

Technologie informacyjne

przygotowanie środowiska

Andrzej Giniewicz

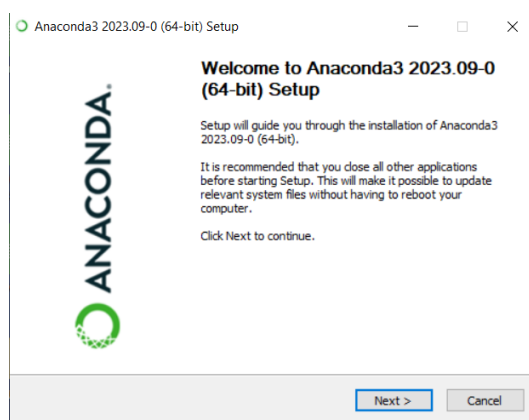
06.03.2024

1 Python

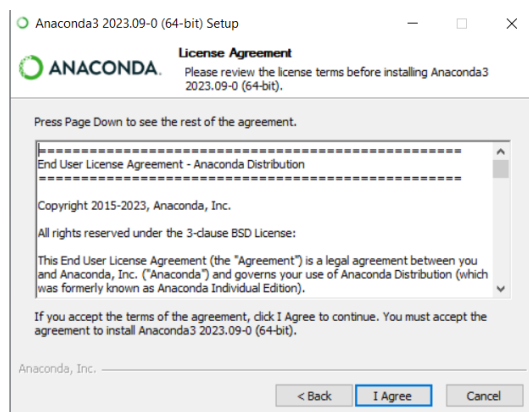
1.1 System Windows

Na systemie Windows polecamy dystrybucję Anaconda, która zawiera wiele preinstalowanych bibliotek. Aby pobrać Pythona i wszystkie biblioteki potrzebne do zajęć, wejdź na stronę <https://www.anaconda.com/download> i ściągnij instalator dla swojej wersji systemu operacyjnego (najprawdopodobniej wariant 64-bitowy).

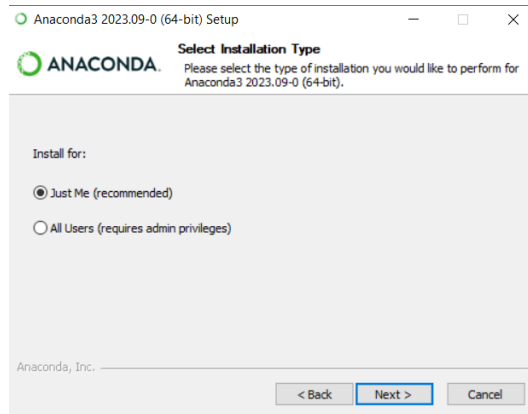
Po uruchomieniu pobranego pliku pojawi się instalator.



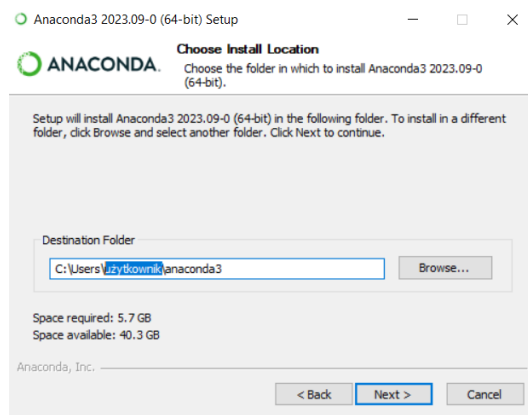
Musimy wyrazić zgodę na licencję użytkownika.



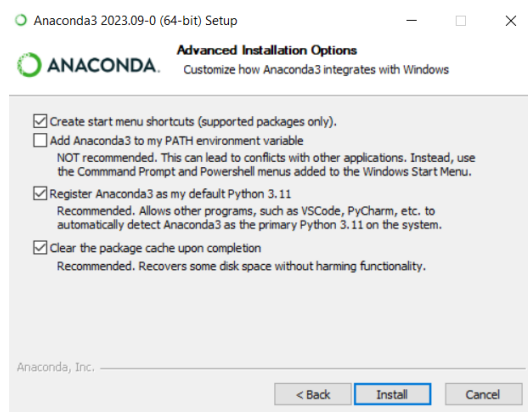
Podczas instalacji pozostaw zaznaczoną opcję „Just me” (domyślna, rekomendowana).



