|  |
| --- |
| **PROYECTO 2** |
| **202000605 – Camilo Ernesto Sincal Sipac** |

**Resumen**

Se realizó la creación de una web que permita la conversión de datos almacenados en archivos CSV en una estructura XML. De tal manera que el usuario tuviera la posibilidad de realizar cualquier modificación a esta para posteriormente extraer los datos y generar una nueva estructura XML con los datos necesarios para la realización de distintos gráficos y listas que cumplían la función de reflejar el análisis de la estructura generada de los archivos indicados por el usuario.

Para cumplir con cada funcionalidad de esta web se implementaron los conceptos de backend y frontend, haciendo uso de frameworks como flask y django respectivamente, siendo la primera de ellas responsable del análisis del XML para el retorno de valores para graficar y el segundo el encargado de analizar los archivos y generar la primera estructura XML, finalmente es necesario indicar el uso del protocolo http para él envió de datos hacia la API perteneciente al backend.

**Palabras clave**

XML, Framework, Django, Flask, HTTP

***Abstract***

*It was made a web that allows the conversion of data stored in CSV files in a XML structure. This web also allows the user to make any change in the structure for later get all the data and generate a new XML with the most important data to create different graphics and lists that fulfilled the function of show the analysis of the generated structure with the data of the files that the user indicated.*

*To accomplish all the functions of the web it was implemented the concepts of Backend and Frontend, using frameworks as flask and Django. Using the first one to analyze the XML and return the most important values to graph and the second one to analyze the CSV files and create the first XML structure.*

*Finally it was used the HTTP protocol to send the data from de Frontend to the API in the Backend.*

***Keywords***

XML, Framework, Django, Flask, HTTP

**Introducción**

Para el desarrollo web y para la creación de un sinfín de otras aplicaciones relacionadas a la informática existen los conocidos Frameworks, estos pueden ser considerados como estructuras realizadas con anterioridad o plantillas que facilitan el desarrollo de las aplicaciones anteriormente ejemplificadas.

Para el desarrollo del proyecto se implementaron dos de estos frameworks, uno de ellos fue Flask y otro Django, siendo usados para el backend y el frontend respectivamente.

De igual manera se usó la estructura de XML para el manejo de los datos, específicamente en la comunicación de ambos frameworks con el HTML, así como para el análisis de los datos y para su posterior uso en las gráficas y listas.

**Desarrollo del tema**

a. Lectura y conversión de CSV

Para la creación de la estructura XML se recibió un conjunto de cuatro archivos CSV, los cuales se enviaron a la API de Flask mediante un formulario en HTML, dado que los archivos se recibieron como bytes primero se realizó la conversión a datos de tipo

String, una vez realizado esto se realizó se utilizó la función de Split con los saltos de línea para guardar cada fila del archivo en una lista, posteriormente se recorrió esta lista a la vez que se volvía a utilizar la función de Split, pero esta vez usando punto y coma de tal manera que se formaran las listas con cada fila separada y dentro de esas filas cada dato por separado.

Con esto se logró tener cada dato separado por su fila y columna correspondiente.

b. Verificación de datos

Una vez se tuvo el conjunto de datos separados se hizo uso de expresiones regulares para su verificación.

Una expresión regular es un patrón que se utiliza para coincidir cadenas de caracteres, para esto existen signos o una combinación de ellos que tienen la funcionalidad de funcionar como los patrones anteriormente mencionados.

Cada patrón es comparado con un texto mediante distintos métodos que permiten extraer o comparar ambas cosas.

Por lo cual, para el análisis de cada dato, se recorrió la lista de datos en cada una des sus columnas y sus expresiones regulares correspondientes mientras estas coincidieran y en caso de no hacerlo se detenía el bucle y se establecía un valor falso a una variable boolena para evitar la creación del XML dado que existía errores.

En caso de que se hayan verificado que todos los datos no contengan errores se procede a la creación del XML.

c. Creación de XML y envío de datos

Para la creación de los XML se utilizó la librerí Element Tree, y de manera similar a la verificación de los datos se fuer recorriendo el arreglo que ya se había verificado saltándose una línea, dado que el encabezado de los archivos CSV no tienen importancia.

Cada grupo de etiquetas del XML fue siendo generado mediante un bucle que iba recorriendo el arreglo de datos creados en base al CSV.

Una vez se termina de generar el XML este es convertido en un string, para posteriormente aplicarle la identación con la librería DOM para finalmente agregarlo a un diccionario global y redirigiéndose al HTML raíz, el cual pasa dicho diccionario al HTML, con lo cual se permite ingresar a todas las variables y finalmente reflejar dicho XML en un área de texto.

d. análisis de XML en el backend.

Dado que el XML generado se encuentra en un área de texto, este ultimo es colocado dentro de un formulario, dado que se realizará un método POST con el texto en el interior de dicha área.

Para esto en la parte del framework de Django se hace un request de datos, el cual obtiene el texto y lo convierte a XML mediante la librería DOM.

Con dicha librería se recorre cada etiqueta XML, y se van almacenando los datos necesarios en arreglos globales, dado que posteriormente se accederán a ellos con un GET para la creación del nuevo XML, es necesario mencionar que para la lista solicitada se crea una lista de estados, el cual almacenará una palabra que indique como se encuentra el stock de los juegos, siento estas la palabra normal y rojo.

Dichas palabras servirán para crear clases en una tabla de HTML, lo cuál permitirá modificar sus atributos en la parte de CSS.

El GET anteriormente mencionado se realiza posterior a haber almacenado los datos necesarios para el nuevo XML que se enviará a HTML para mostrarse en un área de texto y de igual manera a Django para generar las gráficas.

e. Generación de gráficas

Para la generación de las gráficas primero se llamará al método GET de la parte del backend con Flask para poder obtener los datos que se usarán en dichas gráficas, para esto se recorrerá el XML que se retornó en la API, y se irán almacenando en un arreglo, el cual a la vez se guardará en el diccionario global que se enviará a la parte de HTML.

Una vez en el HTML mediante Java Script se importará una biblioteca conocida como Chart.js.

Dicha biblioteca mediante variables que contienen un conjunto de atributos permite generar distintas gráficas.

Dichos valores obtenidos serán los arreglos que se generaron Django.

Para el caso de las listas se recorre cada una de las listas y se irá generando cada fila en base a la cantidad de datos que se encuentren.

f. Conexión de Django y HTML

Para poder manejar Django como frontend se creó una aplicación, la cual fue configurada en el archivo de settings, indicando dicha aplicación en las aplicaciones instaladas, además se creo la variable de templates que permite el uso de HTML con Django.

Para poder acceder a cada uno de estos fue necesario configurar cada dirección el archivo de urls indicando el nombre de la función en la parte de view, un nombre a la url y finalmente un nombre para la ruta de manera general.

**Conclusiones**

El uso de Frameworks facilita de gran manera la creación de aplicaciones, en el caso de este proyecto permitió trabajar ambas partes de la web de manera sencilla y sin tantos problemas dado que facilitó las conexiones entre ambas partes.

De igual manera es necesario mencionar la versatilidad que estas generan al igual que el uso de estructuras de XML que permiten control información de manera rápida y sencilla.

**Referencias bibliográficas**

M. Gabriela, (2020) *Guía completa del Framework: qué es, cuáles tipos existen y por qué es importante en Internet.* Blog personal.

Anónimo, (2021) *Expresiones Regulares.* MDN Web DOCS