



ALDABA SERVICIOS PROFESIONALES S.L

Deseño API REST QRNG

Juan Vilariño Fernández

[15/09/2023]



Unha maneira
de facer Europa.



Fondos Europeos



Despregamento dunha infraestrutura baseada en tecnoloxías cuánticas da información que permita impulsar a I+D+i en Galicia.

Apoiar a transición cara a una economía dixital.

Operación financiada pola Unión Europea, a través do FONDO EUROPEO DE DESENVOLVEMENTO REXIONAL (FEDER) como parte da resposta da Unión á pandemia da COVID-19

DATA	AUTOR	CAMBIOS	VERSIÓN
15/09/2023	Juan Vilariño Fernández		1

Táboa de contidos

1. Servizo API REST	4
1.1) Xerar un token de acceso	4
1.2) Xerar paquetes de números mediante o QRNG	5
1.3) Devolver números aleatorios en continuo	5
2. Frontal de acceso WEB.....	6
2.1) Parte pública.....	6
2.2) Parte privada	6
3. Librería API_QRNG	8
4. Xerador de certificados e chaves ssh.....	9

1. Servizo API REST

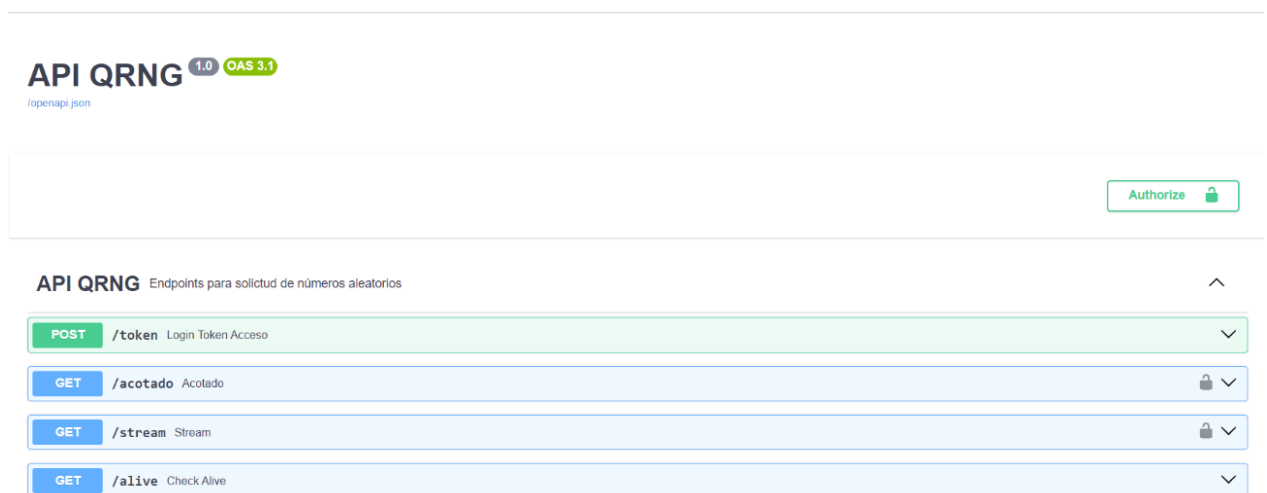
Se deseña e desenvolve un sistema de acceso multiusuario para facilitar o acceso o Xerador Cuántico de Números Aleatorios (QRNG).

O servizo rest API REST Basease en FastApi, que e un framework para Python usando os estándares OpenAPI.

O sistema pode xerar números binarios, enteiros e flotantes de 32 ou de 64 bits.

Na dirección do api, está incluída a documentación de uso.

Por exemplo : <http://193.144.42.120:8000/docs>



The screenshot shows the Swagger UI for the API QRNG. At the top left, it says "API QRNG" with version "1.0" and "OAS 3.1" in a green badge. Below that is the OpenAPI spec file path: "/openapi.json". On the right side, there is an "Authorize" button with a lock icon. The main content area is titled "API QRNG" and "Endpoints para solicitud de números aleatorios". It lists four endpoints:

- POST** /token Login Token Acceso
- GET** /acotado Acotado
- GET** /stream Stream
- GET** /alive Check Alive

O servizo API conta con tres endpoints principais e un auxiliar para comprobar o estado do api.

1.1) Xerar un token de acceso

Para poder facer peticións os endpoints de “/acotado” ou de “/stream”, o primeiro que temos que facer e una chamada de post o token de /token, que nos devolverá un token JWT cunha validad de 15 minutos.

O token firmase usando una chave secreta auto xerada e o algoritmo HS256.

Débese incluír no BODY da petición os campos de *username* e *password*, para poder autenticarse contra o servidor LDAP e garantir o acceso.

1.2) Xerar paquetes de números mediante o QRNG

Unha vez que temos o token xerado, debemos engadilo a cabeceira das peticións.

No endpoint de “/acotado” podemos facerlle una petición o api para que nos devolva un conxunto de números aleatorios.

Podemos pasarlle un parámetro “paquete” para indicar a cantidade de números que queremos xerar(máximo 1000), se non se indica, devolveranse 50.

