
Proyecto 2

201906085– Sebastian Alejandro de Leon Tenaz

Resumen

El Proyecto 2 se solicitaba una aplicación web la cual se programa ron con ayuda de los Frameworks Django para el Frontend y Flask para el Backend que pertenecen a Python.

Básicamente en este proyecto se pedí cargar 4 archivos csv los cuales se tenían que leer para ver que no tuviera errores de escritura y así poder generar un xml. Después de que los 4 csv's fueran aceptadas por el programa entonces se procedería a generar el xml que representa a los csv's.

Luego este xml se enviaba al servidor para poder generar otro xml los datos mas importantes del primer xml. Al obtener los datos de nuevo xml este podría generar diferentes reportes, graficas sobre los datos almacenados en estos archivos xml.

Por lo tanto los xml se utilizaron como una base de datos para la empresa de videojuegos y poder observar de forma grafica los datos mas importantes de su empresa.

Palabras clave

XML, Django, csv, Flask, Python, Frontend, Backend, Postman, GET, POST

Abstract

Project 2 was a project for a web application which was programmed with the help of the Django Frameworks for the Frontend and Flask for the Backend that belong to Python.

Basically in this project I was asked to load 4 csv files which had to be read to see that there were no writing errors and thus be able to generate an xml. After the 4 csvs were accepted by the program, then the xml that represents the csvs would be generated.

Then this xml was sent to the server to generate another xml with the most important data from the first csv. When obtaining the data again xml this could generate different reports, graphs on the data stored in these xml files.

Therefore, the xml was used as a database for the video game company and to be able to graphically observe the most important data of your company.

Keywords

XML, Django, csv, Flask, Python, Frontend, Backend, Postman, GET, POST

Introducción

El Proyecto 2 es un proyecto de aplicación web, que está escrito con la ayuda del framework Django front-end que pertenece a Python y el back-end Flask. Básicamente en este proyecto, se requiere cargar 4 archivos csv, estos archivos deben leerse para ver que no hay errores de escritura, para que pueda generar un xml.

Una vez que el programa acepta 4 csvs, continuará generando xml que representa el csv. Luego, este xml se envía al servidor con ayuda de los métodos HTTPS que son GET Y POST para que los datos más importantes del primer xml se puedan utilizar para generar otro xml.

Cuando los datos xml se recuperan nuevamente, esto puede generar un informe diferente, un gráfico de los datos almacenados en estos archivos xml.

Por tanto, este xml se utiliza como base de datos para empresas de videojuegos, pudiendo visualizar gráficamente los datos más importantes de tu empresa.

Desarrollo del tema

Para el desarrollo del proyecto 1 se necesitan saber conceptos importantes de programación y partes importantes para la realización de dicho proyecto, para ello a continuación se definirán lo que se utilizo en este proyecto:

-Python(como lenguaje de programacion):

Se presenta como un lenguaje de programación de propósito general, cuya expansión y popularidad es relativamente reciente. Se trata de Python, una apuesta por la simplicidad, versatilidad y rapidez de

desarrollo. A continuación veremos unas notas imprescindibles del lenguaje, extractadas básicamente de la página de inicio del lenguaje que se puede ver en www.python.org

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

-Django(Para Frontend):

Django es un framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Desarrollado por programadores experimentados, Django se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago.

Django fue desarrollado inicialmente entre 2003 y 2005 por un equipo que era responsable de crear y mantener sitios web de periódicos. Después de crear varios sitios, el equipo empezó a tener en cuenta y reutilizar muchos códigos y patrones de diseño comunes. Este código común se convirtió en un framework web genérico, que fue de código abierto, conocido como proyecto "Django" en julio de 2005.

-Flask(como Backend):

Flask es un “micro” Framework escrito en Python y concebido para facilitar el desarrollo de Aplicaciones Web bajo el patrón MVC.

