
PROYECTO 3 – APLICACIÓN WEB

202000562 – Pedro Luis Pu Tavico

Resumen

La empresa Tecnologías Chapinas, S.A. está desarrollando una herramienta que sea capaz de facturar detalladamente los servicios de infraestructura de nube que aprovisiona a sus clientes.

La estructura de la tecnología de nube desarrollada por Tecnologías Chapinas, S.A. consiste en crear configuraciones de infraestructura que agrupan recursos necesarios para que una empresa pueda construir las arquitecturas de despliegue de aplicaciones que requiera.

Para esta aplicación se hará uso de un algoritmo que permita a la empresa llevar un control ordenado de todos los servicios de infraestructura que los clientes solicitan y así poder facturar de manera correcta cada uno de los servicios.

Palabras clave

POO: La Programación Orientada a Objetos es un paradigma de programación que parte del concepto de "objetos" como base.

Flask: Framework minimalista escrito en Python que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código.

NodeJS: Es un entorno en tiempo de ejecución multiplataforma, para la capa del servidor basado en el lenguaje de programación JavaScript.

API: conjunto de estructuras que permiten que los componentes de un software se comuniquen con otros.

Abstract

The company Tecnologías Chapinas, S.A. is developing a tool that is capable of detailed billing for the cloud infrastructure services it provides to its customers. The structure of the cloud technology developed by Tecnologías Chapinas, S.A. It consists of creating infrastructure configurations that group the necessary resources so that a company can build the application deployment architectures that it requires.

For this application, an algorithm will be used that allows the company to keep an orderly control of all the infrastructure services that customers request and thus be able to bill each of the services correctly.

Keywords

OOP: Object Oriented Programming is a programming paradigm that starts from the concept of "objects" as a base.

Flask: Minimalist framework written in Python that allows you to create web applications quickly and with a minimum number of lines of code.

NodeJS: It is a cross-platform runtime environment for the server layer based on the JavaScript programming language.

API: set of structures that allow software components to communicate with others.

Introducción

Esta aplicación web está compuesta por dos aplicaciones principales, un servidor de Python el cual funciona como Backend, el cual ayuda con el manejo de datos, y una plataforma Web que servirá como Frontend, el cual ayuda para presentar la información de manera más resumida y completa posible.

Dado que actualmente el mercado de desarrollo de software tiene bastante competencia, nuestro equipo de desarrollo desarrolló una aplicación original, siendo muy personalizada para que el usuario en cuestión pueda manejarse entre ventanas de una manera rápida y sencilla, y tener una interfaz capaz para cualquier persona que opte por primera vez al puesto de trabajo, siendo esta muy intuitiva para cualquier persona que requiera realizar una solicitud, teniendo como puntos fuertes las opciones de agregar, eliminar y generar un reporte de los datos de la empresa.

Django

Es un software gratis y de código abierto, que se encarga de la mayor parte del desarrollo web y permite que el desarrollador se concentre en la creación de la aplicación. Django maneja autenticación de usuario, administración de contenido, mapa del sitio, y otras tareas. Permite tener una escala rápida y flexible.

Flask

Es la decisión de diseño de hacer tareas simples de manera simple, no debe de tener mucho código y se usan request válidos.

Métodos de petición HTTP

Los métodos de petición HTTP conforman una de las funcionalidades más importantes del lenguaje de marcado, pues se encargan de acceder a los servidores para realizar peticiones.

JSON

JavaScript Object Notation (JSON) es un formato basado en texto estándar para representar datos estructurados en la sintaxis de objetos de JavaScript. Es comúnmente utilizado para transmitir datos en aplicaciones web (por ejemplo: enviar algunos datos desde el servidor al cliente, así estos datos pueden ser mostrados en páginas web, o viceversa)

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.

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.

Un archivo XML se divide en dos partes: prolog y body. La parte prolog consiste en metadatos administrativos, como declaración XML, instrucción de procesamiento opcional, declaración de tipo de documento y comentarios. La parte del body se compone de dos partes: estructural y de contenido (presente en los textos simples).

El diseño XML se centra en la simplicidad, la generalidad y la facilidad de uso y, por lo tanto, se utiliza para varios servicios web. Tanto es así que hay sistemas destinados a ayudar en la definición de lenguajes basados en XML, así como APIs que

ayudan en el procesamiento de datos XML - que no deben confundirse con HTML.

API

Las API son mecanismos que permiten a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos.

API significa “interfaz de programación de aplicaciones”. En el contexto de las API, la palabra aplicación se refiere a cualquier software con una función distinta. La interfaz puede considerarse como un contrato de servicio entre dos aplicaciones. Este contrato define cómo se comunican entre sí mediante solicitudes y respuestas.

La arquitectura de las API suele explicarse en términos de cliente y servidor. La aplicación que envía la solicitud se llama cliente, y la que envía la respuesta se llama servidor.

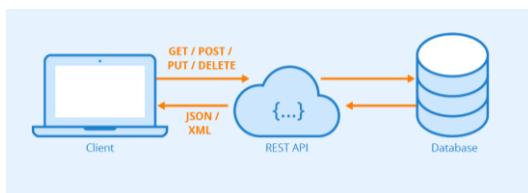


Figura 1. Ejemplo de función de un API.
Fuente: elaboración propia.

Como se puede observar en la imagen anterior, el funcionamiento de un api es bastante sencillo de entender. Las consultas que el cliente realiza son peticiones mediante http.

