



LAB. 1:

Familiarização com ferramentas para programação em C

Algoritmos e Programação II – Turmas 02D

2º semestre de 2024

Prof. Jefferson Zanutto

Prof. Bruno da Silva Rodrigues

Prof. Joaquim Pessoa Filho



Introdução

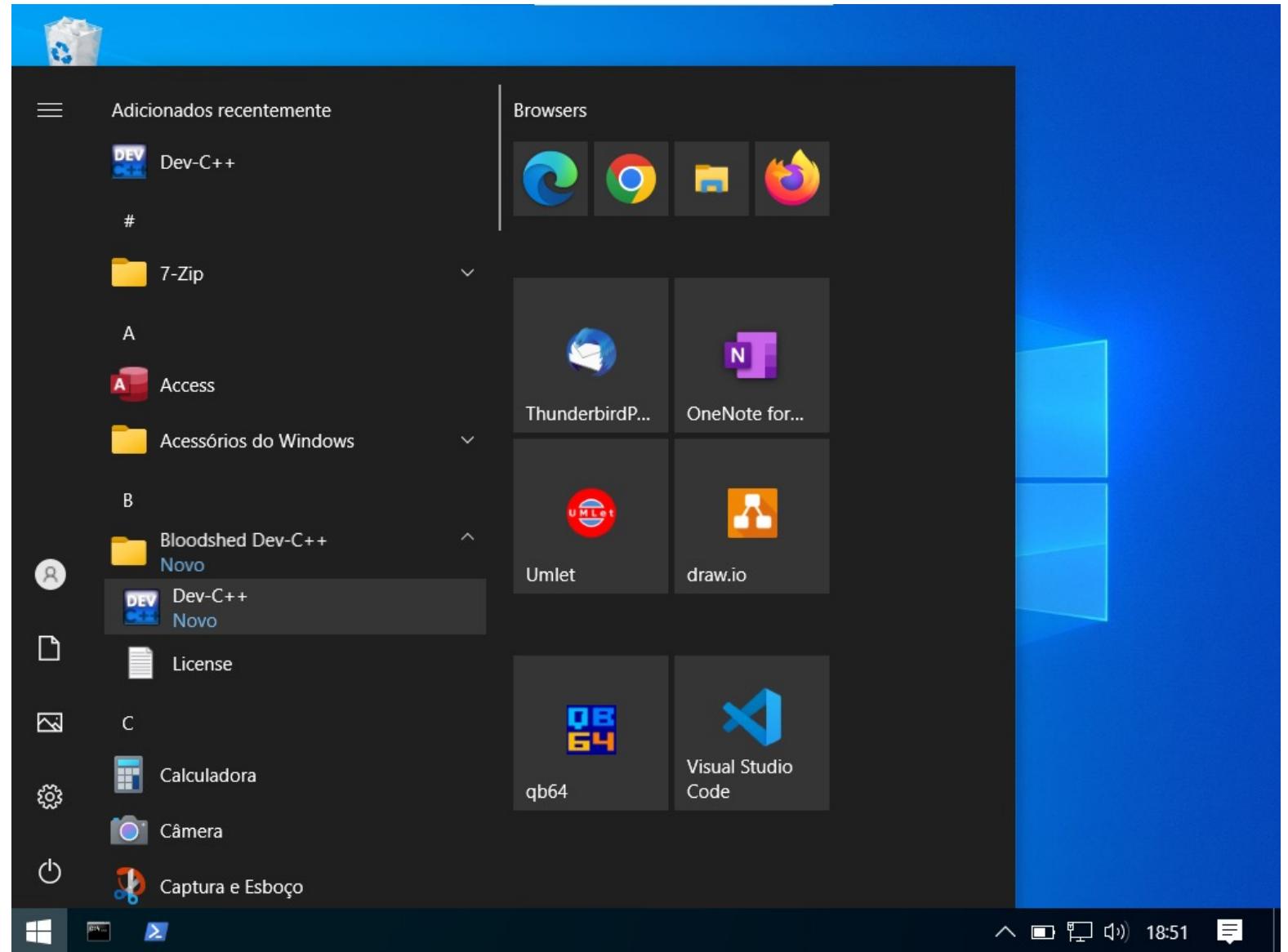
- Há várias ferramentas que podem ser utilizadas para o desenvolvimento de programas na linguagem C.
- Nesta aula vamos escrever um primeiro programa em C utilizando as seguintes ferramentas:
 - Dev-C++
 - repl.it
 - GitHub Codespaces



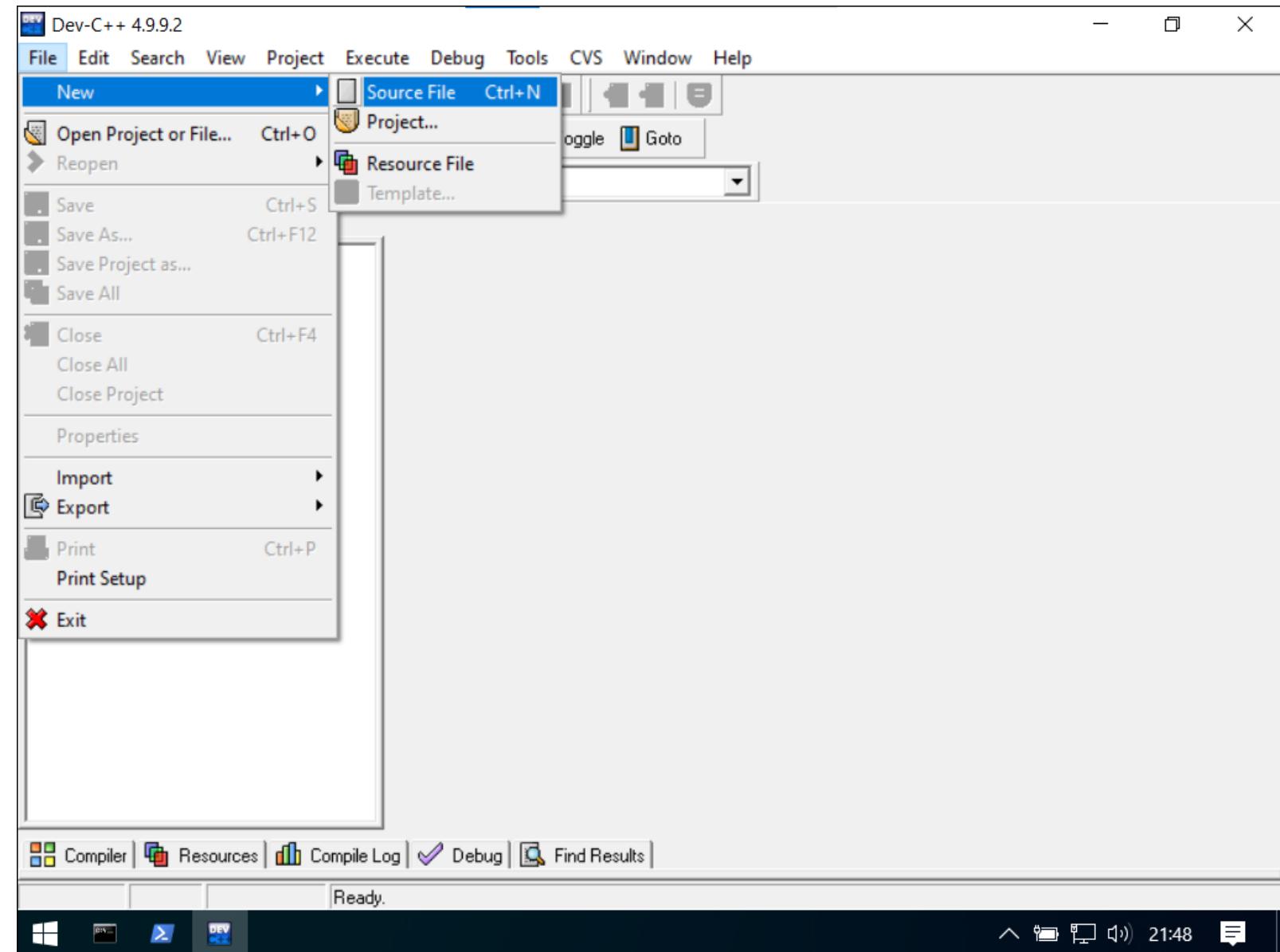
Dev-C++



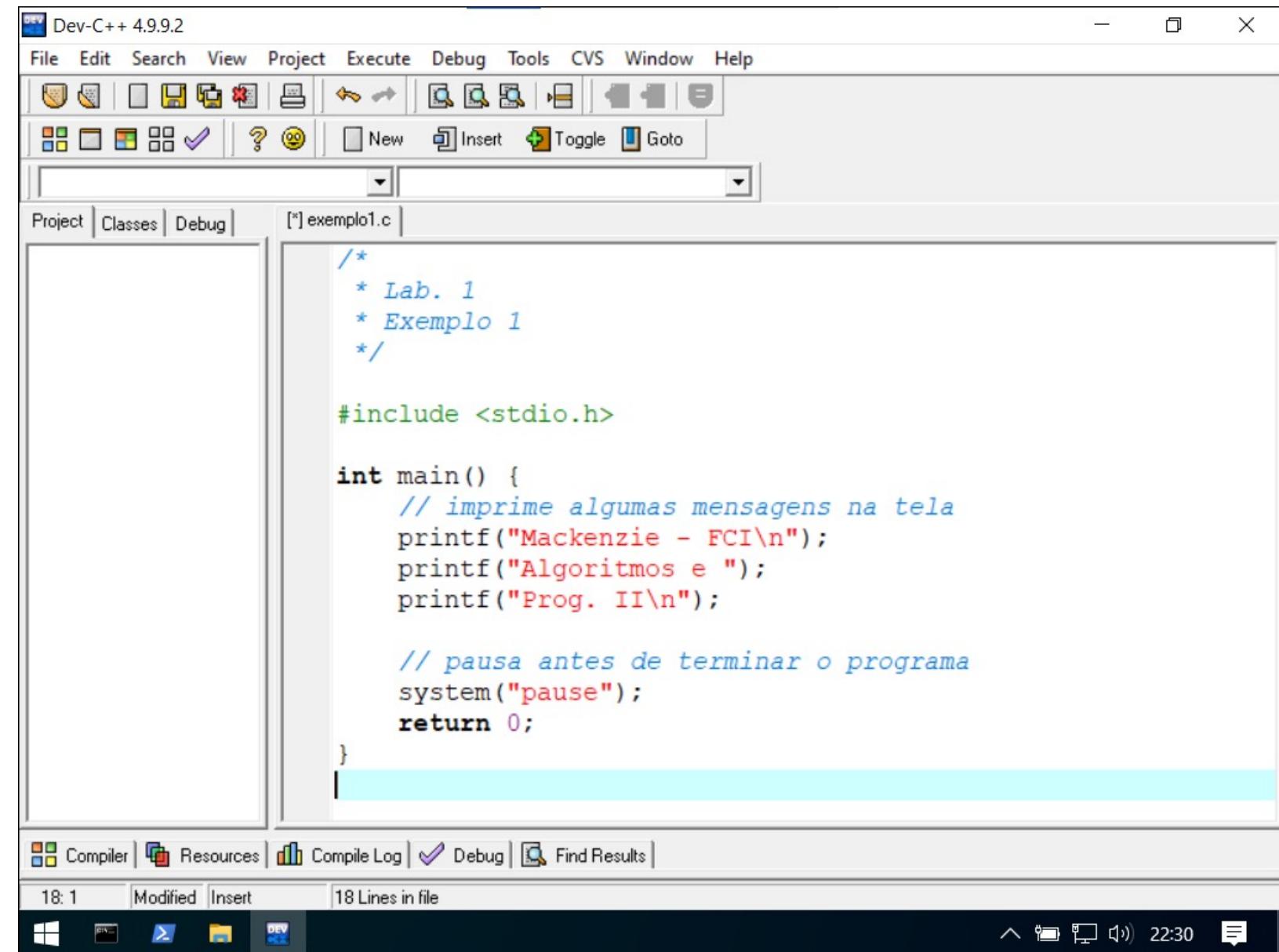
Abra o menu iniciar e execute o programa Dev-C++



