**Informe experimento 2 Solumovil**

**Pre experimentación**

**Problemática:**

La ubicación y el seguimiento de un gran número de vehículos es un aspecto de vital importancia para la empresa de transporte Tbc. Por esto Tbc ha decidido contratar a SoluMovil para diseñar y construir una aplicación que sea capaz de responder ante un gran número de solicitudes simultáneas de sus vehículos (Mobibus, vcub y tranvía eléctrico) así como una respuesta rápida ante estas. Adicionalmente la aplicación construida debe atender todas las solicitudes que se le hagan asegurando que el 100% de la información trasferida por alguno de los vehículos logueados en el sistema.

**Objetivo del experimento:**

Aceptar, procesar y responder a las solicitudes de múltiples usuarios simultáneos en la aplicación SoluMovil con el fin de satisfacer las necesidades de la empresa Tbc. Específicamente, responder al 100% de las solicitudes enviadas por los vehículos manteniendo los atributos de desempeño y escalabilidad.

Descripción del experimento:

Para cumplir con los requerimientos del cliente se siguieron los siguientes pasos:

1. Investigación de métodos de logueo e investigación de formas de cifrado de información. Selección de métodos
2. Implementación de los métodos seleccionados
3. Pruebas de carga
4. Corregimiento de métodos hasta alcanzar los resultados deseados.

**Artefactos a construir.**

Módulo de balanceo de carga en la aplicación. Aplicaciones móviles completamente terminadas. Servicio REST.

**Recursos de Experimentación**

Seleccionar los recursos no fue una tarea sencilla. Esto ya que se debía respetar los elementos de la arquitectura que proporcionaban la escalabilidad, desempeño y disponibilidad a la vez que se cumplía la confidencialidad y la integridad.

**Hardware.**

Se utilizó un PC con sistema operativo Windows 10 y RAM de 4GB como servidor de la aplicación. Este fue el encargado de redirigir las peticiones a través de nginx.

Se utilizaron como nodos del servidor dos MAC con RAM de 4GB utilizando gUnicor como servidor interno para desplegar Django.

**Resultados esperados.**

Se espera que cualquier petición al servicio REST sea resuelta en menos de 1 s además de que el 100% de las peticiones tengan respuesta y que 4500 peticiones tengan un tiempo de respuesta medio menor a 1s. Así como que estas peticiones estén cifradas y solo sean procesadas para usuarios logueados.

**Duración y etapas:**

AL hacer el planeamiento se tienen 3 etapas, planeación, construcción y pruebas. EL punto crítico se encuentra en completar los servicios REST de las aplicaciones móviles así como del servidor puesto que son las que más trabajo representan además que deben tener uniformidad.

|  |  |  |
| --- | --- | --- |
|  | | Tiempo estimado (Horas) |
| Planeación | Investigación técnicas de encriptamineto | 2 |
| Asignación de tareas | 1 |
| Planeación | 2 |
| Desarrollo | Mobibus Stand Alone | 1 |
| Vcub Stand Alone | 1 |
| Tren Eléctrico Stand Alone | 1 |
| Logueo usuarios | 10 |
| Encriptación datos | 10 |
| Bono | 10 |
| Pruebas | Pruebas | 5 |
| Corregimiento | 5 |
| Pruebas | 2 |
| Otros | Instalación software | 10 |
| Desarrollo reporte | 4 |
|  | Total Estimado horas | 64 |

**Post experimentación**

Resultados:

Stormpath es uno de los servicios de logueo y autenticación más populares y fáciles de utilizar para desarrolladores. Además su acoplamiento con Django lo hace aún mejor para este proyecto. Esto ya que además de poder utilizar el servicio de autenticación predefinido en Django así como el proveído por Stormpath.

