

Facy Wyden
Campus Batista Campos

Professor/Coordenador: Alexandre Abreu de Freitas

Aluno: Lucas Vinicius Silveira (202402390562).

Tema: Implementação de programas em C e em Python.

Belém / Pará

2024

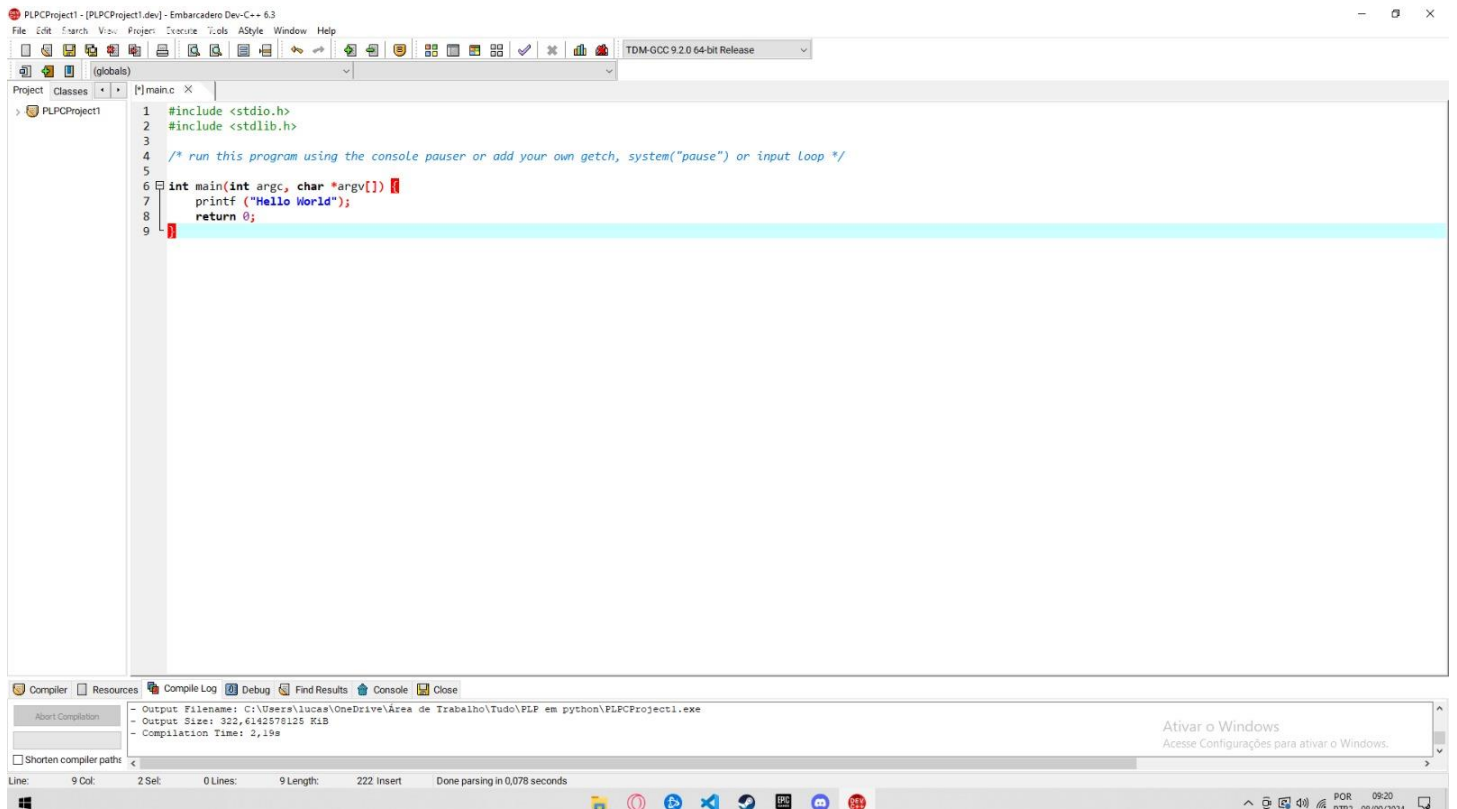
1. Introdução

Neste trabalho será realizado uma pesquisa e comparação entre as linguagens de programação C e python, com o objetivo de testar a efetividade de cada uma. Será desenvolvido dois programas em cada linguagem, um que tem como objetivo imprimir na tela do usuário a mensagem “Hello world”, e outra que tem como objetivo fazer um contador que irá imprimir todos os números de 1 até 1000. Por fim, será comparado a efetividade de cada linguagem levando em relação o tempo de execução e a facilidade de escrita.

2. Implementação do programa “Hello World”.

2.1 Implementação do programa em C.

Abaixo será mostrado a imagem do código por completo junto com sua execução e o tempo de resposta para a primeira execução.



