

Sistemas Operacionais Embarcados

Internet

Internet

- Conteúdo:
 - Conexão à internet
 - Download de arquivos
 - Download de arquivos em serviços em nuvem (Dropbox, Google Drive etc.)
 - Buscas pelo Google
 - Hora mundial
 - Envio de informações (formulários, arquivos etc.)
 - E-mail
 - Servidor web local
 - *Streaming* de video
 - Referências

Conexão à internet

Modelos do Raspberry Pi sem WiFi

- Conecte o cabo ethernet ao RPi. Ele deve se configurar automaticamente para acessar a internet.

Modelos do Raspberry Pi com WiFi

- Execute `sudo iwlist wlan0 scan | grep SSID` para obter os nomes das redes WiFi disponíveis
- Abra o arquivo `/etc/wpa_supplicant/wpa_supplicant.conf`, e acrescente o seguinte texto ao final dele:

```
network={  
    ssid="NOME_DA_REDE"  
    psk="SENHA_DA_REDE"  
}
```

onde `ssid` é o nome da rede, e `psk` é a senha da rede.

Conexão à internet

Modelos do Raspberry Pi com WiFi

- Se quiser criptografar a senha, execute `wpa_passphrase NOME_DA_REDE` e digite a senha da rede. Escreva no arquivo `/etc/wpa_supplicant/wpa_supplicant.conf` o texto retornado no terminal:

```
network={  
    ssid="testing"  
    #psk="testingPassword"
```

```
psk=131e1e221f6e06e3911a2d11ff2fac9182665c004de85300f9cac208a6a80531  
}
```

(Retire o texto precedido pelo símbolo # para maior segurança.)

Conexão à internet

Modelos do Raspberry Pi com WiFi

- Na UnB, utilize a rede eduroam com a seguinte configuração:

```
network={
    ssid="eduroam"
    proto=RSN
    key_mgmt=WPA-EAP
    eap=PEAP
    identity="SEU_EMAIL@unb.br"
    password="SUA_SENHA"
    phase1="peaplabel=0"
    phase2="auth=MSCHAPV2"
}
```

onde os campos `identity` e `password` contém seu e-mail unb.br e sua senha.

Download de arquivos

- A maneira mais simples de baixar arquivos da internet pelo RPi é utilizando o comando `wget`. Por exemplo,

```
wget  
https://raw.githubusercontent.com/DiogoCaetanoGarcia/Sistemas_Embarcados/master/README.md
```

baixa um arquivo no GitHub, e `wget www.unb.br` baixa o arquivo HTML da página principal da UnB.

Download de arquivos

- O comando `curl` oferece muito mais opções do que o `wget`, sendo bastante interessante para o que veremos mais à frente. Execute

```
curl -O  
https://raw.githubusercontent.com/DiogoCaetanoGarcia/Sistemas_Embarcados/master/README.md
```

```
curl -O www.unb.br
```

para obter os mesmos resultados.

- Execute `curl -o saida.html www.unb.br` para salvar o resultado no arquivo local `saida.html`

Download de arquivos em serviços em nuvem (Dropbox, Google Drive etc.)

- Para fazer o *download* de arquivos em serviços de armazenamento em nuvem, tais como o Dropbox, o Google Drive e o Microsoft OneDrive, pode ser necessário realizar alguns passos extra. Antes de mais nada, é necessário liberar o acesso ao arquivo no serviço em nuvem.
- Considere o arquivo `Arquivo_na_nuvem1.pdf`, disponível em <https://drive.google.com/file/d/11uI5-BAXzIhKO14clUwmQlAo8LyqS-Ui/view?usp=sharing>. É necessário separar nesta URL o identificador do arquivo (uma sequência aparentemente aleatória de caracteres) para fazer o download adequadamente:

```
filename="Arquivo na nuvem1.pdf"
fileid="11uI5-BAXzIhKO14clUwmQlAo8LyqS-Ui"
curl -L "https://drive.usercontent.google.com/download?id=${fileid}&export=download&confirm=t" -o $filename
```


Download de arquivos em serviços em nuvem (Dropbox, Google Drive etc.)

- Anteriormente, o Google Drive pedia uma espécie de autenticação para arquivos com mais de 100MB, mas ela não é necessária com o script anterior. Considere o arquivo `Arquivo_na_nuvem2.pdf`, disponível em https://drive.google.com/file/d/1EPVcIVl0UiNtjw9S751ZRiw_FQdTLHf0/view?usp=sharing:

S

```
filename="Arquivo na nuvem2.pdf"  
fileid="1EPVcIVl0UiNtjw9S751ZRiw_FQdTLHf0"  
curl -L "  
"https://drive.usercontent.google.com/download?id=${fileid}&export  
=download&confirm=t" -o $filename
```

- Procedimento semelhante pode ser necessário com o Dropbox e o OneDrive.

Buscas pelo Google

- Para buscar pelo termo “raspberry pi” em www.google.com, por exemplo, execute

```
curl -o saida.html https://www.google.com/search?q=raspberry+pi
```

- Este arquivo contém bastante código Javascript, mas é possível encontrar uma série de links dentro dele.
- Se você abrir o arquivo salvo `saida.html` em um *browser*, verá os resultados renderizados pelo interpretador Javascript do *browser*.

