
PROYECTO 3

202113293 – Genesis Nahomi Aparicio Acan

Resumen

Este proyecto representa una aplicación para la empresa “a Tecnologías Chapinas, S.A.”, la cual esta construyendo 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 la resolución a esta aplicación plantea la utilización de un Frontend (parte grafica que el usuario puede ver) y un Backend (parte de código que esta oculta al usuario) utilizando principalmente las aplicaciones y o librerías que proporcionan Flask para la parte de el Bakend y Django para la parte que corresponde al Frontend , conectando ambas para una mejor apreciación . la función principal del programa es mediante la lectura de dos XML de entrada generar facturas de diversos productos que adquiere un cliente, de igual forma se pueden crear mas datos como recursos categorías... al igual que consultar estos

Palabras clave

Flask, Django, Backend, Frontend, POST, GET.

Abstract

This project represents an application for the company "a Tecnologías Chapinas, S.A.", which is building a tool to is building a tool that is able to bill in detail the cloud infrastructure services it provides to its customers. The structure of the cloud technology developed by Tecnologías Chapinas, S.A. consists of creating infrastructure configurations that group the necessary resources so that a company can build the application deployment architectures it requires.

For the resolution to this application proposes the use of a Frontend (graphic part that the user can see) and a Backend (part of code that is hidden from the user) using mainly applications and or librerias that provide Flask for the part of the Bakend and Django for the part that corresponds to the Frontend, connecting both for a better appreciation . the main function of the program is by reading two XML input generate invoices for various products purchased by a customer, just as you can create more data as resources categories ... as well as consult these

Keywords

Flask, Django, Backend, Frontend, POST, GET

Introducción

Se desarrollo una aplicación la cual permite la lectura de dos archivo con formato XML, uno de estos archivos tiene la información de varias configuraciones que contienen recursos , datos, categorías, configuraciones, instancias . El otro archivo tienen la información de los consumos que se harán teniendo información del nit del cliente a quien se le tramitara la factura, fecha y hora de la factura , id de la instancia .

Esta aplicación permite el manipular estos datos y hacer una simulación de una factura al ingresar el nit la fecha de inicio y la fecha final de el cliente a generar factura

Desarrollo del tema

1.Requisitos para el programa

Para la solución de este problema fue utilizado el lenguaje de programación Python el cual se debe de instalar en el ordenador para el correcto desarrollo del programa, al igual que contar con los requisitos mínimos de una computadora con sistema operativo Windows .

2. FLASK

Flask es un framework escrito en Python que permite crear aplicaciones de forma sencilla y rápida. Es decir, un acelerador de tareas que funciona con pocas líneas de código y que ejecuta las aplicaciones rápidamente. Flask es La forma más rápida y ágil de desarrollar una aplicación básica ya que es un micro Framework .

Actualmente en el desarrollo moderno de aplicaciones web se utilizan distintos Frameworks que son herramientas que nos dan un esquema de trabajo y una serie de utilidades y funciones que nos facilita y nos abstrae de la construcción de páginas web dinámicas.

En general, los frameworks están relacionados con los lenguajes de programación (Ruby on Rails (Ruby), Symphony (PHP)), Django es el más conocido en el mundo de Python, pero Flask es uno que puede no tener una curva de aprendizaje tan alta. , pero nos permite crear una aplicación web tan compleja, que se puede crear en Django.

3.Django

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.

Django es una herramienta que se puede usar para el desarrollo full-stack de aplicaciones y páginas web, así como para el desarrollo de servidores.

