

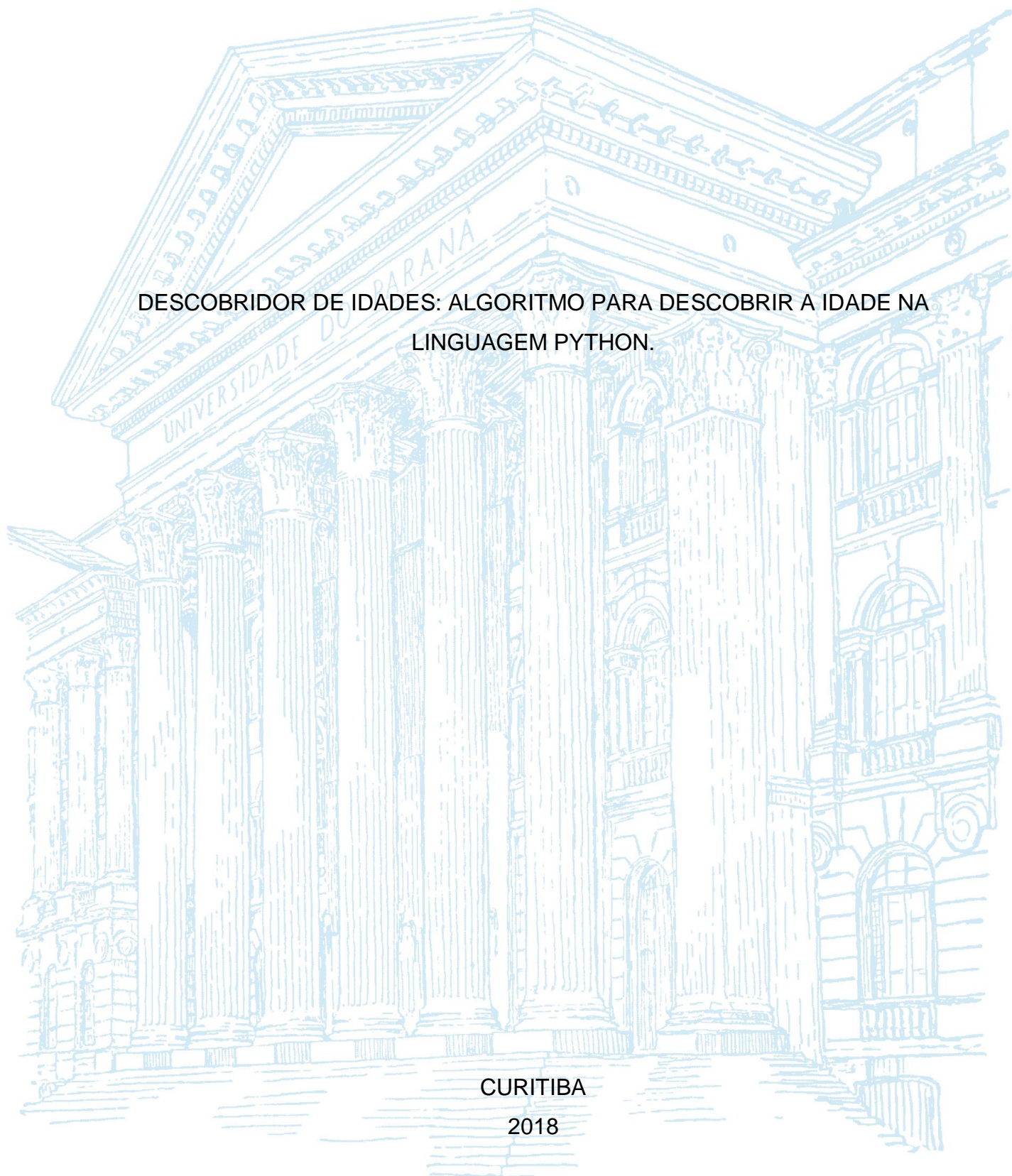
UNIVERSIDADE FEDERAL DO PARANÁ

JAQUELINE CHEZANOSKI
MATHEUS KINCESKI PIRES

DESCOBRIDOR DE IDADES: ALGORITMO PARA DESCOBRIR A IDADE NA
LINGUAGEM PYTHON.

CURITIBA

2018



JACQUELINE CHEZANOSKI (GRR 20170406)
MATHEUS KINCESKI PIRES (GRR 20185661)

DESCOBRIDOR DE IDADES: ALGORITMO PARA DESCOBRIR A IDADE NA
LINGUAGEM PYTHON.

Relatório apresentado à disciplina de Fundamento
da Programação em Computadores do curso de
Graduação em Matemática da Universidade
Federal do Paraná

Orientador: Prof. Jackson Antônio do Prado Lima

CURITIBA
2018

SUMÁRIO

1	INTRODUÇÃO	4
2	OBJETIVOS	5
3	DESENVOLVIMENTO	6
4	CONCLUSÃO	9
	REFERÊNCIAS.....	10

1 INTRODUÇÃO

Esse relatório apresenta informações relacionadas ao trabalho realizado em Python, que coloca em prática o conteúdo aprendido no semestre na matéria de Fundamentos da Programação em Computadores.

