

TRABAJO PRÁCTICO N° 1

ASIGNATURA: TALLER DE PROYECTO 2.

INTEGRANTES:

- Becerra Agustín
- Buscaglia Joaquín
- Molina Rodrigo

Fecha de entrega: 14/09/2017

1) **Generar el archivo 'requirements.txt' con las dependencias necesarias para poder levantar un servidor con Flask. Explicar un ejemplo de uso con la secuencia de acciones y procesos involucrados desde el inicio de la interacción con un usuario hasta que el usuario recibe la respuesta.**

Requirements.txt:

click==6.7

Flask==0.12.2

itsdangerous==0.24

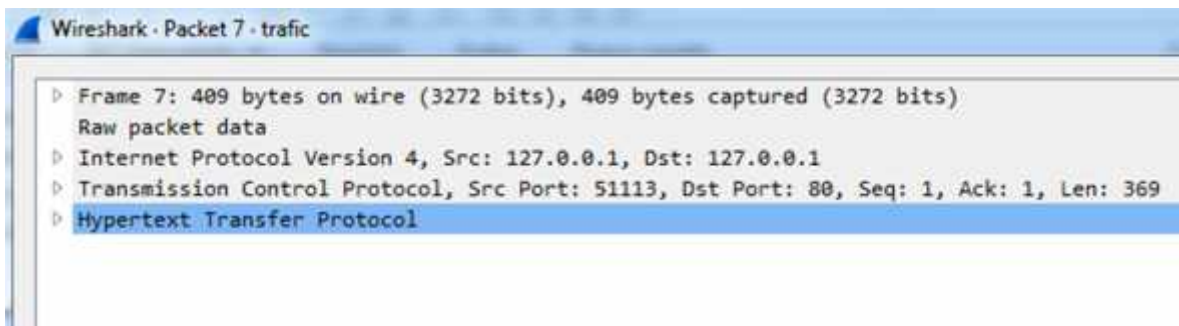
Jinja2==2.9.6

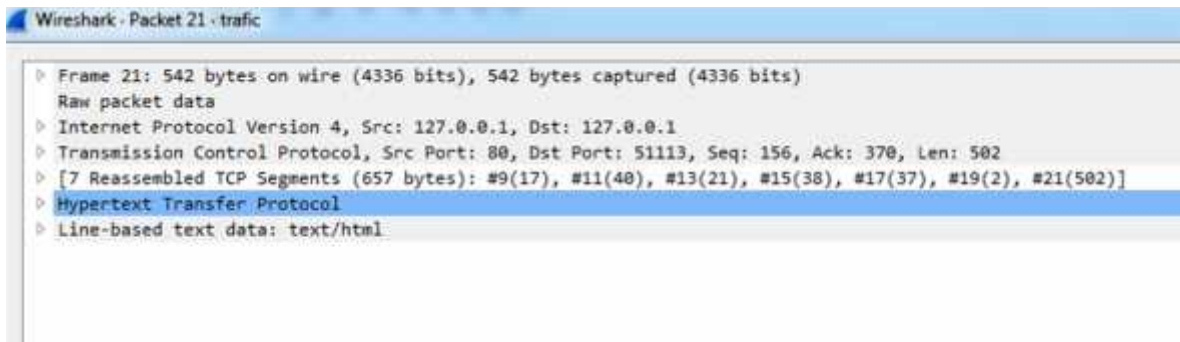
MarkupSafe==1.0

Werkzeug==0.12.2

El proceso de interacción entre el usuario y el servidor comienza cuando el usuario carga la página principal, ahí se manda una petición al servidor, el cual va al @app.route correspondiente al directorio que se trata de cargar, ejecuta el código correspondiente y luego devuelve al cliente el html a mostrar con los parámetros que se le especifiquen.

