Proyecto 3 IPC2

201801155 – Bryan Eduardo Caal Racanac

Resumen

Actualmente el manejo de datos es muy importante para el desarrollo de muchas cosas, como aplicaciones, proyecciones, la toma de decisiones y muchas cosas más, por lo que miles y miles de datos se generan cada segundo para esto, se tiene que tomar en cuenta que cuando vamos a organizar los datos debemos de tener métodos eficientes, esto para reducir el tiempo para la toma de decisiones importantes que nos pueden llevar a una conclusión de suma importancia para ciertas situaciones, la siguiente aplicación ordena estos datos de manera en la que podemos consultar los datos que necesitamos y esta nos retornara la información que le estamos solicitando, la ventaja de la siguiente aplicación es que es una aplicación web por lo que no tomara muchos recursos de nuestra maquina como lo haría una aplicación de escritorio normal, ya que toda la información será alojada en un servidor remoto.

Palabras clave

POO, XML, Python, API, TDA

Abstract

Currently data management is very important for the development of many things, such as applications, projections, decision making and much more, so thousands and thousands of data are generated every second for this, it has to be taken into account that when we are going to organize the data we must have efficient methods, this to reduce the time for important decision-making that can lead us to a conclusion of paramount importance for certain situations, the following application sorts this data in a way that we can query the data we need and it will return the information that we are requesting, the

advantage of the following application is that it is a web application so it will not take many resources from our machine as a normal desktop application would, since all the information will be hosted on a remote server.

Keywords

POO, XML, Python, API, TDA

Introducción

Actualmente el uso de programas para el manejo de la información es crucial, cuando se tienen mucha información y se quiere dividir en pequeños pedazos para comprender mejor los datos se tiene un factor muy importante que es el factor del tiempo, por lo que si se hace a mano, llevaría mucho tiempo y esfuerzo, pero con el uso de herramientas tecnológicas podremos reducir estos tiempos a casi nada, por lo que se empezaron a desarrollar aplicaciones para poder manejar la información que nosotros queramos ordenas, buscar o seccionar, estas aplicaciones eran llamadas aplicaciones de escritorio pero consumían mucho espacio en memoria y si teníamos muchos datos el proceso que quisiéramos realizar se vuelve mucho más complicado y tardado por lo que estas aplicaciones, se empezaron a implementar en el internet, para que la información que necesitábamos organizar anteriormente ya no consumiera la memoria de la máquina y los procesos fueran más rápidos, la siguiente aplicación es un ejemplo de esto.

Desarrollo del tema

Una empresa de software le ha solicitado construir un software que pueda ser consumido desde Internet como un servicio. Este software recibirá un mensaje de la bitácora del software principal de la compañía y producirá una serie de información estadística relacionada. El mensaje que la bitácora enviará contendrá la siguiente información:

FECHA: dd/mm/yyyy
USUARIO: correo electrónico del usuario que genera el error
AFECTADO: lista de correos electrónicos separados por coma
ERROR: código numérico: descripción del error
El programa a desarrollar, luego de recibir el mensaje antes mencionado deberá almacenar la información necesaria en un archivo XML que permitirá mostrar la siguiente información:
FECHA: dd/mm/yyyy

Cantidad total de mensajes recibidos en esta fecha Listado de usuarios distintos que reportaron mensajes
Usuario1 cantidad mensajes generados
Usuario2 cantidad mensajes generados
… Listado de usuarios afectados
UsuarioX1
UsuarioX2
… Listado de errores distintos reportados
Código numérico del error: cantidad de mensajes recibidos
Código numérico del error: cantidad de mensajes recibidos
…
FECHA: dd/mm/yyyy
…

OJO: La sumatoria de los mensajes generados en el listado de usuarios que reportaron el mensaje y la sumatoria de los errores reportados deben cuadrar con la cantidad de mensajes recibidos en una fecha dada.

ARCHIVOS DE ENTRADA Y SALIDA

Archivo de entrada

Si el archivo de entrada posee un error de sintaxis con respecto a la estructura XML entonces el evento con el error será ignorado y se continuará con el análisis de los eventos restantes. Dentro de la etiqueta <eventos> pueden venir 1 o más etiquetas <evento>.

