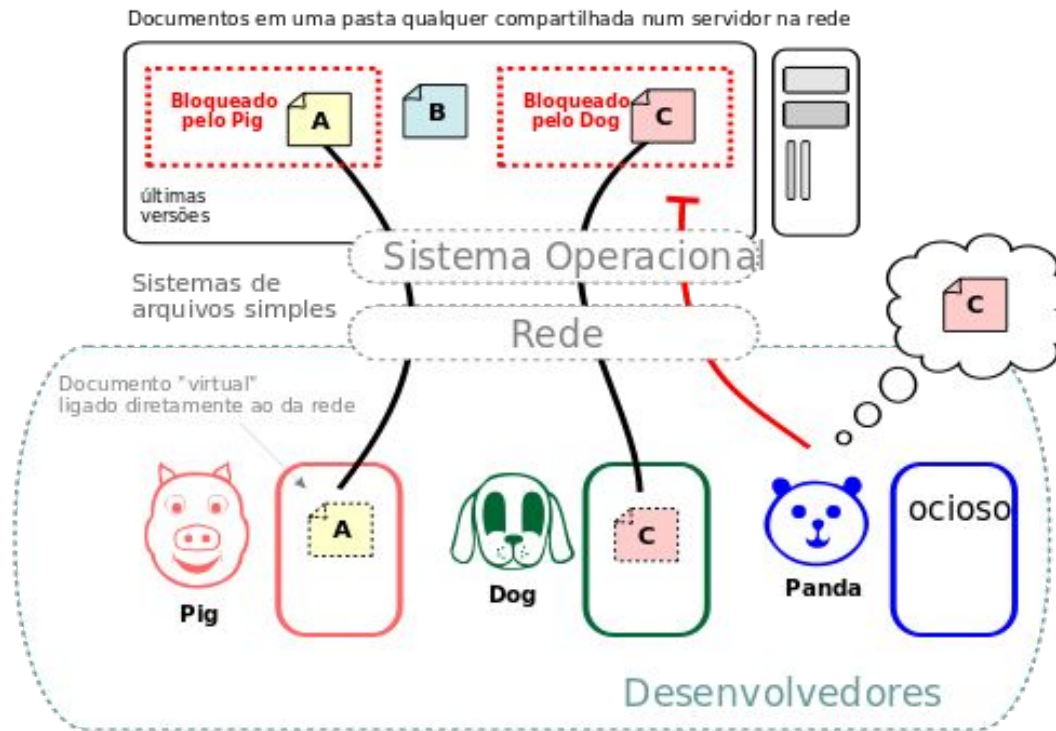
# Versionamento de Código

- **Benefícios do Controle de versão**
  - Histórico de alterações
  - Desenvolvimento Paralelo
  - Desenvolvedores trabalhando no mesmo código
  - Restaurar uma determinada versão
- **Sistemas de controle de versão também podem ser usados para armazenar**
  - documentação
  - manuais
  - relatórios, etc



# Controle de Versão

- Sem controle de versão
  - Arquivos ficam bloqueados enquanto estão sendo utilizados



