API's Y SERVICIOS WEB. CONCEPTOS.

INTERFAZ.

- Capa de abstracción para que 2 sistemas se comuniquen.
- Una capa de abstracción permite la interacción con un sistema sin necesidad de conocer que ocurre por debajo.
- Ejemplos:
    - Volante – Ruedas, no es necesario conocer el mecanismo interno que sincroniza el movimiento del volante con el giro de las ruedas.
    - Para la identificación en un sitio web, solo es necesario escribir usuario y contraseña, no que procesos ocurren durante la identificación.

API.

- Un API es una interfaz que permite que se comuniquen aplicaciones entre sí y compartan datos, bajo ciertas normas, estándares o protocolos.
- La filosofía es aprovechar el desarrollo de otros para una aplicación.
- Pueden ser:
    - **Públicas.**
        - Cualquier persona puede acceder a una API y consultar la información.
    - **Privadas.**
        - Requieren autorización y autenticación.
        - **Token.**
            - Objeto que contiene todos los datos de una autenticación.
            - Al acceder de nuevo, o al solicitar información adicional, el servidor revisará si el token sigue vigente y no pedirá una nueva identificación.
            - Formato de token más común para REST es JWT (JSON Web Token).
    - **Locales**.
        - Se ejecutan en el mismo entorno.
        - Ejemplo:
            - Aplicación del móvil que se comunica con la API de vibración del móvil.
    - **Remotas.**
        - Se consumen datos de una aplicación que está en otro lugar.
        - Utilizan servicios web.
- **Ejemplos API's.**
    - Uso de los mapas de Google.
    - Pasarelas de pago en comercios electrónicos.

ARQUITECTURA DE SOFTWARE.

- Forma en que está diseñado un sistema.

- Como están organizados sus componentes.
- Como se comunican entre sí.
- Qué funciones cumplen éstos.

SERVICIO WEB.

- Sistema que permite la comunicación entre equipos que estén en una red.
- Intercambio automático de datos entre 2 aplicaciones basado en normas.
- Estos sistemas deben seguir los mismo estándares y protocolos de comunicación como HTTP.
- Tipos:
    - Basados en protocolo SOAP.
    - Basados en protocolo o arquitectura REST:
        - A estos servicios web se les denomina Restful.
- Necesario:
    - **Una red**.
        - Si se comunican 2 aplicaciones de una misma empresa, organización, etc., se usa una red local.
        - Si se comunican 2 aplicaciones de distintas empresas u organizaciones, etc., se usa conexiones cifradas https vía Internet.
    - **Un identificador único del servicio**.
        - URL endpoint. (Nombre del servidor y una ruta).
        - **endopoint (punto final).**
            - Similar a la de un sitio web, es una URL de una API que se encarga de contestar a una petición y que permite a un cliente de un servicio web acceder a un recurso proporcionado por una aplicación web desde una API.
    - **Idioma de comunicación común**.
        - Debe ser conocido por las 2 partes:
        - Los más utilizados son:
            - XML.
            - JSON.
    - **PARTES**:
        - Cliente Web Service o Service Consumer Proxy.
            - Inicia la llamada o conexión para utilizar o consumir un servicio web.
        - Servicio o Servidor Web Service o Service Provider.
            - Recibe la llamada y suministra un servicio.
            - Debe existir antes para que los clientes pueden solicitarlo.
            - Debe dar los datos de conexión y definir qué cosas podemos preguntar (esto se incluye en un fichero WDSL para SOAP).

SERVICIO WEB REST Y JSON.

