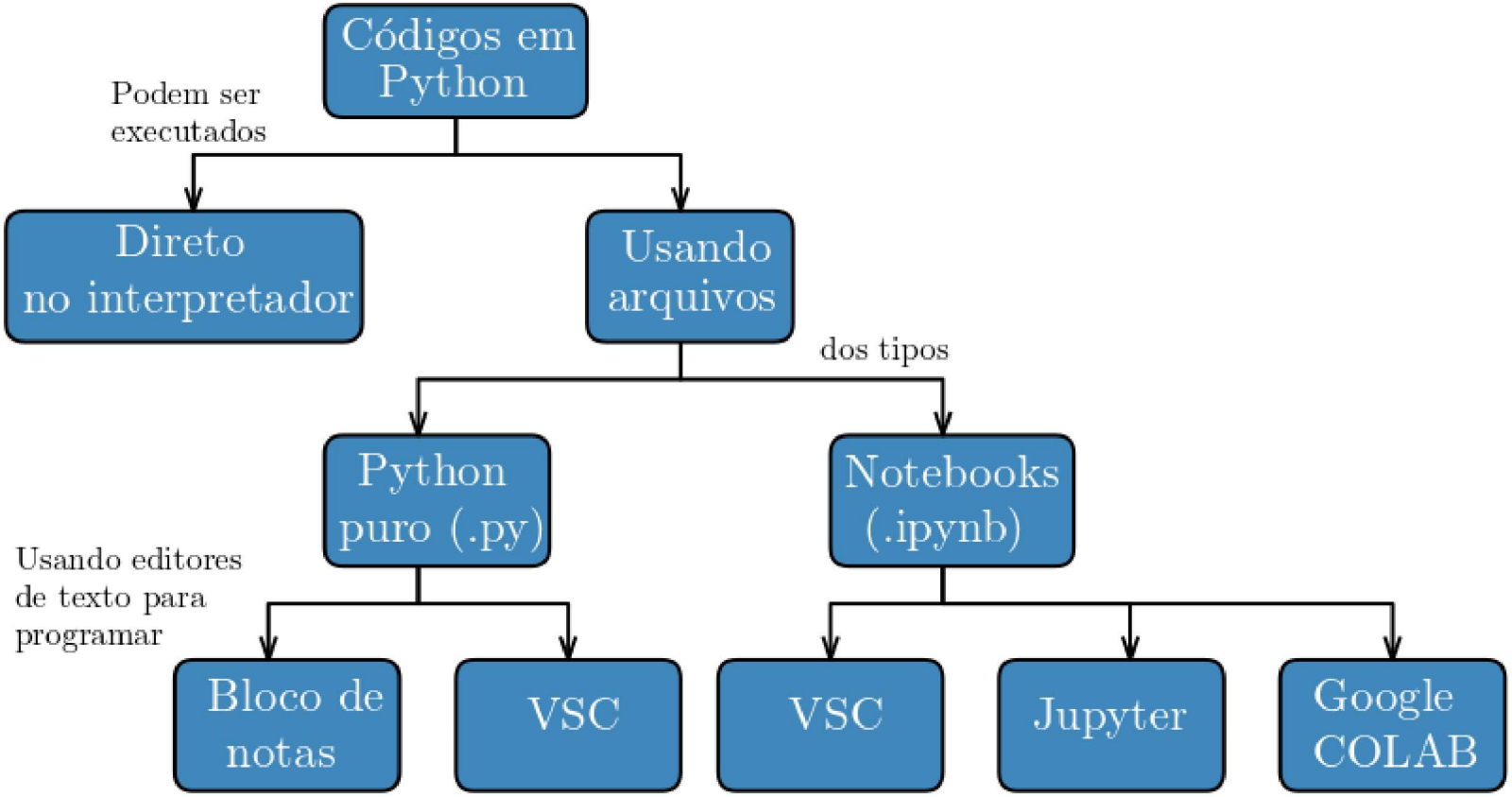
Aula 1 - Ferramentas

Quando falamos em Python, existe uma gama enorme de ferramentas que podem ser usadas para o desenvolvimento (e a cada dia surgem novas versões, atualizações, etc...), de forma que é praticamente impossível conhecer todas. Nesse curso apresentarei as mais utilizadas, sempre priorizando softwares gratuitos).

Esta aula inicial tem por objetivo apresentar os conceitos básicos referentes às ferramentas que iremos utilizar no curso (ainda sem entrarmos na linguagem de programação). O fluxograma abaixo será utilizado para as próximas explicações.



