



Template

Roteiro Atividade Prática II

Unidade de Estudo 3



Atividade Prática – Autenticação Gmail

Este roteiro vai auxiliar o estudante na realização da segunda etapa da atividade prática. A proposta desta atividade é fornecer base para implementação de um segundo fator de autenticação. A atividade consiste em demonstrar como é possível enviar email utilizando a biblioteca smtplib. Estaremos utilizando a linguagem de programação Python em conjunto com ferramenta de desenvolvimento Pycharm. Como pré-requisito para realização desta atividade o estudante deve ter uma conta no Gmail.

Roteiro Atividade:

- Primeiramente devemos habilitar a autenticação multifator do Gmail, para isto acesse o seguinte link <https://myaccount.google.com/signinoptions/two-step-verification>
- Será solicitado que autentique com sua conta do Gmail.

Google

Jhonatan Geremias

prof.jgeremias@gmail.com

Para continuar, primeiro confirme sua identidade

Digite sua senha

☐ Mostrar senha

[Esqueceu a senha?](#) [Próxima](#)

- Para habilitar a autenticação em duas etapas é necessário que você forneça o número do seu dispositivo móvel. Selecione a opção “Mensagem de texto” e clique em “Próxima”.

← Verificação em duas etapas

Vamos configurar seu smartphone

Qual número de telefone você quer usar?

41 9876-5431

O Google só usará este número para fins de segurança da conta.
Não use um número do Google Voice.
Podem ser cobradas tarifas padrão para o envio de mensagens e dados.

Como deseja receber os códigos?

☒ Mensagem de texto ☐ Chamada telefônica


[Mostrar mais opções](#)

Etapa 1 de 3 [PRÓXIMA](#)




- Será enviado um código SMS para seu número de celular, insira o código para habilitar a autenticação de dois fatores.

← Verificação em duas etapas




Confirmar se ele funciona

O Google acaba de enviar uma mensagem de texto com um código de verificação para
(41) 9876-5431


Digite o código 

Não recebeu? [Reenviar](#)

[VOLTAR](#) Etapa 2 de 3 [PRÓXIMA](#) 


- Clique na opção “Ativar” para habilitar a autenticação de dois fatores.

← Verificação em duas etapas



Deu certo. Concluir a ativação da verificação em duas etapas?

Agora que você viu como a verificação em duas etapas funciona, deseja ativá-la para sua
Conta do Google [prof.jgeremias@gmail.com](#)?

Etapa 3 de 3 [ATIVAR](#) 



- A autenticação de dois fatores no Gmail foi ativada.

← Verificação em duas etapas

A verificação em duas etapas está ATIVADA desde 1 de jul. de 2021

DESATIVAR

Opções de segunda etapa disponíveis

Depois que você digita a senha, uma segunda etapa confirma que é você que está fazendo login.

[Saiba mais](#)

Observação: se você fizer login com sua Conta do Google em qualquer smartphone qualificado, as solicitações do Google serão adicionadas como outro método para a verificação em duas etapas.



Mensagem de texto ou voz (Padrão) ?

(41) 9876-5432 Verificado

Os códigos de verificação são enviados por mensagem de texto



- Na sequência precisamos criar uma senha para autenticação por aplicativo, para isto acesse o seguinte link: <https://security.google.com/settings/security/apppasswords>
- Será solicitado suas credenciais do Gmail.



Jhonatan Geremias

prof.jgeremias@gmail.com

Para continuar, primeiro confirme sua identidade

Digite sua senha

|

☐ Mostrar senha

[Esqueceu a senha?](#)

Próxima



- Agora precisamos adicionar uma senha para o aplicativo. Na opção “Selecionar app”, adicione a E-mail. Na opção “Selecionar dispositivo” adicione “Computador Windows”.

← Senhas de app

Senhas de app permitem que você faça login na sua Conta do Google a partir de apps em dispositivos que não sejam compatíveis com a verificação em duas etapas. Como só será necessário informar a senha uma vez, você não precisa memorizá-la. [Saiba mais](#)

Você não tem nenhuma senha de app.

Selecione o app e o dispositivo para o qual você quer gerar a senha de app.

Selecionar app ▼ Selecionar dispositivo ▼

↑ ↑

GERAR

- Após selecionar o app e o dispositivo será habilitado o botão “Gerar”. Para prosseguir clique no botão “Gerar”.

← Senhas de app

Senhas de app permitem que você faça login na sua Conta do Google a partir de apps em dispositivos que não sejam compatíveis com a verificação em duas etapas. Como só será necessário informar a senha uma vez, você não precisa memorizá-la. [Saiba mais](#)

Você não tem nenhuma senha de app.

