**NodeJS**

* **Concepto API**

Intermediario entre el usuario y la base de datos. Sirve para tratar datos. GET, POST, PUT, DELETE. Los datos que van y vienen entre el usuario y la base de datos puede ser en formato JSON o XML.

Los navegadores solo usan GET, en cambio las aplicaciones usan los todos comandos.

* **Request, Response**

Request: Objeto que contiene la información que se ha enviado de la aplicación a la API.

Response: Respuesta que te da la API según la solicitud que haya recibido y luego de haber procesado los datos recibidos.

* **Verbose HttP**
* **Header**
* **Content-type**
* **Express**
* Tratamiento de datos simples
* Cantidad pequeña de datos
* Orientado para SPA o aplicaciones simples
* **NodeJS**
* **Hilos**
* **Async/Await**
* **Postman (Restful client)**
* **DNS**

**Montar un servidor**

NPM init 🡪 License: MIT

Crear index.js

NPM i express

NPM i nodemon

Package.json 🡪 En scripts, añadir

Montar server 🡪 npm run dev

**Funcionamiento API**

El cliente envía información: GET, POST, PUT, DELETE (Cliente):

* XML
* JSON
* Multipart

La API procesa información (API):

* Mapping de objetos
* Acceso a BBDD (Crear, eliminar…)
* Tratamiento información

Consistencia de datos (BBDD):

* Relacional (MySQL)
* No relacional (Mongo, Azure, Cosmos…)

**Enviar datos:**

* **QueryString** 🡪 API/Usuario?Nombre=Marta&Apellido=deHaro

Los parámetros empiezan después de la “?”

Se anidan parámetros usando “&”

* **Params** 🡪 API/Usuario/Marta/deHaro -> Barra navegación/Postman
  + API/Usuario/:nombre/:apellido -> URLservidor/
* **Body** 🡪 Postman/Aplicación Cliente
  + Se envía la información, generalmente, como XML o JSON
  + Se usa sobre todo para mandar POST y PUT y DELETE
* **Postman** 🡪
  + Software que simula la respuesta de un cliente (lo que manda la app al servidor) y obtiene la respuesta del servidor.
  + Únicamente envía información al servidor