Hora mundial

- Para conferir a hora atual via IP da máquina, execute

```
curl http://worldtimeapi.org/api/ip.txt
```

- Para conferir a hora atual via o fuso-horário de Brasília/São Paulo, execute

```
curl http://worldtimeapi.org/api/timezone/America/Sao_Paulo.txt
```

- Para conferir os fuso-horários disponíveis, execute

```
curl http://worldtimeapi.org/api/timezone.txt
```

- Para pegar somente um dos campos da saída, use o grep:

```
curl http://worldtimeapi.org/api/ip.txt | grep unixtime | sed  
"s/unixtime: //"
```

Envio de informações (formulários, arquivos etc.)

- Páginas web possuem diversos campos destinados a formulários. Por exemplo, a página de login de uma página geralmente possui campos para usuário e senha, além de um botão para envio destas informações. Barras de busca são outro exemplo relevante.
- Quando se baixa uma página na internet, é feita uma requisição HTTP GET. No caso do envio destas informações de formulários, é feita uma requisição HTTP POST. O comando `curl` permite automatizar este tipo de requisição.

Envio de informações (formulários, arquivos etc.)

- Por exemplo, entre na página <https://fga.unb.br>, e faça uma busca na barra correspondente pela palavra “eletronica”. Você será redirecionado para a página <https://fga.unb.br/search/articles?query=eletronica>. Uma forma simples de automatizar essa busca é executando

```
curl -v -o exemplo_POST.txt -d query=eletronica  
https://fga.unb.br/search/articles
```

```
less exemplo_POST.txt
```

- Repare na saída do comando curl que foi feita uma requisição POST.
- Pelo resultado da busca, pode-se perceber que uma requisição GET também é suficiente:

```
curl -v -o exemplo_POST.txt  
https://fga.unb.br/search/articles?query=eletronica
```

Envio de informações (formulários, arquivos etc.)

- Atualização de serviços em nuvem:
 - A instalação de aplicativos de serviços em nuvem pode ser uma solução simples, mas custosa computacionalmente, já que estes aplicativos continuamente conferem se houve mudanças em suas pastas principais e secundárias. Para fazer o upload de arquivos pelo terminal, cada serviço oferece uma solução diferente. Confira algumas delas nos links ao final deste documento.

Envio de informações (formulários, arquivos etc.)

- Comunicação via Telegram

- Com o comando `curl`, é possível criar bots no Telegram. Por exemplo, para mandar uma mensagem “Olá mundo”, execute

```
curl -s -X POST https://api.telegram.org/bot<TOKEN>/sendMessage -d chat_id=<CHAT_ID> -d text="Hello World"
```

onde <TOKEN> é uma string necessária para autorizar o bot a mandar pedidos à API do Telegram (por exemplo, 4334584910:AAEPmjlh84N62Lv3jGWEgOftlxxAfMhB1gs), e <CHAT_ID> é o identificador da conversa.

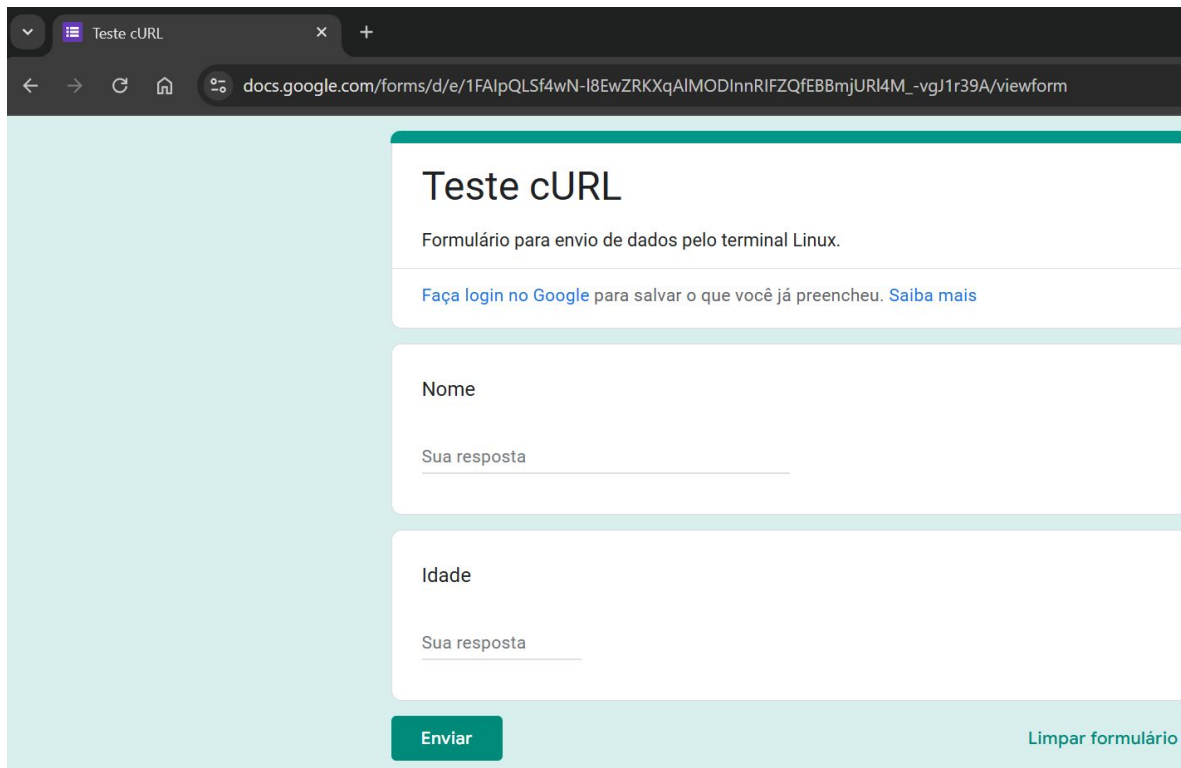
Envio de informações (formulários, arquivos etc.)

