

## Solução - PROJETO: Jogo do galo.

*Instalação do software, consultar vídeo:*

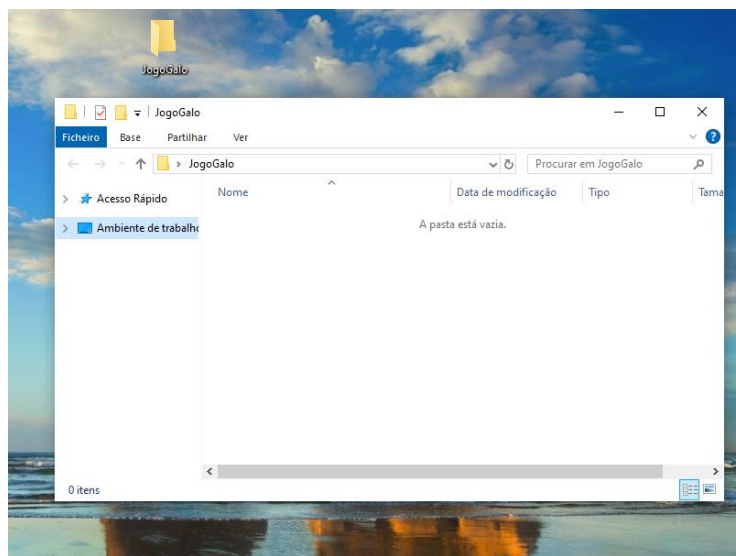
### **Vídeo 2: Instalação**

Por João Martiniano

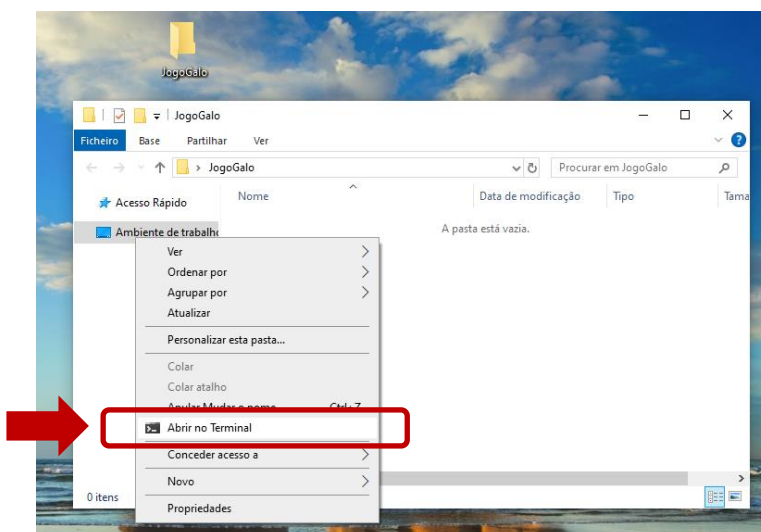
[https://www.youtube.com/watch?v=uV6r\\_deuusM](https://www.youtube.com/watch?v=uV6r_deuusM)

### **Criação do ambiente GIT usando a linha de comando:**

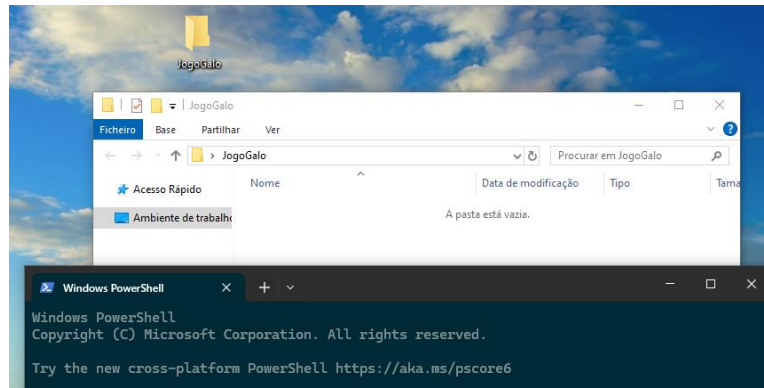
Neste exemplo no ambiente de trabalho, criamos uma pasta de nome “JogoGalo”:



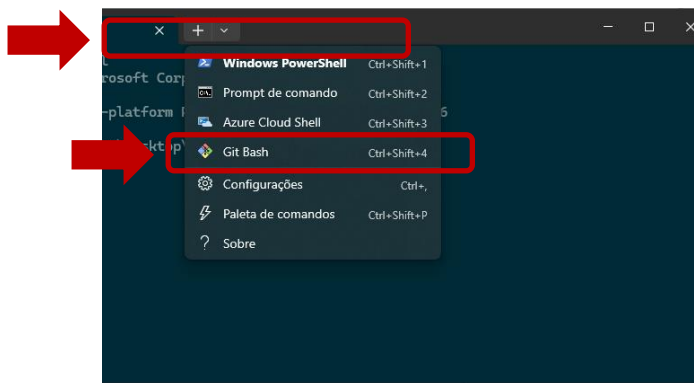
Botão do lado direito do rato, escolher opção “Abrir Terminal”:



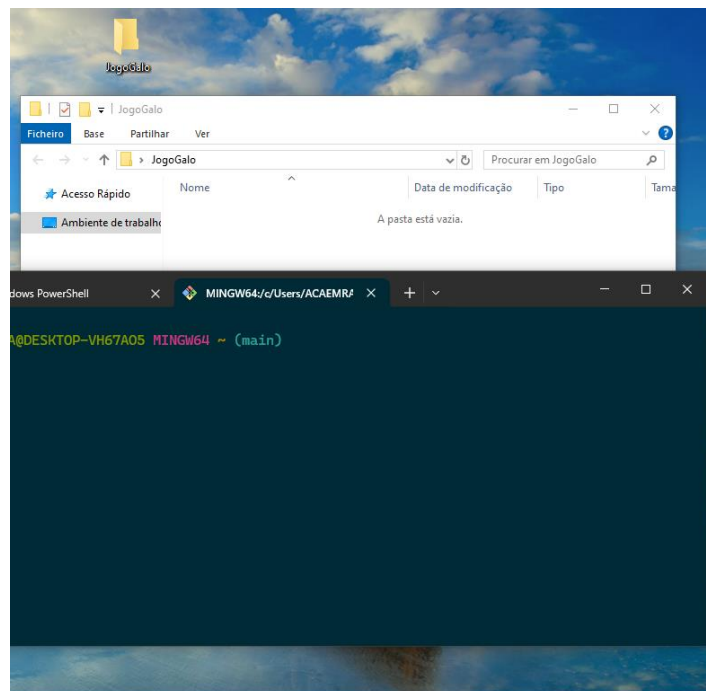
E obtêm-se:



Carregar na linha de cima da janela e selecionar **Git Bash**:

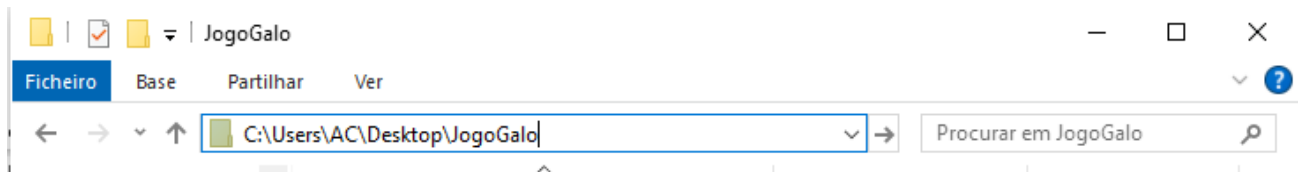


Obtêm-se:

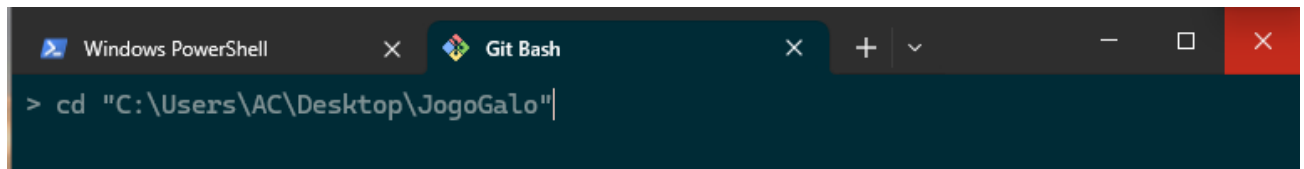


No *Git Bash* vamos mudar para a pasta do nosso trabalho com o comando **cd**.

Para obter o caminho, copiar o caminho na janela da pasta de trabalho:



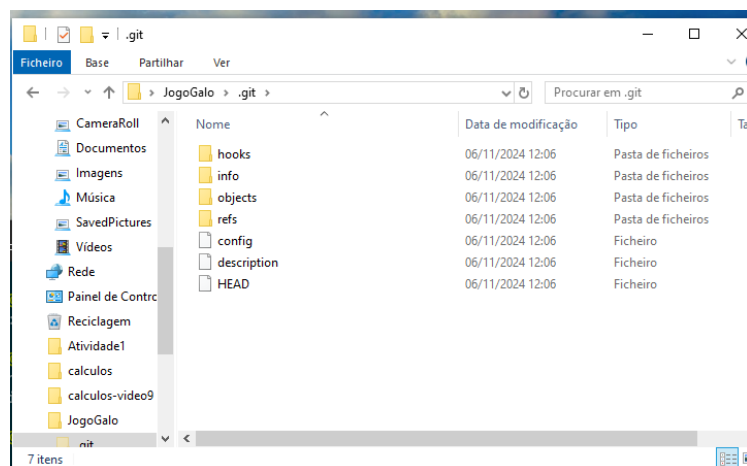
Atenção, não esquecer de colocar o caminho entre aspas:



Depois iniciar o processo de git através do comando:

```
git init
```

Notar que o comando cria uma estrutura na pasta de trabalho:



Em caso de dúvida podemos sempre executar o comando:

```
git help
```

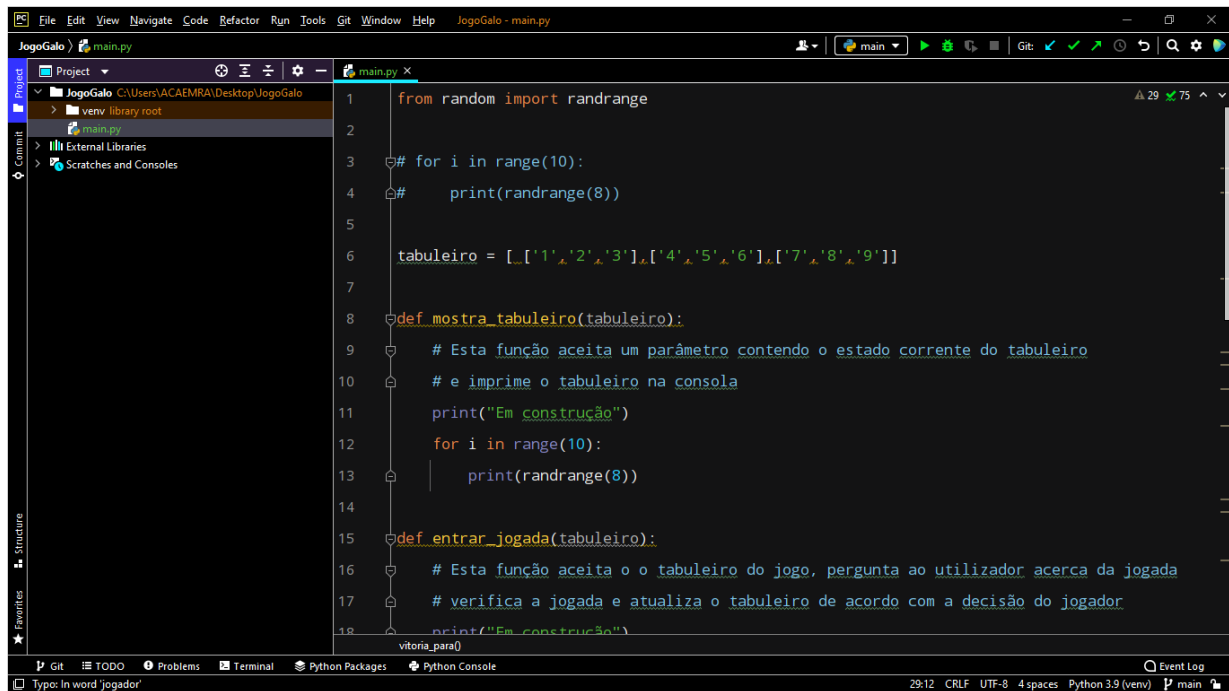
Que nos dá ajuda de como usar o comando.

