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 1 - Predicción de propagación vírica

1. Introducción

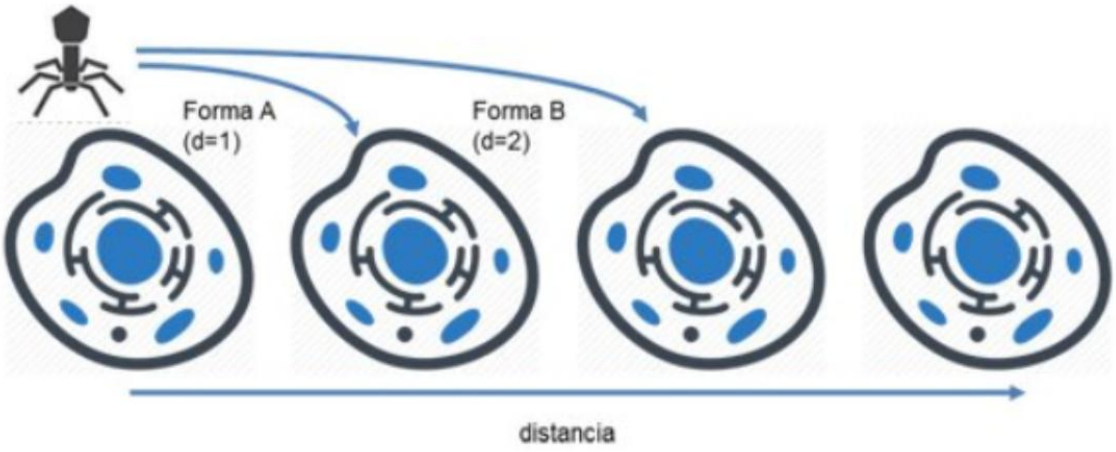
Para desarrollar un sistema de predicción en el ámbito sanitario, necesitamos implementar un algoritmo que calcule el número de patrones de propagación de un virus.

Para ello, es necesario saber que el virus en cuestión tiene dos formas de propagación en un momento determinado:

- **Forma A**: propagación por salto a una célula contigua (**distancia = 1**).
- **Forma B**: propagación por salto a una célula inmediatamente posterior a la contigua (**distancia = 2**).

El reto consiste en desarrollar un algoritmo que calcule el número de patrones diferentes para que la propagación llegue a una distancia determinada, suponiendo que se propaga en una única dimensión y en un único sentido.

Por ejemplo, si la distancia es 3, los patrones son 3: "AB", "BA" y "AAA".



2. Datos

Algunos ejemplos para que puedas probar tu algoritmo:

Distancia	N° de patrones
3	3
10	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

Distancia	N° de patrones
20	10,946
50	20,365,011,074

3. **Indicaciones técnicas**

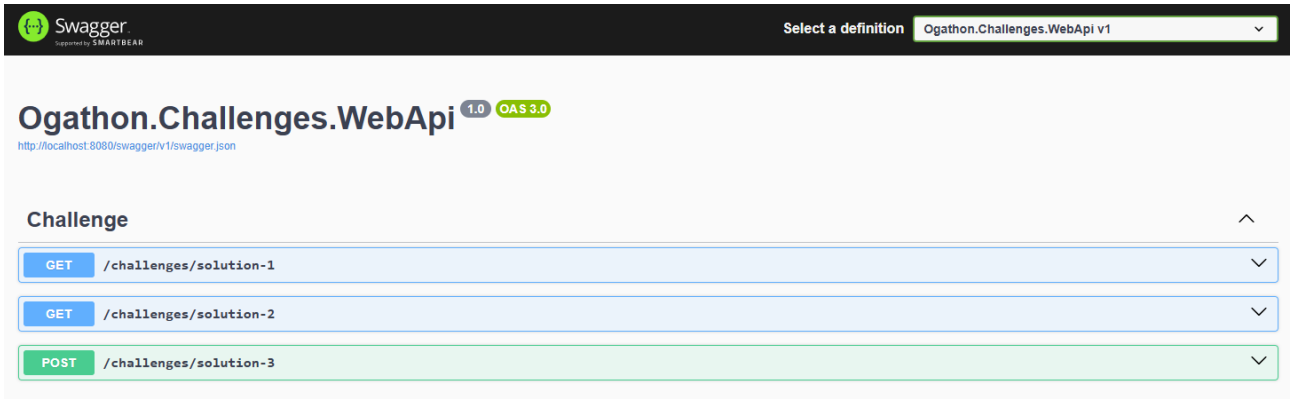
El algoritmo debe ser **consumible vía API** (será la misma para el resto de los retos de desarrollo), a través de un endpoint (cada reto tendrá el suyo) con la siguiente estructura:

- GET → http://localhost:8080/challenges/solution-1?n=

  * El parámetro *n* será la distancia a partir de la cual se tiene que calcular el número de patrones.

- La respuesta debe ser directamente el valor numérico correspondiente al número de patrones calculado. Por ejemplo: *10946*

- 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

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

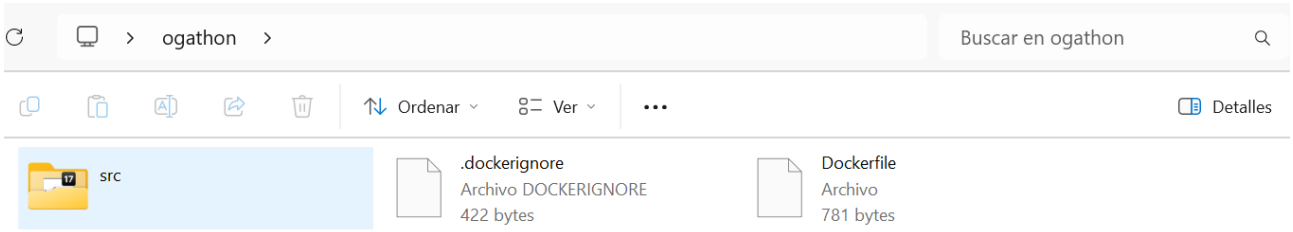
La **estructura del repositorio** debe ser (se incluirán todos los retos de desarrollo en el mismo repositorio, con lo cual habrá un solo Dockerfile que desplegará la API con un endpoint disponible para cada reto):

ogathon/

├── src/

│ ├── (código de la solución)

├── Dockerfile



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

4. Evaluación

Se pide encontrar el resultado que se obtiene para el caso de **distancia = 91**.

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

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

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
        - 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.