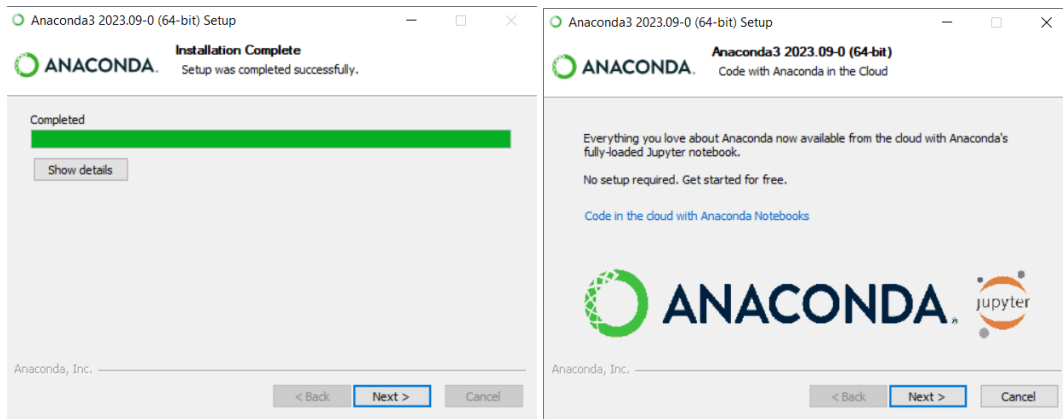
Pozostawiamy również domyślną ścieżkę instalacji (może różnić się nazwą użytkownika).



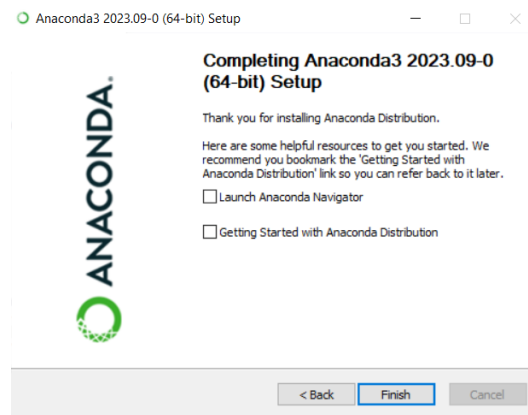
Zaznaczamy rekomendowane opcje. Pamiętajmy dodać program do menu start.



Proces instalacji może potrwać długo, zostaw go na kilka do kilkunastu minut. Gdy zakończy się, aktywny będzie przycisk „Next”. Możemy z niego skorzystać, aby przejść dwa ekrany instalatora.



Na ostatnim ekranie odznaczamy dwie opcje uruchamiające dodatkowe programy.



1.2 System Linux

Istnieje wiele dystrybucji Linuxa. Niektóre posiadają gotowe pakiety do instalacji wszystkich potrzebnych elementów, inne nie. W pierwszej kolejności sprawdź, jak instalować pakiety w Twojej dystrybucji Linuxa. Następnie upewnij się:

1. Czy masz już zainstalowane potrzebne biblioteki i narzędzia? Wiele dystrybucji Linuxa od razu ma zainstalowanego Pythona, ale niektóre w innej wersji;
2. Czy masz możliwość instalacji bibliotek przez Swój system (na przykład komendą `apt-get`)? Jeśli tak, jest to preferowana forma instalacji. Biblioteki mogą się nazywać na przykład tak jak `python-matplotlib` lub `python3-matplotlib`;
3. Jeśli pozostałe opcje zawiodą, zainstaluj biblioteki przez komendę `pip3` lub `pip`, pisząc przykładowo `pip install matplotlib`.

Upewnij się, że masz Pythona przynajmniej w wersji 3.9 oraz biblioteki:

- `numpy`,
- `scipy`,
- `matplotlib`,
- `sympy`,

1.3 System Mac OS X

Na systemie Mac OS X konieczne jest zainstalowanie Homebrew https://brew.sh/index_pl. Instalacja Homebrew polega na wpisaniu w terminalu komendy

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

(komendę tę można też skopiować ze strony https://brew.sh/index_pl). Po zainstalowaniu Homebrew należy zainstalować kilka bibliotek, z których będziemy korzystać:

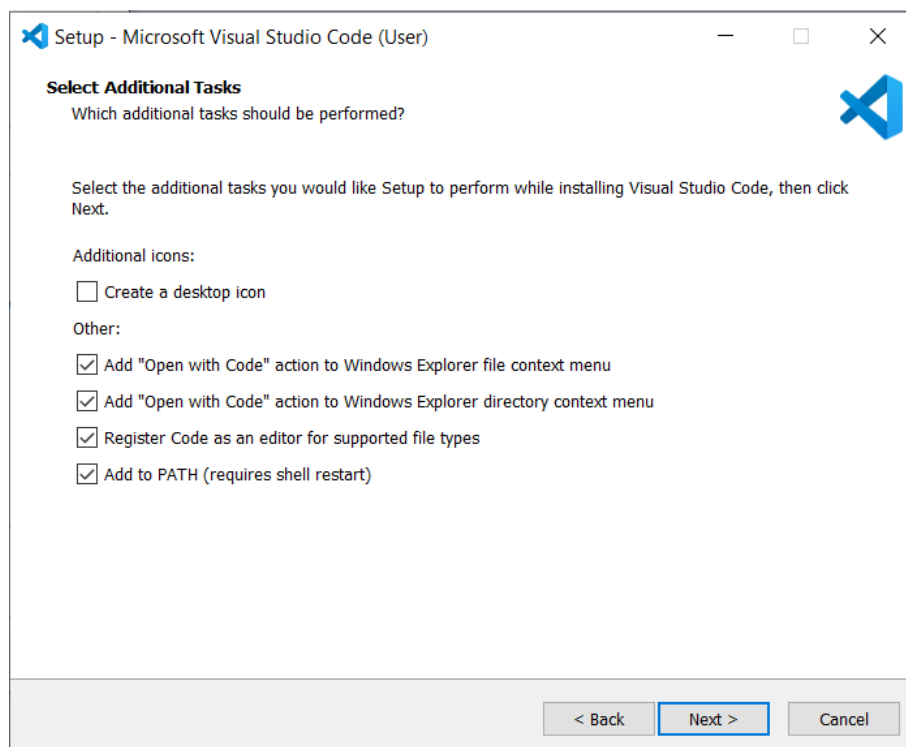
```
brew install scipy  
pip install matplotlib  
pip install sympy
```

2 VS Code i git

Do laboratoriów potrzebujemy dwóch narzędzi, Visual Studio Code oraz git.

