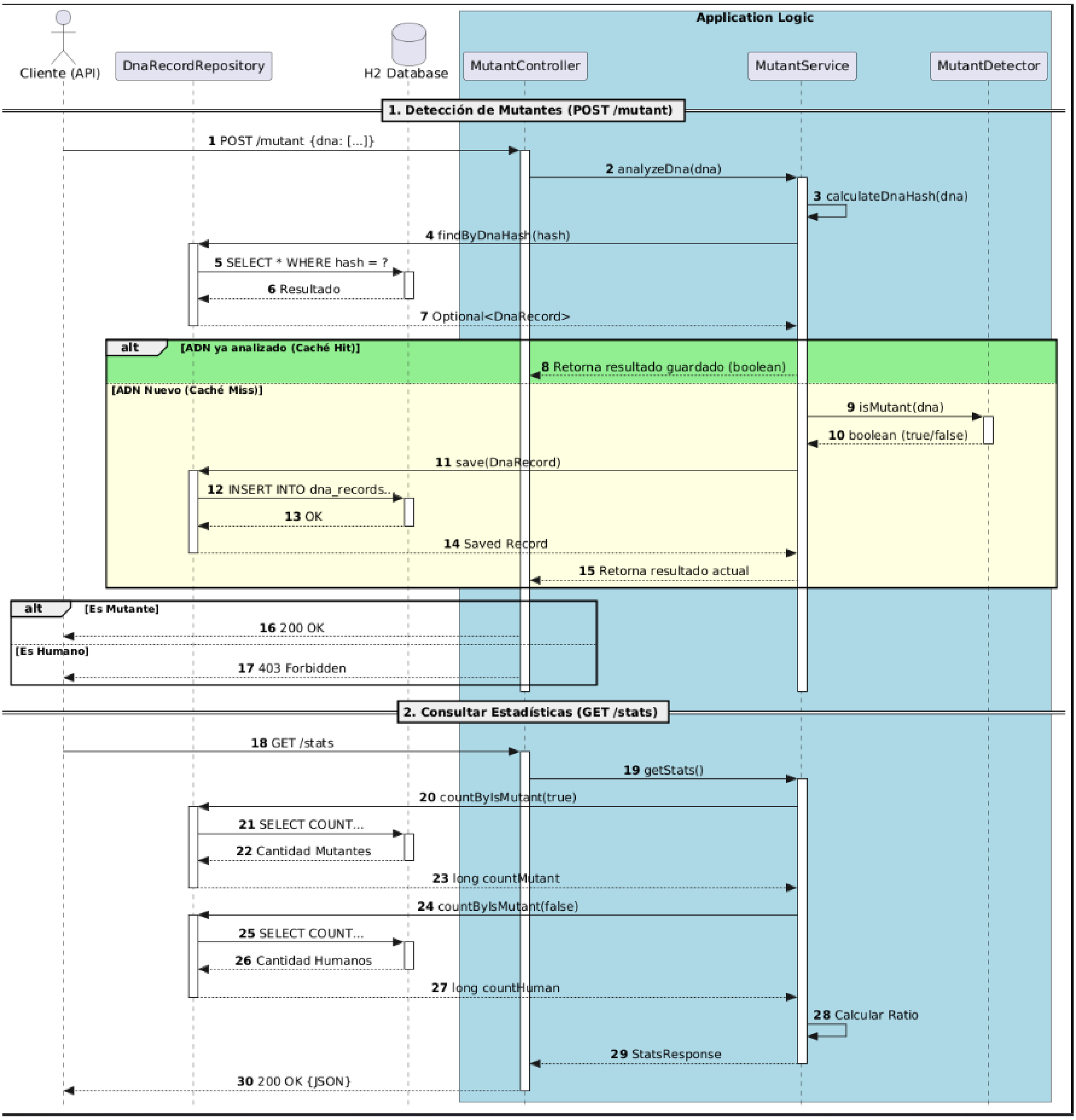
Este diagrama corresponde al diagrama de secuencia de la aplicación de detector de mutantes entregable para el proyecto integrador de la materia.

A continuación de detallarán tanto el diagrama como el código en PlantUML por si desea verlo mas detalladamente y no en una imagen estática.

Diagrama en PNG:



Código del diagrama en la herramienta PlantUML Editor:

```
@startuml
autonumber
skinparam style strictuml
skinparam sequenceMessageAlign center

actor Cliente as "Cliente (API)"
participant Controller as "MutantController"
participant Service as "MutantService"
participant Detector as "MutantDetector"
participant Repo as "DnaRecordRepository"
database DB as "H2 Database"

box "Application Logic" #LightBlue
  participant Controller
  participant Service
  participant Detector
end box

== 1. Detección de Mutantes (POST /mutant) ==

Cliente -> Controller: POST /mutant {dna: [...]}
activate Controller

Controller -> Service: analyzeDna(dna)
activate Service

Service -> Service: calculateDnaHash(dna)
Service -> Repo: findByDnaHash(hash)
activate Repo
Repo -> DB: SELECT * WHERE hash = ?
activate DB
DB --> Repo: Resultado
deactivate DB
Repo --> Service: Optional<DnaRecord>
deactivate Repo

alt #LightGreen ADN ya analizado (Caché Hit)
  Service --> Controller: Retorna resultado guardado (boolean)
else #LightYellow ADN Nuevo (Caché Miss)
  Service -> Detector: isMutant(dna)
  activate Detector
  Detector --> Service: boolean (true/false)
  deactivate Detector

  Service -> Repo: save(DnaRecord)
  activate Repo
  Repo -> DB: INSERT INTO dna_records...
  activate DB
  DB --> Repo: OK
  deactivate DB
  Repo --> Service: Saved Record
  deactivate Repo

  Service --> Controller: Retorna resultado actual
end
```

```
alt Es Mutante
   Controller --> Cliente: 200 OK
else Es Humano
   Controller --> Cliente: 403 Forbidden
end

deactivate Service
deactivate Controller

== 2. Consultar Estadísticas (GET /stats) ==

Cliente -> Controller: GET /stats
activate Controller

Controller -> Service: getStats()
activate Service

Service -> Repo: countByIsMutant(true)
activate Repo
Repo -> DB: SELECT COUNT...
activate DB
DB --> Repo: Cantidad Mutantes
deactivate DB
Repo --> Service: long countMutant
deactivate Repo

Service -> Repo: countByIsMutant(false)
```

```
activate Repo
Repo -> DB: SELECT COUNT...
activate DB
DB --> Repo: Cantidad Humanos
deactivate DB
Repo --> Service: long countHuman
deactivate Repo

Service -> Service: Calcular Ratio

Service --> Controller: StatsResponse
deactivate Service

Controller --> Cliente: 200 OK {JSON}
deactivate Controller

@enduml
```