
TERCER PROYECTO IPC2

202106208 – Dwight Fernando Gabriel Chinchilla Hernández

Resumen

Aplicación web creada para Tecnologías chapinas S.A para la facturación detallada de sus servicios de infraestructura de nube que aprovisiona a sus clientes, la cual consiste en la creación de cargas de sus configuraciones que agrupan los recursos necesarios.

Permitiendo para la creación de estos, crearlos manualmente o realizar una carga masiva a través de un archivo xml con una estructura específica para la lectura. Para la creación de la aplicación web se utilizaron los framework: flask para el backend y django para el frontend.

La forma en que muestra la información es por medio de archivos en .pdf y a través del programa de postman o en la misma consola.

Palabras clave

Xml: Formato de archivo de carga

Iteraciones: Son ciclos que realiza el programa del tejido de cada paciente.

Pdf: Formato escritura.

Abstract

Web application created for Tecnologías chapinas S.A for the detailed billing of its cloud infrastructure services that it supplies to its clients, which consists of creating loads of its configurations that group the necessary resources.

Allowing for the creation of these, create them manually or perform a massive load through an xml file with a specific structure for reading. For the creation of the web application, the frameworks were used: flask for the backend and django for the frontend.

The way it displays the information is through .pdf files and through the postman program or in the same console.

Keywords

Xml: Upload File Format

Iterations: These are cycles carried out by the tissue program for each patient.

Pdf: Writing format.

Introducción

La simulación de muestra el comportamiento de n numero de clientes, en alguna empresa la cual requiera servicio para alquilar algunos componentes computacionales, la lógica del programa se puede aplicar a unas ciertas empresas, pero se requieren ciertos parámetros para que el programa realice las funciones correctamente, para ello se necesita de los archivos de carga, los cuales son dos.

La aplicación web inicia activando el servidor del backend para el manejo de la información. Como programa para pruebas se utilizo el programa llamado como Postman, el cual posee muchas funciones para simular la actividad del backend, es este apartado del programa se pueden crear las instancias solicitadas, las cuales son, las de crear nuevos recursos, nuevas categorías, nuevas configuraciones, y también nuevos clientes e instancias.

Desarrollo De La Aplicación

La creación de los endpoint obtenidas fue por medio del software de Postman, lo que se hizo fue crear un simulaciones, para cada request que solicite el usuario.

La aplicación está elaborada con el paradigma orientado a objetos, las clases utilizadas fueron varias para almacenar de manera dinámica la información, entre algunas de las clases que se utilizo están: Recursos, Usuarios, Instancias, Categoría, entre otras muchas

Una vez cargado el archivo y presionando el botón enviar, se mostrarán en los un mensaje que la carga se realizó con éxito con el contenido del archivo en que se haya cargado.

Para crear un tipo de cliente, recurso o categoría nueva debemos ir al endpoint crear clientes, crear recursos o crear categorías, estos endpoint son requeridos para saber que información se crea debe almacenar o pedir al usuario.

Para empezar el paso a realizar fue el de crear la estructura del backend.

```
@app.route('/cargarArchivo',methods=['POST'])
def readXML(): ...

@app.route('/cargarArchivoConsumos',methods=['POST'])
def readXML2(): ...

@app.route('/recursos',methods=['GET'])
def obtener_recursos(): ...
```

Figura 1. Backend

Fuente: elaboración propia.

Para poder estructurar los demás pasos se tomo de referencia los primeros métodos creados en el archivo main, luego de eso se procedió a la realización de los métodos para poder hacer cada una de las opciones.

1. Cargar archivo xml: Aquí se utilizó el programa Postman para solicitarle al usuario el documento y xml.dom.minidom para poder recibir la información de los dos archivos de carga.

Durante la carga del archivo se generan de una vez la carpeta que corresponde a cada recurso o parámetro necesitado con el propósito de ir almacenando la información desea, las listas creadas, y se va desplegando a mas listas, las cuales son, la lista de recusos, lista de categoria, lista de eclientes, cada una

de estas listas con sus distintos métodos para extraer la información.

```
global lista_recursos, 1
lista_recursos = []
lista_categorias = []
lista_clientes = []
```

Figura II. Listas creadas

Fuente: elaboración propia.

2. **Crear Datos:** Esta función de crear datos hace referencia a la creación de recursos, categorías y sus derivantes y la creación de clientes y sus instancias, para ello se hace uso del programa postman, el cual con el método de crear y con el endpoint de tipo post pide la información de tipo json al usuario.

```
@app.route('/crear_clientes', methods=['POST'])
def crear_clientes():
    json=request.get_json()
    existe=False

    for i in lista_clientes:
        if(i.nit==json['nit']):
            existe=True

    if(existe==False):
        lista_clientes.append(Usuario(json['nit'], json['nombre'], json['usuario']
        return jsonify({'ok':True, 'data': 'Cliente añadido con éxito'}),200)
    else:
        return jsonify({'ok':False, 'data': 'Este usuario ya está registrado'})
```

Figura III. Menú dinámico.

Fuente: elaboración propia.

- POST Crear Recursos
- POST Crear Categorías
- POST Crear Configuración

Figura 1. Submenú de mostrar pacientes.

