



**Universidad Tecnológica
de Chihuahua**

TECNOLOGÍAS DE LA INFORMACIÓN: DESARROLLO DE SOFTWARE
MULTIPLATAFORMA

APLICACIONES WEB ORIENTADAS A SERVICIOS

Reporte de investigación de nombres y funciones de
diferentes API

Docente: Addai Guerrero

Oscar Ariel Quintana Merino
TID41M

Funciones de APIs

Geolocalización:

- **Definición:** Las APIs de geolocalización permiten a las aplicaciones web acceder a la ubicación física del dispositivo del usuario.
- **Funcionamiento:** Utilizan la información de GPS, direcciones IP o puntos de acceso Wi-Fi para determinar la ubicación.
- **Implementación:** JavaScript y la API Geolocation del navegador.

Ejemplo de Código:

```
1  if ("geolocation" in navigator) {  
2    navigator.geolocation.getCurrentPosition(function(position) {  
3      console.log("Latitud: " + position.coords.latitude + ", Longitud: " + position.coords.longitude);  
4    });  
5  }  
6
```

Redes Sociales:

- **Definición:** Las APIs de redes sociales permiten interactuar con plataformas como Facebook, Twitter o Instagram para compartir contenido y obtener datos de perfiles.
- **Funcionamiento:** Se autentican mediante tokens de acceso y realizan solicitudes HTTP a las APIs de las redes sociales.
- **Implementación:** Varios lenguajes, pero JavaScript es común para la integración en aplicaciones web.

Ejemplo de Código:

```
// Utilizando la API de Twitter para publicar un tweet  
const Twitter = require('twitter');  
const client = new Twitter({ /* Credenciales */ });  
  
client.post('statuses/update', { status: '¡Mi primer tweet desde una app!' }, function(error, tweet, response) {  
  if (!error) {  
    console.log(tweet);  
  }  
});
```

E-commerce:

- **Definición:** Las APIs de comercio electrónico permiten la integración de tiendas en línea y sistemas de pago en aplicaciones web.
- **Funcionamiento:** Facilitan la consulta de productos, carritos de compra y procesos de pago.

- **Implementación:** Lenguajes como PHP, JavaScript y Python para el desarrollo de aplicaciones de comercio electrónico.

Ejemplo de Código:

```
// Integración de carrito de compra utilizando una API de e-commerce
// Ejemplo con Shopify
const apiKey = 'tu-api-key';
const storeDomain = 'tudominio.myshopify.com';

fetch(`https://${storeDomain}/admin/api/2021-07/carts.json`, {
  method: 'POST',
  headers: {
    'X-Shopify-Access-Token': apiKey,
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({ line_items: [{ variant_id: 123456789, quantity: 2 }] }),
})
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error(error));
```

Bases de Datos:

- **Definición:** Las APIs de bases de datos permiten interactuar con sistemas de gestión de bases de datos (DBMS) para almacenar, recuperar y administrar datos.
- **Funcionamiento:** Utilizan consultas SQL u otros protocolos de comunicación para acceder a los datos en la base de datos.
- **Implementación:** Lenguajes como Python (para Django o Flask), JavaScript (Node.js), Java, etc.

Ejemplo de Código:

```
// Ejemplo de conexión y consulta a una base de datos MySQL en Node.js
const mysql = require('mysql');

const connection = mysql.createConnection({
  host: 'localhost',
  user: 'usuario',
  password: 'contraseña',
  database: 'basededatos',
});

connection.connect();

connection.query('SELECT * FROM usuarios', function (error, results, fields) {
  if (error) throw error;
  console.log(results);
});

connection.end();
```

Protocolos de Comunicación:

- **Definición:** Las APIs de protocolos de comunicación permiten la interacción con protocolos como HTTP, WebSocket o MQTT para el intercambio de datos en tiempo real o sincronización.
- **Funcionamiento:** Facilitan la creación de solicitudes, manejo de respuestas y comunicación en tiempo real a través de los protocolos.
- **Implementación:** Lenguajes como JavaScript (para aplicaciones web), Python, Java, etc.

Ejemplo de Código:

```
// Ejemplo de uso de la API Fetch para hacer una solicitud HTTP
fetch('https://api.example.com/data', {
  method: 'GET',
  headers: {
    'Content-Type': 'application/json',
  },
})
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.error(error));
```

Plataformas en Línea:

- **Definición:** Las APIs de plataformas en línea permiten interactuar con servicios en la nube como Google Drive, Dropbox o Firebase para almacenar, acceder o procesar datos en línea.
- **Funcionamiento:** Se autentican mediante tokens de acceso y permiten realizar operaciones en los servicios en línea.
- **Implementación:** JavaScript, Python, Java, etc.

