

# Técnicas de Programação

---

Prof. Protásio

# Informações gerais da disciplina

---

- Técnicas de Programação
  - Professor: Protásio
  - E-mail: [protasio@cear.ufpb.br](mailto:protasio@cear.ufpb.br)
  - Slides: envio pelo SIGAA
  - Horário:
    - terça e quinta (10h00-12h00)
  - Horário de atendimento:
    - terça e quinta (14h00-16h00)



# Avaliações

- **Reposição**
- Seguindo as normas da UFPB

- As avaliações serão totalmente **práticas** (verificação de códigos desenvolvidos):
- **Número de avaliações:** 3 + 1 final
  - **1<sup>a</sup> e 2<sup>a</sup> avaliações:**
    - Avaliação por meio de solicitação de um programa
    - Serão analisados:
      - Compilação dos códigos
      - Execução correta e completa dos códigos.
    - Avaliação Individual
  - **3 avaliação:**
    - Trabalho em grupo de 4 alunos com projeto aplicado em Engenharia.

## ▪ Listas de exercícios

- Todo final de slides, será disponibilizada uma lista de exercícios para fixação de conteúdo (não será cobrado)



# Conteúdo programático

---

- **Revisão de programação estruturada em C++**
  - Etapas da criação de um programa em C++, operadores, espaço de nomes, estruturas de controle de decisão e de laços iterativos, funções, passagens de parâmetros, funções inline, sobrecarga de funções, recursividade, modelos de funções, arrays unidimensionais e multidimensionais, variáveis ponteiro, alocação dinâmica de memória, passagem de parâmetros para programas com argc e argv
- **Conceitos e exemplos de algoritmos** (Ordenação de elementos e Busca sequencial).
- **Ponteiros e alocação dinâmica**
- **Manipulação de arquivos em C++** (canais, arquivos, leitura e escrita, acesso sequencial e aleatório em arquivos) e tratamento de exceção.
- **Programação de estruturas de dados** (containers, filas, pilhas, listas ligadas, Árvores binárias).
- **Programação orientada a objeto**
  - classes e objetos, visibilidade dos membros: público ou privado, encapsulamento, herança, polimorfismo
- Aplicação de conhecimentos em POO em **projeto aplicado**.

# CONTEÚDO PROGRAMÁTICO

- Ambiente de programação

Linha de Comando

IDE (VSCode)

- Paradigma de programação

Programação Estruturada

Programação Orientada a Objeto (POO)

- Interface com o usuário

console

CONSOLE

GUI

Duração da Disciplina

# SO, Editores de Códigos e Compiladores

---

- Sistemas Operacionais
  - Linux e Windows
- Editores de Códigos
  - Nano (Linux) e Notepad++ (Windows)
- Compiladores
  - gcc (linguagem C) e g++ (linguagem C++)
    - Em Linux, o gcc e g++ é **nativo**
    - Em Windows, é preciso instalar o **MinGW**
      - **MinGW** = *Minimalist GNU for Windows*
      - Nos últimos slides, tem-se os procedimentos de instalação.

# Ambiente de programação em Windows

---

Linha de Comandos

# Ambiente de programação em Windows

---

## ■ EDIÇÃO DO CÓDIGO-FONTE

### ■ Instale o **notepad++**

- É um editor de código (editor de texto + colorização)
- <https://notepad-plus-plus.org/downloads/>
  - Procedimentos de instalação no fim dos slides

## ■ COMPILADOR PARA WINDOWS

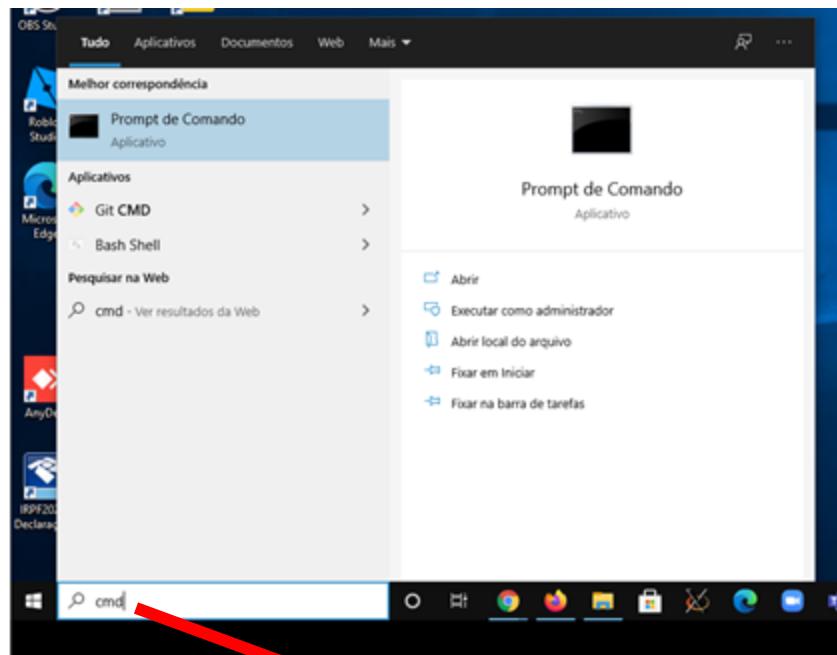
### ■ Instale o **MinGW** (**Minimalist GNU for Windows**)

- <https://sourceforge.net/projects/mingw/files/>
  - Procedimentos de instalação no fim dos slides

# Ambiente de programação em Windows

- Abrir “terminal” do windows

- Em executar, digitar:
  - cmd

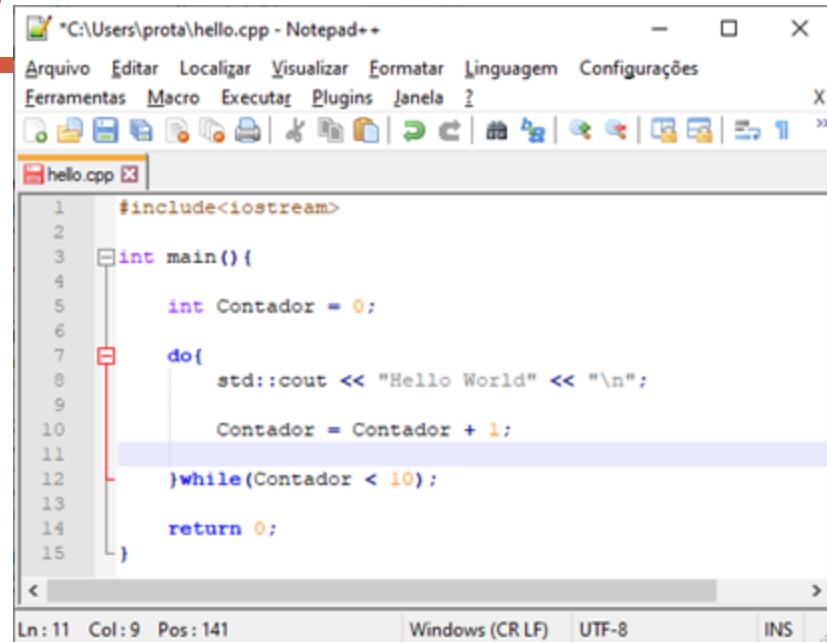


- No terminal, digitar:
  - g++

A screenshot of a Microsoft Windows Command Prompt window titled 'Prompt de Comando'. The window shows the command 'g++' being entered, followed by an error message: 'g++: fatal error: no input files compilation terminated.' The prompt then shows 'C:\Users\prota>' again.