Seguridad:

Ya que la aplicación se maneja con HTTPS el cual sin estado se debía encontrar una forma de almacenar información de la sesión actual del usuario sin correr riesgos. La solución encontrad fue el uso de tokens esto, ya que su uso permite delegar la seguridad permitiendo no solo la escalabilidad de la aplicación sino que además seguridad extra puesto que un profesional se encarga de la seguridad de la aplicación. Esto, puesto que la información de autenticación no queda guardad en el servidor por lo cual posibles ataques para robar la información no son una preocupación. Esto, además permite identificar claramente que usuarios de la aplicación hacen peticiones puesto que para hacerlas deben tener el token de seguirdad por lo cual ataques CSFR (Cross-Site Request Forgery) [1]no son un problema puesto que no existe una sesión a la cual atacar. Adicionalmente, ducho token expirara en algún tiempo por lo cual un usuario deberá hacer log in con lo cual hay más seguridad. [2]

Para implementar este sistema de logueo en base de token se utiizo “Stormpath for OAuth 2.0 and Access/Refresh Token Management”. Se escogió este sistema puesto que permite utilizar tokens tanto de autenticación como de refrescamiento así como manejar la duración de cada uno de estos [3].

Cómo ya se dijo anteriormente el uso de este sistema basado en tokens permite que los ataques realizados a la información sean inútiles, por lo cual se asegura que el 100% de los datos trasmitidos tras el logueo de un usuario no han sido alterados.

Comparación resultados con y sin balanceo de carga.