Selecione o app e o dispositivo para o qual você quer gerar a senha de app.

E-mail ▼ Computador Windows ▼

↓

GERAR



- **Atenção:** Para cada usuário será criado uma senha diferente, você vai precisar desta senha para autenticar a aplicação no Python, efetue a cópia da senha para ser utilizada posteriormente. Tenha cuidado ao manipular sua senha, por meio é possível acessar a sua conta.

Senha de app gerada

Sua senha de app para computador Windows

→ gcpd xhvy fqtn yfnb

Como usar

1. Abra o app "Mail".
2. Abra o menu "Configurações".
3. Selecione "Contas" e selecione sua Conta do Google.
4. Substitua sua senha pela senha de 16 caracteres mostrada acima.

Assim como sua senha normal, esta senha de app concede acesso total à sua Conta do Google. Não é necessário memorizá-la, por isso não a anote ou a compartilhe com outras pessoas.

[Saiba mais](#)

CONCLUÍDO

- Certifique-se que a senha foi criada. Quando a senha não for mais útil, lembre-se de removê-la.

← Senhas de app

Senhas de app permitem que você faça login na sua Conta do Google a partir de apps em dispositivos que não sejam compatíveis com a verificação em duas etapas. Como só será necessário informar a senha uma vez, você não precisa memorizá-la. [Saiba mais](#)

Suas senhas de app

Nome	Criada	Usada pela última vez em
E-mail no meu Computador Windows	17:08	—

Selecione o app e o dispositivo para o qual você quer gerar a senha de app.

