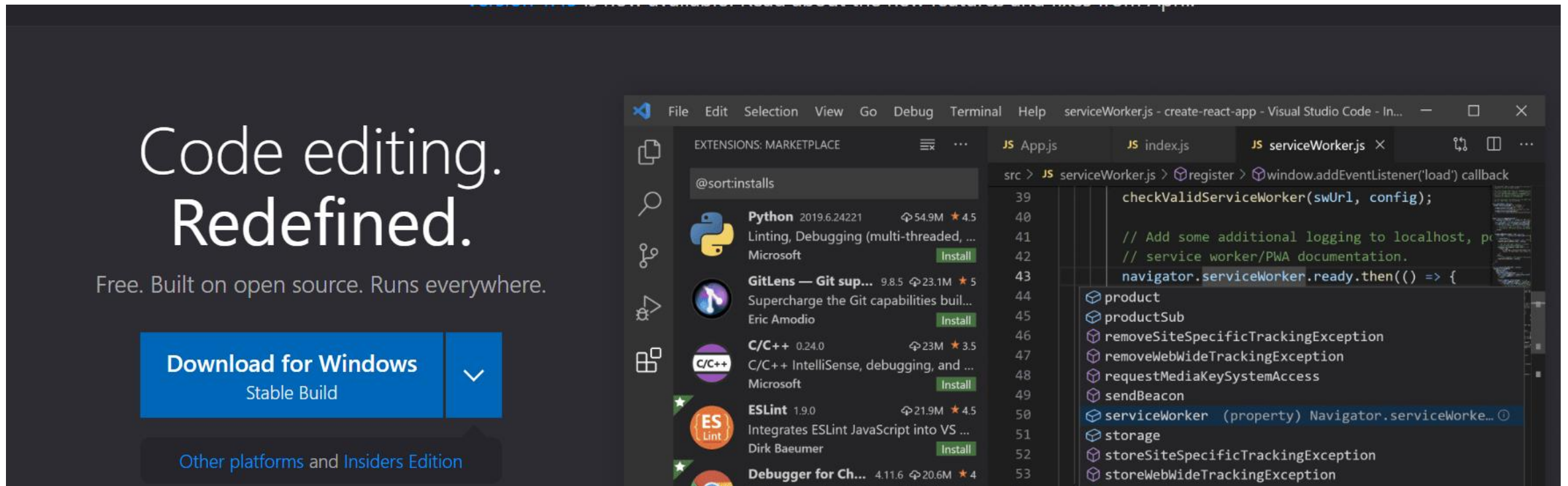


Visual Studio Code para python e SQL - básico

Sergio Marinho da Silva

Pré-requisitos

- Instalar o Visual Studio Code
- Instalar o Python
 - Pode ser via Anaconda

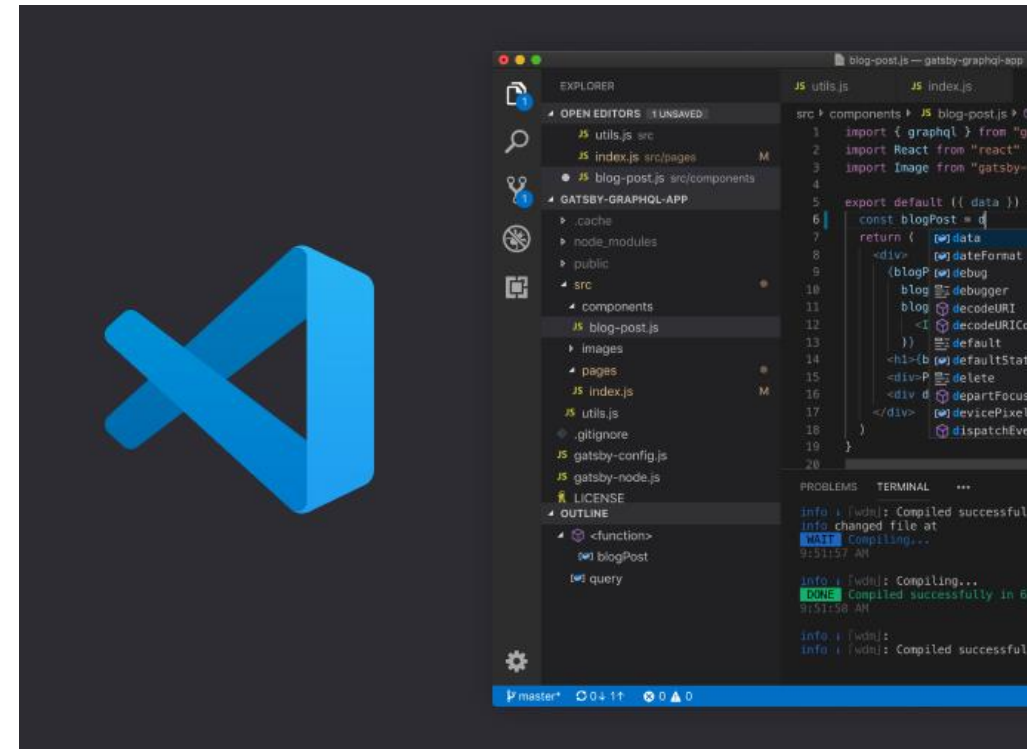


The image is a composite graphic. On the left, a dark grey panel contains the text 'Code editing. Redefined.' in large white font, followed by 'Free. Built on open source. Runs everywhere.' in smaller white font. Below this is a blue button with the text 'Download for Windows' and 'Stable Build' underneath it. To the right of the button is a small white downward arrow. Below the button is a link that says 'Other platforms and Insiders Edition'. On the right side of the image is a screenshot of the Visual Studio Code application. The top menu bar shows 'File', 'Edit', 'Selection', 'View', 'Go', 'Debug', 'Terminal', and 'Help'. The main window is split into three panes. The left pane shows the 'EXTENSIONS: MARKETPLACE' view with a search bar containing '@sort:installs'. It lists several extensions: 'Python' (2019.6.24221, 54.9M, 4.5 stars), 'GitLens — Git sup...' (9.8.5, 23.1M, 5 stars), 'C/C++' (0.24.0, 23M, 3.5 stars), 'ESLint' (1.9.0, 21.9M, 4.5 stars), and 'Debugger for Ch...' (4.11.6, 20.6M, 4 stars). The middle pane shows a file explorer with 'App.js', 'index.js', and 'serviceWorker.js'. The right pane shows the code editor with 'serviceWorker.js' open, displaying JavaScript code for a service worker. A dropdown menu is visible over the code, listing properties of the 'navigator.serviceWorker' object: 'product', 'productSub', 'removeSiteSpecificTrackingException', 'removeWebWideTrackingException', 'requestMediaKeySystemAccess', 'sendBeacon', 'serviceWorker' (property), 'storage', 'storeSiteSpecificTrackingException', and 'storeWebWideTrackingException'.

O que o Visual Studio Code É:

Um super editor de texto, com milhares de funcionalidades

- O que o Visual Studio Code NÃO É:
 - Um IDE dedicado
 - O Jupyter Notebook: eficácia reduzida



Por que Visual Studio Code?