- Outros sistemas de mensagens e notificações
 - Pushbullet: conexão entre dispositivos
 - IFTTT (If This, Than That): conexão entre serviços na nuvem
 - MQTT: protocolo de comunicações máquina/máquina (M2M - Internet of Things)
 - PUSHETTA: notificações em tempo real para diferentes dispositivos (smartphones, browsers, smart TV etc.)
 - yowsup: comunicação via WhatsApp

Envio de informações (formulários, arquivos etc.)

- Armazenamento de dados no Google Sheets

No Google Drive, é possível criar formulários com campos específicos, e definir uma planilha online para receber os resultados do preenchimento deste formulário. Por exemplo, confira o formulário a seguir:

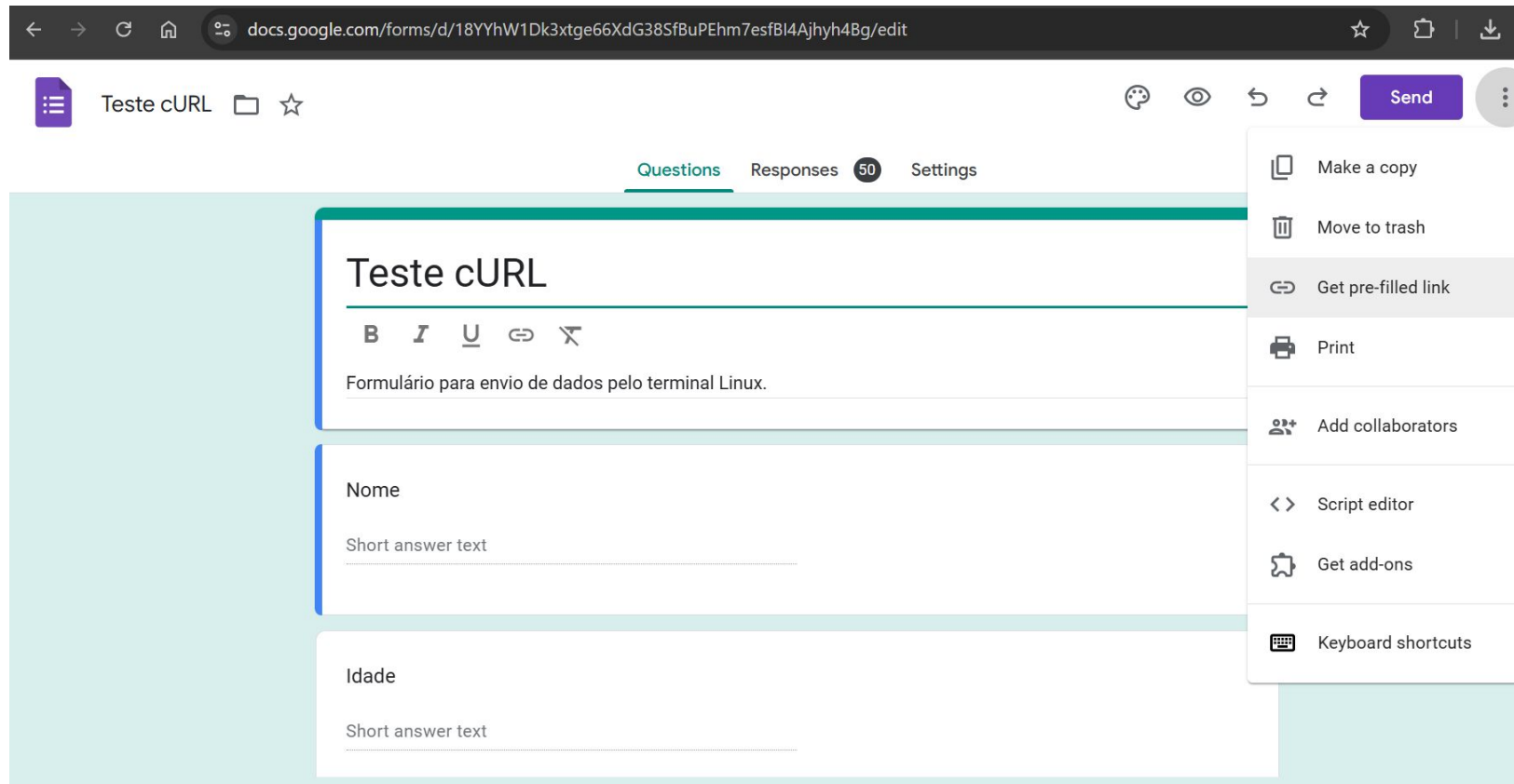


The screenshot shows a web browser window with a single tab titled 'Teste cURL'. The address bar displays the URL: docs.google.com/forms/d/e/1FAIpQLSf4wN-l8EwZRKXqAlMODInnRIFZQfEBBmjURl4M_-vgJ1r39A/viewform. The form itself has a