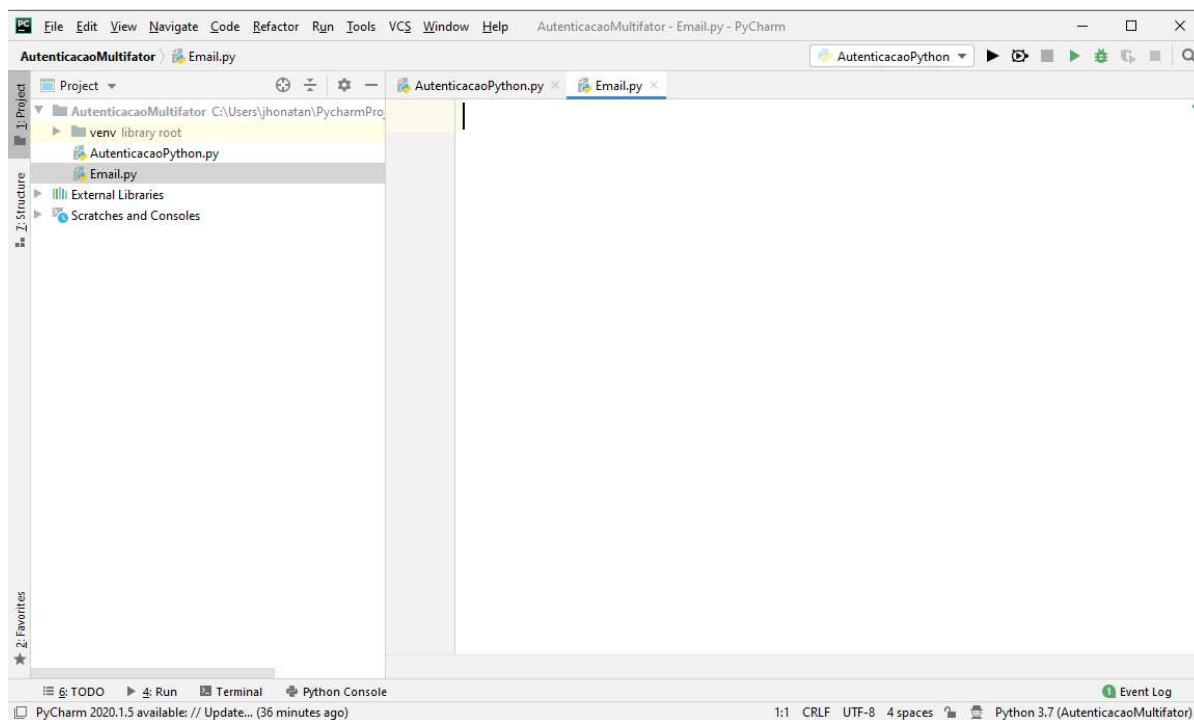
Selecionar app

Selecionar dispositivo

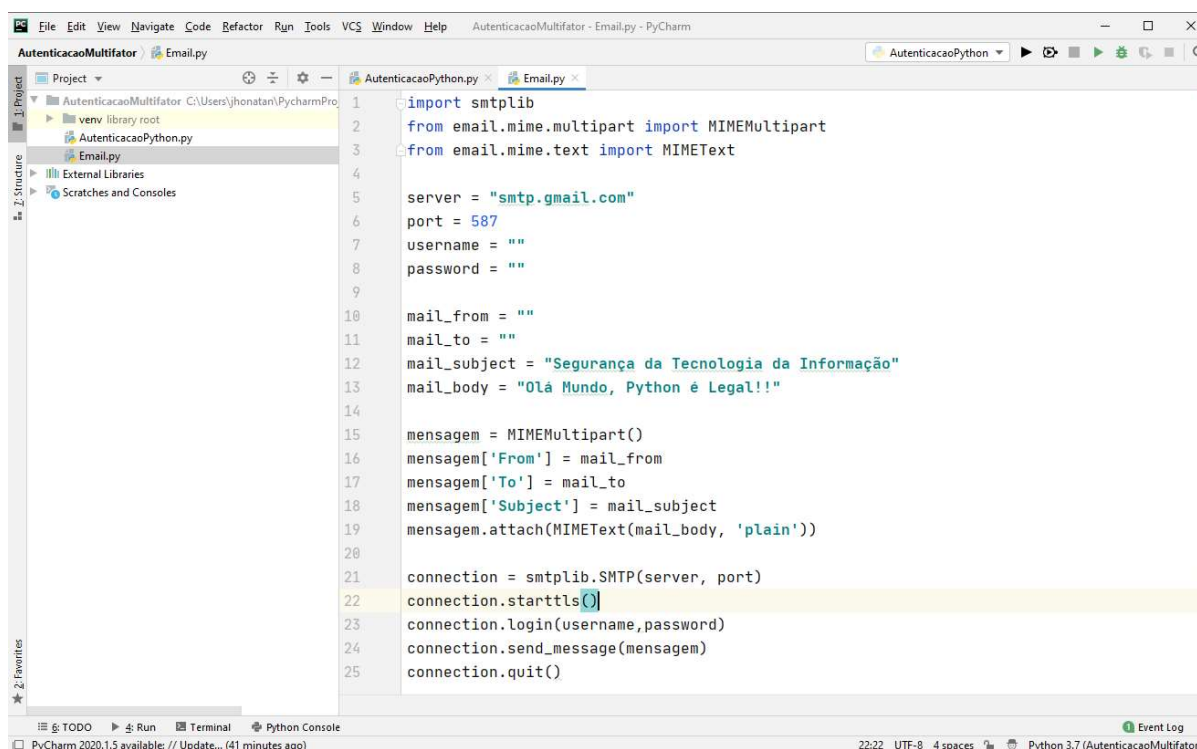
GERAR



- A configuração no Gmail está pronta. Agora podemos iniciar as atividades no Python, você pode utilizar o mesmo projeto no PyCharm criado para o Firebase, adicione apenas mais um arquivo Python.



- Copie o código disponível no arquivo email.py para o novo arquivo.





- Primeiramente defina o usuário da conta do Gmail.

```

1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = ""
9
10 mail_from = ""
11 mail_to = ""
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username,password)
24 connection.send_message(mensagem)
25 connection.quit()
    
```

- Copie a senha da aplicação que você habilitou no Gmail.

```

1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = "gcpdxhvyfqtnyfnb"
9
10 mail_from = ""
11 mail_to = ""
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username,password)
24 connection.send_message(mensagem)
25 connection.quit()
    
```




- Adicione a conta de e-mail do usuário remente.

```
1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = "gcpdxhvyfqtnyfnb"
9
10 mail_from = "prof.jgeremias@gmail.com"
11 mail_to = ""
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username,password)
24 connection.send_message(mensagem)
25 connection.quit()
26
```

- Adicione a conta de e-mail do usuário destinatário.

```
1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = "gcpdxhvyfqtnyfnb"
9
10 mail_from = "prof.jgeremias@gmail.com"
11 mail_to = "prof.jgeremias@gmail.com"
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username,password)
24 connection.send_message(mensagem)
25 connection.quit()
26
```



- Sua configuração está completa, vamos analisar o código. Primeiramente importamos a biblioteca smtplib. Esta biblioteca estabelece uma conexão como com servidor de e-mail utilizando o protocolo SMTP para permitir o envio das mensagens.

```

1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = "gcpdxhvyfqtnyfnb"
9
10 mail_from = "prof.jgeremias@gmail.com"
11 mail_to = "prof.jgeremias@gmail.com"
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username, password)
24 connection.send_message(mensagem)
25 connection.quit()
26