Teremos que indicar que tipo de número imos xerar co parámetro “tipo_num”, o parámetro e un enteiro que pode ser:

Tipo_num	
0	Binarios
1	Enteiros
2	Flotantes de 32 bits
3	Flotantes de 64 bits

Se non se especifica o tipo de número, por defecto será binario.

1.3) Devolver números aleatorios en continuo

Poderemos arrancar un servizo que nos devolverá números aleatorios en continuo mentres o usuario non cerre a conexión.

Pódese elixir que tipo de número quere xerar indicando o parámetro “tipo_num”

2. Frontal de acceso WEB

2.1) Parte pública

Mostrase información estática sobre o QRNG.



XERADOR CUÁNTICO DE NÚMEROS ALEATORIOS

O Xerador Cuántico de Números Aleatorios (QRNG) é un dispositivo que foi incorporado no mes de novembro de 2021 á infraestrutura de computación e comunicacións do CESGA, incorporouse a actualización á versión 2.0.0 no mes de xaneiro do 2023, a cal reduce os requisitos de procesamento de datos por parte de Python, mellorando o rendemento das aplicacións dos usuarios.

Que permite o QRNG?

Cales son os seus usos?

Datos técnicos

Que permite o QRNG?

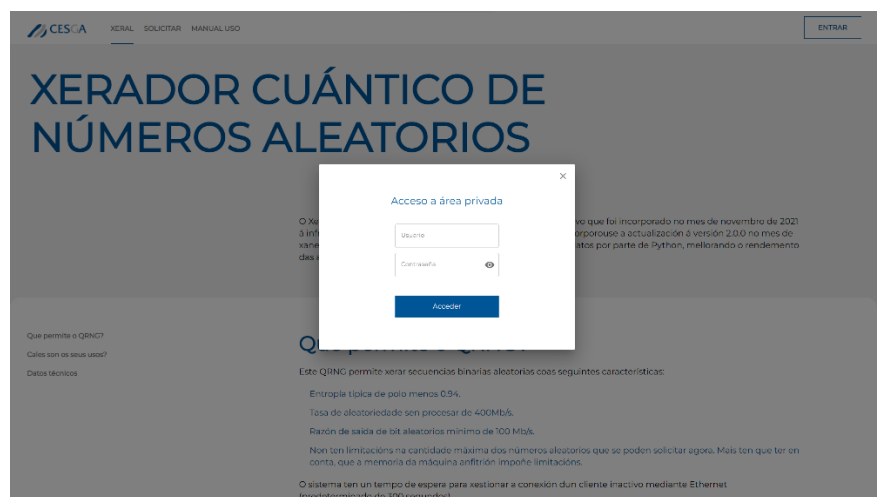
Este QRNG permite xerar secuencias binarias aleatorias coas seguintes características:

- Entropía típica de polo menos 0.94.
- Tasa de aleatoriedade sen procesar de 400Mb/s.
- Razón de saída de bit aleatorios mínimo de 100 Mb/s.

Non ten limitacións na cantidade máxima dos números aleatorios que se poden solicitar agora. Mais ten

2.2) Parte privada

Facendo uso do botón de “Entrar” poderemos autenticarnos do mesmo xeito que no api, de esta forma firmaremos as peticións co token



XERADOR CUÁNTICO DE NÚMEROS ALEATORIOS

O Xerador Cuántico de Números Aleatorios (QRNG) é un dispositivo que foi incorporado no mes de novembro de 2021 á infraestrutura de computación e comunicacións do CESGA, incorporouse a actualización á versión 2.0.0 no mes de xaneiro do 2023, a cal reduce os requisitos de procesamento de datos por parte de Python, mellorando o rendemento das aplicacións dos usuarios.

Acceso a área privada

Usuario

Contraseña

Aceptar

Que permite o QRNG?

Cales son os seus usos?

Datos técnicos

Que permite o QRNG?

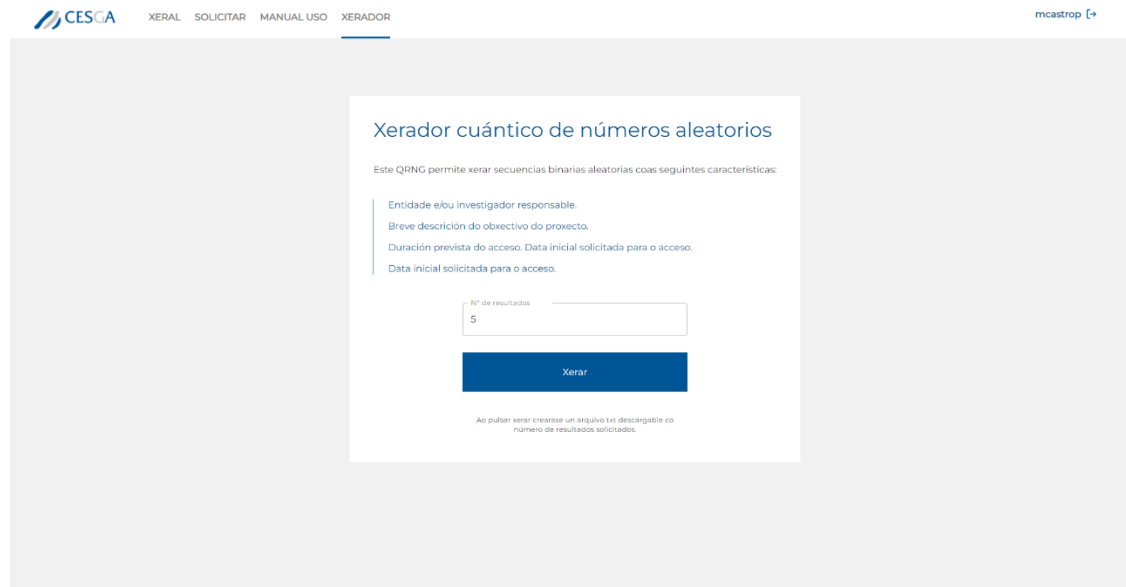
Este QRNG permite xerar secuencias binarias aleatorias coas seguintes características:

- Entropía típica de polo menos 0.94.
- Tasa de aleatoriedade sen procesar de 400Mb/s.
- Razón de saída de bit aleatorios mínimo de 100 Mb/s.

Non ten limitacións na cantidade máxima dos números aleatorios que se poden solicitar agora. Mais ten que ter en conta, que a memoria da máquina anfitrión impoñe limitacións.

O sistema ten un tempo de espera para xestionar a conexión dun cliente inactivo mediante Ethernet (predeterminado de 300 segundos)

Despois de autenticarse, os usuarios poden pedir un paquete de números, a páxina fará una chamada o api, e descargará o paquete, despois se gardarán nun ficheiro .txt que se descarga dende o navegador do usuario.



The screenshot shows a web interface for generating random numbers. At the top left is the CESGA logo, and at the top right is the user name 'mcastrop' with a dropdown arrow. The navigation menu includes 'XERAL', 'SOLICITAR', 'MANUAL USO', and 'XERADOR', with 'XERADOR' being the active page. The main content area is titled 'Xerador cuántico de números aleatorios'. Below the title, it states: 'Este QRNG permite xerar secuencias binarias aleatorias coas seguintes características:'. A list of characteristics follows: 'Entidade e/ou investigador responsable.', 'Breve descripción do obxectivo do proxecto.', 'Duración prevista do acceso. Data inicial solicitada para o acceso.', and 'Data inicial solicitada para o acceso.'. There is a text input field labeled 'Nº de resultados' containing the number '5'. Below the input field is a blue button labeled 'Xerar'. At the bottom of the form, a note reads: 'Ao pulsar xerar crease un arquivo txt descargable co número de resultados solicitados.'

3. Librería API_QRNG

Desenvolveuse unha librería python para facilitar o acceso o API QRNG.

Para instalar a librería de poderemos facelo a través do instalador pip de forma local .

```
pip install api_qrng.whl
```

Facendo uso da librería pódese xerar o token de acceso, paquetes de números aleatorios ou ben arrancar un servizo para devolver números aleatorios ben en continuo ou durante un tempo determinado.

Por exemplo para xerar o token de acceso debemos instanciar a clase de ApiQrng e pasarlle a dirección url e o porto onde está correndo a API.

Despois debemos chamar a función de get_token para autenticarnos contra o servidor LDAP e xerar o token de acceso

```
from api_qrng import ApiQrng  
test = ApiQrng(url = url, puerto=puerto)  
test.get_token('jrfernandez@aldaba.es', "****")
```

Cando xa esteamos autenticados poderemos facer uso das funciones de acotado() ou de stream()

Por exemplo, seguindo o exemplo anterior, se quixésemos xerar un paquete con 100 números e que fosen do tipo binario, bastaría con comprobar que temos o token xerado e invocar a función de acotado indicando que queremos 100 números do tipo_numero 0.

```
if test.token:  
    datos_acotados = test.acotado(paquete = 100,tipo_numero=0)
```


Se o que queremos e iniciar un proceso de que nos devolva números en continuo do tipo binario bastara con facer

```
if test.token:
    datos = test.stream(tipo_numero=0)
    for dato in datos:
        print(dato.decode())
```

Tamén temos dispoñible un parámetro opcional chamado “**timeout**”, por se queremos limitar o tempo do servizo, se quixeramos números durante 5 minutos deberiamos indicar o tempo en segundos da forma:

```
if test.token:
    datos = test.stream(tipo_numero=0, timeout=300)
    for dato in datos:
        print(dato.decode())
```

A librería conta cun script de exemplo de uso chamado “test_api.py” que permite facer unha proba de cada unha dos posibilidades do api.

4. Xerador de certificados e chaves ssh

Inclúese un script chamado “generador_certificados.py” a modo de exemplo, que facendo uso da librería API_QRNG permítenos xerar unha chave ssh engadindo aleatoriedade o proceso.

A chave se exporta automaticamente no directorio do programa co nome “keygen.pub”

Tamén permite xerar un certificado dixital .509 facendo uso do número xerado aleatoriamente.

```
python generador_certificados.py ssh
python generador_certificados.py .509
```