

Técnicas de Programação

Prof. Protásio

Informações gerais da disciplina

- Técnicas de Programação
 - Professor: Protásio
 - E-mail: 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**
 - **1ª e 2ª avaliações:**
 - Avaliação por meio de solicitação de um programa
 - Serão analisados:
 - Compilação dos códigos
 - Execução correta e completa dos códigos.
 - Avaliação Individual
 - **3ª 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* uni 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** (**M**inimalist **G**NU for **W**indows)
 - <https://sourceforge.net/projects/mingw/files/>
 - Procedimentos de instalação no fim dos slides

Ambiente de programação em Windows

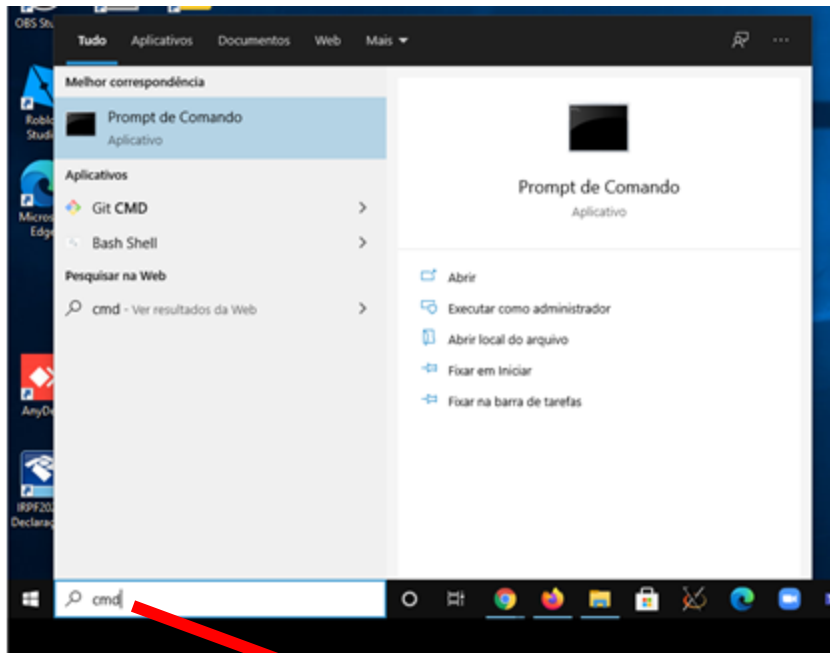
- Abrir “terminal” do windows

- Em executar, digitar:

- **cmd**

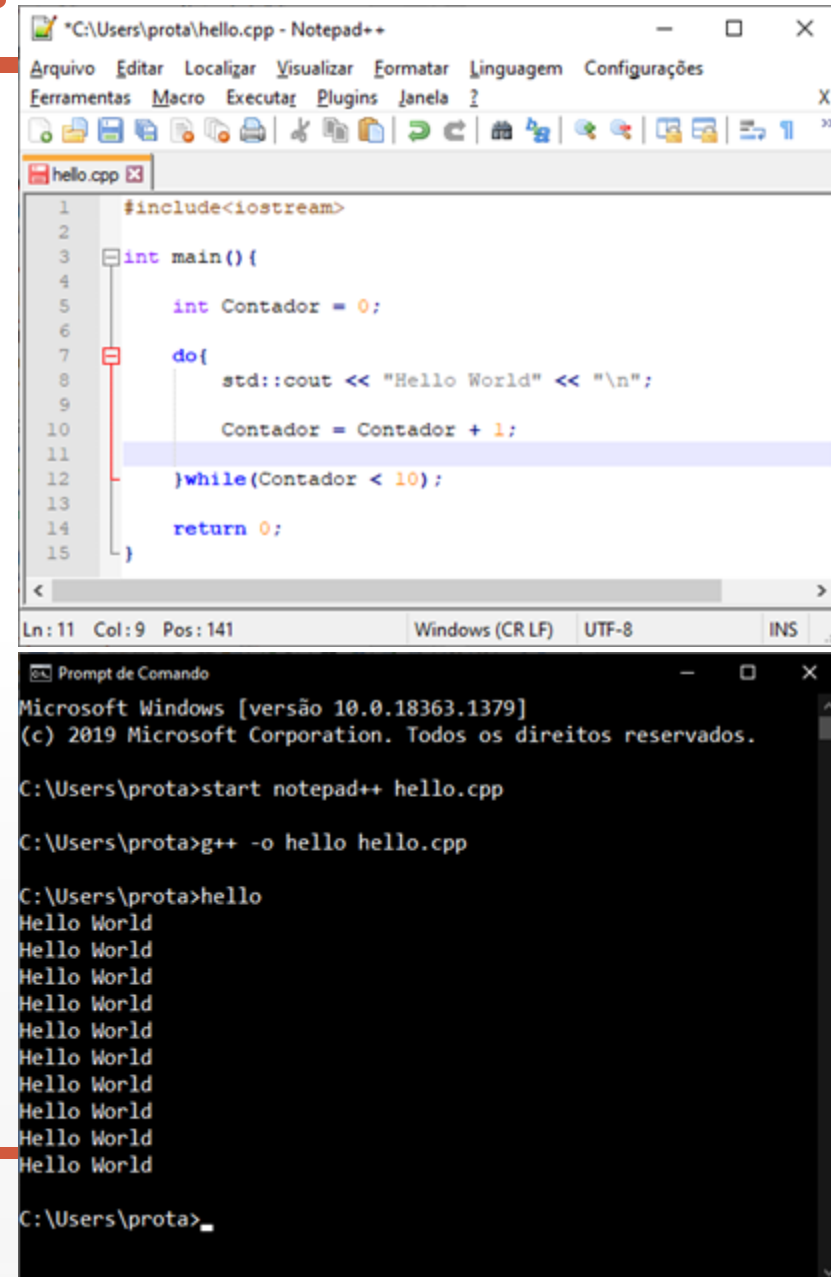
- No terminal, digitar:

- **g++**

A screenshot of a Windows Command Prompt window. The title bar reads 'Prompt de Comando'. The window content shows the following text: 'Microsoft Windows [versão 10.0.18363.1379] (c) 2019 Microsoft Corporation. Todos os direitos reservados. C:\Users\prota>g++ g++: fatal error: no input files compilation terminated. C:\Users\prota>'. The text is displayed in a monospaced font on a black background.

