

Reto 3 - Reciclaje efectivo

1. Introducción

Para desarrollar un sistema de reciclaje, necesitamos implementar un algoritmo que calcule los movimientos a realizar para que los residuos queden perfectamente separados.

Para ello, tendremos **3 contenedores**, cada uno de ellos con **3 tipos de residuos** que, en general, podrán venir mezclados:

- **Vidrio**
- **Plástico**
- **Cartón**

El reto consiste en **calcular el número mínimo de movimientos** de los elementos de residuo indicados, para que los contenedores queden con los residuos separados.

2. Datos

Por ejemplo, con la siguiente distribución de elementos:

Contenedor	Vidrio	Cartón	Plástico
1	2	0	1
2	0	3	1
3	1	1	1

El **mínimo número de elementos a mover** es de **4**, como se indica en la siguiente figura, para que los residuos queden separados:

- **1 de plástico** del contenedor 1.
- **1 de plástico** del contenedor 2.
- **1 de vidrio** y **1 de cartón** del contenedor 3.

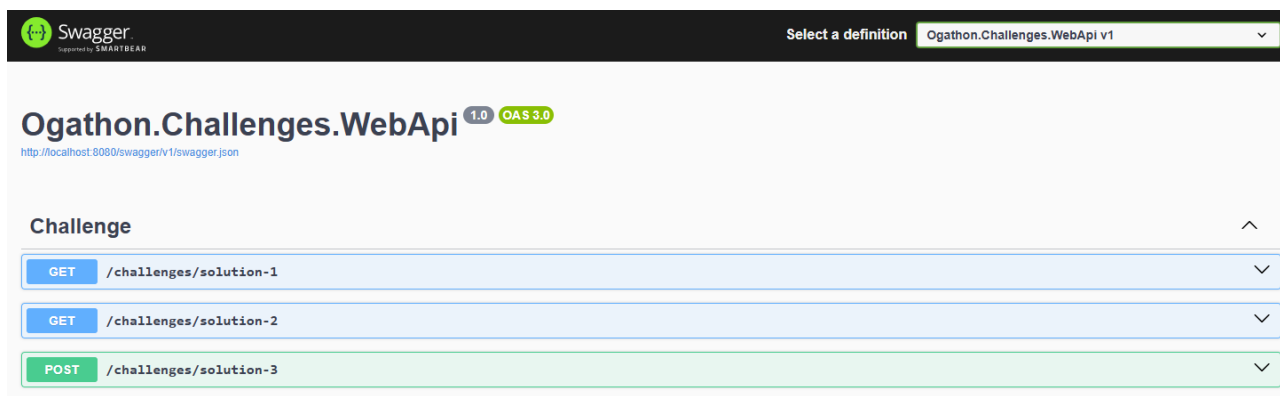
3. Indicaciones técnicas



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:

- POST → <http://localhost:8080/challenges/solution-3>
 - * Body (int[][]). La primera posición representa al contenedor y la segunda al residuo. Ejemplo (el caso de la tabla anterior):


```
[
    [2, 0, 1],
    [0, 3, 1],
    [1, 1, 1]
]
```
- La respuesta debe ser directamente el valor numérico correspondiente al mínimo número de movimientos a realizar para ordenar los contenedores. Por ejemplo: 4
- 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.

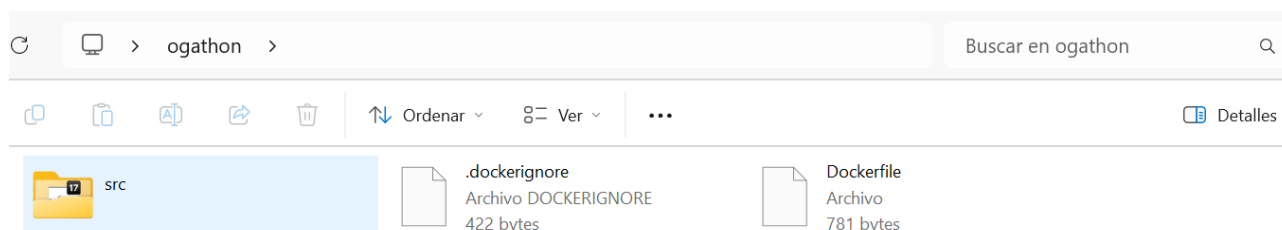


Esta API debe poder ser desplegada mediante un contenedor **Docker**, con lo cual se debe añadir el fichero **Dockerfile** correspondiente, que exponga la API en el **puerto 8080**. La construcción de la imagen se hará mediante:

- ```
> 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
```



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:

| Contenedor | Vidrio | Cartón | Plástico |
|------------|--------|--------|----------|
| 1          | 1      | 3      | 2        |
| 2          | 2      | 1      | 3        |
| 3          | 3      | 2      | 1        |

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
  - $\leq 100\text{ms}$ : 4 puntos
  - 101ms - 500ms: 2 puntos
  - 501ms - 1s: 1 punto
  - 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
    - Más de 20 → 0 puntos
  - Cobertura de Código (8 puntos)
    - $\geq 85\%$  → 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.