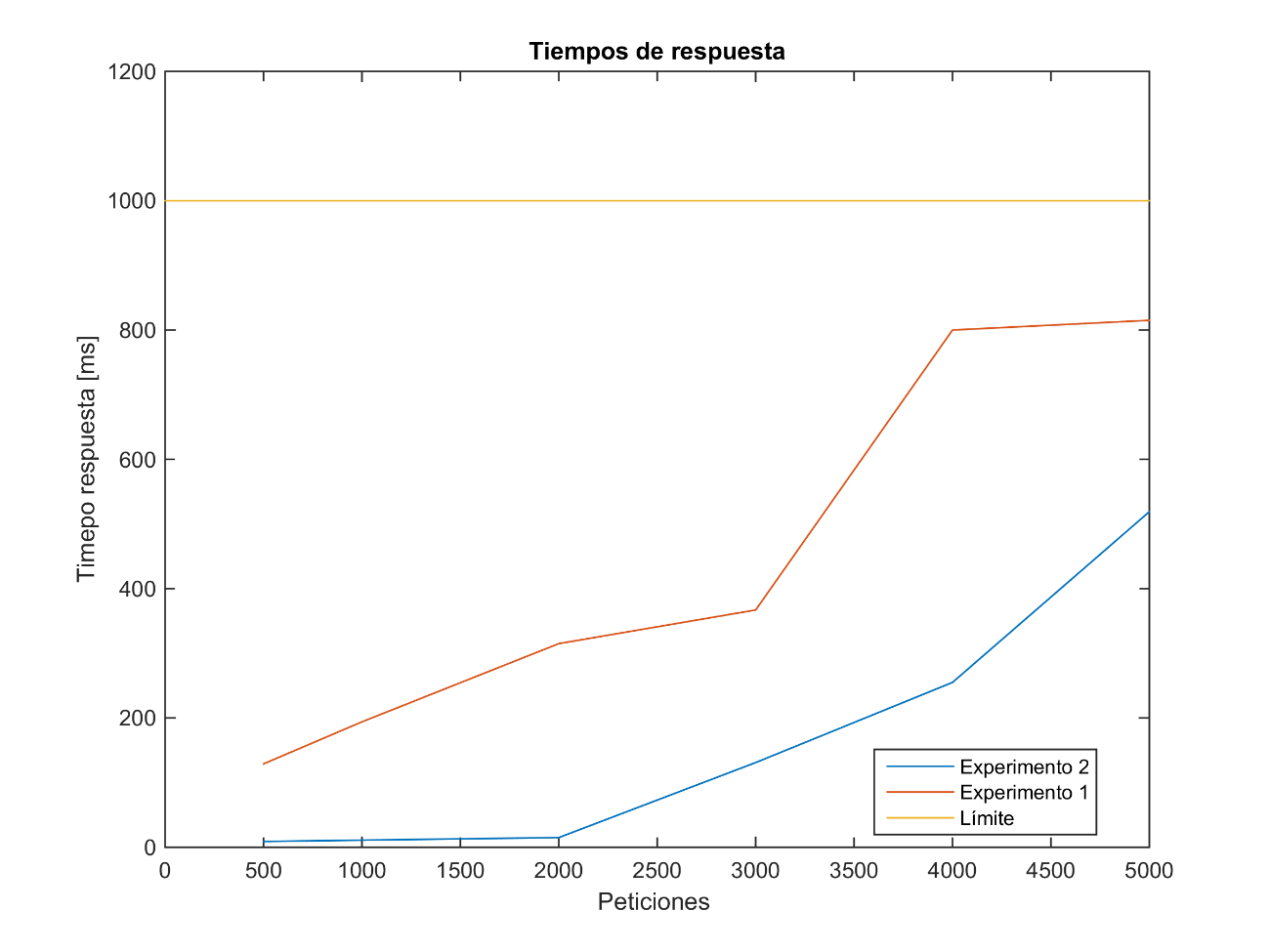


Ilustración 1 Comparación tiempos de respuesta método GET entrega 1 vs entrega 2