- Es una arquitectura de software para crear web services.
- Mejora aspectos de SOAP.
- Inconveniente, es un conjunto de recomendaciones que puede dar lugar a distintas interpretaciones.
- Diferencias con SOAP:
    - **Formato de datos**:
        - Objeto JSON en REST y etiquetas o marcas XML en SOAP.
    - **Cabecera**.
        - En JSON no tiene cabecera o envoltorio por lo que en el mensaje van los datos. Lo que simplifica el desarrollo y acelera la conexión. (Postal).
        - En XML es obligatorio envolver el fichero con soap-envelope. (<soap: envelope> … </soap: envelope>). (Carta dentro de un sobre).
    - **Cantidad de URL Endpoint**.
        - **JSON**:
            - Varias, tantas como acciones u operaciones (recursos) más GET, POST, PUT y DELETE.
        - **XML**:
            - Una sólo para todas alas acciones u operaciones más POST.
    - **Fichero de descripción del lenguaje**.
        - **JSON**.
            - No es obligatorio, es sólo una recomendación, por lo que muchos servicios restful no incluyen este tipo de ficheros.
            - Herramientas para servicios web restful:
                - **Swagger.**
                    - Conjunto de herramientas de software de código abierto para diseñar, construir, documentar, y utilizar servicios web RESTful.
                    - Normas *openAPIs.org.*
                - **RAML.**
                    - RESTful API Modeling Language (RAML) es un lenguaje de modelado para definir APIs RESTful con una sintaxis sencilla y comprensible.
                    - Normas *raml.org.*
                - **json-schema,**
                    - JSON Schema establece un conjunto de reglas que modelan y validan una estructura de datos.
                    - Equivaldría a XSD SCHEMA para XML.
                - **WADL**
                    - Web Application Description Language.

            - Es un XML que sirve para describe Servicios HTTP, normalmente Servicios REST.
        - **MS Entity framework.**
            - Genera archivos EDMX.
    - **XML.**
        - Obligatorio un fichero WSDL por formar parte del estándar.
        - Desde el esquema del modelo de datos se genera automáticamente el WSDL, que contiene los esquemas XSD.
- **Utilización preferente.**
    - **REST.**
        - Comunicaciones entre partes de un mismo sistema.
        - Consumo masivo de datos vía http.
        - Aplicaciones que tiene una parte de servidor y otra de cliente como Webs, aplicaciones móviles y microservicios.
        - Ventajas:
            - Popular entre programadores por ser usado en desarrollo web y JavaScript.
            - Rápida puesta en funcionamiento.
            - Mejor rendimiento para web, aplicaciones móviles e Internet de las cosas.
            - Usado en backend y frontend.
        - Ejemplos:
            - Aplicaciones móviles->Twitter / X (lectura de tweets).
            - Chrome ->LinkedIn (lectura de contactos).
    - **SOAP.**
        - Comunicaciones transaccionales entre empresas, ya que:
            - Siempre hay un documento de especificaciones WSDL.
            - Fácil control de cambios y versiones.
            - Fácil testeo y seguridad.
        - Ejemplos:
            - Magento-> Santander (Pago con tarjeta de crédito).
            - SAP -> Agencia Tributaria. (Facturas).

DESARROLLAR UNA API REST.

- Consultar recursos a través de su URI, URL, etc.
- Códigos de estado o respuesta del servidor al contestar cuando se consulta una API y así, conocer que ha ocurrido con nuestra petición:
    - **2xx.**
        - Correcto. Petición con éxito.
    - **3xx.**
        - Redirección.

    - **4xx.**
        - Errores. Por ejemplo, solicitud inválida a un recurso que no existe o para el que no se tiene autorización.
    - **5xx.**
        - Errores en el servidor.
- **Métodos http.**
    - **GET.**
        - Solicitar información.
    - **POST.**
        - Enviar nueva información, por ejemplo, enviar datos de registro.
    - **PUT.**
        - Actualizar información existente, por ejemplo, modificar o actualizar datos de registro.
    - **DELETE.**
        - Borrar información.
- **Formatos.**
    - La API's pueden devolver la información en distintos formatos:
        - XML.
        - JSON.
        - Texto plano.
- **Buenas prácticas.**
    - **HATEOAS.**
        - Las API's se autodescriben.
        - Cada recurso contiene información de cuál es el recurso siguiente o de la cantidad de recursos totales que hay.
    - **Seguridad.**
        - Protección de API's privadas.
    - **Checkear.**
        - Comprobar que las API's funcionen correctamente.
    - **Documentar**.
        - Las API's se hacen para que otros las consuman, por lo que debe estar bien documentado cómo funcionan y como utilizarlas.