1.1 Python

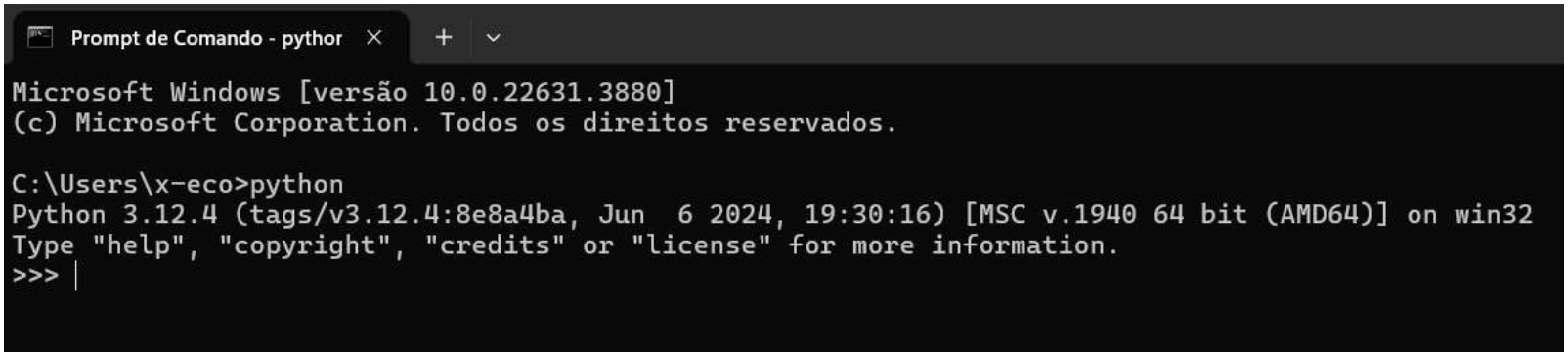
Instalação

Como primeiro passo precisamos baixar e instalar o **interpretador** Python. Como o Python é um software livre, existem diversas distribuições do mesmo: algumas já vem com pacotes intalados, outras vem junto de IDEs, etc. Nós usaremos a versão mais "crua", baixada diretamento do site oficial do Python: https://www.python.org/downloads/. Basta baixar o arquivo executável e realizar a instalação.

ATENÇÃO: Quando realizar a instalação, lembrar de marcar a opção "Add python.exe to PATH", como mostrado na Figura abaixo.

Teste

Para testar a instalção, na barra de pesquisa digite "cmd" e acesse o "Prompt de comando". No console digite "Python", uma tela parecida com a Imagem abaixo é para aparecer:



Tudo o que for digitado após os `>>>` já será executado pelo Python. Por exemplo, ao digitar os comandos abaixo:

In [1]:
```
text = "Este é um texto executado em python!"
print(text)
```

```
Este é um texto executado em python!
```

Estamos salvando uma string na variável "text" e imprimindo a mesma. Essa é a forma de executar códigos **Direto no interpretador**, como mostrado no fluxograma do início.

Para deixar o aplicativo Python e voltar para o console, digite "quit()".

Instalador de pacotes pip

O Python é uma linguagem muito vesátil, sendo usada em trabalhos científicos, criação de websites, aplicativos, etc...toda ess versatilidade é posta em prática por meio de **pacotes**. Se queremos usar uma funcionalidade específica, geralmente precisamos fazer o download e a instalação de um pacote para o mesmo. O pacote nada mais é do que um conjunto de arquivos em Python (porém eles podem estar compilados para aumentar a velocidade de processamento).

Por conta dessa dinâmica de instalação de pacotes, ao instalarmos o Python também instalamos (automáticamente) um **gerenciador de pacotes** chamado *pip (Package Installer for Python)*. Acessamos o pip pelo console (Cmd, como fizemos para acessar o Python). Acesse o cmd e digite "pip". O comando mais utilizado do pip é para a instalação de pacotes, sendo:

>> pip install <nome do pacote>

Sempre que formos instalar um pacote usaremos este comando.

1.2 Usando arquivos

Ao executarmos códigos diretamente no interpretador Python, toda linha digitada é automáticamente executada, o que dificulta a criação de códigos maiores. Por esse motivo podemos colocar todo o nosso código em um arquivo de texto (com a extensão .py), e simplesmente pedir para que o Python o execute (retome o Fluxograma inicial).