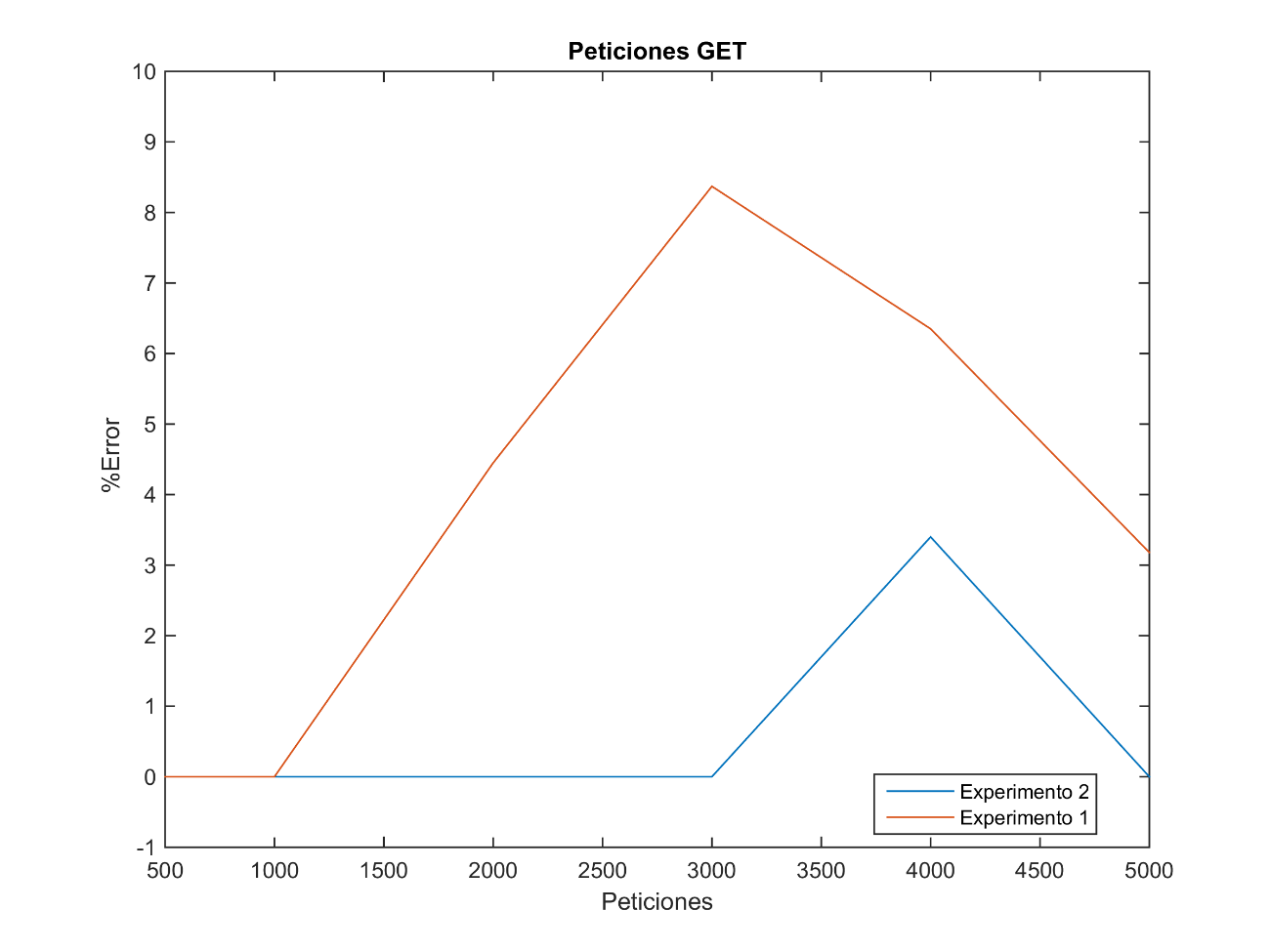


Ilustración 2 Comparación % error obtenido entrega 1 vs entrega 2 método GET