# Ambiente de programação em Windows

- Usando o **notepad++**
    - No terminal
      - Abrir **editor de código**
        - No terminal, digite:
          - **start notepad++ hello.cpp**
        - **Digitar o código e salvar**
        - **Comando de compilação**
          - **g++ -o hello hello.cpp**
        - **Comando de execução**
          - **hello**



Microsoft Windows [versão 10.0.18363.1379]  
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\Users\proto>start notepad++ hello.cpp

```
C:\Users\proto>g++ -o hello hello.cpp
```

C:\Users\proto>hello

Hello World

# Hello World

C:\Users\ppnotes



# Ambiente de programação em Windows da disciplina

The screenshot shows a Windows desktop environment with two windows open:

- Notepad++ (File Path: G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2\hello.c - Notepad++)**: Contains the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Hello UFPB");
5     return 0;
}
```
- Prompt de Comando (File Path: Microsoft Windows [versão 10.0.17134.345])**: Shows the terminal session:

```
G:\>cd "Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2"
G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2>dir
O volume na unidade G é Google Drive File Stream
O Número de Série do Volume é 1983-1116

Pasta de G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2

21/11/2018 19:18    <DIR>      .
21/11/2018 17:58    <DIR>      ..
21/11/2018 18:27          165 ~$1_TP_Introducao.pptx
21/11/2018 18:27          924.177 1_TP_Introducao.pptx
21/11/2018 19:18          78 hello.c
                           3 arquivo(s)    924.420 bytes
                           2 pasta(s)   13.127.200.768 bytes disponíveis

G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2>gcc hello.c -o hello

G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2>dir
O volume na unidade G é Google Drive File Stream
O Número de Série do Volume é 1983-1116

Pasta de G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2

21/11/2018 19:23    <DIR>      .
21/11/2018 17:58    <DIR>      ..
21/11/2018 18:27          165 ~$1_TP_Introducao.pptx
21/11/2018 18:27          924.177 1_TP_Introducao.pptx
21/11/2018 19:18          78 hello.c
21/11/2018 19:23          27.930 hello.exe
                           4 arquivo(s)    952.350 bytes
                           2 pasta(s)   13.125.976.064 bytes disponíveis

G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2>hello
Hello UFPB
G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2>
```

The taskbar at the bottom shows the following icons and status:

- Windows Start button
- Search bar: "Digite aqui para pesquisar"
- Icons for File Explorer, Task View, Edge browser, and File Explorer
- Network connection icon
- Power icon
- Speaker icon
- Screen resolution icon
- File sharing icon
- File history icon
- POR (Portuguese) language indicator
- INTL (International) language indicator
- Date and time: 19:23 21/11/2018

# Meu primeiro fluxograma

---



- Tradução do fluxograma em um programa de computador
  - Linguagem C++
  - Ambiente de linha de comando
  - Sistema Operacional WINDOWS

# Meu primeiro programa



- Criar o **arquivo-fonte** do programa
  - start **notepad++ hello.cpp**

Arquivo-fonte em C++ deve ter **extensão .cpp**

Um editor de texto básico

Comando do Linux para Executar programas gráficos

```

1 #include <iostream>
2
3 int main(){
4
5     std::cout << "Hello World \n";
6
7     return 0;
8 }
9
10
11
12
13
14
15
  
```

Arquivo: C:\Users\prota\hello.cpp - Notepad++

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações  
Ferramentas Macro Executar Plugins Janela ?

Ln : 11 Col : 9 Pos : 141 Windows (CR LF) UTF-8 INS

# Compilação

- Compilar é traduzir o **código-fonte** em um **arquivo executável**.
- Comando para compilar

`g++ -o hello hello.cpp`

- Comando para executar

`hello`



A screenshot of a Microsoft Windows Command Prompt window titled "Prompt de Comando". The window shows the following text:  
Microsoft Windows [versão 10.0.18363.1379]  
(c) 2019 Microsoft Corporation. Todos os direitos reservados.  
C:\Users\prota> g++ -o hello hello.cpp  
  
C:\Users\prota> hello

# Tarefa

- Crie uma pasta com seu nome
- Entre na pasta
- Crie um novo código-fonte
- Mude o código para imprimir na tela:

Hello world

Seu Nome Completo



## Comandos de CMD

**pwd**

Mostra o diretório atual

**dir**

Lista arquivos do diretório atual

**mkdir nome\_diretorio**

Cria diretório

**cd nome\_diretorio**

Muda de diretório

**cd ..**

Retorna para diretório anterior

**cd ~**

Retorna para diretório do usuário

**start notepad++ hello.cpp**

Edição do código-fonte

**g++ -o hello hello.cpp**

Comando para compilar

**hello**

Comando para executar

# Tarefa

- Compile e execute o código abaixo:

```
#include <iostream>
using namespace std;

int main(){

    std::cout << "Calculadora Simples: \n";

    cout << "Soma" << 14 + 2 << "\n";
    cout << "Subtração" << 14 - 2 << "\n";
    cout << "Multiplicação" << 14 * 2 << "\n";
    cout << "Divisão" << 14 / 2 << "\n";

    return 0;
}
```

Calcule a média de 4, 8 e 9

Calcule  $3^2$

- Editar o código  
start **notepad++ calc.cpp**
- Compilar  
**g++ -o calc calc.cpp**
- Executar  
**calc**



# Tarefa

- Compile e execute o código abaixo:

```
#include <iostream>
using namespace std;

int main(){
    int x = 14;
    int y = 2;

    std::cout << "Calculadora Simples: \n";

    cout << "Soma" << x + y << "\n";
    cout << "Subtração" << x - y << "\n";
    cout << "Multiplicação" << x * y << "\n";
    cout << "Divisão" << x / y << "\n";

    return 0;
}
```

Calcule para x=22 e y = 11

- Editar o código  
start **notepad++ calc2.cpp**
- Compilar  
**g++ -o calc2 calc2.cpp**
- Executar  
**calc2**



# Tarefa

<< saída de dados (monitor)  
>> entrada de dados (teclado)

## ▪ Compile e execute o código abaixo:

```
#include <iostream>
using namespace std;

int main(){
    int x, y;
    cout << "Calculadora Simples: \n";
    cout << "Digite o valor de x: \n";
    cin >> x;
    cout << "Digite o valor de y: \n";
    cin >> y;

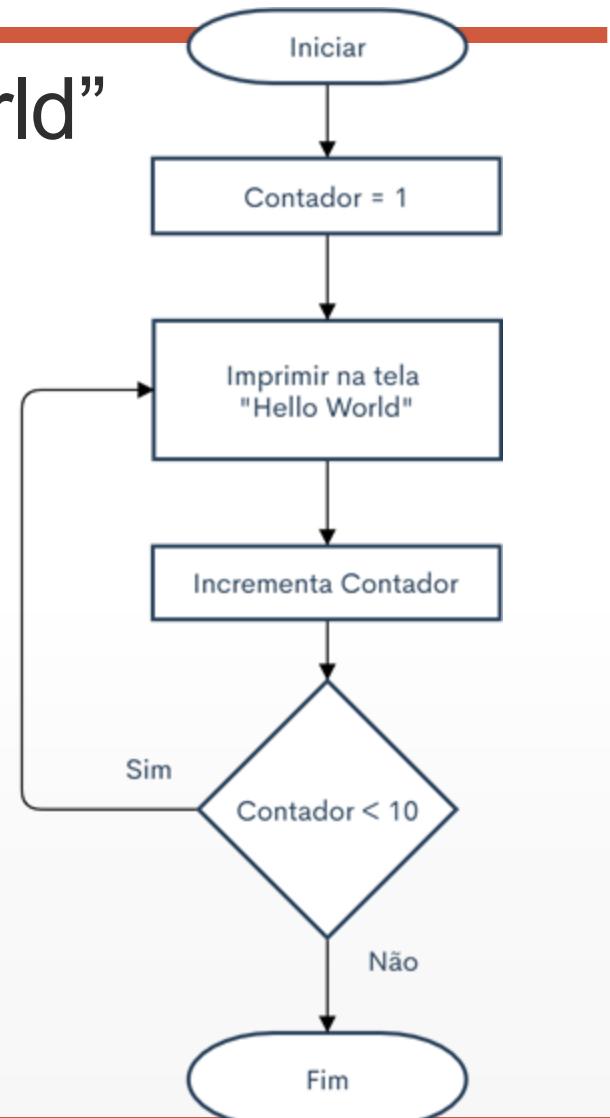
    cout << "Soma" << x + y << "\n";
    cout << "Subtração" << x - y << "\n";
    cout << "Multiplicação" << x * y << "\n";
    cout << "Divisão" << x / y << "\n";

    return 0;
}
```

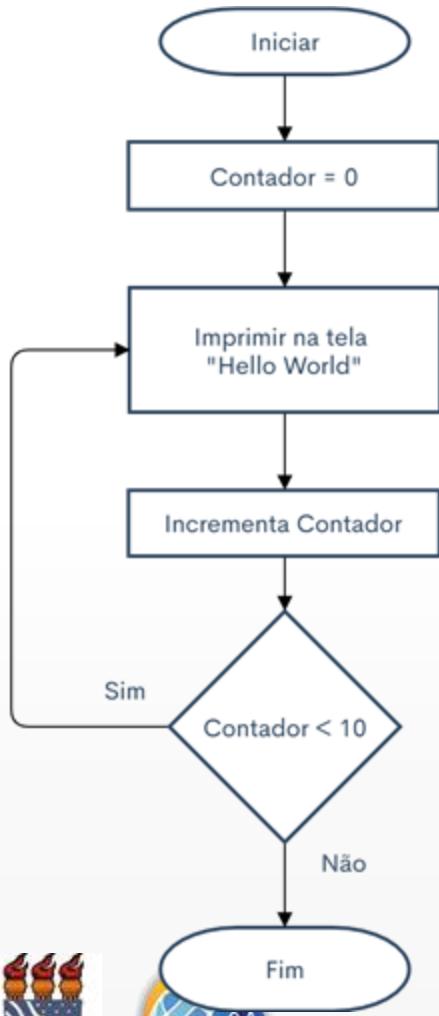
- Editar o código  
start **notepad++ calc3.cpp**
- Compilar  
**g++ -o calc3 calc3.cpp**
- Executar  
**calc3**

# Meu primeiro loop

- Imprime 10 vezes “hello world”
  - Analise o fluxograma
  - Avalie se está correto
    - Se não, corrija-o
  - Ao fim, inicie a programação



# Meu primeiro loop



