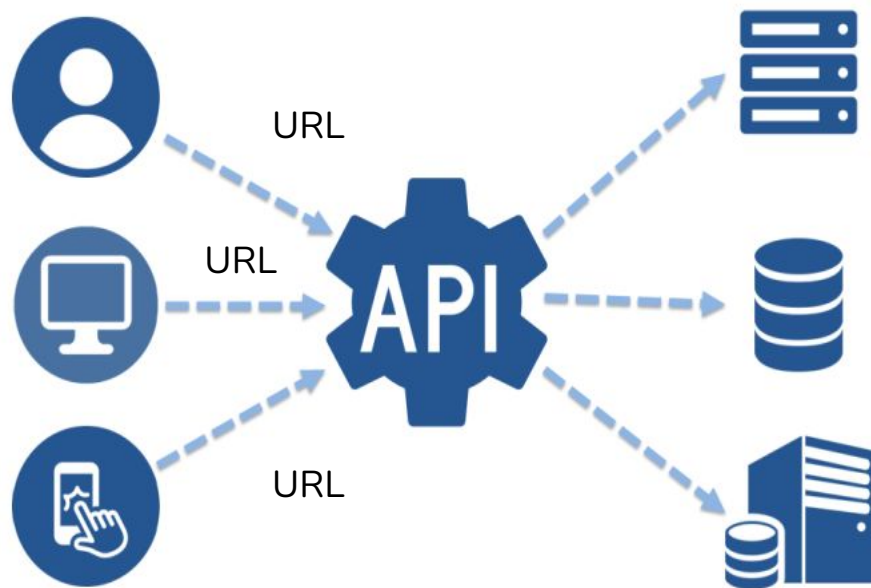


API (Application Programming Interfaces)

DEV.F
DESARROLLAMOS(PERSONAS);