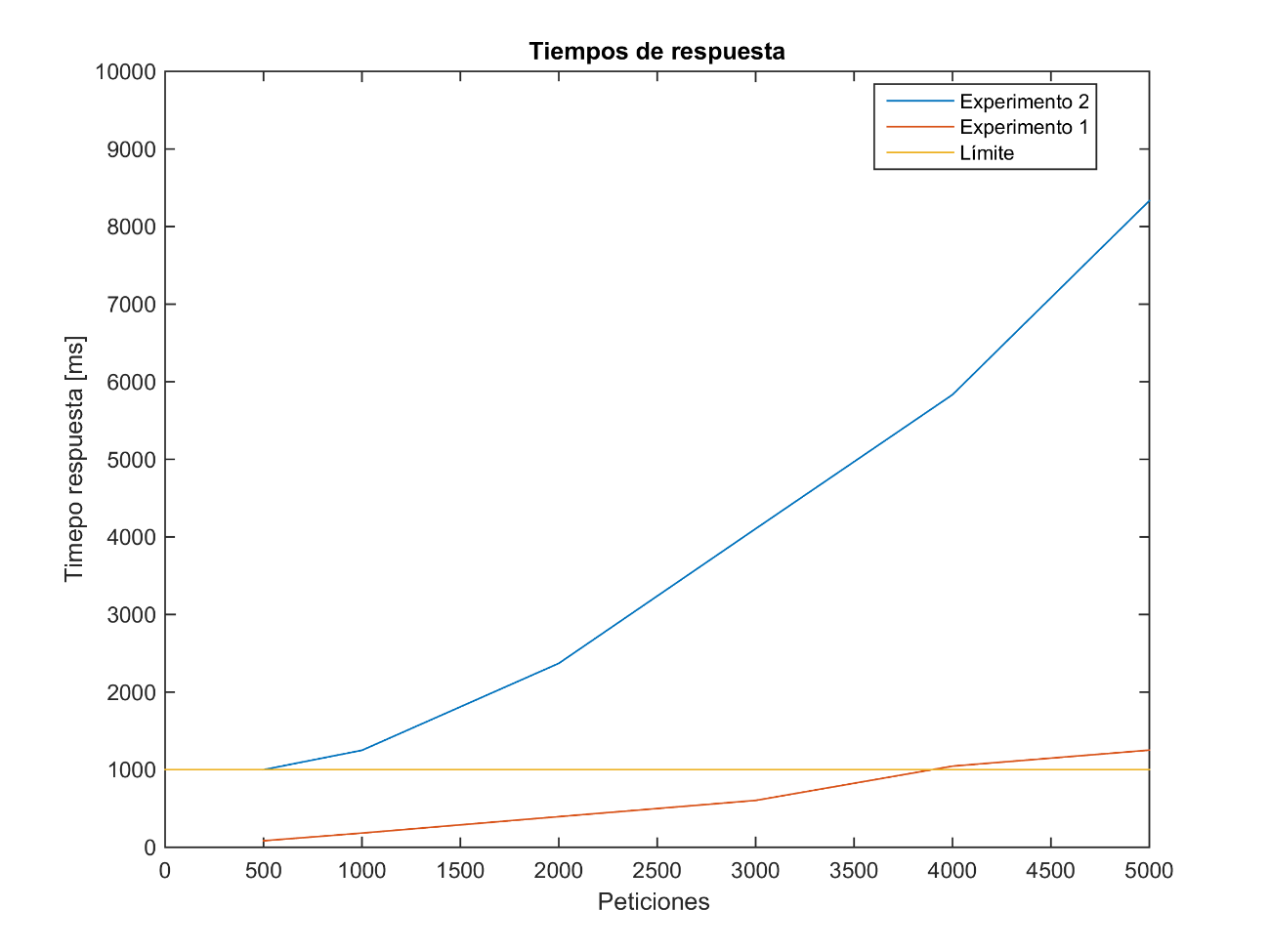


Ilustración 3 Comparación tiempos de respuesta método POST entrega 1 vs entrega 2

