

1ºER PARCIAL: DESAFÍO MERCADO LIBRE

Este informe busca explicar y documentar el funcionamiento de la **API** que se ha escrito en las últimas semanas.

Directivas:

- Escribir algoritmo que cumpla con los requisitos del desafío propuesto por **Mercado Libre**.
- Escribir una **API REST** que haga uso del algoritmo previamente codificado y lo conecte con una base de datos.
- Exponer endpoints para ingresar secuencias de **ADN** y para visualizar estadísticas numéricas.
- Hostear la **API REST** en un servicio de cloud computing (En el caso de este proyecto en específico, se ha utilizado **Render**).

ALGORITMO

Para facilitar la lectura de este documento, no se mostrarán capturas del código, ya que se encuentra en el mismo repositorio que el informe que se está leyendo ahora mismo. Además, se asume que el lector comprende y conoce el desafío de **Mercado Libre** previamente a la lectura de este documento.

El algoritmo escrito para la solución del desafío emplea una estrategia conocida como “**Retorno Temprano**” la cual retorna un valor dentro de una función, antes de que la misma finalice su ejecución. Esta estrategia nos permite obviar la iteración a través de **NxN** datos, ya que el retorno temprano detendrá la ejecución de la función al momento de detectar una secuencia mutante.

La aplicación mas relevante del **retorno temprano** en esta solución se encuentra antes de comenzar la iteración por la secuencia de **ADN**. Mientras el algoritmo convierte el array unidimensional a un array bidimensional (También conocido como **Matriz**), busca un patrón mutante en cada **String**. En caso de encontrar uno de esos patrones, el **retorno temprano** para completamente la ejecución del algoritmo y retornara un valor **Verdadero**.

Al parar la ejecución mientras se convierte el array se evita completamente la secuencia de iteración a través de la matriz.

Esta solución ha sido sometida a **pruebas de estrés intensas**, con entradas de hasta 10.000 x 10.000. Tomando alrededor de 160 milisegundos (0.16 segundos).

API REST

Esta API REST utiliza el patron de **MVC (Modelo-Vista-Controlador)**, y expone 2 endpoints:

- **/mutant**
 - Método: POST
 - Ingresa una cadena de Strings, que representa una secuencia de ADN.
 - Ejemplo de cadena: {"dna": ["AGCCCT", "CATTGC", "TATTGT", "AAATGG", "TACACT", "TCACTG"] }
 - Posibles resultados:
 - **200-OK**: La request fue aceptada y se encontró una secuencia mutante.
 - **403-FORBIDDEN**: La request fue aceptada, pero no se encontró una secuencia mutante.
 - **400-BAD-REQUEST**: La request no fue aceptada ya que el formato es incorrecto.
- **/stats**
 - Método: GET
 - Devuelve estadísticas numéricas de la base de datos (Cantidad de mutantes, cantidad de humanos y ratio de mutantes a humanos).
 - Posibles resultados:
 - **200-OK**: La request fue aceptada y se devolvieron las estadísticas numéricas.

CONCLUSIONES

Este proyecto fue un excelente ejercicio mental, tanto por el algoritmo como por la aplicación en forma de servicio web. Personalmente, he aprendido como utilizar herramientas como **Docker** y **DockerHub**, las cuales son de suma importancia a la hora de desplegar aplicaciones en servicios como **Render**.