API
(Application Programming Interfaces)

DEV.F:
DESARROLLAMOS(PERSONAS);

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.

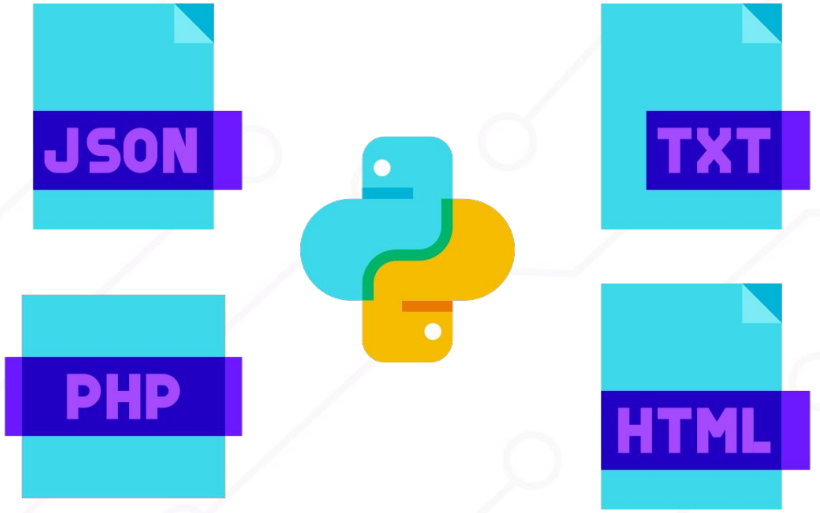
API SOAP

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)	REST (REpresentational State Transfer)
Protocol	Architecture
Function driven	Data driven
Requires advanced security, but can define it, too	Relies on underlying network which can be less secure
Needs more bandwidth	Only needs minimum bandwidth
Stricter rules to follow	Easier for developers to suggest recommendations
Cannot use REST	Can use SOAP
Only works in XML	Works in different data formats such as HTML, JSON, XML, and plain text
Supports HTTP and SMTP protocols	Only requires HTTP

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.

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

¿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.

MICRO SERVICIO

DEV.F

Microservicios

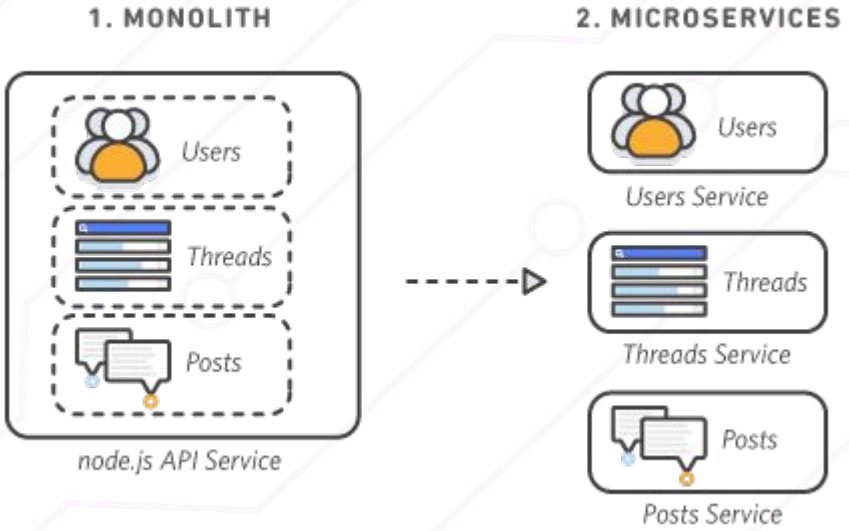
¿Qué son o cómo se comen esas cosas que nos está enseñando el sensei Dani?

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);

PERO INICIAMOS CON LA PRÁCTICA

DEV.F::
DESARROLLAMOS(PERSONAS);