



Instalação, configuração e administração.

Manual do utilizador

versão 1.0 Beta

Junho de 2020.

MiniQRNG é uma marca registada da OpenQbit Inc., sob uma licença de uso livre e comercial. Termos e condições de utilização em: www.OpenQbit.com

Conteúdo

1. Introdução.....	3
2. O que é a programação Blockly?	4
3. O que é o Termux?	4
4. O que é o Mini QRNG?	4
5. Configuração de armazenamento dentro da Termux.....	8
6. Instalação de servidor SSH (Secure Shell).....	9
7. Configuração do servidor SSH no telemóvel (smartphone).....	10
8. Ambientes Blockly (App Inventor, AppyBuilder y Thunkable).	16
9. Definição e utilização de blocos em Mini QRNG.....	17
10. Criação de um dispositivo "Hardware" de um QRNG.	28
11. Anexo "OpenQbit Quantum Computing".	34
12. Licenciamento e utilização de software.	39

1. Introdução.

A criptografia actual é baseada em sequências de números aleatórios. Os geradores de números pseudo-aleatórios actualmente em uso parecem fornecer sequências de bits aleatórios, mas na realidade estas sequências de bits têm certos padrões, pelo que existe o risco de serem pirateados ou manipulados para fazer a informação viajar através de redes públicas e pirateadas.

A integração de fontes de entropia física em geradores de números aleatórios é o método mais comum para superar esta ameaça à segurança. No entanto, a física clássica é causal, pelo que a imprevisibilidade de uma pequena sequência gerada com a física clássica não pode ser provada.

A física quântica, por outro lado, é essencialmente aleatória. Os números gerados por um gerador de números quânticos aleatórios (QRNG) não podem ser previstos: o QRNG é comprovadamente imprevisível. Assim, se um gerador de números quânticos aleatórios for utilizado num sistema de segurança, mesmo um supercomputador rápido com operação aritmética rápida não pode prever as sequências de bits aleatórios utilizadas por este sistema de segurança.

A física quântica utiliza métodos baseados num conceito fundamental chamado entropia.

TIPOS DE ENTROPIA

Existem dois tipos gerais de fontes de entropia que podem ser medidas para gerar números verdadeiramente aleatórios. O primeiro tipo inclui um processo físico que é difícil ou impossível de medir ou demasiado intensivo em termos computacionais para prever, ou ambos. Isto é uma fonte de caoticentropia. Um exemplo comum conhecido pela maioria das pessoas é uma máquina de lotaria. Um conjunto de bolas numeradas sequencialmente são colocadas numa câmara e misturadas constantemente, girando a câmara ou soprando ar através da câmara. Várias das bolas podem cair para fora da câmara e os números marcados nas bolas representam o sorteio da lotaria. O sorteio é aleatório devido ao grande número de interacções entre as bolas e a câmara resultando num rápido aumento do número de movimentos possíveis de cada bola. Não só a complexidade destas interacções é extremamente elevada, como não existe uma forma aparente de observar ou medir com precisão todas as variáveis internas das bolas, a câmara e o fluxo de ar. Um segundo tipo, muito diferente de fonte de entropia é a mecânica quântica. Muitas partículas microscópicas ou ondas, tais como fótons, electrões e prótons têm propriedades mecânicas quânticas que incluem rotação, polarização, posição, e momento. Dada a configuração adequada para produzir estas partículas, os valores específicos da sua rotação ou polarização, por exemplo, não só são desconhecidos e teoricamente imprevisíveis, como também são determinados fisicamente até que seja feita uma medição.

2. O que é a programação Blockly?

Blockly é uma **linguagem de programação visual** composta por um simples conjunto de comandos que podemos combinar como se fossem as peças de um puzzle. É uma ferramenta muito útil para aqueles que querem **aprender a programar de forma** intuitiva e simples ou para aqueles que já sabem programar e querem ver o potencial deste tipo de programação.

Blockly é uma forma de programação em que não é necessário qualquer tipo de linguagem informática, isto porque é apenas juntar blocos gráficos como se estivéssemos a jogar lego ou um puzzle, só é preciso ter alguma lógica e mais nada!

Qualquer pessoa pode criar programas para telemóveis (smartphones) sem se meter com as linguagens de programação difíceis de compreender, basta juntar blocos de forma gráfica de uma forma simples, fácil e rápida de criar.

3. O que é o Termux?

Termux é um emulador de terminal Android e uma aplicação de ambiente Linux que funciona directamente sem a necessidade de encaminhamento ou configuração. É instalado automaticamente um sistema básico mínimo.

Utilizaremos Termux pela sua estabilidade e facilidade de instalação e gestão, no entanto, poderá utilizar um ambiente instalado do Ubuntu Linux para Android.

Neste ambiente Linux, terá o "núcleo" dos processos de comunicação do MiniQRNG.

4. O que é o Mini QRNG?

Mini QRNG é Software e Hardware que inclui três soluções tecnológicas para criar QRNGs (Quantum Random Number Generators). Classificados da seguinte forma:

- a.- QRNG API. - Gerador de números quânticos aleatórios obtidos a partir de servidores externos.
- b.- MiniQRNG / Software. - Gerador de números quânticos aleatórios obtidos utilizando as propriedades físicas (quantum) da câmara do telemóvel.
- c.- MiniQRNG / Hardware. - Gerador de número quântico obtido usando hardware baseado nas propriedades físicas quânticas de um laser. Mais tarde dir-lhe-emos como construí-lo em casa a um baixo custo.

1. Instalação e configuração do Terminal Termux.

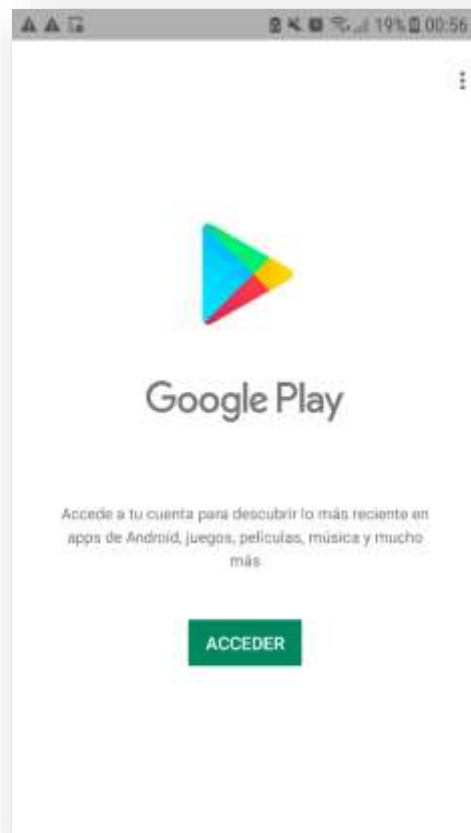
Primeiro precisamos de um ambiente Linux uma vez que cada sistema Android é baseado em Linux para segurança e flexibilidade em ferramentas, utilizaremos o terminal "Termux" que contém aquele ambiente onde instalaremos a(s) ferramenta(s) que nos ajudará(ão) a criar QRNGs.

Termux é um emulador Linux onde iremos instalar os pacotes necessários para criar números quânticos.

Uma das principais vantagens de utilizar Termux é que pode instalar programas sem ter de "rodar" o telemóvel (Smartphone), o que garante que não se perde a garantia do fabricante devido a esta instalação.

Instalação de Termux.

A partir do seu telemóvel, vá para a aplicação ícone do Google Play (play.google.com).



Pesquisar por aplicação "Termux", seleccioná-la e iniciar o processo de instalação.



Início da aplicação Termux.

Depois de começarmos, teremos de executar os dois comandos seguintes para efectuar actualizações do emulador do sistema operativo Linux:

\$ apt update

\$ apt upgrade

Confirmar todas as opções Y(Sim)...

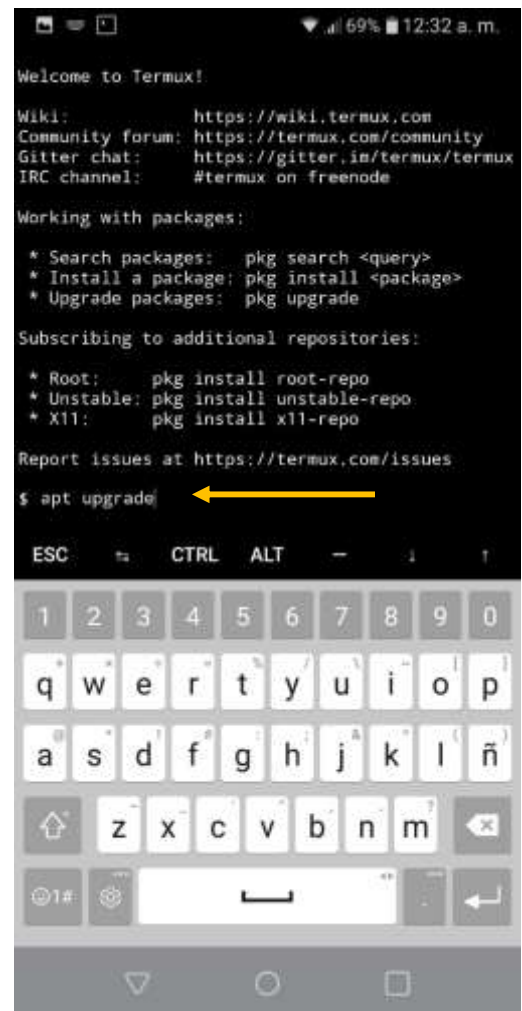
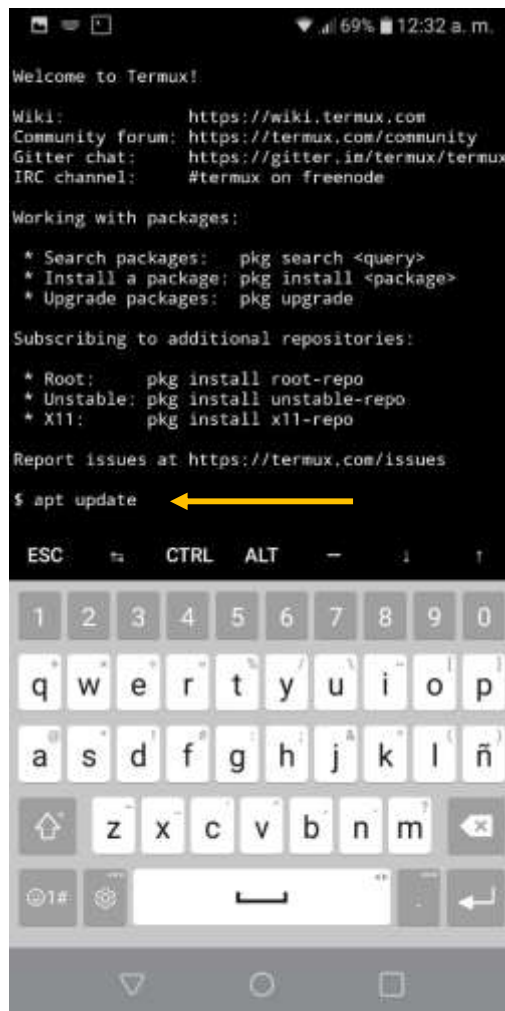
Termux

Home \$ actualização

do apt \$ actualização do



apt



5. Configuração de armazenamento dentro da Termux.

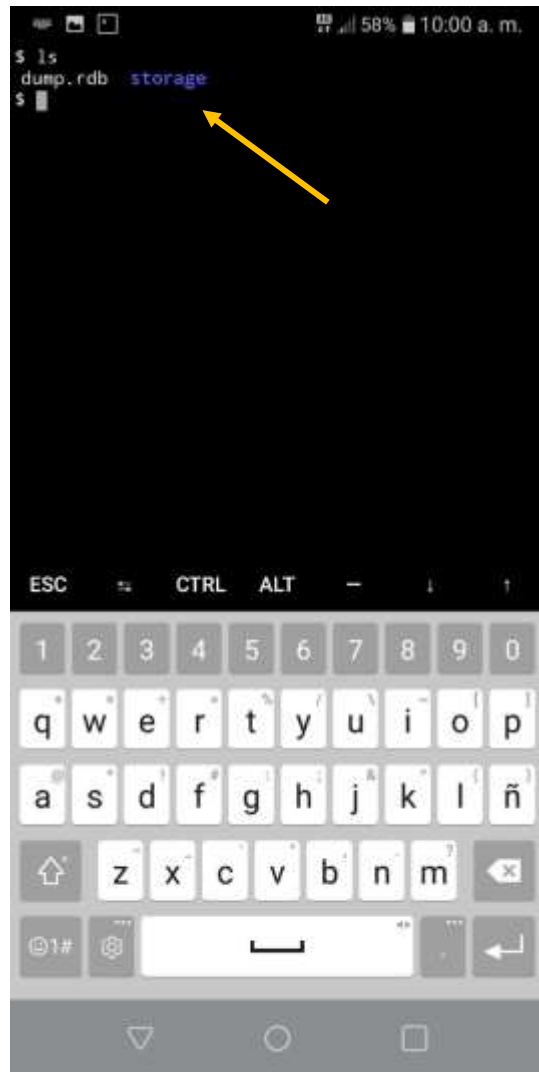
Após ter actualizado e actualizado o sistema Termux, começaremos a configurar a forma de visualizar o armazenamento interno do telefone no sistema Termux, o que o ajudará a poder trocar informações entre o Termux e as nossas informações no telefone.