La palabra “micro” no designa a que sea un proyecto pequeño o que nos permita hacer páginas web pequeñas sino que al instalar Flask tenemos las herramientas necesarias para crear una aplicación web funcional pero si se necesita en algún momento una nueva funcionalidad hay un conjunto muy grande extensiones (plugins) que se pueden instalar con Flask que le van dotando de funcionalidad.

El patrón MVC es una manera o una forma de trabajar que permite diferenciar y separar lo que es el modelo de datos (los datos que van a tener la App que normalmente están guardados en BD), la vista (página HTML) y el controlador (donde se gestiona las peticiones de la app web).

Archivo CSV:

Las siglas CSV vienen del inglés "Comma Separated Values" y significan valores separados por comas. Dicho esto, un archivo CSV es cualquier archivo de texto en el cual los caracteres están separados por comas, haciendo una especie de tabla en filas y columnas. Las columnas quedan definidas por cada punto y coma (;), mientras que cada fila se define mediante una línea adicional en el texto. De esta manera, se pueden crear archivos CSV con gran facilidad (lo explicamos más adelante). Es por esto que los archivos .csv están asociados directamente a la creación de tablas de contenido.

Archivo XML:

XML es el acrónimo de Extensible Markup Language, es decir, es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos. ¿Te parece complicado? Entonces, vamos a simplificarlo.

El lenguaje de marcado es un conjunto de códigos que se pueden aplicar en el análisis de datos o la lectura de textos creados por computadoras o personas. El lenguaje XML proporciona una plataforma para definir elementos para crear un formato y generar un lenguaje personalizado.

Método de petición HTTP:

HTTP define un conjunto de métodos de petición para indicar la acción que se desea realizar para un recurso determinado. Aunque estos también pueden ser sustantivos, estos métodos de solicitud a veces son llamados HTTP verbs. Cada uno de ellos implementan una semántica diferente, pero algunas características similares son compartidas por un grupo de ellos: ej. un request method puede ser safe, idempotent (en-US), o cacheable.

-GET: El método GET solicita una representación de un recurso específico. Las peticiones que usan el método GET sólo deben recuperar datos.

-POST: El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.

Frontend:

Front End es la parte de una aplicación que interactúa con los usuarios, es conocida como el lado del cliente. Básicamente es todo lo que vemos en la pantalla cuando accedemos a un sitio web o aplicación: tipos de letra, colores, adaptación para distintas pantallas(RWD), los efectos del ratón, teclado, movimientos, desplazamientos, efectos visuales... y otros elementos que permiten navegar dentro de una página web. Este conjunto crea la experiencia del usuario.

Como hemos dicho, el desarrollador front end se encarga de la experiencia del usuario, es decir, en el momento en el que este entra a una página web, debe ser capaz de navegar por ella, por lo que el usuario verá una interface sencilla de usar, atractiva y funcional.

Un desarrollador front end debe conocer los siguientes lenguajes de programación: HTML5, CSS3, JavaScript, JQuery, Ajax.

Backend:

Como hemos dicho antes, Front end es todo con lo que el usuario se encuentra directamente en la web o aplicación, entonces cuando hablamos de “Back end” nos referimos al interior de las aplicaciones que viven en el servidor y al que a menudo se le denomina “el lado del servidor”.

El back end del sitio web consiste en un servidor, una aplicación y una base de datos. Se toman los datos, se procesa la información y se envía al usuario. Los

desarrolladores de Front end y Back end suelen trabajar juntos para que todo funcione correctamente.

Un desarrollador Back end debe tener amplios conocimientos de los siguientes lenguajes: frameworks y los tipos de base de datos. No siendo necesario conocer todos los lenguajes pero sí entender y saber trabajar con algunos de ellos.