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**



The screenshot shows two windows. The top window is Notepad++ editing a file named 'hello.cpp'. The code is a C++ program that prints 'Hello World' 10 times using a loop. The bottom window is the Windows Command Prompt, showing the commands used to compile and run the program.

```
*C:\Users\prota\hello.cpp - Notepad++
Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações
Ferramentas Macro Executar Plugins Janela ?

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

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

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

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

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

C:\Users\prota>hello
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World

C:\Users\prota>
```



Ambiente de programação em Windows da disciplina

```
G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2\hello.c - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
new 1 x hello.c x
1 #include <stdio.h>
2
3 int main() {
4     printf("Hello UFPB");
5
6     return 0;
7 }
8

length: 78 lines: 8 Ln: 8 Col: 2 Sel: 0|0 Dos\Windows UTF-8 INS

Prompt de Comando
Microsoft Windows [versão 10.0.17134.345]
(c) 2018 Microsoft Corporation. Todos os direitos reservados.

C:\Users\prota>g:

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>
```

Meu primeiro fluxograma



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

Meu primeiro programa

Comando do Linux para Executar programas gráficos

Um editor de texto básico

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

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




The screenshot shows a Notepad++ window titled "C:\Users\prota\hello.cpp - Notepad++". The menu bar includes Arquivo, Editar, Localizar, Visualizar, Formatar, Linguagem, Configurações, Ferramentas, Macro, Executar, Plugins, Janela, and ?. The toolbar contains various icons for file operations and editing. The code editor shows the following C++ code:

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

The status bar at the bottom indicates "Ln: 11 Col: 9 Pos: 141", "Windows (CR LF)", "UTF-8", and "INS".

Compilação

- Compilar é traduzir o **código-fonte** em um **arquivo executável**.
- Comando para compilar
`g++ -o hello hello.cpp`
- Comando para executar
`hello`



```
Prompt de Comando
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;

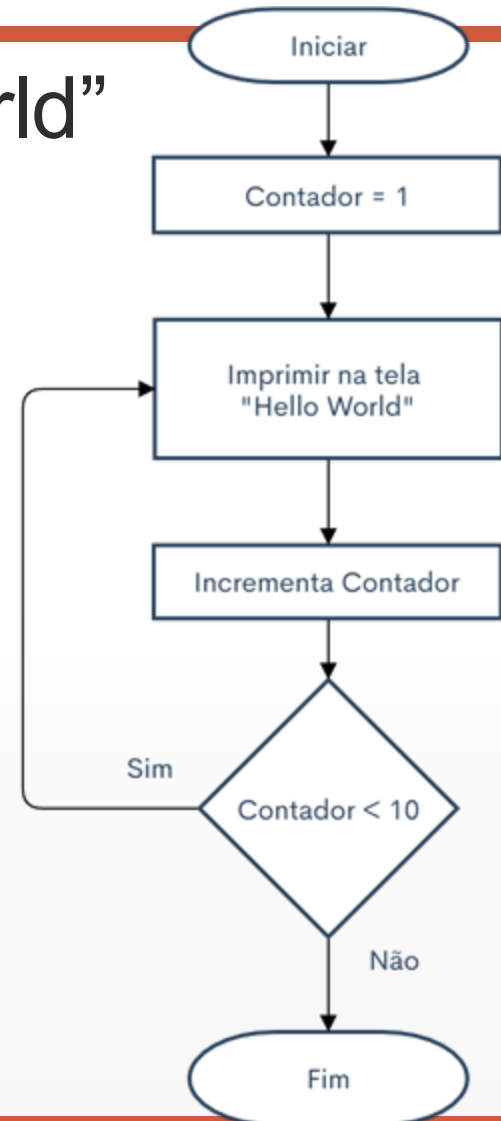
    cout << "Soma" << x + y << "\n";
    cout << "Subtração" << x - y << "\n";
    cout << "Multiplicação" << x * y << "\n";
    cout << "Divisão" << x / y << "\n";

    return 0;
}
```

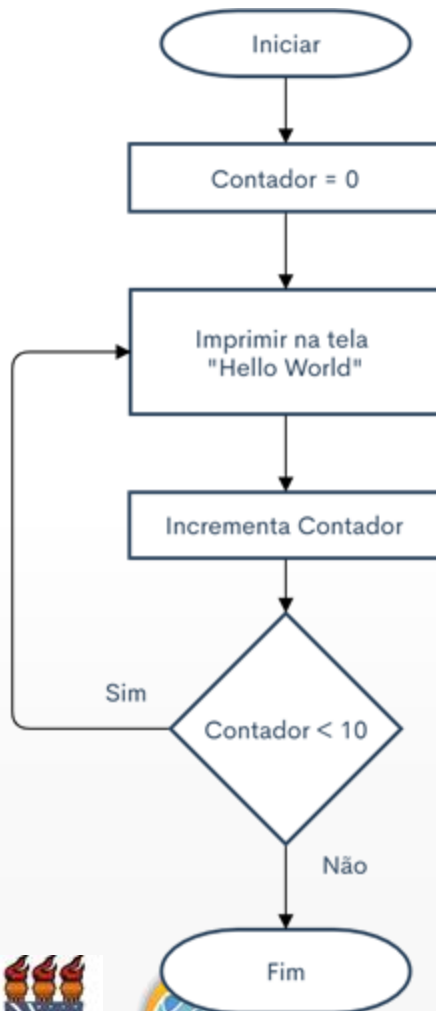
- Editar o código
start **notepad++** **calc3.cpp**
- Compilar
g++ -o **calc3** **calc3.cpp**
- Executar
calc3

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);

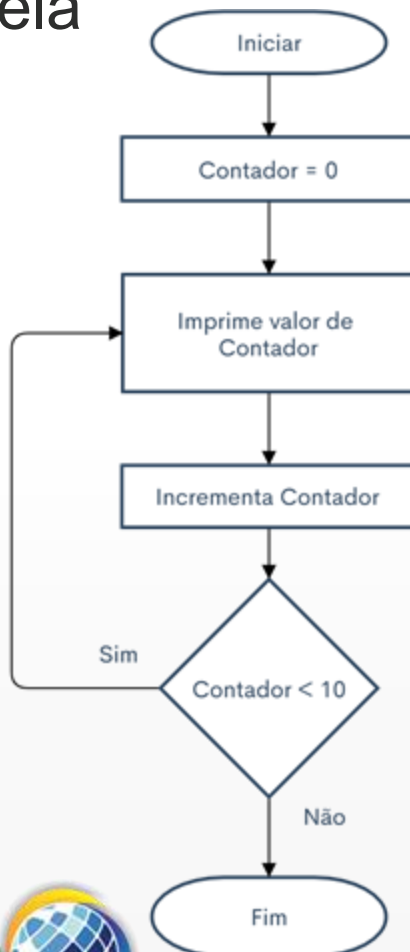
    return 0;
}
```

- 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;

}
```