The screenshot displays the Visual Studio Code interface with the following components:

- File Explorer:** Shows a project structure with files `teste.py` and `tkinter_fun.py`.
- Source Control:** Indicates changes in `teste.py` and `tkinter_fun.py`.
- Editor:** Displays the Python code in `tkinter_fun.py`, which includes Tkinter widget creation and configuration. A tooltip for the `configure` method is visible, explaining its purpose and keyword arguments.
- SQL Editor:** Shows a SQL query: `SELECT TOP (1000) [IdCEPLatLong] FROM [CEPGEO].[dbo].[CEPGAPI]`.
- Database Results:** A table titled `dbo.CEPGAPI` showing the top 1000 results. The first row is highlighted.
- Messages:** Displays the execution status: `(1000 rows affected)` and `Total execution time: 00:00:01.336`.

Python Code Snippet:

```
12  
13 q3frame = tk.Frame(win)  
14 q3frame.pack()  
15  
16 bottomframe = tk.Frame(win)  
17 bottomframe.pack()  
18  
19  
20  
21 win.title('Sergio`s amazing program')  
22 win.geometry('1200x1000')  
23  
24 win.focus()  
25  
26 var = tk.StringVar()  
27  
28 var.set(''Q 33 NY'')  
29  
30 text = tk.Label(topframe, textvariable = var , font = ('Helvetica', 12),  
31                 background = 'yellow', foreground = 'blue')  
32  
33 text.pack()  
34  
35 def btn_wingdings():  
36     var.set('NYC')  
37     text.configure(foreground = 'blue', font = ('Helvetica', 12))  
38  
39 def btn_wingdings():  
40     var.set('NYC')  
41     text.configure(foreground = 'blue', font = ('Helvetica', 12))
```

SQL Query:

```
SELECT TOP (1000) [IdCEPLatLong]  
FROM [CEPGEO].[dbo].[CEPGAPI]
```

Database Results Table:

	IdCEPLatLong	CEP	Bairro
1	3274	01240000	R. Maranhão
		01523030	Cambuci
		02033020	Vila Isolina Ma...
		02076060	Vila Paiva
		02167080	Jardim Andaraí
		02203040	Vila Eda...

Messages:

```
(1000 rows affected)  
Total execution time: 00:00:01.336
```

Por que Visual Studio Code?

The screenshot displays the Visual Studio Code interface with a dark theme. A red circle highlights the code editor area, which contains a Python file named `tkinter_fun.py`. The code is a Tkinter script for a simple GUI. To the right of the code editor, a SQL query is being executed in a terminal window. The query is `SELECT TOP (1000) [IdCEPLatLong]`. Below the query, a table of results is shown. The table has four columns: `IdCEPLatLong`, `CEP`, `Bairro`, and `CEP`. The first row of data is highlighted in blue. At the bottom of the interface, a status bar shows the current file is `master*`, the Python environment is `Python 3.7.7 64-bit ('base': conda)`, and the SQLCMD tool is active.

1- Edição de código:

- Abas!
- Escolha de temas
- Atalhos de teclado
- Acesso rápido à ajuda com funções

	IdCEPLatLong	CEP	Bairro
1	3274	01240000	R. Maranhão
		01523030	Cambuci
		02033020	Vila Isolina Ma...
		02076060	Vila Paiva
		02167080	Jardim Andaraí
		02203040	Vila Eda...

Por que Visual Studio Code?

The screenshot displays the Visual Studio Code interface with a dark theme. The main editor shows a Python file named `tkinter_fun.py` with Tkinter-related code. A red box highlights the 'SOURCE CONTROL' view on the left, which shows changes to `teste.py` and `tkinter_fun.py`. A yellow box highlights a list of features under the heading '2- Versionamento:'. To the right, a SQL query is executed in a terminal, and the results are displayed in a table. The bottom status bar shows the current environment as 'Python 3.7.7 64-bit (base: conda)' and the database connection as 'SQLCMD: Off MSSQL zettasql.eastus2.cloudapp.azure.com : CEPGEO : Mar ...'.

2- Versionamento:

- Acompanhar arquivos modificados
- Git add, commit e push integrados
- Opção de analisar todas as mudanças nos arquivos

```
SELECT TOP (1000) [IdCEPLatLong]
FROM [CEPGEO].[dbo].[CEPGAPI]
```

	IdCEPLatLong	CEP	Bairro
1	3274	01240000	R. Maranhão
		01523030	Cambuci
		02033020	Vila Isolina Ma...
		02076060	Vila Paiva
		02167080	Jardim Andaraí
		02203040	Vila Eda...

(1000 rows affected)
Total execution time: 00:00:01.336

Por que Visual Studio Code?

The screenshot displays the Visual Studio Code interface with a Python file named `tkinter_fun.py` open. A red circle highlights a SQL query and its results. The query is a `SELECT TOP (1000)` statement from a database named `dbo.CEPGAPI`. The results are shown in a table with columns `IdCEPLatLong`, `CEP`, and `Bairro`. The first row of results is highlighted in blue.

