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

Sistema de Facturación Automatizada para Servicios de Nube – Tecnologías Chapinas S.A

202405516 – Selvin Raúl Chuquiej Andrade

Resumen

El presente proyecto desarrolla una solución integral para la empresa Tecnologías Chapinas, S.A., orientada a automatizar el proceso de facturación de servicios en la nube mediante el uso de programación orientada a objetos (POO) y tecnologías web modernas. El sistema implementa una arquitectura cliente-servidor compuesta por un backend en Flask, encargado del procesamiento de datos y la generación de facturas en formato XML, y un frontend en Django, que funciona como simulador de la aplicación principal para enviar configuraciones, consumos y ejecutar el proceso de facturación. La persistencia de datos se gestiona mediante archivos XML que actúan como base de datos estructurada, garantizando trazabilidad y compatibilidad con los mensajes de entrada definidos en el enunciado. Además, el sistema permite la generación de reportes detallados y análisis de ventas, contribuyendo a una gestión eficiente de recursos tecnológicos y consumos de clientes. Este proyecto demuestra la aplicación práctica de conceptos de API REST, POO, manejo de datos y desarrollo web integral..

Palabras clave
Facturación — API REST — XML — Django — Flask

Abstract

El presente proyecto desarrolla una solución integral para la empresa Tecnologías Chapinas, S.A., orientada a automatizar el proceso de facturación de servicios en la nube mediante el uso de programación orientada a objetos (POO) y tecnologías web modernas. El sistema implementa una arquitectura cliente-servidor compuesta por un backend en Flask, encargado del procesamiento de datos y la generación de facturas en formato XML, y un frontend en Django, que funciona como simulador de la aplicación principal para enviar configuraciones, consumos y ejecutar el proceso de facturación. La persistencia de datos se gestiona mediante archivos XML que actúan como base de datos estructurados, garantizando trazabilidad y compatibilidad con los mensajes de entrada definidos en el enunciado. Además, el sistema permite la generación de informes detallados y análisis de ventas, contribuyendo a una gestión eficiente de recursos tecnológicos y consumos de clientes. Este proyecto demuestra la aplicación práctica de conceptos de API REST, POO, manejo de datos y desarrollo web integral..

Keywords
Facturación — API REST — XML — Django — Flask

Introducción

La facturación automatizada de servicios tecnológicos en la nube representa un componente esencial para las empresas que administran infraestructuras virtualizadas y múltiples recursos de cómputo. En este contexto, Tecnologías Chapinas, S.A. requiere un sistema capaz de registrar configuraciones, clientes, instancias y consumos, para posteriormente calcular y generar las facturas de manera eficiente y precisa.

Este proyecto implementa una solución integral basada en los principios de la programación orientada a objetos (POO), el uso de archivos XML como medio de persistencia y la comunicación mediante API REST. El sistema está compuesto por un backend desarrollado en Flask y un frontend en Django, los cuales interactúan para procesar los datos y generar los reportes requeridos. Con ello, se optimiza la administración de los servicios en la nube, garantizando escalabilidad, transparencia y control en los procesos de facturación.

Desarrollo del tema

a. Arquitectura general del sistema

El sistema implementado adopta una arquitectura cliente-servidor, compuesta por dos componentes principales: un frontend desarrollado en Django y un backend en Flask, los cuales se comunican a través del protocolo HTTP mediante solicitudes API REST. Este diseño garantiza la independencia entre la lógica de presentación y la lógica de negocio, lo que mejora la escalabilidad y el mantenimiento del software.

El módulo de frontend cumple la función de simulador de la aplicación principal que utilizaría la empresa Tecnologías Chapinas, S.A., brindando una interfaz web intuitiva que permite cargar los archivos XML de configuración, registrar consumos y ejecutar los procesos de facturación y generación de reportes. Por su parte, el backend desarrollado con Flask se encarga de recibir, procesar y almacenar los datos provenientes del frontend, gestionando internamente las operaciones CRUD (crear, leer, actualizar y eliminar) mediante rutas específicas para cada tipo de entidad del sistema (clientes, instancias, configuraciones, recursos y consumos).
Esta separación de responsabilidades permite mantener un diseño modular, escalable y fácilmente extensible, además de facilitar la interoperabilidad entre ambos entornos. Al implementar esta arquitectura, se logra un flujo de comunicación claro y eficiente entre las capas del sistema, donde Django envía las solicitudes al servidor Flask, el cual responde con los resultados procesados en formato JSON o XML según el tipo de operación. En conjunto, ambos componentes representan una aplicación web integral que combina usabilidad, automatización y persistencia de datos..