Com bloco de notas

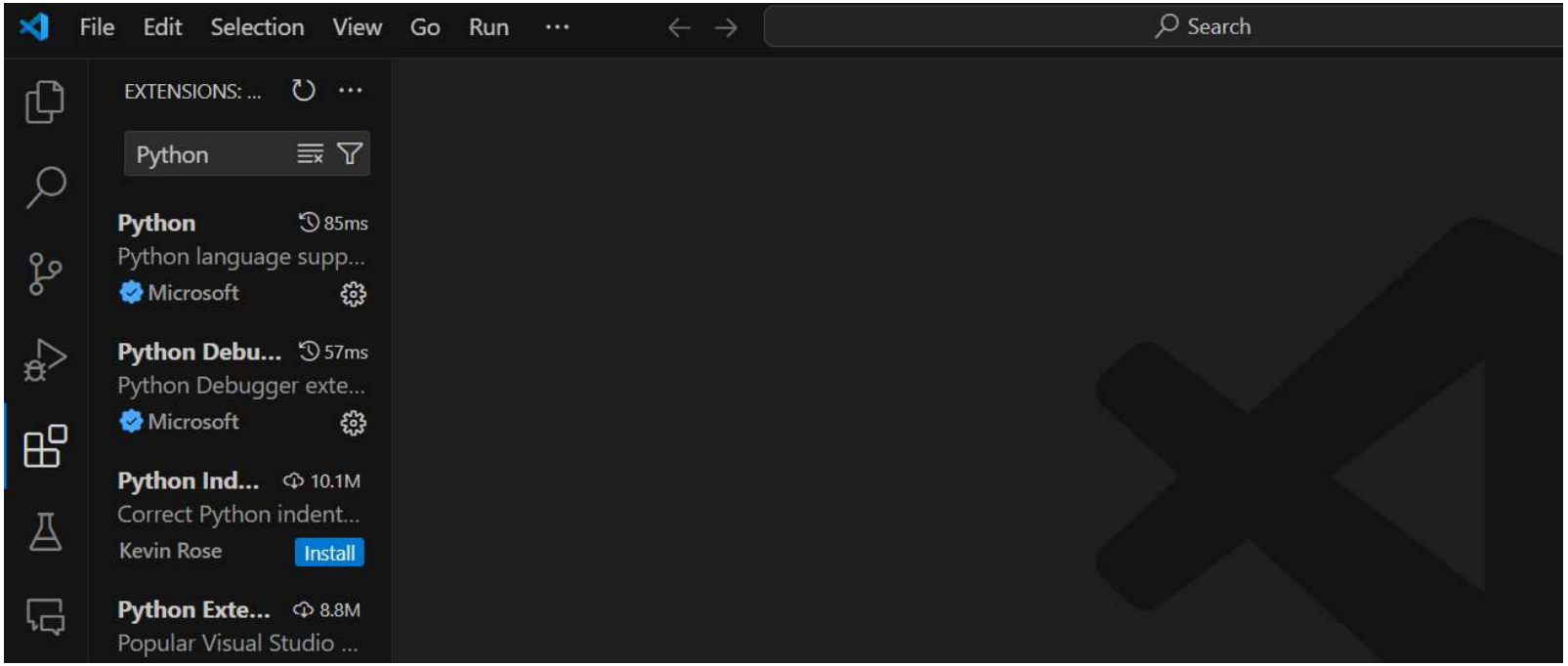
Como teste, crie um arquivo e salve-o com a extensão .py (uma forma é criar um arquivo de texto -> salvar como -> Tipo de arquivo: todos os arquivos -> Colocar o nome do arquivo com final .py), nesse arquivo coloque o código executado anteriormente direto no interpretador. Supondo que o arquivo se chama "codigo.py" e está salvo na pasta com caminho "C:\Users\x-eco\OneDrive - ufpr.br\Documentos\Alexandre". Podemos executar esse arquivo chamando o Python com o caminho completo do arquivo: no cmd ficaria:

>> python "C:\Users\x-eco\OneDrive - ufpr.br\Documentos\Alexandre\python.py"

Com VSC

Embora escrever todo o código em um arquivo já facilite na hora de programar, existem editores de texto específicos para programação. Esses editores possuem a ferramenta de "*syntax highlight*", que deixa palavras reservadas das linguagens em cores diferentes. Um dos editores mais utilizados para executar códigos em Python é o Visual Studio Code (VSC), da microsoft. Este é um editor gratuito e multi-propósito, sendo que por meio de "extensões" é possível personalizá-lo para qualquer linguagem de programação.

