**01 Montando circuito no Fritzing**

## Transcrição

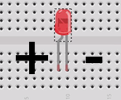
Chegou a hora de fazermos nosso primeiro projeto, e ele será o famoso **Hello World**, só que na versão eletrônica. Vamos utilizar o Raspberry PI para acender um LED. Nesse capítulo usaremos:

* Raspberry PI.
* Protoboard.
* Jumpers.
* Resistor de 220 Ohms ou mais.
* LED.

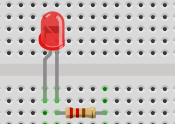
Precisamos que o Raspberry PI esteja ligado e conectado à rede, para podermos fazer o acesso remoto e executar os comandos. E uma recomendação é a instalação do programa **Fritzing**, você pode fazer o download dele [aqui](http://fritzing.org/download/). O Fritzing nos permite montar o nosso projeto na tela do computador, para depois passar para o ambiente físico.

## Hello World

Com a protoboard e o Raspberry PI na tela do Fritzing, pegaremos um LED e podemos reparar que a sua "perna" maior é o seu lado positivo, e a menor o lado negativo, conforme podemos ver na imagem abaixo:



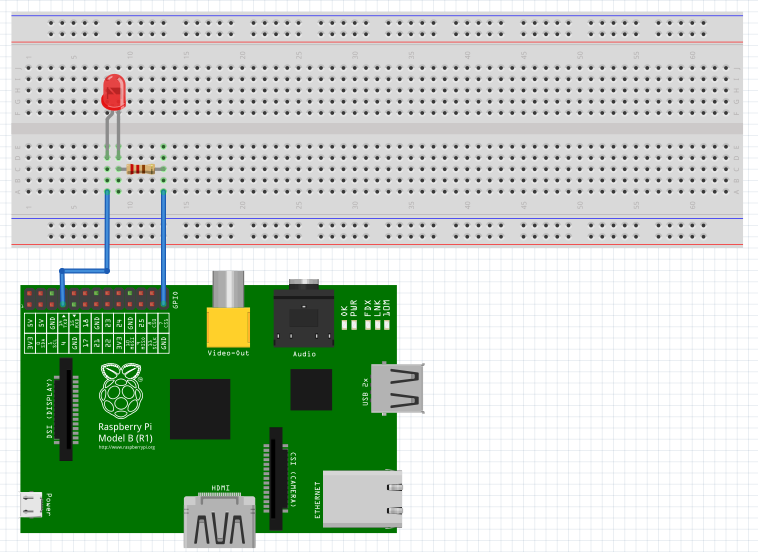
Pegaremos o **resistor de 220 ohms** e ligaremos-o ao lado negativo do LED e à protoboard:



Por último, conectaremos a protoboard ao Raspberry PI. Para isso, precisamos escolher um pino terra (**GND**) e um **GPIO**, escolheremos os pinos **25** e **7**, respectivamente.

Então conectamos **o outro lado do resistor** ao **pino terra** (**pino 25**), e o **lado positivo do led** ao **GPIO** (**pino 7**).

O resultado final ficará assim:



Você pode baixar o projeto pronto [aqui](https://s3.amazonaws.com/caelum-online-public/raspberry2/img/02/led.fzz), mas sempre aconselhamos realmente colocar as mãos na massa e fazer a montagem com o Fritzing antes.

# 02 Tensão do GPIO

**Sobre o GPIO do Raspberry, é correto afirmar que:**

Top of Form

* Alternativa incorreta



Sua tensão de trabalho é de 5V.

* Alternativa incorreta



Sua tensão de trabalho pode variar de 3.3V a 5V.

* Alternativa incorreta



Sua tensão de trabalho é de 3.3V.

* Alternativa incorreta

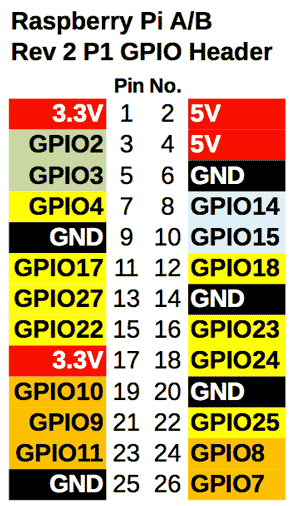


Sua tensão de trabalho sera determinada pela característica da fonte que estiver usando.