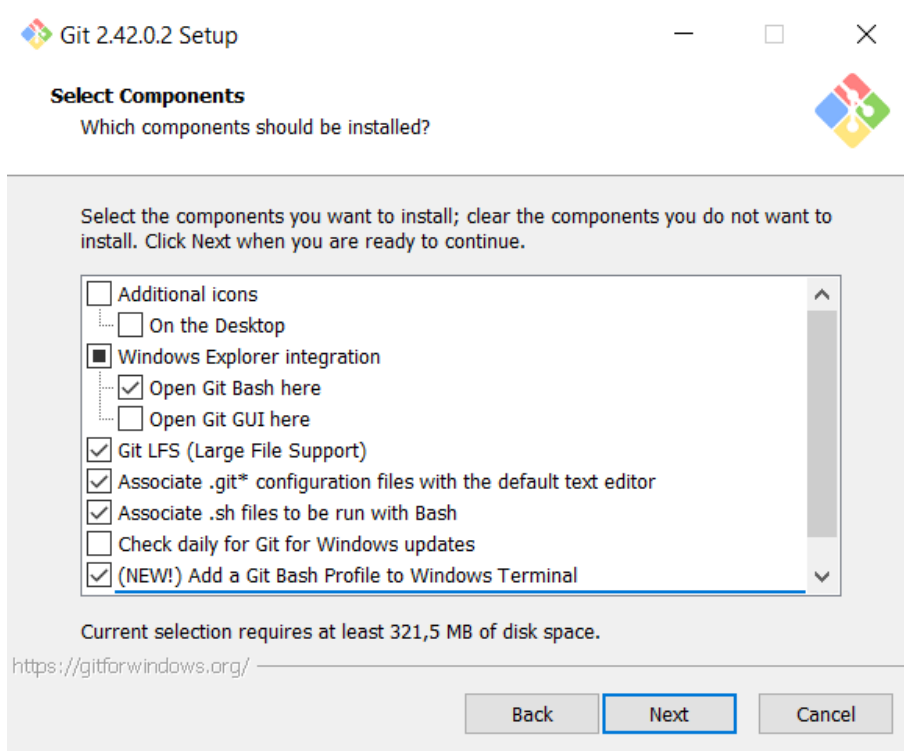
2.1 Windows

Na systemie Windows zaczynamy od instalacji Visual Studio Code. Wchodzimy na stronę <https://code.visualstudio.com/> i pobieramy instalator. Podczas instalacji upewniamy się, że edytor jest dodany do ścieżki oraz jest zintegrowany z eksploratorem plików.

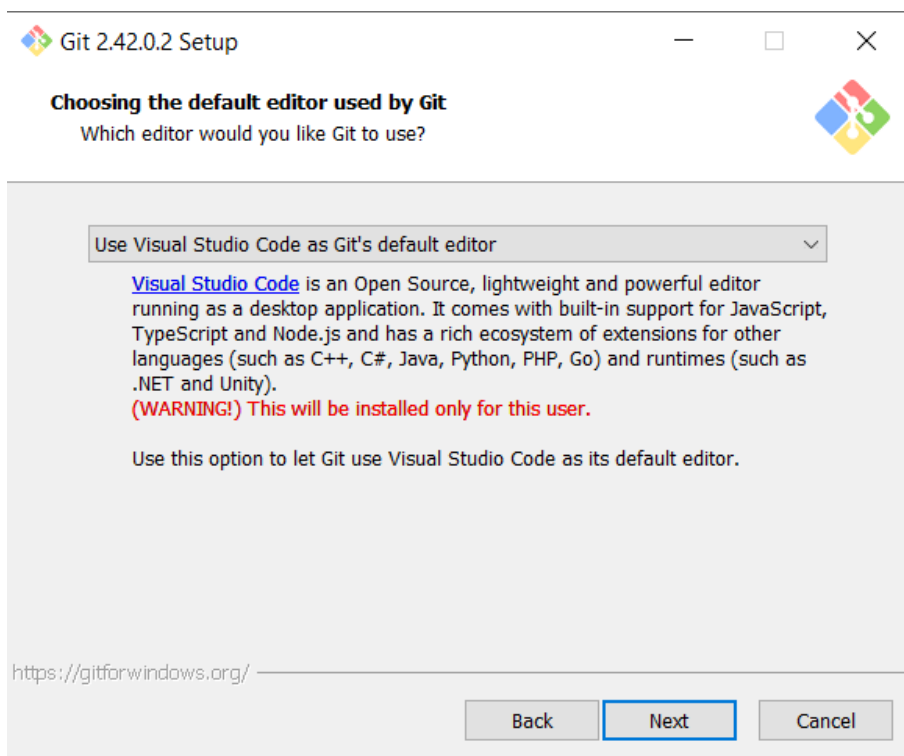


Po zainstalowaniu Visual Studio Code wchodzimy na stronę <https://gitforwindows.org/> i pobieramy instalator. Podczas instalacji musimy zwrócić uwagę na kilka kroków (w pozostałych, pozostawiamy wartości domyślne).