2) **Desarrollar un experimento que muestre si el servidor HTTP agrega o quita información a la genera un programa Python. Nota: debería programar o utilizar un programa Python para conocer exactamente lo que genera y debería mostrar la información que llega del lado del cliente, a nivel de HTTP o, al menos, a nivel de HTML (preferentemente HTTP).**





Se utilizó y ejecuto el programa hecho en python provisto por la cátedra, el cual envía un dato de tipo String. Se utilizó el programa RawCap para monitorear el tráfico de datos y mediante wireshark se identifican los paquetes HTTP involucrados en la comunicación. Como se puede observar en las imágenes ilustradas se le agrega información como:

- Protocolos de internet (fuente, destino, tamaño, etc).
- Protocolo de transmisión (puerto de destino y fuente, tamaño del headers).
- Texto plano de tipo HTML.

3) Generar un proyecto de simulación de acceso a valores de temperatura, humedad, presión atmosférica y velocidad del viento.

a) Un proceso simulará una placa con microcontrolador y sus correspondientes sensor/es o directamente una estación meteorológica proveyendo los valores almacenados en un archivo o en una base de datos. Los valores se generan periódicamente (frecuencia de muestreo).

b) Un proceso generará un documento HTML conteniendo:

- i) Frecuencia de muestreo
- ii) Promedio de las últimas 10 muestras
- iii) La última muestra

c) El documento HTML generado debe ser accesible y responsivo.

Aclaración: Se deberá detallar todo el proceso de adquisición de datos, cómo se ejecutan ambos procesos (ya sea threads o procesos separados), el esquema general, las decisiones tomadas en el desarrollo de cada proceso y la interacción del usuario.

Para llevar a cabo dicha simulación se crearon dos procesos, uno que se encarga de controlar el servidor y otro de generar datos y cargarlos en la base de datos implementada. Estos procesos trabajan a una frecuencia de 1 Hz.

La base de datos que se utiliza fue implementada con MySQL.

4) Agregar a la simulación anterior la posibilidad de que el usuario elija entre un conjunto predefinido de períodos de muestreo (ej: 1, 2, 5, 10, 30, 60 segundos). Identifique los cambios a nivel de HTML, de HTTP y de la simulación misma.

En cuanto al código del HTML lo que se agrego fue:

- Una página principal donde se puede seleccionar la frecuencia deseada y mandarla por parámetro al servidor flask.
- En la página donde se muestran los datos (humedad, velocidad del viento, temperatura y presión) se actualiza con el valor de frecuencia que el usuario haya seleccionado.

En cuanto a la simulación se toman los últimos 10 valores que corresponden a la frecuencia elegida.

5) Comente la problemática de la concurrencia de la simulación y específicamente al agregar la posibilidad de cambiar el período de muestreo. Comente lo que estima que podría suceder en el ambiente real ¿Podrían producirse problemas de concurrencia muy difíciles o imposibles de simular? Comente brevemente los posibles problemas de tiempo real que podrían producirse en general.

Los problemas de concurrencia que aparecen en nuestra simulación al cambiar el periodo de muestreo se debe a la no sincronización entre el productor y el consumidor a la hora de acceder a la base de datos. Por ejemplo, si el productor produce valores con un periodo menor al que son requeridos por el consumidor, este último consumirá valores repetidos en instantes de tiempo consecutivos. Y por otro lado, si el consumidor trabaja a menor frecuencia que el productor, habrá pérdida de datos en ciertos tiempos.

En un ambiente real podría pasar que los sensores no puedan tomar muestra de valores a la frecuencia que sean requeridas por el sistema real.

6) ¿Qué diferencias supone que habrá entre la simulación planteada y el sistema real? Es importante para planificar un conjunto de experimentos que sean significativos a la hora de incluir los elementos reales del sistema completo.

Para implementar esta simulación de muestreo de señales de sensores en un sistema real, habrá que programar el sistema de adquisición de datos para que en un instante de tiempo determinado pueda acceder a los sensores y reflejar dicho valor. Por lo tanto, en este caso, no habrá problemas de concurrencia de datos, por no estar almacenando los valores leídos en una base de datos como se realizó en nuestra simulación.