Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 2do. Semestre 2025

PROYECTO 3 LABORATORIO – SISTEMA TECNOLOGIAS CHAPINAS, S.A

201906795 – Javier Ricardo Yllescas Barrios

Resumen

El programa presentado propone el desarrollo un sistema de facturacion para servicioes en la nube, compuesto por un frontend en Django y un backend en Flask. El sistema procesa datos mediante archivos XML para gestionar recursos, clientes y configuraciones, calculando consumos y generando facturas y reportes.

El objetivo de la aplicacion es usar paradigmas de programacion orientada a objetos para la construccion del software, la utlilizacion de base datos XML y la interaccion entre dos aplicaciones con diferenes tecnologias utilizando el protocolo HTTP para interactuar con el backend.

Palabras clave

Proyecto2, Sistema Frontend (Django), Sistema Backend (Flask), USAC, IPC2.

Introducción

El presente proyecto tiene como objetivo desarrollar una solución tecnológica integral para la empresa Tecnologías Chapinas, S.A., que le permita administrar su portafolio de servicios de nube y realizar los procesos de facturación correspondientes al consumo de recursos por parte de sus clientes.

La solución propuesta implementa una arquitectura moderna que combina un frontend web desarrollado con Django, que servirá como interfaz de gestión, y un backend construido con Flask que proveerá servicios API para el procesamiento de datos. El sistema utilizará archivos XML para la comunicación y persistencia de datos, aplicando los principios de la programación orientada a objetos y expresiones regulares para el procesamiento de información.

Desarrollo del tema

Para una mejor compresion del proyecto se aplicaron los siguientes archivos para desarrollarlo.

Estructura del Proyecto [Archivos]

– DOCUMENTACION_IPC2_Proyecto3_201 90695.pdf

—>ArchivosdePrueba

—>Backed

- AppFlask.py
- ArchivoConfiguraciones.xml
- ArchivoConsumos.xml
- entrada.xml

- entradaconsumos.xml

—>Sistemas

-SistemaCentral.py

-SistemaLecturaXML.py

-SistemaLecturaXMLconsumos.py

-SistemaSalidaXML.py

-SistemaSalidaXMLconsumos.py

-SistemaValidaciones.py

—>Clases

-ArchivoConfiguracion.py

-ArchivoConsumos.py

—>Frontend

—>app

-__init__.py

-admin.py

-apps.py

-models.py

-tests.py

-urls.py

-views.py

—>migrations

—>static

—>templates

—>Frontend

(Generados por Django)

Funcionamiento Frontend

Donde:

—>app

Guarda todos los relacionado a la aplicacion en Django.

app/urls.py

Estan todas las Rutas y asignacion de sus funciones

app/views.py

Funciones para captaurar datos y renderizar html

Funcionamiento Backend

—>Backed

Almacena todo lo realcionado a la aplicacion con flask

Backed/AppFlask.py

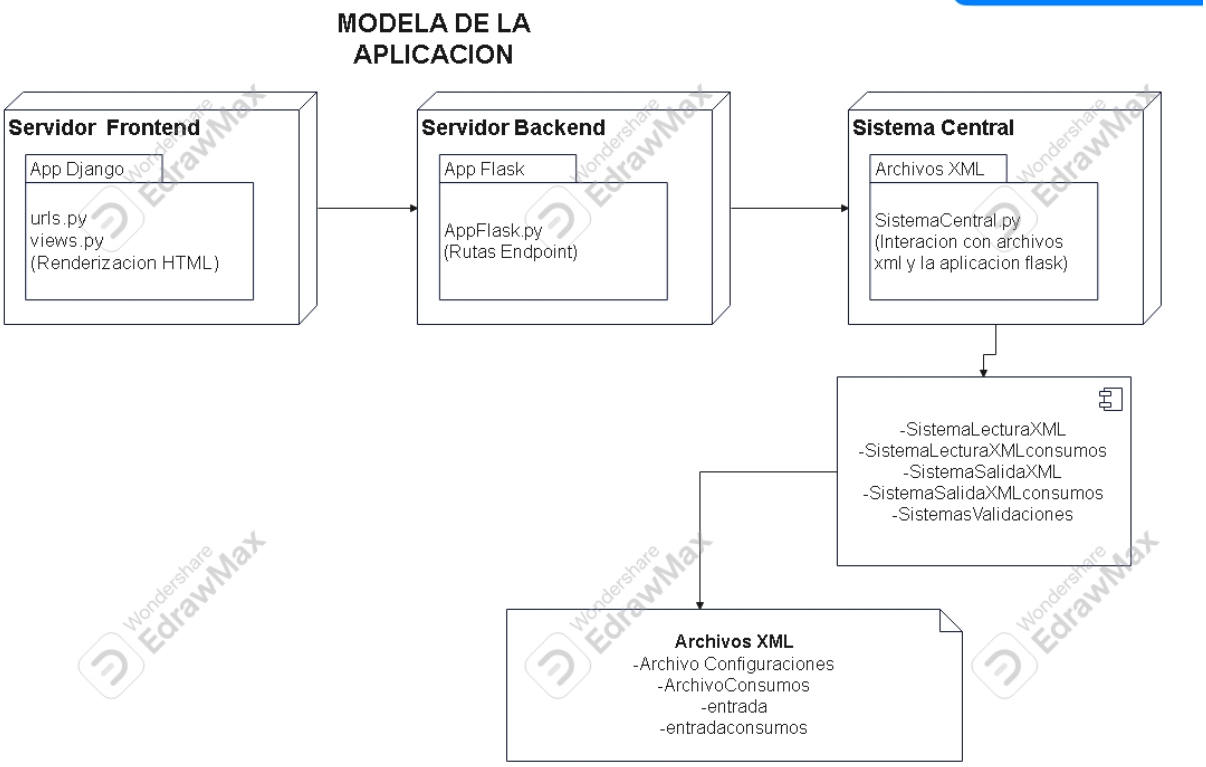
Contiene todas las rutas para los endpint para la interaccion con la aplicaion

—>Backed/Sistemas

Estan todos los sistemas para ejecutar la aplicacon con la base de datos XML.

Backed/Sistemas/SistemaCentral.py

Es el encargado de recibir de flask las instrucciones para poder interactura con los archivos XML y obtener las clases y creacion, eliminacion y moficacion de los archivos XML


MODELA DE LA APLICACION
Servidor Frontend
App Django
urls.py
views.py
(Renderizacion HTML)
Servidor Backend
App Flask
AppFlask.py
(Rutas Endpoint)
Sistema Central
Archivos XML
SistemaCentral.py
(Interacion con archivos xml y la aplicacion flask)
-SistemaLecturaXML
-SistemaLecturaXMLconsumos
-SistemaSalidaXML
-SistemaSalidaXMLconsumos
-SistemasValidaciones
Archivos XML
-Archivo Configuraciones
-ArchivoConsumos
-entrada
-entradaconsumos

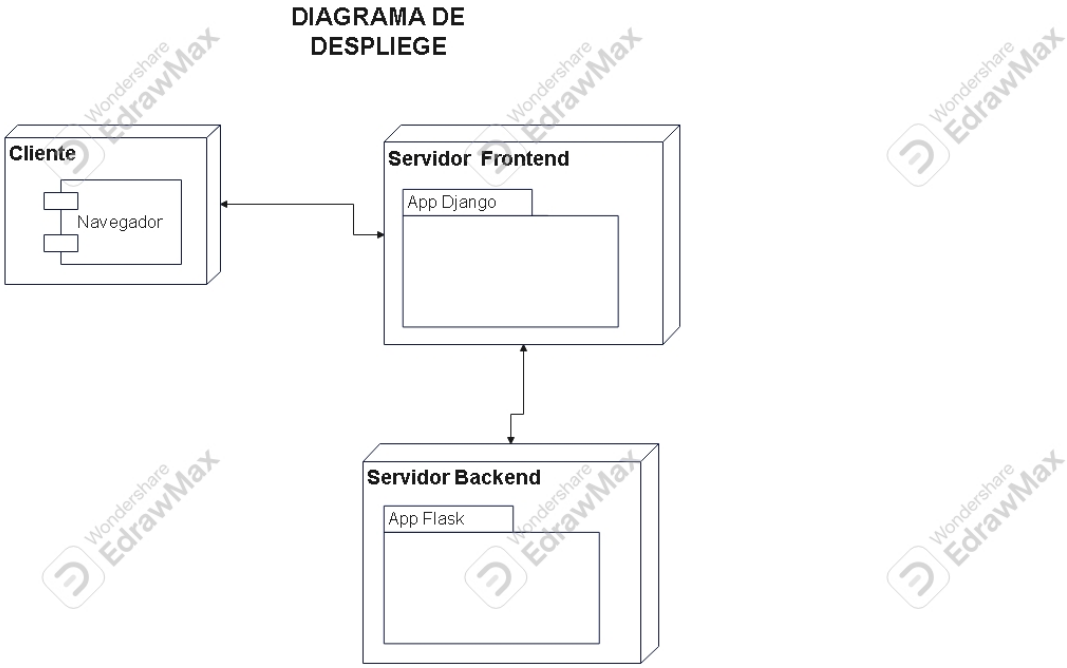
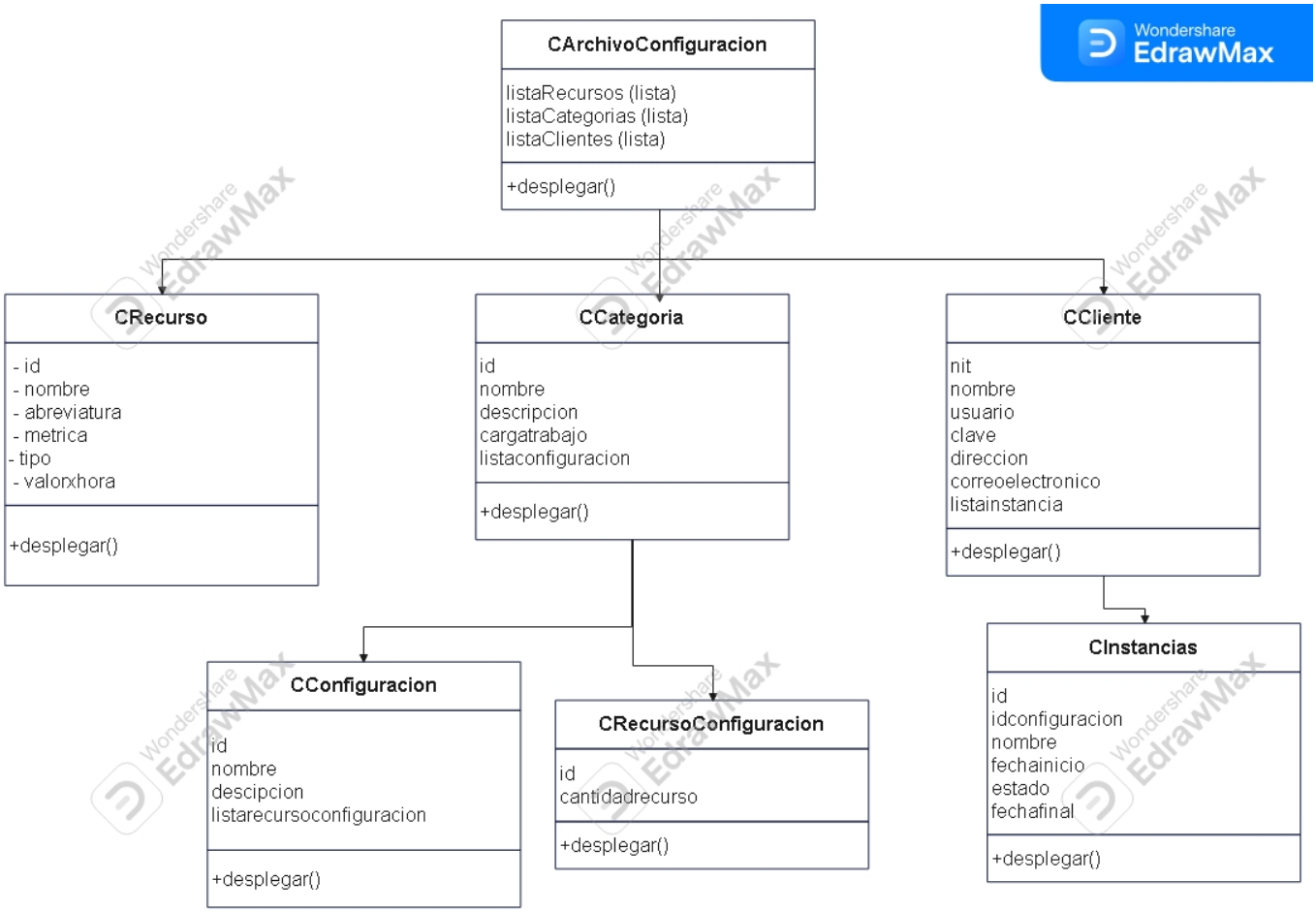