Bottom of Form

Diferente de outros equipamentos, como o **Arduino**, o **Raspberry PI** trabalha com **3.3V** em seu **GPIO**, portanto deve-se calcular os circuitos a partir dessa tensão de referência.

Agora você pode reclamar que na imagem dos GPIOs aparece 3,3V e 5V:



Isso é verdade, mas agora estamos falando sobre os GPIOs que podem ser programados ou controlados. Os GPIOs, que possuem números, são programáveis e sempre terão uma saída de 3,3V!

Os pinos 1, 2, 4, 6, 9, 14, 17, 20 e 25 (na imagem acima) são GPIOs especiais e não podem ser controlados.

# 03 O que é um LED?

O termo LED é muito corriqueiro no dia a dia, mas qual é o significado dessa sigla? Qual sua finalidade? Pesquise :)

**Assim que você responder, o instrutor dará a opinião dele e ampliará a discussão sobre o assunto**.

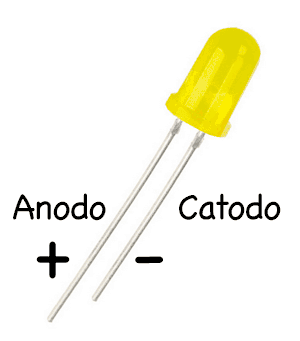
### Opinião do instrutor

A sigla LED significa ***L***ight ***E***mitting ***D***iode (Diodo Emissor de Luz). Um LED é usado para a emissão de luz em locais e instrumentos onde se torna mais conveniente a sua utilização no lugar de uma lâmpada comum.

E Diodo, você sabe o que é?

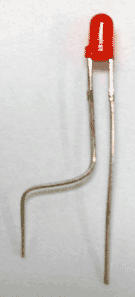
Diodo é um componente eletrônico que faz com que a corrente elétrica circule em sentido único, nunca permitindo que ela circule no sentido oposto.

Certo, mas como visualizar isso em um LED? Vejamos a seguinte foto:



O LED é "aleijado", ele tem uma perna maior do que a outra. Por quê? É uma maneira de distinguir o positivo do negativo. A perna maior sempre é positiva (Anodo) e a menor é sempre negativa (Catodo). É por isso que dizemos que o LED é polarizado (tem dois pólos, positivo e negativo).

Já tentou colocar um LED de pé? Como ele tem uma perna maior do que a outra você terá problemas, é por isso que é comum entornarmos a perna maior para que ambas fiquem com o mesmo tamanho, facilitando assim seu encaixe na protoboard. Vejamos um LED que teve sua perna maior dobrada:



# 04 Esquematizando nossos projetos com o GPIO

**Um excelente programa para nos ajudar a esquematizar nossos projetos com o GPIO é o:**

Top of Form

* Alternativa correta



GPIO-Config

* Alternativa correta



Putty

* Alternativa correta



Raspi-Config

* Alternativa correta



Fritzing

Bottom of Form

O programa **Fritzing** é uma iniciativa Open Source, que irá te auxiliar na esquematização e desenvolvimento de projetos eletrônicos:

[http://fritzing.org/home/](http://fritzing.org/home/" \t "_blank)

No site também tem alguns tutorias disponíveis (a maioria em inglês):