Ajuda em opções específicas do comando *git*:

```
git help commit
```

Atenção que no comando *git* há que ter atenção à ordem das opções.

Vamos criar o projeto de *python* base no enunciado, na pasta onde criamos a estrutura *git*. Neste caso adicionamos o ficheiro *main.py*:

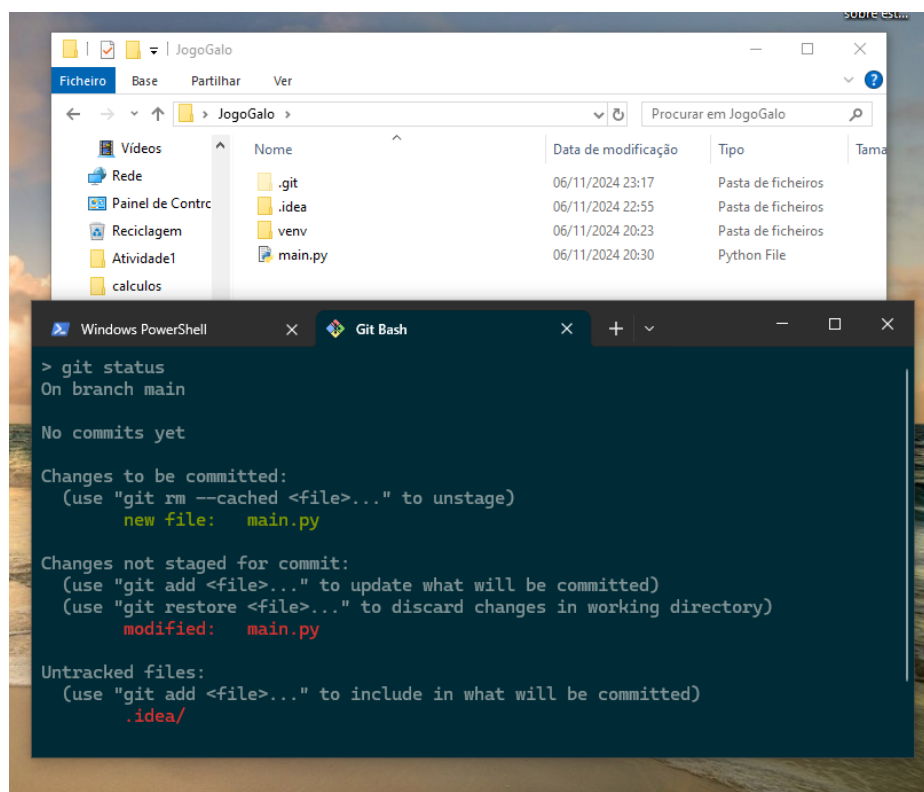


```
1 from random import randrange
2
3 # for i in range(10):
4 #     print(randrange(8))
5
6 tabuleiro = [['1','2','3'],['4','5','6'],['7','8','9']]
7
8 def mostra_tabuleiro(tabuleiro):
9     # Esta função aceita um parâmetro contendo o estado corrente do tabuleiro
10    # e imprime o tabuleiro na consola
11    print("Em construção")
12    for i in range(10):
13        print(randrange(8))
14
15 def entrar_jogada(tabuleiro):
16    # Esta função aceita o o tabuleiro do jogo, pergunta ao utilizador acerca da jogada
17    # verifica a jogada e atualiza o tabuleiro de acordo com a decisão do jogador
18    print("Em construção")
19    vitoria_para()
```

Vamos verificar o estado do git com o comando:

```
git status
```

Que verifica como está o nosso status (estado) e nos diz que o ficheiro *main.py* ainda não foi sujeito a um commit:



Windows File Explorer - JogoGalo

Nome	Data de modificação	Tipo
.git	06/11/2024 23:17	Pasta de ficheiros
.idea	06/11/2024 22:55	Pasta de ficheiros
venv	06/11/2024 20:23	Pasta de ficheiros
main.py	06/11/2024 20:30	Python File

Git Bash

```
> git status
On branch main

No commits yet

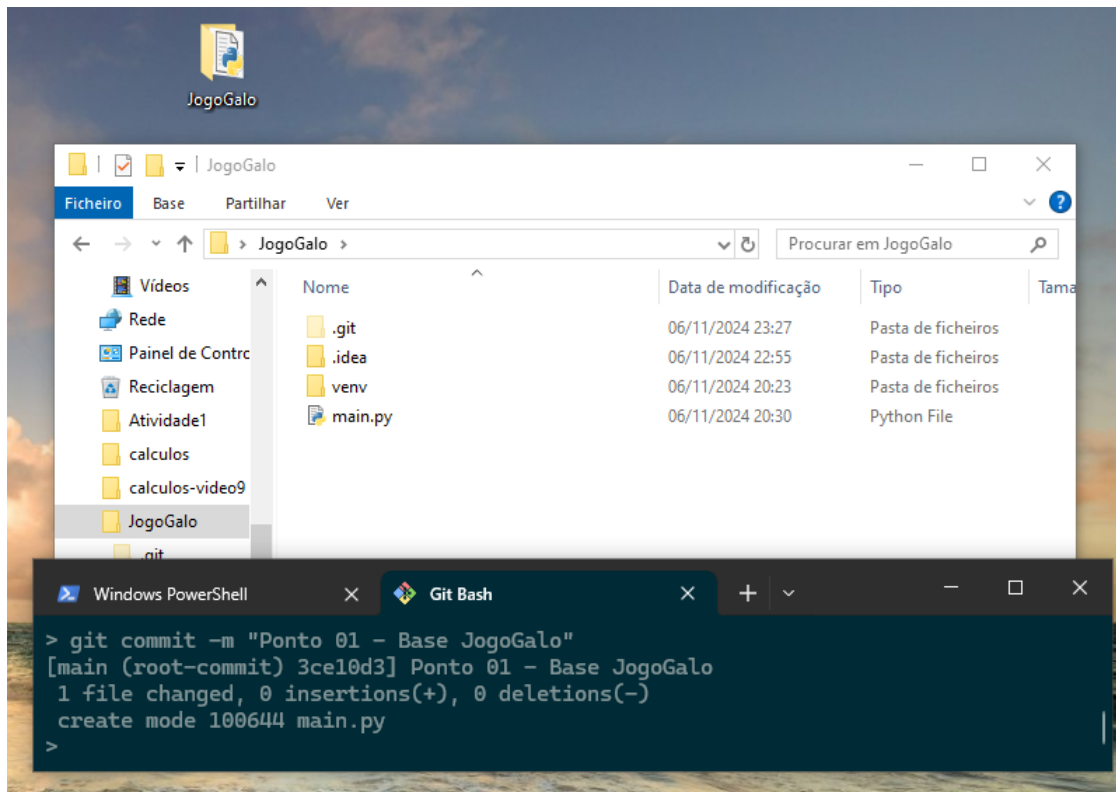
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   main.py

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   main.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  .idea/
```

Vamos fazer o nosso commit com o nome de "Ponto 01 - Base JogoGalo", via comando:

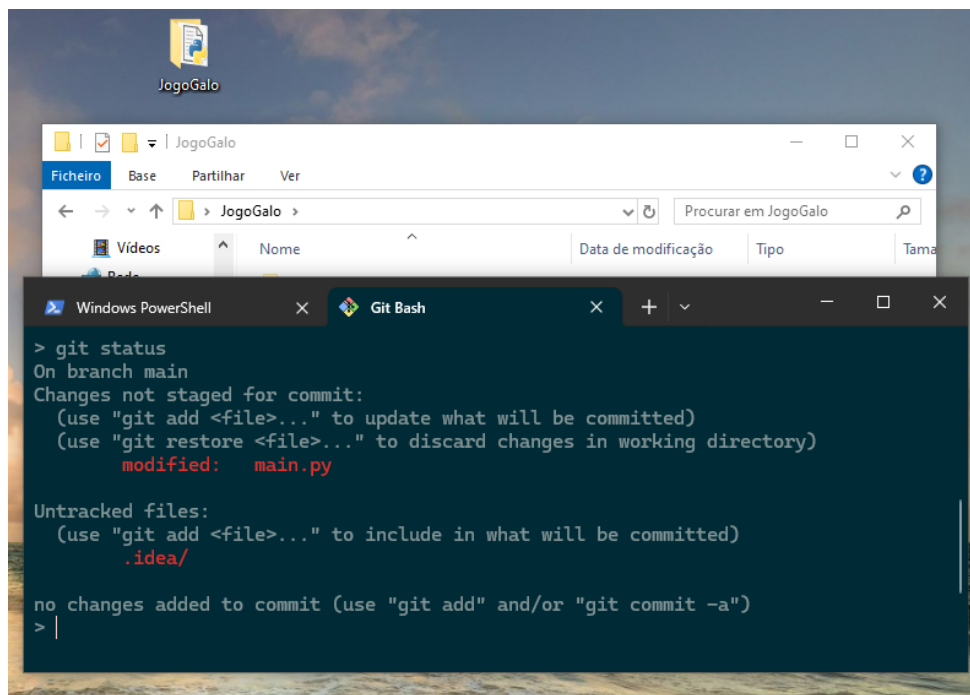
```
git commit -m "Ponto 01 - Base JogoGalo"
```



Se introduzirmos novamente o comando:

```
git status
```

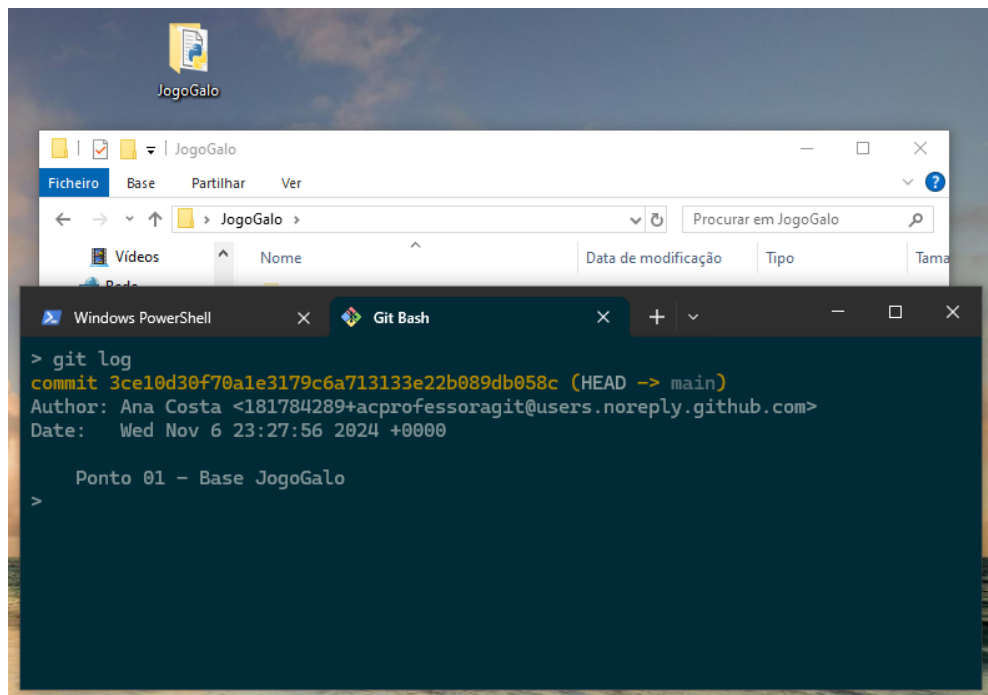
Vai mostrar que agora não há nada a ser guardado (em commit):



