

Trabalhando com e-mail

Enviando e-mails simples com Python

Enviando e-mails simples com Python

Introdução

Para enviar *e-mails* simples (sem formatação) utilizando o *Python* podemos utilizar o módulo *smtplib* que é um módulo disponível por padrão no *Python*. O módulo *smtplib* define um objeto de sessão de cliente *SMTP* que pode ser utilizado para enviar e-mails para todos computadores conectados à *Internet* que possuam o serviço de processamento *SMTP* ou *ESMTP*.

Antes de implementarmos nossos exemplos vamos primeiro ver alguns conceitos:

Enviando e-mails simples com Python

Alguns conceitos

SMTP

Simple Mail Transfer Protocol (abreviado *SMTP*. Traduzido do inglês, significa “Protocolo de transferência de correio simples”) é o protocolo padrão para envio de e-mails através da *Internet*.

É um protocolo relativamente simples, em texto plano, onde um ou vários destinatários de uma mensagem são especificados (e, na maioria dos casos, validados) sendo, depois, a mensagem transferida. Esse protocolo usa por padrão a porta *TCP* 25 (ou 465 para conexão criptografada via *SSL*). No Brasil, porém, desde 2013, provedores e operadoras de internet passaram a utilizar a porta 587, como medida de segurança para diminuir o número de spams.

O *SMTP* é um protocolo de envio, o que significa que ele não permite que um usuário descarregue as mensagens de um servidor. Para isso, é necessário um cliente de e-mail com suporte ao protocolo *POP3* ou *IMAP*, o que é o caso da maioria dos clientes atuais.

Enviando e-mails simples com Python

Alguns conceitos

SSL

O *SSL (Secure Sockets Layer)* é um protocolo de segurança criado pela *Netscape* que se tornou padrão internacional para troca de informações sigilosas na *Internet*.

TLS

O *TLS (Transport Layer Security)*, assim como o seu antecessor *SSL*, é um protocolo de segurança que protege as telecomunicações via *Internet* para serviços como e-mail (*SMTP*), navegação por páginas (*HTTPS*) e outros tipos de transferência de dados.

Enviando e-mails simples com Python

Enviando e-mail sem criptografia

Para enviar e-mails sem *SSL* ou *TLS* o servidor de e-mail deve estar configurado para permitir o envio sem a criptografia.

```
import smtplib as smtp

def enviar_email():
    de = "****@provedor.com.br" # Usuário da conta de e-mail de envio
    senha = '****' # Senha do usuário.
    para = ["****@provedor.com.br"] # Informe o(s) destino(s)
    mensagem = "Subject: Teste\n\nTeste de envio de e-mail" # Mensagem a ser enviada

    try:
        with smtp.SMTP('smtp.provedor.com.br', 587) as s:
            s.login(de, senha) # Efetuando login com o usuário e senha
            s.sendmail(de, para, mensagem) # Enviando e-mail
            s.close() # Fechando a conexão
        print("E-mail enviado!")
    except Exception as erro:
        print("Não foi possível enviar o e-mail. Erro:", erro)

if __name__ == '__main__':
    enviar_email()
```

Enviando e-mails simples com Python

Enviando e-mail sem criptografia

A classe *SMTP* suporta a declaração *with*.

Quando usamos *with*, o comando QUIT é executado automaticamente. Se não utilizarmos *with*, teremos que usar o método *quit()* ao final do envio.

```
with smtp.SMTP('smtp.provedor.com.br', 587) as s:  
    s.login(de, senha) # Efetuando login com o usuário e senha  
    s.sendmail(de, para, mensagem) # Enviando e-mail  
    s.close() # Fechando a conexão
```

Enviando e-mails simples com Python

Enviando e-mail com criptografia SSL usando Gmail.

```
import smtplib as smtp

def enviar_email():
    de = "*****@gmail.com" # Usuário do GMail para envio
    senha = '*****' # Senha
    para = ["evaldowolkers@gmail.com"] # Destinatário
    mensagem = "Subject: Teste\n\nTeste de envio de e-mail com SSL" # Mensagem a ser enviada

    try:
        with smtp.SMTP_SSL('smtp.gmail.com', 465) as s:
            s.login(de, senha) # Efetuando login com o usuário e senha
            s.sendmail(de, para, mensagem) # Enviando e-mail
            s.close() # Fechando a conexão
        print("E-mail enviado!")
    except Exception as erro:
        print("Não foi possível enviar o e-mail. Erro:", erro)

if __name__ == '__main__':
    enviar_email()
```

Enviando e-mails simples com Python

Erro no envio usando conta do GMail

Ao tentar enviar *e-mail* usando o *SMTP* do *GMail*, se ocorrer um erro como este erro apresentado a seguir, você terá que entrar na conta que está utilizando e permitir a execução de aplicativos menos seguros.

