



Back end nivel intermedio

**Infraestructura
back end**

Para que puedas comprender de manera más adecuada el diseño e implementación de una infraestructura debes primero entender componentes, tareas, requerimientos y operaciones necesarias para dicha aplicación.

Según mencionan Martínez y Eterovic (2018), el término infraestructura en general es invisible para la gran mayoría de los usuarios. Es por ello por lo que se consideran distintos tipos de infraestructura de TI, dependiendo del enfoque del proceso de negocio.



Diseño de la infraestructura

La infraestructura de software se refiere al conjunto de elementos necesarios para que una aplicación funcione, esto considera el hardware, software, redes etc. Es decir, todo aquello que sea necesario para desarrollar, dar soporte y probar estos servicios y tener en producción el producto.

La definición de infraestructura que se utiliza se basa en un modelo de componentes no funcionales. Dentro de los muchos atributos no funcionales que existen, el modelo de componentes considera tres que afectan al diseño y configuración: la disponibilidad, la seguridad y el rendimiento.

La infraestructura de las tecnologías de la información (TI) abarca un mundo invisible de protocolos, redes y *middleware* que permiten que los sistemas informáticos de la empresa puedan trabajar juntos y faciliten el flujo eficiente de la información.

Según IBM (s.f.), hoy en día existen principalmente dos tipos de infraestructura:

1. La **infraestructura tradicional** en la cual las empresas son propietarias de todos los elementos como centros de datos e infraestructura de telecomunicaciones, los cuales son gestionados en sus propias instalaciones y con sus propios recursos. Este modelo de infraestructura se considera costoso por el espacio físico, gastos derivados del mantenimiento del equipamiento y energía, es decir, se requieren grandes inversiones iniciales. Una de las fortalezas de este modelo, debería ser seguridad de los datos, sin embargo, algunas organizaciones no invierten demasiado en este rubro a pesar del riesgo que esto conlleva.
2. La **infraestructura en la nube** se refiere a todos aquellos elementos que se pueden alquilar a un tercero. Una de las fortalezas que se tiene en este modelo, es que no se requieren grandes inversiones iniciales y que solo se paga por el espacio o capacidad utilizada, además que se puede incrementar o decrementar en cualquier momento. El gran inconveniente es que la portabilidad para migrar hacia otro proveedor de la nube es muy costosa y difícil de realizar.

El diseño de la infraestructura varía mucho según los requisitos del proyecto de software que se va a implementar y en el que se tienen que considerar diversos factores, por ejemplo:

- Tipo de disponibilidad necesaria.
- Tiempos de respuesta considerados como aceptables por sus usuarios o la gerencia.
- Número de usuarios que ingresarán convergentemente.
- Escalabilidad.
- Cantidad de datos que se necesitan almacenar a corto, mediano y largo plazo.

Para efectos de esta experiencia educativa, nos enfocaremos en la arquitectura cliente servidor, sobre la cual se basa el desarrollo web. En esta arquitectura, como su nombre lo indica, existen dos partes:

- 1) El **cliente** que es el que se ocupa de realizar las peticiones, recibir respuestas y presentarlas al usuario. Un cliente puede ser cualquier tipo de dispositivo capaz de realizar una petición, esto incluye smartphones y dispositivos inteligentes como bocinas, luces, refrigeradores, relojes, termostatos, etc.
- 2) El **servidor** es una aplicación diseñada para satisfacer las demandas de información que realizan los clientes. El servidor recibe las peticiones del cliente, las procesa y responde con la información solicitada.

Los servidores suelen realizar tareas complejas y especializadas, están intercomunicados con otros sistemas como bases de datos, servicios externos e internos (microservicios), servicios de correo electrónico, otros servidores web, etcétera.

Considerando esta arquitectura, es común que se dividan las responsabilidades y los equipos de desarrollo en front end y back end, donde los primeros se encargan de desarrollar el código de la interfaz del usuario que se ejecutará en un navegador y los segundos, el código con las reglas de negocio y las configuraciones del lado del servidor.



Cuando se planean este tipo de proyectos es necesario realizar un **diagrama de infraestructura**, consiste en un mapa visual que muestra cómo interactúan los diferentes componentes tecnológicos, hacia adentro y hacia afuera.