Isto pode ser feito de forma simples e rápida, executando o seguinte comando num terminal Termux.

Termux-setup-storage \$

Quando executa o comando anterior, aparece uma janela a pedir-lhe para confirmar a criação de um **armazenamento** virtual (directório) no Termux. Verificamos ao dar o comando:

\$ ls



6. Instalação de servidor SSH (Secure Shell).

\$ apt install openssh

\$ apt install sshpass

\$ apt install openssh

\$

apt install sshpass

A terminal window on a smartphone screen showing the command '\$ apt install openssh'. The output displays the process of reading package lists, building a dependency tree, and identifying additional packages to be installed (krb5, libdns, libldb, libedit, termux-auth). It also shows the disk space requirements and the progress of downloading packages from a repository. The terminal interface includes a virtual keyboard at the bottom and status icons at the top.

```
$ apt install openssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  krb5 libdns libldb libedit termux-auth
The following NEW packages will be installed:
  krb5 libdns libldb libedit openssh termux-auth
0 upgraded, 6 newly installed, 0 to remove and 0
not upgraded.
Need to get 2255 kB of archives.
After this operation, 11.9 MB of additional disk
space will be used.
Do you want to continue? [Y/n] Y
Get:1 https://dl.bintray.com/termux/termux-packa
ges-24 stable/main arm libldb arm 18.1.32-4 [465
kB]
Get:2 https://dl.bintray.com/termux/termux-packa
ges-24 stable/main arm krb5 arm 1.18.1 [839 kB]
24% [2 krb5 131 kB/839 kB 16%]
```

A terminal window on a smartphone screen showing the command '\$ apt install sshpass'. The output shows the package lists being read, the dependency tree being built, and the identification of the new package 'sshpass'. It indicates that 1 new package will be installed and shows the disk space requirements. The progress bar at the bottom shows '0% [Working]'. A yellow arrow points from the left terminal to this one.

```
$ apt install sshpass
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  sshpass
0 upgraded, 1 newly installed, 0 to remove and 0
not upgraded.
Need to get 7158 B of archives.
After this operation, 57.3 kB of additional disk
space will be used.
0% [Working]
```

Terminámos com a instalação da rede de comunicação para servidor SSH localhost no Smartphone móvel.

7. Configuração do servidor SSH no telemóvel (smartphone).

Permitiremos que o servidor SSH no telemóvel seja capaz de se ligar do nosso PC ao telemóvel e de trabalhar de uma forma mais rápida e confortável, também nos servirá para verificar se o serviço do servidor SSH no telemóvel funciona correctamente uma vez que o utilizaremos na comunicação com Mini QRNG.

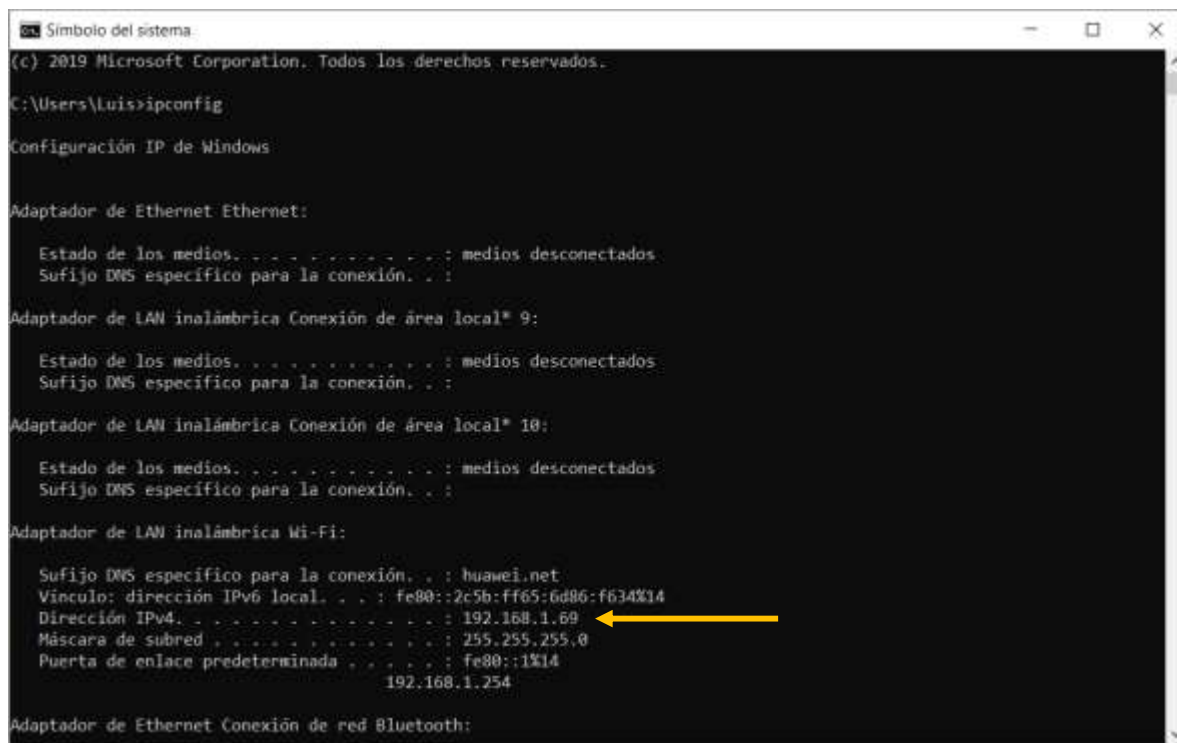
A primeira coisa que temos de fazer é ligar o telemóvel e o PC à **mesma rede WiFi para que** eles se possam ver uns aos outros. Os IPs ou endereços devem ser semelhantes a 192.168.XXX.XXX os valores XXX são números variáveis que são atribuídos aleatoriamente em cada computador.

Este exemplo foi testado num telemóvel LG Q6 e num PC com Windows 10 Home.

Verificar o IP ou endereço que o PC tem ligado ao WiFi temos de abrir um terminal no Windows.

No painel inferior onde a lupa de pesquisa é escrita cmd e premir a tecla Enter. Um terminal será aberto e nele escreveremos o comando:

C:Nome_de_utilizador> ipconfig



```
(c) 2019 Microsoft Corporation. Todos los derechos reservados.
C:\Users\Luis>ipconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

Adaptador de LAN inalámbrica Conexión de área local* 9:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

Adaptador de LAN inalámbrica Conexión de área local* 10:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

Adaptador de LAN inalámbrica Wi-Fi:

    Sufijo DNS específico para la conexión. . . : huawei.net
    Vinculo: dirección IPv6 local. . . : fe80::2c5b:ff65:6d86:f634%14
    Dirección IPv4. . . . . : 192.168.1.69
    Máscara de subred. . . . . : 255.255.255.0
    Puerta de enlace predeterminada. . . : fe80::1%14
                                     192.168.1.254

Adaptador de Ethernet Conexión de red Bluetooth:
```

Mostrar-nos-á que o IP atribuído ao PC neste é 192.168.1.69 mas é muito provável que seja diferente em cada caso.

NOTA: O endereço onde diz "endereço IPv4" deve ser tomado, não deve ser confundido com o Gateway.

Agora, no caso do telemóvel no terminal Termux devemos digitar o seguinte comando para saber o nome do nosso utilizador que vamos utilizar para nos ligarmos ao servidor SSH que tem o nosso telefone, executamos o seguinte comando:

\$ whoami

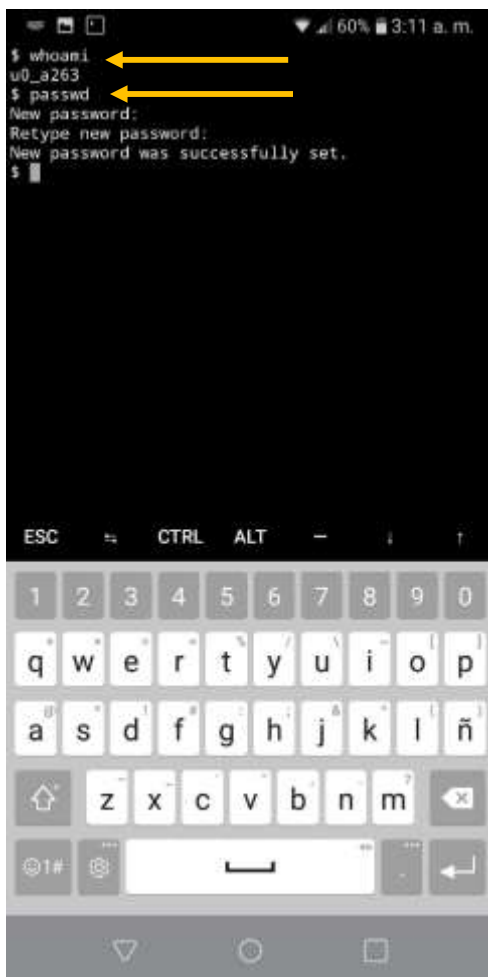
Mais tarde temos de dar uma palavra-chave a este utilizador, pelo que temos de executar o seguinte comando:

Senha de \$

Irá pedir-nos para escrever uma palavra-passe e premir Enter, mais uma vez pede-nos a palavra-passe que confirmámos e premir Enter, se tiver sido **"Nova palavra-passe foi definida com sucesso"** em caso de marcação de um erro é possível que a palavra-passe não tenha sido digitada correctamente. Realizar o procedimento de novo.

E depois, para saber que IP temos no Termux digitamos o seguinte comando, o IP está após a palavra "inet":

\$ ifconfig -a



The screenshot shows a Termux terminal on a mobile device. The status bar at the top indicates 60% battery and 3:11 a.m. The terminal output shows the command '\$ whoami' being executed, resulting in 'u0_a263'. Then, the command '\$ passwd' is entered, followed by prompts for 'New password:' and 'Retype new password:', which are both filled with asterisks. The final output is 'New password was successfully set.' and the prompt '\$' returns. A yellow arrow points to the 'whoami' command, and another points to the 'passwd' command.



The screenshot shows a Termux terminal on a mobile device. The status bar at the top indicates 61% battery and 2:57 a.m. The terminal output shows the command '\$ ifconfig -a' being executed. The output displays network interface details for 'wlan0', including flags, MTU, and IP addresses. A yellow arrow points to the IP address '192.168.1.68' under the 'inet' section.

Agora é altura de iniciar o serviço de servidor SSH no seu telefone para que possa receber sessões do seu PC. Executamos o seguinte comando no terminal Termux, este comando não dá qualquer resultado.

\$ sshd



Agora teremos de instalar um programa no PC que irá comunicar com o servidor SSH do telefone a partir do PC.

Temos de ir a <https://www.putty.org>

Seleccione onde está o link "Pode descarregar PuTTY aqui".



Bitvise SSH Client

Bitvise SSH Client is an SSH and SFTP client for Windows. It is developed and supported prof supports all features supported by PuTTY, as well as the following:

- graphical SFTP file transfer;
- single-click Remote Desktop tunneling;
- auto-reconnecting capability;
- dynamic port forwarding through an integrated proxy;
- an FTP-to-SFTP protocol bridge.

Bitvise SSH Client is **free to use**. You can [download it here](#).

Escolha a versão de 32 bits, não importa se o seu sistema é de 64 bits, funcionará bem.

Download PuTTY: latest release

[Home](#) | [FAQ](#) | [Feedback](#) | [Licence](#) | [Updates](#) | [Mirror](#)
Download: [Stable](#) | [Snapshot](#) | [Docs](#) | [Contact](#)

This page contains download links for the latest released version of PuTTY. Currently this is 0.73, released on 2019-09-29.

When new releases come out, this page will update to contain the latest, so this is a good page to bookmark or link to. Alternative

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date, so you may want to try out the [development snapshots](#), to see if the problem has already been fixed in those versions.

Package files

You probably want one of these. They include versions of all the PuTTY utilities.