- Salve como loop2.cpp
- Compile e execute
 - `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;
}
```

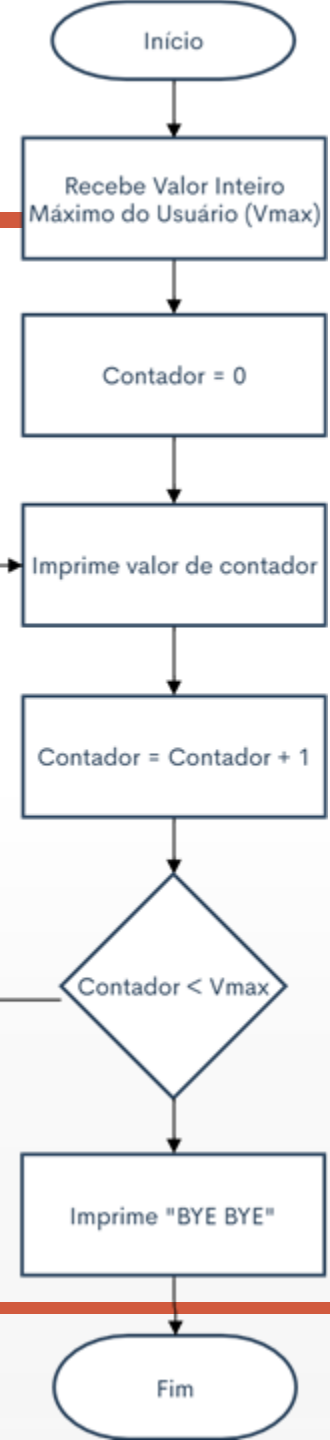
loop4.cpp

Mude para que seja
impresso de
[0 até 2 vezes Vmax]

loop5.cpp

Mude para que seja
impresso de
[-Vmax até Vmax]

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



1ª 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

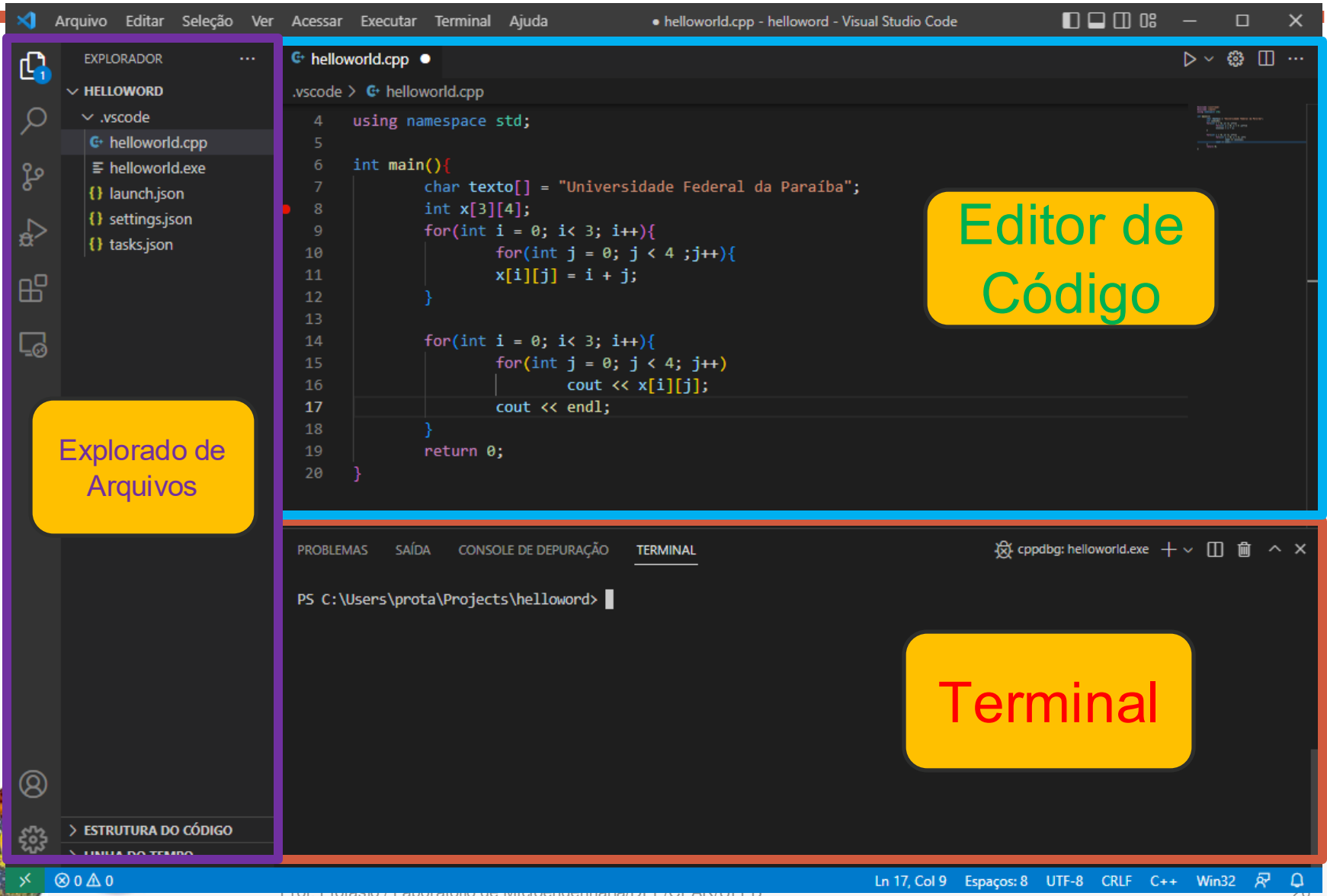
Ambiente de programação em Windows

- 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



Meu primeiro fluxograma



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

Meu primeiro programa

Comando do Linux para Executar programas gráficos

Um editor de texto básico

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

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



The screenshot shows the Visual Studio Code interface with the 'helloworld.cpp' file open. The Explorer sidebar on the left shows the project structure with files like 'helloworld.exe', 'launch.json', 'settings.json', and 'tasks.json'. The main editor window displays the C++ code for the 'Hello World' program. The code includes the necessary header and the main function that prints the phrase 'Hello World' to the console.