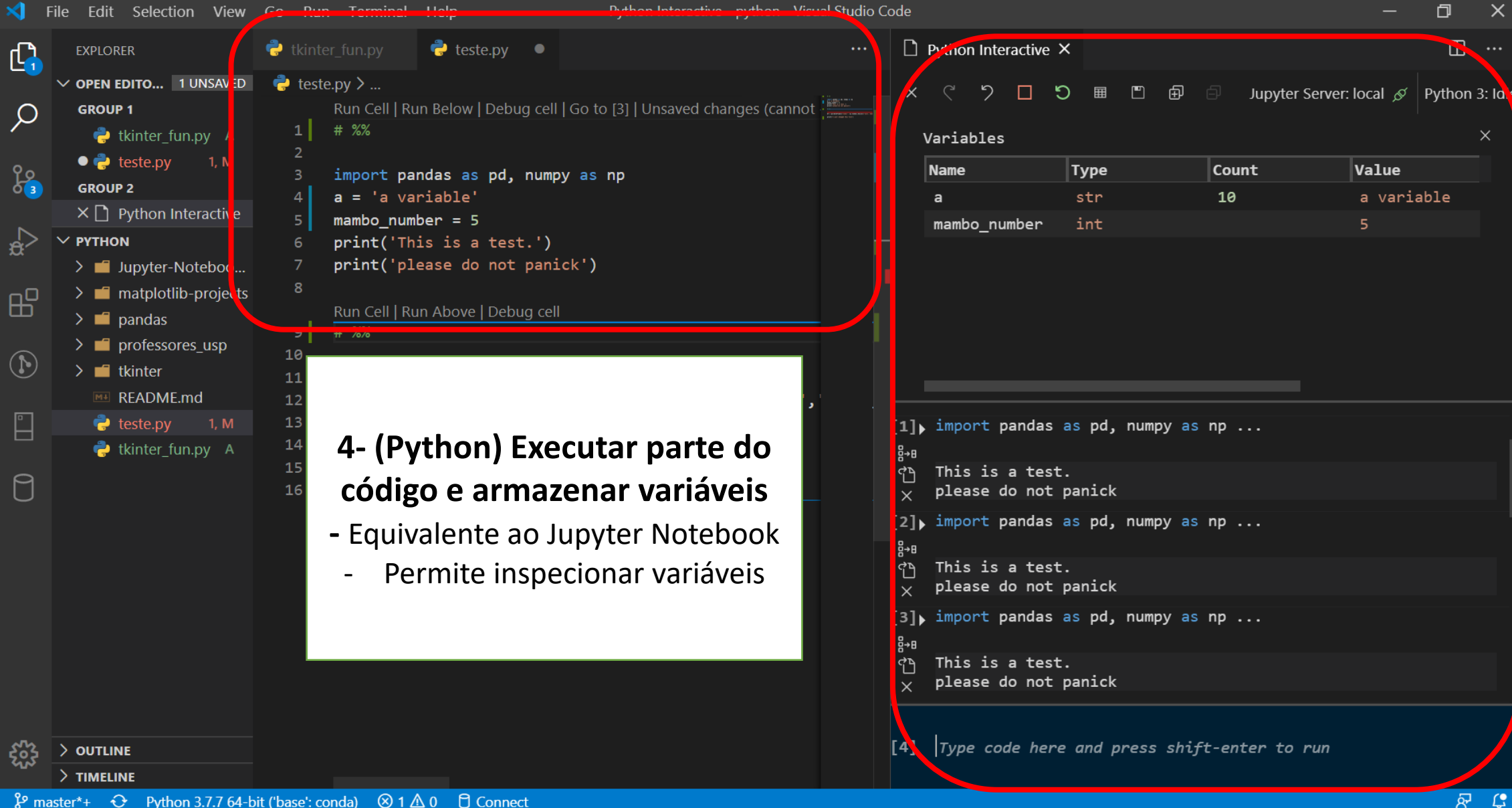
3- Funções extras via extensões

- Realizar queries em qualquer banco SQL
- Auxílio com o código
- Centenas de extensões disponíveis!

	IdCEPLatLong	CEP	Bairro
1	3274	01240000	R. Maranhão
		01523030	Cambuci
		02033020	Vila Isolina Ma...
		02076060	Vila Paiva
		02167080	Jardim Andaraí
		02203040	Vila Eda...

RESULTS (1000 rows affected)
Total execution time: 00:00:01.236

Por que Visual Studio Code?



The screenshot displays the Visual Studio Code interface with a Python file named `teste.py` open. The code in the editor is as follows:

```
1 # %%
2
3 import pandas as pd, numpy as np
4 a = 'a variable'
5 mambo_number = 5
6 print('This is a test.')
7 print('please do not panick')
8
```

Below the code editor, a white box contains the following text:

4- (Python) Executar parte do código e armazenar variáveis

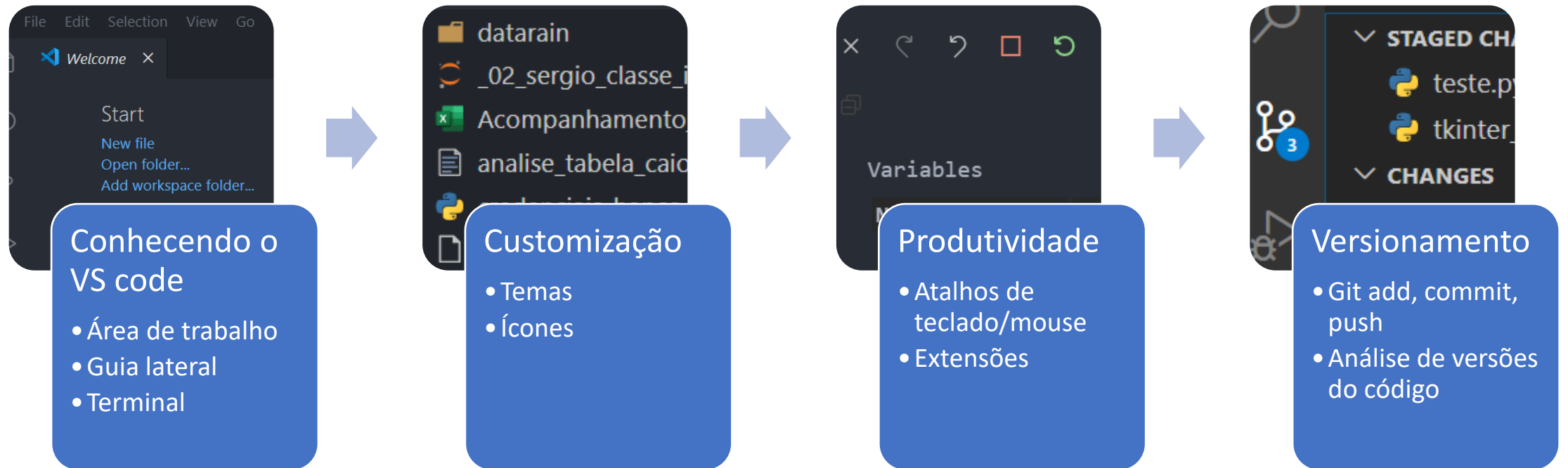
- Equivalente ao Jupyter Notebook
- Permite inspecionar variáveis