dev

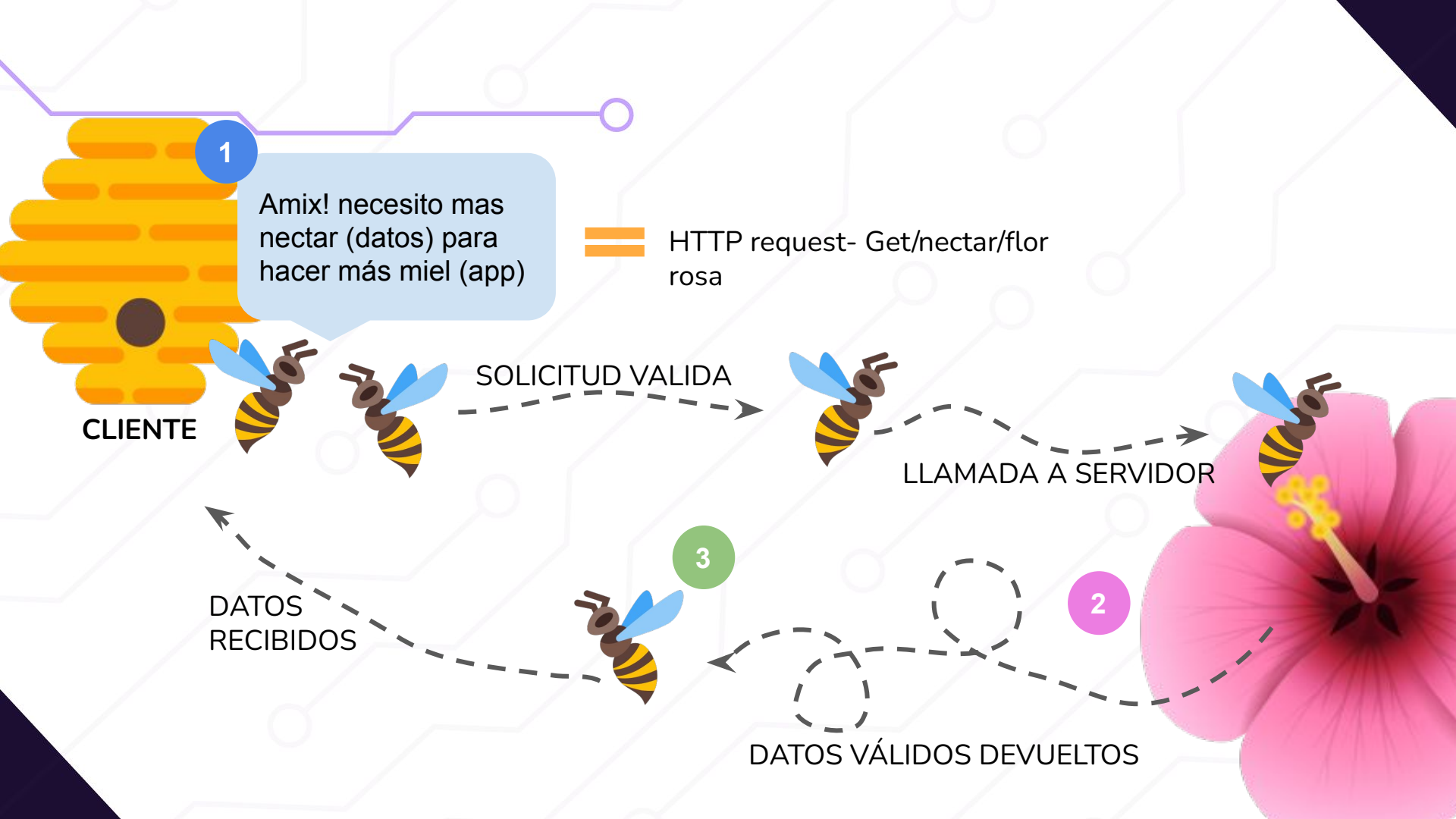


API

(Application Programming Interface)

Una interfaz de programación de aplicaciones (API) es un conjunto de herramientas, definiciones y protocolos que se utiliza para integrar los servicios y el software de aplicaciones.

Es lo que permite que sus productos y servicios se comuniquen con otros, sin tener que diseñar permanentemente una infraestructura de conectividad nueva.





The diagram illustrates the API request-response cycle using a bee analogy. A purple line at the top represents the flow of the process. Below it, three numbered steps are shown: 1. REQUEST (blue circle), 2. RECEIVE (pink circle), and 3. RESPONSE (green circle). Each step is followed by a description of the action. The background features a light purple circuit-like pattern with diagonal lines and circles.

1

REQUEST

La llamada a API es iniciada por la aplicación cliente a través de una solicitud HTTP

2

RECEIVE

La abeja trabajadora actúa como una API, que irá a la flor (servidor) para obtener néctar (datos)

3

RESPONSE

La API transfiere la información solicitada de vuelta a la solicitud de la app, usualmente en formato JSON

REPASO

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.



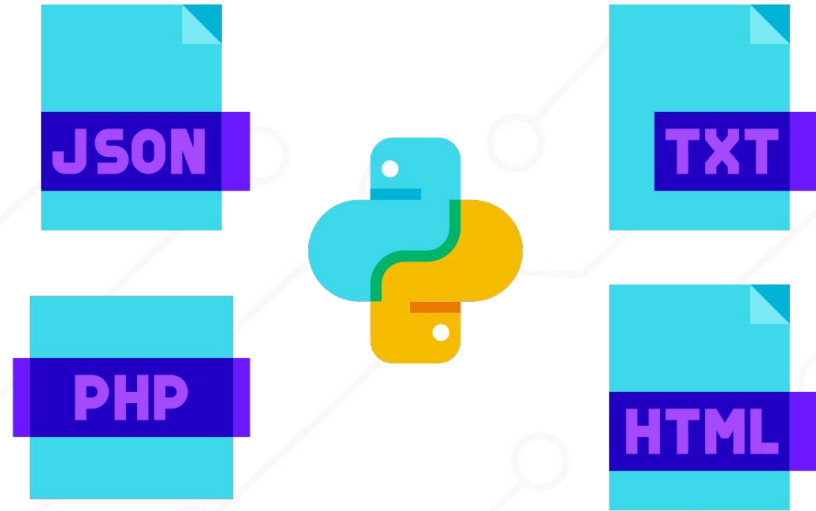
API SOAP

SOAP es un protocolo estándar que se creó originalmente para posibilitar la comunicación entre las aplicaciones que se diseñan con diferentes lenguajes y en distintas plataformas. Como es un protocolo, impone reglas integradas que aumentan la complejidad y la sobrecarga, lo cual puede retrasar el tiempo que tardan las páginas en cargarse. Sin embargo, estos estándares también ofrecen normas integradas que pueden ser ideales para el sector empresarial. Los estándares de cumplimiento integrados incluyen la seguridad, la atomicidad, la uniformidad, el aislamiento y la durabilidad

API REST

Una API de REST, o API de RESTful, es una interfaz de programación de aplicaciones (API o API web) que se ajusta a los límites de la arquitectura REST y permite la interacción con los servicios web de RESTful.

REST no es un protocolo ni un estándar, sino más bien un conjunto de límites de arquitectura. Los desarrolladores de las API pueden implementarlo de distintas maneras.