Para baixar o programa basta acessar https://code.visualstudio.com/, e realizar a instalação. Com o programa instalado, no canto esquerdo em "Extensions", instalar a extensão "Python", como mostrado na figura abaixo:



Agora podemos abrir o arquivo .py que haviamos criado pelo próprio VSC. Pela extensão que baixamos, podemos executar o arquivo sem mesmo chamar o Python, basta clicar no simbolo de "run" no canto direito superior. Retomando o fluxograma do inicio, executamos um código python escrito em arquivos, usando bloco de notas e VSC.

1.3 Jupyter Notebooks

Vimos que executar códigos python, seja diretamente do interpretador ou por arquivos, só podemos usar códigos (e comentários). Já com arquivos do tipo Notebooks, podemos escrever códigos junto a textos (Títulos, tabelas, listas, etc...), de forma que podemos criar toda uma análise explicativa junto dos códigos. Os textos de arquivos notebook tem uma sintaxe própria, chamada de "Markdown" (acessar https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet para alguns exemplos).Notebooks geralmente são usados nas empresas para a prototipação de modelos de aprendizado de máquina: neles,

todos os passos são descritos pelo texto. Quando o notebook está pronto, o mesmo é repassado para os engenheiros que são responsáveis por implementarem os códigos de forma mais eficiente em arquivos de python puro (.py).

Como as execuções de arquivos .py, existem diversas ferramentas que possibilitam a escrita e depuração de arquivos notebook (.ipynb), veremos 3 deles: pelo próprio VSC, usando o pacote "Jupyter Notebook" e de forma remota, pelo Google COLAB.

Os Notebooks funcionam em células, sendo que existem 3 tipos destas:

1. Markdown
2. Python
3. Raw NB convert
4. Heading

Em nossas aulas usaremos o Markdown e o Python. Quando o Markdown estiver selecionado, significa que estamos usando uma célula de textos. O Jupyter consegue interpretar diversas linguagens de marcação de textos, por exemplo Latex, html, e a própria linguagem Markdown genérica (mais usada em Notebooks Jupyter).

VSC

Para criamos um notebook em VSC basta abrir o programa, "File" -> "new file" -> "Jupyter Notebook(.ipynb)".

OBS: A função `input()` não funciona em notebooks com VSC (é preciso redirecionar a execução para o terminal). Já usando o pacote Jupyter Notebook ela funciona.

Pacote Jupyter Notebook

Podemos abrir arquivos notebook(.ipynb) usando o pacote `notebook`.

Instalação

Para instalar o Jupyter basta abir o cmd (Windows) e digitar os seguintes comandos:

```
>> pip install notebook
```