Seleccione
File > New > Source File



Digite o código
do programa



The screenshot shows the Dev-C++ 4.9.9.2 IDE interface. The menu bar includes File, Edit, Search, View, Project, Execute, Debug, Tools, CVS, Window, and Help. The toolbar contains various icons for file operations like New, Insert, Toggle, and Goto. The left sidebar has tabs for Project, Classes, and Debug, with 'exemplo1.c' selected. The main code editor window displays the following C code:

```
/*
 * Lab. 1
 * Exemplo 1
 */

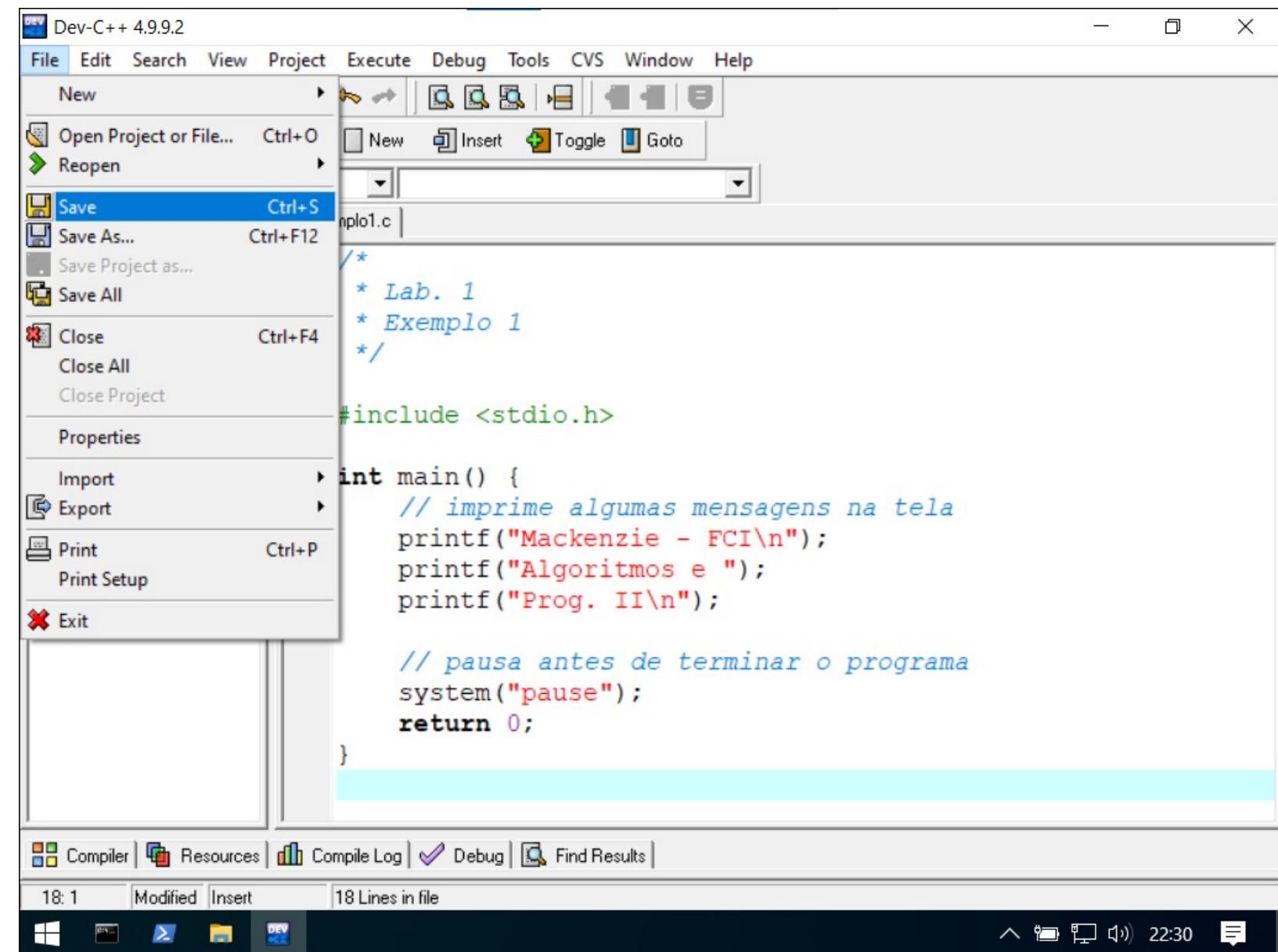
#include <stdio.h>

int main() {
    // imprime algumas mensagens na tela
    printf("Mackenzie - FCI\n");
    printf("Algoritmos e ");
    printf("Prog. II\n");

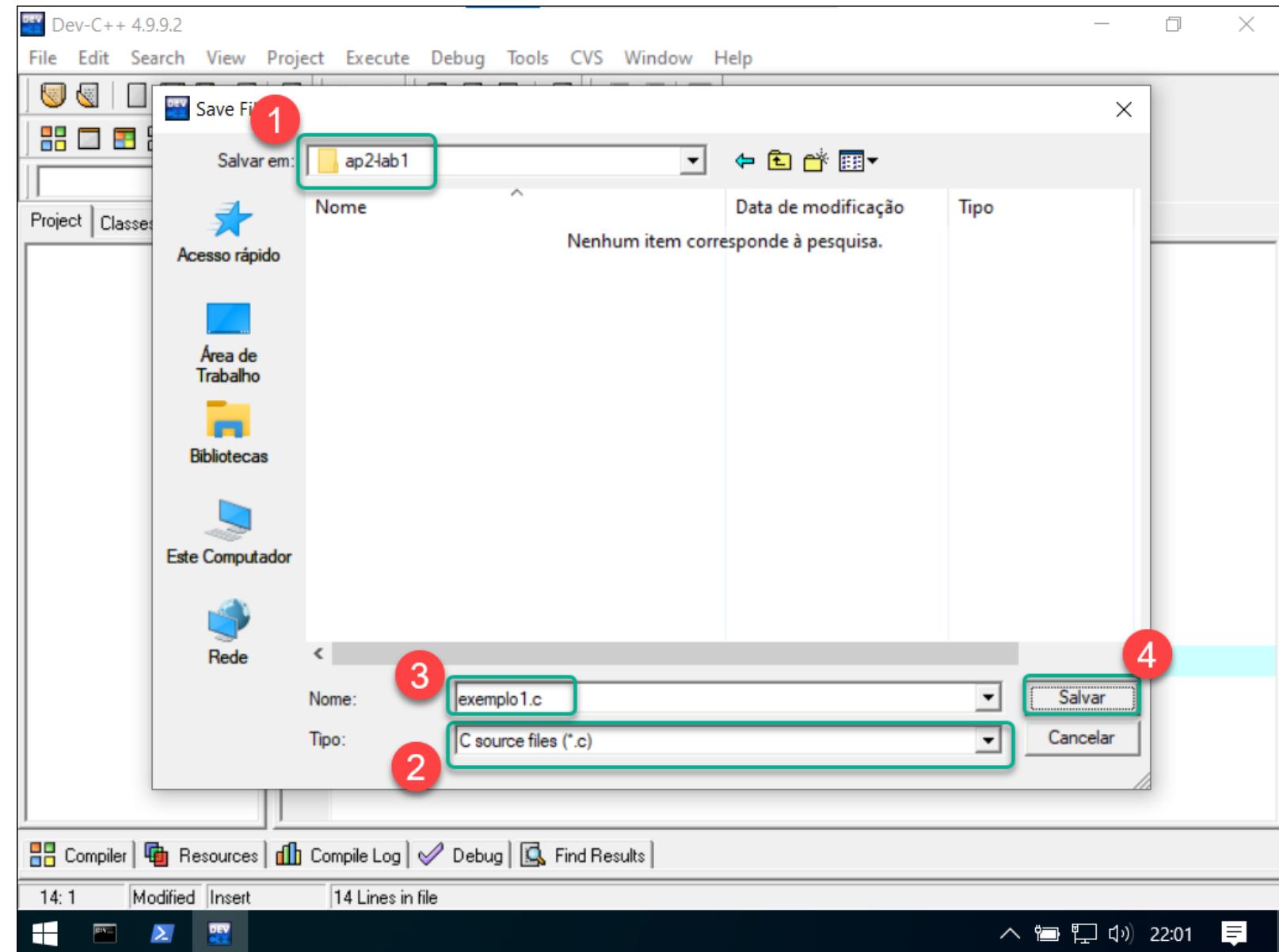
    // pausa antes de terminar o programa
    system("pause");
    return 0;
}
```

The status bar at the bottom shows '18: 1 Modified Insert 18 Lines in file' and the system tray indicates it's 22:30.