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

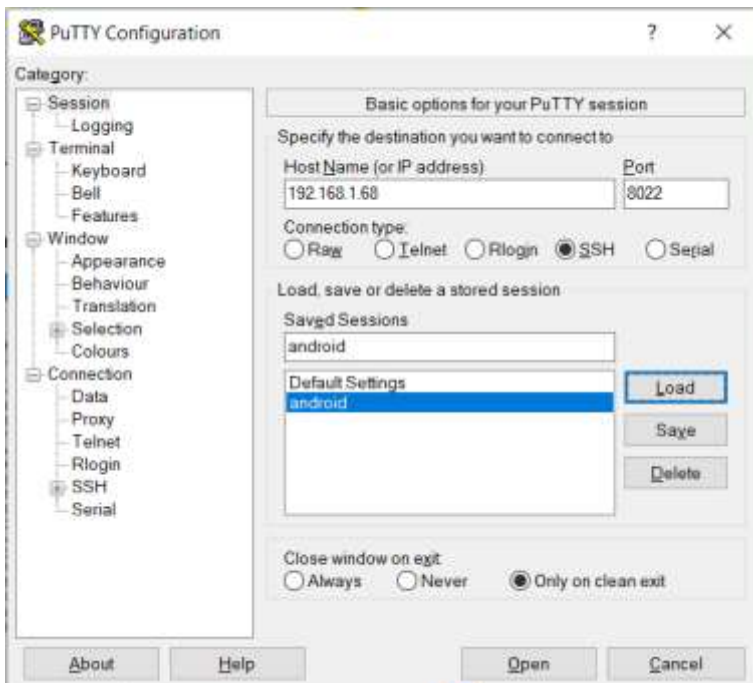
MSI ('Windows Installer')

32-bit:	putty-0.73-installer.msi	(or by FTP)	(signature)
64-bit:	putty-64bit-0.73-installer.msi	(or by FTP)	(signature)

Unix source archive

.tar.gz:	putty-0.73.tar.gz	(or by FTP)	(signature)
----------	-----------------------------------	-----------------------------	-----------------------------

Uma vez descarregado para o seu PC, execute-o e instale-o com as opções predefinidas. Em seguida, iniciar a aplicação PuTTY.



Nesta sessão iremos introduzir os dados do nosso servidor Openssh que instalámos no telemóvel.

Introduza o IP do telemóvel.

HostName ou endereço IP:

192.168.1.68 (exemplo IP)

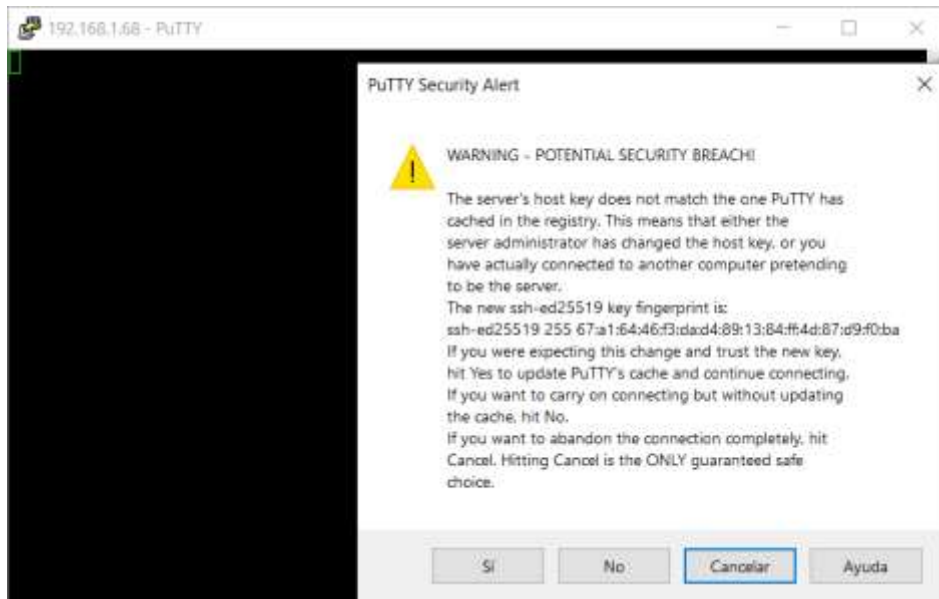
Porto:

8022 (Porta predefinida do servidor SSH móvel).

Podemos dar um nome à sessão em "Sessões Guardadas" e clicar no botão Guardar.

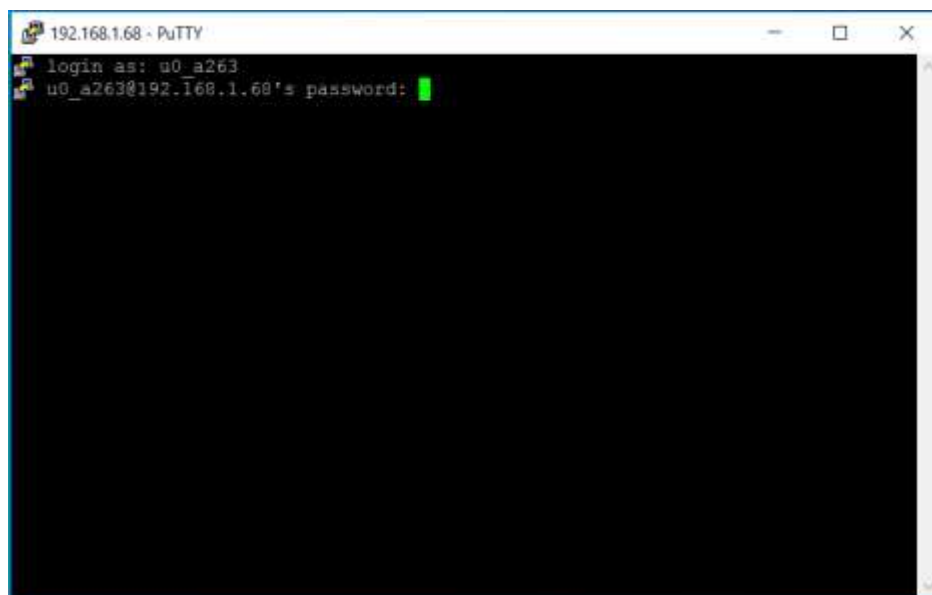
Mais tarde, na parte inferior, premimos para abrir uma ligação ao servidor dando o botão "Open".

No PC, quando se ligar pela primeira vez, **ser-lhe-á pedida a confirmação da** chave de encriptação da informação, clicando no botão "Sim".

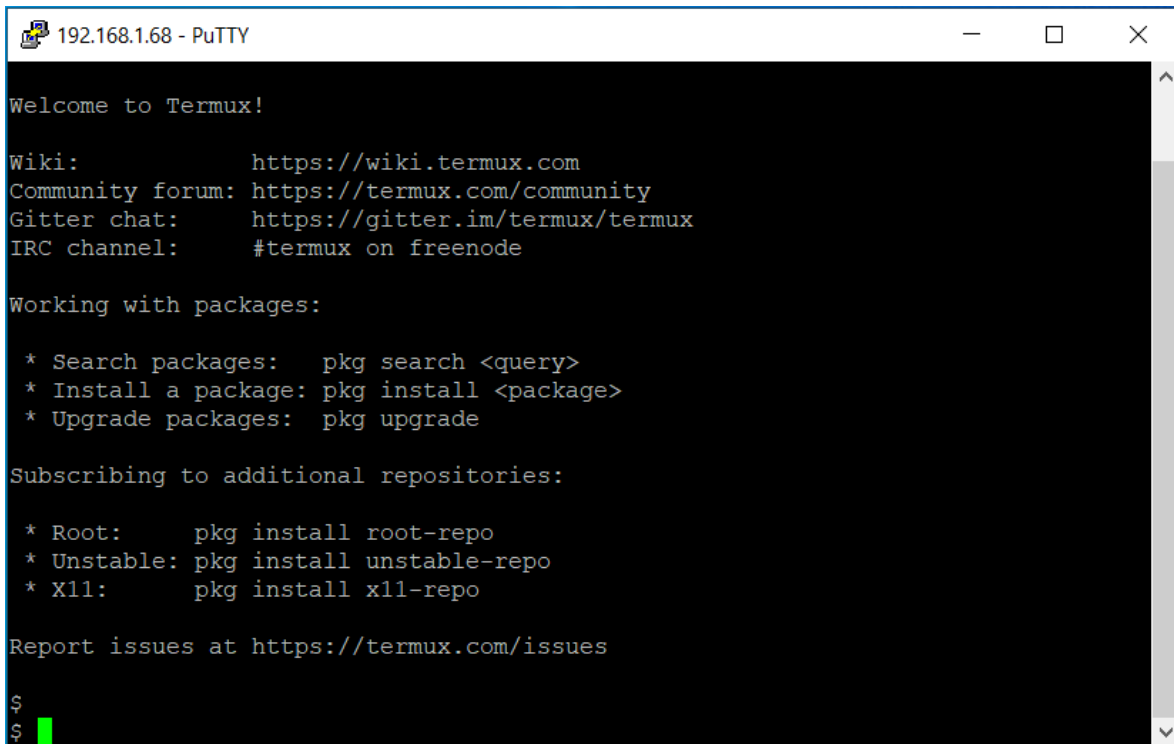


Mais tarde, ser-nos-á pedido o utilizador com o qual nos vamos ligar. Utilizaremos a informação que obtivemos anteriormente (utilizador e palavra-chave).

No **Login como:** temos de introduzir o nosso utilizador e dar Enter, depois pediremos novamente a palavra-passe e daremos o botão Enter.



Se os dados estiverem correctos, estaremos numa sessão SSH (Secure Shell) realizada a partir do PC (Cliente) ao telefone (Servidor SSH).



```
192.168.1.68 - PuTTY
Welcome to Termux!

Wiki:          https://wiki.termux.com
Community forum: https://termux.com/community
Gitter chat:   https://gitter.im/termux/termux
IRC channel:   #termux on freenode

Working with packages:

* Search packages:  pkg search <query>
* Install a package: pkg install <package>
* Upgrade packages: pkg upgrade

Subscribing to additional repositories:

* Root:    pkg install root-repo
* Unstable: pkg install unstable-repo
* X11:     pkg install x11-repo

Report issues at https://termux.com/issues

$
$
```

NOTA IMPORTANTE: Lembre-se que o IP (endereço) do PC e o IP (endereço) do telemóvel ligado ao mesmo WiFi irá provavelmente mudar cada vez que desligarmos e voltarmos a ligar, pelo que temos de verificar duas vezes quais os endereços de cada dispositivo, o que garantirá o sucesso da ligação entre os dispositivos através do servidor SSH do telefone e o PC (Cliente).

8. Ambientes Blockly (App Inventor, AppyBuilder y Thunkable).

App Inventor é um ambiente de desenvolvimento de software criado pelo Google Labs para construir aplicações para o sistema operativo Android. O utilizador pode, visualmente e a partir de um conjunto de ferramentas básicas, ligar uma série de blocos para criar a aplicação. O sistema é gratuito e pode ser facilmente descarregado a partir da web. As aplicações criadas com App Inventor são muito fáceis de criar porque não é necessário nenhum conhecimento de qualquer linguagem de programação.

Todos os ambientes actuais que utilizam a tecnologia Blockly, tais como AppyBuilder e Thunkable, entre outros, têm a sua versão gratuita, a sua forma de utilização pode ser através da Internet nos seus diferentes sites ou pode também ser instalada em casa.

Os blocos que compõem a arquitectura Mini BloclChain foram testados em App inventor e AppyBuilder, mas devido à sua optimização de código, devem trabalhar nas outras plataformas.

Versões online:

App Inventor.

<https://appinventor.mit.edu/>

AppyBuilder.

<http://appybuilder.com/>

Aconchegável.

