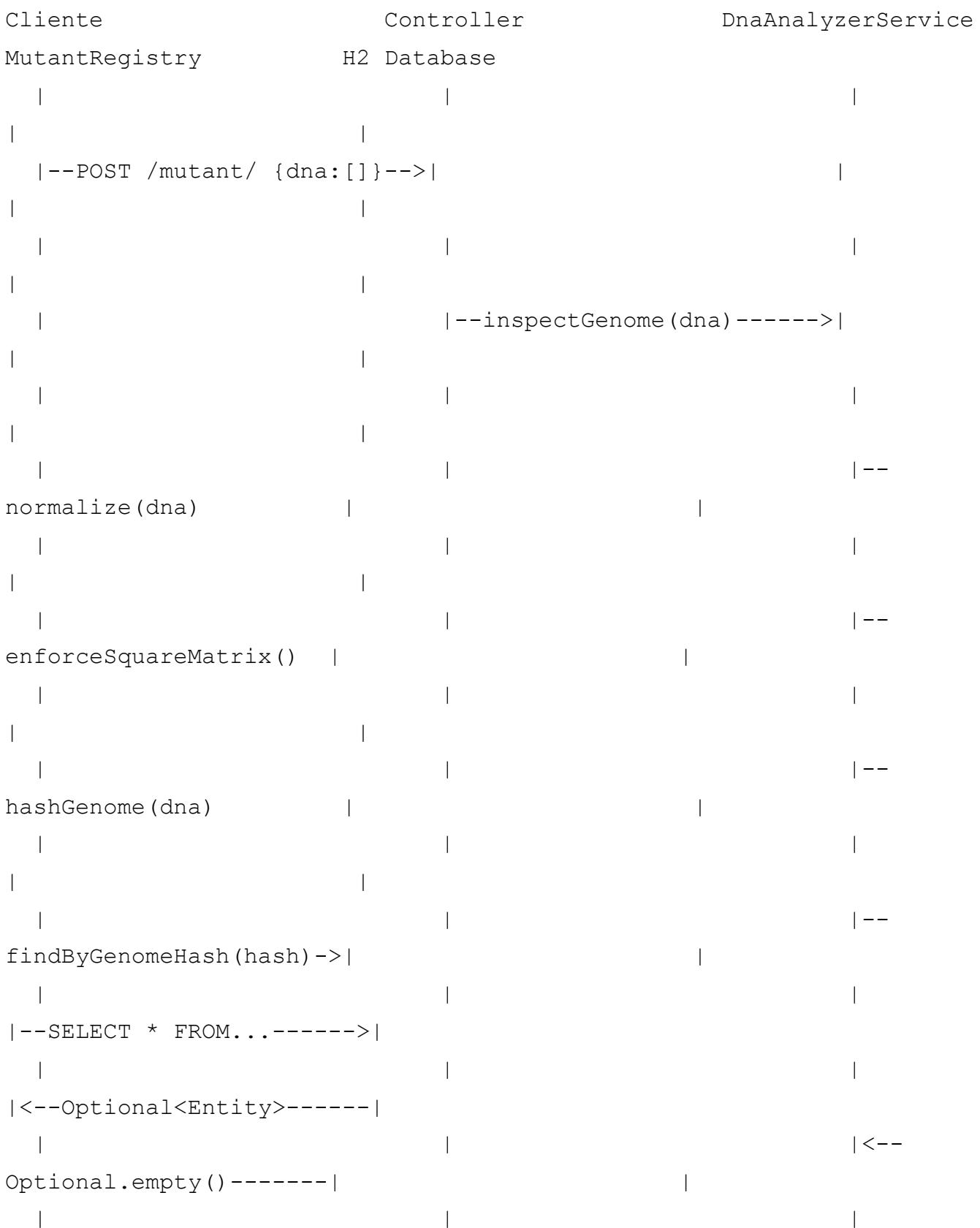


Diagrama de Secuencia - Mutant Detector API

Flujo: POST /mutant/



```

|           |           |
|           |           |           | --
buildMatrix(dna)   |           |           |
|           |           |
|           |           |           | --
detectMutations(matrix) |           |           |
|           |           |
|           |           |           |   | --
|           |           |           |   |   | --
exploreRows()      |           |           |
|           |           |
|           |           |           |   |   | --
exploreColumns()    |           |           |
|           |           |
|           |           |           |   |   | --
explorePrimaryDiagonals() |           |
|           |           |
|           |           |           |   |   | --
exploreSecondaryDiagonals() |           |
|           |           |
|           |           |           |   |   | --
|           |           |           | <--true
(mutant)-----|           |
|           |           |
|           |           |           | --
persistResult()     |           |           |
|           |           |
|           |           |           | --
save(entity)----->|           |
|           |           |
| --INSERT INTO...----->|
|           |           |
| <--Entity saved-----|           |
|           |           |           | <--true-----
-----|           |
|           |           | <--true (mutant)-----|
|           |           |
| <--200 OK-----|           |
|           |

