La búsqueda de los artificios

En un mundo medieval fantástico, diversas facciones compiten por encontrar artefactos antiguos que otorgan habilidades mágicas a sus portadores. La facción de los "Guardianes de la Luz", liderada por el astuto y sabio anciano *Elowen*, ha decidido que es momento de rastrear y asegurar estos artefactos antes de que caigan en manos equivocadas.

Tu tarea: *Elowen* te ha contratado como un experto en tecnología mágica para desarrollar un sistema que analice antiguos manuscritos y determine si contienen pistas sobre la ubicación de los artefactos. Para ello, necesitas crear una función con la siguiente firma (en alguno de los siguientes lenguajes: Java / Golang / C-C++ / Javascript (node) / Python / Ruby):

```
boolean containsArtifactClue(String[] manuscript); // Ejemplo en Java
```

La función recibirá como parámetro un array de *Strings* que representan cada línea de un manuscrito antiguo. Las letras pueden ser cualquier carácter alfabético.

Criterio de Identificación

Un manuscrito contiene una pista sobre la ubicación de un artefacto si se encuentran secuencias de cuatro letras consecutivas en cualquier dirección (horizontal, vertical o diagonal).

Ejemplo (*Caso de Clue sobre el Artefacto*):

```
String[] manuscript = {"RTHGQW", "XRLORE", "NARURR", "REVRAL", “EGSILE", "BRINDS"};
```

En este caso, el llamado a la función `containsArtifactClue(manuscript)` devolvería `true`, ya que hay cuatro letras "R" en la diagonal desde (0, 0) a (3, 3).

Desafíos

- **Nivel 1:** Implementar el método solicitado por *Elowen*.
- **Nivel 2:** Crear una API REST, hospedada en un servicio de cloud computing (Google App Engine, Amazon AWS, etc.). Exponer el servicio “*/clue/*” donde se pueda enviar el manuscrito mediante un **HTTP POST** con el siguiente formato JSON:

  ```
  POST → /clue/ { "manuscript":
  ["RTHGQW","XRLORE","NARURR","REVRAL","EGSILE","BRINDS"] }
  ```

  En caso de que se encuentre una pista sobre el artefacto, debería devolver un **HTTP 200-OK**; en caso contrario, un **403-Forbidden**.
- **Nivel 3:** Incorporar una base de datos para almacenar los manuscritos analizados y sus resultados. Solo se debe registrar un manuscrito único. Exponer un servicio adicional “/stats” que devuelva un JSON con las estadísticas de las búsquedas:

```
{"count_clue_found": 40,"count_no_clue": 100,"ratio": 0.4}
```

Tener en cuenta que la API puede recibir un tráfico muy variable, desde **100** hasta **1 millón** de peticiones por segundo.

Requisitos de entrega

1. Código fuente (para Niveles 2 y 3, en
2. un repositorio de GitHub).
3. Instrucciones para ejecutar el programa o la API (para Niveles 2 y 3, en el README de GitHub).
4. URL de la API (para Niveles 2 y 3).
5. Realizar una prueba de performance en *JMeter* o *Postman*, y publicar los resultados en github donde se pueda las transacciones por segundo alcanzadas y el percentil 90 de los tiempos de respuesta.