Comando

```
git log
```

Mostra-nos que entradas de *commit* temos:



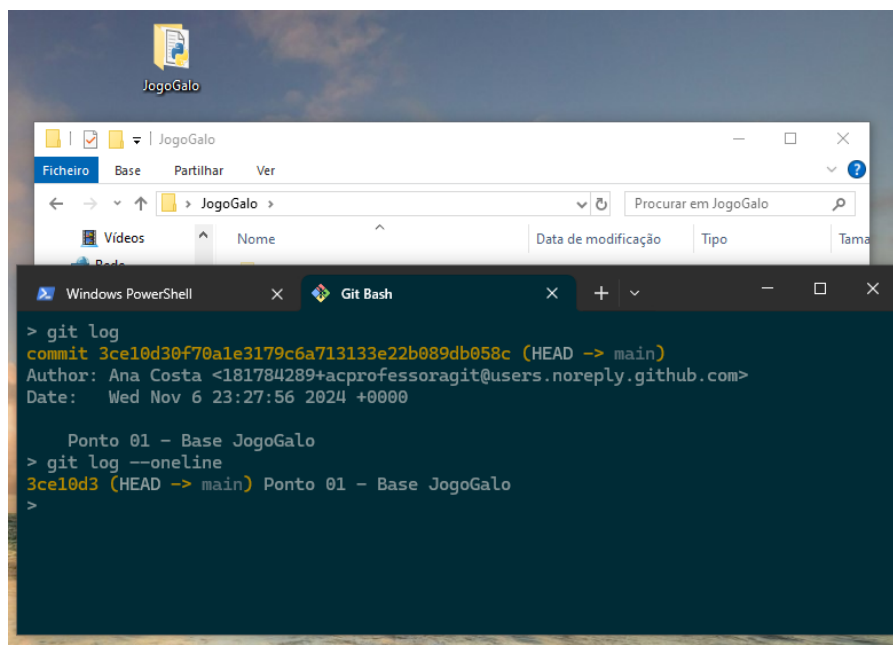
```
> git log
commit 3ce10d30f70a1e3179c6a713133e22b089db058c (HEAD -> main)
Author: Ana Costa <181784289+acprofessoragit@users.noreply.github.com>
Date:   Wed Nov 6 23:27:56 2024 +0000

    Ponto 01 - Base JogoGalo

>
```

Para vermos a versão reduzida, utilizamos o comando:

```
git log --oneline
```



```
> git log
commit 3ce10d30f70a1e3179c6a713133e22b089db058c (HEAD -> main)
Author: Ana Costa <181784289+acprofessoragit@users.noreply.github.com>
Date:   Wed Nov 6 23:27:56 2024 +0000

    Ponto 01 - Base JogoGalo

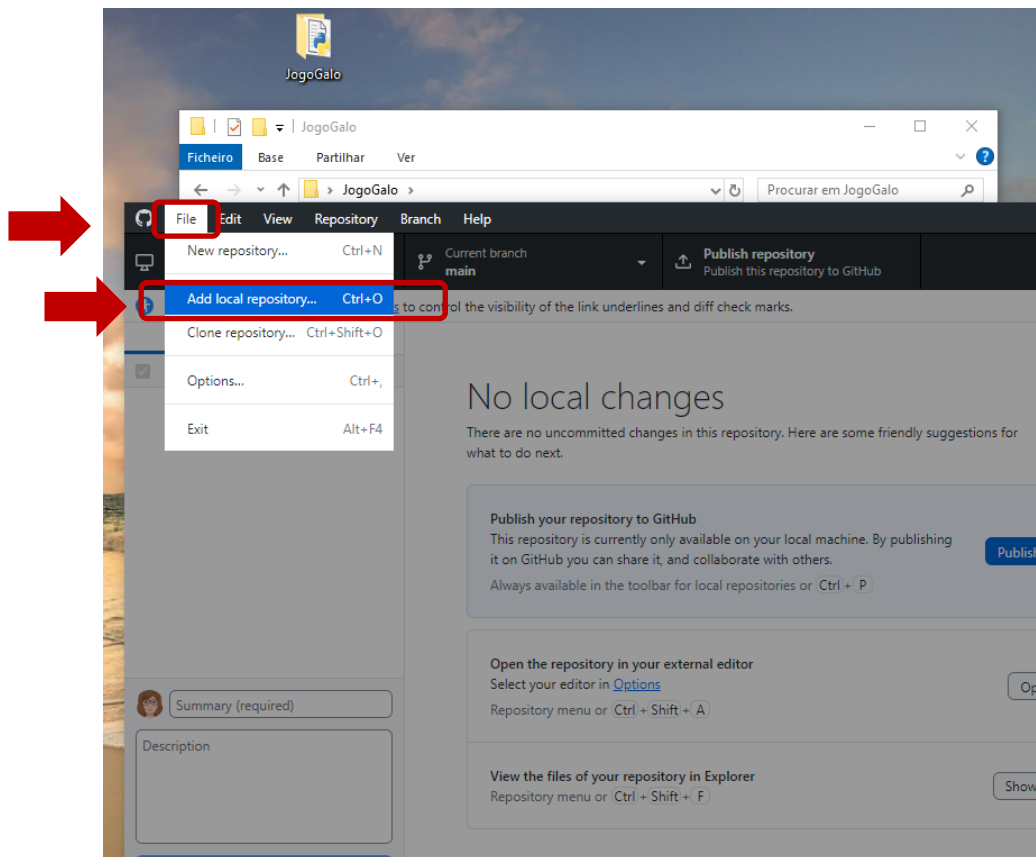
> git log --oneline
3ce10d3 (HEAD -> main) Ponto 01 - Base JogoGalo

>
```

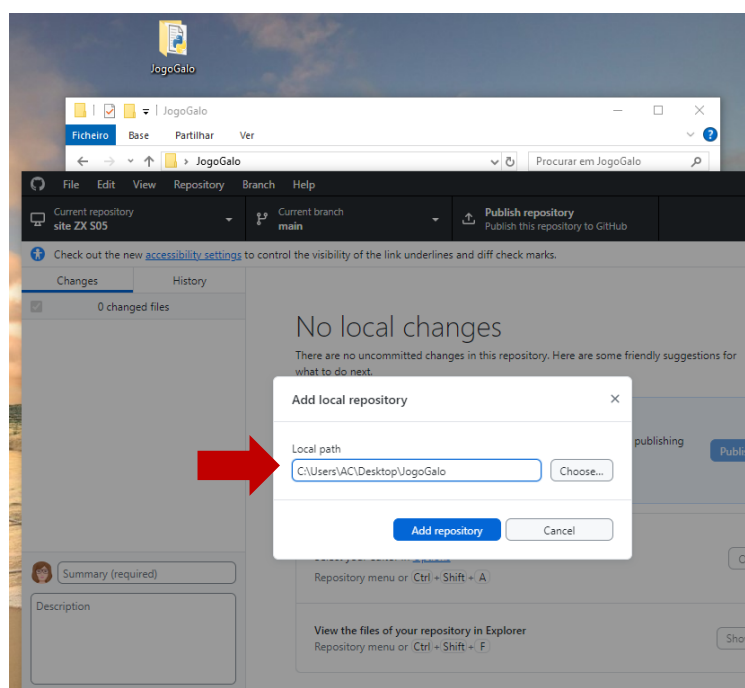
## Programa Git Hub

O software GitHub permite criar a nossa estrutura de versões do nosso software sem utilizarmos comandos na linha de comando do Gitbash.

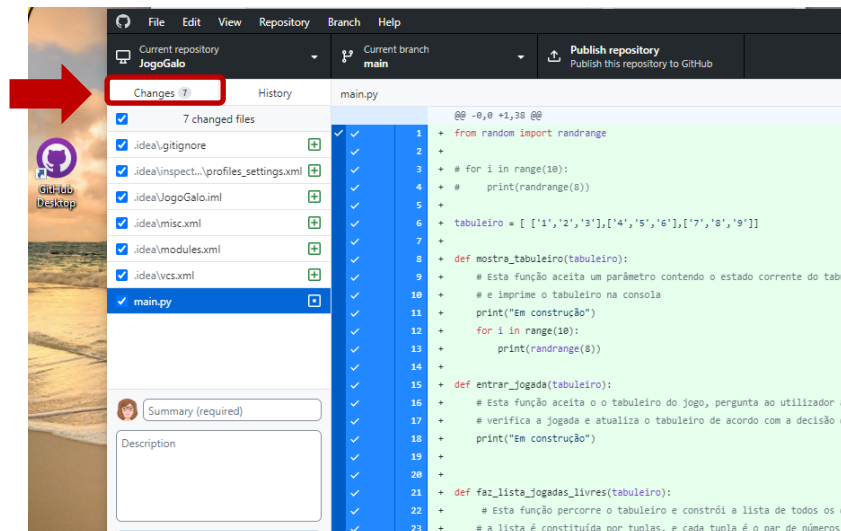
Temos de acrescentar o caminho do nosso trabalho no GitHub:



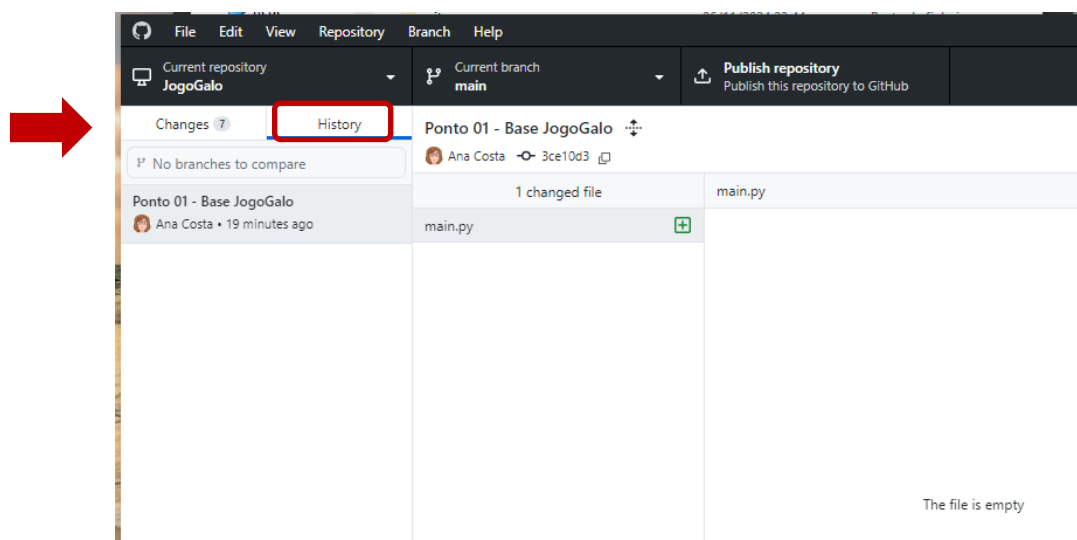
Escrever o caminho da pasta do nosso trabalho (projeto):



Notar que já conseguimos ver o conteúdo do nosso ficheiro main.py, via separador “Changes”:



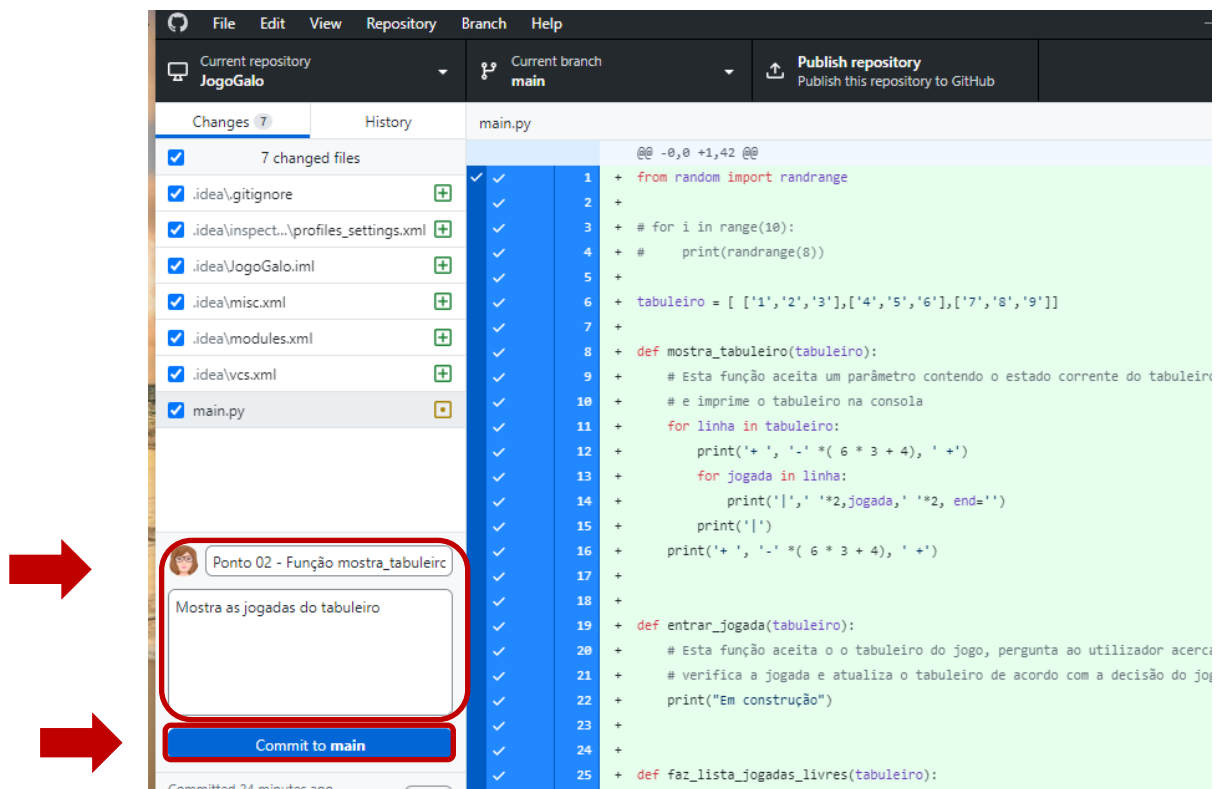
No separador “History” conseguimos visualizar os nossos pontos de commit:



Entretanto, no nosso programa python (main.py) acrescentamos o código para a função **mostra\_tabuleiro**

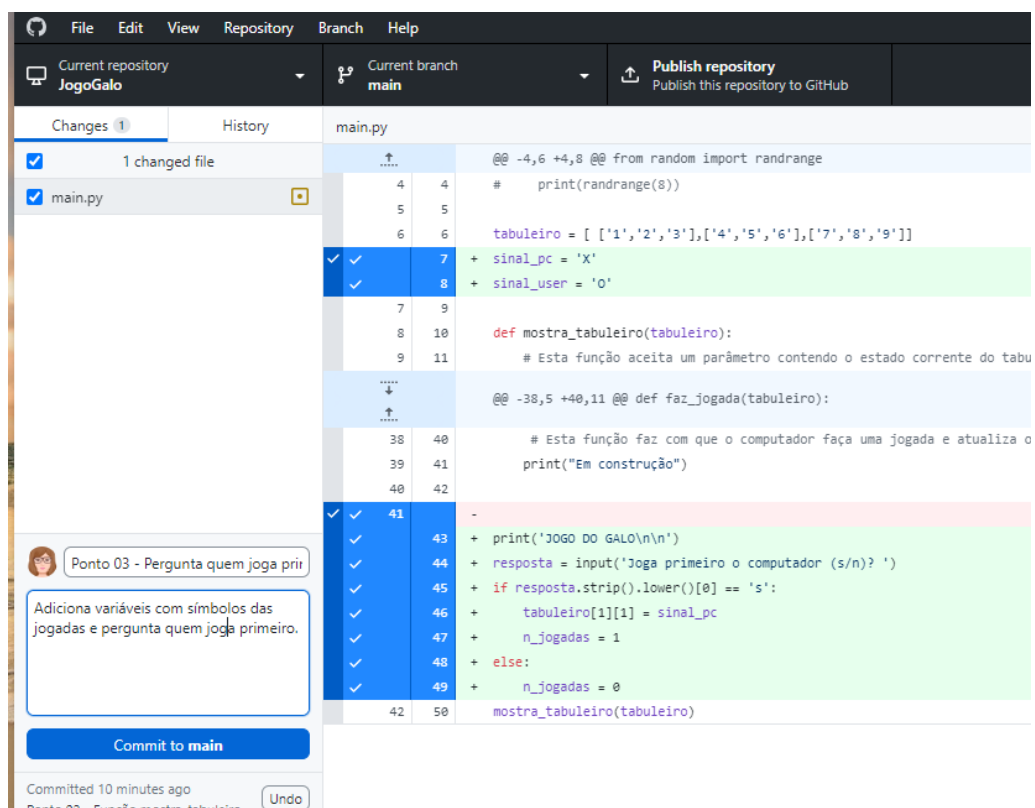


Queremos salvar as nossas alterações através de um *commit*, bastando para isso escrever o nome do *commit* e eventuais observações no retângulo inferior. Depois é só carregar no botão azul "Commit main".

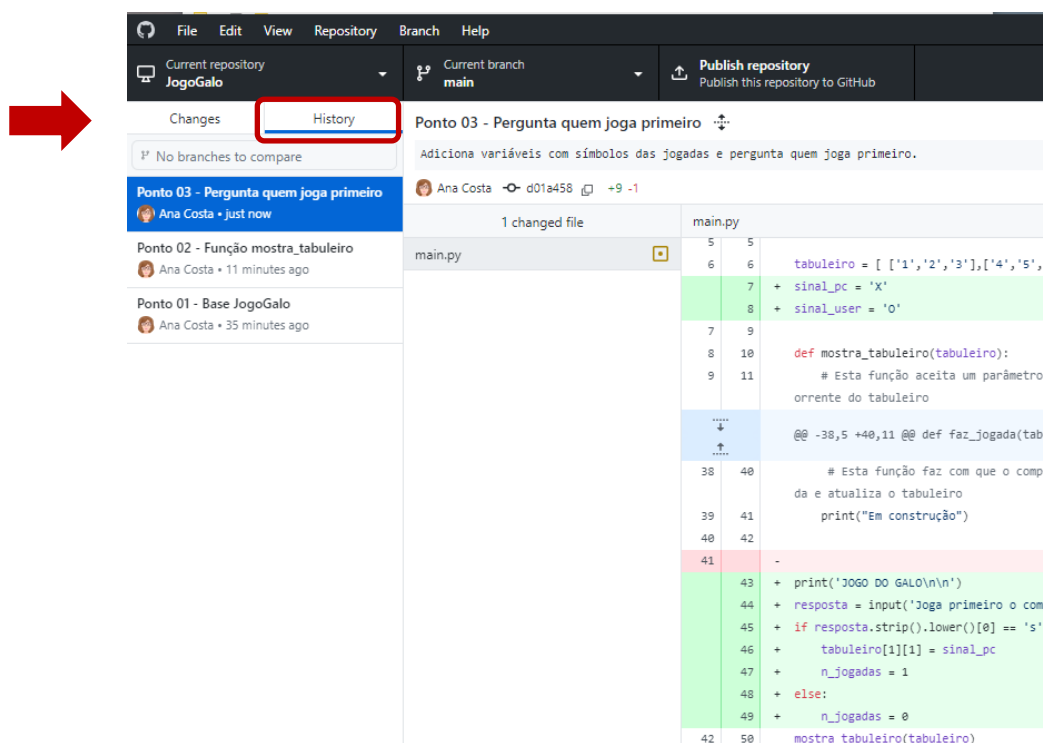


Vamos agora adicionar mais código ao nosso programa e fazer mais um ponto de commit.

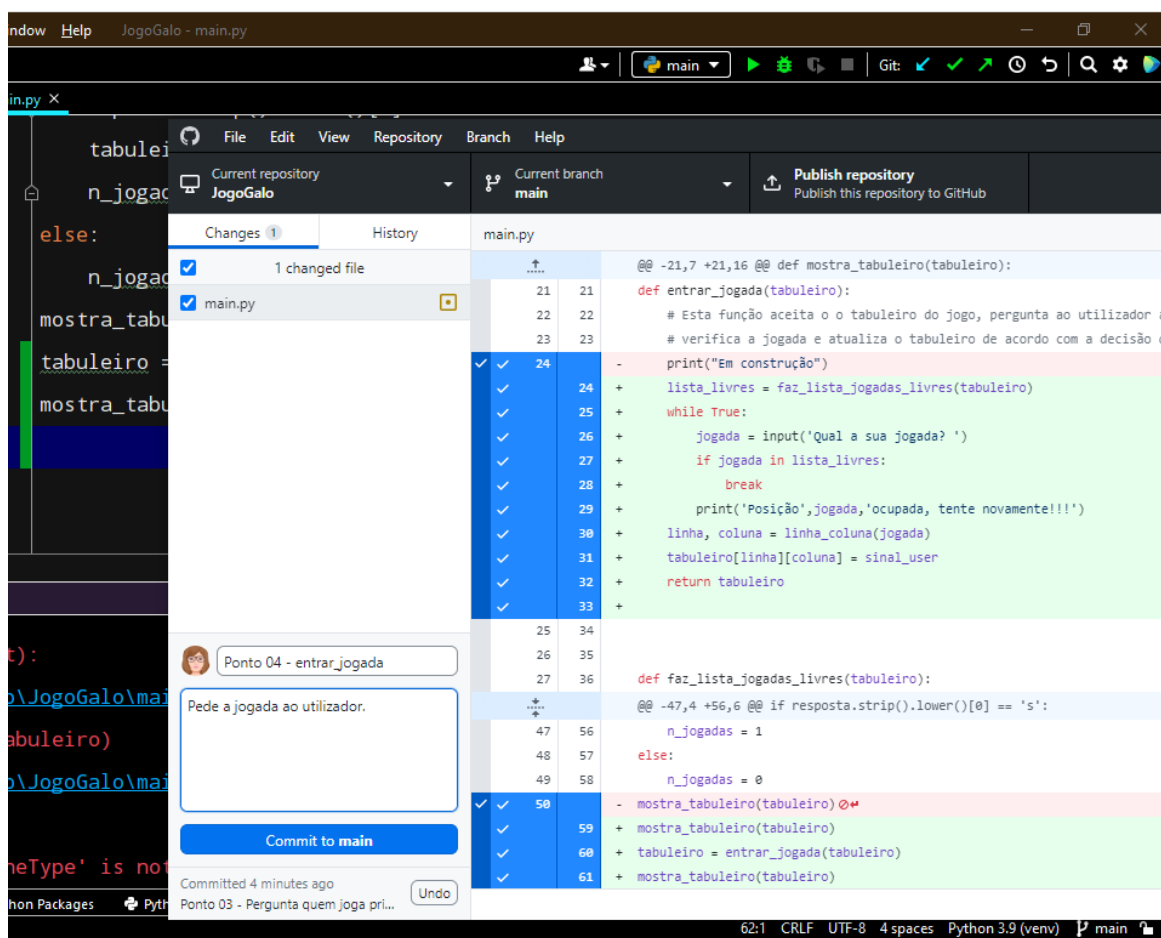
Notar que mostra sublinhado a azul o código que acrescentamos e a rosa o que retiramos (neste caso apagamos uma linha).