light blue background. At the top, the title 'Teste cURL' is displayed in a bold, dark font. Below the title, a subtitle reads 'Formulário para envio de dados pelo terminal Linux.' A link 'Faça login no Google para salvar o que você já preencheu. Saiba mais' is positioned below the subtitle. The form contains two text input fields. The first field is labeled 'Nome' and the second is labeled 'Idade'. Each field has a placeholder text 'Sua resposta' and a horizontal line indicating where to type. At the bottom of the form, there is a green button labeled 'Enviar' and a link 'Limpar formulário'.

https://docs.google.com/forms/d/e/1FAIpQLSf4wN-l8EwZRKXqAlMODInnRIFZQfEBBmjURl4M_-vgJ1r39A/viewform

Envio de informações (formulários, arquivos etc.)

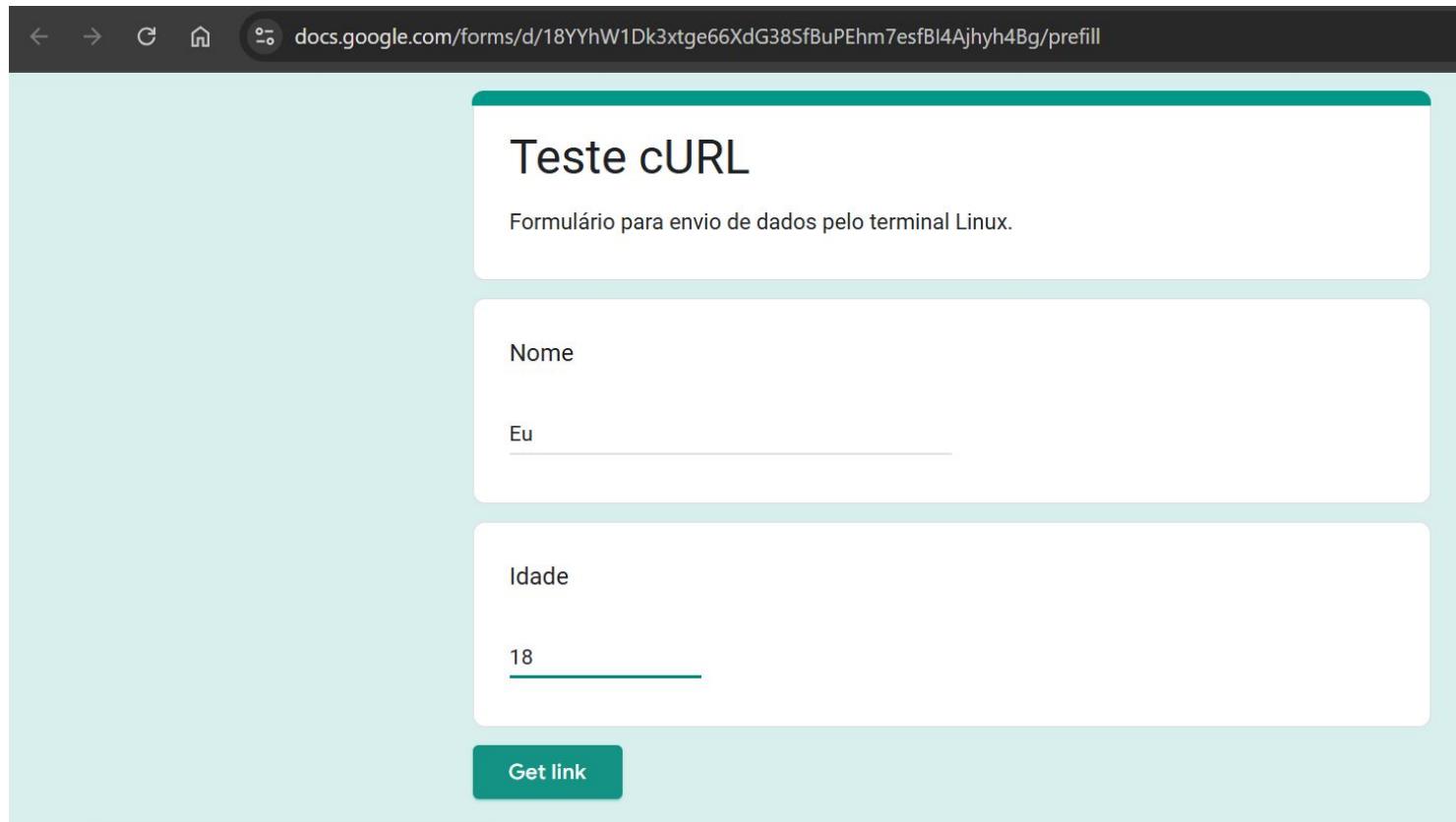
É possível automatizar o preenchimento deste formulário pelo comando curl. Vá para o campo de edição do formulário e clique em “Get pre-filled link”:



