



El futuro digital
es de todos

MinTIC



CICLO IV:

Desarrollo de Aplicaciones Web

Mision
TIC2022



El futuro digital
es de todos

MinTIC

UN UNIVERSIDAD
DEL NORTE

Vigilada Mineducación

Sesión 01: Desarrollo de Aplicaciones Web

Introducción a la programación de aplicaciones web

Mision
TIC2022



Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

1. Conocer las generalidades del desarrollo Web.
2. Identificar las diferencias entre Back-End, Front-End y Full-Stack.
3. Desarrollar peticiones en HTML a servicios RESTful.



El futuro digital
es de todos

MinTIC

UN UNIVERSIDAD
DEL NORTE

Vigilada Mineducación

INTRODUCCIÓN A LA PROGRAMACIÓN DE APLICACIONES WEB

Misión
TIC2022



Introducción

- Entender conceptos generales como que es un desarrollador **Front-End**, **Back-End** o **Full-Stack** (desarrolladores web) y el funcionamiento de las aplicaciones web.
- Una aplicación web un conjunto de archivos que se encuentran almacenados en un computador que cuenta con una conexión a internet (**servidor**).
- A través de un navegador web nosotros como **clientes** podemos cargar la aplicación web.
- Siempre que navegamos por internet estamos descargando datos del servidor y enviando datos de vuelta.

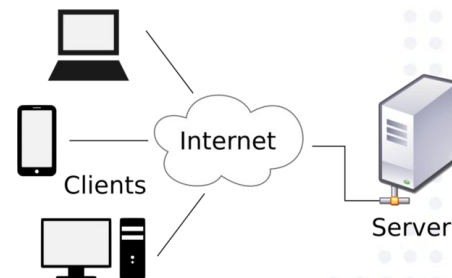


Imagen tomada de [Medium/@subhanqdx](https://medium.com/@subhanqdx)



Generalidades

- Un cliente carga la información disponible de un servidor mediante lo que se conoce como una petición **HTTP**.
- **HTTP** es el **protocolo** mediante el cual el navegador web se comunica con el servidor a través de un **dominio web**, un **puerto**, una **ruta de acceso**, etc.
- El servidor se encarga de analizar la solicitud y darle una respuesta.
- Estas respuestas suelen ser contenido web o información en formato JSON, XML o con información en formato blob.

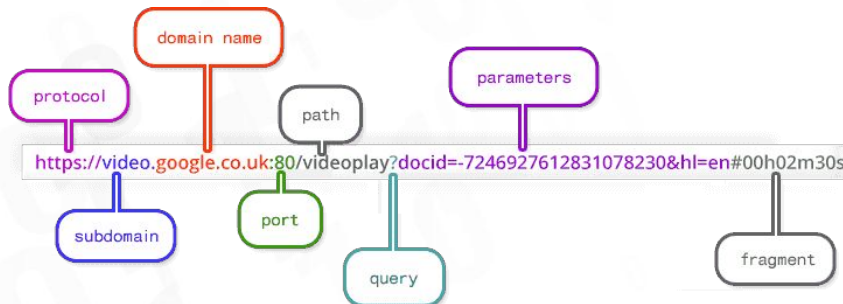


Imagen tomada de [Medium/@joseph.pyram](#)



Front-End

- Consta de todos los aspectos visuales que pueda generar nuestra aplicación web. Estas interfaces de usuario por lo general están definidas bajo tres tipos de archivos:
 - HTML, (Hyper Text Markup Language).
 - CSS (Cascading Style Sheets).
 - Javascript (JS).
- Un desarrollador Front-End define la estructura de una página web a través de los archivos HTML.
- Un desarrollador Front-End define los estilos, ya sea diseños o animaciones, de una página web a través de los archivos CSS.
- Un desarrollador Front-End agrega interactividad a su sitio web a través de javascript, tal como hacer peticiones HTTP, escuchar eventos, etc.