```
#include <iostream>

int main(){

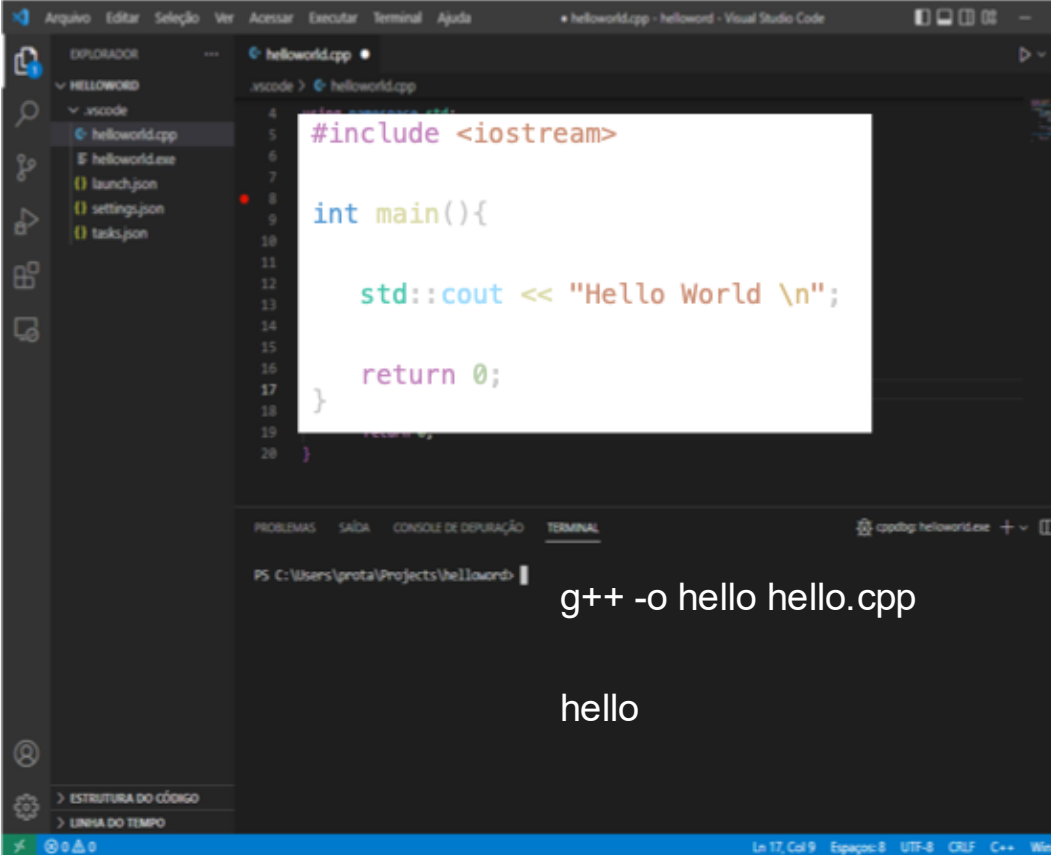
    std::cout << "Hello World \n";

    return 0;
}
```

The terminal at the bottom shows the command prompt: `PS C:\Users\prota\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. The Explorer panel on the left shows a project named 'HELLOWORD' with files 'helloworld.cpp', 'helloworld.exe', 'launch.json', 'settings.json', and 'tasks.json'. The main editor displays the contents of 'helloworld.cpp':

```
#include <iostream>

int main(){

    std::cout << "Hello World \n";

    return 0;
}
```

The TERMINAL panel at the bottom shows the command prompt with the following commands and output:

```
PS C:\Users\prota\Projects\helloworld> g++ -o hello hello.cpp

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

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

PS C:\Users\prota\Projects\helloworld> |

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;

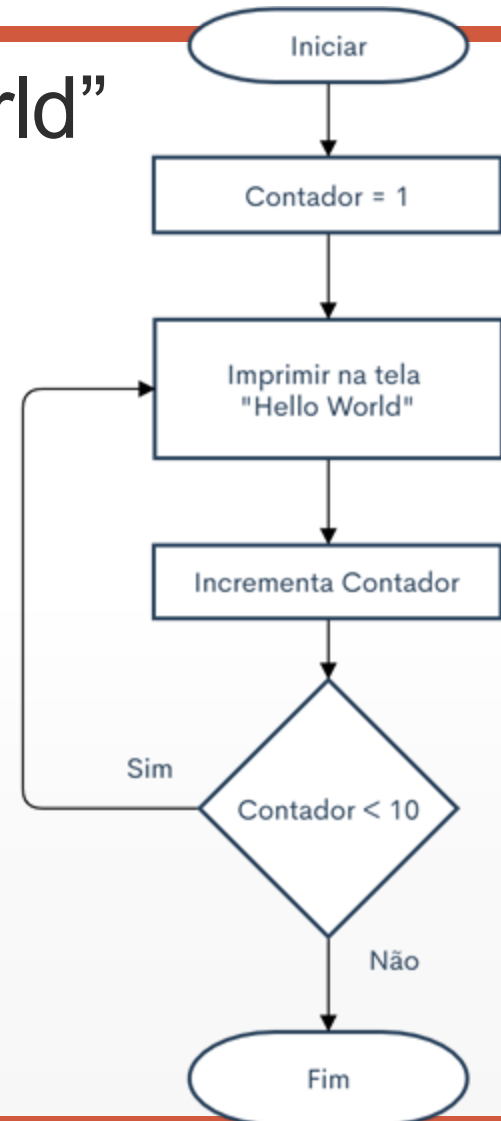
    cout << "Soma" << x + y << "\n";
    cout << "Subtração" << x - y << "\n";
    cout << "Multiplicação" << x * y << "\n";
    cout << "Divisão" << x / y << "\n";