```
#include <iostream>

int main(){
    int Contador = 0;
    do{
        std::cout << "Hello World \n";
        Contador = Contador + 1;
    }while(Contador < 10);

    return 0;
}
```

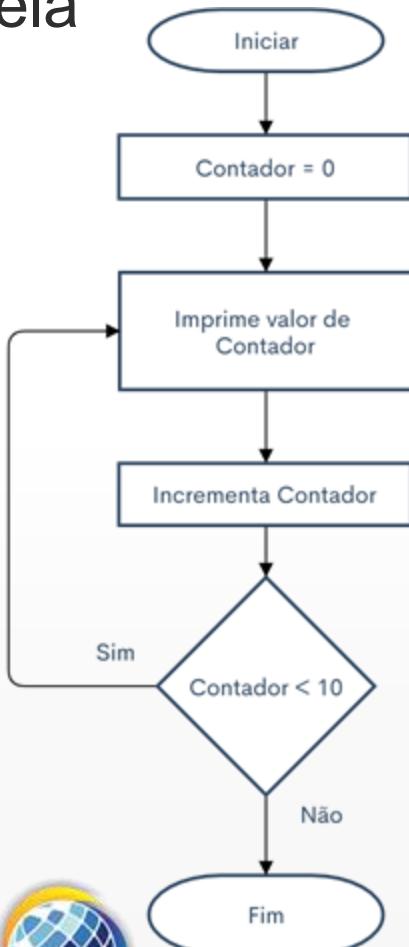
A red circular arrow icon is placed next to the 'do' keyword in the code, indicating the start of a loop.

- Salve como `loop1.cpp`
- Compile e execute
  - `g++ -o loop1 loop1.cpp`
  - `loop1`

Mude para que seja impresso 100 vezes

# Meu segundo loop

- Faça um programa que imprima os números de 1 a 10 na tela



```
#include <iostream>

int main(){

    int Contador = 0;

    do{
        std::cout << Contador << "\n";

        Contador = Contador + 1;

    }while(Contador < 10);

    return 0;
}
```

- Save as loop2.cpp
- Compile and run
  - g++ -o loop2 loop2.cpp
  - loop2

Change to print  
10, 20, 30,..., 100

# Meu loop controlado pelo usuário

- Faça um programa que recebe do usuário o valor máximo a ser impresso

```
#include <iostream>

int main(){
    int Vmax;

    std::cout << " Digite Valor Máximo: \n";
    std::cin >> Vmax;

    int Contador = 0;

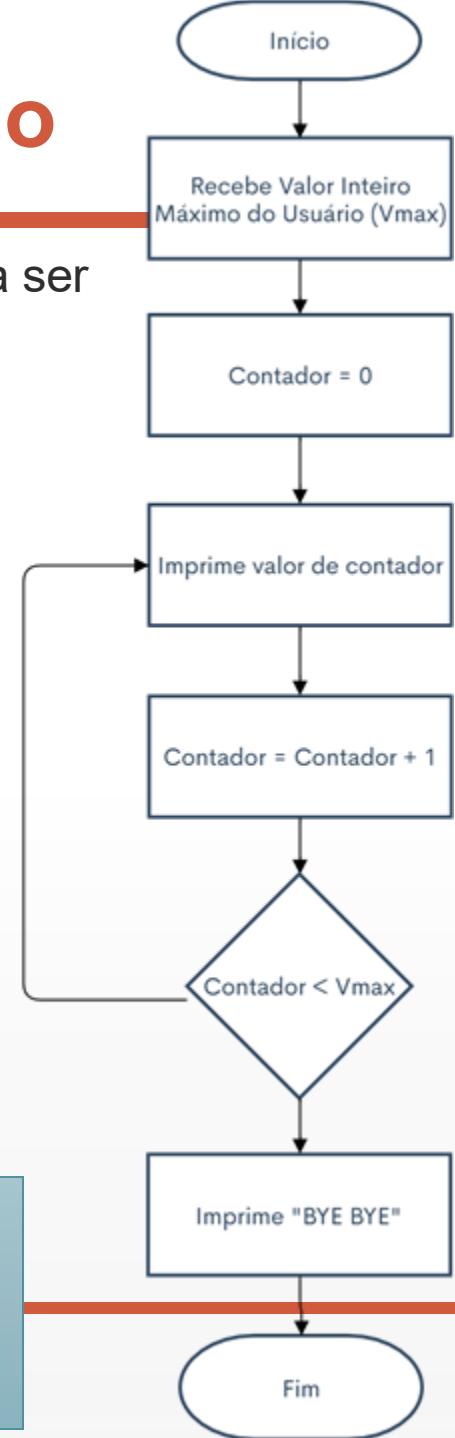
    do{
        std::cout << Contador << "\n";
        Contador = Contador + 1;
    }while(Contador < Vmax);

    return 0;
}
```

- Salve como loop3.cpp
- Compile e execute
  - g++ -o loop3 loop3.cpp
  - loop3

**loop4.cpp**  
Mude para que seja  
impresso de  
[0 até 2 vezes Vmax]

**loop5.cpp**  
Mude para que seja  
impresso de  
[-Vmax até Vmax]



# 1<sup>a</sup> Lista de Exercício

- Para cada tarefa abaixo, faça **um programa com código-fonte**:
  - **lista1\_nome\_sobrenome\_x.cpp** em que x é o número da tarefa
  - Imprima na tela:
    1. Seu nome completo e os números entre 0 e sua idade.
    2. Os valores entre 1 e 100 e em seguida de 100 a 1.
    3. Os valores entre 1 e N com passo P, em que N e P são dados pelo usuário.
      - 1. Exemplo: se N = 7 e P = 2, o resultado deve ser  
1 3 5 7
    4. O padrão ao lado em que N é dado pelo usuário.
- **Fluxogramas**
  - Coloque em um arquivo DOC todos os fluxogramas **lista1\_nome\_sobrenome.doc**
- **Entrega**
  - Compacte todos os 4 códigos-fonte e o DOC em um arquivo ZIP e envie este pelo SIGAA

