

# Experimento 2 entrega final

Grupo **St. ValenTeam**

## Arquitectura

Se tomaron consideraciones de diseño para mejorar los atributos de calidad de disponibilidad y seguridad e integridad de los datos.

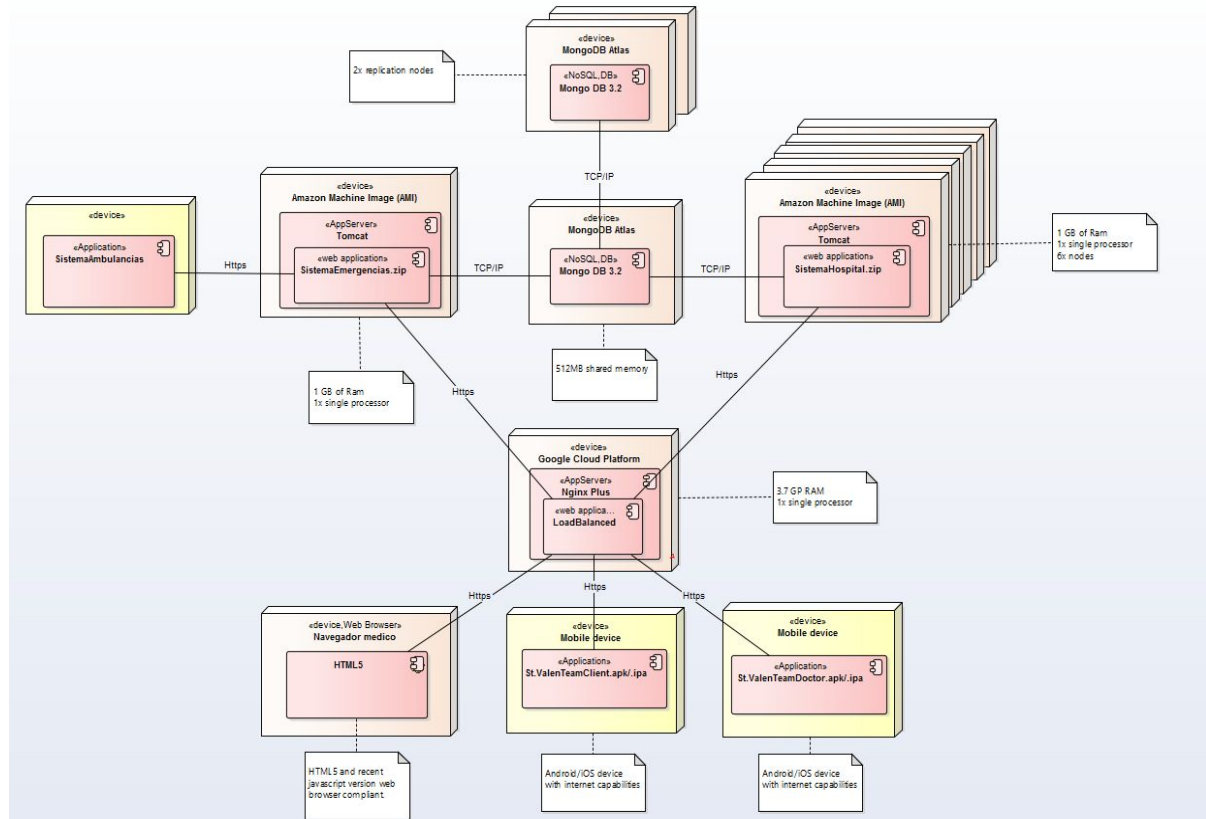
Los patrones incluídos a la arquitectura para satisfacer el atributo de calidad de disponibilidad fueron los siguientes:

- Se mejoró la infraestructura de la aplicación  
La infraestructura de Heroku tiene una capacidad mucho mayor de cómputo que la infraestructura gratuita de AWS que se estaba utilizando.