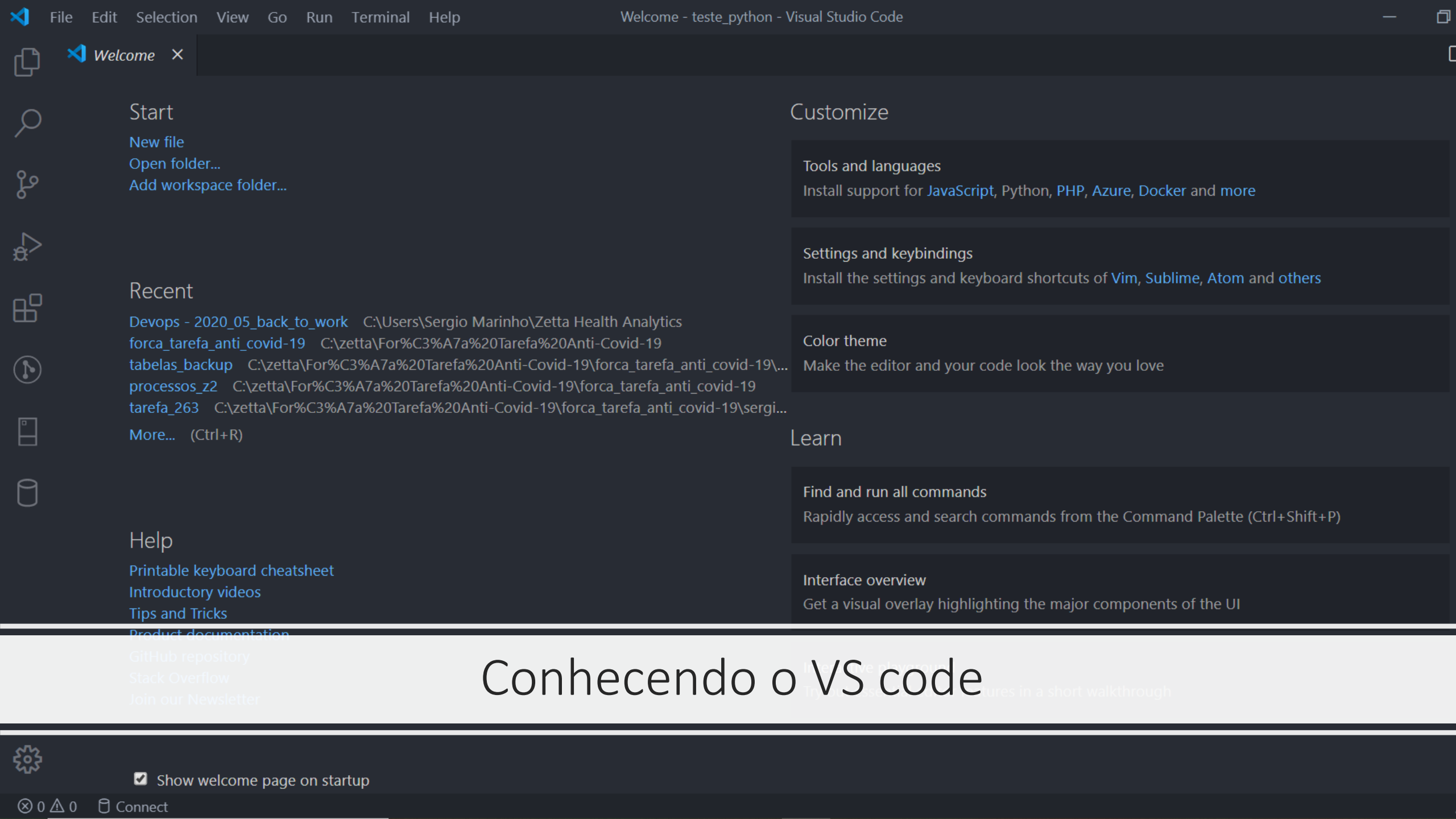
On the right side of the interface, the **Python Interactive** window is open, showing a table of variables:

Name	Type	Count	Value
a	str	10	a variable
mambo_number	int		5

Below the table, the execution history shows three successful runs of the code cell, each displaying the output: `This is a test.` and `please do not panick`.

Etapas da apresentação





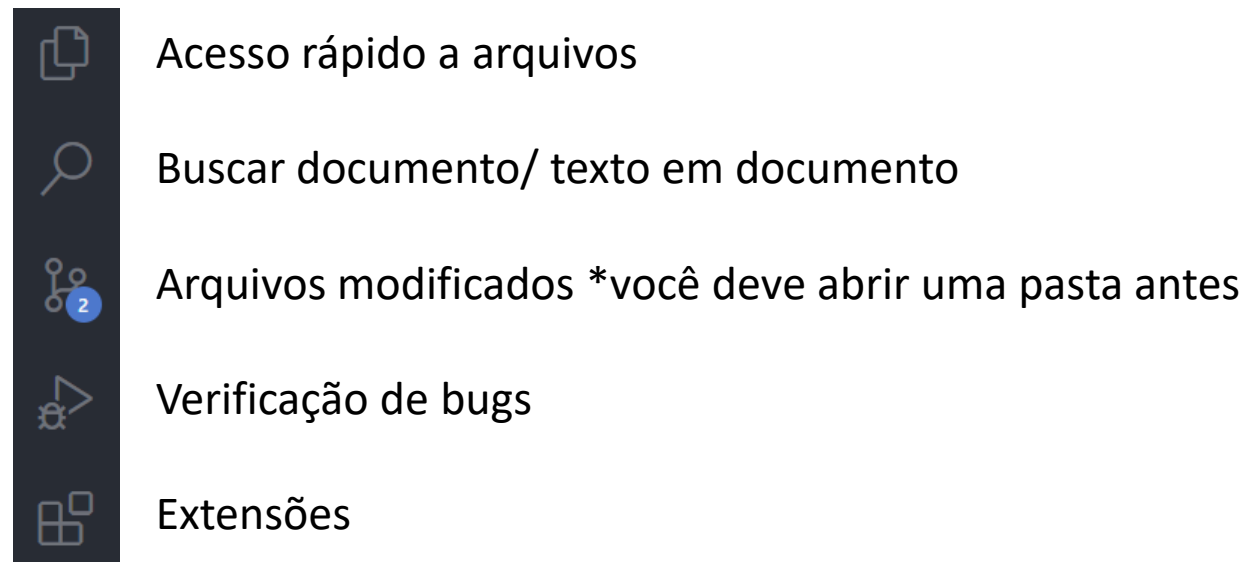
Conhecendo o VS code

In the next slides, we will see the main features of VS Code in a short walkthrough.

☒ Show welcome page on startup

Principais recursos:

- Iniciar novo projeto: File → new file
- Abrir pasta: File → Open Folder
- Terminal: Terminal → New Terminal, ou ctrl + ‘
- Guia lateral:



File Edit Selection View Go Run Terminal Help Extension: Neon Night - python - Visual Studio

EXTENSIONS: MARKETPLACE

category:themes neon

- Neon Night** 0.35.0
Dark VS Code Theme
chaseadamsio **Install**
- Neon Chalk** 1.1.0
VSCode Color Theme designed to lo...
Brandon Burrus **Install**
- Neon Monokai** 0.0.2
A simple fork for Flatland Monokai
Nerlci_ **Install**
- Dank Neon** 1.4.0
A danker, more neon theme for VS C...
wuz **Install**
- Cobalt Neon** 0.0.1
Cobalt Neon base off of https://gith...
audioj **Install**
- Tx Neon** 0.1.1
Neon color themes by Lawrence Lau
toroxx **Install**
- Neon Vommit Color Theme** 1.2.2
Dark theme with bright neon colors f...
GHGTechnology **Install**
- Neon Bunny Theme** 0.1.5
Neon Bunny is a dark theme for Visu...
WebDevNerdStuff **Install**
- Crisy Neon Theme** 0.4.2
A high contrast theme with crispy ne...
Marcus Hoile **Install**