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 /* run this program using the console pauser or add your own getch, system("pause") or input loop */
5
6 int main(int argc, char *argv[]) {
7     printf("Hello World");
8     return 0;
9 }
```

Compiler: TDM-GCC 9.2.0 64-bit Release

Output File: C:\Users\lucas\OneDrive\Área de Trabalho\Tudo\PLP em python\PLCProject1.exe
Output Size: 322,6142578125 KiB
Compilation Time: 2,19s

Line: 9 Col: 2 Sel: 0 Lines: 9 Length: 222 Insert Done parsing in 0,078 seconds

Código:

```
1 #include <stdio.h>

2 #include <stdlib.h>

3

4 int main (int argc, char *argv[]) {

5     printf ("Hello World");

6     return 0;

7 }
```

Explicação:

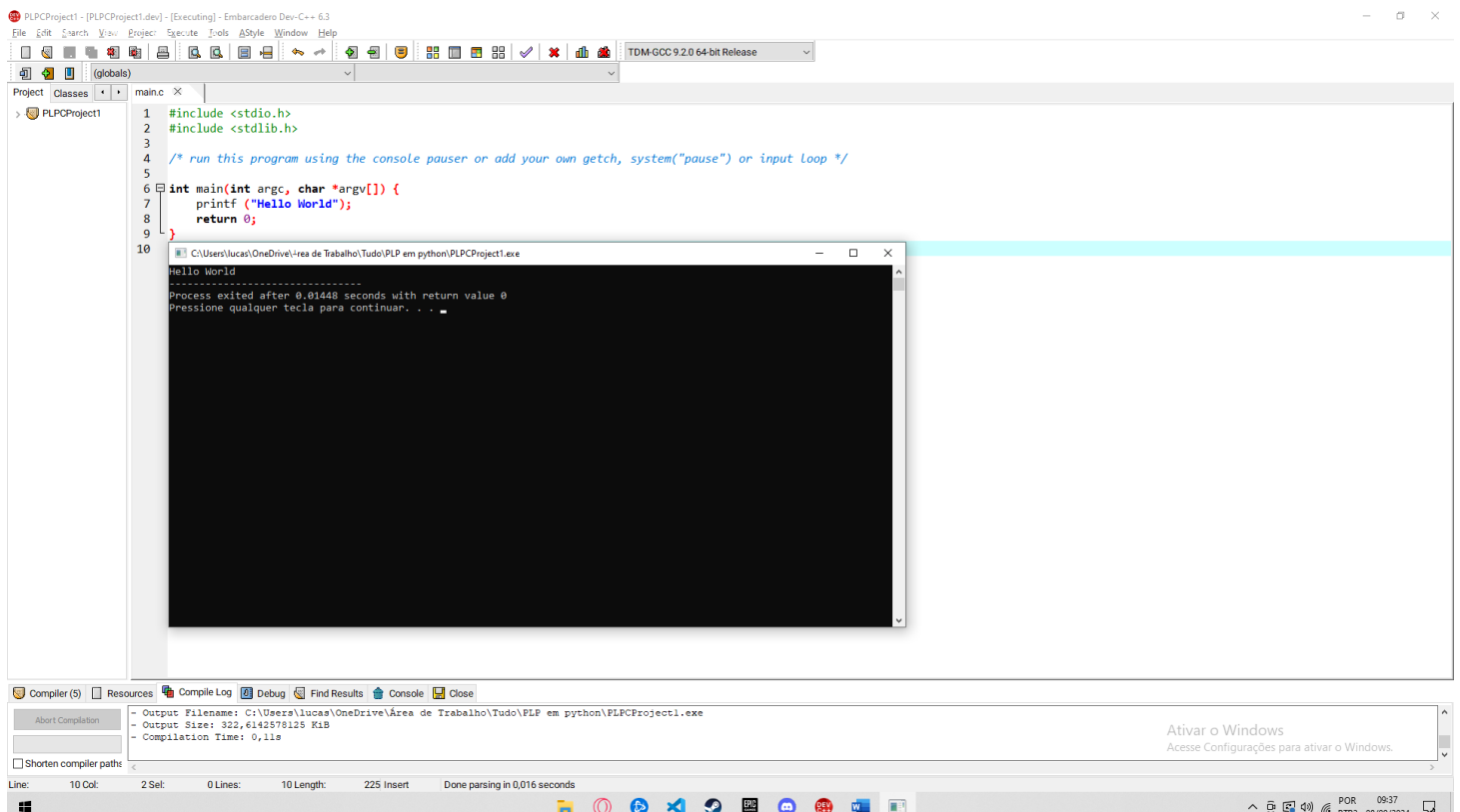
Nas duas primeiras linhas existe a implementação das bibliotecas necessárias para a confecção do código.

Na 4ª linha existe a criação do Int Main, No qual todo o código será escrito.

Na 5ª linha é dado a ordem de imprimir a mensagem “Hello World” para o usuário.

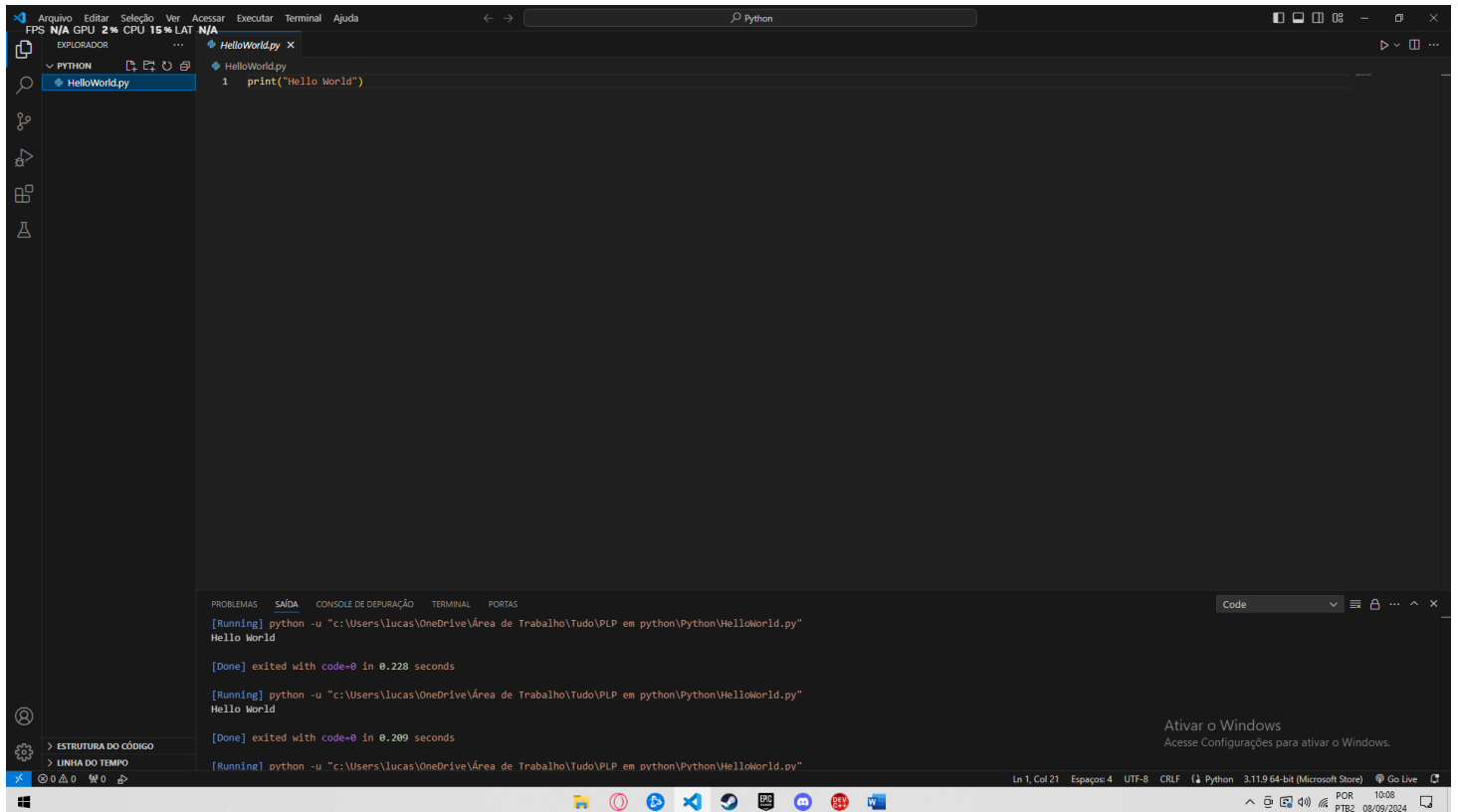
Na 6ª linha é escrita a função que retornará 0 caso o código seja executado corretamente.

Abaixo será mostrado a execução do código pelo console:



2.2 Implementação do programa em Python.

Abaixo será mostrado a imagem do código por completo junto com sua execução e o tempo de resposta para a primeira execução.



Código:

```
1 print("Hello World")
```

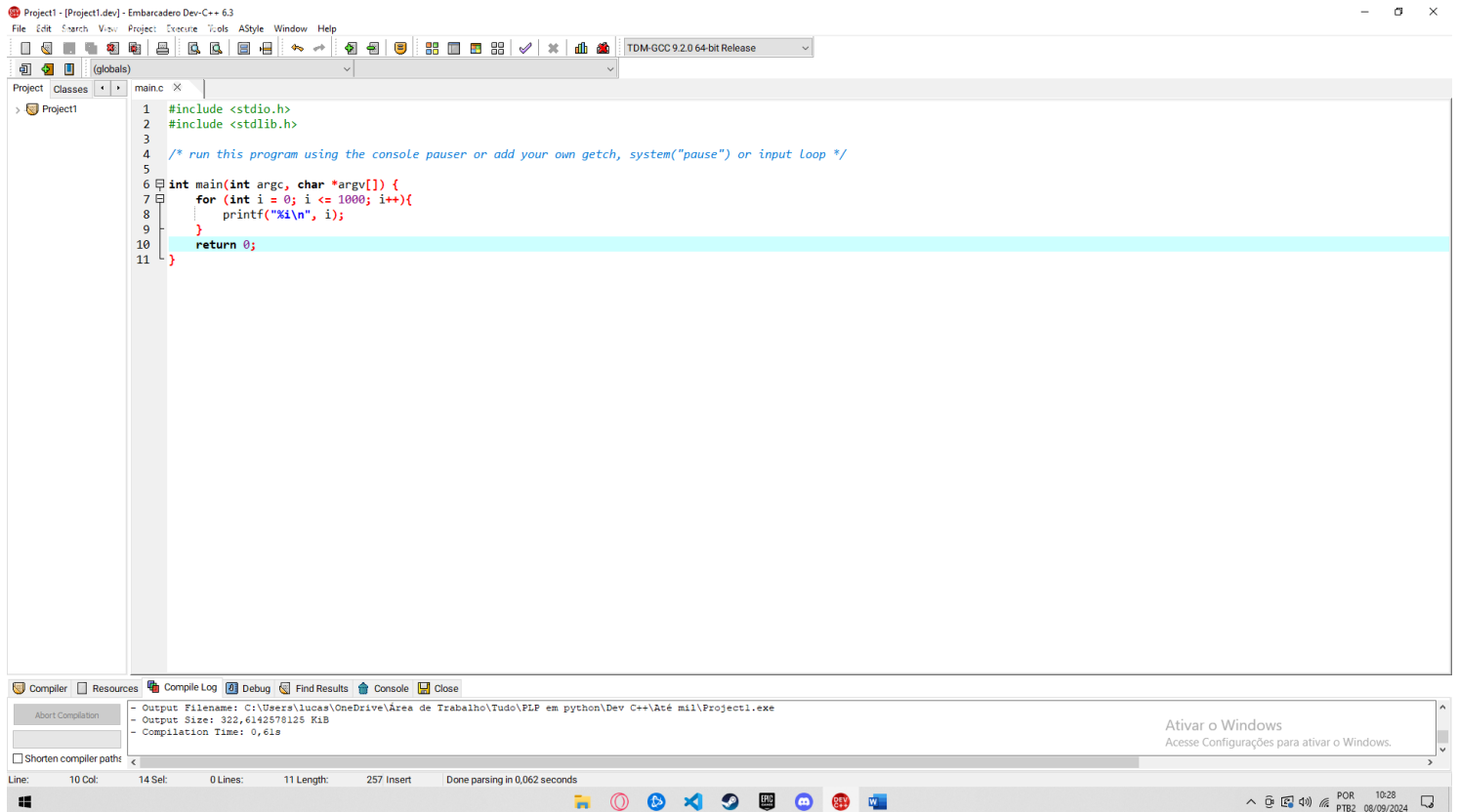
Explicação:

Na primeira e única linha é dada a ordem de imprimir na tela do usuário a mensagem "Hello world".

3. Implementação do programa "Até 1000"

3.1 Implementação do programa em C.

Abaixo será mostrado a imagem do código por completo junto com sua execução e o tempo de resposta para a primeira execução



Código:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5
6 int main(int argc, char *argv[]) {
7     for (int i = 0; i <= 1000; i++){
8         printf("%i\n", i);
9     }
10    return 0;
11 }
```