En estos diagramas se incluyen puntos de acceso, sistemas, aplicaciones, redes, bases de datos, máquinas virtuales, contenedores y seguridad con que se deberá contar.

Un diagrama de infraestructura es esencial al momento de realizar la planeación de la infraestructura y en muchos casos se convierte en la columna vertebral del área de TI ya que, gracias a él, los ingenieros y profesionales pueden entender mejor la infraestructura que se desea implementar para determinado proyecto o para saber con la que ya cuenta la empresa. Este mapa ofrece una excelente herramienta para poder saber qué se implementará, qué se debe administrar, mejorar y mantener seguro.

Así como en una construcción el encargado de realizar la obra requiere de un plano para conocer el lugar donde se colocarán los materiales, las empresas que tienen o desean implementar infraestructura tecnológica deben contar con un mapeo que les ayude a identificar con exactitud los errores y así tomar decisiones de mejora informadas y resolver las necesidades eficientemente.

Este tipo de herramientas no es exclusivo de grandes empresas o para proyectos complejos, también se presentan en empresas pequeñas o emergentes, donde la infraestructura es limitada y la diagramación abarca la representación de los contenidos de un producto digital y las relaciones entre estos. En algunas empresas y especialmente aquellas que pretenden tener algún tipo de certificación tipo ISO, se convierte en un documento obligatorio.

El siguiente es un ejemplo de un diagrama de infraestructura:

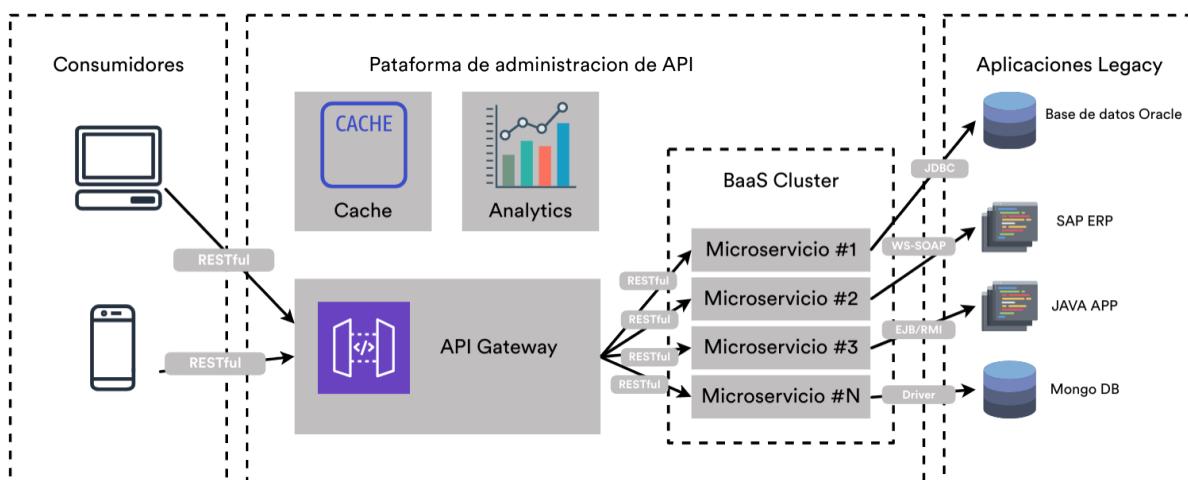


Diagrama 1.

Fuente: Hackmetrix. (2021). Recreado de <https://blog.hackmetrix.com/wp-content/uploads/2021/10/Asi-quedó.png> solo para fines educativos.

Implementación de la infraestructura

Como se mencionó anteriormente, para una infraestructura de software se deben incluir los componentes como el hardware, el software y las bases de datos que puedan trabajar en conjunto para gestionar las operaciones de la aplicación.

La implementación es la base primordial ya que permite optimizar sus recursos de manera que las aplicaciones o sistemas tengan un correcto funcionamiento y que estos también puedan atender de manera efectiva las reglas del negocio y sus requerimientos del mercado. Dicha implementación es la combinación de diferentes términos.

En cuestión de desarrollo existen tres principales componentes:

1. Servidor

2. Bases de datos

3. Middleware

El **middleware** es la conexión entre el *front end* y el *back end*. Este componente es una interfaz de comunicación que se encarga de llevar los recursos del back end hacia el usuario en el front end.

