

```
'''Pontifícia Universidade Católica do Paraná  
  
Disciplina: Segurança da Tecnologia da Informação (Turma 01)  
  
Professor-Tutor: Wellington Rodrigo Monteiro  
  
Alunos: BÁRBARA DE PAULA E SILVA  
        DAVI RAMON GONÇALVES  
        MARCOS ALARCON  
  
Grupo: 87  
  
Atividade: Atividade Somativa 1  
  
Título: Desafio Autenticação e Controle de Acesso no banco de Tóquio'''
```

```
import Autentica  
  
while True:  
    print("1 - Cadastrar Usuário")  
    print("2 - Verificar Email")  
    print("3 - Autenticar Usuário")  
    print("4 - Sair\n")  
  
    opcao = input("Selecione uma opção: ")  
  
    # cadastrar um novo usuário  
    if opcao == "1":  
        Autentica.cadastrar()  
  
    # verificar o email  
    elif opcao == "2":  
        Autentica.verificar_email()  
  
    # autenticar um usuário  
    elif opcao == "3":  
        Autentica.autenticar()  
  
    # sair do programa  
    elif opcao == "4":  
        break # Sai do programa
```

```

import pyrebase
import random
import Email
import acesso

firebaseConfig = {
    "apiKey": "AIzaSyAhpYlI3dA7PQ5GKxv5FaQv0FaZs3hAS80",
    "authDomain": "fir-pucpr-4d42a.firebaseio.com",
    "projectId": "fir-pucpr-4d42a",
    "databaseURL": "https://" + "fir-pucpr-4d42a" + ".firebaseio.com",
    "storageBucket": "fir-pucpr-4d42a.appspot.com",
    "messagingSenderId": "454293494879",
    "appId": "1:454293494879:web:f99cb8da933fc2a5d9cfea",
    "measurementId": "G-CX58JLRCD"
}

# função para criar um novo usuário
def cadastrar():
    firebase = pyrebase.initialize_app(firebaseConfig)
    auth = firebase.auth()

    user = input("Digite seu e-mail: ")
    password = input("Digite sua senha, com pelo menos 6 caracteres: ")

    # Se a senha tiver pelo menos 6 caracteres
    if len(password) >= 6:
        status = auth.create_user_with_email_and_password(user, password)
        if status['idToken']:
            print("\nUsuário:", user+" cadastrado com sucesso!")
        else:
            print("\nErro ao cadastrar usuário!")
    else:
        print("\nSenha deve ter pelo menos 6 caracteres!")

# função para logar um usuário
def autenticar():
    firebase = pyrebase.initialize_app(firebaseConfig)
    auth = firebase.auth()

    user = input("Informe seu e-mail de acesso: ")
    password = input("Informe sua senha: ")
    status = auth.sign_in_with_email_and_password(user, password)
    idToken = status['idToken']
    info = auth.get_account_info(idToken)
    users = info["users"]
    verifyEmail = users[0]["emailVerified"]

    # Se o email foi verificado
    if verifyEmail:
        print("\nAutenticação em dois fatores")
        codigo = random.randint(100, 1000)

        Email.enviar_email(codigo, user)

        codigoEmail = int(input("\nInforme o código enviado para o seu e-mail: "))

        if codigo == codigoEmail:
            print("\nUsuário Autenticado com Sucesso!!!")
            acesso.executarcontroleacesso(user)
        else:
            print("\nCódigo Inválido!!\n")
    else:
        print("\nE-mail não verificado!\n")

# função para enviar um email de verificação
def verificar_email():
    firebase = pyrebase.initialize_app(firebaseConfig)
    auth = firebase.auth()

    user = input("Digite seu e-mail: ")
    password = input("Digite sua senha: ")
    status = auth.sign_in_with_email_and_password(user, password)
    idToken = status['idToken']
    auth.send_email_verification(idToken)

```

```
print("\nEmail de verificação enviado para: ", user)
```

# Arquivo Email.py

```
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

server = "smtp.gmail.com"
port = 587
username = "omega.marcos@gmail.com"
password = "wmfylvtpowarwepto"

# função para enviar um email com o código de verificação
def enviar_email(code, email):
    mail_from = "omega.marcos@gmail.com"
    mail_to = email
    mail_subject = "Código E-mail"
    mail_body = "Código de validação: %d " % code

    msg = MIMEMultipart()
    msg['From'] = mail_from
    msg['To'] = mail_to
    msg['Subject'] = mail_subject
    msg.attach(MIMEText(mail_body, 'plain'))

    connection = smtplib.SMTP(server, port)
    connection.starttls()
    connection.login(username, password)
    connection.send_message(msg)
    connection.quit() # Fecha a conexão
```

```
import datetime
import os
import stat

def executarcontroleacesso(usuario):

    if os.path.isfile('acesso.txt'):
        # Dá Permissão de leitura, escrita e execução para o dono do arquivo
        os.chmod("acesso.txt", stat.S_IRWXU)

        # Abre o arquivo para adicionar as informações de acesso
        arquivo = open("acesso.txt", "w")

        # Adiciona as informações de acesso
        arquivo.write("\nUsuário: " + usuario + " - Data: " + str(datetime.datetime.now()))

    else:
        # Criar um arquivo texto pelo Python (arquivo "acesso.txt");
        arquivo = open("acesso.txt", "x")

        # Adiciona as informações de acesso
        arquivo.write("Usuário: " + usuario + " - Data: " + str(datetime.datetime.now()) + "\n")

    # Fecha o arquivo
    arquivo.close()

    # Permissão de somente leitura
    os.chmod("acesso.txt", stat.S_IRUSR)

    print("\nAs Informações de acesso foram registradas com sucesso!\n")
```