<http://fritzing.org/learning/>

# 05 Mãos na massa: Montando o projeto no Fritzing

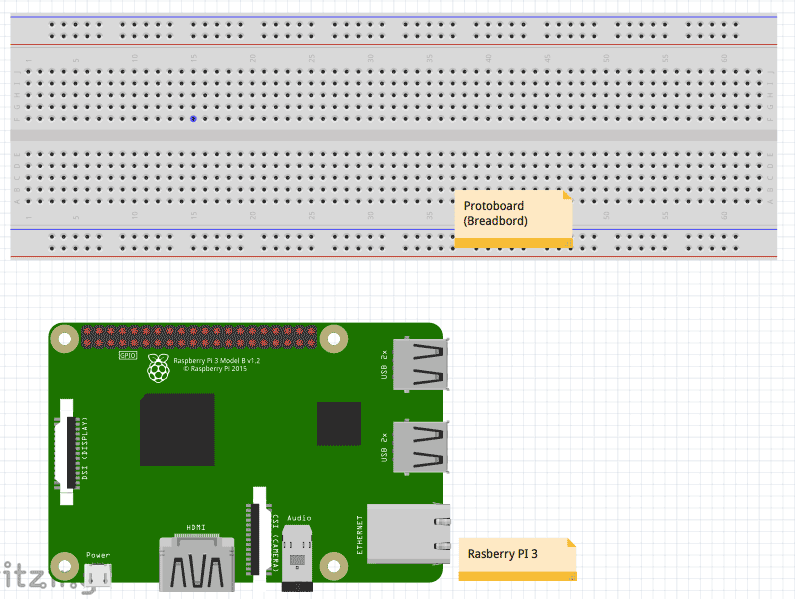
Antes de montar o projeto físico, o circuito, é recomendado montá-lo antes no programa **Fritzing**, ele nos permite montar o nosso projeto na tela do computador. Você pode realizar o download dele aqui: [Fritzing Download](http://fritzing.org/download/).

Uma vez o Fritzing instalado, você pode baixar e abrir esse arquivo do Fritzing: [led-base.fzz](https://s3.amazonaws.com/caelum-online-public/raspberry2/led-base.fzz)

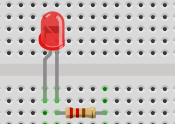
**Abrindo o arquivo você encontrará uma tarefa para resolver**!

*Alternativamente* você pode seguir os passos abaixo:

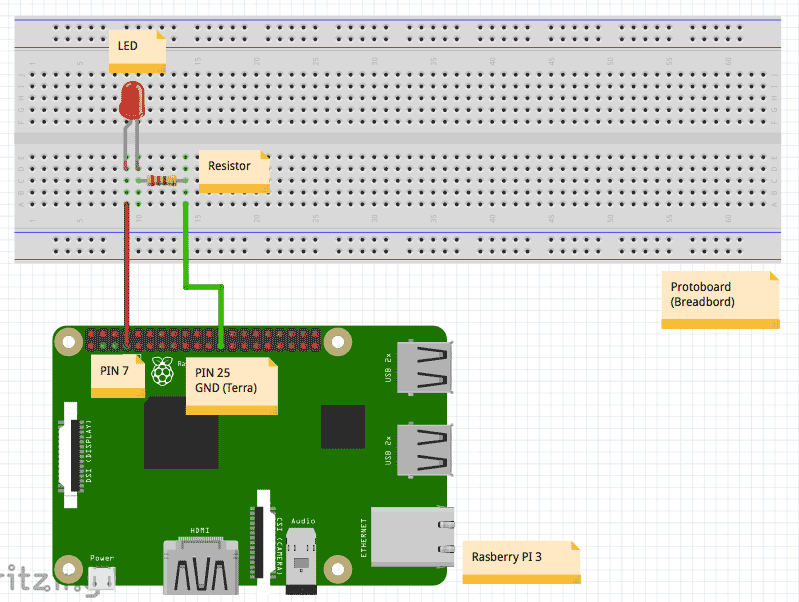
1) Primeiramente adicionamos uma Protoboard e o Raspberry PI:



Pegaremos o resistor de 220 ohms e ligaremos-o ao lado negativo do LED (lembrando que o lado negativo é a "perna menor" do LED, e o positivo é a "perna maior") à Protoboard:



Por último, conectamos o lado positivo do LED ao **pino 7** (equivalente ao **GPIO4**) e o **outro lado do resistor** (o lado que não está ligado ao lado negativo do LED) ao **pino terra 25** (**GND**). O resultado final ficará assim:



Para pegar o arquivo prontinho, basta clicar em *Continuar* :)

# 06 Lei de Ohm

## Transcrição

Estamos utilizando um resistor de **220 ohms**, mas esse não é um valor aleatório, é um valor que precisa ser calculado.