O Python é uma linguagem de alto nível, onde uma de suas maiores características é uma fácil leitura do código e a utilização de poucas linhas, se comparado a outras linguagens. Foi criada em 1989, pelo holandês Guido van Rossum, e atualmente se encontra na versão 3.7.0.

2 OBJETIVO

O objetivo geral é criar um programa em Python com base em uma brincadeira de “adivinhar” a idade do usuário, utilizando o conteúdo aprendido na disciplina de Fundamentos da Programação em Computadores.

3 DESENVOLVIMENTO

3.1 O Trabalho

Nosso programa conta com uma função, que tem como objetivo imprimir as matrizes dadas na tela do usuário.

```
def imprime_matriz(m):
    #Essa função ira imprimir as matrizes necessárias para a brincadeira acontecer.
    linha = len(m) # Número de linhas da matriz m
    coluna = len(m[0]) # Número de colunas da matriz m
    # Nas próximas linhas iremos percorrer todos os elementos da matriz para podermos imprimi-los em tela.
    for i in range(linha):
        for j in range(coluna):
            print(m[i][j], " ", end="")
        print("\n")
```

Figura 1 - Função para imprimir matrizes.

Definimos *imprime_matriz*, recebendo um dado externo, que é uma matriz, para fazer uma operação. Declaramos a variável *linha* e *coluna*, onde, respectivamente, é a quantidade de linhas, dado por *len(m)*, e a quantidade de colunas, dado por *len(m[0])*.

Em um laço for que vai de 0 até o número de linhas, com outro laço for que vai de 0 até o número de colunas, vamos de elemento por elemento da matriz e imprimimos na tela com o comando da penúltima linha, deixando um espaço entre cada elemento com o comando *end=""*. Após imprimir os elementos da primeira linha, isto é, com o fim do primeiro laço *for* das colunas, fazemos com que a próxima linha a ser impressa vá para uma linha abaixo da primeira já imprimida, com o comando *print("\n")*. Com esse comando feito, vamos continuar o primeiro laço for, que vai continuar e imprimir a segunda linha, terceira linha, até acabarem as linhas e finalizar o comando da função.

Em seguida optamos por apresentar ao usuário uma mensagem inicial, instruindo o que ele deverá fazer enquanto utiliza o programa, para que se tenha um maior proveito do mesmo. Explica-se que as únicas respostas devem ser "s" para sim e "n" para não.

```
# Mensagem inicial
print('''Olá usuário, vamos adivinhar a sua idade.
Para isso, responda apenas 's' para sim ou 'n' para não e tecle enter.
Preste muita atenção para não errar e deixar passar algum número!''')
```

Figura 2 - Mensagem inicial

Definimos as matrizes que serão utilizadas. Seus números serão as possíveis idades do usuário. O primeiro elemento de cada matriz é uma potência de 2. Por exemplo, na matriz m1 temos "2⁰" e na m4 "2³". Podemos escrever qualquer número que aparece nas matrizes como a soma dessas potências, novamente exemplificando, podemos escrever o número 27 como $(1) * 2^0 + (1) * 2^1 + (0) * 2^2 + (1) * 2^3 + (1) * 2^4 + (0) * 2^5 = 1 + 2 + 8 + 16$. Podemos, também, escrever essa combinação em forma de um número binário, no caso do 27 será 11011, que são os números que aparecem multiplicando as potências.

```
# A seguir, iremos denotar as matrizes que serão utilizadas
m1 = [[1, 3, 5, 7, 9, 11, 13, 15], [17, 19, 21, 23, 25, 27, 29, 31], [33, 35, 37, 39, 41, 43, 45, 47], [49, 51, 53, 55, 57, 59, 61, 63]]
m2 = [[2, 3, 6, 7, 10, 11, 14, 15], [18, 19, 22, 23, 26, 27, 30, 31], [34, 35, 38, 39, 42, 43, 46, 47], [50, 51, 54, 55, 58, 59, 62, 63]]
m3 = [[4, 5, 6, 7, 12, 13, 14, 15], [20, 21, 22, 23, 28, 29, 30, 31], [36, 37, 38, 39, 44, 45, 46, 47], [52, 53, 54, 55, 60, 61, 62, 63]]
m4 = [[8, 9, 10, 11, 12, 13, 14, 15], [24, 25, 26, 27, 28, 29, 30, 31], [40, 41, 42, 43, 44, 45, 46, 47], [56, 57, 58, 59, 60, 61, 62, 63]]
m5 = [[16, 17, 18, 19, 20, 21, 22, 23], [24, 25, 26, 27, 28, 29, 30, 31], [48, 49, 50, 51, 52, 53, 54, 55], [56, 57, 58, 59, 60, 61, 62, 63]]
m6 = [[32, 33, 34, 35, 36, 37, 38, 39], [40, 41, 42, 43, 44, 45, 46, 47], [48, 49, 50, 51, 52, 53, 54, 55], [56, 57, 58, 59, 60, 61, 62, 63]]
```