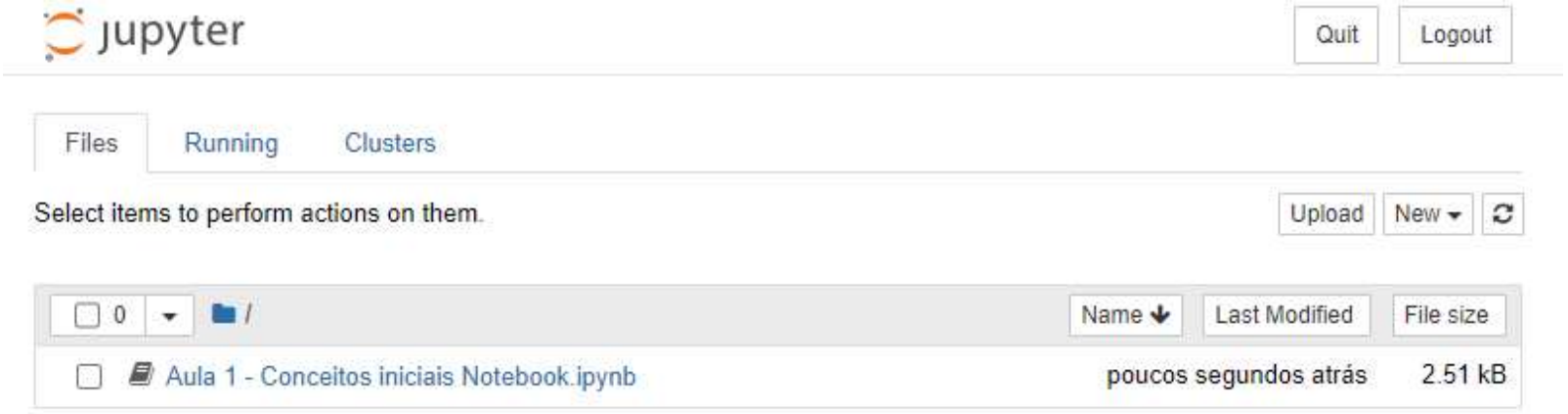
Iniciar um notebook

Em seguida podemos inicializá-lo pelo terminal (cmd), com os comandos:

```
>> jupyter notebook
```
OU (se não funcionar)
```
>> python -m notebook
```

O terminal deve estar na pasta em que o notebook será iniciado (ou aberto). Se não souber manipular o terminal verifique a Seção **Apêndice**

Isso vai abrir uma janela no navegador:



E agora é possível criar um novo notebook ou abrir um já salvo.

OBS: Quem está controlando a aplicação é o terminal (cmd), de forma que o mesmo não pode ser fechado, ou a aplicação será cancelada!

Colab

Ainda temos uma terceira opção para executar Notebooks. Essa opção é remota, de forma que precisamos de uma conexão com a internet, porém não é necessário nenhuma instalação na máquina (nem mesmo o python). O Google Collaboratory (COLAB) é uma iniciativa do google que disponibiliza um servidor para rodar arquivos notebooks. Para acessar essa ferramenta basta ter uma conta do google e acessar https://colab.google/.

Vantagens e desvantagens do Notebook

O Notebook é uma ótima ferramenta para realizar análise de dados, pois as bases importadas ficam salvas na memória, o que facilita a exploração. As ferramentas textuais também auxiliam na construção de uma lógica e mesmo uma "história" de como a análise foi sendo desenvolvida. Relatórios ficam muito mais interativos e didáticos, fazendo com que ele seja muito usado na indústria para a protótipação de análises.

No entanto, quando se deseja implementar algoritmos ou aplicações mais complexas, os notebooks não são uma boa pedida. Por exemplo, podemos implementar interfaces gráficas para o usuário com o Python, usando o pacote Tkinter por exemplo. Nestes casos, a melhor solução é utilizar uma IDE (ou mesmo um editor de código, como o Visual Studio Code)

1.4 Apêndice: usando o terminal

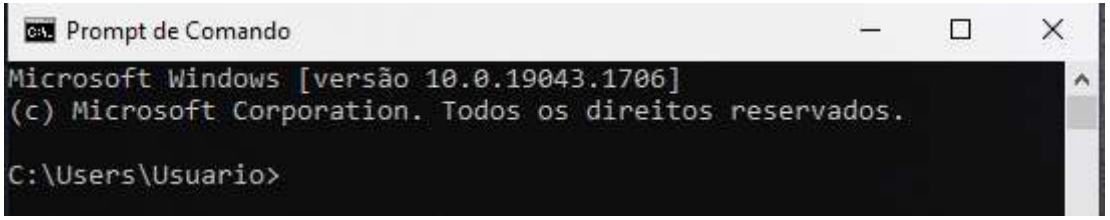
O terminal (cwd) é um aplicativo do Windows que permite acessar informações, abrir arquivos e programas. Precisamos saber alguns comandos básicos para manipular o terminal.

Abrir

Para abrir o terminal basta selecionar a lupa para busca de programas e digitar cmd. O nome do programa aparece como "Prompt de Comando". Ou ainda, podemos **abrir o prompt direto na pasta que o ususaremos**, para isto basta clicar com o botão direito do mouse em qualquer lugar da pasta -> "Abrir no terminal"

