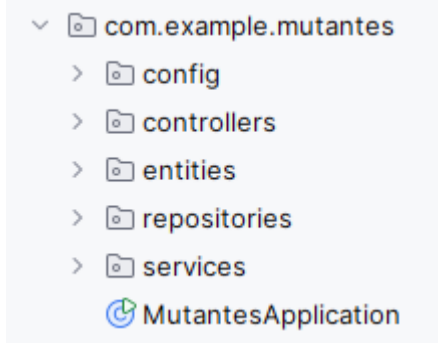


Examen BackEnd

Luciana Ortiz - 3k10

Nivel 1: Arquitectura del proyecto



- **config**: se encuentra la clase encargada de la auditoria
- **controllers**: controladores BaseController de la cual hereda MutantController
- **entities**: entidades Base de la cual hereda la entidad Mutant, entidad Revision para la auditoría
- **repositories**: repositorios BaseRepository y MutantRepository
- **services**: interfaces e implementaciones de Base y Mutant

Nivel 2: API en Render

Despliegamiento en [Swagger](#):

- **Método GET api/v1/mutant**: obtiene todos los adn ingresados con la estructura
{
 "id": id autogenerado autoincremental,
 "dna": array del adn ingresado,
 "mutant": booleano que devuelve false si es humano y true si es mutante
}
- **Método POST api/v1/mutant**: obtiene el adn por body y devuelve el objeto creado en la base de datos (id, dna, mutant)
- **Método GET api/v1/stats**: devuelve la cantidad de humanos, la cantidad de humanos y la relación entre mutantes y humanos

Nivel 3: Documentación

- **Jmeter:** Luego de realizar pruebas en el método GET y POST con distintos escenarios cambiando la cantidad de hilos podemos concluir que la capacidad máxima de usuarios a la vez que soporta el método GET es de ± 100 , en cuanto al método POST se ejecuta sin problemas. (el método post POST falla el 100% ya que la aplicación devuelve un error si el adn ya se ingresó anteriormente).

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
Petición HTTP GET	100	4247	663	6958	1468,64	2,00%	14,1/sec	8,86	2,35	643,7
Petición HTTP POST	100	521	352	2852	347,00	100,00%	11,0/sec	5,12	2,59	476,0

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
Petición HTTP GET	200	15138	9268	17834	2017,85	54,50%	11,2/sec	21,90	0,87	1999,7
Petición HTTP POST	200	16073	351	30200	14176,42	100,00%	5,2/sec	9,83	0,55	1947,0