Ahora hay que recordar que el back end es la parte que se encuentra detrás de una aplicación la cual define el funcionamiento de una página o aplicación web, encargándose de la conexión a la base de datos y el servidor.

Desde la óptica del back end, un servidor esta, encargado de intercambiar información y almacenar el contenido como lo puede ser HTML, CSS y JavaScript, para que así pueda funcionar como un *host*.

Entonces, cuando por medio del navegador se desea ir a alguna página en específico, se le da la dirección al navegador; esto es una petición y este responderá yendo a buscar los archivos y mostrarlos como respuesta a la petición que se le hizo.

Un servidor utiliza distintos protocolos de comunicación para realizar una transferencia de datos, los protocolos principales son:

HTTP/HTTPS

SMTP

FTP

Estos protocolos permiten la comunicación del cliente para realizar peticiones sobre el servidor el cual almacena, procesa y envía como respuesta el resultado de las páginas o aplicaciones web.

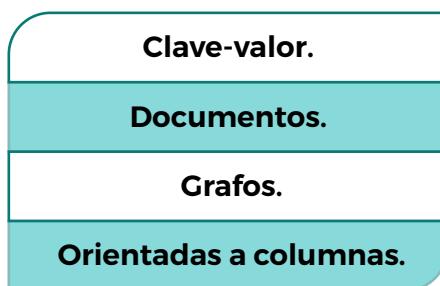
Bases de datos

Una **base de datos** es un conjunto de datos en el cual la información se puede encontrar de manera agrupada o estructurada.

Dentro de las bases de datos existe el modelo **relacional** el cual relaciona la información por medio de tablas, esta información se identifica con un número único el cual permite que se pueda relacionar con más datos.

También existen los modelos **no relacionales o NoSQL**, estos modelos son más flexibles para almacenar la información.

Algunos de los tipos de bases de datos NoSQL son:



Dentro de la infraestructura el diseño tradicional para las aplicaciones se centraba en una arquitectura monolítica, es decir, todo lo que componía una aplicación estaba contenido en la misma.

De acuerdo con RedHat (2019), este enfoque tiene sus desventajas ya que cuanto más grande es la aplicación más difícil es solucionar los problemas que se presentan y agregar funciones nuevas rápidamente.

Los microservicios dividen los elementos de las aplicaciones en partes pequeñas e independientes entre sí. Una de las ventajas de usar estos microservicios es la rapidez que se puede lograr al distribuir un software.

Dentro de la infraestructura existe una arquitectura llamada **MVC** (modelo vista controlador). MVC separa los datos de la aplicación, la interfaz con la que el usuario interactuará y la lógica de la aplicación; ésta se divide en tres componentes:

- **El modelo** es la representación de los datos y la persistencia dentro de la aplicación.
- **La vista** es a donde se enviará la información al cliente para que el usuario interactúe con esos datos.
- **El controlador** es el componente donde se gestionará el uso entre el modelo y la vista, éste hará la adaptación a las necesidades de la aplicación.

Una **API** (Application Programming Interface, por sus siglas en inglés), es un mecanismo de comunicación entre componentes y se define como una interfaz de programación de aplicaciones. Amazon Web Services (2022), menciona que una API puede considerarse como un contrato de servicio entre dos aplicaciones.

En esta imagen se muestra la utilización de una aplicación para hacer llamadas a una API que permite comunicar a dos aplicaciones o servicios entre sí. La aplicación se llama Postman y la puedes encontrar en el siguiente enlace:



The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Scratch Pad' selected, showing options for 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', and 'History'. The main area is titled 'Untitled Request' and shows a 'GET' request to 'https://jsonplaceholder.typicode.com'. The 'Params' tab is active, displaying a single entry: 'Key' under 'KEY' and 'Value' under 'VALUE'. Below this is a 'Response' section containing a cartoon illustration of an astronaut launching a rocket. A message at the bottom says 'Enter the URL and click Send to get a response'.