<https://thunkable.com/>

Versão a ser instalada no seu computador (PC):

<https://sites.google.com/site/aprendeappinventor/instala-app-inventor>

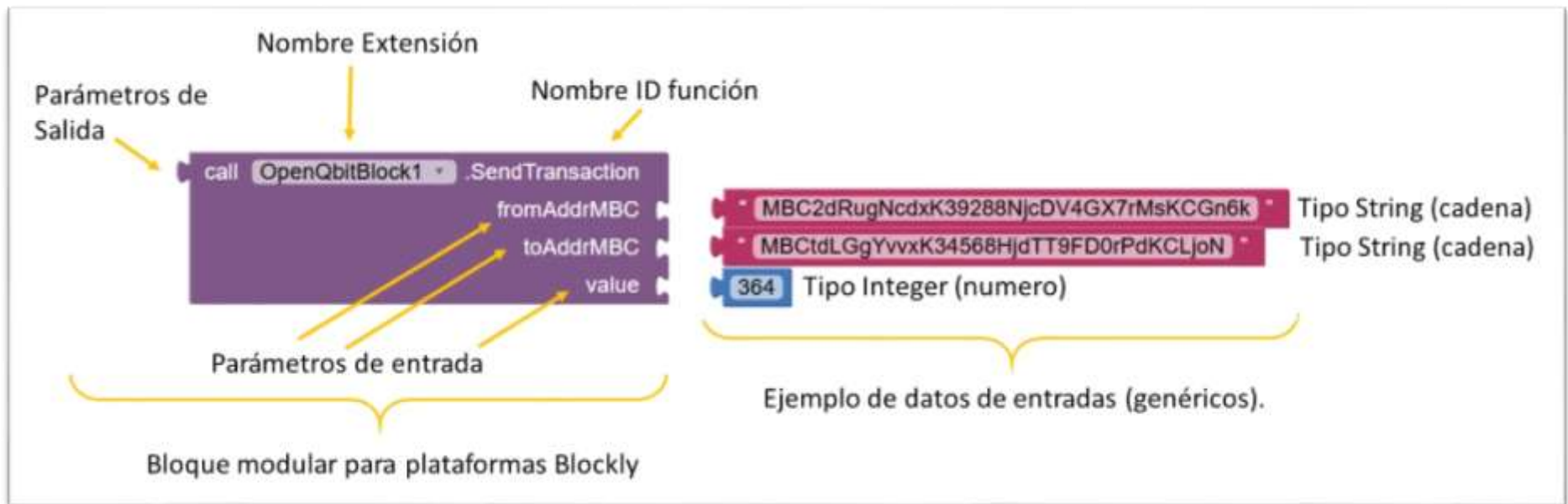
Ambiente para programadores de blocos Blockly.

<https://editor.appybuilder.com/login.php>

9. Definição e utilização de blocos em Mini QRNG.

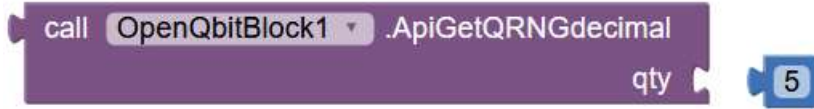
Começaremos por explicar a distribuição dos dados que todos os blocos terão, a sua sintaxe de utilização e configuração.

No exemplo seguinte podemos ver um bloco modular e os seus parâmetros de entrada e saída, assim como os tipos de dados de entrada, estes dados podem ser do tipo String (cadeia de caracteres) ou Inteiro (inteiro ou decimal). Mostramos como é utilizado e configuramos para o seu bom funcionamento.



Cada bloco de módulos terá a sua descrição e será nomeado caso tenha alguma dependência(s) obrigatória(s) ou opcional(s) de outros blocos utilizados como parâmetros de entrada, o processo de integração será anunciado. Começemos com os blocos da extensão OpenQbitQRNG comSSH.

Bloco para gerar números quânticos decimais aleatórios - (ApiGetQRNGdecimal)



Parâmetros de entrada: **qty** <Integer>

Parâmetros de saída: Dá a quantidade "qty" de números quânticos decimais aleatórios introduzidos nos números de entrada estão dentro do intervalo de 0 e 1 no formato JSON.

Exemplo:

qty = 5; saída: {"resultado": [0,5843012986202495, 0,7746497687824652, 0,05951126805960929, 0,1986079055812694, 0,03689783439899279]}

Descrição: Gerador de números aleatórios quânticos (QRNG) API

Bloco para gerar números quânticos decimais aleatórios - (ApiGetQRNGdecimal)



Parâmetros de entrada: **qty** <Integer>, min <Integer>, max <max>

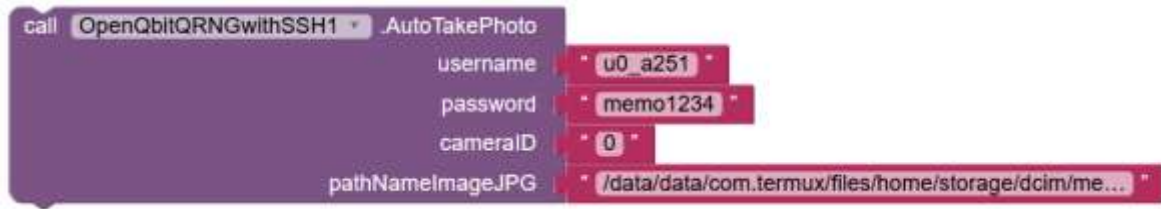
Parâmetros de saída: Dá a quantidade "qty" de inteiros quânticos aleatórios introduzidos na entrada os números estão dentro do intervalo de min e max no formato JSON.

Exemplo:

qty = 8, min = 1, max = 100; saída: {"resultado": [3, 53, 11, 2, 66, 44, 9, 78]}

Descrição: Gerador de números aleatórios quânticos (QRNG) API

Bloco para tirar uma fotografia automaticamente - (AutoTakePhone)



Parâmetros de entrada: **username** < String>, **password** < String>, **cameraID** < String> , **pathNameImageJPG** < String>.

Dependência obrigatória: Para utilizar este bloco deve preencher duas dependências de software; instalar no terminal Termux o módulo Termux-API. Este módulo contém o processo para tirar fotografias e carregar automaticamente o servidor SSH que foi previamente instalado.

Parâmetros de saída: entrega uma fotografia (imagem) em formato JPG no caminho especificado. No caminho deve

Descrição: Cria uma foto JPG automaticamente sem intervenção do utilizador.

Para instalar o Termux-API, o seguinte comando deve ser executado no Terminal Termux:

\$ pkg instalar termux-api

```
$ pkg install termux-api
Ign:2 https://dl.bintray.com/grimler/game-packag
es-24 games InRelease
Ign:3 https://dl.bintray.com/grimler/science-pac
kages-24 science InRelease
Ign:1 https://dl.bintray.com/termux/termux-packa
ges-24 stable InRelease
Get:5 https://dl.bintray.com/grimler/game-packag
es-24 games Release [5344 B]
Get:6 https://dl.bintray.com/grimler/science-pac
kages-24 science Release [6191 B]
Get:4 https://dl.bintray.com/termux/termux-packa
ges-24 stable Release [8255 B]
Get:7 https://dl.bintray.com/grimler/game-packag
es-24 games Release.gpg [475 B]
Get:8 https://dl.bintray.com/grimler/science-pac
kages-24 science Release.gpg [475 B]
Get:9 https://dl.bintray.com/termux/termux-packa
ges-24 stable Release.gpg [821 B]
0% [8 Release.gpg gpgv 6191 B]
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  termux-api
1 upgraded, 0 newly installed, 0 to remove and 7
4 not upgraded.
Need to get 21.2 kB of archives.
After this operation, 4096 B of additional disk
space will be used.
Get:1 https://dl.bintray.com/termux/termux-packa
ges-24 stable/main arm termux-api arm 0.50-1 [21
.2 kB]
Fetched 21.2 kB in 1s (18.7 kB/s)
(Reading database ... 25317 files and directorie
s currently installed.)
Preparing to unpack .../termux-api_0.50-1_arm.de
b ...
Unpacking termux-api (0.50-1) over (0.50) ...
Setting up termux-api (0.50-1) ...
$
```

Para descobrir o número, quantos e posição de IDs (identificadores de lentes fotográficas) tem o seu dispositivo móvel (Smartphone), execute o seguinte comando no terminal Termux.

\$ termux-camera-info

```
$ termux-camera-info
[
  {
    "id": "0",
    "facing": "back",
    "jpeg_output_sizes": [
      {
        "width": 4160,
        "height": 3120
      },
      {
        "width": 4160,
        "height": 2340
      },
      {
        "width": 4160,
        "height": 2080
      },
      {
        "width": 3264,
        "height": 2448
      },
      {

```

The screenshot shows the output of the 'termux-camera-info' command. It lists camera details for ID '0', which is the back camera. Yellow arrows point to the command, the ID '0', and the 'facing' field. The terminal interface includes a status bar at the top showing battery and time, and a virtual keyboard at the bottom.

```
    "manual_sensor",
    "manual_post_processing",
    5,
    6,
    4,
    7,
    "raw"
  ],
  {
    "id": "1",
    "facing": "front",
    "jpeg_output_sizes": [
      {
        "width": 2560,
        "height": 1920
      },
      {
        "width": 2560,
        "height": 1600
      },
      {
        "width": 2560,

```

This screenshot shows the continuation of the camera information, specifically for ID '1', which is the front camera. Yellow arrows highlight the ID '1' and the 'facing' field. The terminal interface is consistent with the previous screenshot, showing status bar and keyboard.

No nosso exemplo, o LG Q6 smarphone que utilizámos tem duas identificações "0" no verso e "1" na frente.

Agora vamos testar o API para tirar uma fotografia usando o ID "0" da lente traseira e dar-lhe um nome alfanumérico no nosso caso teste.jpg

Lembre-se que a API só fornece fotografias em formato JPG:

\$ termux-camera-photo -c 0 teste.jpg

O comando anterior deve ter criado um ficheiro com o nome teste.jpg automaticamente, se assim pudermos utilizar o bloco (AutoTakePhoto), não se esqueça de iniciar o nosso servidor SSH local com o comando: **\$ sshd**

NOTA: No caminho variávelNameImageJPG deve ser considerado que o caminho dentro do terminal Termux para aceder ao armazenamento do Smartphone deve ser utilizado:

/data/data/com.termux/files/home/storage/dcim/example.jpg

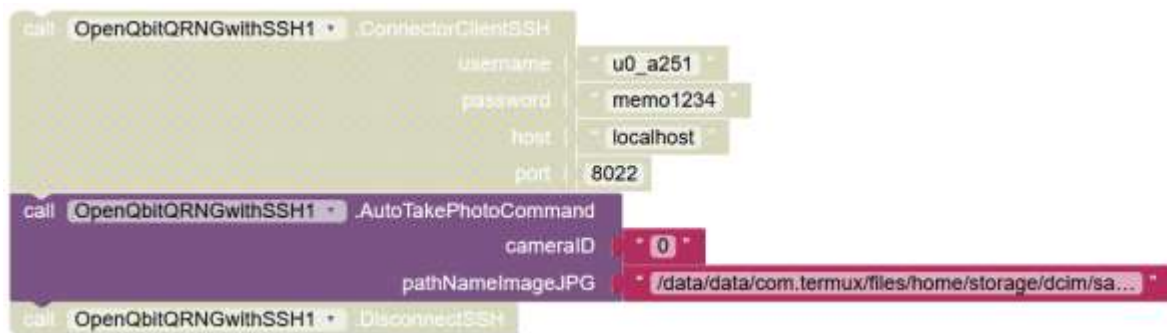
A rota anterior no andróide seria a mesma que a anterior:

/mnt/sdcard/dcim/example.jpg

No entanto, devemos lembrar que no terminal Termux a rota válida para a visualização da ligação do telemóvel deve ser sempre considerada como a rota padrão:

/dados/dados/com.termux/ficheiros/home/armazém

Bloco para tirar uma foto automaticamente SÓ COMANDO - (AutoTakePhoneCommand)



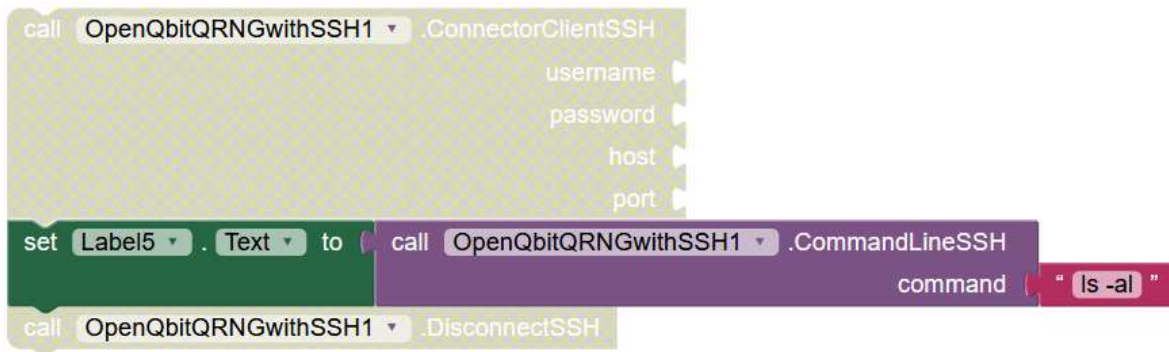
Parâmetros de entrada: **cameraID** <String>, **pathNameImageJPG** <String>.

Dependência Obrigatória: Bloco (ConnectorClientSSH), Bloco (DisconnectSSH).

Parâmetros de saída: entrega uma fotografia (imagem) em formato JPG no caminho especificado.

Descrição: Cria uma foto JPG automaticamente sem intervenção do utilizador. No entanto, a diferença em relação ao bloco (AutoTakePhoto), este bloco contém apenas o comando para criar a foto e é necessário que o bloco se ligue primeiro ao servidor SSH (ConnectClientSSH) e depois utilize o bloco (DisconnectSSH).

