**Documentación Pruebas Mercado Libre**

**Sebastián Tobón Carvajal**

**Medellín, Mayo 2022**

**Componente realizado en Node Js.**

1. Descripción del proyecto: Api capaz de analizar cadenas de ADN en busca de coincidencias mutantes. Sus métodos son capaces de entregar un análisis completo o parcial del ADN y permite obtener las estadísticas de las cadenas analizadas con anterioridad. Este componente contiene toda la lógica utilizada para determinar si una secuencia de ADN es mutante o no. El api no interactúa directamente con la base de datos, sino que consume otro servicio creado para realizar las operaciones sobre el BD.
2. Tecnologías:

* Node js 16.15.0.
* Instalador de paquetes npm 8.5.5.

1. Publicación: El desarrollo fue publicado en un servidor Heroku utilizando la línea de comandos otorgados por Heroku CLI.
2. Preparación de ambiente de desarrollo:

* Clonar el proyecto.
* Crear una nueva rama con **git checkout –b “nombre\_rama”.**
* Abrir id Visual Studio Code y cargar el proyecto.
* Entrar a <https://nodejs.org/es/download/> y descargar el instalador de Node, posteriormente hacer la instalación.
* Ir a <https://nodejs.org/es/download/> y descargar node js. Posteriormente hacer la instalación.
* Ir a <https://devcenter.heroku.com/articles/heroku-cli> y descargar heroku cli. Posteriormente hace la instalación.
* En caso de necesitar más componentes hacer uso de npm para su instalación.

1. Endpoints del api:

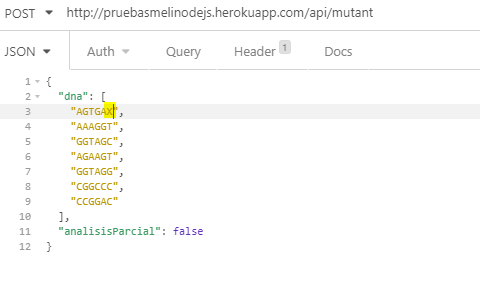
* Get: /api/mutant
  + Descripción: Levanta un ambiente de swagger con documentación e instrucciones de consumo.
* Get: /api/ stats
  + Descripción: Devuelve las estadísticas de numero de mutantes (mutants), humanos (humans) y razón (ratio).
  + Respuesta: Retorna objeto JSON.
  + Parámetros de la respuesta:
    - count\_mutant\_dna: Cantidad de mutantes identificados hasta la fecha. (Integer).
    - cont\_human\_dna: Cantidad de humanos identificados hasta la fecha. (Integer).
    - ratio: Razón entre mutantes y humanos. (String).
  + Códigos de respuesta:
    - 200: Se retorna cuando se hizo el análisis correctamente.
    - 400: Se retorna cuando hay errores en la petición.
* Post: /api/mutant
  + Descripción: Permite evaluar una cadena de ADN y determinar si pertenece a un mutante o no. El método permite hacer un análisis parcial o un análisis completo.
  + Entrada: Recibe un body tipo JSON.
  + Parámetros del body:
    - dna: Contiene las bases nitrogenadas a evaluar. Recibe un arreglo de cadenas de texto y no es case sensitive. (Array).
    - analisisParcial: Recibe true para hacer análisis parcial, es decir, en caso de encontrar dos cadenas mutantes en el ADN interrumpe el análisis y retorna el resultado. Recibe false para hacer el análisis completo del ADN y retorna no solo si es mutante o no sino cuantas y cuáles fueron las secuencias mutantes encontradas. (Boolean).
  + Headers: Content-Type = application/json.
  + Respuesta: Retorna objeto JSON.
  + Parámetros de la respuesta:
    - secuenciaValida: Retorna true en caso de que las bases nitrogenadas ingresadas en el body sean solo A, C, G o T. Retorna false si se ingresa una base nitrogenada diferente a estas. (Boolean).
    - mutante: Retorna true si se encontró un mutante y false en si no se encontró. (Boolean).
    - AnalisisParcial: Retorna el valor ingresado en el body de entrada. (Boolean).
    - secuenciasMutantesDetectadas: Se retorna el número de secuencias mutantes encontrada. Este valor solo se calcula si el analisisParcial definido es false. (Integer).
    - analisisCompleto: Retorna las bases nitrogenadas de las secuencias mutantes encontradas. Este valor solo se calcula si el analisisParcial definido es false. (Array).
  + Códigos de respuesta:
    - 200: Se retorna cuando se encontró un mutante.
    - 403: Se retorna cuando no se encontró mutante o cuando la secuencia de ADN ingresada no es válida.
    - 400: Se retorna cuando hay errores en la petición.