    return 0;
}
```

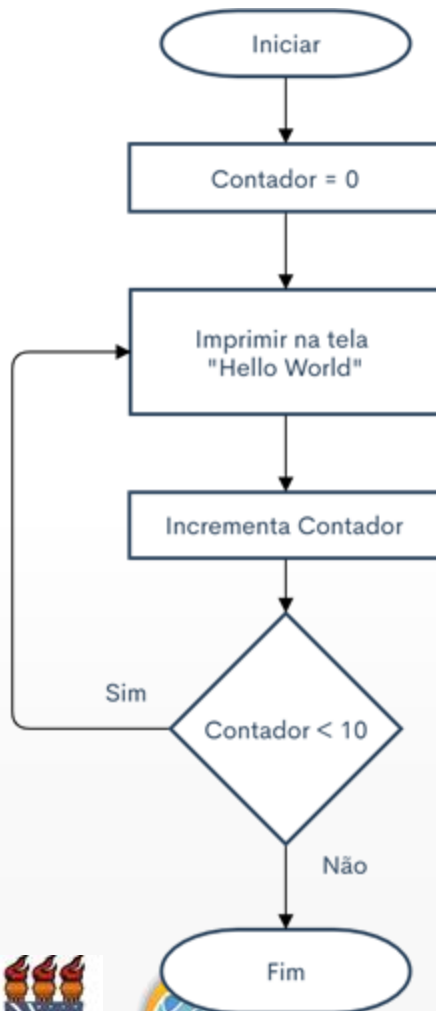
- Editar o código
code **calc3.cpp**
- Compilar
g++ -o **calc3** **calc3.cpp**
- Executar
calc3

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);

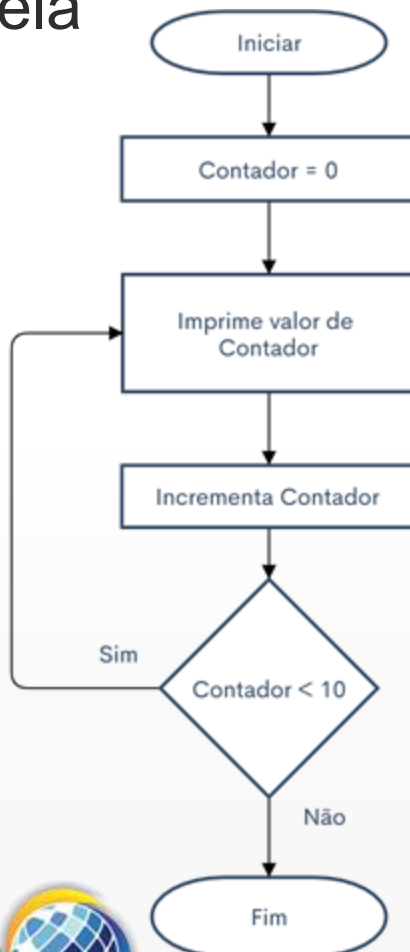
    return 0;
}
```

- 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;

}
```

- Salve como loop2.cpp
- Compile e execute
 - `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;
}
```

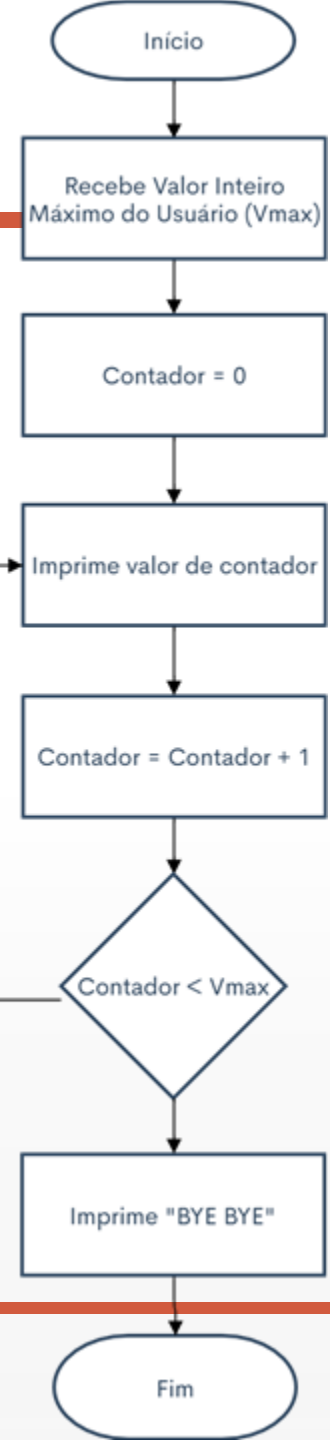
loop4.cpp

Mude para que seja
impresso de
[0 até 2 vezes Vmax]

loop5.cpp

Mude para que seja
impresso de
[-Vmax até Vmax]

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



1ª 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:
 - Seu nome completo e os números entre 0 e sua idade.
 - Os valores entre 1 e 100 e em seguida de 100 a 1.
 - Os valores entre 1 e N com passo P, em que N e P são dados pelo usuário.
 - Exemplo: se N = 7 e P = 2, o resultado deve ser
1 3 5 7
 - 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 executando 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

