oga data driven solutions

Judería 2, Planta 1, Módulo 3A
Parque Empresarial Vega del Rey
41900, Camas - Sevilla - España
44 Wall Street, Suite 1200
NY 10005, New York – Estados Unidos
+34.900.848.508
oga.ai

Reto 2 - Cifrado mediante secuencias

1. Introducción

Para desarrollar un sistema de cifrado se necesita implementar un algoritmo que identifique los números enteros que cumplan una característica determinada.

Para ello, consideremos la secuencia de números que se forma a partir de un entero, cuando cada elemento se calcula sumando los cuadrados de los dígitos del elemento anterior. Ejemplos:

- **$44 \rightarrow (4^2 + 4^2) = 32 \rightarrow (3^2 + 2^2) = 13 \rightarrow (1^2 + 3^2) = 10 \rightarrow (1^2 + 0^2) = 1 \rightarrow (1^2) = 1$ ...**
- **$2 \rightarrow 4 \rightarrow 16 \rightarrow 37 \rightarrow 58 \rightarrow 89 \rightarrow 145 \rightarrow 42 \rightarrow 20 \rightarrow 4$ ...**

Como se puede observar, las cadenas que contienen **1** o **89** generan ciclos.

El reto consiste en desarrollar un algoritmo que calcule el número de enteros, menores o iguales a un máximo dado, a partir del cual se generen secuencias que contengan el número **89**.

2. Datos

Algunos ejemplos para que puedas probar tu algoritmo:

Máximo	N° de enteros menores o iguales que generan ciclos con 89
2	1
7	2
44	34
85	70

3. Indicaciones técnicas

El algoritmo debe ser **consumible vía API** (será la misma para el resto de los retos de desarrollo), a través de un endpoint (cada reto tendrá el suyo) con la siguiente estructura:

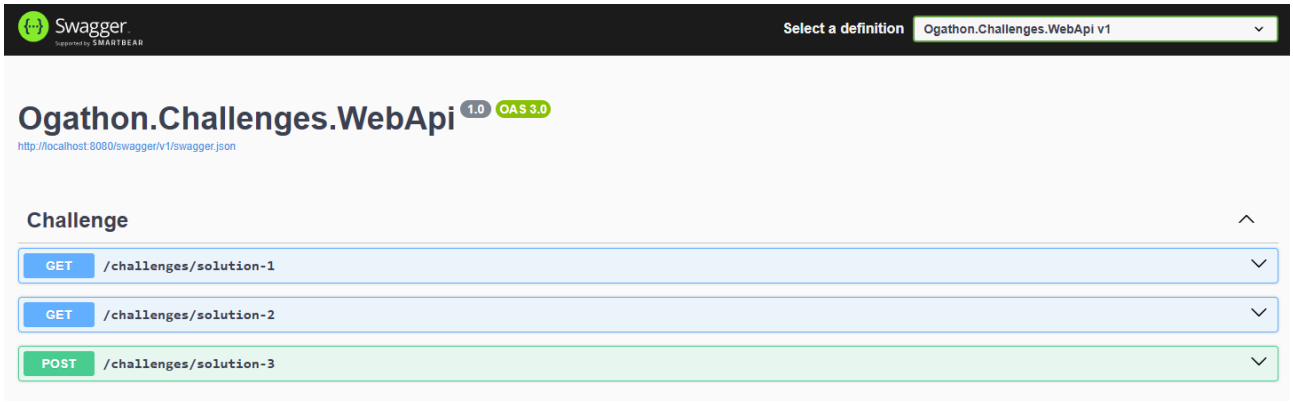
- GET → http://localhost:8080/challenges/solution-2?n=

  * El parámetro *n* el valor que se tomará como máximo para encontrar enteros menores iguales a él, que generen ciclos con 89.

Judería 2, Planta 1, Módulo 3A
Parque Empresarial Vega del Rey
41900, Camas - Sevilla - España
44 Wall Street, Suite 1200
NY 10005, New York – Estados Unidos
+34.900.848.508
oga.ai

- La respuesta debe ser directamente el valor numérico correspondiente al número de enteros menores o iguales a valor dado, que generan ciclos con 89. Por ejemplo: *70*

- Se debe documentar la API mediante **OpenAPI (Swagger o similar)**. Debe ser accesible en http://localhost:8080/swagger y deberá estar tan completa como sea posible.