```
1
12
123
1234
:
12345..N
```

# Usando o VSCode

---

# Ambiente de programação em Windows

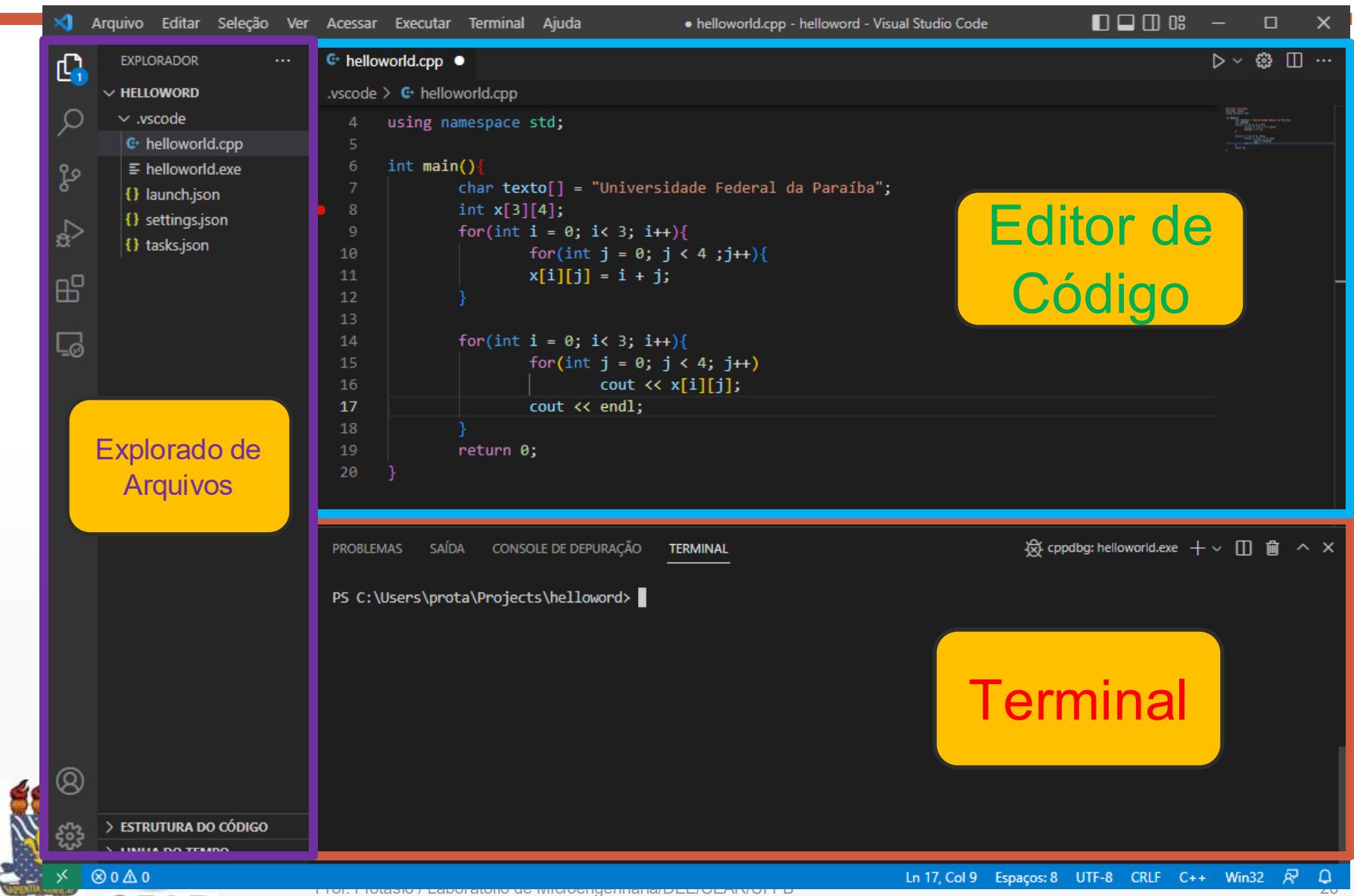
---

- Usando o VSCode
  - No terminal
    - Abrir editor de código
      - No terminal, digite:
        - `.code hello.cpp`
      - Digitar o código e salvar
      - Comando de compilação
        - `g++ -o hello hello.cpp`
      - Comando de execução
        - `hello`



Visual Studio Code

# VSCode



Terminal

# Meu primeiro fluxograma

---



- Tradução do fluxograma em um programa de computador
  - Linguagem C++
  - Ambiente de linha de comando
  - Sistema Operacional WINDOWS

Comando do Linux para Executar programas gráficos

# Meu primeiro programa



- Criar o **arquivo-fonte** do programa
  - `.code hello.cpp`

A screenshot of the Visual Studio Code interface. The title bar shows "Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda" and the path "helloworld.cpp - helloworld - Visual Studio Code". The Explorer sidebar shows a project structure with "EXPLORADOR", "HELLWORD", ".vscode", and files "helloworld.cpp", "helloworld.exe", "launch.json", "settings.json", and "tasks.json". The main editor area displays the following C++ code:

```
#include <iostream>

int main(){
    std::cout << "Hello World \n";
    return 0;
}
```

The code editor has syntax highlighting for C++. The terminal at the bottom shows the command prompt "PS C:\Users\protá\Projects\helloworld>".



# Compilação

- Compilar é traduzir o **código-fonte** em um **arquivo executável**.

- Comando para compilar

```
g++ -o hello hello.cpp
```

- Comando para executar

```
hello
```

The screenshot shows the Visual Studio Code interface with a dark theme. On the left is the Explorer sidebar showing a project folder 'HELLWORD' containing files: 'hello.cpp', 'helloworld.exe', '.vscode' (with 'launch.json', 'settings.json', and 'tasks.json'), and '.gitignore'. The main editor area displays the following C++ code:

```
#include <iostream>

int main(){
    std::cout << "Hello World \n";
    return 0;
}
```

Below the editor, the 'TERMINAL' tab is active, showing the command 'g++ -o hello hello.cpp' entered and the output 'hello'.

# Tarefa

- Crie uma pasta com seu nome
- Entre na pasta
- Crie um novo código-fonte
- Mude o código para imprimir na tela:

Hello world

Seu Nome Completo

A screenshot of a terminal window titled "TERMINAL". The window has tabs at the top: PROBLEMAS, SAÍDA, CONSOLE DE DEPURAÇÃO, and TERMINAL. The TERMINAL tab is selected. The terminal output is as follows:

```
PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL
PS C:\Users\prota\Projects\helloworld> 
          ^
```

The terminal shows the user's path: C:\Users\prota\Projects\helloworld. There is a small red bar at the bottom of the terminal window.

## Comandos de CMD

**pwd**

Mostra o diretório atual

**dir**

Lista arquivos do diretório atual

**mkdir nome\_diretorio**

Cria diretório

**cd nome\_diretorio**

Muda de diretório

**cd ..**

Retorna para diretório anterior

**cd ~**

Retorna para diretório do usuário

**start notepad++ hello.cpp**

Edição do código-fonte

**g++ -o hello hello.cpp**

Comando para compilar

**hello**

Comando para executar

# Tarefa

- Compile e execute o código abaixo:

```
#include <iostream>
using namespace std;

int main(){

    std::cout << "Calculadora Simples: \n";

    cout << "Soma" << 14 + 2 << "\n";
    cout << "Subtração" << 14 - 2 << "\n";
    cout << "Multiplicação" << 14 * 2 << "\n";
    cout << "Divisão" << 14 / 2 << "\n";

    return 0;
}
```

Calcule a média de 4, 8 e 9

Calcule  $3^2$

- Editar o código  
code calc.cpp
- Compilar  
g++ -o calc calc.cpp
- Executar  
calc



# Tarefa

- Compile e execute o código abaixo:

```
#include <iostream>
using namespace std;

int main(){
    int x = 14;
    int y = 2;

    std::cout << "Calculadora Simples: \n";

    cout << "Soma" << x + y << "\n";
    cout << "Subtração" << x - y << "\n";
    cout << "Multiplicação" << x * y << "\n";
    cout << "Divisão" << x / y << "\n";

    return 0;
}
```

Calcule para x=22 e y = 11

- Editar o código  
code calc2.cpp
- Compilar  
g++ -o calc2 calc2.cpp
- Executar  
calc2



# Tarefa

<< saída de dados (monitor)  
>> entrada de dados (teclado)

## ▪ Compile e execute o código abaixo:

```
#include <iostream>
using namespace std;

int main(){
    int x, y;
    cout << "Calculadora Simples: \n";
    cout << "Digite o valor de x: \n";
    cin >> x;
    cout << "Digite o valor de y: \n";
    cin >> y;

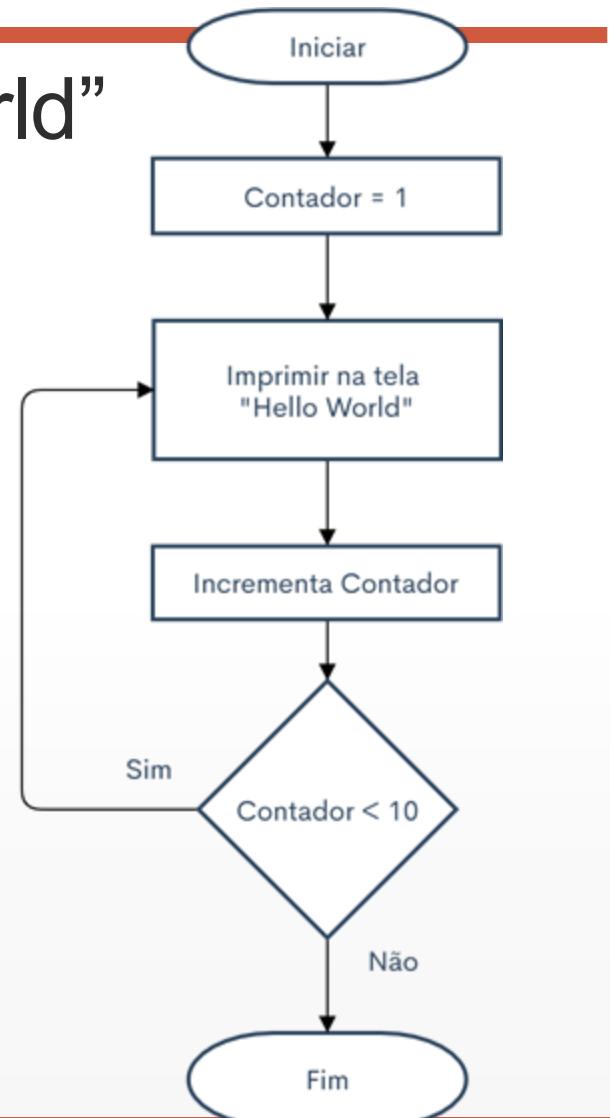
    cout << "Soma" << x + y << "\n";
    cout << "Subtração" << x - y << "\n";
    cout << "Multiplicação" << x * y << "\n";
    cout << "Divisão" << x / y << "\n";