SOAP (Simple Object Access Protocol)

Protocol

Function driven

Requires advanced security, but can define it, too

Needs more bandwidth

Stricter rules to follow

Cannot use REST

Only works in XML

Supports HTTP and SMTP protocols

REST (REpresentational State Transfer)

Architecture

Data driven

Relies on underlying network which can be less secure

Only needs minimum bandwidth

Easier for developers to suggest recommendations

Can use SOAP

Works in different data formats such as HTML, JSON, XML, and plain text

Only requires HTTP



API

Abstracción de
funciones y
procedimientos

REST

Lógica de restricciones y
recomendaciones bajo
la cual podemos
construir una API
(arquitectura)

RESTful API

Es una API ya
implementa utilizando
la lógica REST

Principales características de una API REST

Los objetos REST son manipulados a través de una URI (Uniform Resource Identifier). Esta URI (endpoint) hace de identificador único de cada recurso del sistema REST, por lo que no puede ser compartida por más de un recurso. La estructura básica de una URI es la siguiente:

```
{protocolo}://{hostname}:{puerto}/{ruta del recurso}?{parámetros de filtrado (opcional)}
```

El nombre de la URI no debe contener palabras que impliquen acciones, por lo que deben evitarse los verbos en su construcción. Además, las URI siguen una jerarquía lógica de capas que permite ordenar los recursos y englobar las distintas funcionalidades entre sí.

O bien agregando un cuerpo a la llamada REST en cualquier tipo de formato, siendo los más usados JSON y XML.

¿Qué es un endpoint?

En pocas palabras, un **endpoint** es un extremo de un canal de comunicación. Cuando una API interactúa con otro sistema, los puntos de contacto de esta comunicación se consideran endpoints.

Para las API, un endpoint puede incluir una URL de un servidor o servicio. Cada endpoint es la ubicación desde la cual las API pueden acceder a los recursos que necesitan para llevar a cabo su función.

Las API funcionan mediante "request" y "responses". Cuando una API solicita información de una aplicación web o un servidor web, recibirá una respuesta. El lugar donde las API envían solicitudes y donde vive el recurso se denomina punto final.

The logo consists of the text "DEV.F." in a bold, white, sans-serif font. The period after "F" is replaced by a small grid of four squares, with the top-right square missing, creating a pixelated effect.

DEV.F.

MICRO SERVICIO

dev

Microservicios

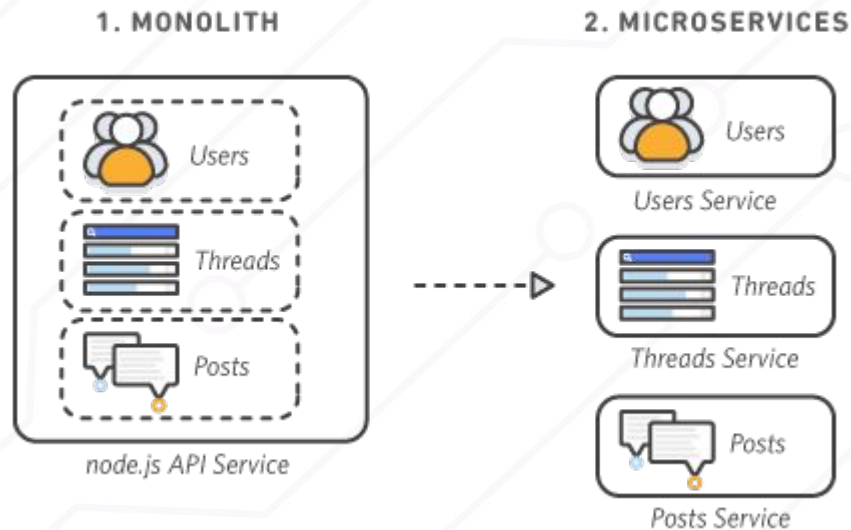
¿Qué son o cómo se comen esas cosas que nos está enseñando el sensei Dani?

Los microservicios son un enfoque arquitectónico y organizativo para el desarrollo de software donde el software está compuesto por pequeños servicios independientes que se comunican a través de API bien definidas. Los propietarios de estos servicios son equipos pequeños independientes.

Las arquitecturas de microservicios hacen que las aplicaciones sean más fáciles de escalar y más rápidas de desarrollar. Esto permite la innovación y acelera el tiempo de comercialización de las nuevas características.

Con una arquitectura de microservicios, una aplicación se crea con componentes independientes que ejecutan cada proceso de la aplicación como un servicio. Estos servicios se comunican a través de una interfaz bien definida mediante API ligeras.

- Autónomos
- Especializados
- Ágiles
- Fácil escalamiento
- Implementación “sencilla”
- Resistentes
- Código reutilizable
- Libertad tecnológica



Y como me dijo el del transporte público

hasta aquí llegamos joven



DEV.F.
DESARROLLAMOS(PERSONAS);

dev