Esta API debe poder ser desplegada mediante un contenedor **Docker**, con lo cual se debe añadir el fichero **Dockerfile** correspondiente, que exponga la API en el **puerto 8080**. La construcción de la imagen se hará mediante:

> *> cd C:\Repos\Dev\ogathon (aquí debe estar el Dockerfile)*
>
> *> docker build -t ogathon-challenges-api -f Dockerfile .*
>
> *> docker run -d -p 8080:8080 --env ASPNETCORE_HTTP_PORTS=8080 --name ogathon-challenges-api ogathon-challenges-api*

La **estructura del repositorio** debe ser (se incluirán todos los retos de desarrollo en el mismo repositorio, con lo cual habrá un solo Dockerfile que desplegará la API con un endpoint disponible para cada reto):

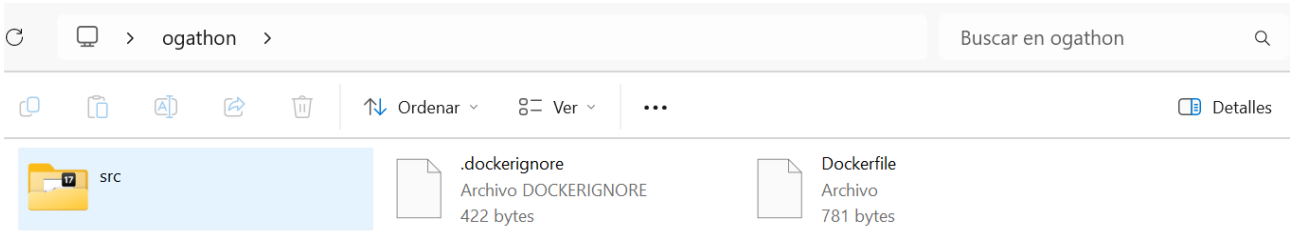
ogathon/

├── src/

│ ├── (código de la solución)

├── Dockerfile

Judería 2, Planta 1, Módulo 3A
Parque Empresarial Vega del Rey
41900, Camas - Sevilla - España
44 Wall Street, Suite 1200
NY 10005, New York – Estados Unidos
+34.900.848.508
oga.ai



El **lenguaje de programación será de libre elección**, siempre y cuando haga posible cumplir con las exigencias del reto.

4. Evaluación

El ganador del reto será la primera persona que implemente el algoritmo indicado, informando del resultado que obtiene para el caso de **máximo = 9100**.

La puntuación máxima que se podrá obtener en el conjunto en los retos de desarrollo será de **100 puntos**. Quedarán repartidos de la forma que sigue.

Se podrán sumar un total de **24 puntos** en este reto.

- Exposición vía API y Docker: 12 puntos **(será requisito indispensable para seguir evaluando)**
- Resultado correcto: 8 puntos
- Tiempos de respuesta: 4 puntos
  - ≤ 100ms: 4 puntos
  - 101ms - 500ms: 2 puntos
  - 501ms - 1s: 1 puntos
  - Más de 1s: 0 puntos

Adicionalmente, para el global de los retos, se podrán obtener otros **28 puntos**.

- Documentación OpenAPI: 5 puntos
- Calidad en código: 20 puntos
  - Bugs y Vulnerabilidades (8 puntos)
    - 0 errores → 8 puntos
    - 1-3 errores → 4 puntos
    - 4+ errores → 0 puntos
  - Code Smells (4 puntos)
    - Menos de 10 → 4 puntos
    - Entre 10-20 → 2 puntos

Judería 2, Planta 1, Módulo 3A
Parque Empresarial Vega del Rey
41900, Camas - Sevilla - España
44 Wall Street, Suite 1200
NY 10005, New York – Estados Unidos
+34.900.848.508
oga.ai

        - Más de 20 → 0 puntos

    - Cobertura de Código (8 puntos)
        - ≥ 85% → 8 puntos
        - 70%-85% → 4 puntos
        - Menos de 70% → 0 puntos

- Tiempo de Resolución (3 puntos)
    - Se otorgan 3 puntos al equipo que termine los 3 retos primero.