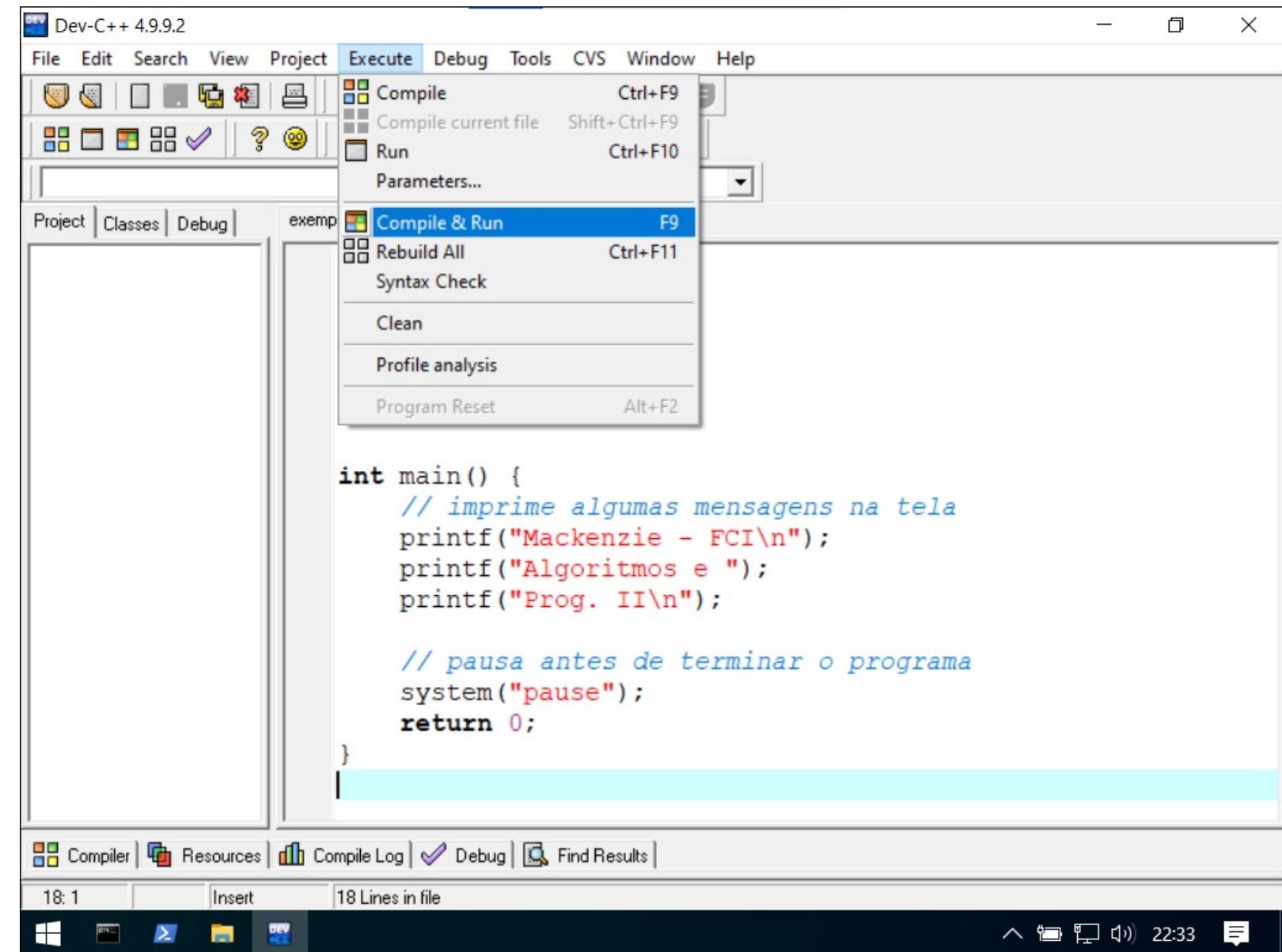
Save o arquivo



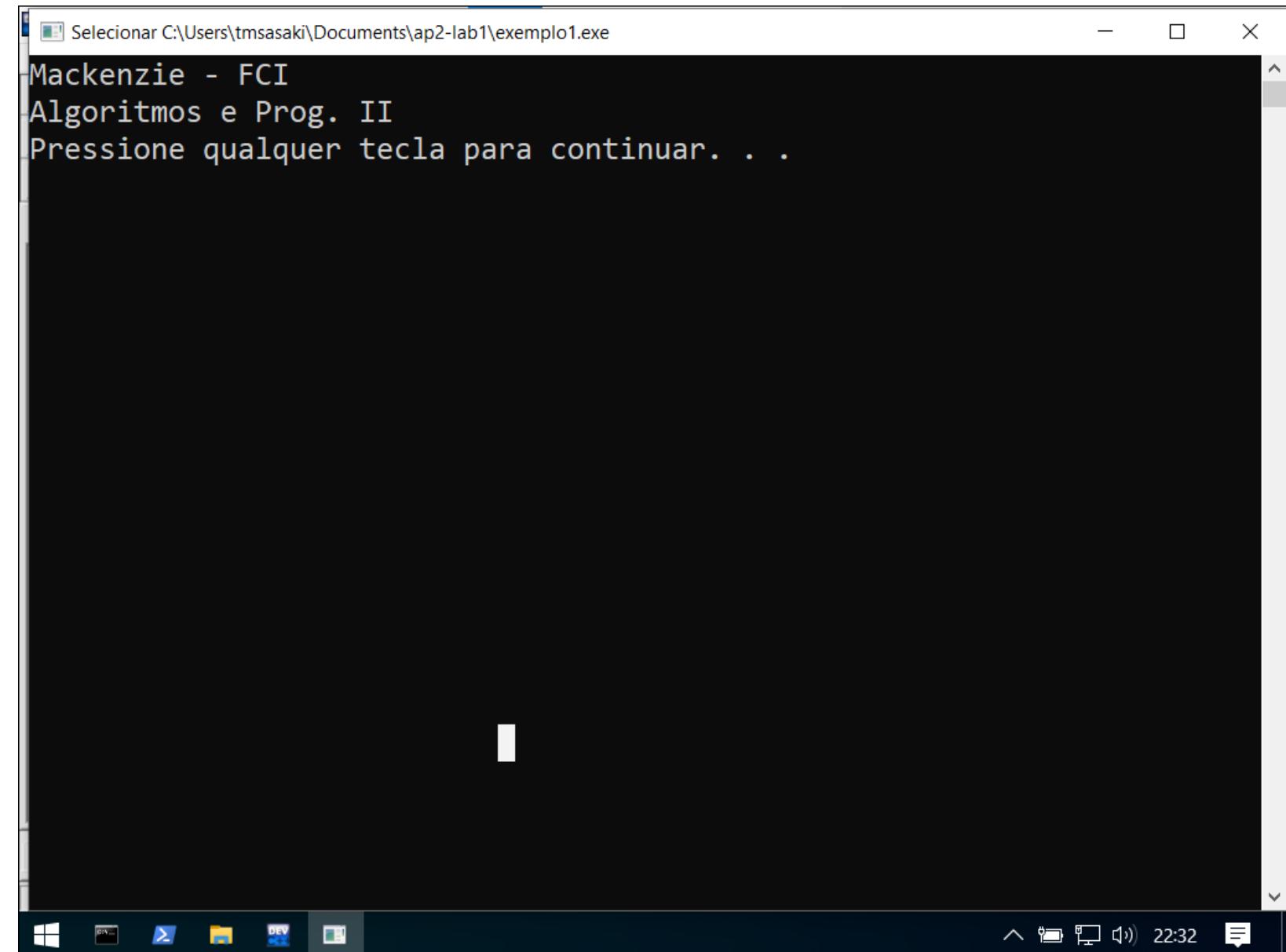
Selecione a pasta e defina o nome



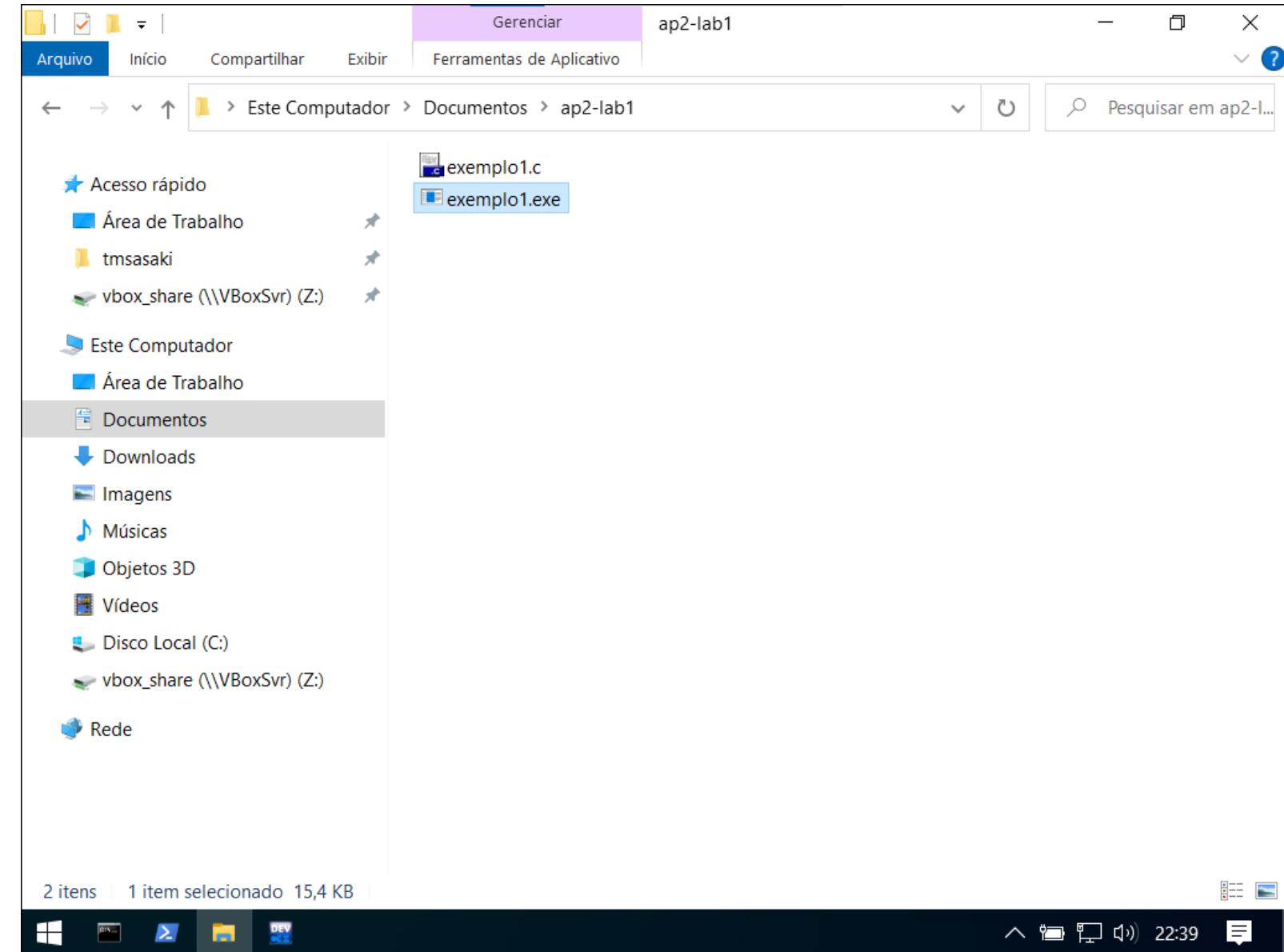
Selecione
Execute > Compile & Run



**Verifique o
resultado da
execução**



Você agora pode usar o executável



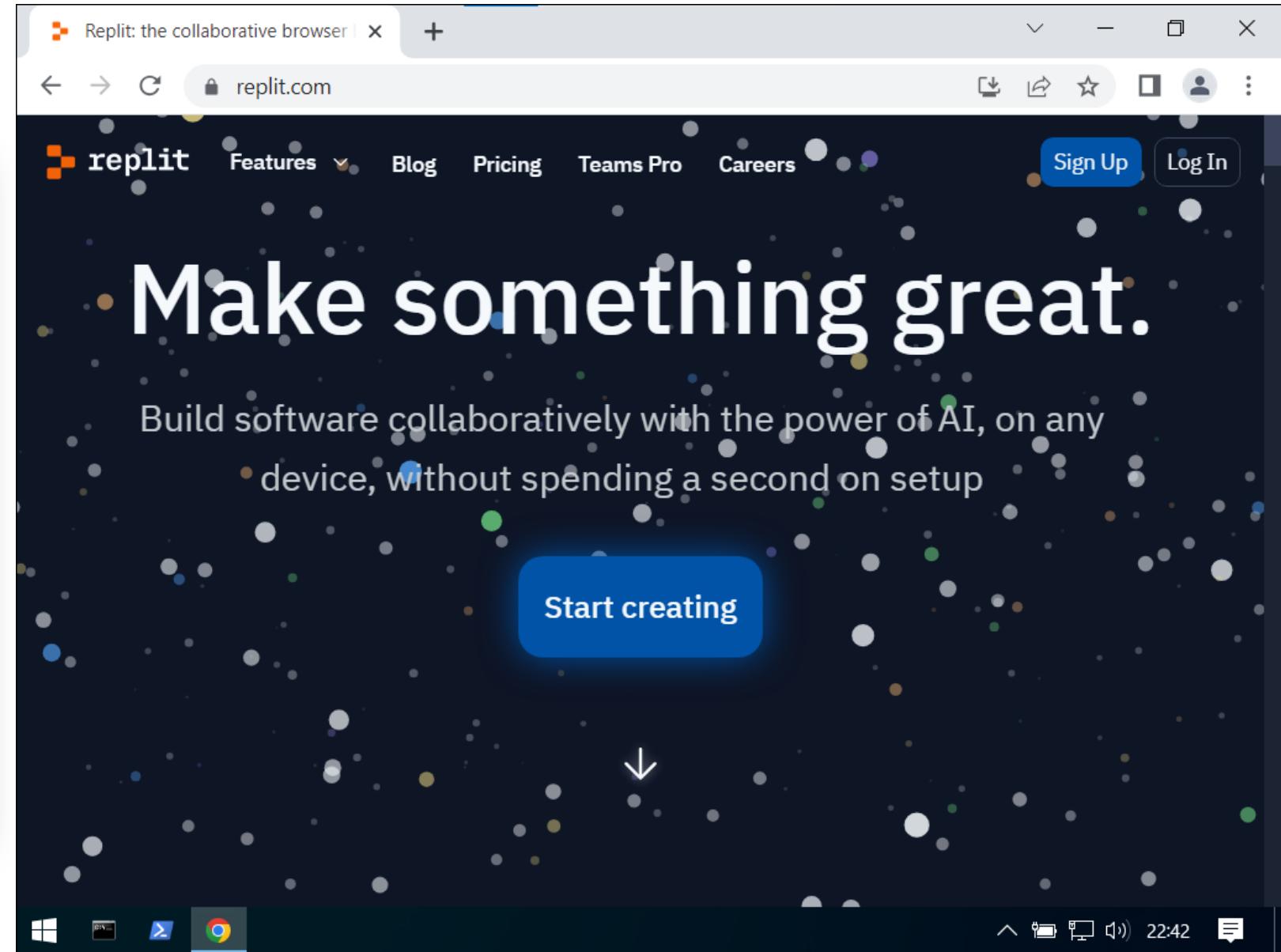
Exercício 1

Altere o código do programa de forma a imprimir também o nome da sua turma.

