**¿ Porque un Api Rest ?**

**Objetivo:**

Explicar porque se debe realizar un api rest

Diagrama

Descripción generada automáticamente

Imagen 1. *Modelo de Api Rest*

**Recursos necesarios:**

Modelo de Api Rest

**Pasos:**

1. **Definir el objetivo de un Api Rest**

El objetivo de un Api Rest es proporcionar una interfaz estándar para que los sistemas informáticos puedan comunicarse y compartir información de manera eficiente.

1. **Explicar que es un Api Rest**

Un **API REST** es una interfaz de programación de aplicaciones que se basa en el protocolo HTTP para permitir la comunicación entre diferentes sistemas y aplicaciones, utiliza una serie de reglas y convenciones para permitir que los sistemas se comuniquen a través de la web. Los recursos se identifican mediante URL únicas, y las operaciones se realizan a través de los métodos HTTP estándar como GET, POST, PUT, DELETE, etc. Cada una de estas acciones tiene una respuesta asociada en formato JSON, XML, HTML, entre otros. El Api Rest se deriva de la combinación de los términos **Api** ( forma en que diferentes aplicaciones o sistemas pueden interactuar entre sí) y **Rest** (estilo de arquitectura web que se utiliza para diseñar estas interfaces). Un API se puede utilizar para acceder a los datos y funcionalidades de una aplicación o servicio de terceros, sin necesidad de conocer todos los detalles internos de cómo se implementa esa funcionalidad y los servicios REST permiten una alta escalabilidad, ya que se basan en una arquitectura "stateless" (sin estado), lo que significa que cada solicitud se procesa de forma independiente, sin mantener información de estado entre solicitudes.

En resumen, un API REST es una interfaz de programación de aplicaciones que utiliza el protocolo HTTP para permitir que diferentes sistemas informáticos se comuniquen y compartan información de manera eficiente y estandarizada.

1. **Realización del modelo Api Rest**

Se debe crear el modelo Api Rest en un editor de imágenes (ver imagen 1).

El modelo de API REST es importante porque proporciona una arquitectura escalable, basada en estándares e independiente de la plataforma para desarrollar aplicaciones y servicios que se pueden integrar de manera eficiente en diferentes entornos. Este modelo facilita la comunicación entre sistemas, la independencia de plataforma, la escalabilidad y desarrollo modular (desarrollo de software que se basa en dividir una aplicación o sistema en módulos más pequeños e independientes). Estas características hacen que las API REST sean ampliamente utilizadas en el diseño de sistemas distribuidos y aplicaciones web.

1. **Explicación del modelo**

Las API REST se utilizan comúnmente en la creación de servicios web y aplicaciones móviles, permitiendo la comunicación entre clientes y servidores a través de internet utilizando el protocolo HTTP.

A continuación, explicaremos el modelo del api rest. En este modelo la comunicación se realiza de la siguiente manera:

**Cliente**: El cliente es la aplicación o sistema que envía una solicitud al API REST para obtener o enviar datos. Esto puede ser una aplicación web, una aplicación móvil u otra aplicación que requiere interactuar con el servidor a través del API REST.

**API REST:** El API REST es la interfaz de programación de aplicaciones que se expone en el servidor y define cómo las solicitudes y respuestas deben estructurarse para comunicarse con el cliente. El API REST recibe las solicitudes del cliente y las procesa, interactuando con la base de datos y aplicando la lógica de negocio necesaria para manejar la solicitud.

**Base de datos:** La base de datos es el sistema de almacenamiento de datos donde se almacenan los datos necesarios para que el API REST funcione. El API REST puede consultar, actualizar o modificar los datos en la base de datos según las solicitudes recibidas del cliente.

**Servidor:** El servidor es el sistema en el que se aloja el API REST. Es responsable de recibir las solicitudes del cliente, enrutarlas al API REST correspondiente, y recibir las respuestas del API REST para enviarlas de vuelta al cliente. El servidor puede realizar tareas adicionales, como la autenticación y autorización de las solicitudes del cliente antes de pasarlas al API REST.

**Respuestas:** Las respuestas del API REST incluyen la información solicitada o el resultado de las operaciones solicitadas, y se envían de vuelta al cliente a través del servidor. Las respuestas pueden estar en formatos comunes como JSON o XML, y contienen los datos solicitados o el resultado de las operaciones realizadas en la base de datos.