
PROYECTO 3 – IPC 2

201800665 – Samuel Alejandro Pajoc Raymundo

Resumen

Se creó una aplicación web utilizando dos frameworks distintos, uno para Back-end y otro para Front-end; la función principal del programa es leer un archivo seleccionado por un usuario, con formato XML, luego, este archivo será procesado y filtrado, luego se almacenarán los datos clasificados dentro de un archivo XML de salida, esto se realizará de esta forma ya que se simulará un almacenamiento permanente, debido a que aún no es posible realizar conexiones a bases de datos.

El usuario podrá elegir entre dos opciones, el método de ordenar los datos almacenados en el archivo XML de salida, luego podrá visualizarlos de forma gráfica.

Palabras clave

- Archivo
- Ruta
- Etiqueta
- Filtrar
- Expresión Regular

Abstract

It was created a web application with two different frameworks, the first one was used to create the Back-end part and the second one was used to create the Front-end part; the principal function of the program is read a selected file by the user, with an XML format, then this file will be processed and filtered, then, the classified data will be saved in an output XML file, this will be done in this way, because it isn't possible yet to connect a DataBase.

The user will can choose between two options, the method to order the data saved recently in the output XML file, after that it'll be posible to watch graphically the result.

Keywords

- File
- Route
- Label
- Filter
- Regular Expression

Introducción

Se realizó una aplicación con web por medio de dos servidores desarrollados en dos frameworks distintos, se trabajó la parte Back-end con Flask, y fue por medio de este que se construyó los endpoint para poder establecer los distintos procesos por los cuales pasará toda la información recopilada en el archivo XML de entrada.

Mientras que la parte del Front-end fue desarrollado con el framework llamado Django, este posee múltiples archivos que facilitan la creación de distintas vistas para el control de la aplicación web. El propósito general de trabajar con estos dos frameworks es porque ambos utilizan el lenguaje Python, y ya que durante el curso de IPC 2 se ha estado trabajando en diversas aplicaciones de escritorio con este lenguaje, esta práctica no sería la excepción por lo que fue necesario alcanzar el siguiente paso con este lenguaje, y por lo que se planteó un programa que funcionara en un navegador siendo este una aplicación web.

Desarrollo del tema

El funcionamiento de este programa inicia en una ventana principal llamada “pagina principal”, dentro de esta ventana se encuentra cinco botones los cuales son: Cargar Archivo, Peticiones, Ayuda, Enviar, Reset. Además, se podrá visualizar un área de texto en donde posteriormente se podrá visualizar el archivo seleccionado.

El botón Peticiones y Enviar, se encontrarán bloqueados, y para poder interactuar con ellos es necesario seleccionar primero un archivo XML que contenga únicamente las etiquetas EVENTO, EVENTOS.

El archivo será seleccionado por el usuario por medio de una ventana emergente disponible después de presionar el botón Cargar Archivo.

Al momento de presionar el botón Cargar Archivo, se desplegará una ventana en la cual abrirá el explorador de archivos, a continuación deberá buscar y seleccionar el archivo XML que desee analizar, el cual deberá contar con la estructura planteada para este proyecto; Este archivo debe contener las etiquetas EVENTOS, EVENTO (dentro de esta etiqueta se podrá encontrar múltiples líneas con cierta información previamente preestablecida según las especificaciones del programa).

Cuando se realice un clic sobre el botón Enviar este archivo será enviado hacia el Back-end, por medio de una solicitud http, ya estando en la función contenida en esa ruta, se iniciará con los procesos de agrupación y filtración de datos, luego se retornará un mensaje indicando que el proceso ha finalizado.

Luego, cuando el usuario realice un clic sobre el botón llamado Peticiones, el usuario podrá ver cuatro botones nuevos, los cuales son: Regresar (lleva a la página principal), Consultar Datos (después de realizar la consulta se bloqueará), Fecha, Código Error, Filtrar (por defecto estará bloqueado).

Cuando se de un clic sobre Consultar Datos, se podrá visualizar dentro de un cuadro de texto todo el contenido del archivo XML de salida generado después de haberse realizado los procesos de filtración, mientras que los botones de Fecha y Código Error, son únicamente para poder ordenar de forma gráfica los datos obtenidos en el archivo de salida. Estos datos son presentados en una tabla con especificaciones elegidas por medio del ordenamiento previamente elegido por el usuario, las

fechas son elegidas por medio de un combobox y el código de error es ingresado directamente en un área de texto, si este código no existe, se mostrará un mensaje indicando que no existe tal código, mientras que las fechas son elegidas directamente del archivo de salida, por lo que siempre existirán todas las fechas posibles por ordenar.

Para poder visualizar estos resultados es necesario dar un clic sobre el botón Filtrar, luego se mostrará un mensaje indicando que se ha generado el grafico, el cual se presentara por medio del formato png.

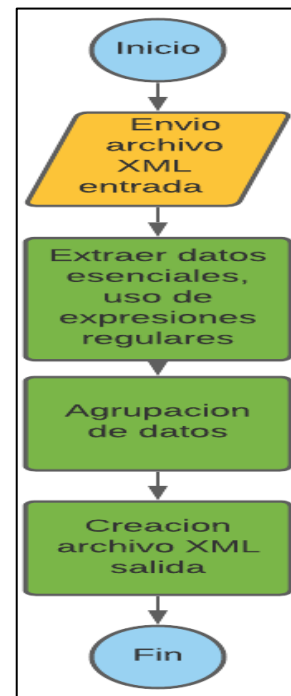
Continuando nuevamente desde la pagina principal, regresando con el botón Regresar, es posible visualizar el botón llamado Ayuda, luego de dar un clic sobre él, se desplegará una nueva ventana en donde se podrá ver dos botones nuevos y un texto con la información del estudiante, los botones son: Regresar y Documentación, al momento de dar un clic sobre Documentación se abrirá el archivo pdf, el cual contendrá todas las especificaciones sobre el desarrollo de esta aplicación web.