Explicação:

Nas duas primeiras linhas foi declarado as bibliotecas necessárias.

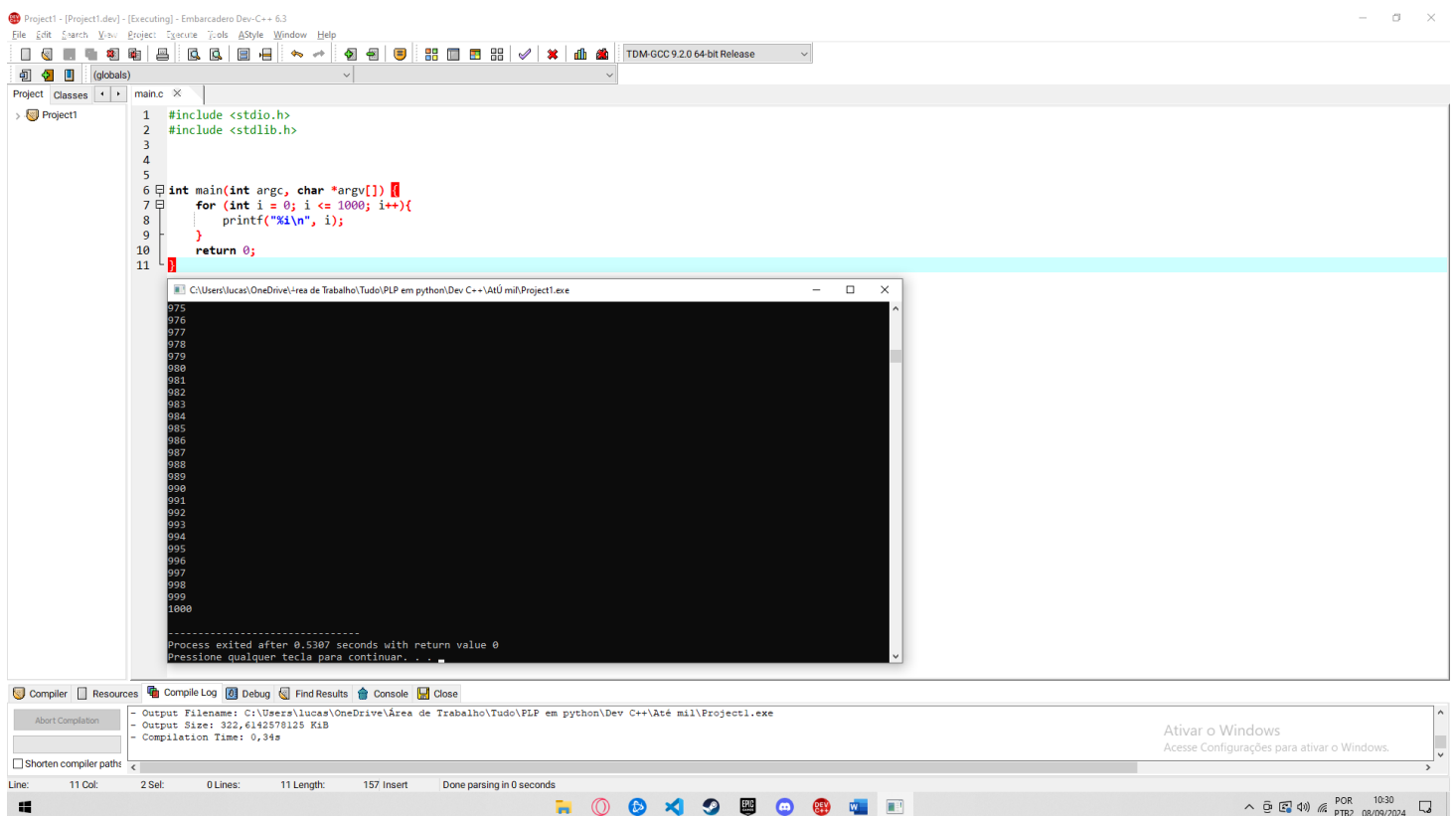
Na 6ª linha existe a criação do `int main`, No qual todo o código será escrito.

