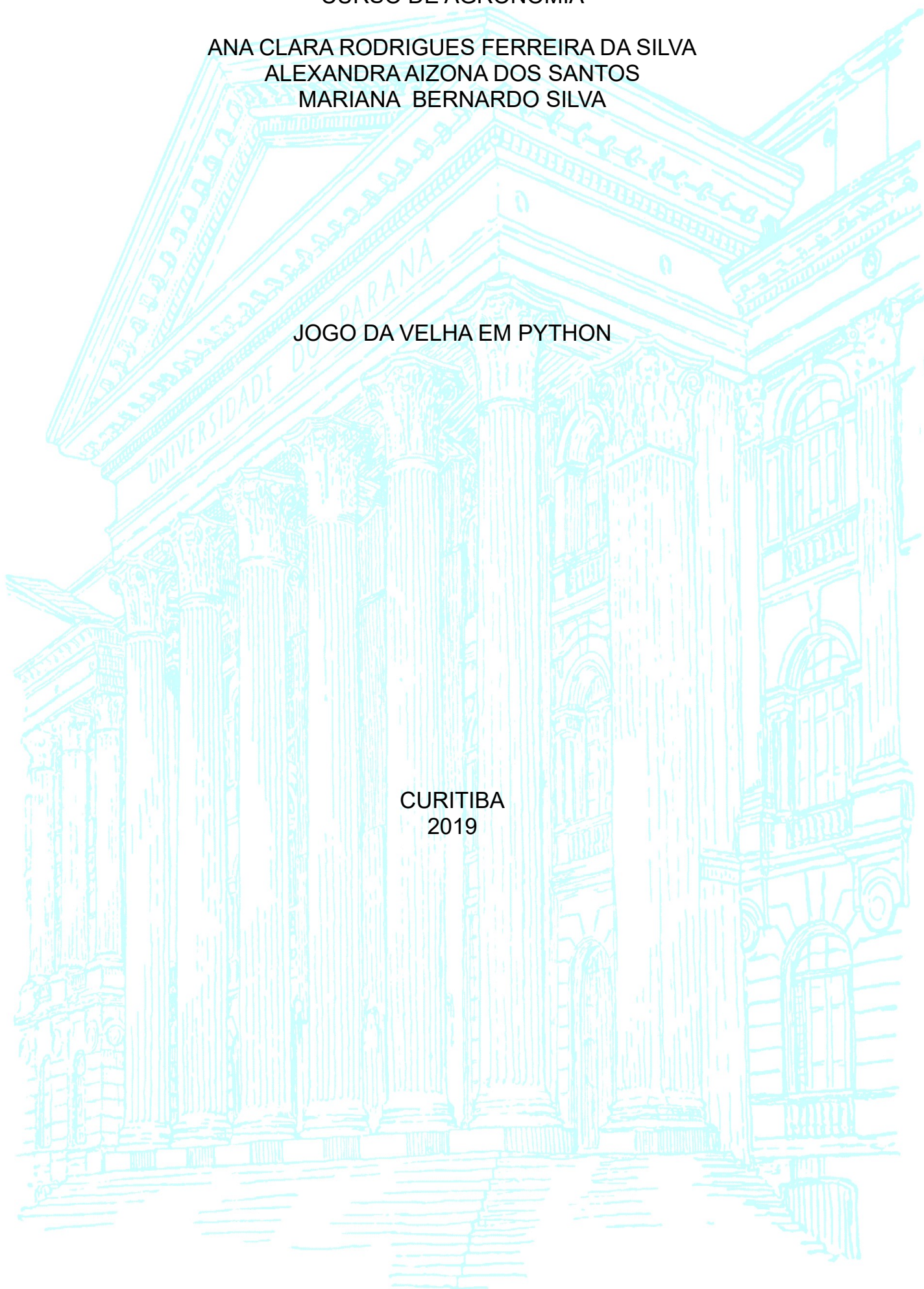


UNIVERSIDADE FEDERAL DO PARANÁ
CURSO DE AGRONOMIA

ANA CLARA RODRIGUES FERREIRA DA SILVA
ALEXANDRA AIZONA DOS SANTOS
MARIANA BERNARDO SILVA

JOGO DA VELHA EM PYTHON

CURITIBA
2019



Ana Clara Rodrigues Ferreira da Silva
Alexandra Aizona dos Santos
Mariana Bernardo Silva

JOGO DA VELHA EM PYTHON

Relatório apresentado à disciplina de Fundamentos da Programação do curso de Graduação de Agronomia da Universidade Federal do Paraná.