The screenshot shows a Google Forms editor interface. The browser address bar displays the URL: docs.google.com/forms/d/18YYhW1Dk3xtge66XdG38SfBuPEhm7esfBI4Ajhyh4Bg/edit. The form title is "Teste cURL". Below the title is a description: "Formulário para envio de dados pelo terminal Linux." The form contains two text input fields: "Nome" and "Idade", both labeled "Short answer text". A dropdown menu is open on the right side of the form, showing options: "Make a copy", "Move to trash", "Get pre-filled link" (highlighted), "Print", "Add collaborators", "Script editor", "Get add-ons", and "Keyboard shortcuts".

Envio de informações (formulários, arquivos etc.)

Preencha o formulário e clique em “Get link”:



The image shows a web browser window with the address bar displaying `docs.google.com/forms/d/18YYhW1Dk3xtge66XdG38SfBuPEhm7esfBI4Ajhyh4Bg/prefill`. The form itself has a light blue background. At the top, there is a white box with the title "Teste cURL" and a subtitle "Formulário para envio de dados pelo terminal Linux." Below this, there are two text input fields. The first is labeled "Nome" and contains the text "Eu". The second is labeled "Idade" and contains the number "18". At the bottom of the form, there is a teal button labeled "Get link".

Teste cURL

Formulário para envio de dados pelo terminal Linux.