b. Implementación de la base de datos XML y APIs

A diferencia de los sistemas tradicionales que utilizan bases de datos relacionales, este proyecto utiliza archivos XML como medio de persistencia de datos, con el propósito de simular el comportamiento de un entorno real donde las configuraciones, consumos y facturas deben ser intercambiadas en este formato. Cada archivo XML representa una “tabla lógica” que almacena información jerárquica y estructurada sobre categorías, configuraciones, clientes, instancias, consumos y facturas.
El manejo de estos archivos se implementa mediante el módulo xml.etree.ElementTree de Python, el cual permite crear, leer, modificar y eliminar nodos XML de forma dinámica. De esta manera, el sistema

garantiza la integridad de los datos, evitando duplicidades mediante verificaciones de identificadores únicos (por ejemplo, el idRecurso o el nitCliente).
Cada entidad del sistema está representada por una clase independiente, diseñada bajo los principios de la programación orientada a objetos (POO). Por ejemplo, las clases Cliente, Instancia, Recurso y Configuracion encapsulan atributos y métodos específicos que facilitan la creación de objetos y la conversión de sus datos a formato XML mediante métodos to_dict().
El backend Flask expone las APIs necesarias para interactuar con estos archivos, permitiendo registrar nuevos elementos, consultar la información existente, generar consumos, inicializar el sistema y ejecutar el proceso de facturación. Todas las comunicaciones se realizan mediante peticiones HTTP POST y GET, devolviendo respuestas en formato JSON, lo que permite la interoperabilidad con cualquier cliente externo o sistema de integración.

c. Proceso de facturación automatizada

El módulo de facturación constituye el núcleo del sistema, ya que se encarga de realizar los cálculos económicos asociados al uso de los recursos de infraestructura en la nube. Este proceso se inicia cuando el usuario selecciona un rango de fechas, momento en el cual el sistema recorre los registros del archivo consumos.xml, identificando aquellos consumos que se encuentren dentro del intervalo solicitado y que aún no hayan sido facturados.
Una vez filtrados los datos, los consumos se agrupan por cliente utilizando su número de NIT, con el objetivo de generar una factura consolidada para cada uno. Para cada instancia consumida, el sistema determina la configuración utilizada y los recursos asociados, multiplicando la cantidad de cada recurso por su valor por hora y el tiempo total consumido. El resultado de este cálculo representa el subtotal por recurso, el cual se acumula para determinar el monto total de la factura.
Posteriormente, el sistema genera un nuevo archivo facturas.xml o actualiza el existente, registrando la información completa de cada factura, incluyendo su número único, fecha de emisión, NIT del cliente y monto total. De forma adicional, se marca cada consumo correspondiente como “facturado”, lo que evita duplicaciones en futuras ejecuciones. Este proceso elimina completamente la necesidad de cálculos manuales y reduce significativamente los errores humanos, garantizando la trazabilidad, precisión y transparencia de la información financiera.
Gracias a esta automatización, Tecnologías Chapinas, S.A. puede procesar grandes volúmenes de datos y generar facturas de manera rápida, ordenada y coherente con los consumos reales de cada cliente.

d. Reportes y análisis de datos

Además del proceso de facturación, el sistema cuenta con un módulo de reportes y análisis que amplía sus capacidades administrativas. Este componente permite generar dos tipos de reportes principales: el detalle de factura y el análisis de ventas.
El reporte de detalle de factura permite seleccionar una factura específica y visualizar todos los datos relacionados: el número de factura, el NIT del cliente, la fecha, las instancias involucradas, los tiempos de consumo y el aporte económico de cada recurso. Este nivel de detalle proporciona transparencia tanto para el cliente como para la empresa, mostrando de manera clara cómo se compone el monto total facturado.
Por otro lado, el reporte de análisis de ventas ofrece una visión más global del rendimiento económico del sistema, permitiendo analizar qué categorías, configuraciones o recursos generan mayores ingresos durante un período determinado. Este tipo de análisis es fundamental para la toma de decisiones estratégicas, ya que permite identificar los servicios más rentables y optimizar la asignación de recursos tecnológicos.

Ambos reportes pueden exportarse en formato PDF o visualizarse desde la interfaz del frontend, brindando flexibilidad y facilidad de acceso. En conjunto, este módulo convierte al sistema en una herramienta integral no solo para la gestión técnica de la facturación, sino también para el control y monitoreo financiero de los servicios en la nube.

Conclusiones

El desarrollo del sistema de facturación automatizada para Tecnologías Chapinas, S.A. permitió aplicar de forma práctica los fundamentos de la programación orientada a objetos, la manipulación de archivos XML y la implementación de API REST para el intercambio de información entre aplicaciones. A través de la integración del backend en Flask y el frontend en Django, se logró una solución funcional, modular y escalable que automatiza los procesos de registro, consumo y facturación de servicios en la nube.

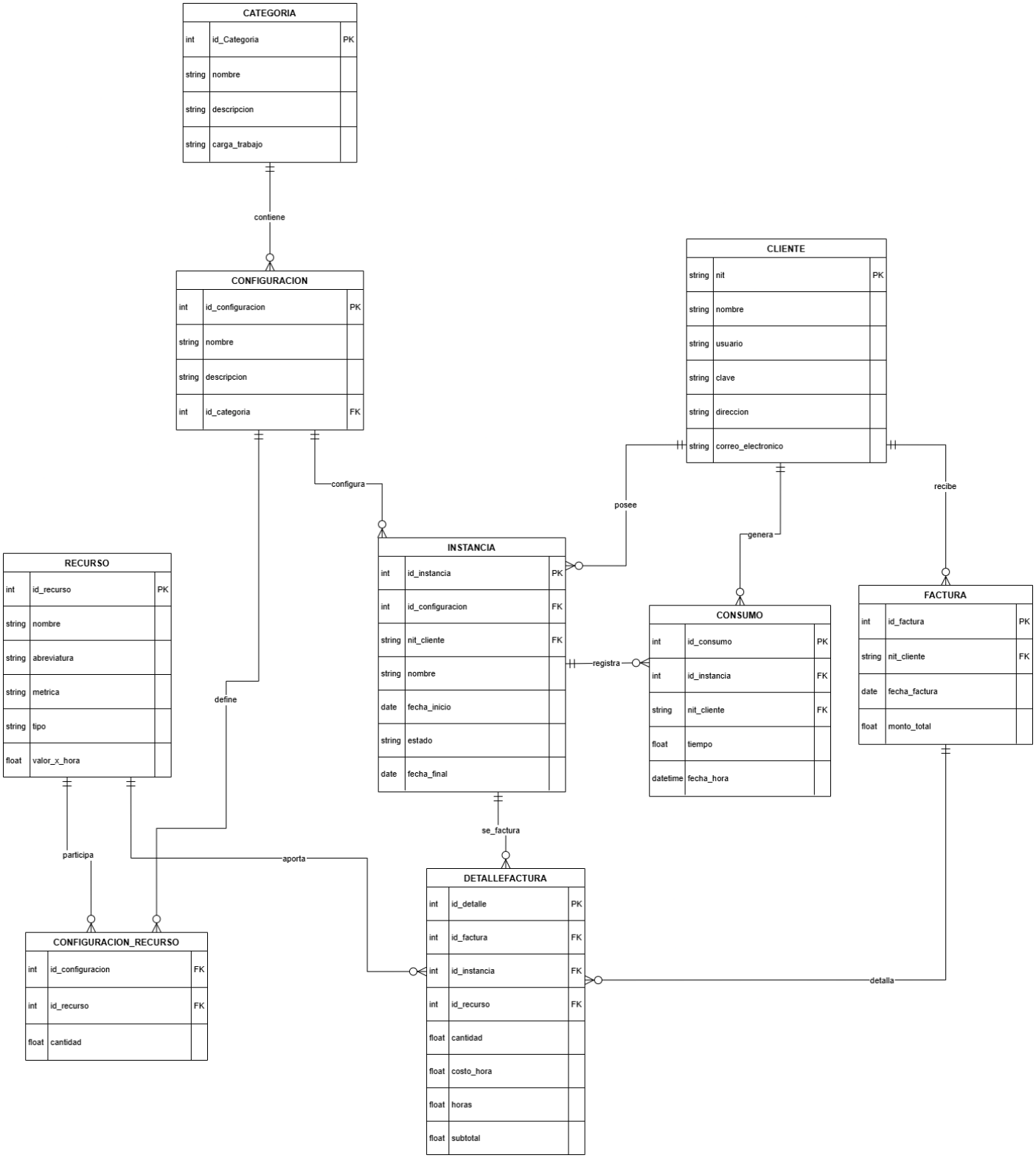
El uso de estructuras XML como base de datos demostró ser una alternativa viable para la persistencia de datos en entornos de simulación, manteniendo la trazabilidad de cada entidad y facilitando la generación de reportes. Asimismo, la automatización del cálculo de montos y generación de facturas contribuye a una gestión más eficiente, reduciendo errores humanos y optimizando los tiempos operativos.

Finalmente, este proyecto refuerza la importancia de integrar diversas tecnologías y paradigmas de programación para resolver problemáticas reales, demostrando cómo el diseño estructurado y la orientación a objetos permiten construir sistemas robustos y mantenibles.

Referencias

- Date, C. J. (1991). *An Introduction to Database Systems*. Addison-Wesley Publishing Company, Inc.
- Django Software Foundation. (2024). *Django Documentation*. Recuperado de https://docs.djangoproject.com/
- Palet, M. (2024). *Flask API Best Practices: Building Scalable Python Backends*. RealPython. Recuperado de https://realpython.com/
- Universidad de San Carlos de Guatemala. (2025). *Guía del Proyecto 3 – Introducción a la Programación y Computación 2*. Facultad de Ingeniería, Escuela de Ciencias y Sistemas.
- Van Rossum, G., & Drake, F. L. (2023). *Python 3.11 Documentation*. Python Software Foundation. Recuperado de https://docs.python.org/3/

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



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