```

- Aqui estamos importando a biblioteca MIMEMultipart para permitir trabalhar com os e-mail no formato MIME, padrão para envio de e-mail, estaremos enviando as mensagens em modo texto.

```

1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = "gcpdxhvyfqtnyfnb"
9
10 mail_from = "prof.jgeremias@gmail.com"
11 mail_to = "prof.jgeremias@gmail.com"
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username, password)
24 connection.send_message(mensagem)
25 connection.quit()
26

```



- A variável “server” recebe o nome do servidor SMTP.

```

1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = "gcpdxhvyfqtnyfnb"
9
10 mail_from = "prof.jgeremias@gmail.com"
11 mail_to = "prof.jgeremias@gmail.com"
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username, password)
24 connection.send_message(mensagem)
25 connection.quit()
26

```

- A variável “port” recebe o número da porta do servidor SMTP para trabalhar com o algoritmo de criptografia STARTTLS.

```

1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = "gcpdxhvyfqtnyfnb"
9
10 mail_from = "prof.jgeremias@gmail.com"
11 mail_to = "prof.jgeremias@gmail.com"
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username, password)
24 connection.send_message(mensagem)
25 connection.quit()
26

```



- As variáveis “username” e “password”, recebem as credenciais para utilizar o usuário do Gmail. Reforçando que foi criado uma senha específica para utilizar no projeto.

```

1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = "gcpdxhvyfqtnyfnb"
9
10 mail_from = "prof.jgeremias@gmail.com"
11 mail_to = "prof.jgeremias@gmail.com"
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username,password)
24 connection.send_message(mensagem)
25 connection.quit()
26

```

- Aqui são criadas as variáveis que contêm as informações do e-mail: e-mail do remetente, e-mail do destinatário, assunto da mensagem e corpo do e-mail.

```

1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = "gcpdxhvyfqtnyfnb"
9
10 mail_from = "prof.jgeremias@gmail.com"
11 mail_to = "prof.jgeremias@gmail.com"
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username,password)
24 connection.send_message(mensagem)
25 connection.quit()
26

```




- Criando um objeto Python do tipo MIMEMultipart para trabalhar com o e-mail no formato MIME.

```

1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = "gcpdxhvyfqtnyfnb"
9
10 mail_from = "prof.jgeremias@gmail.com"
11 mail_to = "prof.jgeremias@gmail.com"
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username, password)
24 connection.send_message(mensagem)
25 connection.quit()
26

```

- Atribui as informações do remetente, destinatário e assunto para estrutura mensagem.

```

1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = "gcpdxhvyfqtnyfnb"
9
10 mail_from = "prof.jgeremias@gmail.com"
11 mail_to = "prof.jgeremias@gmail.com"
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username, password)
24 connection.send_message(mensagem)
25 connection.quit()
26

```



- Adicionando o corpo da mensagem. O e-mail será anexo no formato plain (texto simples).

```

1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = "gcpdxhvyfqtnyfnb"
9
10 mail_from = "prof.jgeremias@gmail.com"
11 mail_to = "prof.jgeremias@gmail.com"
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username, password)
24 connection.send_message(mensagem)
25 connection.quit()
26

```

- Abrindo uma conexão como o servidor SMTP.

```

1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = "gcpdxhvyfqtnyfnb"
9
10 mail_from = "prof.jgeremias@gmail.com"
11 mail_to = "prof.jgeremias@gmail.com"
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username, password)
24 connection.send_message(mensagem)
25 connection.quit()
26

```




- Habilitando a utilização do protocolo de criptografia STARTTLS.

```

1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = "gcpdxhvyfqtnyfnb"
9
10 mail_from = "prof.jgeremias@gmail.com"
11 mail_to = "prof.jgeremias@gmail.com"
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username, password)
24 connection.send_message(mensagem)
25 connection.quit()
26

```

- Realizando a autenticação com o servidor de e-mail.

```

1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = "gcpdxhvyfqtnyfnb"
9
10 mail_from = "prof.jgeremias@gmail.com"
11 mail_to = "prof.jgeremias@gmail.com"
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username, password)
24 connection.send_message(mensagem)
25 connection.quit()
26