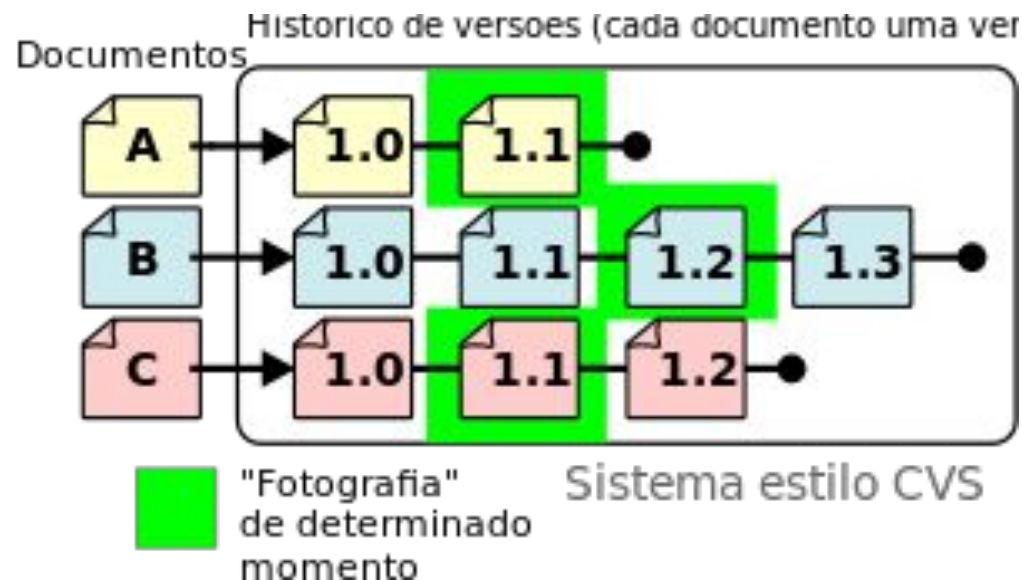
# Git e GitHub

- **Git**
  - Sistema de controle de versões distribuído
- **GitHub**
  - Serviço de hospedagem de código que usa o sistema Git para prover controle de versões
  - Repositórios públicos e gratuitos
  - Sistemas similares:
    - GitLab
    - BitBucket



# Versionamento Semântico

- Conjunto de políticas para gerenciar as diversas versões de um sistema
  - Útil para o gerenciamento de dependências
- Como vocês fazem para definir a versão do software?



# Versionamento Semântico

- Dado um número de versão MAJOR.MINOR.PATCH, incremente a:
  - versão Maior (MAJOR): quando fizer mudanças incompatíveis
  - versão Menor (MINOR): quando adicionar funcionalidades mantendo compatibilidade
  - versão de Correção (PATCH): quando corrigir falhas mantendo compatibilidade



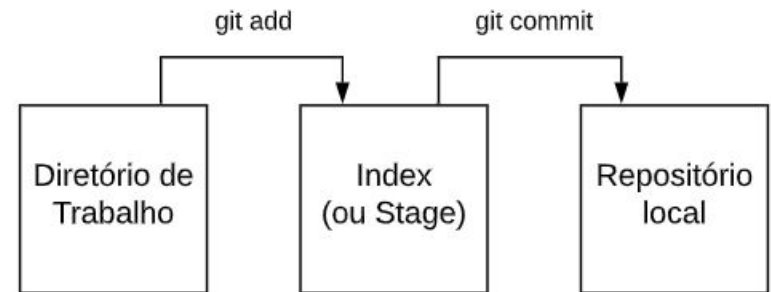
# Git

- Sistema de controle de versão distribuído
  - <https://git-scm.com/>
  - O pacote inclui o próprio Git, o Git Bash e o Git GUI, uma interface gráfica para o Git



# Comandos Básicos do Git

- Criar um repositório local vazio
  - **git init**
- Clonar um repositório
  - **git clone <nome do repositório>**
    - ele cria o repositório vazio e depois copiar todos os commits do repositório remoto
- Adicionar um ou mais arquivos
  - **git add <nome-arquivo>**
  - **git add \*.extensão**
  - **git add \***