Orientador: Prof. Msc. Jackson Antônio do Prado Lima

CURITIBA
2019

lista de figuras

Figura 1:.....6

Figura 2 :.....6

Figura 3.....7

Figura 4.....7

Figura 5.....7

Figura 6.....8

Figura 7.....8

Figura 8.....8

Figura 9.....9

Sumário

1 INTRODUÇÃO.....	5
1.1 JUSTIFICATIVA.....	5
1.2 OBJETIVOS.....	5
2 DESENVOLVIMENTO.....	6
2.1 O PROGRAMA.....	6
2.2 O CÓDIGO.....	6
3 CONSIDERAÇÕES FINAIS.....	10
REFERÊNCIAS.....	11

1 INTRODUÇÃO

O relatório descreve em detalhes informações relativas ao programa realizado em Python, linguagem de programação de alto nível criada por Guido Van Rossum em 1989. O programa visa desenvolver um dos mais populares jogos, Jogo da Velha.

1.1 JUSTIFICATIVA

É uma forma de aprofundar e aplicar os conhecimentos desenvolvidos durante o semestre.

1.2 OBJETIVOS

O objetivo do trabalho é mostrar o quanto programação pode ser útil nas mais diversas áreas, incluindo na de entretenimento. Baseado nisso, buscamos criar um Jogo da Velha programado na linguagem Python.

2 DESENVOLVIMENTO

2.1 O PROGRAMA

Durante a construção do código não foi feito o uso de nenhuma biblioteca. O programa fundamentou-se a partir de matriz, funções e condicionais.

2.2 O CÓDIGO

O código começa com a própria matriz, a qual é a base do jogo, variáveis das quais serão usadas ao decorrer do programa e dois prints, um de apresentação e outro de regras.

Figura 1:

```
m=[["-","-","-"],["-","-","-"],["-","-","-"]]
ind=1
gx=0
go=0
empate=0
print('=== JOGO DA VELHA === ')
print("OBS:As linhas são de cima para baixo e colunas são da esquerda para direita. Todas começam de 1 e terminam em 3\n")
```

A função jogador definirá a vez de cada jogador, sendo par a jogada, será vez de “x”, e ímpar, “o”.

Figura 2 :

```
#algoritmo de indentificação
def jogador(ind):
    if ind%2==0:
        ind="x"
        return "x"
    if ind%2==1:
        ind="o"
        return "o"

#jogador x
if jogador(ind)=="x":
    print("<=== Vez do jogador dos x ===>")

#jogador o
elif jogador(ind)=="o":
    print("<=== Vez do jogador dos o ===>")
```

A função vitória, determina todas as vitórias possíveis, se um dos jogadores realizarem uma delas, o ganhador vence a partida.

Figura 3

```

#algoritmo que avalia vitória
def vitória(linha,coluna):
    if m[linha-1][0]==jogador(ind) and m[linha-1][1]==jogador(ind) and m[linha-1][2]==jogador(ind):
        print("<::: O jogador dos ",jogador(ind)," GANHOU!! :::>")
        for i in m:
            print(i)
        return vitória

    elif m[0][coluna-1]==jogador(ind) and m[1][coluna-1]==jogador(ind) and m[2][coluna-1]==jogador(ind):
        print("<::: O jogador dos",jogador(ind)," GANHOU!! :::>")
        for i in m:
            print(i)
        return vitória

    elif m[0][0]==jogador(ind) and m[1][1]==jogador(ind) and m[2][2]==jogador(ind):
        print("<::: O jogador dos ",jogador(ind)," GANHOU!! :::>")
        for i in m:
            print(i)
        return vitória

    elif m[2][0]==jogador(ind) and m[1][1]==jogador(ind) and m[0][2]==jogador(ind):
        print("<::: O jogador dos ",jogador(ind)," GANHOU!! :::>")
        for i in m:
            print(i)
        return vitória

```

A função conclusão será necessária para avaliar se a jogada é válida. Linhas e colunas deverão ser escolhidas entre 1 a 3, e não será possível escolher uma posição já ocupada. Caso a coordenada esteja dentro das especificações, o jogo continuará.