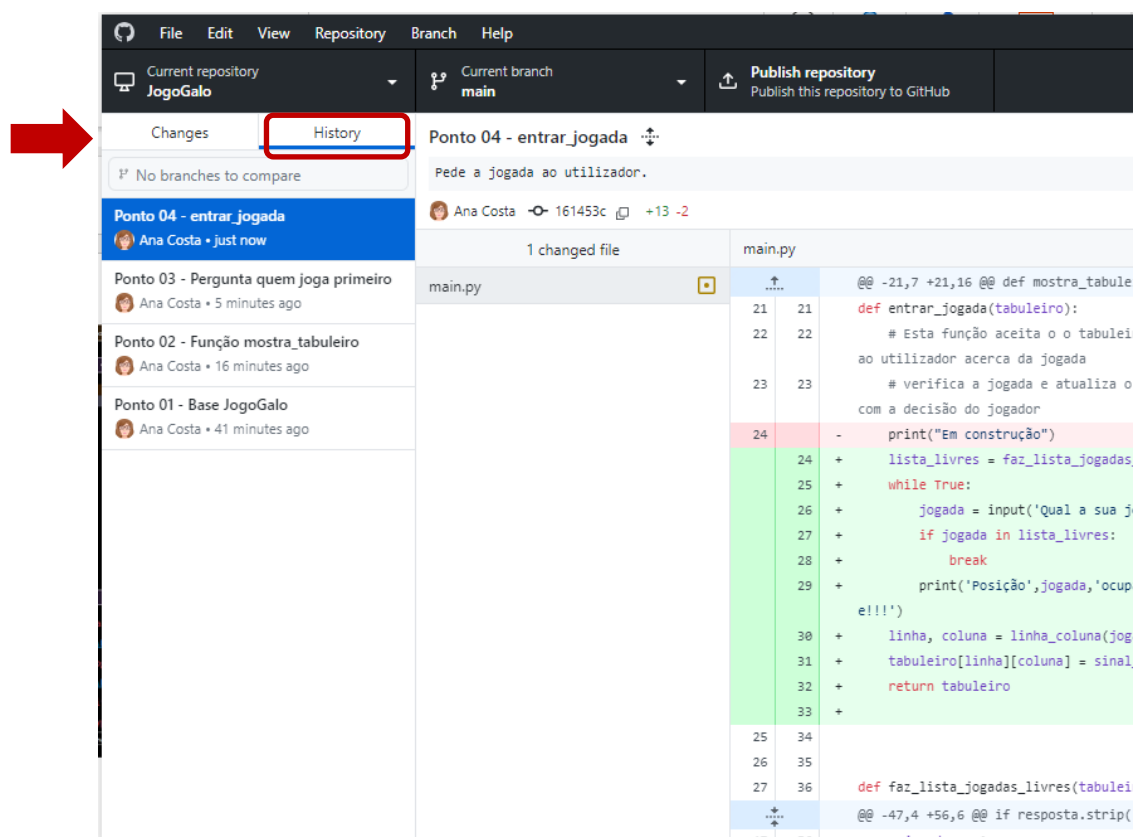
Novamente em "History", conseguimos visualizar os nossos pontos de *commit* e alterações:



Vamos criar mais um ponto de *commit*, com o código escrito em *python* para a função `entrar_jogada`

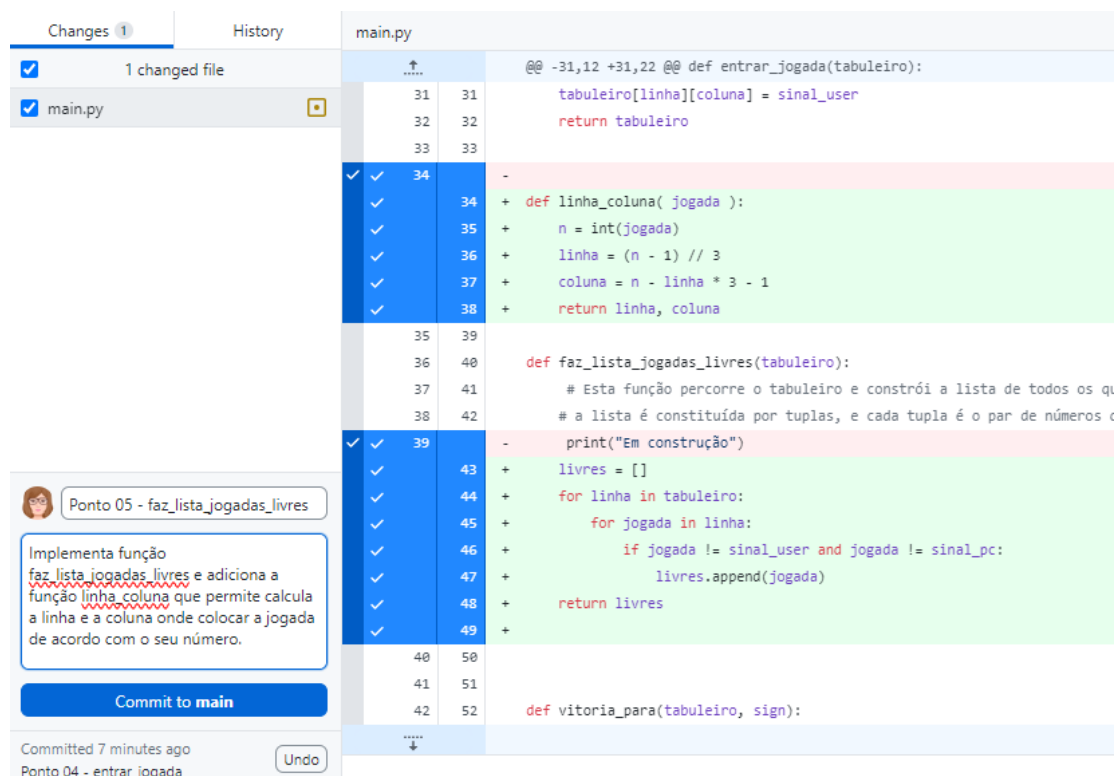


Novamente em “History”, conseguimos visualizar os nossos pontos de *commit* e alterações:

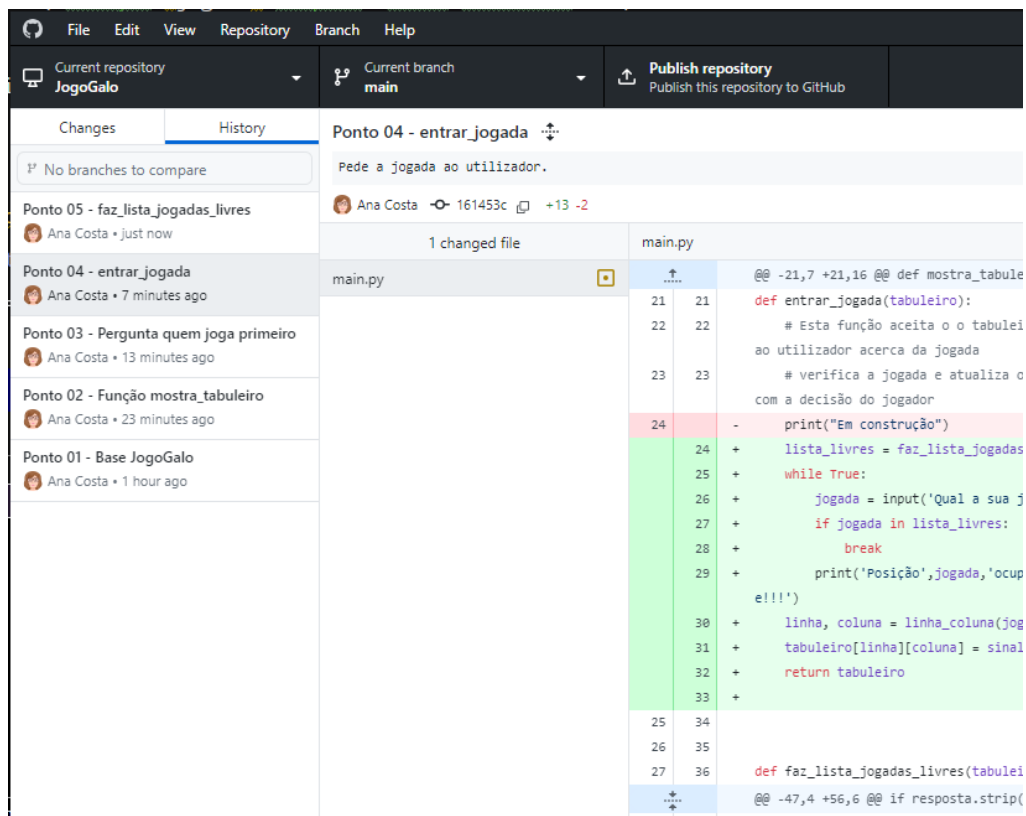


Mais um ponto de *commit*, com o código escrito em *python* para a função `faz_lista_jogadas_livres`

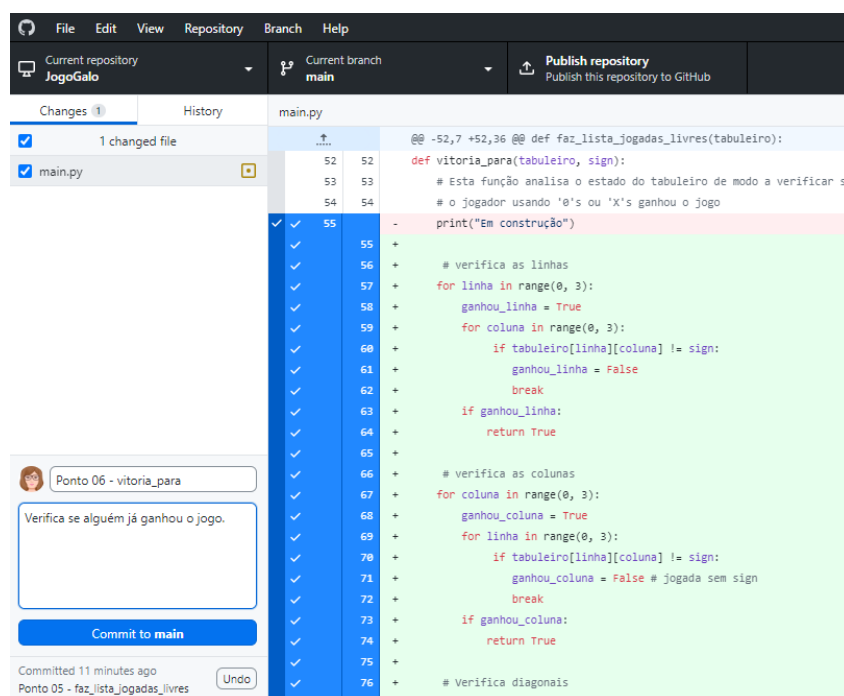
Notar que removemos a linha: `print ("Em construção")`



Novamente em “History”, conseguimos visualizar os nossos pontos de *commit* e alterações:



Mais um ponto de *commit*:



Em “History”, conseguimos outra vez visualizar os nossos pontos de *commit* e alterações:

The screenshot shows the Git GUI interface. The top bar includes menus: File, Edit, View, Repository, Branch, and Help. Below the bar, the 'Current repository' is 'JogoGalo' and the 'Current branch' is 'main'. A 'Publish repository' button is visible. The left sidebar shows the commit history with the following entries:

- Ponto 06 - vitoria\_para (Ana Costa • just now)
- Ponto 05 - faz\_lista\_jogadas\_livres** (Ana Costa • 12 minutes ago)
- Ponto 04 - entrar\_jogada (Ana Costa • 19 minutes ago)
- Ponto 03 - Pergunta quem joga primeiro (Ana Costa • 24 minutes ago)
- Ponto 02 - Função mostra\_tabuleiro (Ana Costa • 35 minutes ago)
- Ponto 01 - Base JogoGalo (Ana Costa • 1 hour ago)

The main area displays the diff for 'Ponto 05 - faz\_lista\_jogadas\_livres'. The commit message is: 'Implementa função faz\_lista\_jogadas\_livres e adiciona a função linha\_coluna que permite calcular a linha e coluna onde colocar a jogada de acordo com o seu número.' The commit hash is 'ddc8c87' with a net change of '+12 -2'. One file, 'main.py', is shown as changed. The diff content is as follows:

```

@@ -31,12 +31,22 @@ def entrar_jogada(tabuleiro):
     31     31     tabuleiro[linha][coluna] = sinal
     32     32     return tabuleiro
     33     33
+    34     -
+    34     + def linha_coluna( jogada ):
+    35     +     n = int(jogada)
+    36     +     linha = (n - 1) // 3
+    37     +     coluna = n - linha * 3 - 1
+    38     +     return linha, coluna
+
     35     39
     36     40     def faz_lista_jogadas_livres(tabuleiro):
     37     41         # Esta função percorre o tabuleiro
     38     42         # a de todos os quadrados livres;
     39     43         # a lista é constituída por tuplas
     40     44         # par de números de linha e coluna
     41     45         print("Em construção")
     42     46         livres = []
     43     47         for linha in tabuleiro:
     44     48             for jogada in linha:
     45     49                 if jogada != sinal_usuario:
     46     50                     livres.append((linha, coluna))
     47     51         return livres

```

Mais um ponto de *commit* e *history*:

The screenshot shows the Git GUI interface. The left sidebar shows the commit history with the following entries:

- Ponto 07 - Ciclo de jogadas** (Ana Costa • just now)
- Ponto 06 - vitoria\_para (Ana Costa • 4 minutes ago)
- Ponto 05 - faz\_lista\_jogadas\_livres (Ana Costa • 15 minutes ago)
- Ponto 04 - entrar\_jogada (Ana Costa • 23 minutes ago)
- Ponto 03 - Pergunta quem joga primeiro (Ana Costa • 28 minutes ago)
- Ponto 02 - Função mostra\_tabuleiro (Ana Costa • 39 minutes ago)
- Ponto 01 - Base JogoGalo (Ana Costa • 1 hour ago)

The main area displays the diff for 'Ponto 07 - Ciclo de jogadas'. The commit message is: 'Coloca o ciclo de jogadas no programa principal.' The commit hash is 'fc92b88' with a net change of '+20 -2'. One file, 'main.py', is shown as changed. The diff content is as follows:

```

@@ -96,5 +96,23 @@ if resposta.strip() != '':
     96     96     else:
     97     97         n_jogadas = 0
     98     98         mostra_tabuleiro(tabuleiro)
+    99     99         tabuleiro = entrar_jogada(tabuleiro)
+   100    100         mostra_tabuleiro(tabuleiro)
+
+   101    101         while True:
+   102    102             tabuleiro = entrar_jogada(tabuleiro)
+   103    103             mostra_tabuleiro(tabuleiro)
+   104    104             if vitoria_para(tabuleiro, sinal_usuario):
+   105    105                 print('Parabéns ganhou.')
+   106    106                 break
+   107    107             n_jogadas += 1
+   108    108             if n_jogadas == 9:
+   109    109                 print("Empate.")
+   110    110                 break
+   111    111             tabuleiro = faz_jogada(tabuleiro)
+   112    112             mostra_tabuleiro(tabuleiro)
+   113    113             n_jogadas += 1
+   114    114             if vitoria_para(tabuleiro, sinal_usuario):
+   115    115                 print('Perdeu!!!')
+   116    116                 break

```

Mais um ponto de *commit*:

The screenshot shows the PyCharm IDE with the 'Commit' window open. The window has a dark theme. At the top, it shows the current repository 'JogoGalo' and the current branch 'main'. Below this, there are tabs for 'Changes' and 'History'. The 'Changes' tab is active, showing a list of files to be committed. The file 'main.py' is selected, and its changes are displayed in a diff view. The diff shows a function 'faz\_jogada' that takes a 'tabuleiro' parameter and returns a 'tabuleiro'. The function includes a while loop that continues until a move is made. The commit message is 'Ponto 08 - faz\_jogada' and the description is 'Implementa a execução da jogada.' The commit button is labeled 'Commit to main'.

Current repository: JogoGalo  
Current branch: main  
Publish repository: Publish this repository to GitHub

Changes: 1 changed file  
main.py

Commit message: Ponto 08 - faz\_jogada  
Description: Implementa a execução da jogada.  
Commit to main

Committed 3 minutes ago  
Ponto 07 - Ciclo de jogadas

A sua *history*:

The screenshot shows the PyCharm IDE with the 'History' tab active. The history list on the left shows a series of commits from 'Ponto 01' to 'Ponto 08'. The commit 'Ponto 08 - faz\_jogada' is selected, and its details are shown on the right. The details include the commit message 'Implementa a execução da jogada.', the author 'Ana Costa', and the commit hash 'f65846e'. The diff view shows the changes made in this commit, which include the implementation of the 'faz\_jogada' function.

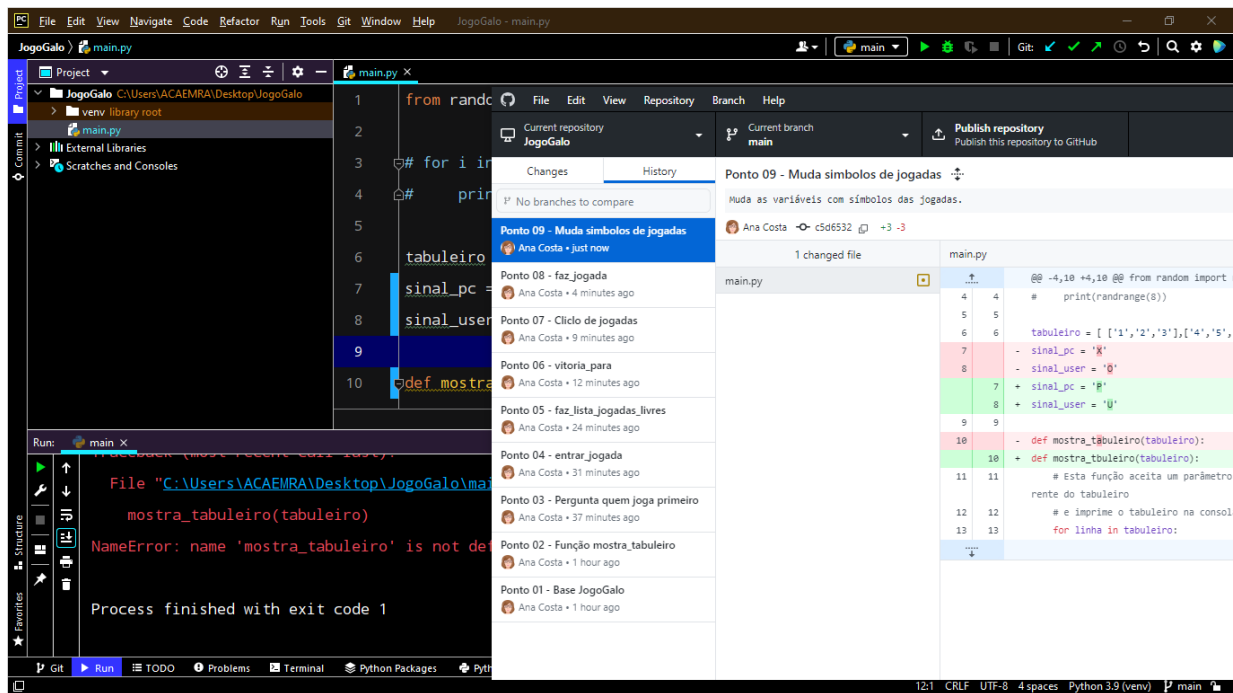
History: Ponto 08 - faz\_jogada  
Implementa a execução da jogada.  
Ana Costa • f65846e • +10 -1

1 changed file: main.py

main.py

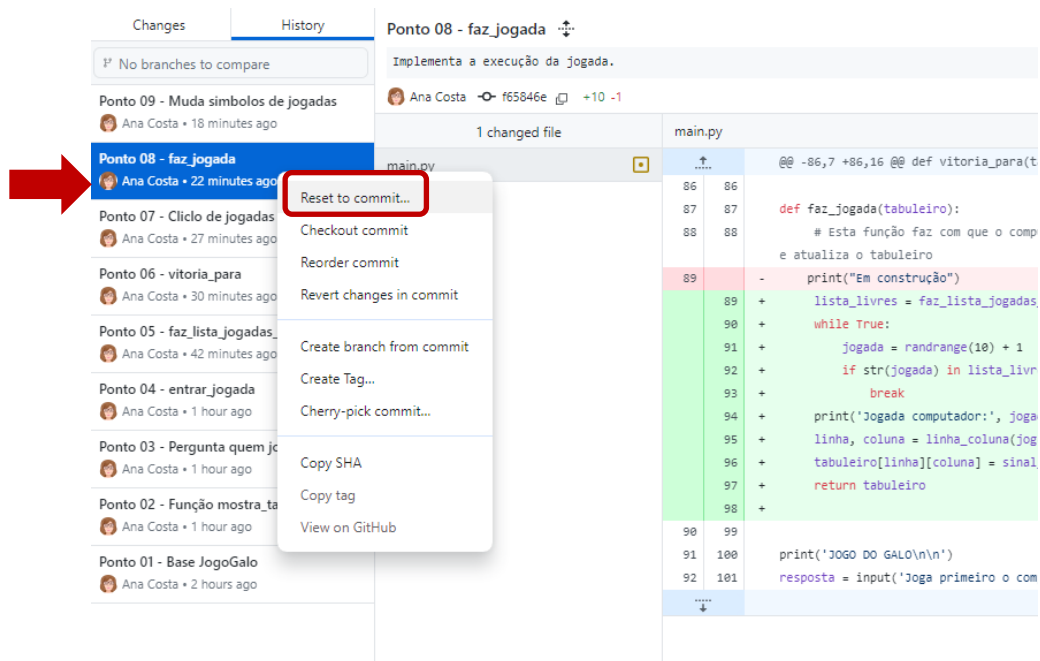
```
@@ -86,7 +86,16 @@ def vitoria_para(tabuleiro, sign):  
86 86  
87 87  
88 88  
89 89  
90 90  
91 91  
92 92  
93 93  
94 94  
95 95  
96 96  
97 97  
98 98  
99 99  
100 100  
101 101
```

Mas no commit 9, o nosso programa deixa de funcionar....



Podemos retornar ao ponto anterior, através de Reset commit.