Nome

Eu

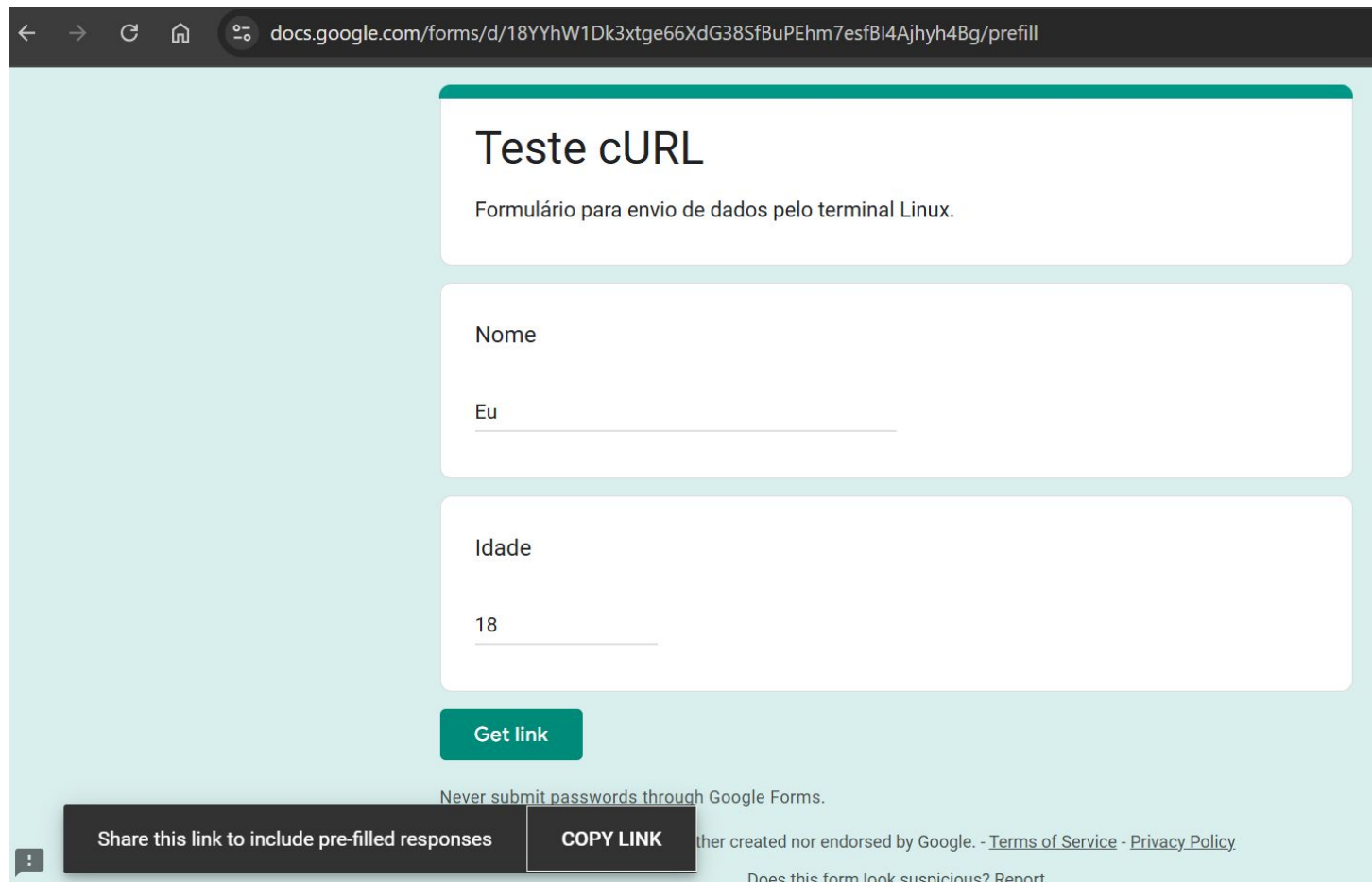
Idade

18

Get link

Envio de informações (formulários, arquivos etc.)

Clique em “Copy link”:



The screenshot shows a web browser window with the address bar displaying the URL: docs.google.com/forms/d/18YYhW1Dk3xtge66XdG38SfBuPEhm7esfBI4Ajhyh4Bg/prefill. The form itself is titled "Teste cURL" and has a subtitle "Formulário para envio de dados pelo terminal Linux." Below the title, there are two text input fields. The first field is labeled "Nome" and contains the text "Eu". The second field is labeled "Idade" and contains the number "18". At the bottom of the form, there is a green button labeled "Get link". Below the button, there is a small text warning: "Never submit passwords through Google Forms." At the very bottom of the page, there is a dark grey bar with two buttons: "Share this link to include pre-filled responses" and "COPY LINK". To the right of these buttons, there is a small text link: "either created nor endorsed by Google. - Terms of Service - Privacy Policy". At the bottom right corner, there is a small text link: "Does this form look suspicious? Report".

docs.google.com/forms/d/18YYhW1Dk3xtge66XdG38SfBuPEhm7esfBI4Ajhyh4Bg/prefill

Teste cURL

Formulário para envio de dados pelo terminal Linux.

Nome

Eu

Idade

18

Get link

Never submit passwords through Google Forms.

Share this link to include pre-filled responses

COPY LINK

either created nor endorsed by Google. - [Terms of Service](#) - [Privacy Policy](#)

Does this form look suspicious? [Report](#)

Envio de informações (formulários, arquivos etc.)

O resultado vem com a URL correspondente ao formulário que você preencheu:

https://docs.google.com/forms/d/e/1FAIpQLSf4wN-l8EwZRKXqAlMODInnRIFZQfEBBmjURL4M_-vgJ1r39A/viewform?usp=pp_url&entry.1962235247=Eu&entry.146553730=18

Para converter essa URL para uma chamada curl, separe os campos e faça um pedido POST:

```
curl -G -Ss --data-urlencode "entry.1962235247=Eu"  
--data-urlencode "entry.146553730=18"  
https://docs.google.com/forms/d/e/1FAIpQLSf4wN-l8EwZRKXqAlMODInnRI  
FZQfEBBmjURL4M_-vgJ1r39A/formResponse?usp=pp_url
```

E-mail

- O comando `curl` trabalha com diversos protocolos, incluindo SMTP. Assim, ele permite o envio de e-mails.
- Os provedores modernos de e-mail exigem mais graus de segurança. No Gmail, é possível habilitar login simplificado para aplicativos específicos (<https://myaccount.google.com/apppasswords>)

← App passwords

App passwords help you sign into your Google Account on older apps and services that don't support modern security standards.

App passwords are less secure than using up-to-date apps and services that use modern security standards. Before you create an app password, you should check to see if your app needs this in order to sign in.

[Learn more](#)

Your app passwords

geeksforgeeks

Created on 3:33 PM



To create a new app specific password, type a name for it below...

App name

E-mail

Crie um arquivo “email.conf” com os dados de login simplificado:

```
Username: USUARIO  
Password: SENHA
```

Altere as permissões deste arquivo para que outros usuários não tenham acesso a seus dados de login:

```
chmod 600 email.conf
```

E-mail

Rode os seguintes comandos para enviar um email de teste com esta conta:

```
TO="EMAIL DE DESTINO"
SMTP USERNAME=$(grep "Username:" email.conf|sed "s/Username: //")
SMTP PASSWORD=$(grep "Password:" email.conf|sed "s/Password: //")
SUBJECT="Hello from Curl"
BODY="This is the email body sent using Curl and SMTP."
MESSAGE="Subject: $SUBJECT\n\n$BODY"
curl --url "smtp://smtp.gmail.com:587" --ssl-reqd \
--mail-from "$SMTP USERNAME" --mail-rcpt "$TO" \
--user "$SMTP USERNAME:$SMTP PASSWORD" \
--tlsv1.2 -T <(echo -e "$MESSAGE")
```