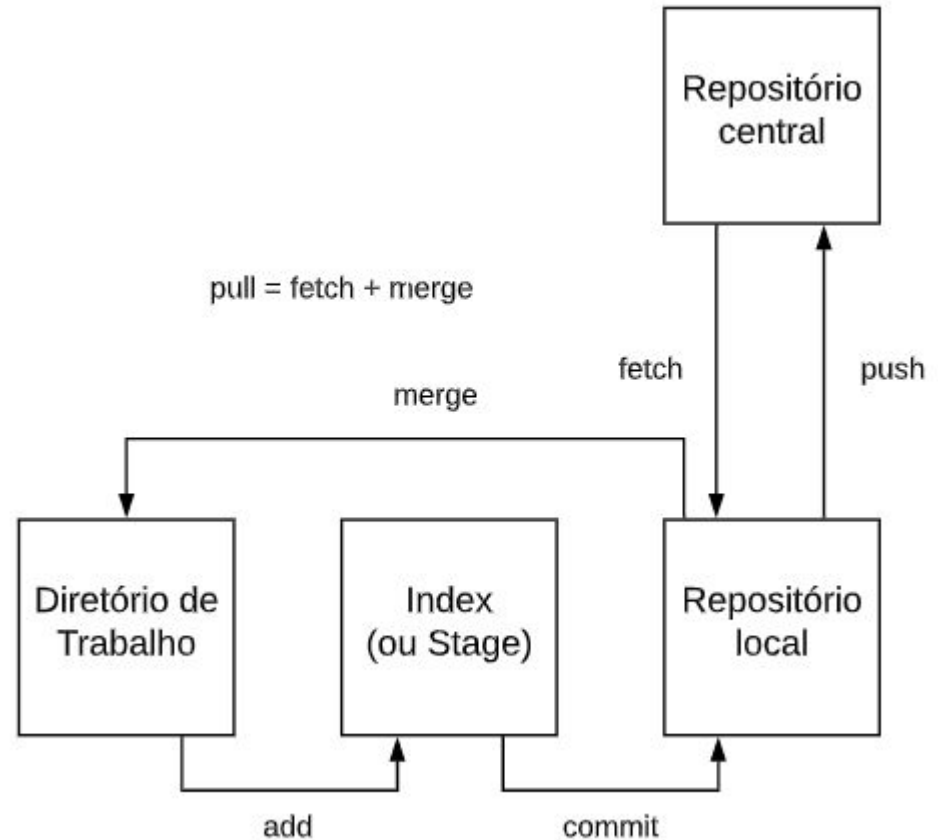




# Comandos Básicos do Git

## Push & Pull

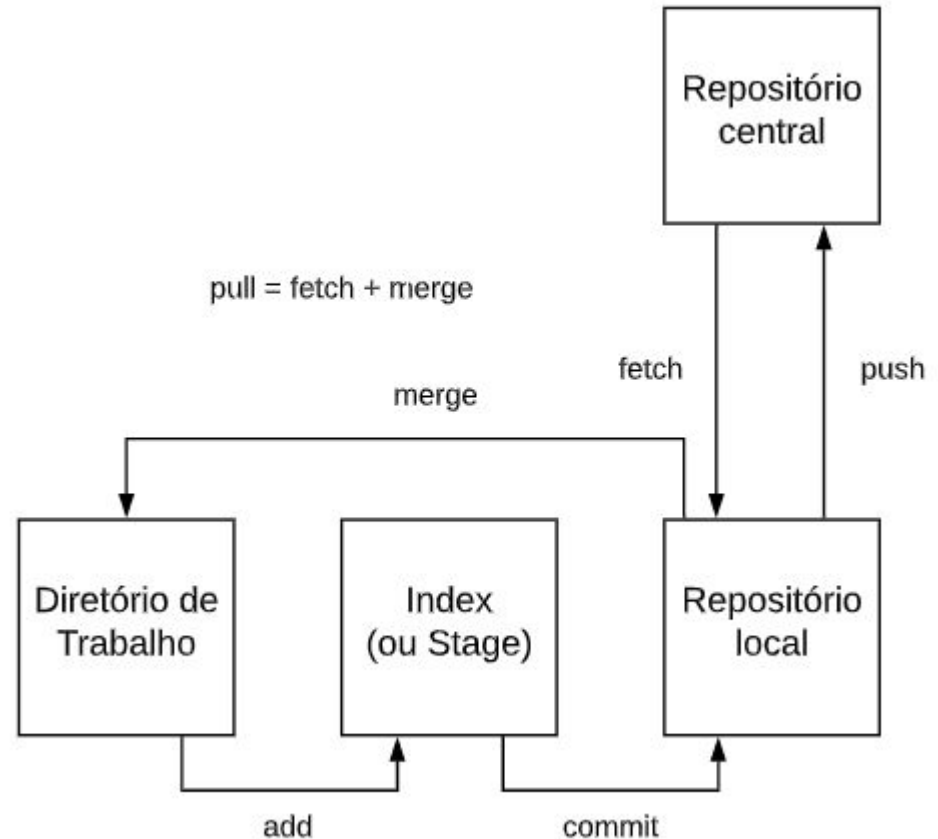
- Push
  - copia os commits mais recentes do repositório local para o repositório remoto