Fuente: elaboración propia.

realizan las operaciones mas importantes, las cuales brinda la información de los tiempos y la información de la facturación y de los montos, al igual gracias a esto, la empresa tiene estadísticas de sus clientes y su alcance de metas.

Conclusiones

Una vez más el paradigma de POO es un método que brinda demasiada utilidad a la hora de trabajar con información encerrada en categorías.

Adicionalmente, pueden incluirse preguntas abiertas a la reflexión y debate, temas concatenados con el tema expuesto o recomendaciones para profundizar en la temática expuesta.

Los archivos de marcado como lo es xml, markdown, etc. Brindan grandes posibilidades poder estructurar la información de forma que sea más rápida su lectura por programas secundarios.

Postman es una herramienta que brinda todo lo esencial para poder simular la actividad de un servidor, con sus distintos tipos de request.

Adicionalmente, se puede decir que la lógica del sistema, es compleja, pero gracias a los programas se pueden hallar valores, los cuales se pueden utilizar para realizar estadísticas, y para poder realizar estimaciones o cálculos a cerca del comportamiento de los clientes de la empresa.

Referencias bibliográficas

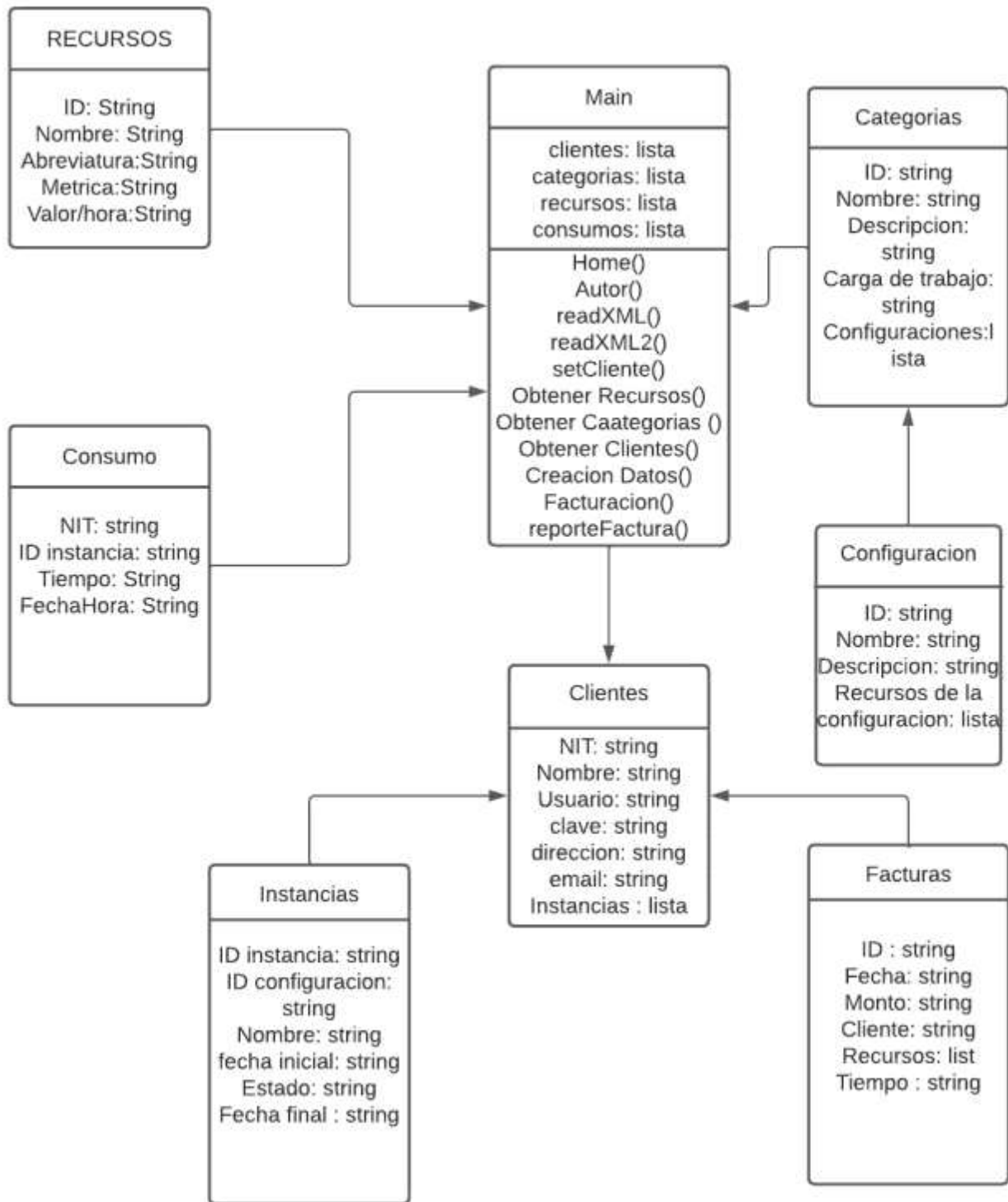
3. Manejo Facturacion: Este es el apartado mas importante del programa, ya que en el se

- Guagliano, C. (2019). *Programación en Python II:*

Programación orientada a objetos. RedUsers

- *Python 3: los fundamentos del lenguaje.* (2016).

Ediciones ENI.



Anexos