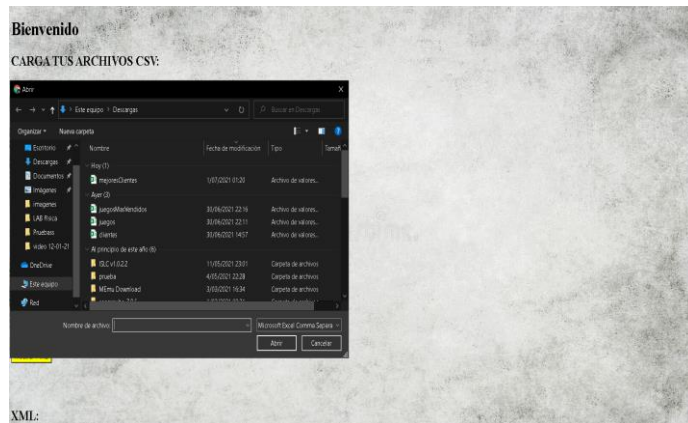
ASP.NET , PHP, Python, Ruby, Node.js, Java, MySQL, SQL Server, PostgreSQL, Oracle, MongoDB

SOLUCION:

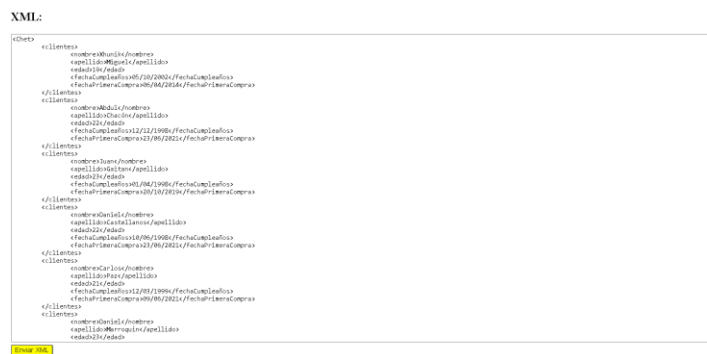
Ahora que ya poseemos estos conceptos para la realización de proyecto empezamos con la solución.

Para empezar se creo una aplicación en Django para poder usar las diferentes funciones de este Framework. Estas funciones se utilizaron para el frontend, por lo que se crearon una interfaz con html y css para este proyecto.

Django se utilizo para hacer el proceso de cargar los archivoscsv's(clientes.csv,juegos.csv , juegosMasVendidos.csv y mejoresClientes.csv), luego de haberlos cargado estos archivos csv pasaban una función de verificación para ver si no contenían algún error en ellos, para ello se uso la librería regex para poder validar las diferentes expresiones regulares para todos los campos.



Luego de que fueran validados que los archivos no contuviesen ningún error se puede mostrar el xml, el cual el cliente tendrá la opción de modificar el xml antes de enviarlo al servidor.



Luego este archivo se envia al servidor usando el método POST para conectar con el backend que se realizo en framework Flask y devolver un archivo xml nuevo el cual contiene los campos mas importantes.

Debido a este nuevo xml se podrá realizar diferentes reportes sobre ello como por ejemplo:

Para los mejores clientes se podrá observar una grafica de barras por cada cliente.

Para los juegos mas vendidos se podrá observar una grafica de pie.

Para la clasificación se mostrara una grafica para la cual devolverá el numero de juegos que pertenecen a esa clasificación.

Para cumpleaños una lista de los cumpleaños
ordenados por mes

Y por ultimo una listado de juegos que muestra un listado de juegos disponibles y el stock que hay sobre cada uno.

Conclusiones

-Se determino que es importante utilizar diferentes frameworks para una mayor seguridad del contenido de la aplicación.

-Se determino django y flask son frameworks muy distintos.

-Se determino que conocimiento de los métodos HTTP son útiles en la programación web.

Referencias bibliográficas

-Stefaniak, P. (2019, 10 septiembre). ¿Qué es Backend y Frontend? Descubre Comunicación.
<https://descubrecomunicacion.com/que-es-backend-y-frontend/>

- Gonzales, J. G. (2021, 1 julio). Introducción a Django - Aprende sobre desarrollo web | MDN.
<https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>

- Williams, R. W. (2021, 1 julio). Métodos de petición HTTP - HTTP | MDN. HTTP.
<https://developer.mozilla.org/es/docs/Web/HTTP/Methods>