E-mail

Para acrescentarmos arquivos em anexo, usamos codificação base64 no corpo do e-mail:

```
TO="EMAIL DE DESTINO"  
TO="srcaetano@gmail.com"  
SMTP USERNAME=$(grep "Username:" email.conf | sed "s/Username: //")  
SMTP PASSWORD=$(grep "Password:" email.conf | sed "s/Password: //")  
SUBJECT="Hello from Curl"  
BODY="This is the email body sent using Curl and SMTP."  
nome imagem="unb.png"  
email from="EMAIL REMETENTE"  
passw from="SENHA REMETENTE"  
email to="EMAIL DESTINATARIO"  
echo "From: <${SMTP_USERNAME}@gmail.com>" > email.txt  
echo "To: <${TO}>" >> email.txt  
echo Subject: $SUBJECT >> email.txt  
echo Date: $(date) >> email.txt  
echo Content-Type: multipart/mixed\; boundary=corpo_msg >>  
email.txt
```

E-mail

```
echo >> email.txt
echo --corpo msg >> email.txt
echo Content-Type: text/plain\; charset=UTF-8 >> email.txt
echo >> email.txt
echo $BODY >> email.txt
echo >> email.txt
echo --corpo msg >> email.txt
echo Content-Type: image/png\; name=\"$nome_imagem\" >> email.txt
echo Content-Transfer-Encoding: base64 >> email.txt
echo Content-Disposition: attachment\; filename=\"$nome_imagem\"
>> email.txt
echo >> email.txt
cat $nome_imagem | base64 >> email.txt
echo --corpo msg-- >> email.txt
curl --url "smtp://smtp.gmail.com:587" \
--ssl-reqd --mail-from "$SMTP_USERNAME" \
--mail-rcpt "$TO" --user "$SMTP_USERNAME:$SMTP_PASSWORD" \
--tlsv1.2 -T email.txt
rm email.txt
```

E-mail

- Neste exemplo, baixamos uma imagem de <https://fga.unb.br/articles/0001/7219/guia-unb-gama.png>, e a acrescentamos ao arquivo `email.txt` utilizando o comando `base64`
- Também indicamos no arquivo `email.txt` que este anexo é uma imagem do tipo PNG em

```
echo Content-Type: image/png\; name=\"$nome_imagem\" >>  
email.txt
```

Para outros tipos de arquivos (JPEG, PDF etc.), este campo deve ser modificado.

Servidor web local

- É possível criar um servidor web no Raspberry Pi para que seus dados estejam acessíveis via internet, seja na rede local ou globalmente. No segundo caso, é necessário adquirir um domínio. Dentre os diversos servidores web disponíveis no Linux, vejamos o `apache`

```
sudo apt-get update  
sudo apt-get install apache2 -y
```


para instalar o `apache2` no Raspberry Pi. Acesse <http://localhost/> em um *browser* no Raspberry Pi, ou [http://IP DO RASPBERRY PI](http://IP_DO_RASPBERRY_PI) em um computador, *tablet* ou celular ligado à mesma rede para visualizar a seguinte página:

Servidor web local

- É possível
este
seg
serv

```
sudo apt-get install apache2  
sudo apt-get install apache2-doc
```

para instalar
browser
computador
seguinte



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in** [/usr/share/doc/apache2/README.Debian.gz](#). Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/  
|-- apache2.conf  
|   |-- ports.conf  
|-- mods-enabled  
|   |-- *.load  
|   |-- *.conf  
|-- conf-enabled  
|   |-- *.conf  
|-- sites-enabled  
|   |-- *.conf
```

dados
e. No
rsos

n um

Servidor web local

- Quando você acessa esta página, você faz um requerimento HTTP GET da página HTML localizada em `/var/www/html/index.html` no Raspberry Pi com o `apache2` instalado e executando.
- Mude esta página HTML para apresentar o que quiser ao usuário.
- (Por precaução, execute `sudo cp /var/www/html/index.html /var/www/html/index_original.html` para manter uma cópia desta página no seu Raspberry Pi.)

Servidor web local

- Você deve ter percebido que a página web funcionou logo após a instalação do `apache2`. Isso quer dizer que ele está executando continuamente. Execute

```
sudo /etc/init.d/apache2 stop  
sudo update-rc.d apache2 disable
```

para parar sua execução, e

```
sudo update-rc.d apache2 enable  
sudo /etc/init.d/apache2 start
```

para reinicia-lo.

Servidor web local

- **Envio de dados do cliente para o servidor:**
 - O envio de informações de um cliente para um servidor pode ser feito através de requisições HTTP GET e POST. Com o `apache2` em execução, execute

