Descrição: Estácio

**UNIVERSIDADE ESTÁCIO DE SÁ**

**MBA EM CIÊNCIA DE DADOS E BIG DATA ANALYTICS**

**Trabalho**

**Aluno: Daniel Mendes Guedes**

**Trabalho da disciplina**

**PRÁTICA E LABORATÓRIO II**

**Tutor: Prof. André Luiz Braga**

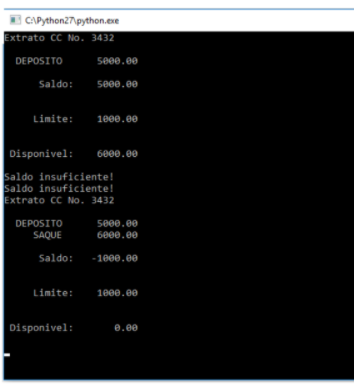
**Brasília/DF**

**2021**

**Trabalho a ser desenvolvido:**

FAÇA UM PROGRAMA ORIENTADO A OBJETOS QUE PERMITA AO USUÁRIO CRIAR UMA CLASSE CLIENTE (NOME E TELEFONE) E UMA CLASSE CONTA (COM NUMERO E SALDO), LOGICAMENTE VINCULADA AO CLIENTE, QUE PERMITA SIMULAR DUAS OPERAÇÕES (SAQUE E EXTRATO, SIMPLES). DEVE SER ADMITIDA A MANIPULAÇÃO DE UMA CONTA ESPECIAL AGREGANDO A ESTA O ATRIBUTO LIMITE. NO FINAL SIMULE A LISTAGEM DE EXTRATO BANCÁRIO, COM BASE NA SEGUINTE FIGURA (AS INFORMAÇÕES DO EXTRATO

BANCÁRIO DEVEM SER AS MESMAS QUE ESTÃO SENDO MOSTRADAS):



**Resolução:**

A linguagem, orientada a objeto, utilizada para a construção do sistema foi a linguagem Python. A construção foi realizada através do ferramenta *Colaboratory* que permite escrever o código pelo navegador sem nenhum configuração necessária e permite fácil compartilhamento.

***SCRIPT***

# Cria Classe Cliente(Nome e Telefone)

class Cliente:

def \_\_init\_\_(self, nome, telefone):

self.nome = nome

self.telefone = telefone

# Cria Classe Conta (Número e Saldo), vinculada ao Cliente

# Cria simulação de duas Operações (SAQUE E EXTRATO, SIMPLES).

class Conta:

def \_\_init\_\_(self, clientes, número, saldo = 0):

self.saldo = 0

self.clientes = clientes

self.número = número

self.operações = []

self.deposito(saldo)

def resumo(self):

print("CC N°%s Saldo: %10.2f" %

(self.número, self.saldo))

def pode\_sacar(self, valor):

return self.saldo >= valor

def saque(self, valor):

if self.pode\_sacar(valor):

self.saldo -= valor

self.operações.append(["SAQUE", valor])

return True

else:

print("Saldo insuficiente!")

return False

def deposito(self, valor):

self.saldo += valor

self.operações.append(["DEPÓSITO", valor])

def extrato(self):

print("Extrato CC N° %s\n" % self.número)

for o in self.operações:

print("%10s %10.2f" % (o[0],o[1]))

print("\n Saldo: %10.2f\n" % self.saldo)

# Criar Conta Especial com Limite

class ContaEspecial(Conta):

def \_\_init\_\_(self, clientes, número, saldo = 0, limite=0):

Conta.\_\_init\_\_(self, clientes, número, saldo)

self.limite = limite

def pode\_sacar(self, valor):

return self.saldo + self.limite >= valor

def extrato(self):

Conta.extrato(self)

print("\n Limite: %10.2f\n" % self.limite)

print("\n Disponivel: %10.2f\n" % (self.limite + self.saldo))

# Simulação de Listagem de Extrato Bancário solicitada

Daniel = Cliente("Daniel", "92000-5966")

conta = ContaEspecial([Daniel], 3432, 5000, 1000)

conta.extrato()

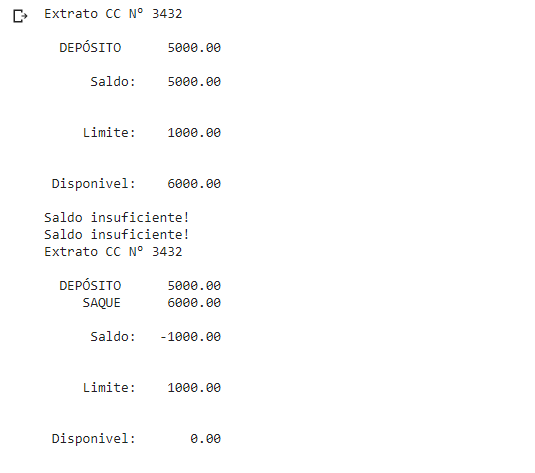
conta.saque(6000)

conta.saque(3000)

conta.saque(1000)

conta.extrato()

**Resultado de Saída**



*Colaboratory*: <https://colab.research.google.com/drive/1nwwgCN4lMnKOh-Ea5zgSf-CQE9vSHFvO?usp=sharing>

Repositório Github: