

Las APIs y su aprovechamiento en soluciones IoT e IIoT

Objetivo

Lograr que el estudiante no solo comprenda este concepto (API) sino que lo haga en el contexto de IoT, comprobando el valor que tienen en las soluciones IoT o IIoT.

Conocimientos previos

Una API

Una API, o Interfaz de Programación de Aplicaciones (por sus siglas en inglés, Application Programming Interface). Es la clave de la programación distribuida en la nube. Imaginemos que se trata de una caja o sistema que tiene entradas, realiza tareas usando los datos de esas entradas y entrega resultados a través de unas salidas, pero lo relevante es que esa caja tiene usualmente una URL, igual que la tiene una página Web. Pero esa URL va acompañada de comandos que obedecen a reglas muy propias para recibir y entregar la información. Usualmente una API es un producto muy maduro, muy bien documentado, con muy buen soporte técnico y de recursos para garantizar que muchos usuarios puedan ser atendidos con alta calidad. Por eso es común escuchar que una API es “un conjunto de reglas y protocolos que permite que diferentes programas se comuniquen entre sí a través de la web”. También es importante aclarar que una URL no caracteriza una API, más bien las API pueden usar URL y comandos para ofrecer servicios. No siempre una API usa URL pero si siempre una API pertenece a un servidor, por ejemplo Google brinda API que se usan por comandos de javascript o Python en APIs como Google Maps API, Google Drive API, Google Calendar API, Google Cloud Speech-to-Text API. Algunas cualidades claves son:

- Abstracción de funcionalidad: Una API oculta los detalles internos de cómo funciona un sistema o servicio y proporciona una forma clara y definida para que otros programas accedan a su funcionalidad. Entonces no importa en qué lenguaje está escrita una API.
- Comunicación estándar: Las APIs establecen reglas y formatos estandarizados para la comunicación, lo que facilita que las aplicaciones se entiendan entre sí.
- Reutilización de código: Las APIs permiten a los desarrolladores utilizar funciones y servicios existentes en lugar de reinventar la rueda. Esto promueve la reutilización de código y acelera el desarrollo de software.
- Seguridad y control: Las APIs pueden controlar qué datos o funciones están disponibles para otros programas y establecer niveles de seguridad para proteger los recursos y la información.
- Independencia de plataforma: Las APIs pueden permitir la interoperabilidad entre sistemas en diferentes plataformas, lo que facilita la integración de aplicaciones en diversas tecnologías.

- Las API usan protocolo HTTP.
- Documentación: Las APIs suelen estar acompañadas de documentación que describe cómo usarlas, qué parámetros se requieren y qué respuestas se pueden esperar.

Una API también puede ser vista como una interfaz de programación en el sentido de que ofrece herramientas para crear más funcionalidades.

Ejemplos comunes: Las APIs se encuentran en todas partes, desde las redes sociales (como la API de Twitter o la API de Facebook, la de GPT) hasta los sistemas operativos (como la API de Windows o la API de Android) y los servicios en la nube (como la API de Google Maps o la API de AWS). En el caso de Google surgen algunas confusiones de términos, por ejemplo “SpreadsheetApp” es una clase que google ofrece en JavaScript. Pero Google también ofrece la API de Google Sheets que es una API RESTful que permite a las aplicaciones externas interactuar con hojas de cálculo de Google, cómo leer y escribir datos en ellas.

Las partes de una URL para html y API

Primer caso (la URL de una página web)

<https://sites.google.com/e3t.uis.edu.co/homeroortega/>

Segundo caso (la URL de API):

https://v6.exchangerate-api.com/v6/API_KEY

Donde API_KEY es un código que cada usuario tiene para acceder a unos recursos puntuales de una API.

Tercer caso (la URL de una API con órdenes - un endpoint):

https://v6.exchangerate-api.com/v6/API_KEY/algo

Donde “algo” se refiere a comandos que la API tiene predefinidos para prestar diferentes servicios.

Reto general para esta práctica

En la práctica anterior se había implementado una solución de neveras que reportan la falta de productos a un broker EMQX gracias a los sensores que se tienen instalados. Las tiendas interesadas se suscriben a la información que esas neveras emiten, cada nevera se diferencia porque tiene un tópico diferente. En la práctica pasada, la tienda solo estaba interesada en surtir huevos.

Continuaremos con esa idea, pero vamos a agregar más y más elementos. Por ahora, vamos a aprovechar una API que ofrece el valor del cambio del dólar en el instante dado. La idea consiste en que el suscriptor conoce el precio del producto (los huevos) en dólares. Será necesario traducir ese precio a la moneda del país donde se encuentra la nevera, en

este caso a pesos colombianos. El problema es que la traducción a la moneda debe considerar el valor real y actual de la moneda en los mercados internacionales. Por eso es necesario acudir a una API que nos garantice la información correcta del precio del dólar.

Tarea 1. Realizar pruebas con Python a una API que ofrece el precio del dólar respecto al peso colombiano

Esta tarea busca conocer cómo funcionan las API, que las podemos invocar por comando para obtener servicios. Para ser coherentes con el reto de este taller, justamente le haremos pruebas a una API que brinda el precio del dólar en el momento. Por ahora lo haremos desde nuestro código de python.

Paso 1: Visita el sitio de [API ExchangeRate](#) lee y observa todo para comprender de qué se trata una API, revisa los planes de pagos.

Escribe aquí tus impresiones sobre lo que una API puede llegar a ofrecer. Imagina que le estás contando a un amigo que has aprendido cómo funciona el negocio de las API y explícale lo que te impresiona de las API, las oportunidades de negocio que hay en ellas.

Las API (Interfaz de Programación de Aplicaciones) es una herramienta increíble que permite a diferentes apps comunicarse efectivamente entre sí, cambiando datos y funcionalidades. Algo curioso de las APIs es que funcionan como “puentes” que conectan sistemas complejos y plataformas en tiempo real. Por ejemplo, una API puede permitir que una aplicación de banca consulte en tiempo real los tipos de cambio de divisas, o que una app de clima obtenga datos actualizados sobre las condiciones meteorológicas.

El negocio de las API es enorme porque permite a las empresas ofrecer servicios específicos sin tener que construir soluciones completas desde cero. Esto abre muchas oportunidades. Las compañías pueden monetizar sus datos y servicios, creando planes de suscripción para acceder a ellos.

Paso 2. Obtén una “API Key” apropiada para obtener el precio del dólar en el momento.

API_KEY obtenida

The screenshot shows the ExchangeRate-API dashboard. The left sidebar contains navigation links: Dashboard, API Keys, Request Usage, Account Details, Manage Plan, Billing, Invoices, Documentation, Docs Overview, Supported Currencies, Standard API, Requests, and Product Homepage. The main content area is titled 'API Access' and displays the API Key: **ec49b72870e83327952f63f8**. Below this, it shows an example request: <https://v6.exchangerate-api.com/v6/ec49b72870e83327952f63f8/latest/USD>. The 'API Request Quota Usage' section features a progress bar for 'API request quota available' at 30,000. To the right, a table shows request counts: 0 for today, this month, and last month, with a monthly quota reset on the 5th. The bottom section is titled 'Account & Plan Summary'.

API request quota available	API requests today	API requests this month	API requests last month	Monthly quota reset
30 000	0	0	0	5th

Paso 3. Obtén el End-Point para la consulta del precio del dólar frente al peso en el momento. La API ExchangeRate ofrece instrucciones para obtenerlo en: Menú > Documentation > Fast Start > Standard endpoint

El endpoint de la API ExchangeRate tiene la forma: <https://v6.exchangerate-api.com/v6/YOUR-API-KEY/algo>.

Dónde, "YOUR-API-KEY" es la API_KEY, "algo" son comandos que dependen de lo que requieras.