Bloco para executar comando no terminal Termux - (CommandLineSSH)



Parâmetros de entrada: **comando** <Cordar>

Dependência Obrigatória: Bloco (ConnectorClientSSH), Bloco (DisconnectSSH).

Parâmetros de saída: Executar o comando introduzido no terminal Termux.

Descrição: Um comando introduzido é executado e o bloco é necessário primeiro para se ligar ao servidor SSH (ConnectorClientSSH) e depois para utilizar o bloco (DisconnectSSH).

Bloco para ligação a um servidor SSH remoto ou local - (ConnectorClientSSH).

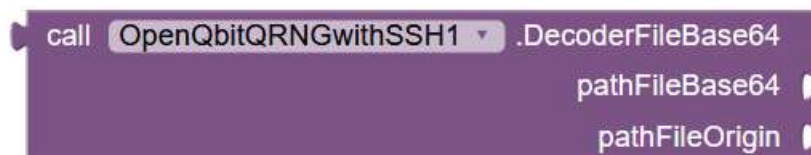


Parâmetros de entrada: **username** <string>, **password** <string>, **host** <string>, **port**<integer>

Parâmetros de saída: Se a ligação com o servidor ssh do terminal Termux for bem sucedida, dá-nos uma mensagem; "Connect SSH", se não for bem sucedida, dá-nos uma mensagem NULL.

Descrição: Bloco de comunicação para ligar o servidor SSH escolhido ao terminal Termux, através do protocolo de comunicação SSH (Secure Shell).

Bloco para decodificar um ficheiro com algoritmo Base64 (DecoderFileBase64).

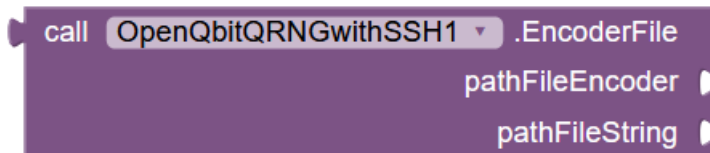


Parâmetros de entrada: **pathFileBase64** <String>, **pathFileOrigin** <String>

Parâmetros de saída: ficheiro fonte que foi introduzido no bloco (**EncoderFileBase64**)

Descrição: Um ficheiro Base64 é convertido para o ficheiro original que foi inserido no bloco (**EncoderFileBase64**).

Block converte um ficheiro para o formato Base64 - (**EncoderFile**)

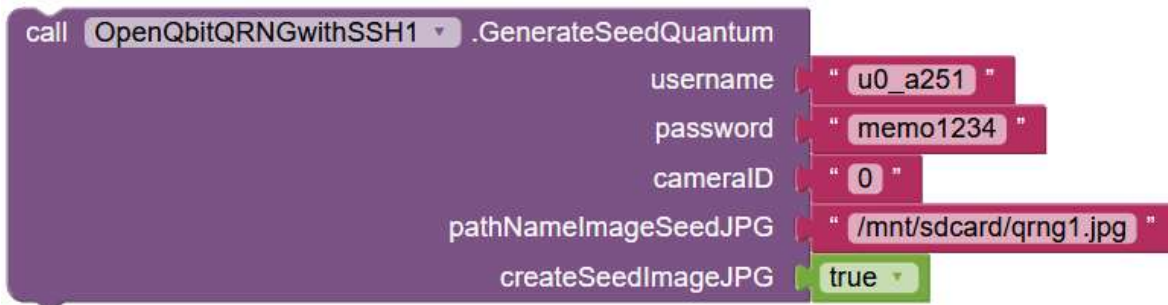


Parâmetros de entrada: **pathFileOrigin** <String> , **pathFileBase64** <String>

Parâmetros de saída: ficheiro codificado com base64.

Descrição: Converte um ficheiro fonte de qualquer formato para um ficheiro Base64. Os nomes dos ficheiros podem ser arbitrários e escolhidos pelo utilizador.

Bloco para gerar QRNG (Quantum Random Number Generator) - (**GenerateSeedQuantum**)



Parâmetros de entrada: **username** <String>, **password** <String>, **cameralID** <String>, **pathNameImageJPG** <String>. **createSeedImageJPG** <Blean>

Se o valor booleano for "Verdadeiro" quando o bloco é executado cada vez que uma nova imagem JPG de semente será criada com o nome do caminho introduzido. Se o valor Boolean for "Falso" desactivamos a opção de tirar uma imagem JPG (foto) e podemos indicar manualmente onde se encontra uma imagem na nossa selecção, que pode ser de qualquer formato.

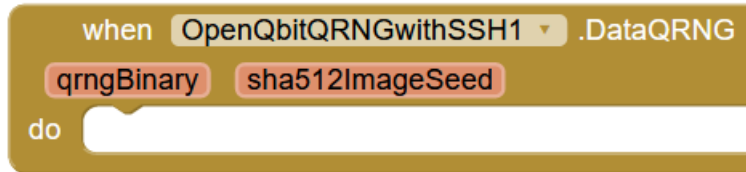
NOTA: Os melhores resultados para gerar um QRNG são baseados numa imagem formatada "RAW". Exemplo de formato DNG.

Dependência obrigatória: O Termux-API acima mencionado deve ser instalado no bloco (**AutoTakePhoto**).

Parâmetros de saída: O evento é executado (**DataQRNG**) e dá-nos dois valores:

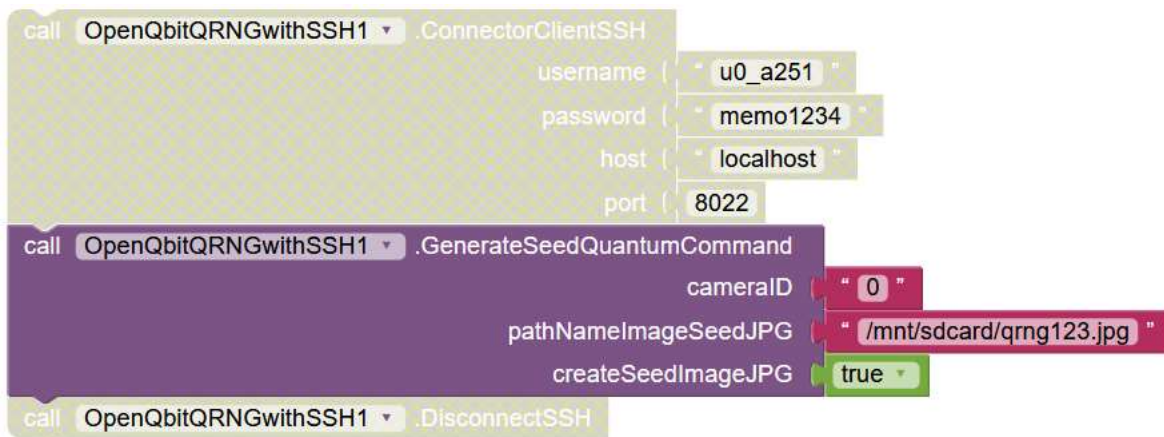
qrngBinary.- Uma série de números binários aleatórios

Sha512ImageSeed - Sha512 da imagem da semente JPG de onde vieram os números aleatórios.



Descrição: Gera números quânticos aleatórios (QRNG) através do sensor óptico da câmara do telemóvel. O algoritmo é baseado na captura de fotos aleatórias e o algoritmo é-lhe aplicado para fornecer uma sequência de números binários.

Bloco para gerar QRNG (Quantum Random Number Generator) - (GenerateSeedQuantumCommand).



Parâmetros de entrada: **cameraID** <String>, **pathNameImageJPG** <String>. **createSeedImageJPG** <Boolean>

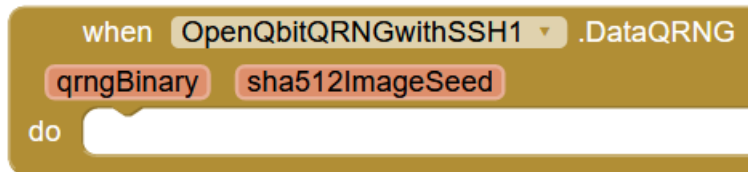
Se o valor booleano for "Verdadeiro" quando o bloco é executado cada vez que uma nova imagem JPG de semente será criada com o nome do caminho introduzido. Se o valor Boolean for "Falso" desactivamos a opção de tirar uma imagem JPG (foto) e podemos indicar manualmente onde se encontra uma imagem na nossa selecção, que pode ser de qualquer formato.

Dependência Obrigatória: Bloco (**ConnectorClientSSH**), Bloco (**DisconnectSSH**).

Parâmetros de saída: O evento é executado (**DataQRNG**) e dá-nos dois valores:

qrngBinary.- Uma série de números binários aleatórios

Sha512ImageSeed - Sha512 da imagem da semente JPG de onde vieram os números aleatórios.



Descrição: Gera números quânticos aleatórios (QRNG) através do sensor óptico da câmara do telemóvel. O algoritmo é baseado na captura de fotos aleatórias e o algoritmo é-lhe aplicado para fornecer uma sequência de números binários.

No entanto, a diferença em relação ao bloco (**GenerateSeedQuantum**), este bloco contém apenas o comando para executar o algoritmo de reacção QRNG e precisa do bloco primeiro para se ligar ao servidor SSH (**ConnectClientSSH**) e depois utilizar o bloco (**DisconnectSSH**).

Bloco para obter entropia de Shannon numa Imagem (foto) - (**GetShannonEntropyFile**).



Parâmetros de entrada: username <String> , password <String>, pathFileImage <String>

Dependência obrigatória: O módulo Shannon_entropy precisa de ser instalado no terminal Termux.

Parâmetros de saída: Fornece a entropia de uma imagem

Exemplo:

Produção: 8.94596789873

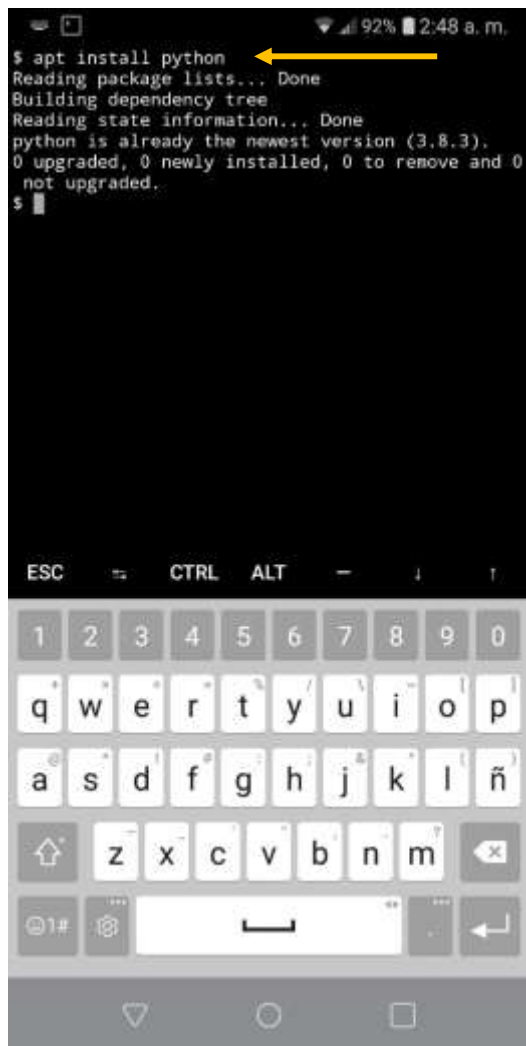
Descrição: Dá-nos a entropia de uma imagem. A entropia é o parâmetro fundamental para a geração de números aleatórios de boa qualidade, quanto maior for a entropia, melhores serão os resultados.

Para instalar o módulo de entropia Shannon, precisamos primeiro de instalar o pacote Python e depois instalar o módulo Pillow e Shannon_entropy com os seguintes comandos no terminal Termux.