1. Url base: <http://pruebasmelinodejs.herokuapp.com>
2. Datos del autor:

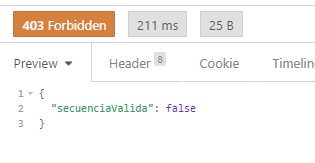
* Nombre: Sebastián Tobón Carvajal.
* Correo: [sebastiantc19@gmail.com](mailto:sebastiantc19@gmail.com)
* Contacto: 3207523478.

1. Ejemplos de consultas y respuestas realizados con insomnia:

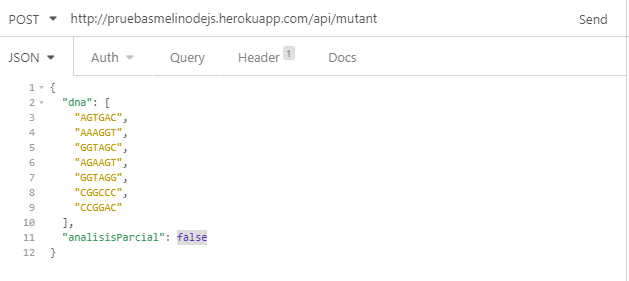
* Get: /api/mutant
  + Entrada: Se ingresa una cadena que tiene un gen diferente a A, C, G o T.



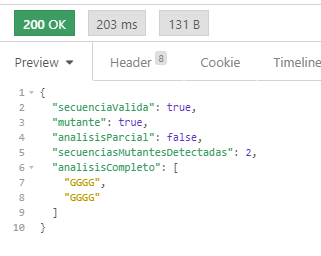
* + Respuesta: Se retorna secuenciaValida false indicando que es una cadena que no corresponde a una cadena humana analizable.



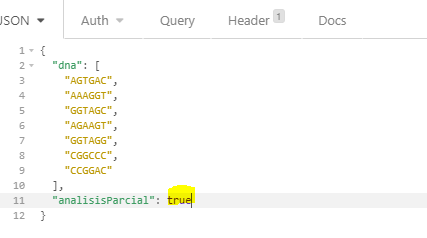
* + Entrada: Se ingresa una secuencia valida con analisisParcial false.



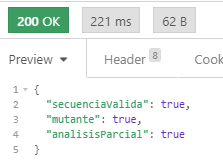
* + Respuesta: Se retorna secuenciaValida true y los datos correspondientes al análisis. Como el análisis hecho es completo se indica la cantidad y la naturaleza de las secuencias mutantes.



* + Entrada: Se ingresa una secuencia valida con analisisParcial true.



* + Respuesta: Se retorna secuenciaValida true y los datos correspondientes al análisis. Como el análisis es parcial es retorna el detalle de las secuencias mutantes detectadas. Este método una vez encuentra dos coincidencias mutantes se interrumpe en su ejecución y retorna.



**Swagger**

1. Versión: 3.0.0.
2. Descripción Framework utilizado para la documentación del api hecha en node js.

**Componente realizado en .NET**

1. Descripción del proyecto: Api capaz interactuar con la base de datos que almacena los registros de los ADN analizados, su fecha y su resultado del análisis.
2. Tecnologías:

* .NET framework 4.5.
* Lenguaje C#.

1. Publicación: El desarrollo fue publicado en un servidor somee: <http://somee.com/>
2. Preparación de ambiente de desarrollo:

* Clonar el proyecto.
* Abrir Visual Studio y cargar el proyecto.
* Correr el proyecto.
* En caso de necesitar componentes adicionales hacer uso de la terminal de Visual Studio para su instalacion.