Estas son algunas de las características por las que Django es uno de los frameworks más usados para el desarrollo web:

- Completo
- Escalable
- Versátil
- Seguro
- Portátil

- Mantenimiento

3. Desarrollo del tema

La aplicación web inicia en su menú principal el cual se muestran las diferentes opciones

1. Cargar datos
2. Cargar consumos
3. Mostrar datos
4. Agregar recursos
5. Agregar categoría
6. Agregar cliente
7. Agregar instancia
8. Crear factura
9. Ayuda



Figura1 pagina principal
Fuente: elaboración propia

3.1 Cargar datos y consumos

La función de estos botones es el dirigir al usuario a las respectivas paginas donde deben de ingresar el nombre del archivo para poder cargarlos al sistema ,si el archivo se subió correctamente se mostrar en pantalla un mensaje

3.2 Mostrar datos

Este botón lo dirige al apartado donde se mostraran los datos cargados al sistema

3.3 Agregar recursos, categorías, cliente, instancia

Este botón lo dirige a los respectivos apartados donde el usuario puede ingresar los respectivos datos y cargarlos al sistema

Figura2. pagina de crear recursos
Fuente: elaboración propia

3.4 Crear factura

Este botón lo lleva a una página donde debe de ingresar su nit, fecha de inicio y fecha final del usuario para poder generar una factura al usuario con el nit correspondiente , al igual que se generara un reporte de lo que el esta adquiriendo

3.5 Ayuda

Este botón lo llevara a un apartado donde se mostrara la información del usuario a su vez que mostrara este documento en pantalla

4 Backend

El backend se conforma de varios métodos GET y POST para las diferentes funciones que tiene la aplicación como el crear recursos, cargar archivos , mostrar archivos , etc. Este fue hecho y con la ayuda de Flask es posible su ejecución

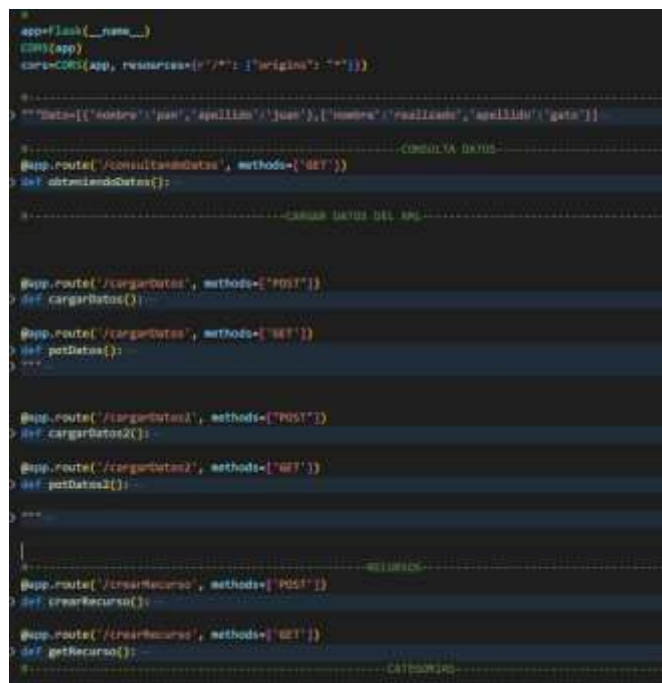


Figura3 metodos del Bakend
Fuente: elaboración propia

5. Frontend

El Frontend fue hecho con Django el cual facilito la creación de este , contiene una gran cantidad de modulos y carpetas que se conectan entre si y a su vez este se conecta con el bacekd para hacer una interacción mas amigable para el usuario



Figura5. metodos del Frontend que interactúan con el Backend

Fuente: elaboración propia

Conclusiones

Django es uno de los frameworks mas completos para la creación de aplicaciones web , junto con Python se pueden crear paguinias bonitas y funcionales , solo que al principio de ello es un poco tedioso y confuso el trabajar con el dado a la gran cantidad de modulos que genera que uno tiende a tener miedo de tocar alguna clase y o un modulo por equivocación y escribir código en el haciendo que su programa se arruine, pero con practica y paciencia poco a poco se puede aprender

Referencias bibliográficas

- Introducción a Django Recuperado 2 de noviembre de 2022, de <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction> .
- Qué es Flask, José Domingo Muñoz Recuperado 2 de noviembre de 2022, de <https://openwebinars.net/blog/que-es-flask/>