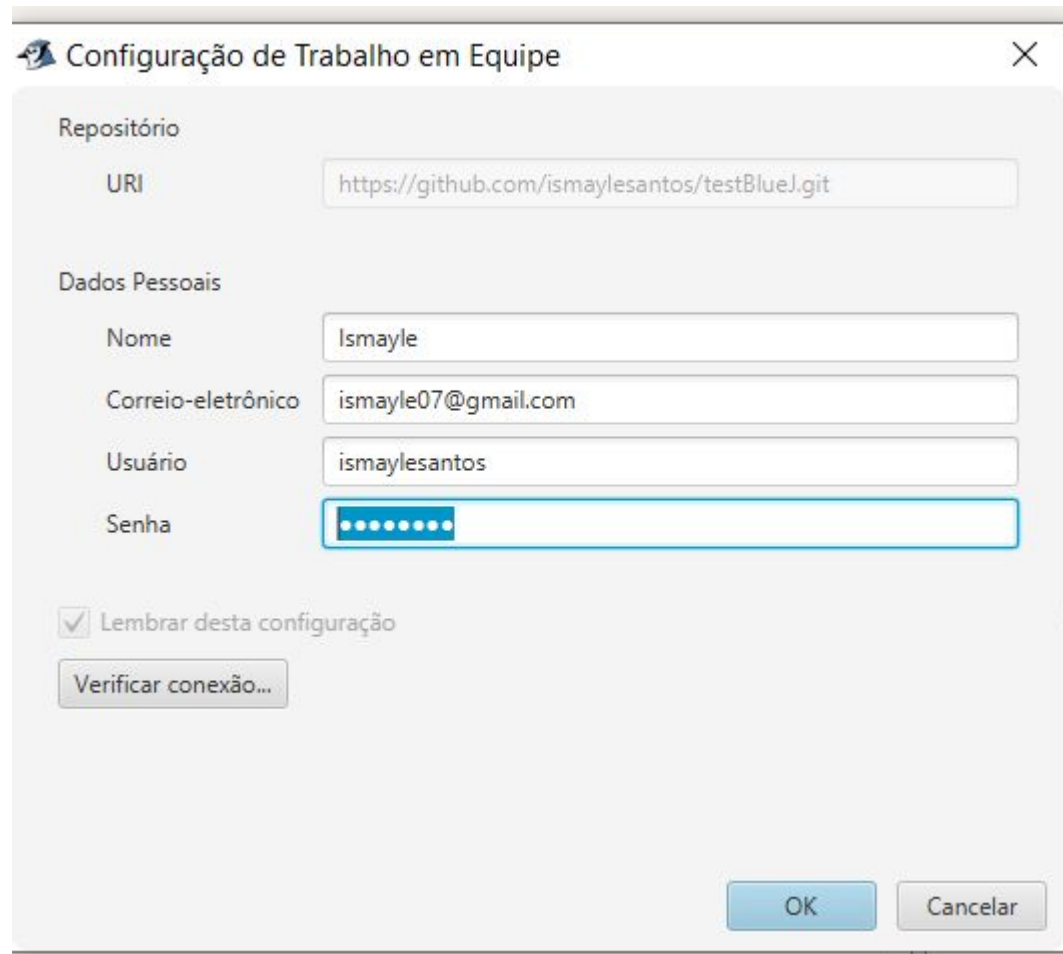
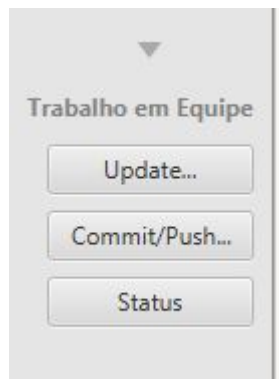
# Comandos Básicos do Git

## Push & Pull

- Pull
  - copia os arquivos do repositório central para o repositório local (fetch)
  - atualiza os arquivos do diretório de trabalho (merge)



# BlueJ + GitHub



A dialog box titled 'Configuração de Trabalho em Equipe' with a close button (X) in the top right corner. The dialog is divided into two sections: 'Repositório' and 'Dados Pessoais'.