Na 7ª foi criado usando a função `for` um laço de repetição que só irá terminar quando a variável criada `i` foi maior que 1000.

Na 8ª foi dada a ordem de imprimir a variável `i`.

Na 10ª linha é escrita a função que retornará 0 caso o código seja executado corretamente.

Abaixo será mostrado a execução do código pelo console:



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5
6 int main(int argc, char *argv[]) {
7     for (int i = 0; i <= 1000; i++){
8         printf("%i\n", i);
9     }
10    return 0;
11 }
```

Process exited after 0.5307 seconds with return value 0
Pressione qualquer tecla para continuar.

Compiler: TDM-GCC 9.2.0 64-bit Release
Output Filename: C:\Users\lucas\OneDrive\Área de Trabalho\Tudo\PLP em python\Dev C++\Até mil\Project1.exe
Output Size: 322,6142578125 KiB
Compilation Time: 0,34s

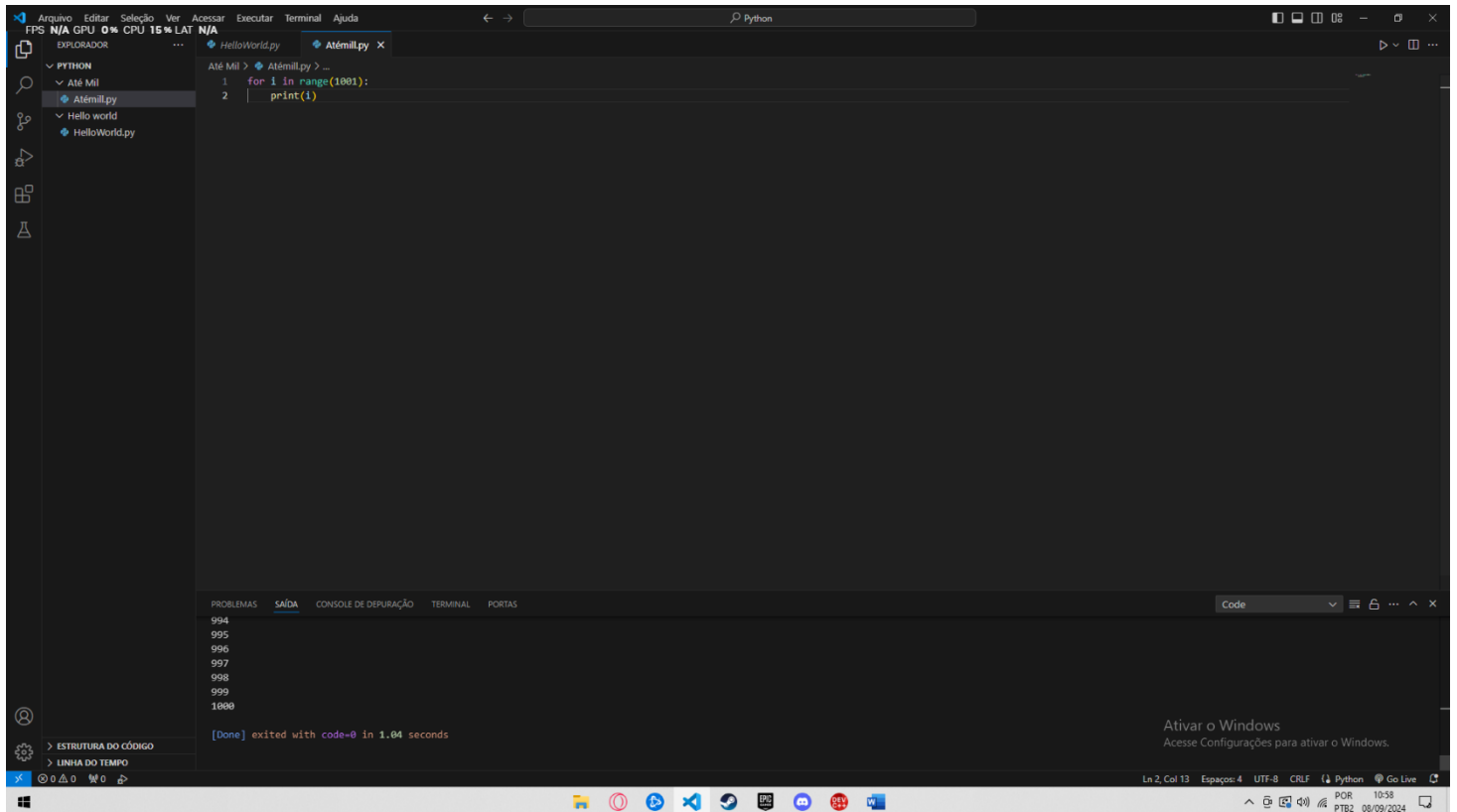
Line: 11 Col: 2 Sel: 0 Lines: 11 Length: 157 Insert Done parsing in 0 seconds

Ativar o Windows
Acesse Configurações para ativar o Windows.

POR PTB2 10:30 08/09/2024

3.2 Implementação do programa em Python.

Abaixo será mostrado a imagem do código por completo junto com sua execução e o tempo de resposta para a primeira execução.



Código:

```
1 for i in range (1001):
```

```
2  print(i)
```

Explicação:

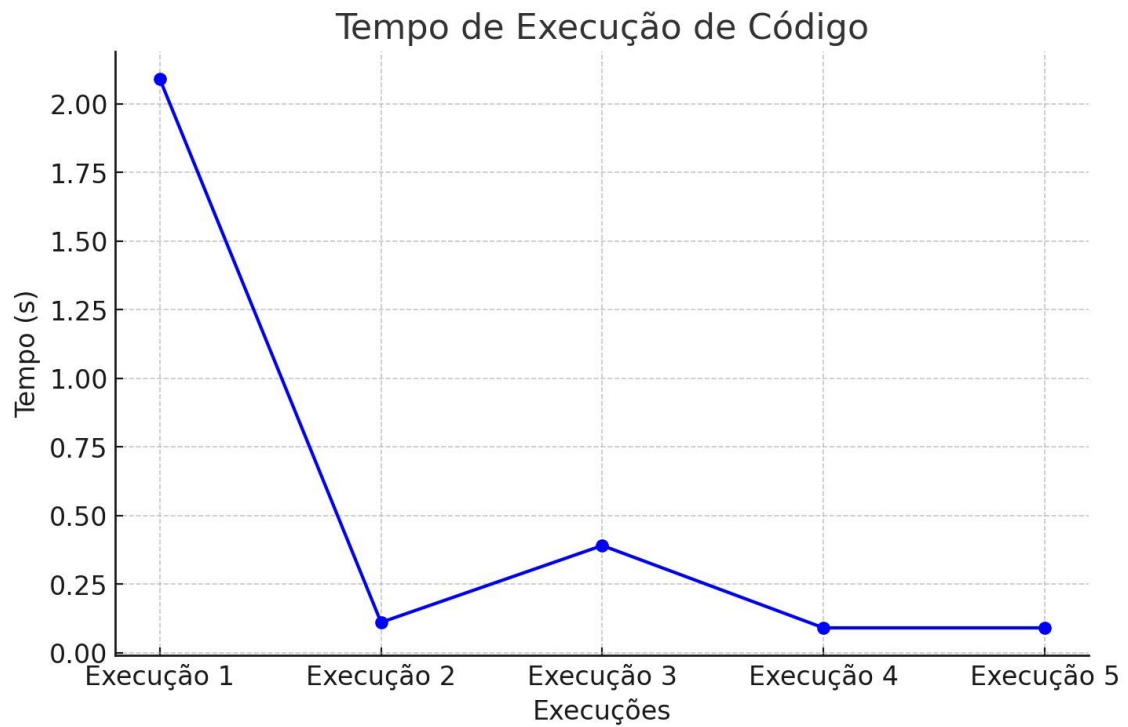
Na primeira linha é criado um loop for que atribui pra variável i o valor de 1 a 1000 usando a função range().

Na segunda linha é dada a ordem de imprimir a variável i para o usuário.

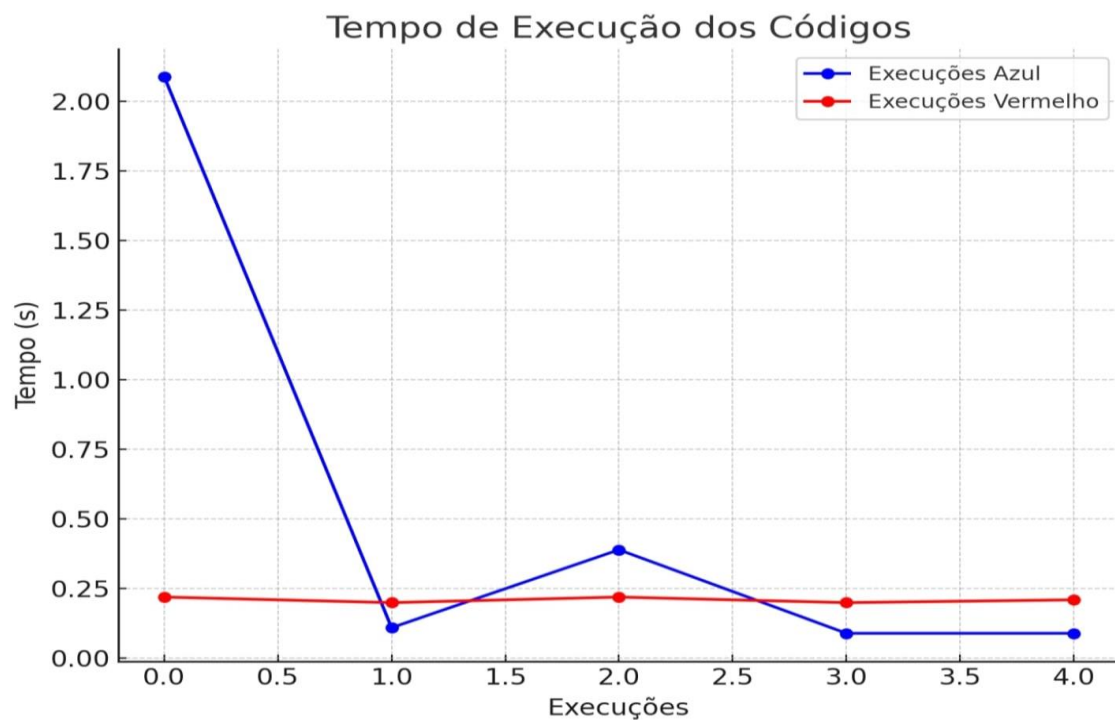
4. Conclusão

4.1 Comparação de tempo de execução

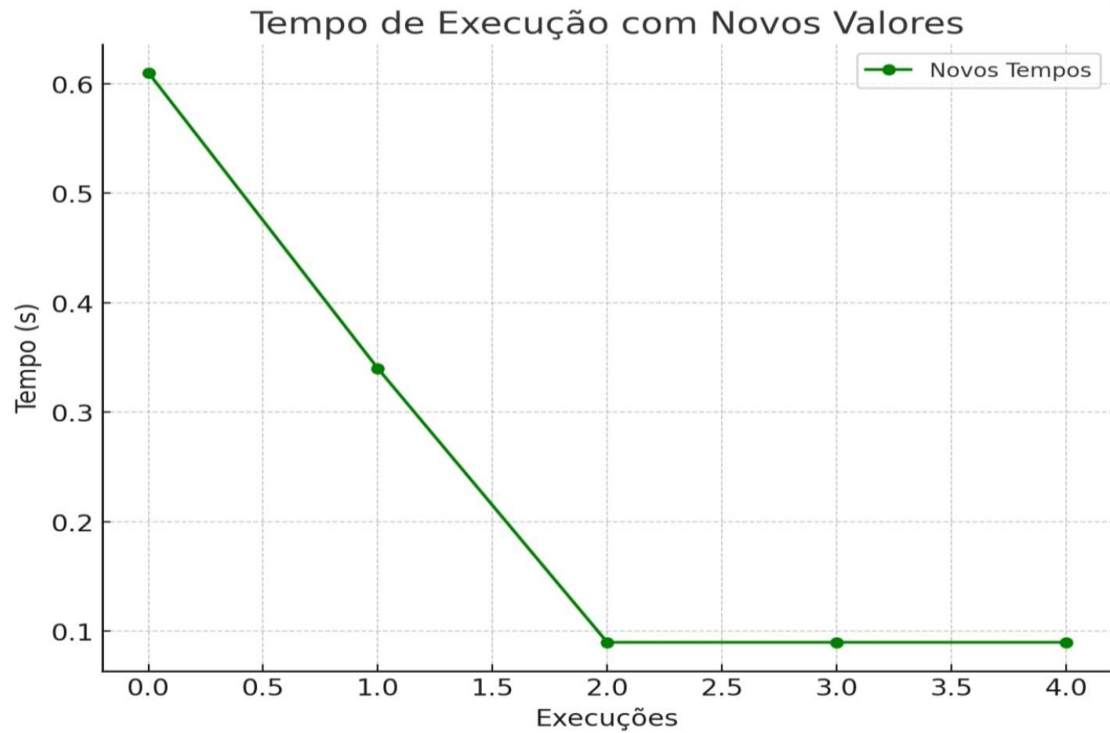
1 – Gráfico do tempo de execução do programa “Hello World” realizado em C.



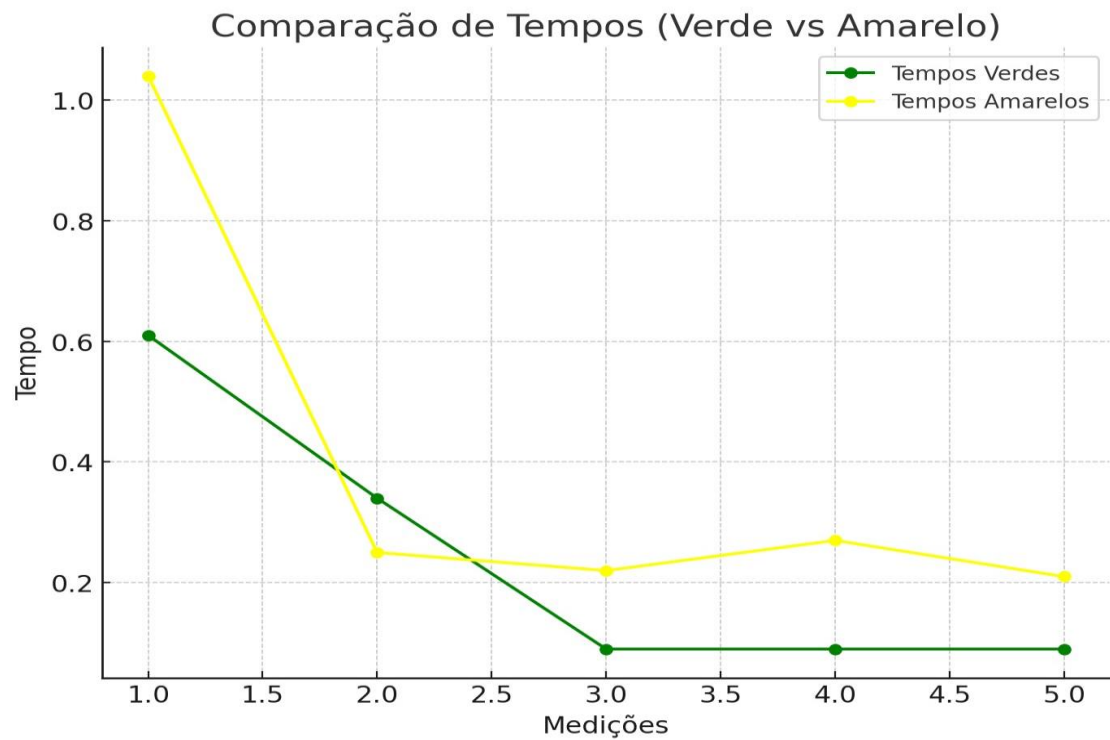
2 – Gráfico da comparação do tempo de execução do programa “Hello World” realizado em C e em Python.



3 – Gráfico do tempo de execução do programa “Até mil” realizado em C



4 – Gráfico da comparação do tempo de execução do programa “Até mil” realizado em C e em Python.



4.2 Implementações das linguagens

A linguagem de programação Python permite escrever códigos de maneira significativamente mais simples e concisa em comparação com a linguagem C. Um código que em C pode exigir 10 linhas, muitas vezes pode ser escrito em apenas uma linha em Python, graças à sua sintaxe mais enxuta e ao alto nível de abstração que oferece. A linguagem de programação Python permite escrever códigos de maneira significativamente mais simples e concisa em comparação com a linguagem C. Um código que em C pode exigir 10 linhas, muitas vezes pode ser escrito em apenas uma linha em Python, graças à sua sintaxe mais enxuta e ao alto nível de abstração que oferece.

Quanto ao tempo de execução, nota-se que nas primeiras execuções do código em C, há uma necessidade considerável de tempo devido à criação do programa executável. No entanto, após essa etapa inicial, o código é executado de forma muito mais rápida, com tempos variando entre 0,11 e 0,09 segundos. Em contraste, no Python, não há uma diferença tão significativa entre a primeira e as execuções subsequentes. O tempo de execução se mantém mais estável, em torno de 0,20 segundos, refletindo um padrão mais consistente, mesmo sem a necessidade de compilação prévia.

4.3 Descrição de hardware e sistema operacional

Nesta pesquisa foi usado os respectivos aplicativos:

Dev C++: usado para o desenvolvimento de códigos em C.

Visual Studio Code: Usado para o desenvolvimento de códigos em Python.

Sistema operacional:

O sistema operacional utilizado neste trabalho foi o Windows 10 Home, versão 22H2, rodando em arquitetura de 64 bits.

Hardware:

Processador (CPU): Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz 2.90 GHz
Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz 2.90 GHz

Memoria Ram: 16,0 GB DDR4, 2666 mhz

Armazenamento: SSD M.2 512 GB.

Placa de Vídeo: NVIDIA GeForce RTX 3060 Ti.