Para isso, precisamos lembrar da **Lei de Ohm**, lá do ensino médio. Essa lei faz a relação entre **tensão**, **resistência** e **intensidade da corrente**, respectivamente, dessa maneira:

V = R \* I

No nosso circuito, temos a saída do Raspberry PI como **3.3V**, diferente do Arduino, que é **5V**. Porém, precisamos ver que para ligar o LED, ele também irá consumir um pouco dessa voltagem. Mais precisamente, o LED vermelho consome **2V**. Substituindo na nossa fórmula, temos:

3.3 - 2 = R \* I

1.3 = R \* I

Agora, precisamos olhar para a nossa corrente elétrica (**I**). No caso, a corrente que o LED vermelho necessita é de **20mA**, ou seja, **0.02A**. Substituindo novamente na nossa fórmula, e realizando a conta:

1.3V = R \* 20mA

1.3 = R \* 0,02A

R = 1.3 / 0,02

R = 65 ohms

Vimos que a resistência necessária para que o nosso LED não queime é de no mínimo **65 ohms**. Mas como estamos falando de resistência, é sempre bom **errarmos para cima**. Ou seja, podemos selecionar a primeira resistência após o valor de 65. No curso estamos utilizando a de **220 ohms**, ela funciona para o projeto, o único detalhe é que o LED acenderá um pouco mais fraco do que com um resistor de valor mais baixo.

# 07 Corrente elétrica

**Vimos no vídeo a Lei de Ohm a fórmula V = R \* I**

**Qual é o significado do I?**

Top of Form

* Alternativa correta



Igualdade da corrente elétrica.

* Alternativa correta



Intensidade da corrente elétrica.

* Alternativa correta



Irregularidade da corrente elétrica.

Bottom of Form

É a **intensidade da corrente elétrica**, no entanto muitas vezes se fala apenas da corrente elétrica (sem a palavra intensidade).

Quando existe uma diferença de potencial elétrico entre componentes (polarizado), acontece um fluxo de partículas para equilibrar o campo elétrico. Esse fluxo é a corrente elétrica.

**A corrente elétrica** é medida em ampère (A), que nada mais é do que a quantidade de elétrons que passa nesse fio em um tempo determinado!

# 08 Calculando o valor do resistor

**Um conhecimento básico em eletrônica certamente irá ajudar muito na elaboração de projetos com o Raspberry PI, por isso é correto afirmar que:**

Top of Form

* Alternativa incorreta



Para calcularmos o valor da tensão, recorremos à **Lei de Ohm**:

V = R / I

* Alternativa incorreta



Para aplicarmos a **Lei das Malhas** (Kircchoff), basta somar a tensão da fonte com as demais tensões em cada componente.

* Alternativa incorreta



Para calcularmos o valor de um resistor, recorremos à **Lei de Ohm**:

R = V / I

* Alternativa incorreta



Um exemplo da aplicação da **Lei de Kirchhoff** (Lei dos Nós) é:

R = V \* I

Bottom of Form

A **Lei de Ohm** é muito importante para que possamos calcular os valores das resistências, um exemplo prático está no dimensionamento do resistor para “proteger” o LED. Essa lei faz a relação entre **tensão**, **resistência** e **intensidade da corrente**, lembrando do "quem **V**ê, **RI**" , dessa maneira:

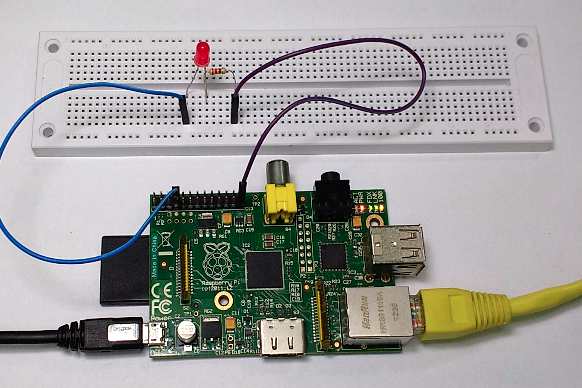
V = R \* I

Ou:

R = V / I

# 09 Montando o Raspberry PI

Com isso, podemos passar o projeto do Fritzing para o ambiente físico:



Para acender o LED, primeiramente devemos nos conectar remotamente ao Raspberry PI, utilizando o SSH ou VNC, como visto no treinamento anterior:

ssh pi@IP\_DO\_RASPBERRY

Como conectamos o LED ao **pino 7**, precisamos defini-lo como saída. Para isso utilizamos o seguinte comando:

gpio mode 7 out

Após isso, escrevemos (***write***) que queremos a porta do pino 7 em alta (ligada). Para representar uma porta em alta, utilizamos o número **1**:

gpio write 7 1

Após esse comando, o LED acende! Para apagá-lo, colocamos a porta em baixa (número **0**):

gpio write 7 0

Então a partir de agora, a nossa intenção é aprender essa interação com o GPIO, fazendo isso via linha de comando e via Python!

# 10 Configurando um pino como saída

**Através do console (terminal), é possível definir se uma porta irá trabalhar com entrada ou saída. Imaginando que você queira configurar o pino 11 como saída, é necessário utilizar o seguinte comando:**

Top of Form

* Alternativa correta



gpio set 11 out

* Alternativa correta



gpio 11 mode out

* Alternativa correta



GPIO-Config 11 out

* Alternativa correta



gpio mode 11 out

Bottom of Form

Através do comando **gpio**, podemos "conversar" diretamente com o GPIO sem utilizar uma linguagem de programação, como o Python. A sintaxe correta do comando para definição da porta como saída é:

gpio mode NUMERO\_PINO out

Considerando o pino 11, a sintaxe fica:

gpio mode 11 out

Lembrando que a saída é sempre de 3.3V.

# 11 Interpretando um programa

**Dado o programa abaixo:**

**gpio mode 7 out**

**gpio write 7 0**

**Qual a sua funcionalidade?**

Top of Form

* Alternativa correta



Definir o pino 7 como saída e configurá-lo com **HIGH**.

* Alternativa correta



Definir o pino 7 como saída e configurá-lo com **LOW**.

* Alternativa correta



Alternar o valor da porta 7 de **LOW** para **HIGH**.

* Alternativa correta



Definir o pino 7 como entrada e configurá-lo com **LOW**.

Bottom of Form

A sequência acima faz com que primeiramente seja definido o pino 7 como **SAÍDA**:

gpio mode 7 out

E depois configurado com **LOW** (**0**):

gpio write 7 0

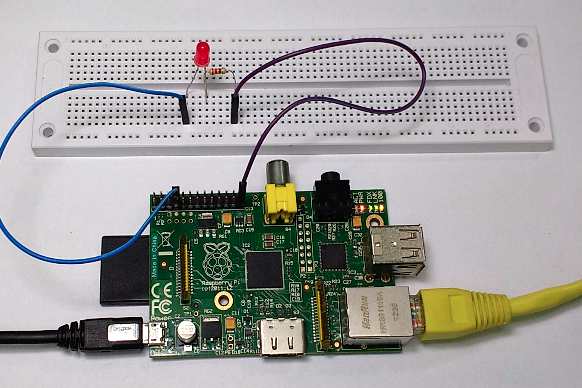
Para configurar ele como **HIGH**, podemos usar:

gpio write 7 1

**HIGH** ou **1**, significa 3.3V.

# 12 Mãos na massa: Montando o circuito e acendendo o led

O projeto montado deve ficar semelhante a esse:



Para acendermos o LED, precisamos estar conectados ao Raspberry PI, seja com teclado e monitor, ou remotamente (SSH ou VNC).

Como o LED está conectado ao **pino 7**, é nele que precisamos trabalhar. Primeiro definindo-o como **saída**:

gpio mode 7 out

Para acender o LED, **escrevemos** (***write***) no **pino 7**, ligando-o (número **1**):

gpio write 7 1

Para apagar o LED, basta desligar a porta (número **0**):

gpio write 7 0

### Opinião do instrutor

Aprendemos o básico para interagir com o GPIO. No próximo capítulo, a intenção é aprender essa interação com o GPIO via Python!