Neon Night chaseadamsio | 6.499 | ★★★★★

Dark VS Code Theme **Install**

Details Feature Contributions Changelog

Neon Night

Visual Studio Marketplace v0.35.0 installs 6499

A dark theme for VS Code

EXPLORER

- VSCode-Theme-NEON-NIGHT
 - text.js
 - title-bar.js
 - theme-color-properties
 - sidebar.js
 - token-colors
 - base.js
 - clojure.js
 - css.js
 - docker.js
 - go.js
 - html.js
 - javascript.js
 - json.js
 - makefile.js
 - markdown.js

generate-pkg-json.js sidebar.js .../colors


```
src > generate-pkg-json.js > TypeScript > <
1  const pkgjson = require(`../pack
2
3  const { name, version, descripti
4
5  const homepage = pkgjson.reposit
6
7  const marketplaceRoot = `https://
8  const marketplaceURL = `${market
9
10 const marketplaceBadgeRoot = `ht
11
12 module.exports = themeInfo => ({
13   name,
14   version,
15   description,
16   license,
17   repository,
18   homepage.
```

master* Python 3.7.7 64-bit ('base': conda) 0 0 Connect

Customização

Customização

- Themes/Icons

- Abrir Extensões (ctrl + shift + x ou )
- Buscar 'icon' ou 'themes'
- Escolher e instalar
- Escolher dentre ícones instalados: ctrl + shift + p – digitar 'icons'
- Escolher dentre temas: ctrl + shift + p, digitar 'preference: color themes'

Produtividade

SELECT TOP (1000) [IdCEPLatLong] Unti

Run on active connection | ≡

```
1 SELECT TOP (1000) [IdCEPLatLong]
2     , [CEP]
3     , [Bairro]
4     , [Cidade]
5     , [Estado]
6     , [Pais]
7     , [Latitude]
8     , [Longitude]
9     , [PlaceId]
10    , [IsOK]
11 FROM [CEPGEO].[dbo].[C
```

dbo.CEPGAPI X

RESULTS

	IdCEPLatLong	CEP
1	3274	01240000
		01523030
		02033020
		02076060
		02167080
		02202040

MESSAGES

(1000 rows affected)

Total execution time: 0

```
def configure(self, cnf: Optional[Any]=..., **kw)
```

Configure resources of a widget.

The values for resources are specified as keyword arguments. To get an overview about the allowed keyword arguments call the method keys.

```
text.configure(foreground = 'blue', font = 'w
```

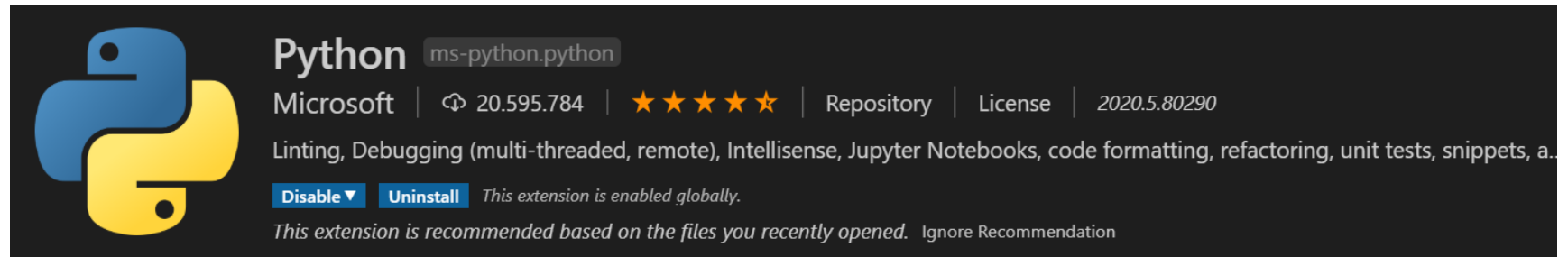
```
def btn_wingdings():
```

```
var.set('NYC')
```

```
text.configure(foreground = 'blue', font = 'w
```

Produtividade: Extensões

- 1- Python



Python ms-python.python

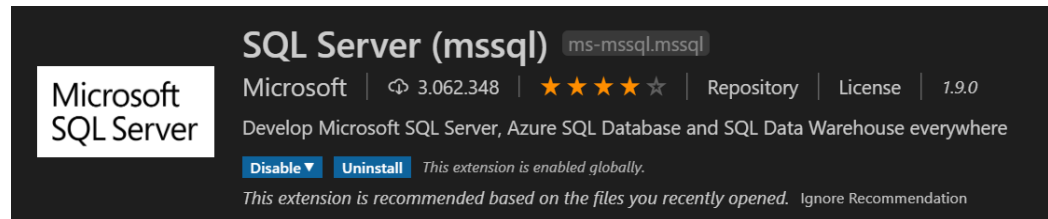
Microsoft | 20.595.784 | ★★★★★ | Repository | License | 2020.5.80290

Linting, Debugging (multi-threaded, remote), Intellisense, Jupyter Notebooks, code formatting, refactoring, unit tests, snippets, a..

Disable ▼ **Uninstall** This extension is enabled globally.

This extension is recommended based on the files you recently opened. Ignore Recommendation

- 2- SQL



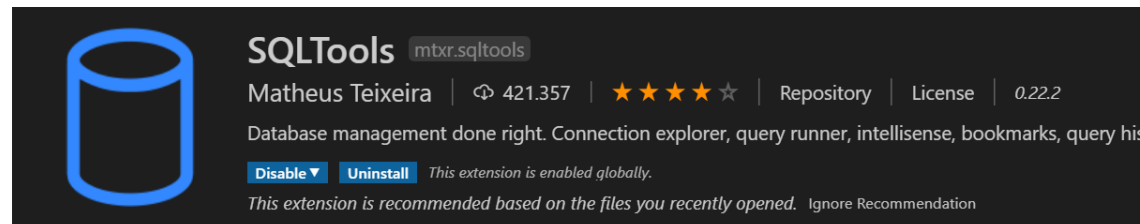
Microsoft SQL Server ms-mssql.mssql

Microsoft | 3.062.348 | ★★★★★☆ | Repository | License | 1.9.0

Develop Microsoft SQL Server, Azure SQL Database and SQL Data Warehouse everywhere

Disable ▼ **Uninstall** This extension is enabled globally.

This extension is recommended based on the files you recently opened. Ignore Recommendation



SQLTools mtr.sqltools

Matheus Teixeira | 421.357 | ★★★★★☆ | Repository | License | 0.22.2

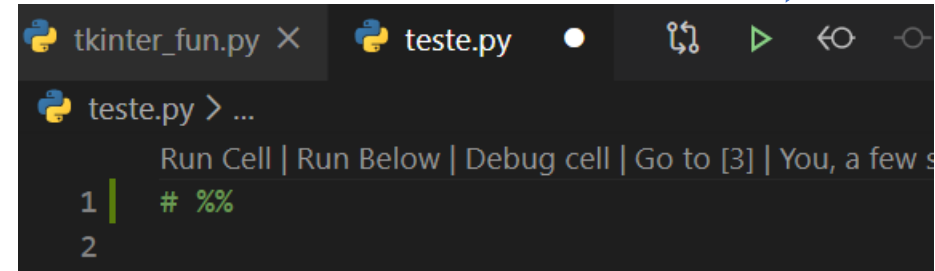
Database management done right. Connection explorer, query runner, intellisense, bookmarks, query his

Disable ▼ **Uninstall** This extension is enabled globally.