    return 0;
}
```

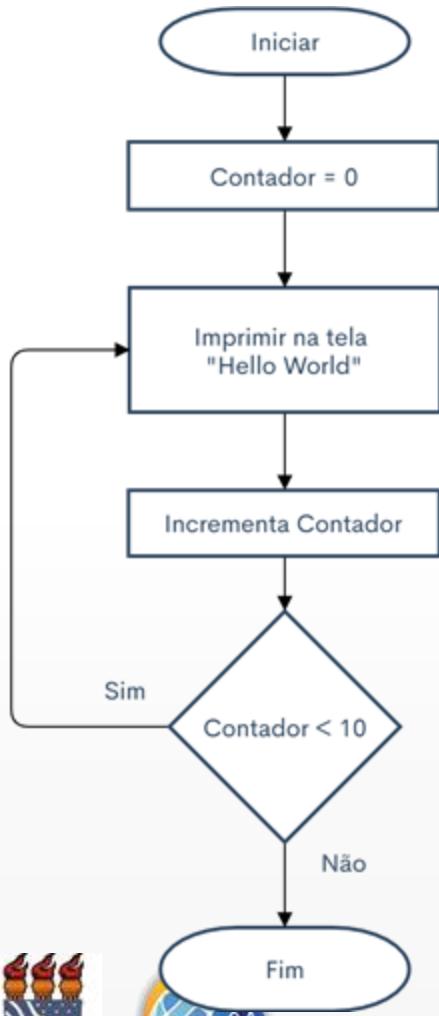
- Editar o código  
code calc3.cpp
- Compilar  
g++ -o calc3 calc3.cpp
- Executar  
calc3

# Meu primeiro loop

- Imprime 10 vezes “hello world”
  - Analise o fluxograma
  - Avalie se está correto
    - Se não, corrija-o
  - Ao fim, inicie a programação



# Meu primeiro loop



```
#include <iostream>

int main(){
    int Contador = 0;
    do{
        std::cout << "Hello World \n";
        Contador = Contador + 1;
    }while(Contador < 10);

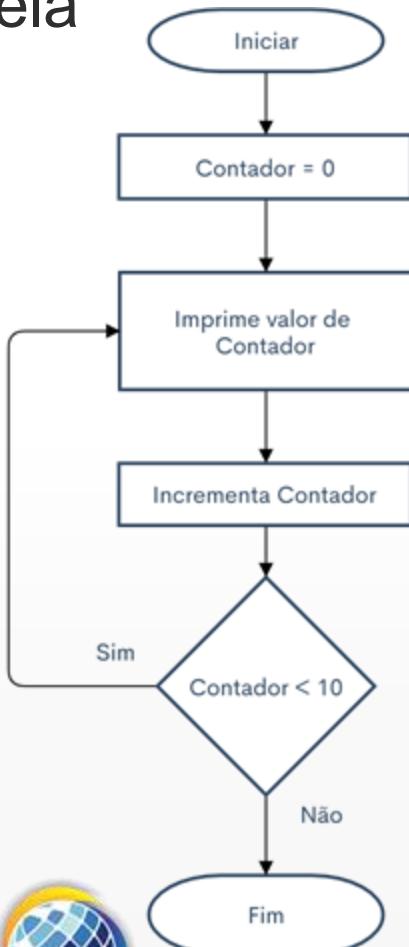
    return 0;
}
```

- Salve como `loop1.cpp`
- Compile e execute
  - `g++ -o loop1 loop1.cpp`
  - `loop1`

Mude para que seja impresso 100 vezes

# Meu segundo loop

- Faça um programa que imprima os números de 1 a 10 na tela



```
#include <iostream>

int main(){
    int Contador = 0;
    do{
        std::cout << Contador << "\n";
        Contador = Contador + 1;
    }while(Contador < 10);

    return 0;
}
```

- Save as loop2.cpp
- Compile and run
  - g++ -o loop2 loop2.cpp
  - loop2

Mude para que seja impresso  
10, 20, 30,..., 100

# Meu loop controlado pelo usuário

- Faça um programa que recebe do usuário o valor máximo a ser impresso

```
#include <iostream>

int main(){
    int Vmax;

    std::cout << " Digite Valor Máximo: \n";
    std::cin >> Vmax;

    int Contador = 0;

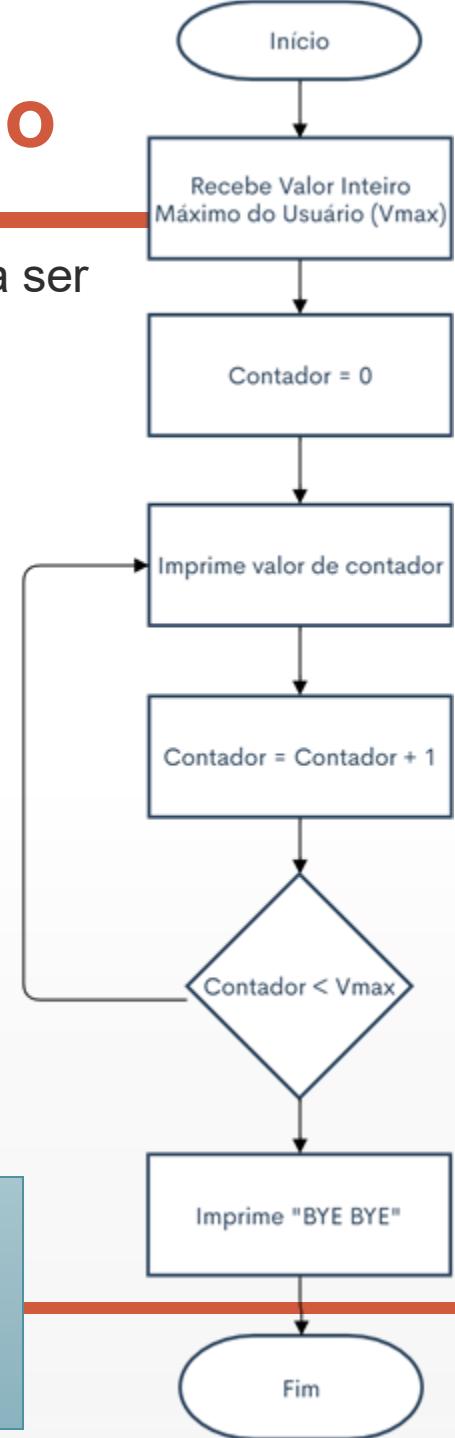
    do{
        std::cout << Contador << "\n";
        Contador = Contador + 1;
    }while(Contador < Vmax);

    return 0;
}
```

- Salve como loop3.cpp
- Compile e execute
  - g++ -o loop3 loop3.cpp
  - loop3

**loop4.cpp**  
Mude para que seja  
impresso de  
[0 até 2 vezes Vmax]

**loop5.cpp**  
Mude para que seja  
impresso de  
[-Vmax até Vmax]



# 1<sup>a</sup> Lista de Exercício

- Para cada tarefa abaixo, faça **um programa com código-fonte**:
  - **lista1\_nome\_sobrenome\_x.cpp** em que x é o número da tarefa
  - Imprima na tela:
    1. Seu nome completo e os números entre 0 e sua idade.
    2. Os valores entre 1 e 100 e em seguida de 100 a 1.
    3. Os valores entre 1 e N com passo P, em que N e P são dados pelo usuário.
      - 1. Exemplo: se N = 7 e P = 2, o resultado deve ser  
1 3 5 7
    4. O padrão ao lado em que N é dado pelo usuário.
- **Fluxogramas**
  - Coloque em um arquivo DOC todos os fluxogramas **lista1\_nome\_sobrenome.doc**
- **Entrega**
  - Compacte todos os 4 códigos-fonte e o DOC em um arquivo ZIP e envie este pelo SIGAA