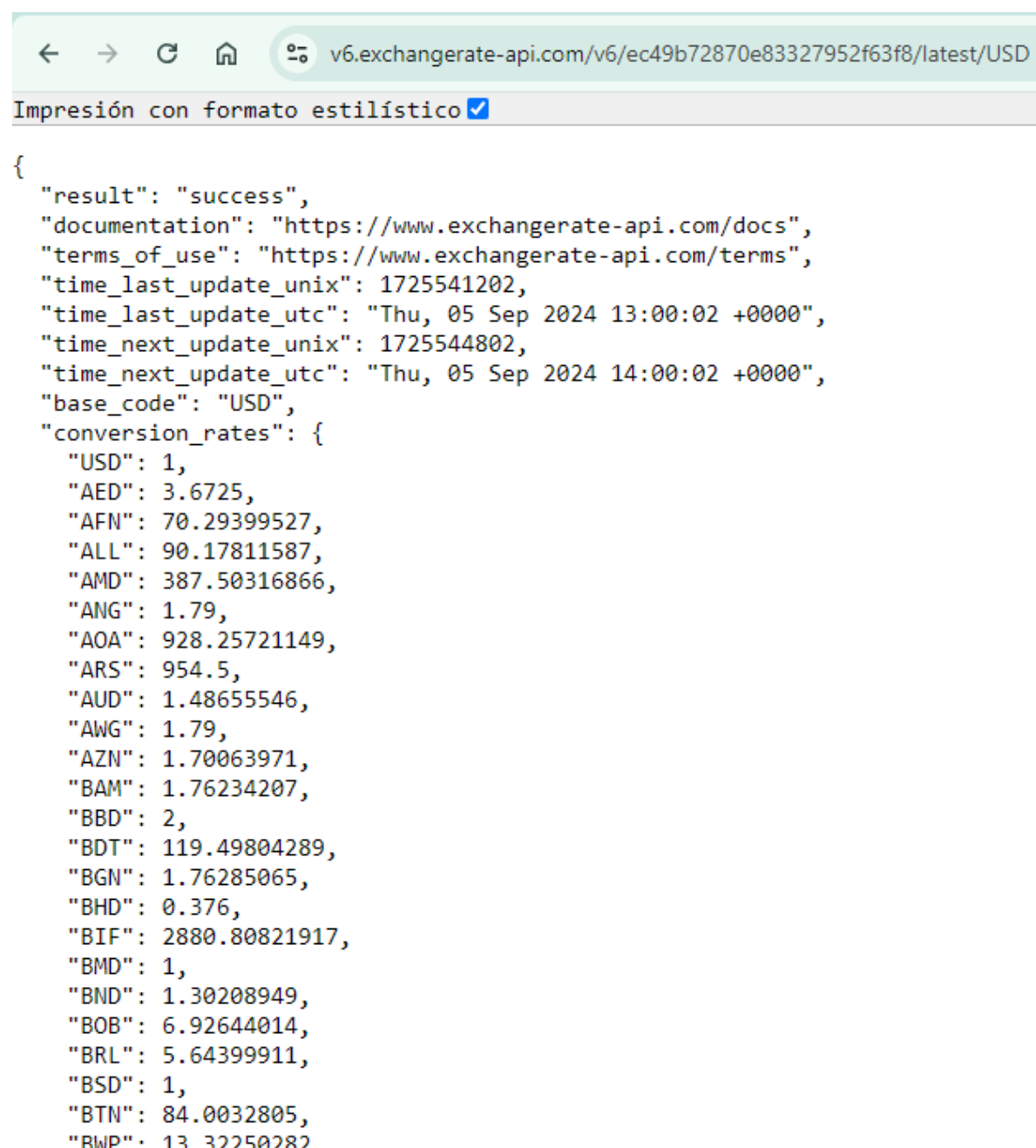
Tu End-Point para consultar el precio del dólar en el momento actual:

<https://v6.exchangerate-api.com/v6/ec49b72870e83327952f63f8/latest/USD>

Paso 4. Comprueba que tu End-Point funciona en un navegador.

Como las API usan protocolo HTTP, es posible escribir el End-Point en un navegador como una dirección URL y obtener un resultado

Resultado obtenido en un navegador al usar el End-Point



The screenshot shows a web browser with the address bar containing the URL: `v6.exchangerate-api.com/v6/ec49b72870e83327952f63f8/latest/USD`. Below the address bar, there is a checkbox labeled "Impresión con formato estilístico" which is checked. The main content area displays a JSON response from the API:

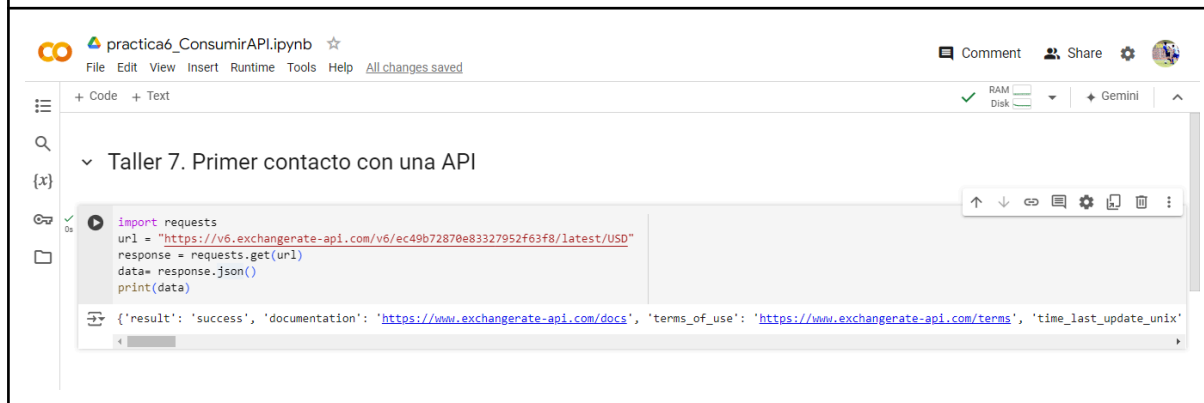
```
{
  "result": "success",
  "documentation": "https://www.exchangerate-api.com/docs",
  "terms_of_use": "https://www.exchangerate-api.com/terms",
  "time_last_update_unix": 1725541202,
  "time_last_update_utc": "Thu, 05 Sep 2024 13:00:02 +0000",
  "time_next_update_unix": 1725544802,
  "time_next_update_utc": "Thu, 05 Sep 2024 14:00:02 +0000",
  "base_code": "USD",
  "conversion_rates": {
    "USD": 1,
    "AED": 3.6725,
    "AFN": 70.29399527,
    "ALL": 90.17811587,
    "AMD": 387.50316866,
    "ANG": 1.79,
    "AOA": 928.25721149,
    "ARS": 954.5,
    "AUD": 1.48655546,
    "AWG": 1.79,
    "AZN": 1.70063971,
    "BAM": 1.76234207,
    "BBD": 2,
    "BDT": 119.49804289,
    "BGN": 1.76285065,
    "BHD": 0.376,
    "BIF": 2880.80821917,
    "BMD": 1,
    "BND": 1.30208949,
    "BOB": 6.92644014,
    "BRL": 5.64399911,
    "BSD": 1,
    "BTN": 84.0032805,
    "BWP": 13.32250282
```

Paso 5. Escribe el código en Python para obtener en una variable lo mismo que has obtenido en un navegador.

Escribe el siguiente código en Python para lanzar una solicitud GET a la API y obtener la misma respuesta del paso anterior pero por comando

```
import requests
url = "https://v6.exchangerate-api.com/v6/4995be34b73555d65adb3c08/latest/USD"
response = requests.get(url)
data= response.json()
print(data)
```

Resultado al correr el código



Paso 6. Complementa el código anterior para obtener el valor del dólar en pesos colombianos

Código para obtener el valor del dólar en pesos colombianos en el momento

```
import requests

# URL de la API para obtener los valores de cambio a partir del USD
url =
"https://v6.exchangerate-api.com/v6/ec49b72870e83327952f63f8/latest/USD"

# Realiza la solicitud GET a la API
response = requests.get(url)

# Convierte la respuesta en formato JSON
data = response.json()

# Extrae el valor del dólar en pesos colombianos
dolar_a_cop = data['conversion_rates']['COP']

# Imprime el valor del dólar en pesos colombianos
print(f"El valor actual del dólar en pesos colombianos es: {dolar_a_cop}")
```

Resultado al correr el código



El valor actual del dólar en pesos colombianos es: 4180.26238719

+ Code



Tarea 2. Complementa el código del taller (taller 5) anterior para que la información que se registra en Google Sheet incluya el precio del dólar como si fuese un sensor adicional.