Figura 4

```

#função de erro
def conclusão(linha,coluna):
    if linha<1 or linha>3 or coluna<1 or coluna>3:
        print("::: <-Erro->Você digitou algum número fora do limite. Digite novamente :::")
        pergunta()

    elif m[linha-1][coluna-1]==jogador(ind) or m[linha-1][coluna-1]==jogador(ind+1):
        print("::: <-Erro->Você digitou uma posição já existente. Digite novamente :::")
        pergunta()

    else:
        m[linha-1][coluna-1]=jogador(ind)
        if vitória(linha,coluna)==vitória:
            return True
        return False

while True:
    while True :
        ind+=1
        x=False

```

A função pergunta, será usada a cada jogada, para os dois jogares, na ordem demonstrada.

Figura 5

```

#função da pergunta
def pergunta():
    for i in m:
        print(i)
    a=int(input("Qual a linha voce escolhe? "))
    b=int(input("Qual coluna voce escolhe? "))
    if conclusão(a,b)==True:
        return True
    return False

```

Caso o programa detecte o empate da partida, será acrescentado um empate ao saldo do jogo. Uma mensagem de empate é mostrada.

Figura 6

```
#algoritmo que avalia empate ou leva para funcao
if pergunta()==True:
    if jogador(ind)=="x":
        gx+=1
    elif jogador(ind)=="o":
        go+=1
    break
elif ind==10:
    print("#=== DEU VELHA! EMPATOU!! ===#")
    for i in m:
        print(i)
    empate+=1
    break
```

Se no final da partida, tiver um vencedor, será acrescentado nas vitórias do participante. Em seguida, aparecerá a opção de jogar novamente.

Figura 7

```
#algoritmo que avalia vitoria
if pergunta()==True:
    if jogador(ind)=="x":
        gx+=1
    elif jogador(ind)=="o":
        go+=1
    break

x=input("::: Vocês querem jogar novamente?(s/n) :::")
```

O programa apenas irá aceitar “s” se desejar continuar jogando e “n”, se caso não, qualquer letra ou número diferente digitado, fará com que apareça uma mensagem de erro, e logo em seguida a pergunta novamente.

Figura 8

```
#algoritmo de erro de letra invalida
while x!="s" and x!="n":
    print("Letra inválida, digite novamente")
    x=input("::: Vocês querem jogar novamente?(s/n) :::")
```

Tendo uma ou mais vitórias de um dos jogares, ou tendo como finalização do jogo, um empate, o código imediatamente dará a opção dos jogares disputarem mais uma partida. Caso sim, aparecerá novamente a matriz, se não, por meio de prints, aparecerá o saldo do jogo.

Figura 9

```
#algoritmo de confirmação
if x=="s":
    x=True
    m=[["-", "-", "-"], ["-", "-", "-"], ["-", "-", "-"]]
    ind=1
else:
    break

print("< FIM DO JOGO! >")
print("<===== Dados do jogo =====>")
print("O jogador x ganhou: ",gx)
print("O jogador o ganhou: ",go)
print("Empate: ",empate)

sair=input("Aperte o botão ENTER para sair")
```

3 CONSIDERAÇÕES FINAIS

Neste trabalho foi possível aplicar e integrar o conhecimento adquirido ao longo do semestre. Embora o jogo criado possa ser considerado simples, é possível perceber o uso de diversas funções e ferramentas do Python.

Por fim, percebemos a importância da programação e de seus conceitos, não apenas como uma disciplina, mas como uma ferramenta capaz de facilitar as mais diversas atividades no cotidiano.

REFERÊNCIAS

Labaki, Josue – Introdução a Python – Disponível em:

<chrome-extension://oemmndcbldboiebfnladdacbfdmadadm/https://www.dcc.ufrj.br/~fabiom/mab225/pythonbasico.pdf>

Python Documentation – Disponível em:

<https://docs.python.org/3/>