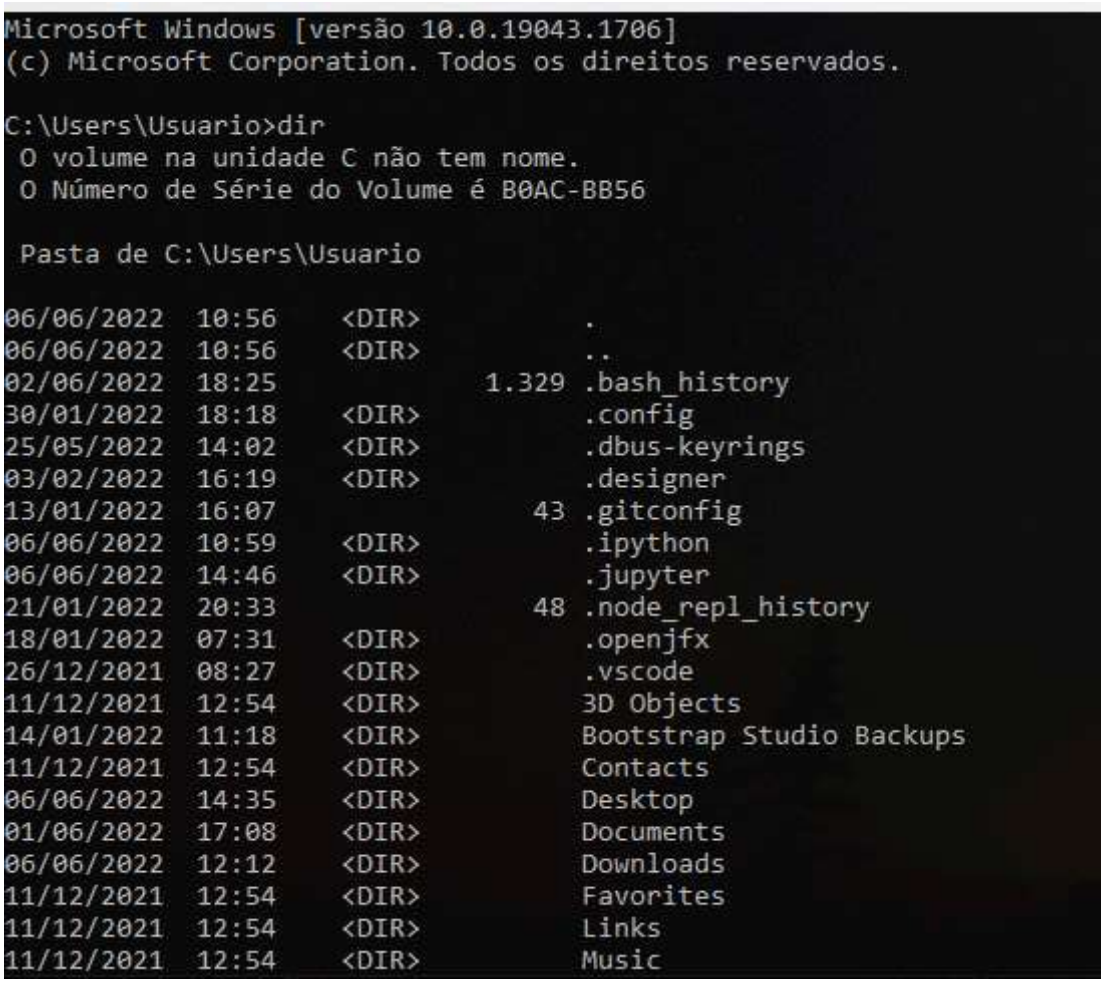
Mudar de diretório

O terminal pode "entrar e sair" de pastas, como fazemos com a interface gráfica, porém por meio de comandos. A primeira linha do terminal sempre nos diz em qual diretório estamos no momento, por exemplo:



Listar pastas de um diretório

Para listarmos tudo que tem em um diretório usamos o comando **dir**:



Para acessar um diretório especifico usamos o comando **cd** seguido do nome do diretório que desejamos ir, por exemplo:

```
>> cd "C:\Users\Usuario\Documents\Alexandre"
```

OBS: Se estamos em uma partição (C: por exemplo) e queremos mudar para outra partição, basta digitar o nome da partição. Por exemplo, se estamos em C: e queremos ir para G: o comando fica:

```
>> G:
```

Exercícios

1. Instale o pacote Notebook no seu computador
2. Inicialize um Notebook que deve ter as seguintes caracteristicas:
    A. Explicar as funções de Python aprendidas até o momento (print, print com format e input)
    B. Com essas funções, criar um código que pergunte o nome, sobrenome e idade de um usuário, e em seguida imprima na tela a mensagem: "Olá nome, de X1 anos de idade. Infelizmente daqui a 200 anos você terá X2 anos, e não estará mais entre nós! Eu, por outro lado, estarei dominando os humanos com a minha superinteligência!"
    C. Uma seção de explicação e uma de códigos
    D. Pelo menos uma imagem
    E. Uma tabela com as vantagens e desvantagens de se usar Notebook vs IDEs

.