(mas atenção a esta funcionalidade quando a trabalhar em equipa ou o código já publicado)

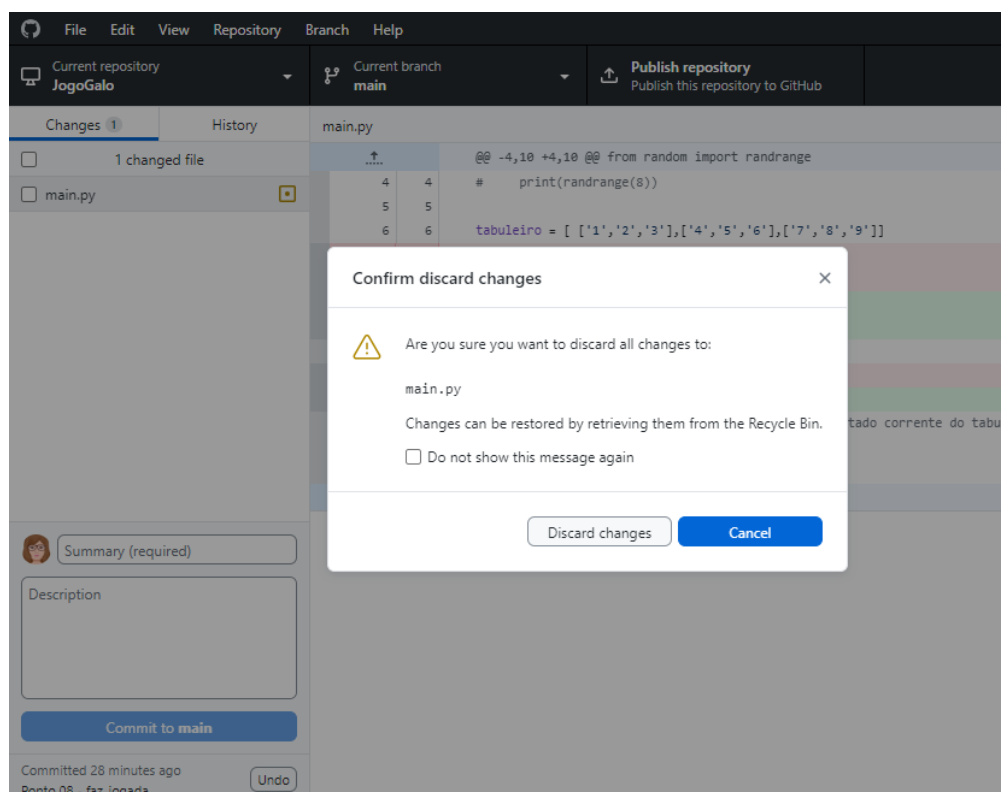
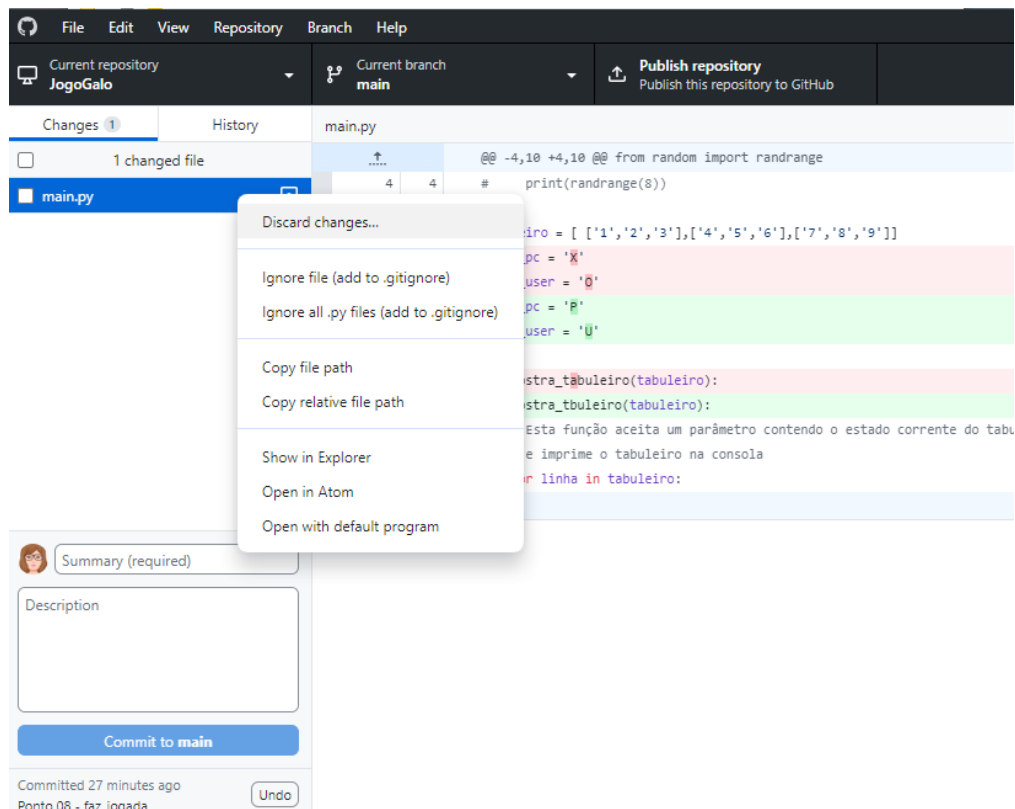


The screenshot shows the PyCharm IDE with the 'Commit History' tab active. The commit list on the left shows 'Ponto 08 - faz\_jogada' by Ana Costa, 9 minutes ago. A context menu is open over this commit, showing options like 'Reset to commit...', 'Checkout commit', 'Reorder commit', 'Revert changes in commit', 'Create branch from commit', 'Create Tag...', 'Cherry-pick commit...', 'Copy SHA', 'Copy tag', and 'View on GitHub'. The main pane shows a diff for 'main.py', highlighting changes in the 'faz\_jogada' function.

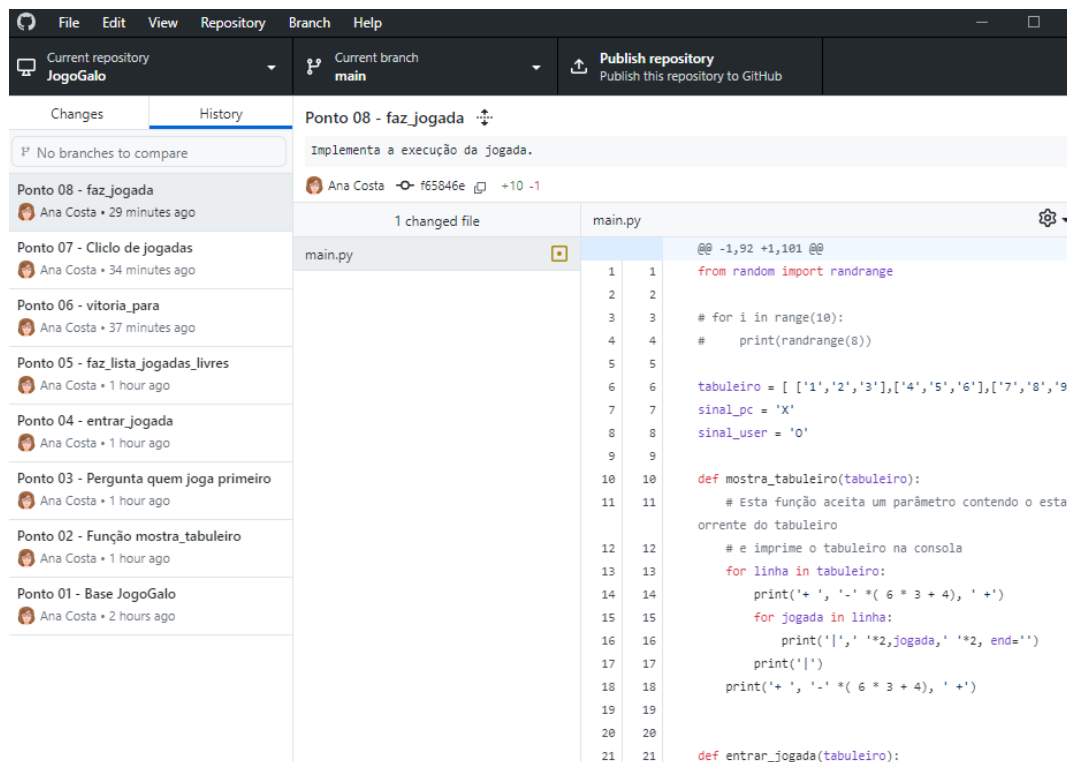
Agora posso descartar as alterações incorretas com opção *discard changes* seleccionando as alterações:

The screenshot shows the PyCharm IDE with the 'Changes' tab active. The 'main.py' file is selected, and the diff view shows changes in the 'mostra\_tabuleiro' function. A commit dialog box is open at the bottom, showing the commit message 'Ponto 08 - faz jogada' and the 'Commit to main' button.





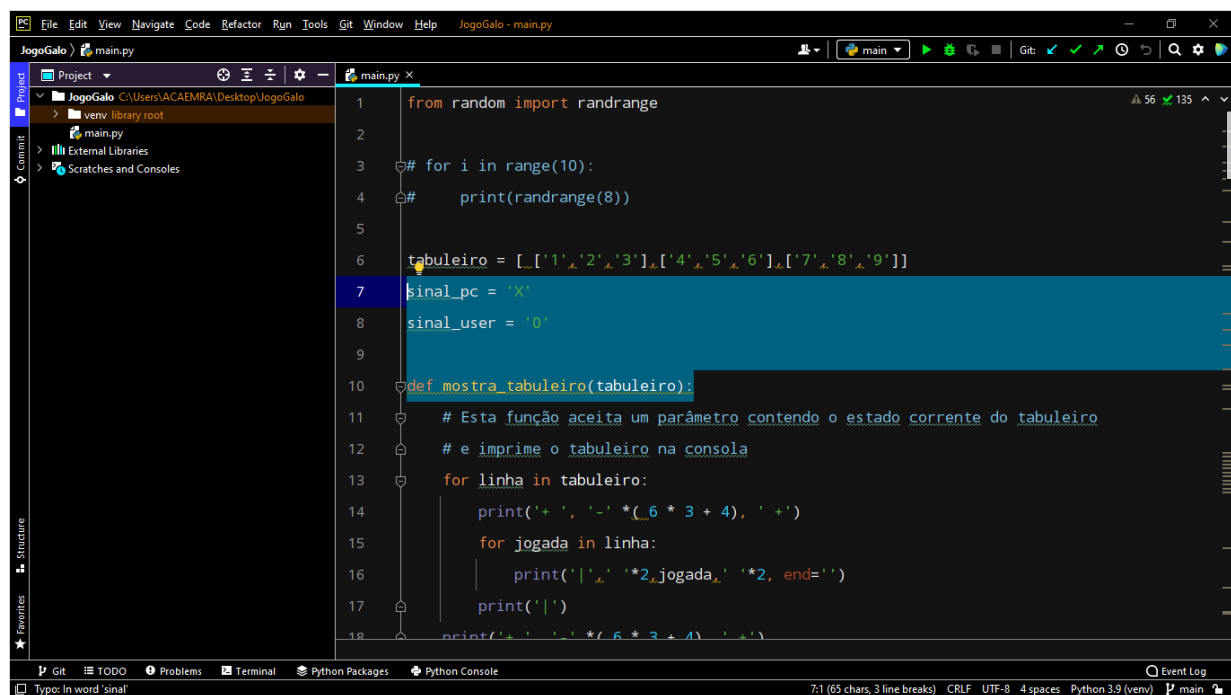
Voltamos ao ponto 8 de commit:



The screenshot shows the GitHub web interface for a repository named 'JogoGalo'. The 'History' tab is selected, showing a list of commits. The commit 'Ponto 08 - faz\_jogada' by Ana Costa is highlighted. The diff for this commit is shown, indicating that the file 'main.py' was changed. The diff shows the addition of code for the 'faz\_jogada' function and the 'mostra\_tabuleiro' function.

```
@@ -1,92 +1,101 @@
1 1  from random import randrange
2 2
3 3  # for i in range(10):
4 4  #     print(randrange(8))
5 5
6 6  tabuleiro = [ ['1','2','3'], ['4','5','6'], ['7','8','9']
7 7  sinal_pc = 'X'
8 8  sinal_user = 'O'
9 9
10 10 def mostra_tabuleiro(tabuleiro):
11 11     # Esta função aceita um parâmetro contendo o estado
    corrente do tabuleiro
    # e imprime o tabuleiro na consola
    for linha in tabuleiro:
14 14         print('+ ', '- ' * ( 6 * 3 + 4), ' +')
15 15         for jogada in linha:
16 16             print('|', ' '*2, jogada, ' '*2, end='')
17 17         print('|')
18 18         print('+ ', '- ' * ( 6 * 3 + 4), ' +')
19 19
20 20
21 21 def entrar_jogada(tabuleiro):
```

E o nosso código *python* também:



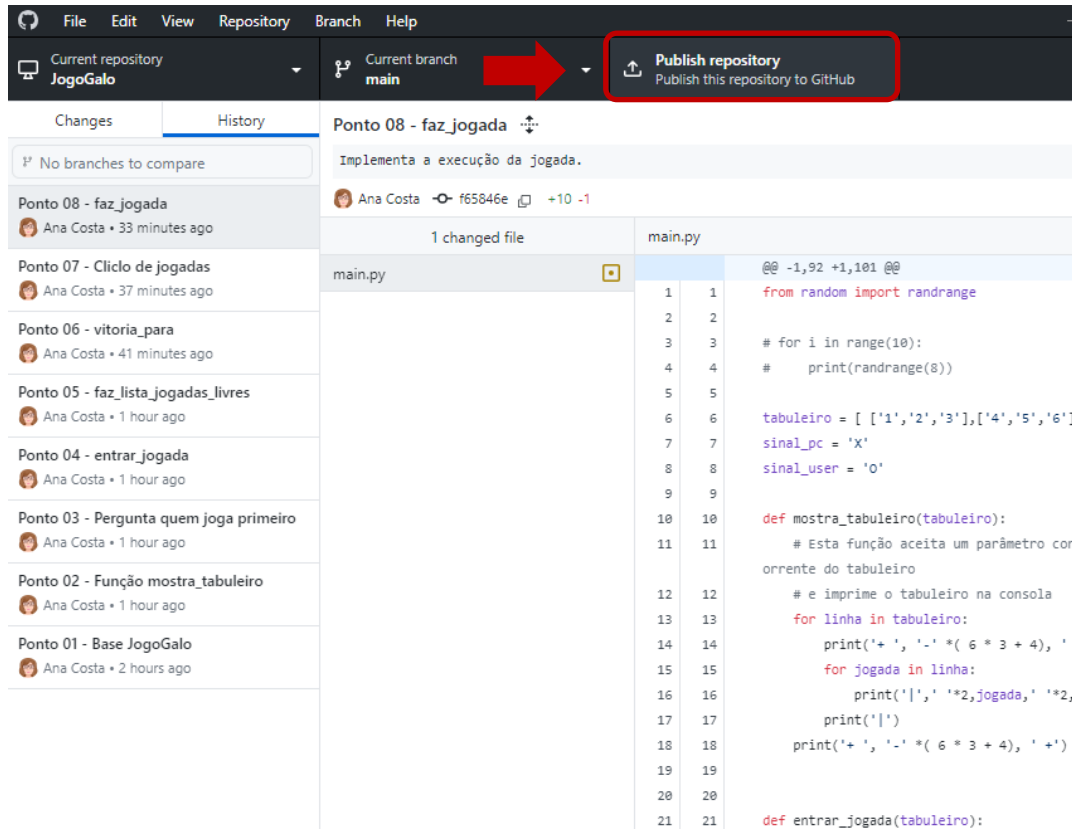
The screenshot shows the PyCharm IDE with the file 'main.py' open. The code is written in Python and includes the same logic as the GitHub diff, including the 'mostra\_tabuleiro' and 'entrar\_jogada' functions.

```
1 from random import randrange
2
3 # for i in range(10):
4 #     print(randrange(8))
5
6 tabuleiro = [['1','2','3'], ['4','5','6'], ['7','8','9']]
7 sinal_pc = 'X'
8 sinal_user = 'O'
9
10 def mostra_tabuleiro(tabuleiro):
11     # Esta função aceita um parâmetro contendo o estado corrente do tabuleiro
12     # e imprime o tabuleiro na consola
13     for linha in tabuleiro:
14         print('+ ', '- ' * ( 6 * 3 + 4), ' +')
15         for jogada in linha:
16             print('|', ' '*2, jogada, ' '*2, end='')
17         print('|')
18         print('+ ', '- ' * ( 6 * 3 + 4), ' +')
```

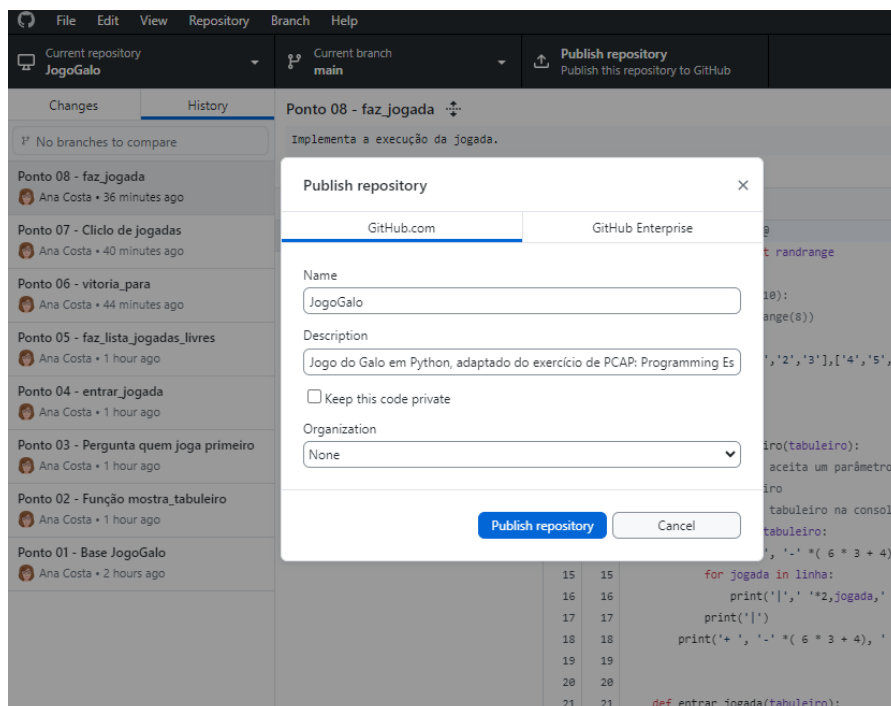
## Enviar o programa para a nossa conta *Git Hub online*:

Agora vamos enviar para a nossa conta GitHub online, para que possamos aceder de qualquer computador e ficarmos com uma cópia na nuvem.

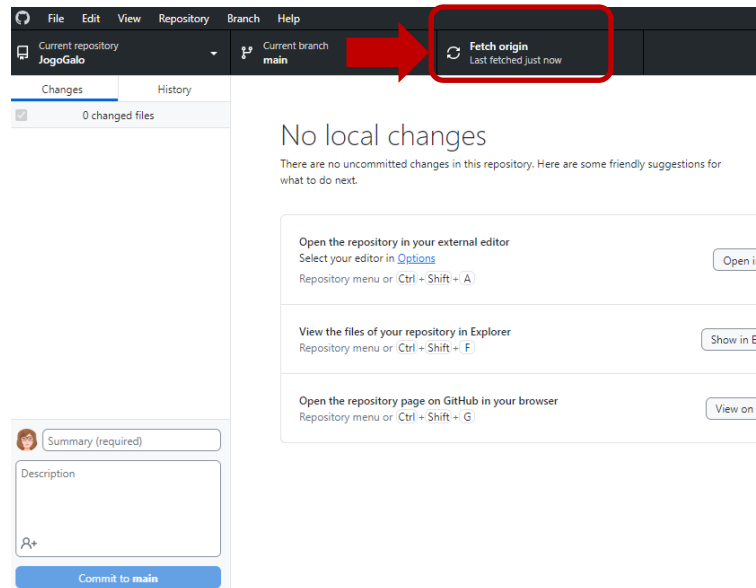
Escolhemos a opção *"Publish repository"*:



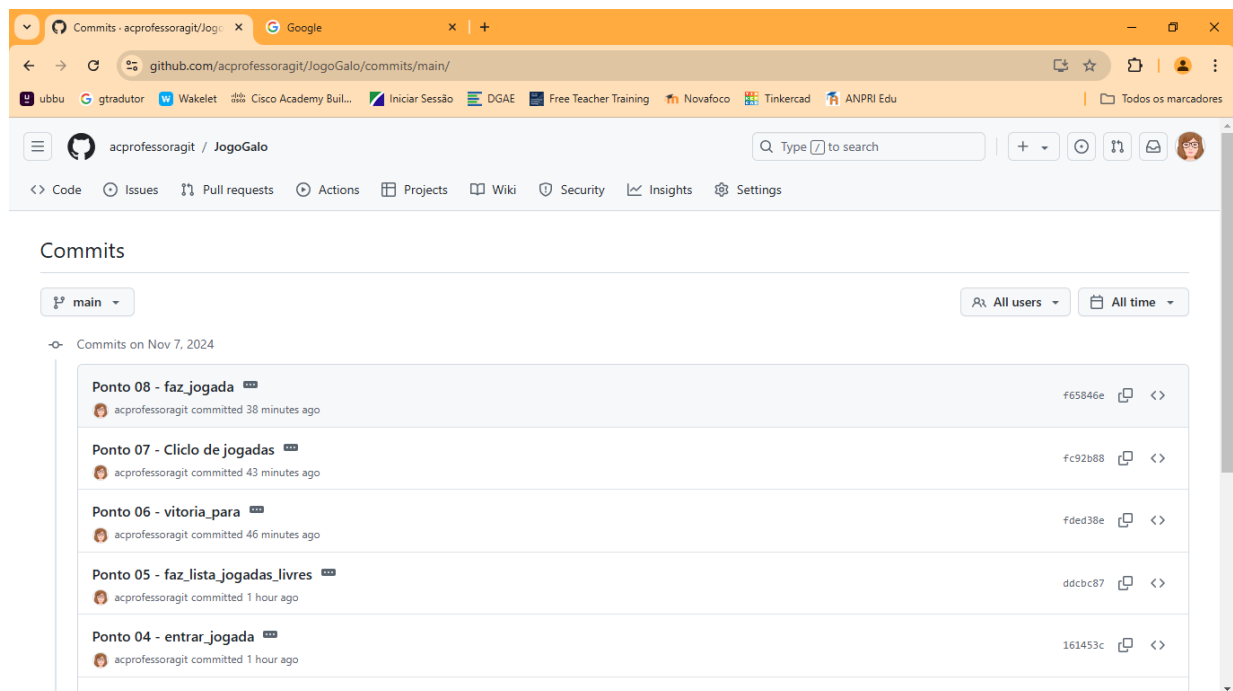
E depois preenchemos os dados:



Notar que agora já temos um novo status, com *"Fetch origin"*:



E online, na nossa conta GitHub, temos:



Sendo agora possível consultar de qualquer computador e se desejarmos descarregar o programa e alterar. E eventualmente trabalhar em equipa com outros programadores.

Existem muitas mais funcionalidades, quer usando a linha de comando, quer através da interface *GitHub*. Mas este exemplo, demonstra o cuidado que devemos ter ao realizarmos o nosso trabalho de programação, pois acidentes podem ocorrer e é sempre bom podermos recuperar o nosso trabalho. Mesmo que o nosso computador avarie, podemos retirar o nosso trabalho do *GitHub* online em:

<https://github.com/acprofessoragit/JogoGalo>