<EVENTOS>
<EVENTO>
Guatemala, 15/01/2021
Reportado por: <"Nombre Empleado 1" xx@ing.usac.edu.gt>
Usuarios afectados: aa@ing.usac.edu.gt, <bb@ing.usac.edu.gt>
Error: 20001 - Desbordamiento de búfer de memoria RAM
en el servidor de correo electrónico.
</EVENTO>
…
</EVENTOS>

Archivo de salida

Nombre del archivo: estadistica.xml
<ESTADISTICAS>
<ESTADISTICA>
<FECHA> 15/01/2021 </FECHA>
<CANTIDAD_MENSAJES>3</CANTIDAD_MENSAJES>
<REPORTADO_POR>
<USUARIO>
<EMAIL> xx@ing.usac.edu.gt </EMAIL>
<CANTIDAD_MENSAJES> 1 </CANTIDAD_MENSAJES>
</USUARIO>
<USUARIO>
<EMAIL> yy@ing.usac.edu.gt </EMAIL>

<CANTIDAD_MENSAJES> 2
</CANTIDAD_MENSAJES>
</USUARIO>
</REPORTADO_POR>
<AFECTADOS>
<AFECTADO> aa@ing.usac.edu.gt
</AFECTADO>
<AFECTADO> bb@ing.usac.edu.gt
</AFECTADO>
…
</AFECTADOS>
<ERRORES>
<ERROR>
<CODIGO> 20001 </CODIGO>
<CANTIDAD_MENSAJES>2</CANTIDAD_MENSAJES>
</ERROR>
<ERROR>
<CODIGO> 20002 </CODIGO>
<CANTIDAD_MENSAJES>
1</CANTIDAD_MENSAJES>
</ERROR>
...
</ERRORES>
</ESTADISTICA>
…
</ESTADISTICAS>

ARQUITECTURA

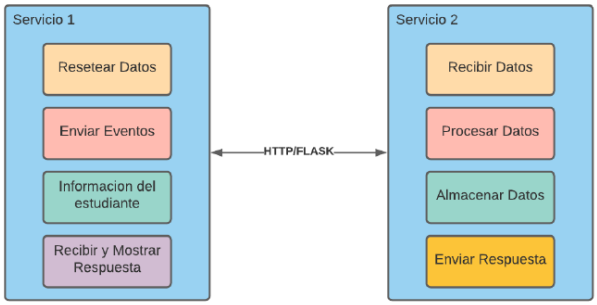
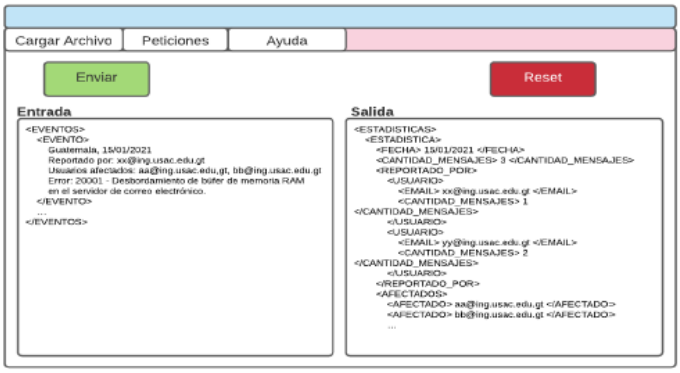


Figura 1: Arquitectura de la aplicación

Servicio 1 - Frontend

Este servicio consiste en una aplicación Web y consistirá en un simulador de la aplicación principal, contendrá únicamente las funcionalidades necesarias para testear el buen funcionamiento de la Api (Servicio 2), en esta aplicación se podrán mostrar los eventos que se procesarán y los datos estadísticos que fueron almacenados en la base de datos XML de salida).



Componentes:

Cargar Archivo: Se desplegará una pantalla para gestionar la carga de los archivos de entrada

con extensión .xml con uno o varios eventos. Se especifica en la sección de archivos de entrada y salida.

Peticiones: En este apartado se debe de tener las siguientes opciones:

Consultar Datos: Al seleccionar esta opción se deben de consultar los datos almacenados en el archivo estadisticas.xml y se mostrarán los datos en el recuadro de texto de salida.