**Repositório**

URI:

**Dados Pessoais**

Nome:

Correio-eletrônico:


Usuário:

Senha:

☒ Lembrar desta configuração


# Eclipse + GitHub


Owner \*      Repository name \*

 ismaylesantos ▾ / repositorioEclipse ✓

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

Description (optional)

☐  **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:**  
Skip this step if you're importing an existing repository.


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

☐ **Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)


☐ **Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).


Go to file    Add file ▾    **↓ Code ▾**


 **Clone** ⓘ

HTTPS   SSH   GitHub CLI

**https://github.com/ismaylesantos/repositorioEclipse** 

Use Git or checkout with SVN using the web URL.

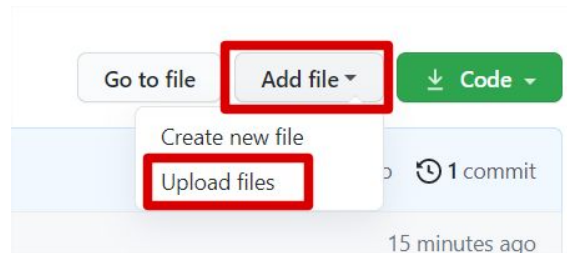
 **Open with GitHub Desktop**

 **Download ZIP**

# Eclipse + GitHub

**Existem várias maneiras, uma forma:**

1. Vamos criar o projeto no eclipse
2. Fazer o upload no GitHub
  - a. Basta fazer o upload a pasta toda do projeto



Menu do GitHub

3. Deletar o projeto local (marcando a opção para remover do disco)
4. Importar do GitHub, adicionando o repositório e fazendo o clone dele no Eclipse

Sugestão de vídeo: <https://www.youtube.com/watch?v=MdiolaiK9-Y>

---

# Eclipse + GitHub

## Outra forma:

### 1. Usando o Git Bash

```
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ismaylesantos/retoapagar.git
git push -u origin main
```

Quem quiser conhecer um pouco mais:

[https://learngitbranching.js.org/?locale=pt\\_BR](https://learngitbranching.js.org/?locale=pt_BR)



**Tem alguns  
links úteis no  
repositório**

# Projeto Final

*Comecem a  
organizar os  
grupos!*

- **Objetivo**
    - Desenvolver um sistema utilizando OO e um banco de dados relacional
  - **Deadline**
    - APF-1: 01.07.21 - Descrição do projeto
    - APF-2: 29.07.21 - Entrega parcial (foco nas classes e OO)
    - APF-3: 24.08.21 - Entrega do projeto completo
  - **Apresentações dos trabalhos Finais**
    - 26.08.21 à 01.09.21
  - **Linguagem: Python (usando OO) ou Java**
  - **Individual ou em grupos de até 4 alunos**
-

# Trabalho Prático - TP1

- TRABALHO PRÁTICO - TP1 (Aula Prática 18/05/2021)
    - Instalar Java JDK
      - <https://www.oracle.com/java/technologies/java-se-jdk16-downloads.html>
    - Instalar Eclipse IDE e/ou BlueJ
      - <https://www.eclipse.org/downloads/>
      - <https://www.bluej.org/>
    - Fazer Primeiro commit
      - Criar conta no GitHub e compartilhar com Professor/monitora
      - Criar 1º projeto e fazer 1º commit/push
-



# Trabalho Prático - TP1

- TRABALHO PRÁTICO - TP1 (Aula Prática 18/05/2021)
    - Forma de Entrega
      - Projeto do GitHub compartilhado comigo
      - 1 Print da tela da IDE no Google Classroom
-

# *Obrigado!*

## *Por hoje é só pessoal...*

### **Dúvidas?**



**rfbrkh3**



**ismaylesantos@great.ufc.br**



**@IsmayleSantos**

---