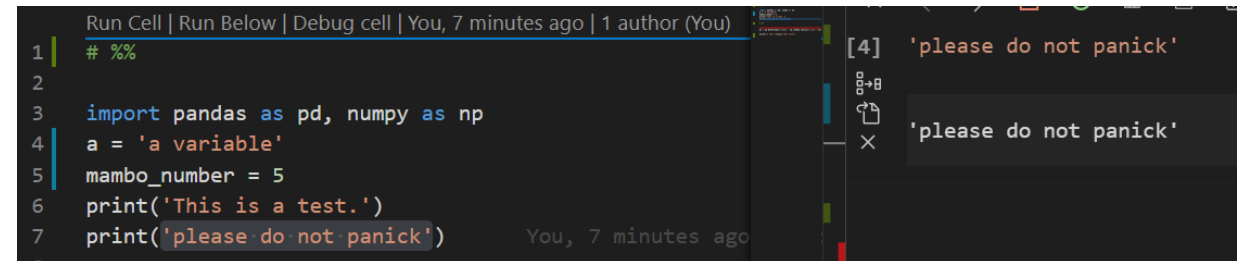
This extension is recommended based on the files you recently opened. Ignore Recommendation

Produtividade (Python): edição

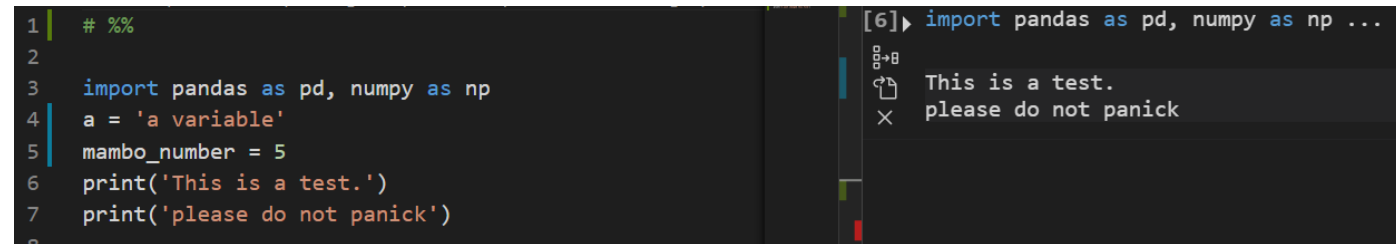
- Executar o código inteiro:
 - Pressione o botão play sobre o código



- Executar parte do código:
 - Selecione a parte desejada e pressione shift Enter

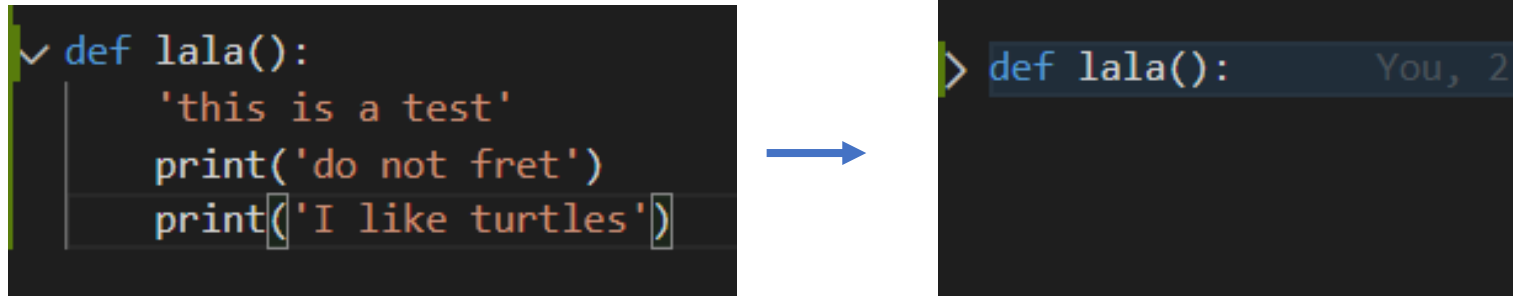


- Selecionar parte do código estilo Jupyter:
 - Selecione a célula e pressione shift-enter



Produtividade: Atalhos

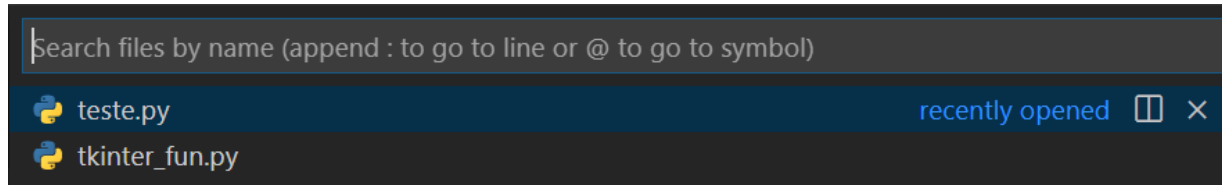
- Encolher grupo:
 - Clicar na cabeça de seta do lado do grupo



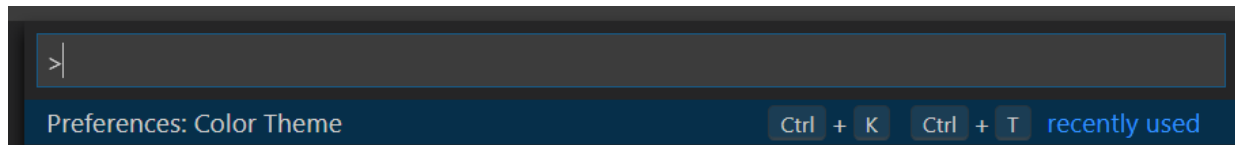
Produtividade: Atalhos de teclado

Buscas

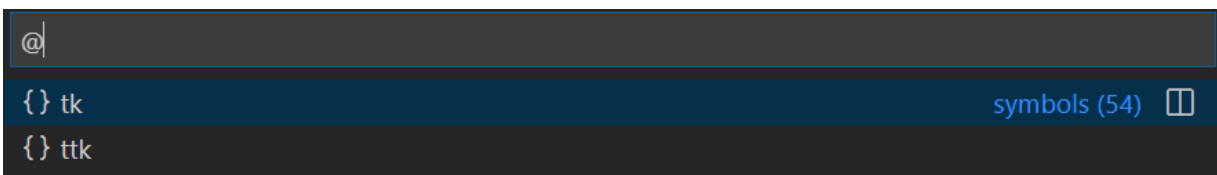
- Buscar arquivos: ctrl p



- Buscar funções do VScode: ctrl shift p



- Buscar funções/objetos: ctrl shift o



OU:
Ctrl p

E depois digitar:
> - funções
@ - arquivos

Produtividade: Atalhos de teclado

Localizar no arquivo

- Selecionar próximo igual: selecionar texto; ctrl d
- Selecionar todos os iguais: selecionar texto: ctrl shift l
- Buscar todas as referências: selecionar texto; shift F12

```
var = tk.StringVar()      You, 13 minutes ago • Uncommitted changes
var.set(''Q 33 NY'')      You, 13 minutes ago • Uncommitted changes
text = tk.Label(topframe, textvariable = var , font = 'WingDings 40',
                  background = 'yellow', foreground = 'red')
```

```
var = tk.StringVar()      You, 14 minutes ago • Uncommitted changes
var.set(''Q 33 NY'')      You, 14 minutes ago • Uncommitted changes
text = tk.Label(topframe, textvariable = var , font = 'WingDings 40',
                  background = 'yellow', foreground = 'red')
```