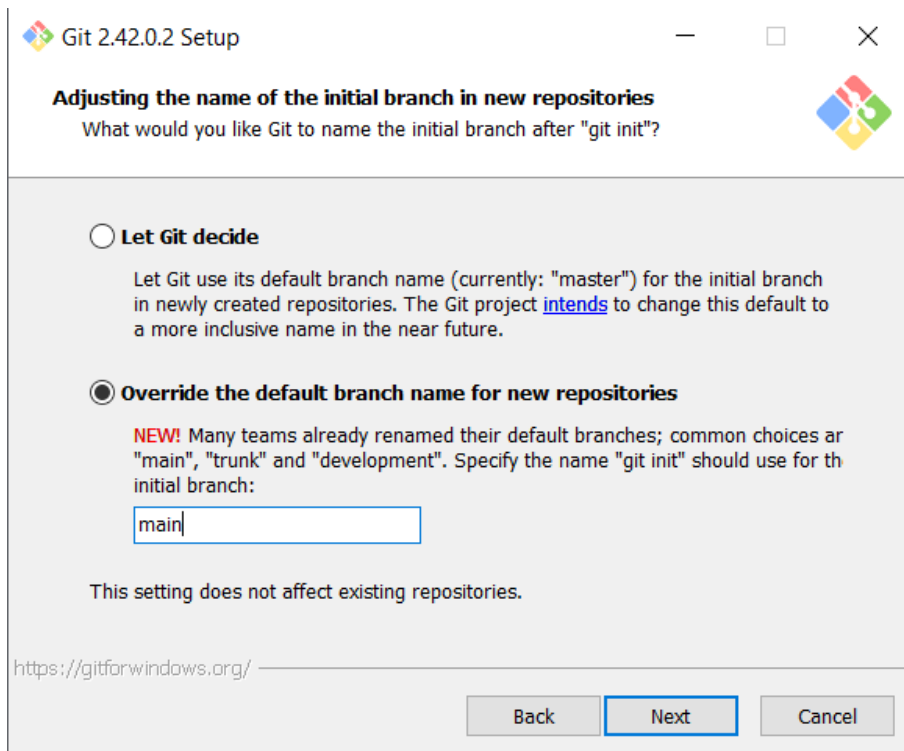
Odznaczamy opcję „Git GUI here” (będziemy korzystać z linii komend) oraz dodajemy profil Git Bash do aplikacji Windows Terminal.



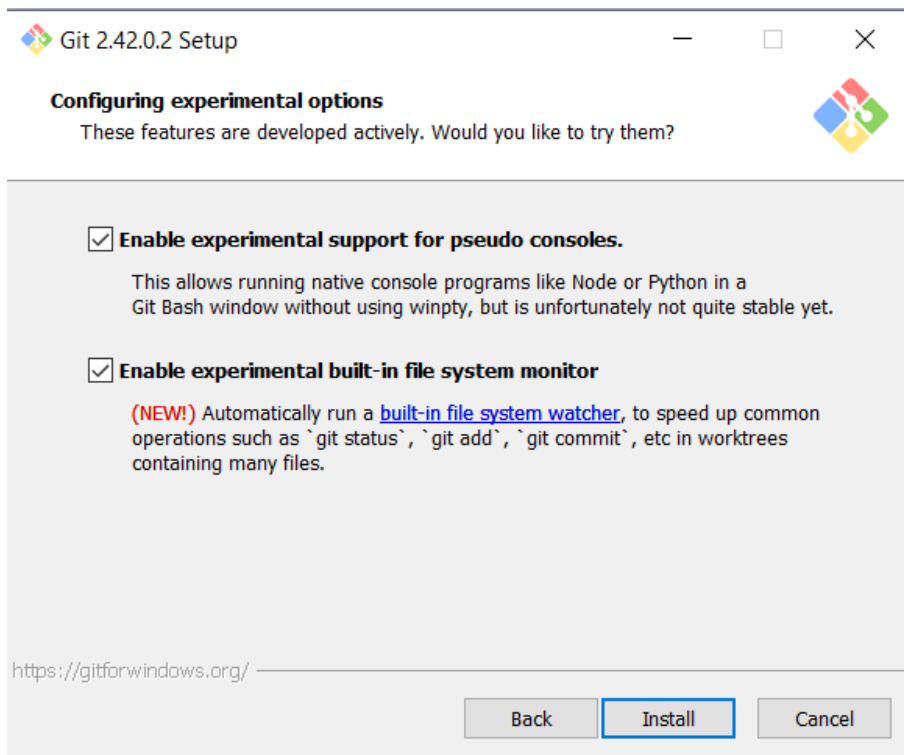
Jako edytor wybieramy Visual Studio Code (do tego potrzebowaliśmy zainstalować go w poprzednim kroku).



Ustawiamy nazwę głównego brancha dla nowych repozytoriów na main.



Włączamy eksperymentalne funkcjonalności, w szczególności obsługę pseudo-terminala.