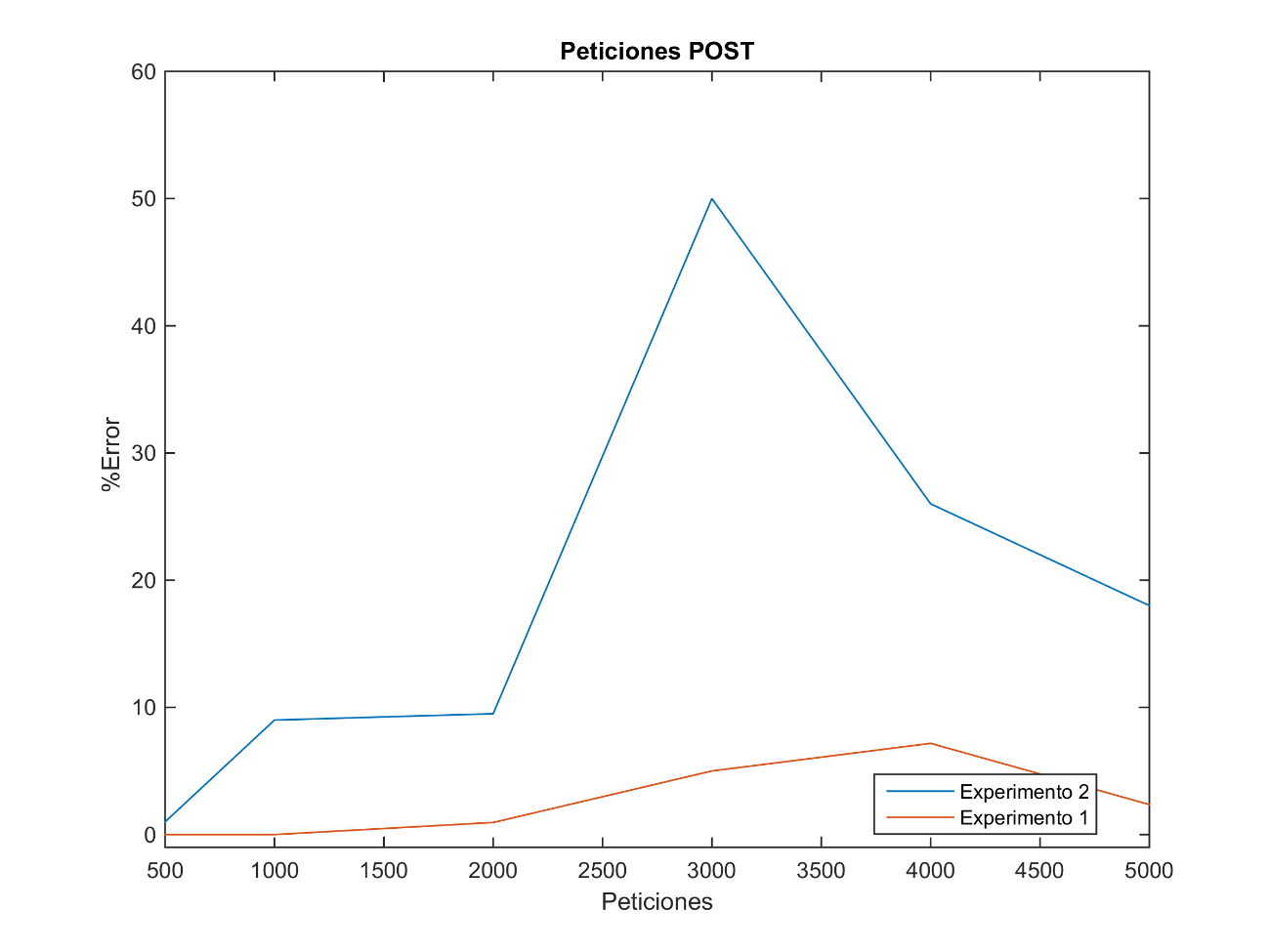


Ilustración 4 Comparación % error obtenido entrega 1 vs entrega 2 método POST

Tiempos reales

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Tiempo estimado (Horas) | Jairo | Juan | Meili | Simón |
| Planeación | Investigación técnicas de encriptamineto | 2 | 2 | 2 | 0 | 2 |
| Asignación de tareas | 1 | 2 | 1 | 1 | 0 |
| Planeación | 2 | 1 | 0 | 2 | 2 |
| Desarrollo | Mobibus Stand Alone | 1 | 0 | 1 | 1 | 0 |
| Vcub Stand Alone | 1 | 2 | 1 | 1 | 0 |
| Tren Eléctrico Stand Alone | 1 | 2 | 0 | 1 | 0 |
| Logueo usuarios | 10 | 1 | 3 | 10 | 4 |
| Encriptación datos | 10 | 1 | 0 | 9 | 4 |
| Bono | 10 | 0 | 1 | 2 | 4 |
| Pruebas | Pruebas | 5 | 2 | 1 | 4 | 5 |
| Corregimiento | 5 | 0 | 5 | 3 | 2 |
| Pruebas | 2 | 2 | 0 | 1 | 0 |
| Otros | Instalación software | 10 | 2 | 9 | 10 | 9 |
| Desarrollo reporte | 4 | 2 | 1 | 2 | 0 |
|  | Total Estimado | 64 | 19 | 13 | 35 | 52 |
|  | Real | 119 |  |  |  |  |
|  | Diferencia | 86% |  |  |  |  |

Artefactos construidos

3 aplicaciones móviles: Mobibus, Vcub y tranvía eléctrico.

Servidor con base de datos en postgres SQL local. Todos los métodos deseados en la entrega 1. Archivo config de nginx que permite hacer el balance de carga a través del puerto 90. Además conexión de la aplicación con strompath para el logueo de usuarios y la protección de datos. Así como la autorización de usuarios por dos medios, Strompath y Django.

Análisis

Conclusiones

El trabajo en equipo de esta entrega fue mejor en comparación a las pasadas. Además, hubo comunicación así como distribución de tareas. También se aprendió sobre el tema de seguridad en cuanto a confidencialidad e integridad de datos.