Las consideraciones incluidas a la arquitectura para satisfacer el atributo de calidad de seguridad fueron las siguientes:

- Se utiliza una función de hash para garantizar la integridad de los datos de los sensores
- Se realizan solicitudes **Https** para garantizar confidencialidad e integridad en la comunicación web
- Autenticación de usuarios mediante un servicio login y password
- Autorización de usuarios mediante tokens
- Capa de seguridad en Node Red

A continuación se presenta el **diagrama de despliegue**:



## Pruebas de carga

Se volvieron a ejecutar las pruebas de carga propuestas en el experimento 1 y repetidas en la primera entrega del experimento. A continuación se pueden ver los resultados y una clara de explicación del deterioro en los resultados:

- **Peticion tipo POST:**

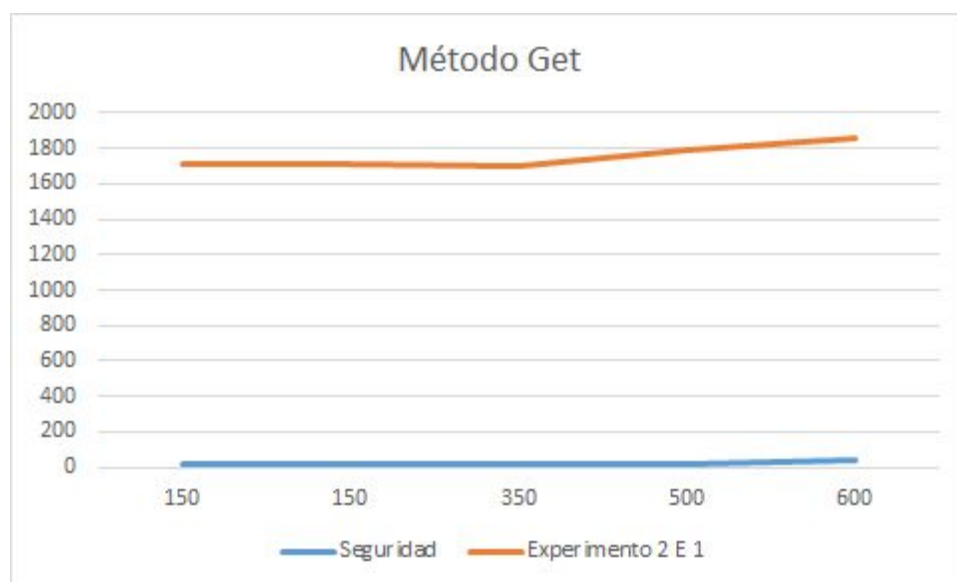
### Procesar medición (petición POST) - Seguridad

#Samples	Average(ms)	Min(ms)	Max(ms)	Error (%)	Throughput
9	18	7	77	0	10
17	18	7	189	0	19
167	18	6	291	0	168
333.966667	17	9	1206	0	336
666.983333	37	10	3157	0	671
926.466667	337	5	20063	0.5	931

### Analisis:

Los datos arrojados por las pruebas de loader.io fueron muy favorables en comparación a la entrega pasada. Esto se debe principalmente al cambio que se hizo de infraestructura, de AWS a Heroku. Por otro lado, los tiempos son mayores en esta entrega que en la primera entrega, donde también se utilizó Heroku, pero es de esperarse que la capa de seguridad aumente la latencia en un valor más o menos fijo.

- **Petición tipo GET:**



### Petición historia médica - Seguridad

#Samples	Average(ms)	Min(ms)	Max(ms)	Error(%)	Throughput
540	15	10	78	0%	18
1020	14	10	253	0%	24
10020	12	12	120	0%	215
20038	18	13	1899	0%	357
40019	16	15	1500	0%	795
55588	19	13	985	0%	1100
90000	40	10	1899	0%	1820
120000	350	16	9956	3,1%	2113

### Analisis:

Se evidencia que el tiempo en el que se responde una petición Get que requiere autenticación con respecto al tiempo que tomaba en la primera entrega del experimento 2 es mucho menor, esto no es lo esperado ya que debería aumentar el tiempo de respuesta. Sin embargo, esto se debe a que se decidió volver a utilizar Heroku ya que la disponibilidad y desempeño de Heroku es mucho mayor. Sin embargo, al comparar con la entrega 1 es claro que el tiempo promedio de respuesta es mayor.