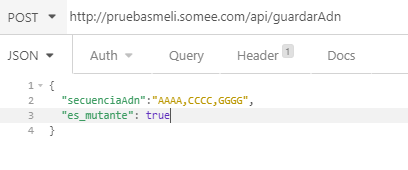
1. Endpoints del api:

* Get: / api/estadisticas
* Descripción: Permite obtener de la base de datos la cantidad total de humanos y mutantes hasta la fecha Respuesta: Retorna objeto JSON.
  + Parámetros de la respuesta:
    - count\_mutant\_dna: Cantidad de mutantes identificados hasta la fecha. (Integer).
    - cont\_human\_dna: Cantidad de humanos identificados hasta la fecha. (Integer).
  + Códigos de respuesta:
    - 200: Se retorna cuando se hizo el análisis correctamente.
    - 400: Se retorna cuando hay errores en la petición.
* Post: /api/guardarAdn
  + Descripción: Permite insertar en la base de datos el ADN, el resultado y la fecha de procesamiento; asociado a un id autoincrementable. Si una secuencia ya fue insertada previamente el método no la vuelve a insertar en el BD.
  + Entrada: Recibe un body tipo JSON.
  + Parámetros del body:
    - secuenciaAdn: Contiene las bases nitrogenadas de la secuencia de ADN. (String).
    - es\_mutante: Valor a almacenar con el resultado del análisis y que determina si es mutante o no. (Boolean).
  + Headers: Content-Type = application/json.
  + Respuesta: Retorna objeto JSON.
  + Parámetros de la respuesta:
    - response.codigo: 0 si todo esta ok o 1 si hay un error. (Integer).
    - response.respuesta: Mensaje de respuesta de la operación. Contiene un ok o la tipificación del error. (String).
    - response.mensaje: Retorna si el usuario pudo ser insertado o no en la base de datos. En caso de que el usuario ya exista no lo inserta nuevamente. (String).
    - status: Devuelve el código de respuesta de la operación. (Integer).
    - exeption: Retorna la descripción de la posible excepción. (String).
  + Códigos de respuesta:
    - 200: Se retorna cuando se encontró un mutante.
    - 400: Se retorna cuando hay errores en la petición.

1. Url base: <http://pruebasmeli.somee.com>
2. Datos del autor:

* Nombre: Sebastián Tobón Carvajal.
* Correo: [sebastiantc19@gmail.com](mailto:sebastiantc19@gmail.com)
* Contacto: 3207523478.

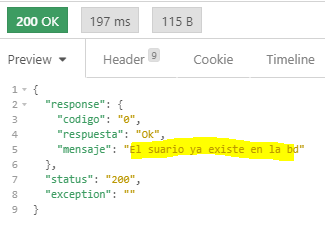
1. Ejemplos de consultas y respuestas realizados con insomnia:
   * Entrada: Se ingresa una secuencia de ADN.



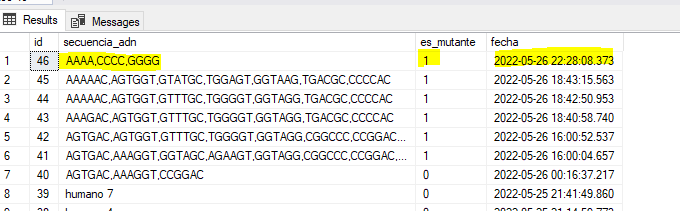
* + Respuesta: Inserción correcta.



* + Si se vuelve a intentar insertar esa misma cadena ya responderá que no se puede insertar debido a que ya fue insertada previamente.



* + Resultado en la base de datos.



**Componente de base de datos**

1. Publicación: Servidor <http://somee.com/>
2. Nombre BD: PruebasMeli
3. Nombre tabla: ResultadosPruebasAdn
4. Columnas tabla:

* id: int. Llave primaria y autoincrementadle.
* secuencia\_adn: varchar(MAX).
* es\_mutante: bit.
* fecha: datetime