

Experimento: Lab 3 - SMTP

Ao final deste laboratório, você terá adquirido um melhor entendimento do protocolo SMTP. Você também ganhará experiência na implementação de um protocolo padrão usando Python.

Sua tarefa é desenvolver um cliente de e-mail simples que envie e-mails para qualquer destinatário. Seu cliente precisará se conectar a um servidor de e-mail (como o <https://temp-mail.org/pt/>, por exemplo), dialogar com o servidor usando o protocolo SMTP e enviar uma mensagem de e-mail para o servidor. O Python fornece um módulo chamado `smtplib`, que possui métodos embutidos para enviar e-mails usando o protocolo SMTP. No entanto, não utilizaremos esse módulo neste laboratório, pois ele esconde os detalhes do SMTP e da programação com *sockets*.

Para limitar o spam, alguns servidores de e-mail não aceitam conexões TCP de fontes arbitrárias. Para o experimento descrito abaixo, você pode tentar se conectar tanto ao servidor de e-mail da sua universidade quanto a um servidor de Webmail popular, como o servidor de e-mail AOL. Também pode tentar fazer sua conexão tanto de casa quanto do campus da universidade.

Código

Abaixo, você encontrará o código base para o cliente. Você deve completar o código base. Os locais onde você precisa adicionar código estão marcados com `#Fill in start` e `#Fill in end`. Cada local pode exigir uma ou mais linhas de código.

Notas Adicionais

Em alguns casos, o servidor de e-mail receptor pode classificar seu e-mail como lixo eletrônico. Certifique-se de verificar a pasta de spam quando procurar o e-mail enviado pelo seu cliente.

O que apresentar

Em sua apresentação, você deve mostrar e explicar o código completo do seu cliente de e-mail SMTP e mostrar o seu código funcionando.

Esta atividade vale nota. A implementação básica resulta em nota máxima 6,0.

Código Base do Cliente SMTP em Python:

```
from socket import *

msg = "\r\n Eu amo redes de computadores!"
endmsg = "\r\n.\r\n"

# Escolher um servidor de e-mail (por exemplo, servidor do Google) e chamá-lo de
mailserver
mailserver = # Fill in start    # Fill in end

# Criar um socket chamado clientSocket e estabelecer uma conexão TCP com o servidor de
e-mail
# Fill in start
# Fill in end
```

```

recv = clientSocket.recv(1024).decode()
print(recv)
if recv[:3] != '220':
    print('220 resposta não recebida do servidor.')

# Enviar comando HELO e imprimir a resposta do servidor.
heloCommand = 'HELO Alice\r\n'
clientSocket.send(heloCommand.encode())
recv1 = clientSocket.recv(1024).decode()
print(recv1)
if recv1[:3] != '250':
    print('250 resposta não recebida do servidor.')

# Enviar comando MAIL FROM e imprimir a resposta do servidor.
# Fill in start
# Fill in end

# Enviar comando RCPT TO e imprimir a resposta do servidor.
# Fill in start
# Fill in end

# Enviar comando DATA e imprimir a resposta do servidor.
# Fill in start
# Fill in end

# Enviar os dados da mensagem.
# Fill in start
# Fill in end

# A mensagem termina com um único ponto.
# Fill in start
# Fill in end

# Enviar comando QUIT e obter a resposta do servidor.
# Fill in start
# Fill in end

```

Exercícios Opcionais

1. Servidores de e-mail como o Google (endereço: `smtp.gmail.com`, porta: 587) exigem que seu cliente adicione Transport Layer Security (TLS) ou Secure Sockets Layer (SSL) para autenticação e segurança, antes de enviar o comando MAIL FROM. Adicione comandos TLS/SSL ao seu cliente existente e implemente a comunicação com o servidor de e-mail do Google utilizando o endereço e a porta indicados.
2. O cliente SMTP atual só envia mensagens de texto no corpo do e-mail. Modifique seu cliente para que ele possa enviar e-mails com texto e imagens.