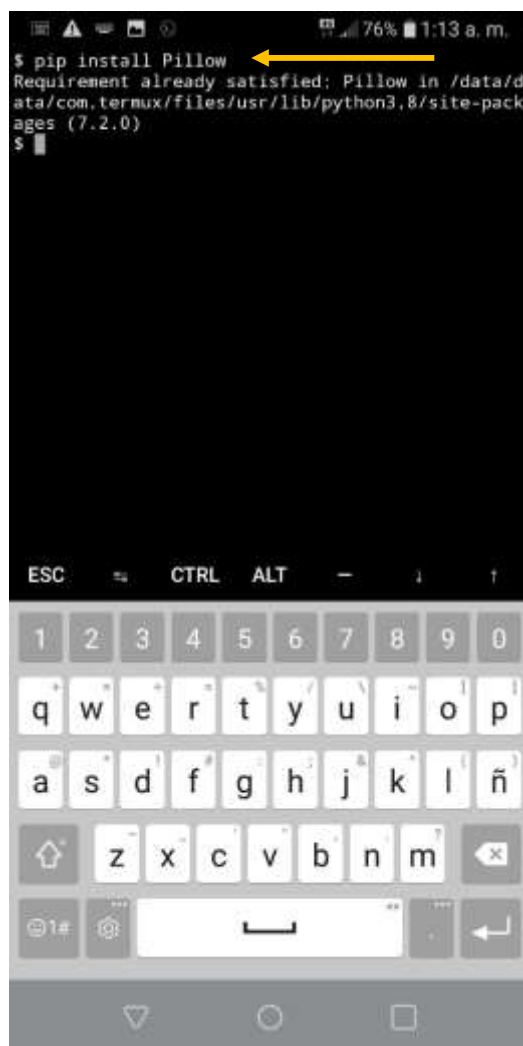
\$ apt instalar Python

\$ pip instalar almofada

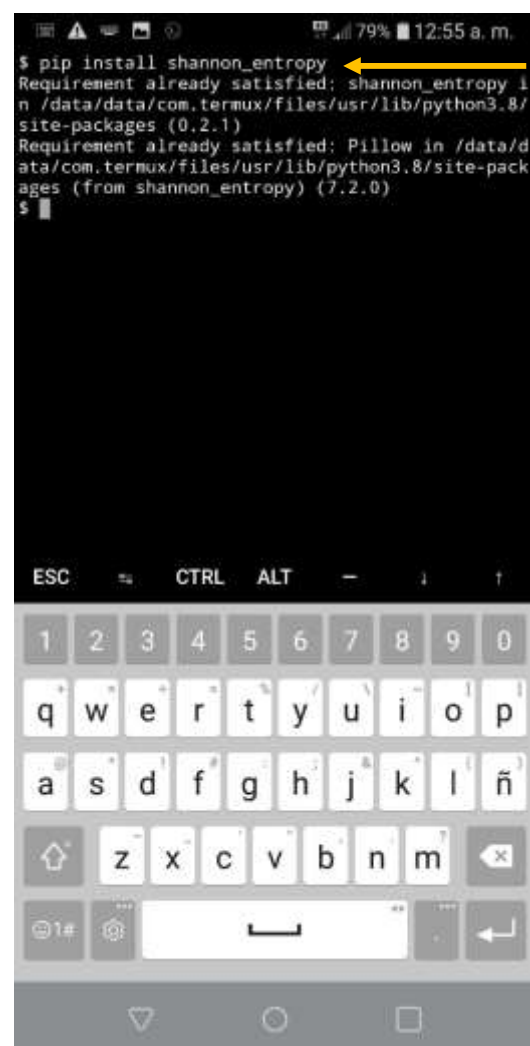
\$ pip instalar Shannon_entropy



```
$ apt install python
Reading package lists... Done
Building dependency tree
Reading state information... Done
python is already the newest version (3.8.3).
0 upgraded, 0 newly installed, 0 to remove and 0
not upgraded.
$
```



```
$ pip install Pillow
Requirement already satisfied: Pillow in /data/d
ata/com.termux/files/usr/lib/python3.8/site-pack
ages (7.2.0)
$
```



```
$ pip install shannon_entropy
Requirement already satisfied: shannon_entropy i
n /data/data/com.termux/files/usr/lib/python3.8/
site-packages (0.2.1)
Requirement already satisfied: Pillow in /data/d
ata/com.termux/files/usr/lib/python3.8/site-pack
ages (from shannon_entropy) (7.2.0)
$
```

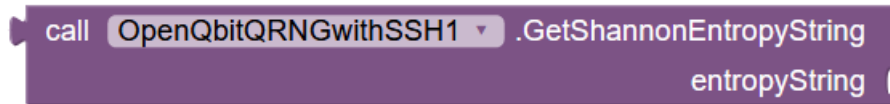
Depois temos de criar no directório "Home" do Termux o seguinte ficheiro Python chamado "entropy.py" com o seguinte código no interior.

```
Da imagem de importação da PIL  
  
Matemática de importação  
  
Da importação de Shannon_entropy *  
  
Sistema de importação  
  
Img=Image.open(sys.argv[1])  
  
print(Shannon_entropy(img))
```

Guardamos o ficheiro e temos o nosso ambiente para utilizar com o bloco (GetShannonEntropyFile).

Dica: De facto, com esta instalação Python pode criar os seus próprios programas nesta linguagem e executá-los através do bloco (ConnectorClientSSH).

Bloco para obter entropia Shannon a partir de um cordel - (GetShannonEntropyString).



Parâmetros de entrada: entropiaString>String>

Parâmetros de saída: Fornece a entropia de uma cadeia de caracteres.

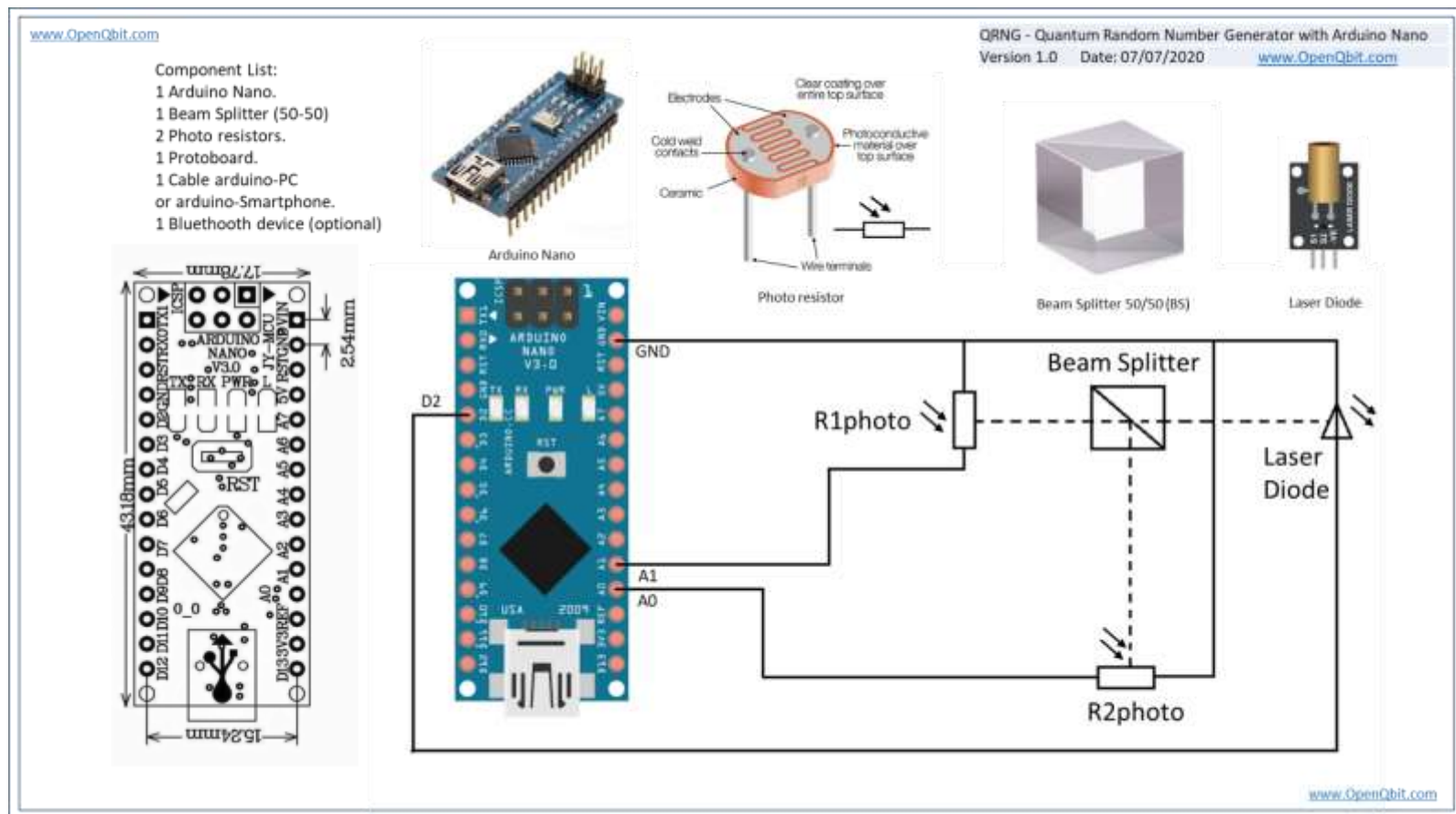
Exemplo:

Produção: 5.76002345671

Descrição: Dá-nos a entropia de uma cadeia de caracteres. A entropia é o parâmetro fundamental para a geração de números aleatórios de boa qualidade, quanto maior for a entropia, melhores serão os resultados.

10. Criação de um dispositivo "Hardware" de um QRNG.

Vamos agora criar um dispositivo físico "Hardware" para gerar Números Quânticos Aleatórios (QRNG) com componentes baratos que podem ser facilmente montados em casa.



www.OpenQbit.com

QRNGv1.0.ino

Software
Program to arduino nano.

```

/* OpenQbitQRNG Firmware V1.0
*Author: Guillermo Vidal
*Copyright © 2020 OpenQbit, Inc.
*License: MIT
*/

```

```

int triggerQ = 2; // This pin will pulse our quantum circuit
int QuA0Pin = A0; // This pin measures the horizontal polarized photons
int QuA1Pin = A1; // This pin measures the vertically polarized photons
float Qu0 = 0;
float Qu1 = 0;

void setup() {
  // Just setting up triggerPin and serial connection
  pinMode(triggerQ, OUTPUT); // sets the digital pin 2 as output
  Serial.begin(9600);
}

int Random() {
  // Pulse the laser
  digitalWrite(triggerQ, HIGH);
  delay(300);
  digitalWrite(triggerQ, LOW);
  delay(300);
  // Read the photoresistors
  Qu0 = analogRead(QuA0Pin);
  Qu1 = analogRead(QuA1Pin);
  // Determine random bit
  if(Qu0>Qu1) { // More photons in the Qu0 mode, return 0
    return 0;
  } if(Qu0 < Qu1) { // More photons in the Qu1 mode, return 1
    return 1;
  } else {
    /* The same number of photons are in both modes!
    This is actually not an uncommon occurrence, for our
    purposes we will simply run the function recursively until
    a random bit can be generated.
    */
    Random();
  }
}

void loop() {
  Serial.print(Random());
}

```


Output console

0010110101011110101011010.....

QRNG - Quantum Random Number Generator with Arduino Nano
Version 1.0 Date: 07/07/2020 www.OpenQbit.com

www.OpenQbit.com

Compilando o programa QRNGv10.ino e carregando para o nano árduo...



```
QRNGv10 Arduino 1.8.10
Archivo Editar Programa Herramientas Ayuda

QRNGv10

int triggerQ = 2; // This pin will pulse our quantum circuit.
int Qu0Pin = A0; // This pin measures the horizontal polarised photons
int Qu1Pin = A1; // This pin measures the vertically polarised photons
float Qu0 = 0;
float Qu1 = 0;

void setup() {
  // Just setting up triggerPin and serial connection
  pinMode(triggerQ, OUTPUT); // sets the digital pin 2 as output
  Serial.begin(5000);
}

int Random() {
  // Pulse the laser
  digitalWrite(triggerQ, HIGH);
  delay(300);
  digitalWrite(triggerQ, LOW);
  delay(300);
  // Read the photodiodes
  Qu0 = analogRead(Qu0Pin);
  Qu1 = analogRead(Qu1Pin);
  // Determine random bit
  if(Qu0>Qu1) { // More photons in the Qu0 mode, return 0
    return 0;
  } if(Qu0 < Qu1) { // More photons in the Qu1 mode, return 1
    return 1;
  } else {
    // The same number of photons are in both modes!
    // This is actually not an uncommon occurrence, but our
    // program will simply run the function repeatedly until
    // a random bit can be generated.
    return Random();
  }
}

void loop() {
  Serial.println(Random());
}
```

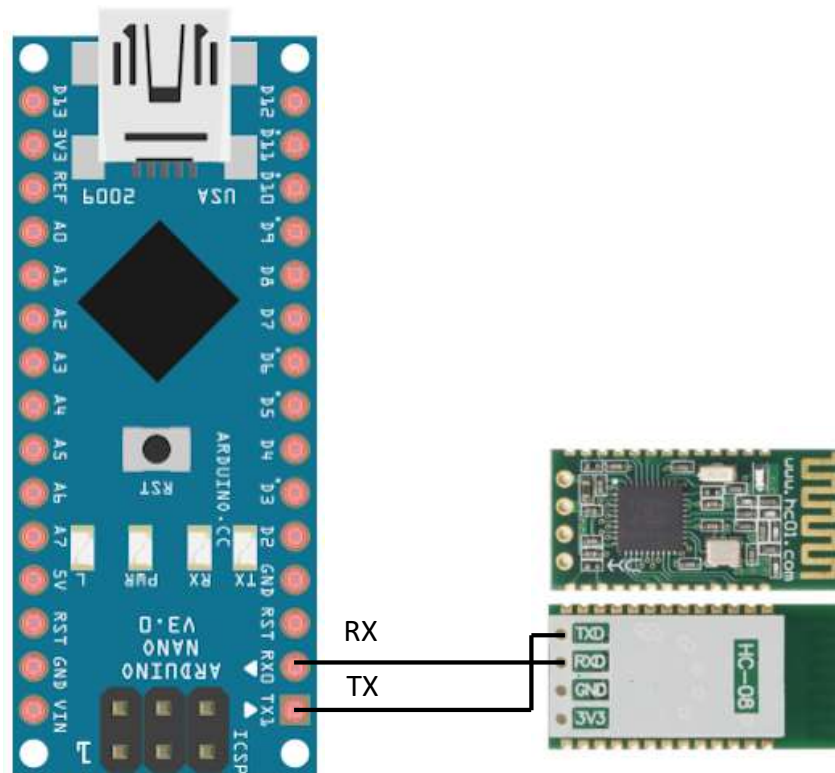
Compilando...