Diagrama de clases

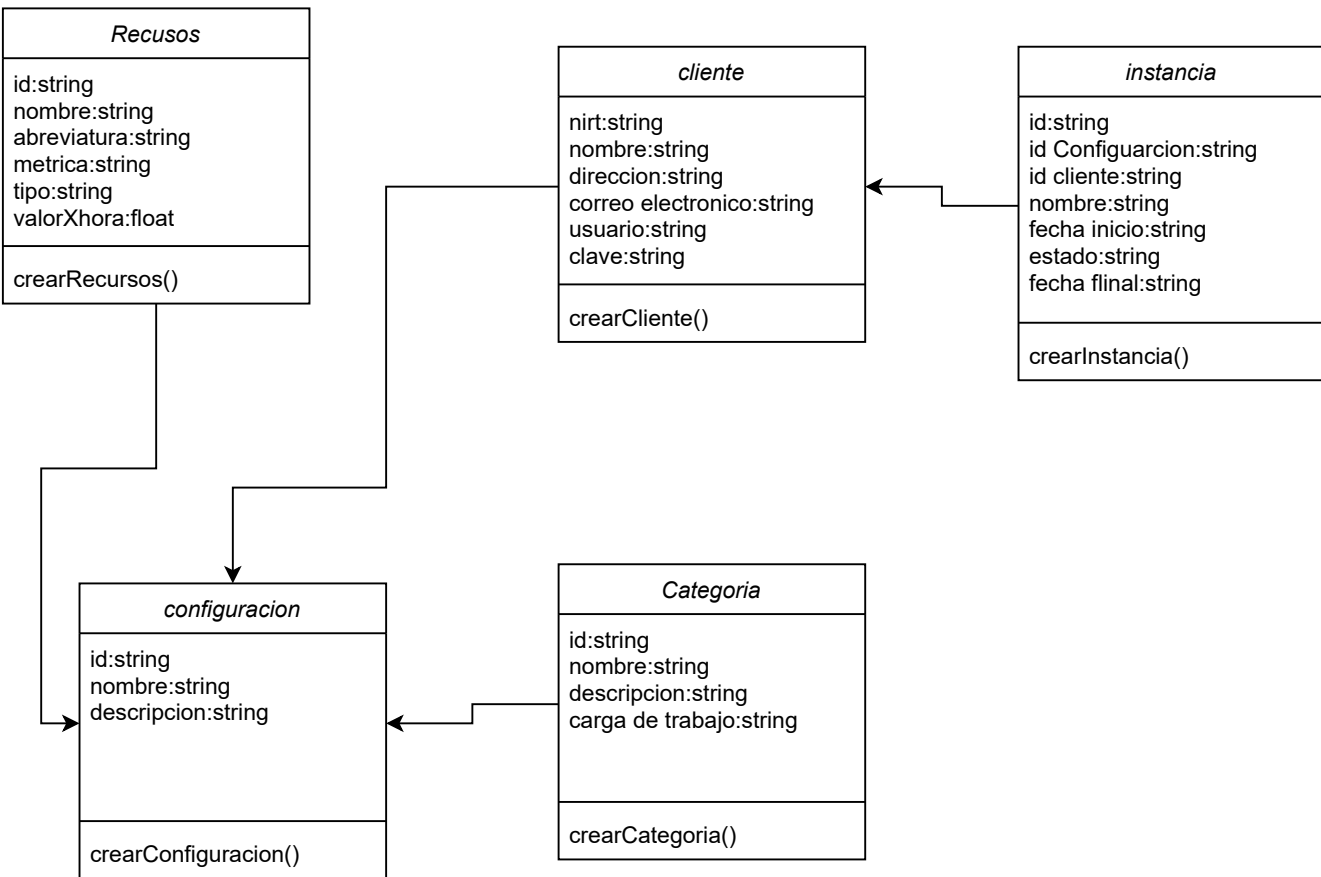


figura 5 diagrama de clases
fuente : elaboracion propia