Escribe aquí el nuevo código del suscriptor. Es decir el código del usuario remoto que envía los datos al servidor.

```
import gspread
from oauth2client.service_account import ServiceAccountCredentials
import random
from datetime import datetime
from google.colab import drive
import time # Para retrasos en la simulación
import requests
from google.colab import files
import time # Importar la librería time

# Subir el archivo JSON desde la computadora
files.upload()

# Montar Google Drive
#drive.mount('/content/drive')

# Ruta al archivo JSON en Google Drive
json_keyfile_name = 'practica-5-v2-d2d3049d3f0a.json'

#json_keyfile_name = '/practica-5-v2-d2d3049d3f0a.json'

# Configuración de la autenticación para acceder a Google Sheets
scope = ["https://spreadsheets.google.com/feeds",
"https://www.googleapis.com/auth/drive"]
creds =
ServiceAccountCredentials.from_json_keyfile_name(json_keyfile_name,
scope)
client = gspread.authorize(creds)

# Accede a la hoja de cálculo
spreadsheet_url =
"https://docs.google.com/spreadsheets/d/1xStIjQV--TuZWE0jL5cMnK9HneVw
m5BC42n7lgLP4Lc/edit?usp=sharing"
sheet = client.open_by_url(spreadsheet_url).get_worksheet(1) # Hoja
```

```

2 (índice 1)

# URL de la API para obtener los valores de cambio a partir del USD
url =
"https://v6.exchangerate-api.com/v6/ec49b72870e83327952f63f8/latest/USD"

# Función para obtener el valor actual del dólar y su conversión a COP
def obtener_valores_monedas():
    response = requests.get(url)
    data = response.json()
    dolar_a_cop = data['conversion_rates']['COP']
    return 1.0, dolar_a_cop # Retorna el valor del dólar en USD y COP

# Función para generar un ID único
def unico_ID():
    return datetime.now().strftime("%Y-%m-%d%H%M%S%f") # Basado en la fecha y hora actual

# Función para simular los datos del sensor
def simular_dato_sensor(nombre_sensor, ubicacion, tipo, unidades_medida):
    id_unico = unico_ID()
    valor = round(random.uniform(10.0, 100.0), 2) # Valor aleatorio entre 10.0 y 100.0
    fecha = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    # Obtener valores de monedas
    usd, cop = obtener_valores_monedas()

    # Agregar los valores de USD y COP al final
    return [fecha, id_unico, nombre_sensor, ubicacion, tipo, unidades_medida, valor, usd, cop]

# Función para enviar datos simulados al Google Sheets
def enviar_datos_al_sheet(datos_sensores):
    for dato in datos_sensores:
        sheet.append_row(dato)
        print(f"Datos simulados agregados: {dato}")
        time.sleep(3) # Esperar 3 segundos antes de agregar el siguiente dato

# Asegurarse de que los encabezados de las nuevas columnas estén agregados
sheet.update_cell(1, 8, "USD") # Columna H (8) para USD
sheet.update_cell(1, 9, "COP") # Columna I (9) para COP

# Ejecución de la simulación
try:
    while True:
        # Simulación de datos de sensores con las columnas adicionales USD y COP
        datos_sensores = [
            simular_dato_sensor("Sensor_Temperatura", "Ubicación 1", "Temperatura", "Celsius"),

```



```

        simular_dato_sensor("Sensor_Humedad", "Ubicación 2",
"Humedad", "Porcentaje"),
        simular_dato_sensor("Sensor_Presion", "Ubicación 3",
"Presión", "Pascal")
    ]
    enviar_datos_al_sheet(datos_sensores)
    time.sleep(3)
except KeyboardInterrupt:
    print("Simulación finalizada.")

# Mostrar los datos actuales en la hoja de cálculo
datos_en_sheet = sheet.get_all_records()
print("\nDatos actuales en la hoja de cálculo:")
for fila in datos_en_sheet:
    print(fila)

```

docs.google.com/spreadsheets/d/1xStjJQV--TuZWE0jL5cMnK9HneVwm58C42n71gLP4Lc/edit?gid=2129246401#gid=2129246401