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

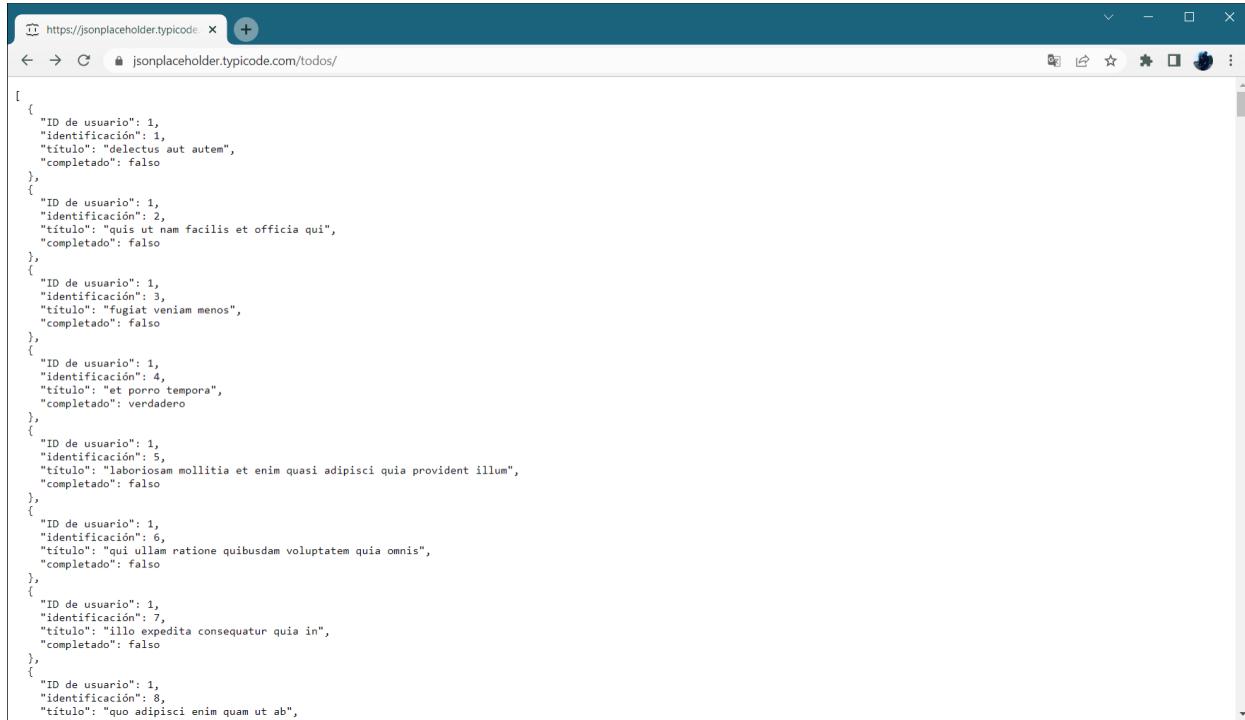
Dentro de las acciones que se pueden realizar con una API se encuentran las principales llamadas GET, POST, PUT y DELETE. Haz clic en el siguiente ejemplo en donde se usa un API:



JSONPlaceholder. (2021). Resources. Recuperado de <https://jsonplaceholder.typicode.com/todos/>

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

Al abrir el enlace en tu navegador podrás visualizarlo de la siguiente manera:



```
[{"ID de usuario": 1, "identificación": 1, "título": "delectus aut autem", "completado": falso}, {"ID de usuario": 1, "identificación": 2, "título": "quis ut nam facilis et officia qui", "completado": falso}, {"ID de usuario": 1, "identificación": 3, "título": "fugiat veniam minus", "completado": falso}, {"ID de usuario": 1, "identificación": 4, "título": "et porro tempora", "completado": verdadero}, {"ID de usuario": 1, "identificación": 5, "título": "laboriosam mollitia et enim quasi adipisci quia provident illum", "completado": falso}, {"ID de usuario": 1, "identificación": 6, "título": "qui ullam ratione quibusdam voluptatem quia omnis", "completado": falso}, {"ID de usuario": 1, "identificación": 7, "título": "illo expedita consequatur quia in", "completado": falso}, {"ID de usuario": 1, "identificación": 8, "título": "quo adipisci enim quam ut ab", "completado": falso}]
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Ahora la información visualizada en la página está en un formato conocido como JSON (JavaScript Object Notation).

JSON (s.f.) menciona que esta anotación es un formato ligero para el intercambio de datos, de fácil lectura y escritura para los usuarios, es fácil de analizar y generar por parte de las máquinas.

El formato es construido por colecciones de pares de nombre/valor conocido como un objeto definido como un conjunto desordenado de pares nombre/valor. Un objeto comienza con **{ (llave de apertura)** y termine con **}** (llave de cierre). Cada nombre es seguido por : (dos puntos) y los pares nombre/valor están separados por , (coma), así como en una lista de valores comúnmente representadas como arreglos, vectores, listas o secuencias.

Para hacer esta llamada dentro del programa los debes colocar de la siguiente manera:

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Scratch Pad' selected, showing options like 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', and 'History'. The main area has a 'Overview' tab with a 'GET https://jsonplaceholder.typicode.com/users' request. Below it, under 'Body', the JSON response is shown:

```
1  [
2   {
3     "id": 1,
4     "name": "Leanne Graham",
5     "username": "Bret",
6     "email": "Sincere@april.biz",
7     "address": {
8       "street": "Kulas Light",
9       "suite": "Apt. 556",
10      "city": "Gwenborough",
11      "zipcode": "92998-3874",
12      "geo": {
13        "lat": "-37.3159",
14        "lng": "81.1496"
15      }
16    },
17    "phone": "1-770-736-8031 x56442",
18  }
19 ]
```

The status bar at the bottom indicates 'Status: 200 OK Time: 4.75 s Size: 6.47 KB Save Response'.

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

La llamada tendrá que estar configurada en GET.

Esto lo verás más a fondo en tu actividad guiada de tema.

Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.

Durante este tema conociste el concepto de infraestructura y todo lo que la compone. También conociste la comunicación entre diferentes capas de la infraestructura por medio de los middlewares. Leíste acerca de los formatos JSON y del funcionamiento de las APIs y cómo se puede interactuar con estas por medio de Postman ejecutando llamadas.

También se mencionaron los microservicios consistentes en dividir las funciones de una aplicación en partes más pequeñas e independientes.

Referencias bibliográficas

- IBM. (s.f.). ¿Qué es *infraestructura de TI*? Recuperado de <https://www.ibm.com/mx-es/topics/infrastructure>
- JSONPlaceholder. (2021). Resources. Recuperado de <https://jsonplaceholder.typicode.com/todos/>
- JSON. (s.f.). *Introducción a JSON*. Recuperado de <https://www.json.org/json-es.html>
- Martínez, E., y Eterovic, J. (2018). *Infraestructura de los Sistemas de Información para la Salud*. Argentina: Kindle.
- Postman. (2022). *Build APIs together*. Recuperado de <https://www.postman.com/>
- RedHat. (2019). ¿Qué es *infraestructura de TI*? Recuperado de <https://www.redhat.com/es/topics/cloud-computing/what-is-it-infrastructure>

Para saber más

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

Lecturas

Para conocer más acerca de **infraestructura back end**, te sugerimos leer lo siguiente:

- Amazon Web Services. (s.f.). *Cómo crear una API pública con AWS*. Recuperado de <https://aws.amazon.com/es/what-is/api/>
- Hernandez, R. (2021). *El patrón modelo-vista-controlador: Arquitectura y frameworks explicados*. Recuperado de <https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern/>

Videos

Para conocer más acerca de las herramientas de **infraestructura back end**, te sugerimos revisar lo siguiente:

- freeCodeCamp.org. (2021, 1 de marzo). *Postman Beginner's Course - API Testing* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=VywxIQ2ZXw4>

- Vida. MRR - Programacion web. (2018, 16 de julio). *Un ejemplo de microservicios #CafeConRivas* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=qAcUGw7HhxM&t=448s>

Checkpoints

Asegúrate de:

- Comprender qué es una API.
- Comprender las llamadas a una API.
- Identificar las partes de una infraestructura.
- Comprender el formato JSON.

Requerimientos Técnicos

- Computadora con acceso a internet.
- Programa Postman instalado.
- Navegador instalado.

Prework

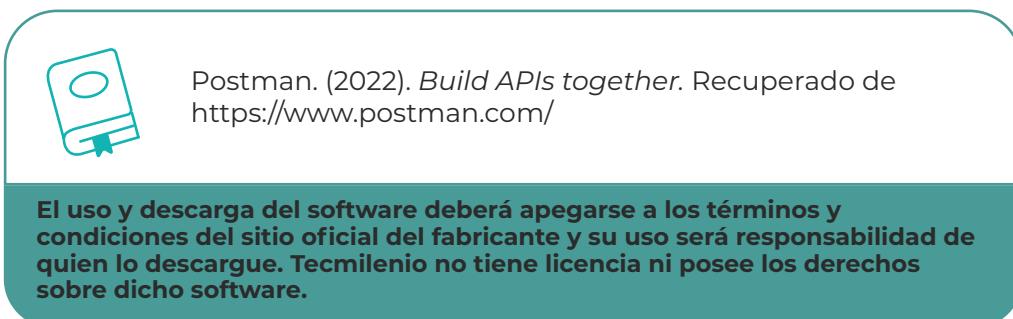
Para tu actividad guiada usarás los conocimientos adquiridos en el tema para construir una infraestructura consumiendo un API pública para mostrar los datos de esta API en tu propia aplicación. Para la realización de esta aplicación será necesario que descargues un editor de código, se te recomienda que uses Visual Studio Code y lo puedes encontrar en este enlace:



Visual Studio Code. (2022). *Download Visual Studio*. Recuperado de <https://code.visualstudio.com/download>

El uso y descarga del software deberá apegarse a los términos y condiciones del sitio oficial del fabricante y su uso será responsabilidad de quien lo descargue. Tecmilenio no tiene licencia ni posee los derechos sobre dicho software.

También será necesario que descargues e instales la aplicación de Postman, puedes descargarla de su página oficial:



Dentro de la página se requiere que selecciones tu sistema operativo y comiences con la descarga:

The screenshot shows the Postman desktop application. On the left, there's a sidebar with sections like 'Build', 'APIs together', and 'Documentation'. The main area has tabs for 'Workspaces', 'New', and 'Import'. A specific request is being edited: 'Twitter API v2 / Tweet / Single Tweet' with the URL 'GET - https://api.twitter.com/2/tweets/:id'. The 'Params' tab is selected, showing fields like 'User ID' and 'Query params'. The 'Body' tab shows a JSON payload. To the right, there's a 'Documentation' panel with sample responses and a 'Request params' section. At the bottom, there's a cartoon illustration of three characters interacting with a computer screen.

What is Postman?
Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.

API Tools
A comprehensive set of tools that help accelerate the API Lifecycle - from design, testing, documentation, and mocking to discovery.

API Repository
Easily store, iterate and collaborate around all your API artifacts on one central platform used across teams.

Workspaces
Organize your API work and collaborate with teammates across your organization or stakeholders across the world.

Intelligence
Improve API operations by leveraging advanced features such as search, notifications, alerts and security warnings, reporting, and much more.

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Ahora es momento de que pruebes las llamadas a la API con el programa Postman de la siguiente manera:

The screenshot shows the Postman interface with a GET request to `https://jsonplaceholder.typicode.com/users`. The response body is a JSON object representing a user:

```

1  [
2   {
3     "id": 1,
4     "name": "Leanne Graham",
5     "username": "Bret",
6     "email": "Sincere@april.biz",
7     "address": [
8       {
9         "street": "Kulas Light",
10        "suite": "Apt. 556",
11        "city": "Gwenborough",
12        "zipcode": "92998-3874",
13        "geo": [
14          {
15            "lat": "-37.3159",
16            "lng": "81.1496"
17          }
18        ]
19      }
20    ]
21  ]

```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Prueba con las siguientes urls en Postman y en un documento describe los datos que arrojan:



JSONPlaceholder. (2021). Resources.
Recuperado de
<https://jsonplaceholder.typicode.com/posts>
<https://jsonplaceholder.typicode.com/comments>
<https://jsonplaceholder.typicode.com/albums>
<https://jsonplaceholder.typicode.com/photos>
<https://jsonplaceholder.typicode.com/todos>
<https://jsonplaceholder.typicode.com/users>

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

Una vez que tengas la descripción de estos datos estarás listo para desarrollar una página web en la que los mostrarás en una tabla para que se puedan leer de manera más sencilla. Se iniciará utilizando HTML, JS Y CSS.

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educacional y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.