

---

## PROYECTO 2

---

202109715 – Pablo Josué Barahona Luncey

### 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.

### Palabras clave

nube, clientes, organizaciones, configuraciones, estados

### 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.*

### Keywords

*cloud, clients, organizations, configurations, states*

## Introducción

Este programa se realizó a través del lenguaje de programación Python. El objetivo de este proyecto fue crear y analizar los consumos que realizan algunos clientes en algunos servicios que presta una empresa este se realizó haciendo un backend y un frontend, el cual es amigable con el usuario

## Desarrollo del tema

Este programa se desarrolló en el lenguaje de programación Python, el cual es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Una de sus ventajas es que es un lenguaje interpretado, lo que significa que no necesita compilar el código fuente para poder ejecutarlo.

Para este proyecto se tiene como objetivo principal la utilización de entornos de virtualización como también un back end realizado con flask y también un front end realizado con django los cuales se deben de asociar para poder completar el programa y se haga un entorno visual amigable con el usuario.

Para su realización también se usaron clases listas y base de Datos creada también con un archivo xml. En este programa únicamente se podrá visualizar el backend ya que no hay tiempo para realizar el frontend.

Se realizó el protocolo http el cual se describe a continuación:

HTTP, de sus siglas en inglés: "Hypertext Transfer Protocol", es el nombre de un protocolo el cual nos permite realizar una petición de datos y recursos, como pueden ser documentos HTML. Es la base de cualquier intercambio de datos en la Web, y un protocolo de estructura cliente-servidor, esto quiere decir que una petición de datos es iniciada por el elemento que recibirá los datos (el cliente), normalmente un navegador Web. Así, una página web completa resulta de la unión de distintos sub-documentos recibidos, como, por ejemplo: un documento que especifique el estilo de maquetación de la página web (CSS), el texto, las imágenes, vídeos, scripts, etc...

Clientes y servidores se comunican intercambiando mensajes individuales (en contraposición a las comunicaciones que utilizan flujos continuos de datos). Los mensajes que envía el cliente, normalmente un navegador Web, se llaman peticiones, y los mensajes enviados por el servidor se llaman respuestas.

Diseñado a principios de la década de 1990, HTTP es un protocolo ampliable, que ha ido evolucionando con el tiempo. Es lo que se conoce como un protocolo de la capa de aplicación, y se transmite sobre el protocolo TCP, o el protocolo encriptado TLS (en-US), aunque teóricamente podría usarse cualquier otro protocolo fiable. Gracias a que es un protocolo capaz de ampliarse, se usa no solo para transmitir documentos de hipertexto (HTML), si no que además, se usa para transmitir imágenes o vídeos, o enviar datos o contenido a los servidores, como en el caso de los formularios de datos. HTTP puede incluso ser utilizado para transmitir partes de documentos, y actualizar páginas Web en el acto.

También se utiliza el sistema de seguridad CORS:

El Intercambio de Recursos de Origen Cruzado (CORS) es un mecanismo que utiliza cabeceras HTTP adicionales para permitir que un user agent (en-US) obtenga permiso para acceder a recursos seleccionados desde un servidor, en un origen distinto (dominio) al que pertenece. Un agente crea una petición HTTP de origen cruzado cuando solicita un recurso desde un dominio distinto, un protocolo o un puerto diferente al del documento que lo generó.

Un ejemplo de solicitud de origen cruzado: el código JavaScript frontend de una aplicación web que es localizada en `http://domain-a.com` utiliza XMLHttpRequest para cargar el recurso `http://api.domain-b.com/data.json`.

Por razones de seguridad, los exploradores restringen las solicitudes HTTP de origen cruzado iniciadas dentro de un script. Por ejemplo, XMLHttpRequest y la API Fetch siguen la política de mismo-origen. Ésto significa que una aplicación que utilice esas APIs XMLHttpRequest sólo puede hacer solicitudes HTTP a su propio dominio, a menos que se utilicen cabeceras CORS.

El W3C Grupo de Trabajo de Aplicaciones Web recomienda el nuevo mecanismo de Intercambio de Recursos de Origen Cruzado (CORS, por sus siglas en inglés). CORS da controles de acceso a dominios cruzados para servidores web y transferencia segura de datos en dominios cruzados entre navegadores y servidores Web. Los exploradores modernos utilizan CORS en un contenedor API (como XMLHttpRequest o Fetch) para ayudar a mitigar los riesgos de solicitudes HTTP de origen cruzado.

Se realizaron muchos métodos pero la mayoría son muy parecidos ya que al utilizar un back end como lo es con flash se usan métodos muy parecidos

visualmente ya que se repiten patrones y se automatizan los objetivos.

Gracias a eso nosotros podemos iniciar nuestro background con un código único y en una ruta específica para evitar aqueos en la misma red o en la misma computadora, esto es muy importante ya que en las empresas o en cosas personales estamos muy expuestos a ser hackeados y es por ello que se utilizan los back end y las rest api, entre otros.

El término API es una abreviatura de Application Programming Interfaces, que en español significa interfaz de programación de aplicaciones. Se trata de un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones de software a través de un conjunto de reglas.

Así pues, podemos hablar de una API como una especificación formal que establece cómo un módulo de un software se comunica o interactúa con otro para cumplir una o muchas funciones. Todo dependiendo de las aplicaciones que las vayan a utilizar, y de los permisos que les dé el propietario de la API a los desarrolladores de terceros.

Para iniciar el primer método es el método para leer archivos el cual lee archivos xml el formato de archivos xml se podrá visualizar en la siguiente imagen, para leer eso se utiliza la librería minidom la cual permite leer correctamente un archivo xml y transformarlo en un dato limpio.



```
1 <?xml version="1.0"?>
2 <archivoConfiguraciones>
3   <listaRecursos>
4     <recurso id="1000">
5       <nombre>Nucleo de unidad central de procesamiento</nombre>
6       <abreviatura>CPU</abreviatura>
7       <metrica>Ghz </metrica>
8       <tipo>Procesamiento</tipo>
9       <valorXhora>0.02</valorXhora>
10    </recurso>
11    <recurso id="1001">
12      <nombre>Memoria de Acceso Aleatorio</nombre>
13      <abreviatura>RAM</abreviatura>
14      <metrica>GB </metrica>
15      <tipo>Almacenamiento</tipo>
16      <valorXhora>0.03</valorXhora>
17    </recurso>
18    <recurso id="1002">
19      <nombre>Unidad de estado solido</nombre>
20      <abreviatura>SSD</abreviatura>
21      <metrica>GB</metrica>
22      <tipo>Almacenamiento</tipo>
23      <valorXhora>0.03</valorXhora>
24    </recurso>
25    <recurso id="1003">
26      <nombre>Unidad de disco duro</nombre>
27      <abreviatura>DD</abreviatura>
28      <metrica>GB</metrica>
29      <tipo>Almacenamiento</tipo>
30      <valorXhora>0.01</valorXhora>
31    </recurso>
32  </listaRecursos>
```

Para poder entender estos conceptos, se realizó un diagrama de clases, el cual puede encontrarse en la parte de Anexos como “Imagen 1”.

Para entender el programa desde un punto de vista técnico, iniciamos presentando el programa y el creador, en este caso soy yo (Pablo Josué Barahona Luncey) y las respectivas identificaciones.

Para realizar los endpoints se usaron las opciones @approute la cual es nativa de flask. A través de eso los endpoints se utilizan y se pueden visualizar a través de diferentes programas como postman insomnia o una extensión de visual studio la cual se llama Thunder. el siguiente método es la carga de consumo la cual se puede cargar a través de una base de Datos o es de la extensión zander la cual es la que yo utilizo esto va a permitirnos poder utilizarlos las configuraciones iniciales y podernos ejecutar el programa correctamente.

Luego tenemos diferentes tipos de endpoints, uno de ellos es consultar categorías los cuales va a consultar y va a revisar todas las categorías que se

tienen guardadas en las bases de Datos y las va a devolver en un formato json con el nombre de las mismas para esto se utilizó el método get de flask.

luego de eso también tenemos el consultar configuraciones que se utilizó el mismo método get también tenemos el consultar clientes el crear el consultar instancias y el consultar recursos para terminar así obteniendo todos los Datos que se leen en nuestro archivo.

luego de eso también tenemos otro tipo de métodos que es para poder crear y para eso se utiliza el método post el cual crea recursos también podemos crear categorías crear configuraciones crear clientes o crear instancias esto nos va a servir para poder crear e ingresar Datos manualmente y también lo podemos utilizar para actualizar nuestra base de Datos la cual está en un archivo xml.

Por último también para para usar el back end tenemos el método post y él endpoint generar factura, este endpoint se encarga de poder generar y crear todas las archivos que se necesitan para poder terminar la evaluación del programa al generar una factura podemos visualizar la persona que utilizó esa factura y los gastos que hizo en la misma.

## Referencias bibliográficas

- Barahona, P. (5 de septiembre de 2022). *Lucidchart*.  
Obtenido de Lucidchart:  
[https://lucid.app/documents?canInterruptWithPayments=1#/templates?folder\\_id=home](https://lucid.app/documents?canInterruptWithPayments=1#/templates?folder_id=home)
- Desarrollo Web. (5 de septiembre de 2022).  
*desarrolloweb.com*. Obtenido de  
desarrolloweb.com:  
<https://desarrolloweb.com/articulos/1325.php>
- Desconocido. (5 de septiembre de 2022). *Listas enlazadas simples*. Obtenido de Listas enlazadas simples:  
[http://www.it.uc3m.es/java/2011-12/units/pilas-colas/guides/2/guide\\_es\\_solution.html#:~:text=Una%20lista%20enlazada%20simple%20es,los%20elementos%20de%20la%20misma](http://www.it.uc3m.es/java/2011-12/units/pilas-colas/guides/2/guide_es_solution.html#:~:text=Una%20lista%20enlazada%20simple%20es,los%20elementos%20de%20la%20misma).
- Stefunko, M. (2018). *Acción humana*. Guatemala: The Scholar's edition.