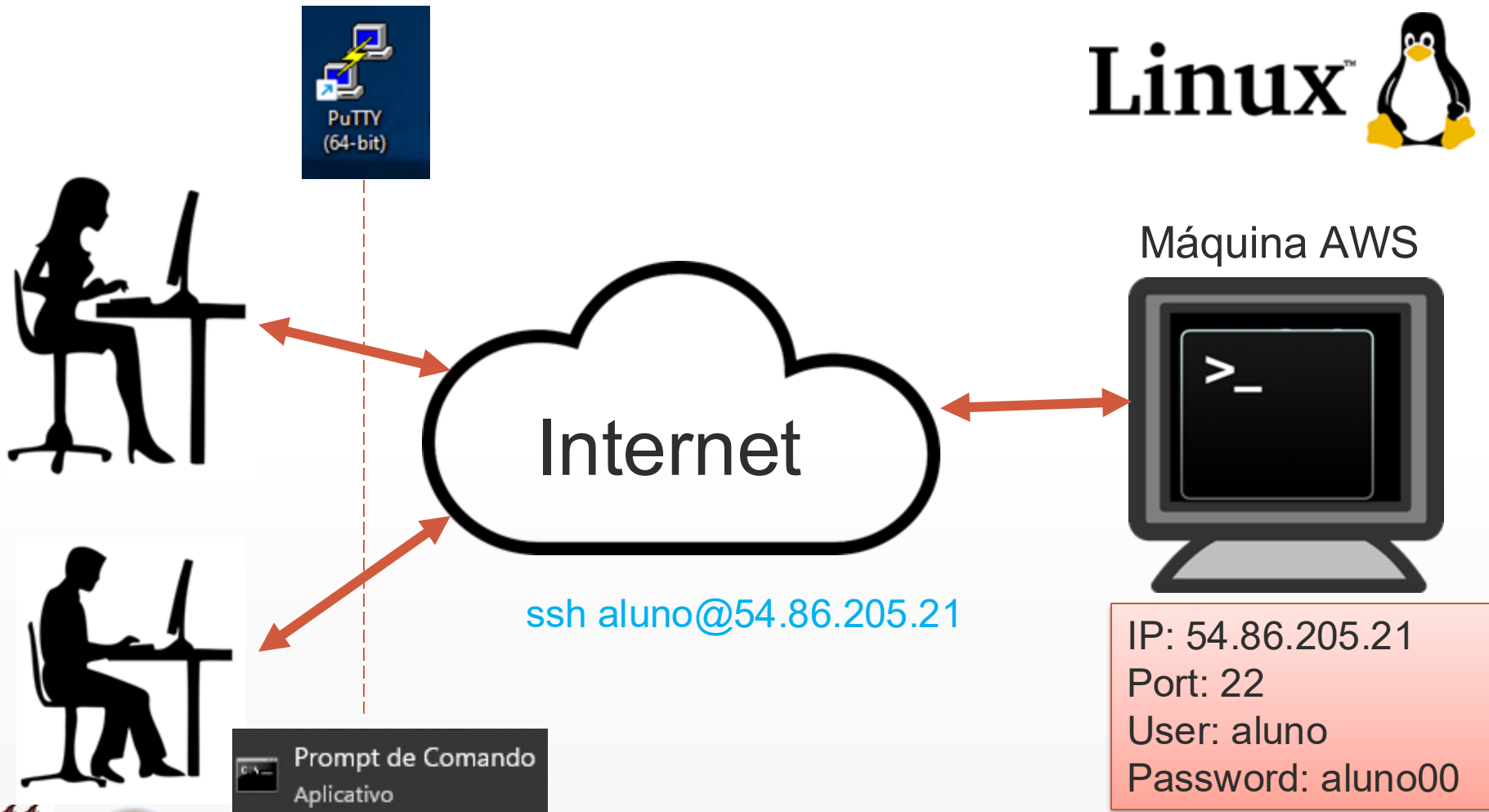
Windows

- **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

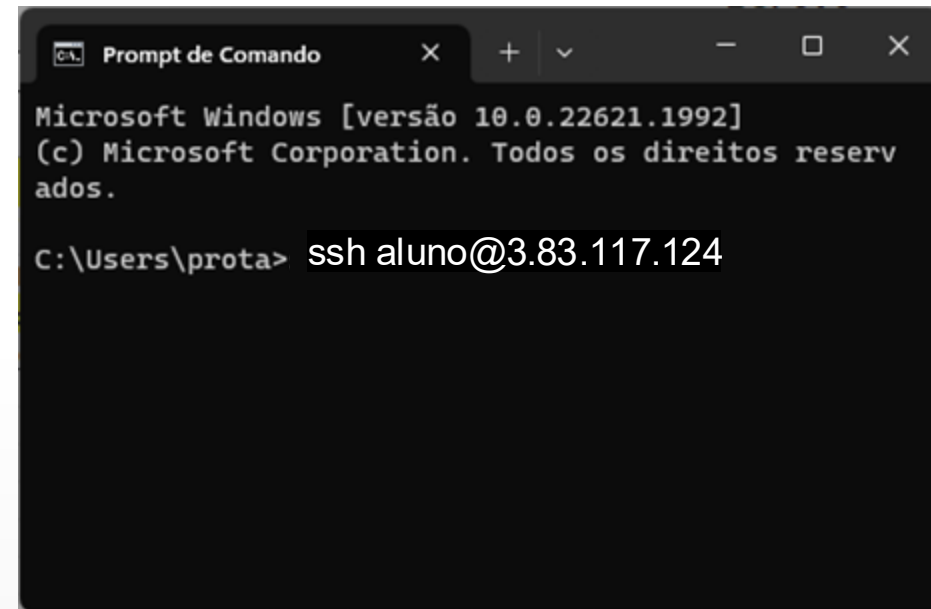
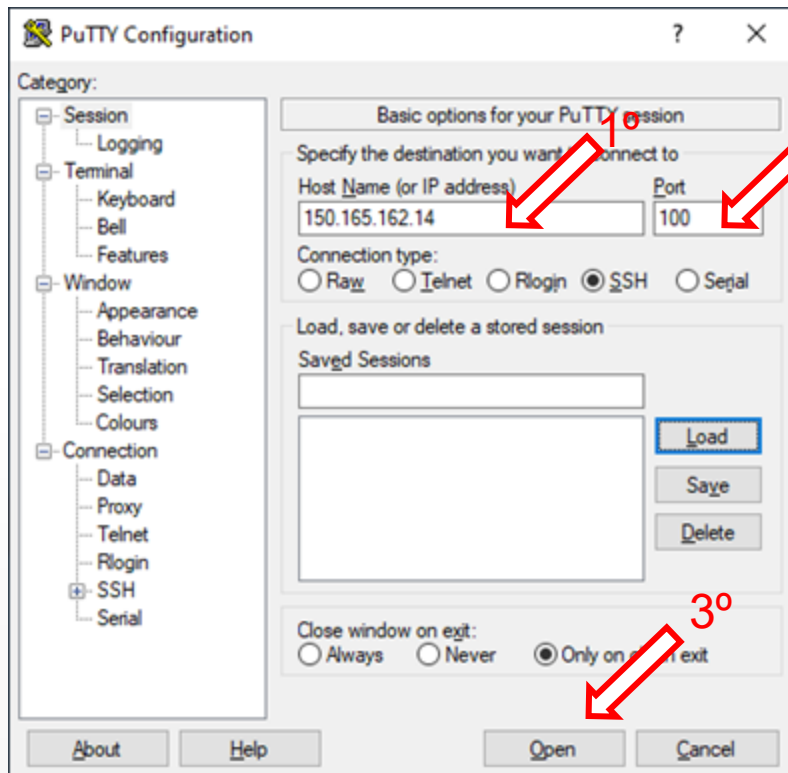
IP: 3.83.117.124

