

## PROJETO: Jogo do galo.

### Objetivos

- Aperfeiçoar as competências do aluno na utilização de *Python* para a resolução de problemas complexos.
- Integração de técnicas de programação num programa constituído por várias partes.
- Uso da plataforma Git para gestão de versões.

### Cenário

A sua tarefa é escrever **um programa simples para jogar o jogo do galo com o utilizador**. Para lhe facilitar as coisas, decidimos simplificar o jogo. Aqui estão os nossos pressupostos:

- o computador (ou seja, o seu programa) deve jogar o jogo utilizando 'X's;
- o utilizador (por exemplo, você) deve jogar o jogo utilizando 'O's;
- o primeiro movimento pertence ao computador - coloca sempre o seu primeiro 'X' no meio do tabuleiro;
- todos os quadrados são numerados fila a fila começando por 1 (veja exemplo abaixo, para referência)
- o utilizador introduz a sua jogada inserindo o número do quadrado que escolhe - o número deve ser válido, ou seja, deve ser um número inteiro, deve ser maior do que 0 e menor do que 10, e não pode apontar para um campo que já esteja ocupado;
- o programa verifica se o jogo terminou - há quatro vereditos possíveis: o jogo deve continuar, ou o jogo termina com um empate, a sua vitória ou a vitória do computador;
- o computador responde com a sua jogada e a verificação é repetida;
- não implemente nenhuma forma de inteligência artificial - uma escolha de campo aleatória feita pelo computador é suficientemente boa para o jogo.

A sessão de exemplo com o programa pode ter a aparência seguinte:

```
+-----+-----+-----+
|       |       |       |
|   1   |   2   |   3   |
|       |       |       |
+-----+-----+-----+
|       |       |       |
|   4   |   X   |   6   |
|       |       |       |
+-----+-----+-----+
|       |       |       |
|   7   |   8   |   9   |
|       |       |       |
+-----+-----+-----+
Introduza a sua jogada: 1
```

+-----+-----+-----+			
	0		2
			3
+-----+-----+-----+			
	4		X
			6
+-----+-----+-----+			
	7		8
			9
+-----+-----+-----+			
+-----+-----+-----+			
	0		X
			3
+-----+-----+-----+			
	4		X
			6
+-----+-----+-----+			
	7		8
			9
+-----+-----+-----+			
Introduza a sua jogada: 8			
+-----+-----+-----+			
	0		X
			3
+-----+-----+-----+			
	4		X
			6
+-----+-----+-----+			
	7		0
			9
+-----+-----+-----+			
+-----+-----+-----+			
	0		X
			3
+-----+-----+-----+			
	4		X
			X
+-----+-----+-----+			
	7		0
			9
+-----+-----+-----+			
Introduza a sua jogada: 4			

+-----+-----+-----+			
	O		X
			3
+-----+-----+-----+			
	O		X
			X
+-----+-----+-----+			
	7		O
			9
+-----+-----+-----+			
+-----+-----+-----+			
	O		X
			X
+-----+-----+-----+			
	O		X
			X
+-----+-----+-----+			
	7		O
			9
+-----+-----+-----+			
Introduza a sua jogada: 7			
+-----+-----+-----+			
	O		X
			X
+-----+-----+-----+			
	O		X
			X
+-----+-----+-----+			
	O		O
			9
+-----+-----+-----+			
Ganhou!			

## Requisitos

Implemente as seguintes características:

- o tabuleiro deve ser armazenado como uma lista de três elementos, enquanto cada elemento é outra lista de três elementos (as listas internas representam linhas) de modo que todos os quadrados possam ser acedidos utilizando a seguinte sintaxe:

```
tabuleiro[linha][coluna]
```

- cada um dos elementos da lista interna pode conter 'O', 'X', ou um dígito que representa o número do quadrado (tal quadrado é considerado livre)
- a aparência do tabuleiro deve ser exatamente a mesma que a apresentada no exemplo.
- implemente as funções definidas para si no editor.<sup>1</sup>

O desenho de um número inteiro aleatório pode ser feito utilizando uma função Python chamada `randrange()`. O programa de exemplo abaixo mostra como utilizá-lo (o programa imprime dez números aleatórios de 0 a 8).

Nota: a instrução `from-import` fornece um acesso à função `randrange` definida dentro de um módulo externo de Python chamado `random`.

```
from random import randrange
```

```
for i in range(10):  
    print(randrange(8))
```

*Continua na próxima página...*

---

<sup>1</sup> Adaptado de *PCAP: Programming Essentials in Python*.

```

from random import randrange

# for i in range(10):
#     print(randrange(8))

tabuleiro = [['1','2','3'], ['4','5','6'], ['7','8','9']]

def mostra_tabuleiro(tabuleiro):
    # Esta função aceita um parâmetro contendo o estado
    corrente do tabuleiro
    # e imprime o tabuleiro na consola

def entrar_jogada(tabuleiro):
    # Esta função aceita o o tabuleiro do jogo, pergunta ao
    utilizador acerca da jogada
    # verifica a jogada e atualiza o tabuleiro de acordo com
    a decisão do jogador

def faz_lista_jogadas_livres(tabuleiro):
    # Esta função percorre o tabuleiro e constrói a lista
    de todos os quadrado livres;
    # a lista é constituída por tuplas, e cada tupla é o par
    de números de linha e coluna

def vitoria_para(tabuleiro, sign):
    # Esta função analisa o estado do tabuleiro de modo a
    verificar se
    # o jogador usando '0's ou 'X's ganhou o jogo

def faz_jogada(tabuleiro):
    # Esta função faz com que o computador faça uma jogada
    e atualiza o tabuleiro

```

**Partilhe a ligação Git do seu projeto pela *classroom*.  
Deve criar vários pontos de Commit.**

Bom trabalho 😊