EL MODELO DE DATOS RELACIONAL

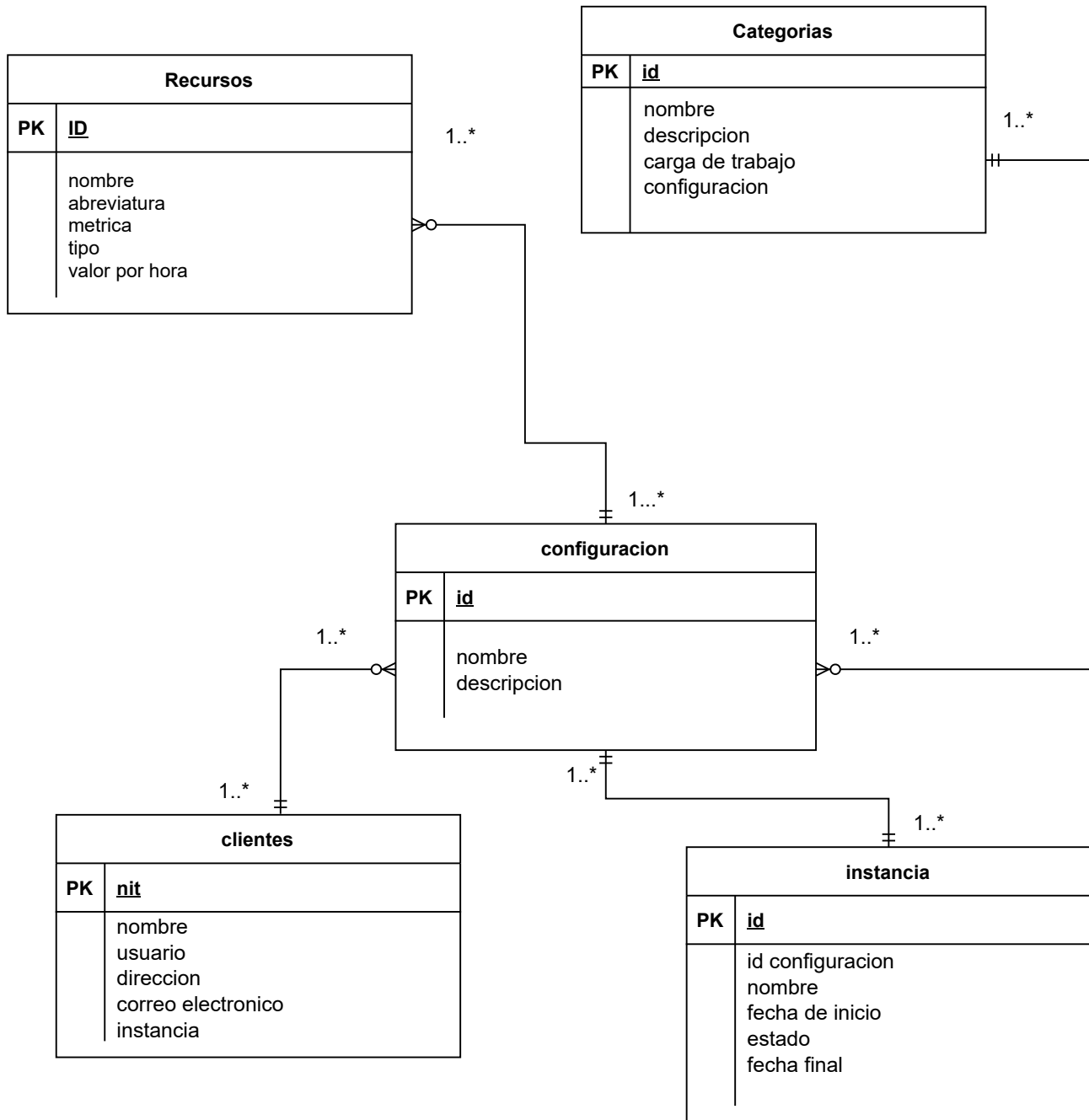


Figura6. modelo de datos

Fuente: elaboración propia