

Universidad Nacional de La Plata
Facultad de Informática

Taller de Proyecto II

Práctica N°1: Estación Meteorológica

Agosto 2018

Integrantes:

- Gurmendi, Alan (1130/9)
- Palavecino, Facundo (1025/0)

Adquisición de Datos

El proceso de generación y adquisición de datos es implementado a través de Flask. Flask es un framework cuyo punto de inicio se encuentra en el archivo *app.py*. Este archivo define métodos para distintas rutas dentro del proyecto, comenzando por el *root* (“/”).

La generación propiamente dicha de los datos se hace a través de una clase llamada *Process* (*process.py*) que consiste en un loop infinito que genera valores aleatorios para las 4 variables a medir, arma una tupla con estos valores y los guarda en una base de datos cada 1 segundo.

Para que se inicie la generación de los datos, Flask espera que se renderize la página *index.html* para la página *root* del proyecto. Cuando el usuario cargue por primera vez la página, dará comienzo al script *process.py* que generará y guardará los datos.

En este punto, reconocemos 2 procesos que están trabajando simultáneamente: *process.py* que genera y guarda la información, y *app.py* que es el proceso de Flask que levanta la página web para que el usuario interactúe con el sistema.

Interacción con el usuario

Para que el usuario pueda interactuar con el sistema, se sirve una única página web que por un lado, muestra de manera dinámica la información requerida, es decir, promedios de las últimas 10 muestras y la última muestra tomada. Por otro lado, permite que el usuario elija la frecuencia con la que la página actualiza la información.

Esta página web es responsive, es decir, se adapta a distintos tamaños de pantalla. Además, cuenta con 2 tablas, 1 para mostrar las últimas 10 muestras de la base de datos con los promedios de los atributos, y otra para mostrar el detalle de la última muestra recolectada, ofrece una barra de progreso que le indica al usuario cuánto tiempo falta para recibir nueva información y 4 botones que permiten mostrar/ocultar las tablas, comenzar la actualización y cambiar la frecuencia de ésta última.

Para modificar dinámicamente los elementos, se usa JavaScript y jQuery, donde se enlazan los distintos elementos presentes en HTML a funcionalidades descritas en estos lenguajes.

Punto 2

El problema más recurrente que se encuentra se asocia a los accesos concurrentes por parte de los procesos a la base de datos.

Por un lado, el simulador de las muestras accede 1 vez por segundo a la base para guardar una nueva muestra. Por el otro, la página web accede a la base de datos 2 veces para pedirle información nueva cada cierto tiempo (según seleccione el usuario puede ser 1 segundo o 60 segundos).

Encontramos que estos accesos pueden generar problemas en la base. Para poder agregar/modificar/sacar información, la base de datos registra transacciones. Estas transacciones envuelven la consulta correspondiente para que la base no pierda consistencia en sus datos. Cuando una transacción falla, la base pierde consistencia y no admite que otra transacción tome lugar, ya que la información que maneja puede ser errónea.

Esto conlleva a que cuando una consulta (por parte del simulador o la página) falla, la base no acepte ninguna otra subsiguiente, y por ejemplo la página ya no pueda actualizar la información periódicamente.

Esto se agrava si además el sistema le permite al usuario cambiar el período de actualización, que termina siendo el período con el que la página hace consultas a la base de datos. En el caso crítico, la base recibe 1 consulta por segundo por parte de la página, congestionando el tráfico de los datos.

En un ambiente real la problemática seguirá presente, porque el responsable del error es el motor de la base de datos que no soporta la cantidad de accesos que el sistema requiere. Es por ello, que por más que las muestras provengan de una placa real, es necesario realizar las transacciones para guardar la información en la base, de la misma manera que se realiza ahora.

En general, los problemas de tiempo real se presentan en el tráfico de la información, por ejemplo, desde una placa hasta el servidor, o desde el servidor hasta la aplicación que presenta la información al usuario. También, los problemas pueden aparecer si el poder de procesamiento del navegador o el equipo no es lo suficientemente alto para el volumen de información que recibe.

Punto 3

La diferencia más grande reside en cómo se generan los datos y en cómo el generador de estos datos le manda la información a la base.

En el simulador, los datos se generan de forma aleatoria y, como la base reside en el mismo equipo que él, guarda la información rápidamente.

Pero, en el caso de una placa externa que sense las variables, va a ser necesario que envíe la información al servidor presente en otro equipo.