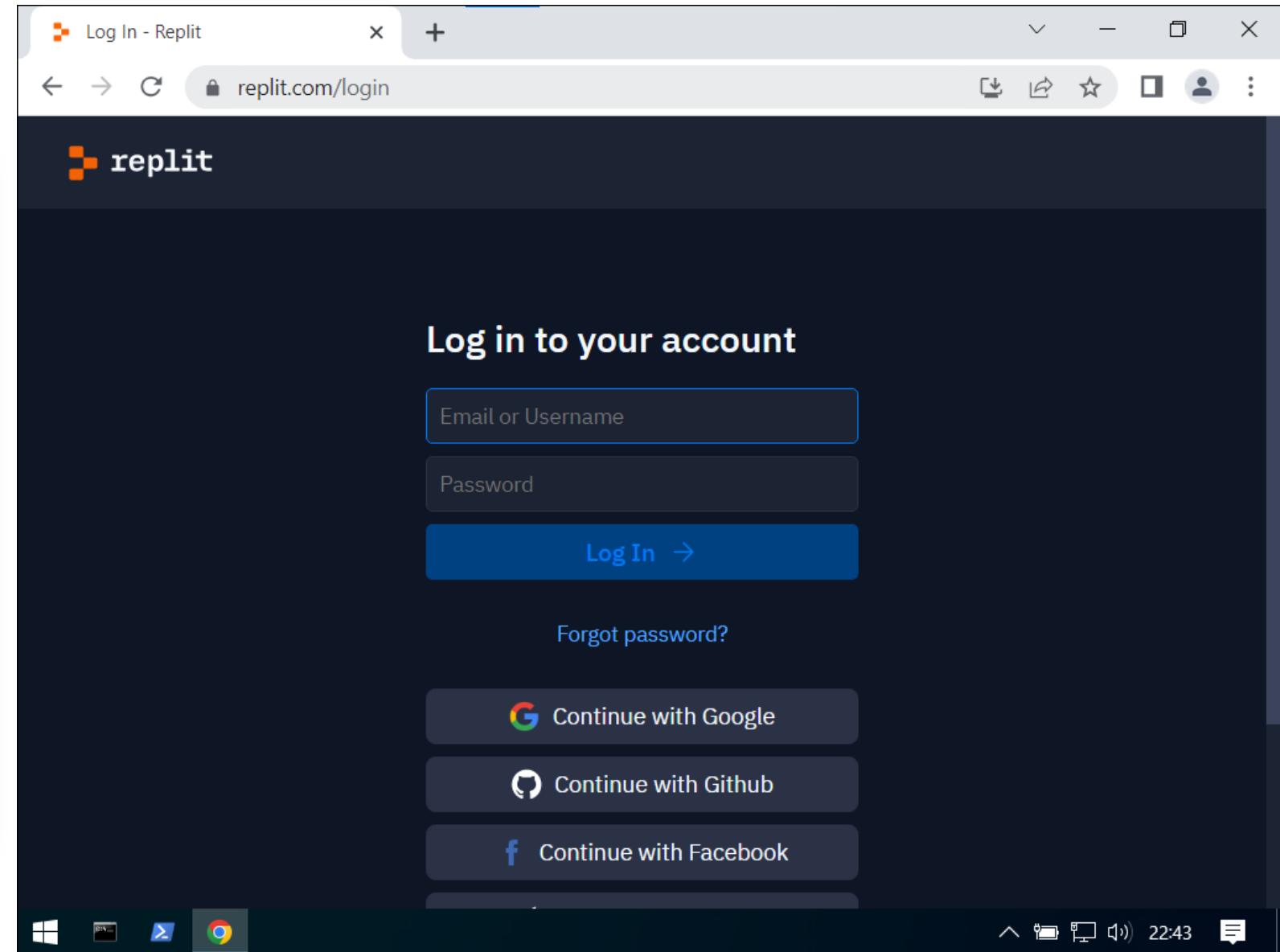
repl.it



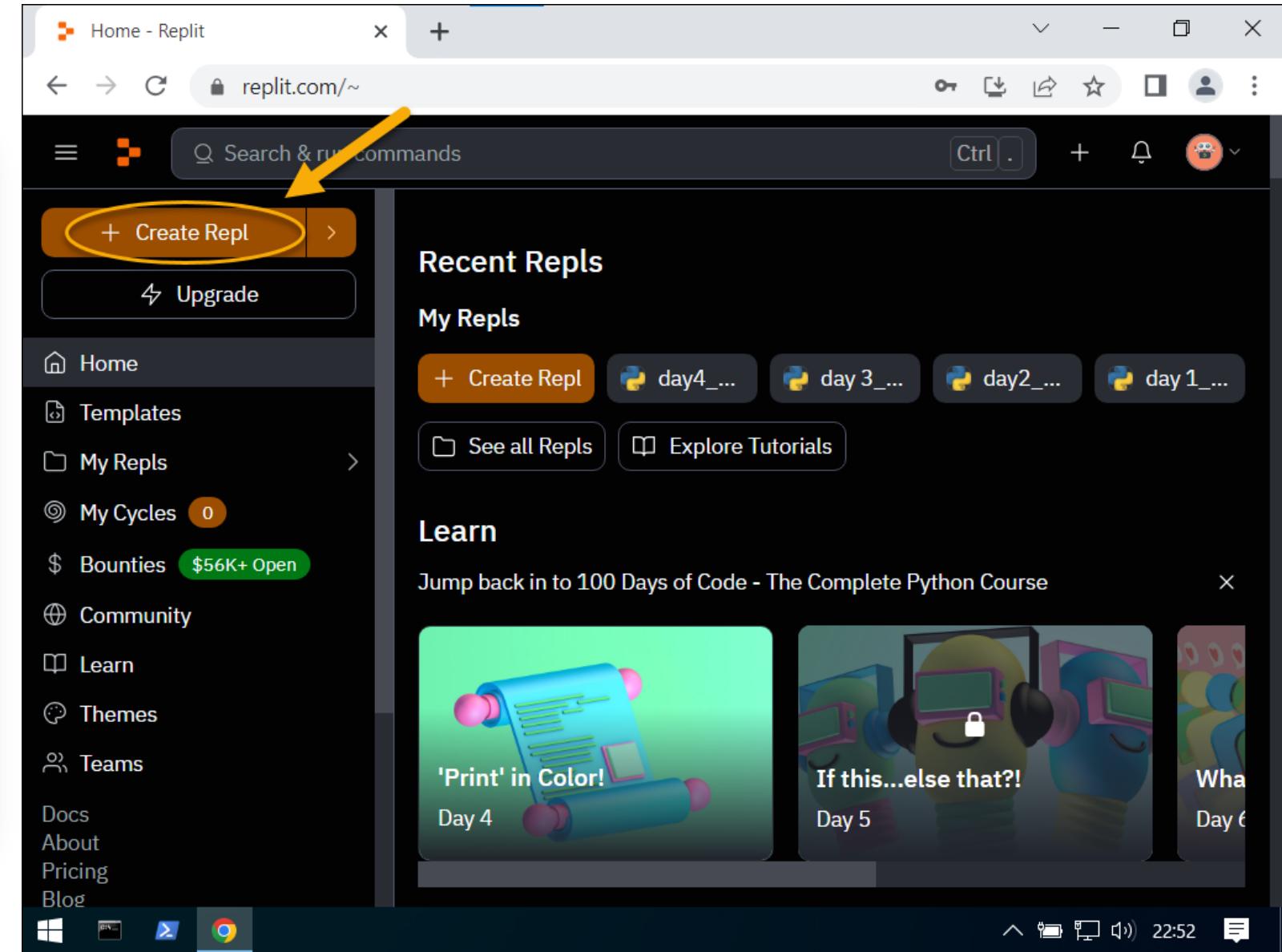
Acesse
<https://replit.com>



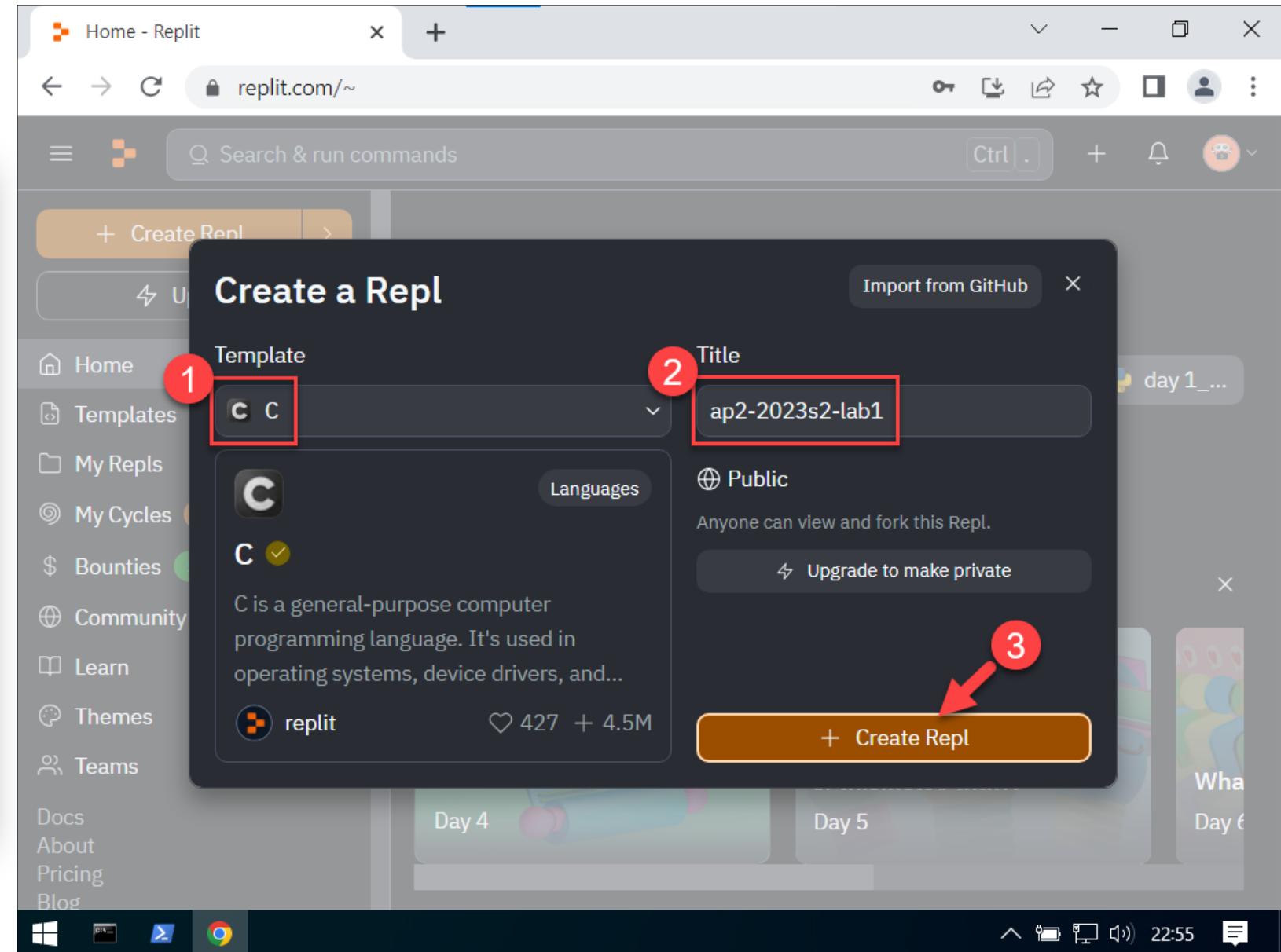
Efetue o login com sua conta



Inicie a criação de um novo Repl



Escolha o template C e defina o nome do Repl



Digite o código do programa

The screenshot shows the Replit IDE interface. At the top, the title bar reads "main.c - ap2-2023s2-lab1 - Replit". The address bar shows the URL "replit.com/@tmsdev/ap2-2023s2-lab1#main.c". Below the address bar, there's a navigation bar with icons for back, forward, search, and file operations. To the right of the navigation bar are buttons for "Deploy" and "Console".

The left sidebar has a "Search" field and a "Files" section containing a file named "main.c".

The main workspace displays the "main.c" code:1 /*
2 * Lab. 1
3 * Exemplo 1
4 */
5
6 #include <stdio.h>
7
8 int main() {
9 // imprime mensagens na tela
10 printf("Mackenzie - FCI\n");
11 printf("Algoritmos e ");
12 printf("Prog. II\n");
13 return 0;
14 }
15