```
1
12
123
1234
:
12345..N
```

# Conteúdo Extra

---

# Comparação da execução em linha de comandos: Windows e Linux

## Windows

- Abrir **terminal**
- Abrir **editor de código**
  - No terminal, digite:
    - `start notepad++ hello.c`
- **Digitar o código e salvar**
- **Comando de compilação**
  - `g++ -o hello hello.cpp`
- **Comando de execução**
  - `hello`

## Linux

- Abrir **terminal**
- Abrir **editor de código**
  - No terminal, digite:
    - `nano hello.cpp`
- **Digitar o código e salvar**
- **Comando de compilação**
  - `g++ -o hello hello.cpp`
- **Comando de execução**
  - `./hello`



# Alguns comandos LINUX e WINDOWS

## Linux

- **ls**
  - Lista arquivos do diretório atual
- **pwd**
  - Mostra a pasta atual
- **mkdir nome\_diretorio**
  - Criar diretório
- **cd**
  - Mudar diretório atual
- **mv origem destino**
  - Mover arquivo
- **rm nome\_arquivo**
  - Deletar arquivo
- **more nome\_arquivo**
  - Visualizar arquivos (uma tela por vez)
- **man nome\_comando**
  - Ajuda e descrição do comando

## Windows

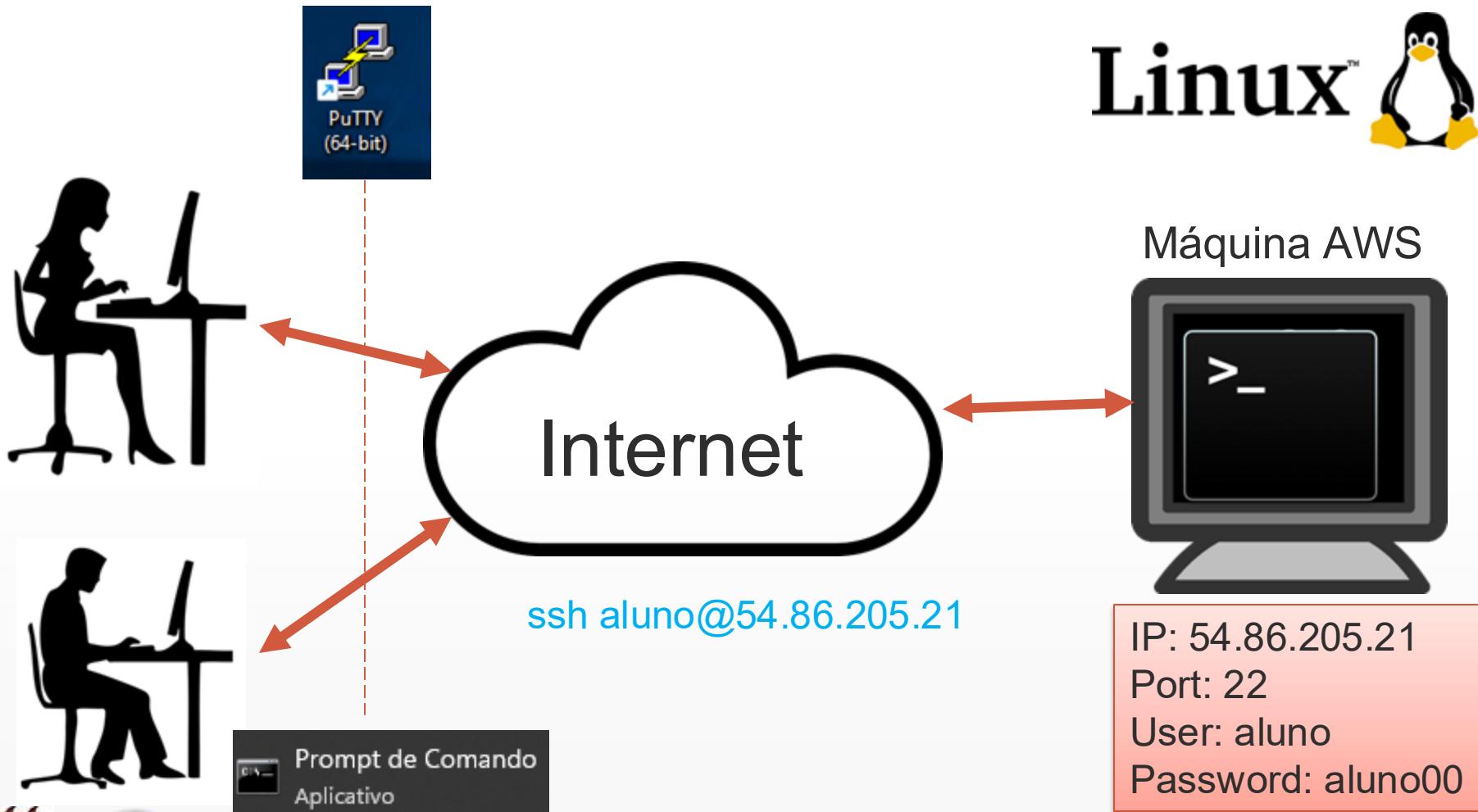
- **dir**
  - Lista arquivos do diretório atual
- **pwd**
  - Mostra a pasta atual
- **mkdir nome\_diretorio**
  - Criar diretório
- **cd**
  - Mudar diretório atual
- **move origem destino**
  - Mover arquivo
- **del nome\_arquivo**
  - Deletar arquivo
- **more nome\_arquivo**
  - Visualizar arquivos (uma tela por vez)
- **help nome\_comando**
  - Ajuda e descrição do comando



# Ambiente de programação em Linux/Máquina Remota

---

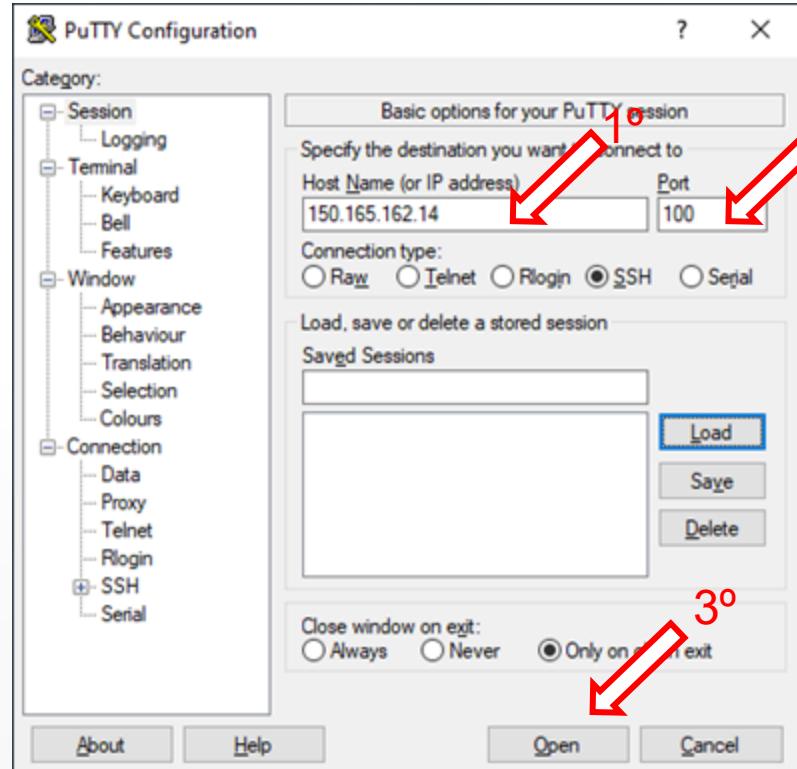
# Meu primeiro acesso remoto



# Meu primeiro acesso remoto

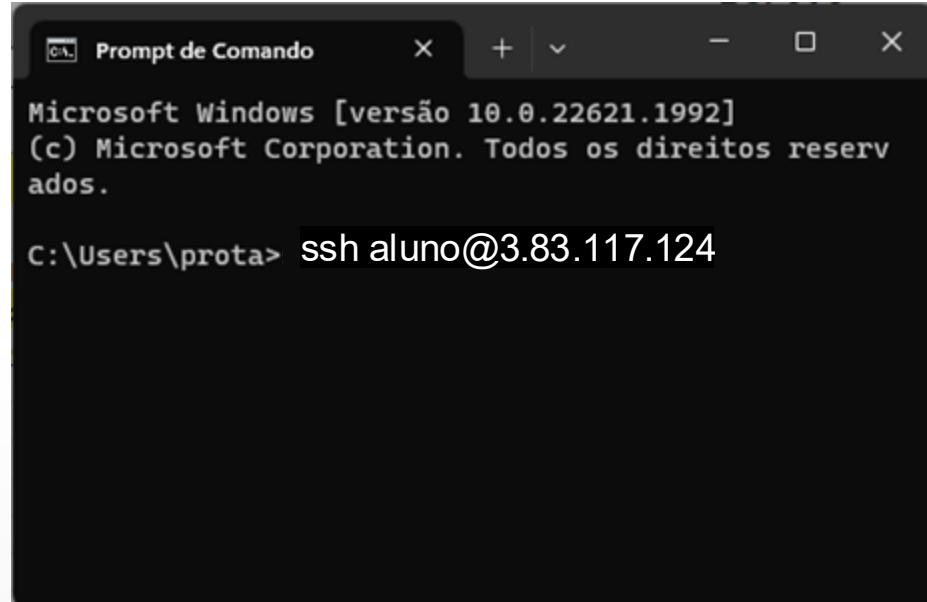
AWS  
IP: 3.83.117.124  
Port: 22  
User: aluno  
Password: aluno00

- Execute o PuTTY



OU

- Execute o CMD



ssh aluno@150.165.162.14 -p 100

# Meu primeiro acesso remoto

AWS  
IP: 3.83.117.124  
Port: 22  
User: aluno  
Password: aluno00