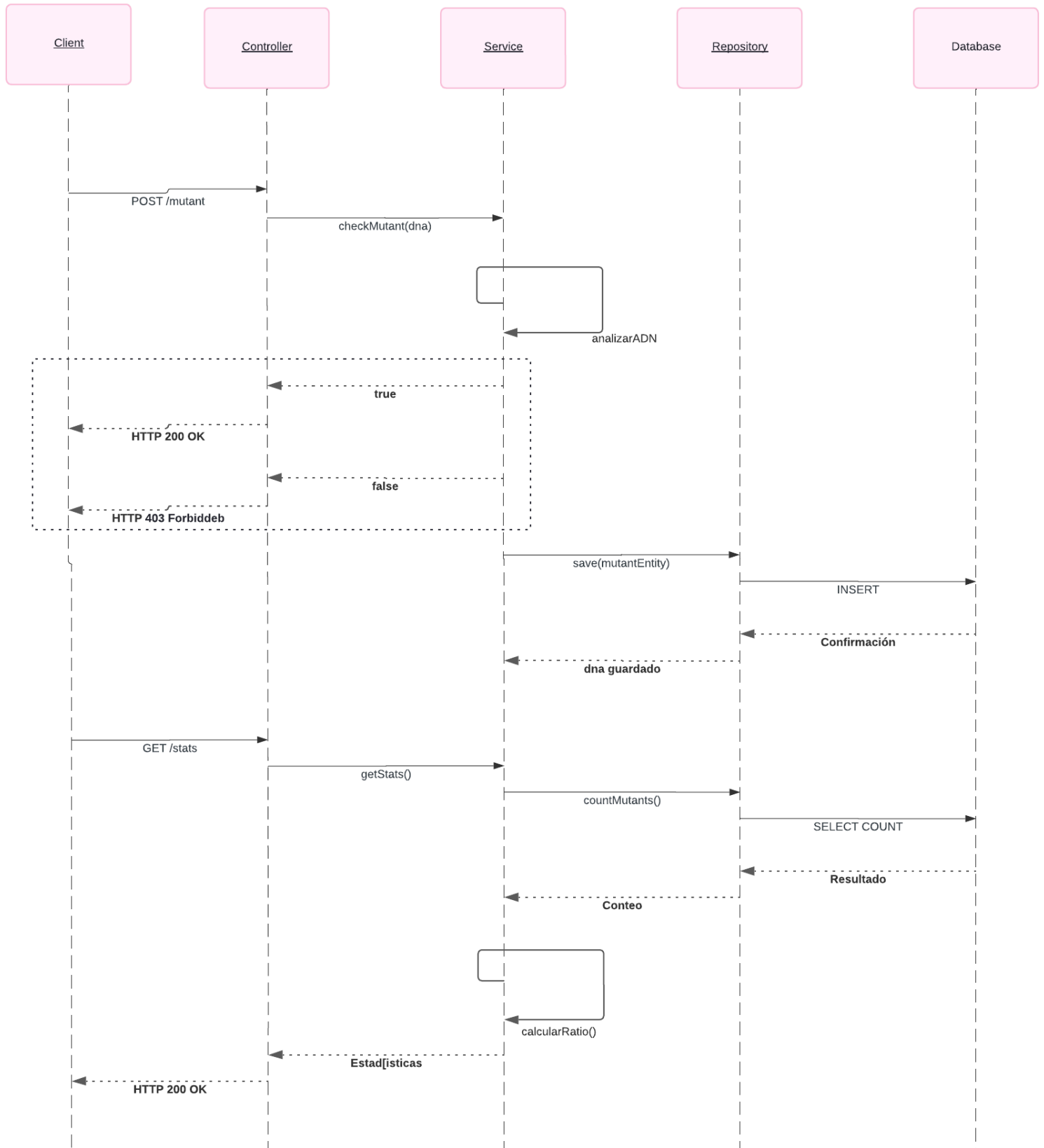
- **Code Coverage:** Luego de realizar el test unitario mediante JUnit logramos un promedio de 93% de coverage en todo el código.

Coverage Tests in 'mutantes.test' x				
Element ^				
	Class, %	Method, %	Line, %	Branch, %
com.example.mutantes	85% (6/7)	94% (17/18)	96% (77/80)	94% (36/38)
config	100% (1/1)	100% (1/1)	100% (1/1)	100% (0/0)
CustomRevisionListener	100% (1/1)	100% (1/1)	100% (1/1)	100% (0/0)
controllers	100% (2/2)	100% (9/9)	96% (61/63)	100% (36/36)
BaseController	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
BaseControllerImpl	100% (1/1)	100% (8/8)	96% (48/50)	100% (34/34)
MutantController	100% (1/1)	100% (1/1)	100% (13/13)	100% (2/2)
entities	0% (0/1)	0% (0/1)	0% (0/1)	0% (0/2)
audit	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
Base	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
Mutant	0% (0/1)	0% (0/1)	0% (0/1)	0% (0/2)
repositories	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
BaseRepository	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
MutantRepository	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
services	100% (2/2)	100% (6/6)	100% (14/14)	100% (0/0)
BaseService	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
BaseServiceImpl	100% (1/1)	100% (4/4)	100% (11/11)	100% (0/0)
MutantService	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
MutantServiceImpl	100% (1/1)	100% (2/2)	100% (3/3)	100% (0/0)
MutantesApplication	100% (1/1)	100% (1/1)	100% (1/1)	100% (0/0)

Overall Coverage Summary

Package	Class, %	Method, %	Branch, %	Line, %
all classes	85,7% (6/7)	95,5% (21/22)	94,7% (36/38)	96,4% (81/84)

- Diagrama de Secuencia



El diagrama muestra dos flujos principales:

- Detección de mutantes:
 - El cliente envía una solicitud POST con la secuencia de ADN
 - El controlador pasa los datos al servicio para su análisis
 - El servicio determina si es mutante o no
 - Dependiendo del resultado devuelve un código HTTP 200 o 403
 - Los resultados se guardan en la base de datos
- Obtención de estadísticas:
 - El cliente solicita mediante el método GET
 - El controlados pide las estadísticas al servicio
 - El servicio obtiene los conteos de repositorio, que a su vez consulta la base de datos
 - Se calculan las estadísticas y se devuelven al cliente