The status bar at the bottom shows "Ln 15, Col 1 History".

At the very bottom, there's a taskbar with icons for Windows, Task View, File Explorer, Google Chrome, and File Explorer again.

Execute e verifique o resultado

The screenshot shows the Replit IDE interface. At the top, the URL is `replit.com/@tmsdev/ap2-2023s2-lab1#main.c`. A red arrow labeled '1' points to the play button icon in the toolbar. The code editor displays the `main.c` file:

```
1 /*  
2 * Lab. 1  
3 * Exemplo 1  
4 */  
5  
6 #include <stdio.h>  
7  
8 int main() {  
9     // imprime mensagens na tela  
10    printf("Mackenzie - FCI\n");  
11    printf("Algoritmos e ");  
12    printf("Prog. II\n");  
13    return 0;  
14 }  
15
```

A red box labeled '2' highlights the output in the terminal window, which shows the printed text:

```
>_ make -s  
>_ ./main  
Mackenzie - FCI  
Algoritmos e Prog. II
```

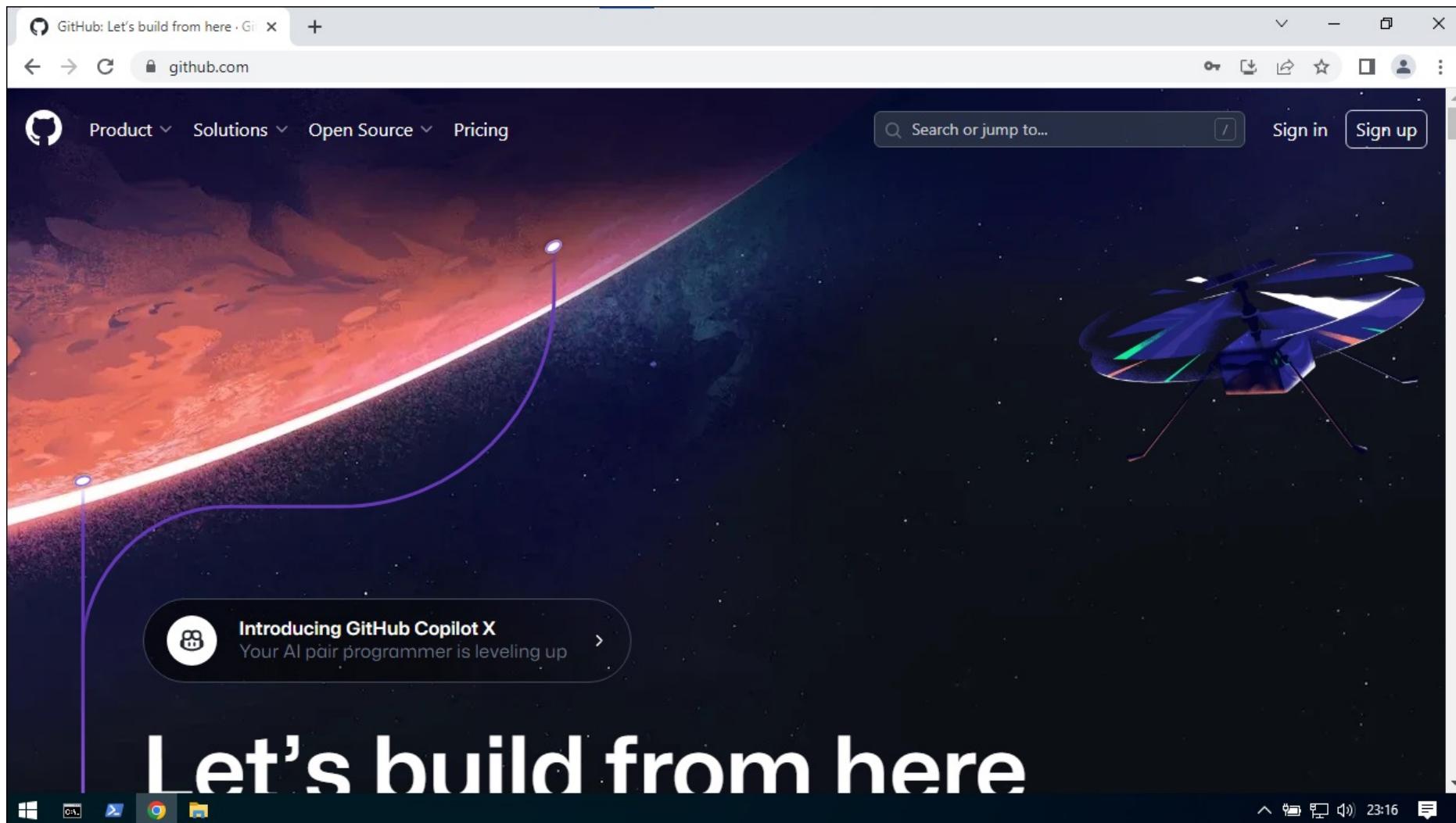
Exercício 2

Altere o código do programa de forma a imprimir também o nome da linguagem de programação que estudaremos nesta disciplina.