Figura 3 - Matrizes

Definimos a variável que terá o controle sobre nosso primeiro laço de repetição. Esse laço é o que controlará o reinício do jogo, caso essa seja a vontade do usuário ao final do programa.

```
answer_end="s" # Esse será nosso controle para entrarmos no laço de repetição e finalizarmos o programa no futuro
while answer_end!="n" or answer_end!="s": # laço de repetição para reiniciar a brincadeira.
```

Figura 4 - Laço de Repetição

Declaramos a variável "soma" como 0, será onde somaremos os números que irão compor a idade do usuário, para podermos imprimi-la ao final, antes de encerramos ou repetirmos o primeiro laço.

Começamos, então, um processo que se repetirá com as seis matrizes do programa, uma a uma. Cada matriz será impressa na tela do usuário utilizando a função definida como *imprime_matriz(m)*, logo aparece ao usuário uma mensagem o questionando se a idade dele se encontra na matriz apresentada. Caso o usuário digite algo diferente de "s" ou "n" (comandos explicados ao início), ele acabará entrando em um novo laço de repetição, que irá mostrar novamente uma mensagem explicando as respostas que são aceitas. O usuário sairá desse laço apenas quando sua resposta for uma das desejadas.

Caso a resposta seja "s", o programa irá passar pela condição dada, o elemento que corresponde ao [0][0] da matriz será adicionado a variável "soma" e o programa irá executar o mesmo processo com a próxima matriz. Caso seja "n", apenas passa-se para a próxima matriz.

```
soma=0 # Variável para a idade
imprime_matriz(m1)
# Imprimimos a primeira matriz para o usuário
answer = input("Sua idade está na tabela acima? ")
''' O usuário deverá procurar o número correspondente a sua idade na matriz
    Caso o usuário digite algo diferente de "s" ou "n", ele acabará vindo para esse laço'''
while not (answer=="s" or answer=="n"):
    print("Por favor, responda a pergunta com s ou n.\n")
    # Onde repassamos a informação de que as respostas devem ser da forma que desejamos
    answer=input("Sua idade está na tabela? ")
# caso a resposta seja "s", iremos adicionar à variável "soma" o elemento que corresponde a posição [0][0] da matriz
if answer=="s":
    soma+=1
print("\n\n")
# caso a resposta seja "n", nada acontece
```

Figura 5 - Apresentação das matrizes

Após repetirmos o processo anterior com todas as matrizes, mostramos ao usuário uma mensagem que contém a variável *soma*, apresentada como sendo sua idade.

```
print("Você tem {0} anos".format(soma))
'''
```

Figura 6 - Apresentação da idade

Por último, damos a opção de poder jogar novamente ou não. Caso a resposta seja diferente das desejadas, caímos novamente em um laço de repetição explicando as respostas desejadas. Caso seja "s", retornamos ao primeiro laço. Caso seja "n", apresentamos uma mensagem de agradecimento e encerramos o programa.

```
answer_end=input("Deseja fazer novamente? ")
while not (answer_end=="s" or answer_end=="n"): #respostas diferentes são redirecionadas
    print("Por favor, responda a pergunta com s ou n.\n")
    answer_end=input("Deseja fazer novamente? ")

print("Agradecemos por usar o nosso programa.") # Caso seja "n", encerramos o programa.
```

Figura 7 - Parte final

3.2 Dificuldades

Durante o processo de desenvolvimento do trabalho, encontramos algumas dificuldades. A ideia inicial era aplicar este programa em um 'bot' no Telegram (aplicativo de mensagens), mas não teríamos condições de deixar as matrizes visualmente bem proporcionadas, deixando o programa com uma aparência estranha e que dificultaria a utilização por parte do usuário.

4 CONCLUSÃO

Apesar das dificuldades encontradas, acreditamos que esse trabalho tenha nos agregado muito, pois a partir dele, revisamos e aplicamos o conteúdo do semestre, além de termos aprendido um pouco sobre números binários e como começar um *'bot'* no Telegram, apesar de não termos conseguido aplicá-lo à nossa ideia. É interessante descobrirmos como, com coisas simples do Python, podemos solucionar grandes problemas em poucas linhas ou então apenas nos divertimos com programas simples, mas com uma linha de pensamento e lógica muito bacana por trás deles.

REFERÊNCIAS

MANUAL DO MUNDO comunicação. Disponível em <<http://www.manualdomundo.com.br/2012/07/adivinhacao-surpreendente-do-numero-usando-cartelas/>>. Acesso em 24 de nov. de 2018.

UNIVERSIDADE FEDERAL DO PARANA (UFPR). Sistema de Bibliotecas. Disponível em <<https://portal.ufpr.br/normalizacao>>. Acesso em: 24 de nov. de 2018.