Filtrar información por fecha y usuario que reporta: Al seleccionar esta opción se podrá elegir la fecha por la cual se requiere filtrar y se debe de **mostrar gráficamente** los usuarios que reportaron errores en esa fecha.

Filtrar por fecha y código de error: Al seleccionar esta opción se podrá ingresar un código de error, se deberá **presentar gráficamente** el total de mensajes que contienen ese código por cada fecha en donde se hizo el Reporte de dicho error.

Ayuda: desplegará 2 opciones, una para visualizar información del estudiante y otra para visualizar la documentación del programa.

Botón Enviar: Enviará los eventos del recuadro de texto a la Api para su posterior procesamiento.

Botón Reset: Este botón mandará la instrucción a la Api para devolver al estado inicial la Api, es decir sin datos.

Servicio 2 – Backend

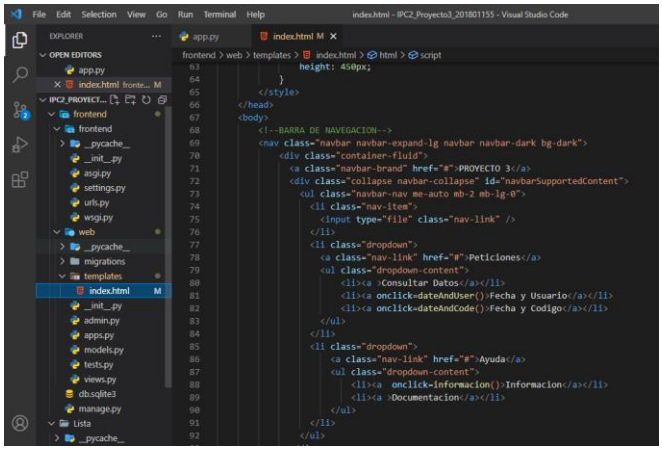
Este servicio consiste en una API que brindara servicios utilizando el protocolo HTTP, su funcionalidad principal es procesar los datos recibidos del servicio 1, luego de procesar los datos es necesario que estos sean almacenados en un archivo xml, este archivo está especificado en la sección de archivos de entrada y salida, este servicio también tiene la funcionalidad de devolver los datos que fueron almacenados para que sean mostrados como anteriormente se indica.

NOTA: Durante la calificación de este proyecto, el Servicio 2 podrá ser consumido desde otro cliente, por ejemplo, postman.

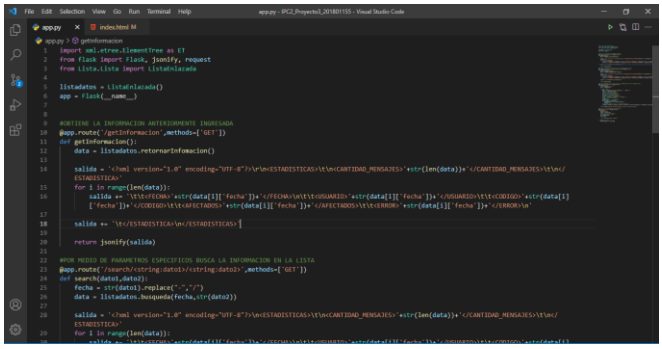
Para la correcta funcionalidad de la aplicación se utilizó lo que serían frameworks muy famosos dentro de python, uno de ellos es flask el cual por medio de rutas se puede usar completamente como lo sería un backend o un servidor, para la parte del frontend se usó lo que es django el cual es un framework de python con el cual se pueden hacer varias cosas como consumir apis e implementar de manera más practica el frontend como lo podemos ver en la parte de anexos.

Anexos

Implementación de django en el frontend



Implementación de flask en el backend



Referencias bibliográficas

Antonakos, James L. and Mansfield, Kenneth C., Jr. *Practical Data Structures Using C/C++* (1999). Prentice-Hall. ISBN 0-13-280843-9, pp. 165–190

Collins, William J. *Data Structures and the Java Collections Framework* (2002,2005) New York, NY: McGraw Hill. ISBN 0-07-282379-8, pp. 239–303

Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford *Introductions to Algorithms* (2003). MIT Press. ISBN 0-262-03293-7, pp. 205–213, 501–505