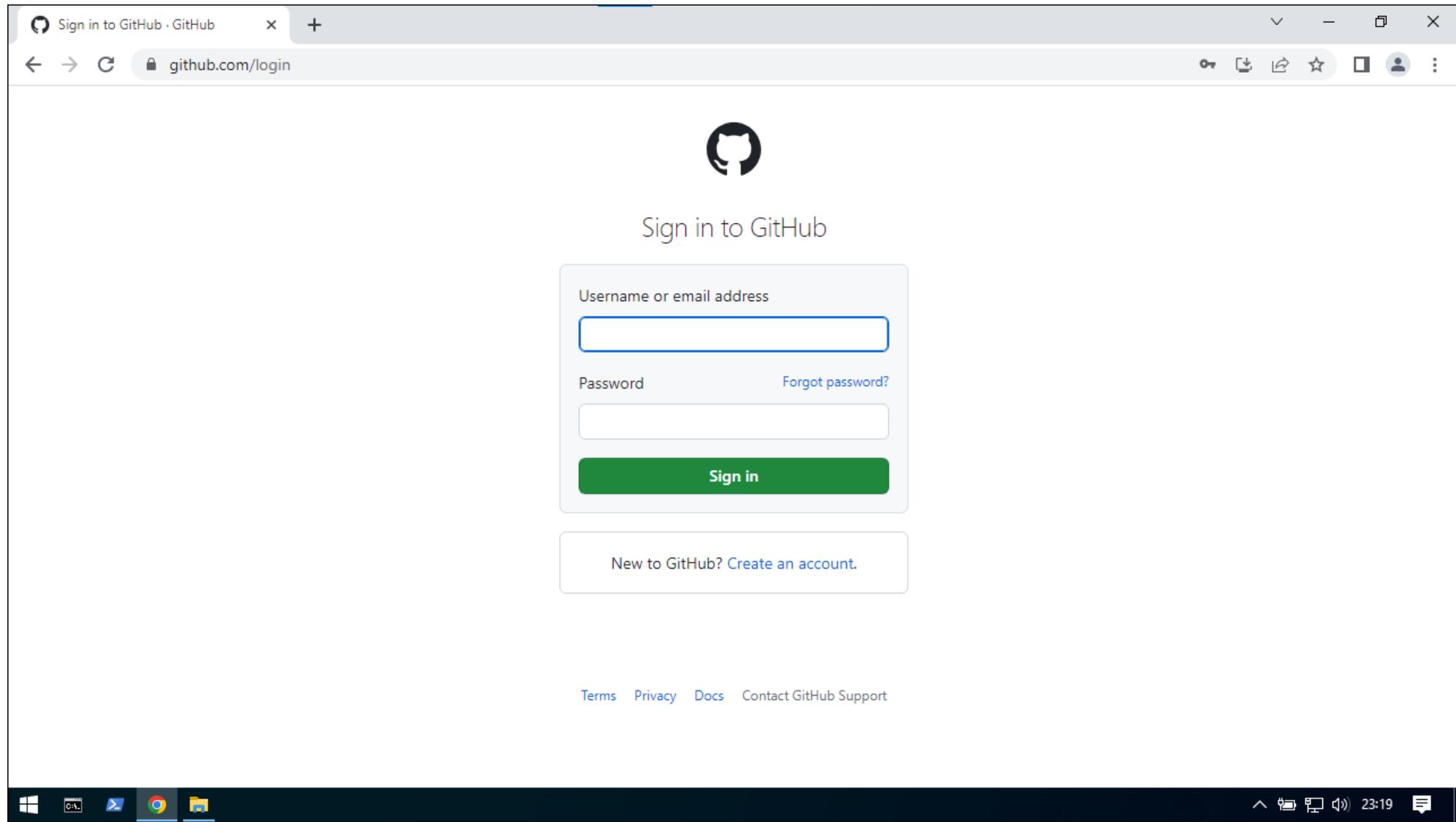
GitHub Codespace



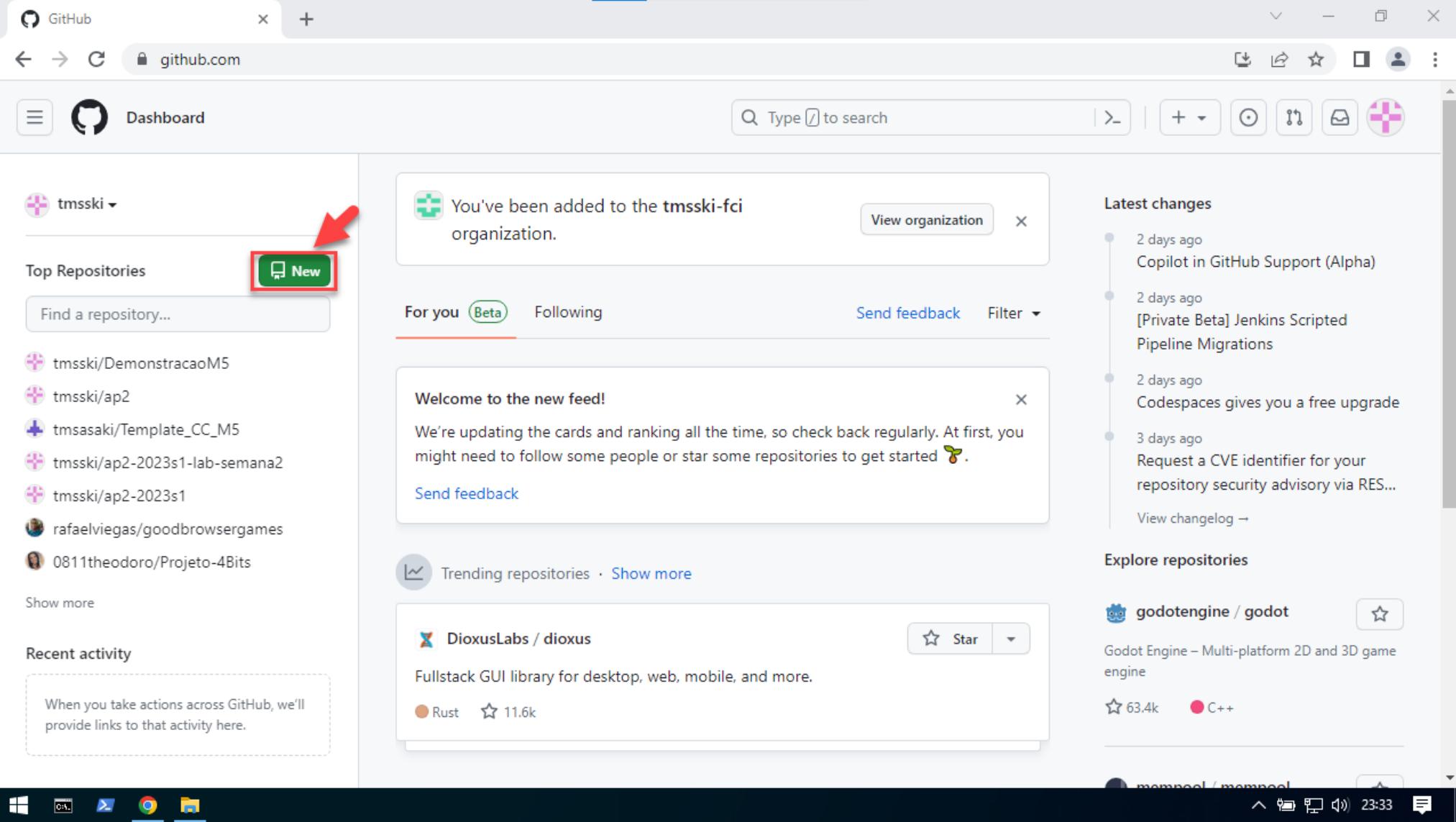
Acesse <https://github.com>



Efetue o login na sua conta



Inicie a criação de um novo repositório



A screenshot of the GitHub Dashboard. On the left, under 'Top Repositories', there is a green button labeled 'New' with a red arrow pointing to it. The dashboard also features a search bar at the top right, a sidebar with 'Latest changes' and 'Explore repositories', and a central feed area with a 'Welcome to the new feed!' message.

tmsski

New

Top Repositories

Find a repository...

tmsski/DemonstracaoM5

tmsski/ap2

tmsasaki/Template_CC_M5

tmsski/ap2-2023s1-lab-semana2

tmsski/ap2-2023s1

rafaelviegas/goodbrowsergames

0811theodoro/Projeto-4Bits

Show more

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

You've been added to the tmsski-fci organization.

View organization

For you Beta Following Send feedback Filter

Welcome to the new feed!

We're updating the cards and ranking all the time, so check back regularly. At first, you might need to follow some people or star some repositories to get started 🌱.

Send feedback

Trending repositories · Show more

DioxusLabs / dioxus

Fullstack GUI library for desktop, web, mobile, and more.

Rust ★ 11.6k

Latest changes

- 2 days ago Copilot in GitHub Support (Alpha)
- 2 days ago [Private Beta] Jenkins Scripted Pipeline Migrations
- 2 days ago Codespaces gives you a free upgrade
- 3 days ago Request a CVE identifier for your repository security advisory via RES...

View changelog →

Explore repositories

godotengine / godot

Godot Engine – Multi-platform 2D and 3D game engine

63.4k C++

mempool / mempool

Windows Taskbar icons

23:33

Defina o nome, adicione um README e crie o repositório

The screenshot shows the GitHub 'New repository' creation interface. A red circle labeled '1' highlights the 'Repository name' field, which contains 'ap2-2023s2-lab1'. A red circle labeled '2' points to the 'Initialize this repository with:' section, specifically the 'Add a README file' checkbox, which is checked. A red circle labeled '3' points to the green 'Create repository' button at the bottom right.

New repository

github.com/new

Required fields are marked with an asterisk (*).

Owner * tmsski Repository name * 1 / ap2-2023s2-lab1 ap2-2023s2-lab1 is available.

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

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:

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](#).

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

You are creating a public repository in your personal account.

Create repository

Crie um novo arquivo

The screenshot shows a GitHub repository page for 'tmsski/ap2-2023s2-lab1'. The repository is public and contains one commit named 'tmsski Initial commit' with a file named 'README.md'. On the right side of the page, there is a sidebar with various repository statistics and links. At the top center, there is a navigation bar with buttons for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. Below the navigation bar, there is a search bar and a toolbar with icons for pinning, unwatching, forking, and starring the repository. A red circle with the number '1' is placed over the 'Add file' button in the toolbar. A red arrow with the number '2' points to the 'Create new file' option in a dropdown menu that appears when the 'Add file' button is clicked. The dropdown menu also includes 'Upload files'.



Defina o nome do arquivo, digite o código e inicie o *commit*

The screenshot shows a GitHub commit interface. At the top, the repository is set to `tmsski / ap2-2023s2-lab1`. A red circle labeled '1' highlights the file path `ap2-2023s2-lab1/exemplo1.c`, which is circled in red. The code editor contains the following C code, with lines 1 through 15 highlighted by a red box labeled '2':

```
1  /* Lab. 1
2   * Exemplo 1 */
3
4 #include <stdio.h>
5
6 int main() {
7     // imprime algumas mensagens na tela
8     printf("Mackenzie - FCI\n");
9     printf("Algoritmos e ");
10    printf("Prog. II\n");
11
12
13
14    return 0;
15 }
```

A red arrow labeled '3' points to the green **Commit changes...** button.

At the bottom, a note says: "Use `Control + Shift + m` to toggle the `tab` key moving focus. Alternatively, use `esc` then `tab` to move to the next interactive element on the page."

M 27

Confirme a operação de *commit*

Commit changes ×

Commit message

Extended description

Add an optional extended description..

Commit directly to the `main` branch
 Create a **new branch** for this commit and start a pull request
[Learn more about pull requests](#)

Cancel Commit changes

Crie um codespace

The screenshot shows a GitHub repository page for 'ap2-2023s2-lab1'. The repository is public and has one branch ('main') and one file ('exemplo1.c'). The 'Code' tab is selected. A dropdown menu is open at the top right, with the 'Code' button circled in red (step 1). Inside the dropdown, the 'Codespaces' tab is also circled in red (step 2). A large red arrow points from step 3 to the 'Create codespace on main' button in the center of the dropdown menu. The 'About' section indicates no description, website, or topics are provided.

tmsski / ap2-2023s2-lab1

Type ⌘ to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

ap2-2023s2-lab1 Public

Pin Unwatch 1 Fork 0 Star 0

main 1 branch 0 tags

Go to file Add file ▾ <> Code ▾

Local Codespaces

No codespaces

You don't have any codespaces with this repository checked out

Create codespace on main

Learn more about codespaces...

About

No description, website, or topics provided.