El Sketch usa 2352 bytes (7%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.
Las variables globales usan 198 bytes (7%) de la memoria dinámica, dejando 1840 bytes para las variables locales. El máximo es 2048 bytes.

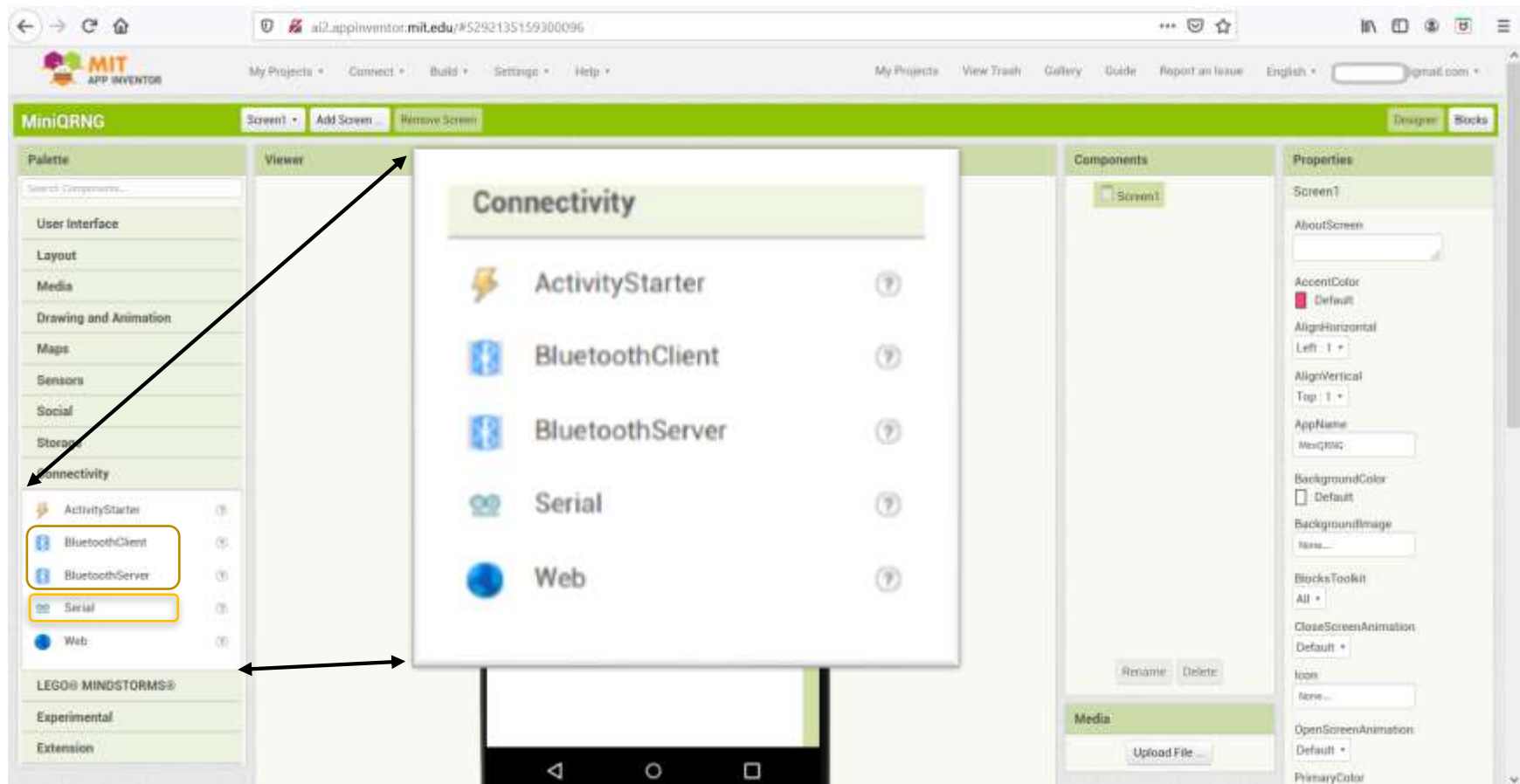
40 ArduinoGetters Ltd en CDM

Há duas formas de comunicar com o nano árduo, uma é através da porta Serial e a outra é através de uma ligação Bluetooth.

Para a ligação bluetooth é muito simples, só temos de comprar o módulo HC-08 ou um semelhante e ligá-lo como se segue:



Os seguintes componentes Serial ou Bluetooth podem ser utilizados para ligar o App Inventor ao Arduino:



Agora compilado e carregado o programa QRNGv10.ino só falta comunicar com o nano árduo para guardar os dados (números quânticos aleatórios) estes serão em formato binário, no entanto, os dados obtidos podem ser facilmente passados para outro formato, tal como hexadecimal ou decimal, dependendo do requisito final.

Finalmente, para ver um exemplo de como funciona a ligação em série ou Bluetooth, aqui estão algumas ligações de referência.

Lembre-se que tudo é através de programação Blockly para ser testado com App Inventor este já tem blocos para comunicação com arduino serial ou outro sistema blockly pode ser para tarves bluetooth similares online.

http://kio4.com/appinventor/9A0_bluetooth_RXTX.htm

<http://kio4.com/appinventor/index.htm#bluetooth>

<https://community.appinventor.mit.edu/>

11. Anexo "OpenQbit Quantum Computing".

Como funciona a computação quântica? ⁽²⁾

A transformação digital está a provocar mudanças no mundo mais rapidamente do que nunca. Acreditaria que a era digital está prestes a terminar? A **literacia digital** já foi identificada como uma área onde o conhecimento aberto e as oportunidades acessíveis de aprender sobre a tecnologia são urgentes para colmatar lacunas no desenvolvimento social e económico. Aprender com os conceitos-chave da era digital tornar-se-á ainda mais crítico com a chegada iminente de outra nova onda tecnológica capaz de transformar os modelos existentes com velocidade e potência espantosas: **as tecnologias quânticas**.

Neste artigo, comparamos os conceitos básicos da computação tradicional e da computação quântica; e também começamos a explorar a sua aplicação em outras áreas relacionadas.

O que são tecnologias quânticas?

Ao longo da história, os seres humanos desenvolveram a tecnologia à medida que compreendiam como a natureza funciona através da ciência. Entre 1900 e 1930, o estudo de alguns fenómenos físicos que ainda não eram bem compreendidos deu origem a uma nova teoria física, a **Mecânica Quântica**. Esta teoria descreve e explica o funcionamento do mundo microscópico, o habitat natural das moléculas, átomos ou electrões. Graças a esta teoria, não só foi possível explicar estes fenómenos, como também foi possível compreender que a realidade subatômica funciona de uma forma completamente contra-intuitiva, quase mágica, e que no mundo microscópico ocorrem eventos que não ocorrem no mundo macroscópico.

Estas **propriedades quânticas** incluem sobreposição quântica, enredamento quântico e teleportação quântica.

- A **sobreposição quântica** descreve como uma partícula pode estar em diferentes estados ao mesmo tempo.
- O **enredamento quântico** descreve como duas partículas tão distantes quanto desejado podem ser correlacionadas de tal forma que, ao interagirem com uma, a outra está ciente disso.
- O **teletransporte quântico** utiliza emaranhamento quântico para enviar informação de um lugar para outro no espaço sem ter de viajar através dele.

As tecnologias quânticas baseiam-se nestas propriedades quânticas da natureza subatômica.

Neste caso, hoje em dia, a compreensão do mundo microscópico através da Mecânica Quântica permite-nos inventar e conceber tecnologias capazes de melhorar a vida das pessoas. Existem muitas e muito diferentes tecnologias que utilizam fenómenos quânticos e algumas delas, tais como lasers ou ressonância magnética (MRI), estão connosco há mais de

meio século. Contudo, estamos actualmente a assistir a uma revolução tecnológica em áreas como a computação quântica, informação quântica, simulação quântica, óptica quântica, metrologia quântica, relógios quânticos ou sensores quânticos.

O que é computação quântica? Primeiro, é preciso compreender a computação clássica.




FIGURA 1.
Ejemplos de caracteres en lenguaje binario.

Caracter	Bits
7	111
A	01000001
\$	00100100
:)	0011101000101001

Para compreender como funcionam os computadores quânticos, é conveniente explicar primeiro como funcionam os computadores que utilizamos todos os dias, que iremos referir neste documento como computadores digitais ou clássicos. Estes, tal como os restantes dispositivos electrónicos, tais como comprimidos ou telemóveis, utilizam bits como unidades fundamentais de memória. Isto significa que os programas e aplicações são codificados em bits, ou seja, em linguagem binária de zeros e uns. Sempre que interagimos com qualquer um destes dispositivos, por exemplo, premindo uma tecla no teclado, são criadas, destruídas e/ou modificadas cordas de zeros e uns dentro do computador.

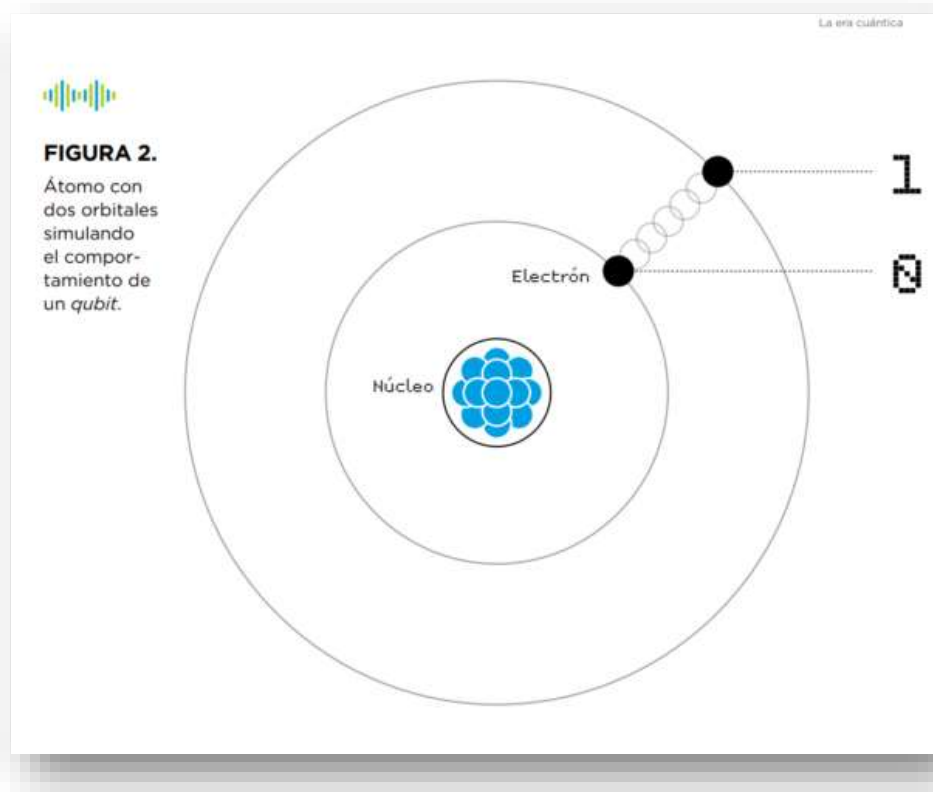
A questão interessante é, o que são estes zeros e uns fisicamente dentro do computador? O zero e um estados correspondem à corrente eléctrica que circula, ou não, através de peças microscópicas chamadas transístores, que actuam como interruptores. Quando não está a fluir corrente, o transistor está "desligado" e corresponde ao bit 0, e quando está a fluir está "ligado" e corresponde ao bit 1.