Erro exibido ao executar o sistema:

Please log in via your web browser and then try again. Learn more at <https://support.google.com/mail/answer/78754>

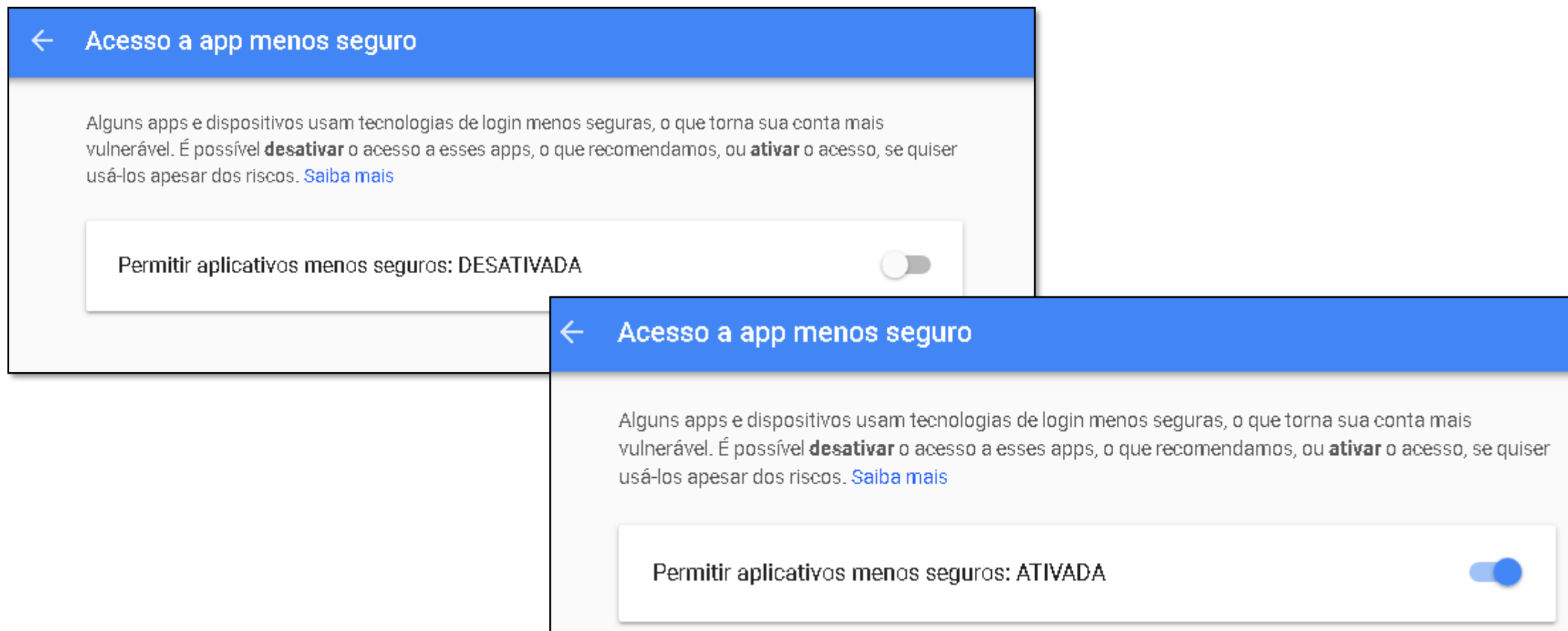
Link para permitir aplicativos menos seguros:

<https://myaccount.google.com/lesssecureapps>

Observação: O GMail utiliza a porta 465 para *SSL* e 587 para *TLS*.

Enviando e-mails simples com Python

Erro no envio usando conta do GMail



Após ativar esta opção, você conseguirá usar a conta para enviar *e-mail* via *smtplib*.

Enviando e-mails simples com Python

Erro no envio usando conta do GMail

Se a conta que for utilizar para enviar e-mail utiliza autenticação em dois fatores, será necessário criar uma senha de aplicativo para utilizar em vez de utilizar sua senha do e-mail.

Esse vai ser o erro exibido em nossa aplicação neste caso:

Não foi possível enviar o e-mail. Erro: (534, b'5.7.9 Application-specific password required. Learn more at\n5.7.9 <https://support.google.com/mail/?p=InvalidSecondFactor> m12sm5429230qkk.123 - gsmtip')

Enviando e-mails simples com Python

Erro no envio usando conta do GMail

Essas são as instruções apresentadas no link indicado no erro:

Como gerar uma senha de app

Acesse sua Conta do Google.

No painel de navegação à esquerda, clique em Segurança.

No painel "Como fazer login no Google", clique em Senhas de app. Se for solicitado, digite sua senha.

Observação: se não for possível acessar a página, isso significa que a verificação em duas etapas:

- não foi configurada na sua conta;
- foi configurada apenas com chaves de segurança.

Na parte inferior da página, clique em Selecionar app e escolha o app que você está usando.

Clique em Selecionar dispositivo e escolha o dispositivo que você está usando.

Clique em Gerar.

Siga as instruções para inserir a senha de app (o código de 16 caracteres na barra amarela) no seu dispositivo.

Clique em Concluído.

Depois que você terminar, o código da senha de app não será mais exibido. No entanto, você verá uma lista de apps e dispositivos para os quais criou senhas de app.

Enviando e-mails simples com Python

Enviando e-mail com criptografia TLS usando Gmail.

```
import smtplib as smtp

def enviar_email():
    de = "****@gmail.com" # Usuário do GMail para envio
    senha = '****' # Senha
    para = ["evaldowolkers@gmail.com"] # Destinatário
    mensagem = "Subject: Teste\n\nTeste de envio de e-mail com TLS" # Mensagem a ser enviada

    try:
        with smtp.SMTP('smtp.gmail.com', 587) as s:
            s.starttls()
            s.login(de, senha) # Efetuando login com o usuário e senha
            s.sendmail(de, para, mensagem) # Enviando e-mail
            s.close() # Fechando a conexão
        print("E-mail enviado!")
    except Exception as erro:
        print("Não foi possível enviar o e-mail. Erro:", erro)

if __name__ == '__main__':
    enviar_email()
```

FIM