```



- Enviando uma mensagem.

```

1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = "gcpdxhvyfqtnyfnb"
9
10 mail_from = "prof.jgeremias@gmail.com"
11 mail_to = "prof.jgeremias@gmail.com"
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username,password)
24 connection.send_message(mensagem)
25 connection.quit()
26

```

- Finalizando a conexão com o servidor de e-mail.

```

1 import smtplib
2 from email.mime.multipart import MIMEMultipart
3 from email.mime.text import MIMEText
4
5 server = "smtp.gmail.com"
6 port = 587
7 username = "prof.jgeremias@gmail.com"
8 password = "gcpdxhvyfqtnyfnb"
9
10 mail_from = "prof.jgeremias@gmail.com"
11 mail_to = "prof.jgeremias@gmail.com"
12 mail_subject = "Segurança da Tecnologia da Informação"
13 mail_body = "Olá Mundo, Python é Legal!!"
14
15 mensagem = MIMEMultipart()
16 mensagem['From'] = mail_from
17 mensagem['To'] = mail_to
18 mensagem['Subject'] = mail_subject
19 mensagem.attach(MIMEText(mail_body, 'plain'))
20
21 connection = smtplib.SMTP(server, port)
22 connection.starttls()
23 connection.login(username,password)
24 connection.send_message(mensagem)
25 connection.quit()
26

```



- Enfim, vamos executar o código. Após executar este código você deve verificar a caixa do e-mail do destinatário para checar se o e-mail foi enviado.

```

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

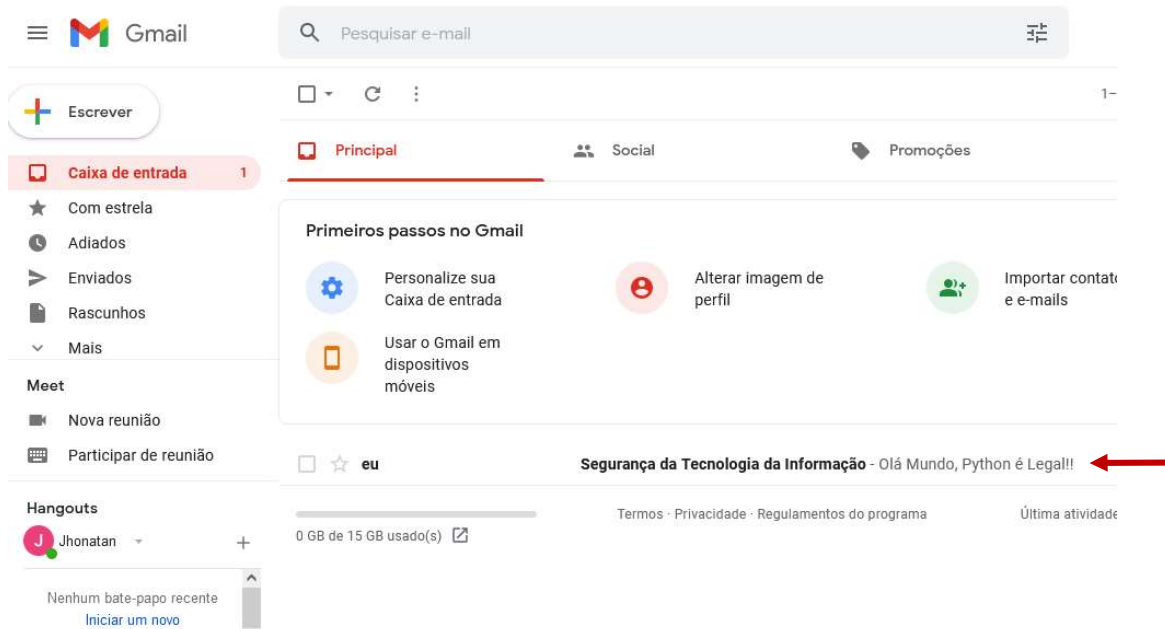
server = "smtp.gmail.com"
port = 587
username = "prof.jgeremias@gmail.com"
password = "gcpdxhvyfqtnyfnb"

mail_from = "prof.jgeremias@gmail.com"
mail_to = "prof.jgeremias@gmail.com"
mail_subject = "Segurança da Tecnologia da Informação"
mail_body = "Olá Mundo, Python é Legal!!"

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

Process finished with exit code 0

- Se tudo estiver certo é para você ter recebido um novo e-mail.



Neste roteiro aprendemos como enviar uma mensagem no Python por e-mail. Esta atividade tem por propósito fornecer recursos para que você implemente uma autenticação multifator.





PROFESSOR-AUTOR

Escreva aqui o nome



PUCPR
GRUPO MARISTA