Concepto de API

Interfaces de programación de aplicaciones (APIs)

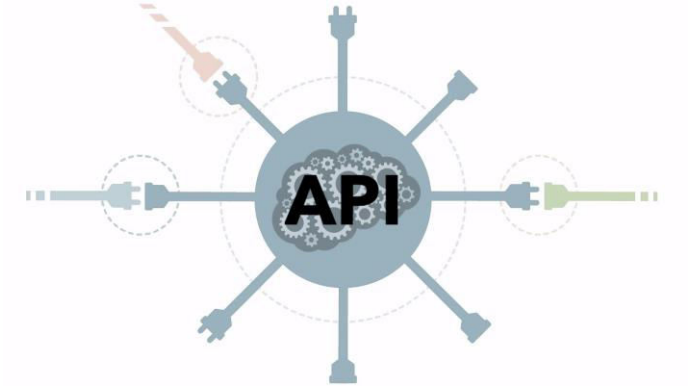
Hasta ahora tenemos:

- Capacidad de intercambiar datos entre aplicaciones utilizando el protocolo HTTP
- Un modo de representar estructuras complejas de datos para poder enviar y recibir datos entre las aplicaciones a través de XML o JSON

El paso siguiente consiste en definir y documentar “contratos” entre aplicaciones usando estas técnicas. Estos contratos reciben el nombre de **Interfaces de Programación de Aplicaciones** (***A****pplication* ***P****rogramming* ***I****nterfaces*), o **API**s

Interfaces de programación de aplicaciones (APIs)

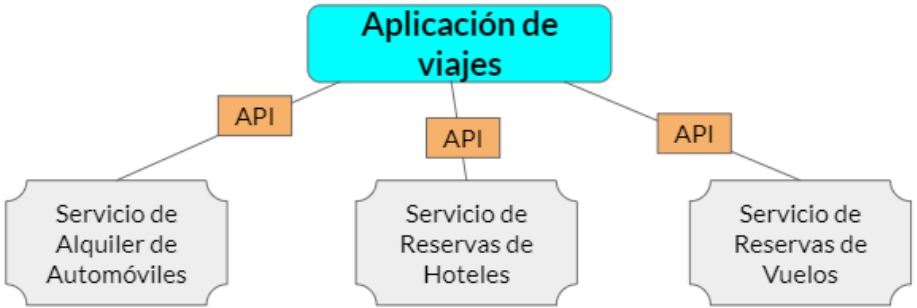
Cuando se utiliza una API, un programa crea un conjunto de servicios disponibles para que los usen otras aplicaciones y publica las APIs (“reglas”) que deben seguirse para acceder a los servicios proporcionados por el programa, permitiendo que las aplicaciones se conecte, se comuniquen y compartan información.



Arquitectura Orientada a Servicios (SOA)

- Cuando los programas acceden a servicios proporcionados por otros, se utiliza un planteamiento llamado **Arquitectura Orientada a Servicios** o **SOA**

- Un planteamiento SOA es aquel en el que una aplicación principal usa los servicios de otras aplicaciones



Tipos de APIs

- **APIs de servicios web**: permiten el intercambio de información entre un servicio web y una aplicación
  - SOAP
  - **REST**

- APIs basadas en bibliotecas
- APIs basadas en clases
- APIs de sistemas operativos

API REST

- La mayoría de las empresas utilizan API REST para crear servicios web
- Funcionan de manera similar a una web

- Las llamadas a la API se implementan como peticiones HTTP
- Las operaciones que nos permiten obtener datos en HTTP son **GET** y **POST**
- Ejemplo: https://docs.openaq.org/

Métodos HTTP: GET y POST

- El concepto **GET** consiste en obtener información del servidor. Esto puede realizarse a través de un navegador.

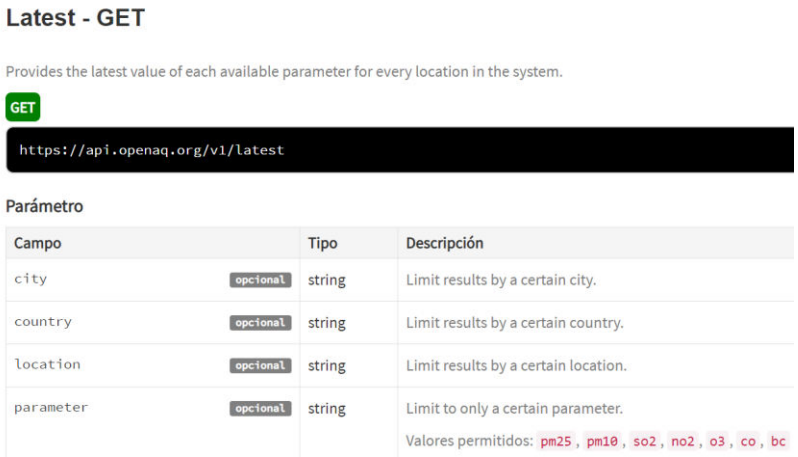
- El concepto **POST** consiste en **enviar** información desde el cliente para que sea procesada y actualice o agregue información en el servidor (por ejemplo, el envío de un formulario). No puede realizarse con un navegador

- Existen herramientas, como **Postman**, que permiten realizar peticiones HTTP, y probar cualquier API REST

Llamada a una API

- En primer lugar, se recomienda leer detenidamente la **documentación** de la API
- Ejemplo:

Para llamar a esta API utilizaremos una URL “fija”, a la cual se le introducen los parámetros



Latest - GET

Provides the latest value of each available parameter for every location in the system.

GET

```
https://api.openaq.org/v1/latest
```

Parámetro

Campo		Tipo	Descripción
city	opcional	string	Limit results by a certain city.
country	opcional	string	Limit results by a certain country.
location	opcional	string	Limit results by a certain location.
parameter	opcional	string	Limit to only a certain parameter. Valores permitidos: pm25 , pm10 , so2 , no2 , o3 , co , bc

https://api.openaq.org/v1/latest?city=Madrid&parameter=no2

Se utiliza el interrogante (?) para introducir parámetros, separados por &

¿Por qué utilizar APIs?

- Permiten **ofrecer** datos a otras aplicaciones o desarrolladores en un formato estándar
- Permiten **consumir** datos de otras aplicaciones
- Existen infinidad de proveedores de APIs (Facebook, Youtube, Amazon, Twitter, etc.)
- Directorio de proveedores de APIs:

https://www.programmableweb.com/apis/directory

¡GRACIAS!

¿Preguntas?