4ª edição

COMPLETAMENTE ATUALIZADO PARA **RASPBERRY PI 400**

O GUIA PARA INICIANTES DO Raspberry Pi Oficial

Como utilizar o seu novo computador

por Gareth Halfacree

O GUIA PARA INICIANTES DO Raspberry Pi Oficial

Como utilizar o seu novo computador

Publicado pela primeira vez em 2021 pela Raspberry Pi Trading Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS

Diretor de publicação: Russell Barnes • Editor: Phil King
Design: Critical Media • Ilustrações: Sam Alder
CEO: Eben Upton

ISBN: 978-1-912047-80-2

Bem-vindo ao Guia para Iniciantes do Raspberry Pi oficial

Achamos que vai adorar o Raspberry Pi. Qualquer que seja o modelo que possui – uma placa Raspberry Pi padrão, ou o novo Raspberry Pi 400 com teclado integrado – este computador acessível pode ser utilizado para aprender a programar, construir robôs e criar todos os tipos de projetos estranhos e maravilhosos.

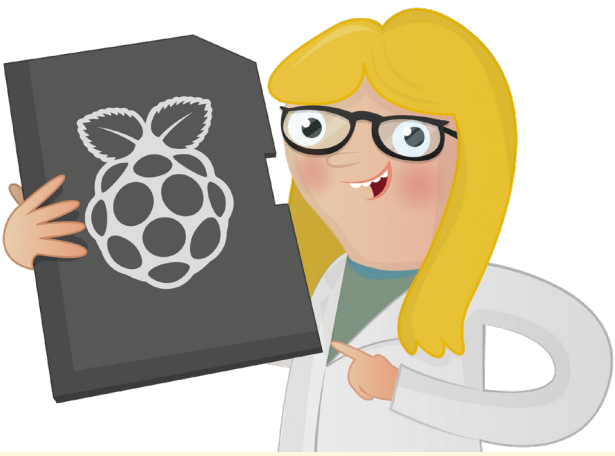
O Raspberry Pi é capaz de fazer tudo o que se espera de um computador – incluindo navegar na Internet e jogar jogos, ver filmes e ouvir música. Mas o Raspberry Pi é muito mais do que um computador moderno.

Com um Raspberry Pi, pode entrar no coração de um computador. Pode configurar o seu próprio sistema operativo e ligar fios e circuitos diretamente aos respetivos pinos GPIO. Foi concebido para ensinar os jovens a programar em linguagens como Scratch e Python, e todas as principais linguagens de programação estão incluídas no sistema operativo oficial.

O mundo precisa mais do que nunca de programadores, e o Raspberry Pi despertou o amor pela ciência da computação e pela tecnologia na nova geração.

Pessoas de todas as idades utilizam o Raspberry Pi para criar projetos emocionantes: tudo desde consolas de jogos retro até estações meteorológicas ligadas à Internet.

Se quiser criar jogos, construir robôs ou aprender uma variedade de projetos incríveis, este livro está aqui para ajudá-lo a começar.



Sobre o autor

Gareth Halfacree é um jornalista de tecnologia freelance, escritor e ex-administrador de sistemas no setor educacional. Com uma paixão por software e hardware de código aberto, foi um dos primeiros a adotar a plataforma Raspberry Pi e escreveu várias publicações sobre as suas capacidades e flexibilidade. Encontra-se no Twitter como **@ghalfacree** ou no seu Web site em **freelance.halfacree.co.uk**.

Índice

APÊNDICES

Capítulo 1

Conheça o seu Raspberry Pi

Familiarize-se com o seu novo computador do tamanho de um cartão de crédito, fazendo uma visita guiada ao Raspberry Pi. Descubra os seus numerosos componentes e o que fazem

O Raspberry Pi é um dispositivo notável: um computador totalmente funcional num pacote minúsculo e de baixo custo. Se está à procura de um dispositivo que possa usar para navegar na Web ou jogar jogos, se está interessado em aprender a programar os seus próprios programas, ou se quer criar os seus próprios circuitos e dispositivos físicos, o Raspberry Pi – e a sua incrível comunidade – irá apoiá-lo ao longo de todo o processo.

O Raspberry Pi é conhecido como um *computador de placa única*, o que significa exatamente o que parece: é um computador, tal como um computador de secretária, portátil ou smartphone, mas construído sobre uma única *placa de circuito impresso*. Como a maioria dos computadores de placa única, o Raspberry Pi é pequeno – com aproximadamente a área útil de um cartão de crédito – mas isso não significa que não seja poderoso: um Raspberry Pi pode fazer tudo o que um computador maior e com mais consumo de energia pode fazer, embora não seja necessariamente tão rápido.

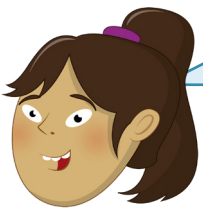
A família Raspberry Pi nasceu de um desejo de incentivar uma educação mais prática sobre computadores em todo o mundo. Os seus criadores, que se juntaram para formar a Fundação sem fins lucrativos Raspberry Pi, não tinham ideia de que este se tornaria tão popular: os poucos milhares construídos em 2012 para testar as águas ficaram imediatamente esgotados, e foram enviados milhões de unidades para todo o mundo nos anos seguintes. Estas placas foram enviadas para casas, salas de aula, escritórios, centros de dados, fábricas e até mesmo barcos autónomos e balões de exploração espacial.

Foram lançados vários modelos do Raspberry Pi desde o Modelo B original, cada um com novas especificações melhoradas ou funcionalidades específicas para um caso de uso particular. A família Raspberry Pi Zero, por exemplo, é uma versão minúscula do Raspberry Pi em tamanho

real sem algumas caraterísticas – em particular as várias portas USB e a porta de rede com fios – em favor de um layout significativamente menor e requisitos de energia reduzidos.

No entanto, todos os modelos Raspberry Pi têm uma coisa em comum: são *compatíveis*, o que significa que o software escrito para um modelo pode ser executado em qualquer outro modelo. Até pode ter a última versão do sistema operativo do Raspberry Pi e executá-la num protótipo original de pré-lançamento do Modelo B. É verdade que será executado mais devagar, mas ainda assim será executado.

Neste livro irá aprender sobre o Raspberry Pi 4 Modelo B e o Raspberry Pi 400, as mais recentes e mais poderosas versões do Raspberry Pi. O que aprende, porém, pode ser facilmente aplicado a outros modelos da família Raspberry Pi, por isso não se preocupe se estiver a usar uma versão diferente.

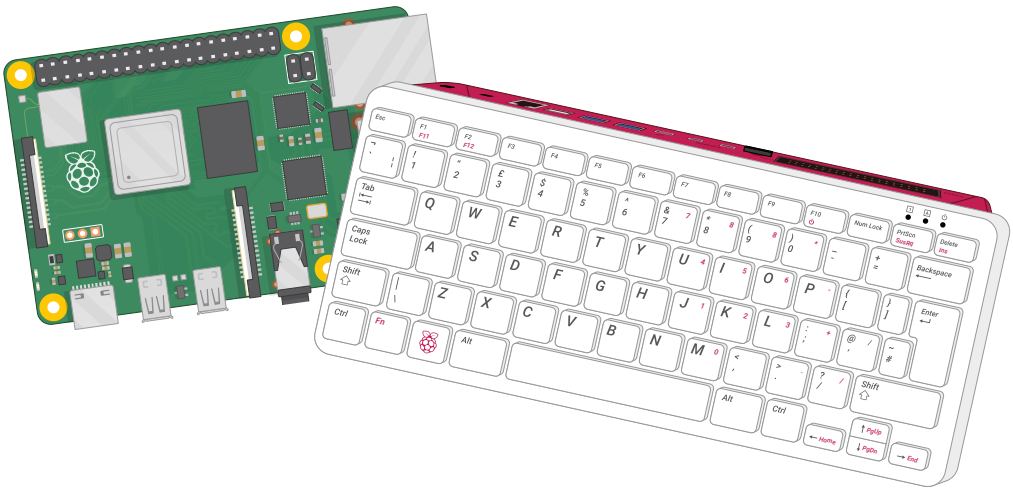


RASPBERRY PI 400

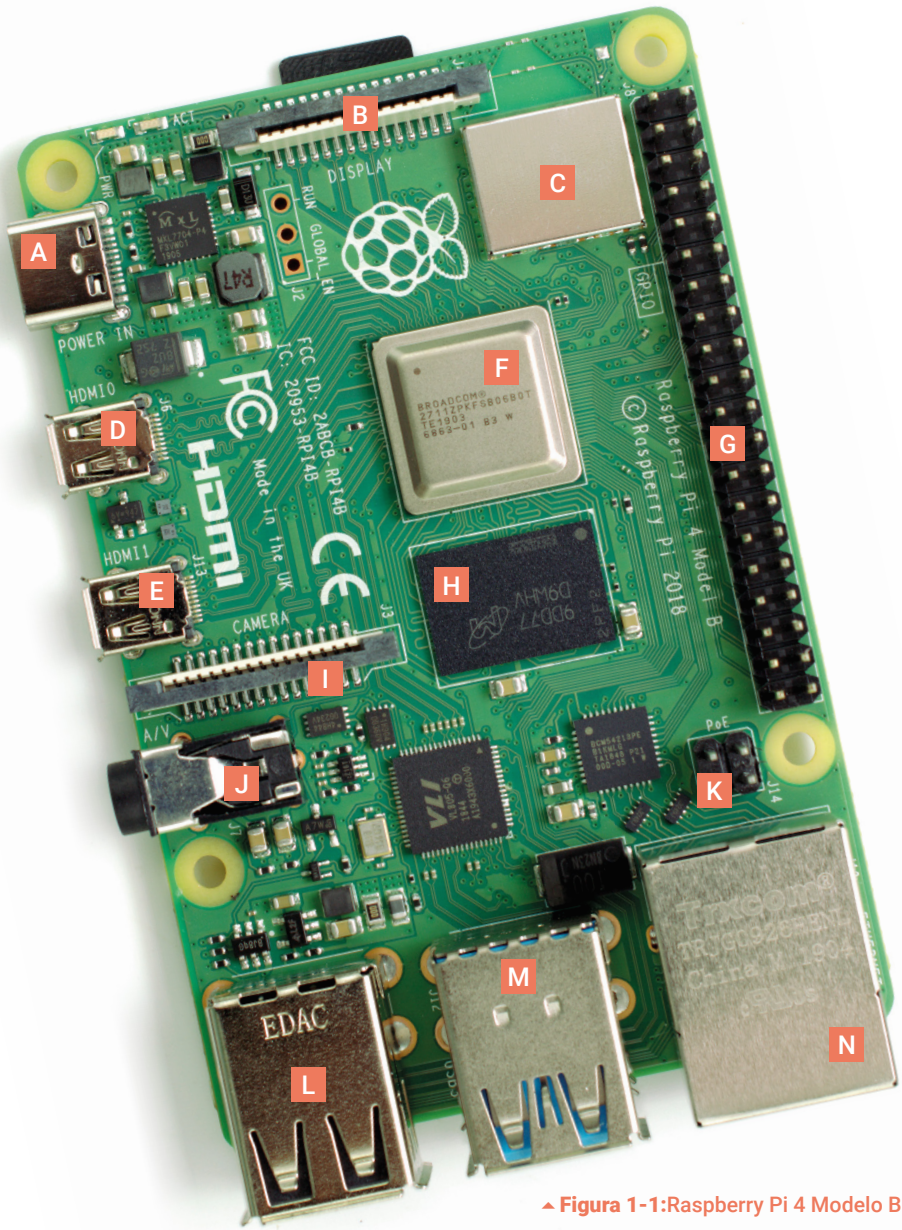
Se tiver um Raspberry Pi 400, a placa de circuito está integrada na caixa do teclado. Continue a ler para obter mais informações sobre todos os componentes que fazem o Raspberry Pi funcionar, ou avance para a página 20 para fazer uma visita guiada ao seu dispositivo.

Uma visita guiada ao Raspberry Pi

Ao contrário de um computador tradicional, que esconde as respetivas peças dentro de uma caixa, um Raspberry Pi padrão tem todos os seus componentes, portas e funcionalidades à vista – embora possa comprar uma caixa para fornecer proteção extra, se preferir. Isto torna-o uma excelente ferramenta para aprender o que as várias partes de um computador fazem, e também facilita a aprendizagem da colocação dos componentes quando for o momento de ligar os vários extras – conhecidos como *periféricos* – que precisará para começar.



- A Alimentação USB Tipo C
- B Porta de monitor DSI
- C Sem fios / Bluetooth
- D Micro-HDMI 0
- E Micro-HDMI 1
- F Sistema em chip
- G GPIO
- H RAM
- I Porta da câmara CSI
- J Ficha AV de 3,5 mm
- K PoE
- L USB 2.0
- M USB 3.0
- N Porta de Ethernet



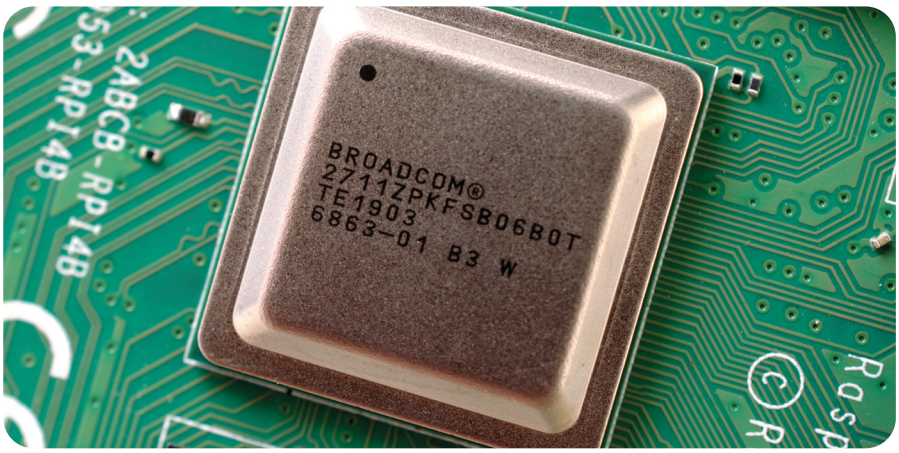
▲ **Figura 1-1:** Raspberry Pi 4 Modelo B

Figura 1-1 mostra um Raspberry Pi 4 Modelo B visto de cima. Quando estiver a usar um Raspberry Pi com este livro, tente mantê-lo na mesma posição da foto; se estiver virado, pode ser confuso quando for necessário usar coisas como o Conetor GPIO (detalhado em **Capítulo 6, Computação física com Scratch e Python**).

Embora possa parecer que há muito nesta placa minúscula, um Raspberry Pi é muito fácil de perceber – a começar pelos respetivos *componentes*, o funcionamento interno que faz o dispositivo funcionar.

Componentes do Raspberry Pi

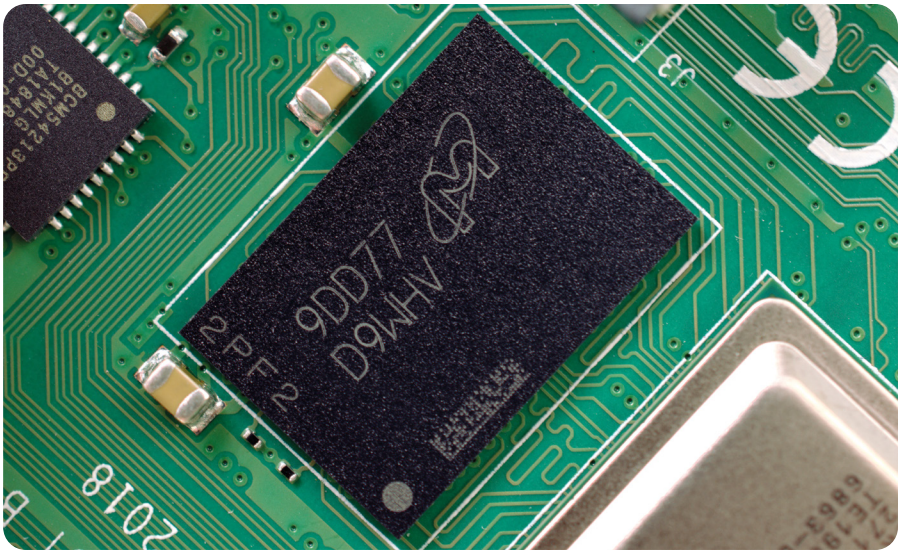
Como qualquer computador, o Raspberry Pi é composto por vários componentes, cada um dos quais tem um papel a desempenhar no respetivo funcionamento. O primeiro e, sem dúvida o mais importante, pode encontrá-lo logo acima do ponto central na parte superior da placa (**Figura 1-2**), coberta por uma tampa metálica: o *sistema em chip* (SoC).



▲ **Figura 1-2:** Sistema em chip do Raspberry Pi (SoC)

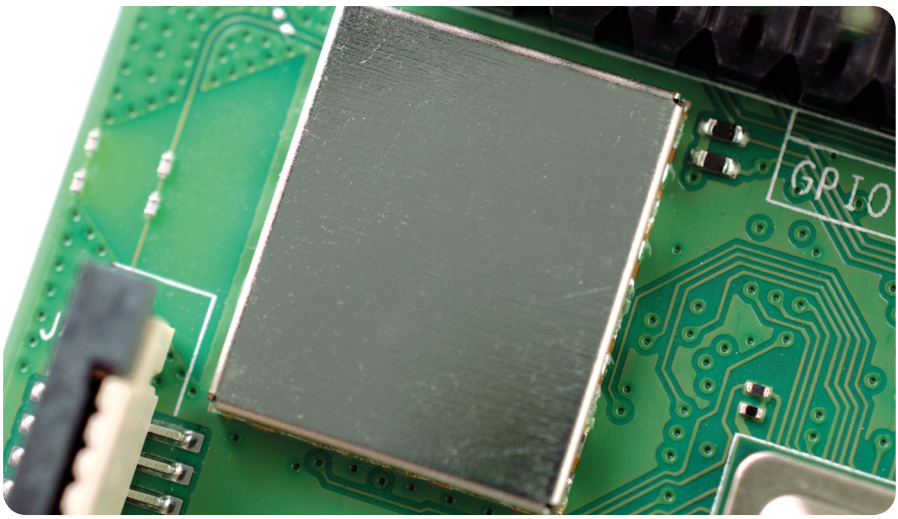
O nome sistema em chip é um grande indicador do que encontraria se forçasse a abertura da tampa metálica: um chip de silício, conhecido como um *circuito integrado,* que contém a maior parte do sistema do Raspberry Pi. O que inclui a *unidade de processamento central* (CPU), geralmente considerada o "cérebro" de um computador, e a *unidade de processamento gráfico* (GPU), que processa os aspetos visuais.

Um cérebro não funciona sem memória, no entanto, junto ao SoC encontrará exatamente isso: outro chip, que parece um pequeno quadrado preto de plástico (**Figura 1-3**, no verso). Esta é a *memória de acesso aleatório (RAM)* do Raspberry Pi. Quando está a trabalhar no Raspberry Pi, é a RAM que guarda o que está a fazer; só quando guarda o seu trabalho é que este será escrito no cartão microSD. Juntos, estes componentes formam as memórias voláteis e não voláteis do Raspberry Pi: a RAM volátil perde o respetivo conteúdo sempre que o Raspberry Pi é desligado, enquanto o cartão microSD não volátil mantém os respetivos conteúdos.



▲ **Figura 1-3:** Memória de acesso aleatório (RAM) do Raspberry Pi

Na parte superior direita da placa encontrará outra tampa metálica (**Figura 1-4**) cobrindo a ligação de *rádio*, o componente que dá ao Raspberry Pi a capacidade de comunicar com os dispositivos sem fios. A própria ligação rádio atua como dois componentes principais, na verdade: um *Rádio Wi-Fi* para ligação a redes de computadores; e um *Rádio Bluetooth* para ligação a periféricos, como ratos, e para enviar ou receber dados de dispositivos inteligentes próximos, como sensores ou smartphones.



▲ **Figura 1-4:** Módulo de rádio do Raspberry Pi

Outro chip preto, coberto de plástico, pode ser visto na extremidade inferior da placa, mesmo atrás do conjunto central de portas USB. Este é o *Controlador USB*, e é responsável pelo funcionamento das quatro portas USB. Junto, há um chip ainda mais pequeno, o *controlador de rede*, que lida com a porta de rede Ethernet do Raspberry Pi. Um último chip preto, mais pequeno que os restantes, encontra-se um pouco acima da tomada de alimentação USB Tipo C na parte superior esquerda da placa (**Figura 1-5**); que é conhecido como um *circuito integrado de gestão de energia (PMIC)*, e transforma a energia que vem da porta micro USB na energia que o Raspberry Pi precisa para funcionar.

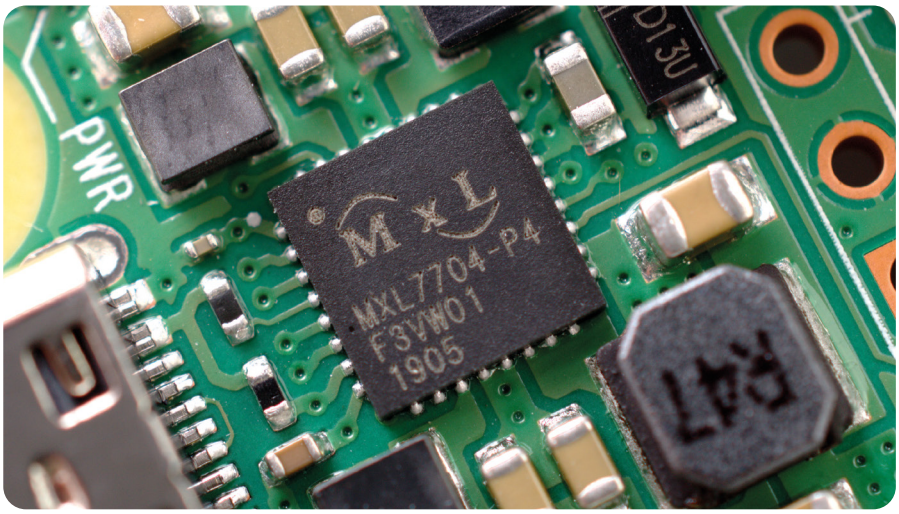


Figura 1-5: Circuito integrado de gestão de energia (PMIC) do Raspberry Pi

Não se preocupe se isto parecer muita informação para assimilar: não precisa de saber o que é cada componente ou onde encontrá-lo na placa para poder usar o Raspberry Pi.

Portas do Raspberry Pi

O Raspberry Pi tem uma série de portas, começando com quatro *Portas de Barramento Série Universal (USB)* (**Figura 1-6**) no meio e no lado direito da extremidade inferior da placa. Estas portas permitem ligar qualquer periférico compatível com USB, desde teclados e ratos a câmaras digitais e unidades flash, ao Raspberry Pi. Em linguagem técnica, existem dois tipos de portas USB: as com partes pretas no interior são portas USB 2.0, baseadas na versão dois do padrão do Barramento Série Universal; as com partes azuis são portas USB 3.0 mais rápidas, baseadas na versão três, mais recente.

▲ **Figura 1-6:** Portas USB do Raspberry Pi

À direita das portas USB está uma *Porta de Ethernet* também conhecida como uma *porta de rede* (**Figura 1-7**). Pode usar esta porta para ligar o Raspberry Pi a uma rede de computadores com fios usando um cabo com um conetor RJ45 na respetiva ponta. Se olhar atentamente para a porta Ethernet, verá dois díodos emissores de luz (LEDs) na parte inferior; estes são LEDs de estado, e indicam-lhe de que a ligação está a funcionar.

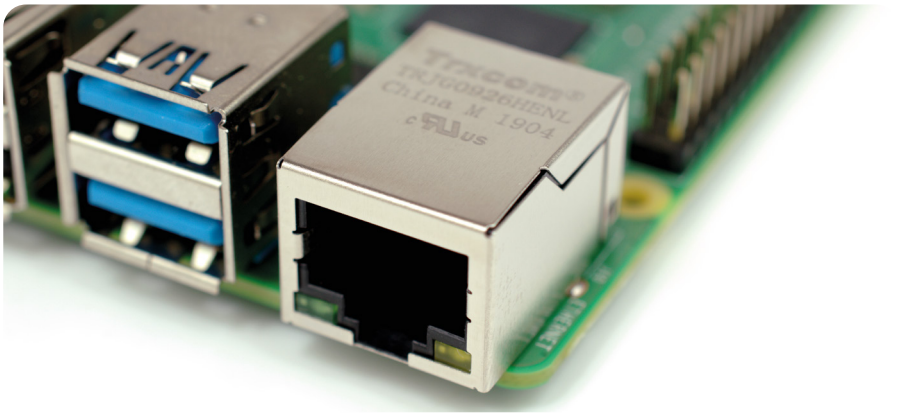


Figura 1-7: Porta de Ethernet do Raspberry Pi

Logo acima das portas USB, na extremidade esquerda do Raspberry Pi, está uma *tomada audiovisual (AV) de 3,5 mm* (**Figura 1-8**). Que também é conhecida como a *tomada para auscultadores*, e pode ser usada exatamente para esse fim – embora obtenha melhor sonoridade se a ligar a colunas amplificadas em vez de a auscultadores. No entanto, tem uma caraterística extra e oculta: além do áudio, a tomada AV de 3,5 mm transporta um sinal de vídeo que pode ligar a TVs, projetores e outros monitores que suportem um *sinal de vídeo composto* usando um cabo especial conhecido como um adaptador da *ficha para auscultadores (TRRS)*.

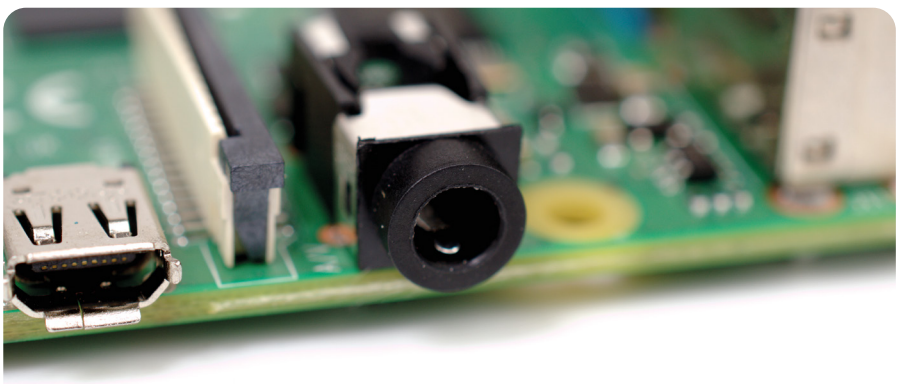


Figura 1-8: Tomada AV de 3,5 mm do Raspberry Pi

Diretamente acima da tomada AV de 3,5 mm está um conetor de aspeto estranho com uma tampa de plástico que pode ser puxada para cima; que é o *conetor da câmara*, também conhecido como *Interface Série da Câmara (CSI)* (**Figura 1-9**). Isso permite que use o Módulo Câmera do Raspberry Pi (sobre o qual obterá mais informações em **Capítulo 8, Módulo Câmera do Raspberry Pi** .)

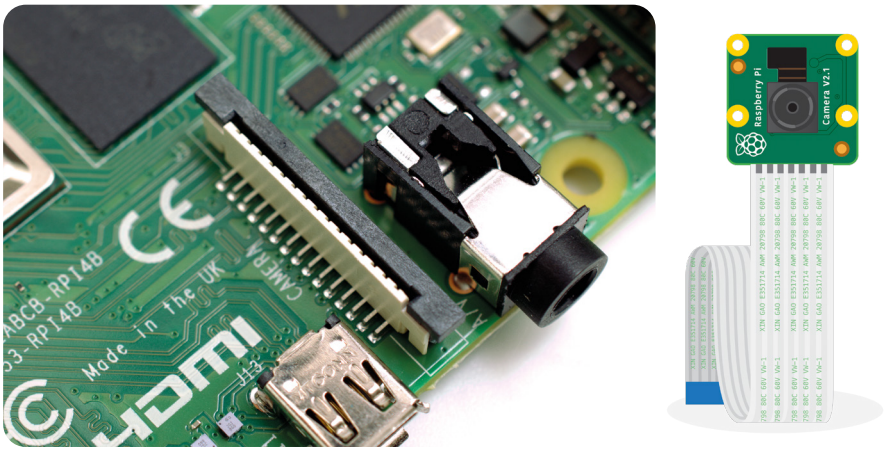


Figura 1-9: Conetor da câmara do Raspberry Pi

Acima disso, ainda na extremidade esquerda da placa, estão as *portas micro Interface Multimédia de Alta Definição (micro-HDMI)*, que são uma versão mais pequena dos conetores que encontra numa consola de jogos, descodificador, ou TV (**Figura 1-10**). A parte multimédia do respetivo nome indica que transporta sinais de áudio e vídeo, enquanto que a alta definição indica que pode esperar uma excelente qualidade. Usa-as para ligar o Raspberry Pi a um ou dois dispositivos de visualização: um monitor de computador, TV, ou projetor.

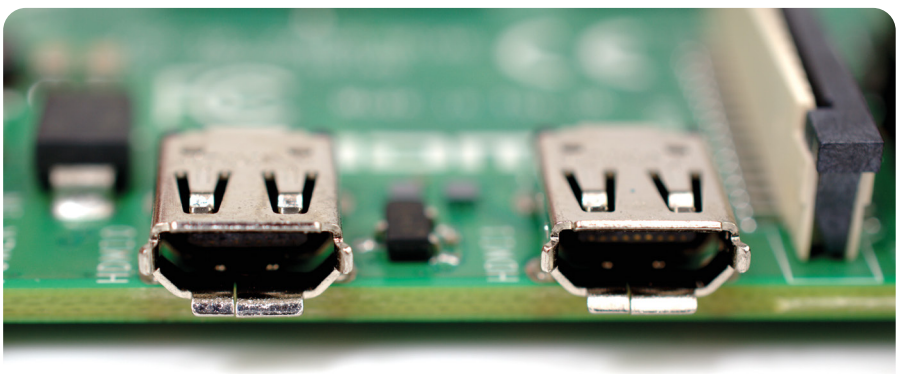


Figura 1-10: Portas micro-HDMI do Raspberry Pi

Acima das portas HDMI está uma *Porta de alimentação USB Tipo C* (**Figura 1-11**), que usará para ligar o Raspberry Pi a uma fonte de alimentação. A porta USB Tipo C é muito comum em smartphones, tablets e outros dispositivos portáteis. Apesar de ser possível usar um carregador de telemóvel padrão para alimentar o Raspberry Pi, para obter melhores resultados deve usar a fonte de alimentação USB Tipo C oficial do Raspberry Pi.

Figura 1-11: Porta de alimentação USB Tipo C do Raspberry Pi

Na extremidade superior da placa está outro conetor de aspeto estranho (**Figura 1-12**), que à primeira vista parece ser idêntico ao conetor da câmara. Este é exatamente o oposto: um *conetor de visualização* ou *Interface Série de Visualização (DSI)* concebido para ser utilizado com o ecrã tátil Raspberry Pi (**Figura 1-13**, no verso).

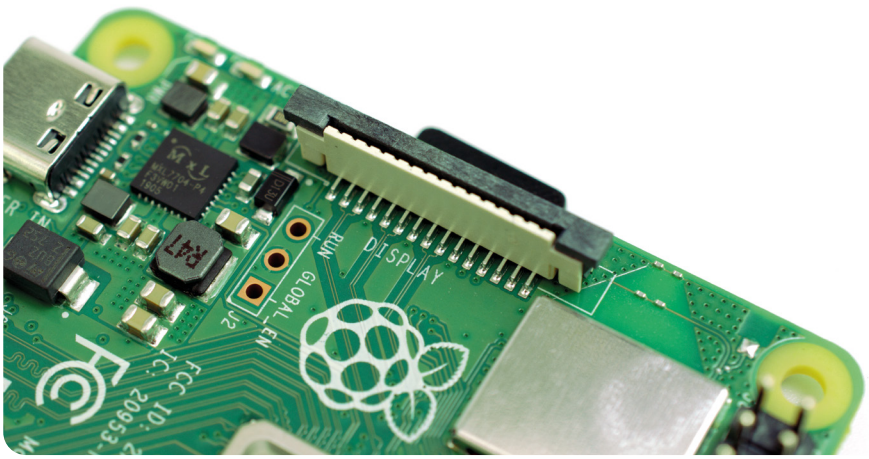


Figura 1-12: Conetor de visualização do Raspberry Pi (DSI)

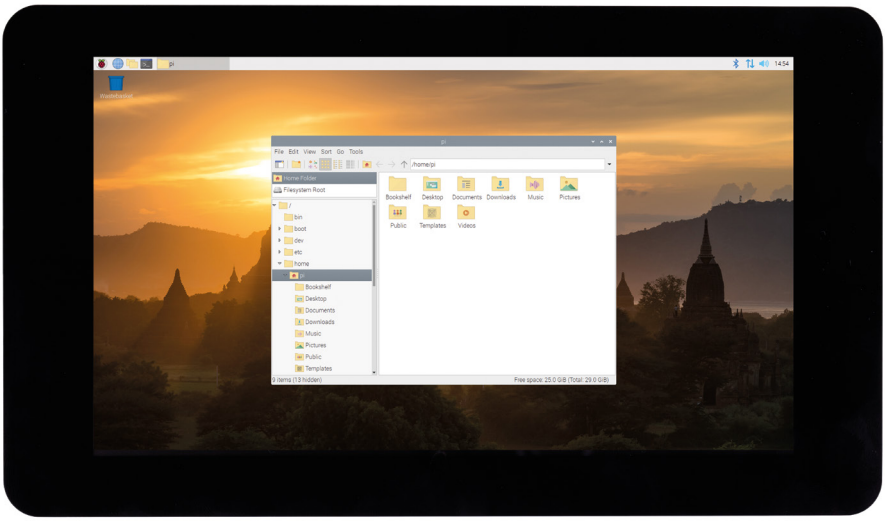


Figura 1-13: O ecrã tátil do Raspberry Pi

Na margem direita da placa encontrará 40 pinos metálicos, divididos em duas fileiras de 20 pinos (**Figura 1-14**). Este é o *Conetor GPIO (entrada/saída para fins gerais)*, uma característica do Raspberry Pi usada para comunicar com hardware adicional, de LEDs e botões até sensores de temperatura, joysticks e medidores de tensão arterial. Mais informações sobre o conetor GPIO em **Capítulo 6, Computação física com Scratch e Python**. Logo abaixo e à esquerda deste conetor está outro conetor mais pequeno com quatro pinos: este é usado para ligar o Power over Ethernet (PoE) HAT, um suplemento opcional que permite ao Raspberry Pi receber energia de uma ligação de rede em vez da porta USB Tipo C.

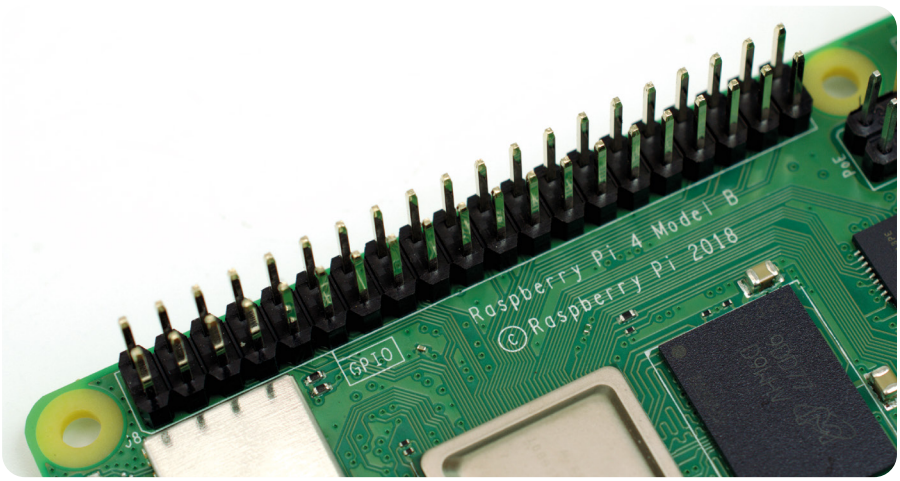


Figura 1-14: Conetor GPIO do Raspberry Pi

Há uma última porta no Raspberry Pi, mas não conseguirá vê-la na parte de cima. Vire a placa para encontrar um *conetor de cartão microSD* no lado oposto do conetor de visualização na placa (**Figura 1-15**). Este é o armazenamento do Raspberry Pi: o cartão microSD aqui introduzido contém todos os ficheiros que guardar, todo o software que instalar e o sistema operativo que o Raspberry Pi executa.

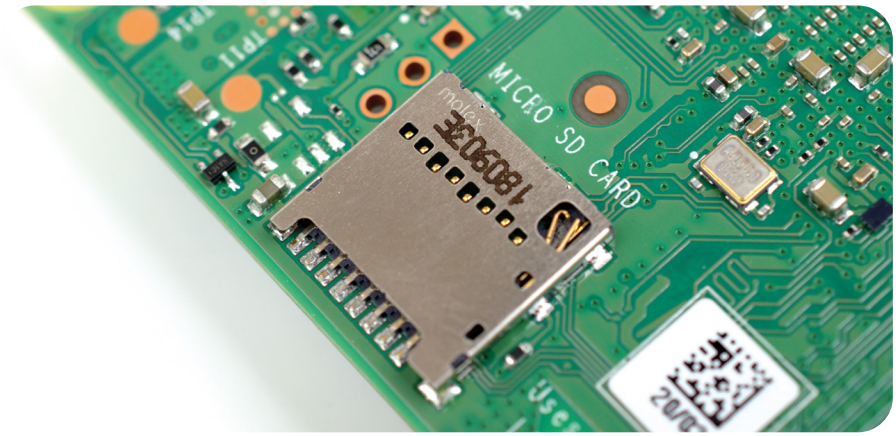


Figura 1-15: Conetor microSD do Raspberry Pi

▲ **Figura 1-16:** O Raspberry Pi 400 tem um teclado integrado

Raspberry Pi 400

O Raspberry Pi 400 tem os mesmos componentes que o Raspberry Pi 4 e coloca-os dentro de uma caixa de teclado. Além de protegê-los, a caixa do teclado ocupa menos espaço na sua mesa e ajuda a manter os cabos mais arrumados.

O Raspberry Pi 400 é composto pelos mesmos componentes centrais do Raspberry Pi 4, incluindo o sistema em chip e a memória. Não os consegue ver, mas estão lá. O que pode ver são as partes externas, começando com o teclado (**Figura 1-16**). No canto superior direito estão três díodos emissores de luz (LEDs): o primeiro acende quando prime a tecla Num Lock, que liga algumas das teclas para agir como um teclado numérico num teclado de tamanho completo; o segundo acende quando prime o Caps Lock, que faz com que as teclas das letras sejam maiúsculas em vez de minúsculas; e o último acende quando Raspberry Pi 400 é ligado.

Na parte traseira do Raspberry Pi 400 (**Figura 1-17**) estão localizadas as portas. A porta mais à esquerda, vista por trás, é o conetor de entrada/saída para fins gerais (GPIO). Este é o mesmo conetor que é descrito na página 17, mas invertido: o primeiro pino, o Pino 1, está na parte superior direita, enquanto o último pino, o Pino 40, está na parte inferior esquerda. Pode obter mais informações sobre o conetor GPIO no **Capítulo 6, Computação física com Scratch e Python.**

Figura 1-17: As portas encontram-se na traseira do Raspberry Pi 400

Junto ao conetor GPIO está a ranhura para o cartão microSD. Este armazena o cartão microSD que atua como o armazenamento do Raspberry Pi 400 para o sistema operativo, aplicações e outros dados. O cartão microSD vem pré-instalado no Raspberry Pi 400; pode removê-lo empurrando suavemente o cartão até este fazer um clique, sair e depois puxá-lo para fora. Quando voltar a colocar o cartão, certifique-se de que os contactos metálicos brilhantes estão virados para baixo; o cartão deve deslizar para dentro com um clique suave.

As duas portas seguintes são as portas do micro-HDMI, para ligar a um monitor, TV ou outro dispositivo de visualização. Como o Raspberry Pi 4, o Raspberry Pi 400 suporta até dois dispositivos de visualização. Ao lado destes está a porta de alimentação USB Tipo C, para ligação à fonte de alimentação do Raspberry Pi ou uma fonte de alimentação USB compatível.

As duas portas azuis são portas USB 3.0, que fornecem uma ligação de alta velocidade a dispositivos onde se incluem unidades de estado sólido (SSDs), pens USB, impressoras, e muito mais. A porta branca à direita destas é uma porta USB 2.0 de baixa velocidade, à qual pode ligar o Rato Raspberry Pi incluído no pacote.

A última porta é uma porta de rede Ethernet Gigabit, que lhe permite ligar o Raspberry Pi 400 à sua rede usando um cabo de rede como alternativa à utilização da rede Wi-Fi sem fios integrada. Pode obter mais informações sobre como ligar o Raspberry Pi 400 a uma rede no **Capítulo 2, Começar com o seu Raspberry Pi.**

Capítulo 2

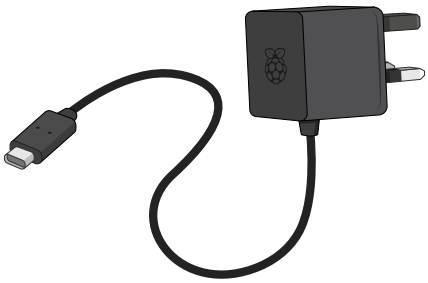
Começar com o seu Raspberry Pi

Descubra os artigos essenciais que precisará para o seu Raspberry Pi e como ligá-los a todos para configurar e pôr o Raspberry Pi a funcionar

O Raspberry Pi foi concebido para ser o mais rápido e fácil de configurar e usar, mas, como qualquer computador, depende de vários componentes externos, denominados *periféricos*. Quando olha para a placa do circuito impresso aberta do Raspberry Pi, com um aspeto muito diferente dos computadores instalados em caixas a que está habituado, pode ficar preocupado porque parece complicado, mas isso não é verdade. O Raspberry Pi pode estar pronto a usar em menos de dez minutos; para isso, siga os passos deste guia.

Se recebeu este livro juntamente com o Kit completo Raspberry Pi Raspberry Pi ou com um Raspberry Pi 400, já terá quase tudo o que precisa para começar: basta ter um monitor de computador ou uma TV com uma ligação HDMI – o mesmo tipo de conetor usado por caixas descodificadoras, leitores Blu-ray e consolas de jogos – para que possa ver o que o seu Raspberry Pi está a fazer.

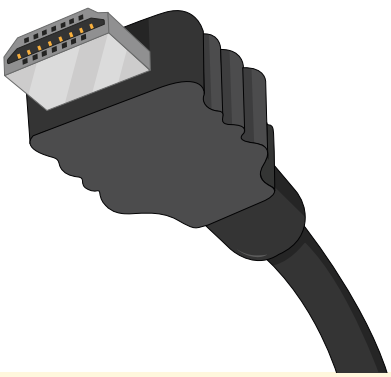
Se comprou o seu Raspberry Pi sem acessórios, vai precisar também de:



- **Fonte de alimentação USB** – Uma fonte de alimentação de 5 V com corrente de 3 amperes (3 A) e com um conetor USB Tipo C. A Fonte de Alimentação oficial do Raspberry Pi é a escolha recomendada, uma vez que é compatível com as necessidades de energia de comutação rápida do Raspberry Pi.

- **cartão microSD com NOOBS** – O cartão microSD funciona como o armazenamento permanente do Raspberry Pi; todos os ficheiros que cria e o software que instala, juntamente com o próprio sistema operativo, são armazenados no cartão. Um cartão de 8 GB será suficiente para começar, contudo um de 16 GB oferece mais espaço para crescer. Utilizando um cartão com o NOOBS (Novo Software Predefinido) pré-instalado irá poupar-lhe tempo; caso contrário, veja o **Apêndice A** para obter as instruções de instalação de um sistema operativo (SO) num cartão vazio.

- **Teclado e rato USB** – O teclado e o rato permitem-lhe controlar o seu Raspberry Pi. Quase todos os teclados e ratos com ou sem fios com conetor USB funcionarão com o Raspberry Pi, embora alguns teclados "gaming" com luzes coloridas possam consumir demasiada energia para serem usados de forma fiável.



- **Cabo Micro-HDMI** – Transporta o som e as imagens do Raspberry Pi para a sua TV ou monitor. Uma ponta do cabo tem um conetor micro-HDMI para o Raspberry Pi; a outra, um conetor HDMI de tamanho completo para o seu dispositivo de visualização. Em alternativa, pode usar um adaptador micro-HDMI para HDMI e um cabo HDMI padrão de tamanho completo. Se usar um monitor sem uma tomada HDMI, pode comprar adaptadores micro-HDMI para DVI-D, DisplayPort ou VGA. Para ligar a uma TV antiga que usa vídeo composto ou tem uma tomada SCART, use um cabo de áudio/vídeo para auscultadores (TRRS) de 3,5 mm.

É seguro utilizar o Raspberry Pi sem uma caixa, desde que não seja colocado numa superfície metálica que possa conduzir eletricidade e causar um curto-circuito. Uma caixa opcional, porém, pode fornecer proteção adicional; o Kit completo Raspberry Pi inclui a Caixa Raspberry Pi Oficial, e existem caixas de terceiros disponíveis em todos os revendedores.

Se quiser usar o Raspberry Pi numa rede com fios, em vez de uma rede sem fios (Wi-Fi), também precisará de um cabo de rede. Deve ser ligado numa das pontas ao switch ou router da sua rede. Se está a planear usar a rede sem fios integrado do Raspberry Pi, não precisará de um cabo; no entanto, precisa de saber o nome e a chave ou frase de acesso da sua rede sem fios.

Configuração do Raspberry Pi 400

As seguintes instruções são para a configuração do Raspberry Pi 4 ou de outro membro da família Raspberry Pi. As instruções para configurar o Raspberry Pi 400 podem ser encontradas na página 32.

Configurar o hardware

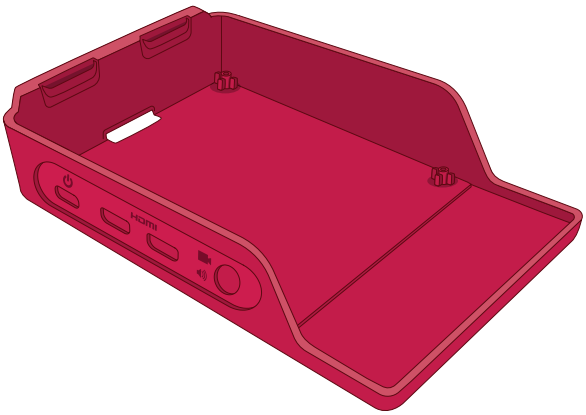
Comece por retirar o seu Raspberry Pi da respetiva caixa. O Raspberry Pi é uma peça robusta de hardware, mas isso não significa que seja indestrutível: tente adquirir o hábito de segurar a placa pelas pontas, em vez de pelas laterais planas, e tenha um cuidado extra ao redor dos pinos metálicos levantados. Se estes pinos estiverem dobrados, na melhor das hipóteses, vai dificultar o uso de placas adicionais e de outro hardware extra e, na pior das hipóteses, pode causar um curto-circuito que vai danificar o seu Raspberry Pi.

Se ainda não o fez, consulte o **Capítulo 1, Conhecer o seu Raspberry Pi** para obter detalhes sobre onde estão exatamente as várias portas e para que servem.

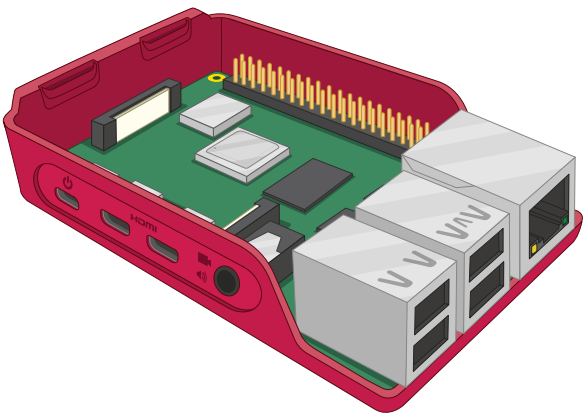
Montar a caixa

Se estiver a instalar o Raspberry Pi numa caixa, este deve ser o seu primeiro passo. Se estiver a usar a Caixa Raspberry Pi Oficial, comece por dividi-la nas respetivas duas peças individuais: a base vermelha e a tampa branca.

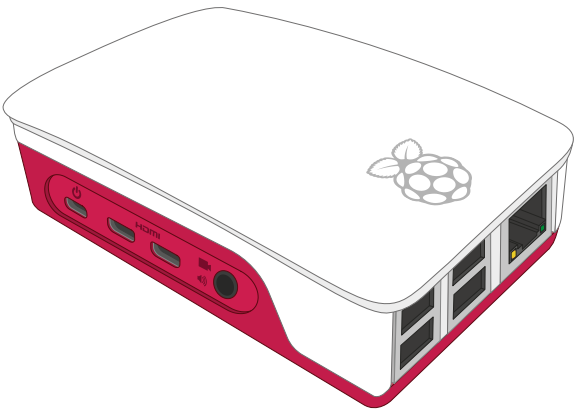
1 **Pegue na base e segure-a para que a extremidade levantada esteja à sua esquerda e a extremidade baixa à sua direita.**



2 **Ao segurar no seu Raspberry Pi (sem cartão microSD introduzido) pelas respetivas portas USB e Ethernet, num ângulo ligeiramente inclinado, encaixe os respetivos conetores (USB Tipo-C, 2 ×micro-HDMI, e 3,5 mm) nos respetivos orifícios na parte lateral da base, depois baixe suavemente para o outro lado para ficar plana.**

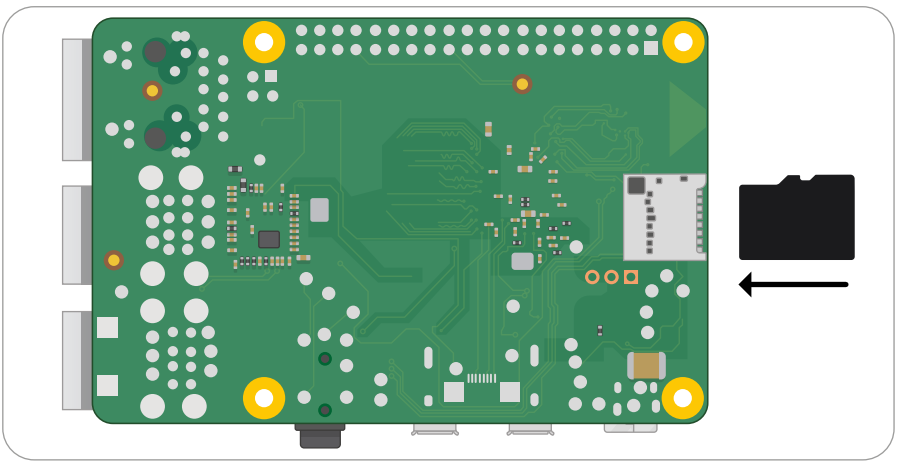


3 **Segure na tampa branca e coloque os dois clipes à esquerda nos orifícios correspondentes à esquerda da base, por cima da ranhura do cartão microSD. Quando estiverem colocados, empurre a parte lateral direita (acima da porta USB) para baixo até ouvir um clique.**

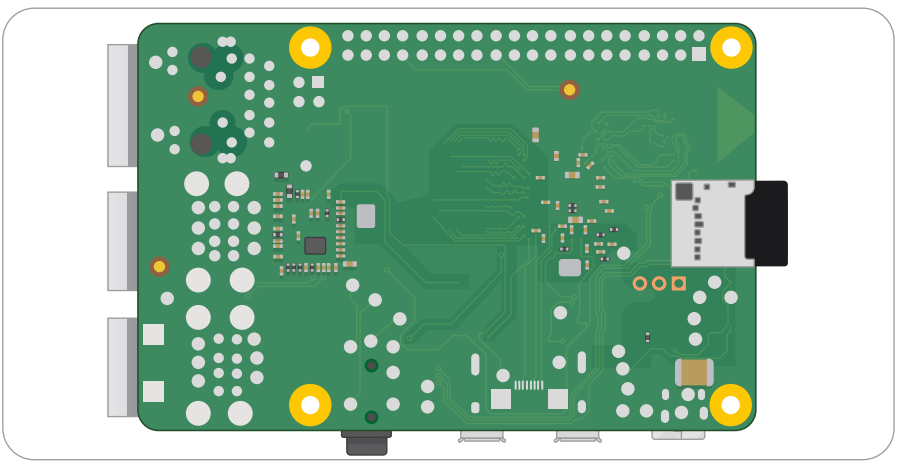


Ligar o cartão microSD

Para instalar o cartão microSD, que é o *armazenamento* do Raspberry Pi, vire o Raspberry Pi (na respetiva caixa se estiver a utilizar uma) e deslize o cartão para a ranhura do cartão microSD com a etiqueta afastada do Raspberry Pi. Só pode ser introduzido de uma forma, e deve deslizar para a posição inicial sem ser necessária muita pressão.



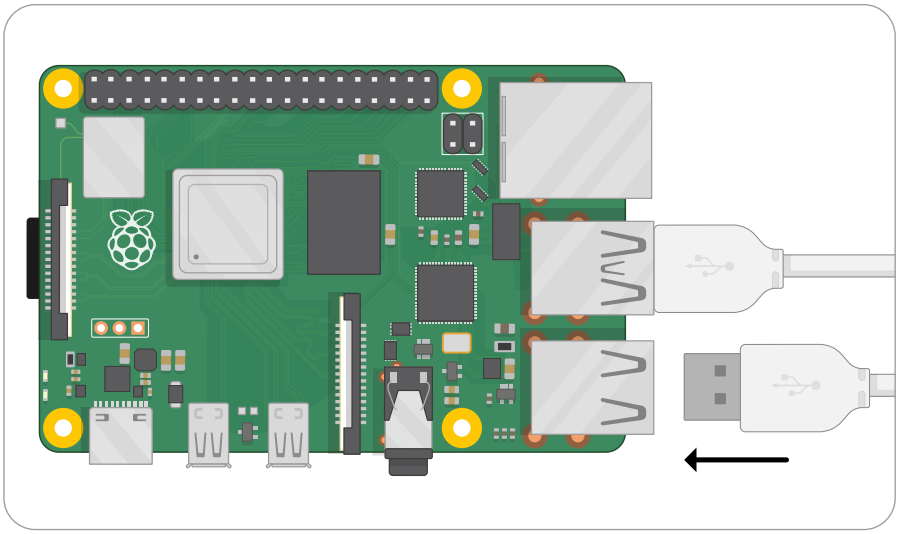
O cartão microSD deslizará para dentro do conetor e, em seguida, para (sem ouvir nenhum clique).



Se quiser removê-lo mais tarde novamente, basta segurar a extremidade do cartão e puxá-lo suavemente para fora. Se estiver a utilizar um modelo antigo do Raspberry Pi, poderá precisar de forçar levemente o cartão para poder desbloqueá-lo; tal não será necessário num Raspberry Pi 3 ou 4.

Ligar um teclado e um rato

Ligue o cabo USB do teclado a qualquer uma das quatro portas USB (2.0 ou 3.0) no Raspberry Pi. Se estiver a usar o Teclado Raspberry Pi Oficial, há uma porta USB na parte de trás para o rato; se não, basta ligar o cabo USB do seu rato noutra porta USB do Raspberry Pi.



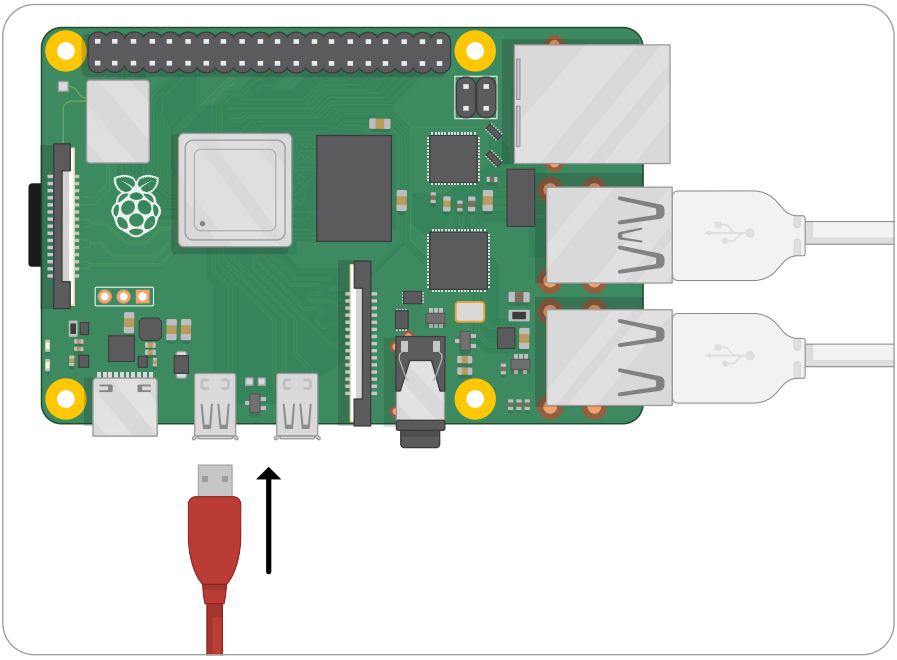
Os conetores USB para o teclado e para o rato devem deslizar para a posição inicial sem ser necessária muita pressão; se tiver de forçar o conetor para o introduzir, significa que ocorreu um problema. Verifique se o conetor USB está na posição correta!

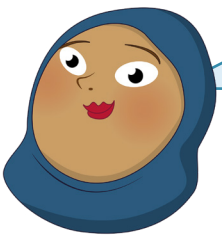
TECLADO E RATO

O teclado e o rato atuam como a sua forma principal de comunicar ao Raspberry Pi o que fazer; em computação, estes são conhecidos como *dispositivos de entrada* em contraste com o dispositivo de visualização que é um *dispositivo de saída*.

Ligar a um dispositivo de visualização

Segure no cabo micro-HDMI e ligue a extremidade mais pequena à porta micro-HDMI mais próxima da porta USB Tipo C no seu Raspberry Pi, e a outra extremidade ao seu dispositivo de visualização. Se o seu ecrã tiver mais do que uma porta HDMI, verifique se existe um número de porta ao lado do próprio conetor; terá de mudar a TV para esta entrada para ver o ecrã do Raspberry Pi. Se não conseguir ver um número de porta, não se preocupe: basta alternar entre cada uma das entradas até encontrar o Raspberry Pi.



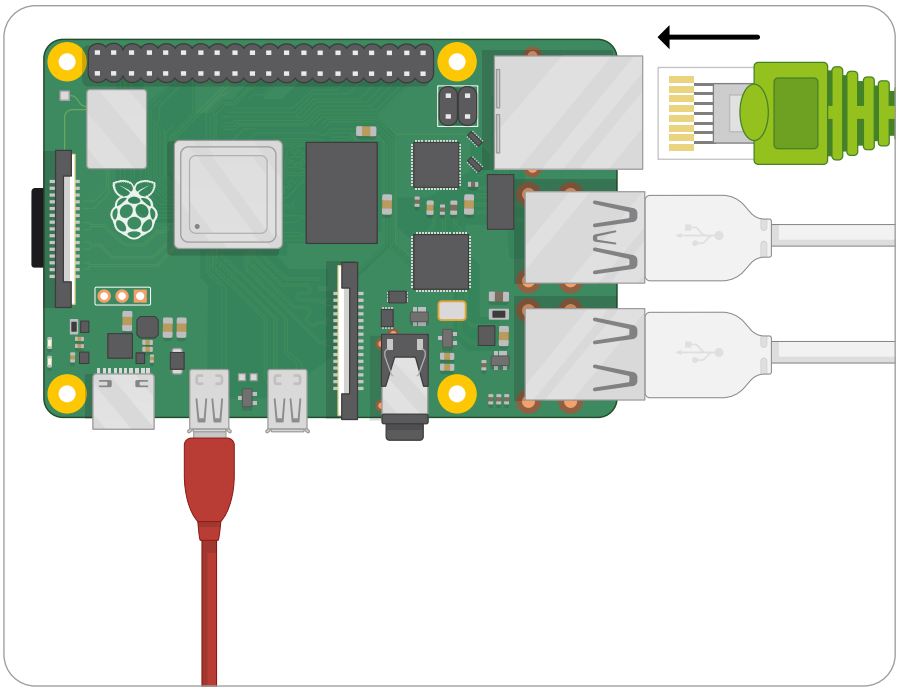


LIGAÇÃO TV

Se a sua TV ou monitor não tiver um conetor HDMI, não significa que não possa usar o Raspberry Pi. Os cabos adaptadores, disponíveis em qualquer revendedor de peças eletrónicas, permitem a conversão da porta micro-HDMI do Raspberry Pi para DVI-D, DisplayPort ou VGA para utilizar com monitores de computador antigos; estes são ligados simplesmente à porta micro-HDMI do Raspberry Pi e, em seguida, usa um cabo adequado para ligar o cabo adaptador ao monitor. Se a sua TV tiver apenas uma entrada de vídeo composto ou SCART, pode adquirir cabos adaptadores para auscultadores (TRRS) de 3,5 mm e adaptadores de vídeo composto para SCART que se ligam ao conetor AV de 3,5 mm.

Ligar um cabo de rede (opcional)

Para ligar o seu Raspberry Pi a uma rede com fios, pegue num cabo de rede – conhecido como cabo Ethernet – e introduza-o na porta Ethernet do Raspberry Pi, com o clipe plástico voltado para baixo, até ouvir um clique. Se precisar de remover o cabo, basta premir o clipe de plástico para dentro na direção da tomada e deslizar suavemente o cabo para o libertar.

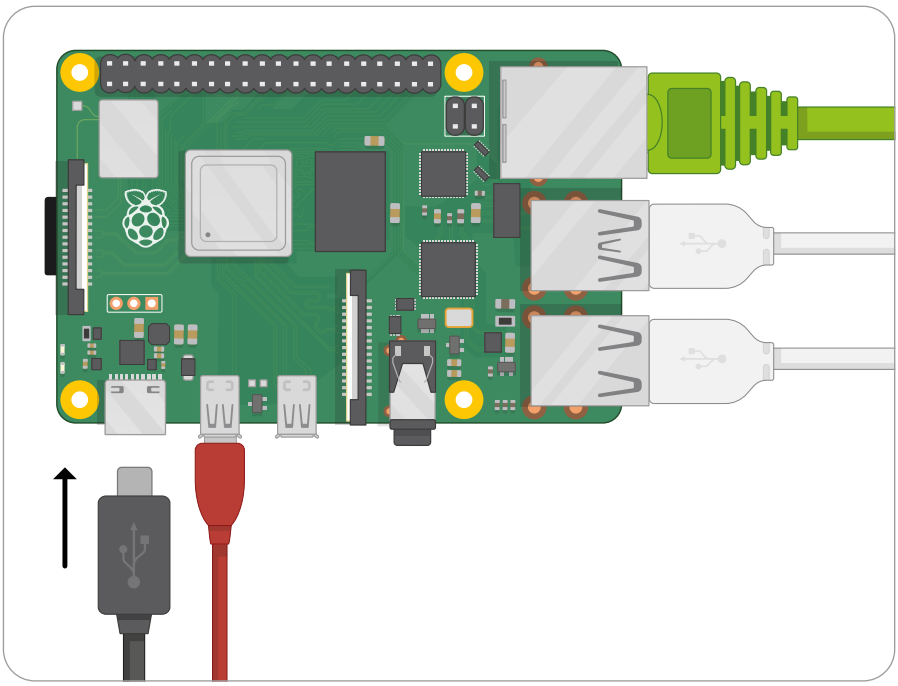


A outra extremidade do cabo de rede deve ser ligada a qualquer porta livre no seu hub de rede, switch ou router do mesmo modo.

Ligar uma fonte de alimentação

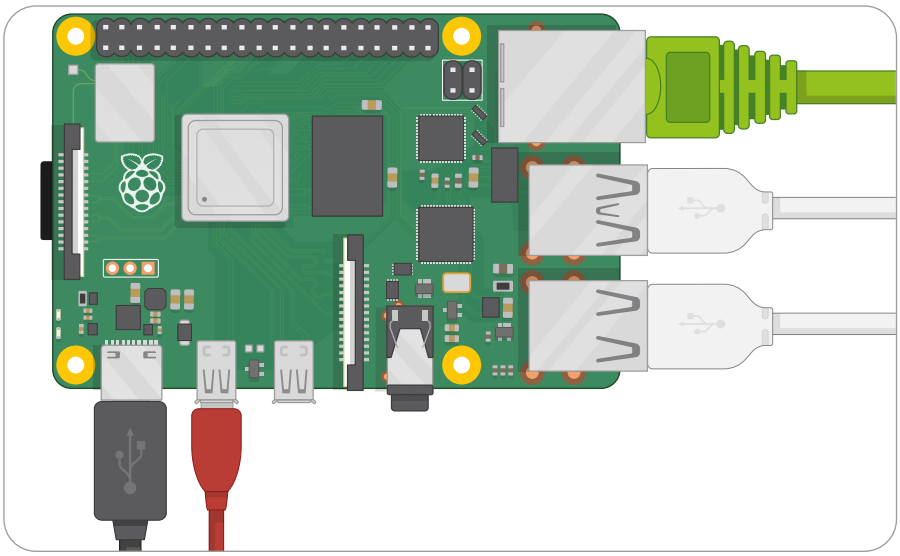
O último passo no processo de configuração do hardware é ligar o Raspberry Pi a uma fonte de alimentação, e só deve ser feito quando estiver pronto para configurar o respetivo software: O Raspberry Pi não tem um botão de alimentação e liga-se assim que estiver ligado a uma fonte de alimentação sob tensão.

Primeiro, ligue a extremidade do cabo de alimentação USB Tipo C ao conetor de alimentação USB Tipo C no Raspberry Pi. Pode ser introduzido nos dois sentidos e deve deslizar suavemente para a posição correta. Se a sua fonte de alimentação tem um cabo removível, certifique-se de que a outra ponta está ligada ao corpo da fonte de alimentação.

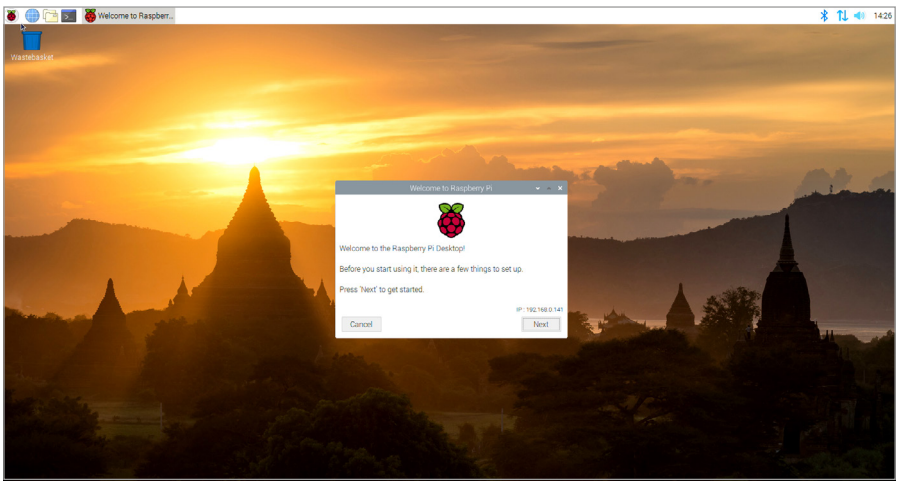


Por fim, ligue a fonte de alimentação a uma tomada e ligue-a; o seu Raspberry Pi começará imediatamente a funcionar.

Parabéns: montou o seu Raspberry Pi!



Verá de forma breve quatro logótipos Raspberry Pi na parte superior esquerda de um ecrã preto, e poderá ver um ecrã azul aparecer à medida que o software é redimensionado automaticamente para utilizar em pleno o seu cartão microSD. Se vir um ecrã preto, espere alguns minutos; na primeira vez que o Raspberry Pi arranca, este tem de fazer alguma manutenção interna em segundo plano. Após algum tempo, verá o ambiente de trabalho do Raspberry Pi OS e o assistente de configuração, como na **Figura 2-1**. O seu sistema operativo está agora pronto para ser configurado, o que aprenderá a fazer em **Capítulo 3, Usar o seu Raspberry Pi**.



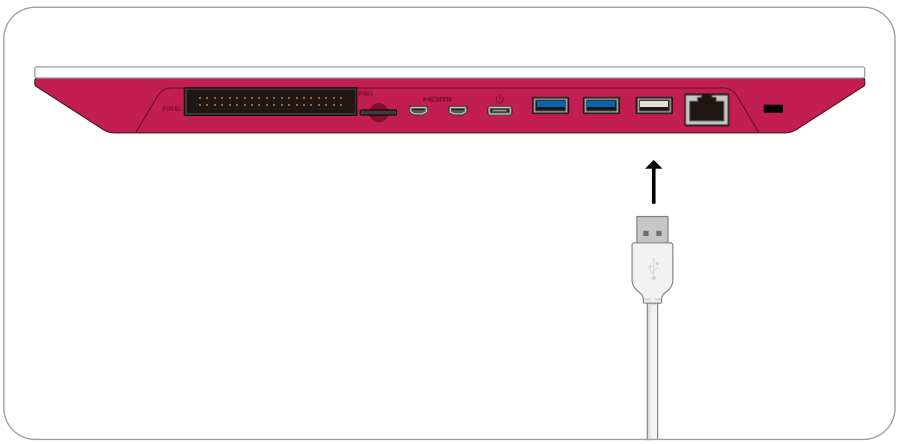
▲ **Figura 2-1:** O ambiente de trabalho do Raspberry Pi OS e o assistente de configuração

Configurar o Raspberry Pi 400

Ao contrário do Raspberry Pi 4, o Raspberry Pi 400 vem com um teclado incorporado e o cartão microSD já instalado. Ainda precisa de ligar alguns cabos para começar, mas isso deve apenas demorar alguns minutos.

Ligar um rato

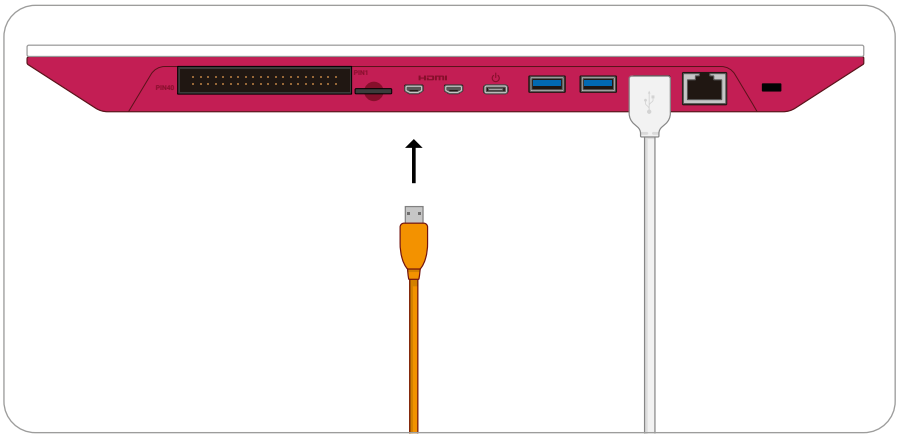
O teclado do Raspberry Pi 400 já está ligado, falta apenas ligar o rato. Segure no cabo USB na extremidade do rato e introduza-o numa das três portas USB (2.0 ou 3.0) na parte traseira do Raspberry Pi 400. Se quiser guardar as duas portas USB 3.0 de alta velocidade para outros acessórios, use a porta branca.



O conetor USB deve entrar sem exercer muita pressão; se tiver de forçar o conetor para o introduzir, significa que ocorreu um problema. Verifique se o conetor USB está na posição correta!

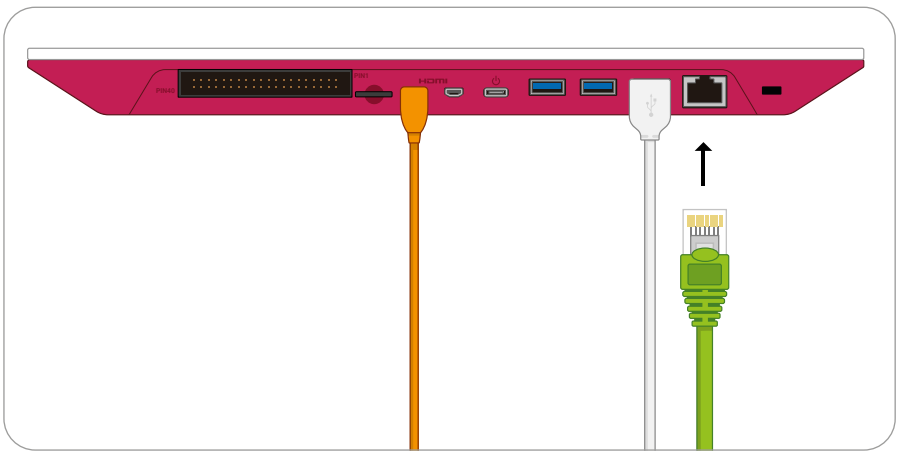
Ligar a um dispositivo de visualização

Segure no cabo micro-HDMI e ligue a extremidade mais pequena à porta micro-HDMI mais próxima da porta USB Tipo C no seu Raspberry Pi 400, e a outra extremidade ao seu dispositivo de visualização. Se o seu ecrã tiver mais do que uma porta HDMI, verifique se existe um número de porta ao lado do próprio conetor; terá de mudar a TV para esta entrada para ver o ecrã do Raspberry Pi. Se não conseguir ver um número de porta, não se preocupe: basta alternar entre cada uma das entradas até encontrar o Raspberry Pi.



Ligar um cabo de rede (opcional)

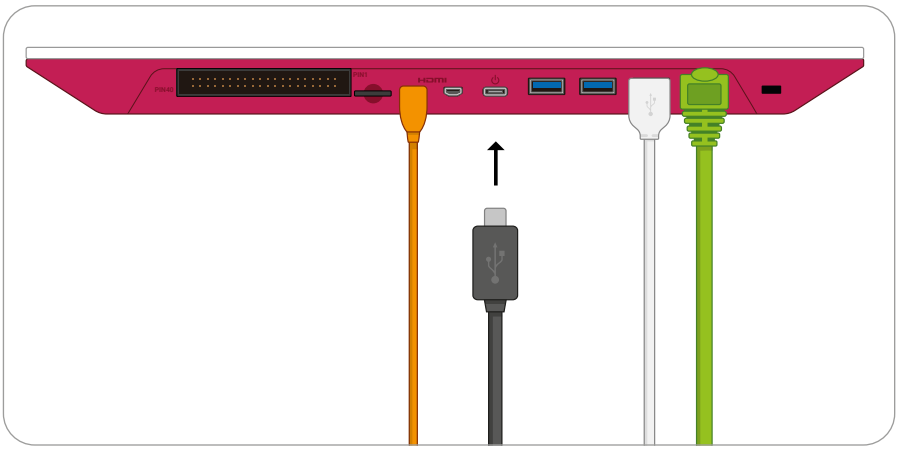
Para ligar o seu Raspberry Pi 400 a uma rede com fios, utilize um cabo de rede, conhecido como cabo Ethernet, e introduza-o na porta Ethernet do Raspberry Pi 400, com o clipe plástico voltado para cima, até ouvir um clique. Se precisar de remover o cabo, basta premir o clipe de plástico para dentro na direção da tomada e deslizar suavemente o cabo para o libertar.



A outra extremidade do cabo de rede deve ser ligada a qualquer porta livre no seu hub de rede, switch ou router do mesmo modo.

Ligar uma fonte de alimentação

O último passo no processo de configuração do hardware é ligar o Raspberry Pi 400 a uma fonte de alimentação, e só deve ser feito quando estiver pronto para configurar o respetivo software: O Raspberry Pi 400 não tem um botão de alimentação e liga-se assim que estiver ligado a uma fonte de alimentação sob tensão. Primeiro, ligue a extremidade do cabo de alimentação USB Tipo C ao conetor de alimentação USB Tipo C no Raspberry Pi. Pode ser introduzido nos dois sentidos e deve deslizar suavemente para a posição correta. Se a sua fonte de alimentação tem um cabo removível, certifique-se de que a outra ponta está ligada ao corpo da fonte de alimentação.



Por fim, ligue a fonte de alimentação a uma tomada e ligue a tomada; o seu Raspberry Pi 400 começará imediatamente a funcionar. Parabéns: montou o seu Raspberry Pi 400!

Verá de forma breve quatro logótipos Raspberry Pi na parte superior esquerda de um ecrã preto, e poderá ver um ecrã azul aparecer à medida que o software é redimensionado automaticamente para utilizar em pleno o seu cartão microSD. Se vir um ecrã preto, espere alguns minutos; na primeira vez que o Raspberry Pi arranca, este tem de fazer alguma manutenção interna em segundo plano. Após algum tempo, verá o ambiente de trabalho do Raspberry Pi OS e o assistente de configuração, como na **Figura 2-2**. O seu sistema operativo está agora pronto para ser configurado, o que aprenderá a fazer em **Capítulo 3, Usar o seu Raspberry Pi**.

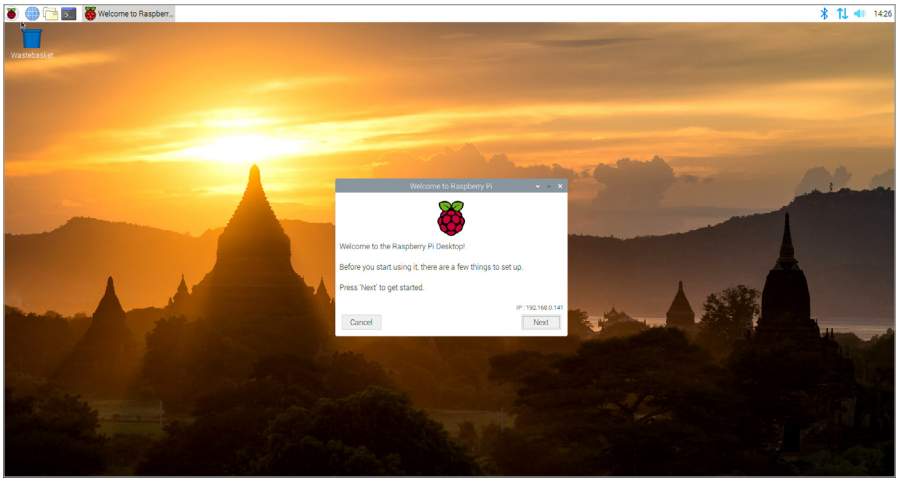
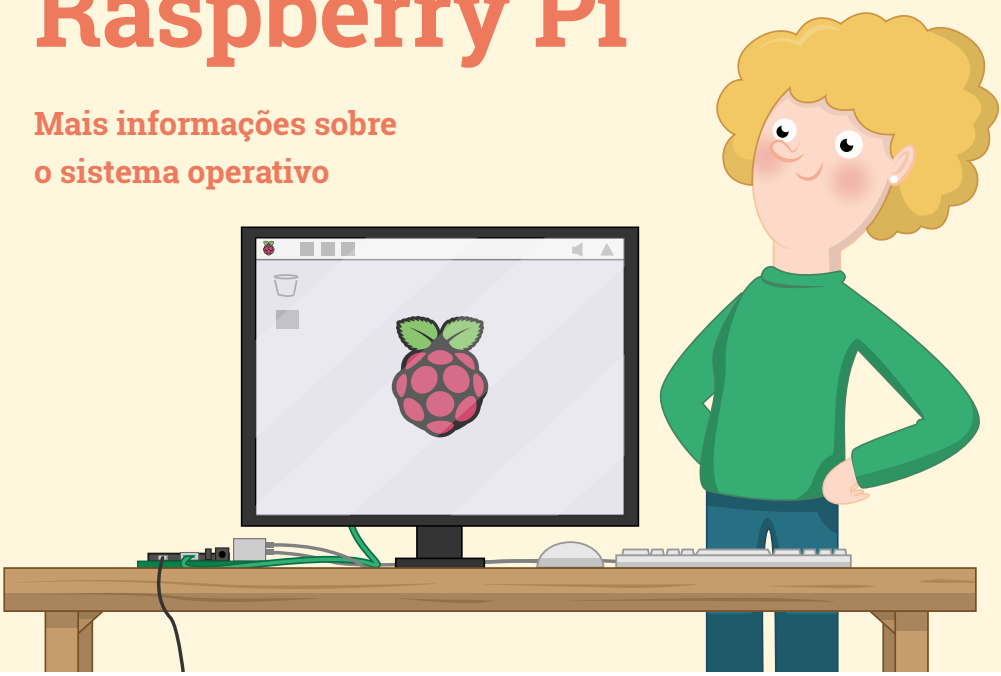


Figura 2-2: O ambiente de trabalho do Raspberry Pi OS e o assistente de configuração

Capítulo 3

Usar o seu Raspberry Pi

Mais informações sobre o sistema operativo



O Raspberry Pi é capaz de executar uma vasta gama de software, incluindo uma série de sistemas operativos diferentes – o software principal que faz um computador funcionar. O mais popular destes, e o sistema operativo oficial da Fundação Raspberry Pi, é o Raspberry Pi OS. Baseado no Debian Linux, está concebido especificamente para o Raspberry Pi e inclui uma gama de extras pré-instalados e prontos a usar.

Se apenas usou o Microsoft Windows ou o Apple macOS, não se preocupe: O Raspberry Pi OS é baseado nos mesmos princípios de janelas, ícones, menus e ponteiro (WIMP), e deve sentir-se familiarizado rapidamente. O capítulo seguinte irá ajudá-lo a começar e apresentar-lhe algum do software integrado.

Assistente de boas-vindas

A primeira vez que executar o sistema operativo Raspberry Pi OS, verá o Assistente de boas-vindas (**Figura 3-1**). Esta ferramenta útil irá guiá-lo através da alteração de algumas configurações no Raspberry Pi OS, conhecido como a *configuração*, para adaptar ao modo e onde vai usar o Raspberry Pi.

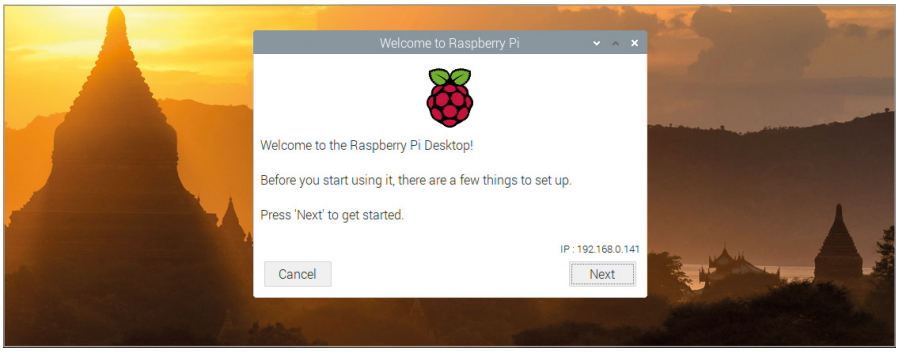
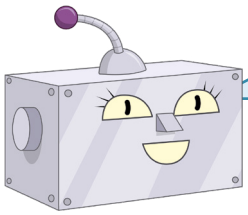


Figura 3-1: Assistente de boas-vindas



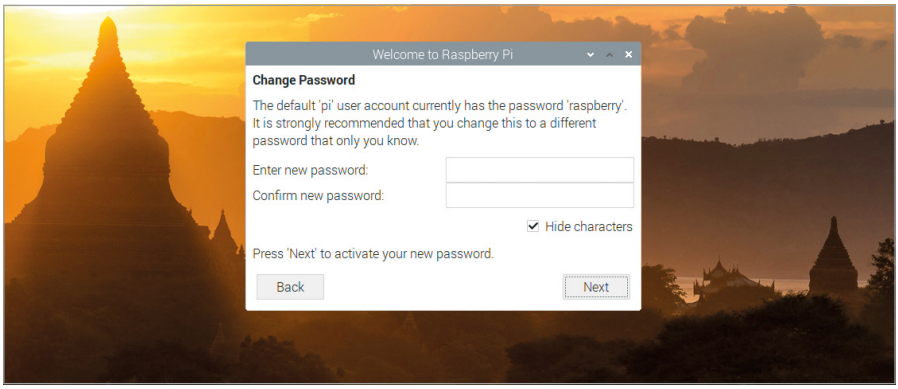
FECHAR O ASSISTENTE

Pode optar por fechar o Assistente de Boas-vindas; para isso, clique no botão Cancelar, mas algumas funcionalidades do Raspberry Pi, como a rede sem fios, não funcionarão até que responda a pelo menos ao primeiro conjunto de perguntas.

Clique no botão Next e, em seguida, escolha o seu país, idioma e fuso horário; para isso, clique em cada uma das caixas pendentes e selecione a sua resposta da lista (**Figura 3-2**). Se estiver a usar um teclado com o esquema dos EUA, clique na caixa de verificação para garantir que o Raspberry Pi OS usa o esquema de teclado correto. Se quiser que o ambiente de trabalho e os programas apareçam em inglês, independentemente do idioma nativo do seu país, marque a caixa de verificação "Use English language". Quando terminar, clique em Next.

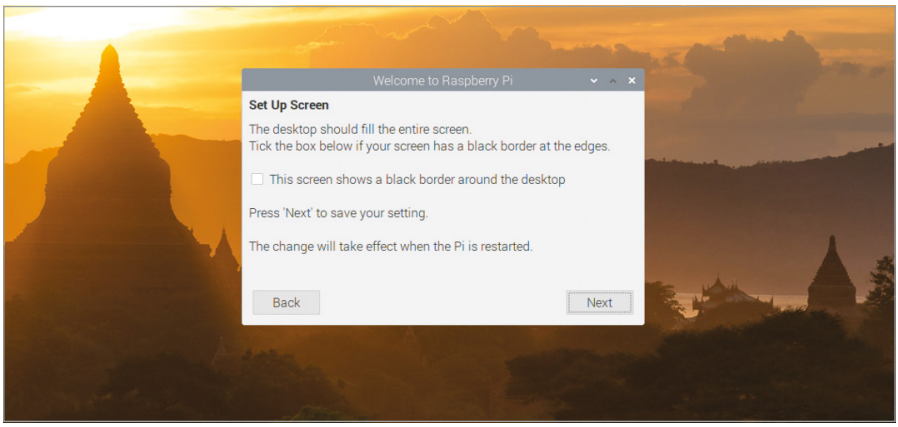
Figura 3-2: Selecionar um idioma, entre outras opções

O próximo ecrã solicita que altere a palavra-passe do utilizador "pi" (a palavra-passe predefinida é "raspberry"). Por razões de segurança, convém criar uma nova palavra-passe. Introduza-a nas caixas (**Figura 3-3**). Quando estiver satisfeito, clique em Next.



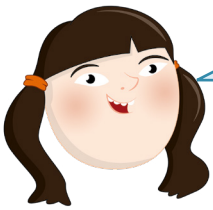
▲ **Figura 3-3:** Definir uma nova palavra-passe

O ecrã seguinte pergunta se existe uma margem preta em volta do ecrã (**Figura 3-4**). Se o ambiente de trabalho do Raspberry Pi preencher a sua TV ou monitor por completo, deixe a caixa desmarcada; se tiver uma margem preta à volta e for mais pequeno do que a sua TV ou monitor, marque a caixa. Quando estiver pronto, clique em Next.



▲ **Figura 3-4:** A verificar se não há margem preta

O ecrã seguinte permitirá que escolha a sua rede Wi-Fi numa lista (**Figura 3-5**). Percorra a lista de redes com o rato ou com o teclado, encontre o nome da sua rede, clique sobre a mesma e, em seguida, clique em Next. Assumindo que a sua rede sem fios é segura (deve mesmo ser), será solicitada a sua palavra-passe (também conhecida como chave pré-partilhada); está normalmente escrita num cartão fornecido com o router ou na parte inferior



REDE SEM FIOS

A rede sem fios integrada só está disponível nas famílias dos dispositivos Raspberry Pi 3, Pi 4, e Pi Zero W. Se quiser usar outro modelo de Raspberry Pi com uma rede sem fios, precisará de um adaptador Wi-Fi USB.

do próprio router. Clique em Next para ligar à rede. Se não quiser ligar a uma rede sem fios, clique em Skip.

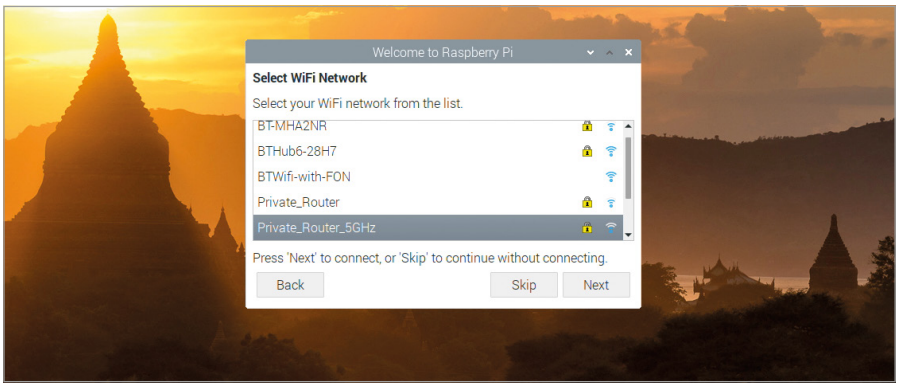


Figura 3-5: Escolher uma rede sem fios

O próxima ecrã permitirá que verifique e instale as atualizações para o Raspberry Pi OS e para o restante software no Raspberry Pi (**Figura 3-6**). O Raspberry Pi OS é atualizado regularmente para corrigir erros, adicionar novas funcionalidades e melhorar o desempenho. Para instalar estas atualizações, clique em Next; caso contrário, clique em Skip. Transferir as atualizações pode demorar vários minutos, seja paciente. Quando as atualizações forem instaladas, aparecerá uma janela que mostra "System is up to date"; clique no botão OK.

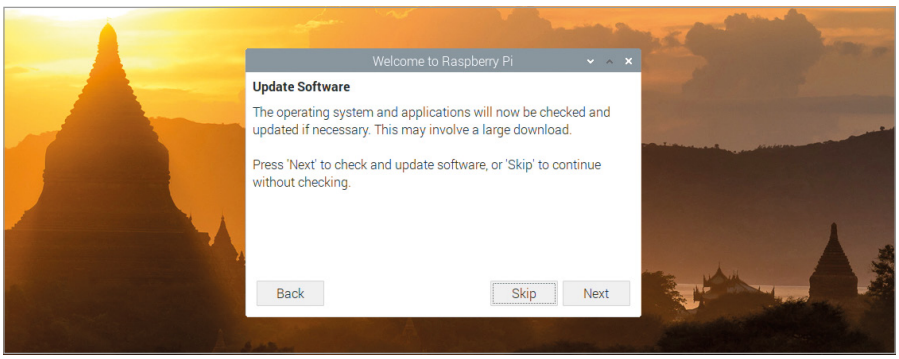


Figura 3-6: A verificar atualizações

O ecrã final do Assistente de boas-vindas (**Figura 3-7**) tem um objetivo simples: algumas alterações efetuadas apenas serão aplicadas quando reiniciar o seu Raspberry Pi (um processo que também é conhecido como reiniciar). Se solicitado, clique no botão Restart e o Raspberry Pi irá reiniciar. Desta vez o Assistente de boas-vindas não aparecerá; este já fez o que lhe competia e o seu Raspberry Pi está pronto a usar.

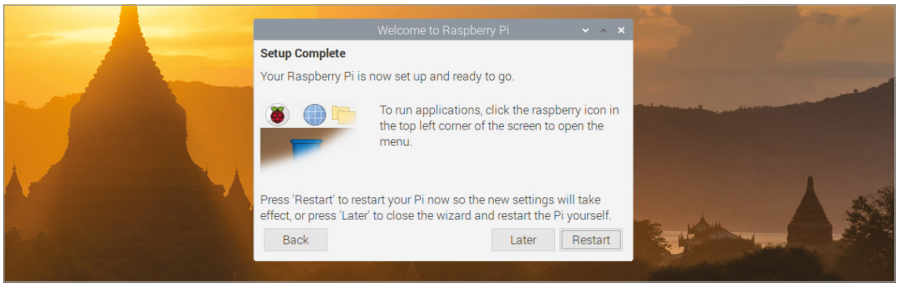


Figura 3-7: Reiniciar o Raspberry Pi

Navegar pelo ambiente de trabalho

A versão do Raspberry Pi OS instalada na maioria das placas Raspberry Pi é conhecida como "Raspberry Pi OS com ambiente de trabalho", referindo-se à principal interface de utilizador gráfica (**Figura 3-8**). A maior parte do ambiente de trabalho é ocupado por uma fotografia, conhecida como papel de parede (**A** em **Figura 3-8**), sobre os quais aparecerão os programas que executar. Na parte superior do ambiente de trabalho encontra uma barra de tarefas (**B**), que lhe permite carregar efetivamente cada um dos programas; estes são então indicados por tarefas (**C**) na barra de tarefas.

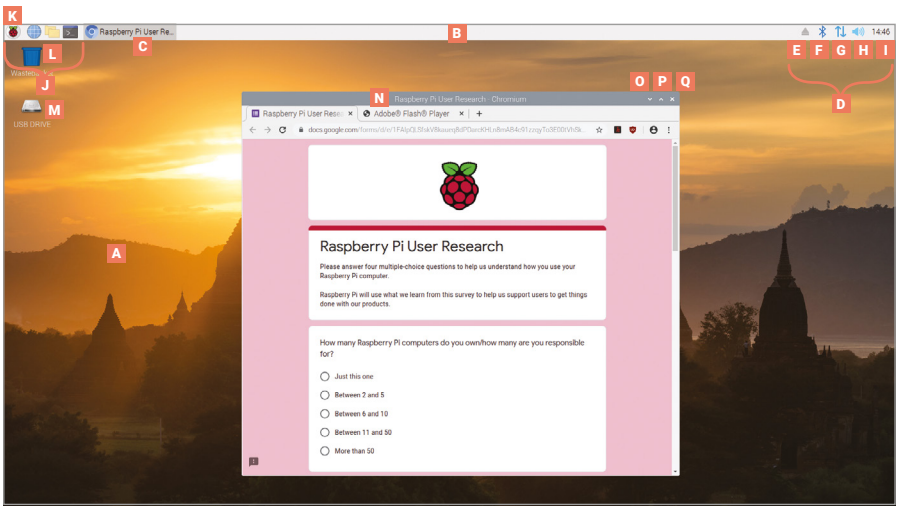


Figura 3-8: O ambiente de trabalho do Raspberry Pi OS

- **A** Papel de parede
- **B** Barra de tarefas
- **C** Tarefa
- **D** Tabuleiro do sistema
- **E** Ejeção de multimédia
- **F** Ícone de Bluetooth
- **G** Ícone de Rede
- **H** Ícone de Volume
- **I** Relógio
- **J** Iniciador
- **K** Ícone do Menu (ou Framboesa)
- **L** Ícone do Lixo
- **M** Ícone da Unidade amovível
- **N** Barra de título da janela
- **O** Minimizar
- **P** Maximizar
- **Q** Fechar

O lado direito da barra de menus aloja o *tabuleiro do sistema*(**D**). Se tiver algum *armazenamento amovível*, como, por exemplo, memórias USB, ligadas ao Raspberry Pi, verá um símbolo de ejeção (**E**); ao clicar no mesmo, poderá ejetá-los e removê-los com segurança. Na extremidade direita está o relógio (**I**); clique sobre o mesmo para aceder ao calendário digital (**Figura 3-9**).

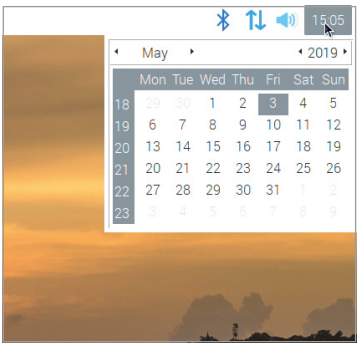


Figura 3-9: O calendário digital

Ao lado encontra um ícone de altifalante (**H**); clique com o botão esquerdo do rato para ajustar o volume de áudio do Raspberry Pi ou clique com o botão direito do rato para escolher qual a saída que o Raspberry Pi deve usar. Ao lado do altifalante há um ícone de rede (**G**); se estiver ligado a uma rede sem fios verá a intensidade do sinal apresentada como uma série de barras, enquanto que se estiver ligado a uma rede com fios verá apenas duas setas. Ao clicar no ícone da rede, abre uma lista de redes sem fios próximas (**Figura 3-10**), e ao clicar no ícone do Bluetooth (**F**) ao lado, permite ligar a um dispositivo Bluetooth próximo.

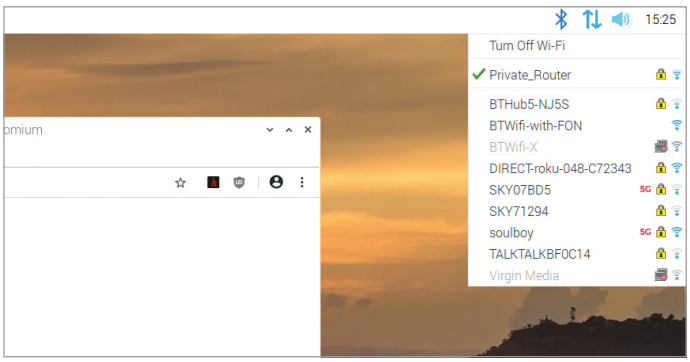
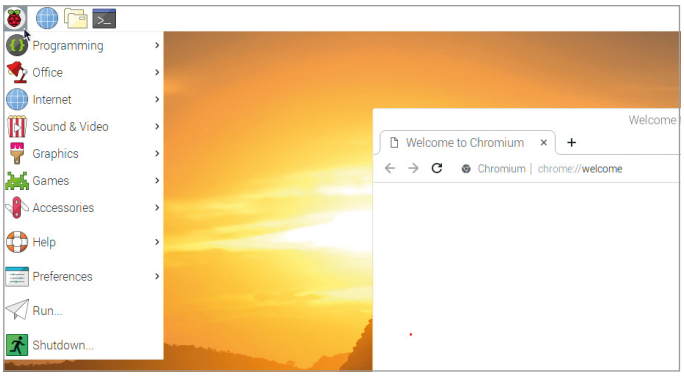


Figura 3-10: Lista de redes sem fios próximas

O lado esquerdo da barra de menus é onde se encontra o *iniciador*(**J**), que é onde encontrará os programas instalados com o Raspberry Pi OS. Alguns destes são visíveis como ícones de atalho; outros estão ocultos no menu, ao qual pode aceder clicando no ícone da framboesa (**K**) na extremidade esquerda (**Figura 3-11**, no verso).

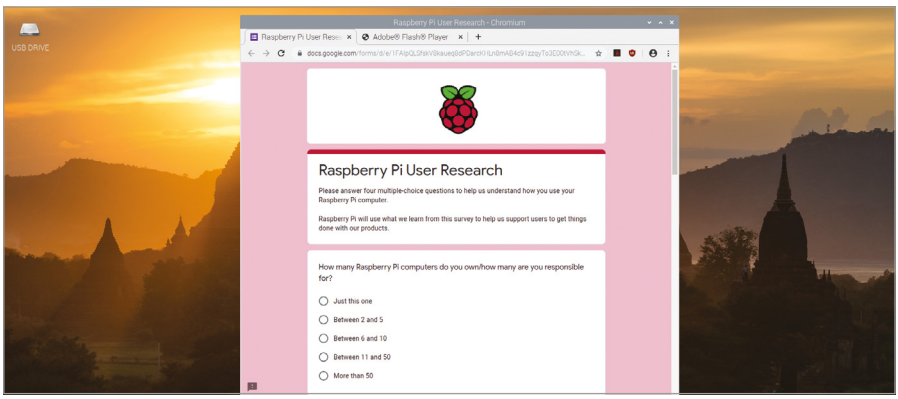


▲ **Figura 3-11:** O menu do Raspberry Pi OS

Os programas no menu estão separados por categorias, os nomes indicam o conteúdo: a categoria Programar, por exemplo, contém software concebido para o ajudar a escrever os seus próprios programas – como explicado a partir do **Capítulo 4, Desenvolvimento com Scratch** – enquanto a categoria Jogos ajuda-o a passar o tempo. Nem todos os programas serão detalhados neste guia; sinta-se à vontade para fazer experiências e aprender mais.

O navegador Web Chromium

Para praticar com o seu Raspberry Pi, comece por carregar o navegador Chromium: clique no ícone da framboesa no canto superior esquerdo para aceder ao menu, mova o ponteiro do rato para selecionar a categoria de Internet e clique no navegador Web Chromium para carregá-lo (**Figura 3-12**).



▲ **Figura 3-12:** O navegador Web Chromium

Se usou o navegador Chrome da Google noutro computador, reconhecerá o Chromium imediatamente. Como navegador Web, o Chromium permite que visite sites, reproduza vídeos, jogos e até comunicar com pessoas de todo o mundo em fóruns e sites de chat.

Comece a usar o Chromium maximizando a respetiva janela de modo a ocupar uma maior área do ecrã: encontre os três ícones no canto superior direito da barra de título da janela do Chromium (**N**) e clique no meio, no ícone da seta para cima (**P**). Este é o botão *maximizar*, que abre a janela em ecrã inteiro. À esquerda de maximizar está o botão *minimizar* (**O**), que oculta a janela até que clique nela na barra de tarefas na parte superior do ecrã. À direita da maximização, está a cruz *fechar*(**Q**), e faz exatamente o que esperaria: fecha a janela.

FECHE E GUARDE

É uma má ideia fechar uma janela antes de guardar o seu trabalho; alguns programas avisam-no para guardar quando clicar no botão fechar, outros não o fazem.

Clique na barra de endereço na parte superior da janela do Chromium – a grande barra branca com uma lupa no lado esquerdo – e introduza **www.raspberrypi.org**e, em seguida, prima o botão **ENTER** no seu teclado. O site Raspberry Pi irá carregar (**Figura 3-13**). Também pode pesquisar na barra de endereço: tente pesquisar por "Raspberry Pi", "Raspberry Pi OS", ou "Computação Educacional".

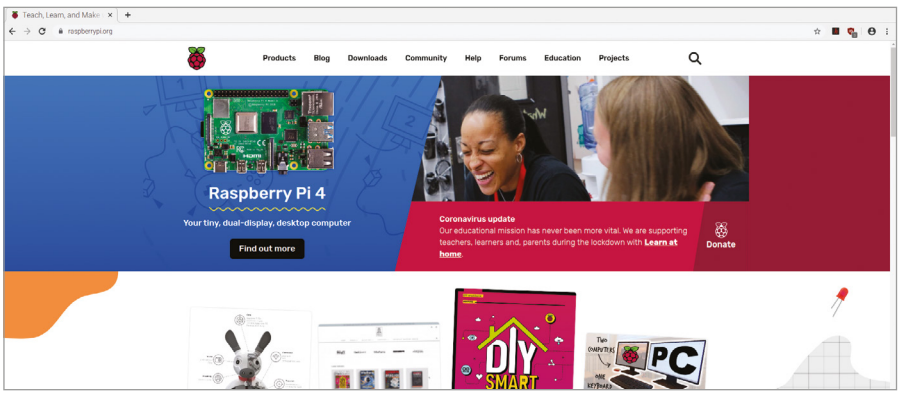


Figura 3-13: Carregar o site do Raspberry Pi no Chromium

A primeira vez que carregar o Chromium, este poderá abrir vários *separadores* na parte superior da janela. Para mudar para um separador diferente, clique no mesmo; para fechar um separador sem fechar o próprio Chromium, clique na cruz na margem direita do separador que pretende fechar. Para abrir um novo separador, que é uma maneira prática de ter vários sites abertos sem ter de estar a alternar entre várias janelas do Chromium, clique no botão do separador à direita do última separador da lista, ou mantenha pressionada a tecla **CTRL** no teclado e pressione a tecla **T** com **CTRL** premido.

Quando terminar de usar o Chromium, clique no botão Fechar no canto superior direito da janela.

O Gestor de ficheiros

Os ficheiros que guarda – por exemplo, programas, vídeos, imagens – vão todos para o seu *diretório pessoal*. Para ver o diretório pessoal, clique no ícone da framboesa novamente para aceder ao menu, mova o ponteiro do rato para selecionar Acessórios e, em seguida, clique em Gestor de Ficheiros para o carregar (**Figura 3-14**).

O Gestor de ficheiros permite que navegue pelos ficheiros e pastas, também conhecidos como *diretórios*, no cartão microSD do Raspberry Pi, assim como em qualquer dispositivo de armazenamento amovível, como unidades flash USB, que ligar às portas USB do Raspberry Pi. Quando o abre pela primeira vez, acede automaticamente ao seu diretório inicial. Aqui encontrará uma série de outras pastas, conhecidas como *subdiretórios*que, tal como o menu, estão organizadas em categorias. Os subdiretórios principais são:

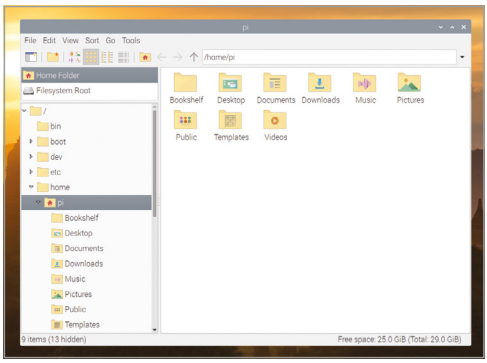


Figura 3-14: O Gestor de ficheiros

- **Bookshelf:** Contém cópias digitais de livros, revistas e outras publicações da Raspberry Pi Press, incluindo uma cópia deste *Guia para iniciantes*. Pode ler estes, e transferir ainda mais, com a aplicação Bookshelf; encontre-a na secção Ajuda do menu.
- **Desktop:** Esta pasta é o que vê quando carrega o Raspberry Pi OS pela primeira vez; se guardar aqui um ficheiro, aparecerá no ambiente de trabalho, tornando-o mais fácil de encontrar e carregar.
- **Documents:** É o destino da maioria dos ficheiros que vai criar, desde contos a receitas.
- **Downloads:** Quando transfere um ficheiro da Internet com o navegador Web Chromium, este será guardado automaticamente nas Transferências.
- **Music:** Qualquer música que criar ou colocar no Raspberry Pi pode ser armazenada aqui.
- **Pictures:** Esta pasta é especificamente para fotos, conhecidas pelo termo técnico *ficheiros de imagem*.
- **Public:** Enquanto a maioria dos seus ficheiros são privados, tudo o que colocar em Público estará disponível para outros utilizadores do Raspberry Pi, mesmo que tenham um nome de utilizador e palavra-passe pessoais.

- **Templates:** Esta pasta contém quaisquer modelos, documentos em branco com um esquema ou estrutura básica já implementados, criados ou instalados pelas suas aplicações.
- **Videos:** Uma pasta para os vídeos, e o primeiro lugar que a maioria dos programas de reprodução de vídeo vai pesquisar.

A janela do Gestor de Ficheiros divide-se em dois painéis: o painel esquerdo mostra os diretórios no seu Raspberry Pi, e o painel direito mostra os ficheiros e subdiretórios do diretório selecionado no painel esquerdo. Se ligar um dispositivo de armazenamento amovível na porta USB do Raspberry Pi, aparece uma janela perguntando se gostaria de abri-lo no Gestor de Ficheiros (**Figura 3-15**); clique em OK para poder ver os respetivos ficheiros e diretórios.

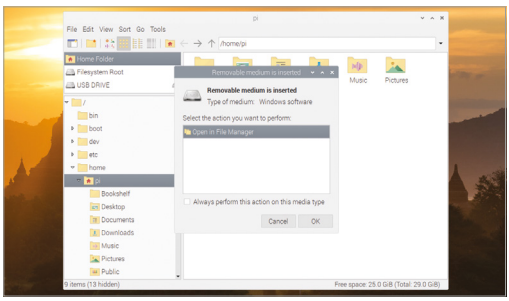
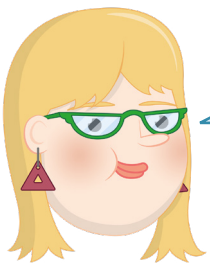


Figura 3-15: Inserir um dispositivo de armazenamento amovível

Os ficheiros podem ser facilmente copiados entre o cartão microSD do Raspberry Pi e um dispositivo amovível: com o seu diretório pessoal e o dispositivo amovível abertos em janelas separadas do Gestor de ficheiros, mova o ponteiro do rato para o ficheiro que pretende copiar, clique e mantenha premido o botão esquerdo do rato, deslize o ponteiro do rato para a outra janela, e solte o botão do rato (**Figura 3-16**, no verso). Esta ação é conhecida como *arrastar e largar*.

Outro método é clicar uma vez no ficheiro, clicar no menu Editar, clicar em Copiar, clicar na outra janela, clicar no menu Editar, e clicar em Colar.

A opção Cortar, também disponível no menu Editar, é semelhante, mas elimina o ficheiro na localização original após ter copiado. Ambas as opções também podem ser utilizadas através dos atalhos do teclado **CTRL+C** (copiar) ou **CTRL+X** (cortar), e colar através de **CTRL+V**.



ATALHOS DE TECLADO

Quando vir um atalho de teclado como **CTRL+C**, significa que deve manter premida a primeira tecla no teclado (**CTRL**), premir a segunda tecla (**C**) e, em seguida, soltar as duas teclas.

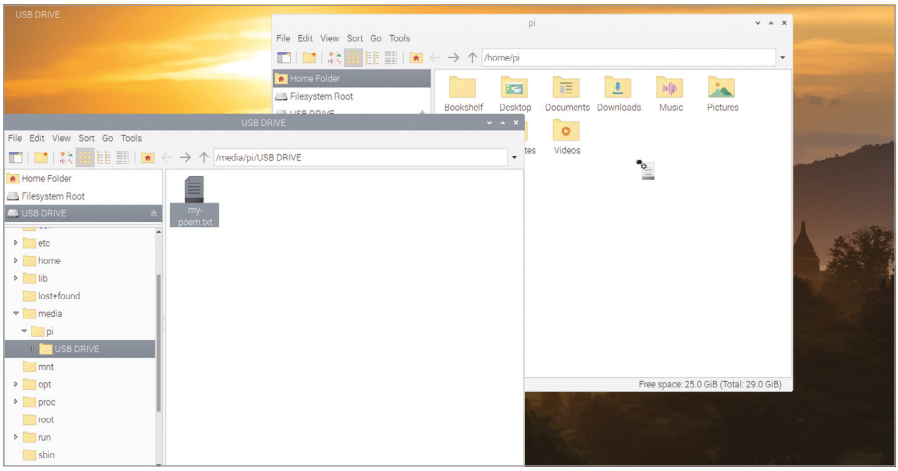
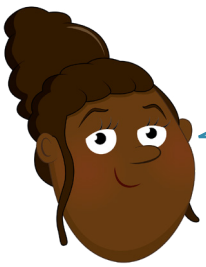


Figura 3-16: Arrastar e largar um ficheiro

Quando terminar as experiências, feche o Gestor de Ficheiros; para isso, clique no botão fechar na parte superior esquerda da janela. Se tiver mais do que uma janela aberta, feche-as todas. Se ligou um dispositivo de armazenamento amovível ao seu Raspberry Pi, ejete-o clicando no botão ejetar na parte superior direita do ecrã, encontrando-o na lista, e clicando no mesmo antes de o desligar.



EJETE OS DISPOSITIVOS

Use sempre o botão ejetar antes de desligar um dispositivo de armazenamento externo; se não o fizer, os ficheiros nele contidos podem ficar corrompidos e inutilizáveis.

O conjunto de aplicações de produtividade do LibreOffice

Para mais uma amostra do que o Raspberry Pi pode fazer, clique no menu do ícone da framboesa, mova o ponteiro do rato para Produtividade e clique em LibreOffice Writer. Esta ação irá carregar a parte do processador de texto do LibreOffice (**Figura 3-17**), um *conjunto de aplicações de produtividade* popular – se já usou o Microsoft Office ou o Google Docs, já usou um conjunto de aplicações de produtividade.

Nota: O LibreOffice pode não ser instalado por predefinição em todas as imagens do Raspberry Pi OS; caso contrário, use a ferramenta Recommended Software (ver página 49) para instalá-lo.

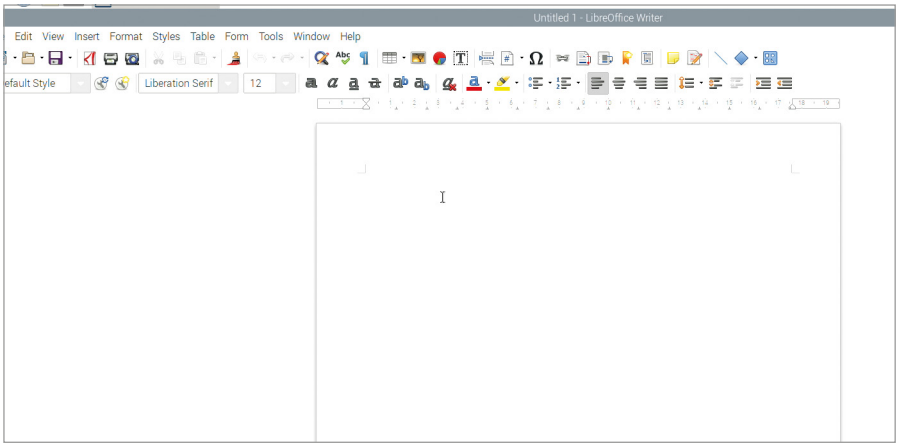


Figura 3-17: O programa Writer do LibreOffice

Um processador de texto permite-lhe não só escrever documentos, mas também formatá-los de forma inteligente: pode alterar o estilo do tipo de letra, cor, tamanho, adicionar efeitos e até mesmo inserir imagens, gráficos, tabelas e outros conteúdos. Um processador de texto também permite verificar se o seu trabalho tem erros, realçando problemas de ortografia e gramática em vermelho e verde, respetivamente, à medida que escreve.

Comece a escrever um parágrafo sobre o que aprendeu sobre o Raspberry Pi e o respetivo software até agora. Faça experiências com os diferentes ícones na parte superior da janela para ver o que eles fazem: veja se pode tornar o seu texto maior, e mudar a respetiva cor. Se não sabe bem como fazê-lo, basta sobrepor o ponteiro do rato sobre cada ícone para ver uma "descrição" que lhe diz o que esse ícone faz. Quando estiver satisfeito, clique no menu File e, em seguida, na opção Save para guardar o seu trabalho (**Figura 3-18**). Dê-lhe um nome e clique no botão Save.

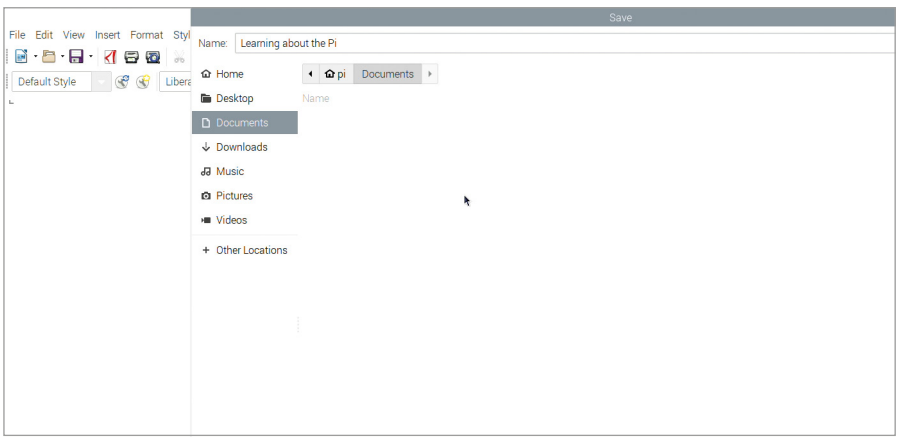


Figura 3-18: Guardar um documento

GUARDE O SEU TRABALHO

Habitue-se a guardar o seu trabalho, mesmo que ainda não o tenha terminado. Evitará muitos problemas se houver um corte de energia e o seu trabalho for interrompido!

O LibreOffice Writer é apenas uma parte do conjunto geral de aplicações de produtividade do LibreOffice. As outras partes, que encontrará na mesma categoria do menu Office do LibreOffice Writer, são:

- **LibreOffice Base:** Uma base de dados; uma ferramenta para armazenar informação, procurando-a rapidamente, e analisá-la.
- **LibreOffice Calc**: Uma folha de cálculo; uma ferramenta para manipulação de números e criação de gráficos.
- **LibreOffice Draw**: Um programa de ilustração; uma ferramenta para a criação de imagens e diagramas.
- **LibreOffice Impress**: Um programa de apresentação, para a criação de slides e executar apresentações.
- **LibreOffice Math**: Um editor de fórmulas; uma ferramenta para criar fórmulas matemáticas devidamente formatadas que podem em seguida ser usadas noutros documentos.

O LibreOffice também está disponível para outros computadores e sistemas operativos. Se gostar de usá-lo no seu Raspberry Pi, pode transferi-lo gratuitamente de **libreoffice.org** e instalá-lo em qualquer computador Microsoft Windows, Apple macOS ou Linux.

Se quiser saber mais sobre a utilização do LibreOffice, clique no menu Help. Caso contrário, feche o LibreOffice Writer clicando no botão fechar no ponto superior direito da janela.

OBTENHA AJUDA

A maioria dos programas inclui um menu de Ajuda que tem tudo, desde informações sobre o que é o programa até guias sobre como usá-lo. Se alguma vez se sentir perdido ou não entender o funcionamento de um programa, procure o menu Ajuda para se orientar.

A ferramenta Recommended Software

Embora o Raspberry Pi OS inclua uma vasta gama de software pré-instalado, é compatível com muitos mais. Pode encontrar uma seleção do melhor software na ferramenta Recommended Software.

Tenha em atenção que a ferramenta Recommended Software necessita de uma ligação à Internet. Se o seu Raspberry Pi estiver ligado, clique no menu do ícone da framboesa, mova o ponteiro do rato para Preferências e clique em Recommended Software. A ferramenta será carregada e depois começará a transferir informação sobre o software disponível.

Após alguns segundos, aparecerá uma lista de pacotes de software compatíveis (**Figura 3-19**). Estes, tal como o software do menu da framboesa, estão organizados em várias categorias. Clique numa categoria no painel à esquerda para ver o software dessa categoria, ou clique em Todos os programas para ver todos.

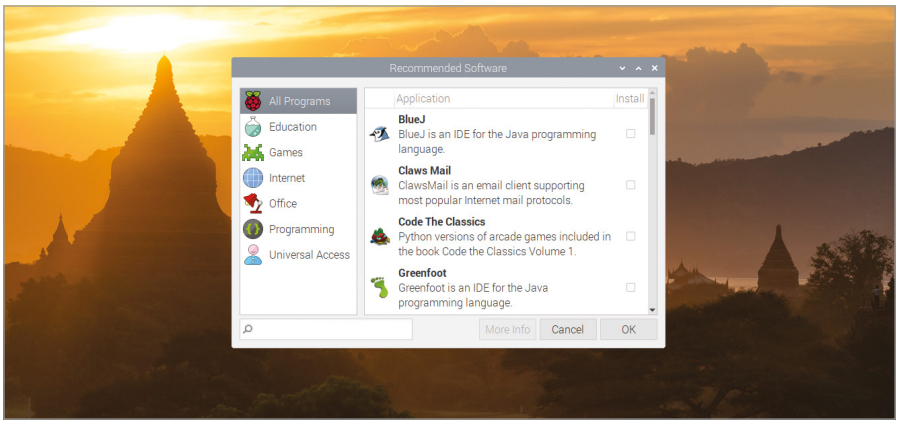


Figura 3-19: A ferramenta Recommended Software

Se um elemento de software tiver um visto de verificação junto ao mesmo, significa que já está instalado no seu Raspberry Pi. Caso contrário, pode clicar na caixa de verificação junto ao mesmo para adicionar um visto de verificação e marcá-lo para instalação. Pode marcar as peças de software que quiser antes de as instalar todas de uma vez, mas se estiver a usar um cartão microSD mais pequeno que o recomendado, poderá não ter espaço para todas elas.

Também pode desinstalar o software da mesma forma: encontre um elemento de software que já tenha um visto de verificação na respetiva caixa de verificação e, em seguida, clique no visto de verificação para removê-lo. Se cometeu um erro ou mudou de ideias, basta clicar novamente para voltar a colocar o visto de verificação.

Quando estiver satisfeito com o software que escolheu, clique no botão OK para iniciar o processo de instalação ou de desinstalação (**Figura 3-20**, no verso). Após transferir e instalar o novo software escolhido, aparecerá uma caixa de diálogo; clique em OK para fechar a ferramenta Recommended Software.

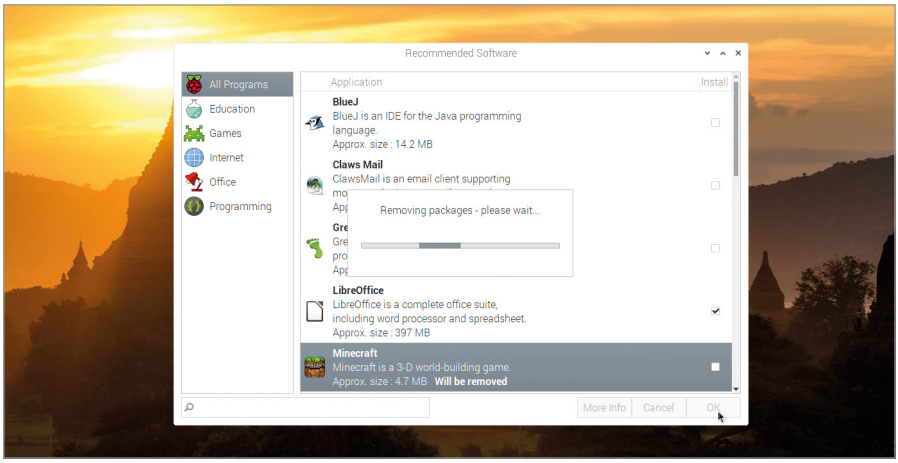


Figura 3-20: Desinstalar software

Pode encontrar uma ferramenta adicional para instalar ou desinstalar software, a ferramenta Add/Remove Software, na mesma categoria de Preferências do menu do Raspberry Pi OS. Esta opção oferece uma seleção mais ampla de software, mas que ainda não foi aprovada pela Fundação Raspberry Pi.

A Ferramenta Raspberry Pi Configuration

O último programa que sobre o qual aprenderá neste capítulo é conhecido como a ferramenta Raspberry Pi Configuration, e é muito parecido com o Assistente de boas-vindas que usou no início: permite alterar várias definições no Raspberry Pi OS. Clique no ícone da framboesa novamente para aceder ao menu, mova o ponteiro do rato para selecionar a categoria das Preferências e, em seguida, clique em Raspberry Pi Configuration para o carregar (**Figura 3-21**).

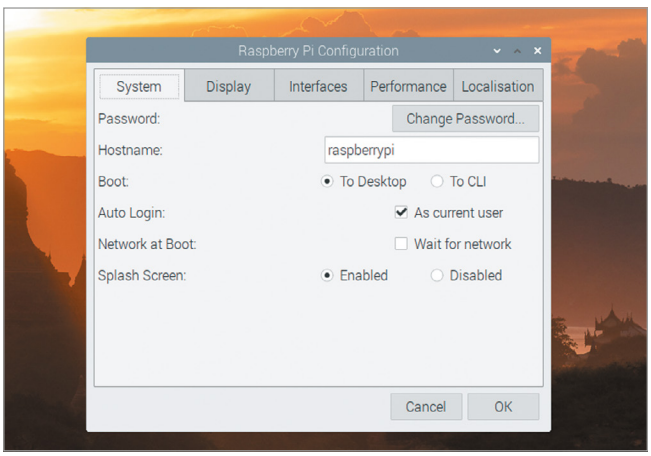


Figura 3-21: A Ferramenta de Configuração do Raspberry Pi

A ferramenta está dividida em cinco separadores. O primeiro é o System: que permite alterar a palavra-passe da sua conta, definir um nome de anfitrião – o nome que o Raspberry Pi usa na sua rede local sem fios ou com fios – e alterar uma série de outras definições. No entanto, não deve ser necessário alterar qualquer um destes. Clique no separador Display para abrir a próxima categoria. Aqui poderá alterar, se necessário, as definições de visualização do ecrã para se adequar à sua TV ou monitor.

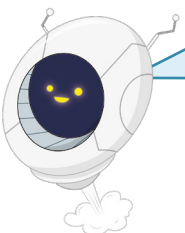
MAIS DETALHES

Esta breve descrição geral é simplesmente para que se habitue à ferramenta. Pode encontrar informações mais detalhadas sobre cada uma das respetivas definições em **Apêndice E, A Ferramenta de Configuração do Raspberry Pi**.

O separador Interface oferece uma série de definições, todas desativadas por predefinição. Estas definições apenas devem ser alteradas se adicionar novo hardware, como o Raspberry Pi Camera Module, e apenas se instruído pelo fabricante do hardware. As exceções a esta regra são: SSH, que ativa uma "Secure Shell" e permite que aceda ao Raspberry Pi através de outro computador na sua rede com um cliente SSH; VNC, que ativa um "Computador de uma Rede Virtual" e permite que veja e controle o ambiente de trabalho do Raspberry Pi OS através de outro computador na sua rede com um cliente VNC; e o GPIO remoto, que permite que use os pinos GPIO do Raspberry Pi, sobre os quais obterá mais informações em **Capítulo 6, Computação física com Scratch e Python** – através de outro computador na sua rede.

Clique no separador Performance para ver a quarta categoria. Aqui pode definir a quantidade de memória utilizada pela unidade de processamento gráfico (GPU) do Raspberry Pi e, para alguns modelos, aumentar o desempenho do Raspberry Pi através de um processo conhecido como *overclocking*. No entanto, tal como anteriormente, é melhor não alterar estas definições, a menos que seja necessário.

Por fim, clique no separador Localisation para ver a última categoria. Aqui pode mudar a sua região, que controla opções como o idioma usado no Raspberry Pi OS e como os números são apresentados, alterar o fuso horário, alterar o esquema do teclado e definir o seu país para o Wi-Fi. Por enquanto, basta clicar em Cancelar para fechar a ferramenta sem efetuar quaisquer alterações.



AVISO!

Países diferentes têm regras diferentes sobre as frequências de rádio Wi-Fi pode usar. Se configurar o país do Wi-Fi na Ferramenta de Configuração do Raspberry Pi para um país diferente de onde realmente está, é provável que torne mais difícil ligar-se às suas redes e pode até ser ilegal de acordo com as leis de licenciamento de rádio, portanto não o faça!

Encerramento

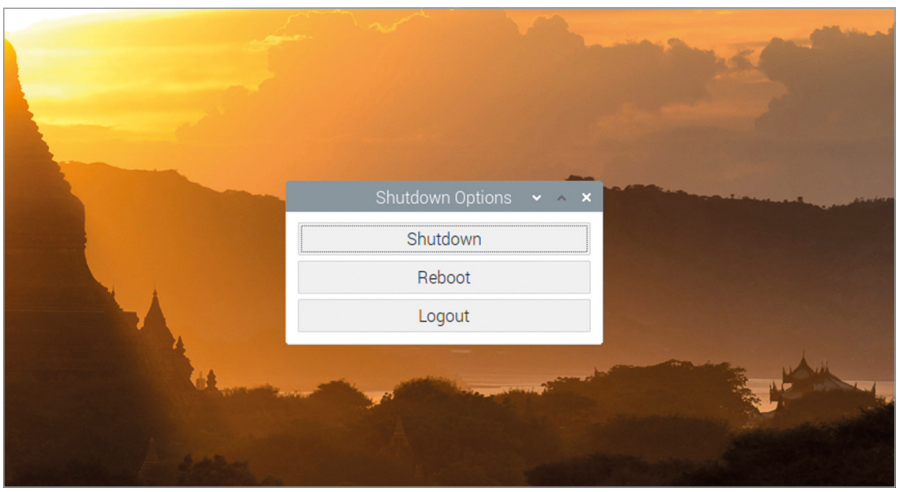
Agora que explorou o ambiente de trabalho do Raspberry Pi OS, está na altura de aprender uma competência muito importante: encerrar o seu Raspberry Pi em segurança. Como qualquer computador, o Raspberry Pi guarda os ficheiros em que está a trabalhar na *memória volátil* – memória que é esvaziada quando o sistema é desligado. Para documentos que está a criar, é suficiente guardar um de cada vez – o que transforma o ficheiro da memória volátil em *memória não volátil*, a do cartão microSD – para garantir que não perde nada.

Mas os documentos em que está a trabalhar não são os únicos ficheiros abertos. O próprio Raspberry Pi OS mantém vários ficheiros abertos enquanto está a funcionar, e desligar o cabo de energia do seu Raspberry Pi enquanto estes ainda estão abertos pode corromper o sistema operativo e ser necessário reinstalá-lo.

Para evitar que isso aconteça, precisa de ter a certeza que o Raspberry Pi OS guarda todos os seus ficheiros e prepará-lo para desligar – um processo conhecido como *encerramento* do sistema operativo.

Clique no ícone da framboesa no canto superior esquerdo do ambiente de trabalho e, em seguida, clique em Shutdown. Aparecerá uma janela com três opções (**Figura 3-22**): Shutdown, Reboot, Logout... Encerrar é a opção que usará mais: ao clicar na mesma fará com que o Raspberry Pi OS feche todo o software e ficheiros abertos e, em seguida, desligue o Raspberry Pi. Assim que o ecrã ficar preto, espere alguns segundos até que a luz verde intermitente no Raspberry Pi se apague; só então será seguro desligar a fonte de alimentação.

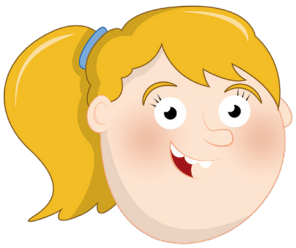
Para voltar a ligar o Raspberry Pi, basta desligar e voltar a ligar o cabo de alimentação ou ligar a tomada de parede.



▲ **Figura 3-22:** Encerrar o Raspberry Pi

A opção Reboot passa por um processo semelhante à opção Shutdown, fecha todas as janelas, mas em vez de desligar o Raspberry Pi, reinicia o Raspberry Pi, quase da mesma forma como se selecionasse a opção Shutdown e, em seguida, desligasse e voltasse a ligar o cabo de alimentação. Terá de usar a opção Reboot se fizer certas alterações que requerem que reinicie o sistema operativo, como instalar algumas atualizações no respetivo software principal, ou se tiver ocorrido um erro no software, conhecido como *Falha* e deixado o Raspberry Pi SO num estado inutilizável.

Por fim, a opção Logout só é realmente útil se tiver mais de uma conta de utilizador no seu Raspberry Pi: fecha quaisquer programas que tem atualmente abertos e direciona-o para o ecrã de início de sessão no qual é solicitado que introduza um nome de utilizador e uma palavra-passe. Se premir Logout por engano e quiser voltar a iniciar sessão, basta escrever "pi" como o nome de utilizador e a palavra-passe que escolheu no Assistente de Boas-vindas no início deste capítulo.



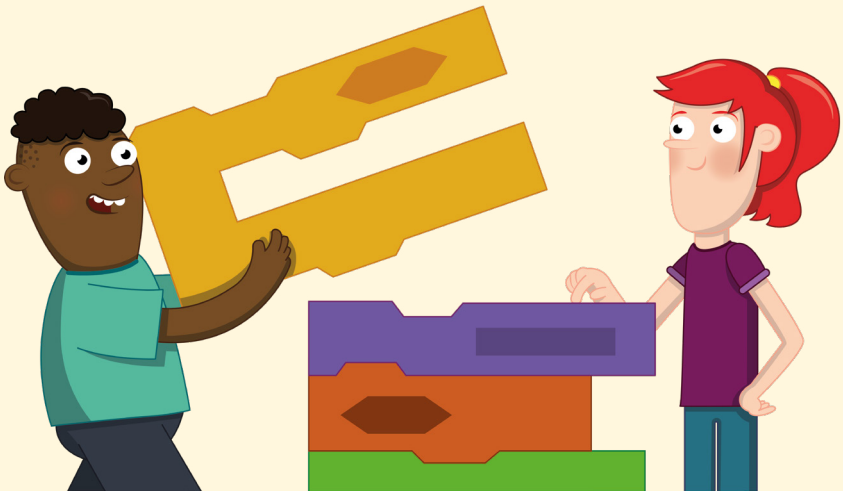
AVISO!

Nunca desligue o cabo de alimentação de um Raspberry Pi sem encerrar primeiro o Raspberry Pi. Se o fizer, pode corromper o sistema operativo e perder os ficheiros que criou ou transferiu.

Capítulo 4

Programar com Scratch 3

Comece a aprender a programar com o Scratch, a linguagem de programação com base em blocos

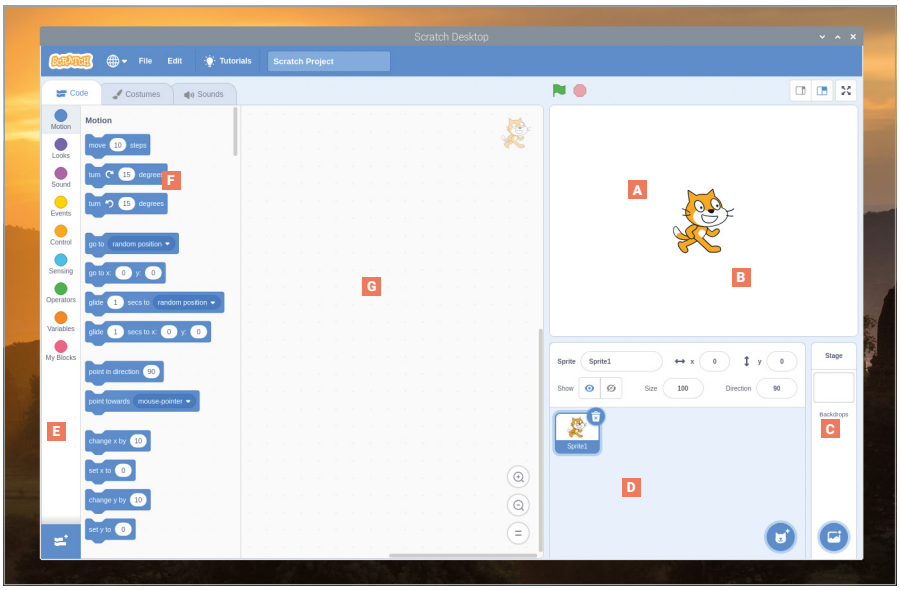


Usar o Raspberry Pi não se trata apenas de utilizar software que outras pessoas criaram; trata-se de criar o seu próprio software, tendo como base tudo o que possa imaginar. Quer tenha experiência anterior com a criação dos seus próprios programas, um processo conhecido como programação ou escrever código, ou não, vai encontrar no Raspberry Pi uma excelente plataforma para criar e experimentar.

O Scratch é essencial para a acessibilidade da programação no Raspberry Pi, uma linguagem de programação visual desenvolvida pelo Massachusetts Institute of Technology (MIT). Enquanto que as linguagens tradicionais de programação necessitam que introduza instruções em suporte de texto para o computador realizar, da mesma forma que escreve uma receita para fazer um bolo, o Scratch faz com que construa o seu programa passo a passo utilizando blocos, os pedaços de código pré-escritos ocultos por trás de peças de puzzle codificadas por cores.

O Scratch é excelente como a primeira linguagem de programação tanto para jovens como para os mais velhos, mas não se deixe enganar pela sua aparência amigável: é um ambiente de programação poderoso e totalmente funcional em que pode criar tudo, desde jogos e animações simples até projetos de robótica interativa complexos.

Apresentamos a interface do Scratch 3



A **Área do palco** – Tal como os atores de uma peça, os seus sprites movem-se pelo palco controlados pelo seu programa.

B **Sprite** – As personagens ou objetos que controlar num programa do Scratch são conhecidos como sprites, e estão no palco.

C **Controlos de palco** – O seu palco pode ser alterado, incluindo adicionar as suas próprias fotos como fundo, com os controlos de palco.

D **Lista de Sprites** – Todos os sprites que criou ou carregou no Scratch aparecerão nesta secção da janela.

E **Palete de Blocos** – Todos os blocos disponíveis para o seu programa aparecem na palete de blocos, que apresenta categorias codificadas por cores.

VERSÕES DO SCRATCH

Quando este livro foi escrito, o Raspberry Pi OS foi lançado com três versões do Scratch: Scratch 1, 2 e 3, todos incluídos na secção de Programação do menu do Raspberry Pi OS. Este capítulo foi escrito para o Scratch 3. Tenha em atenção que o Scratch 3 só irá ser executado no Raspberry Pi 4. Se em vez disso quiser usar o Scratch 2, essa versão não pode ser executada no Raspberry Pi Zero, Modelo A, A+, B, ou B+.

F **Blocos** – Os blocos de código pré-escritos do programa permitem que construa o seu programa passo a passo.

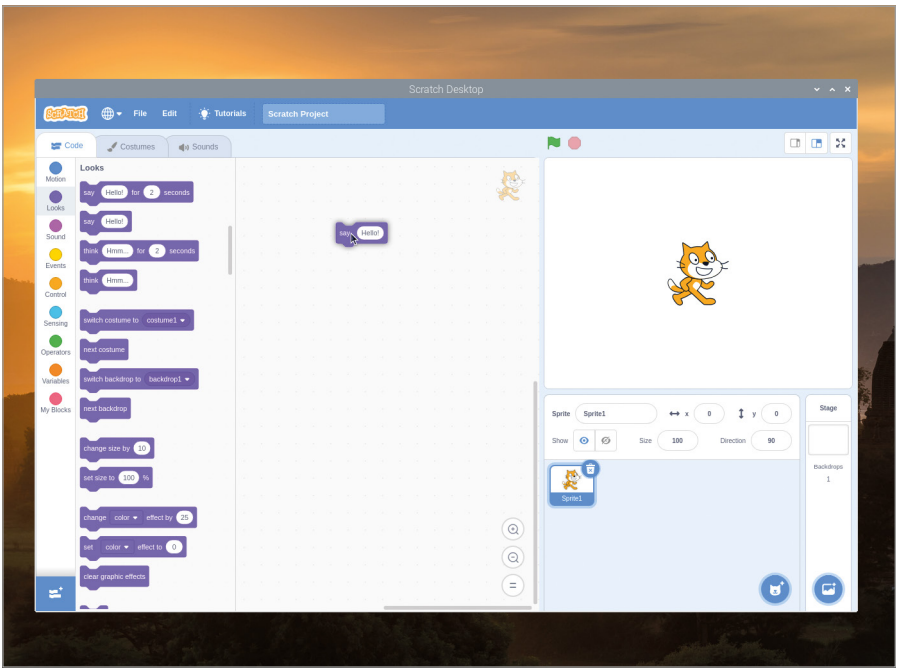
G **Área de Programação** – A área de programação é onde o seu programa é construído ao arrastar e soltar blocosdapalete de blocos para formar scripts.

O seu primeiro programa do Scratch: Olá, Mundo!

O Scratch 3 carrega como qualquer outro programa no Raspberry Pi: clique no ícone da framboesa para carregar o menu do Raspberry Pi OS, mova o cursor para a secção de Programação, e clique no Scratch 3. Após alguns segundos, a interface de utilizador do Scratch 3 será carregada.

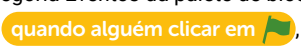
Embora na maioria das linguagens de programação seja necessário indicar ao computador o que fazer através de instruções escritas, com o Scratch não é o caso. Comece por clicar na categoria Aparência na palete de blocos, à esquerda da janela do Scratch. Esta ação dá acesso aos blocos dessa categoria, de cor roxa. Encontre o bloco 

, clique e mantenha premido o botão esquerdo do rato sobre este, e arraste-o para a área de programação no centro da janela do Scratch antes de soltar o botão do rato (**Figura 4-1**).



▲ **Figura 4-1:** Arraste e solte o bloco na área de programação

Veja a forma do bloco que acabou de deixar cair: tem um orifício na parte de cima, e uma peça a condizer a sair na parte de baixo. Como uma peça de puzzle, esta disposição significa que é necessário que o bloco tenha algo acima e algo abaixo dele. Para este programa, o que ficar acima é um *acionador*.

Clique na categoria Eventos da palete de blocos, de cor dourada e, em seguida, clique e arraste o bloco 

, conhecido como um bloco *chapéu* – para a área de programação. Posicione-o de modo a que o pedaço no fundo se ligue ao buraco na

parte superior do seu bloco 

até ver um contorno branco e, em seguida, solte o botão do rato. Não é preciso ser preciso; se estiver suficientemente perto, o bloco encaixa no lugar como uma peça de um puzzle. Caso contrário, clique e mantenha premido novamente no bloco para ajustar a respetiva posição até encaixar.

O seu programa está concluído. Para que funcione, ação também conhecida como *executar* o programa, clique no ícone da bandeira verde na parte superior esquerda da área do palco. Se tudo correu bem, o sprite do gato vai cumprimentá-lo com um alegre "Olá!" (**Figura 4-2**), o seu primeiro programa é um sucesso!

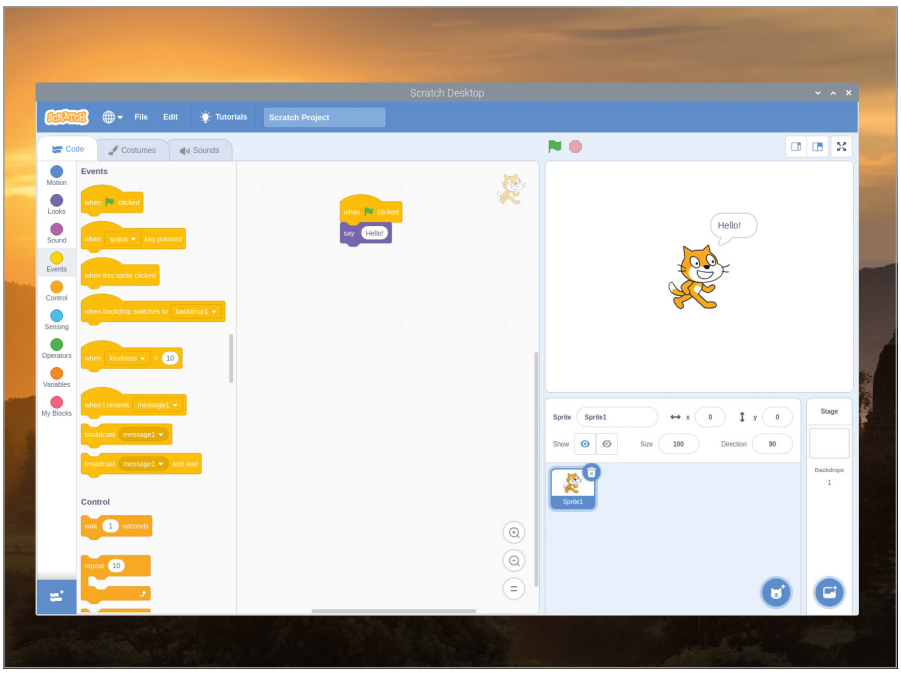


Figura 4-2: Clique na bandeira verde acima do palco e o gato dirá "Olá"

Antes de continuar, nomeie e guarde o seu programa. Clique no menu Arquivo e, em seguida, em "Descarregar para o seu computador". Introduza um nome e clique no botão Guardar (**Figura 4-3**).

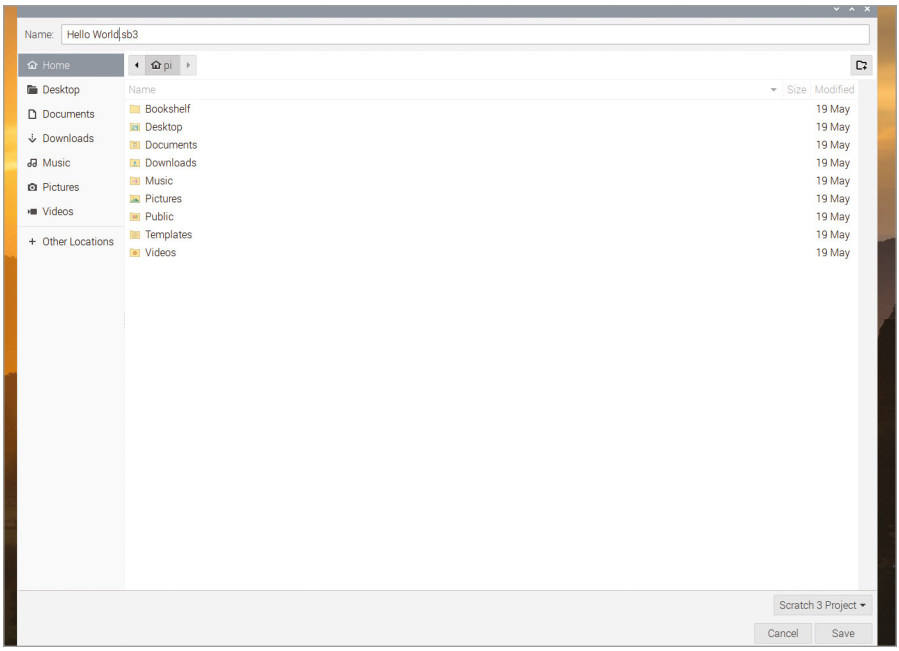
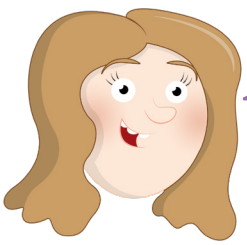


Figura 4-3: Guarde o seu programa com um nome fácil de memorizar



O QUE PODE DIZER?

Alguns blocos no Scratch podem ser alterados. Experimente clicar na palavra "Olá!" e escrever outra coisa, em seguida, clique na bandeira verde novamente. O que acontece no palco?

Próximos passos: sequenciação

Embora o seu programa tenha dois blocos, só tem realmente uma instrução: dizer "Olá!" sempre que se clicar na bandeira e o programa for executado. Para fazer mais, precisa de conhecer a *sequenciação*. Os programas de computador, na sua forma mais simples, são uma lista de instruções, tal como uma receita. Cada instrução segue a instrução precedente numa progressão lógica conhecida como *sequência linear*.

Comece por clicar e arrastar o bloco diz Olá! da área de programação de volta para a palete de blocos (**Figura 4-4**). Esta ação elimina o bloco, retirando-o do seu programa e deixando apenas o bloco acionador, quando alguém clicar em .

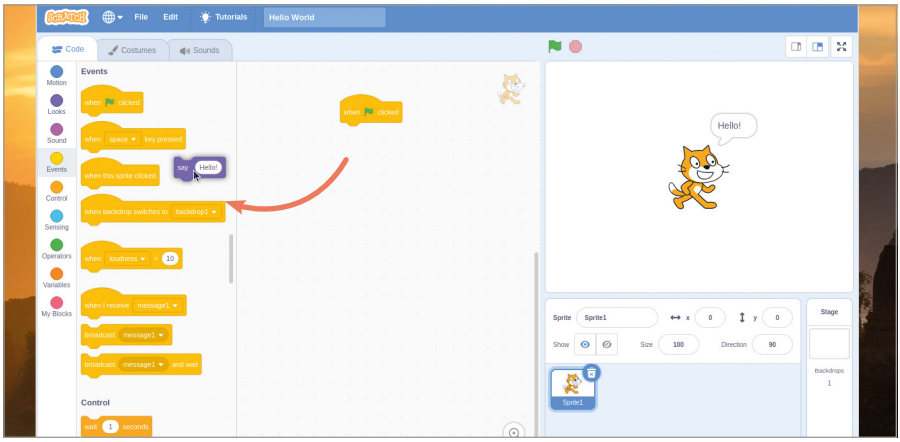
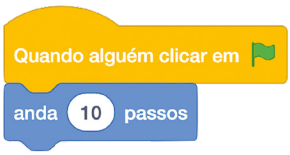
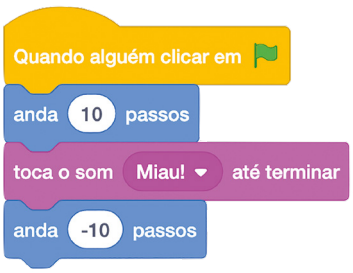


Figura 4-4: Para eliminar um bloco, basta arrastá-lo para fora da área de programação

Clique na categoria Movimento na palete de blocos, em seguida, clique e arraste o bloco **anda 10 passos** para que este se encaixe no bloco acionador na área de programação. Como o nome sugere, esta ação indica ao seu sprite, o gato, para se deslocar uma série de passos na direção para onde está virado.

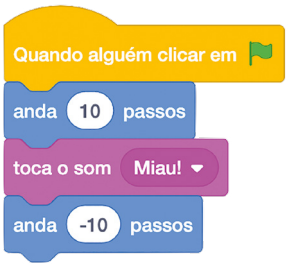


Adicione mais instruções ao seu programa para criar uma sequência. Clique na palete de Som, codificada a cor-de-rosa e, em seguida, clique e arraste o bloco **toca o som Miau até terminar** para que encaixe debaixo do bloco **anda 10 passos**. Continue: clique novamente na categoria Movimento e arraste outro bloco **anda 10 passos** para baixo do seu bloco Som, mas desta vez clique no "10" para selecioná-lo e introduza "-10" para criar um bloco **anda -10 passos**.

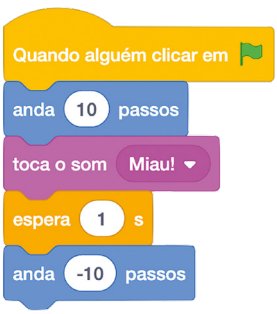


Clique na bandeira verde acima do palco para executar o programa. Verá o gato mover-se para a direita, e miar, certifique-se de que tem altifalantes ou auscultadores ligados para ouvi-lo e, em seguida, volte para o início novamente. Clique na bandeira novamente, e o gato repetirá as respetivas ações.

Parabéns: criou uma sequência de instruções, que o Scratch está a executar uma de cada vez, de cima para baixo. Embora o Scratch só execute uma instrução da sequência de cada vez, fá-lo muito rapidamente: experimente apagar o bloco `toca o som Miau até terminar` ; para isso, clique e arraste o bloco `anda -10 passos` para o destacar, arrastando o bloco `toca o som Miau até terminar` para a palete de blocos, depois substitua-o pelo bloco `toca o som Miau` mais simples antes de arrastar o seu bloco `anda -10 passos` de volta para o fundo do seu programa.



Clique na bandeira verde para executar o seu programa novamente, e o sprite do gato parecer que está parado. O sprite está, de facto, a mover-se, mas este move-se de volta para a posição inicial tão rapidamente que parece que está parado. Esta situação acontece porque ao usar o bloco `toca o som Miau` este não espera que o som termine de tocar antes de realizar o próximo passo; porque o Raspberry Pi "pensa" tão rapidamente, que a próxima instrução é executada antes de poder ver o sprite do gato mover-se. Há outra maneira de corrigir isso, além de usar o bloco `toca o som Miau até terminar`: clique na categoria de Controlo laranja claro da palete de blocos e, em seguida, clique e arraste um bloco `espera 1 s` entre o bloco `toca o som Miau` e o bloco `anda -10 passos` inferior.



Clique na bandeira verde para executar o seu programa uma última vez, e verá que o sprite do gato espera um segundo após mover-se para a direita antes de voltar para a esquerda novamente. Esta ação é conhecida como um *atraso*, e é essencial para controlar o tempo que a sequência de instruções leva para ser executada.

DESAFIO: ADICIONAR MAIS PASSOS

Experimente adicionar passos à sua sequência e altere os valores nos passos atuais. O que acontece quando o número de passos de um bloco de movimento não coincide com o número de passos de outro bloco? O que acontece se tentar tocar um som enquanto outro som ainda estiver a tocar?

Repetir o ciclo

A sequência que criou até agora só é executada uma vez: você clica na bandeira verde, o sprite do gato move-se e mia e, em seguida, o programa para até que clique novamente na bandeira verde. No entanto, não terá de parar, porque no Scratch vem incluído um tipo de bloco de Controlo conhecido como *ciclo*.

Clique na categoria Controlo na palete de blocos e encontre o bloco repete para sempre. Clique e arraste-o para a área de programação, em seguida solte-o debaixo do bloco quando alguém clicar em e acima do primeiro bloco. anda 10 passos

Repare na forma como o bloco em forma de C cresce automaticamente para rodear os outros blocos na sua sequência. Clique na bandeira verde agora, e verá rapidamente o que o bloco repete para sempre faz: em vez do seu programa ser executado uma vez e terminar, irá ser executado vezes sem conta, literalmente para sempre. Em programação, isto é conhecido como um *ciclo infinito*, literalmente, um ciclo que nuncaacaba.

Se o som de miado constante estiver a ficar um pouco exagerado, clique no octógono vermelho junto à bandeira verde acima da área do palco para parar o seu programa. Para mudar o tipo de ciclo, clique e arraste o primeiro bloco **anda 10 passos** e puxe-o juntamente com os blocos debaixo dele para fora do bloco **repete para sempre** e, em seguida, solte-os debaixo do bloco **quando alguém clicar em** . Clique e arraste o bloco **repete para sempre** para a palete de blocos para eliminá-lo e, em seguida, clique e arraste o bloco **repete 10 vezes** debaixo do bloco **quando alguém clicar em** para que contorne os outros blocos.

Clique na bandeira verde para executar o seu novo programa. No início, parece estar a fazer a mesma coisa que na sua versão original: repetir a sua sequência de instruções vezes sem conta. Desta vez, no entanto, em vez de continuar para sempre, o ciclo terminará após dez repetições. Esta ação é conhecida como um *ciclo definido*: você define quando irá terminar. Os ciclos são ferramentas poderosas, e a maioria dos programas – especialmente jogos e programas de deteção, utilizam muitas vezes ciclos infinitos e definidos.

O QUE ACONTECE AGORA?

O que acontece se mudar o número no bloco de ciclo para o tornar maior? O que acontece se for menor? O que acontece se colocar o número 0 no bloco de ciclo?

Variáveis e condicionais

Os conceitos finais que tem de saber antes de começar a programar programas no Scratch a sério estão estritamente relacionados: *variáveis* e *condicionais*. Uma variável é, como o nome sugere, um valor que pode variar, noutras palavras, mudar, ao longo do tempo e sob o controlo do programa. Uma variável tem duas propriedades principais: o respetivo nome, e o valor

que armazena. Esse valor também não tem de ser um número: podem ser números, texto, verdadeiro ou falso, ou estar completamente vazio, conhecido como um *valor nulo*.

As variáveis são ferramentas poderosas. Tenha em mente tudo o que tem de monitorizar num jogo: os pontos de vida de uma personagem, a velocidade do objeto em movimento, o nível que está a ser jogado atualmente e a pontuação. Todas estas são monitorizadas como variáveis.

Primeiro, clique no menu Arquivo e guarde o seu programa atual, clicando em "Descarregar para o seu computador". Se já guardou o programa anteriormente, ser-lhe-á solicitado se que escrever por cima do mesmo, substituindo a antiga cópia guardada pela nova versão atualizada. Em seguida, clique em Arquivo e, em seguida, em Novo para iniciar um novo projeto em branco (clique em OK quando lhe perguntarem se deseja substituir os conteúdos do projeto atual). Clique na categoria laranja escura Variáveis na palete de blocos e, em seguida, no botão "Criar uma variável". Introduza "ciclos" como o nome da variável (**Figura 4-5**) e, em seguida, clique em OK para fazer aparecer uma série de blocos napalete de blocos.

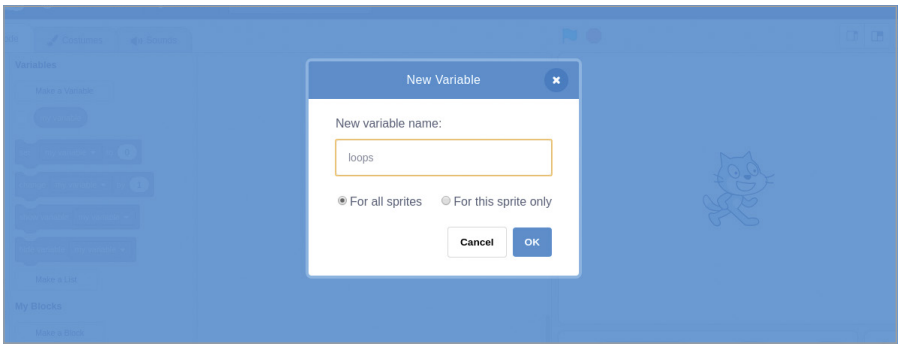


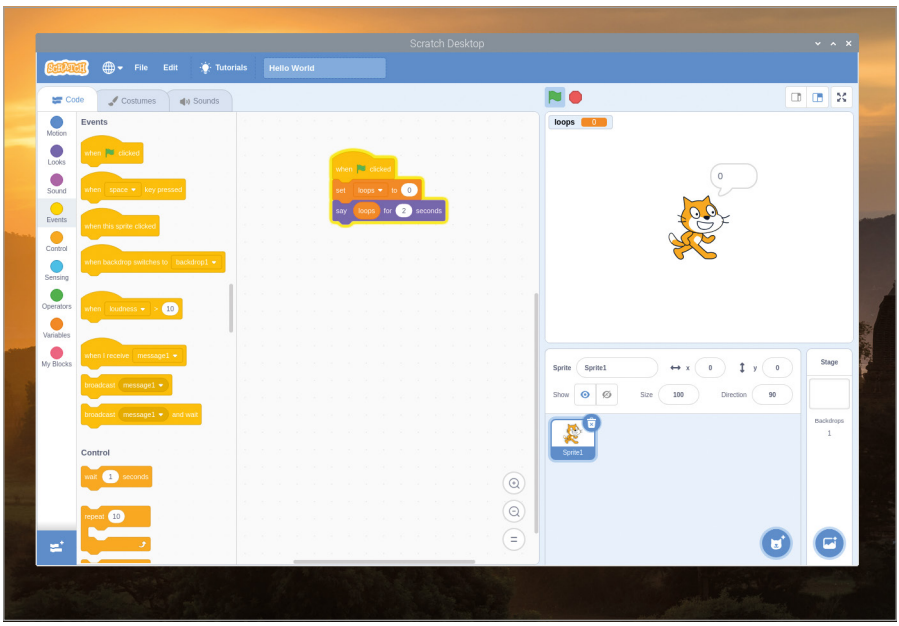
Figura 4-5: Dê um nome à sua nova variável

Clique e arraste o bloco altera ciclos para 0 para a área de programação. Esta ação indica ao seu programa para *inicializar* a variável com um valor de 0. Em seguida, clique na categoria Aparência da palete de blocos e arraste o bloco diz Olá! durante 2 s para debaixo do seu bloco altera ciclos para 0.

Como descobriu anteriormente, os blocos diz Olá! fazem com que o sprite do gato diga o que estiver escrito neles. No entanto, em vez de você próprio escrever a mensagem no bloco, pode usar uma variável no respetivo lugar. Clique novamente na categoria Variáveis da palete de blocos e, em seguida, clique e arraste o bloco arredondado ciclos, conhecido como *blocorepórter* que está no topo da lista, junto a uma caixa de verificação, em cima da

palavra "Olá!" no seu bloco diz Olá! durante 2 s. Esta ação cria um novo bloco combinado: diz ciclos durante 2 s.

Clique na categoria Eventos na palete de blocos e, em seguida, clique e arraste o bloco quando alguém clicar em para o colocar no topo da sua sequência de blocos. Clique na bandeira verde acima da área do palco, e verá o sprite do gato dizer "0" (**Figura 4-6**), o valor que deu à variável "ciclos".



▲ **Figura 4-6:** Desta vez o gato vai dizer o valor da variável

As variáveis no entanto, não são imutáveis. Clique na categoria Variáveis na palete de blocos e, em seguida, clique e arraste o bloco adiciona a ciclos o valor 1 para o fundo da sua sequência de blocos. Em seguida, clique na categoria Controlo e, em seguida, clique e arraste um bloco repete 10 vezes e largue-o para que comece diretamente por baixo do seu bloco altera o ciclo para 0 e se molde à volta dos blocos restantes nasua sequência.

Clique na bandeira verde novamente. Desta vez, verá o gato contar de forma crescente de 0 a 9. Esta ação funciona porque o seu programa está agora a mudar, ou *modificando*, a variável em si: cada vez que o ciclo é executado, o programa adiciona um ao valor na variável "ciclos" (**Figura 4-7**).

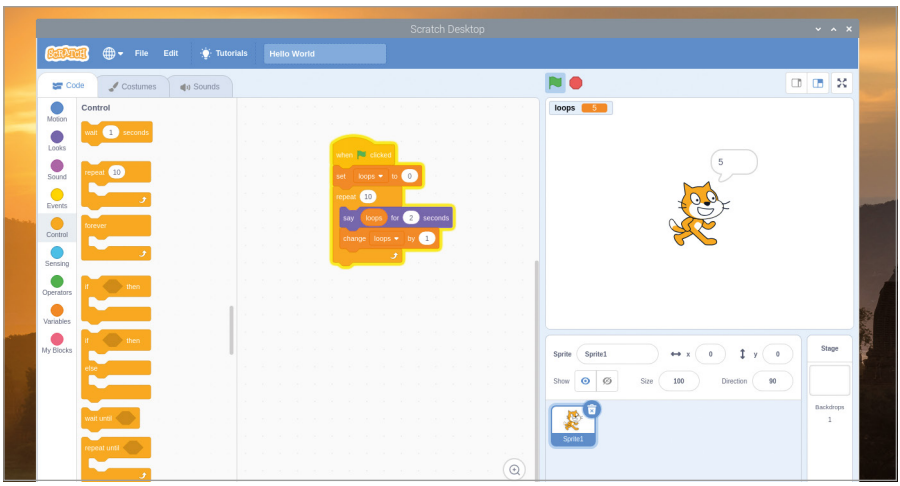
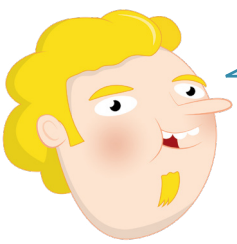


Figura 4-7: Graças ao ciclo, o gato agora conta de forma crescente

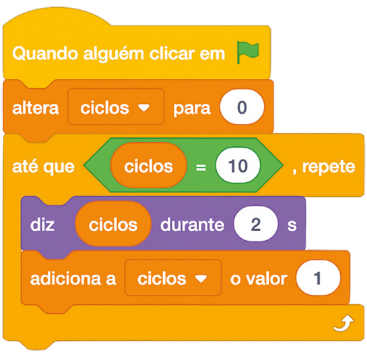


A CONTAR A PARTIR DE ZERO

Embora o ciclo que criou seja executado dez vezes, o sprite do gato só conta até nove. Esta situação acontece porque a nossa variável está a começar com um valor zero. Incluindo o zero e o nove, há dez números entre zero e nove, portanto o programa para antes que o gato diga "10". Para alterar isso, pode definir o valor inicial da variável para 1 em vez de 0.

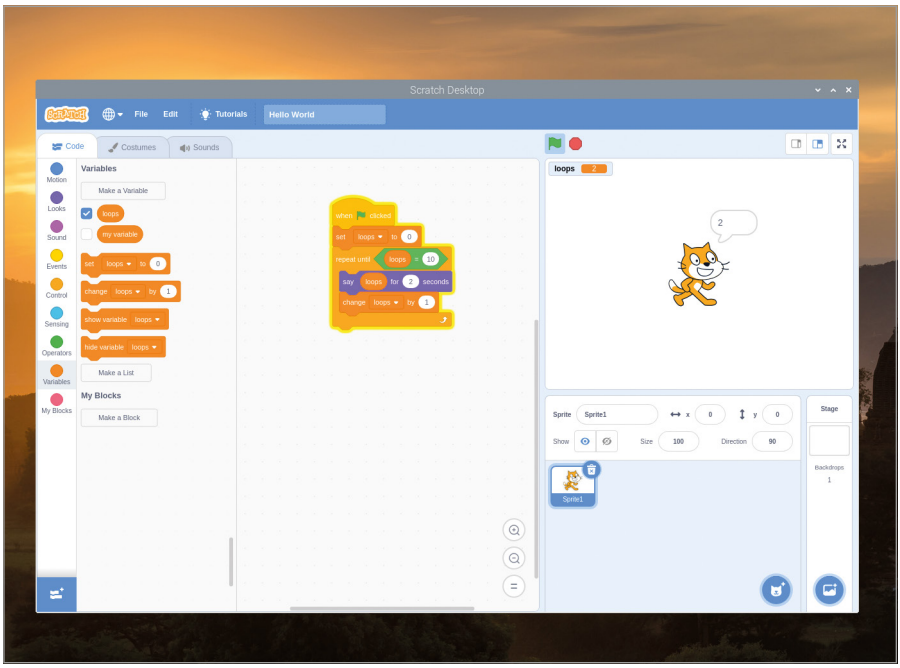
Pode fazer mais com uma variável do que modificá-la. Clique e arraste o bloco diz ciclos durante 2 s para libertá-lo do bloco repete 10 vezes e largá-lo abaixo do bloco repete 10. Clique e arraste o bloco repete 10 para a palete de blocos para eliminá-lo e, em seguida, substitua-o com um bloco até que , repete, certificando-se de que o bloco está ligado ao fundo do bloco altera o ciclo para 0 e rodeia ambos os outros blocos na sua sequência. Clique na categoria Operadores na palete de blocos e, em seguida, clique e arraste o bloco em forma de diamante (=) e largue-o no orifício em forma de diamante correspondente no bloco até que , repete.

Este bloco Operadores permite-lhe comparar dois valores, incluindo variáveis. Clique na categoria Variáveis, arraste o bloco repórter ciclos para o espaço vazio no bloco Operadores (=) e, em seguida, clique no espaço com "50" e introduza o número "10".



Clique na bandeira verde acima da área do palco e verá que o programa funciona da mesma forma que anteriormente: o sprite do gato conta de 0 a 9 (**Figura 4-8**) e depois o programa

para. Esta ação acontece porque o bloco até que , repete funciona exatamente da mesma forma que o bloco repete 10, mas em vez de contar o número de ciclos em si, está a comparar o valor da variável "ciclos" com o valor que introduziu à direita do bloco. Quando a variável "ciclos" chega ao 10, o programa para.



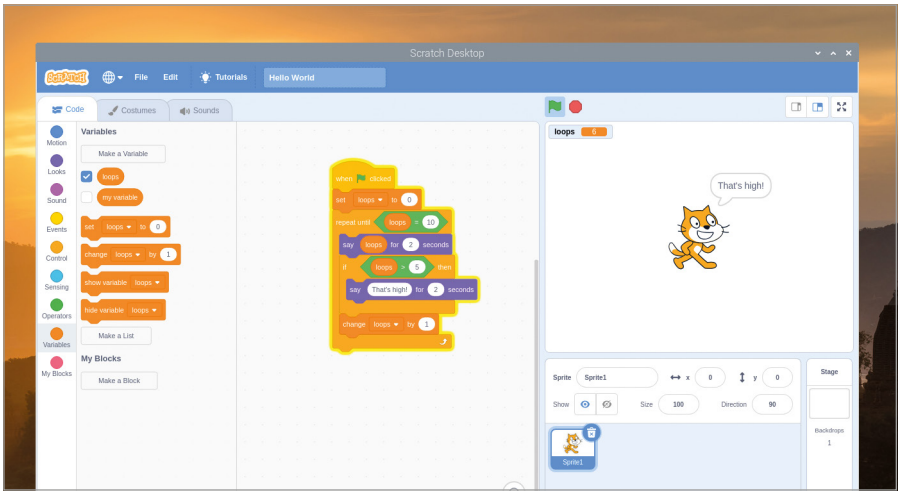
▲ **Figura 4-8:** Usando um bloco "repetir até" com um operador comparativo

Isto é conhecido como um *operador comparativo* literalmente compara, dois valores. Clique na categoria Operadores da palete de blocos, e encontre os outros dois blocos em forma de diamante acima e abaixo daquele com o símbolo "=". Estes também são operadores comparativos: "<" compara dois valores e é acionado quando o valor da esquerda é menor que o da direita, e ">" é acionado quando o valor da esquerda é maior que o da direita.

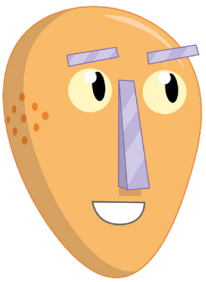
Clique na categoria Controlo da palete de blocos, encontre o bloco se então e, em seguida, clique e arraste-o para a área de programação antes de largá-lo diretamente por baixo do bloco diz ciclos durante 2 s. Irá automaticamente rodear o bloco altera o ciclo para 1, em seguida, clique e arraste-o sobre o mesmo para movê-lo de modo a que se ligue ao fundo do seu bloco se então. Clique na categoria Aparência da palete de blocos e, em seguida, arraste e largue um bloco diz Olá! durante 2 s para o interior do seu bloco se então. Clique na categoria Operadores na palete de blocos e, em seguida, clique e arraste o bloco = para o orifício em forma de diamante no seu bloco se então.

O bloco **se então** é um bloco condicional, o que significa que os blocos no seu interior apenas serão executados se uma determinada condição for cumprida. Clique na categoria Variáveis da palete de blocos, arraste e largue o bloco repórter **ciclos** para o espaço vazio no seu bloco **= =** e, em seguida, clique no espaço com um "50" escrito no mesmo e introduza o número "5". Finalmente, clique na palavra "Olá!" no seu bloco **diz Olá! durante 2 s** e escreva "Isso é alto".

Clique na bandeira verde. Em primeiro lugar, o programa funcionará como antes, com o sprite do gato a contar de forma crescente a contar do zero. Quando o número chegar a 6, o primeiro número maior que 5, o bloco **se então** começará a ser acionado e o sprite do gato comentará sobre o quão alto os números estão a ficar (**Figura 4-9**). Parabéns: já pode trabalhar com variáveis e condicionais!



▲ **Figura 4-9:** O gato comenta quando o número chegar ao 6



DESAFIO: ALTO E BAIXO

Como poderá alterar o programa para que o sprite do gato comente o quão baixos são os números abaixo de 5? Pode alterar o programa para que o gato comente tanto os números altos como os baixos? Experimente com o bloco **se então senão** para tornar este desafio mais fácil!

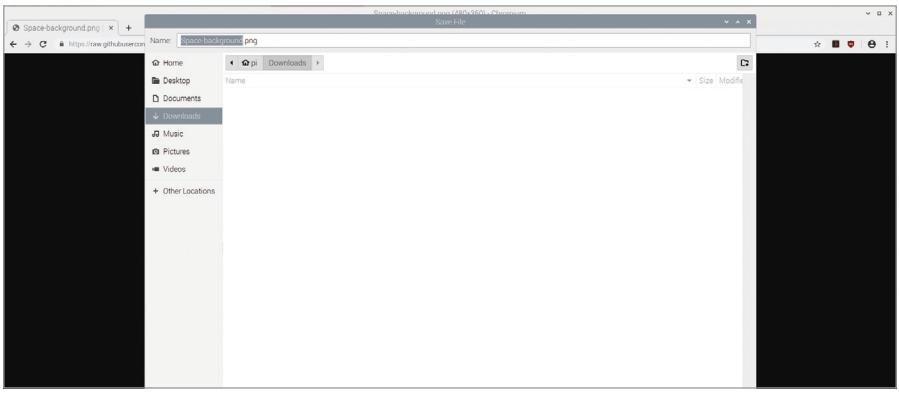
Projeto 1: Temporizador de Reação dos Astronautas

Agora que já percebe como funciona o Scratch, está na hora de fazer algo um pouco mais interativo: um temporizador de reação, concebido em honra do astronauta britânico da ESA Tim Peake e o respetivo tempo a bordo da Estação Espacial Internacional.

Guarde o seu programa atual, se quiser mantê-lo e, em seguida, abra um novo projeto, clicando em Arquivo e Novo. Antes de começar, dê-lhe um nome clicando em Arquivo e "Descarregar para o eu computador": chame-lhe "Temporizador de reação dos astronautas".

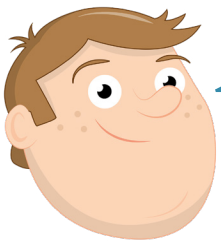
Este projeto tem base em duas imagens, uma como fundo do palco, outra como sprite, que não estão incluídas nos recursos integrados do Scratch. Para transferi-las, clique no ícone da framboesa para carregar o menu do Raspberry Pi OS, mova o cursor para a secção de

Programação, e clique no navegador web Chromium. Quando o navegador tiver carregado, escreva **rpf.io/astronaut-backdrop**na barra de endereço e, em seguida, prima a tecla **ENTER**. Clique com o botão direito do rato na imagem do espaço e clique em "Guardar imagem como..." e, em seguida, clique no botão Guardar (**Figura 4-10**). Clique novamente na barra de endereço, e escreva **rpf.io/astronaut-sprite** e, em seguida, prima a tecla **ENTER**.



▲ **Figura 4-10:** Guardar a imagem de fundo

Novamente, clique no botão direito na fotografia de Tim Peake e clique em "Guardar imagem como..." em seguida, selecione a pasta das Transferências e clique no botão Guardar. Com essas duas imagens guardadas, pode fechar o Chromium ou deixá-lo aberto e usar a barra de tarefas para alternar para o Scratch 3.



INTERFACE DE UTILIZADOR

Se tiver seguido este capítulo desde o início, deve estar familiarizado com a interface de utilizador do Scratch 3. As seguintes instruções do projeto dependem se sabe onde está tudo; caso se esqueça onde encontrar algo, volte atrás e consulte a imagem da interface do utilizador no início deste capítulo para se recordar.

Clique com o botão direito do rato no sprite do gato na lista e clique em "eliminar". Sobreponha o ponteiro do rato sobre o ícone Escolher um pano de fundo e, em seguida, clique no ícone Carregar pano de fundo na lista que aparece. Encontre o ficheiro **Space-background.png** na pasta Transferências, clique sobre esta para a selecionar e, em seguida, clique em OK. O fundo branco liso do palco mudará para a imagem do espaço, e a área de programação será substituída pela área dos panos de fundo (**Figura 4-11**). Aqui pode desenhar sobre o pano de fundo, mas por enquanto basta clicar no separador com o nome Programação na parte superior da janela do Scratch 3.

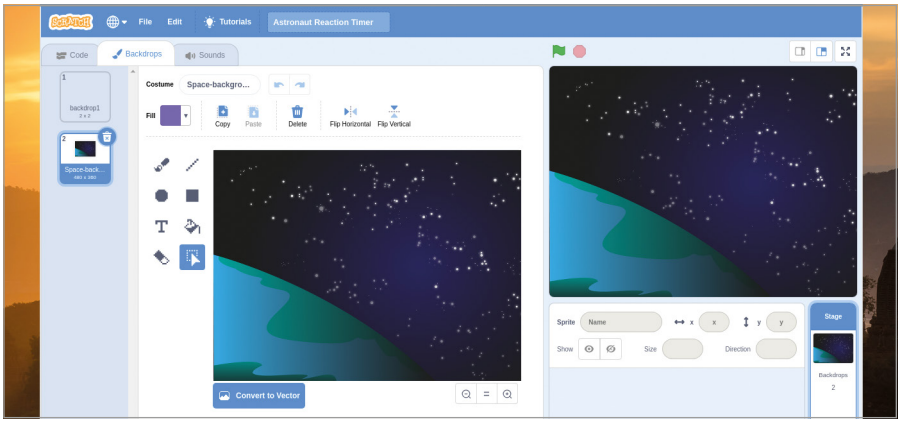


Figura 4-11: O espaço aparece como fundo no palco

Carregue o seu novo sprite; para isso, sobreponha o ponteiro do rato sobre o ícone Escolher um sprite, e, em seguida, clique no ícone Carregar sprite na parte superior da lista que aparece. Encontre o ficheiro **Astronaut-Tim.png** na pasta Transferências, clique sobre esta para a selecionar e, em seguida, clique em OK. O sprite aparece no palco automaticamente, mas pode não estar no centro: clique e arraste-o com o rato e largue-o para que fique perto do centro inferior (**Figura 4-12**).

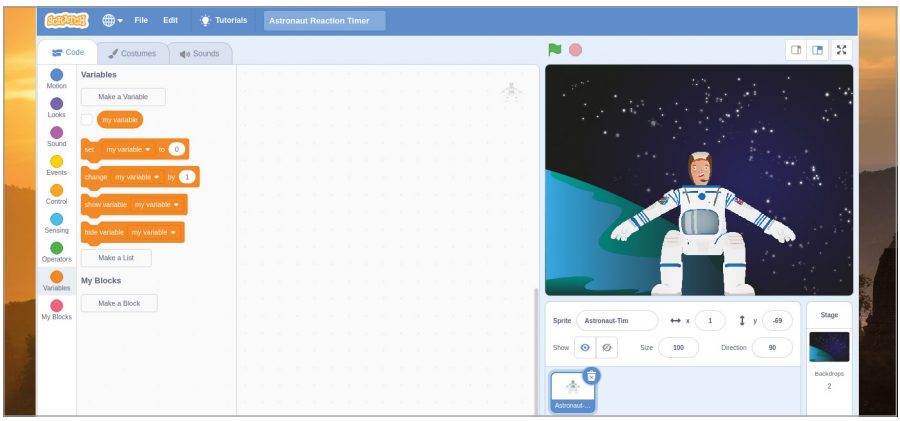
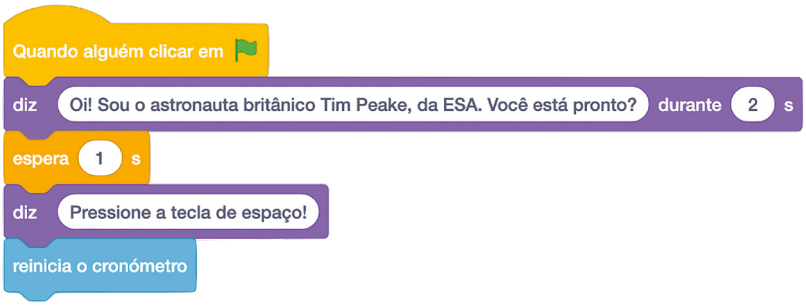


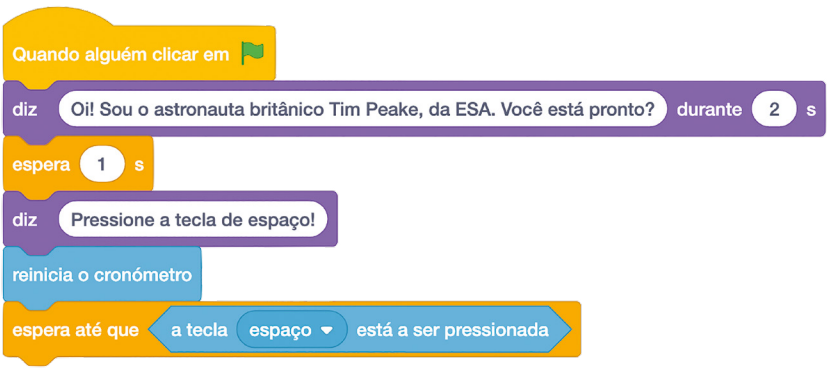
Figura 4-12: Arraste o sprite do astronauta para o centro inferior do palco

Com o seu novo pano de fundo e o sprite nos respetivos lugares, está pronto para criar o seu programa. Comece por criar uma nova variável chamada "tempo", certificando-se de que a opção "Para todos os sprites" está selecionada antes de clicar em OK. Clique no seu sprite, no palco ou no painel de sprites, para selecioná-lo e, em seguida, adicione um bloco quando alguém clicar em da categoria Eventos à área de programação. Em seguida, adicione um bloco diz Olá! durante 2 s da categoria Aparência e, em seguida, clique no mesmo para alterá-lo para dizer "Olá! O astronauta britânico da ESA Tim Peake está aqui. Está pronto?"

Adicione um bloco espera 1 s da categoria Controlo e, em seguida, um bloco diz Olá!. Altere este bloco para dizer "Prima Espaço!" e, em seguida, adicione um bloco reinicia o cronómetro da categoria Deteção. Esta ação controla uma variável especial incorporada no Scratch para cronometrar as coisas, e será usado para cronometrar a rapidez com que pode reagir no jogo.

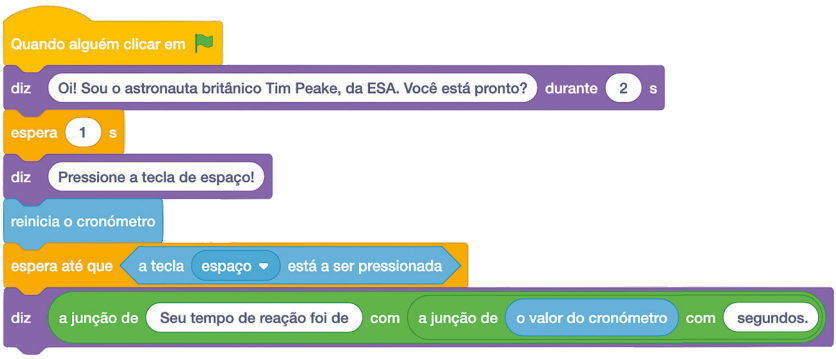


Adicione um bloco de controlo espera até que e, em seguida, arraste um bloco de deteção a tecla espaço está a ser pressionada? para o respetivo espaço em branco. Esta ação irá suspender o programa até que prima a tecla **ESPAÇO** no teclado, mas o temporizador continuará a funcionar, a cronometrar exatamente quanto tempo passa desde que recebe a mensagem que lhe indica para "Carregar no Espaço" e o momento em que de facto carrega na tecla **ESPAÇO**.



Agora, precisa que o Tim lhe indique quanto tempo demorou a premir a tecla **ESPAÇO** mas de uma forma que seja fácil de ler. Para fazer isso, precisa de um bloco Operadores a junção de. Esta ação usa dois valores, incluindo variáveis, e junta-os um após o outro, processo conhecido como *concatenação*.

Comece com um bloco **diz Olá!** e, em seguida, arraste e solte um bloco Operadores **a junção de** sobre a palavra "Olá!". Clique em "maçã" e escreva "O seu tempo de reação foi", certifique-se de que adiciona um espaço em branco no final, em seguida, arraste outro bloco de junção sobre a parte superior de "banana" na segunda caixa. Arraste um bloco repórter **cronómetro** da categoria Deteção para a que é agora a caixa do meio e escreva " segundos" na última caixa, certificando-se de que inclui um espaço em branco no início.



Por fim, arraste um bloco de variáveis **altera a minha variável para 0** para o fim da sua sequência. Clique na seta para baixo junto a "a minha variável" e clique em "tempo" na lista, em seguida, substitua o "0" por um bloco repórter **cronómetro** da categoria Deteção. O seu jogo já está pronto para testar clicando na bandeira verde acima do palco. Prepare-se, e assim que vir a mensagem "Prima Espaço!", prima **ESPAÇO**o mais rápido possível (**Figura 4-13**), tente bater o nosso recorde!

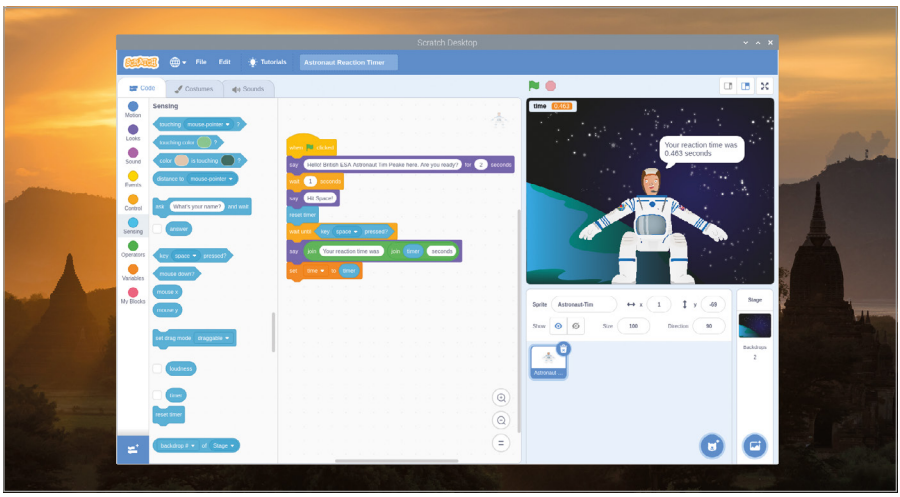
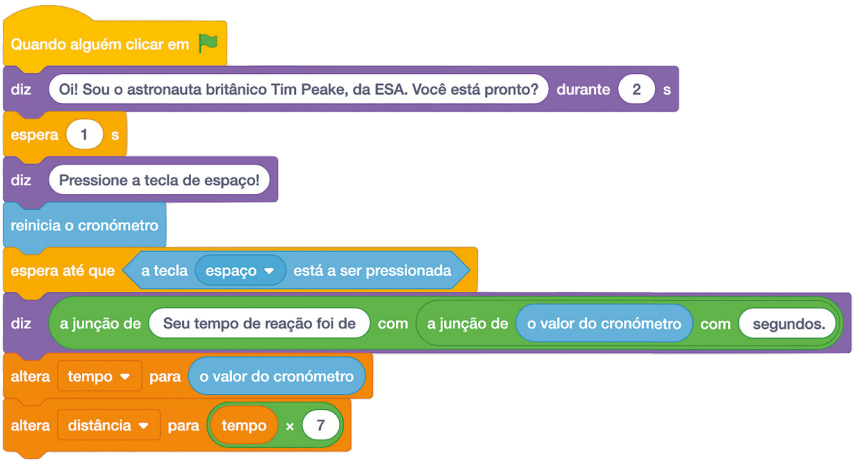


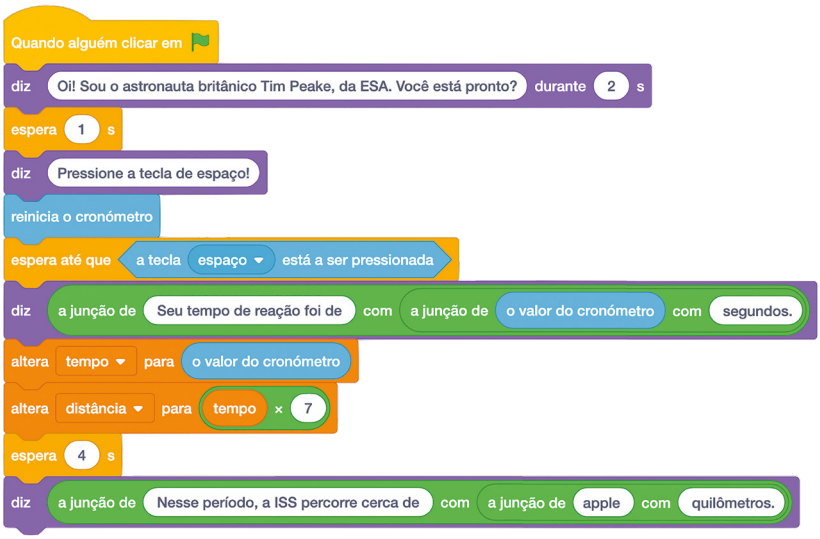
Figura 4-13: Está na hora de jogar!

Pode usufruir ainda mais deste projeto ao fazer com que este calcule aproximadamente a distância percorrida pela Estação Espacial Internacional no tempo que demorou a premir **ESPAÇO** com base na velocidade da estação de sete quilómetros por segundo. Primeiro, crie uma nova variável chamada "distância". Repare como os blocos da categoria Variáveis mudam automaticamente para mostrar a nova variável, mas os blocos das variáveis tempo atuais no seu programa permanecem iguais.

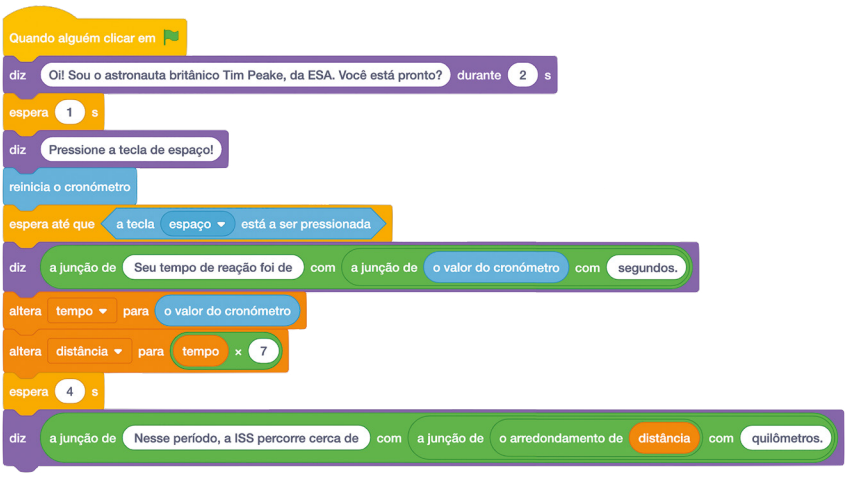
Adicione um bloco altera distância até 0 e, em seguida, arraste um bloco Operadores (*), indicando multiplicação, sobre o "0". Arraste um bloco repórter tempo sobre o primeiro espaço em branco e, em seguida, escreva o número "7" no segundo espaço. Quando terminar, no seu bloco combinado constará altera distância até tempo * 7. Esta ação irá ter em conta o tempo que demorou a premir a tecla **ESPAÇO** e irá multiplicá-lo por sete, para obter a distância percorrida em quilómetros pela EEI.



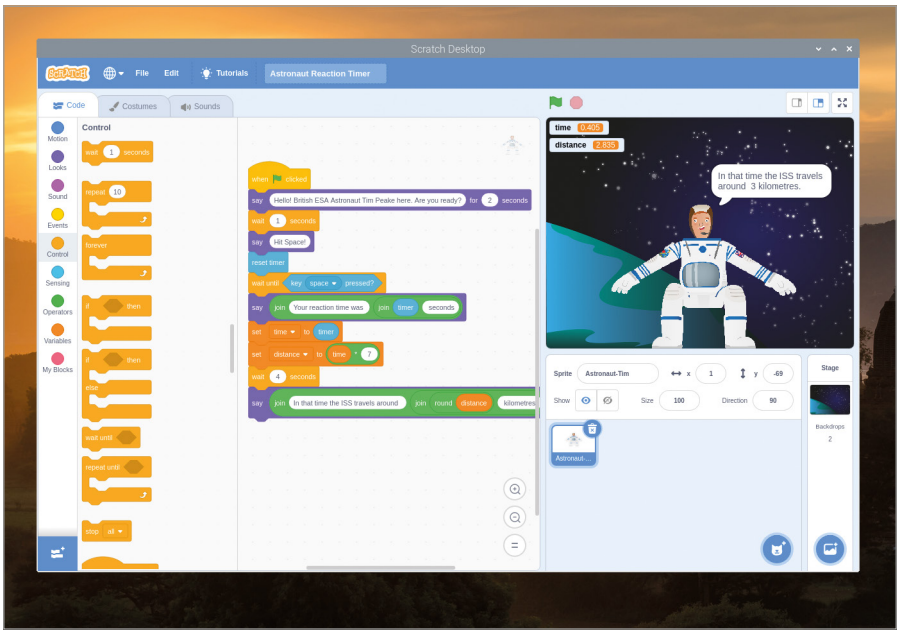
Adicione um bloco espera 1 s e altere-o para "4". Por fim, arraste outro bloco diz Olá! para o fim da sua sequência e adicione dois blocos a junção de, como anteriormente. No primeiro espaço, sobre "maçã" introduza "O tempo em que a EEI viaja em volta", não se esqueça de incluir o espaço no final; no espaço "banana" escreva "quilómetros", não se esqueça de colocar o espaço no início.



Por fim, arraste um bloco Operadores **o arredondamento de** para o espaço em branco do meio e, em seguida, arraste um bloco repórter **distância** para o novo espaço em branco que este cria. O bloco **o arredondamento de** arredonda os números para cima ou para baixo para o respetivo número inteiro mais próximo, por isso, em vez de obter um número de quilómetros extremamente preciso mas difícil de ler, obterá um número inteiro fácil de ler.



Clique na bandeira verde para executar o seu programa, e veja quão longe viaja a EEI no tempo que demorar a carregar na tecla **ESPAÇO**. Lembre-se de guardar o seu programa quando tiver terminado, para que no futuro possa facilmente carregá-lo novamente sem ter de começar do início!



▲ **Figura 4-14:** O Tim indica-lhe a distância que a ISS já percorreu

DESAFIO: QUEM É RÁPIDO?

Para além do astronauta, que outras profissões necessitam de reflexos de frações de segundos? Pode desenhar os seus próprios sprites e fundos para apresentar uma dessas profissões?

Projeto 2: Natação Sincronizada

A maioria dos jogos utiliza mais do que um único botão, e este projeto demonstra-o oferecendo um controlo de dois botões usando as teclas ← e → no teclado.

PROJETO ONLINE

Este projeto também está disponível online em **rpf.io/synchro-swimming**

Crie um novo projeto e guarde-o como "Natação Sincronizada". Clique no Palco na secção de controlo de palco e, em seguida, clique no separador Panos de fundo na parte superior esquerda. Clique no botão Converter em Bitmap por baixo do pano de fundo. Escolha uma cor semelhante à cor da água na palete Preencher, clique no ícone Preencher e, em seguida, clique no pano de fundo axadrezado para preenchê-lo com a cor azul (**Figura 4-15**).

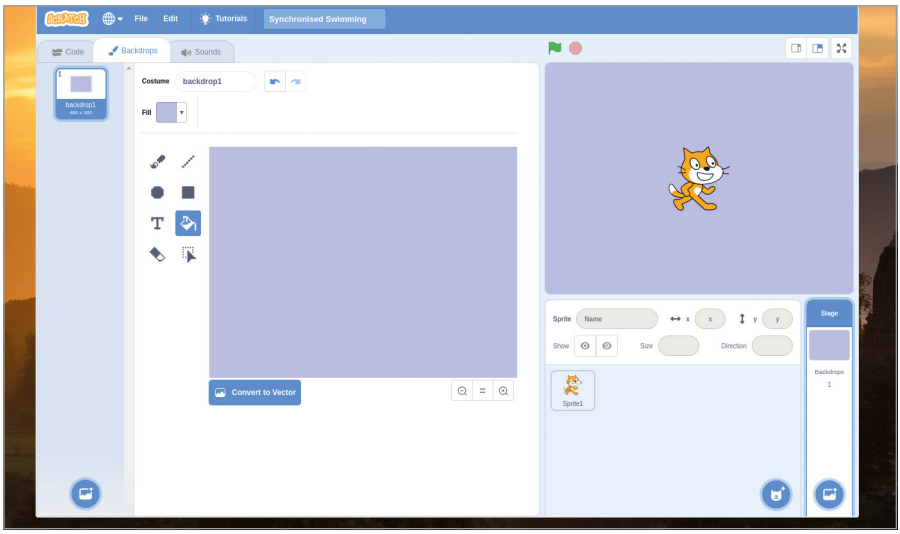


Figura 4-15: Preencha o fundo com uma cor azul

Clique com o botão direito do rato no sprite do gato na lista e clique em "eliminar". Clique no ícone "Escolher um Sprite" para ver uma lista de sprites integrados. Clique na categoria Animais, em seguida, em "Gato a voar" (**Figura 4-16**), e em OK. Este sprite também é indicado para projetos de natação.

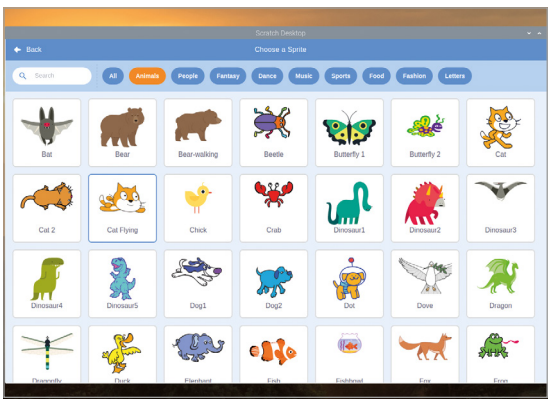
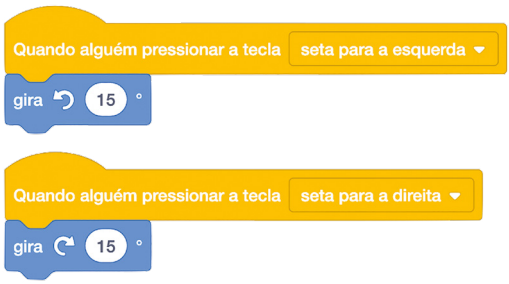


Figura 4-16: Escolha um sprite da biblioteca

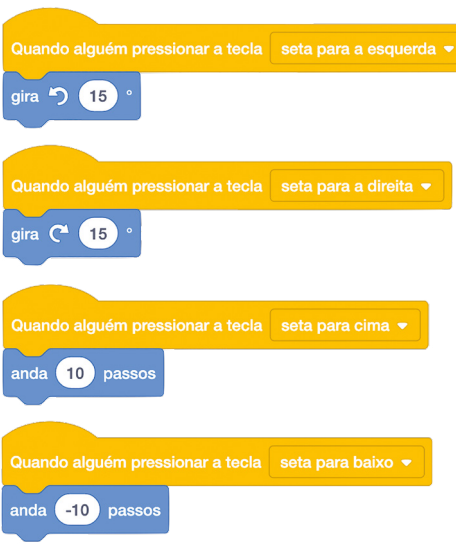
Clique no novo sprite e, em seguida, arraste dois blocos de Eventos quando alguém pressionar a tecla para a área de programação. Clique na pequena seta para-baixo junto à palavra "espaço" no primeiro bloco e selecione "seta para a esquerda" na lista de opções possíveis. Arraste um bloco de Movimento gira 15º sob o seu bloco

quando alguém pressionar a tecla seta para cima, em seguida, faça o mesmo com o seu segundo bloco de Eventos, exceto não selecionar "seta para a direita" na lista e usar um bloco de gira 15º Movimento.



Prima as teclas ← ou → para testar o seu programa. Verá o sprite do gato girar consigo, na direção que está a premir no teclado. Constatará que, não precisou de clicar na bandeira verde desta vez; esta situação acontece porque os blocos acionadores de Eventos que usou estão sempre ativos, mesmo quando o programa não está a ser "executado" como normalmente.

Faça os mesmos passos novamente duas vezes, mas desta vez selecione "seta para cima" e "seta para baixo" para os blocos acionadores de Eventos e, em seguida, anda 10 passos e anda -10 passos para os blocos de Movimento. Prima as setas agora e verá que o seu gato também pode virar-se e nadar para frente e para trás!



Para tornar o movimento do sprite do gato mais realista, pode mudar como este aparece – processo conhecido na terminologia do Scratch como o respetivo *fato*. Clique no sprite do

gato e, em seguida, clique no separador Fatos por cima da palete de blocos. Clique no fato "gato a voar-a" e clique no ícone do X no caixote que aparece no canto superior direito para o eliminar. Em seguida, clique no fato "gato a voar-b" e use a caixa de nome na parte superior para mudar-lhe o nome para "direita" (**Figura 4-17**).

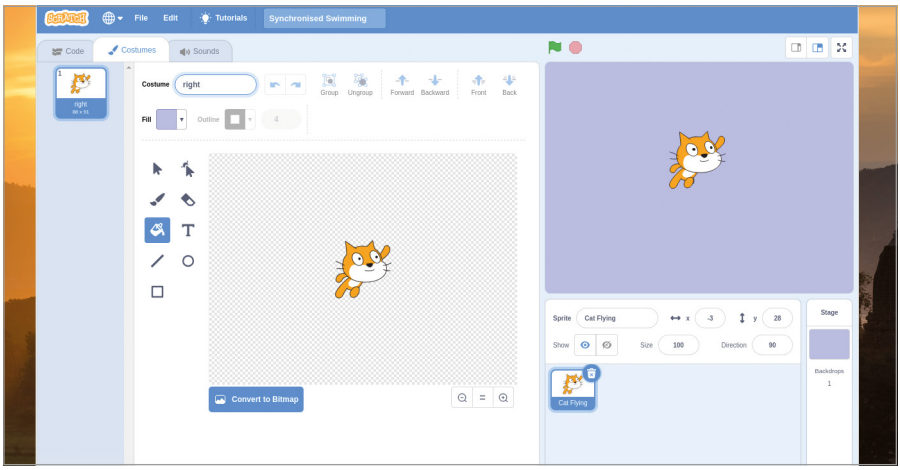


Figura 4-17: Mude o nome do fato para "direita"

Clique com o botão direito do rato sobre o fato cujo nome alterou recentemente para "direita" e clique em "duplicar" para criar uma cópia. Clique nesta cópia para selecioná-la, clique no ícone Selecionar, clique em Inverter Horizontalmente e, em seguida, altere o respetivo nome para 'esquerda' (**Figura 4-18**). Vai acabar com dois "fatos" para o seu sprite, que são exatamente imagens espelhadas: um chamado "direita" com o gato virado para a direita, e um chamado "esquerda" com o gato virado para a esquerda.

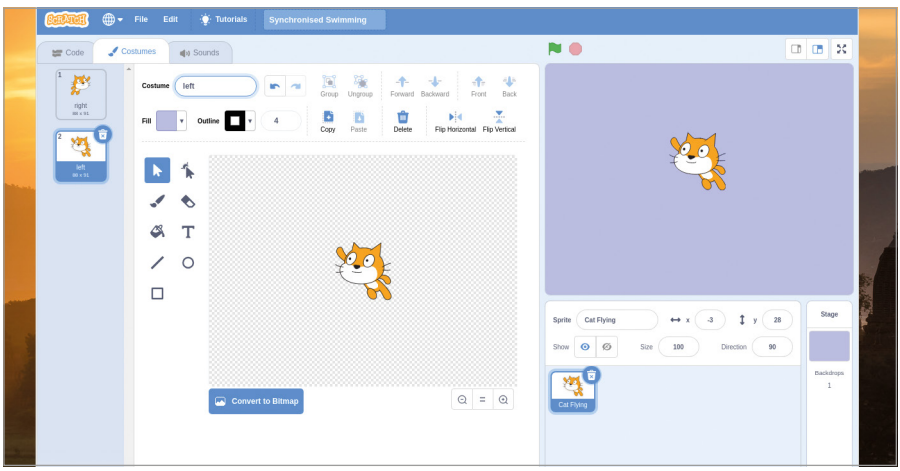
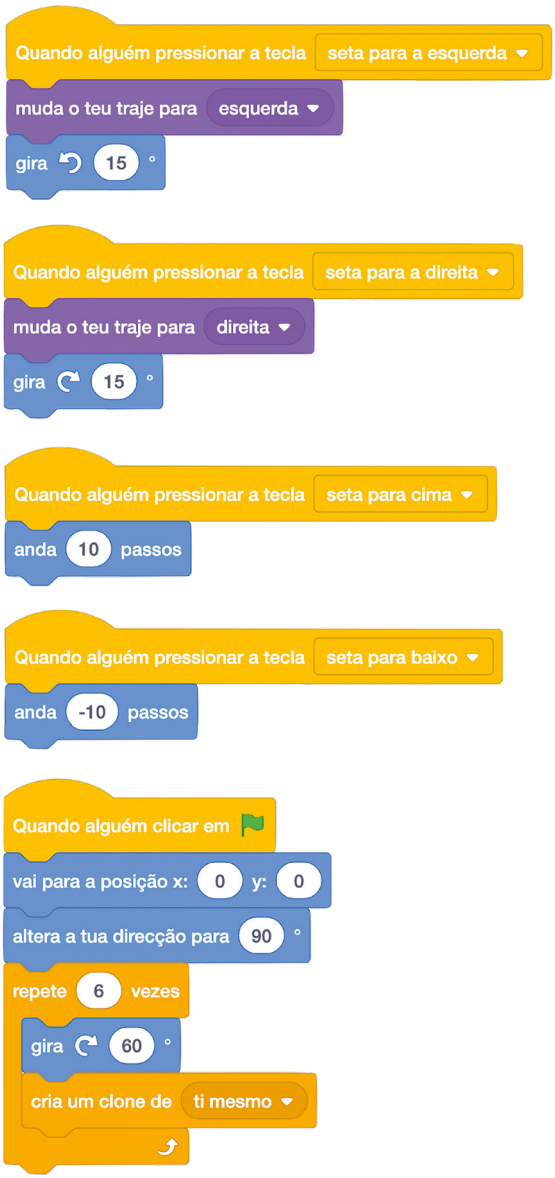


Figura 4-18: Duplique o fato, inverta-o e mude-lhe o nome para "esquerda"

Clique no separador Programação por cima da área dos fatos e, em seguida, arraste dois blocos **muda o teu traje para esquerda** Aparência por baixo dos seus blocos de Eventos seta para a esquerda e seta para a direita, altere o bloco por baixo do bloco da seta para a direita para **muda o teu traje para direita** . Experimente novamente as teclas do teclado; o gato parece agora virar-se para a direção em que está a nadar

No entanto, para a natação sincronizada ao estilo olímpico, precisamos de mais nadadores e de uma maneira de repor a posição do sprite do gato. Adicione um bloco Eventos quando alguém clicar em, em seguida, por baixo, adicione um bloco Movimento vai para a posição x: 0 y: 0, alterando os valores se necessário, e um bloco Movimento altera a tuo direcção a 90. Agora, quando clicar na bandeira verde, o gato irá ser movido para o meio do palco em direção à direita.

Para criar mais nadadores, adicione um bloco repete 6 vezes, alterando o valor padrão "10", e adicione um bloco de controlo cria um clone de ti mesmo no seu interior. Para que os nadadores não estejam todos a nadar na mesma direção, adicione um bloco gira 60º acima do bloco cria um clone de mas ainda no interior do bloco repete 6 . Clique na bandeira verde e experimente as setas do teclado agora para ver os seus nadadores ganharem vida!



Para completar o sentimento olímpico, precisará de adicionar algumas músicas. Clique no separador Sons acima da palete de blocos e, em seguida, clique no ícone Escolher um som . Clique na categoria Ciclos e, em seguida, navegue através da lista (**Figura 4-19**) até encontrar algumas músicas de que goste, escolhemos "Dance Around". Clique no botão OK para escolher a música e, em seguida, clique no separador Programação para abrir novamente a área de programação.

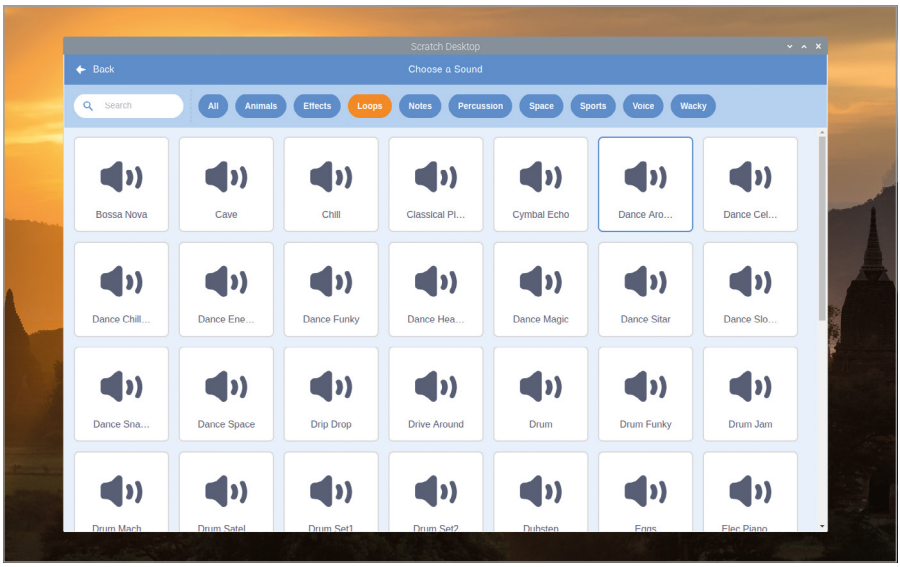
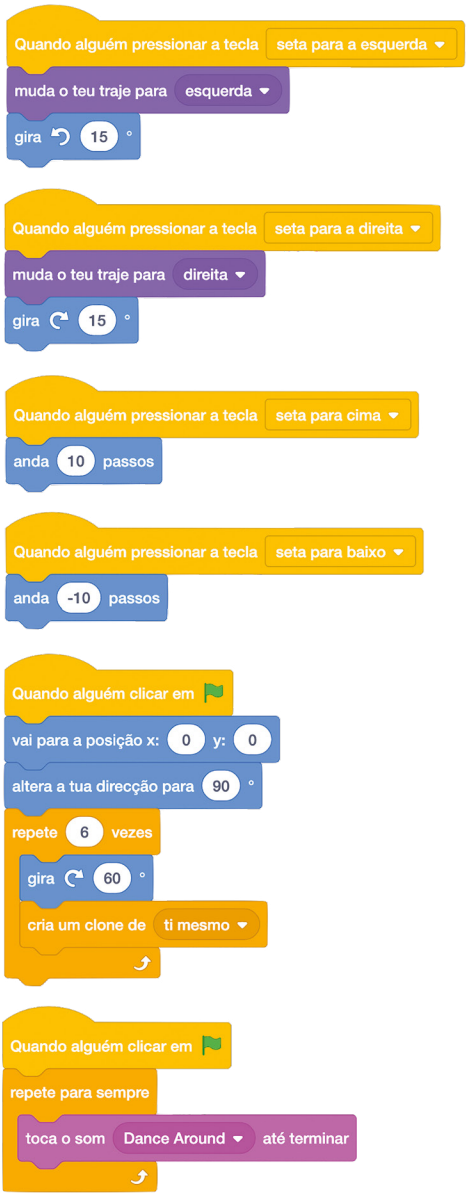
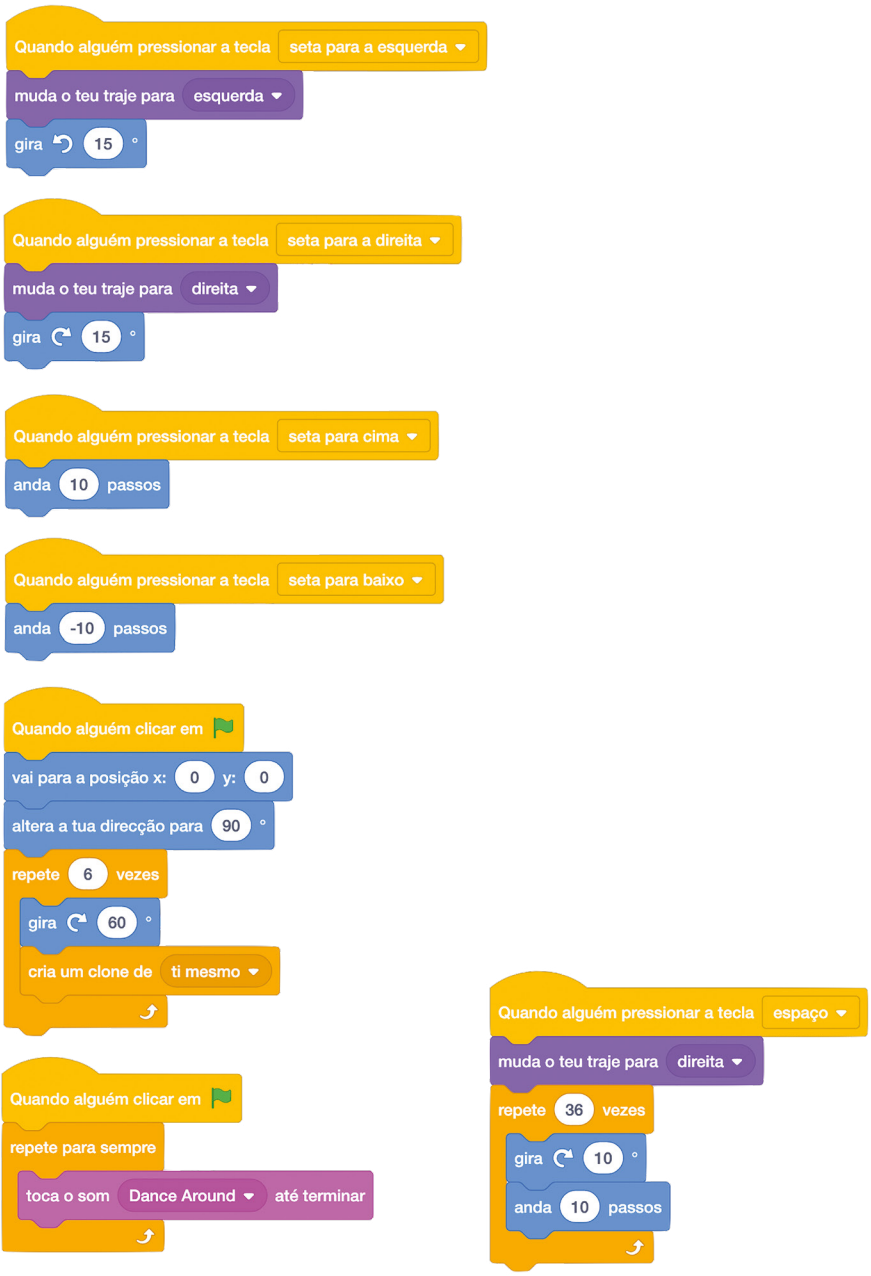


Figura 4-19: Selecione um ciclo de música da biblioteca de sons

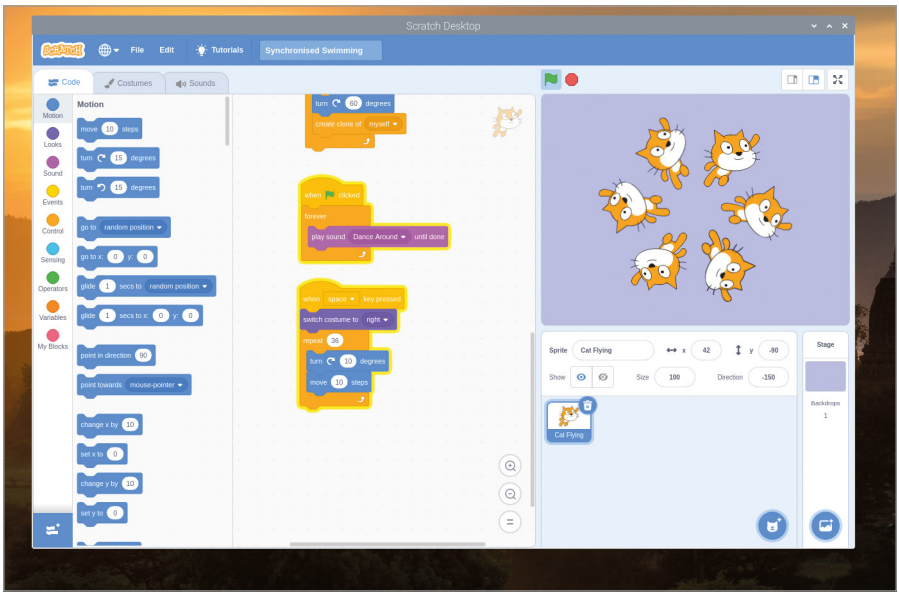
Adicione outro bloco de Eventos quando alguém clicar em à sua área de programação e, em seguida, adicione um bloco de Controlo repete para sempre. Dentro deste bloco de Controlo, adicione um bloco toca o som Dance Around até terminar, lembrando-se de procurar o nome da música que escolheu, e clique na bandeira verde para testar o seu novo programa. Se quiser parar a música, clique no octógono vermelho para parar o programa e silenciar o som!



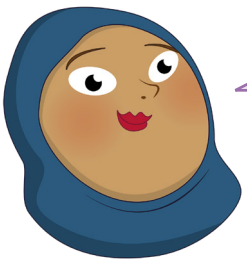
Por fim, pode simular uma coreografia de dança completa ao adicionar um novo acionador de evento ao seu programa. Adicione um bloco de Eventos **quando alguém pressiona a tecla** e, em seguida, um bloco **muda o teu traje para direita**. Por baixo disto, adicione um bloco **repete 36 vezes**, lembrando-se de alterar o valor do padrão, e no interior deste, um bloco **gira 10º** e um bloco **anda 10 passos**.



Clique na bandeira verde para iniciar o programa e, em seguida, prima a tecla **ESPAÇO** para experimentar a nova coreografia (**Figura 4-20**)! Não se esqueça de guardar o seu programa quando terminar.



▲ **Figura 4-20:** A coreografia de natação sincronizada terminada



DESAFIO COREOGRAFIA PERSONALIZADA

Consegue criar a sua própria coreografia de natação sincronizada com ciclos? O que precisaria de alterar se quisesse mais ou menos nadadores? Consegue adicionar várias coreografias de natação que podem ser acionadas usando diferentes teclas no teclado?

Projeto 3: Jogo de Tiro com Arco

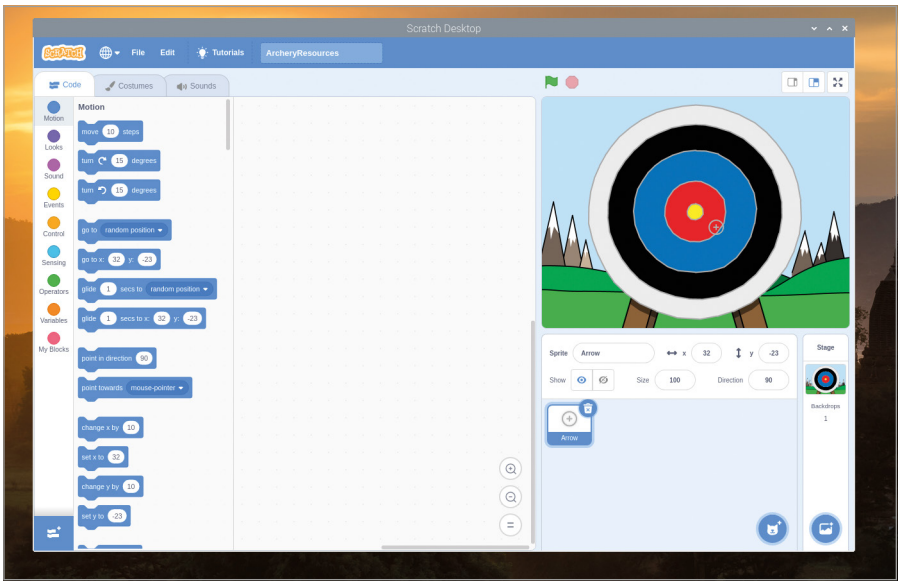
Por esta altura, já está a começar tornar-se quase um especialista em Scratch, está na hora de trabalhar em algo um pouco mais desafiante: um jogo de tiro com arco, onde o jogador tem de acertar num alvo com um arco e uma flecha que balançam de um lado para o outro aleatoriamente.

PROJETO ONLINE

Este projeto também está disponível online em **rpf.io/archery**

Comece por abrir o navegador Web Chromium e escreva **rpf.io/p/en/archery-go** e, em seguida, prima a tecla **ENTER**. Os recursos para o jogo são transferidos como um ficheiro zip, portanto terá de descompactá-lo (clique com o botão direito do rato e selecione Extrair Aqui). Volte ao Scratch 3 e clique no menu Ficheiro e, em seguida, em "Carregar do seu computador". Clique em

ArcheryResources.sb3 e, em seguida, no botão Abrir. Pergunta se quer substituir o conteúdo do seu projeto atual: se não guardou as suas alterações, clique em Cancelar e guarde-as, caso contrário clique em OK.

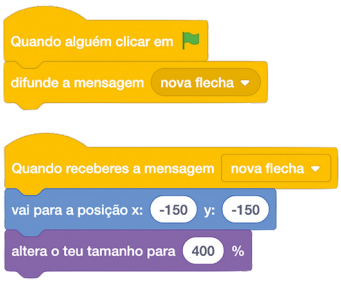


▲ **Figura 4-21:** Projeto de recursos carregado para o jogo de tiro com arco

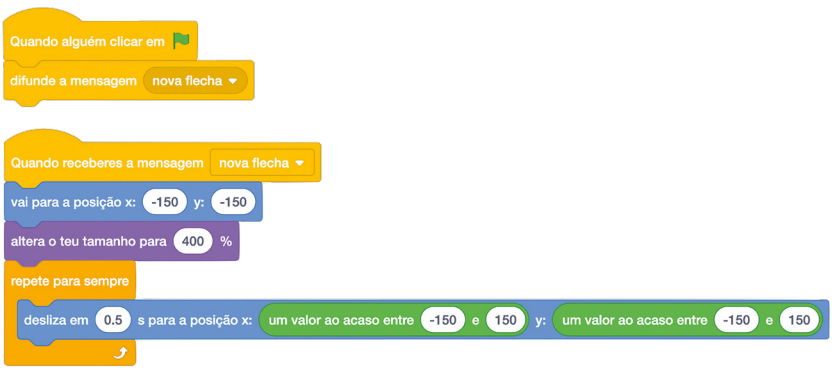
O projeto que acabou de carregar contém um pano de fundo e um sprite (**Figura 4-21**), mas não tem nenhum linha de código para fazer o jogo: adicioná-lo é o seu trabalho. Comece por adicionar um bloco de Eventos quando alguém clicar em e, em seguida, um bloco difunde a mensagem mensagem1. Clique na seta para baixo no final do bloco e, em seguida, em "Nova Mensagem", e introduza "nova seta" antes de clicar no botão OK. No seu bloco consta agora difunde a mensagem nova flecha.

Uma transmissão é uma mensagem de uma parte do seu programa que pode ser recebida por qualquer outra parte do seu programa. Para que realmente faça algo, adicione um bloco quando receberes a mensagem mensagem1, e altere-o novamente para dizer quando receberes a mensagem nova flecha. Desta vez pode apenas clicar na seta para baixo e escolher "nova seta" na lista; não precisa criar a mensagem novamente.

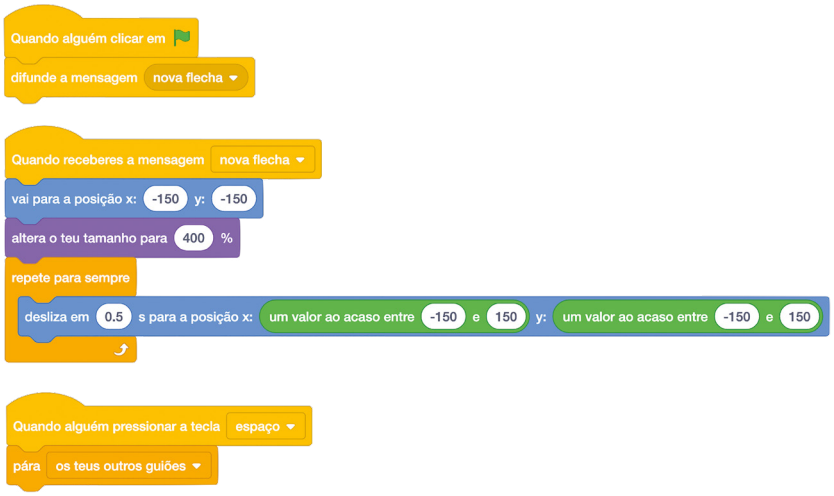
Abaixo do seu bloco quando receberes a mensagem nova flecha, adicione um bloco vai para a posição x: -150 y: -150 e um bloco altera o teu tamanho para 400 %. Lembre-se de que estes não são os valores padrão para esses blocos, portanto terá de alterá-los assim que os arrastar para a área de programação. Clique na bandeira verde para ver o que fez até agora: a seta do sprite, que o jogador usa para apontar para o alvo, saltará para o fundo esquerdo do palco e quadruplicará de tamanho.



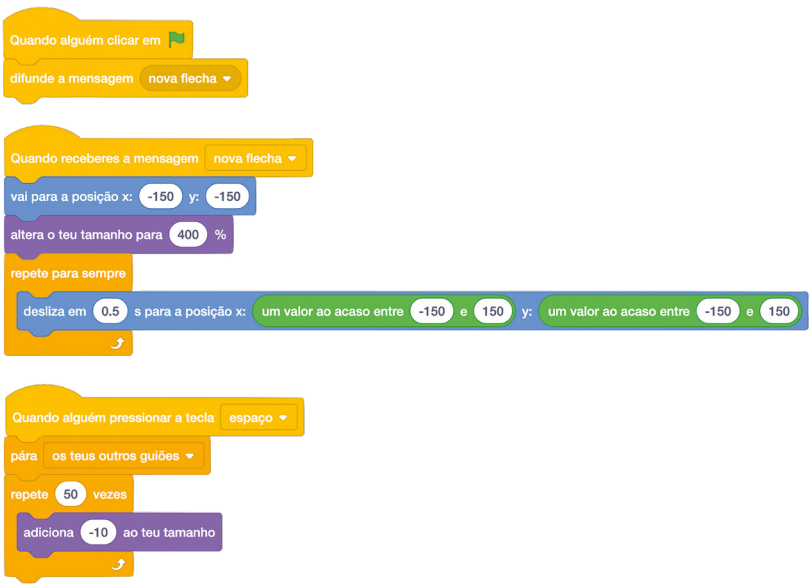
Para desafiar o jogador, adicione movimento ao simular o balanço à medida que o arco é armado e o arqueiro apontar. Arraste um bloco repete para sempre, e, em seguida um bloco desliza em 1 s para a posição x: -150 y: -150. Edite a primeira caixa branca para dizer "0,5" em vez de "1", em seguida, coloque um bloco Operadores um valor ao acaso entre -150 e 150 em cada uma das outras duas caixas brancas. Isto significa que a flecha irá andar à deriva pelo palco numa direção e distância aleatórias, tornando muito mais difícil acertar no alvo!



Clique novamente na bandeira verde, e verá o que esse bloco faz: o sprite da sua flecha está agora à deriva pelo palco, cobrindo diferentes partes do alvo. No entanto, de momento, não é possível que a flecha erre o alvo. Arraste um bloco quando alguém pressiona a tecla para a área de programação e, em seguida, um bloco de Controlo pára tudo. Clique na seta para baixo no final do bloco e altere-o para um bloco pára teus outros guiões.

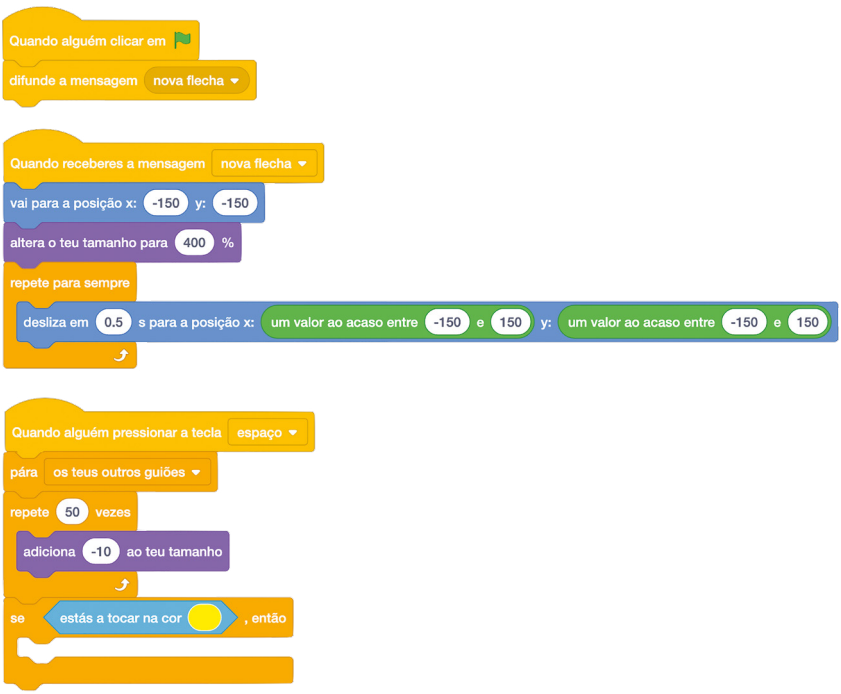


Se interrompeu o seu programa para adicionar os novos blocos, clique na bandeira verde para iniciá-lo novamente e, em seguida, prima a tecla **ESPAÇO**: verá que o sprite da flecha irá parar de se mover. Já é um começo, mas terá de transparecer que a flecha está a voar para o alvo. Adicione um bloco repete 50 vezes um bloco adiciona -10 ao teu tamanho e, em seguida clique na bandeira verde para testar o seu jogo novamente. Desta vez, a seta parece estar a voar na direção contrária ao utilizador e em direção ao alvo.



Para tornar o jogo divertido, tem de adicionar uma forma de estar a par da pontuação. Ainda na mesma pilha de blocos, adicione um bloco se então , certificando-se de que está abaixo do

bloco repete 50 e não no seu interior, com um bloco de Deteção estás a tocar na cor? no respetivo espaço em forma de diamante. Para escolher a cor correta, clique na caixa colorida no final do bloco de Deteção, no ícone de conta-gotas e, em seguida, clique na parte amarela do centro do seu alvo no palco.



Para que o jogador saiba que acertou, adicione um bloco toca o som cheer e um bloco diz 200 pontos durante 2 s no interior do bloco se então. Por fim, adicione um bloco difunde a mensagem nova flecha à última posição da pilha de blocos, abaixo e fora do bloco se então, para dar ao jogador outra flecha cada vez que disparar uma. Clique na bandeira verde para iniciar o seu jogo e tentar acertar na parte amarela do centro do alvo: quando o fizer, será recompensado com um aplauso da multidão e 200 pontos!

Para mais projetos no Scratch para experimentar, consulte o **Apêndice D: Leitura adicional**

O jogo funciona, mas é um pouco desafiante. Ao usar o que aprendeu neste capítulo, experimente estendê-lo para adicionar pontuações para acertar em partes do alvo que não no centro: 100 pontos para o vermelho, 50 pontos para o azul, e assim por diante.

DESAFIO: CONSEGUE MELHORÁ-LO?

Como tornaria o jogo mais fácil? Como tornaria o jogo mais difícil? Consegue usar variáveis para que a pontuação do jogador aumente à medida que dispare mais flechas? Consegue adicionar um contador regressivo para colocar mais pressão sobre o jogador?

Capítulo 5

Programar com Python

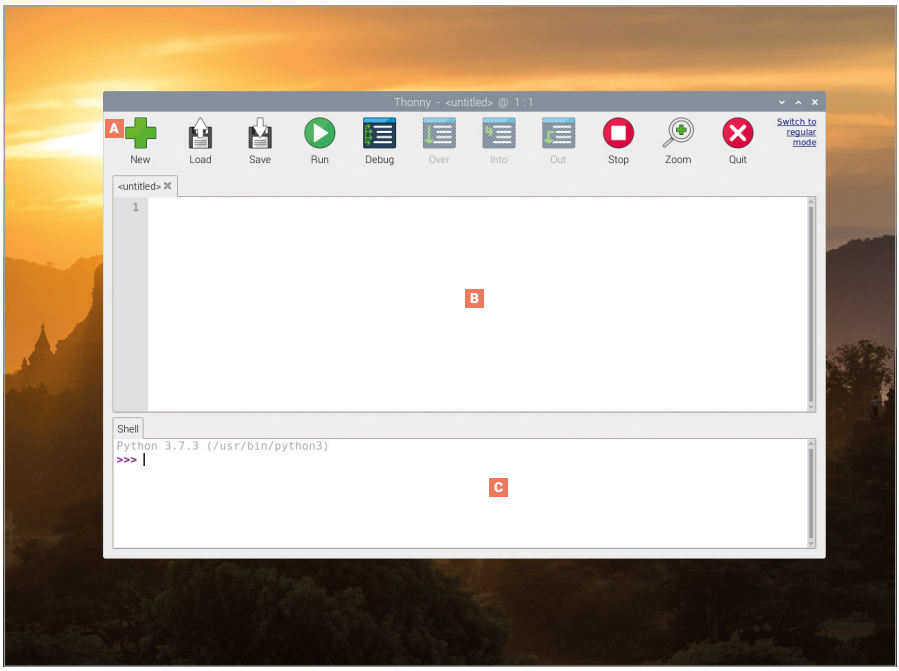
Agora que já domina o Scratch, vamos mostrar-lhe como fazer codificação com base em texto com o Python

Batizado com o nome da trupe de comédia Monty Python, o Python de Guido van Rossum evoluiu de um projeto de hobby apresentado ao público pela primeira vez em 1991 para uma linguagem de programação muito apreciada que alimenta uma vasta gama de projetos. Ao contrário do ambiente visual do Scratch, o Python é baseado em texto: escreve instruções numa linguagem simplificada e num formato específico para o computador executar.

O Python é um ótimo próximo passo para quem já utilizou o Scratch, oferecendo maior flexibilidade e um ambiente de programação mais "tradicional". Isso não quer dizer que é difícil de aprender: com um pouco de prática, qualquer pessoa pode escrever programas Python para tudo, desde cálculos simples até jogos surpreendentemente complicados.

Este capítulo desenvolve os termos e conceitos introduzidos no **Capítulo 4, Desenvolvimento com o Scratch 3**. Este capítulo é mais fácil se já tiver feito os exercícios do capítulo anterior, por isso, se ainda não o fez, volte atrás para os concluir.

Apresentamos o Thonny Python IDE



A **Toolbar** – A interface "Modo simples" do Thonny utiliza uma barra de ícones amigáveis como menu, permitindo-lhe criar, guardar, carregar e executar os seus programas Python, bem como testá-los de várias formas.

B **Script Area** – A área Script é onde os seus programas Python são escritos e está dividida numa área principal para o seu programa e uma pequena margem lateral para mostrar os números das linhas.

C **Shell do Python** – A Shell do Python permite escrever instruções individuais que são executadas de imediato quando prime o botão **ENTER**, e também fornece informações sobre os programas em execução.

VERSÕES DO THONNY

O Thonny tem duas versões de interface: Um 'Modo normal' e um 'Modo simples', que é melhor para iniciantes. Este capítulo utiliza o Modo simples, que é carregado por predefinição quando abre o Thonny na secção Programação do menu framboesa.

O seu primeiro programa em Python: Olá, Mundo!

Como qualquer outro programa pré-instalado no Raspberry Pi, o Thonny está disponível no menu: clique no ícone da framboesa, mova o cursor para a secção de Programação, e clique no Thonny Python IDE. Após alguns segundos, a interface de utilizador do Thonny (Modo simples predefinido) será carregada.

O Thonny é um pacote conhecido como *ambiente de programação integrado (integrated development environment – IDE)*, um nome que soa complicado, mas com uma explicação simples: reúne, ou *integra*, todas as diferentes ferramentas de que necessita para escrever, ou *programar*, software numa única interface de utilizador, ou *ambiente*. Há muitos IDEs disponíveis, alguns dos quais suportam muitas linguagens de programação diferentes, ao passo que outros, como o Thonny, priorizam o suporte de uma única linguagem.

Ao contrário do Scratch, que fornece blocos de construção visuais como base para o seu programa, o Python é uma linguagem de programação mais tradicional onde tudo é escrito. Comece o seu primeiro programa clicando na área Shell do Python na parte inferior da janela do Thonny, e depois introduza a seguinte instrução antes de premir a tecla **ENTER**:

```
print("Olá, Mundo!")
```

Quando premir **ENTER**, o seu programa começará a ser executado de imediato: O Python responderá na mesma área Shell com a mensagem "Olá, Mundo!" (**Figura 5-1**), conforme solicitado. Isso porque a Shell é uma linha direta para o *intérprete* do Python, cujo trabalho é ver as suas instruções e *interpretar* o que querem dizer. A isto se chama *modo interativo*. Pode compará-lo a uma conversa cara a cara com alguém: assim que uma pessoa termina o que está a dizer, a outra pessoa responde e depois aguarda a sua próxima resposta.

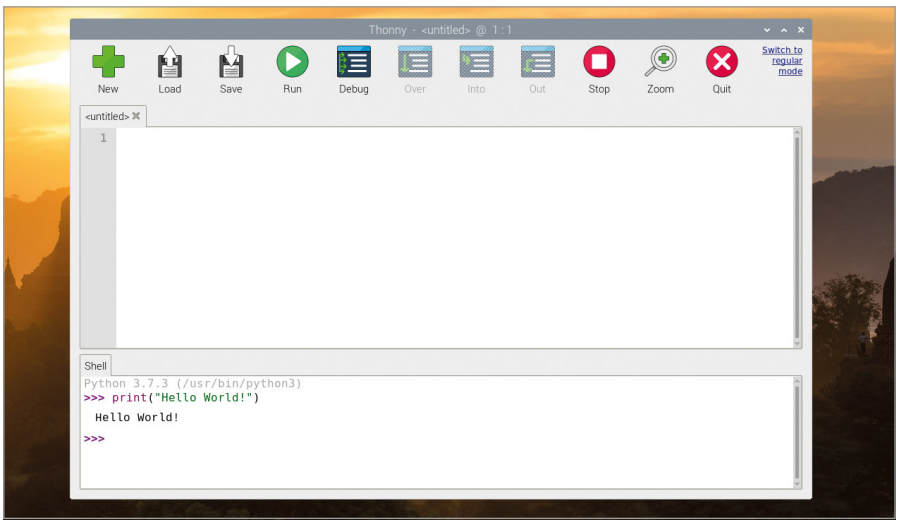
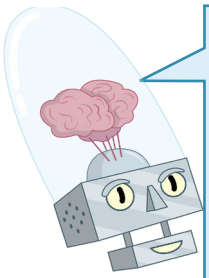


Figura 5-1: O Python imprime a mensagem "Olá, Mundo!" (Hello World!) na área Shell



ERRO DE SINTAXE

Se o seu programa não executa, mas imprime uma mensagem 'erro de sintaxe' na área da Shell, existe um erro em algum ponto do que escreveu. O Python requer que as suas instruções sejam escritas de uma forma muito específica: se falhar um parêntese ou uma aspa, escrever 'print' errado ou com um P maiúsculo, ou adicionar símbolos em algum ponto na instrução, a mesma não será executada. Tente escrever a instrução novamente e certifique-se de que corresponde à versão neste livro antes de premir a tecla **ENTER**!

No entanto, não tem de utilizar o Python no modo interativo. Clique na área de scripts no centro da janela do Thonny e introduza o seu programa novamente:

```
print("Olá, Mundo!")
```

Desta vez, quando premir a tecla **ENTER**, não acontece nada, a não ser obter uma nova linha em branco na área de scripts. Para que esta versão do seu programa funcione, terá que clicar no ícone Run (Executar) na barra de ferramentas do Thonny. Quando o fizer, será solicitado que guarde primeiro o programa. Introduza um nome descritivo, como "Olá, Mundo", e clique no botão guardar. Quando o programa estiver guardado, serão apresentadas duas mensagens na área Shell do Python (**Figura 5-2**):

```
>>> %Run 'Olá, Mundo.py'
    Olá, Mundo!
```

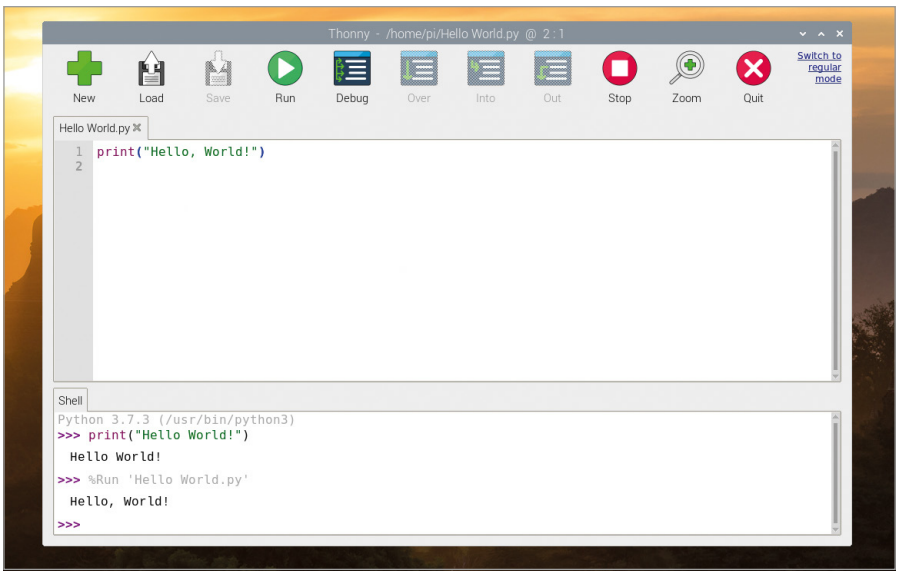


Figura 5-2: Executar o seu programa simples

A primeira destas linhas é uma instrução do Thonny para indicar ao intérprete do Python que execute o programa que acabou de guardar. A segunda é a saída do programa – a mensagem que indicou ao Python para imprimir. Parabéns: escreveu e executou o seu primeiro programa Python em ambos os modos, interativo e script!

DESAFIO: NOVA MENSAGEM

Pode alterar a mensagem que o programa Python imprime como a sua saída? Se pretendesse adicionar mais mensagens, utilizaria o modo interativo ou o modo script? O que acontece se remover os parênteses ou as aspas do programa e depois tentar executá-lo novamente?

Próximos passos: ciclos e indentação de código

Tal como o Scratch utiliza pilhas de blocos de tipo puzzle para controlar que bits do programa estão ligados a que outros bits, o Python tem a sua própria forma de controlar a sequência em que os seus programas são executados: *indentação*. Crie um novo programa clicando no ícone New (Novo) na barra de ferramentas do Thonny. Não perderá o seu programa existente, o Thonny criará um novo separador por cima da área de scripts. Comece por escrever o seguinte:

```
print("Início do ciclo!")
for i in range(10):
```

A primeira linha imprime uma mensagem simples para a Shell, tal como o seu programa Olá, Mundo. A segunda começa um ciclo *definido* que funciona da mesmo forma que no Scratch: um contador, `i`, é atribuído ao ciclo e recebe uma série de números – a instrução de `range`, com indicação para começar no número 0 e subir em direção ao número 10, sem nunca o atingir – para contar. O símbolo dois pontos (`:`) indica ao Python que a instrução seguinte deve fazer parte do ciclo.

No Scratch, as instruções a incluir no ciclo estão literalmente incluídas dentro do bloco em forma de C. O Python utiliza uma abordagem diferente: código de indentação. A linha seguinte começa com quatro espaços em branco, que o Thonny deve ter adicionado quando premiu **ENTER** após a linha 2:

```
    print("Número de ciclos", i)
```

Os espaços em branco empurram esta linha para dentro em comparação com as outras linhas. Esta indentação é a forma de o Python reconhecer a diferença entre instruções internas e externas ao ciclo; o código indentado é conhecido como sendo *aninhado*.

Constatará que, ao premir **ENTER** no final da terceira linha, o Thonny indentou automaticamente a linha seguinte, assumindo que seria parte do ciclo. Para remover, basta premir a tecla **BACKSPACE** uma vez antes de escrever a quarta linha:

```
print("Ciclo concluído!")
```

O seu programa de quatro linhas está concluído. A primeira linha fica fora do ciclo e só é executada uma vez; a segunda linha configura o ciclo; a terceira fica dentro do ciclo e é executada uma vez por cada vez que o ciclo conclui um ciclo; e a quarta linha fica fora do ciclo novamente.

```
print("Início do ciclo!")
for i in range(10):
    print("Número de ciclos", i)
print("Ciclo concluído!")
```

Clique no ícone Executar, guarde o programa como **Indentação** e procure a respetiva saída na área Shell (**Figura 5-3**):

```
Início do ciclo!
Número de ciclos 0
Número de ciclos 1
Número de ciclos 2
Número de ciclos 3
Número de ciclos 4
Número de ciclos 5
Número de ciclos 6
Número de ciclos 7
Número de ciclos 8
Número de ciclos 9
Ciclo concluído!
```

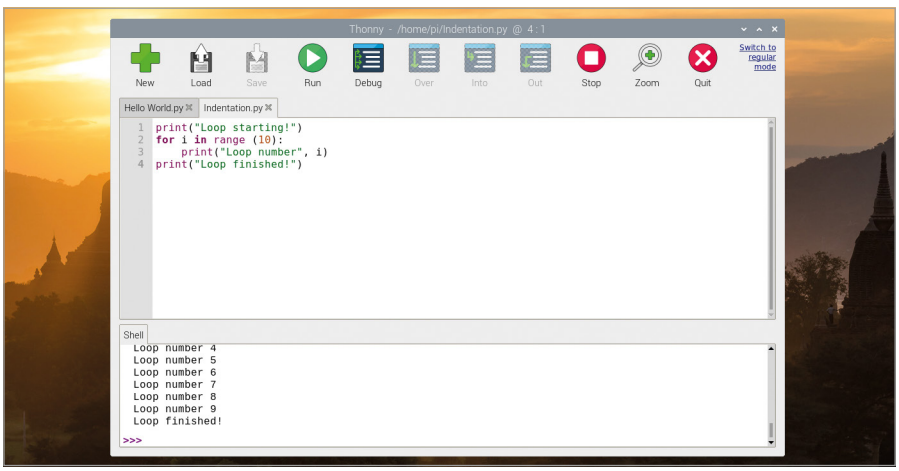
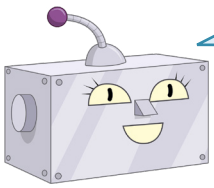


Figura 5-3: Executar um ciclo



CONTAR A PARTIR DE ZERO

O Python é uma linguagem indexada a zero – o que significa que começa a contar a partir de 0, não a partir de 1 – o motivo pelo qual o seu programa imprime os números de 0 a 9 em vez de 1 a 10. Se quiser, pode alterar esse comportamento mudando a instrução `range(10)` para `range(1, 11)`, ou qualquer outros números à sua escolha.

A indentação é uma parte poderosa do Python e um dos motivos mais comuns para que um programa não funcione conforme o previsto. Ao procurar problemas num programa, um processo conhecido como *depuração*, verifique sempre a indentação – especialmente quando começar a aninhar ciclos dentro de ciclos.

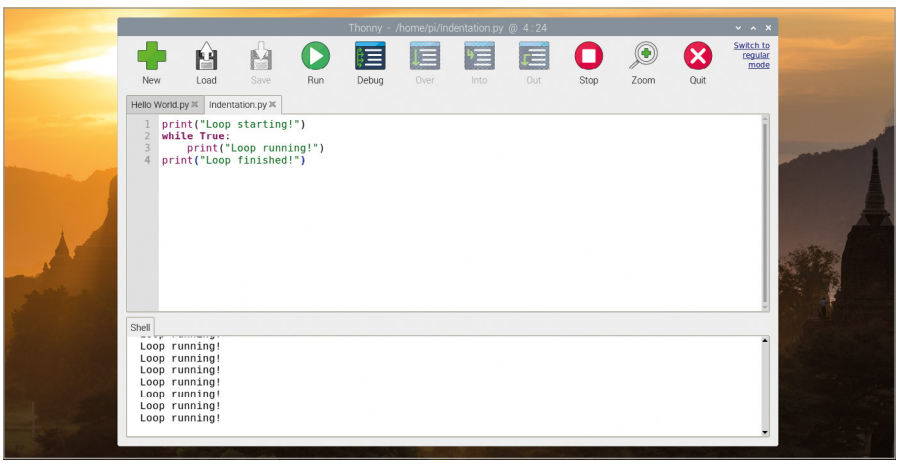
O Python também suporta ciclos *infinitos* que são executados sem fim. Para mudar o seu programa de um ciclo definido para um ciclo infinito, edite a linha 2 para ler:

```
while True:
```

Se clicar agora no ícone Executar, obterá um erro: **`name 'i' is not defined`**. Este erro ocorre porque eliminou a linha que criou e atribuiu um valor à variável **`i`**. Para o corrigir, basta editar a linha 3 para que deixe de utilizar a variável:

```
    print("Ciclo em execução!")
```

Clique no ícone Executar e, se for rápido, verá a mensagem "Início do ciclo!" seguida de uma sequência interminável de mensagens "Início do ciclo" (**Figura 5-4**). A mensagem "Ciclo concluído!" nunca será impressa porque o ciclo não tem fim: sempre que o Python terminar a impressão da mensagem "Ciclo em execução!", ela volta ao início do ciclo e imprime novamente.



▲ **Figura 5-4:** Um ciclo infinito continua até que pare o programa

Clique no ícone Stop (Parar) na barra de ferramentas do Thonny para indicar ao programa que pare o que está a executar – conhecido como interrupção do programa. Verá uma mensagem apresentada na área Shell do Python e o programa parará – sem nunca alcançar a linha 4.

DESAFIO: CICLO NO CICLO

Pode reverter o ciclo para um ciclo definido novamente? Pode adicionar um segundo ciclo definido ao programa? Como adicionaria um ciclo dentro de um ciclo, e como esperaria que funcionasse?

Condicionais e variáveis

As variáveis, como em todas as linguagens de programação, existem para mais funções do que apenas controlar ciclos. Comece um novo programa clicando no ícone Novo no menu do Thonny e, em seguida, escreva o seguinte na área de scripts:

```
userName = input ("Como se chama? ")
```

Clique no ícone Executar, guarde o seu programa como **Teste de nome** e veja o que acontece na área da Shell: será perguntado o seu nome. Escreva o seu nome na área da Shell, seguido de **ENTER**. Uma vez que essa é a única instrução no seu programa, não acontece mais nada (**Figura 5-5**). Se efetivamente quiser fazer alguma coisa com os dados que colocou na variável, precisará de mais linhas no seu programa.

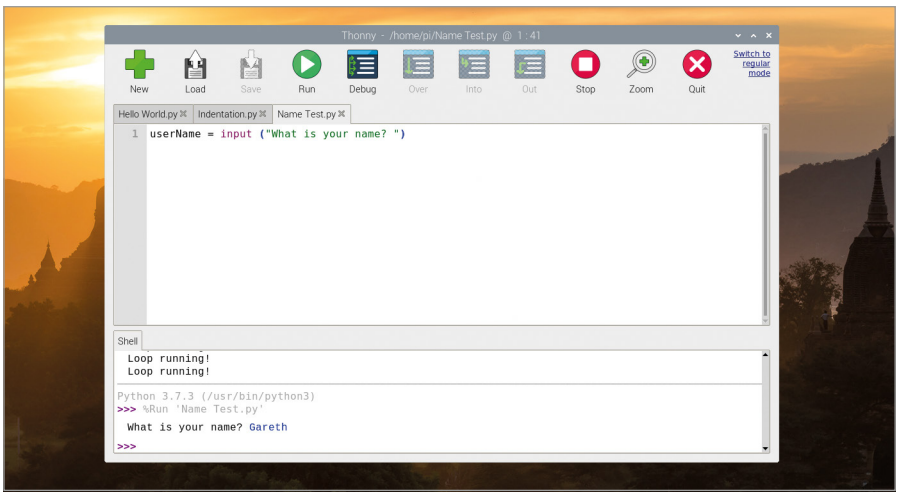


Figura 5-5: A função `input` permite-lhe solicitar introdução de texto a um utilizador

Para que o seu programa faça algo útil com o nome, adicione uma *declaração condicional* escrevendo o seguinte:

```
if userName == "Clark Kent":
    print("Você é o Super-Homem!")
else:
    print("Você não é o Super-Homem!")
```

Tenha em atenção que, quando o Thonny verifica que o seu código tem de ser indentado, executa a ação automaticamente, mas não sabe quando deve parar a indentação, pelo que terá de apagar os espaços por si mesmo.

Clique no ícone Executar e escreva o seu nome na área da Shell. A menos que o seu nome seja Clark Kent, verá a mensagem "Você não é o Super-Homem!". Clique novamente em Executar e, desta vez, escreva o nome "Clark Kent" – certificando-se de que o escreve exatamente como no programa, com um C e K maiúsculo. Desta vez, o programa reconhece que é, de facto, o Super-Homem **Figura 5-6**).

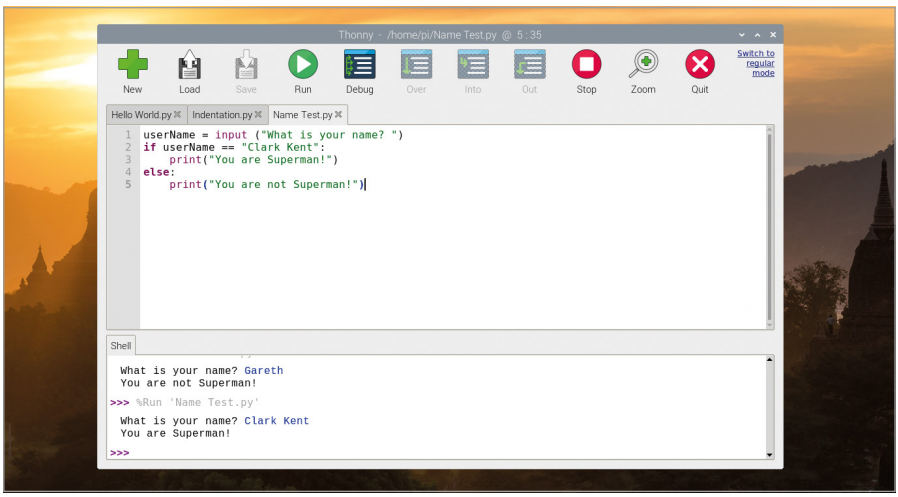


Figura 5-6: Não devia estar a salvar o mundo?

Os símbolos **==** indicam ao Python que faça uma comparação direta, procurando ver se a variável **userName** corresponde ao texto – conhecido como uma *string* – no seu programa. Se estiver a trabalhar com números, há outras comparações que pode fazer: **>** para verificar se um número é superior a outro número, **<** para verificar se é inferior a, **=>** para verificar se é igual ou superior a, **=<** para verificar se é igual ou inferior a. Há também **!=**, que significa que não é igual a – é exatamente o oposto de **==**. Estes símbolos são tecnicamente conhecidos como *operadores de comparação*.

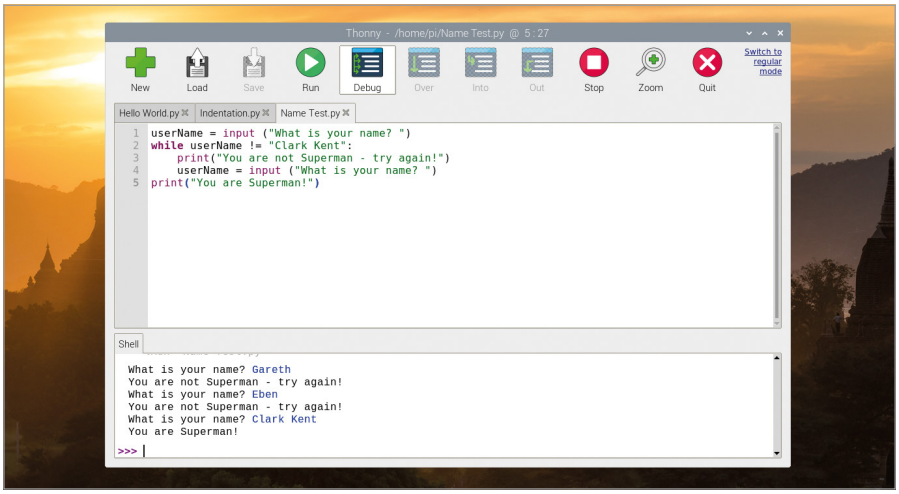
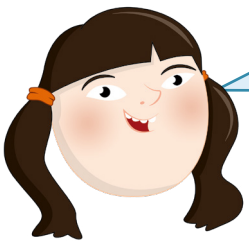


Figura 5-7: Continuará a perguntar o seu nome até responder que é "Clark Kent"



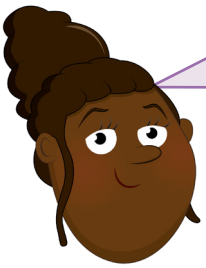
UTILIZAR = AND ==

A chave para utilizar as variáveis é aprender a diferença entre = e ==. Lembre-se: =significa "tornar esta variável igual a este valor", enquanto == significa "verificar se a variável é igual a este valor". Se os confundir, é provável que resulte num programa que não funciona!

Os operadores de comparação também podem ser utilizados em ciclos. Elimine as linhas de 2 a 5 e escreva o seguinte em substituição:

```
while userName != "Clark Kent":
    print("Você não é o Super-Homem - tente novamente!")
    userName = input ("Como se chama? ")
print("Você é o Super-Homem!")
```

Clique novamente no ícone Executar. Desta vez, em vez de desistir, o programa vai continuar a pedir o seu nome até confirmar que você é o Super-Homem (**Figura 5-7**) – como uma espécie de palavra-passe muito simples. Para sair do ciclo, escreva "Clark Kent" ou clique no ícone Parar na barra de ferramentas do Thonny. Parabéns: agora sabe como utilizar condicionantes e operadores de comparação!



DESAFIO: ADICIONAR MAIS PERGUNTAS

Pode mudar o programa para fazer mais de uma pergunta, armazenando as respostas em diversas variáveis? Pode criar um programa que utilize condicionais e operadores de comparação para imprimir se um número introduzido pelo utilizador for maior ou menor que 5, como o programa que criou em **Capítulo 4, Programar com o Scratch**?

Projeto 1: Flocos de neve de tartaruga

Agora que compreende como funciona o Python, chegou a hora de brincar com gráficos e criar um floco de neve utilizando uma ferramenta conhecida como *tartaruga*.

PROJETO ONLINE

Este projeto também está disponível online em **rpf.io/turtle-snowflakes**

Originalmente robôs físicos com a forma dos seus nomes de animais, as tartarugas são desenhadas para se moverem em linha reta, virarem, e levantarem e baixarem uma caneta – na versão digital, significa simplesmente começar ou parar de desenhar uma linha à medida que se move. Ao contrário de algumas outras linguagens, nomeadamente Logo e as suas muitas variantes, o Python não tem uma ferramenta de tartaruga embutida – mas é fornecido com uma *biblioteca* de código adicional para proporcionar o poder da tartaruga. As bibliotecas são pacotes de código que adicionam novas instruções para expandir as capacidades do Python, e são importadas para os seus próprios programas utilizando um comando importar.

Crie um novo programa clicando no ícone Novo e escreva o seguinte:

```
import turtle
```

Quando utilizar as instruções incluídas numa biblioteca, tem que usar o nome da biblioteca seguido de um ponto final e, depois, o nome da instrução. Pode ser irritante ter de escrever sempre isto, como tal, pode atribuir um nome de variável mais curto – pode ser apenas uma letra, mas achámos que seria interessante o nome funcionar como nome de animal de estimação da tartaruga. Escreva o seguinte:

```
pat = turtle.Turtle()
```

Para testar o seu programa, terá de atribuir uma tarefa à sua tartaruga. Escreva:

```
pat.forward(100)
```

Clique no ícone Executar e guarde o seu programa como **Flocos de neve de tartaruga**. Quando o programa estiver guardado, será apresentada uma nova janela chamada "Gráficos de tartaruga" e verá o resultado do seu programa: a sua tartaruga, Pat, avançará 100 unidades, desenhando uma linha reta (**Figura 5-8**).

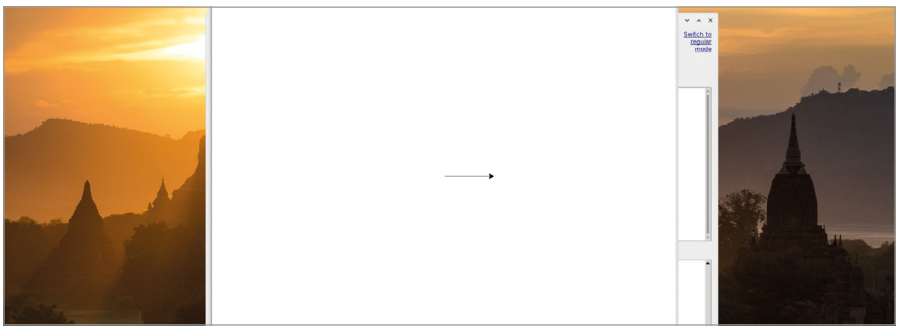


Figura 5-8: A tartaruga avança para traçar uma linha reta

Volte para a janela principal do Thonny – se estiver oculta atrás da janela Gráficos de tartaruga, clique no botão minimizar na janela Gráficos de tartaruga ou clique na entrada Thonny na barra de tarefas na parte superior do ecrã, e clique no botão Parar para fechar a janela Gráficos de tartaruga.

Ter de introduzir cada instrução de movimento à mão seria fastidioso, por isso, elimine a linha 3 e crie um ciclo para fazer o trabalho árduo de criar formas:

```
for i in range(2):
    pat.forward(100)
    pat.right(60)
    pat.forward(100)
    pat.right(120)
```

Execute o seu programa, e o Pat desenhará um único paralelogramo (**Figura 5-9**).

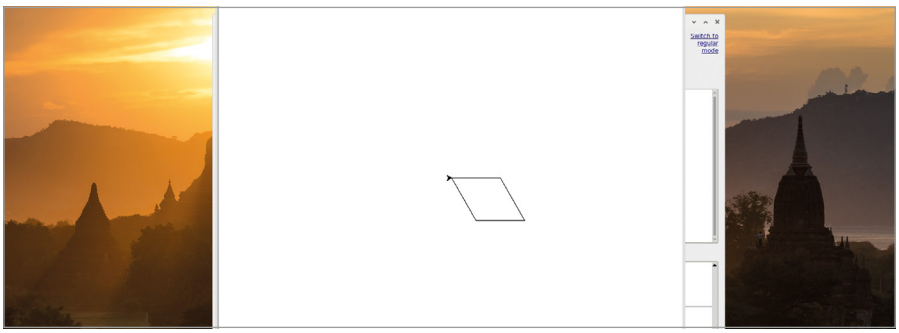


Figura 5-9: Pode desenhar formas combinando voltas e movimentos

Para transformar isso numa forma de floco de neve, clique no ícone Parar na janela principal do Thonny e crie um ciclo em torno do seu ciclo adicionando a seguinte linha como linha 3:

```
for i in range(10):
```

...e o seguinte na parte inferior do seu programa:

```
    pat.right(36)
```

O seu programa não funcionará tal como se encontra porque o ciclo existente não está indentado corretamente. Para corrigir esse problema, clique no início de cada linha do ciclo existente – linhas 4 a 8 – e prima a tecla **SPACE** quatro vezes para corrigir a indentação. O seu programa deve ter o seguinte aspeto agora:

```
import turtle
pat = turtle.Turtle()
for i in range(10):
    for i in range(2):
        pat.forward(100)
        pat.right(60)
        pat.forward(100)
        pat.right(120)
    pat.right(36)
```

Clique no ícone Executar e observe a tartaruga: vai desenhar um paralelogramo como antes, mas, quando terminar, vai rodar 36 graus e desenhar outro, e mais outro, e assim por diante, até haver dez paralelogramos sobrepostos no ecrã, assemelhando-se um pouco a um floco de neve (**Figura 5-10**).

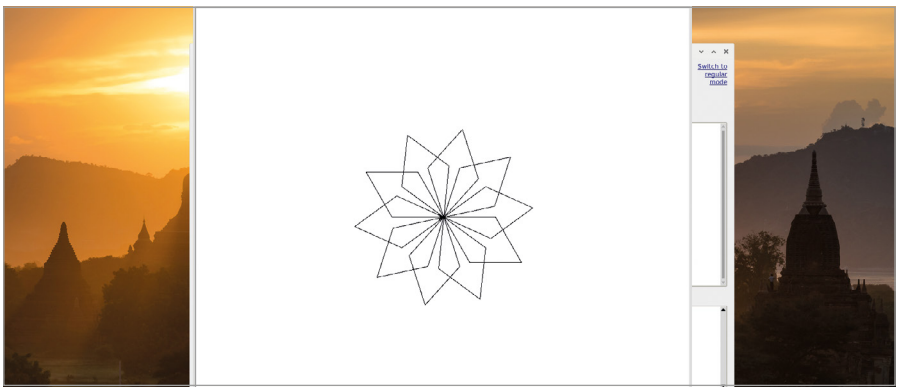


Figura 5-10: Repetir a forma para criar uma mais complexa

Uma tartaruga robótica desenha numa única cor num grande pedaço de papel, mas a tartaruga simulada do Python pode utilizar uma gama de cores. Adicione uma nova linha 3 e 4, empurrando aslinhas existentes para baixo:

```
turtle.Screen().bgcolor("blue")
pat.color("cyan")
```

Execute o seu programa novamente para ver o efeito do seu novo código: a cor de fundo da janela Gráficos de tartaruga mudou para azul e o floco de neve passou a ser ciano (**Figura 5-11**).

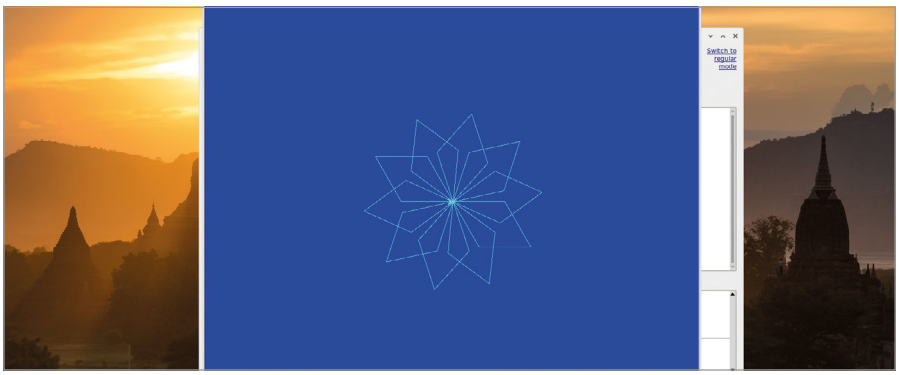


Figura 5-11: Mudar as cores do fundo e dos flocos de neve

Também pode atribuir a seleção das cores aleatoriamente a partir de uma lista utilizando a biblioteca **`random`**. Volte à parte superior do seu programa e introduza o seguinte como linha 2:

```
import random
```

Mude a cor de fundo no que é agora a linha 4 de "azul" para "cinza" e crie uma nova variável chamada "cores" inserindo uma nova linha 5:

```
colours = ["cyan", "purple", "white", "blue"]
```

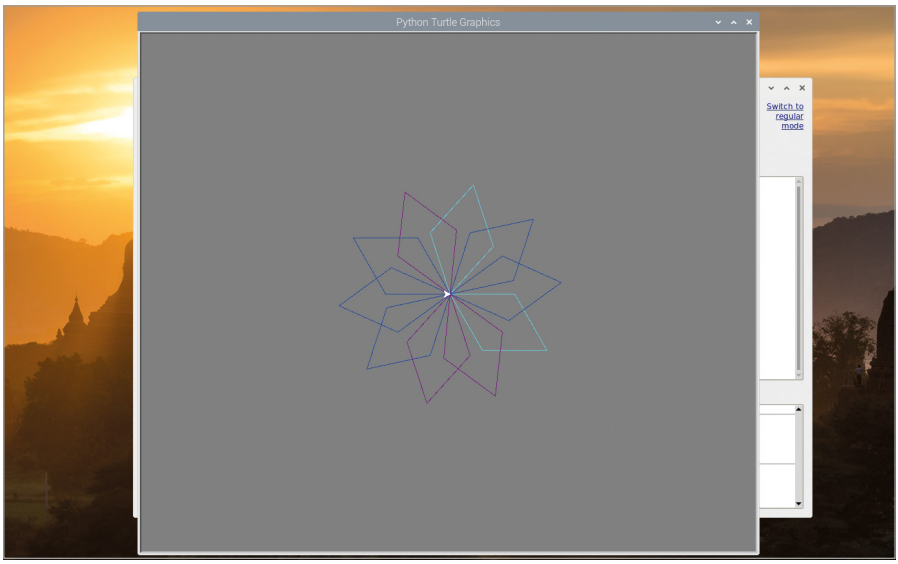
GRAFIAS DOS EUA

Muitas linguagens de programação utilizam grafias em inglês americano e o Python não é exceção: o comando para mudar a cor da caneta da tartaruga é *color*, pelo que se escrever em inglês britânico como *colour*, simplesmente não funciona. Contudo, as variáveis podem ter qualquer grafia que pretenda. É por isso que pode chamar à sua nova variável *colours* e o Python compreender.

Este tipo de variável é conhecido como uma lista e é marcado por parênteses retos. Neste caso, a lista está preenchida com as cores possíveis para os segmentos dos flocos de neve, mas é na mesma necessário indicar ao Python que escolha uma cor sempre que o ciclo se repete. No fim do programa, introduza o seguinte, assegurando que está indentado com quatro espaços para fazer parte do ciclo externo, exatamente como a linha acima:

```
    pat.color(random.choice(colours))
```

Clique no ícone Executar e snowflake-stroke-ninja-star será desenhado novamente. Desta vez, porém, o Python escolherá uma cor aleatória da sua lista conforme desenha cada pétala, dando ao floco de neve um acabamento agradável e multicolorido (**Figura 5-12**).



▲ **Figura 5-12:** Utilizar cores aleatórias para as "pétalas"

Para que o floco de neve pareça menos uma estrela ninja e mais um floco de neve real, acrescente uma nova linha 6 abaixo da lista de **cores** e introduza o seguinte:

```
pat.penup()
pat.forward(90)
pat.left(45)
pat.pendown()
```

Se estivesse a utilizar um robô tartaruga, as instruções **penup** e **pendown** permitiriam pôr e tirar a caneta física do papel, mas no mundo virtual, basta dizer à tartaruga que pare e comece a desenhar linhas. No entanto, em vez de utilizar um ciclo, desta vez vai criar uma

função – um segmento de código que pode chamar em qualquer momento, como criar a sua própria instrução do Python.

Comece por eliminar o código para desenhar os seus flocos de neve com base no paralelogramo: abrange tudo entre e incluindo a instrução **`pat.color("cyan")`** na linha 10 e **`pat.right(36)`** na linha 17. Deixe a instrução **`pat.color(random.choice(colours))`** mas acrescente um símbolo hash (**#**) no início da linha. Isto é conhecido como *comentar* uma instrução e significa que o Python vai ignorá-la. Pode utilizar comentários para adicionar explicações ao seu código, para ser mais fácil de entender quando voltar a trabalhar com ele passados alguns meses ou se o enviar a outra pessoa!

Crie a sua função, que será chamada de "branch", introduzindo a seguinte instrução na linha 10, por baixo de **`pat.pendown():`**

```
def branch():
```

Isto *define* a sua função, **`branch`**. Quando premir a tecla **ENTER**, o Thonny adicionará automaticamente indentação para as instruções da função. Introduza o seguinte, prestando muita atenção à indentação porque a dado momento vai aninhar código com três níveis de indentação!

```
    for i in range(3):
        for i in range(3):
            pat.forward(30)
            pat.backward(30)
            pat.right(45)
        pat.left(90)
        pat.backward(30)
        pat.left(45)
    pat.right(90)
    pat.forward(90)
```

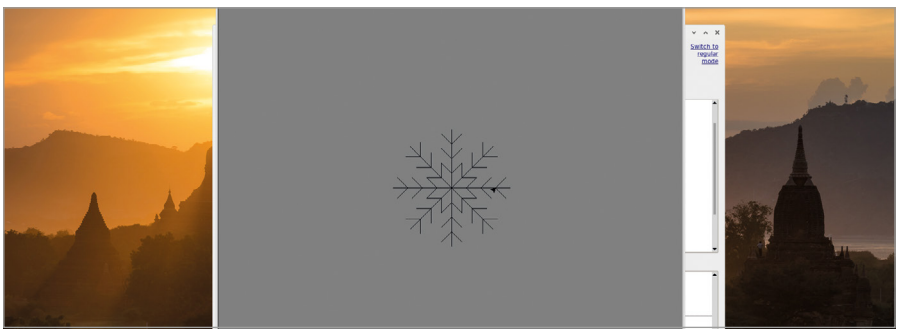
Por fim, crie um novo ciclo na parte inferior do seu programa, mas por cima da linha de cor comentada, para executar ou *invocar* a sua nova função:

```
for i in range(8):
    branch()
    pat.left(45)
```

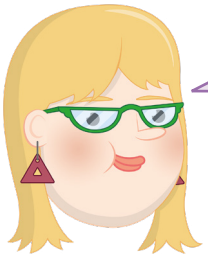
O seu programa final deve ter o seguinte aspeto:

```
import turtle
import random
pat = turtle.Turtle()
turtle.Screen().bgcolor("grey")
colours = ["cyan", "purple", "white", "blue"]
pat.penup()
pat.forward(90)
pat.left(45)
pat.pendown()
def branch():
    for i in range(3):
        for i in range(3):
            pat.forward(30)
            pat.backward(30)
            pat.right(45)
        pat.left(90)
        pat.backward(30)
        pat.left(45)
    pat.right(90)
    pat.forward(90)
for i in range(8):
    branch()
    pat.left(45)
#    pat.color(random.choice(colours))
```

Clique em Executar e observe a janela gráfica à medida que o Pat desenha, seguindo as suas instruções. Parabéns: o seu floco de neve agora assemelha-se muito mais a um floco de neve (**Figura 5-13**)!



▲ **Figura 5-13:** Os ramos adicionais fazem com que pareça um floco de neve



DESAFIO: O QUE SE SEGUE?

Pode utilizar as suas instruções comentadas para que os ramos do floco de neve sejam desenhados em cores diferentes? Pode criar uma função "floco de neve" e utilizá-la para desenhar muitos flocos de neve no ecrã? Pode fazer o seu programa mudar o tamanho e a cor dos flocos de neve aleatoriamente?

Projeto 2: Encontrar as diferenças assustador

O Python também pode processar imagens e sons, bem como gráficos com base em tartarugas, que podem ser utilizados com um efeito espetacular como uma brincadeira para os seus amigos – um jogo de encontrar as diferenças com um segredo assustador no seu coração, perfeito para o Halloween!

PROJETO ONLINE

Este projeto também está disponível online em **rpf.io/scary-spot**

Para este projeto precisa de duas imagens – a sua imagem para encontrar as diferenças mais uma imagem surpresa "assustadora" – e um ficheiro de som. Clique no ícone da framboesa para carregar o menu do Raspberry Pi OS, escolha a categoria Internet e clique no navegador Web Chromium. Quando tiver carregado, escreva **rpf.io/spot-pic** na barra de endereço, seguido da tecla **ENTER**. Clique com o botão direito do rato na imagem e clique em "Guardar imagem como...", selecione a pasta Início na lista do lado esquerdo e clique em Guardar. Clique novamente na barra de endereço do Chromium e introduza **rpf.io/scary-pic** seguido da tecla **ENTER**. Como antes, clique com o botão direito do rato na imagem, clique em "Guardar imagem como...", selecione a pasta Início e clique em Guardar.

Para o ficheiro de som de que necessita, clique novamente na barra de endereço e introduza **rpf.io/scream** seguido da tecla **ENTER**. Este ficheiro, o som de um grito para dar ao seu jogador um verdadeiro susto, será reproduzido automaticamente, mas tem de guardá-lo para poder utilizá-lo. Clique com o botão direito do rato no pequeno leitor de áudio, clique em "Guardar como...", selecione a pasta Início e clique em Guardar. Agora pode fechar a janela do Chromium.

Clique no ícone Novo na barra de ferramentas do Thonny para iniciar um novo projeto. Como antes, vai utilizar uma biblioteca para expandir as capacidades do Python: a biblioteca Pygame, que tal o nome sugere, foi criada com os jogos em mente. Escreva o seguinte:

```
import pygame
```

Vai precisar de algumas partes de outras bibliotecas e também de uma subsecção da biblioteca Pygame. Importe-as introduzindo o seguinte:

```
from pygame.locals import *
from time import sleep
from random import randrange
```

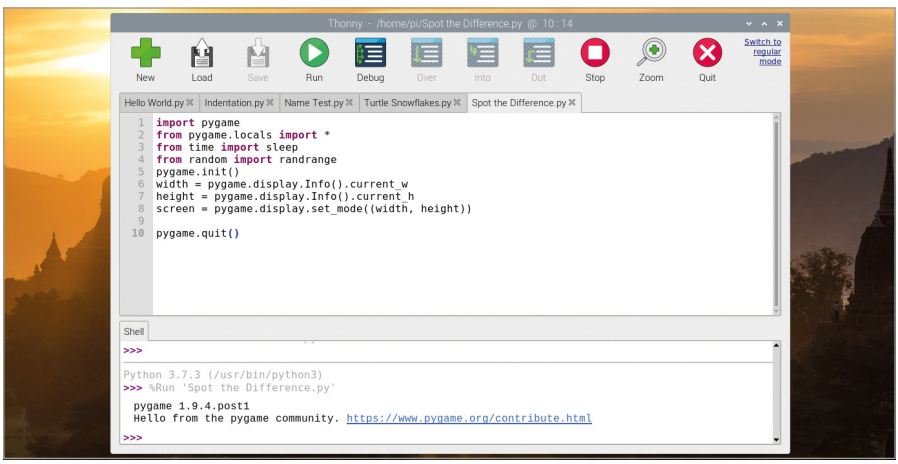
A instrução **`from`** funciona de forma diferente da instrução **`import`**, permitindo-lhe importar apenas as partes de uma biblioteca de que necessita, e não toda a biblioteca. Em seguida, tem de configurar o Pygame; isto é conhecido como *inicialização*. O Pygame precisa de saber a largura e altura do monitor ou TV do jogador, conhecidos como a *resolução*. Escreva o seguinte:

```
pygame.init()
width = pygame.display.Info().current_w
height = pygame.display.Info().current_h
```

O passo final na configuração do Pygame é criar a respetiva janela, a que o Pygame chama ecrã. Escreva o seguinte:

```
screen = pygame.display.set_mode((width, height))

pygame.quit()
```

Tenha em atenção a linha em branco no meio porque é o destino do seu programa. Por enquanto, clique no ícone Executar, guarde o seu programa como **Encontrar as diferenças** e observe: O Pygame criará uma janela, preenchendo-a com um fundo preto, que desaparecerá quase imediatamente quando chegar à instrução para sair. Além de uma pequena mensagem na Shell (**Figura 5-14**), o programa não fez muito até agora.



▲ **Figura 5-14:** O seu programa é funcional, mas ainda não faz muito

Para apresentar a sua imagem de encontrar as diferenças, escreva a seguinte linha no espaço acima de **pygame.quit()**:

```
difference = pygame.image.load('spot_the_diff.png')
```

Para garantir que a imagem preenche o ecrã, deve dimensioná-la de acordo com a resolução do seu monitor ou TV. Escreva o seguinte:

```
difference = pygame.transform.scale(difference, (width, height))
```

Agora que a imagem está na memória, tem de indicar ao Pygame para apresentá-la efetivamente no ecrã – um processo conhecido como *blitting* ou *transferência de blocos de bits*. Escreva o seguinte:

```
screen.blit(difference, (0, 0))
pygame.display.update()
```

A primeira destas linhas copia a imagem para o ecrã, começando no canto superior esquerdo; a segunda diz ao Pygame para redesenhar o ecrã. Sem esta segunda linha, a imagem ficará no local correto na memória, mas nunca será visível para si!

Clique no ícone Executar e a imagem será brevemente apresentada no ecrã (**Figura 5-15**).

Figura 5-15: A sua imagem Encontre as diferenças

Para ver a imagem por mais tempo, adicione a seguinte linha imediatamente acima de **pygame.quit()**:

```
sleep(3)
```

Clique em Executar novamente e a imagem ficará no ecrã durante mais tempo. Adicione a sua imagem surpresa introduzindo o seguinte logo abaixo da linha **pygame.display.update():**

```
zombie = pygame.image.load('scary_face.png')
zombie = pygame.transform.scale(zombie, (width, height))
```

Adicione um atraso para que a imagem do zombie não apareça de imediato:

```
sleep(3)
```

Em seguida, transfira a imagem para o ecrã e atualize para que seja mostrada ao jogador:

```
screen.blit(zombie, (0,0))
pygame.display.update()
```

Clique no ícone Executar e veja o que acontece: Pygame carregará a sua imagem de encontrar as diferenças, mas após três segundos será substituída pelo assustador zombie (**Figura 5-16**)!

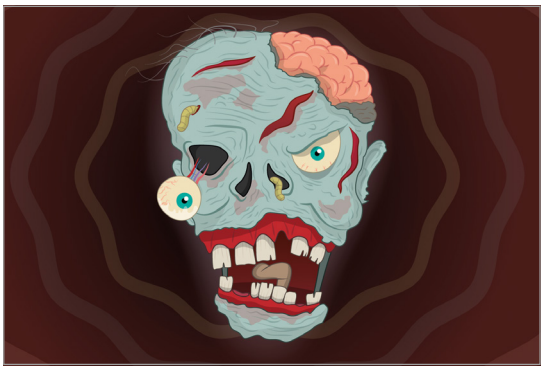


Figura 5-16: Vai pregar um valente susto a alguém

O atraso definido em três segundos, no entanto, torna as coisas um pouco previsíveis. Mude a linha **`sleep(3)`** por cima de **`screen.blit(zombie, (0,0))`** para:

```
sleep(randrange(5, 15))
```

Isto escolhe um número aleatório entre 5 e 15 e atrasa o programa durante esse tempo. Em seguida, adicione a seguinte linha acima da sua instrução **`sleep`**< para carregar o ficheiro de som do grito:

```
scream = pygame.mixer.Sound('scream.wav')
```

Mova abaixo da instrução **`sleep`** e escreva o seguinte numa nova linha para iniciar a reprodução do som, para que comece a tocar um pouco antes da imagem assustadora mostrada ao jogador:

```
scream.play()
```

Por fim, diga ao Pygame para parar de tocar o som escrevendo a seguinte linha por cima de **`pygame.quit()`**:

```
scream.stop()
```

Clique no ícone Executar e admire o seu trabalho: após alguns segundos de diversão inocente, o seu zombie assustador aparecerá ao mesmo tempo que um grito arrepiante, pregando um susto valente aos seus amigos! Se constatar que a imagem do zombie aparece antes do som começar a tocar, pode compensar adicionando um pequeno atraso imediatamente após a instrução **`scream.play()`** e antes da instrução **`screen.blit`**:

```
sleep(0.4)
```

O seu programa final deve ter o seguinte aspeto:

```
import pygame
from pygame.locals import *
from time import sleep
from random import randrange
pygame.init()
width = pygame.display.Info().current_w
height = pygame.display.Info().current_h
screen = pygame.display.set_mode((width, height))
difference = pygame.image.load('spot_the_diff.png')
difference = pygame.transform.scale(difference, (width, height))
screen.blit(difference, (0, 0))
pygame.display.update()
zombie = pygame.image.load('scary_face.png')
zombie = pygame.transform.scale (zombie, (width, height))
scream = pygame.mixer.Sound('scream.wav')
sleep(randrange(5, 15))
scream.play()
screen.blit(zombie, (0,0))
pygame.display.update()
sleep(3)
scream.stop()
pygame.quit()
```

Agora só falta convidar os seus amigos para brincar às diferenças e assegurar que as colunas estão ligadas, é claro!

DESAFIO: ALTERAR O ASPETO

Pode mudar as imagens para tornar a brincadeira mais apropriada para outros eventos, como o Natal? Pode desenhar as suas próprias imagens assustadoras e para encontrar as diferenças (com um editor gráfico como o GIMP)? Poderia rastrear o utilizador clicando numa diferença, para ser mais convincente?

Projeto 3: Labirinto RPG

Agora que está a apanhar o jeito do Python, chegou a hora de utilizar o Pygame para criar algo um pouco mais complicado: um jogo de labirinto totalmente funcional com base em texto, baseado em jogos clássicos de RPG (Role Playing Games). Conhecidos como aventuras de texto ou ficção interativa, estes jogos são do tempo quando os computadores não conseguiam processar gráficos, mas ainda têm os seus fãs, que argumentam que nenhum gráfico é tão vívido como a nossa imaginação!

PROJETO ONLINE

Este projeto também está disponível online em **rpf.io/scary-spot**

Este programa é bastante mais complexo do que os restantes neste capítulo, portanto, para facilitar as coisas, vamos começar com uma versão já parcialmente escrita. Abra o navegador Web Chromium e aceda ao seguinte endereço: **rpf.io/p/pt-PT/rpg-go**.

O navegador Web Chromium vai transferir automaticamente o código do programa para a pasta Transferências, mas avisando que o tipo de ficheiro – um programa Python – pode danificar o seu computador. Transferiu o ficheiro da Raspberry Pi Foundation, uma origem fidedigna, por isso, clique no botão Manter na mensagem de aviso apresentada na parte inferior do ecrã. Regresse ao Thonny e clique no ícone Load (Carregar) . Encontre o ficheiro **rpg.py** na sua pasta Transferências e clique no botão Carregar.

Comece por clicar no ícone Executar para se familiarizar com o funcionamento de uma aventura de texto. A saída do jogo aparece na área Shell na parte inferior da janela do Thonny. Aumente a janela do Thonny clicando no botão maximizar para facilitar a leitura.

O jogo, tal como se encontra, é muito simples: há duas divisões e não há objetos. O jogador começa na Entrada, a primeira das duas divisões. Para ir para a Cozinha, basta escrever "vai sul" seguido da tecla **ENTER** (**Figura 5-17**). Quando estiver na Cozinha, pode escrever "vai norte" para regressar à Entrada. Também pode experimentar escrever "vai oeste" e "vai este", mas como não há divisões nessas direções, o jogo mostrará uma mensagem de erro.

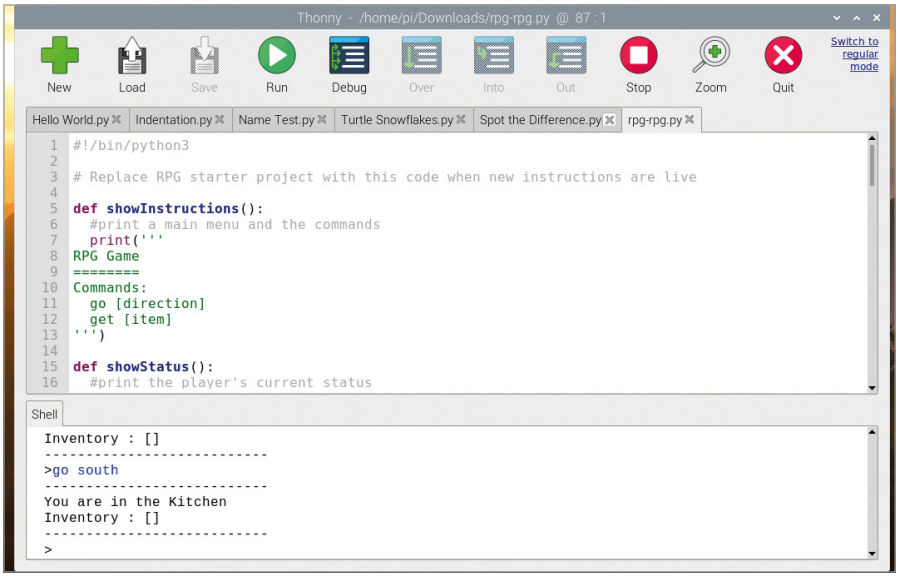


Figura 5-17: Para já apenas existem duas divisões

Desloque até a linha 29 do programa na área de scripts para encontrar uma variável chamada **`divisões`**. Este tipo de variável é conhecido como um *dicionário* e indica ao jogo as divisões, as respetivas saídas, e a que divisão uma determinada saída vai dar.

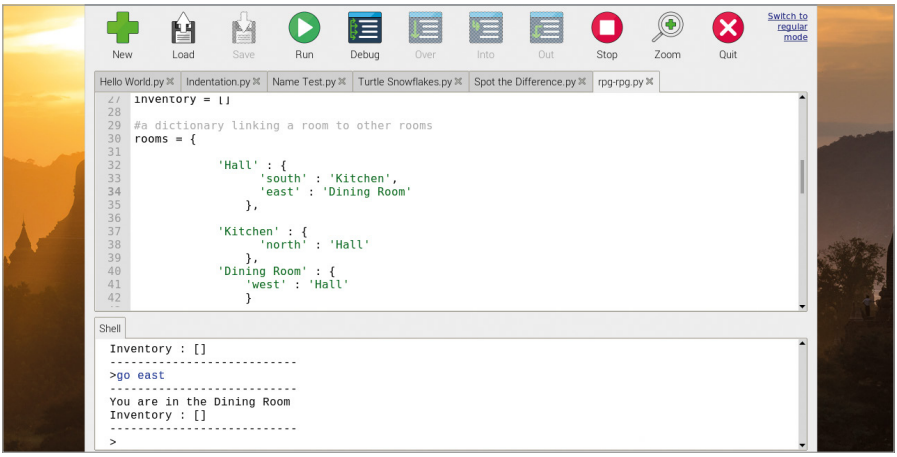
Para tornar o jogo mais interessante, acrescente outra divisão: uma Sala de jantar, a leste do Átrio. Encontre a variável **`divisões`** na área de scripts e expanda-a adicionando um símbolo de vírgula (**`,`**) após o **`}`**na linha 38, escrevendo o seguinte (a indentação exata não é essencial num dicionário):

```
'Sala de jantar' : {
      'oeste' : 'Entrada'
   }
```

Também vai precisar de uma nova saída na Entrada, já que não é criada uma automaticamente para si. Vá para o final da linha 33, adicione uma vírgula e, em seguida, adicione a seguinte linha:

```
'este' : 'Sala de jantar'
```

Clique no ícone Executar e experimente a sua nova divisão: escreva 'vai leste' quando estiver na Entrada para entrar na Sala de jantar (**Figura 5-18**), e escreva 'vai oeste' quando estiver na Sala de jantar para entrar na Entrada. Parabéns: criou a sua própria divisão!

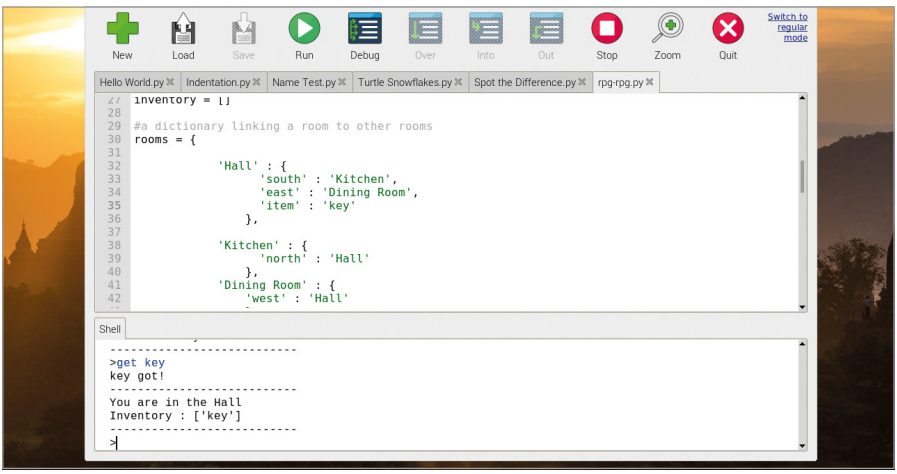


▲ **Figura 5-18:** Adicionou outra divisão

As divisões vazias não são lá muito divertidas. Para adicionar um item a uma divisão, tem de modificar o dicionário dessa divisão. Pare o programa clicando no ícone Parar. Encontre o dicionário **`Entrada`** na área de scripts e adicione uma vírgula ao final da linha **`'este' : 'Sala de jantar'`** antes de premir **ENTER** e escrever a seguinte linha:

```
'item' : 'chave'
```

Clique novamente em Executar. Desta vez, o jogo indicará que pode ver o seu novo item: uma chave. Escreva 'apanhar chave' (**Figura 5-19**), pegue nela e adicione-a à lista de itens que possui – conhecida como *inventário*. O seu inventário fica consigo enquanto se desloca de divisão em divisão.



▲ **Figura 5-19:** A chave recolhida é adicionada ao seu inventário

Clique no ícone Parar e torne o jogo mais interessante ao adicionar um monstro a evitar. Encontre o dicionário **Cozinha** e adicione um item 'monstro' da mesma forma que adicionou o item 'chave' – sem se esquecer de adicionar uma vírgula no final da linha acima:

```
'item' : 'monstro'
```

Para que o monstro seja capaz de atacar o jogador, é necessário adicionar alguma lógica ao jogo. Desloque até ao fundo do programa na área de scripts e adicione as seguintes linhas – incluindo o comentário, marcado com um símbolo hash, que o ajudará a entender o programa se regressar noutro dia – e assegure que indenta as linhas:

```
    # o jogador perde se entrar numa divisão com um monstro
    if 'item' in divisões[divAtual] and "monstro" in
divisões[divAtual]['item']:
        print('Um monstro apanhou-o... FIM DO JOGO!')
        break
```

Clique em Executar e tente ir para a divisão Cozinha (**Figura 5-20**) – o monstro não vai ficar muito impressionado!

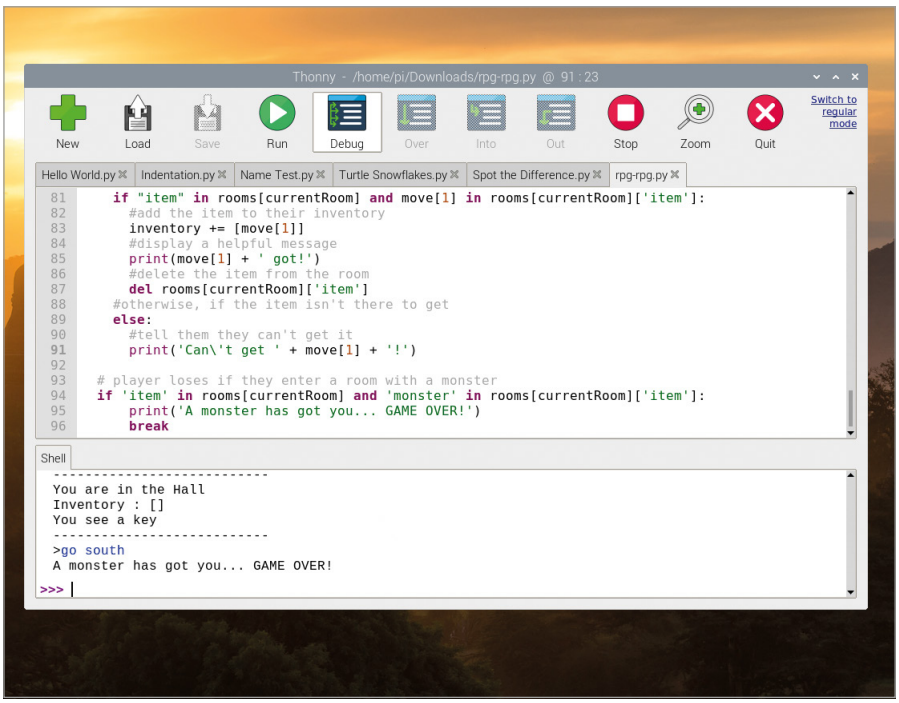


Figura 5-20: Deixe lá os ratos, há um monstro na cozinha

Para transformar esta aventura num verdadeiro jogo, vai precisar de mais itens, mais uma divisão e a capacidade de "vencer" ao deixar a casa com todos os itens em segurança no seu inventário. Comece por adicionar outra divisão, como fez para a Sala de jantar, só que desta vez trata-se de um Jardim. Adicione uma saída do dicionário Sala de jantar e não se esqueça de adicionar uma vírgula ao final da linha acima:

```
'sul' : 'Jardim'
```

Em seguida, adicione a sua nova divisão ao dicionário **`divisões`** principal, novamente sem se esquecer de adicionar uma vírgula após **`}`** na linha acima, como anteriormente:

```
'Jardim' : {
      'norte' : 'Sala de jantar'
    }
```

Adicione um objeto "poção" ao dicionário Sala de jantar, novamente sem se esquecer de adicionar a vírgula necessária à linha acima:

```
'item' : 'poção'
```

Por fim, desloque até à parte inferior do programa e adicione a lógica necessária para verificar se o jogador tem todos os itens e, em caso afirmativo, diga-lhes que ganharam o jogo:

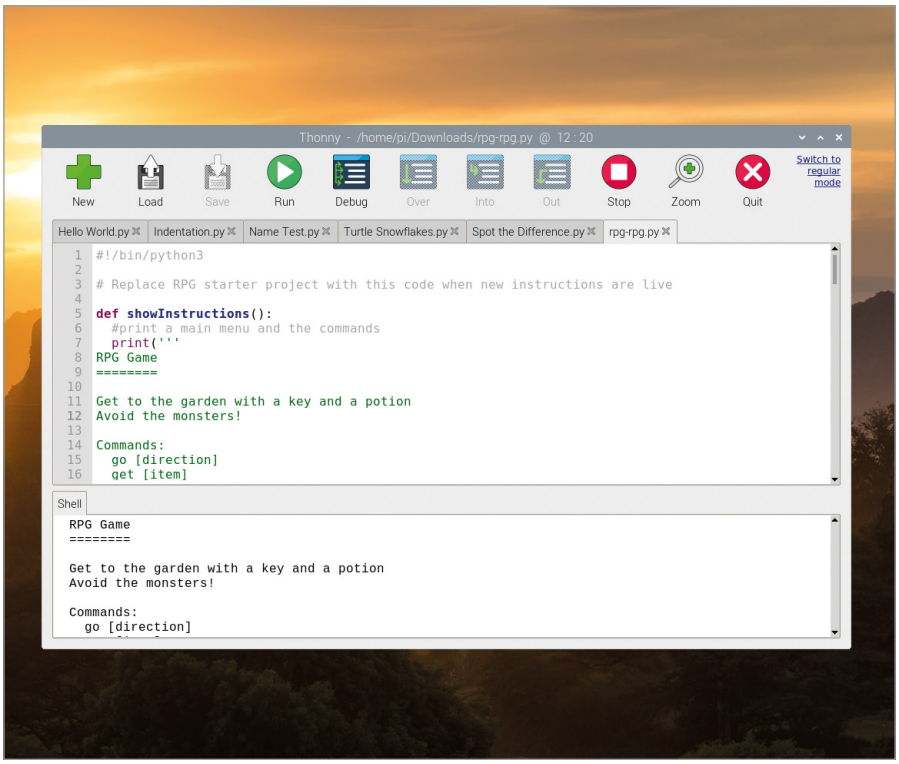
```
  # o jogador vence se chegar ao jardim com uma chave e uma poção
      if divAtual == 'Jardim' and 'chave' in inventario and 'poção'
in inventario:
          print('Escapou da casa... VENCEU!')
          break
```

Clique em Executar e tente terminar o jogo recolhendo a chave e a poção antes de ir para o jardim. Lembre-se, não entre na Cozinha porque é lá que está o monstro!

Como último ajuste para o jogo, adicione algumas instruções a indicar ao jogador como concluir o jogo. Desloque até à parte superior do programa, onde a função **`mostraInstruções()`** está definida, e acrescente o seguinte:

```
Chegue ao jardim com uma chave e uma poção
Evite os monstros!
```

Execute o jogo uma última vez e verá as suas novas instruções apresentadas logo no início (**Figura 5-21**). Parabéns: criou um jogo de labirinto interativo com base em texto!



▲ **Figura 5-21:** Agora o jogador sabe o que tem de fazer

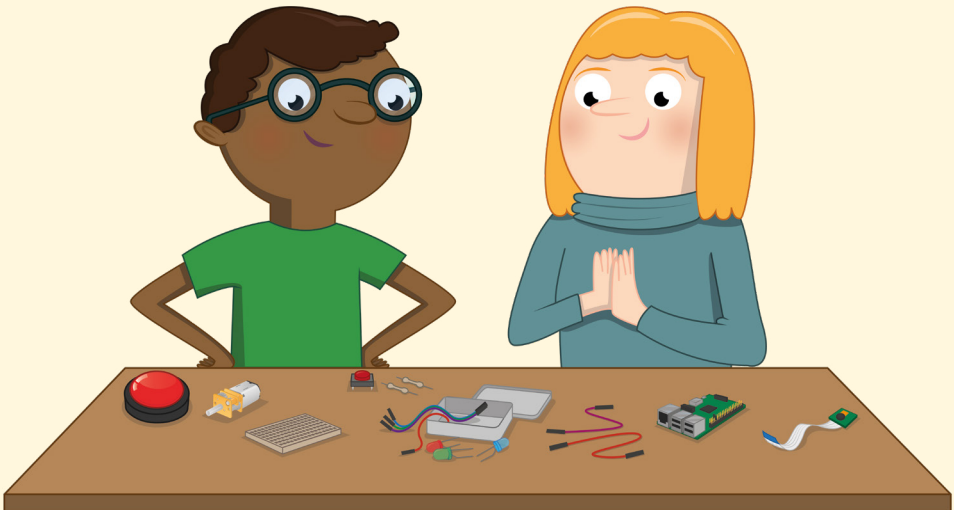
DESAFIO: EXPANDIR O JOGO

Pode adicionar mais divisões para que o jogo demore mais tempo? Pode adicionar um item para protegê-lo do monster? Como adicionaria uma arma para matar o monster? Pode adicionar divisões por cima e por baixo das divisões existentes, com acesso por escadas?

Capítulo 6

Computação física com Scratch e Python

Programar implica muito mais que apenas fazer coisas no ecrã. Também pode controlar componentes eletrónicos ligados aos pinos GPIO do Raspberry Pi



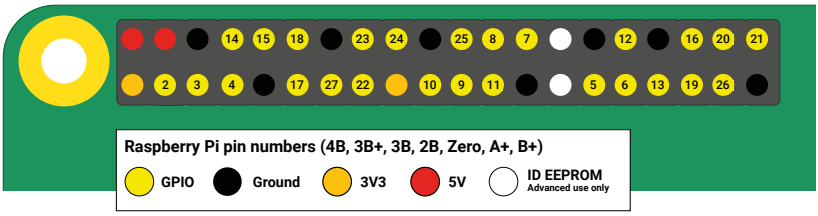
Quando as pessoas pensam em "programação" ou "codificação", geralmente, e naturalmente, pensam em software. No entanto, a programação pode ser mais do que apenas software: pode afetar o mundo real através do hardware. Isto é conhecido como *computação física*.

Como o nome sugere, a computação física pretende controlar coisas no mundo real com os seus programas: hardware, ao invés de software. Quando define o programa na máquina de lavar roupa, altera a temperatura no termóstato programável ou prime um botão nos semáforos para atravessar a estrada em segurança, está a utilizar a computação física.

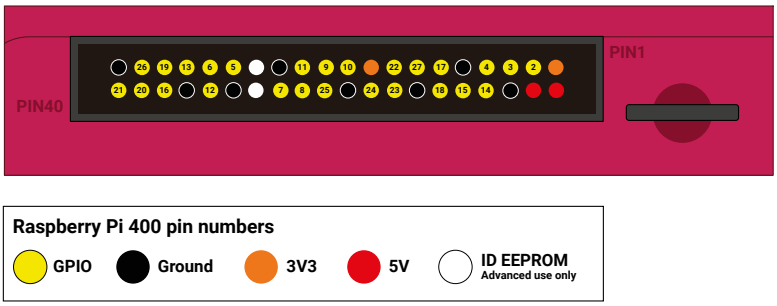
O Raspberry Pi é um ótimo dispositivo para aprender sobre computação física graças a uma funcionalidade chave: o conetor *entrada/saída para fins gerais*(*GPIO*).

Apresentamos o conetor GPIO

Situado na margem superior da placa de circuito do Raspberry Pi, ou na traseira do Raspberry Pi 400, assemelhando-se a duas longas filas de pinos de metal, o conetor GPIO (general-purpose input/output) é uma forma de ligar hardware, como LEDs e interruptores, ao Raspberry Pi para controlar nos programas criados por si. Os pinos podem ser utilizados para entrada e saída.



O conetor GPIO do Raspberry Pi é composto de 40 pinos macho. Alguns pinos estão disponíveis para utilizar nos seus projetos de computação física, alguns pinos fornecem energia, enquanto outros pinos são reservados para comunicação com hardware adicional como o Sense HAT (consulte o **Capítulo 7**).



O Raspberry Pi 400 tem o mesmo conetor GPIO com os mesmos pinos, mas está virado de cabeça para baixo em comparação com outros modelos Raspberry Pi. Este diagrama assume que está a observar o conetor GPIO desde a traseira do Raspberry Pi 400. Verifique sempre os cabos ao fazer qualquer ligação no GPIO do Raspberry Pi 400 – é fácil esquecer, apesar das etiquetas "Pino 40" e "Pino 1" na caixa!

EXTENSÕES DE GPIO

É perfeitamente possível utilizar o conetor GPIO do Raspberry Pi 400 tal como se encontra, mas pode achar mais fácil utilizar uma extensão. Com uma extensão, os pinos encontram-se na lateral do Raspberry Pi 400, o que significa que pode verificar e ajustar os fios sem necessidade de contornar pela traseira.

As extensões compatíveis incluem a gama Black HAT Hack3r de **pimoroni.com** e o Pi T-Cobbler Plus de **adafruit.com**.

Se comprar uma extensão, verifique sempre as ligações – algumas, como o Pi T-Cobbler Plus, mudam o esquema dos pinos GPIO. Quando em dúvida, utilize sempre as instruções do fabricante.

Existem várias categorias de tipos de pinos, cada uma com uma função particular:

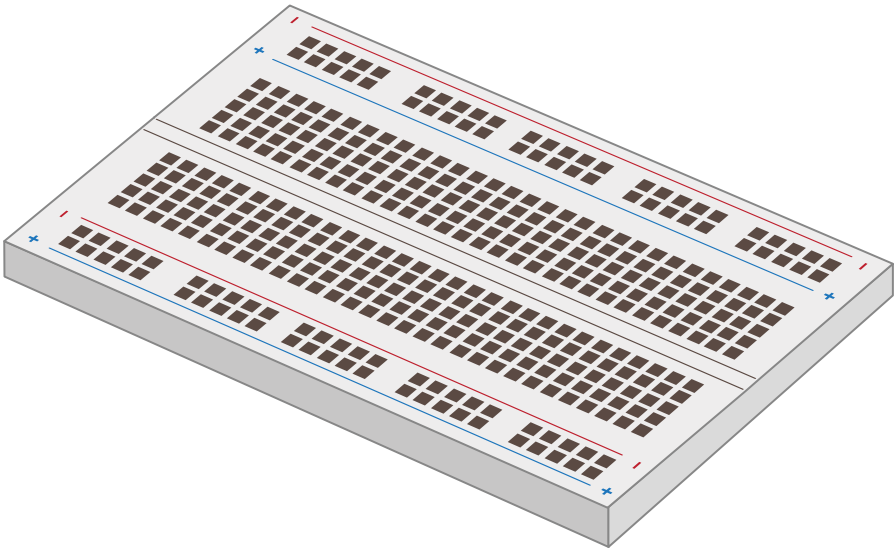
3V3	**3,3 volts de potência**	Uma fonte de alimentação de 3,3 V constantemente ligada, a mesma voltagem do Raspberry Pi a nível interno
5 V	**5 volts de potência**	Uma fonte de alimentação de 5 V constantemente ligada, a mesma voltagem do Raspberry Pi no conetor de alimentação micro USB
Terra (GND)	**0 volts de potência**	Uma ligação à terra, utilizada para completar um circuito ligado à fonte de alimentação
GPIO XX	**Número de pino de entrada/saída de uso geral "XX"**	Os pinos GPIO disponíveis para os seus programas, identificados por um número de 2 a 27
ID EEPROM	**Pinos reservados para fins especiais**	Pinos reservados para utilização com Hardware Attached on Top (HAT) e outros acessórios

AVISO!

O conetor GPIO Raspberry Pi é uma forma divertida e segura de experimentar com computação física, mas deve ser tratado com cuidado. Tenha cuidado para não dobrar os pinos ao ligar e desligar o hardware. Nunca ligue dois pinos diretamente juntos, acidentalmente ou deliberadamente, a menos que seja expressamente indicado nas instruções de um projeto: isso é conhecido como um curto-circuito e, dependendo dos pinos, pode danificar permanentemente o seu Raspberry Pi.

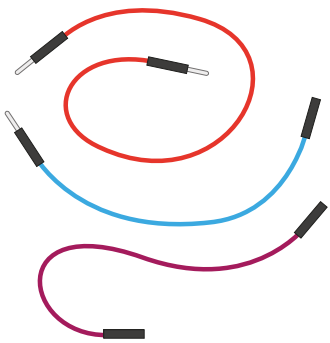
Componentes eletrónicos

O conetor GPIO é apenas uma parte do que necessita para começar a trabalhar com computação física; a outra metade é composta por componentes elétricos, os dispositivos que controlará a partir do conetor GPIO. Há milhares de componentes diferentes disponíveis, mas a maioria dos projetos GPIO são criados com as seguintes partes comuns.

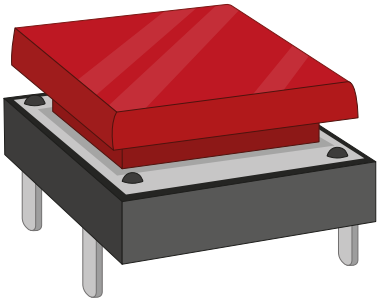


Uma *placa universal*, também conhecida como *placa universal sem solda*, pode facilitar imenso os projetos de computação física. Ao invés de ter diversos componentes separados que têm de ser ligados com fios, uma placa universal permite inserir componentes e ligá-los através de faixas metálicas ocultadas sob a sua superfície. Muitas placas universais também incluem secções para distribuição de alimentação, facilitando a construção dos seus circuitos. Não precisa de uma placa universal para começar com a computação física, mas certamente ajuda.

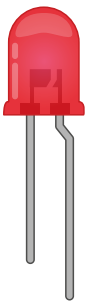
Os *Cabos jumper*, também conhecidos como *fios jumper*, ligam os componentes ao Raspberry Pi e, se não utilizar uma placa universal, uns aos outros. Estão disponíveis em três versões: macho para fêmea (M2F), necessários para ligar uma placa universal aos pinos GPIO; fêmea para fêmea (F2F), que podem ser utilizados para ligar componentes individuais se não utilizar uma placa universal; e macho para macho (M2M), utilizados para fazer ligações de uma parte da placa universal a outra. Consoante o seu projeto, pode precisar dos três tipos de cabos jumper; se utilizar uma placa universal, geralmente pode utilizar apenas cabos M2F e M2M.



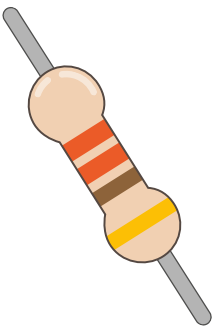
Um *interruptor de pressão*, também conhecido como *interruptor momentâneo*, é o tipo de interruptor utilizado para controlar uma consola de jogos. Normalmente disponível com dois ou quatro pés – ambos os tipos funcionam com o Raspberry Pi – o botão é um dispositivo de entrada: pode dizer ao seu programa para estar atento ao premir do botão e executar uma tarefa. Outro tipo de interruptor comum é o *interruptor com bloqueio*. Ao passo que um botão de pressão apenas está ativo enquanto é premido, um interruptor com bloqueio – semelhante aos interruptores de luz – é ativado quando o prime e mantém-se nesse estado até premir para mudar a posição.



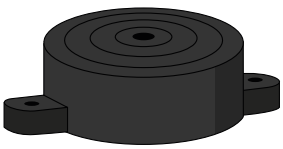
Um *díodo emissor de luz* (*LED*) é um *dispositivo de saída*; é diretamente controlado no seu programa. Os LED acendem quando estão ligados e encontram-se por toda a casa, desde os pequenos LED que indicam que a máquina de lavar roupa está ligada aos grandes que podem servir para iluminar as divisões. Os LED estão disponíveis numa vasta gama de formas, cores e tamanhos, mas nem todos são adequados para utilização com o Raspberry Pi: evite todos os que se destinam a fontes de alimentação de 5 V ou 12 V.



As *resistências* são componentes que controlam o fluxo de *corrente elétrica* e estão disponíveis em diferentes valores, medidos utilizando uma unidade chamada *ohms* (*Ω*). Quanto maior o número de ohms, mais resistência é fornecida. Para projetos de computação física com o Raspberry Pi, a sua utilização mais comum é impedir que os LED consumam demasiada corrente e se danifiquem ou ao seu Raspberry Pi; por esse motivo, procure resistências com classificação de aproximadamente 330 Ω, apesar de muitos fornecedores de eletricidade venderem pacotes práticos com uma série de diferentes valores comummente usados para lhe dar mais flexibilidade.



Um *alarme piezoelétrico*, normalmente apenas chamado alarme ou campainha, é outro dispositivo de saída. No entanto, enquanto um LED produz luz, um alarme produz um ruído – um zumbido, na verdade. Dentro da caixa de plástico do alarme há um par de placas metálicas; quando são ativadas, essas placas vibram umas contra as outras para produzirem o som do zumbido. Há dois tipos de campainhas: *alarmes ativos* e *alarmes passivos*. Certifique-se de que obtém uma campainha ativa, pois são as mais simples de usar.



Outros componentes elétricos comuns incluem motores, que precisam de uma placa de controlo especial para poderem ser ligados ao Raspberry Pi, sensores infravermelhos que detetam movimento, sensores de temperatura e humidade que podem ser utilizados para prever o tempo, e resistências dependentes da luz (LDR) – dispositivos de entrada que funcionam como um LED ao inverso ao detetar a luz.

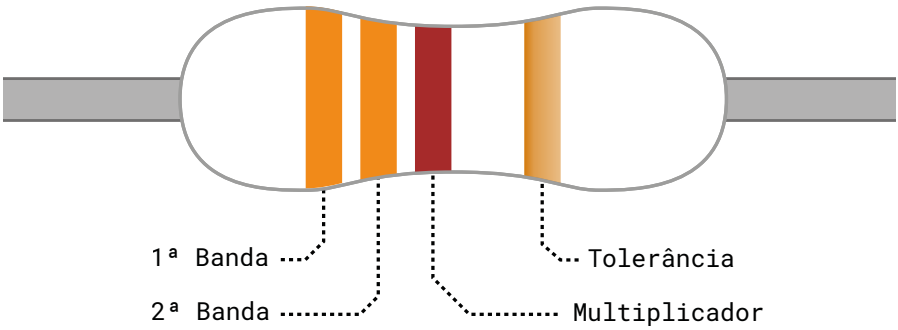
Vendedores em todo o mundo fornecem componentes para computação física com o Raspberry Pi, seja como peças individuais ou em kits que fornecem tudo o que precisa para começar. Para encontrar vendedores, visite **rpf.io/products** e clique em Raspberry Pi 4 para obter uma lista de lojas online parceiras do Raspberry Pi (revendedores aprovados) no seu país ou região.

Para concluir os projetos neste capítulo, deve ter pelo menos:

- 3 × LED: vermelho, verde e amarelo ou âmbar
- 2 × interruptores de botão de pressão
- 1 × alarme ativo
- Cabos jumper macho para fêmea (M2F) e fêmea para fêmea (F2F)
- Como opção, uma placa universal e cabos jumper macho para macho (M2M)

Leitura dos códigos de cores das resistências

As resistências são apresentadas numa vasta gama de valores, desde as versões de resistência zero, que são efetivamente apenas pedaços de arame, até às versões de alta resistência do tamanho da sua perna. Muito poucas destas resistências têm os seus valores impressos em números: em vez disso, utilizam um código especial impresso como faixas coloridas ao redor do corpo da resistência.



	1ª/2ª Banda	Multiplicador	Tolerância
Preto	0	$\times 10^{0}$	-
Castanho	1	$\times 10^{1}$	±1%
Vermelho	2	$\times 10^{2}$	±2%
Laranja	3	$\times 10^{3}$	-
Amarelo	4	$\times 10^{4}$	-
Verde	5	$\times 10^{5}$	±0.5%
Azul	6	$\times 10^{6}$	±0.25%
Tolet	7	$\times 10^{7}$	±0.1%
Cinza	8	$\times 10^{8}$	±0.05%
Branco	9	$\times 10^{9}$	-
Ouro	-	$\times 10^{-1}$	±5%
Prata	-	$\times 10^{-2}$	±10%
Nenhum	-	-	±20%

Para ler o valor de uma resistência, posicione-a para que o grupo de faixas esteja à esquerda e a faixa isolada à direita. A contar da primeira faixa, procure a respetiva cor na coluna "1ª/2ª Banda" da tabela para obter o primeiro e segundo dígitos. Este exemplo tem duas faixas laranja, ambas significam um valor de "3" para um total de "33". Se a sua resistência tem quatro faixas agrupadas em vez de três, anote também o valor da terceira faixa (para resistências de cinco/seis faixas, consulte **rpf.io/5-6band**).

Passando para a última faixa agrupada – a terceira ou quarta – procure a respetiva cor na coluna "Multiplicador". Isso indica o número necessário pelo qual deve multiplicar o seu número atual para obter o valor real da resistência. Este exemplo tem uma faixa castanha, que significa "$\times 10^1$". Pode parecer confuso, mas trata-se apenas de *notação científica*: "$\times 10^1$" significa simplesmente "adicionar um zero no fim do número". Se fosse azul, para "$\times 10^6$", significaria "adicionar seis zeros no fim do número".

33, das faixas laranja, mais o zero adicionado dada faixa castanha, dá 330 – que é o valor da resistência, medido em ohms. A faixa final, à direita, é a *tolerância* da resistência. Isto é simplesmente quão próximo do valor nominal é provável que esteja. As resistências mais baratas podem ter uma faixa prateada, indicando que podem ser 10% superiores ou inferiores à sua classificação, ou nenhuma última faixa, indicando que podem ser 20% superiores ou inferiores; as resistências mais caras têm uma faixa cinzenta, indicando que estarão nos 0,05% da sua classificação. Para projetos de hobby, a precisão não é tão importante: qualquer tolerância geralmente funcionará bem.

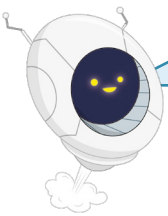
Se o valor da resistência for superior a 1000 ohms (1000 Ω), normalmente é classificado em kilohms (kΩ); se for superior a um milhão de ohms, são megohms (MΩ). Uma resistência de 2200 Ω seria escrita como 2,2 kΩ; uma resistência de 2 200 000 Ω seria escrita como 2,2 MΩ.

SABE A SOLUÇÃO?

Que faixas de cores teria uma resistência de 100 Ω? Que faixas de cores teria uma resistência de 2,2 MΩ? Se quisesse encontrar as resistências mais baratas, que faixa de tolerância de cor procuraria?

O seu primeiro programa de computação física: Olá, LED!

Tal como imprimir "Olá, Mundo" no ecrã é um fantástico primeiro passo na aprendizagem de uma linguagem de programação, fazer uma luz LED acender é a tradicional introdução à aprendizagem da computação física. Para este projeto, vai precisar de um LED e uma resistência de 330 ohm (330 Ω), ou o mais próximo possível de 330 Ω, mais cabos jumper fêmea para fêmea (F2F).

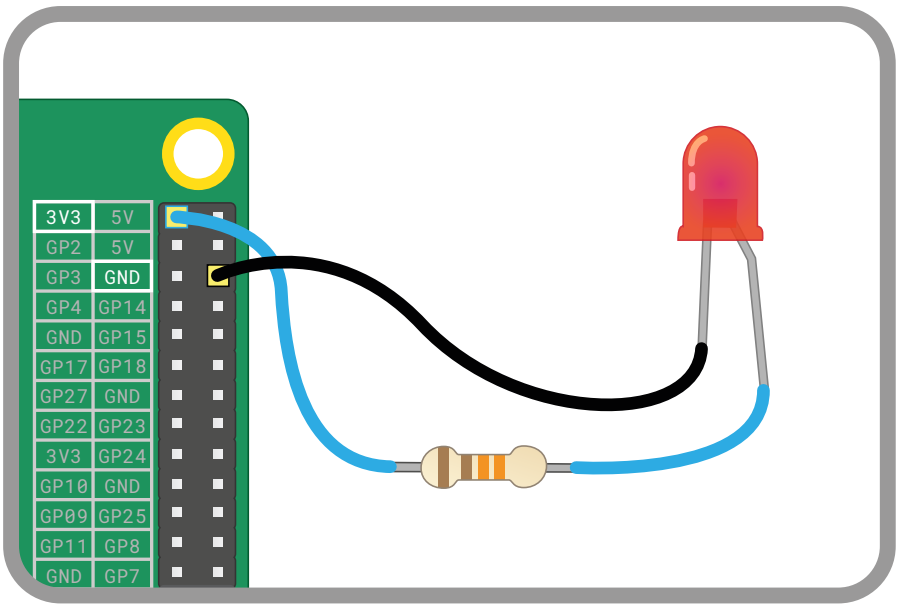


A RESISTÊNCIA É VITAL

A resistência é um componente vital neste circuito: protege o Raspberry Pi e o LED, limitando a quantidade de corrente elétrica que o LED pode absorver. Sem ela, o LED pode absorver muita corrente e entrar em curto-circuito, bem como o Raspberry Pi. Quando utilizada desta forma, a resistência é conhecida como uma *resistência limitadora de corrente*. O valor exato da resistência de que necessita depende do LED que utiliza, mas 330 Ω funciona para os LEDs mais comuns. Quanto mais alto o valor, mais escuro o LED; quanto mais baixo o valor, mais brilhante o LED.

Nunca ligue um LED a um Raspberry Pi sem uma resistência limitadora de corrente, a menos que saiba que o LED tem uma resistência integrada de valor apropriado.

Comece por verificar se o LED funciona. Vire o Raspberry Pi para que o conetor GPIO fique em duas faixas verticais no lado direito. Ligue uma extremidade da resistência de 330 Ω ao primeiro pino de 3,3 V (rotulado 3V3 na **Figura 6-1**) com um cabo jumper fêmea para fêmea, em seguida, ligue a outra extremidade à perna longa – positiva ou ânodo – do LED com outro cabo jumper fêmea para fêmea. Pegue no último cabo jumper fêmea para fêmea e ligue a perna curta – negativa, ou cátodo – do LED ao primeiro pino de terra (rotulado GND na **Figura 6-1**).



▲ **Figura 6-1:** Ligue o LED a estes pinos – não se esqueça da resistência!

Desde que o Raspberry Pi esteja ligado, o LED deve acender. Se não acender, verifique novamente o seu circuito: certifique-se de que não utilizou um valor de resistência demasiado alto, que todos os fios estão devidamente ligados e que escolheu os pinos GPIO corretos para corresponder ao diagrama. Verifique também as pernas do LED, pois os LED só funcionam de uma forma: com a perna mais longa ligada ao lado positivo do circuito e a mais curta ao lado negativo.

Quando o LED estiver a funcionar, pode programá-lo. Desligue o cabo jumper do pino de 3,3 V (rotulado 3V3 na **Figura 6-2**) e ligue-o ao pino GPIO 25 (rotulado GP25 na **Figura 6-2**). O LED vai apagar, mas não se preocupe, isso é normal.

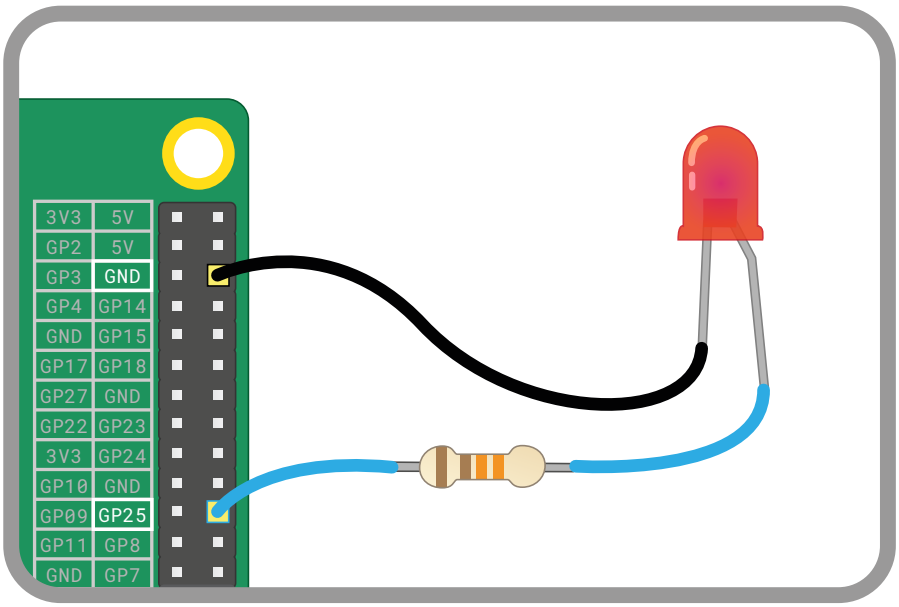
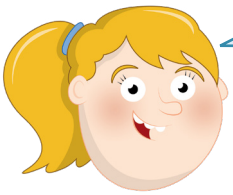


Figura 6-2: Desligue o cabo do 3V3 e ligue-o ao pino GPIO 25

Agora está pronto para criar um programa Scratch ou Python para ligar e desligar o seu LED.



CONHECIMENTO DE PROGRAMAÇÃO

Os projetos deste capítulo requerem que esteja confortável com a utilização do Scratch 3 e do ambiente de programação integrado (IDE) Thonny Python. Se ainda não o fez, procure o **Capítulo 4, Programar com Scratch 3** e o **Capítulo 5, Programar com Python** e trabalhe primeiro com esses projetos.

Controlo de LEDs no Scratch

Carregue o Scratch 3 e clique no ícone Adicionar extensão . Desloque para baixo para encontrar a extensão "Raspberry Pi GPIO" (**Figura 6-3** no verso) e clique. Esta ação carrega os blocos de que necessita para controlar o conetor GPIO do Raspberry Pi a partir do Scratch 3. Verá os novos blocos aparecerem na paleta de blocos; quando precisar deles, estarão disponíveis na categoria Raspberry Pi GPIO.

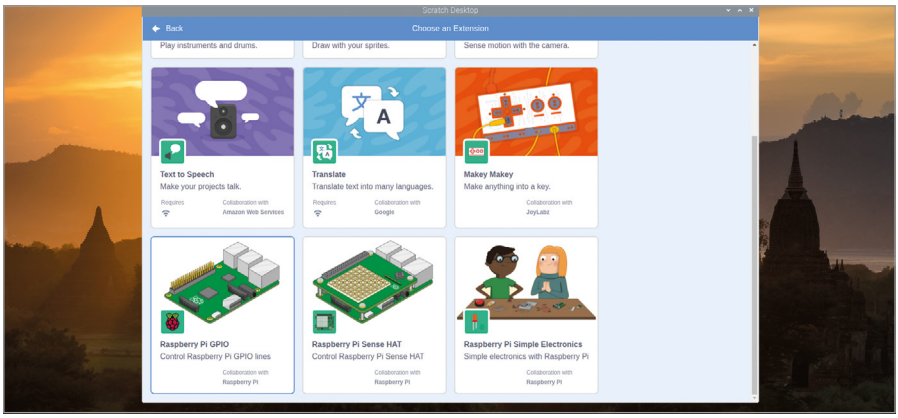
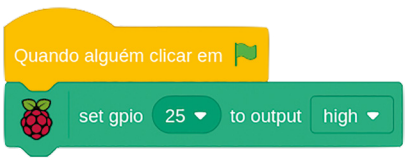
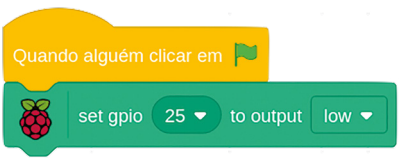


Figura 6-3: Adicionar a extensão Raspberry Pi GPIO ao Scratch 3

Comece por arrastar um bloco Eventos quando alguém clicar em para a área de programação e, em seguida, coloque um bloco set gpio to output high por baixo. Terá de escolher o número do pino que está a utilizar: clique na pequena seta para abrir a seleção pendente e clique em "25" para indicar ao Scratch que está a controlar o pino GPIO 25.

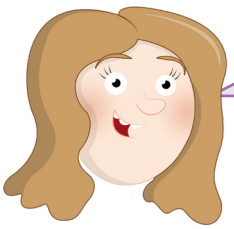


Clique na bandeira verde para executar o seu programa. Vai ver o seu LED acender: programou o seu primeiro projeto de computação física! Clique no octógono vermelho para parar o seu programa: vê como o LED permanece aceso? Isso acontece porque a única indicação do seu programa ao Raspberry Pi foi para ligar o LED – isso é o que significa a parte "output high" do bloco set gpio 25 to output high. Para desligá-lo novamente, clique na seta para baixo no final do bloco e selecione "low" na lista.



Clique na bandeira verde novamente e, desta vez, o programa apaga o LED. Para tornar as coisas mais interessantes, adicione um bloco de controlo repete para sempre e um par de blocos espera 1 s para criar um programa para acender e apagar o LED a cada segundo.

Clique na bandeira verde e observe o LED: vai acender um segundo, apagar um segundo, acender um segundo e continuar a repetir esse padrão até que clique no octógono vermelho para parar o LED. Veja o que acontece quando clica no octógono enquanto o LED está no estado ligado ou desligado.



DESAFIO: PODE ALTERÁ-LO?

Como mudaria o programa para o LED ficar ligado durante mais tempo? E ficar desligado durante mais tempo? Qual é o atraso mais pequeno que pode utilizar enquanto ainda é visível o interruptor LED a ligar e desligar?

Controlo de LEDs no Python

Carregue o Thonny na secção Programação do menu framboesa, depois clique no botão Novo para iniciar um novo projeto e Guardar para o guardar como **Olá, LED**. Para utilizar os pinos GPIO do Python, precisa de uma biblioteca chamada GPIO Zero. Para este projeto, apenas precisa da parte da biblioteca para trabalhar com LEDs. Importe apenas esta secção da biblioteca escrevendo o seguinte na área Shell do Python:

```
from gpiozero import LED
```

A seguir, tem de indicar à GPIO Zero a que pino GPIO o LED está ligado. Escreva o seguinte:

```
led = LED(25)
```

Juntas, estas duas linhas dão ao Python a capacidade de controlar os LED ligados aos pinos GPIO do Raspberry Pi e indicam-lhe qual o pino, ou pinos, se tiver mais do que um LED no seu circuito, que deve controlar. Para controlar efetivamente o LED, escreva o seguinte:

```
led.on()
```

Para desligar o LED novamente, escreva:

```
led.off()
```

Parabéns, agora tem o controlo dos pinos GPIO do Raspberry Pi no Python! Tente escrever essas duas instruções novamente. Se o LED já estiver apagado, **`led.off()`** não vai fazer nada; o mesmo é verdade se o LED já estiver aceso e introduzir **`led.on()`**.

Para criar um programa verdadeiro, escreva o seguinte na área Script:

```
from gpiozero import LED
from time import sleep

led = LED(25)

while True:
    led.on()
    sleep(1)
    led.off()
    sleep(1)
```

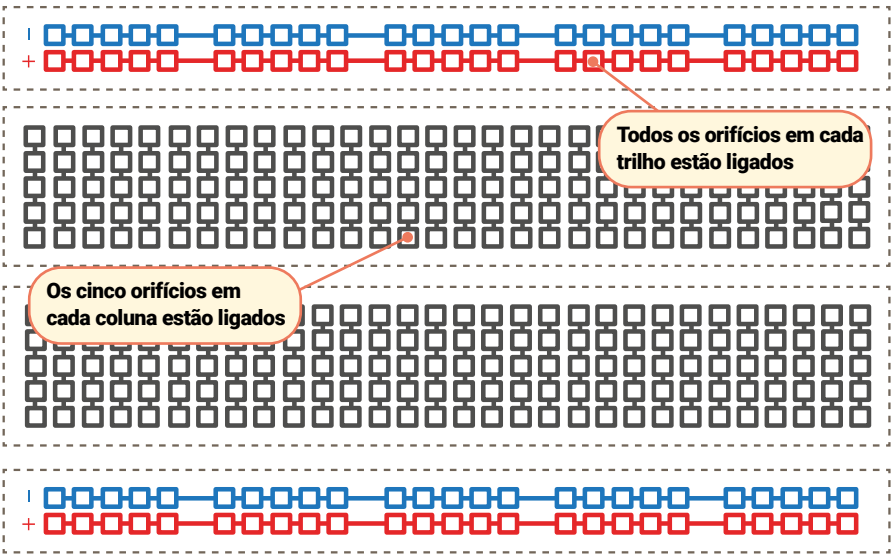
Este programa importa a função LED da biblioteca **`gpiozero`**(GPIO Zero) e a função **`sleep`** da biblioteca **`time`** e, em seguida, constrói um ciclo infinito para ligar o LED durante um segundo, desligá-lo durante um segundo e repetir. Clique no botão Executar para o ver em ação: o LED começará a piscar. Tal como no programa Scratch, anote o comportamento quando clica no botão Parar enquanto o LED está ligado e enquanto está desligado.

DESAFIO: ILUMINAÇÃO MAIS LONGA

Como mudaria o programa para o LED ficar ligado durante mais tempo? E ficar desligado durante mais tempo? Qual é o atraso mais pequeno que pode utilizar enquanto ainda é visível o interruptor LED a ligar e desligar?

Utilizar uma placa universal

Os próximos projetos neste capítulo serão muito mais fáceis de concluir se utilizar uma placa universal para fixar os componentes e fazer as ligações elétricas.



Uma placa universal é coberta de orifícios espaçados 2,54 mm entre si para a correspondência dos componentes. Por baixo desses orifícios há faixas de metal que atual como os cabos jumper que tem utilizado até agora. As faixas estão dispostas em filas ao longo da placa, sendo que a maioria das placas tem uma abertura ao meio para as dividir em duas metades. Muitas placas universais também têm letras na parte superior e números nos lados. Esses número e letras permitem-lhe encontrar um orifício específico: A1 é o canto superior esquerdo, B1 é o orifício imediatamente à direita e B2 é um orifício abaixo desse. A1 liga a B1 através das faixas metálicas ocultas, mas um orifício 1 nunca é ligado a nenhum orifício 2, a menos que adicione um cabo jumper.

As placas universais maiores também têm filas de orifícios nas laterais, tipicamente marcadas com riscas vermelhas e pretas ou vermelhas e azuis. São os *trilhos de corrente*, concebidos para facilitar as ligações dos cabos: pode ligar um único cabo do pino terra do Raspberry Pi a um dos trilhos de corrente – tipicamente marcado com uma risca azul ou preta e um símbolo negativo – para fornecer um *ponto em comum* para muitos componentes na placa universal, e pode fazer o mesmo se o seu circuito precisar de potência de 3,3 V ou 5 V.

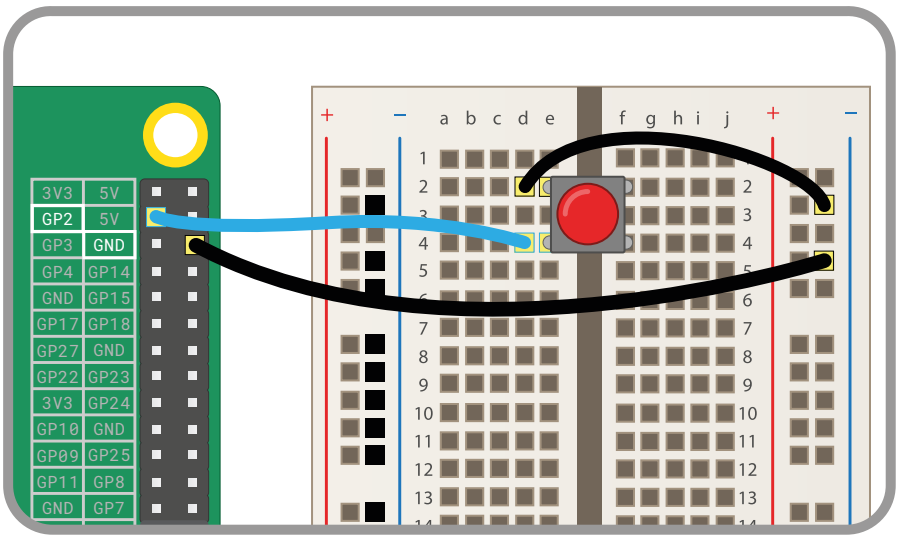
Adicionar componentes eletrónicos a uma placa universal é simples: basta alinhar os respetivos cabos (as peças metálicas salientes) com os orifícios e empurrar ligeiramente até que o componente esteja na posição correta. Para as ligações necessárias além das que a placa universal faz por si, pode utilizar cabos jumper macho para macho (M2M); para as ligações da placa universal ao Raspberry Pi, utilize cabos jumper macho para fêmea (M2F).

Nunca tente inserir mais do que um cabo jumper de um componente num único orifício da placa universal. Lembre-se: os orifícios são ligados em colunas, além da divisão no meio, de modo que um cabo de componente em A1 é eletricamente ligado a tudo o que adicionar a B1, C1, D1, e E1.

Passos seguintes: ler um botão

Saídas como LEDs são uma coisa, mas a parte "entrada/saída" de "GPIO" significa que também pode utilizar pinos como entradas. Para este projeto, vai precisar de uma placa universal, cabos jumper macho para macho (M2M) e macho para fêmea (M2M) e um interruptor de botão de pressão. Se não tiver uma placa universal, pode utilizar cabos jumper fêmea para fêmea (F2F), mas o botão será muito mais difícil de premir sem quebrar acidentalmente o circuito.

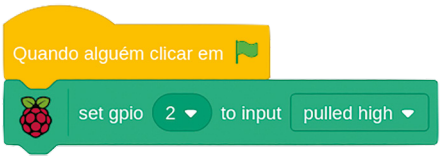
Comece por adicionar o botão de pressão à sua placa universal. Se o botão tiver apenas duas pernas, certifique-se de que estão em filas numeradas diferentes da placa universal; se tiver quatro pernas, vire-o de modo que os lados com as pernas salientes fiquem ao longo das filas da placa e os lados planos sem pernas fiquem na parte superior e inferior. Ligue o trilho terra da placa universal a um pino terra do Raspberry Pi (marcado com GND na **Figura 6-4**) com um cabo jumper macho para fêmea e, em seguida, ligue uma perna do botão de pressão ao trilho terra com um cabo jumper macho para macho. Por fim, ligue a outra perna – a que está no mesmo lado da perna que acabou de ligar, se estiver a utilizar um interruptor de quatro pernas – ao pino GPIO 2 (marcado com GP2 na **Figura 6-4**) do Raspberry Pi com um cabo jumper macho para fêmea.



▲ **Figura 6-4:** Ligar um botão de pressão aos pinos GPIO

Ler um botão no Scratch

Inicie um novo programa no Scratch e arraste um bloco quando alguém clicar em para a área de programação. Ligue um bloco set gpio to input pulled high e selecione o número 2 no menu pendente para corresponder ao pino GPIO que utilizou para o botão de pressão.



Se clicar agora na bandeira verde, não acontece nada. Isso é porque indicou ao Scratch que utilizasse o pino como uma entrada, mas não indicou uma ação correspondente a essa entrada. Arraste um bloco repete para sempre para o final da sequência e, depois, arraste um bloco se, então, senão para o interior. Encontre o bloco gpio is high?, arraste-o para o espaço em forma de diamante na parte se, então do bloco e utilize o menu pendente para selecionar o número 2 e indicar qual o pino GPIO a verificar. Arraste um bloco diz olá! durante 2 s para a parte senão do bloco e edite para dizer "Botão premido!". Deixe a parte "se, então" do bloco vazia por enquanto.

Há muita coisa a decorrer, mas comece por um teste: clique na bandeira verde e prima o botão na placa universal. O sprite deve indicar que o botão foi premido: leu com êxito uma entrada do pino GPIO!

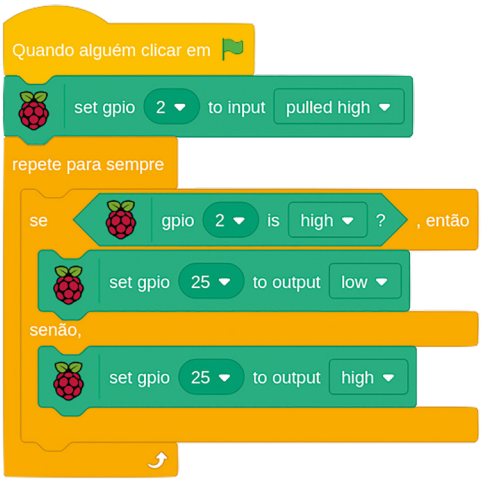
Pode ter reparado que a parte se gpio 2 is high? então do bloco está vazia. O código executado quando o botão é premido está na parte senão do bloco. Parece confuso, não é suposto o botão ficar para cima ao premir? De facto, acontece o oposto: Os pinos GPIO do Raspberry Pi são normalmente altos ou ligados quando definidos como uma entrada, pelo que premir o botão empurra-os para baixo.

Observe novamente o seu circuito: veja como o botão está ligado ao pino GPIO 2, que fornece a parte positiva do circuito, e o pino terra. Quando o botão é premido, a tensão no pino GPIO é puxada para baixo através do pino terra e o programa Scratch para de executar o código no bloco se gpio 2 is high? então para executar o código na parte senão do bloco.

Se tudo isto soar surpreendente, lembre-se: um botão num pino GPIO do Raspberry Pi é premido quando o pino vai para baixo, não quando vai para cima!

Para expandir ainda mais o seu programa, adicione o LED e a resistência novamente ao circuito: não se esqueça de ligar a resistência ao pino GPIO 25 e à perna longa do LED, e a perna mais curta do LED ao trilho terra da placa universal.

Arraste o bloco diz Botão premido! durante 2 s da área de programação para a paleta de blocos para eliminá-lo e, em seguida, substitua-o por um bloco set gpio 25 to output high – sem esquecer que terá de alterar o número GPIO com a seta pendente. Adicione um bloco set gpio 25 to output low, sem se esquecer de alterar os valores, à parte se gpio 2 is high? então do bloco atualmente vazia.



Clique na bandeira verde e prima o botão. O LED ficará aceso enquanto mantiver o botão premido. Se o soltar, o LED apaga. Parabéns: está a controlar um pino GPIO com base numa entrada de outro pino!

DESAFIO: MANTER O LED ILUMINADO

Como mudaria o programa para o LED ficar ligado durante alguns segundos, mesmo após soltar o botão? O que seria necessário mudar para o LED ficar ligado quando não prime o botão e desligado quando prime?

Ler um botão no Python

Clique no botão Novo no Thonny para iniciar um novo projeto e no botão Guardar para o guardar como **Entrada do botão**. Utilizar um pino GPIO como entrada para um botão é muito semelhante a utilizar um pino como saída para um LED, mas é necessário importar uma parte diferente da biblioteca GPIO Zero. Escreva o seguinte na área Script:

```
from gpiozero import Button
button = Button(2)
```

Para que o código seja executado quando o botão é premido, a GPIO Zero fornece a função **`wait_for_press`**. Escreva o seguinte:

```
button.wait_for_press()
print("Premiu o botão!")
```

Clique no botão Executar e prima o botão de pressão. A sua mensagem será impressa na Shell do Python na parte inferior da janela do Thonny: leu com êxito uma entrada do pino GPIO! Se quiser experimentar o seu programa novamente, tem de clicar novamente no botão Executar; como não há nenhum ciclo no programa, este encerra quando termina de imprimir a mensagem na Shell.

Para expandir ainda mais o seu programa, adicione o LED e a resistência novamente ao circuito, se ainda não fez: não se esqueça de ligar a resistência ao pino GPIO 25 e à perna longa do LED, e a perna mais curta do LED ao trilho terra da placa universal.

Para controlar um LED, bem como ler um botão, é necessário importar as funções **`Button`** e **`LED`** da biblioteca GPIO Zero. Também vai precisar da função **`sleep`** da biblioteca **`time`**. Volte à parte superior do seu programa e introduza o seguinte como as novas primeiras duas linhas:

```
from gpiozero import LED
from time import sleep
```

Por baixo da linha **`button = Button(2)`**, escreva:

```
led = LED(25)
```

Apague a linha **`print("Premiu o botão!")`** e substitua-a por:

```
led.on()
sleep(3)
led.off()
```

O seu programa final deve ter o seguinte aspeto:

```
from gpiozero import LED
from time import sleep
from gpiozero import Button

button = Button(2)
led = LED(25)
button.wait_for_press()
led.on()
sleep(3)
led.off()
```

Clique no botão Executar e prima o botão de pressão: o LED acende durante três segundos, depois apaga novamente e o programa encerra. Parabéns: pode controlar um LED com uma entrada de botão no Python!

DESAFIO: ADICIONAR UM CICLO

Como adicionaria um ciclo para que o programa repita em vez de encerrar após premir um botão? O que seria necessário mudar para o LED ficar ligado quando não prime o botão e desligado quando prime?

Vamos fazer barulho: controlar um alarme

Os LED são um ótimo dispositivo de saída, mas não são muito úteis se estiver a olhar noutra direção. A solução: alarmes, que emitem um ruído audível em qualquer parte da divisão. Para este projeto, vai precisar de uma placa universal, cabos jumper macho para fêmea (M2M) e um alarme ativo. Se não tiver uma placa universal, pode ligar o alarme utilizando cabos jumper fêmea para fêmea (F2F).

Um alarme ativo pode ser tratado exatamente como um LED, em termos de circuitos e programação. Repita o circuito que criou para o LED, mas substitua o LED pelo alarme ativo e deixe a resistência de fora, pois o alarme vai precisar de mais corrente para funcionar. Ligue uma perna do alarme ao pino GPIO 15 (rotulado GP15 na **Figura 6-5**) e a outra ao pino terra (rotulado GND no diagrama) utilizando a placa universal e os cabos jumper macho para fêmea.

Se o alarme tiver três pernas, certifique-se de que a perna marcada com o símbolo menos (-) está ligada ao pino terra, e a perna marcada com "S" ou "SIGNAL" está ligada a GPIO 15. Em seguida, ligue a perna restante – normalmente a do meio – ao pino de 3,3 V (rotulado 3V3)

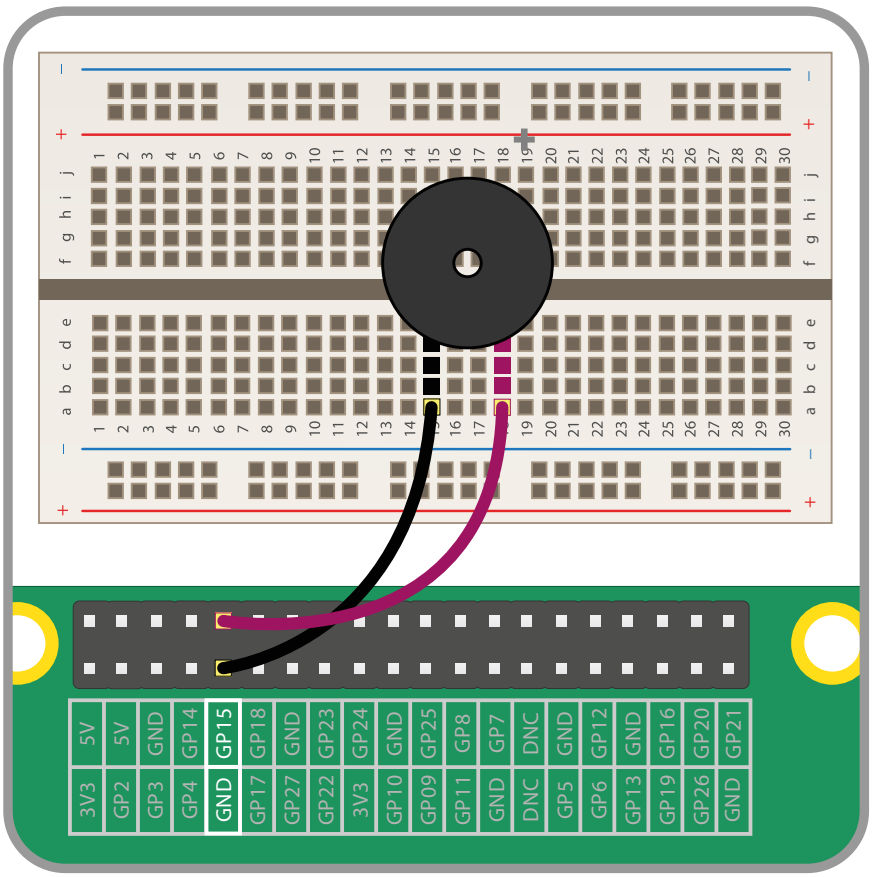


Figura 6-5: Ligar um alarme aos pinos GPIO

Controlar um alarme no Scratch

Recrie o mesmo programa que utilizou para fazer o LED piscar, ou carregue-o, se o tiver guardado antes de criar o projeto do botão. Utilize o menu pendente nos blocos **set gpio to output high** para selecionar o número 15, para que o Scratch controle o pino GPIO correto.

Clique na bandeira verde e o alarme começará a tocar: um segundo ligado, um segundo desligado. Se apenas ouvir o alarme clicar uma vez por segundo, significa que está a utilizar um alarme passivo em vez de um alarme ativo. Ao passo que um alarme ativo gera um sinal que muda rapidamente, conhecido como *oscilação*, para fazer vibrar as placas metálicas, um alarme passivo requer um sinal oscilante. Se apenas o ligar utilizando o Scratch, as placas movem-se só uma vez e param, emitindo o som de "clique" até à próxima vez que o programa ligar ou desligar o pino.

Clique no octógono vermelho para parar o alarme, mas faça-o apenas quando não estiver a emitir som, caso contrário, o alarme continuará a tocar até que execute novamente o seu programa!

DESAFIO: MUDAR O ALARME

Como mudaria o programa para o som do alarme ser mais curto? Pode construir um circuito para o alarme ser controlado por um botão?

Controlar um alarme no Python

Controlar um alarme ativo através da biblioteca GPIO Zero é quase idêntico a controlar um LED, na medida em que tem estados ligados e desligados. Mas precisa de uma função diferente: **`buzzer`**. Inicie um novo projeto no Thonny, guarde-o como **Buzzer** e escreva o seguinte:

```
from gpiozero import Buzzer
from time import sleep
```

À semelhança do que sucede com os LED, a GPIO Zero precisa de saber a que pino está ligado o alarme para poder controlá-lo. Escreva o seguinte:

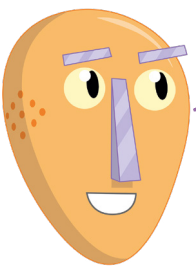
```
buzzer = Buzzer(15)
```

A partir daqui, o seu programa é quase idêntico ao que escreveu para controlar o LED; a única diferença (além de um número de pino GPIO diferente) é que está a utilizar **`buzzer`** em vez de **`led`**. Escreva o seguinte:

```
while True:
    buzzer.on()
    sleep(1)
    buzzer.off()
    sleep(1)
```

Clique no botão Executar e o alarme começará a tocar: um segundo ligado, um segundo desligado. Se estiver a utilizar um alarme passivo em vez de um alarme ativo, apenas ouvirá um breve clique a cada segundo em vez de um zumbido contínuo: isto acontece porque um alarme passivo não tem um *oscilador* para criar o sinal de mudança rápida que faz vibrar as placas dentro do alarme.

Clique no botão Parar para sair do programa, mas certifique-se de que o alarme não está a emitir som no momento, caso contrário, continuará a tocar até que execute o seu programa novamente!



DESAFIO: UM MELHOR ALARME

Como mudaria o programa para o som do alarme ser mais curto? Pode construir um circuito para o alarme ser controlado por um botão?

Projeto do Scratch: Semáforos

Agora que sabe como usar botões, alarmes e LEDs como entradas e saídas, está pronto para construir um exemplo de computação do mundo real: semáforos, completos com um botão que pode premir para atravessar a estrada. Para este projeto, vai precisar de uma placa universal, de LEDs vermelho, amarelo e verde, de três resistências de 330 Ω, de um alarme, de um interruptor de botão de pressão e de uma seleção de cabos jumper macho para macho (M2M) e macho para fêmea (M2M).

Comece por construir o circuito (**Figura 6-6**), ligando o alarme ao pino GPIO 15 (rotulado GP15 na **Figura 6-6**), o LED vermelho ao pino GPIO 25 (rotulado GP25), o LED amarelo ao GPIO 8 (GP8), o LED verde ao GPIO 7 (GP7) e o interruptor ao GPIO 2 (GP2). Não se esqueça de ligar as resistências de 330 Ω entre os pinos GPIO e as pernas longas dos LED, e ligue as segundas pernas em todos os componentes ao trilho terra da placa universal. Por fim, ligue o trilho terra a um pino terra (rotulado GND) no Raspberry Pi para concluir o circuito.

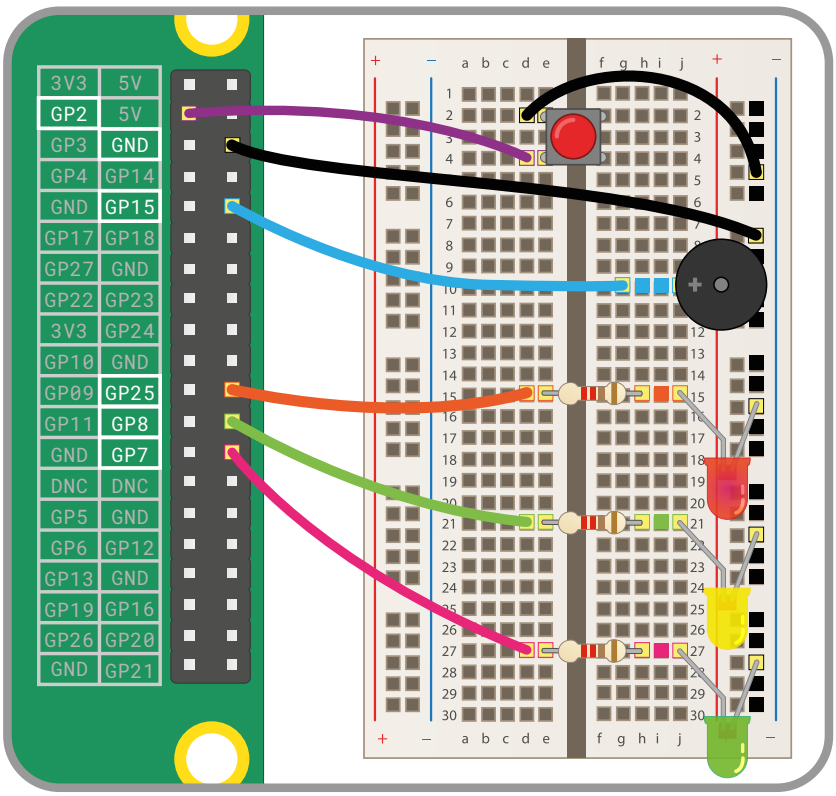
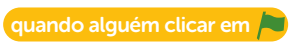
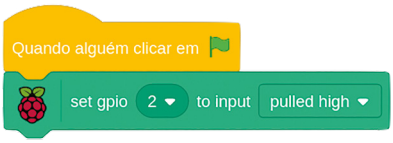


Figura 6-6: Diagrama de cabos para o projeto Semáforos

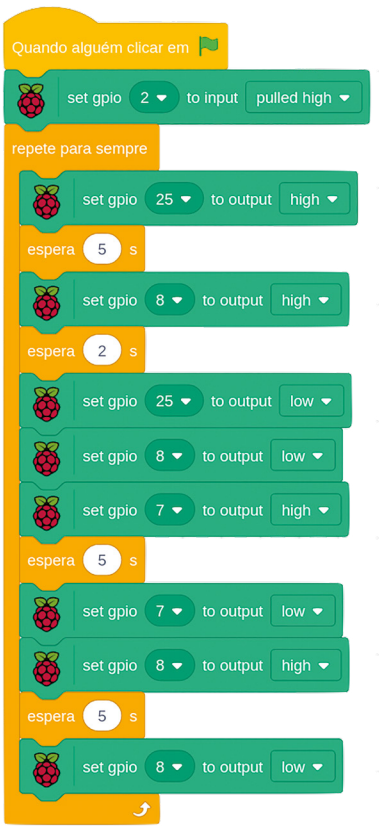
Inicie um novo programa no Scratch 3 e arraste um bloco 

para a área de programação. Em seguida, deve indicar ao Scratch que o pino GPIO 2, que

está ligado ao botão de pressão no seu circuito, é uma entrada e não uma saída: arraste um bloco **set gpio to input pulled high** da categoria Raspberry Pi GPIO da paleta de blocos por baixo do seu bloco **quando alguém clicar em** . Clique na seta para baixo ao lado de "0" e selecione o número 2 na lista pendente.



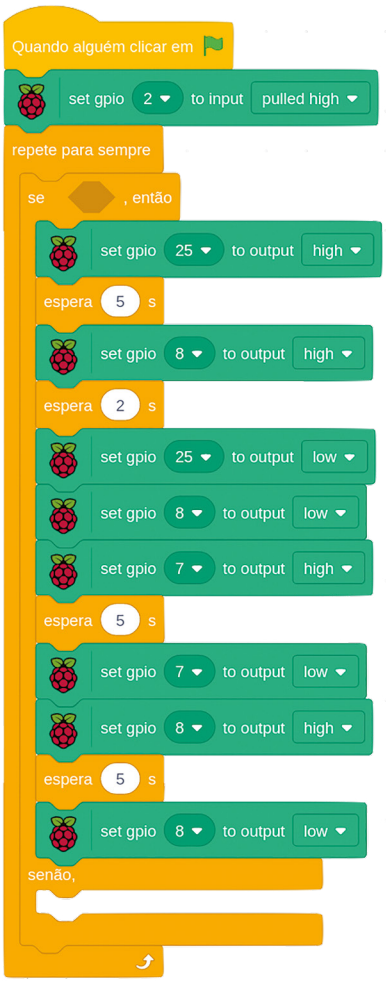
A seguir, tem de criar a sequência de semáforos. Arraste um bloco **repete para sempre** para o seu programa e, em seguida, preencha-o com blocos para ligar e desligar os LEDs do semáforo num padrão. Memorize que pinos GPIO estão ligados a que componentes: quando utiliza o pino 25, está a utilizar o LED vermelho, o pino 8, o LED amarelo, e o pino 7 o LED verde.



Clique na bandeira verde e observe os LED: primeiro acende o vermelho, depois o vermelho e o amarelo, depois o verde, depois o amarelo, e finalmente a sequência repete com a luz

vermelha mais uma vez. Este padrão corresponde ao que é utilizado pelos semáforos no Reino Unido; se desejar, pode editar a sequência para combinar padrões noutros países.

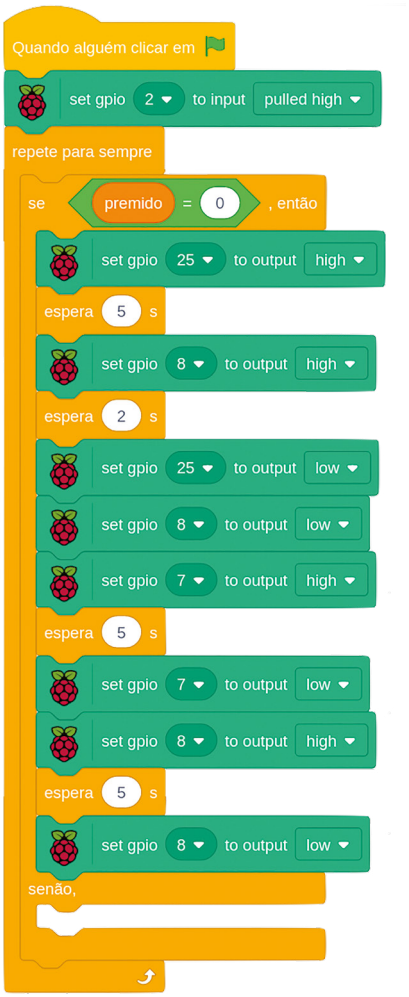
Para simular uma passagem de peões, o programa tem de saber que o botão é premido. Clique no octógono vermelho para parar o seu programa, se estiver em funcionamento. Arraste um bloco **se, então, senão** para a área de scripts e ligue-o de forma que fique diretamente por baixo do seu bloco **repete para sempre**, com a sequência dos semáforos na secção "if then". Deixe o espaço em forma de diamante vazio por agora.



Uma passadeira verdadeira não muda o sinal para vermelho mal o botão é premido, espera pelo próximo sinal vermelho na sequência. Para integrar esse esquema no seu próprio programa, arraste um bloco **when gpio is low** para a área de programação e selecione "2" na lista pendente. Em seguida, arraste um bloco **altera premido para 1** para baixo.

Esta pilha de blocos observa quando o botão é premido e define a variável "premido" para 1. Definir uma variável desta forma permite-lhe guardar o facto de o botão ter sido premido, mesmo que não atue imediatamente.

Regresse à sua pilha de blocos original e encontre o bloco **se, então**. Arraste um bloco Operador **= =** para o diamante vazio do bloco **se, então** e, em seguida, arraste um bloco repórter **premido** para o primeiro espaço em branco. Escreva "0" sobre o "50" do lado direito do bloco.



Clique na bandeira verde e observe a sequência dos semáforos. Prima o interruptor do botão de pressão: no início, parece que não acontece nada, mas quando a sequência chegar ao fim – apenas com o LED amarelo aceso – os semáforos apagam-se e permanecem desligados, graças à variável que introduziu.

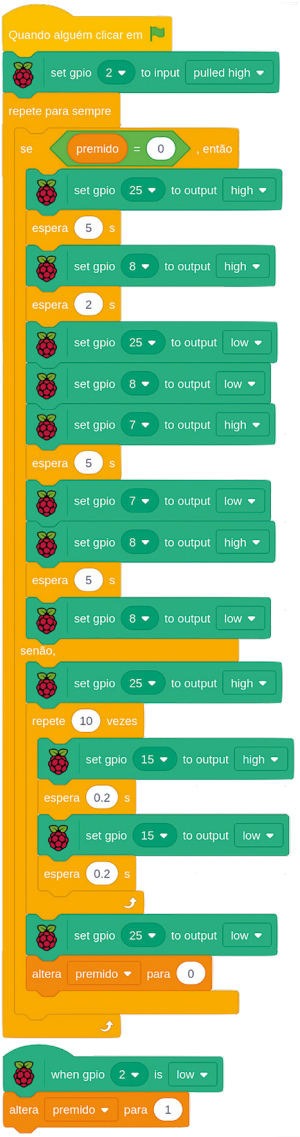
Tudo o que falta fazer é atribuir ao botão da passadeira outra função efetiva além de desligar as luzes. Na pilha de blocos principal, encontre o bloco **senão** e arraste um bloco **set gpio 25 to output high** para o seu interior, sem se esquecer de alterar o número do pino GPIO predefinido para corresponder ao pino a que o LED vermelho está ligado.

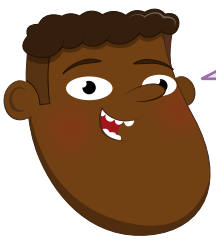
Por baixo, ainda no bloco **senão**, crie um padrão para o alarme: arraste um bloco **repete 10 vezes**, e depois preencha-o com blocos **set gpio 15 to output high**, **espera 0.2 s**, **set gpio 15 to output low** e **espera 0.2 s**, alterando os valores dos pinos GPIO para corresponderem ao pino do componente do alarme.

Por fim, por baixo da parte inferior do seu bloco **repete 10 vezes**, mas ainda no bloco **senão**, adicione um bloco **set gpio 25 to output low** e um bloco **altera premido para 0** – o último bloco repõe a variável que armazena o a pressão do botão, para que a sequência do alarme não se repita para sempre.

Clique na bandeira verde e prima o botão na placa universal. Quando a sequência terminar, verá a luz vermelha acender e o alarme emitir som para que os peões saibam que é seguro atravessar. Após alguns segundos, o alarme para e a sequência de semáforos começa novamente e continua até à próxima vez que premir o botão.

Parabéns: programou o seu próprio conjunto de semáforos totalmente funcionais, completo com passadeira!





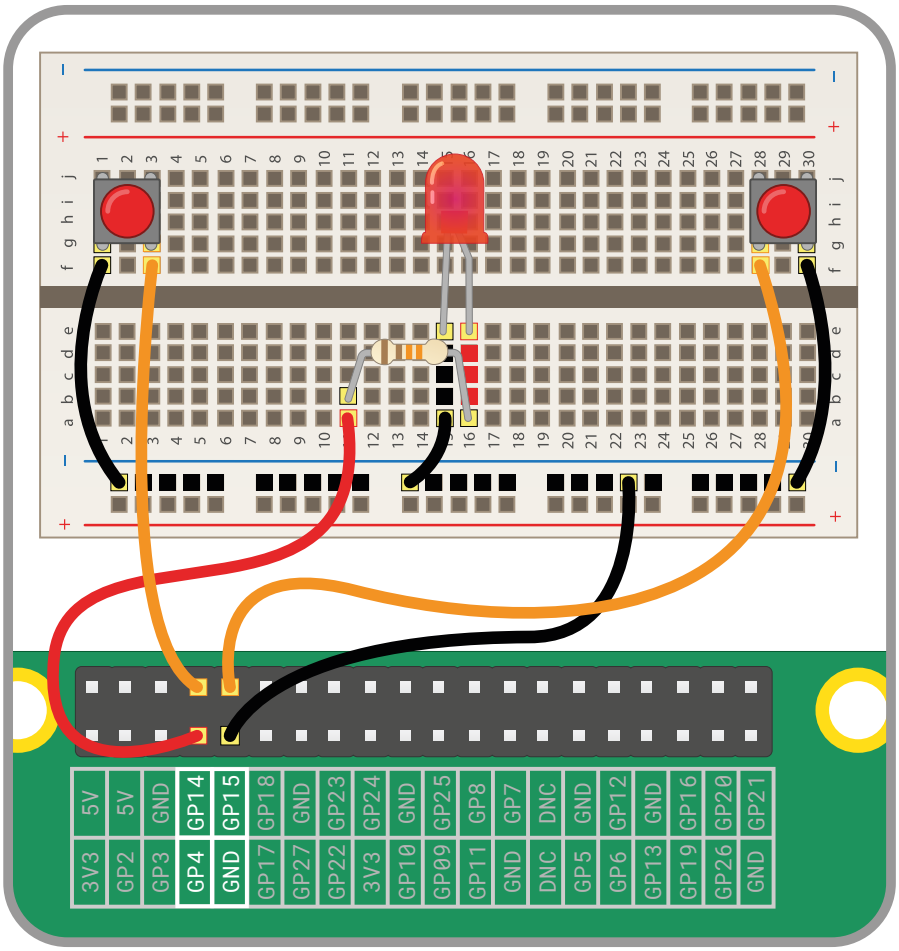
DESAFIO: CONSEGUE MELHORÁ-LO?

Pode alterar o programa para dar ao peão mais tempo para atravessar? Pode encontrar informações sobre os padrões de semáforos de outros países e reprogramar as suas luzes para corresponderem? Como poderia tornar os LED menos brilhantes?

Projeto do Python: Jogo de reação rápida

Agora que sabe como usar botões e LEDs como entradas e saídas, está pronto para construir um exemplo de computação do mundo real: um jogo de reação rápida, concebido para ver quem tem os reflexos de reação mais rápidos! Para este projeto, vai precisar de uma placa universal, de um LED, de uma resistência de 330 Ω, de dois interruptores de botão, de alguns cabos jumper macho para fêmea (M2F) e alguns macho para macho (M2M).

Comece por construir o circuito (**Figura 6-7**): ligue o primeiro interruptor no lado esquerdo da placa universal ao pino GPIO 14 (rotulado GP14 na **Figura 6-7**), o segundo interruptor no lado direito da PLACA UNIVERSAL ao pino GPIO 15 (rotulado GP15), a perna mais longa do LED à resistência de 330 Ω, que depois se liga ao pino GPIO 4 (rotulado GP4) do Raspberry Pi, e a segunda perna de todos os componentes ao trilho terra da placa universal. Por fim, ligue o trilho terra a um pino terra (rotulado GND) no Raspberry Pi.



▲ **Figura 6-7:** Diagrama de cabos para o Jogo de reação rápida

Inicie um novo projeto no Thonny e guarde-o como **Jogo de reação**. Vai utilizar as funções **LED** e **button** da biblioteca GPIO Zero e a função **sleep** da biblioteca time. Em vez de importar as duas funções GPIO Zero em duas linhas separadas, pode poupar tempo e importá-las em conjunto utilizando um símbolo de vírgula (**,**) para separá-las. Escreva o seguinte na área de scripts:

```
from gpiozero import LED, Button
from time import sleep
```

Como antes, tem de indicar a GPIO Zero que a que pinos os dois botões e o LED estão ligados. Escreva o seguinte:

```
led = LED(4)
right_button = Button(15)
left_button = Button(14)
```

Agora, adicione instruções para ligar e desligar o LED, para que possa verificar se está a funcionar corretamente:

```
led.on()
sleep(5)
led.off()
```

Clique no botão Executar: o LED acende durante cinco segundos, depois apaga novamente e o programa encerra. Para efeitos de um jogo de reação, no entanto, é um pouco previsível se o LED apagar sempre após exatamente 5 segundos. Adicione o seguinte por baixo da linha **from time import sleep**:

```
from random import uniform
```

A biblioteca random (aleatória), como o nome sugere, permite gerar números aleatórios (aqui com uma distribuição uniforme, consulte **rpf.io/uniform**). Encontre a linha **sleep(5)** e altere-a para ler:

```
sleep(uniform(5, 10))
```

Clique novamente no botão Executar: desta vez, o LED ficará aceso durante um número aleatório de segundos, entre 5 e 10. Conte para ver quanto tempo demora até o LED apagar e, em seguida, clique no botão Executar mais algumas vezes: verá que o tempo é diferente para cada execução, o que torna o programa menos previsível.

Para transformar os botões em acionadores para cada jogador, tem de adicionar uma função. Desloque até o final do seu programa e escreva o seguinte:

```
def pressed(button):
    print(str(button.pin.number) + " ganhou o jogo")
```

Tenha em atenção que o Python utiliza indentação para saber que linhas fazem parte da sua função; o Thonny automaticamente indenta a segunda linha por si. Por fim, adicione as duas linhas seguintes para detetar quando os jogadores primem os botões, que não devem ser indentadas, caso contrário, o Python trata-as como parte da função criada por si.

```
right_button.when_pressed = pressed
left_button.when_pressed = pressed
```

Execute o seu programa e, desta vez, tente premir um dos dois botões logo que o LED se apague. Verá uma mensagem para premir o primeiro botão impressa na Shell do Python na parte inferior da janela do Thonny. Infelizmente, também verá mensagens sempre que os botões forem premidos, mas com o número do pino em vez de um nome amigável para o botão.

Para corrigir essa situação, comece por perguntar aos jogadores os seus nomes. Por baixo da linha **`from random import uniform`**, escreva o seguinte:

```
left_name = input("Nome do jogador esquerdo é ")
right_name = input("Nome do jogador direito é")
```

Volte à sua função e substitua a linha **`print(str(button.pin.number) + " ganhou o jogo")`** por:

```
    if button.pin.number == 14:
        print (left_name + " ganhou o jogo")
    else:
        print(right_name + " ganhou o jogo")
```

Clique no botão Executar e escreva os nomes de ambos os jogadores na área da Shell do Python. Desta vez, quando premir o botão o mais rápido possível após o LED apagar, verá o nome do jogador impresso em vez do número do pino.

Para corrigir o problema de todas as pressões nos botões serem tratadas como ganhar o jogo, é necessário adicionar uma nova função da biblioteca do *sistema*: **`exit`**. Por baixo da última linha **`import`**, escreva o seguinte:

```
from os import _exit
```

A seguir, no fim da sua função, por baixo da linha **`print(right_name + " ganhou o jogo")`**, escreva o seguinte:

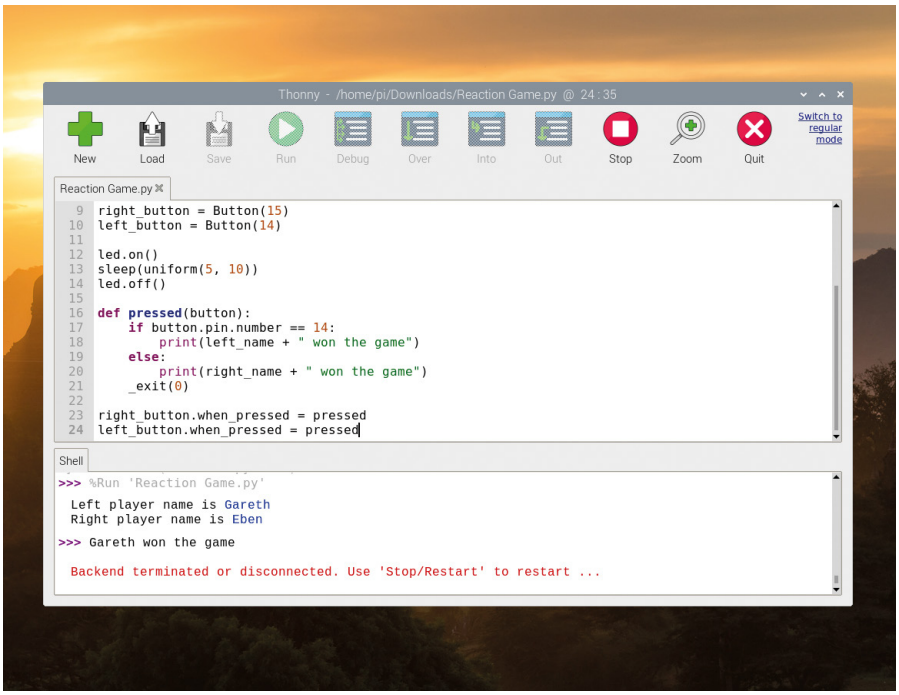
```
    _exit(0)
```

A indentação é importante aqui: **`_exit(0)`** deve ser indentado em quatro espaços, alinhado com **`else:`** duas linhas acima e **`if`** duas linhas acima desse ponto. Esta instrução diz ao Python que pare o programa após o primeiro botão ser premido, o que significa que o jogador cujo botão é premido em segundo lugar não recebe nenhuma recompensa por perder!

O seu programa final deve ter o seguinte aspeto:

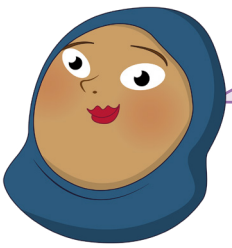
```
from gpiozero import LED, Button
from time import sleep
from random import uniform
from os import _exit

left_name = input("Nome do jogador esquerdo é ")
right_name = input ("Nome do jogador direito é ")
led = LED(4)
right_button = Button(15)
left_button = Button(14)

led.on()
sleep(uniform(5, 10))
led.off()

def pressed(button):
    if button.pin.number == 14:
        print (left_name + " ganhou o jogo")
    else:
        print(right_name + " ganhou o jogo")
    _exit(0)

right_button.when_pressed = pressed
left_button.when_pressed = pressed
```

Clique no botão Executar, introduza os nomes dos jogadores, espere que o LED apague e verá o nome do jogador vencedor. Também verá uma mensagem do próprio Python: **`Backend terminated or disconnected . Use 'Stop/Restart' to restart ...`** Isto significa simplesmente que o Python recebeu o seu comando **`_exit(0)`** e parou o programa, mas que será necessário clicar no ícone Parar para sair completamente e preparar o programa para outra ronda (**Figura 6-8**).



▲ **Figura 6-8:** Quando o vencedor for decidido, terá de parar o programa

Parabéns: criou o seu próprio jogo físico!



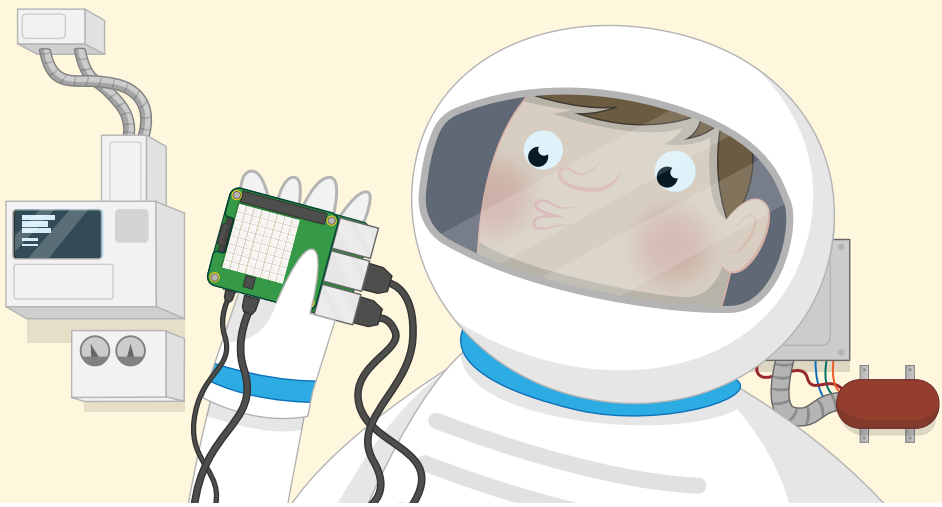
DESAFIO: MELHORAR O JOGO

Pode adicionar um ciclo para que o jogo execute continuamente? Não se esqueça de remover a instrução `_exit(0)` primeiro! Pode adicionar um contador de pontos para poder ver quem está a ganhar nas várias jogadas? E que tal um temporizador, para que possa ver quanto tempo demorou a reagir ao apagar da luz?

Capítulo 7

Computação física com a Sense HAT

Como usado na Estação Espacial Internacional, a Sense HAT é uma placa multifunções adicional para o Raspberry Pi, equipada com sensores e um ecrã de matriz LED



O Raspberry Pi vem com um suporte para um tipo especial de placa adicional intitulado *Hardware Attached on Top (HAT)*. Os HATs podem adicionar de tudo, desde microfones e luzes a relés e telas eletrónicas ao Raspberry Pi, mas um HAT em particular é muito especial: a Sense HAT.

A Sense HAT foi concebida especialmente para a missão espacial Astro Pi. Um projeto conjunto entre a Fundação Raspberry Pi, a Agência Espacial do Reino Unido e a Agência Espacial Europeia, na Astro Pi observou-se placas Raspberry Pi e Sense HATs serem transportadas até a Estação Espacial Internacional a bordo de um foguetão de carga da Orbital Science Cygnus. Desde que chegaram em segurança à órbita acima da Terra, as Sense HATs, apelidadas de Ed e Izzy pelos astronautas, têm sido usadas para executar código e realizar experiências científicas para as quais contribuíram estudantes de toda a Europa.

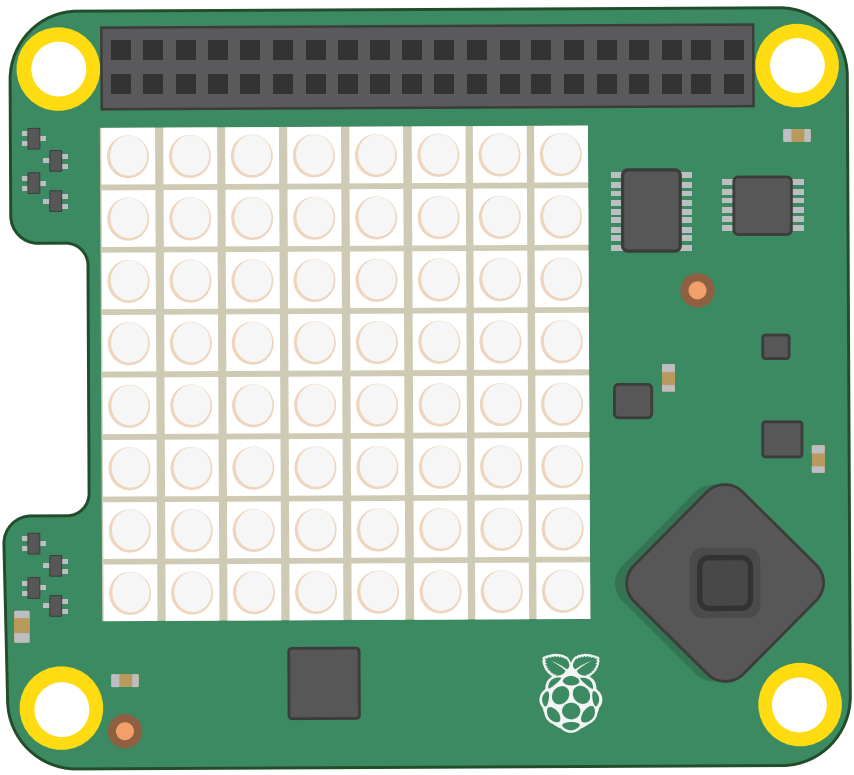
Embora Ed e Izzy estejam um pouco longe para poder usá-los você mesmo, também pode encontrar aqui na Terra o mesmo hardware Sense HAT, em todos os revendedores Raspberry Pi, e se para já não quiser comprar uma Sense HAT, pode simular uma através de software!

REAL OU SIMULADO

Este capítulo é o melhor complemento a uma Sense HAT genuína anexada a um conetor GPIO de um Raspberry Pi, mas alguém que não possuir um exemplar pode ignorar a secção intitulada "Instalar a Sense HAT" e experimentar os projetos no emulador Sense HAT; irão funcionar na mesma!

Apresentamos a Sense HAT

A Sense HAT é um suplemento poderoso e multifunções para o Raspberry Pi. Para além de uma matriz 8×8 de 64 LEDs programáveis vermelhos, verdes e azuis (RGB) que podem ser controlados para produzir qualquer cor a partir de uma gama de milhões, a Sense HAT inclui um comando joystick com cinco direções e seis sensores na placa.



Sensor de giroscópio: Usado para sentir mudanças de ângulo ao longo do tempo, conhecidas tecnicamente como *velocidade angular*, ao monitorizar a direção do campo de gravidade da Terra, a força que puxa tudo para baixo em direção ao centro do planeta. Dito de forma simples, o sensor giroscópico pode indicar-lhe quando roda a Sense HAT em relação à superfície da Terra e quão rápido está a girar.

Acelerómetro: Semelhante ao sensor giroscópio, mas em vez de monitorar um ângulo relativo à gravidade da Terra, mede a força de aceleração em várias direções. Combinadas, as leituras (dados) dos dois sensores podem ajudá-lo a monitorizar para onde uma Sense HAT está direcionada e como está a ser movimentada.

Magnetómetro: Mede a força de um campo magnético, e é outro sensor que pode ajudar a monitorizar os movimentos da Sense HAT: ao medir o campo magnético natural da Terra, o magnetómetro consegue encontrar a direção do norte magnético. O mesmo sensor também pode ser usado para detetar objetos metálicos, e até mesmo campos elétricos. Todos estes três sensores estão incorporados num único chip, rotulado de "ACCEL/GYRO/MAG" na placa de circuito da Sense HAT.

Sensor de humidade: Mede a quantidade de vapor de água no ar, conhecida como a *humidade relativa*. A humidade relativa pode variar entre 0%, por não haver água, e 100%, devido ao facto de o ar estar completamente saturado. Os dados relativos à humidade podem ser usados para detetar quando pode estar quase a chover!

Sensor de pressão barométrica:também conhecido como *barómetro*mede a pressão do ar. Embora a maioria das pessoas esteja familiarizada com a pressão barométrica da previsão do tempo, o barómetro tem uma segunda utilização secreta: pode monitorizar quando está a subir ou a descer uma colina ou montanha, à medida que o ar fica mais fino e a pressão mais baixa quanto mais longe estiver do nível do mar da Terra.

Sensor de temperatura: Mede o nível de calor ou frio do ambiente, embora também seja afetado pelo nível de calor ou frio do Sense HAT: se estiver a usar uma caixa, poderá verificar que as suas leituras estão mais altas do que seria expectável. A Sense HAT não tem um sensor de temperatura em separado; em vez disso, usa sensores de temperatura incorporados nos sensores de humidade e de pressão barométrica. Um programa pode usar um ou ambos os sensores; a decisão é sua.

SENSE HAT NO RASPBERRY PI 400

A Sense HAT é totalmente compatível com o Raspberry Pi 400, e pode ser introduzido diretamente no conetor GPIO na traseira. No entanto, ao fazê-lo significa que os LEDs estarão virados na direção oposta a si e a placa estará virada de cabeça para baixo.

Para corrigi-lo, precisará de uma placa ou de um cabo de extensão GPIO. As extensões compatíveis incluem a gama Black HAT Hack3r da Pimoroni; pode utilizar a Sense HAT com a própria placa Black HAT Hack3r, ou basta usar o cabo plano de 40 pinos incluído como uma extensão. No entanto, verifique sempre as instruções do fabricante, para ter a certeza de que está a ligar o cabo e a Sense HAT da forma correta!

Instalar a Sense HAT

Se tiver um Sense HAT físico, comece por desembalá-lo e certifique-se de que tem todas as peças: deve ter o Sense Hat em si, quatro pilares de metal ou plástico conhecidos como *espaçadores*...e oito parafusos. Também pode ter alguns pinos de metal numa tira de plástico preto, como os pinos GPIO no Raspberry Pi; se sim, introduza esta tira com os pinos para cima através da parte inferior da Sense HAT até ouvir um clique.

Os espaçadores foram concebidos para impedir que a Sense HAT se dobre e flexione enquanto usa o joystick. Embora a Sense HAT funcione sem que estes sejam instalados, usá-los ajudará a proteger a sua Sense HAT, Raspberry Pi e conetor GPIO de serem danificados.

AVISO!

Os módulos Hardware Attached on Top (HAT) só devem ser ligados e removidos do conetor GPIO enquanto o seu Raspberry Pi estiver desligado e desconetado da respetiva fonte de alimentação. Tenha sempre o cuidado de manter o HAT plano quando o instalar, e verifique novamente se está alinhado com os pinos do conetor GPIO antes de os premir para baixo.

Instale os espaçadores empurrando para cima quatro dos parafusos por baixo da parte inferior do Raspberry Pi através dos quatro orifícios de montagem em cada canto, em seguida, aperte os espaçadores nos parafusos. Introduza a Sense HAT para baixo no conetor GPIO do Raspberry Pi, certificando-se de que o alinha corretamente com os pinos por baixo e que o mantém o mais plano possível. Por fim, aparafuse os quatro últimos parafusos através dos orifícios de montagem na Sense HAT e nos espaçadores que instalou anteriormente. Se for instalada corretamente, a Sense HAT deve estar plana e nivelada e não se deve dobrar ou abanar enquanto usa o respetivo joystick.

Ligue novamente a energia ao seu Raspberry Pi, e verá os LEDs da Sense HAT acenderem-se num padrão de arco-íris (**Figura 7-1**) e, em seguida, desligarem-se novamente. A sua Sense HAT já está instalada!

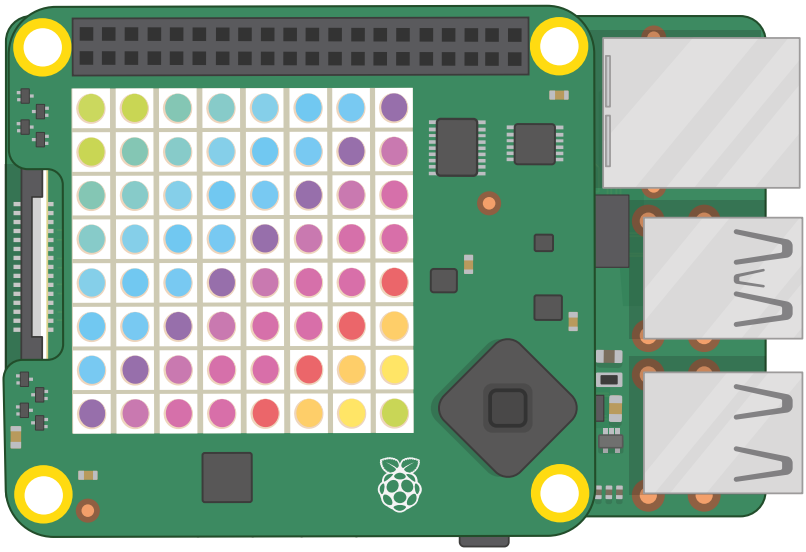


Figura 7-1: Aparece um padrão de arco-íris quando liga a energia pela primeira vez

Se quiser remover novamente a Sense HAT, basta desapertar os parafusos superiores, levantar a HAT, tendo cuidado para não dobrar os pinos no conetor GPIO, já que a HAT se mantém bem apertada (pode ser necessário forçar com uma pequena chave de fendas) e, em seguida, remova os espaçadores do Raspberry Pi.

Olá, Sense HAT!

Como em todos os projetos de programação, há um sítio óbvio para começar com a Sense HAT: deslocar uma mensagem de boas-vindas através do respetivo ecrã LED. Se estiver a usar o emulador Sense HAT, carregue-o agora clicando no ícone do menu Raspberry Pi OS, escolhendo a categoria Programação e clicando no emulador Sense HAT.

EXPERIÊNCIA EM PROGRAMAÇÃO

Este capítulo assume que tem experiência com o Scratch 3 ou o Python e o ambiente de desenvolvimento integrado (IDE) Thonny, dependendo se estiver a trabalhar com os códigos de exemplo do Scratch ou do Python, ou de ambos! Se ainda não o fez, regresse ao **Capítulo 4, Programar com o Scratch** ou ao **Capítulo 5, Programar com o Python** e trabalhe primeiro com os projetos nesse capítulo.

Saudações do Scratch

Carregue o Scratch 3 no Menu do Raspberry Pi OS. Clique no botão Adicionar Extensão na parte inferior esquerda da janela do Scratch. Clique na extensão Sense HAT do Raspberry Pi (**Figura 7-2**). Esta ação carrega os blocos de que necessita para controlar as várias funcionalidades da Sense HAT, incluindo o respetivo ecrã LED. Quando precisar destes, irá encontrá-los na categoria Sense HAT do Raspberry Pi.

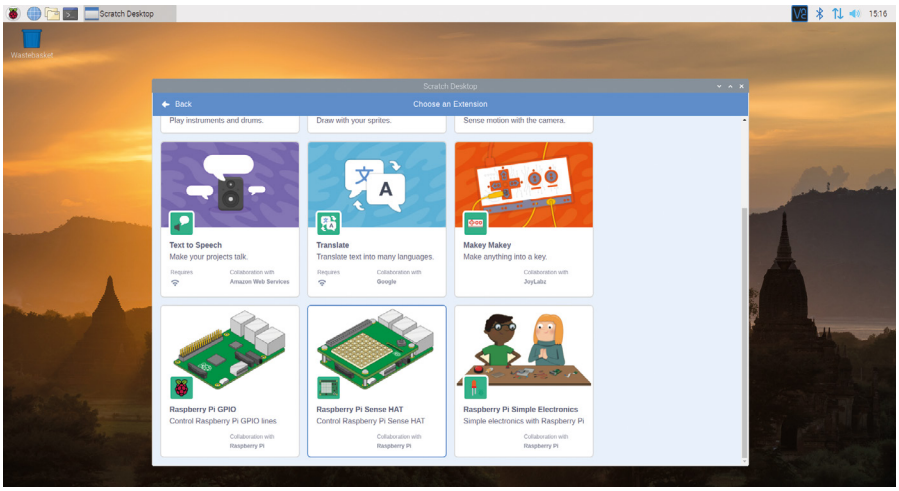
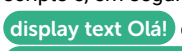


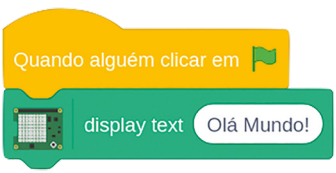
Figura 7-2: Adicionar a extensão Sense HAT do Raspberry Pi ao Scratch 3

Comece por arrastar um bloco de Eventos 

para a área de scripts e, em seguida, arraste um bloco 

diretamente para baixo deste. Edite o texto para que conste 

no bloco.



Clique na bandeira verde na área do palco e observe a sua Sense HAT ou o emulador da Sense HAT: a mensagem irá deslocar-se lentamente pela matriz LED da Sense HAT, iluminando os pixels LED para formar uma letra de cada vez (**Figura 7-3**...no verso). Parabéns: o seu programa é um sucesso!

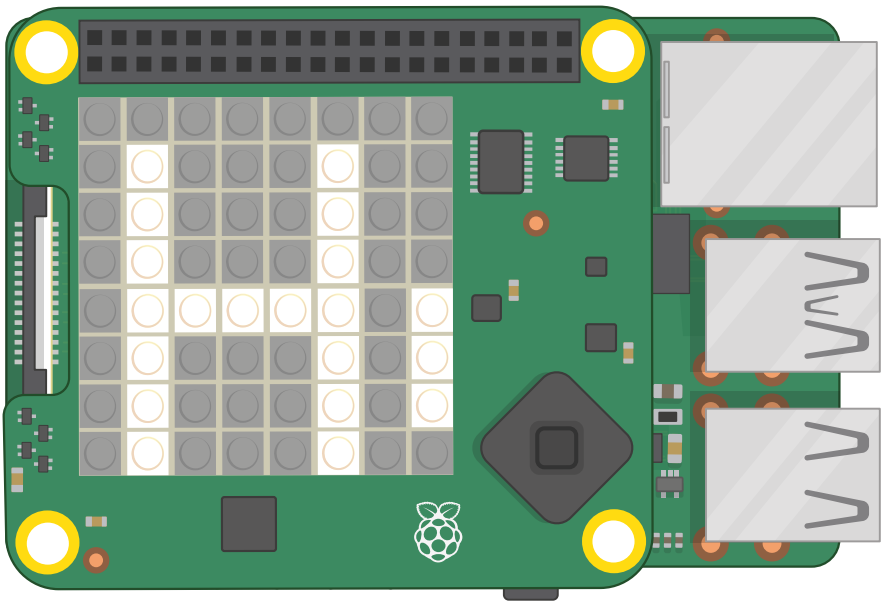


Figura 7-3: A sua mensagem desloca-se pela matriz LED

Já pode apresentar uma mensagem simples, está na hora de controlar a forma como essa mensagem será apresentada. Além de poder modificar a mensagem a ser apresentada, pode alterar a rotação, a forma como a mensagem é apresentada na Sense HAT. Arraste um bloco set rotation to 0 degrees da paleta de blocos e introduza-o debaixo de quando alguém clicar em e acima de display text Olá, Mundo!, em seguida, clique na seta para baixo ao lado de 0 e altere-o para 90. Clique na bandeira verde e verá uma mensagem igual à anterior , mas em vez de ser apresentada da esquerda para a direita, irá deslocar-se da parte inferior para a superior (**Figura 7-4**), vai precisar de virar a sua cabeça, ou a Sense HAT, para lê-la!

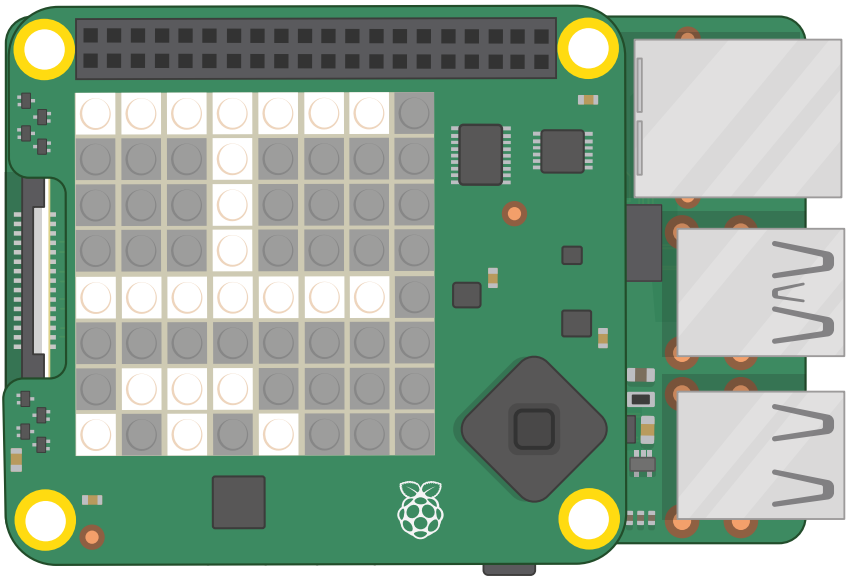


Figura 7-4: Desta vez a mensagem desloca-se verticalmente

Agora altere a rotação para 0 e, em seguida, arraste um bloco **set colour** entre **set rotation to 0 degrees** e **display text Olá, Mundo!**. Clique na cor no final do bloco para aceder ao seletor de cores do Scratch e encontrar uma bonita cor amarela brilhante, e, em seguida, clique na bandeira verde para ver como o resultado do seu programa se alterou (**Figura 7-5**).

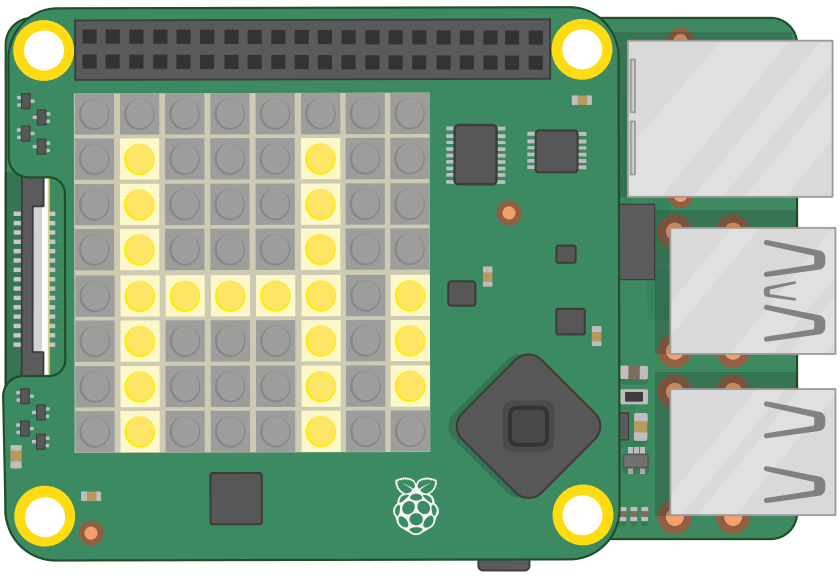
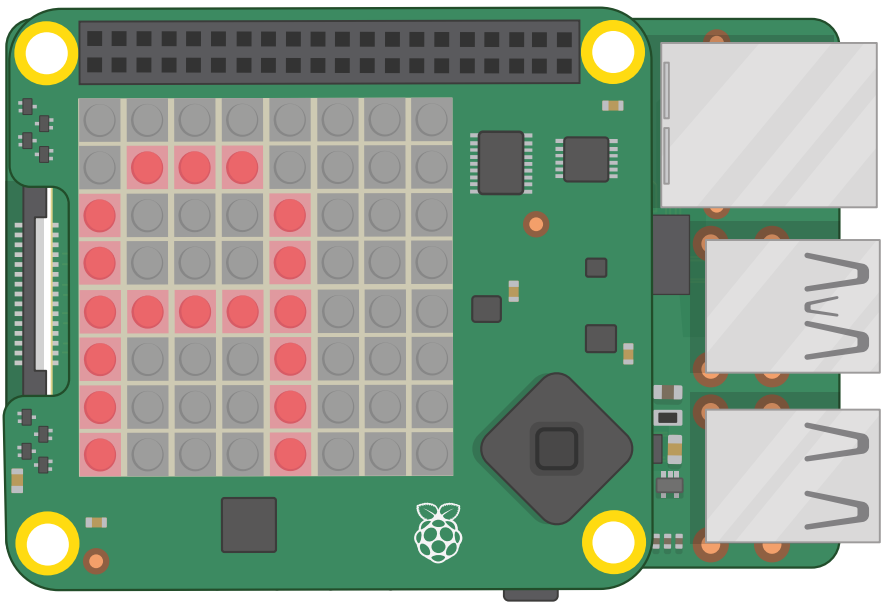


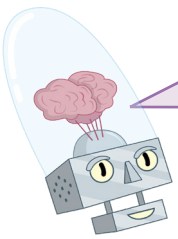
Figura 7-5: Alterar a cor do texto

Por fim, arraste um bloco **set background** entre **set colour to amarelo** e **display text Olá, Mundo!** e, em seguida, clique sobre a cor para aceder ao seletor de cores novamente. Desta vez, ao escolher uma cor não afetará os LEDs que compõem a mensagem, mas sim os LEDs que não o fazem, conhecidos como o fundo. Encontre uma bonita cor azul e, em seguida, clique novamente na bandeira verde: desta vez a sua mensagem estará num amarelo vivo sobre um fundo azul. Experimente alterar estas cores para encontrar a sua combinação favorita, nem todas as cores funcionam bem em conjunto!

Além de ser possível deslocar mensagens inteiras, pode também apresentar letras individuais. Arraste o seu bloco **display text** da área do script para eliminá-lo e em seguida, arraste um bloco **display character A** para a área de scripts no respetivo lugar. Clique na bandeira verde, e constatará a diferença: este bloco mostra apenas uma letra de cada vez, e a letra fica na Sense HAT até que lhe indique o contrário sem se deslocar ou desaparecer. A este bloco aplicam-se os mesmos blocos de controlo de cor que o bloco **display text**: experimente alterar a cor da letra para vermelho (**Figura 7-6**).



▲ **Figura 7-6:** Apresentação de uma única letra



DESAFIO: REPETIR A MENSAGEM

Consegue usar o seu conhecimento dos ciclos para fazer repetir uma mensagem que se desloca? Consegue fazer um programa que soletre uma palavra letra a letra com cores diferentes?

Saudações do Python

Carregue o Thonny; para isso, clique no menu do ícone da framboesa, selecione Programação e clique no Thonny. Se estiver a usar o emulador Sense HAT e este ficar coberto pela janela do Thonny, clique sem largar o botão do rato na barra de título de uma das janelas, na parte superior, em azul, e arraste-a para movê-la ao redor do ambiente de trabalho até que possa ver as duas janelas.

MUDANÇA DE LINHA NO PYTHON

O código Python escrito para um HAT físico é executado no emulador Sense HAT, e vice-versa, mas com uma diferença. Se estiver a usar o emulador Sense HAT com o Python, tem de mudar a linha de `sense_hat import SenseHat` em todos os programas deste capítulo para `from sense_emu import SenseHat` em vez dessa. Se quiser mais tarde executá-los num Sense HAT físico, basta mudar a linha para o que estava!

Para usar a Sense HAT, ou o emulador Sense HAT, num programa Python, tem de importar a biblioteca Sense HAT. Introduza o seguinte na área de scripts, sem se esquecer de usar `sense_emu`(em vez de `sense_hat`) se estiver a usar o emulador Sense HAT:

```
from sense_hat import SenseHat
sense = SenseHat()
```

A biblioteca Sense HAT tem uma função simples para receber uma mensagem, formatá-la para que possa ser apresentada no visor LED e deslocá-la suavemente. Escreva o seguinte:

```
sense.show_message("Olá, Mundo!")
```

Clique no botão Executar e guarde o seu programa como **Olá, Sense HAT**. Verá a sua mensagem deslocar-se lentamente na matriz LED da Sense HAT, iluminando os pixels LED para formar uma letra de cada vez (**Figura 7-7**...no verso). Parabéns: o seu programa é um sucesso!

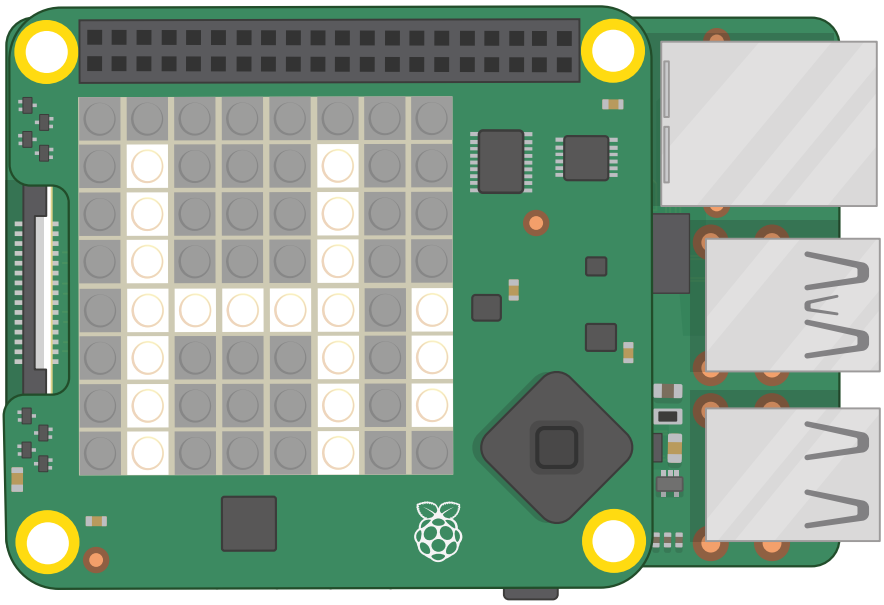


Figura 7-7: Deslocar uma mensagem pela matriz LED

A função **`show_message()`** tem ainda mais truques na manga. Volte para o seu programa e edite a última linha para que diga:

```
sense.show_message("Olá, Mundo!", text_colour=(255, 255, 0),
back_colour=(0, 0, 255), scroll_speed=(0.05))
```

Estas instruções adicionais, separadas por vírgulas, são conhecidas como *parâmetros* e controlam vários aspetos da função **`show_message()`**. A mais simples é **`scroll_speed=()`**, que altera a rapidez com que a mensagem se desloca pelo ecrã. Aqui, um valor de 0,05 desloca-se a cerca do dobro da velocidade habitual. Quanto maior for o número, menor é a velocidade.

Os parâmetros **`text_colour=()`** e **`back_colour=()`**, com a ortografia britânica, ao contrário da maioria das instruções do Python, definem a cor do texto e o fundo, respetivamente. Não aceitam nomes de cores, no entanto; tem de declarar a cor que quer como um trio de números. O primeiro número representa a quantidade de vermelho na cor, do 0 para nenhum vermelho até 255 para o máximo de vermelho possível; o segundo número é a quantidade de verde na cor; e o terceiro número a quantidade de azul. Juntos, estes são conhecidos como *RGB*, para vermelho, verde e azul.

Clique no ícone Executar e observe a Sense HAT: desta vez, a mensagem irá deslocar-se consideravelmente mais rápido, e estará num amarelo vivo sobre um fundo azul (**Figura 7-8**). Tente alterar os parâmetros para encontrar uma combinação de velocidade e cor que funcione para si.

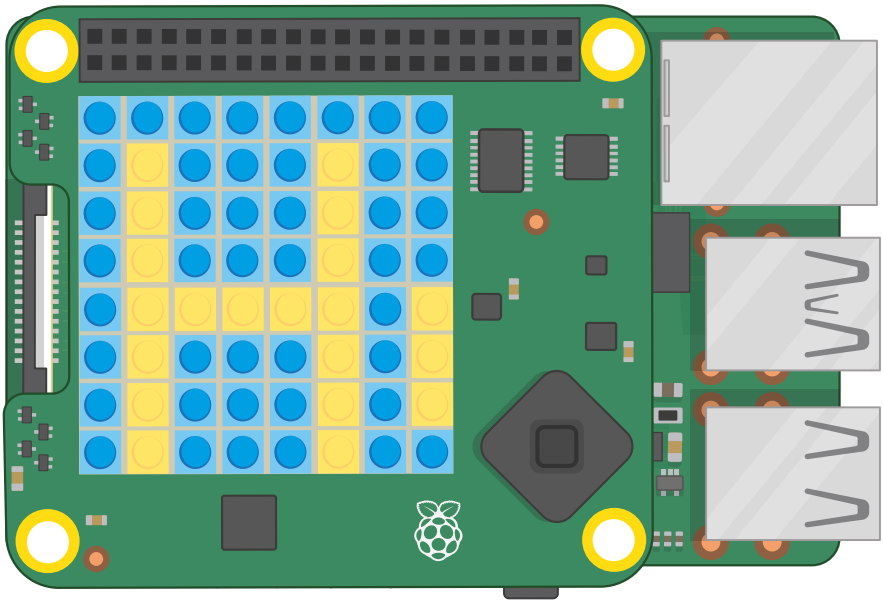


Figura 7-8: Alterar a cor da mensagem e do fundo

Se quiser usar nomes amigáveis em vez de valores RGB para definir as respetivas cores, terá de criar variáveis. Acima da sua linha `sense.show_message()` adicione o seguinte:

```
yellow = (255, 255, 0)
blue = (0, 0, 255)
```

Volte para a sua linha `sense.show_message()` e edite-a para que conste:

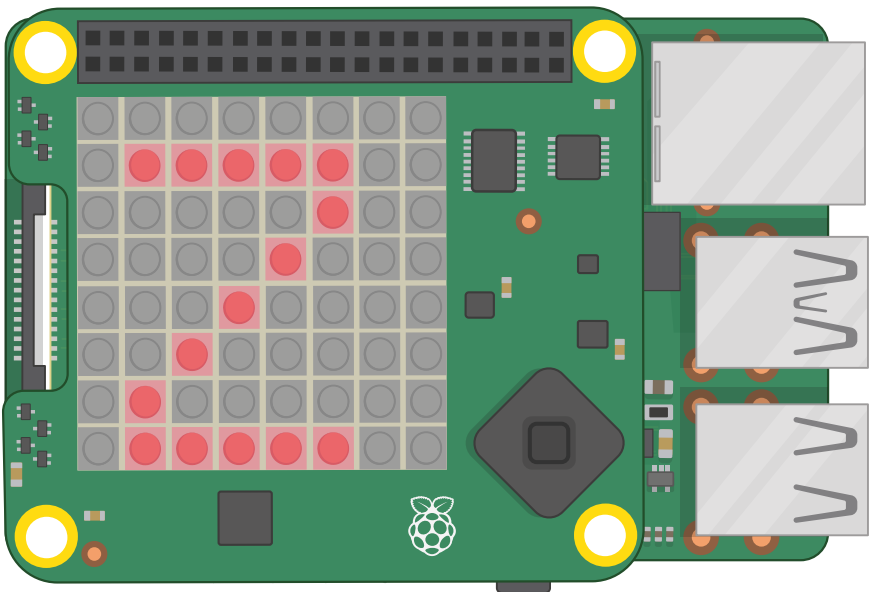
```
sense.show_message("Olá, Mundo!", text_colour=(yellow), back_
colour=(blue), scroll_speed=(0.05))
```

Clique novamente no ícone Executar e verá que nada mudou: a sua mensagem ainda está a amarelo sobre um fundo azul. Desta vez, no entanto, usou os nomes das variáveis para tornar o seu código mais legível: em vez de uma sequência de números, o código explica a cor que estiver a definir. Pode definir quantas cores quiser: tente adicionar uma variável chamada "vermelho" com os valores 255, 0 e 0; uma variável chamada "branco" com os valores 255, 255, 255; e uma variável chamada "preto" com os valores 0, 0 e 0.

Além de ser possível deslocar mensagens inteiras, pode também apresentar letras individuais. Elimine a sua linha `sense.show_message()` e introduza o seguinte no respetivo lugar:

```
sense.show_letter("Z")
```

Clique em Executar, e verá a letra "Z" aparecer no ecrã da Sense HAT. Desta vez, ficará lá: as letras individuais, ao contrário das mensagens, não se deslocam automaticamente. Também pode controlar **`sense.show_letter()`** com os mesmos parâmetros de cor de **`sense.show_message()`**Tente também mudar a cor da carta para vermelho (**Figura 7-9**).



▲ **Figura 7-9:** Apresentação de uma única letra

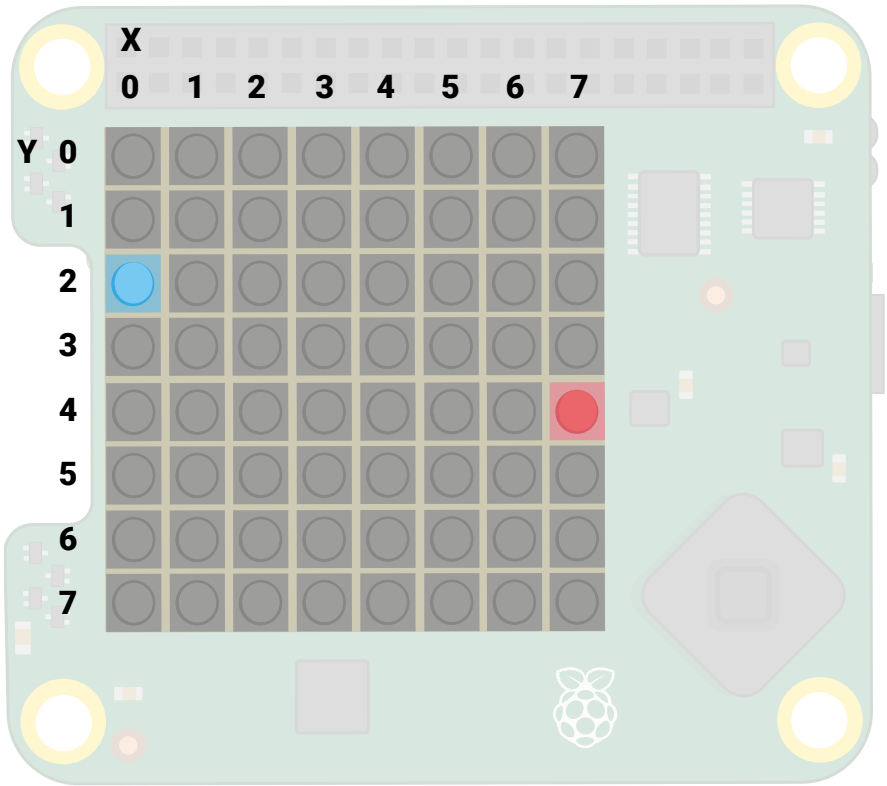
DESAFIO: REPETIR A MENSAGEM

Consegue usar o seu conhecimento dos ciclos para fazer repetir uma mensagem que se desloca? Consegue fazer um programa que soletre uma palavra letra a letra com cores diferentes? Quão rápido pode deslocar uma mensagem?

Próximos passos: Desenhar com luz

O ecrã LED da Sense HAT não serve apenas para mensagens: também pode apresentar imagens. Cada LED pode ser equiparado a um único pixel, abreviatura de *elemento de imagem*, numa imagem à sua escolha, que lhe permite tornar os seus programas mais interessantes com imagens ou até mesmo animação.

No entanto, para criar desenhos, tem de ter a possibilidade de alterar LEDs individuais. Para isso, terá de compreender o esquema da matriz LED da Sense HAT para escrever um programa que ligue ou desligue os LEDs corretos.



▲ **Figura 7-10:** Sistema de coordenadas da matriz LED

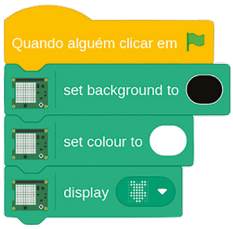
Existem oito LEDs em cada linha do ecrã, e oito em cada coluna (**Figura 7-10**). Ao contar os LEDs, no entanto, deve – como a maioria das linguagens de programação, começar no 0 e terminar no 7. O primeiro LED está no canto superior esquerdo, o último está no canto inferior direito. Com os números das linhas e colunas, pode encontrar as *coordenadas*de qualquer LED na matriz. O LED azul na matriz da imagem está nas coordenadas 0, 2; o LED vermelho está nas coordenadas 7, 4. O eixo X, ao longo da matriz, vem primeiro, seguido pelo eixo Y, descendo a matriz.

Ao planear imagens para desenhar na Sense HAT, pode ajudar desenhá-las à mão primeiro em papel quadriculado, ou pode fazer o planeamento numa folha de cálculo como o LibreOffice Calc.

Imagens no Scratch

Comece um novo projeto no Scratch, guardando o seu programa atual, se quiser mantê-lo. Se trabalhou nos projetos deste capítulo, o Scratch 3 irá manter a extensão Sense HAT do Raspberry Pi carregada; se fechou e voltou a abrir o Scratch 3 desde o seu último projeto, carregue a extensão com o botão Adicionar extensão. Arraste um bloco Eventos quando alguém clicar em para a área de programação e, em seguida, arraste o bloco set background e set colour para baixo

deste. Edite ambos para definir a cor de fundo para preto e a cor para branco: para obter a cor preta deslize os controlos de deslize Luminosidade e Saturação para 0; para obter a cor branca deslize a Luminosidade para 100 e Saturação para 0. Terá de fazer isso no arranque de todos os programas da Sense HAT, caso contrário o Scratch usará simplesmente as últimas cores que escolheu, mesmo que as tenha escolhido num programa diferente. Por fim, arraste um bloco display framboesa para a parte inferior do seu programa.



Clique na bandeira verde: verá os LEDs da Sense HAT acenderem na forma de uma framboesa (**Figura 7-11**).

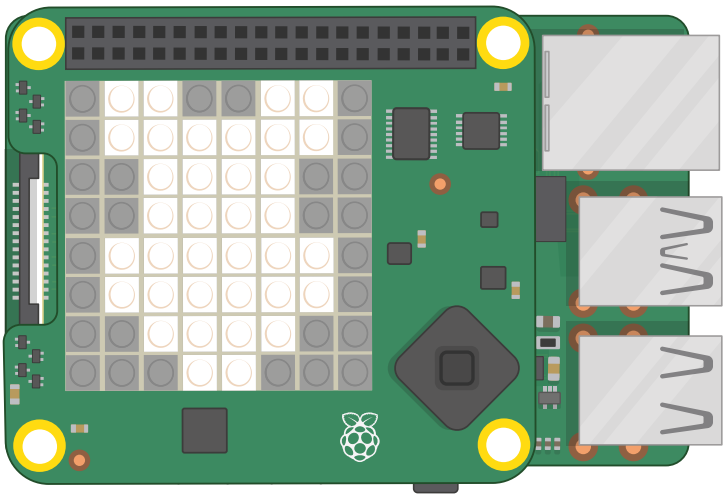
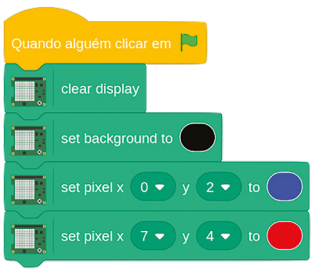


Figura 7-11: Não olhe diretamente para os LEDs enquanto estiverem branco brilhantes

Também não está limitado à forma predefinida da framboesa. Clique na seta para baixo ao lado da framboesa para ativar o modo de desenho: pode clicar em qualquer LED no padrão para ligá-lo ou desligá-lo, enquanto os dois botões na parte inferior desligam ou ligam todos os LEDs. Agora, tente desenhar o seu próprio padrão, em seguida, clique na seta verde para vê-lo na Sense HAT. Tente também alterar a cor e a cor do fundo com os blocos acima.

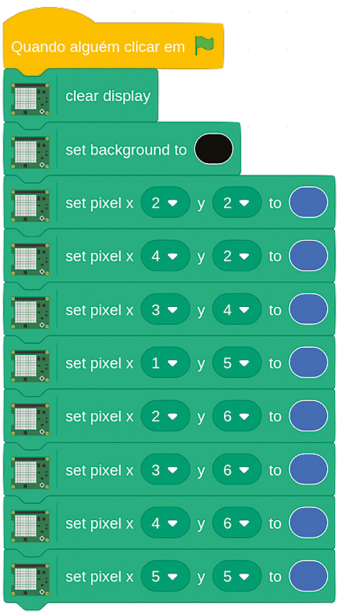
Quando terminar, arraste os três blocos para dentro da palete de blocos para eliminá-los e coloque um bloco clear display em quando alguém clicar em ; clique na bandeira verde, e todos os LEDs irão apagar-se.

Para criar uma imagem, tem de ter a possibilidade de controlar pixéis individuais e dar-lhes cores diferentes. Pode fazer isso encadeando blocos **display framboesa** editados com blocos **set colour**, ou pode controlar cada pixel individualmente. Para criar a sua própria versão do exemplo da matriz LED ilustrada no início desta secção, com dois LEDs especificamente selecionados iluminados a vermelho e azul, deixe o bloco **clear display** na parte superior do seu programa e arraste um bloco **set background** para baixo deste. Altere o bloco **set background** para a cor preta e, em seguida, arraste dois blocos **set pixel x 0 y 0** por baixo deste. Finalmente, edite estes blocos da seguinte forma:

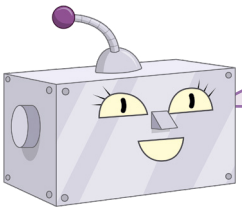


Clique na bandeira verde, e verá os seus LEDs acenderem-se para corresponderem à imagem da matriz (**Figura 7-10**) na página 165. Parabéns: consegue controlar LEDs individuais!

Edite o seu conjunto existente de blocos de pixéis da seguinte forma, e arraste mais para a parte inferior até obter oseguinte programa:



Antes de clicar na bandeira verde, veja se consegue adivinhar que imagem vai aparecer com base nas coordenadas da matriz LED que usou, em seguida execute o seu programa e confirme se está correto!



DESAFIO: NOVOS DESENHOS

Consegue desenhar mais imagens? Experimente primeiro planear a sua imagem à mão no papel quadriculado. Consegue desenhar uma imagem e fazer com que as cores mudem?

Imagens no Python

Inicie um novo programa no Thonny e guarde-o como um Desenho Sense HAT e, em seguida, introduza o seguinte, sem se esquecer de usar **`sense_emu`**(em vez de **`sense_hat`**) se estiver a usar o emulador:

```
from sense_hat import SenseHat
sense = SenseHat()
```

Não se esqueça de que precisa destas duas linhas no seu programa para poder usar a Sense HAT. Em seguida, introduza:

```
sense.clear(255, 255, 255)
```

Sem olhar diretamente para os LEDs do Sense HAT, clique no ícone Executar: deverá vê-los a todos tornarem-se da cor branco brilhante (**Figura 7-12**), e é por isso que não deve estar a olhar diretamente para eles quando executar o seu programa!

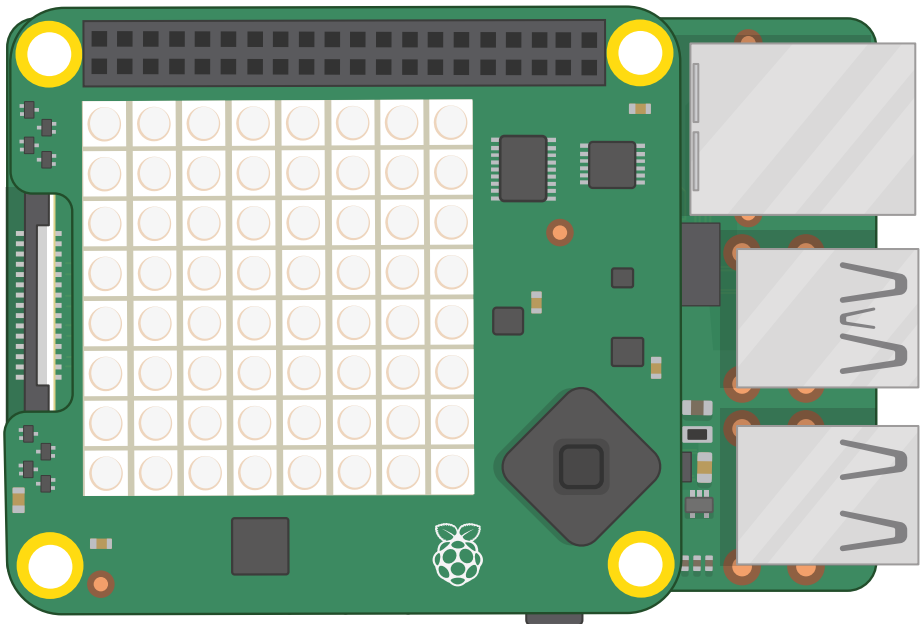


Figura 7-12: Não olhe diretamente para a matriz enquanto estiver iluminada a branco brilhante

O `sense.clear()` foi concebido para limpar os LEDs de qualquer programação anterior, mas aceita parâmetros de cor RGB, o que significa que pode mudar o ecrã para qualquer cor que quiser. Experimente editar a linha para:

```
sense.clear(0, 255, 0)
```

Clique em Executar, e a Sense HAT irá ficar verde brilhante (**Figura 7-13**...no verso). Experimente com cores diferentes, ou adicione as variáveis com o nome da cor que criou para o seu programa Hello World para facilitar aleitura.

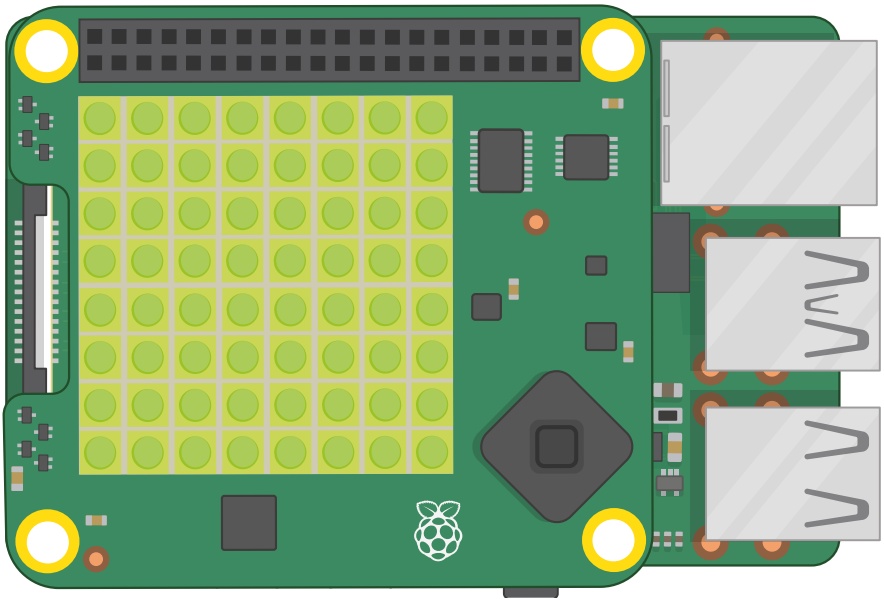


Figura 7-13: A matriz LED acende num verde brilhante

Para limpar os LEDs, tem de usar os valores RGB para o preto: 0 vermelho, 0 azul, e 0 verde. No entanto, há uma maneira mais fácil. Edite a linha do seu programa para dizer:

```
sense.clear()
```

A Sense HAT vai escurecer; na função `sense.clear()`não ter nada entre parênteses equivale a indicar-lhe para mudar todos os LEDS para preto, ou seja, desligá-los (**Figura 7-14**). Quando tem de limpar os LEDs nos seus programas por completo, essa é a função a ser usada.

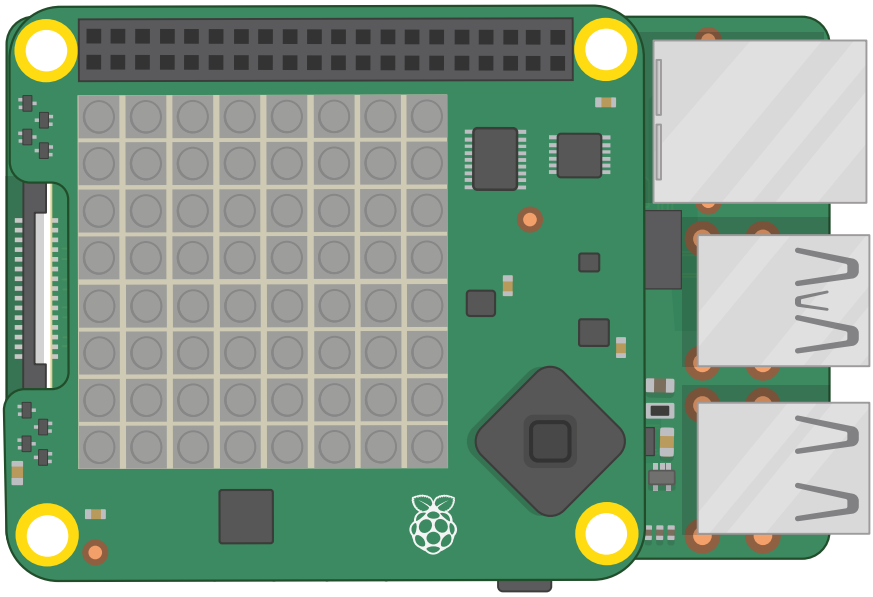


Figura 7-14: Use a função `sense.clear` para desligar todos os LEDs

Para criar a sua própria versão da matriz LED ilustrada anteriormente neste capítulo, com dois LEDs especificamente selecionados iluminados a vermelho e azul, adicione as seguintes linhas ao seu programa depois de **`sense.clear()`** :

```
sense.set_pixel(0, 2, (0, 0, 255))
sense.set_pixel(7, 4, (255, 0, 0))
```

O primeiro par de números é a localização do pixel na matriz, eixo X (horizontal) seguido pelo eixo Y (vertical). O segundo, no seu próprio conjunto de parênteses, são os valores RGB para a cor do pixel. Clique no botão Executar e verá o efeito: acenderão dois LEDs na sua Sense HAT, assim como na **Figura 7-10**na página 165.

Elimine essas duas linhas e introduza o seguinte:

```
sense.set_pixel(2, 2, (0, 0, 255))
sense.set_pixel(4, 2, (0, 0, 255))
sense.set_pixel(3, 4, (100, 0, 0))
sense.set_pixel(1, 5, (255, 0, 0))
sense.set_pixel(2, 6, (255, 0, 0))
sense.set_pixel(3, 6, (255, 0, 0))
sense.set_pixel(4, 6, (255, 0, 0))
sense.set_pixel(5, 5, (255, 0, 0))
```

Antes de clicar em Executar, veja as coordenadas e compare-as à matriz: consegue adivinhar qual é a imagem que essas instruções vão desenhar? Clique em Executar para descobrir se tem razão!

Desenhar uma imagem detalhada com funções **`set_pixel()`** individuais é, no entanto, um processo lento. Para acelerar as coisas, pode alterar vários pixels ao mesmo tempo. Elimine todas as suas linhas **`set_pixel()`** e introduza o seguinte:

```
g = (0, 255, 0)
b = (0, 0, 0)

creeper_pixels = [
    g, g, g, g, g, g, g, g,
    g, g, g, g, g, g, g, g,
    g, b, b, g, g, b, b, g,
    g, b, b, g, g, b, b, g,
    g, g, g, b, b, g, g, g,
    g, g, b, b, b, b, g, g,
    g, g, b, b, b, b, g, g,
    g, g, b, g, g, b, g, g
]

sense.set_pixels(creeper_pixels)
```

Há muitos elementos, mas comece por clicar em Executar para ver se reconhece um certo pequeno creeper. As primeiras duas linhas criam duas variáveis para manter as cores: verde e preto. Para facilitar a escrita e leitura do código do desenho, as variáveis são apenas uma letra: "g" para verde e "b" para preto.

O próximo bloco de código cria uma variável que contém valores de cor para todos os 64 pixels da matriz LED, separados por vírgulas e entre parênteses retos. No entanto, em vez de números, utiliza as variáveis de cor que criou anteriormente: olhe com atenção, sem se esquecer que "g" é para o verde e "b" é para o preto, e já conseguirá ver a imagem que vai aparecer (**Figura 7-15**).

Por fim, **`sense.set_pixels(creeper_pixels)`** assume essa variável e usa a **`sense.set_pixels()`** para desenhar a matriz toda de uma só vez. Muito mais fácil do que tentar desenhar pixel a pixel!

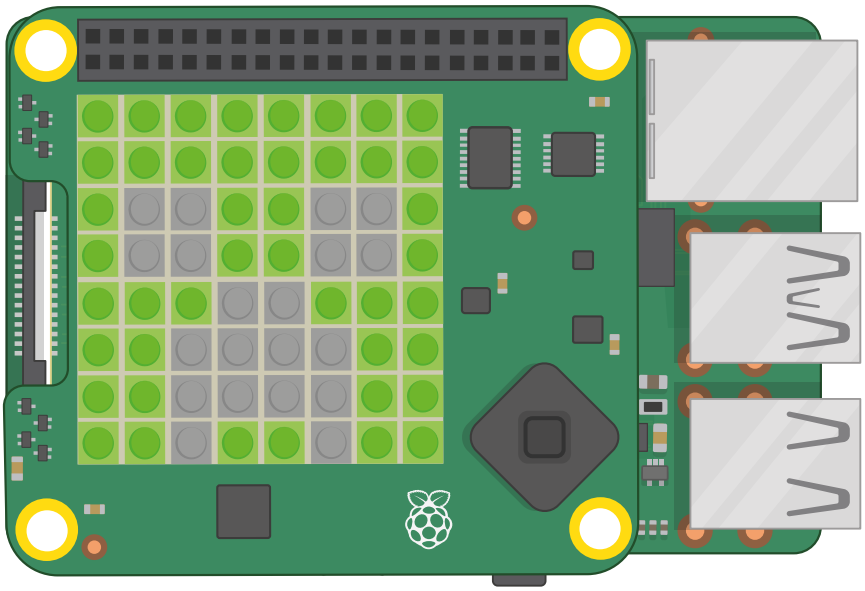


Figura 7-15: Apresentar uma imagem na matriz

Também pode rodar e inverter imagens, como uma forma de mostrar as imagens na posição certa quando a sua Sense HAT roda ou como uma forma de criar animações simples a partir de uma única imagem assimétrica.

Comece por editar o seu `creeper_pixels` para fechar o seu olho esquerdo; para isso, substitua os quatro pixels "b", começando pelos dois primeiros na terceira linha e, em seguida, os dois primeiros na quarta linha, com"g":

```
creeper_pixels = [
    g, g, g, g, g, g, g, g,
    g, g, g, g, g, g, g, g,
    g, g, g, g, g, b, b, g,
    g, g, g, g, g, b, b, g,
    g, g, g, b, b, g, g, g,
    g, g, b, b, b, b, g, g,
    g, g, b, b, b, b, g, g,
    g, g, b, g, g, b, g, g
]
```

Clique em Executar, e verá o olho esquerdo do creeper fechar (**Figura 7-16**, no verso). Para fazer uma animação, aceda à parte superior do seu programa e adicione a linha:

```
from time import sleep
```

Em seguida aceda à parte inferior e introduza:

```
while True:
    sleep(1)
    sense.flip_h()
```

Clique em Executar, e observe o creeper a fechar e a abrir os olhos, um de cada vez!

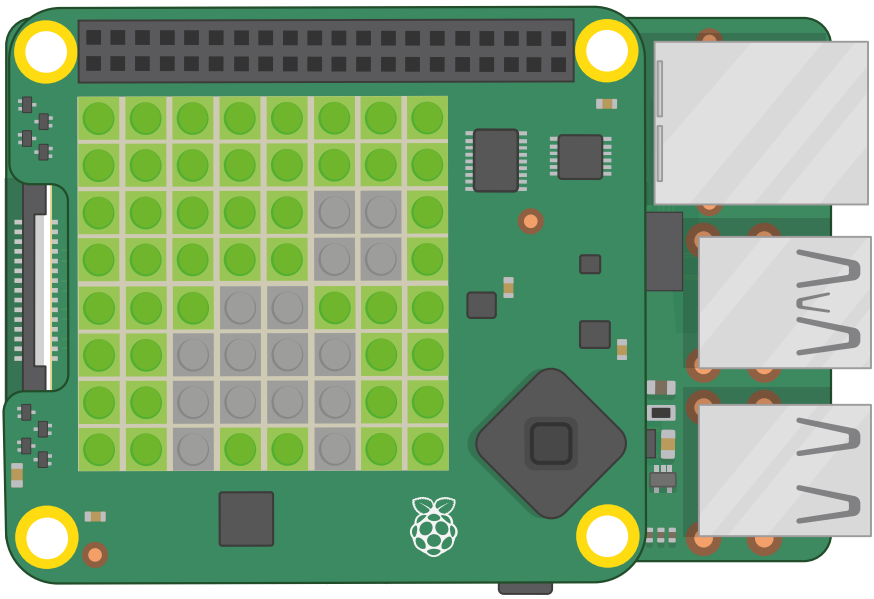
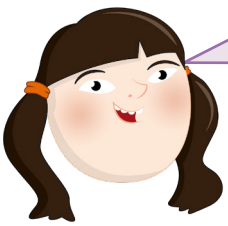


Figura 7-16: Mostrar uma animação simples de dois quadros

A função **`flip_h()`** inverte uma imagem no eixo horizontal, longitudinalmente; se quiser inverter uma imagem no respetivo eixo vertical, substitua **`sense.flip_h()`** por **`sense.flip_v()`** em alternativa. Também pode rodar uma imagem em 0, 90, 180 ou 270 graus com a função **`sense.set_rotation(90)`**, alterando o número de acordo com a quantidade de graus que quer rodar a imagem. Experimente usar esta função para fazer o creeper rodar em volta de si próprio em vez de piscar os olhos!

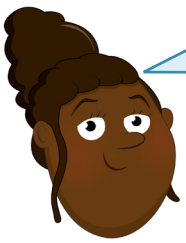


DESAFIO: NOVOS DESENHOS

Consegue desenhar mais imagens e animações? Experimente primeiro planear a sua imagem à mão no papel quadriculado para que seja mais fácil escrever a variável. Consegue desenhar uma imagem e fazer com que as cores mudem? Sugestão: pode alterar as variáveis depois de já as ter usado uma vez.

Sentir o mundo à sua volta

O verdadeiro poder da Sense HAT reside nos vários sensores que possui. Estes permitem-lhe fazer leituras de tudo, desde a temperatura até à aceleração, e utilizá-los nos seus programas da forma que achar melhor.



EMULAR OS SENSORES

Se estiver a usar o emulador Sense HAT, terá de ativar a simulação inercial e ambiental do sensor: no emulador, clique em Edit, em Preferences e, em seguida, marque-as. No mesmo menu escolha "180°..360°|0°..180°" na "Orientation Scale" para se certificar de que os números no emulador correspondem aos números comunicados pelo Scratch e pelo Python e, em seguida, clique no botão Fechar.

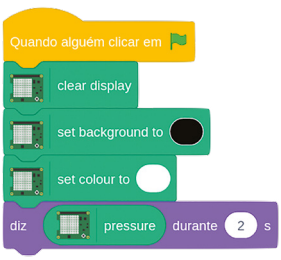
Deteção ambiental

O sensor de pressão barométrica, o sensor de humidade e o sensor de temperatura são todos sensores ambientais; fazem medições a partir do ambiente envolvente à Sense HAT.

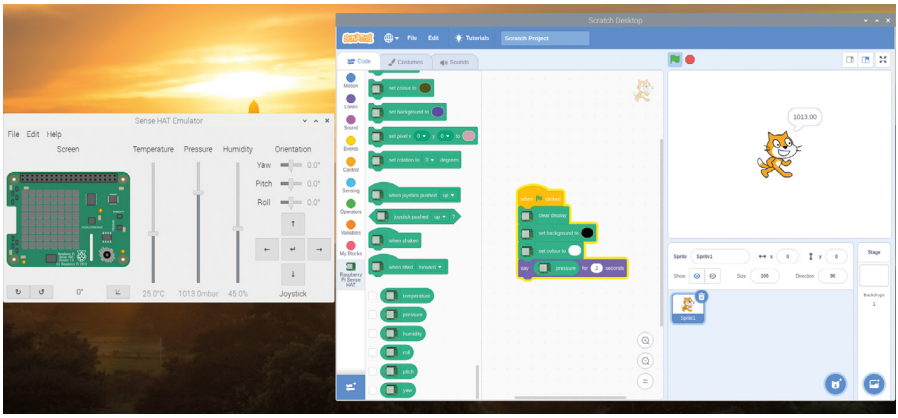
Deteção ambiental no Scratch

Inicie um novo programa no Scratch, guardando o antigo se quiser e adicione a extensão Sense HAT do Raspberry Pi se ainda não estiver carregada. Arraste um bloco Eventos quando alguém clicar em para a sua área de programação e, em seguida, um bloco clear display para baixo e um bloco set background to preto para baixo deste. A seguir, adicione um bloco set colour to branco, use os controlos de deslize de Luminosidade e Saturação para selecionar a cor correta. É sempre uma boa ideia fazer esta ação no início dos seus programas, porque irá garantir que a Sense HAT não está a mostrar nada de um programa antigo, garantindo ao mesmo tempo as cores que está a usar.

Arraste um bloco diz Olá! durante 2 s diretamente para baixo dos seus blocos atuais. Para fazer uma leitura do sensor de pressão, encontre o bloco pressure na categoria Sense HAT do Raspberry Pi e arraste-o sobre a palavra "Hello" no seu bloco diz Olá! durante 2 s.



Clique na bandeira verde e o gato do Scratch irá indicar-lhe a leitura atual do sensor de pressão em *milibares*. Após dois segundos, a mensagem desaparecerá; experimente soprar na Sense HAT (ou mover o controlo de deslize da Pressão do emulador para cima) e clique na bandeira verde para executar o programa novamente; deve ver uma leitura mais alta desta vez (**Figura 7-17**, no verso).

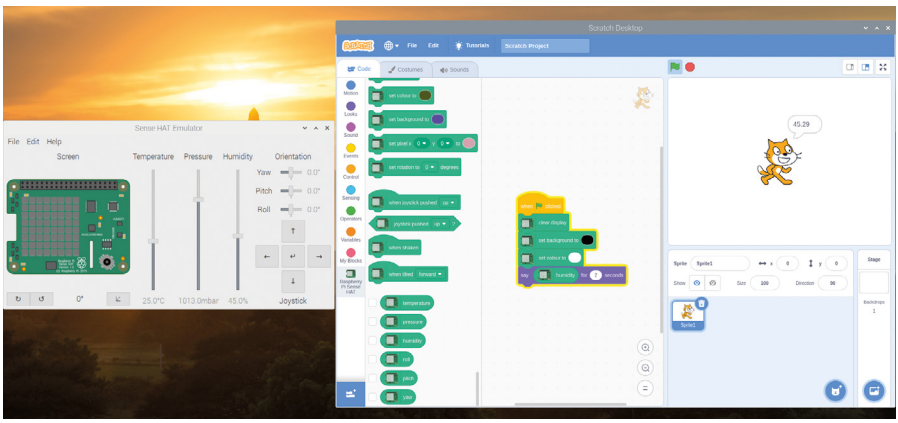


▲ **Figura 7-17:** A mostrar a leitura do sensor de pressão

ALTERAR VALORES

Se estiver a usar o emulador Sense HAT, pode alterar os valores comunicados por cada um dos sensores emulados usando os respetivos controlos de deslize e os botões. Tente deslizar o sensor de pressão para baixo em direção à parte inferior em seguida, clique na bandeira verde novamente.

Para mudar para o sensor de humidade, elimine o bloco `pressure` e substitua-o por `humidity`. Execute seu programa novamente, e verá a humidade relativa atual do seu quarto. Novamente, pode tentar executá-lo novamente enquanto sopra na Sense HAT (ou movendo o controlo de deslize da Humidade do emulador para cima) para alterar a leitura (**Figura 7-18**), o seu hálito é surpreendentemente húmido!



▲ **Figura 7-18:** Exibir a leitura a partir do sensor de humidade

Para o sensor de temperatura, é tão fácil como apagar o bloco humidity e substituí-lo por temperature e, em seguida, executar o seu programa novamente. Verá uma temperatura em graus Celsius (**Figura 7-19**). Esta pode não ser a temperatura exata do seu quarto, no entanto: O Raspberry Pi está sempre a gerar calor, e esta ação aquece a Sense HAT e os respetivos sensores também.

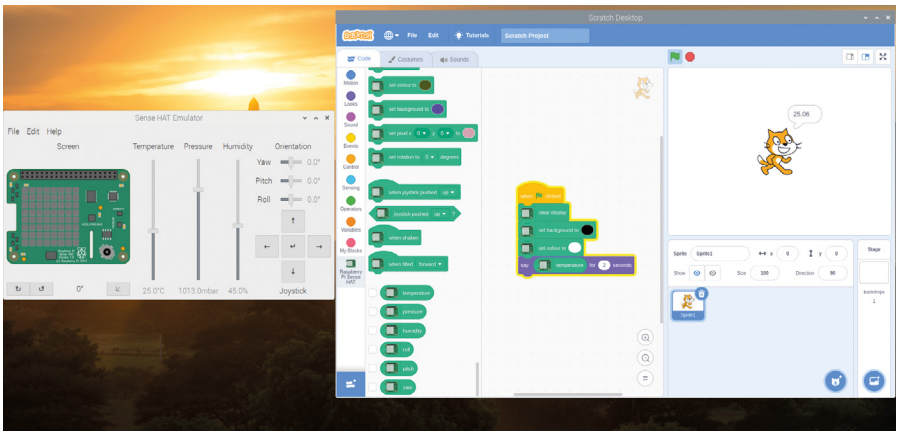
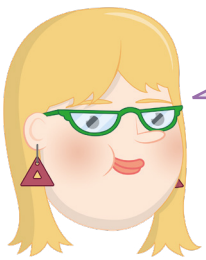


Figura 7-19: Apresenta a leitura do sensor de temperatura



DESAFIO: DESLOCAR E CICLO

Pode alterar o seu programa para tirar uma leitura de cada um dos sensores de cada vez, em seguida, desloque-os pela matriz LED, em vez de imprimi-los na área do palco? Consegue fazer um ciclo ao seu programa, para que esteja sempre a imprimir as condições ambientais atuais?

Deteção ambiental no Python

Para começar a tirar leituras dos sensores, crie um novo programa no Thonny e guarde-o como **Sensores Sense HAT**. Introduza o seguinte na área de scripts, como deve sempre fazer quando estiver a usar a Sense HAT, e não se esqueça de usar **`sense_emu`** se estiver a usar o emulador:

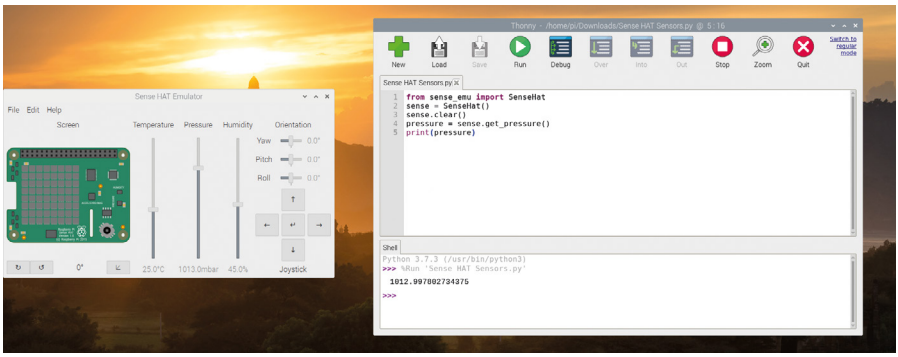
```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()
```

É sempre uma boa ideia incluir **`sense.clear()`** ao iniciar os seus programas, para o caso do ecrã da Sense HAT ainda mostrar algo do último programa que executou.

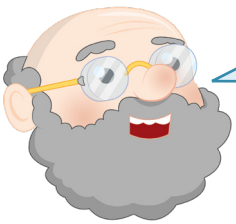
Para efetuar uma leitura do sensor de pressão, introduza:

```
pressure = sense.get_pressure()
print(pressure)
```

Clique em Executar e verá um número impresso na Shell do Python na parte inferior da janela do Thonny. Esta é a leitura da pressão de ar detetada pelo sensor de pressão barométrica, em *milibares* (**Figura 7-20**). Experimente soprar na Sense HAT (ou mover o controlo de deslize da Pressão do emulador para cima) enquanto clica no ícone executar novamente; o número deve ser mais alto desta vez.



▲ **Figura 7-20:** A imprimir uma leitura de pressão do Sense HAT



ALTERAR VALORES

Se estiver a usar o emulador Sense HAT, pode alterar os valores comunicados por cada um dos sensores emulados usando os respetivos controlos de deslize e os botões. Experimente deslizar a definição do sensor de pressão para a parte inferior e, em seguida, clique em Executar novamente.

Para mudar para o sensor de humidade, remova as duas últimas linhas de código e substitua-as por:

```
humidity = sense.get_humidity()
print(humidity)
```

Clique em Executar e verá outro número impresso na shell do Python: desta vez, é a humidade relativa atual do seu quarto como percentagem. Novamente, pode soprar na Sense HAT (ou mover o controlo de deslize da Humidade para cima) e verá o mesmo a ir para cima quando executar o seu programa novamente (**Figura 7-21**), o seu hálito é surpreendentemente húmido!

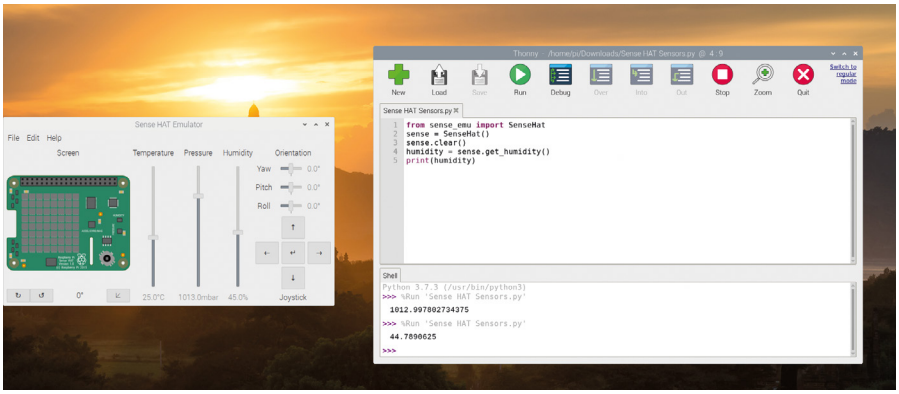


Figura 7-21: Apresenta a leitura do sensor de humidade

Para o sensor de temperatura, remova as últimas duas linhas do seu programa e substitua-as por:

```
temp = sense.get_temperature()
print(temp)
```

Clique em Executar novamente, e verá uma temperatura em graus Celsius (**Figura 7-22**). Esta pode não ser a temperatura exata do seu quarto, no entanto: O Raspberry Pi está sempre a gerar calor, e esta ação aquece a Sense HAT e os respetivos sensores também.

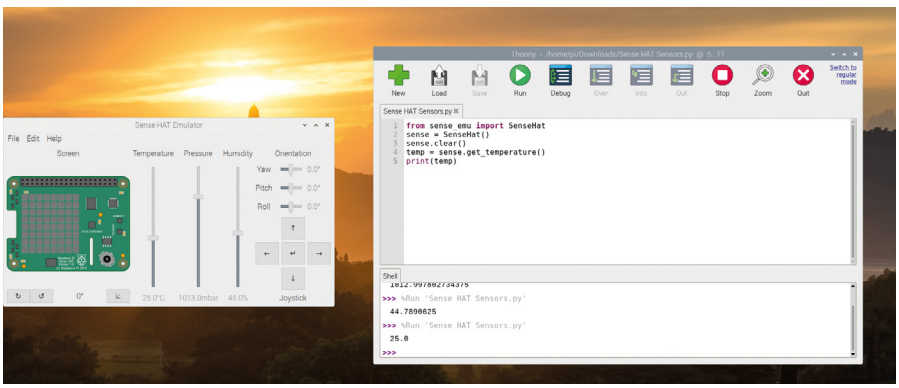
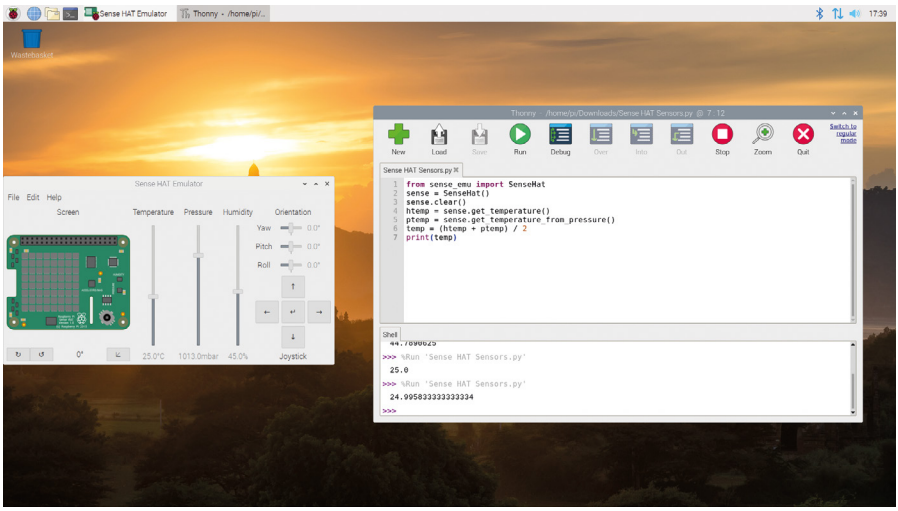


Figura 7-22: A mostrar a leitura da temperatura atual

Normalmente a Sense HAT comunica a temperatura com base numa leitura do sensor de temperatura incorporado no sensor de humidade; se em vez disso quiser usar a leitura do sensor de pressão, deve usar **`sense.get_temperature_from_pressure()`**. Também é possível combinar as duas leituras para obter uma média, que pode ser mais precisa do que usar qualquer um dos sensores sozinho. Elimina as últimas duas linhas do seu programa e introduza:

```
htemp = sense.get_temperature()
ptemp = sense.get_temperature_from_pressure()
temp = (htemp + ptemp) / 2
print(temp)
```

Clique no ícone Executar, e verá um número impresso na consola do Python (**Figura 7-23**). Desta vez, é baseado em leituras de ambos os sensores, as quais somou e dividiu por dois, o número de leituras, para obter uma média de ambos. Se estiver a usar o emulador, todos os três métodos, humidade, pressão e a média, mostrarão o mesmo número.



▲ **Figura 7-23:** Uma temperatura baseada nas leituras de ambos os sensores

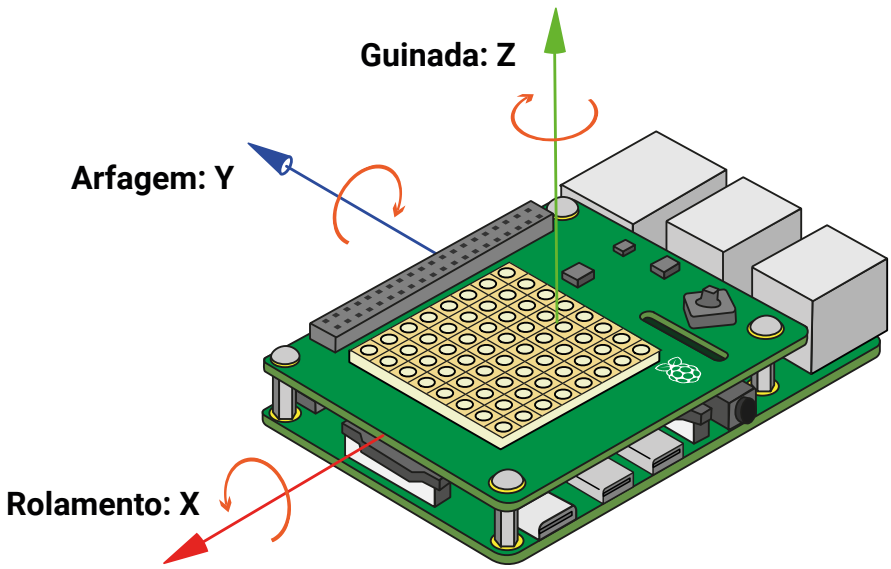
DESAFIO: DESLOCAR E CICLO

Pode alterar o seu programa para tirar uma leitura de cada um dos sensores de cada vez, em seguida, desloque-os pela matriz LED, em vez de imprimi-los na shell? Consegue fazer um ciclo ao seu programa, para que esteja sempre a imprimir as condições ambientais atuais?

Deteção por inércia

O sensor giroscópico, acelerómetro e magnetómetro combinam-se para formar o que é conhecido como uma *unidade de medição por inércia (IMU)*. Embora, tecnicamente, estes sensores façam medições do ambiente ao redor tal como os sensores ambientais, o magnetómetro, por exemplo, mede a intensidade do campo magnético, são normalmente usados para obter dados sobre o movimento da própria Sense HAT. A IMU é a soma de vários sensores; algumas linguagens de programação permitem que tire leituras de cada sensor independentemente, enquanto outras apenas lhe darão uma leitura combinada.

Mas antes que possa compreender a IMU, precisa de compreender como as coisas se movem. A Sense HAT, e o seu Raspberry Pi à qual está ligada, pode mover-se ao longo de três eixos espaciais: lateralmente no eixo X; para a frente e para trás no eixo Y; e para cima e para baixo no eixo Z (**Figura 7-24**). Esta pode também rodar ao longo destes três mesmos eixos, mas os respetivos nomes mudam: a rotação no eixo X é chamada de *rolamento*, a rotação no eixo Y é chamada de *arfagem* e a rotação no eixo Z é chamada de *guinada*. Quando roda a Sense HAT ao longo do respetivo eixo curto, está a ajustar a respetiva arfagem; rode-a ao longo do respetivo eixo longo e é uma volta; rode-a à volta de si própria enquanto a mantém plana sobre a mesa e está a ajustar a respetiva guinada. Pense neles como um avião: quando está a descolar, aumenta a respetiva arfagem para subir; quando está a fazer uma volta de vitória, está literalmente a girar ao longo do respetivo eixo de rotação; quando está a usar o respetivo leme para virar como um carro faria, sem rolar, é uma guinada.

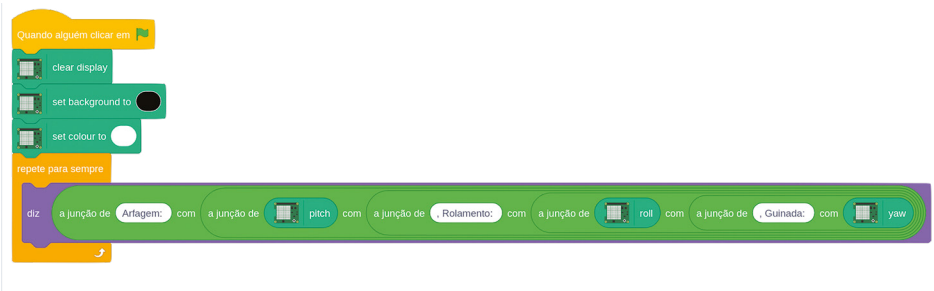


▲ **Figura 7-24:** Os eixos espaciais do IMU da Sense HAT

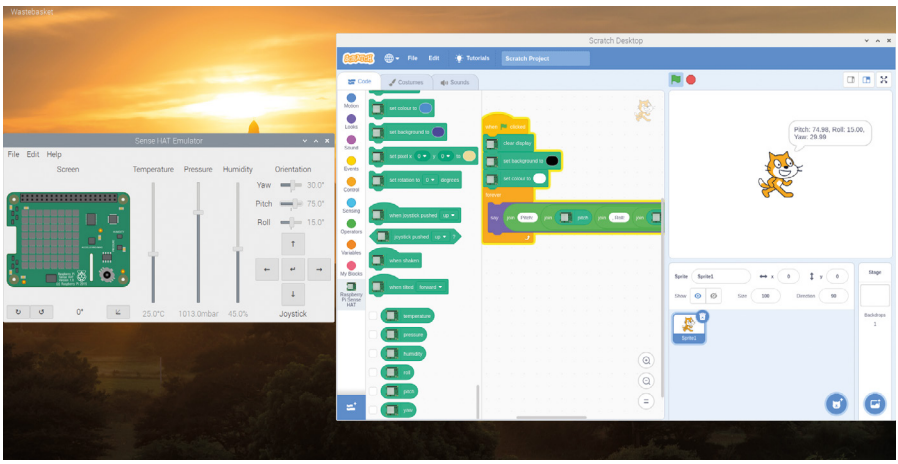
Deteção por inércia no Scratch

Inicie um novo programa no Scratch e carregue a extensão Sense HAT do Raspberry Pi, se ainda não estiver carregada. Inicie o seu programa da mesma forma que anteriormente: arraste um bloco Eventos quando alguém clicar em para a sua área de programação em seguida, arraste um bloco clear display por baixo e, em seguida, arraste e edite um bloco set background to preto e um bloco set colour to branco.

Por fim, arraste um bloco sempre para a parte inferior dos seus blocos atuais e preencha-o com um bloco diz Olá!. Para mostrar uma leitura de cada um dos três eixos da IMU - arfagem, volta, e guinada, terá de adicionar junção Blocos operador mais os blocos Sense HAT correspondentes. Não se esqueça de incluir espaços e vírgulas, para que o resultado seja de fácil leitura.



Clique na bandeira verde para executar o seu programa e experimente reorganizar a disposição da Sense HAT e do Raspberry Pi, tendo cuidado para não retirar nenhum cabo! Enquanto inclina a Sense HAT através dos seus três eixos, verá os valores de arfagem, volta e guinada mudarem de acordo (**Figura 7-25**).



▲ **Figura 7-25:** Apresenta os valores de arfagem, volta e guinada

Deteção por inércia no Python

Inicie um novo programa no Thonny e guarde-o como **Sensores Sense HAT**. Preencha as linhas iniciais habituais, sem se esquecer de usar **`sense_emu`** se estiver a usar o emulador Sense HAT:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()
```

Para utilizar a informação da IMU para descobrir a orientação atual da Sense HAT nos seus três eixos, introduza o seguinte:

```
orientation = sense.get_orientation()
pitch = orientation["pitch"]
roll = orientation["roll"]
yaw = orientation["yaw"]
print("arfagem {0} volta {1} guinada {2}".format(pitch, roll, yaw))
```

Clique em Executar e verá as leituras para a orientação da Sense HAT em separado nos três eixos (**Figura 7-26**). Tente rodar a Sense HAT e clique em Executar novamente; deve ver os números alterarem para refletir a sua nova orientação.

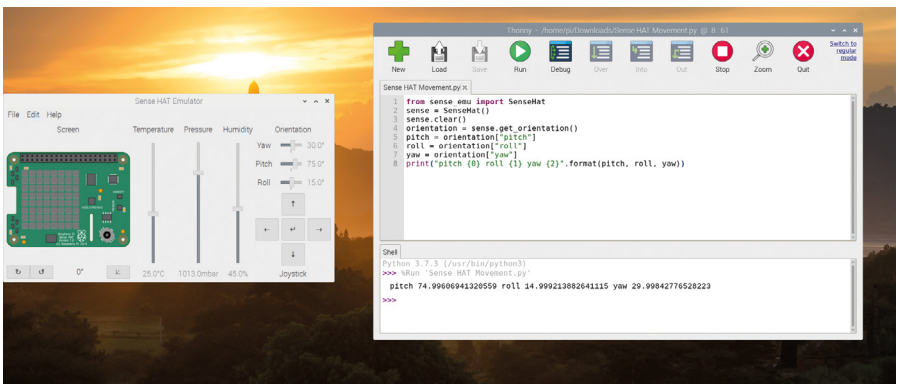


Figura 7-26: A mostrar os valores de arfagem, volta e guinada da Sense HAT

No entanto, para além da IMU poder fazer mais do que medir a orientação: também pode detetar movimento. Para obter leituras precisas para o movimento, é necessário tirar leituras à IMU frequentemente num ciclo: ao contrário da orientação, tirar uma única leitura não lhe dará nenhuma informação útil quando for necessário detetar movimento. Apaga tudo a seguir a **`sense.clear()`** e, em seguida, introduza o seguinte código:

```
while True:
    acceleration = sense.get_accelerometer_raw()
    x = acceleration["x"]
    y = acceleration["y"]
    z = acceleration["z"]
```

Tem agora variáveis que contêm as leituras atuais do acelerómetro para os três eixos espaciais: X, ou esquerda e direita; Y, ou para a frente e para trás; e Z, ou para cima ou para baixo. Os números do sensor do acelerómetro podem ser difíceis de interpretar, por isso pode introduzir o seguinte para facilitar a respetiva compreensão arredondando-os para o número inteiro mais próximo:

```
    x = round(x)
    y = round(y)
    z = round(z)
```

Por fim, imprima os três valores; para isso, escreva a seguinte linha:

```
    print("x={0}, y={1}, z={2}".format(x, y, z))
```

Clique em Executar e verá os valores do acelerómetro impressos na área da shell do Python (**Figura 7-27**). Ao contrário do seu programa anterior, estes irão imprimir continuamente; para parar de imprimi-los, clique no botão vermelho de paragem para parar o programa.

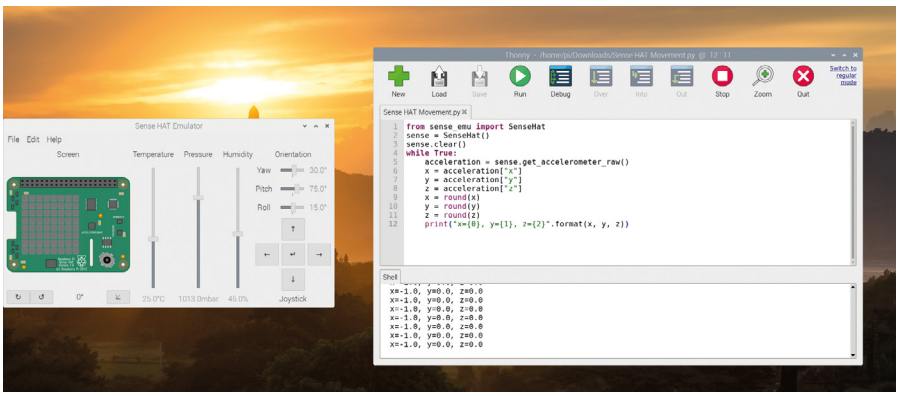


Figura 7-27: Leituras do acelerómetro arredondadas para o número inteiro mais próximo

Poderá ter-se apercebido de que o acelerómetro está a indicar-lhe que um dos eixos, o eixo Z, se o seu Raspberry Pi estiver sobre a mesa, tem um valor de aceleração de 1,0 gravidades (1G), mas a Sense HAT não se está a mover. Esta situação acontece porque está a detetar a força gravitacional da Terra, a força que está a puxar a Sense HAT para baixo em direção ao

centro da Terra, e a razão pela qual se derrubar algo da sua secretária isso cairá para o chão.

Com o seu programa em execução, tente pegar cuidadosamente na Sense HAT e no Raspberry Pi e rode-os em volta deles próprios, mas certifique-se de que não retira nenhum dos respetivos cabos! Com a rede Raspberry Pi e as portas USB na direção do chão, verá os valores alterarem para que no eixo Z conste 0G e no eixo X conste agora 1G; vire-o novamente para que as portas HDMI e a de alimentação estejam na direção do chão e agora é no eixo Y que consta 1G. Se fizer o oposto e tiver a porta HDMI virada para cima, verá em vez disso -1G no eixo Y.

Com o conhecimento de que a gravidade da Terra é cerca de 1G, e o seu conhecimento dos eixos espaciais, pode usar as leituras do acelerómetro para descobrir qual a direção que vai para baixo, e, da mesma forma, qual a direção que vai para cima. Também pode usá-lo para detetar movimentos: tente abanar cuidadosamente a Sense HAT e o Raspberry Pi, e observe os números à medida que o faz: quanto mais forte abanar, maior é a aceleração.

Quando estiver a usar **`sense.get_accelerometer_raw()`** está a indicar à Sense HAT para desligar os outros dois sensores na IMU, o sensor giroscópico e o magnetómetro, e a obter dados apenas do acelerómetro. Naturalmente, também pode fazer o mesmo com os outros sensores.

Encontre a linha **`acceleration = sense.get_accelerometer_raw()`** e altere-a para:

```
orientation = sense.get_gyroscope_raw()
```

Altere a palavra **`acceleration`** em todas as três linhas abaixo para **`orientation`**. Clique em Executar, e verá a orientação da Sense HAT para os três eixos, arredondada ao número inteiro mais próximo. No entanto, ao contrário da última vez que verificou a orientação, desta vez os dados vêm apenas do giroscópio, sem utilizar o acelerómetro ou magnetómetro. Isto pode ser útil se quiser saber, por exemplo, a orientação de uma Sense HAT em movimento nas costas de um robô, sem que o movimento confunda as leituras, ou se estiver a usar a Sense HAT perto de um campo magnético forte

Pare o seu programa clicando no botão vermelho de paragem. Para usar o magnetómetro, elimine tudo do seu programa exceto as primeiras quatro linhas e, em seguida, introduza o seguinte abaixo da linha **`while True`**:

```
north = sense.get_compass()
print(north)
```

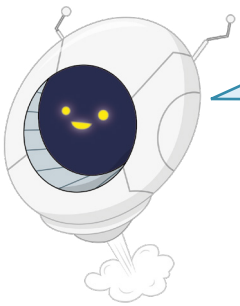
Execute o seu programa e verá a direção do norte magnético impressa repetidamente para a área da shell do Python. Rode cuidadosamente a Sense HAT e constará que o cabeçalho se altera com a orientação da Sense HAT à medida que o norte se altera: construiu uma bússola! Se tiver um íman, um de frigorífico serve, experimente movê-lo em torno da Sense HAT para ver o que acontece às leituras do magnetómetro.

DESAFIO: AUTO-ROTAÇÃO

Com o que aprendeu sobre a matriz LED e os sensores da unidade de medida inercial, consegue escrever um programa que rode uma imagem dependendo da posição da Sense HAT?

Controlo de Joystick

O joystick da Sense HAT, que se encontra no canto inferior direito, pode ser pequeno, mas é surpreendentemente poderoso: além de ser capaz de reconhecer entradas em quatro direções, cima, baixo, esquerda e direita, também tem uma quinta entrada, que é acedida premindo-a de cima para baixo como um interruptor de pressão.



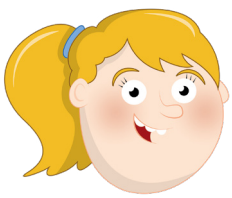
AVISO!

O joystick da Sense HAT só deve ser usado se tiver instalado os espaçadores como descrito no início deste capítulo. Sem os espaçadores, ao premir o joystick para baixo é possível que a placa Sense HAT se dobre e danifique tanto a Sense HAT como o conetor GPIO do Raspberry Pi.

Controlo de Joystick no Scratch

Inicie um novo programa no Scratch com a extensão Sense HAT do Raspberry Pi carregada. Como anteriormente: arraste um bloco Eventos quando alguém clicar em para a sua área de scripts, arraste um bloco clear display por baixo e, em seguida, arraste e edite um bloco set background to preto e um bloco set colour to branco.

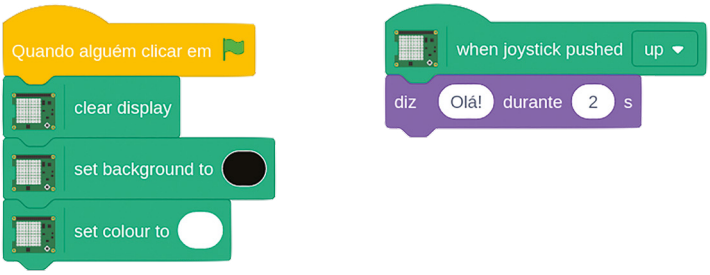
No Scratch, o joystick da Sense HAT mapeia as teclas do cursor no teclado: premir o joystick para cima é o equivalente a premir a tecla da seta para cima, premi-lo para baixo é equivalente a premir a tecla da seta para baixo, premi-lo para a esquerda é o mesmo que premir a tecla da seta para a esquerda e premi-lo para a direita é o mesmo que premir a tecla da seta para a direita; entretanto premir o joystick para dentro como um botão de pressão, é equivalente a premir a tecla **ENTER**.



AVISO!

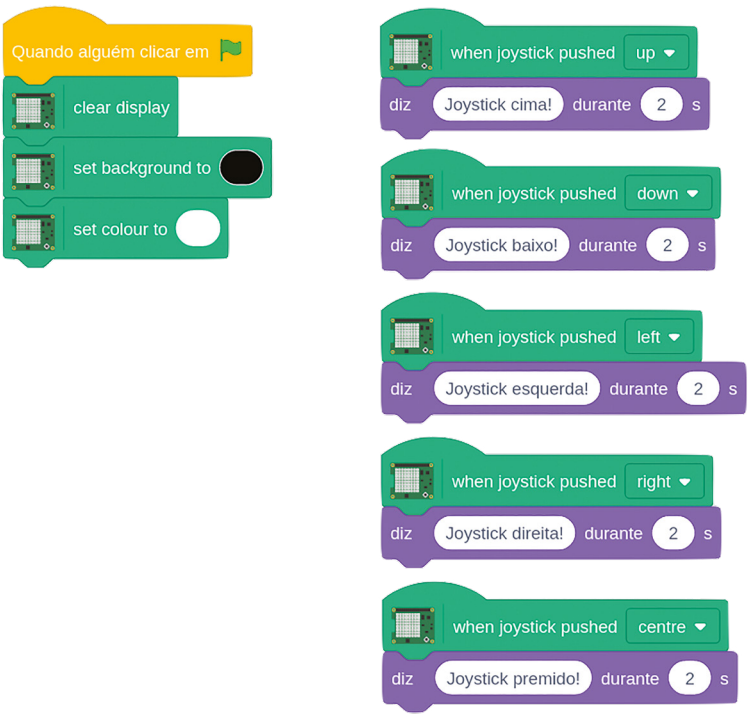
O controlo do joystick só está disponível na Sense HAT física. Ao usar o emulador da Sense HAT, use as teclas correspondentes no seu teclado para simular quando prime o joystick.

Arraste um bloco **when joystick pushed up** para a sua área de programação. Em seguida, para lhe dar algo para fazer, arraste um bloco **diz Olá! durante 2 s** para baixo do mesmo.

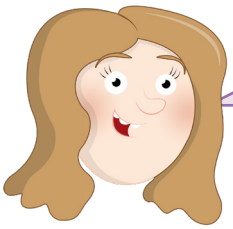


Prima o joystick para cima e constará que o gato do Scratch dirá um alegre "Olá!".

Em seguida, edite o bloco **diz Olá! durante 2 s** para se tornar num bloco **diz Joystick para cima! durante 2 s**, e continue a adicionar blocos de Eventos e de Aparência até obter algo para quando prime cada uma das cinco formas do joystick.



Tente premir o joystick em várias direções para ver as suas mensagens aparecerem!



DESAFIO FINAL

Consegue usar o joystick da Sense HAT para controlar um sprite do Scratch na área do palco? Pode fazer com que quando um sprite recolher outro sprite, representando um objeto, a Sense HAT apresente uma mensagem alegre?

Controlo de Joystick no Scratch

Inicie um novo programa no Thonny e guarde-o como Sense HAT Joystick. Comece com as três linhas habituais que configuram a Sense HAT e limpe a matriz LED:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()
```

Em seguida, configure um ciclo infinito:

```
while True:
```

Em seguida indique ao Python para ouvir as entradas do joystick da Sense HAT com a seguinte linha, que o Thonny indentará automaticamente por si:

```
    for event in sense.stick.get_events():
```

Por fim, adicione a seguinte linha, que, mais uma vez, o Thonny indentará por si, para realmente fazer algo quando for detetado um joystick:

```
        print(event.direction, event.action)
```

Clique em Executar, e experimente premir o joystick em várias direções. Verá a direção que escolheu impressa na área da shell do Python: cima, baixo, esquerda, direita e meio para quando tiver premido o joystick para baixo como um botão de pressão.