Practica_5_v2

Archivo Editar Ver Insertar Formato Datos Herramientas Extensiones Ayuda

100% 123 Predet... 10

	A	B	C	D	E	F	G	H	I	J
1	Fecha	ID	Nombre_Sensor	Ubicación	Tipo	Unidades	Valor	USD	COP	
2	2024-09-05 14:41:43	20240905144143907400	Sensor_Temperatura	Ubicación 1	Temperatura	Celsius	63.09	1	4180.262387	
3	2024-09-05 14:41:44	20240905144144098064	Sensor_Humedad	Ubicación 2	Humedad	Porcentaje	99.36	1	4180.262387	
4	2024-09-05 14:41:44	20240905144144277087	Sensor_Presion	Ubicación 3	Presión	Pascal	60	1	4180.262387	
5	2024-09-05 14:41:57	20240905144157035075	Sensor_Temperatura	Ubicación 1	Temperatura	Celsius	61.84	1	4180.262387	
6	2024-09-05 14:41:57	20240905144157219425	Sensor_Humedad	Ubicación 2	Humedad	Porcentaje	57.07	1	4180.262387	
7	2024-09-05 14:41:57	20240905144157389115	Sensor_Presion	Ubicación 3	Presión	Pascal	13.37	1	4180.262387	
8	2024-09-05 14:42:10	20240905144210222079	Sensor_Temperatura	Ubicación 1	Temperatura	Celsius	23.78	1	4180.262387	
9	2024-09-05 14:42:10	20240905144210585407	Sensor_Humedad	Ubicación 2	Humedad	Porcentaje	50.94	1	4180.262387	
10	2024-09-05 14:42:10	20240905144210948384	Sensor_Presion	Ubicación 3	Presión	Pascal	88.94	1	4180.262387	
11	2024-09-05 14:42:23	20240905144223919000	Sensor_Temperatura	Ubicación 1	Temperatura	Celsius	82.12	1	4180.262387	
12	2024-09-05 14:42:24	20240905144224132344	Sensor_Humedad	Ubicación 2	Humedad	Porcentaje	29.63	1	4180.262387	
13	2024-09-05 14:42:24	20240905144224306701	Sensor_Presion	Ubicación 3	Presión	Pascal	67.64	1	4180.262387	
14	2024-09-05 14:42:37	20240905144237071977	Sensor_Temperatura	Ubicación 1	Temperatura	Celsius	90.56	1	4180.262387	
15	2024-09-05 14:42:37	20240905144237242734	Sensor_Humedad	Ubicación 2	Humedad	Porcentaje	48.9	1	4180.262387	
16	2024-09-05 14:42:37	20240905144237605314	Sensor_Presion	Ubicación 3	Presión	Pascal	56.16	1	4180.262387	
17	2024-09-05 14:46:26	20240905144626216781	Sensor_Temperatura	Ubicación 1	Temperatura	Celsius	19.33	1	4180.262387	
18	2024-09-05 14:46:26	20240905144626405630	Sensor_Humedad	Ubicación 2	Humedad	Porcentaje	29.66	1	4180.262387	
19	2024-09-05 14:46:26	20240905144626587983	Sensor_Presion	Ubicación 3	Presión	Pascal	15.58	1	4180.262387	
20	2024-09-05 14:46:39	20240905144639294641	Sensor_Temperatura	Ubicación 1	Temperatura	Celsius	56.32	1	4180.262387	

+ Hoja 1 CosumoAPI

Conclusiones: explica si has aprendido a valorar las API, como algo que impactará tu futuro. Explica por ejemplo si ves posible crear soluciones donde el cómputo está distribuido por el mundo (por toda la internet), si es posible aprovechar los desarrollos de otras entidades para potenciar tus soluciones.

He aprendido sobre la importancia de las API para el desarrollo de aplicaciones y como estas son clave en la actualidad. Las API no solo facilitan el acceso a recursos y datos distribuidos en todo el mundo, sino que también permiten que los desarrolladores construyan soluciones más rápidas y eficientes sin tener que reinventar la rueda. Este tipo de servicios aprovecha la infraestructura y datos de otras empresas para potenciarlos a los desarrollos propios.

Me queda claro que las API son una forma de computación distribuida, donde diferentes servicios alrededor del mundo están interconectados, cada uno especializado en realizar tareas específicas (como el caso de la API que se usó en este taller para conocer el valor del dólar en pesos colombianos). Esta capacidad para distribuir el cómputo y el almacenamiento de datos a través de Internet nos abre las puertas para construir soluciones globales que sean altamente escalables y eficientes.

Las oportunidades al usar APIs son increíblemente infinitas, ya que puedo imaginar crear soluciones que combinen múltiples servicios en la nube, desde sistemas de pago hasta servicios de machine learning, todos orquestados a través de APIs.