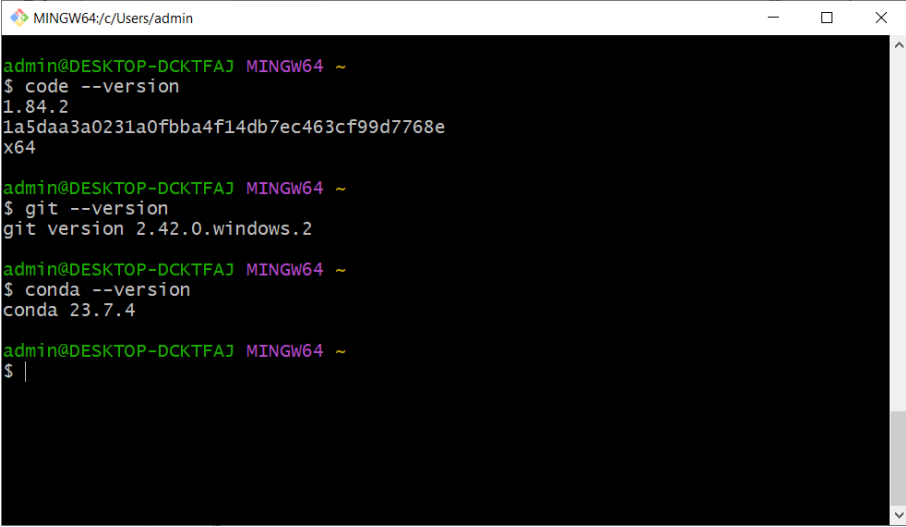
Pozostałe opcje pozostawiamy na wartościach domyślnych.

Teraz przechodzimy (przeklikując się pomiędzy katalogami, w trybie graficznym) do katalogu, w którym zainstalowaliśmy Anacondę na początku semestru, najprawdopodobniej będzie to C:\Users\użytkownik\Anaconda3. Wchodzimy do katalogu etc i dalej do katalogu profile.d. W środku powinien być plik conda.sh. Klikamy prawy przycisk myszy w wolnym

miejscu (nie na ikonke pliku) i z menu wybieramy Git BASH Here, aby otworzyć terminal z katalogiem ustawionym na odpowiednie miejsce. W terminalu wpisujemy komendę, która powie powłóce, gdzie szukać komendy conda (podczas instalacji nie dodawaliśmy Anacondy do zmiennej PATH, jeśli ktoś to zrobił, nie musi wykonywać tego kroku)

```
echo ". '${PWD}"/conda.sh" >> ~/.bashrc
```

Zamykamy terminal. Następnie klikamy prawy klawisz w wolnym miejscu na przykład na pulpicie i wybieramy Git BASH Here. W terminale po wpisaniu poniższych komend, nie powinniśmy widzieć żadnych błędów.



```
MINGW64/c/Users/admin
admin@DESKTOP-DCKTFAJ MINGW64 ~
$ code --version
1.84.2
1a5daa3a0231a0fbba4f14db7ec463cf99d7768e
x64

admin@DESKTOP-DCKTFAJ MINGW64 ~
$ git --version
git version 2.42.0.windows.2

admin@DESKTOP-DCKTFAJ MINGW64 ~
$ conda --version
conda 23.7.4

admin@DESKTOP-DCKTFAJ MINGW64 ~
$ |
```

2.2 MacOS

W systemie MacOS, podczas pierwszego tygodnia zainstalowaliśmy Homebrew. Możemy użyć tego samego narzędzia, aby zainstalować potrzebne narzędzia.

```
brew install --cask visual-studio-code
brew install git
```

Sprawdź, czy `code --version` wpisany w terminalu zgłasza numer wersji. Jeśli system nie może znaleźć komendy, uruchom Visual Studio Code, wciśnij `Cmd+Shift+P`, aby uruchomić paletę komend i zacznij wpisywać „Shell Command: Install code in ”PATH””. Kliknij opcję, która zostanie znaleziona. Po tym kroku, `code` powinien działać z terminala. Następnie wywołaj komendę

```
git config --global core.editor "code --wait"
```

2.3 Linux

W systemie Linux `git` powinien być dostępny bez dodatkowych kroków instalacji. Jeśli nie jest dostępny, zainstaluj zarazem `git`, jak i Visual Studio Code, za pomocą menadżera pakietów dla swojej dystrybucji Linuxa. Obie komendy `git` i `code` powinny być widoczne z terminala bez dodatkowych kroków. Wykonaj komendę

```
git config --global core.editor "code --wait"
```

aby ustawić domyślny edytor dla gita na Visual Studio Code.