```

Flujo: GET /stats

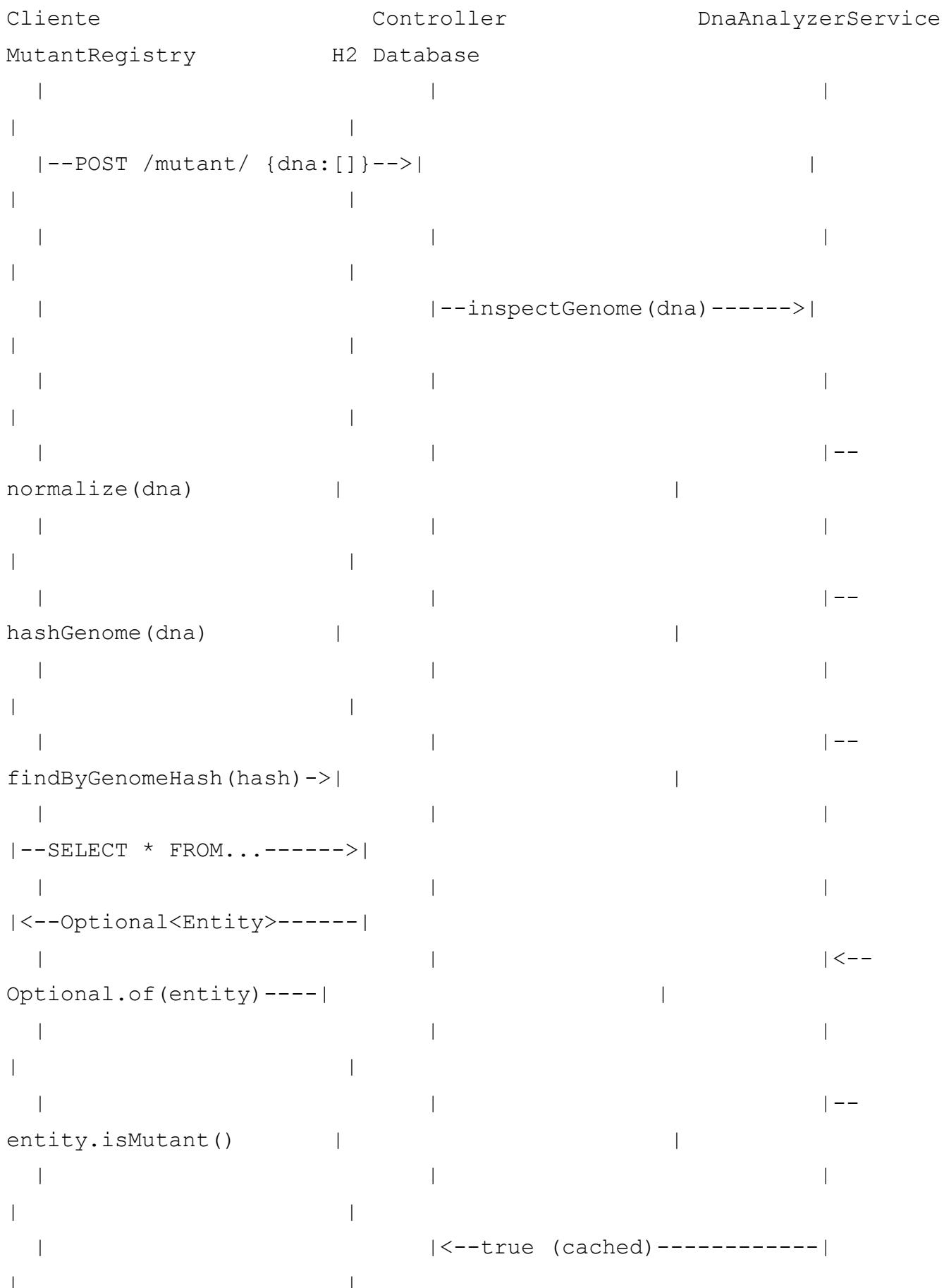
Sequence diagram illustrating the flow from Cliente to Controller to DnaMetricsService:

```
sequenceDiagram
    participant Cliente
    participant Controller
    participant DnaMetricsService

    Cliente->>Controller: MutantRegistry
    activate Controller
    Controller-->>DnaMetricsService: H2 Database
    activate DnaMetricsService
    Note over Controller: |--GET /stats----->
    Controller-->>DnaMetricsService: |--fetchStats ()----->
    activate DnaMetricsService
    DnaMetricsService-->>Controller: countByMutantTrue ()----->
    activate Controller
    Controller-->>DnaMetricsService: |--SELECT COUNT(*) ...---->
    activate DnaMetricsService
    DnaMetricsService-->>Controller: |<--40-----|
    activate Controller
    Controller-->>DnaMetricsService: |<--40-----|
    activate DnaMetricsService
    Note over Controller: |--countByMutantFalse ()--->
    Controller-->>DnaMetricsService: |--SELECT COUNT(*) ...---->
    activate DnaMetricsService
    DnaMetricsService-->>Controller: |<--100-----|
    activate Controller
    Controller-->>DnaMetricsService: |<--100-----|
    activate DnaMetricsService
    Note over Controller: |--calculate ratio (40/100) |
    Controller-->>DnaMetricsService: |--new StatsPayload(40, 100, 0.4) |
    activate DnaMetricsService
    DnaMetricsService-->>Controller: |<--StatsPayload-----|
```

```
|<--200 OK {count_mutant_dna:40, count_human_dna:100, ratio:0.4}--|
```

Flujo: POST /mutant/ (con caché)



| <--200 OK-----|

| |