```
aluno@microengenharia-s5715br: ~
login as: aluno
aluno@150.165.162.14's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-66-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

7 packages can be updated.
1 of these updates is a security update.
To see these additional updates run: apt list --upgradable

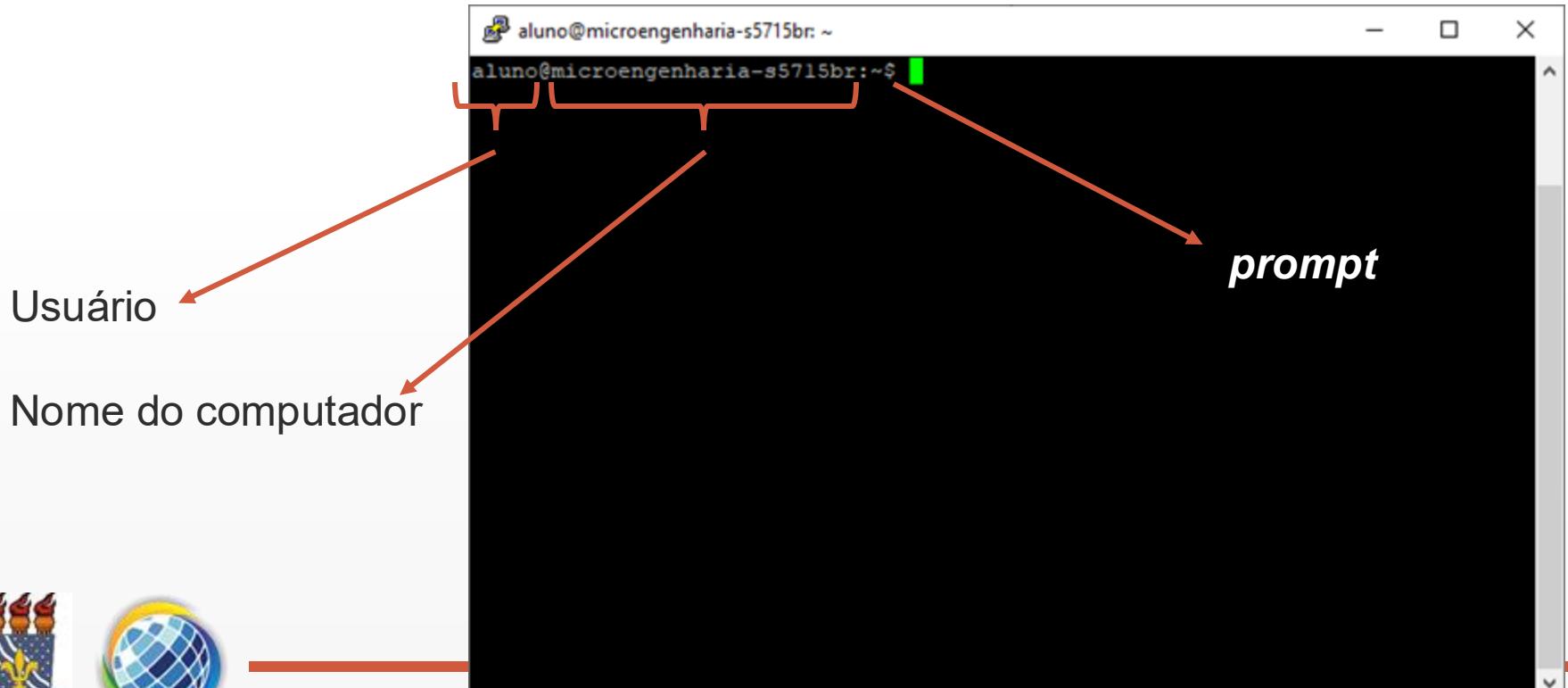
New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Wed Mar  3 17:34:04 2021 from 179.183.184.123
aluno@microengenharia-s5715br:~$
```



# Meu primeiro acesso remoto

- Terminal (CONSOLE) e Linha de Comando
  - No *prompt* \$ digite o comando *clear*



# Meus primeiros comandos LINUX

## ■ **pwd**

- Mostra o diretório atual

## ■ **ls**

- Lista arquivos do diretório atual

### ▪ **ls -all**

- Lista todos os arquivos do diretório atual (ocultos)

## ■ **mkdir nome\_diretorio**

- Criar diretório

## ■ **cd nome\_diretorio**

- Muda de diretório

## ■ **cd ..**

- Retorna para diretório anterior

## ■ **cd ~**

Retorna para diretório do usuário



```
aluno@microengenhariaDELL02: ~/protasio
GabrielNascimento          loop4.cpp
gabriel_veriato           luan
gil                         luan_lacerda
Gisela                      luan_novo
GlauberR                   luan_pedro
Gustavo                     lucas32
gustavohra                 Lucas_Dantas
gustavohra_diretorio       Lucca
Gustavo_Souza              luis_netto
Helio_B                     manoelz
hello                       marcolino_diretorio
hello.cpp                  Marcos
Hyago                       MariaThais
Iaácob                      mario
igor                         Mario
Isabelle                    Mateus_Navarro
aluno@microengenhariaDELL02:~$ pwd
/home/aluno
aluno@microengenhariaDELL02:~$ mkdir protasio
aluno@microengenhariaDELL02:~$ cd protasio
aluno@microengenhariaDELL02:~/protasio$ ls
aluno@microengenhariaDELL02:~/protasio$ pwd
/home/aluno/protasio
aluno@microengenhariaDELL02:~/protasio$
```

Crie um diretório com o seu  
primeiro nome

Execute **ls** para ver  
se o diretório foi  
criado

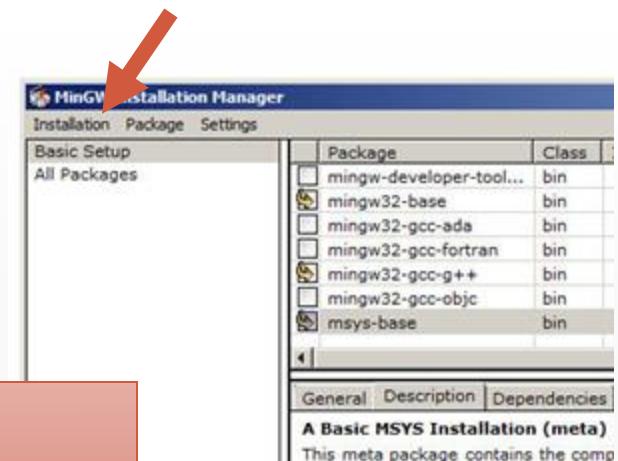
# MinGW: instalação

---

# MinGW C++

## ▪ Instalação

- Clique em <https://sourceforge.net/projects/mingw/files/> e download o arquivo **mingw-get-setup.exe** (86.5 kB)
- Após seguir os passo de instalação, selecione os seguintes pacotes:
  - mingw-developer-tool
  - Mingw32-base
  - Mingw32-gcc-g++
  - Msys-base
- Para efetivar a instalação, clique na aba <Installation> e depois em <Apply Changes>



**Um passo a passo, pode ser visto aqui**

<https://www.ics.uci.edu/~pattis/common/handouts/mingweclipse/mingw.html>

# MinGW

---

- Instalação
  - Após modificar o *path* do Windows
    - Painel de controle
      - Sistemas
        - Aba avançado
          - Variáveis de ambiente
            - Edite o *Path*
              - Adicione no final do path atual: “;c:\MinGW\bin”
      - OU, caso não tenha direitos de administrador,
        - Variáveis de usuário

# Notepad++: instalação

---

# Notepad++

---

- Editor de Código
  - Notepad++ é um editor free para ambientes Windows e licença GPL
  - Instalação
    - <http://notepad-plus-plus.org>
  - Após modificar o *path* do Windows
    - Painel de controle
      - Sistemas
        - Aba avançado
          - Variáveis de ambiente
            - Edite o *Path*
              - Adicione no final do path atual: “; C:\Program Files (x86)\Notepad++”
          - OU, caso não tenha direitos de administrador,
            - Variáveis de usuário



# Ambiente de programação em Windows da disciplina

The screenshot shows a Windows desktop environment. On the left is a Notepad++ window titled "G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2\hello.c - Notepad++". It contains the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Hello UFPB");
5
6     return 0;
7 }
```