Também verá que lhe são dados dois eventos de cada vez que premir o joystick uma vez: um evento, **`pressed`**, para quando prime uma direção pela primeira vez; o outro evento, **`released`** para quando o joystick voltar ao centro. Pode usar isso nos seus programas: pense numa personagem de um jogo, que se pode configurar para começar a mexer-se quando o joystick é premido numa direção e depois parar assim que se solte o joystick.

Também pode usar o joystick para acionar funções, em vez de estar limitado a usá-lo para um ciclo. Elimine tudo por baixo de **`sense.clear()`**, e introduza o seguinte:

```
def red():
    sense.clear(255, 0, 0)

def blue():
    sense.clear(0, 0, 255)

def green():
    sense.clear(0, 255, 0)

def yellow():
    sense.clear(255, 255, 0)
```

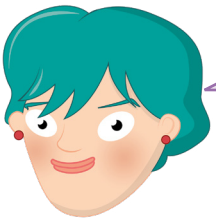
Estas funções alteram toda a matriz LED da Sense HAT para uma única cor: vermelho, azul, verde ou amarelo, o que vai fazer com que seja extremamente fácil de verificar se o seu programa está a funcionar! Para acioná-los realmente, tem de indicar ao Python qual a função que acompanha cada entrada do joystick. Introduza as seguintes linhas:

```
sense.stick.direction_up = red
sense.stick.direction_down = blue
sense.stick.direction_left = green
sense.stick.direction_right = yellow
sense.stick.direction_middle = sense.clear
```

Por fim, o programa precisa de um ciclo infinito, conhecido como o ciclo *principal*, para continuar a ser executado e, desta forma, continuar a vigiar as entradas do joystick, em vez de apenas ser executado pelo código que escreveu uma vez e terminar. Introduza as seguintes duas linhas:

```
while True:
    pass
```

Clique em Executar, e experimente mover o joystick: verá os LEDs acenderem numa cor gloriosa! Para desligar os LEDs, prima o joystick como um botão de pressão: a direção **`middle`** está configurada para usar a função **`sense.clear()`** para os desligar a todos. Parabéns: consegue capturar as entradas do joystick!



DESAFIO FINAL

Consegue usar o que aprendeu para desenhar uma imagem no ecrã e, em seguida, rodá-la no direção para a qual prime o joystick? Consegue que a entrada do meio alterne entre mais do que uma imagem?

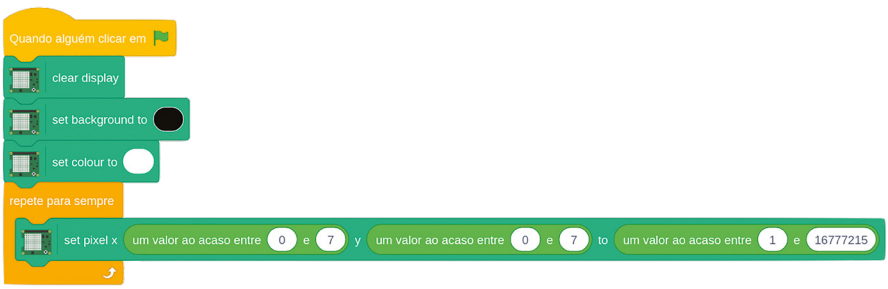
Projeto do Scratch: Sense HAT Sparkler

Agora que já conhece a Sense HAT, está na hora de usar tudo o que aprendeu para criar um fogo de artifício sensível ao calor, um dispositivo que fica mais feliz quando está frio e que vai ficando mais lento à medida que a temperatura aumenta.

Inicie um novo projeto no Scratch e adicione a extensão Sense HAT do Raspberry Pi, se ainda não estiver carregada. Como sempre, comece com quatro blocos: quando alguém clicar em , clear display , set background to preto , e set colour to branco , sem se esquecer que terá de alterar as cores das respetivas definições predefinidas.

Comece por criar um fogo de artifício simples, mas artístico. Arraste um bloco sempre para a área de programação e, em seguida, preencha-o com um bloco set pixel x 0 y 0 to colour. Em vez de utilizar números definidos, preencha cada uma das secções x, y e de cor desse bloco com um bloco um valor ao acaso entre 1 e 10 Operadores.

Os valores de 1 a 10 não são muito úteis neste caso, portanto terá de fazer alguma edição. Os dois primeiros números no bloco set pixel são as coordenadas X e Y do pixel na matriz LED, o que significa que devem ser números entre 0 e 7, portanto, altere os dois primeiros blocos para pick random 0 to 7. A próxima secção é a cor para a qual o pixel deve ser definido. Quando está a usar o seletor de cores, a cor escolhida é apresentada diretamente na área de scripts; no entanto, as cores são representadas por um número, e pode usar o número diretamente. Edite o último bloco aleatório de seleção para que conste um valor ao acaso entre 0 e 16777215.



Clique na bandeira verde e verá os LEDs na Sense HAT começarem a acenderem-se em cores aleatórias (**Figura 7-28**). Parabéns: criou fogo de artifício eletrónico!

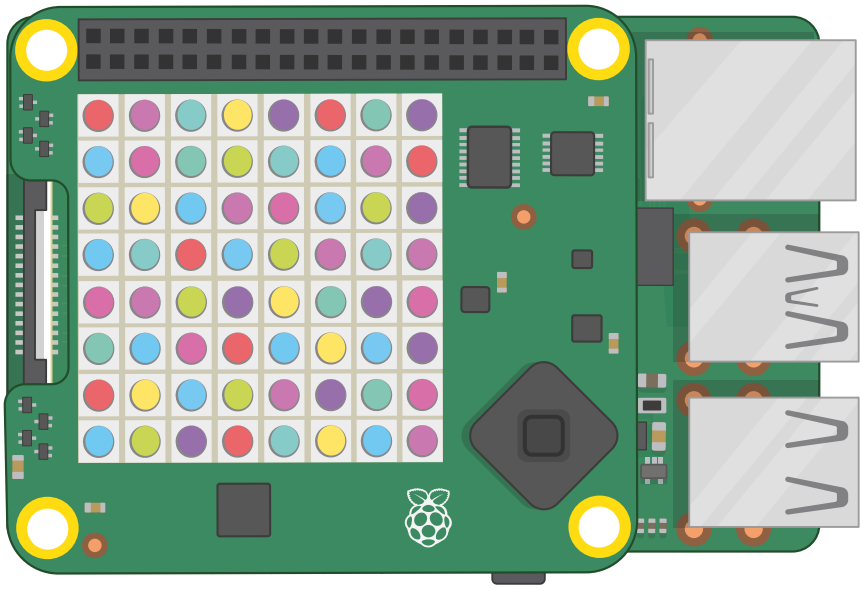
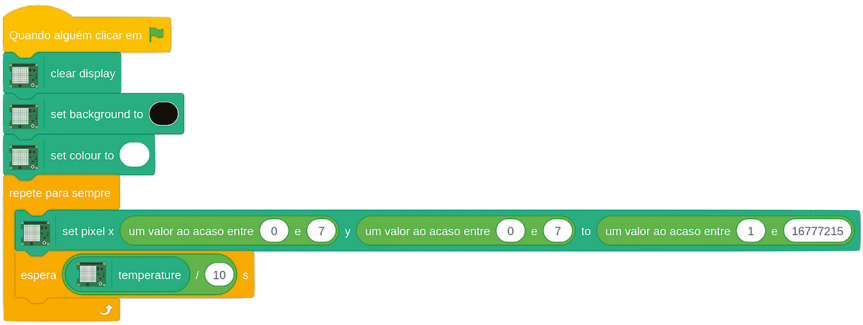


Figura 7-28: Iluminar os pixels em cores aleatórias

O fogo de artifício não é muito interativo. Para alterar isso, comece por arrastar um bloco **espera 1 s** para que fique por baixo do bloco **set pixel**, mas dentro do bloco **sempre**. Arraste um bloco **( ) / ( )** Operadores sobre o 1 e, em seguida, introduza 10 no respetivo segundo espaço. Por fim, arraste um bloco **temperature** sobre o primeiro espaço na separação do bloco Operador.



Clique na bandeira verde e constatará, a menos que viva num lugar muito frio, que o fogo de artifício é muito mais lento do que anteriormente. Isso acontece porque criou um atraso dependente da temperatura: o programa agora espera *a temperatura atual dividida por 10* número de segundos antes de cada ciclo. Se a temperatura no seu quarto for de 20 °C, o programa irá esperar 2 segundos antes de repetir o ciclo; se a temperatura for de 10 °C, irá esperar 1 segundo; se for abaixo de 10 °C, irá esperar menos de um segundo.

Se a sua Sense HAT detetar uma temperatura negativa, abaixo de 0 °C, o ponto de congelamento da água, tentará esperar menos de 0 segundos; o que é impossível, pelo menos sem se inventar as viagens no tempo, verá o mesmo efeito como se esperasse 0 segundos. Parabéns: já tem a possibilidade de integrar as várias funcionalidades da Sense HAT nos seus próprios programas!

Projeto do Python: Sense HAT Tricorder

Agora que já conhece a Sense HAT, está na hora de usar tudo o que aprendeu para criar um tricorder, um dispositivo reconhecido imediatamente pelos fãs de uma determinada franquia de ficção científica, capaz de comunicar as leituras dos vários sensores integrados.

Inicie um novo projeto no Thonny e guarde-o como **Tricorder**, em seguida, tem de fazer um programa Sense HAT com as linhas habituais:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()
```

Em seguida, tem de começar a definir as funções para cada um dos vários sensores da Sense HAT. Comece com a unidade de medição inercial ao escrever:

```
def orientation():
    orientation = sense.get_orientation()
    pitch = orientation["pitch"]
    roll = orientation["roll"]
    yaw = orientation["yaw"]
```

Como vai deslocar os resultados do sensor pelos LEDs, faz sentido arredondá-los para que não esteja à espera de dezenas de casas decimais. No entanto, em vez de números inteiros, arredonde-os para uma casa decimal, ao escrever o seguinte:

```
    pitch = round(pitch, 1)
    roll = round(roll, 1)
    yaw = round(yaw, 1)
```

Por fim, tem de indicar ao Python que desloque os resultados para os LEDs, para que o tricorder funcione como um dispositivo portátil sem a necessidade de o ligar a um monitor ou TV:

```
    sense.show_message("Arfagem {0}, Volta {1}, Guinada {2}".
format(pitch, roll, yaw))
```

Agora que tem uma função completa para ler e exibir a orientação da IMU, tem de criar funções semelhantes para cada um dos outros sensores. Comece com o sensor de temperatura:

```
def temperature():
    temp = sense.get_temperature()
    temp = round(temp, 1)
    sense.show_message("Temperatura: %s graus Celsius" % temp)
```

Olhe cuidadosamente para a linha que imprime o resultado para os LEDs: o **%s** é conhecido como um marcador de posição, e é substituído pelo conteúdo da variável **temp**. Com esta ação, pode formatar adequadamente a saída com uma etiqueta, "Temperatura:", e uma unidade de medida, "graus Celsius", o que torna o seu programa muito mais amigável.

Em seguida, defina uma função para o sensor de humidade:

```
def humidity():
    humidity = sense.get_humidity()
    humidity = round(humidity, 1)
    sense.show_message("Humidade: %s por cento" % humidity)
```

Em seguida o sensor de pressão:

```
def pressure():
    pressure = sense.get_pressure()
    pressure = round(pressure, 1)
    sense.show_message("Pressão: %s millibares" % pressure)
```

E por fim a leitura da bússola do magnetómetro:

```
def compass():
    for i in range(0, 10):
        north = sense.get_compass()
    north = round(north, 1)
    sense.show_message("Norte: %s graus" % north)
```

O ciclo curto **for** nesta função efetua dez leituras do magnetómetro para garantir que tem dados suficientes para fornecer-lhe um resultado preciso. Se verificar que o valor comunicado continua a alterar-se, tente estendê-lo para 20, 30, ou mesmo 100 ciclos para melhorar ainda mais a precisão.

O seu programa tem agora cinco funções, cada uma das quais tira uma leitura de um dos sensores da Sense HAT e desloca-os pelos LEDs. Este necessita de uma forma de escolher qual o sensor que quer usar, e o joystick é perfeito para isso.

Escreva o seguinte:

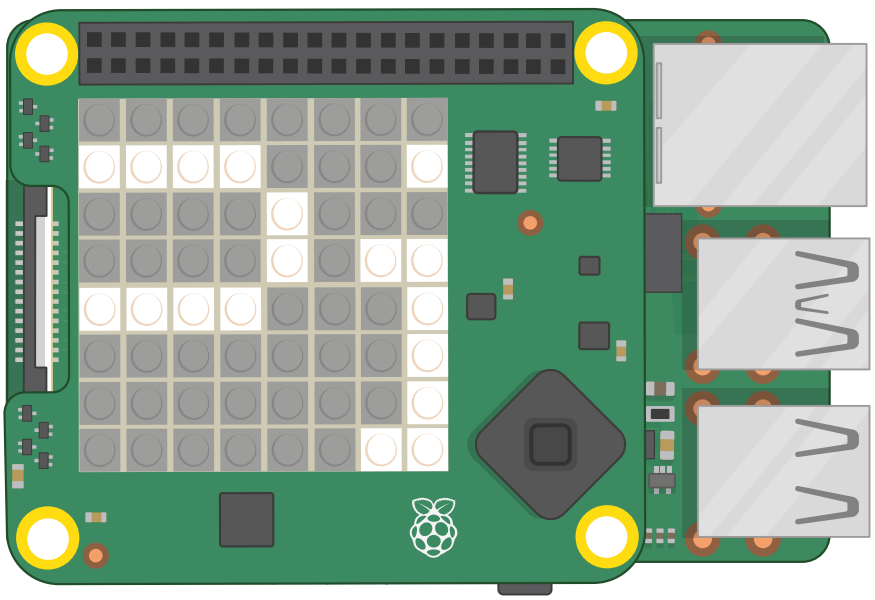
```
sense.stick.direction_up = orientation
sense.stick.direction_right = temperature
sense.stick.direction_down = compass
sense.stick.direction_left = humidity
sense.stick.direction_middle = pressure
```

Estas linhas atribuem um sensor a cada uma das cinco possíveis direções do joystick: para cima faz a leitura do sensor de orientação; para baixo faz a leitura do magnetómetro; para a esquerda faz a leitura do sensor de humidade; para a direita do sensor de temperatura; e premir o botão no meio faz a leitura do sensor de pressão.

Por fim, necessita de um ciclo principal para que o programa continue a detetar as pressões do joystick e que não feche imediatamente. Na parte inferior do seu programa, introduza o seguinte:

```
while True:
    pass
```

Clique em Executar, e experimente mover o joystick para tirar uma leitura de um dos sensores (**Figura 7-29**). Quando tiver terminado de deslocar o resultado, prima uma direção diferente. Parabéns: construiu um tricorder-portátil que deixaria a Federação dos Planetas orgulhosa!



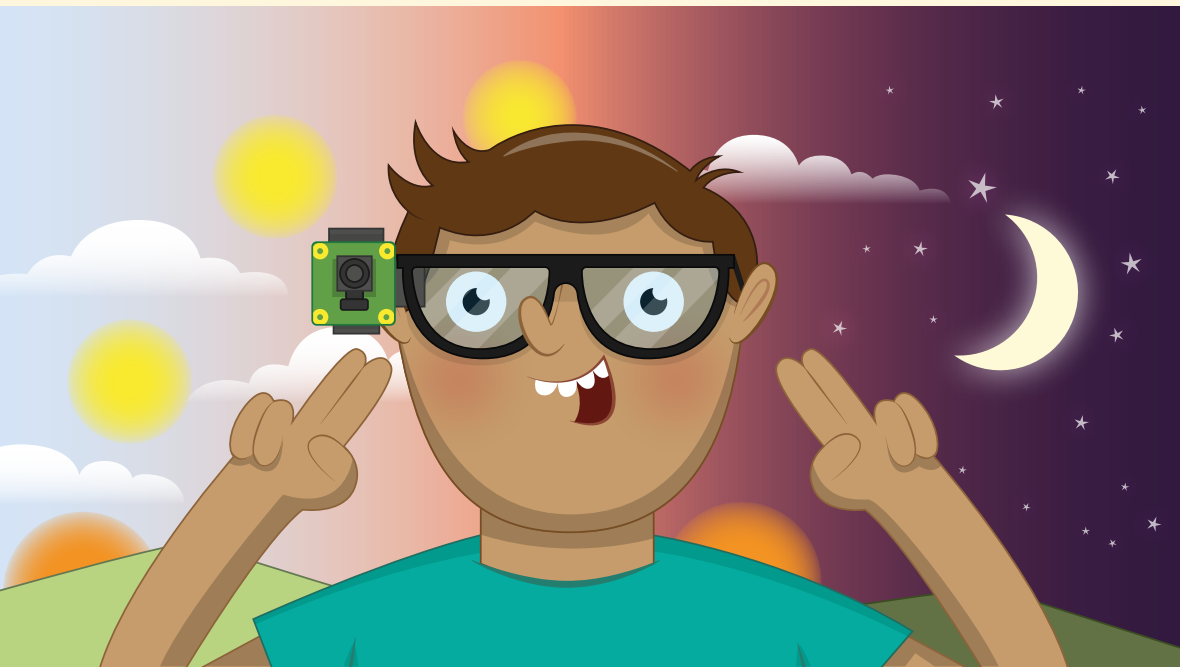
▲ **Figura 7-29:** Cada leitura desloca-se pelo ecrã

Para obter mais projetos da Sense HAT, siga as ligações em **Apêndice D, Leitura adicional**.

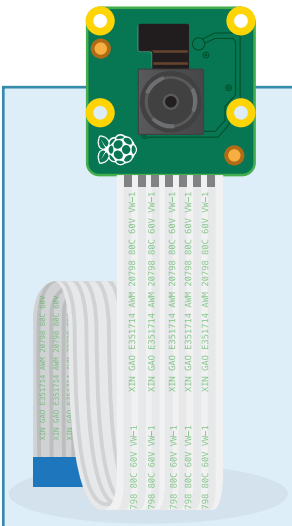
Capítulo 8

Camera Module do Raspberry Pi

A ligação de um Camera Module ou uma HQ Camera ao Raspberry Pi permite-lhe tirar fotografias de alta resolução e filmar vídeos e criar projetos de visão computacional fantásticos



Se algum dia desejou construir algo com capacidade de visão – conhecido no campo da robótica como *visão por computador* – então o Camera Module opcional do Raspberry Pi, ou a nova HQ Camera, é um ótimo ponto de partida. Uma pequena placa de circuito quadrada com um cabo plano fino, o Camera Module/HQ Camera liga-se à porta Camera Serial Interface (CSI) no seu Raspberry Pi (não disponível no Raspberry Pi 400) e fornece imagens estáticas de alta resolução e sinais de vídeo em movimento que podem ser utilizados como se encontram ou integrados nos seus próprios programas.



TIPOS DE CÂMARAS!

Estão disponíveis três tipos de Câmara Raspberry Pi: o módulo de câmara padrão, a versão "NoIR" e a câmara de alta qualidade (HQ). Se quiser tirar fotografias e vídeos normais em ambientes bem iluminados, deve utilizar o módulo de câmara padrão; se quiser usar lentes especiais e pretende obter a melhor qualidade de imagem, utilize o módulo de câmara HQ. O módulo de câmara NoIR – assim chamado porque não possui filtro de infravermelho, ou IR – foi concebido para utilização com fontes de luz infravermelha para tirar fotografias e vídeos na escuridão total. Se está a construir uma caixa de nidificação, câmara de segurança ou outro projeto que envolva visão noturna, utilize a versão NoIR – mas lembre-se de comprar também uma fonte de luz infravermelha!

Os Camera Modules padrão e NoIR do Raspberry Pi são baseados num sensor de imagem Sony IMX219. É um *sensor de 8 megapixels*, o que significa que pode tirar fotografias com até 8 milhões de pixels. Esse resultado é obtido mediante a captura de imagens até 3280 pixels de largura por 2464 de altura. Além de imagens estáticas, o Camera Module pode capturar imagens de vídeo com resolução Full HD a uma velocidade de 30 fotogramas por segundo (30 fps). Para um movimento mais suave ou até mesmo para criar um efeito de movimento lento, a câmara pode ser configurada para capturar a uma velocidade de fotogramas mais elevada através da diminuição da resolução: 60fps para imagens de vídeo de 720p, e até 90 fps para imagens de 480p (VGA).

A HQ Camera utiliza um sensor Sony IMX477 de 12,3 megapixels que também é maior que o sensor dos Camera Modules padrão e NoIR – o que significa que pode absorver mais luz, resultando em imagens de maior qualidade. Contudo, ao contrário dos Camera Modules, a HQ Camera não inclui uma lente, sem a qual não pode tirar fotografias ou vídeos. Pode utilizar qualquer lente com um mecanismo de montagem C ou CS; outras montagens de lentes podem ser utilizadas com um adaptador de montagem C ou CS apropriado. Para obter detalhes sobre como montar uma lente, consulte o **Apêndice F: HQ Camera**.

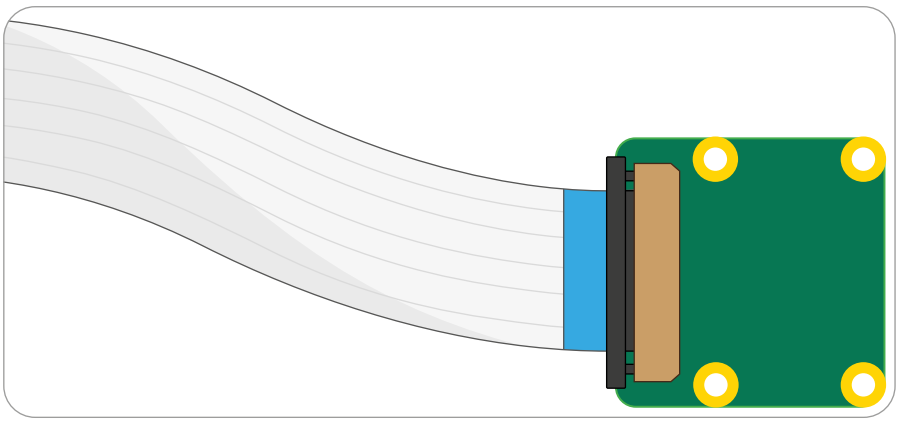
RASPBERRY PI 400

Infelizmente, os módulos de câmara do Raspberry Pi não são compatíveis com o Raspberry Pi 400. Pode utilizar câmaras Web USB como alternativa, mas não poderá usar as ferramentas de software específicas do módulo de câmara do Raspberry Pi incluídas no Raspberry Pi OS.

Instalar a câmara

Como qualquer suplemento de hardware, o Camera Module ou HQ Camera apenas deve ser ligado ou desligado do Raspberry Pi quando a alimentação e o respetivo cabo estiverem desligados. Se o Raspberry Pi estiver ligado, selecione Encerrar no menu da framboesa, aguarde que desligue e retire a ficha da tomada.

Na maioria dos casos, o cabo plano fornecido já está ligado ao Camera Module ou HQ Camera; se não estiver, vire a placa da câmara ao contrário para que o sensor fique na parte inferior e procure um conetor de plástico plano. Use as unhas para segurar cuidadosamente as extremidades salientes e puxe-as para fora até o conetor ficar parcialmente para fora. Deslize o cabo plano, com as extremidades prateadas para baixo e o plástico azul virado para cima, por baixo da patilha que acabou de puxar para fora e, em seguida, empurre a patinha suavemente para a posição correta com um clique (**Figura 8-1**); não importa qual a extremidade do cabo que utiliza. Se o cabo for instalado corretamente, estará reto e não sairá da posição se o puxar ligeiramente. Se não estiver bem instalado, puxe a patilha para fora e tente novamente.



▲ **Figura 8-1:** Ligar o cabo plano ao Camera Module

Instale a outra extremidade do cabo da mesma maneira. Encontre a porta da câmara (ou CSI) no Raspberry Pi e puxe a patilha suavemente para cima. Se o seu Raspberry Pi estiver instalado num estojo, poderá ser mais fácil removê-lo primeiro. Com o Raspberry Pi posicionado de modo que a porta HDMI fique de frente para si, deslize o cabo plano para que as extremidades prateadas fiquem à sua esquerda e o plástico azul à sua direita (**Figura 8-2**) e, em seguida, empurre suavemente a patilha de volta para a posição correta. Se o cabo for instalado corretamente, estará reto e não sairá da posição se o puxar ligeiramente. Se não estiver bem instalado, puxe a patilha para fora e tente novamente.

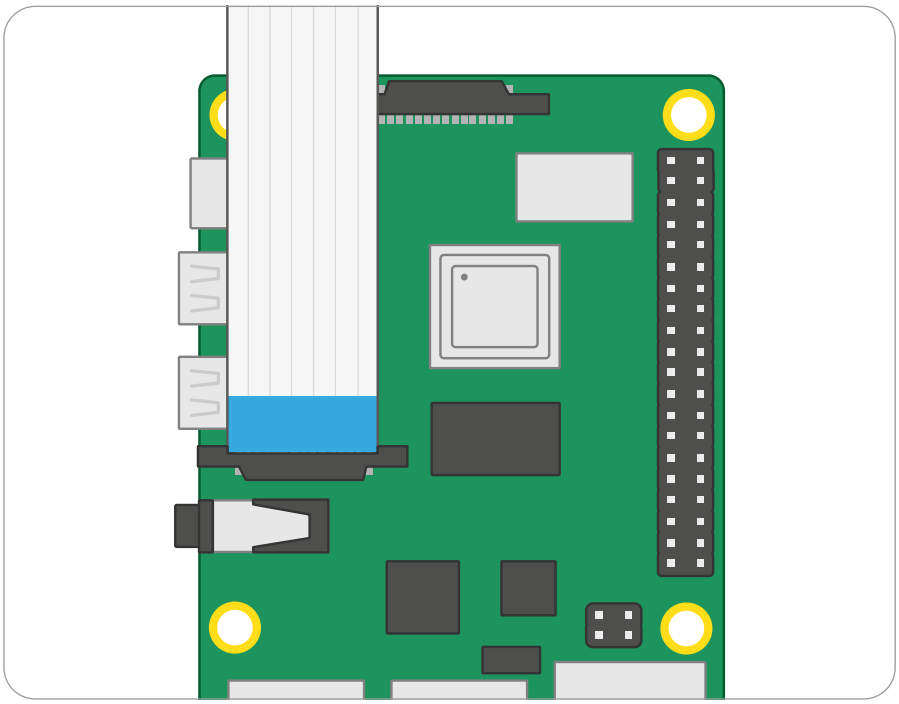
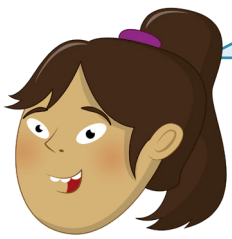


Figura 8-2: Ligar o cabo plano à porta da câmara/CSI no Raspberry Pi

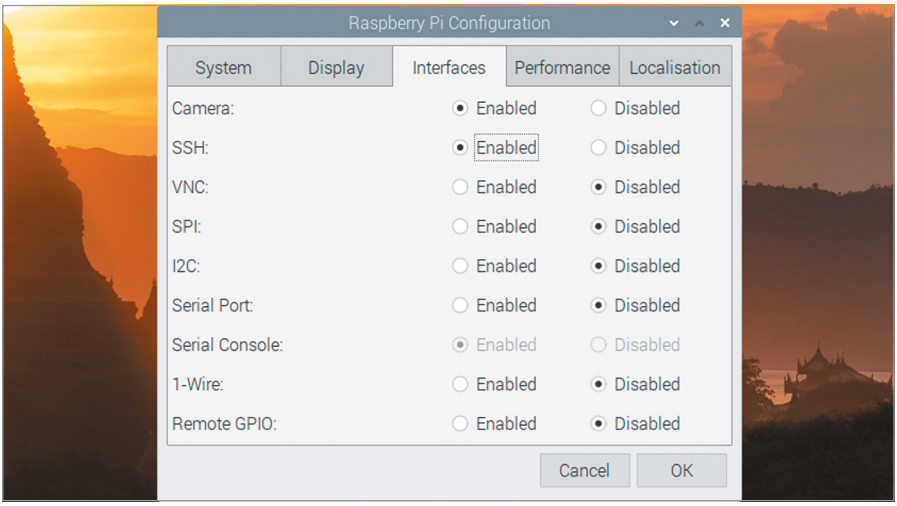
O Camera Module vem com um pequeno pedaço de plástico azul a cobrir a lente para protegê-la contra arranhões durante o fabrico, transporte e instalação. Encontre a pequena patilha de plástico e remova-a da lente com cuidado para que a câmara fique pronta a ser utilizada.



AJUSTAR O FOCO

O módulo de câmara é normalmente fornecido com uma pequena roda de plástico para ajustar o foco da lente. Embora o foco definido de fábrica seja normalmente perfeito, se estiver a utilizar a sua câmara para trabalhos de grande plano, pode deslizar a roda sobre a lente e rodá-la suavemente para ajustar o foco manualmente. Para focagem da câmara HQ, consulte o **Apêndice F**.

Volte a ligar a fonte de alimentação ao Raspberry Pi e deixe-o carregar o Raspberry Pi OS. Para poder utilizar a câmara, terá de indicar ao Raspberry Pi que tem uma câmara ligada: abra o menu ícone da framboesa, escolha a categoria Preferências e clique em Configuração do Raspberry Pi. Quando a ferramenta carregar, clique no separador Interfaces, encontre a entrada Câmara na lista e clique no botão de opção redondo à esquerda de "Ativado" para ligá-la (**Figura 8-3** no verso). Clique em OK e será solicitado pela ferramenta que reinicie o Raspberry Pi. Reinicie e a câmara estará pronta para ser utilizada!

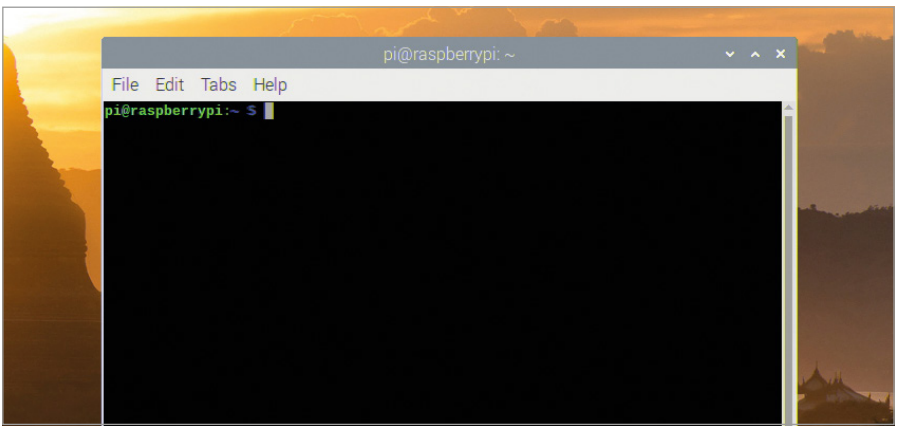


▲ **Figura 8-3:** É necessário ativar a câmara na Raspberry Pi Configuration

Testar a câmara

Para confirmar que o Camera Module ou HQ Camera está devidamente instalado, e que ativou a interface na Ferramenta de configuração Raspberry Pi, pode utilizar a ferramenta **raspistill**. Essa ferramenta, juntamente com a **raspivid** para vídeos, está concebida para capturar imagens da câmara utilizando a *interface de linha de comando (command-line interface – CLI* do Raspberry Pi.

Ao contrário dos programas que tem utilizado até agora, não encontrar a ferramenta raspistill no menu. Clique no ícone da framboesa para carregar o menu, escolha a categoria Acessórios e clique em Terminal. Será apresentada uma janela preta com texto a verde e azul (**Figura 8-4**): é o *terminal* que lhe permite aceder à interface de linha de comando.



▲ **Figura 8-4:** Abrir uma janela Terminal para introduzir comandos

Para testar a câmara, escreva o seguinte no Terminal:

```
raspistill -o test.jpg
```

Quando clicar em **ENTER**, será apresentada no ecrã uma imagem grande do que a câmara consegue ver (**Figure 8-5**). A essa imagem chama-se *pré-visualização em direto* e, a menos que o raspistill receba outra instrução, será visível durante 5 segundos. Após os 5 segundos, a câmara captura uma única imagem fixa e guarda-a na pasta Início com o nome **test.jpg**. Se quiser capturar outra imagem, escreva o mesmo comando novamente, mas certifique-se de que muda o nome do ficheiro de saída a seguir a **`-o`** para não substituir a sua primeira imagem!

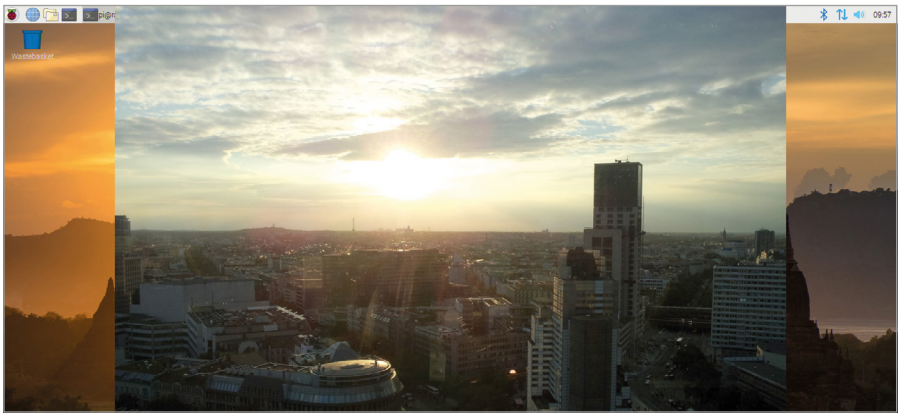


Figura 8-5: A pré-visualização em direto da câmara

Se a pré-visualização em direto estava invertida, terá de indicar ao raspistill que a câmara está virada. O Camera Module está concebido para o cabo plano sair pela margem inferior; se sair da lateral ou da parte superior, como em alguns acessórios de montagem de câmaras de terceiros, pode rodar a imagem em 90, 180, ou 270 graus utilizando o interruptor **`-rot`**. Para uma câmara montada com o cabo a sair da parte superior, basta utilizar o seguinte comando:

```
raspistill -rot 180 -o test.jpg
```

Se o cabo plano sair pela margem direita, utilize um valor de rotação de 90 graus; se sair pela margem esquerda, utilize 270 graus. Se a sua captura original estava no ângulo errado, tente outra utilizando o interruptor **`-rot`** para a corrigir.

Para ver a imagem, abra o Gestor de ficheiros na categoria Acessórios do menu framboesa: a imagem que tirou, chamada **test.jpg**estará na pasta **home/pi**. Encontre-a na lista de ficheiros e faça duplo clique para carregá-la num visualizador de imagens (**Figura 8-6**). Também pode anexar a imagem a e-mails, carregá-la em Web sites através do navegador ou arrastá-la para um dispositivo de armazenamento externo.

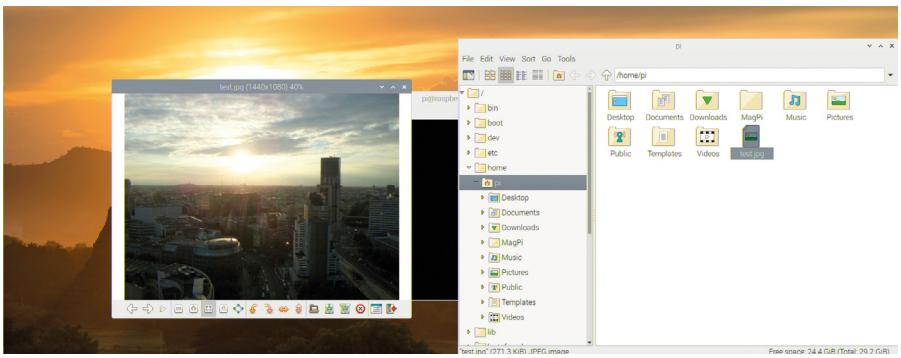
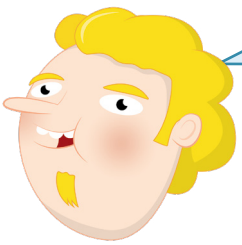


Figura 8-6: A abrir a imagem capturada

Apresentamos a picamera

A forma mais flexível de controlar o Camera Module ou a HQ Camera é utilizando o Python com a prática biblioteca picamera. Dessa forma, tem controlo total sobre as capacidades de pré-visualização, captura de imagem e vídeo da câmara, permitindo-lhe integrá-las nos seus próprios programas e até mesmo combiná-las com programas que utilizam o módulo GPIO através da biblioteca GPIO Zero!



PROGRAMAÇÃO PYTHON

Os projetos deste capítulo assumem que tem experiência com a linguagem de programação Python, Thonny IDE e os pinos GPIO do Raspberry Pi. Se ainda não o fez, procure o **Capítulo 5, Programar com Python** e o **Capítulo 6, Computação física com Scratch e Python** e trabalhe primeiro com esses projetos!

Feche o Terminal, se ainda estiver aberto, clicando no X no canto superior direito da janela e, em seguida, carregue o Thonny na categoria Programação do menu da framboesa. Guarde o seu novo projeto como **Câmara** e comece a importar as bibliotecas de que o seu programa precisa escrevendo o seguinte na área Script:

```
from picamera import PiCamera
from time import sleep
camera = PiCamera()
```

A última linha permite controlar o módulo de câmara ou a câmara HQ utilizando a função **camera**. Para começar, escreva o seguinte:

```
camera.start_preview()
sleep(10)
camera.stop_preview()
```

Clique em Executar e o ambiente de trabalho desaparecerá; no seu lugar, verá uma pré-visualização em ecrã inteiro do que a câmara consegue ver (**Figura 8-7**). Tente movê-la ou acene com a mão em frente à lente para ver a imagem no ecrã mudar. Após 10 segundos, a pré-visualização fecha e o seu programa termina, mas, ao contrário da pré-visualização do raspistill, nenhuma imagem será guardada mais tarde.

Figura 8-7: Uma pré-visualização no ecrã inteiro em direto da visualização da câmara

Se a pré-visualização estava invertida, pode rodar a imagem para colocá-la novamente na posição certa. Por baixo da linha **camera = PiCamera()**, escreva:

```
camera.rotation = 180
```

Se a pré-visualização estava de cabeça para baixo, essa linha vai corrigir a posição. À semelhança do que sucede com o raspistill, **camera.rotation** permite rodar a imagem em 90, 180 ou 270 graus, dependendo se o cabo sai do lado direito, superior ou esquerdo do módulo de câmara. Lembre-se de utilizar **camera.rotation** no início de qualquer programa que escreva para evitar a captura de imagens ou vídeos invertidos!

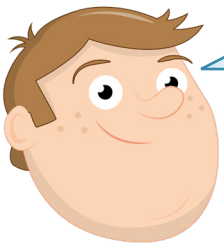
Captura de imagens estáticas

Para capturar uma imagem, em vez de apenas mostrar uma pré-visualização em direto, o seu programa tem de ser alterado. Comece por reduzir o atraso para a pré-visualização: encontre a linha **`sleep(10)`** e mude-a para ler:

```
sleep(5)
```

Diretamente por baixo dessa linha, adicione o seguinte:

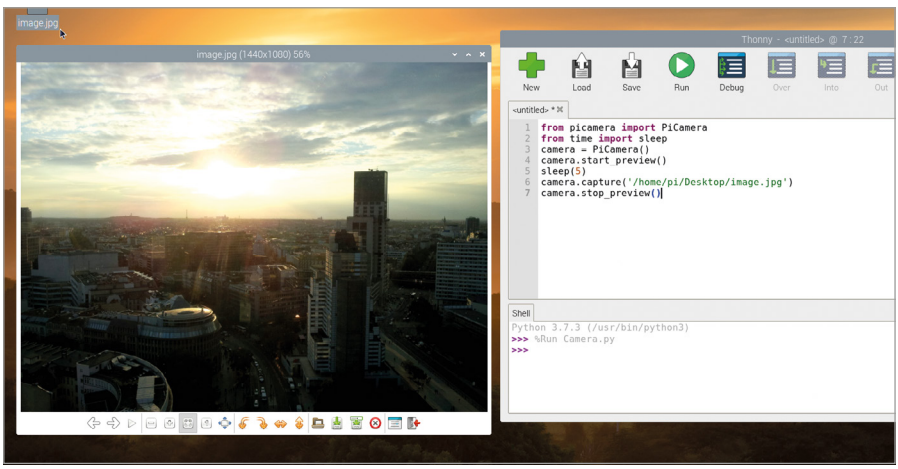
```
camera.capture('/home/pi/Desktop/image.jpg')
```



ESTÁ NA HORA DE AJUSTAR

Quando a câmara está no modo de pré-visualização, analisa o vídeo para ver se é necessário ajustar as suas definições para obter a melhor qualidade. Pode constatar isso se estiver num ambiente muito escuro ou muito claro, com a pré-visualização impossível de ver no início, e depois rapidamente ficando mais nítida. Para dar tempo à câmara para ajustar, adicione sempre um período de pré-visualização de pelo menos 2 segundos ao seu programa antes de capturar uma imagem.

A função **`camera.capture`** diz ao Python para guardar uma imagem estática, que tem de saber não apenas como a imagem deve ser chamada, mas em que pasta deve ser guardada. Neste exemplo, está a guardar no ambiente de trabalho. Procure a imagem logo abaixo do Cesto de papéis. Se a janela do Thonny estiver sobreposta, basta clicar e arrastar a barra do título para a mover. Faça duplo clique no ficheiro para ver a imagem que capturou (**Figura 8-8**). Parabéns: programou uma câmara!



▲ **Figura 8-8:** A abrir a imagem capturada

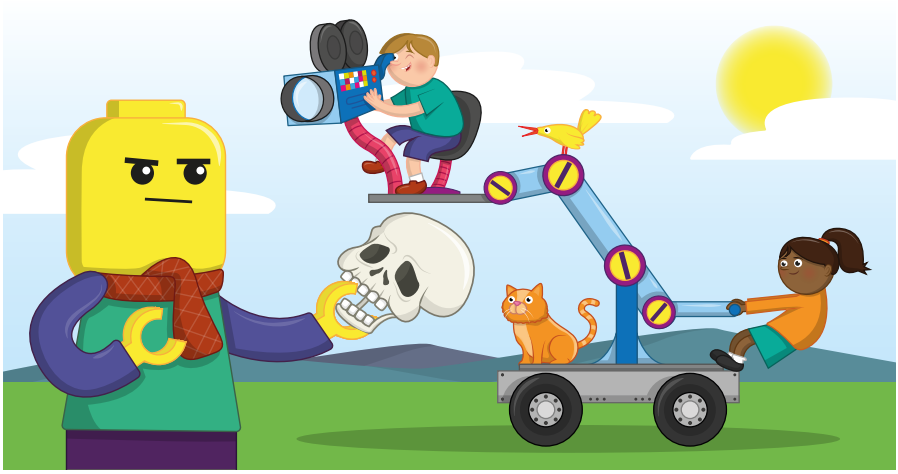
Captura de vídeo em movimento

Além de tirar imagens estáticas, também pode capturar vídeo. Apague tudo entre as linhas **`camera.start_preview()`** e **`camera.stop_preview()`** e escreva o seguinte por baixo de **`camera.start_preview():`**

```
camera.start_recording('/home/pi/Desktop/video.h264')
sleep(10)
camera.stop_recording()
```

A pré-visualização da câmara será apresentada, como antes, mas desta vez também será gravada num ficheiro no ambiente de trabalho. Aguarde os 10 segundos que indicou ao Python para dormir – pode fazer uma dança em frente à câmara para tornar o vídeo mais interessante – e, quando a pré-visualização fechar, o vídeo estará disponível no ambiente de trabalho.

Para reproduzir o vídeo, basta fazer duplo clique no ficheiro **video.h264** no ambiente de trabalho. Começa a reprodução do vídeo e, se fez uma dança, pode ver como ficou! Após o vídeo terminar, o software do leitor fecha com uma mensagem amigável no Terminal. Parabéns: agora pode capturar vídeo utilizando o módulo de câmara ou a câmara HQ do seu Raspberry Pi!



Animação stop-motion com botão de pressão

Recorrendo ao que aprendeu neste capítulo e ao conhecimento sobre como ligar hardware ao conetor GPIO do Raspberry Pi do **Capítulo 6**, **Computaçãofísica**, chegou a hora de construir algo especial: o seu próprio estúdio de animação em stop-motion.

A animação em stop-motion é o processo de tirar muitas fotografias de objetos imóveis, como carros modelo ou figuras de ação, e mover os objetos ligeiramente entre cada foto. Embora os objetos nunca se movam em nenhuma das imagens, se os mostrar um após o outro suficientemente rápido, vai parecer que estão a mover-se à velocidade que desejar!

Para este projeto, vai precisar de um interruptor de botão de pressão, uma placa universal, um cabo jumper macho para macho (M2M) e um par de macho para fêmea (M2M). Se não tiver uma placa universal, pode ligar o interruptor utilizando cabos jumper fêmea para fêmea (F2F), mas vai ser mais difícil premir. Se precisar de rever alguns dos componentes, regresse ao **Capítulo 6, Computação física com Scratch e Python**. Também vai precisar de objetos para animar: podem ser qualquer coisa, desde um pedaço de barro até um carro de brinquedo ou uma figura de ação.

Comece por criar o circuito: adicione o botão de pressão à placa universal, em seguida ligue o trilho terra da placa universal a um pino terra do Raspberry Pi (marcado com GND na **Figura 8-9**) com um cabo jumper macho para fêmea. Utilize um cabo jumper macho para macho para ligar uma perna do interruptor ao trilho terra na placa universal, depois um cabo jumper macho para fêmea para ligar a outra perna do interruptor ao pino GPIO 2 (marcado GP2 em **Figura 8-9**).

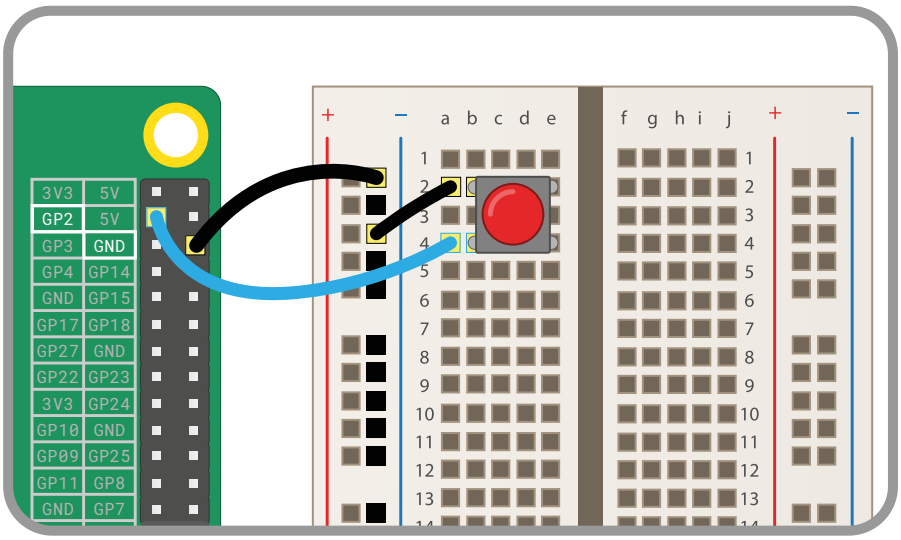


Figura 8-9: Diagrama de cabos para ligar um botão de pressão aos pinos GPIO

Crie um novo projeto no Thonny e guarde-o como **Stop Motion**. Comece por importar e configurar as bibliotecas necessárias para utilizar a câmara e a porta GPIO:

```
from picamera import PiCamera
from gpiozero import Button
camera = PiCamera()
button = Button(2)
```

Depois, escreva o seguinte:

```
camera.start_preview()
button.wait_for_press()
camera.capture('/home/pi/Desktop/image.jpg')
camera.stop_preview()
```

Clique em Executar e verá uma pré-visualização do ângulo que a câmara está a apontar. A pré-visualização permanece no ecrã até premir o botão: prima agora e a pré-visualização fecha após o seu programa guardar uma imagem no ambiente de trabalho. Encontre a imagem, chamada **image.jpg** e faça duplo clique para abri-la e confirmar que o programa está a funcionar.

A animação em stop-motion envolve a criação de muitas imagens estáticas, para dar a impressão de movimento quando são montadas em conjunto. Todas estas imagens resultam num ambiente de trabalho confuso, por isso, precisa de uma pasta para as guardar. Clique com o botão direito do rato em qualquer ponto no ambiente de trabalho que ainda não tenha um ficheiro ou um ícone e selecione Criar novo e Pasta (**Figura 8-10**). Atribua à pasta o nome **animation**, tudo em letras minúsculas e, em seguida, clique no botão OK.

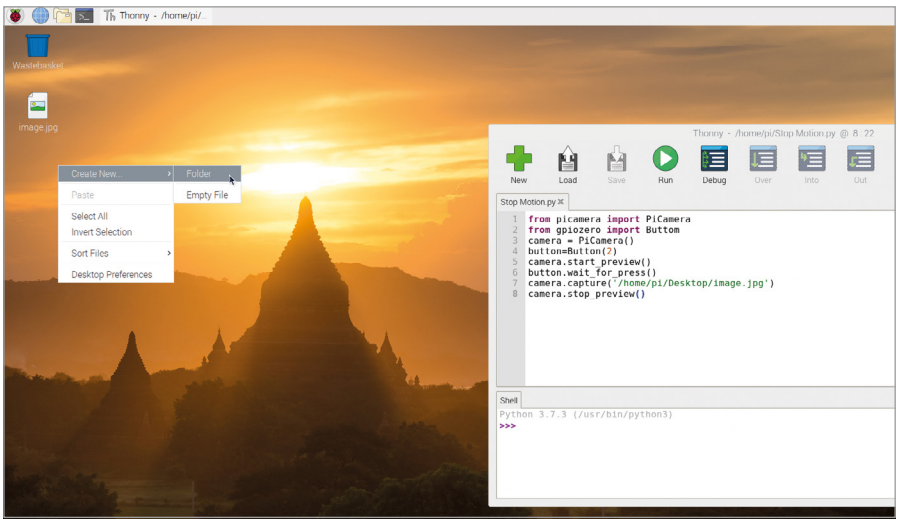


Figura 8-10: Criar uma nova pasta para as imagens capturadas

Ter que reiniciar o seu programa sempre que captura uma imagem para a sua animação não é produtivo, por isso, tem de alterá-lo para executar num ciclo. Ao contrário dos ciclos anteriores que criou, este requer uma forma de encerrar graciosamente, caso contrário, se parar o programa enquanto a pré-visualização da câmara é apresentada, deixará de ver o ambiente de trabalho. Para isso, tem de utilizar duas instruções especiais: `try` e `except`.

Comece por apagar tudo a seguir a **camera.start_preview(),** e depois escreva:

```
frame = 1
```

Isto cria uma nova variável, **frame**, que o seu programa utilizará para armazenar o número de fotogramas atual. Vai utilizar essa variável em breve para garantir que guarda sempre um novo ficheiro; sem isso, apenas seria possível guardar uma imagem por cima de outra, substituindo a primeira imagem sempre que clicasse no botão.

A seguir, configure o ciclo escrevendo:

```
while True:
    try:
```

A nova instrução **try** diz ao Python para executar qualquer código no interior – que será o código para capturar imagens. Escreva:

```
button.wait_for_press()
camera.capture('/home/pi/Desktop/animation/frame%03d.jpg' % frame)
frame += 1
```

Há um par de truques inteligentes nestas três linhas de código. O primeiro está no nome do ficheiro da captura: utilizando **%03d**, diz ao Python para escolher um número e adicionar os zeros à frente que forem necessários para que tenha três dígitos. Assim, "1" passa a "001", "2" passa a "002" e "10" passa a "010". Precisa desta opção no seu programa para manter os ficheiros na ordem correta e ter a certeza de que não está a substituir um ficheiro que já guardou.

O **% frame** no final dessa linha diz ao Python para utilizar o número da variável do fotograma no nome do ficheiro. Para garantir que cada ficheiro é único, a última linha – **frame += 1** – incrementa a variável do fotograma em 1. A primeira vez que premir o botão, **frame** será aumentado de 1 para 2; na próxima vez, de 2 para 3, e assim por diante.

De momento, porém, o seu código não encerra sem problemas quando conclui a captura de imagens. Para que isso aconteça, precisa de um **except** para **try**. Escreva o seguinte, sem se esquecer de remover um nível de indentação na primeira linha para que o Python saiba que não faz parte da secção **try**:

```
    except KeyboardInterrupt:
        camera.stop_preview()
        break
```

O seu programa final terá o seguinte aspeto:

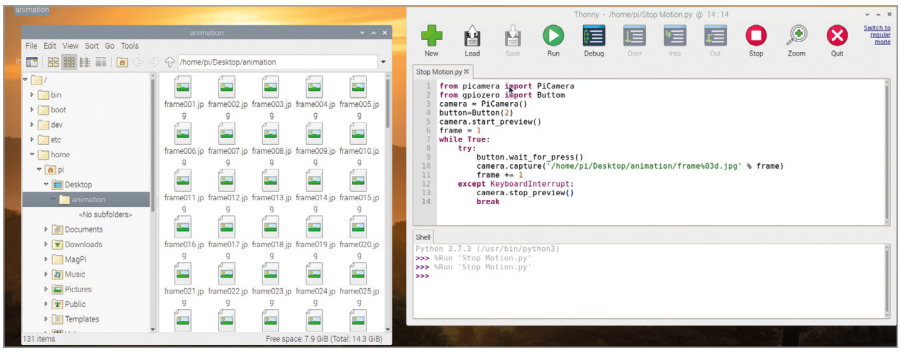
```
from picamera import PiCamera
from time import sleep
from gpiozero import Button
camera = PiCamera()
button = Button(2)
camera.start_preview()
frame = 1
while True:
    try:
        button.wait_for_press()
        camera.capture('/home/pi/Desktop/animation/frame%03d.jpg'
% frame)
        frame += 1
    except KeyboardInterrupt:
        camera.stop_preview()
        break
```

Experimente clicar em Executar, mas em vez de premir o botão, prima as teclas **CTRL** e **C** no teclado. Não é preciso premir ambas as teclas ao mesmo tempo: basta manter premida a tecla **CTRL**, premir e soltar **C** e, em seguida, soltar **CTRL**. Estas duas teclas funcionam como uma interrupção que diz ao Python para parar o que está a fazer. Sem a linha **`except KeyboardInterrupt:`**, o Python fecharia imediatamente e deixaria a pré-visualização da câmara a bloquear o ecrã; com essa linha , no entanto, o Python executa qualquer código que esteja no interior – neste caso, o código que lhe indica que pare a pré-visualização da câmara e feche sem problemas.

Agora está pronto para começar a capturar a sua animação em stop-motion! Posicione o Camera Module ou a HQ Camera onde a lente consiga ver os objetos que vai animar e certifique-se de que não se move – se a câmara se mover, estraga o efeito. Coloque os objetos nas suas posições iniciais, depois clique em Executar para iniciar o seu programa. Verifique se está tudo bem na pré-visualização e prima o botão para capturar o primeiro fotograma.

Mova ligeiramente os objetos – quanto menos os mover entre fotogramas, mais perfeita será a animação concluída – e prima novamente o botão para capturar outro fotograma. Continue a fazer isso até que a animação esteja concluída: quantos mais fotogramas capturar, mais longa será a sua animação.

Quando terminar, prima **CTRL+C** para fechar o seu programa, depois clique duas vezes no botão **animation** no ambiente de trabalho para ver as imagens capturadas (**Figura 8-11** no verso). Faça duplo clique em qualquer imagem para a abrir e ver com mais detalhes!



▲ **Figura 8-11:** As imagens capturadas na pasta

No momento, no entanto, apenas tem uma pasta cheia de imagens estáticas. Para criar uma animação, tem de transformá-las num vídeo. Para o fazer, clique no ícone da framboesa para carregar o menu, escolha a categoria Acessórios e clique em Terminal. Abre a *interface de linha de comando*, discutida com mais detalhes no **Apêndice C**, que lhe permite introduzir comandos para o Raspberry Pi. Quando o Terminal carregar, comece por mudar para a pasta que criou escrevendo:

```
cd Desktop/animation
```

É importante que o "D" de "Desktop" esteja em maiúsculas; o Raspberry Pi OS é o que se chama de *sensível a maiúsculas e minúsculas*, ou seja, se não escrever um comando ou nome de pasta exatamente como foi originalmente escrito, não funciona! Após mudar as pastas, escreva o seguinte:

```
ffmpeg -i frame%03d.jpg -r 10 animation.h264
```

Esta ação utiliza um programa chamado **ffmpeg** para pegar nas imagens estáticas na pasta e convertê-las num vídeo chamado **animation.h264**. (Nota: se o ffmpeg não estiver disponível, pode instalá-lo com `sudo apt-get install ffmpeg`.) Dependendo de quantos fotogramas tirou, este processo pode demorar alguns minutos; saberá que está terminado quando o aviso do Terminal reaparecer.

Para reproduzir o vídeo, procure o ficheiro **animation.h264** na pasta **animation** e faça duplo clique para o abrir. Em alternativa, pode reproduzir o ficheiro a partir do Terminal escrevendo o seguinte:

```
omxplayer animation.h264
```

Quando o vídeo carregar, verá a animação stop-motion ganhar vida. Parabéns: transformou o seu Raspberry Pi num poderoso estúdio de animação!

Se a animação se mover muito rápido ou muito lentamente, mude a parte **`-r 10`** do comando **`ffmpeg`** para um número inferior ou superior: esta é a taxa de fotogramas, ou seja, quantas imagens estáticas existem num segundo de vídeo. Um número mais baixo fará a sua animação correr mais lentamente, mas parecer menos perfeita; um número mais alto parecerá mais suave, mas fará a animação correr mais rapidamente.

Se quiser guardar o seu vídeo, arraste e largue-o do ambiente de trabalho para a pasta de vídeos; caso contrário, da próxima vez que executar o seu programa, acabará por substituir o ficheiro!

Definições avançadas da câmara

Se requer mais controlo do módulo de câmara e câmara HQ do Raspberry Pi, pode utilizar a biblioteca picamera do Python para aceder a diversas definições. Essas definições, juntamente com os seus valores predefinidos, são detalhadas abaixo para incluir nos seus próprios programas.

`camera.awb_mode = 'auto'`

Isto define o modo de equilíbrio automático de brancos da câmara e pode ser ajustado para qualquer um dos seguintes modos: **`off`**, **`auto`**, **`sunlight`**, **`cloudy`**, **`shade`**, **`tungsten`**, **`fluorescent`**, **`incandescent`**, **`flash`** ou **`horizon`**. Se achar que as suas fotos e vídeos parecem um pouco azuis ou amarelos, experimente um modo diferente.

`camera.brightness = 50`

Isto define o brilho da imagem da câmara, do mais escuro em 0 até ao mais brilhante em 100.

`camera.color_effects = None`

Isto muda o efeito da cor atualmente a ser utilizada pela câmara. Normalmente, não é necessário mexer nesta definição, mas se fornecer um par de números, pode alterar a forma como a câmara grava a cor: tente **`(128, 128)`** para criar uma imagem a preto e branco.

`camera.contrast = 0`

Isto define o contraste da imagem. Um número mais alto fará com que as coisas pareçam mais dramáticas e cruas; um número mais baixo fará com que pareçam mais esbatidas. Pode utilizar qualquer número entre -100 para contraste mínimo e 100 para contraste máximo.

`camera.crop = (0.0, 0.0, 1.0, 1.0)`

Isto permite-lhe cortar a imagem nos lados e na parte superior para capturar apenas a parte necessária. Os números representam as coordenadas X e Y, largura e altura e, por predefinição, capturam a imagem completa. Experimente diminuir os dois últimos números – 0,5 e 0,5 é um bom ponto de partida – para ver que efeito tem esta definição.

`camera.exposure_compensation = 0`

Isto define a *compensação* de exposição da câmara, permitindo-lhe controlar manualmente quanta luz é capturada para cada imagem. Ao contrário da definição de brilho, esta definição controla a própria câmara. Os valores válidos variam entre -25 para uma imagem muito escura e 25 para uma imagem muito brilhante.

`camera.exposure_mode = 'auto'`

Isto define o *modo de exposição*, ou a lógica que o módulo de câmara/câmara HQ utiliza para decidir como deve ser a exposição de cada imagem. Os modos possíveis são: **`off`**, **`auto`**, **`night`**, **`backlight`**, **`spotlight`**, **`sports`**, **`snow`**, **`beach`**, **`verylong`**, **`fixedfps`**, **`antishake`** e **`fireworks`**.

`camera.framerate = 30`

Isto define o número de imagens capturadas para criar um vídeo por segundo, ou a *velocidade de fotogramas*. Uma maior taxa de fotogramas cria um vídeo mais perfeito, mas ocupa mais espaço de armazenamento. As taxas de fotogramas mais altas requerem uma resolução mais baixa para serem utilizadas, que pode ser definida através de **`camera.resolution`**.

`camera.hflip = False`

Isto roda a imagem da câmara no eixo horizontal, ou X, quando definido para **`True`**.

`camera.image_effect = 'none'`

Isto aplica um dos vários efeitos de imagem ao fluxo de vídeo, que será visível na pré-visualização, bem como as imagens e vídeos gravados. Os efeitos possíveis são: **`blur`**, **`cartoon`**, **`colorbalance`**, **`colorpoint`**, **`colorswap`**, **`deinterlace1`**, **`deinterlace2`**, **`denoise`**, **`emboss`**, **`film`**, **`gpen`**, **`hatch`**, **`negative`**, **`none`**, **`oilpaint`**, **`pastel`**, **`posterise`**, **`saturation`**, **`sketch`**, **`solarize`**, **`washedout`** e **`watercolor`**.

`camera.ISO = 0`

Isto altera a definição ISO da câmara, o que afeta a sua sensibilidade à luz. Por predefinição, a câmara ajusta o ISO automaticamente, dependendo da luz disponível. Pode optar por definir o ISO manualmente utilizando um dos seguintes valores: 100, 200, 320, 400, 500, 640, 800. Quanto mais alto o ISO, melhor será o desempenho da câmara em ambientes com pouca luz, mas a imagem ou vídeo capturados terão mais granulosidade.

`camera.meter_mode = 'average'`

Isto controla a forma como a câmara decide a quantidade de luz disponível quando define a sua exposição. A predefinição é a média de quantidade de luz disponível em toda a imagem; outros modos possíveis são **`backlit`**, **`matrix`** e **`spot`**.

`camera.resolution = (1920, 1080)`
Isto define a resolução da imagem ou vídeo capturado, representado por dois números para largura e altura. As resoluções mais baixas ocupam menos espaço de armazenamento e permitem que utilize uma velocidade de fotogramas mais alta; as resoluções mais altas são de melhor qualidade, mas ocupam mais espaço de armazenamento.

`camera.rotation = 0`
Isto controla a rotação da imagem, de 0 graus a 90, 180, e 270 graus. Utilize esta definição se não conseguir posicionar a câmara de modo a que o cabo plano saia pela parte inferior.

`camera.saturation = 0`
Isto controla a saturação da imagem, ou quão vibrantes são as cores. Os valores possíveis situam-se entre -100 e 100.

`camera.sharpness = 0`
Isto controla a nitidez da imagem. Os valores possíveis situam-se entre -100 e 100.

`camera.shutter_speed = 0`
Isto controla a rapidez com que o obturador abre e fecha ao capturar imagens e vídeos. Pode definir a velocidade do obturador manualmente em microssegundos, sendo que as velocidades do obturador mais longas funcionam melhor com luz mais fraca e as velocidades do obturador mais rápidas com luz mais forte. Normalmente, esta oção deve ser deixada na predefinição automática.

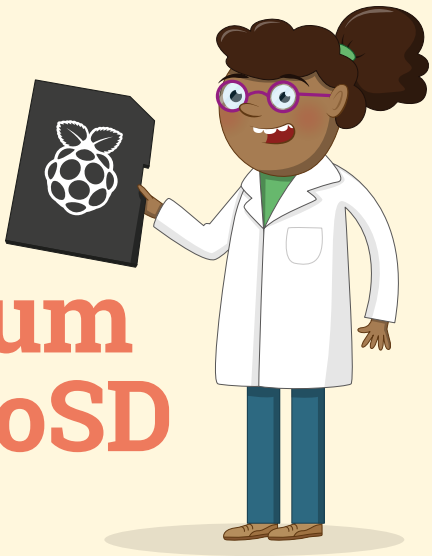
`camera.vflip = False`
Isto roda a imagem da câmara no eixo vertical, ou Y, quando definido para `True`.

`camera.video_stabilization = False`
Quando definido para `True`, ativa a estabilização do vídeo. Esta definição apenas é necessária se o módulo de câmara ou a câmera HQ estiver em movimento enquanto estiver a gravar, como, por exemplo, se estiver ligado a um robô ou a ser transportado, a fim de reduzir o tremor do vídeo capturado.

Mais informações sobre estas definições, bem como sobre definições adicionais não documentadas aqui, encontram-se em **picamera.readthedocs.io**.

Apêndice A

Instalar um sistema operativo num cartão microSD



Pode comprar cartões microSD com NOOBS (Software novo pronto a utilizar) pré-instalado nos mesmos em todos os bons revendedores Raspberry Pi, permitindo-lhe instalar facilmente o Raspberry Pi OS (anteriormente conhecido como Raspbian) no seu Raspberry Pi. Em alternativa, também pode utilizar as seguintes instruções para utilizar o Raspberry Pi Imager para instalar manualmente um sistema operativo no seu próprio cartão microSD vazio (ou reutilizado).

> **AVISO!**
>
> Se comprou um cartão microSD com o NOOBS já pré-instalado, não precisa de fazer mais nada além de ligá-lo ao seu Raspberry Pi. Estas instruções são direcionadas para cartões microSD em branco, ou para cartões usados anteriormente nos quais deseja instalar um novo sistema operativo. Se executar estas instruções num cartão microSD que contenha ficheiros irá perder esses ficheiros, certifique-se de que fez uma cópia de segurança dos seus ficheiros antes de começar!

Transferir o Raspberry Pi Imager

Baseado no Debian, o Raspberry Pi OS é o sistema operativo oficial do Raspberry Pi. A forma mais fácil de instalar o Raspberry Pi OS num cartão microSD para o seu Raspberry Pi é utilizar a ferramenta Raspberry Pi Imager, transferível através de **rpf.io/downloads**. Tenha em atenção que este método substitui a instalação do seu sistema operativo através do NOOBS, embora este último ainda esteja disponível a partir da mesma página de transferências.

A aplicação Raspberry Pi Imager está disponível para computadores Windows, macOS e Ubuntu Linux, por isso escolha a versão relevante para o seu sistema.

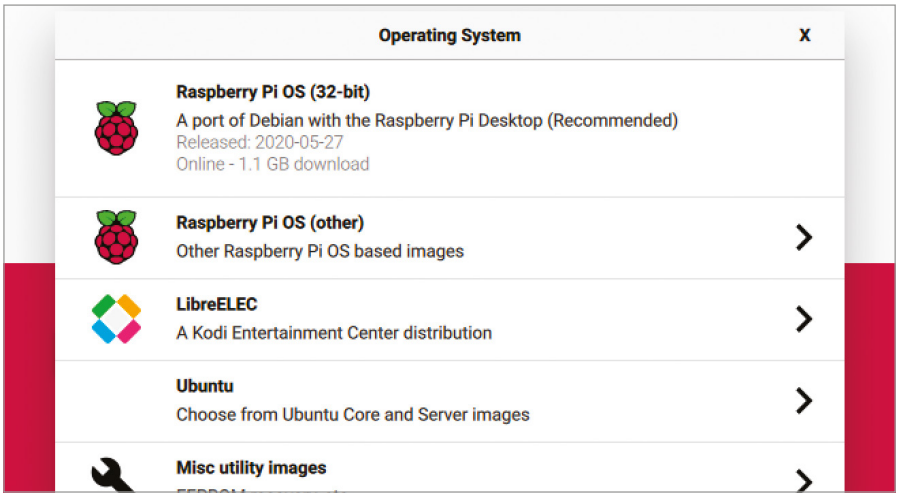
No macOS, clique duas vezes no ficheiro DMG transferido. Pode precisar de alterar as suas definições de Segurança e Privacidade para permitir que as aplicações transferidas da "App Store e programadores identificados" possam ser executadas. Em seguida, basta arrastar o ícone do Raspberry Pi Imager para a pasta Aplicações.

Num PC com Windows, clique duas vezes no ficheiro EXE transferido. Quando solicitado, selecione o botão "Sim" para possibilitar a respetiva execução. Em seguida, clique no botão "Instalar" para iniciar a instalação.

Gravar o SO no cartão microSD

Ligue o seu cartão microSD ao seu PC ou computador Mac: precisará de um adaptador USB para cartões microSD, a menos que o seu computador tenha um leitor de cartões integrado. Tenha em atenção que o cartão não precisa de ser pré-formatado.

Inicie a aplicação Raspberry Pi Imager. Clique no botão "Choose SO" para selecionar o sistema operativo que gostaria de instalar. A melhor opção é o Raspberry Pi OS padrão – se preferir a versão Lite mais simples, ou a versão Full (com todo o software recomendado pré-instalado), selecione "Raspberry Pi OS (outro)". Também há opções para instalar o LibreELEC (escolha a versão para o seu modelo Raspberry Pi) e o Ubuntu Core ou Server.



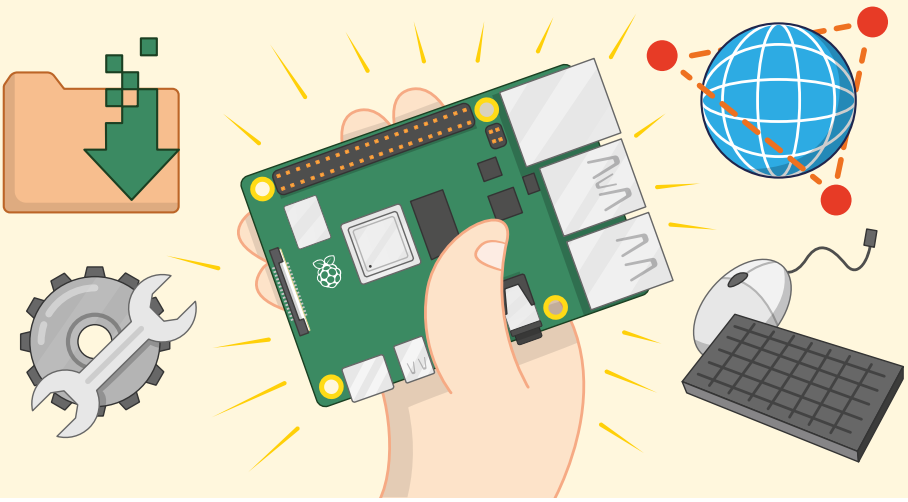
Nota: Se quiser instalar um SO diferente, como o Lakka, basta transferir o respetivo ficheiro de imagem do site relevante e, em seguida, selecionar a opção "Use custom" no Raspberry Pi Imager.

Com um sistema operativo selecionado, clique no botão "Choose SD card" e selecione o seu cartão microSD (normalmente haverá apenas uma opção).

Por fim, clique no botão "Write" e aguarde enquanto o utilitário grava o SO selecionado no seu cartão e, em seguida, verifica-o. Quando estiver concluído, pode remover o cartão microSD. Pode em seguida, introduzi-lo no seu Raspberry Pi e iniciá-lo no sistema operativo que acabou de instalar.

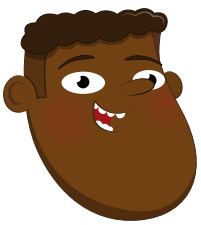
Apêndice B

Instalar e desinstalar o software



O Raspberry Pi OS vem com uma seleção de pacotes de software populares, cuidadosamente selecionados pela Raspberry Pi Foundation, mas estes não são os únicos pacotes que funcionam num Raspberry Pi. Com as seguintes instruções, pode pesquisar por software adicional, instalá-lo e desinstalá-lo novamente para expandir as capacidades do seu Raspberry Pi.

As instruções neste apêndice são adicionais às que encontra no **Capítulo 3, Usar o seu Raspberry Pi**, que explicam como utilizar a ferramenta de Software Recomendado; se ainda não o leu, faça-o antes de usar os métodos descritos neste apêndice.

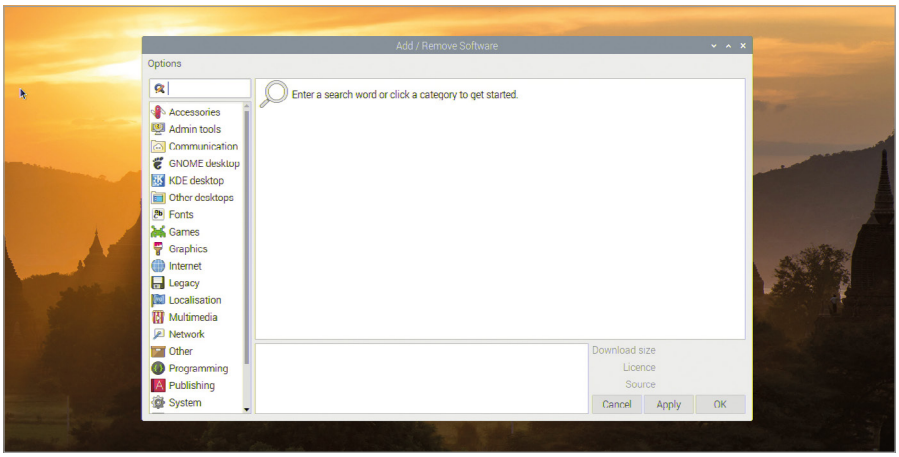


CAPACIDADE DO CARTÃO

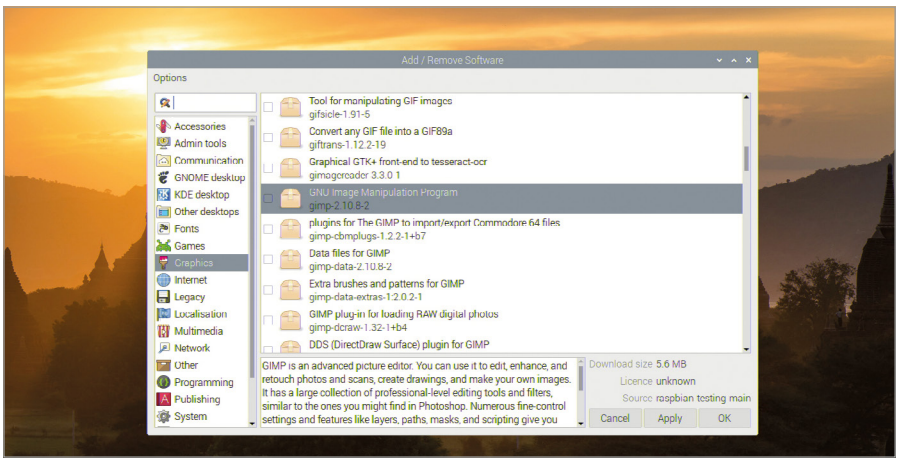
Ao adicionar software ao seu Raspberry Pi, ocupa espaço no seu cartão microSD. Um cartão de 16 GB ou superior permite que instale mais software; para verificar se o cartão que pretende usar é compatível com o Raspberry Pi, visite **rpf.io/sdcardlist**.

Pesquisar software disponível

Para ver e pesquisar na lista de pacotes de software disponíveis para o Raspberry Pi OS, usando o que é conhecido como *repositórios de software*, clique no ícone da framboesa para carregar o menu, selecione a categoria Preferências e, em seguida, clique em Add/Remove Software. Após alguns segundos, é apresentada a janela da ferramenta.



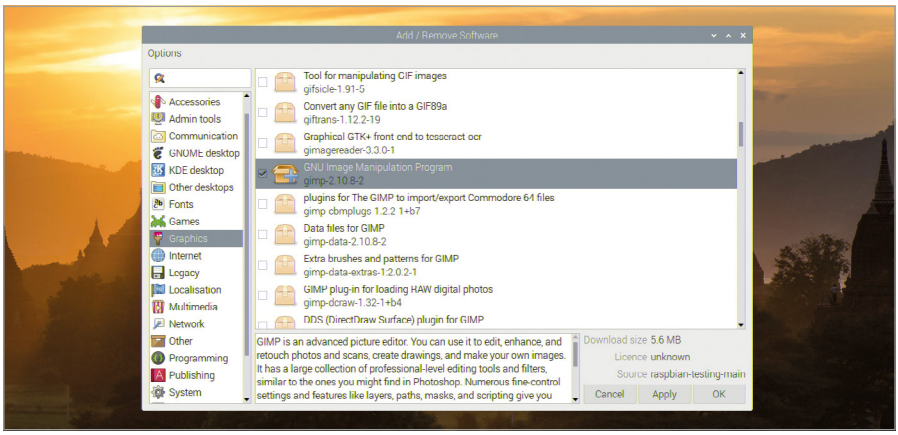
O lado esquerdo da janela Add/Remove Software contém uma lista de categorias – as mesmas categorias que encontra no menu principal quando clica no ícone da framboesa. Ao clicar numa categoria, é apresentada uma lista de software disponível nessa categoria. Pode também introduzir um termo de pesquisa na caixa no canto superior esquerdo da janela, como "'text editor" (editor de texto) ou "games" (jogo), e ver uma lista de pacotes de software correspondentes de qualquer categoria. Ao clicar num pacote, apresenta informações adicionais sobre o mesmo na parte inferior da janela.



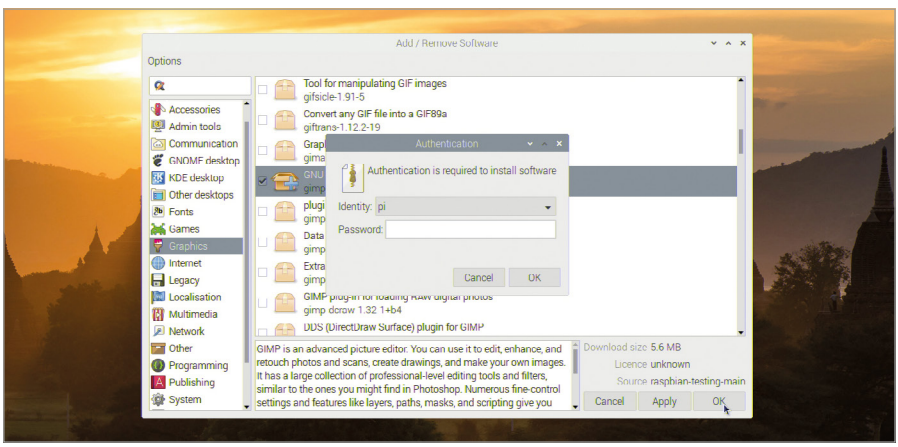
Se a categoria escolhida tiver muitos pacotes de software disponíveis, pode demorar algum tempo até que a ferramenta Adicionar/Remover Software termine de apresentar a lista.

Instalar software

Para selecionar um pacote para instalação, clique na caixa junto do mesmo. Pode instalar mais de um pacote ao mesmo tempo: basta continuar a clicar para adicionar mais pacotes. O ícone ao lado do pacote mudará para uma caixa aberta com o símbolo '+', para confirmar que este será instalado.

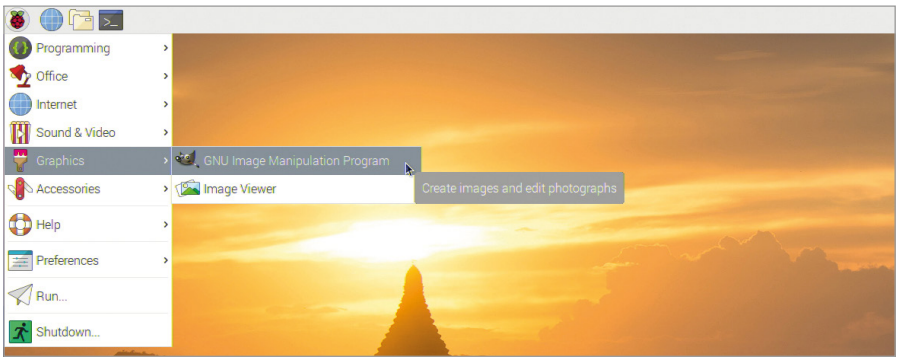


Quando estiver satisfeito com as suas escolhas, clique no botão OK ou Apply (Aplicar); a única diferença é que o botão OK fecha a ferramenta Add/Remove Software quando o seu software estiver instalado, enquanto o botão Apply mantém a ferramenta aberta. Será solicitado que introduza a sua palavra-passe para confirmar a sua identidade – afinal de contas não quer permitir que qualquer pessoa adicione ou remova software do seu Raspberry Pi!



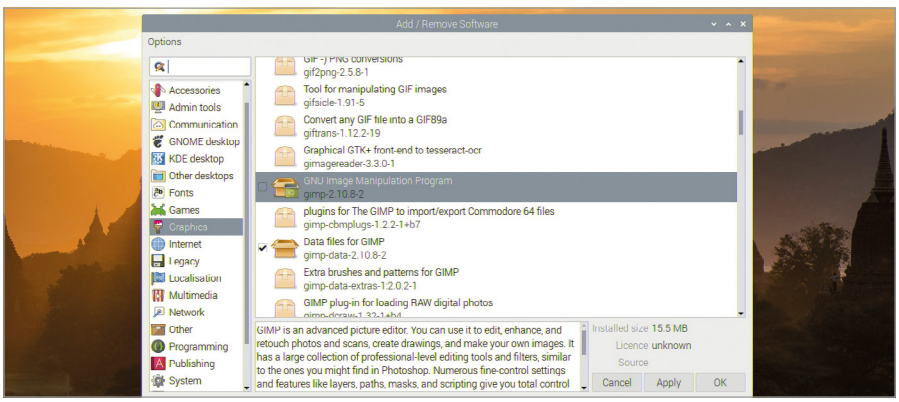
Pode reparar que, quando instala um único pacote, são também instalados outros pacotes; estes pacotes são designados *dependências*, necessários para que o software que instalou funcione corretamente, por exemplo, grupos de efeitos sonoros para um jogo, ou uma base de dados para um servidor Web.

Uma vez instalado o software, pode encontrá-lo clicando no ícone da framboesa para carregar o menu e encontrar a categoria do pacote de software. Tenha em atenção que a categoria do menu nem sempre é igual à categoria da Ferramenta Add/Remove Software, e algum software nem sequer tem uma entrada no menu; este software é conhecido como *software de linha de comandos* e deve ser executado no Terminal. Para obter mais informações sobre a linha de comandos e sobre o Terminal, aceda a **Apêndice C, A interface da linha de comandos**.



Desinstalar software

Para selecionar um pacote para remoção, ou*desinstalação*, localize-o na lista de pacotes – a função de pesquisa é útil aqui – e clique na caixa junto do mesmo para selecionar. Pode desinstalar mais de um pacote ao mesmo tempo: basta continuar a clicar para remover mais pacotes. O ícone junto ao pacote mudará para uma caixa aberta junto a um pequeno caixote do lixo, para confirmar que será desinstalado.



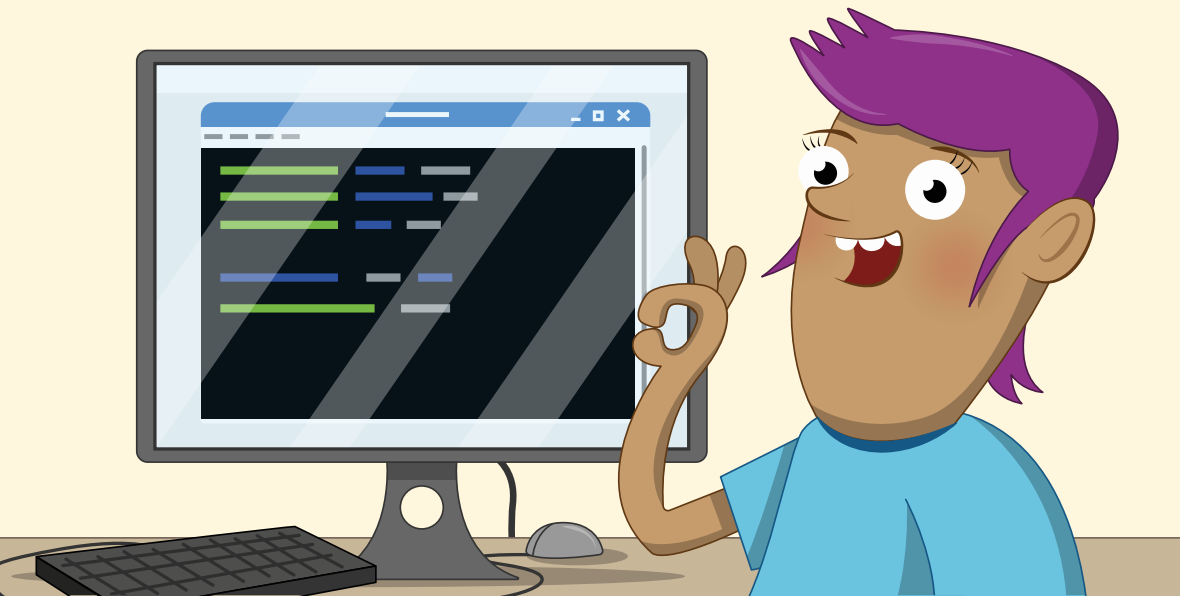
Como anteriormente, pode clicar em OK ou Apply para começar a desinstalar os pacotes de software selecionados. Será solicitado que confirme a sua palavra-passe, a menos que o tenha feito nos últimos minutos, e também que confirme a remoção de quaisquer dependências relacionadas com o seu pacote de software. Quando a desinstalação for concluída, o software desaparecerá do menu do ícone da framboesa, mas os ficheiros que criou com o software, por exemplo, imagens para um pacote gráfico ou os dados guardados de um jogo, não serão removidos.

AVISO!

Todo o software instalado no Raspberry Pi OS aparece em Add/Remove Software, incluindo o software necessário para o seu Raspberry Pi funcionar. É possível remover pacotes até que o computador deixe de funcionar. Para evitar esta situação, não desinstale ficheiros a não ser que tenha a certeza de que já não precisa deles. Se isso já aconteceu, reinstale o Raspberry Pi OS usando as instruções em **Capítulo 2, Começar com o seu Raspberry Pi**, ou reinstale o sistema operativo seguindo as instruções no **Apêndice A**.

Apêndice C

A interface da linha de comandos



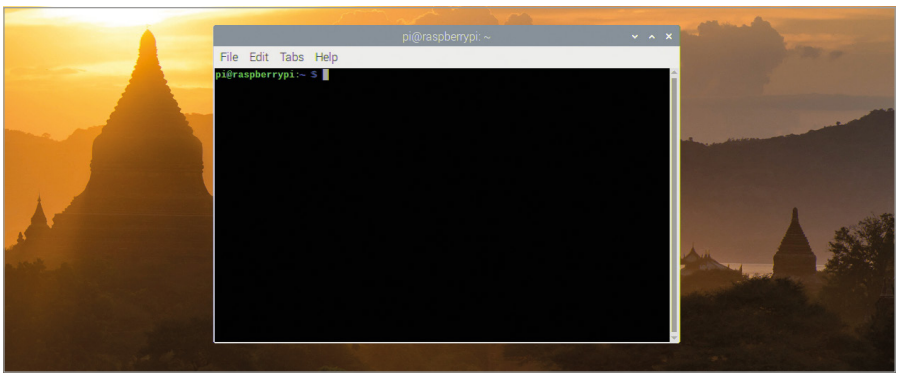
Apesar de poder gerir a maior parte do software num Raspberry Pi através do computador, alguns só podem ser acedidos através de um modo baseado em texto conhecido como a *interface da linha de comandos (CLI)* numa aplicação chamada Terminal. A maioria dos utilizadores nunca irá precisar de usar o CLI, mas para aqueles que querem aprender mais, este apêndice disponibiliza uma introdução básica.

MAIS INFORMAÇÕES

Este apêndice não foi concebido para ser um guia exaustivo da interface da linha de comandos do Linux. Para obter uma visão mais detalhada sobre a utilização do CLI, visite **rpf.io/terminal** num navegador Web.

Carregar o Terminal

O acesso ao CLI é feito através do Terminal, um pacote de software que carrega o que é tecnicamente conhecido como um *terminal de tele-escrita virtual (VTY)*, um nome que remonta aos primórdios dos computadores quando os utilizadores emitiam comandos através de uma enorme máquina de escrever eletromecânica, em vez de um teclado e de um monitor. Para carregar o pacote Terminal, clique no ícone da framboesa para carregar o menu, escolha a categoria Acessórios e, em seguida, clique em Terminal.



A janela do Terminal pode ser arrastada pela área de trabalho, redimensionada, maximizada e minimizada, como qualquer outra janela. Também pode tornar o tamanho do texto maior se for difícil de ver, ou menor se quiser que caiba mais na janela: clique no menu Editar e escolha Ampliar ou Reduzir respetivamente, ou prima a tecla **CTRL** no teclado seguida de **+**ou **-**.

A linha de comandos

O que aparece em primeiro lugar num Terminal é a *linha de comandos*, que está à espera das suas instruções. A janela de um Raspberry Pi a executar o Raspberry Pi OS tem o seguinte aspeto:

```
pi@raspberrypi:~ $
```

A primeira parte da janela, **`pi`**, é o seu nome de utilizador; a segunda parte, a seguir a **`@`**, é o nome do computador que está a usar, que é **`raspberrypi`** por predefinição. A seguir a '**`:`**' há um til, **`~`**, que é uma forma abreviada de se referir ao seu diretório pessoal e representa o seu *diretório de trabalho atual (CWD)*. Por fim, o símbolo do **`$`** indica que o seu utilizador é um *utilizador não privilegiado*, o que significa que precisa de uma palavra-passe para realizar tarefas como adicionar ou remover software.

Explorar

Tente introduzir o seguinte e, em seguida, prima a tecla **ENTER**:

```
cd Desktop
```

Verá a linha de comandos alterar-se para:

```
pi@raspberrypi:~/Desktop $
```

Isso mostra-lhe que o seu diretório de trabalho atual mudou: estava no seu diretório raiz anteriormente, indicado pelo símbolo **~** e agora está no subdiretório **Desktop** por baixo do seu diretório raiz. Para fazer isso, usou o comando **cd** – *alterar diretório*.

MAIÚSCULAS/MINÚSCULAS CORRETAS

A interface da linha de comandos do Raspberry Pi OS é sensível a maiúsculas e minúsculas, o que significa que é importante quando os comandos ou os nomes têm letras maiúsculas e minúsculas. Se recebeu a mensagem "este ficheiro ou diretório não existe" quando tenta mudar de diretório, verifique se tinha um A maiúsculo no início de Ambiente de trabalho.

Há quatro maneiras de voltar para o seu diretório raiz: tente uma de cada vez, alterando para o subdiretório **Desktop** todas as vezes. A primeira é:

```
cd ..
```

Os símbolos **..** são outro atalho, desta vez para o 'diretório acima deste', também conhecido como o *diretório principal*. Como o diretório acima de **Desktop** é o seu diretório raiz, este comando direciona-o para lá. Volte a aceder ao subdiretório **Desktop** e tente a segunda forma:

```
cd ~
```

Este comando utiliza o símbolo **~** e significa literalmente "alterar para o meu diretório raiz". Ao contrário de **cd ..** que apenas o direciona para o diretório principal de qualquer diretório em que esteja atualmente, este comando funciona a partir de qualquer lugar – mas há uma forma mais fácil:

```
cd
```

Sem que seja atribuído o nome de um diretório, o comando **cd** simplesmente está predefinido para voltar para o seu diretório raiz. A última forma de regressar ao seu diretório raiz é escrever:

```
cd /home/pi
```

Este comando usa o que se chama um *caminho absoluto*, que irá funcionar independentemente do diretório de trabalho atual. Por isso, como `cd` por si só ou `cd ~`, irá direcioná-lo para o seu diretório raiz de onde quer que esteja; no entanto, ao contrário dos outros métodos, é necessário o seu nome de utilizador.

Processamento de ficheiros

Para praticar o trabalho com ficheiros, altere para o diretório **Desktop** e escreva o seguinte:

```
touch Test
```

Um ficheiro chamado **Test** aparecerá no ambiente de trabalho. O comando `touch` é usado normalmente para atualizar as informações de data e hora num ficheiro, mas se – como neste caso – o ficheiro não existir, cria-o.

Tente o seguinte:

```
cp Test Test2
```

Outro ficheiro chamado **Test2** aparecerá no ambiente de trabalho. Esta é uma *cópia* do ficheiro original, idêntico em todos os aspetos. Elimine-o ao escrever:

```
rm Test2
```

Este comando *remove* o ficheiro, e irá vê-lo desaparecer.

AVISO!

Ao contrário da eliminação de ficheiros com o Gestor de Ficheiros gráfico, que os guarda no Cesto de papéis para poder recuperá-los mais tarde, os ficheiros eliminados com `rm` não se podem recuperar. Certifique-se de que escreve com atenção!

Em seguida, tente:

```
mv Test Test2
```

Este comando *move* o ficheiro, e verá o seu **Test** original desaparecer e ser substituído por **Test2**. O comando mover, `mv`, pode ser usado desta forma para mudar o nome dos ficheiros.

No entanto, quando não está no ambiente de trabalho, ainda precisa de ser capaz de ver quais os ficheiros que estão num diretório. Tipo:

```
ls
```

Este comando *indica*os conteúdos do diretório atual, ou de qualquer outro diretório fornecido. Para obter mais detalhes, incluindo a listagem de quaisquer ficheiros ocultos e os tamanhos dos ficheiros, tente adicionar alguns interruptores:

```
ls -larth
```

Estes interruptores controlam o comando **ls**: **l** muda a respetiva saída para uma longa lista vertical; **a** manda-lhe mostrar todos os ficheiros e diretórios, incluindo os que geralmente estariam escondidos **r** inverte a ordem normal de ordenação **t** ordena por tempo de modificação, que combinado com o comando **r** dá-lhe os ficheiros mais antigos no topo da lista e os ficheiros mais recentes no fundo; e **h** usa tamanhos de ficheiros legíveis por humanos, tornando a lista mais fácil de compreender.

Executar programas

Alguns programas só podem ser executados na linha de comandos, enquanto outros têm interfaces tanto gráficas como de linha de comandos. Um exemplo destes últimos é a Ferramenta de Configuração do Software Raspberry Pi, que normalmente o utilizador carrega a partir do menu do ícone da framboesa.

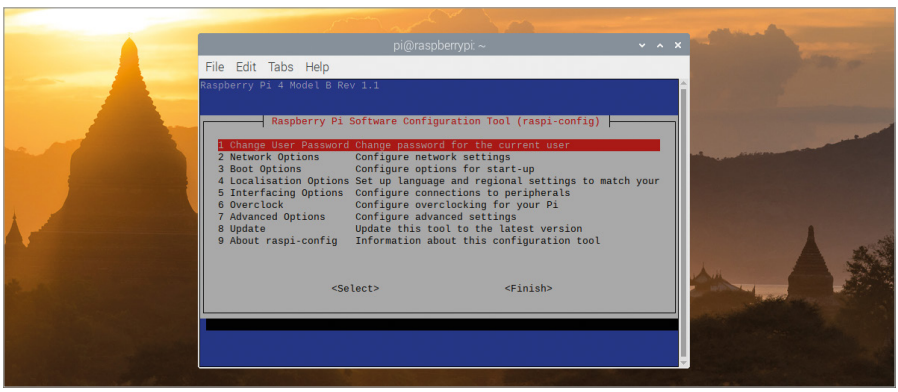
Tipo:

```
raspi-config
```

Receberá um erro indicando que o software só pode ser executado como *raiz*, a conta do superutilizador no seu Raspberry Pi. Também lhe indica como o fazer, ao escrever:

```
sudo raspi-config
```

A parte **sudo** do comando significa *switch-user do* e diz ao Raspberry Pi OS para executar o comando como utilizador raiz.



Só precisará de usar **sudo** quando um programa precisa de *privilégios* elevados como quando está a instalar ou desinstalar software ou a ajustar as definições do sistema. Um jogo, por exemplo, nunca deve ser executado com **sudo**.

Prima a tecla **TAB** duas vezes para selecionar Terminar e prima **ENTER** para sair da Ferramenta de Configuração do Software Raspberry Pi e voltar à interface da linha de comandos. Por fim, escreva:

```
exit
```

Esta ação terminará a sua sessão na interface da linha de comandos e fechará a aplicação Terminal.

Usar as TTYs

A aplicação Terminal não é a única forma de utilizar a interface da linha de comandos: também é possível mudar para um dos vários terminais já em funcionamento conhecidos como *teletipos* ou *TTYs*. Prima sem soltar **CTRL** e **ALT** no seu teclado e prima a tecla **F2** para mudar para 'tty2'.

Precisa de iniciar sessão novamente com o seu nome de utilizador e palavra-passe, após o qual poderá usar a interface da linha de comandos, como no Terminal. Usar estas TTYs é útil quando, por qualquer motivo, a interface principal do computador não está a funcionar.

Para sair da TTY, prima sem soltar **CTRL+ALT**, e, em seguida, prima **F7**: o ambiente de trabalho irá reaparecer. Prima **CTRL+ALT+F2** novamente e irá voltar para 'tty2' - e o que quer que esteja a executar no mesmo ainda estará lá.

Antes de trocar novamente, escreva:

```
exit
```

Em seguida, prima **CTRL+ALT+F7** para voltar para o ambiente de trabalho. O motivo pelo qual terá de sair antes de mudar para a TTY é que qualquer pessoa com acesso ao teclado pode mudar para uma TTY, e se ainda estiver com a sessão iniciada, alguém poderá aceder à sua conta sem terem de saber a sua palavra-passe!

Parabéns: deu os seus primeiros passos na dominação da interface da linha de comandos do Raspberry Pi OS!

Apêndice D

Leitura adicional

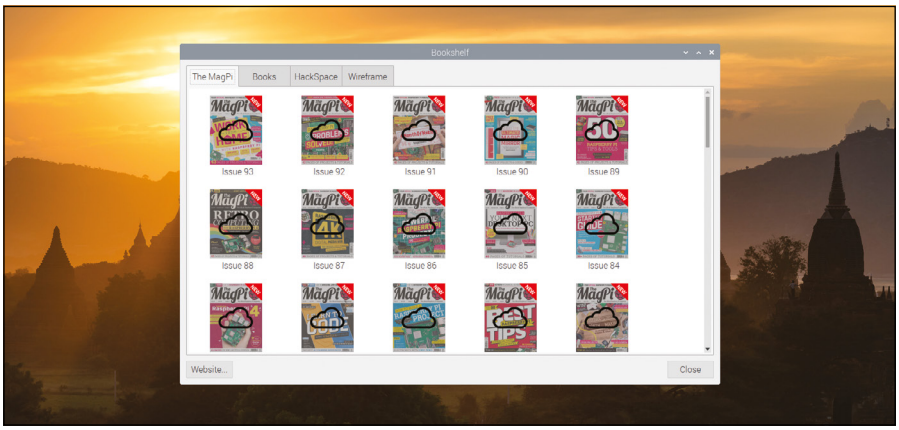
O *guia para Iniciantes do Raspberry Pi oficial* foi projetado para começar a utilizar o seu Raspberry Pi, mas não é, de forma alguma, uma descrição completa sobre tudo o que pode fazer. A comunidade Raspberry Pi é global e vasta, com pessoas usando-os para tudo, desde jogos e aplicações de deteção, até robótica e inteligência artificial, e há muitas ideias por explorar.

Este apêndice destaca algumas fontes de ideias de projetos, planos de aula e outros materiais que podem funcionar como o seu próximo grande passo, agora que conhece o *Guia para iniciantes*.

Bookshelf

Menu Raspberry > Help > Bookshelf

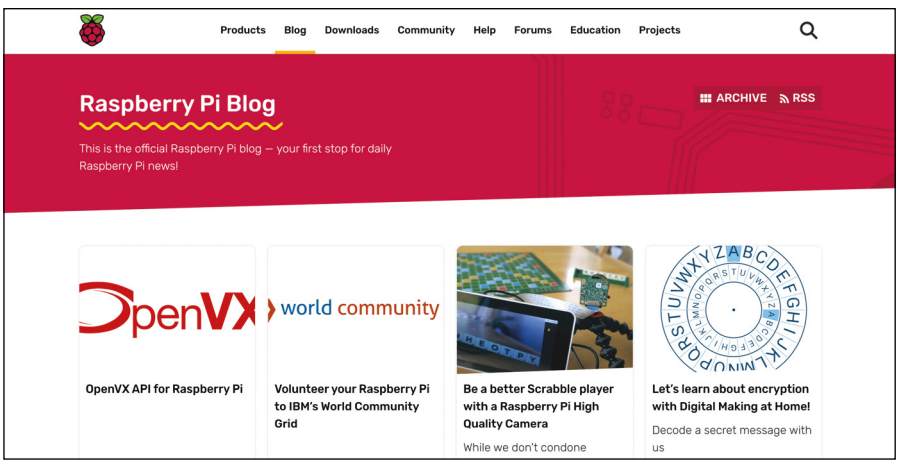
Bookshelf é uma aplicação incluída com o Raspberry Pi OS que o ajuda a procurar, transferir e ler versões digitais das publicações da Raspberry Pi Press – incluindo este *Guia para iniciantes do Raspberry Pi*. Basta carregá-lo clicando no menu do ícone da framboesa, selecione Ajuda e clique em Bookshelf; em seguida, procure entre as revistas e livros, todos gratuitos, para transferir e ler à sua vontade.



O blogue do Raspberry Pi

rpf.io/blog

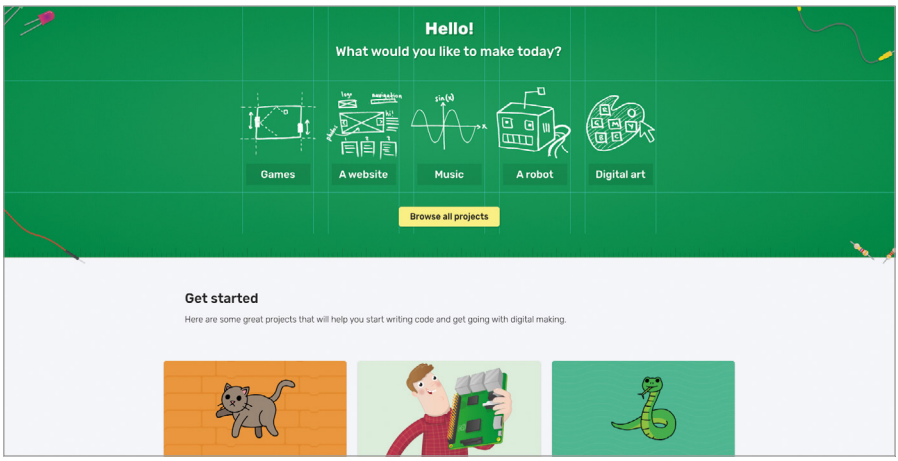
A sua primeira paragem para as últimas notícias sobre o Raspberry Pi, o blogue oficial abrange tudo, desde novos lançamentos de hardware e material educativo, até aos melhores projetos, campanhas e iniciativas da comunidade. Se quer manter-se atualizado sobre o Raspberry Pi, aqui encontra tudo o que precisa.



Projetos do Raspberry Pi

rpf.io/projects

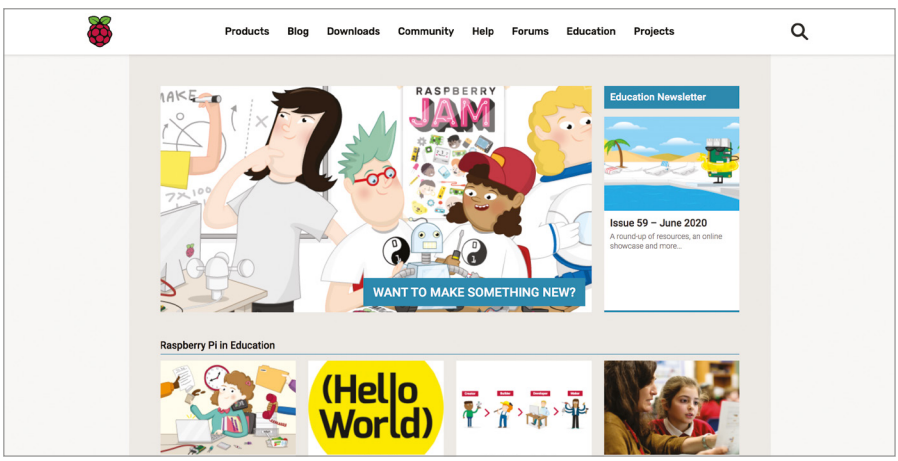
O site oficial de Projetos do Raspberry Pi oferece tutoriais passo a passo de projetos em várias categorias, desde criar jogos e músicas, até construir o seu próprio site ou robô com Raspberry Pi. A maioria dos projetos também está disponível em vários idiomas e abrangem vários níveis de dificuldade adequados para todos, desde iniciantes até criadores experientes.



Educação Raspberry Pi

rpf.io/education

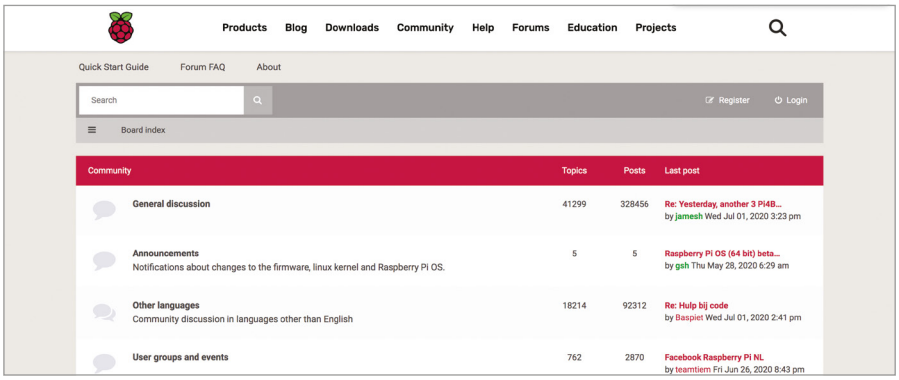
O site oficial de Educação do Raspberry Pi oferece boletins informativos, formação online e projetos direcionados para educadores. O site também inclui ligações para recursos adicionais, incluindo o programa de formação Picademy, programas orientados por voluntários para programação Code Club e CoderDojo, e eventos globais Raspberry Jam.



Os fóruns do Raspberry Pi

rpf.io/forums

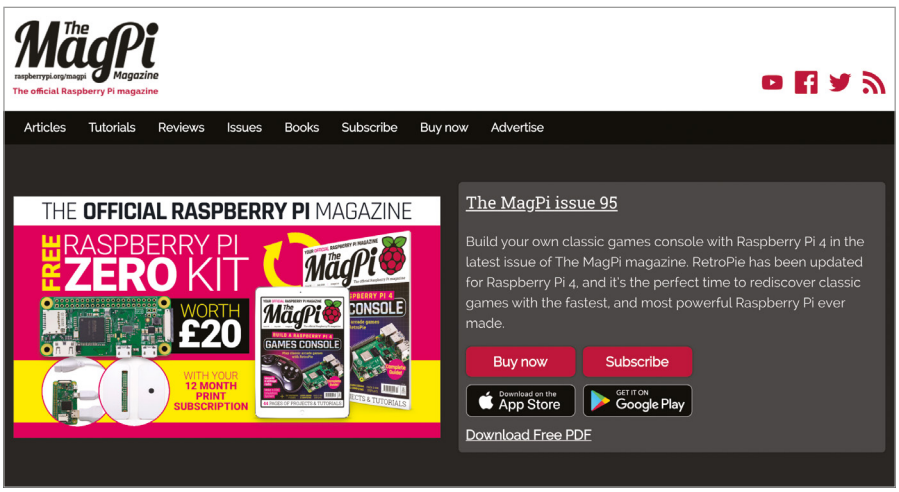
Os fóruns Raspberry Pi são os locais de reunião para os fãs do Raspberry Pi onde podem conversar sobre tudo, desde assuntos de iniciantes até tópicos profundamente técnicos – e há até uma área "sem tópico" para conversas gerais!



A revista MagPi

magpi.cc

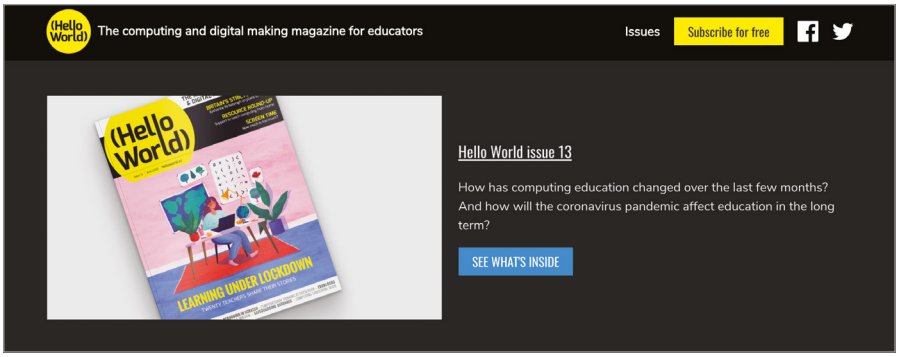
A revista oficial Raspberry Pi, The MagPi, é uma elegante publicação mensal que inclui desde tutoriais e guias até avaliações e notícias, apoiada em grande parte pela comunidade mundial do Raspberry Pi. Os exemplares estão disponíveis em todos os bons quiosques e supermercados, e também podem ser transferidos gratuitamente em formato digital sob a licença Creative Commons. The MagPi também publica livros e revistas sobre diversos temas, que estão disponíveis para compra em formato impresso ou para transferência gratuita.



A revista Hello World

helloworld.cc

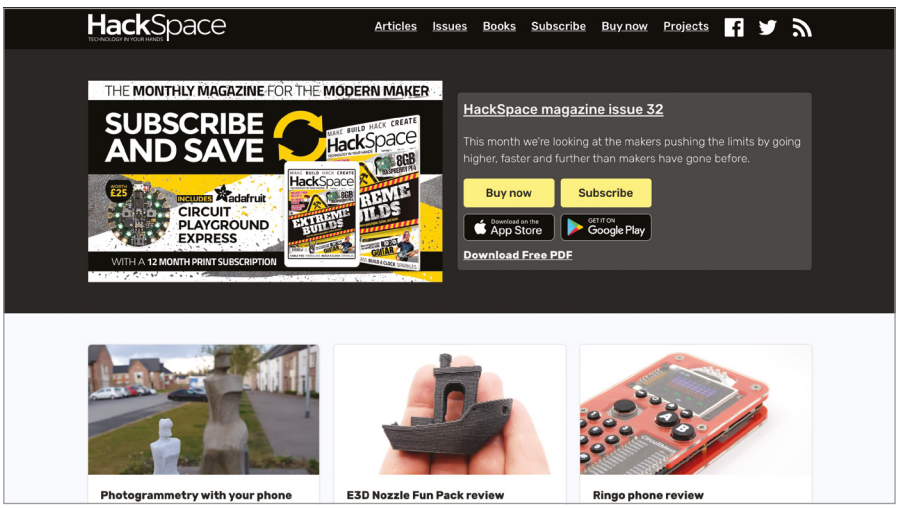
Publicada três vezes por ano, a Hello World está disponível gratuitamente para professores, voluntários e bibliotecários no Reino Unido. Para todos os outros, podem transferir cópias digitais gratuitas sob a licença Creative Commons, estão também disponíveis subscrições comerciais da versão impressa.



A revista HackSpace

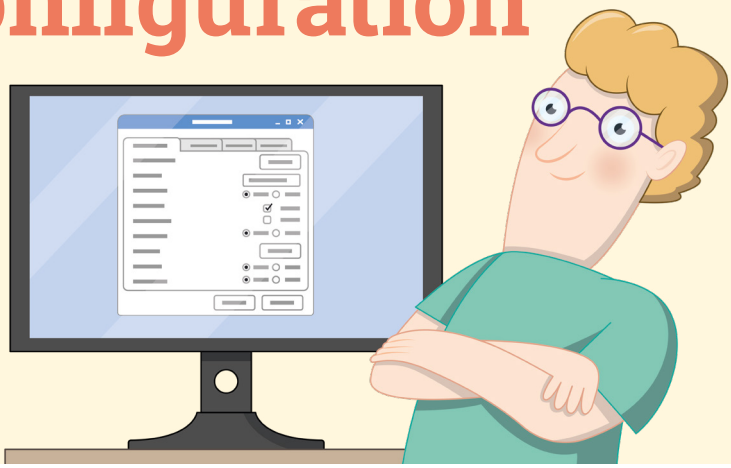
hsmag.cc

Dirigido a um público mais amplo do que The MagPi, a revista HackSpace aborda a comunidade de criadores com críticas de hardware e software, tutoriais e entrevistas. Se está interessado em ampliar os seus horizontes para além do Raspberry Pi, a revista HackSpace é um ótimo lugar para começar. Pode encontrá-la no formato de papel em supermercados e quiosques ou transferi-la gratuitamente em formato digital.



Apêndice E

A ferramenta Raspberry Pi Configuration



A Ferramenta Raspberry Pi Configuration é um poderoso pacote para ajustar as inúmeras definições no seu Raspberry Pi, desde as interfaces disponíveis nos programas até ao respetivo controlo através de uma rede. No entanto, pode ser um pouco assustador para os recém-chegados, por isso, este apêndice irá guiá-lo através de cada uma das definições e explicar os respetivos objetivos.

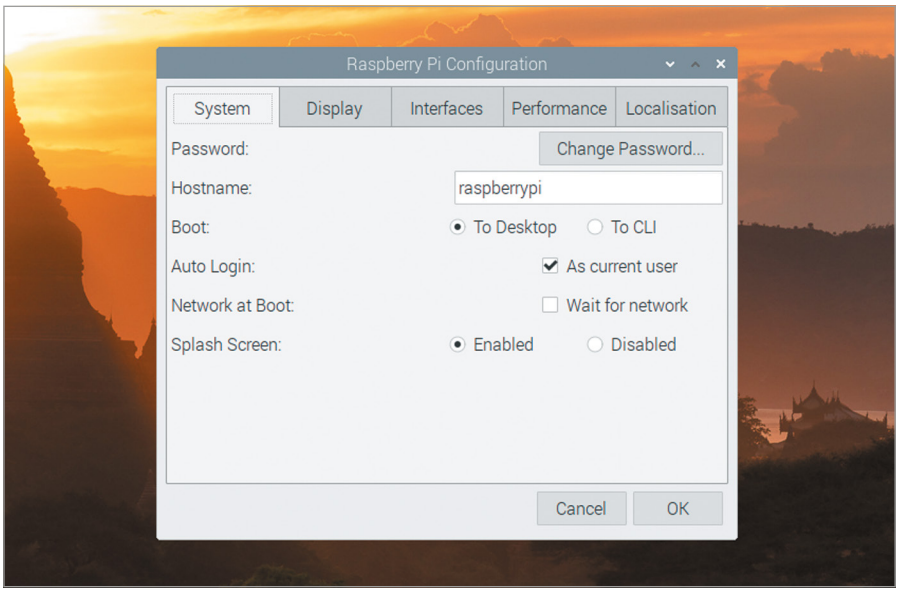
Pode carregar a Ferramenta Raspberry Pi Configuration através do menu do ícone da framboesa, na categoria Preferências. Também pode ser executado a partir da interface da linha de comandos ou do Terminal, com o comando **`raspi-config`**. Os esquemas da versão da linha de comandos e da versão gráfica são diferentes, com opções que aparecem em diferentes categorias, dependendo da versão que usa; este apêndice é baseado na versão gráfica.

AVISO!

A não ser que precise de alterar uma determinada definição, é melhor não usar a Ferramenta Raspberry Pi Configuration. Se estiver a adicionar novo hardware ao seu Raspberry Pi, como um HAT de áudio ou um Camera Module, as instruções indicam-lhe a definição que deve ser alterada; caso contrário, as definições predefinidas geralmente não devem ser alteradas.

Separador System

O separador System contém opções que controlam várias definições de sistema do Raspberry Pi OS.



- **Password:** Clique no botão "Change Password..." para definir uma nova palavra-passe para a sua conta de utilizador atual. Por predefinição esta é a conta "pi".

- **Hostname:** O nome pelo qual se identifica um Raspberry Pi nas redes. Se tiver mais de um Raspberry Pi na mesma rede, cada um deles deve ter o seu próprio nome exclusivo.

- **Boot:** Definir esta opção para "To Desktop" (a predefinição) carrega o ambiente de trabalho familiar do Raspberry Pi OS; defini-lo para "To CLI" carrega a interface da linha de comandos como descrito em **Apêndice C, Ainterface da Linha de Comandos**.

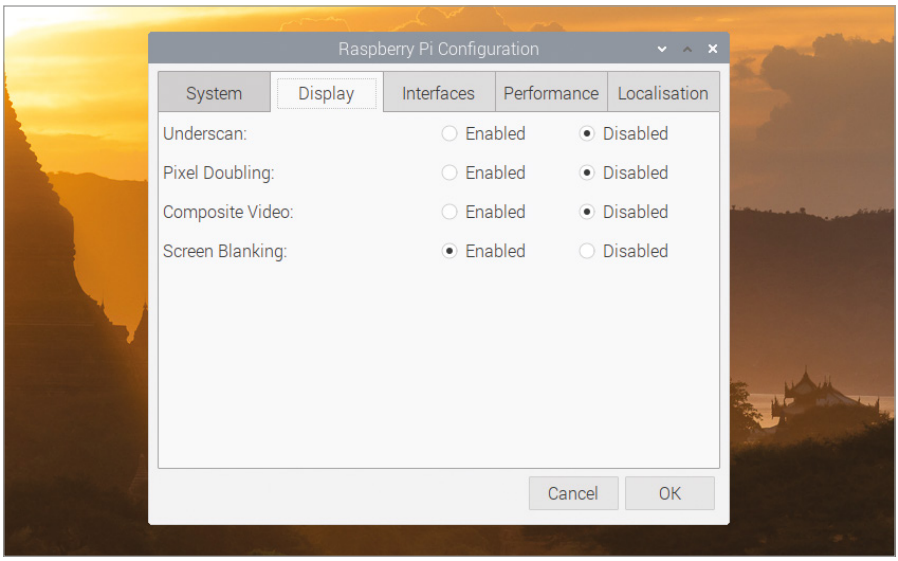
- **Auto Login:** Quando "Login as user "pi"" estiver marcado (a predefinição), o Raspberry Pi OS carrega o ambiente de trabalho sem precisar que introduza o seu nome de utilizador e palavra-passe.

- **Network at Boot:** Quando estiver marcado "Wait for network", o Raspberry Pi OS não carregará até que tenha uma ligação de rede a funcionar.

- **Splash Screen:** Quando definido como "Enabled" (a predefinição), as mensagens de inicialização do Raspberry Pi OS ficam ocultas atrás de um ecrã inicial gráfico.

Separador Display

O separador Display inclui as definições que controlam como o ecrã é apresentado.



- **Overscan:**Esta definição controla se a saída de vídeo no Raspberry Pi inclui ou não barras pretas em torno das respetivas margens, para compensar a proporção de muitas TVs. Se vir barras pretas, defina como "Disabled"; se não, mantenha "Enabled."

- **Pixel Doubling:**Se estiver a usar um ecrã de alta resolução, mas de tamanho pequeno, pode ligar a Pixel Doubling para tornar tudo maior e mais fácil de ver no ecrã.

- **Composite Video:**Esta funcionalidade controla a saída de vídeo composto disponível na ficha combinada de áudio-vídeo (AV), quando for usada com uma ficha para auscultadores (TRRS). Se quiser usar a saída de vídeo composto em vez de HDMI, defina esta opção como "Enabled"; caso contrário, deixe-a desativada.

- **Screen Blanking:** Esta opção permite ligar e desligar o ecrã em branco (o tempo limite que desliga o ecrã após alguns minutos).

Separador Interfaces

O separador Interfaces contém definições que controlam as interfaces de hardware disponíveis no Raspberry Pi.

- **Camera:** Ativa ou desativa a Camera Serial Interface (CSI), para utilização com um Raspberry Pi Camera Module.

- **SSH:** Ativa/desativa a interface Secure Shell (SSH); permitindo que abra uma interface da linha de comandos no Raspberry Pi através de outro computador na sua rede utilizando um cliente SSH.

- **VNC:** Ativa/desativa a interface de Virtual Network Computing (VNC); permitindo que veja uma interface da linha de comandos no Raspberry Pi através de outro computador na sua rede utilizando um cliente VNC.

- **SPI:** Ativa ou desativa a Serial Peripheral Interface (SPI), usada para controlar alguns suplementos de hardware que se ligam aos pinos GPIO.

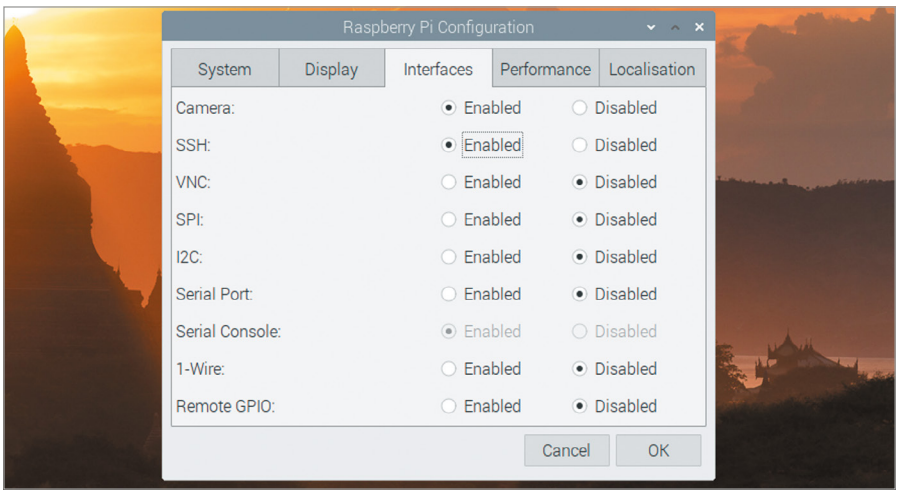
- **I2C:** Ativa ou desativa a interface Inter-Circuito Integrado (I²C), usada para controlar alguns suplementos de hardware que se ligam aos pinos GPIO.

- **Serial Port:** Ativa ou desativa a porta de série do Raspberry Pi, que está disponível nos pinos da GPIO.

- **Serial Console:** Ativa ou desativa a consola de série, uma interface da linha de comandos que está disponível na porta de série. Esta opção só está disponível se a definição da Serial Port acima estiver definida como Ativado.

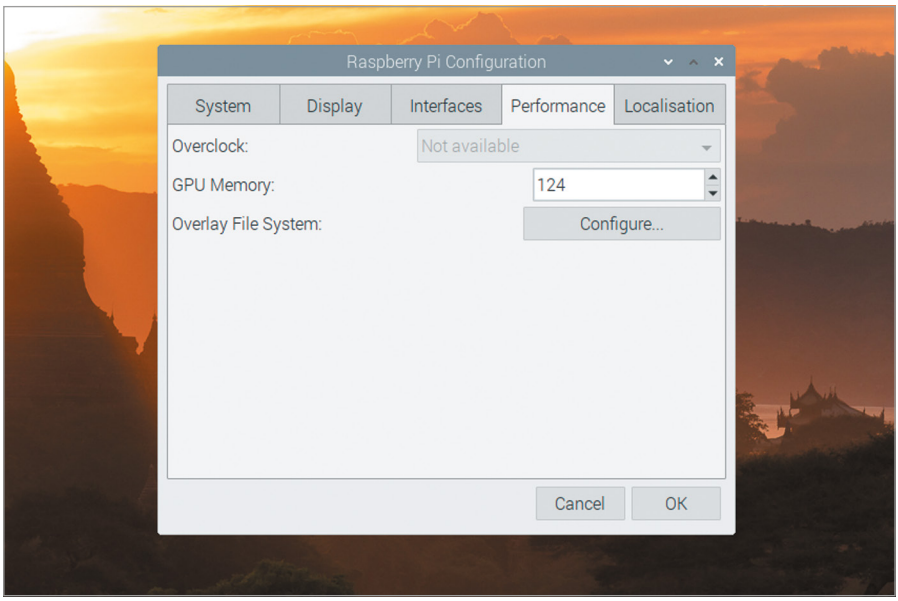
- **1-Wire:** Ativa ou desativa a interface 1-Wire, usada para controlar alguns suplementos de hardware que se ligam aos pinos GPIO.

- **Remote GPIO:** Ativa ou desativa um serviço de rede que lhe permite controlar os pinos GPIO do Raspberry Pi através de outro computador na sua rede utilizando a biblioteca GPIO Zero. Estão disponíveis mais informações sobre o GPIO remoto em **gpiozero.readthedocs.io**.



Separador Performance

O separador Performance contém definições que controlam a quantidade de memória disponível e a velocidade com que o processador do Raspberry Pi é executado.



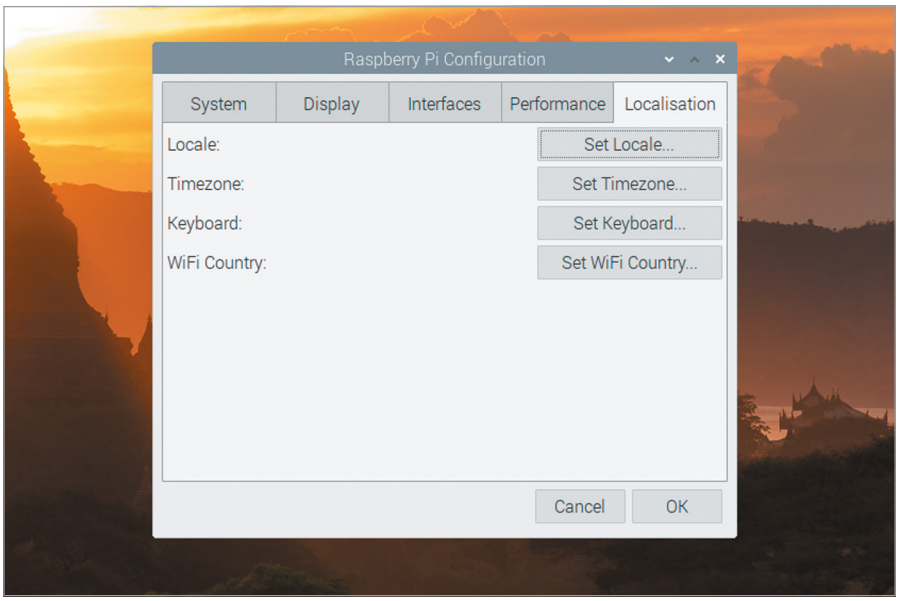
- **Overclock:** Permite-lhe escolher entre uma gama de definições que aumentam o desempenho do seu Raspberry Pi à custa de um maior consumo de energia, geração de calor e possível diminuição em geral do tempo útil de vida. Não está disponível em todos os modelos do Raspberry Pi.

- **GPU Memory:** Permite-lhe definir a quantidade de memória reservada para utilização pelo processador gráfico do Raspberry Pi. Os valores superiores à predefinição podem melhorar o desempenho em tarefas de composição 3D complicadas e de GPGPU (GPU de propósito geral) à custa da redução da memória disponível para o Raspberry Pi OS; os valores mais baixos podem melhorar o desempenho em tarefas de memória intensiva à custa de tornar a composição 3D, a câmara e as funcionalidades de reprodução de vídeo selecionados mais lentos ou mesmo ficarem indisponíveis.

- **Overlay File System:** Permite bloquear o sistema de ficheiros do Raspberry Pi para que as alterações só sejam feitas num disco RAM virtual em vez de serem gravadas no cartão microSD, para que possa voltar a um estado limpo sempre que reiniciar.

Separador Localisation

O separador Localisation contém definições que controlam em que região o seu Raspberry Pi foi projetado para funcionar, incluindo definições do esquema de teclado.



- **Locale:** Permite-lhe escolher a sua região, uma definição do sistema que inclui o idioma, o país e um conjunto de carateres. Tenha em atenção que alterar aqui o idioma só alterará o idioma apresentado nas aplicações para as quais está disponível uma tradução.

- **Timezone:** Permite-lhe escolher o seu fuso horário regional, selecionando uma área do mundo seguida da cidade mais próxima. Se o seu Raspberry Pi está ligado à rede mas o relógio apresenta a hora errada, esta situação geralmente é causada quando é selecionado o fuso horário incorreto.

- **Keyboard:** Permite-lhe escolher o seu tipo de teclado, idioma e esquema. Se achar que o seu teclado introduz as letras ou símbolos incorretamente, pode corrigi-lo aqui.

- **WiFi Country:** Permite-lhe definir o seu país para fins de regulamentação de rádio. Certifique-se de que seleciona o país em que o seu Raspberry Pi está a ser usado: selecionar um país diferente pode impossibilitar a ligação a pontos de acesso Wi-Fi próximos e pode ser uma violação da lei da radiodifusão. Tem de definir um país antes de poder utilizar o rádio Wi-Fi.

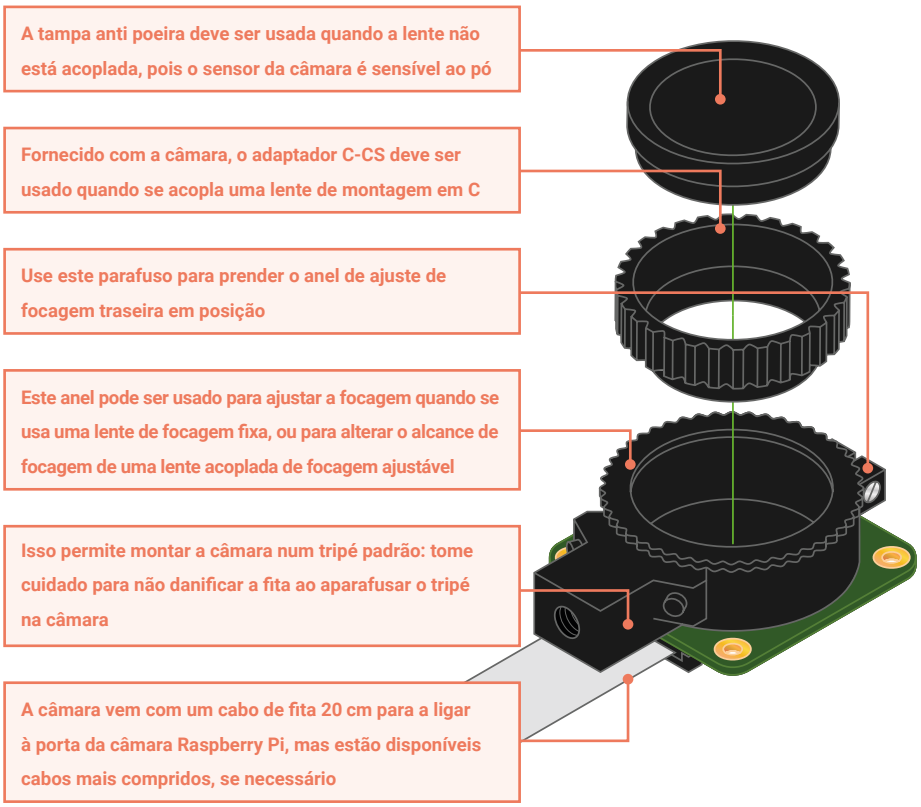
Apêndice F

Configuração da Câmara de Alta Qualidade

Câmara de Alta Qualidade (câmara de alta qualidade) pode capturar imagens com maior resolução do que o módulo de câmara padrão. Ao contrário desta última, não tem uma lente incorporada. Em vez disso, pode ser usada com qualquer lente padrão de montagem em C ou CS; estão disponíveis para aquisição com a câmara lentes de 6 mm ou de 16 mm para ajudá-lo a começar.

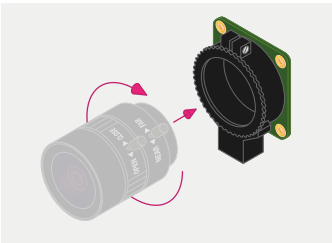
Lente de montagem CS de 6 mm

Está disponível para a HQ Camera uma lente de baixo custo de 6 mm. Esta lente é adequada para fotografia básica. Também pode ser usado para fotografia macro porque pode focar objetos a distâncias muito curtas.



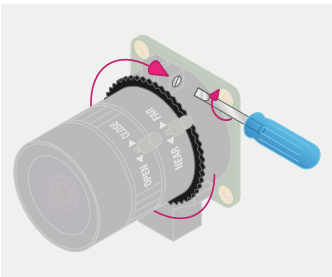
01 Instalar a lente

A lente de 6 mm é um dispositivo de montagem CS, por isso não necessita do anel adaptador C-CS (ver diagrama acima). Não focará corretamente se o adaptador estiver instalado, por isso, se necessário, remova-o. Em seguida, rode a lente no sentido dos ponteiros do relógio por completo até ao anel de ajuste de focagem traseira.



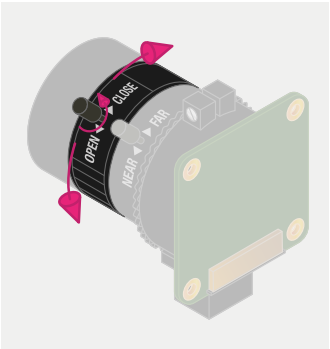
02 Anel de ajuste de focagem traseira e parafuso de blocagem

O anel de ajuste de focagem traseira deve ser aparafusado por completo para obter a menor distância de focagem traseira possível. Use o parafuso de bloqueio de focagem traseira para se certificar de que este não sai desta posição ao ajustar a abertura ou foco.



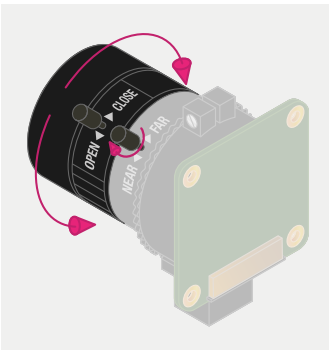
03 Abertura

Para ajustar a abertura, segure a câmara com a lente na direção oposta a si. Rode o anel central enquanto segura com firmeza o anel externo, o mais afastado possível da câmara. Rode no sentido dos ponteiros do relógio para fechar a abertura e reduzir a luminosidade da imagem. Rode no sentido contrário ao dos ponteiros do relógio para abrir a abertura. Assim que estiver satisfeito com o nível de luminosidade, aperte o parafuso na parte lateral da lente para bloquear a abertura.



04 Focagem

Em primeiro lugar, bloqueie o anel de focagem interno, rotulado de "NEAR ◀▶ FAR", na posição ao aparafusar o respetivo parafuso. Agora, segure na câmara com a lente na direção oposta a si. Segure os dois anéis exteriores da lente e rode-os ambos no sentido dos ponteiros do relógio até à imagem entrar em foco – serão necessárias quatro ou cinco voltas completas. Para ajustar o foco, gire os dois anéis externos no sentido horário para focar um objeto próximo. Rode-os no sentido contrário ao sentido dos ponteiros do relógio para focar um objeto distante. Pode ser necessário ajustar a abertura novamente.

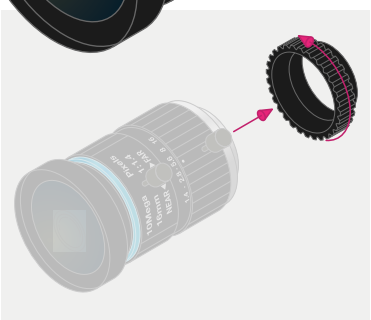


Lente de montagem C de 16 mm

A lente de 16 mm fornece uma imagem com qualidade superior à lente de 6 mm. Tem um ângulo de visão estreito que é mais adequado para visualizar objetos distantes.

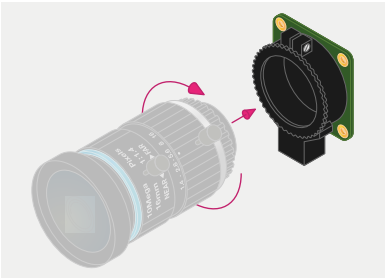
01 Ajustar o Adaptador C-CS

Certifique-se de que o adaptador C-CS incluído com a câmara HQ está montado na lente de 16 mm. A lente é um dispositivo de montagem em C, por isso, tem uma focagem traseira mais distante do que a lente de 6 mm e, portanto, necessita do adaptador.



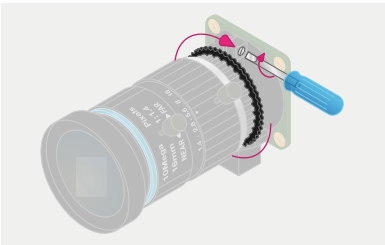
02 Instalar a lente na câmara

Rode a lente de 16 mm no sentido dos ponteiros do relógio por completo até ao anel de ajuste de focagem traseira.



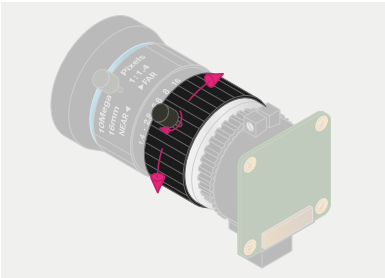
03 Anel de ajuste de focagem traseira e parafuso de bloqueio

O anel de ajuste de focagem traseira deve ser aparafusado por completo. Use o parafuso de bloqueio de focagem traseira para se certificar de que este não sai desta posição ao ajustar a abertura ou foco.



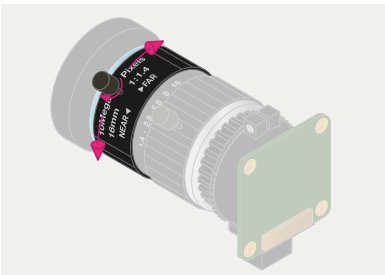
04 Abertura

Para ajustar a abertura, segure a câmara com a lente na direção oposta a si. Rode o anel interno, mais próximo da câmara, enquanto segura a câmara com firmeza. Rode no sentido dos ponteiros do relógio para fechar a abertura e reduzir a luminosidade da imagem. Rode no sentido contrário ao dos ponteiros do relógio para abrir a abertura. Assim que estiver satisfeito com o nível de luminosidade, aperte o parafuso na parte lateral da lente para bloquear a abertura na posição.



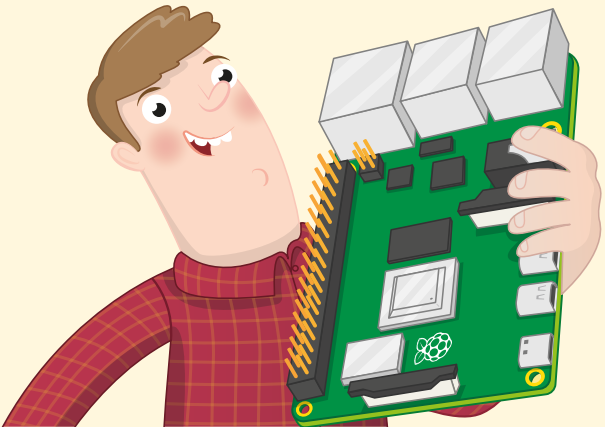
05 Focagem

Para ajustar a focagem, segure a câmara com a lente na direção oposta a si. Rode o anel de focagem com a etiqueta "NEAR ◀▶ FAR", no sentido contrário aos ponteiros do relógio para focar um objeto perto. Rode-os no sentido contrário ao sentido dos ponteiros do relógio para focar um objeto distante. Pode ser necessário ajustar a abertura novamente.



Apêndice G

Especificações do Raspberry Pi

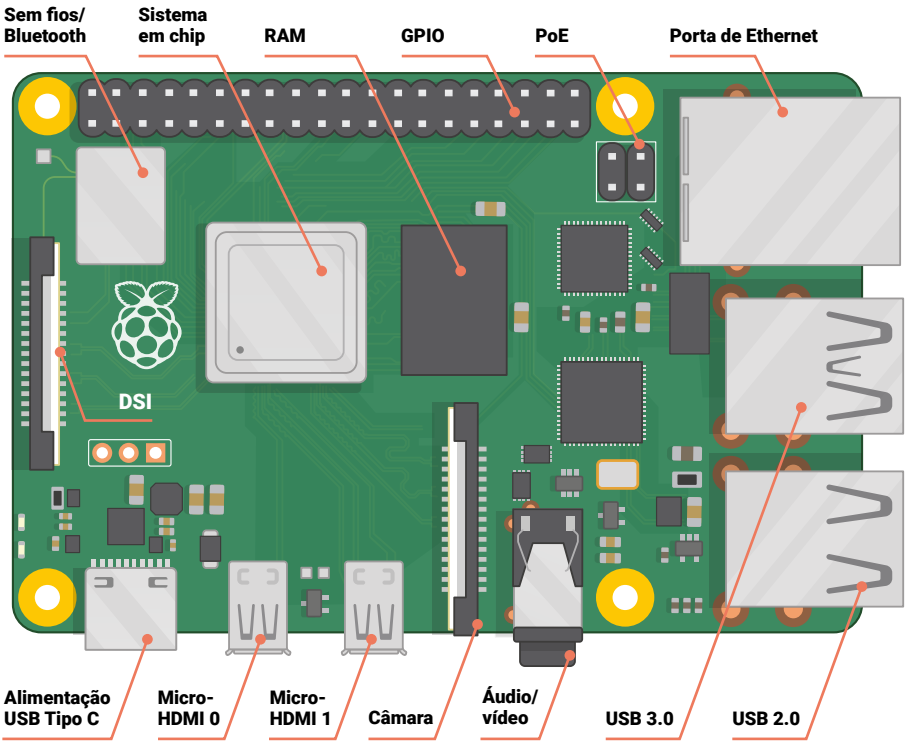


Os vários componentes e caraterísticas de um computador são conhecidos como as suas especificações, e a consulta das especificações indica-lhe as informações necessárias para comparar dois computadores. Estas especificações podem parecer complicadas ao princípio, são muito técnicas, e não é precisa conhecê-las para usar um Raspberry Pi, mas estão incluídas aqui para o leitorinteressado.

O sistema em chip do Raspberry Pi 4 Modelo B e Raspberry Pi 400 é um Broadcom BCM2711B0, que aparece escrito na tampa metálica se olhar com atenção (no Raspberry Pi 4). Inclui uma unidade de processamento central (CPU) de quatro núcleos ARM Cortex-A72 de 64 bits, funcionando cada um a 1,5GHz ou 1,8 GHz (1,5 ou 1,8 mil milhões de ciclos por segundo), e uma unidade de processamento gráfico (GPU) Broadcom VideoCore VI (Seis) funcionando a 500 MHz (500 milhões de ciclos por segundo) para tarefas de vídeo e para tarefas de renderização 3D, como jogos.

O sistema em chip está ligado à RAM (memória de acesso aleatório) LPDDR4 (Low-Power Double-Data-Rate 4) de 2 GB, 4 GB ou 8 GB (dois, quatro, ou oito mil milhões de bytes) – 4 GB no Raspberry Pi 400 – que funciona a 3200 MHz (três mil e duzentos milhões de ciclos por segundo). Esta memória é partilhada entre o processador central e o processador gráfico. A ranhura do cartão microSD suporta até 512 GB (512 mil milhões de bytes) de armazenamento.

A porta de Ethernet suporta ligações até ao gigabit (1000 Mbps, 1000-Base-T), enquanto que a ligação de rádio suporta redes Wi-Fi 802.11ac nas bandas de frequência 2,4 GHz e 5 GHz, ligações Bluetooth 5.0 e Bluetooth Low Energy (BLE).



Divididas numa lista de itens, as especificações do Raspberry Pi 4 têm o seguinte aspeto:

- **CPU:** ARM Cortex-A72 quad-core de 64 bits a 1,5 GHz
- **GPU:** VideoCore VI a 500 MHz
- **RAM:** 1 GB, 2 GB ou 4 GB de LPDDR4
- **Redes:** Gigabit Ethernet, 802.11ac de banda dupla, Bluetooth 5.0, Bluetooth Low Energy
- **Saídas de áudio/vídeo:** Tomada AV analógica de 3,5 mm, 2 × micro-HDMI 2.0
- **Ligação de periféricos:** 2 × portas USB 2.0, 2 × portas USB 3.0, Interface serial de câmara, Interface serial de ecrã (DSI)
- **Armazenamento:** microSD, até 512 GB
- **Alimentação:** 5 volts a 3 amperes por USB Tipo C
- **Extras:** Conetor GPIO de 40 pinos, compatibilidade Power over Ethernet (com hardware adicional)

As especificações do Raspberry Pi 400 são:

- **CPU:** ARM Cortex-A72 quad-core de 64 bits a 1,8 GHz
- **GPU:** VideoCore VI a 500 MHz
- **RAM:** 4 GB de LPDDR4
- **Redes:** Gigabit Ethernet, 802.11ac de banda dupla, Bluetooth 5.0, Bluetooth Low Energy
- **Saídas de áudio/vídeo:** 2 × micro-HDMI 2.0
- **Ligação de periféricos:** 1 × porta USB 2.0, 2 × portas USB 3.0
- **Armazenamento:** microSD, até 512 GB (16 GB fornecido)
- **Alimentação:** 5 volts a 3 amperes por USB Tipo C
- **Extras:** Conetor GPIO de 40 pinos

Apêndice H

Guia de utilizador e segurança do Raspberry Pi

Raspberry Pi

Desenhado e distribuído por
Raspberry Pi Trading Ltd
Maurice Wilkes Building
Cowley Road
Cambridge
CB4 0DS
REINO UNIDO
www.raspberrypi.org

Raspberry Pi Conformidade regulamentar e informações de segurança

Raspberry Pi 4 Modelo B
ID DA FCC: 2ABCB-RPI4B
ID DA IC: 20953-RPI4B

Raspberry Pi 400
ID DA FCC: 2ABCB-RPI400
ID DA IC: 20953-RPI400

IMPORTANTE: Consulte as instruções de instalação antes de ligar à fonte de alimentação em **www.raspberrypi.org/safety**

AVISO: Perigos de cancro e saúde reprodutiva – www.P65Warnings.ca.gov.

Todas as informações e certificados regulamentares podem ser encontrados em **www.raspberrypi.org/compliance**

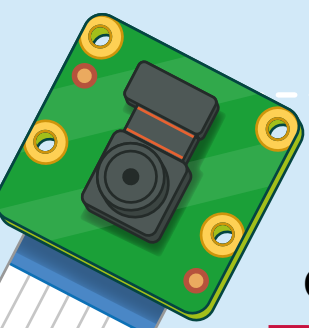
IFETEL: 2019LAB-ANCE4957
Número de certificado para Raspberry Pi 4 Modelo B.

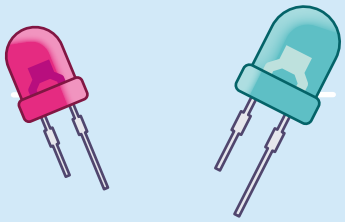
Número de registo TRA
ER73381/19
Número de certificado para Raspberry Pi 4 Modelo B.

CCAO19LP1120T2
Número de certificado para Raspberry Pi 4 Modelo B.

NTC
Tipo aprovado:
N.º: ESD-GEC-1920098C
Número de certificado para Raspberry Pi 4 Modelo B.

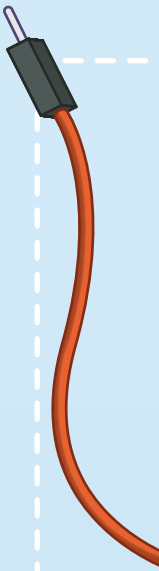
TA-2019/750 APROVADO
Número de certificado para Raspberry Pi 4 Modelo B.





O GUIA PARA INICIANTES DO

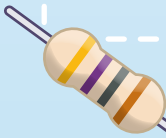
Raspberry Pi Oficial

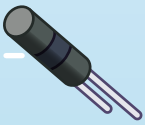


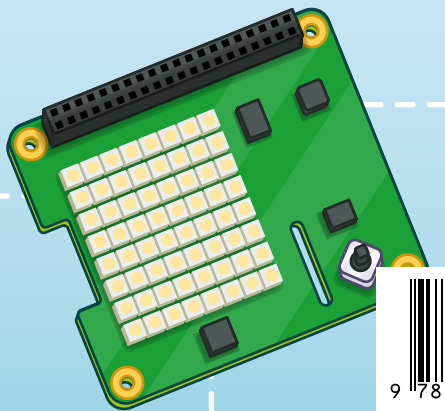
O Raspberry Pi é um pequeno computador inteligente produzido no Reino Unido e com um enorme potencial. Fabricado com a mesma tecnologia que encontra num smartphone, o Raspberry Pi foi projetado para ajudá-lo a aprender a programar, descobrir como funcionam os computadores e construir os seus próprios dispositivos incríveis. Este guia foi escrito para mostrar como é fácil começar.

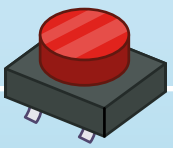
Aprenda como:

- Configurar o seu Raspberry Pi, instalar o sistema operativo e começar a utilizar este computador totalmente funcional.
- Iniciar projetos de programação, com guias passo a passo utilizando as linguagens de programação Scratch 3 e Python.
- Experimentar ligar componentes eletrónicos e divertir-se criando projetos incríveis.









raspberrypi.org

9 781912 047802

Raspberry Pi PRESS