Readme Activity 0 stars 1 watching 0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

1

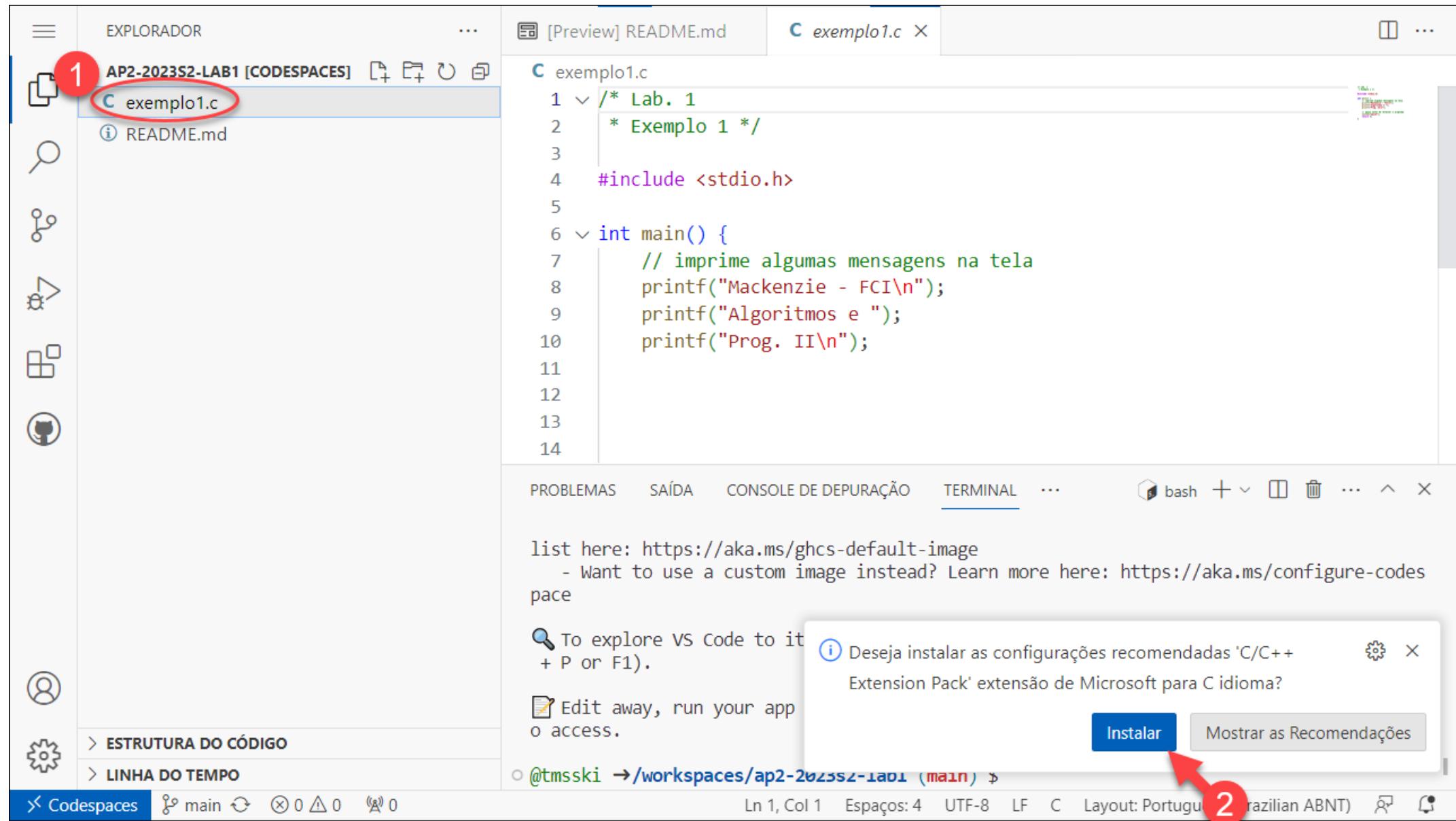
2

3

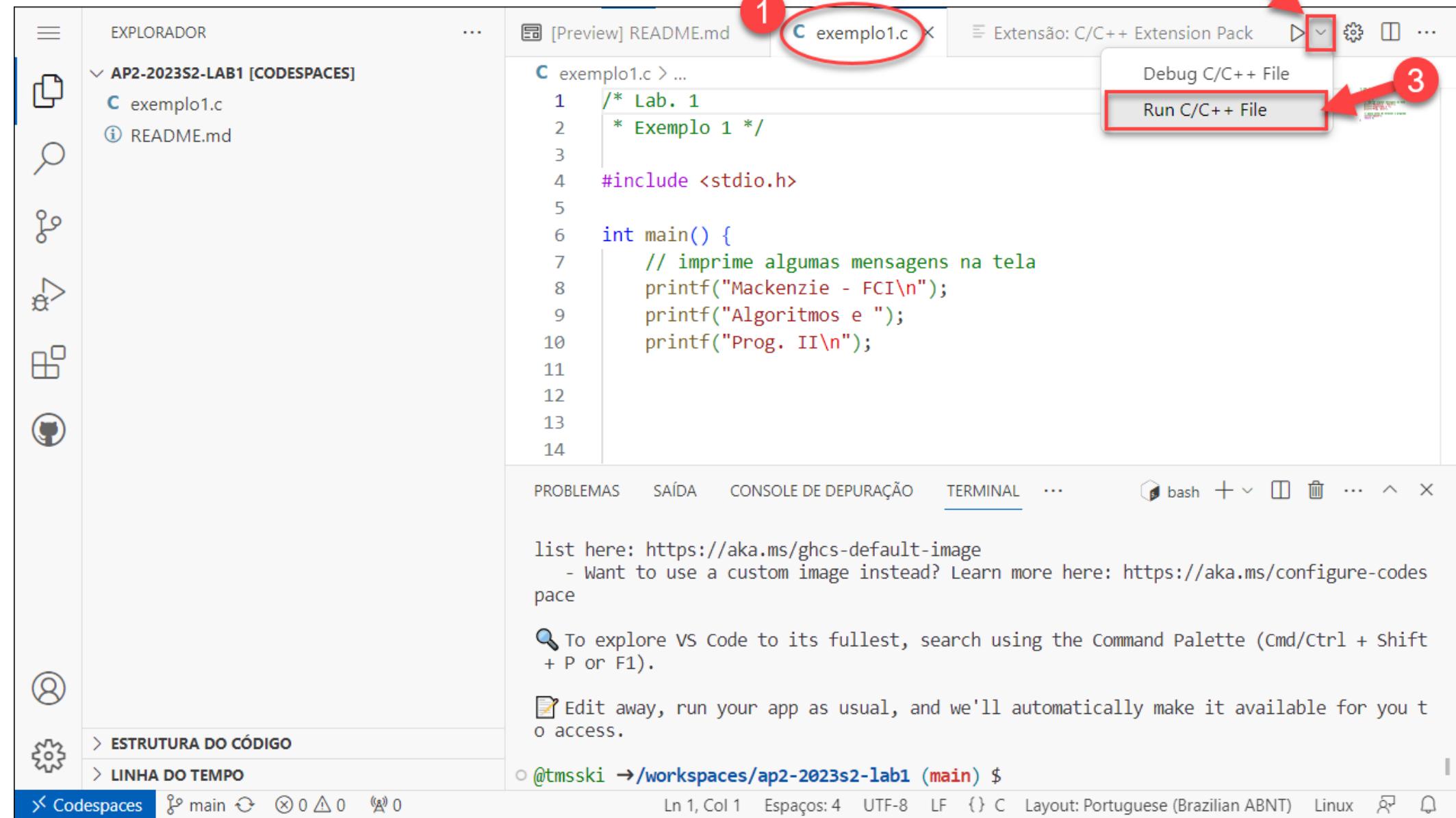
M

29

Selecione exemplo1.c e pressione Instalar



Execute exemplo1.c



Selecione a 1ª opção de compilador

The screenshot shows the Visual Studio Code interface with the following details:

- Left Sidebar (Explorador):** Shows a workspace named "AP2-2023S2-LAB1 [CODESPACE]" containing files "exemplo1.c" and "README.md".
- Top Bar:** Includes a search bar labeled "Selecionar uma configuração de depuração" and a red arrow pointing to it.
- Task Selection:** A dropdown menu lists three options:
 - C/C++: gcc Compilar e depurar o arquivo ativo preLaunchTask: C/C++: gcc arquivo de build ativo
 - Tarefa Detectada (compilador: /usr/bin/gcc)
 - C/C++: gcc-9 Compilar e depurar o arquivo ativo preLaunchTask: C/C++: gcc-9 arquivo de build ativo
 - Tarefa Detectada (compilador: /usr/bin/gcc-9)
 - (gdb) Iniciar
- Code Editor:** Displays a C/C++ code snippet:

```
6     int main() {  
7         // imprime algumas mensagens na tela  
8         printf("Mackenzie - FCI\n");  
9         printf("Algoritmos e ");  
10        printf("Prog. II\n");  
11  
12  
13  
14
```
- Bottom Navigation:** Tabs include PROBLEMAS, SAÍDA, CONSOLE DE DEPURAÇÃO, TERMINAL (selected), and others. The TERMINAL tab shows a message about default image configuration.
- Bottom Status Bar:** Shows file path "@tmsski →/workspaces/ap2-2023s2-lab1 (main) \$", line information "Ln 1, Col 1", encoding "UTF-8", and layout settings.
- Page Number:** "32" in the bottom right corner.

Verifique o resultado da execução

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. On the left is the Explorer sidebar, which displays the project structure under 'AP2-2023S2-LAB1 [CODESPACES]'. The 'exemplo1' folder contains files '.vscode', 'exemplo1.c', and 'README.md'. The 'exemplo1.c' file is open in the main editor area, showing the following code:

```
/* Lab. 1
 * Exemplo 1 */

#include <stdio.h>

int main() {
    // imprime algumas mensagens na tela
    printf("Mackenzie - FCI\n");
    printf("Algoritmos e ");
    printf("Prog. II\n");

    return 0;
}
```

Below the editor, the 'TERMINAL' tab is selected, showing the output of the program execution:

```
Mackenzie - FCI
Algoritmos e Prog. II
[1] + Done          "/usr/bin/gdb" --interpreter=mi --tty=
${DbgTerm} 0<"/tmp/Microsoft-MIEngine-In-wsromh2r.wm2" 1>"/tmp/Microso
t-MIEngine-Out-sfqvxma2.pku"
@tmsski →/workspaces/ap2-2023s2-lab1 (main) $
```

The status bar at the bottom provides information about the current file ('main'), line count (Ln 10), character count (Col 26), encoding (UTF-8), and layout (Portuguese (Brazilian ABNT)). It also indicates the operating system is Linux.

Faça algumas alterações e salve (Control + S)

The screenshot shows the Microsoft Visual Studio Code interface with the following details:

- EXPLORADOR** sidebar: AP2-2023S2-LAB1 [CODESPACES] folder containing .vscode, exemplo1, exemplo1.c, and README.md. The exemplo1.c file is currently selected.
- Editor**: The code editor displays the content of exemplo1.c:

```
* Exemplo 1 *
#include <stdio.h>
int main() {
    // imprime algumas mensagens na tela
    printf("Mackenzie - FCI\n");
    printf("Ciência da Computação\n");
    printf("Algoritmos e Programação II\n");
    return 0;
}
```

The lines from line 9 to line 12 are highlighted with a red rectangle.
- Terminal**: The terminal window shows the output of the program:

```
Mackenzie - FCI
Algoritmos e Prog. II
[1] + Done          "/usr/bin/gdb" --interpreter=mi --tty=
${DbgTerm} 0<"/tmp/Microsoft-MIEngine-In-wsromh2r.wm2" 1>"/tmp/Microso
t-MIEngine-Out-sfqvxma2.pku"
@tmsski →/workspaces/ap2-2023s2-lab1 (main) $
```
- Bottom Status Bar**: Shows the file name main.c, line 13, column 1, and other status information like layout and encoding.
- Bottom Right Corner**: Shows the page number 34.

