

```
"""
GRUPO 22
```

```
Equipe:
Larissa Hey D'Andrade (Inteligência Artificial Aplicada)
João Victor Pires de Campos (Gestão de Tecnologia da Informação)
Luiz Otávio de Azevedo Maciel (Análise e Desenvolvimento de Sistemas)
Wu Hsuan Yi Pedro (Big Data e Inteligência Analítica)
```

```
"""
```

```
import pyrebase, os, dotenv, stat
from pathlib import Path
from datetime import datetime, date

def load_environment():
    dotenv.load_dotenv(
        dotenv_path=os.path.join(Path(__file__).parent, '.env')
    )

class Authentication:
    def __init__(self, firebase):
        self.authentication = firebase.auth()
        self.email=''
        self.password=''
        self.status = ''
        self.info = ''
        self.idToken=''

    def execute(self):
        # """ Rotina principal, responsável pela interação com usuário """

        print("O que deseja fazer?")
        print("1 - Criar uma conta;\n"
              "2 - Checar informações da conta;\n")

        choice = '0'

        while (not self.choice_is_valid(choice)):
            choice = input()
            print()
            if (self.choice_is_valid(choice)):
                self.email=input("Digite seu e-mail: ")
                self.password=input("Digite sua senha, com pelo menos 6 caracteres: ")

                if choice == '1':
                    self.status = self.create_user()
                    break
                elif choice == '2':
                    self.status = self.sign_in_user()
                    break

            else:
                self.status = ''
                print("Você deve escolher 1 ou 2")
                input("Digite 1 ou 2")

    def choice_is_valid(self, choice):
        if (choice == '1' or choice == '2'):
            return True

        return False

    def verify_mail(self):
        # """ Rotina responsável pela verificação de email """
        users=self.info["users"]
        verify_email = users[0]["emailVerified"]

        if verify_email:
            self.create_log_in_file()
            print("Você está autenticado!")

        else:
            self.authentication.send_email_verification(self.idToken)
            print("Sua conta ainda não foi verificada. Enviamos um email com o link.")
```

```

def create_log_in_file (self):
    if os.path.isfile("login.txt"):
        # Modifica a permissão do arquivo para leitura, escrita e execução
        os.chmod("login.txt", stat.S_IRWXU)

    register = datetime.now()
    hora_atual = register.strftime("%H:%M:%S")
    data_atual = date.today()

    # Abre o arquivo para escrita
    arquivo = open("login.txt", 'w')

    # Escreve no arquivo
    arquivo.write(f"[{data_atual}, {hora_atual}] Login efetuado: {self.email}")

    # Fecha o arquivo
    arquivo.close()

    # Modifica o arquivo apenas para leitura
    os.chmod("login.txt", stat.S_IRUSR)

def create_user(self):
    self.status = self.authentication.create_user_with_email_and_password(self.email, self.password)
    self.idToken = self.status['idToken']
    self.info = self.authentication.get_account_info(self.idToken)
    print("Conta criada com sucesso.")

def sign_in_user(self):
    self.status = self.authentication.sign_in_with_email_and_password(self.email, self.password)
    self.idToken = self.status['idToken']
    self.info = self.authentication.get_account_info(self.idToken)
    self.verify_mail()

def main():
    """ Função principal do código
        Carrega as senhas da API a partir do .env """

    load_environment()

    firebaseConfig = {
        "apiKey": os.environ.get("APIKEY"),
        "authDomain": os.environ.get("APIKEY"),
        "projectId": os.environ.get("APIKEY"),
        "databaseURL": os.environ.get("APIKEY"),
        "storageBucket": os.environ.get("APIKEY"),
        "messagingSenderId": os.environ.get("APIKEY"),
        "appId": os.environ.get("APIKEY"),
        "measurementId": os.environ.get("APIKEY"),
    }

    Auth = Authentication(pyrebase.initialize_app(firebaseConfig))
    Auth.execute()

if __name__ == "__main__":
    main()

```