Mais simplesmente, é como se os bits 0 e 1 correspondessem a buracos, de modo que um buraco vazio é um bit 0 e um buraco ocupado por um electrão é um bit 1. Como exemplo, a figura 1 mostra a escrita binária de alguns caracteres. Agora que temos uma ideia de como funcionam os computadores de hoje, vamos tentar compreender como funcionam os quantum.

De bits a qubits

A unidade fundamental de informação na computação quântica é o bit quântico ou qubit. Os Qubits são, por definição, sistemas quânticos de dois níveis - veremos aqui exemplos - que,

tal como os bits, podem estar no nível baixo, que corresponde a um estado de baixa excitação ou energia definida como 0, ou no nível alto, que corresponde a um estado de maior excitação ou definido como 1. No entanto, e aqui reside a diferença fundamental com a computação clássica, as desistências podem também estar em qualquer dos infinitos estados intermédios entre 0 e 1, tais como um estado que é meio 0 e meio 1, ou três quartos de 0 e um quarto de 1.



Algoritmos quânticos, exponencialmente mais potentes e eficientes de computação

O objectivo dos computadores quânticos é tirar partido destas propriedades quânticas dos *qubits*, como sistemas quânticos que são, a fim de executar algoritmos quânticos que utilizam sobreposição e intercalação para fornecer um poder de processamento muito maior do que os clássicos. É importante salientar que a verdadeira mudança de paradigma não consiste em fazer o mesmo que os computadores digitais ou clássicos - os actuais - mas sim mais rápido, como se pode ler em muitos artigos, mas que os algoritmos quânticos permitem realizar certas operações de uma forma totalmente diferente que em muitos casos se revela mais eficiente - ou seja, em muito menos tempo ou utilizando muito menos recursos computacionais -.

Vejamos um exemplo concreto do que isto envolve. Imaginemos que estamos em Bogotá e queremos saber qual é a melhor rota para chegar a Lima de entre um milhão de opções para lá chegar ($N=1,000,000$). A fim de utilizar computadores para encontrar o melhor caminho,

precisamos de digitalizar 1.000.000 opções, o que implica traduzi-las em linguagem bit para o computador clássico e em *qubits* para o computador quântico. Enquanto um computador clássico teria de ir um a um analisando todos os caminhos até encontrar o desejado, um computador quântico tira partido do processo conhecido como paralelismo quântico que lhe permite considerar todos os caminhos ao mesmo tempo. Isto implica que, enquanto o computador clássico necessita da ordem de passos $N/2$ ou iterações, ou seja, 500.000 tentativas, o computador quântico encontrará o caminho óptimo depois de apenas \sqrt{N} operações no registo, ou seja, 1.000 tentativas.

No caso anterior a vantagem é quadrática, mas noutros casos é mesmo exponencial, o que significa que com n *qubits* podemos obter uma capacidade computacional equivalente a 2^n bits. Para exemplificar isto, é comum contar que com cerca de 270 qubits poderíamos ter mais estados base num computador quântico - mais cadeias de caracteres diferentes e simultâneas - do que o número de átomos no universo, que é estimado em cerca de 10^{80} . Outro exemplo é que se estima que com um computador quântico entre 2000 e 2500 *qubits* poderíamos quebrar praticamente toda a criptografia utilizada hoje em dia (a chamada criptografia de chave pública).

Porque é importante saber sobre a tecnologia quântica?

Estamos num momento de transformação digital em que diferentes tecnologias emergentes, tais como cadeias de bloqueio, inteligência artificial, drones, Internet das coisas, realidade virtual, impressoras 3D, robôs ou veículos autónomos, têm cada vez mais presença em múltiplos campos e sectores. Estas tecnologias, chamadas a melhorar a qualidade de vida do ser humano acelerando o desenvolvimento e gerando impacto social, avançam hoje em dia de uma forma paralela. Só raramente vemos empresas a desenvolver produtos que exploram combinações de duas ou mais destas tecnologias, tais como a cadeia de bloqueios e a IoT ou os zangões e a inteligência artificial. Embora estejam destinados a convergir, gerando assim um impacto exponencialmente maior, a fase inicial de desenvolvimento em que se encontram e a escassez de promotores e pessoas com perfis técnicos significam que a convergência ainda é uma tarefa pendente.

Devido ao seu potencial disruptivo, espera-se que as tecnologias quânticas não só converjam com todas estas novas tecnologias, mas que tenham uma influência transversal em praticamente todas elas. A computação quântica ameaçará a autenticação, troca e armazenamento seguro de dados, tendo um grande impacto nas tecnologias em que a criptografia tem um papel mais relevante, como a segurança cibernética ou a cadeia de bloqueio, e um impacto negativo menor, mas também a ser considerado em tecnologias como a 5G, IoT ou drones.

Quer praticar computação quântica?

Dezenas de simuladores quânticos de computador já estão disponíveis na rede com diferentes linguagens de programação já em uso tais como C, C++, Java, Matlab, Maxima, [OpenQbit.com](https://openqubit.com)

Python ou Octave. Também, novas línguas como o Q#, lançado pela Microsoft. Pode explorar e jogar com uma máquina quântica virtual através de plataformas como a IBM e a Rigetti.

Mini QRNG é criado pela empresa OpenQbit.com que se concentra no desenvolvimento de tecnologia baseada na computação quântica para diferentes tipos de sectores, tanto privados como públicos.

Porque é que o Mini QRNG é diferente dos outros QRNG, simplesmente porque o sistema foi criado para ser modular e pode ser facilmente montado em casa a um custo bastante baixo.

- (1) <https://blogs.iadb.org/conocimiento-abierto/es/como-funciona-la-computacion-cuantica/>

12. Licenciamento e utilização de software.

Android

<https://source.android.com/setup/start/licenses>

Termux

<https://github.com/termux/termux-app/blob/master/LICENSE.md>

Nó

<https://raw.githubusercontent.com/nodejs/node/master/LICENSE>

Python

<https://www.python.org/download/releases/2.7/license/>

OpenSSH

<https://www.openssh.com/features.html>

Putty SSH

<https://www.chiark.greenend.org.uk/~sgtatham/putty/licence.html>

MIT App Inventor 2 Companheiro e App Inventor Blockly

<https://appinventor.mit.edu/about/termsofservice>

Extensões externas:

JSOINTOOLS

<https://thunkableblocks.blogspot.com/2017/07/jsontools-extension.html>

Licenciamento de versões opensource e comerciais do sistema QRNG Mini ver o site oficial

<http://www.openqbit.com>

Mini QRNG, Mini BlocklyChain, MiniBlockly, BlocklyCode, MiniBlockMiniChain, QBlockly son marcas registradas por OpenQbit.

O Mini QRNG é de domínio público.

Todo o código e documentação no Mini QRNG foi dedicado ao domínio público pelos seus autores. Todos os autores de códigos e representantes das empresas para as quais trabalham assinaram declarações juramentadas dedicando as suas contribuições ao domínio público e os originais dessas declarações juramentadas são guardados num cofre na sede da OpenQbit México. Qualquer pessoa é livre de publicar, usar ou distribuir as extensões originais do Mini QRNG (OpenQbit), seja como código fonte ou como binários compilados, para qualquer finalidade, comercial ou não comercial, e por qualquer meio.

O parágrafo anterior aplica-se ao código e à documentação entregue no Mini QRNG as partes da biblioteca do Mini QRNG que realmente agrupam e enviam com uma aplicação maior. Alguns scripts utilizados como parte do processo de compilação (por exemplo, scripts de "configuração" gerados pelo autoconf) podem ser incluídos noutras licenças de código

aberto. No entanto, nenhum destes guiões de compilação faz parte da biblioteca final QRNG Mini entregável, pelo que as licenças associadas a esses guiões não devem ser um factor de avaliação dos seus direitos de cópia e utilização da biblioteca QRNG Mini.

Todo o código entregável no Mini QRNG foi escrito de raiz. Nenhum código foi retirado de outros projectos ou da Internet aberta. Cada linha de código pode ser rastreada até ao seu autor original, e todos esses autores têm dedicatórias de domínio público em ficheiro. Portanto, a base de código QRNG Mini é limpa e não contaminada com código licenciado de outros projectos de código aberto, não contribuição aberta

Mini QRNG é código aberto, o que significa que pode fazer quantas cópias quiser e fazer o que quiser com essas cópias, sem limitações. Mas o Mini QRNG não é código aberto. Para manter o Mini QRNG no domínio público e assegurar que o código não seja contaminado com conteúdo proprietário ou licenciado, o projecto não aceita correcções de pessoas desconhecidas. Todo o código no Mini QRNG é original, uma vez que foi escrito especificamente para utilização pelo Mini QRNG. Nenhum código foi copiado de fontes desconhecidas na Internet.

O Mini QRNG é do domínio público e não requer uma licença. No entanto, algumas organizações querem uma prova legal do seu direito a utilizar o Mini QRNG. As circunstâncias em que isto ocorre incluem o seguinte:

- A sua empresa quer uma indemnização por queixas de violação de direitos de autor.
- Está a utilizar o Mini QRNG numa jurisdição que não reconhece o domínio público.
- Está a utilizar o Mini QRNG numa jurisdição que não reconhece o direito de um autor a dedicar o seu trabalho ao domínio público.
- Quer ter um documento legal tangível como prova de que tem o direito legal de usar e distribuir Mini QRNG.
- O seu departamento jurídico diz-lhe que deve comprar uma licença.

Se alguma das circunstâncias acima se aplicar a si, a OpenQbit, a empresa que emprega todos os criadores de Mini QRNG, vender-lhe-á uma Garantia de Título de Mini QRNG. Uma Garantia de Título é um documento legal que afirma que os autores reivindicados do Mini QRNG são os verdadeiros autores, e que os autores têm o direito legal de dedicar o Mini QRNG ao domínio público, e que a OpenQbit se defenderá vigorosamente contra as reivindicações de licenciamento. Todas as receitas da venda das garantias do título Mini QRNG são utilizadas para financiar a melhoria contínua e o apoio do Mini QRNG.

Código Contribuído

Para manter o Mini QRNG completamente livre e livre de royalties, o projecto não aceita adesivos. Se quiser fazer uma sugestão de mudança e incluir um remendo como prova de conceito, isso seria óptimo. Contudo, não se ofenda se reescrevermos o seu remendo a partir do zero. O tipo de licença não comercial ou de fonte aberta que a utiliza nesta modalidade e algumas semelhantes, sem compra de apoio, quer individual quer colectiva, independentemente da dimensão da empresa, será regido pelas seguintes premissas legais.

Exoneração de responsabilidade pela garantia. A menos que exigido pela lei aplicável ou acordado por escrito, o Licenciador fornece a Obra (e cada Contribuinte fornece as suas Contribuições) "TAL COMO ESTÁ", **SEM GARANTIAS OU CONDIÇÕES DE QUALQUER TIPO, quer** expressas ou implícitas, incluindo, sem limitação, quaisquer garantias ou condições de TÍTULO, NÃO-VIOLAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM FIM PARTICULAR. É o único responsável por determinar a correcta utilização ou redistribuição da Obra e por assumir quaisquer riscos associados ao seu exercício das permissões ao abrigo desta Licença.

Quaisquer perdas financeiras ou outras incorridas pela utilização deste software serão suportadas pela parte afectada. Todas as disputas legais as partes submeter-se-ão aos tribunais apenas na jurisdição da Cidade do México, país México.

Para apoio comercial, utilização e licenciamento deve ser estabelecido um acordo ou contrato entre a OpenQbit ou a sua empresa e a parte interessada.

Os termos e condições do marketing de distribuição podem ser alterados sem aviso prévio, por favor ir ao site oficial www.openqbit.com para ver quaisquer alterações às cláusulas de apoio e licenciamento não-comerciais e comerciais.

Qualquer pessoa, utilizador, entidade privada ou pública de qualquer natureza legal ou de qualquer parte do mundo que simplesmente utilize o software aceita sem condições as cláusulas estabelecidas neste documento e as que podem ser modificadas a qualquer momento no portal de www.openqbit.com sem aviso prévio e podem ser aplicadas à discrição do OpenQbit em uso não comercial ou comercial.

Quaisquer perguntas e informações sobre o Mini QRNG devem ser dirigidas à comunidade do App Inventor ou às várias comunidades do sistema Blockly, tal como estão: AppBuilder, Trunkable, etc. e/ou para o e-mail opensource@openqbit.com para a procura de perguntas pode demorar 3 a 5 dias úteis a serem respondidas.

Apoio com uso comercial.

support@openqbit.com

Vendas para uso comercial.

sales@openqbit.com

Informação legal e questões ou preocupações sobre licenças

legal@openqbit.com

