ALDABA

ALDABA SERVICIOS PROFESIONALES S.L

Deseño API REST QRNG

Juan Vilariño Fernández

[15/09/2023]

XUNTA DE GALICIA

Unión Europea

Unha maneira de facer Europa.

Fondos Europeos

CESGA GALICIA SUPERCOMPUTING CENTER

Despregamento dunha infraestructura 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

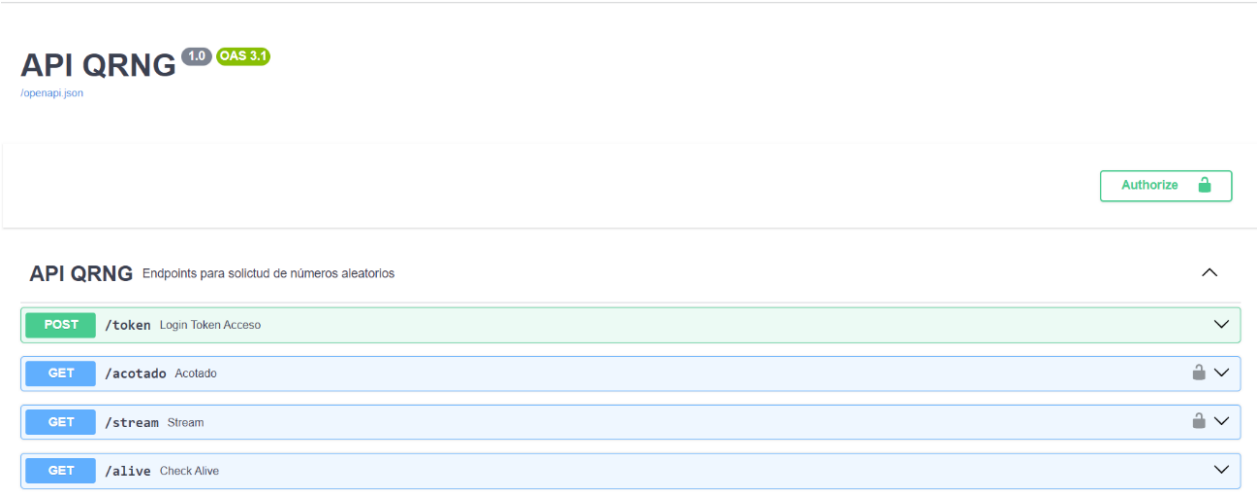
Se deseña e desenvolve un sistema de acceso multiusuario para facilitar o acceso o Xerador Cuántico de Números Aleatorios (QRNG).

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



O servicio 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 valided 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 quérese xerar indicando o parámetro "tipo_num"

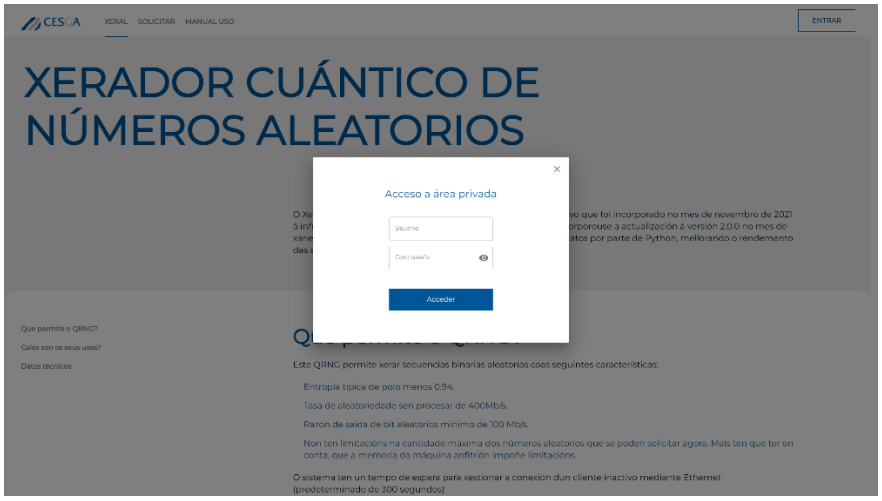
2. Frontal de acceso WEB

2.1) Parte pública

Mostrase información estática sobre o QRNG.

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



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.

3. Librería API_QRNG

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

Para instalar a libraría 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 libraría 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
```