To the right of the Notepad++ window is a Command Prompt window titled "Prompt de Comando". The session starts with the command "cd" followed by the path to the directory containing the "hello.c" file. It then lists the contents of the directory, shows the compilation of the source code into an executable, and finally runs the executable, displaying the output "Hello UFPB".

```
G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2>cd "G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2"
G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2>dir
O volume na unidade G é Google Drive File Stream
O Número de Série do Volume é 1983-1116

Pasta de G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2

21/11/2018 19:18    <DIR>      .
21/11/2018 17:58    <DIR>      ..
21/11/2018 18:27           165 ~$1_TP_Introducao.pptx
21/11/2018 18:27           924.177 1_TP_Introducao.pptx
21/11/2018 19:18           78 hello.c
                           3 arquivo(s)    924.420 bytes
                           2 pasta(s)   13.127.200.768 bytes disponíveis

G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2>gcc hello.c -o hello

G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2>dir
O volume na unidade G é Google Drive File Stream
O Número de Série do Volume é 1983-1116

Pasta de G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2

21/11/2018 19:23    <DIR>      .
21/11/2018 17:58    <DIR>      ..
21/11/2018 18:27           165 ~$1_TP_Introducao.pptx
21/11/2018 18:27           924.177 1_TP_Introducao.pptx
21/11/2018 19:18           78 hello.c
21/11/2018 19:23           27.930 hello.exe
                           4 arquivo(s)    952.350 bytes
                           2 pasta(s)   13.125.976.064 bytes disponíveis

G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2>hello
Hello UFPB
G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2>
```

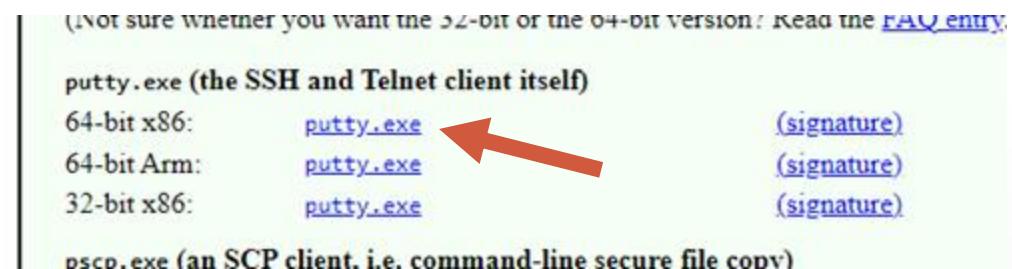
The taskbar at the bottom of the screen shows several icons, including the Start button, a search bar, and icons for File Explorer, Task View, Edge browser, and others. The system tray on the right displays the date and time (19:23, 21/11/2018), battery status, signal strength, and other system information.

# Instalação do PUTTY e do BITVISE

---

# Instalação do Putty

- Vá em  
<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
- Click em **64-bit x86: psftp.exe**

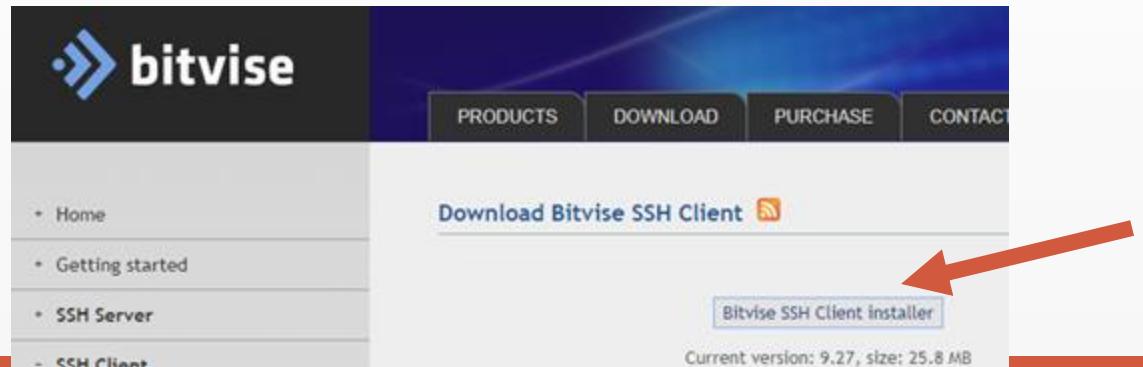


- Preferencialmente, salve-o na Área de Trabalho. Para executar, basta clicar no ícone

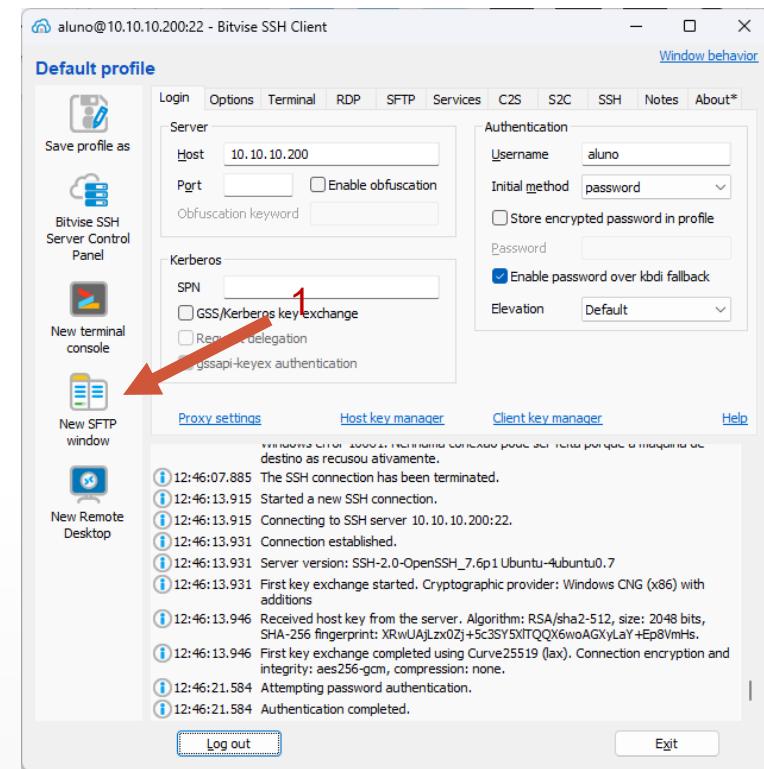
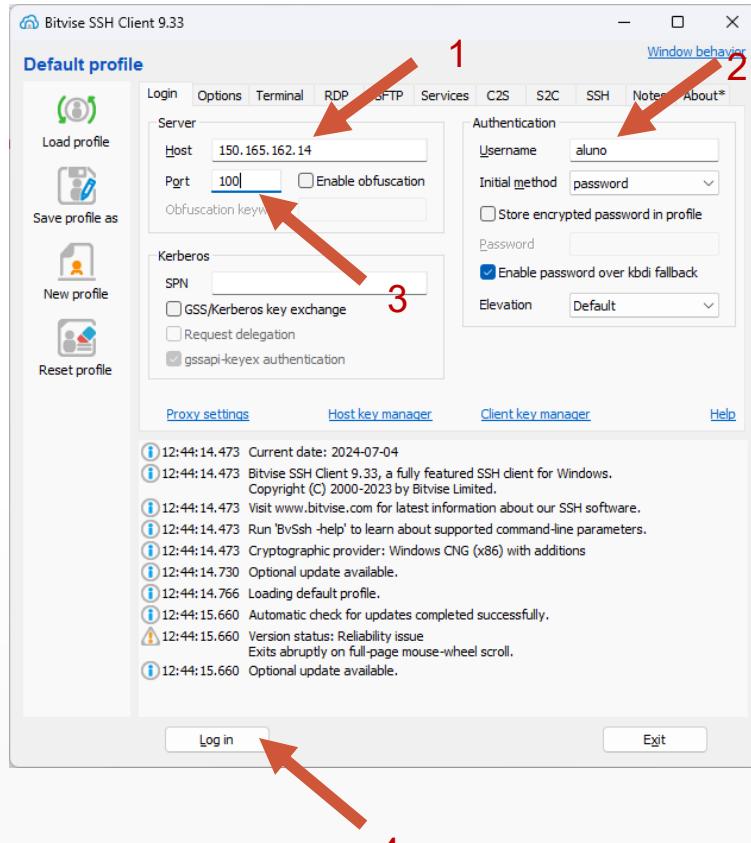


# Instalação do Bitvise

- Vá em <https://www.bitvise.com/>
- Click na aba Download
- Click em [Download Bitvise SSH Client](#)
- Click em [Bitvise SSH Client installer](#)
- Dowload o arquivo e instale



# Uso do Bitvise para baixar arquivos da máquina remota



Usar a aplicação SFTP: Secure File Transfer Protocol

# Uso do Bitvise para baixar arquivos da máquina remota

- Escolha o arquivo na máquina remota e transfira para a pasta desejada na máquina local