Ejemplo de Código:

```
// Ejemplo de uso de la API de Google Drive para subir un archivo
// (requiere autenticación y configuración previa)
const fileMetadata = {
  name: 'mi-archivo.txt',
};
const media = {
  mimeType: 'text/plain',
  body: fs.createReadStream('mi-archivo.txt'),
};
drive.files.create({
  resource: fileMetadata,
  media: media,
  fields: 'id',
}, function (err, file) {
  if (err) {
    console.error(err);
  } else {
    console.log('Archivo subido con éxito, ID:', file.data.id);
  }
});
```

Plataformas de Streaming:

- **Definición:** Las APIs de plataformas de streaming permiten la transmisión de contenido multimedia en tiempo real, como video y audio.
- **Funcionamiento:** Facilitan la codificación, transmisión y decodificación de contenido multimedia en aplicaciones de streaming.
- **Implementación:** Lenguajes como JavaScript (para reproductores web), C++ (para codificación de video), etc.

Ejemplo de Código:

```
// Ejemplo de integración de un reproductor de video en una aplicación web
// utilizando la API de YouTube
<iframe width="560" height="315" src="https://www.youtube.com/embed/VIDEO_ID" frameborder="0" allowfullscreen></iframe>
```

API de Pagos (Payment Gateway API):

- **Definición:** Las APIs de pagos permiten la realización de transacciones financieras en línea, como procesamiento de tarjetas de crédito y pagos electrónicos.
- **Funcionamiento:** Facilitan la conexión con servicios de procesamiento de pagos y proporcionan funciones para realizar cobros y gestionar transacciones financieras.
- **Implementación:** JavaScript, PHP, Ruby, entre otros.

Ejemplo de Código:

```
// Ejemplo de integración de la API de Stripe para procesar pagos en una tienda en línea
const stripe = require('stripe')('tu-clave-privada');

stripe.charges.create({
  amount: 2000,
  currency: 'usd',
  source: 'tok_mastercard', // Tarjeta de prueba
  description: 'Compra de producto',
}, function(err, charge) {
  if (err) {
    console.error(err);
  } else {
    console.log('Pago exitoso, ID de transacción:', charge.id);
  }
});
```

API de Traducción (Translation API):

- **Definición:** Las APIs de traducción permiten la traducción automática de texto o contenido entre diferentes idiomas.
- **Funcionamiento:** A través de estas APIs, se envía texto a una plataforma de traducción que devuelve el texto traducido en el idioma deseado.
- **Implementación:** JavaScript, Python, Ruby, etc.

Ejemplo de Código:

```
// Ejemplo de uso de la API de Google Cloud Translation para traducir texto
// (requiere configuración y autenticación)
const { Translate } = require('@google-cloud/translate').v2;
const translate = new Translate();

async function translateText() {
  const [translation] = await translate.translate('Hello, world!', 'es');
  console.log(`Traducción: ${translation}`);
}

translateText().catch(console.error);
```

APIs de Inteligencia Artificial (AI):

- **Definición:** Las APIs de inteligencia artificial permiten integrar funcionalidades de aprendizaje automático, reconocimiento de patrones y procesamiento de lenguaje natural en una aplicación web.
- **Funcionamiento:** Permiten enviar datos a modelos de IA preentrenados o personalizados para obtener predicciones y análisis.
- **Implementación:** Lenguajes como Python (para entrenamiento de modelos), JavaScript (para integración en la aplicación web), etc.

Ejemplo de Código:

```
// Integración de una API de reconocimiento facial (ejemplo con Microsoft Azure Face API)
const axios = require('axios');

const imageUrl = 'URL_DE_LA_IMAGEN';
const subscriptionKey = 'TU_CLAVE_DE_SUSCRIPCIÓN';

axios.post('https://westcentralus.api.cognitive.microsoft.com/face/v1.0/detect?returnFaceAttributes=age,gender,emotion', { url: imageUrl }, {
  headers: { 'Ocp-Apim-Subscription-Key': subscriptionKey },
})
.then(response => console.log(response.data))
.catch(error => console.error(error));
```