```
sudo curl -o /var/www/html/index.html  
https://raw.githubusercontent.com/DiogoCaetanoGarcia/Sistemas_Em  
barcados/master/Code/24_Internet/formulario_RPi1.html  
sudo curl -o /var/www/html/obrigado.html  
https://raw.githubusercontent.com/DiogoCaetanoGarcia/Sistemas_Em  
barcados/master/Code/24_Internet/obrigado.html
```


Servidor web local

- **Envio de dados do cliente para o servidor:**
 - Acesse o servidor para visualizar uma página com um formulário contendo três campos de preenchimento e um menu *dropdown*. Digite qualquer coisa nestes campos, pressione o botão *Submit* e execute `tail -1 /var/log/apache2/access.log` para ver o último acesso feito ao arquivo `/var/www/html/index.html` via internet.
 - Repare que o que você digitou nos campos do formulário aparecem após a palavra GET, e você pode utilizar isto para ler o que o usuário mandou de informação para o seu Raspberry Pi.
 - Execute `sudo curl -o /var/www/html/index.html https://raw.githubusercontent.com/DiogoCaetanoGarcia/Sistemas_Embarcados/master/Code/24_Internet/formulario_RPi2.html` para visualizar um formulário com melhor aparência.

Servidor web local

- **Atualização da página:**
 - Execute

```
cd ~
echo '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><html
xmlns="http://www.w3.org/1999/xhtml"><head><meta
http-equiv="Content-Type" content="text/html; charset=UTF-8"
/><title>Temperatura do Raspberry Pi</title></head><body><div
class="container"><h1>' > index.html
echo $(date) >> index.html
echo '</h1><p>' >> index.html
echo '$(/opt/vc/bin/vcgencmd measure temp) >> index.html
echo '</p></div></body></html>' >> index.html
sudo mv index.html /var/www/html/index.html
```

e acesse o servidor para visualizar uma página com informação da temperatura da GPU do Raspberry Pi. Isto é possível graças ao comando `/opt/vc/bin/vcgencmd measure temp`

Servidor web local

- **Atualização da página:**

- Atualize a página do servidor, e perceba que o valor indicado na página não muda. Isto acontece porque a página HTML não foi atualizada. Execute

```
curl -o update servidor.sh  
https://raw.githubusercontent.com/DiogoCaetanoGarcia/Sistemas_Emb  
arcados/master/Code/24 Internet/update_servidor.sh  
chmod +x update servidor.sh  
./update_servidor.sh
```

para que a página seja atualizada a cada segundo. (Aperte CONTROL-C para parar esta atualização.)

Streaming de video

- Existem diversas soluções para oferecer *streaming* de video via web no Raspberry Pi, usando o *Raspberry Pi Camera Module* ou *webcams* USB:
 - <https://www.linux-projects.org/uv4l/>
 - https://github.com/silvanmelchior/RPi_Cam_Web_Interface
 - <https://motion-project.github.io/index.html>
 - <https://www.videolan.org/vlc/streaming.html>

Referências

- <https://weworkweplay.com/play/automatically-connect-a-raspberry-pi-to-a-wifi-network/>
- <https://curl.haxx.se/book.html>
- <https://www.hostinger.com.br/tutoriais/comando-curl-linux/>
- <https://www.matthuisman.nz/2019/01/download-google-drive-files-wget-curl.html>
- <http://worldtimeapi.org/>
- <https://www.dropbox.com/developers/documentation/http/documentation>
- <https://riptutorial.com/dropbox-api/example/1356/uploading-a-file-via-curl>
- <https://olivermarshall.net/how-to-upload-a-file-to-google-drive-from-the-command-line/>
- <https://www.shellhacks.com/telegram-api-send-message-personal-notification-bot/>

Referências

- <https://thepihut.com/blogs/raspberry-pi-tutorials/using-ifttt-with-the-raspberry-pi>
- <https://yaz.in/p/submitting-a-google-form-using-the-command-line/>
- <https://www.geeksforgeeks.org/using-curl-to-send-email/>
- <https://stackoverflow.com/questions/14722556/using-curl-to-send-email>
- <https://www.commandlinefu.com/commands/view/6716/send-email-with-curl-and-gmail>
- https://raspberrypi-projects.com/pi/software_utilities/email/ssmtp-to-send-emails
- <https://stackoverflow.com/questions/30351465/html-email-with-inline-attachments-and-non-inline-attachments>
- <https://en.wikipedia.org/wiki/MIME>
- <https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md>

Referências

- <https://raspberrypi.stackexchange.com/questions/64115/how-to-stop-apache-from-running-a-local-web-server-on-boot>
- https://www.w3schools.com/howto/howto_css_contact_form.asp
- <https://raspberrypi.stackexchange.com/questions/75248/apache-server-monitoring>
- <https://www.e-tinkers.com/2018/04/how-to-control-raspberry-pi-gpio-via-http-web-server/>
- <https://www.filipeflop.com/blog/streaming-com-raspberry-pi/>
- <https://medium.com/@gonzalovazquez/raspberry-pi-and-motioneye-setting-up-your-own-video-surveillance-95444e0faad>

Referências

- <https://stackoverflow.com/questions/49846400/raspberry-pi-use-vlc-to-stream-webcam-logitech-c920-h264-video-without-transcode>
- <https://raspberrypi.stackexchange.com/questions/4412/streaming-h264-with-logitech-c920>
- <https://raspberrypi-projects.com/pi/pi-hardware/raspberry-pi-camera/streaming-video-using-vlc-player>