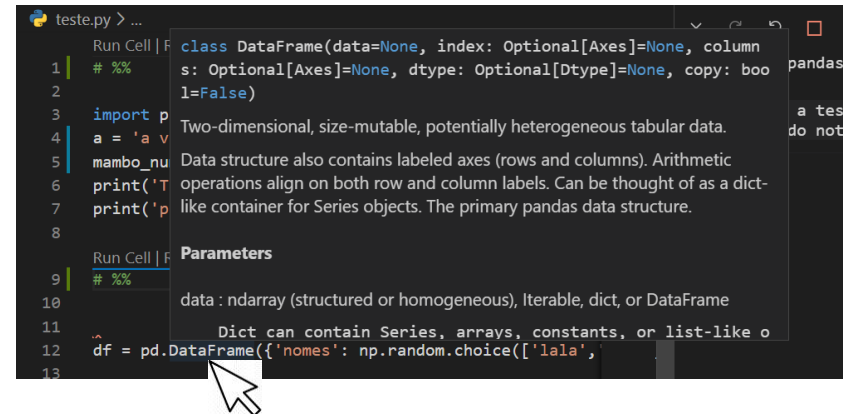
```
26 var = tk.StringVar()
tkinter_fun.py C:\Users\Sergio Marinho\Desktop\teste_python\Sergio\python - References (5)
21 win.title('Sergio's amazing program')
22 win.geometry('1200x1000')
23
24 win.focus()
25
26 var = tk.StringVar()
27
28 var.set(''Q 33 NY'')
29
30 text = tk.Label(topframe, textvariable = var , font = 'WingDings 40',
31                 background = 'yellow', foreground = 'red')
```

var = tk.StringVar()
var.set(''Q 33 NY'')
textvariable = var , font = 'Win
var.set('NYC')
var.set('NYC')

Produtividade: Atalhos de teclado

Obter ajuda

- Funções: deixe o mouse em cima da função



```
teste.py > ...
Run Cell | F
1 | # %%
2 |
3 | import pandas as pd
4 | a = 'a v
5 | mambo_nu
6 | print('T
7 | print('p
8 |
9 | Run Cell | F
9 | # %%
10 |
11 |
12 | df = pd.DataFrame({'nomes': np.random.choice(['lala',
13 |
```

class DataFrame(data=None, index: Optional[Axes]=None, columns: Optional[Axes]=None, dtype: Optional[Dtype]=None, copy: bool=True, verify_integrity: bool=False)

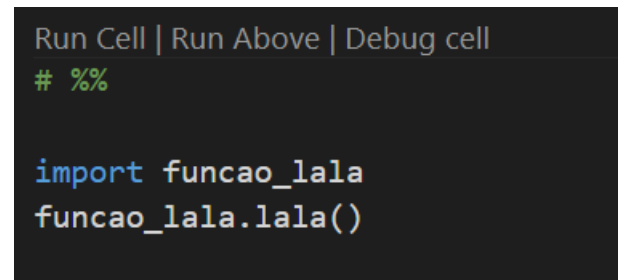
Two-dimensional, size-mutable, potentially heterogeneous tabular data. Data structure also contains labeled axes (rows and columns). Arithmetic operations align on both row and column labels. Can be thought of as a dict-like container for Series objects. The primary pandas data structure.

Parameters

data : ndarray (structured or homogeneous), Iterable, dict, or DataFrame

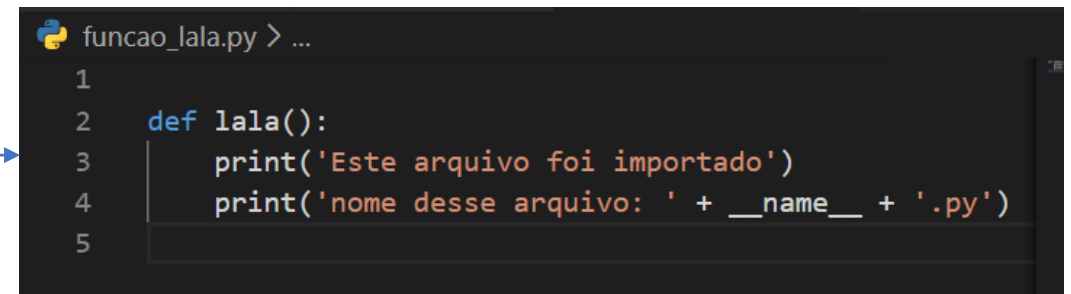
Dict can contain Series, arrays, constants, or list-like objects

- Localizar o arquivo de onde vem essa função:
 - Selecionar função – F12



```
Run Cell | Run Above | Debug cell
# %%

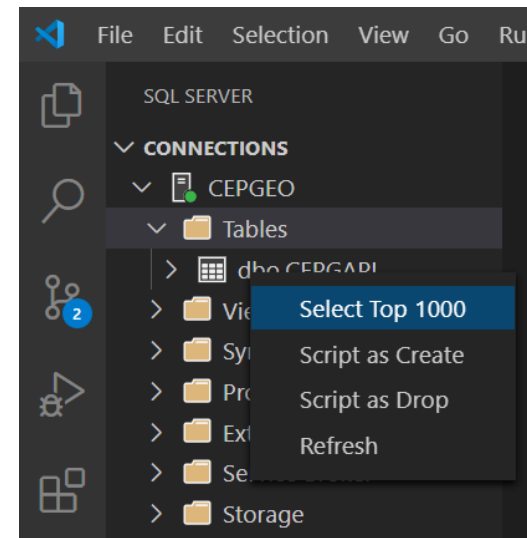
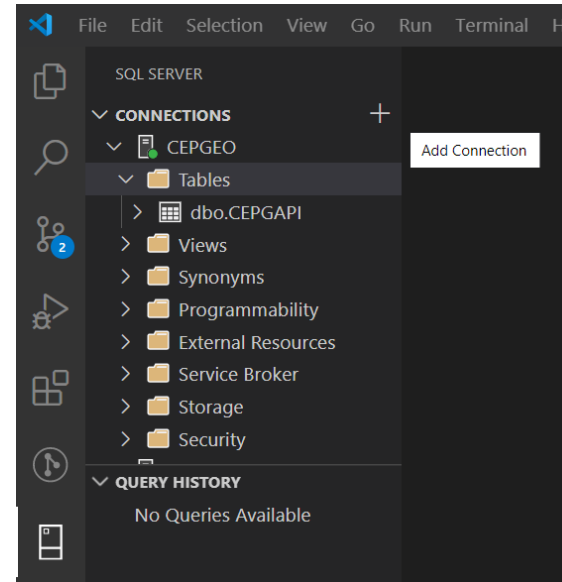
import funcao_lala
funcao_lala.lala()
```