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

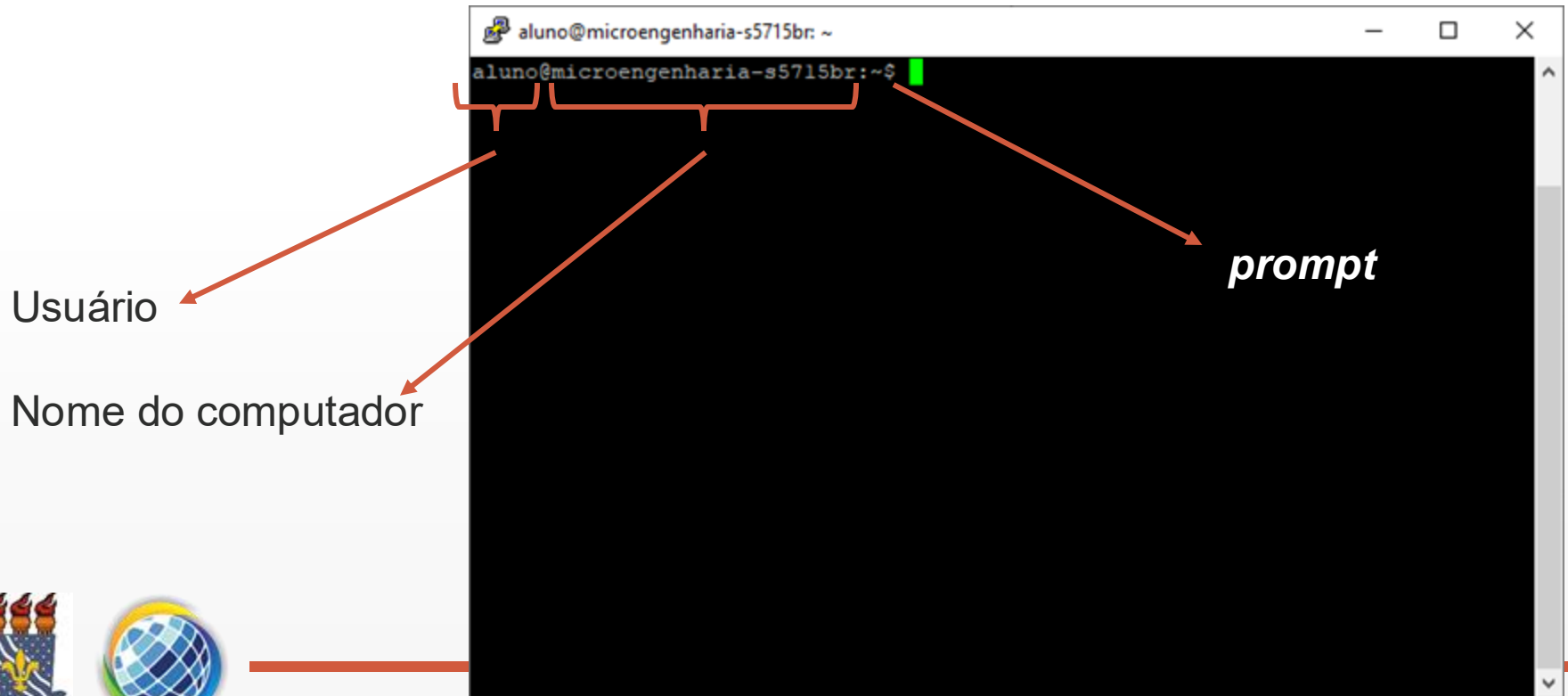
User: aluno

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          teste4
gabriel_veriato    luan               teste4.cpp
gil                luan_lacerda       Thalita
Gisela             luan_novo           Thamyson
GlauberR           luan_pedro          thamyson_aula1
Gustavo            lucas32             thaysa
gustavohra         Lucas_Dantas        thiago
gustavohra_diretorio Lucca               tiago_salles
Gustavo_Souza      luis_netto          victemoteo
Helio_B            manoelz             Victor_Costa
hello              marcolino_diretorio Videos
hello.cpp          Marcos              vinicius
Hyago              MariaThais          ViniciusPontes
Iaácob             mario               vscode-cpptools
igor               Mario               Walklis
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

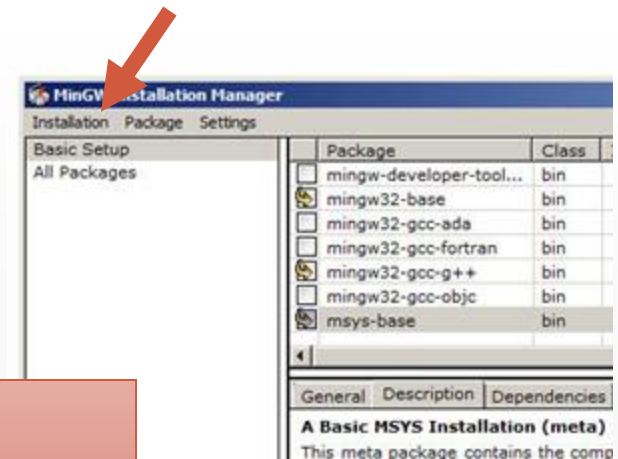


MinGW: instalação

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
 - Ming32-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>

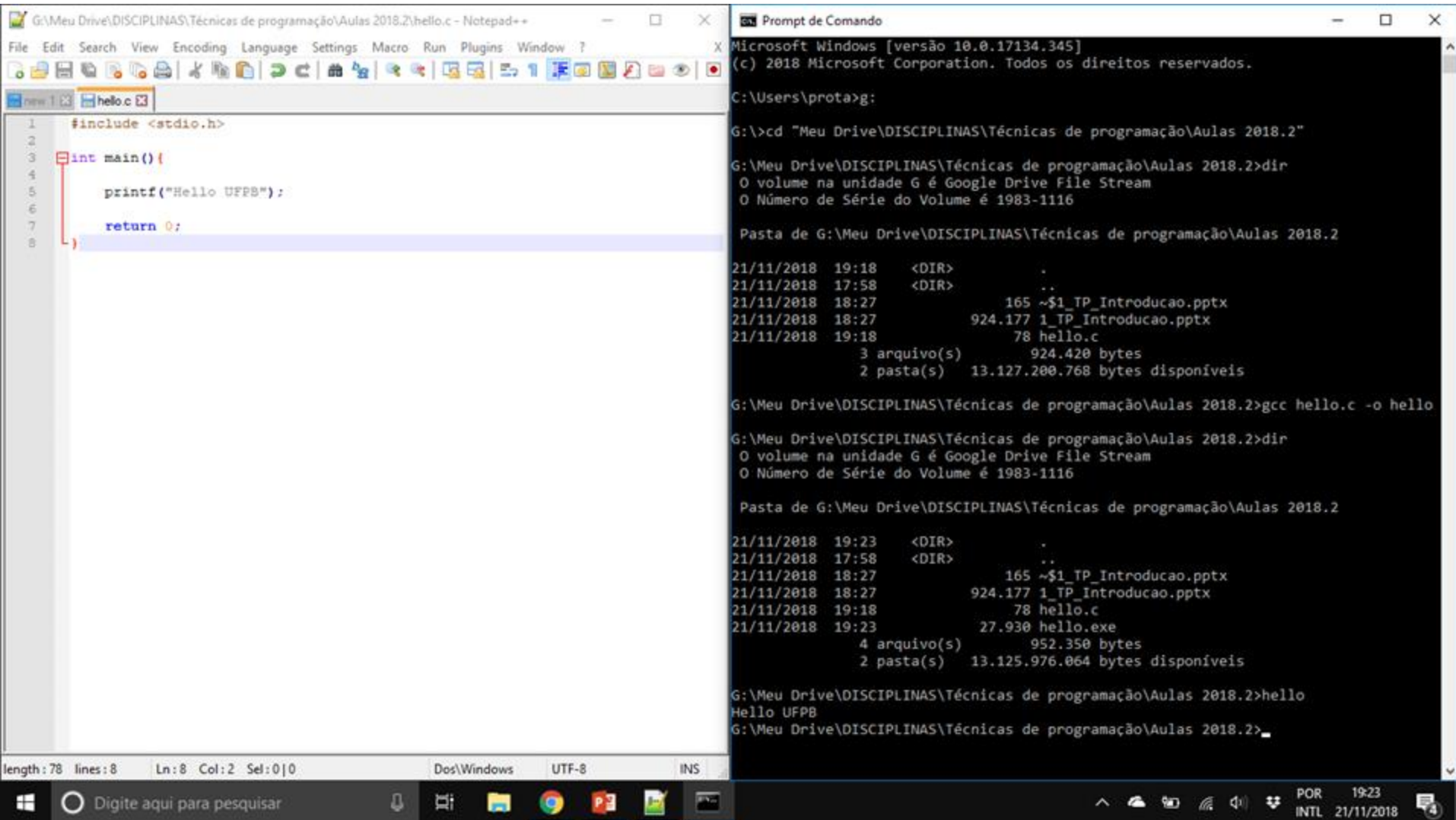
- Instalação
 - Após modificar o *path* do Windows
 - Painel de controle
 - Sistemas
 - Aba avançado
 - Variáveis de ambiente
 - Edite o *Path*
 - Adicione no final do path atual: “;c:\MingW\bin”
 - OU, caso não tenha direitos de administrador,
 - Variáveis de usuário

