



ENUNCIADO DE TRABAJOS DE LABORATORIO 2021

Docentes

Profesor Adjunto: **Lic. Cristian Javier Parise**

Auxiliar: **Lic. Pedro Konstantinoff**

Sistemas Distribuidos

INDICE

TRABAJOS DE LABORATORIO	2
Objetivo general de los Trabajos Prácticos de Laboratorio.	3
Características	3
Forma de aprobación	3
Trabajo de Laboratorio N° 2	4
Sincronización. Timestamping. Docker.	5

TRABAJOS DE LABORATORIO

Objetivo general de los Trabajos Prácticos de Laboratorio.

- Conocimiento y realización de trabajos sobre arquitectura Cliente / Servidor con diferentes tecnologías.
- Conocimiento de la problemática de Sincronización.
- Manejo de la generación de páginas WEB dinámicas.
- Manejo de Código Móvil y Sistemas Distribuidos de archivo.

Características

El Informe deberá reunir las siguientes características:

1. Cada alumno presentará su informe a efectos de su calificado por el profesor. Los trabajos que no reúnan los requisitos mínimos serán devueltos para su corrección.
2. Deberá ser presentado a la cátedra con **discusión individual** por alumno.
3. Para su preparación e impresión, el trabajo práctico deberá ser entregado de la siguiente forma:
 - Deberá identificar al alumno y contener la firma del mismo.
 - Se deberá trabajar sobre el enunciado de cada punto de la práctica, **conservando su enunciado previo a la solución.**
 - Toda la bibliografía utilizada deberá ser referenciada indicando título y autor, en una sección dedicada a tal efecto.
 - De existir algún código de programación, se deberá adjuntar el programa de aplicación que implementa la solución. El código fuente debe estar debidamente comentado. La solución debe ser desarrollada utilizando el lenguaje de programación indicado. Se debe incluir la demostración de prueba de la solución implementada.
 - El código fuente deberá estar correctamente comentado, adaptado para la prueba por parte del docente, y desarrollado en el lenguaje de programación indicado. También se debe incluir un README o instructivo para su correcto testeo.
 - El código fuente además deberá ser compartido mediante un repositorio tipo github.

Forma de aprobación

Se tendrá en cuenta para la aprobación del trabajo práctico y del alumno:

- Estructura general de la presentación, su legibilidad y facilidad de lectura y comprensión.
- Funcionamiento de la aplicación desarrollada. Se evaluará si la funcionalidad cumple con lo solicitado.
- Contenido del informe y el uso de la información técnica para elaborarlo.
- Evaluación del trabajo de laboratorio.

Sistemas Distribuidos

En caso de no verificarse algunas de las condiciones enunciadas, el trabajo práctico se considerará desaprobado.

Trabajo de Laboratorio N° 2

Sincronización. Timestamping. Docker & Servicios.

1. Implemente un servidor de hora empleando Python y llamadas a procedimientos remotos. Desarrollar un cliente que emule un reloj (utilizar hilos para la actualización del mismo) y que se sincronice dicho reloj con el servidor de hora implementado, utilizando el algoritmo de Cristian. Al utilizar la hora patrón, tome un grupo de valores (por ejemplo 5), eligiendo el que haya demandado menor tiempo de tránsito de los mensajes.
2. Desarrolle una aplicación que se comunice con un servidor NTP de internet y que extraiga 8 muestras del retardo con que correspondería ajustar la hora obtenida. Presentar en pantalla una tabla con cada muestra del tiempo obtenido del servidor en formato YYYY/MM/DD HH:MM:SS.mmm en relación con cada retardo. Utilizar la librería NTPLib de Python.

<https://pypi.org/project/ntplib/>

3. Timestamping
 - a) Obtener un sello de tiempo de un archivo de texto con contenido conocido, verificarlo y por último alterarlo e intentar volver a verificarlo. Documentar el proceso utilizando la GUI de dos infraestructuras:
 - i) **bfa.ar** (Nacional). Para el procedimiento deberán obtener un recibo digital temporario *provisorio* (extensión rd.temp) del hash del archivo, luego esperar a que el archivo sea ingresado en un bloque Blockchain y por último obtener el recibo digital temporario *definitivo* (extensión rd)
 - ii) **opentimestamps.org** (Externa). Para el procedimiento deberán obtener un recibo (extensión ots) y luego verificarlo pero se debe esperar a que el hash del archivo forma parte de un bloque Blockchain (mientras dará el mensaje Pending attestation)
 - b) Realizar la operación de sellado por línea de comando

```
pip install opentimestamps-client
ots stamp <file>
```
4. En base al código del repositorio
https://github.com/pkonstantinoff/unpsjb_distribuidos/tree/main/tl2/p4/
 - a) Levante un servicio web nginx y sirva una página que diga "HOLA MUNDO".
 - b) Genere volúmenes persistentes para el servicio del inciso 4.a
 - c) Levante un servicio de base de datos postgres con persistencia de la base mediante volúmenes.
 - d) Levante el servicio del punto 3.a del TL1 empleando docker-compose. Compruebe el funcionamiento del servicio, haciendo uso del mismo con el cliente del punto 3.a del TL1.
 - e) Levante el servicio del punto 3.b del TL1 empleando docker-compose. Compruebe el funcionamiento del servicio, haciendo uso del mismo con el cliente del punto 3.b del TL1.
 - f) Levante un servidor NTP y emplee el cliente del punto 2.