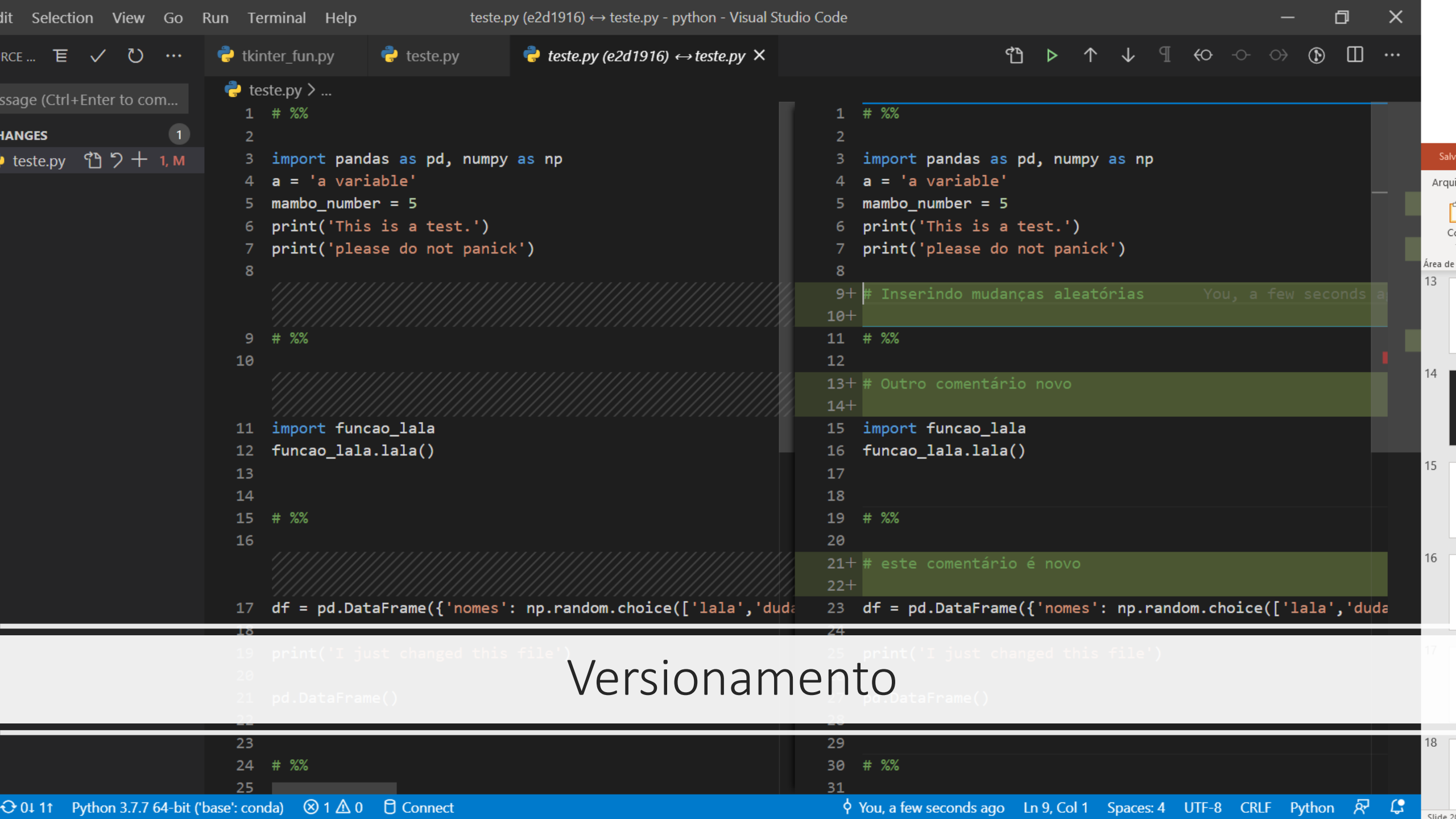


```
funcao_lala.py > ...
1
2 def lala():
3     print('Este arquivo foi importado')
4     print('nome desse arquivo: ' + __name__ + '.py')
5
```

Produtividade: SQL

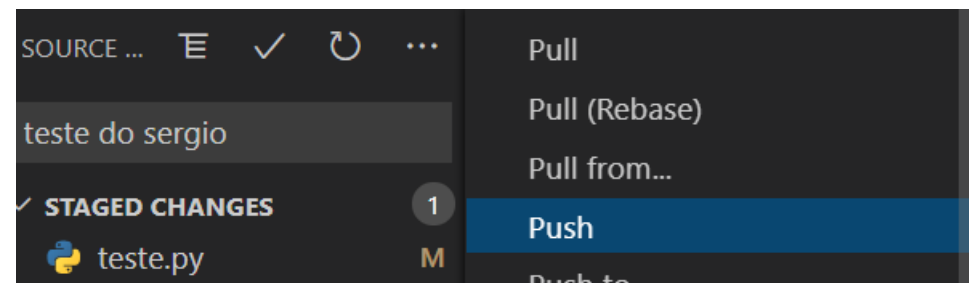
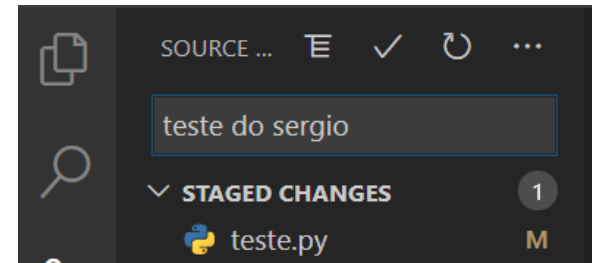
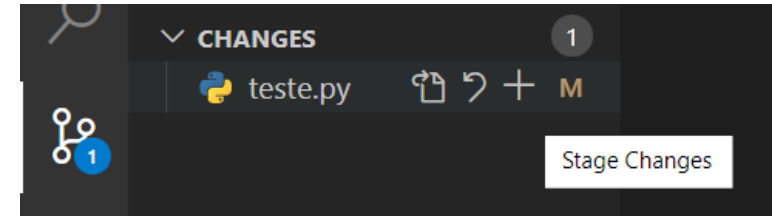
- Criar conexão
 - Clicar em 'add connection'
- Fazer query:
 - Clicar em 'Select Top 1000'





Versionamento:

- Escolher pasta com .git habilitado
- Fazer o git add: subir arquivos salvos
 - Clicar em 'Stage Changes'
- Fazer o git commit -m: comentários
 - Digitar a mensagem e clicar em 'Commit'
- Git push/pull:
 - Clicar em ... → Push/Pull



Exemplo de projeto feito no VS Code

Graças ao VS code, eu pude:

- Obter instruções sobre as funções rapidamente
- Localizar e editar variáveis facilmente
- Testar e validar o código e variáveis a cada instante

