

# Prueba Técnica Star Wars: Operación Firestorm

Son tiempos adversos para la rebelión, las últimas informaciones provenientes de nuestros servicios de inteligencia indican que el imperio está desarrollando la Estrella de la Muerte, un arma definitiva que amenaza la paz en la galaxia. La única esperanza para el ejército rebelde es que el *Red Sq adron* se infiltre en la base Imperial del *plane a Scarif* y robe sus planos, para encontrar sus puntos débiles y poder destruirla.

Como maniobra de distracción se atacará intensamente las defensas de la base enemiga; para ello el *Comandan e Gra en Drei* ha ordenado delimitar con láser las áreas que deben recibir una potente descarga de proyectiles que serán lanzados desde el *Cr cero E elar Mon Calamari* el cual se encuentra orbitando el planeta.

La información técnica para calcular el área de ataque queda descrita a continuación en amarillo.

Calcular el área de la región delimitada entre la gráfica de la hipérbola  $f$  y el eje de abscisas, siendo  $f(x)$ .

$$f(x) = \frac{1}{x}, 2 \leq |x| \leq 3$$

La región está compuesta por dos regiones, una delimitada por las dos siguientes rectas:

$$x = -3, x = -2$$

y la otra por las dos siguientes rectas:

$$x = 3, x = 2$$

Dada la simetría de las áreas es suficiente calcular el área de una de las dos regiones y duplicarla.

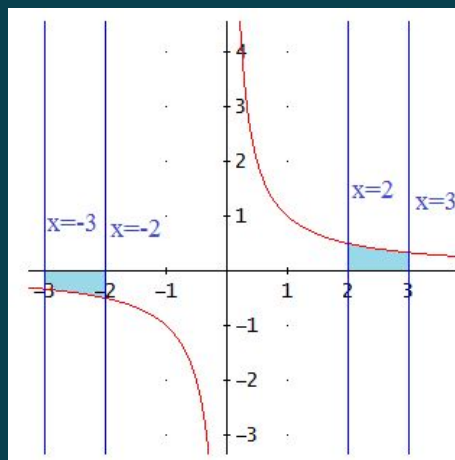


gráfico que dibuja las áreas

Nota: Solo es posible calcular el área utilizando el método de los trapecios. No se debe emplear ningún otro método como integración por diccionarios u otros sistemas de aproximación de cálculo de áreas.

## Como ingeniero de élite de la rebelión se te demanda lo siguiente

Habilitar en los sistemas rebeldes un microservicio que contenga un endpoint Rest y una entrada desde consola para poder calcular el área de ataque y que así el *cr cero e elar Mon Calamari* pueda aprovisionar el número necesario de *pro ec ile pro on orpedo*.

Para cada petición el Alto Mando establecerá la precisión  $n$ , entendiendo  $n$  como el número de intervalos usado en el cálculo del área mediante el método de los trapecios. No es posible que el mando establezca una precisión  $n$  superior a 15.000 ni inferior a 100, si se incumplen estos valores el microservicio debe lanzar una excepción.

Cada petición de cálculo de área será única e identificada por un UUID; asimismo aunque todos los valores que componen la petición sean iguales a los de otra; si sus identificadores difieren, las peticiones serán diferentes.

Para ajustar los sensores de los *mi ile pro on orpedo*, el Alto Mando exige adjuntar la información meteorológica de la *Ci dadela de Scarif* en el día siguiente a cada petición de cálculo de Área. Para ello se debe utilizar el servicio <https://openweathermap.org/api>. (usa la ciudad de Barcelona como si se tratase de la *ci dadela de Scarif*)

Es vital para la causa pasar desapercibidos, así que las peticiones al servicio openweathermap deben ser las mínimas posibles, sería ideal desarrollar un sistema de caché para conseguir tal fin. Como nota importante los climatólogos de la rebelión estiman que las previsiones meteorológicas tienen una validez de 3 días.

Es necesario persistir todas las peticiones de cálculo para que se puedan recuperar mediante su UUID.

## Resumen de la petición

Debes desarrollar dos use cases

- calculateArea:
  - Parámetros de entrada:
    - Un numero natural n.
    - Un UUID para identificar ese calculo de area.
- getAreaById:
  - Parámetros de entrada
    - UUID para obtener ese cálculo de área.
  - La response en formato json debe quedar como ilustra el siguiente ejemplo:

```
{
  "response":{
    "area":0.8109302162,
    "weather":{
      "wind":"Light breeze, 2.6 m/s, East-northeast ( 60 )",
      "humidity":"83%"
    }
  }
}
```
- Los endpoint Rest y los comandos de consola ejecutan los mismos casos de uso y tienen las mismas salidas.

## ¿Ambigüedad en el enunciado?

- No te preocupes si hay puntos ambiguos, que no entiendes correctamente o que consideras que son erróneos, no worries. Desarrolla lo que entiendas y consideres oportuno.
- Si no entiendes el cálculo de áreas que se te demanda mediante del método de los trapecios, devuelve un float random en su lugar intentando seguir todos las otras features que se te piden.
- Explica tus consideraciones en el readme para que podamos entender que has decidido entregarnos.

## ¿Cómo hago la entrega y que es necesario?

- Desarrolla el microservicio usando arquitectura DDD y CRQS.
- Crea un buen dominio y ten en cuenta patrones de diseño, pensando que el alto mando te pedirá más features en el futuro.
- Usa php 7.3 y symfony 4.
- Puedes usar cualquier infraestructura que consideres oportuna, tales como, doctrine con mysql , redis y otros.
- Pon en el readme todo lo necesario para hacer funcionar la aplicación.
- No desarrolles ningún elemento front ni nada más de lo que se te demanda.

## Se valorará

- Arquitectura Event Driven.
- Dockerizado de la aplicación.
- Documentacion con OpenApi.
- Añade todo aquello a nivel de testing que consideres oportuno. (unitarios, e2e ....)
- Si lo consideras oportuno puedes añadir cualquier artefacto de documentación tales como UML, diagramas BPMN , fotos de tu proceso de event storming u otros.

Muchas gracias por estar en el proceso de Nemuru y  
¡Qué la fuerza de acompañe!