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



```
G:\Meu Drive\DISCIPLINAS\Técnicas de programação\Aulas 2018.2\hello.c - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
new 1 x hello.c x
1 #include <stdio.h>
2
3 int main() {
4     printf("Hello UFPB");
5
6     return 0;
7 }
8

length: 78 lines: 8 Ln: 8 Col: 2 Sel: 0|0 Dos\Windows UTF-8 INS

Prompt de Comando
Microsoft Windows [versão 10.0.17134.345]
(c) 2018 Microsoft Corporation. Todos os direitos reservados.

C:\Users\prota>g:

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>
```

Instalação do PUTTY e do BITWISE

Instalação do Putty


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

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

putty.exe (the SSH and Telnet client itself)

64-bit x86:	putty.exe	(signature)
64-bit Arm:	putty.exe	(signature)
32-bit x86:	putty.exe	(signature)

pscp.exe (an SCP client, i.e. command-line secure file copy)



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

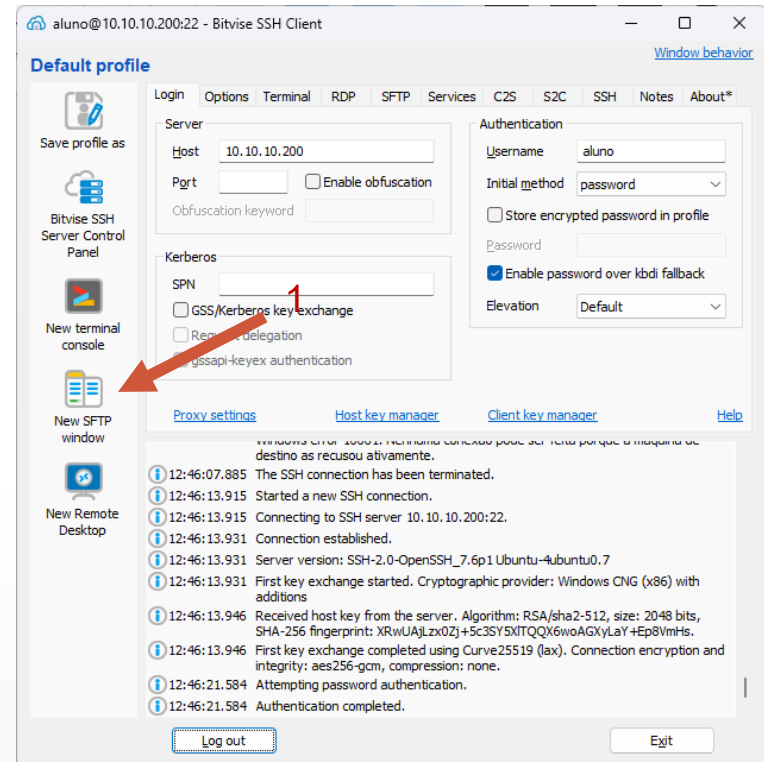
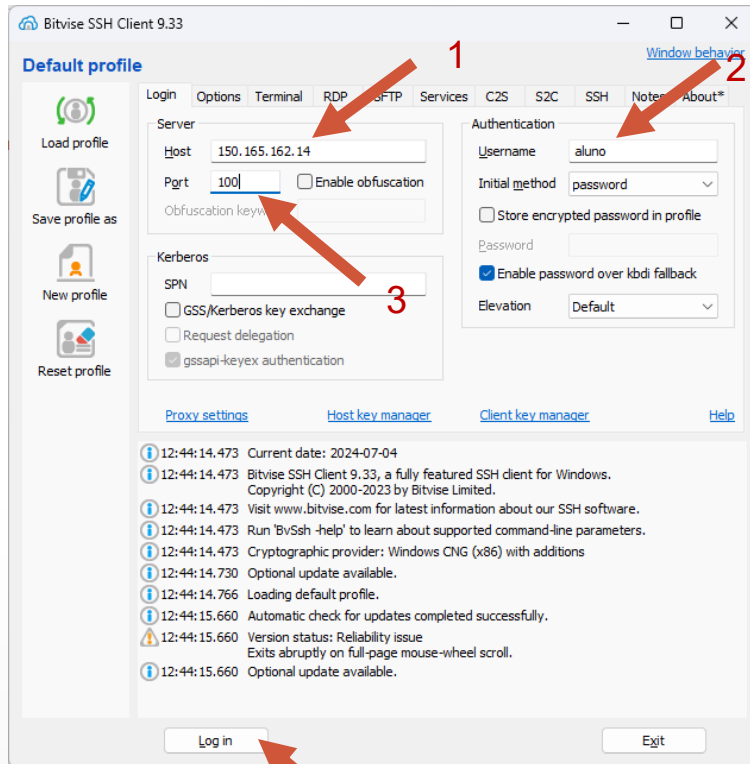


Instalação do Bitvise

- 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 Tranfer 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