Imagen tomada
de [CursosGIS](#)



Front-End

- Un desarrollador Front-End tiene la responsabilidad de estar al día con conocimiento que haga parte de lo que se considere estado del arte.
- Se puede consultar [stateofjs](#) o [roadmap](#).
- Un desarrollador Front-End puede hacer su trabajo perfectamente con HTML, CSS y JS.
- Seguir esta forma de desarrollo nos puede dejar con mucho código que se repite constantemente, dejado de seguir así [la buena práctica de desarrollo DRY](#).
- Se suelen utilizar librerías o frameworks para facilitar ciertos aspectos del desarrollo, entre las más conocidas están:
 - React.
 - Angular.
 - Vue.



Imagen tomada de
[Medium/somoswigou](#)



Back-End

- El desarrollo Back-End consiste en configurar un servidor web, el cual estará escuchando las diferentes peticiones del cliente.
- El desarrollador Back-End se encarga de todo lo que no hace parte de la interfaz gráfica de una aplicación web.
 - Se encarga de definir las funcionalidades del modelo de negocios.
 - Procesos en base de datos.
 - Consumir streaming o archivos.
 - Microservicios en segundo plano como envío de correo electrónico, notificaciones de tipo push, tareas recurrentes, etc.



Back-End

- Cuando se trata de desarrollo de Back-End, hay un múltiples lenguajes de codificación que suelen estar asociados a él. Entre ellos están:
 - Python
 - Java
 - JS
 - PHP
 - Ruby
 - ASP.NET
 - Entre otros

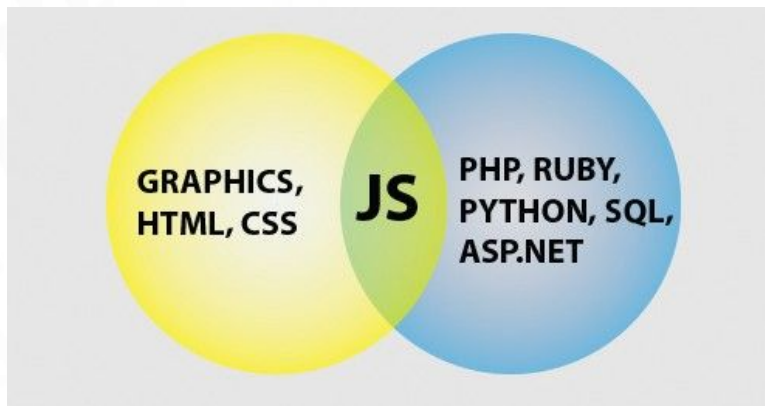


Imagen tomada de [Medium/@larong2122](https://medium.com/@larong2122)



Full-Stack

- Los desarrolladores Full-Stack, es un rol en el cual se trabajan los requerimientos del lado de los desarrolladores Front-End y Back-End.
- Por lo general el desarrollo se divide en tres capas:
 - Presentación.
 - Modelo de negocio.
 - Base de datos.

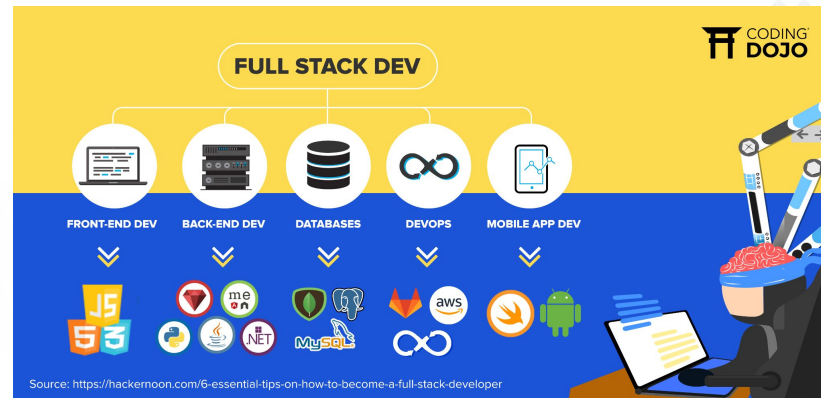


Imagen tomada de [codingdojo](https://codingdojo.com)



Full-Stack

- Las tecnologías usadas para desarrollo Full-Stack son un grupo de otras librerías o frameworks, o frameworks específicos para su desarrollo, como lo son:
 - MERN (Mongo, Express, React, NodeJs).
 - MEAN (Mongo, Express, Angular, NodeJs).
 - MEVN (Mongo, Express, Vue, NodeJs).
 - Django.
 - Ruby on Rails (RoR).
 - Entre otras.