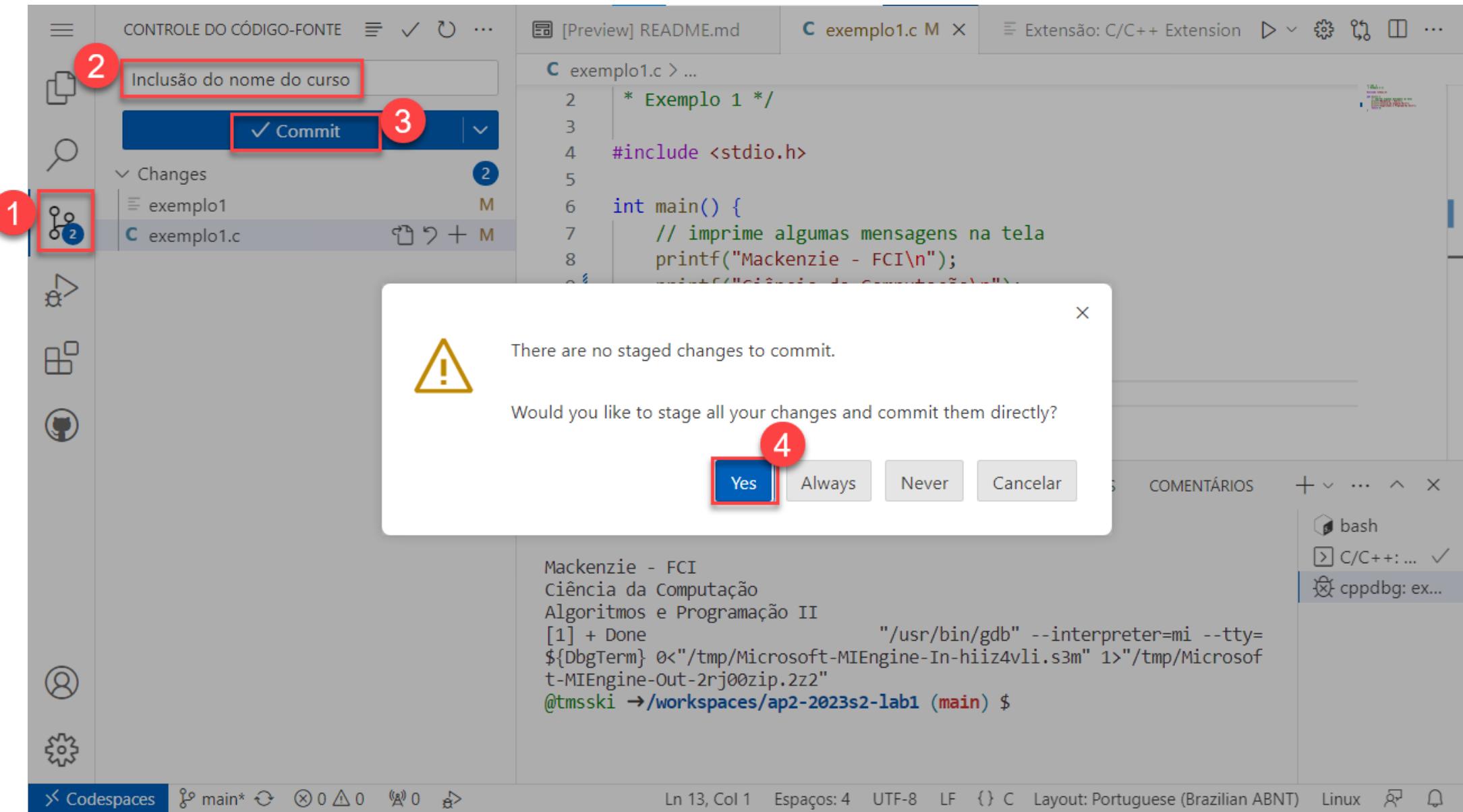
Execute novamente

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORADOR:** Shows the folder structure `AP2-2023S2-LAB1 [CODESPACES]` containing files `.vscode`, `exemplo1`, `exemplo1.c`, and `README.md`.
- EDITOR:** Displays the content of `exemplo1.c` with syntax highlighting.
- TERMINAL:** Shows the output of the program, which includes:
 - Text output: "Mackenzie - FCI", "Ciência da Computação", "Algoritmos e Programação II". This output is highlighted with a purple rectangle.
 - GDB session output:

```
[1] + Done "/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<"/tmp/Microsoft-MIEngine-In-hiiz4vli.s3m" 1>"/tmp/Microsoft-MIEngine-Out-2rj00zip.2z2"
@tmsski →/workspaces/ap2-2023s2-lab1 (main) $
```
- EXTENSIONS:** Shows the extension `C/C++` is installed and active.
- STATUS BAR:** Shows the file is `main*`, has 0 changes, 0 errors, 0 warnings, and 0 info messages. It also shows the current file is `exemplo1.c`, the encoding is UTF-8, and the layout is Portuguese (Brazilian ABNT).

Faça o *commit* das alterações



Envie as alterações para o repositório

The screenshot shows the Visual Studio Code interface with the GitHub Extension Pack installed. The main area displays a C file named 'exemplo1.c' containing a simple program that prints three messages: 'Mackenzie - FCI', 'Ciência da Computação', and 'Algoritmos e Programação II'. Below the code editor, the 'TERMINAL' tab is active, showing the output of running the program with GDB. On the left, the 'CONTROLE DO CÓDIGO-FONTE' (Source Control) sidebar is open, showing a commit message input field and a prominent blue button labeled 'Sync Changes 1↑'. A red arrow points to this button. The bottom status bar indicates the file is at line 13, column 1, with 4 spaces, and the layout is set to Portuguese (Brazilian ABNT).

```
* Exemplo 1 *
#include <stdio.h>
int main() {
    // imprime algumas mensagens na tela
    printf("Mackenzie - FCI\n");
    printf("Ciência da Computação\n");
    printf("Algoritmos e Programação II\n");
    return 0;
}
```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL PORTAS COMENTÁRIOS + ... ^ X

```
Mackenzie - FCI
Ciência da Computação
Algoritmos e Programação II
[1] + Done          "/usr/bin/gdb" --interpreter=mi --tty=
${DbgTerm} 0<"/tmp/Microsoft-MIEngine-In-hiiz4vli.s3m" 1>"/tmp/Microsof
t-MIEngine-Out-2rj00zip.2zz"
@tmsski →/workspaces/ap2-2023s2-lab1 (main) $
```

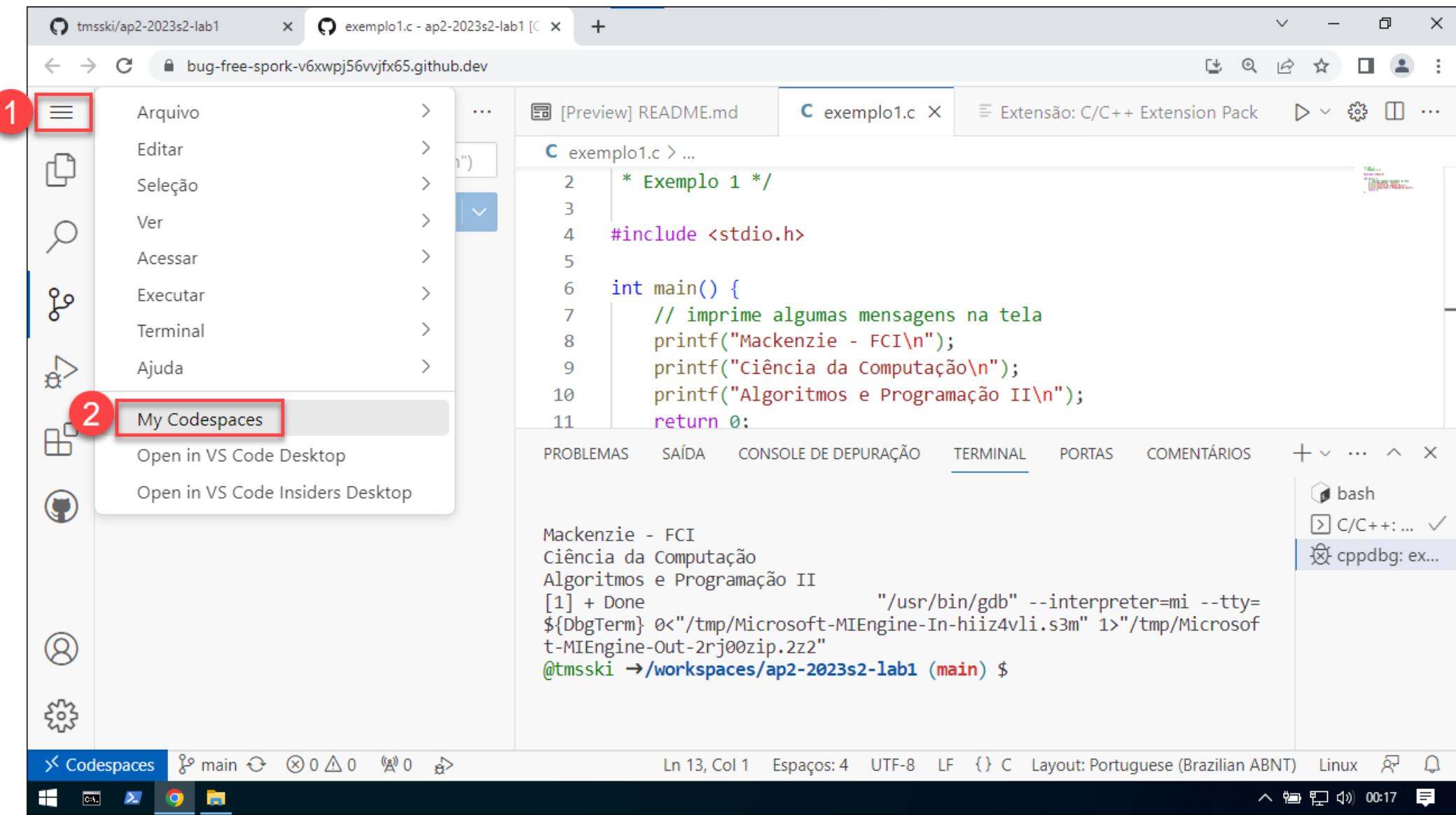
Ln 13, Col 1 Espaços: 4 UTF-8 LF {} C Layout: Portuguese (Brazilian ABNT) Linux

Exercício 3

Altere o código do programa de forma a imprimir também o ano em que a FCI foi fundada.

Lembre-se de efetuar o *commit* e enviar a modificação para o seu repositório.

Quando finalizar, vá para “My Codespaces”



Pare o seu codespace (há um limite mensal para o serviço gratuito)

The screenshot shows the GitHub Codespaces interface. At the top, there are three tabs: 'tmsski/ap2-2023s2-lab1', 'exemplo1.c - ap2-2023s2-lab1 [C]', and 'Codespaces'. The 'Codespaces' tab is active. Below the tabs, the URL 'github.com/codespaces/' is displayed. The main content area shows three template cards: 'Start with a blank canvas or import any packages you need.', 'A popular JavaScript library for building user interfaces based on UI components.', and 'JupyterLab is the latest web-based interactive development environment for notebooks, code, and data.' Each card has a 'Use this template' button.

In the center, there is a list titled 'Owned by tmsski' containing one item: 'tmsski/ap2-2023s2-lab1' (bug-free spork, main). To the right of this list is a context menu for the codespace. The menu items are: 'Open in ...', 'Rename', 'Export changes to a branch', 'Change machine type', 'Keep codespace', 'Stop codespace' (highlighted with a red box and the number 2), and 'Delete'. A red circle with the number 1 is drawn around the three dots at the end of the menu.

At the bottom of the page, there is a navigation bar with links: 'Terms', 'Privacy', 'Security', 'Status', 'Docs', 'Contact GitHub', 'Pricing', 'API', 'Training', and 'Blog'. On the far left, there is a circular icon with a large letter 'M'. The bottom right corner of the screen shows the Windows taskbar with icons for File Explorer, Google Chrome, Task View, and Start.

Entrega



Envie na tarefa “Lab. 1” os seguintes itens:

- os arquivos “exemplo1.c” e “exemplo1.exe” do exercício com o Dev-C++;
- a URL do seu Repl no **repl.it**;
- a URL do seu repositório GitHub.



Universidade Presbiteriana
Mackenzie



Faculdade de
Computação e Informática