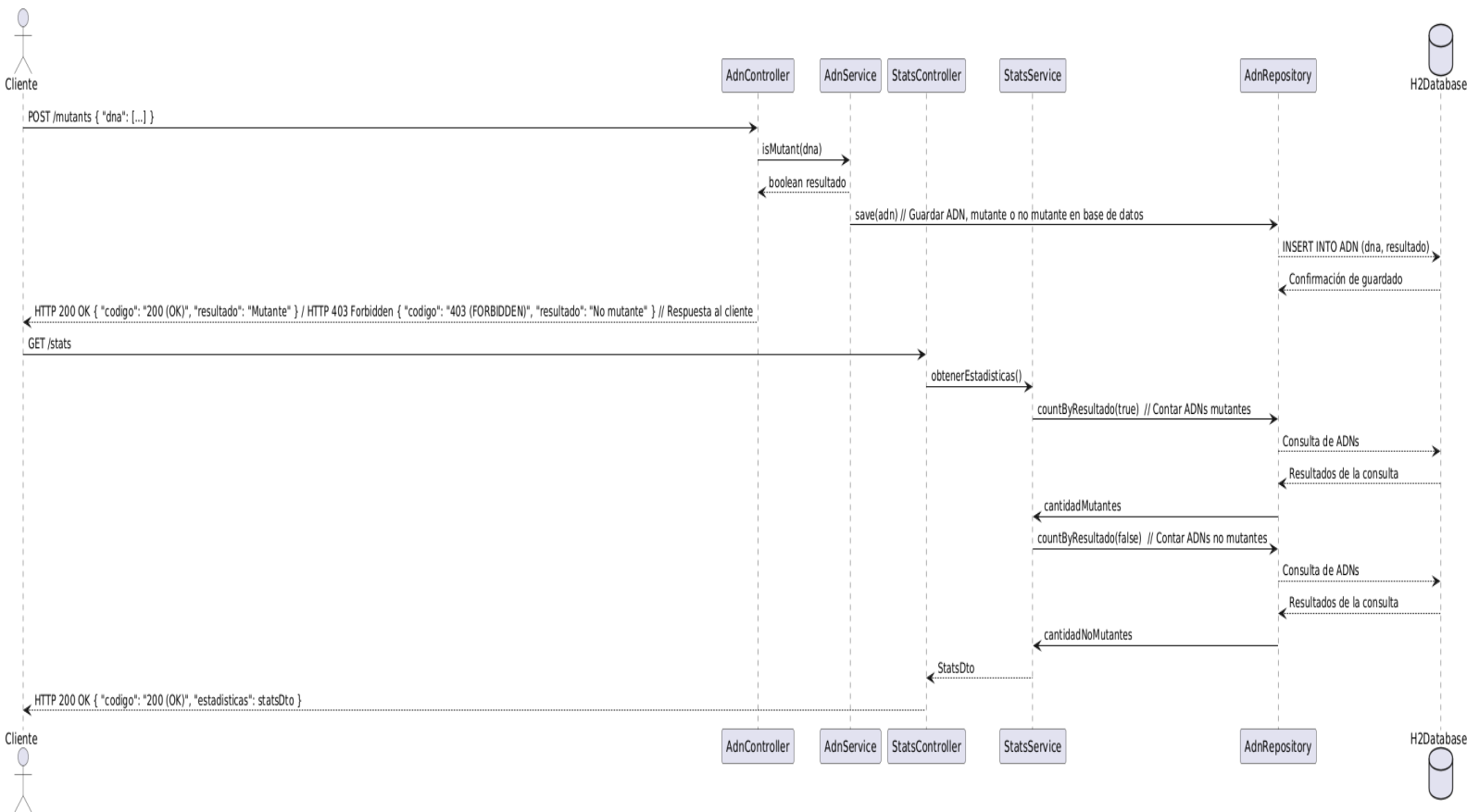


# DIAGRAMA DE SECUENCIA



## DESCRIPCION GENERAL

El diagrama de secuencia muestra las interacciones entre los actores y los componentes del sistema en un flujo de operaciones relacionado con la validación de secuencias de ADN y la obtención de estadísticas de mutantes. Permite entender mejor la arquitectura del sistema y cómo se gestionan las solicitudes del usuario en el backend. La separación de responsabilidades entre los controladores, servicios y repositorios facilita la mantenibilidad y escalabilidad de la aplicación, asegurando que cada componente maneje su función específica de manera eficiente.

### Elementos del Diagrama

#### Actores:

- **Cliente:** Representa al usuario que realiza solicitudes al sistema a través de una API REST.

## Componentes:

- **AdnController**: Controlador encargado de manejar las solicitudes relacionadas con las secuencias de ADN.
- **AdnService**: Servicio que contiene la lógica de negocio para validar y procesar las secuencias de ADN.
- **StatsController**: Controlador responsable de manejar las solicitudes para obtener estadísticas sobre los ADN procesados.
- **StatsService**: Servicio que gestiona la lógica para calcular las estadísticas de ADN mutantes y no mutantes.
- **AdnRepository**: Repositorio que interactúa con la base de datos para almacenar y recuperar información sobre las secuencias de ADN.
- **H2Database**: Base de datos en memoria utilizada para almacenar los datos de ADN.

## Flujo de Mensajes

### Validación de Secuencia de ADN:

1. El **Cliente** envía una solicitud **POST** a **AdnController** con un cuerpo que contiene la secuencia de ADN en formato JSON.
2. **AdnController** invoca el método `isMutant(dna)` de **AdnService** para verificar si la secuencia es mutante.
3. **AdnService** realiza la validación de la secuencia y, además de devolver un valor booleano que indica si la secuencia es mutante o no, se comunica con **AdnRepository** para guardar la secuencia de ADN (dna) y el resultado (resultado) en la base de datos.
4. **AdnRepository** ejecuta la operación de inserción en la base de datos, utilizando el comando:
  - `INSERT INTO ADN (dna, resultado).`
5. **H2Database** confirma la operación de guardado y devuelve la confirmación a **AdnRepository**.
6. **AdnService** devuelve el valor booleano a **AdnController**.
7. **AdnController** responde al **Cliente** con:
  - Un código **HTTP 200 (OK)** y el mensaje "Mutante" si la secuencia es válida y mutante.
  - O un código **HTTP 403 (Forbidden)** y el mensaje "No mutante" si la secuencia no cumple con los criterios de mutación.

### Obtención de Estadísticas:

1. El **Cliente** realiza una solicitud **GET** a **StatsController** para obtener estadísticas sobre las secuencias de ADN procesadas.
2. **StatsController** llama al método `obtenerEstadisticas()` de **StatsService** para recuperar los datos estadísticos.
3. **StatsService** utiliza **AdnRepository** para contar la cantidad de secuencias de ADN mutantes y no mutantes, realizando consultas a **H2Database**.
4. **AdnRepository** devuelve los resultados de las consultas a **StatsService**, que luego prepara un objeto **StatsDto** con las estadísticas.

5. Finalmente, **StatsController** responde al **Cliente** con un código **HTTP 200 (OK)** y el objeto **StatsDto** que contiene las estadísticas.