DIAGRAMA DE DESPLIEGE
Cliente
Navegador
Servidor Frontend
App Django
Servidor Backend
App Flask

Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 2do. Semestre 2025

CLASES SISTEMA CENTRAL (ARCHIVOS XML)




SistemaCentral
SisLeerArhvXML
ArchivoConfiguracion
SisSalidaXML
SisVal
mensajeErroresXML
SisLeerArhvXMLCons
ArchivoConsumos
SisSalidaXMLCons
Test()
LeerArchivo(ruta)
ValidarArchivo()
GuaradarArchivoConfiguraciones(ruta)
LeerBaseDatosArchivoConfiguraciones(ruta)
LeerArchivoConsumos(ruta)
ValidarArchivoConsumosCliente()
GuaradarArchivoConsumos(ruta)
SistemaLeerArchivosXMLconsumos
ruta
contenidoXL
domXML
ArchivoListaConsumos
obtenerArchivoListaConsumos()
msg(mensaje, extra=None)
leerArchivo()
asignarruta(ruta)
obtenerArchivoConsumo()
SegmentarArchivo()
SistemaLeerArchivosXML
ruta
contenidoXML
domXML
XMLArchivonvofiguracion
CArchivoConfiguracion
ListaRecuross
ListaCategorias
ListaClientes
obtenerArchivoConfiguracion()
msg(mensaje, extra=None)
asignarruta(ruta)
SegmentarArchivo()
obtenerlistaclientes()
obtenerlistaCategorias()
obtenerlistaRecursos()
leerArchivo()
SistemaSalidaXML
rutasalida
ArchivoConfig
doc
root
asignarruta(ruta)
asignarArchivoConfiguraciones(archivo)
GuardarArchivoConfiguraicones()
segmentar_archivo_XML()
creararchivoDOC()
crear_archivo()
msg(mensaje, extra=None)

Como ejecutar el Proyecto.

Para ejecutar el proyecto necesitas tener python, flask, request, django, jsonify instalado y saber usar CMD

Paso 1: Ejecutar el Backend accede a la carpeta Backend y ejecuta en CMD “python AppFlask.py”

Paso 2: Ejecuta Frontend buscar la carpeta forntend y ejecuta en CMD “python manage.py runserver”

específicos planteados, incluyendo el consumo y procesamiento de mensajes XML, la validación de datos mediante expresiones regulares, la generación de reportes en PDF y la implementación de un sistema de facturación basado en consumo de recursos. El uso de control de versiones con GitHub facilitó un desarrollo organizado y metódico, resultando en una solución robusta que satisface las necesidades de gestión de infraestructura cloud y facturación requeridas por la empresa.
Flask al ser de un framework basado en python es muy sencillo para ejecutar toda nuestra logica de back-end

.

Comentarios:
Este proyecto represento una excelente oportunidad para aplicar conceptos clave de desarrollo de software en un contexto parecido al real, integrando backend, frontend, bases de datos y APIs,mientras se usa la programacion orientada a objetos enseñada utilizando clases especificas para alamcenar toda la informacion y tener un mejor manejo

.

Referencias bibliográficas

Universidad de San Carlos de Guatemala. (s.f.).
Enunciado proyecto 1.
https://uedi.ingenieria.usac.edu.gt/campus/pluginfile

.php/270932/mod_resource/content/1/%5BIPC2%5DProyecto_3_2S2025-v2.pdf

Argueta, Hesban. (s.f.). *Clases laboratorio IPC2* [Repositorio en GitHub]. GitHub. https://github.com/Hes-007/IPC2-2S2025/tree/main.

Ruiz Juarez, J. M. (s.f.). Contenido Unidad 2 y 3. https://uedi.ingenieria.usac.edu.gt/campus/course/view.php?id=2547

Flask (s.f) Documentacion para uso framework Flask https://flask.palletsprojects.com/en/stable/tutorial/