Para la realización de esta aplicación web se usaron los mecanismos de un api, ya que esta nos permite comunicarnos entre componentes mediante protocolos. Esta aplicación está compuesta por dos componentes, los cuales son el backend y el frontend, mismos que serán detallados más adelante.

Backend

Este término es utilizado para referirse al área lógica de toda página web. Nos referimos a la arquitectura interna del sitio que asegura que todos elementos desarrollen la función correcta. No está visible a ojos del usuario y no incluye ningún tipo de elemento gráfico. Esta área es la encargada, además de la funcionalidad del sitio, de la seguridad y la optimización de los recursos. Al momento de construir el backend este puede ser creado en un lenguaje de programación en conjunto de un framework.

En la actualidad existen muchos lenguajes y frameworks para la creación del backend. Para esta aplicación web se hará uso del lenguaje Python en conjunto con el framework flask para la realización del backend.

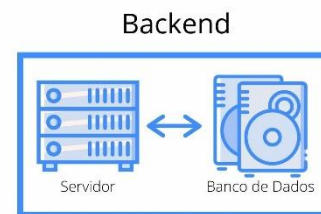


Figura 2. Ejemplo de función del backend.
Fuente: elaboración propia.

Como bien sabemos en el backend se encuentra toda la lógica de nuestra aplicación. Para el desarrollo del backend de esta aplicación se integro la base de datos dentro del backend esta misma será detallada más adelante.

Base de Datos

Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un sistema de gestión de bases de datos (DBMS). En conjunto, los datos y el DBMS,

junto con las aplicaciones asociadas a ellos, reciben el nombre de sistema de bases de datos, abreviado normalmente a simplemente base de datos.

Los datos de los tipos más comunes de bases de datos en funcionamiento actualmente se suelen utilizar como estructuras de filas y columnas en una serie de tablas para aumentar la eficacia del procesamiento y la consulta de datos. Así, se puede acceder, gestionar, modificar, actualizar, controlar y organizar fácilmente los datos.

La mayoría de las bases de datos utilizan un lenguaje de consulta estructurada (SQL) para escribir y consultar datos.

En esta aplicación web la base de datos fue creada a partir de un archivo XML, el cual se encuentra almacenado en la base de datos. En este archivo se almacenarán todos los datos que sean ingresados por el cliente.

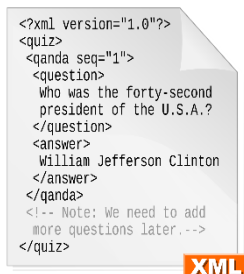


Figura 3. Ejemplo de archivo XML.
Fuente: elaboración propia.

Frontend

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, 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.

Para la creación de esta aplicación web se utilizó el framework Django para el desarrollo del Frontend, ya que este framework es muy optimo y sencillo de utilizar.

Frontend



Figura 4. Representación de función del frontend.
Fuente: elaboración propia.

Conclusiones

- Flask es un framework de Python, el cual ayuda en la realización de apis haciendo que el código sea el menor posible. En la realización del backend de este programa Flask resulto ser una herramienta muy útil y práctica, ya que es muy fácil de entender al momento de programar.
- Django es un framework bastante completo y practico al momento de usarlo. En este proyecto Django fue utilizado en la creación del frontend, y este hizo posible agregar lógica en los archivos html sin la necesidad de javascript, ya que este posee la capacidad de poder realizar operaciones en un html.
- Una base de datos puede ser creado a través de un archivo XML ya que este tipo de archivos posee las capacidades para realizar modificaciones en el.

Referencias bibliográficas

- Problemas resueltos de listas / Mónica Adriana Carreño León 2010.
- CAIRÓ O., GUARDATI S., Estructura de Datos, México, Mc Graw-Hill, 2001.
- JOYANES A. LUIS, Algoritmos y Estructuras de Datos: Una Perspectiva en C., España, Mc Graw-Hill, 2005.

Anexos

Diagrama de Clases "Backend"

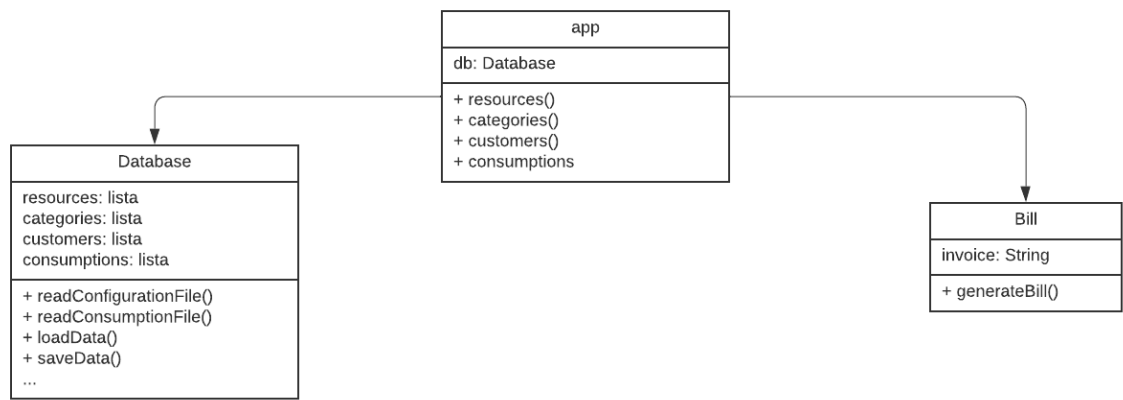


Diagrama de Clases "Frontend"