Regresando nuevamente a la pagina principal, y ahora dando un clic sobre el botón Reset, se reiniciará el funcionamiento del programa, y por lo que ahora será nuevamente posible elegir un nuevo archivo, para poder ser analizado.

Desarrollo Back-end

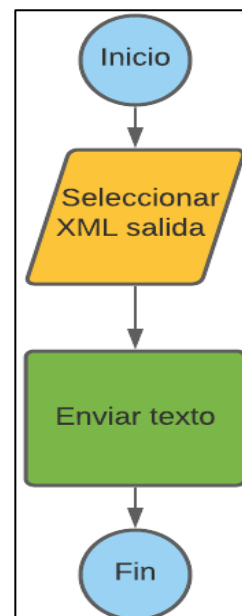
Endpoint “/AlmacenarXML”, esta es la ruta encargada de recibir el archivo XML seleccionado, el cual contiene varios caracteres no deseados, los cuales deben ser removidos; dentro de dicha ruta se encuentra la función llamada extraer datos, la cual realiza los primeros pasos de extracción de información.

Gráfico No.1 – diagrama UML, Filtración



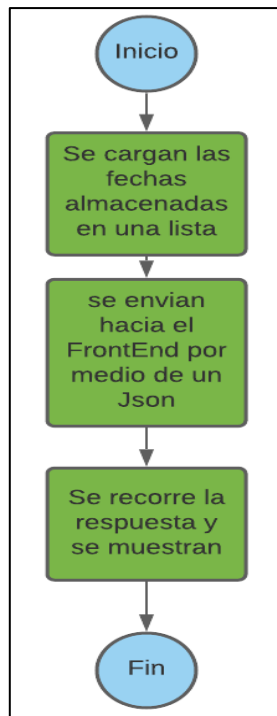
Endpoint “/CargarXMLsalida”, esta ruta es la encargada de leer el archivo de salida, el contenido de este XML es enviado por medio de una lista la cual contiene todos los caracteres escritos en este archivo.

Gráfico No.2 – diagrama UML, Lectura archivo salida



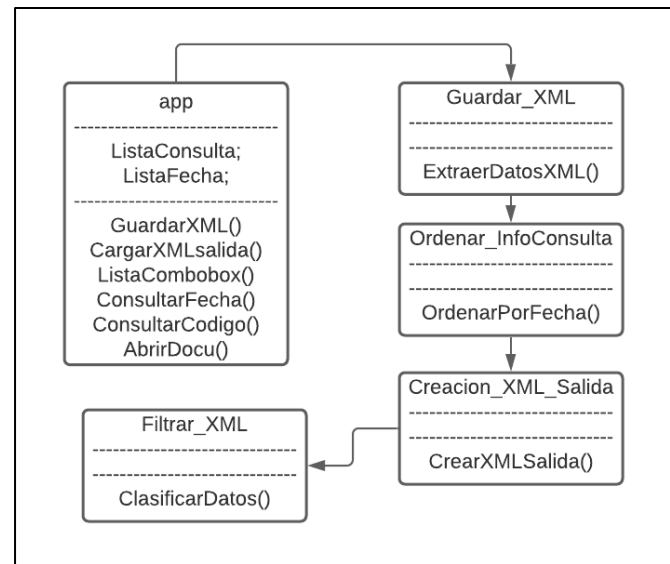
Endpoint “/ListaFechas”, esta ruta es la encargada de cargar las fechas contenidas dentro del archivo de salida previamente generado, después de haber realizado el envío, luego estas fechas son enviadas al frontend para que luego sean mostradas por medio de un combobox.

Gráfico No.3 – Diagrama UML, cargar fechas



Endpoints “/OrdenarFecha” y “/OrdenarCodigo”, estas rutas son llamadas dependiendo el proceso de filtración que haya especificado el usuario desde la vista peticiones, al momento de elegir un proceso de filtración de la información contenida en el archivo de salida. Por ultimo se tiene el Endpoint “/AbrirDocumentacion”, la cual como su nombre lo indica, permite mostrar la documentación perteneciente a dicho proyecto.

Gráfico No.4 - Diagrama de clases



Conclusiones

Se demostró que es posible trabajar con dos frameworks distintos, para poder llevar un mejor control del desarrollo de aplicaciones web.

Se observó la capacidad y limitaciones que posee trabajar con el lenguaje Python, comparado con otros framework que poseen un mayor orden al momento de crear componentes.

Se observó la facilidad que posee para crear endpoints al igual que, vistas, por lo que es muy recomendado para aquellos programadores principiantes que deseen entender un poco más sobre el desarrollo web.

Se observó que cuando se tiene un error en un servidor, la aplicación no se detiene por completo, debido a que su funcionamiento esta separado en dos partes principales en las cuales uno controla toda su lógica y la otra todo lo que tenga que ver con el

entorno gráfico, y por lo que es muy recomendado trabajar de esta forma ya que además de evitar reiniciar a cada momento el servidor, es más fácil detectar fallas, ya que es posible ir indicando que elementos pueden presentar errores.

Referencias bibliográficas

Charles Severance. Date (2013). Python para informáticos. Addison-Wesley Publishing Company, Inc.

David Budgen. Date (2003). Software web design. Pearson and Addison Wesley, editorials.

McGRAW-HILL. Date (2008). Fundamentos para aplicaciones INTERAMERICANA DE ESPAÑA, S. A. U.