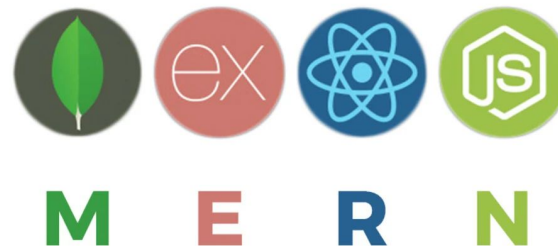


Imagen tomada de [Dev/globalgraphic](https://dev.to/globalgraphic)



Full-Stack

- También han de tener conocimientos en el despliegue de aplicaciones web, su arquitectura y recursos de consumo, etc.
- En la última década esto se ha venido considerando como una disciplina aparte, incluyendo todo lo que serían proveedores de computación en la nube como:
 - Amazon Web Services (AWS).
 - Google Cloud Platform (GCP).
 - Azure.
 - Entre otros.



Google Cloud

Imagen tomada de [Medium](#)



Interfaz de programación de aplicaciones (API)

- Application Programming Interface (API).
- Cuando hablamos de una API normalmente nos referimos a un intermediario o un tercero al cual nuestra aplicación web tiene acceso.
- Usamos una API para obtener o enviar algún tipo de información a un servicio de terceros, por ejemplo:
 - El botón de “Compartir en...” que vemos en diferentes aplicaciones para compartir contenido directamente en redes sociales.
 - Un servicio que analice la información enviada.

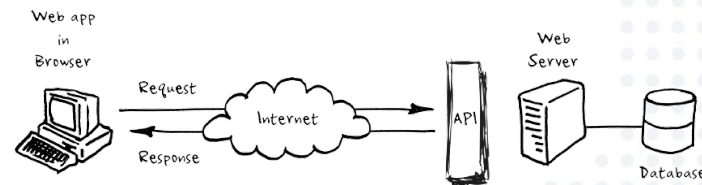


Imagen tomada de [Medium/@perrysetgo](https://medium.com/@perrysetgo)



Interfaz de programación de aplicaciones (API)

- Una API puede ser utilizada para obtener información de terceros como lo sería el caso de SWAPI, una API que aloja información relacionada a Star Wars.
- Para entender más de cómo se definen las API hay que entender más de los principios de desarrollo REST. Donde se define:
 - Todo es un recurso.
 - Cada recurso ha de tener una ruta de acceso.
 - Se utilizan métodos HTTP estándares (GET, PUT, POST, DELETE).
 - Permitir múltiples representaciones para el mismo recurso.
 - La comunicación con el cliente ha de carecer de estado.

REST Design Principles

Everything is a Resource

Each Resource is identifiable by Unique URI

Use the standard HTTP methods

Allow multiple representations for same
Resource

Communication should be always stateless

Imagen tomada de [Medium/@subhangdxt](https://medium.com/@subhangdxt)



Interfaz de programación de aplicaciones (API)

- La mayoría de las implementaciones de la API REST se basan en negociaciones de contenido impulsadas por agentes.
- Esta se basa en el uso de cabeceras de solicitud HTTP o patrones URI de recursos.
- Para determinar el tipo del contenido, el servidor utiliza la cabecera de solicitud HTTP Content-Type. Algunos ejemplos comunes de tipos de contenido son:
 - “text/plain”.
 - “application/xml”.
 - “text/html”.
 - “application/json”.
 - “image/gif”.
 - “image/jpeg”.



Interfaz de programación de aplicaciones (API)

- Es una arquitectura cliente-servidor.
- El **servidor** almacena y/o manipula la información y la pone a disposición del usuario de manera eficiente.
- El **cliente** toma esa información y la muestra al usuario y/o la utiliza para realizar posteriores peticiones.
- Esta separación de intereses permite que tanto el cliente como el servidor evolucionen de forma independiente.
- Sólo requiere que la interfaz siga siendo la misma.

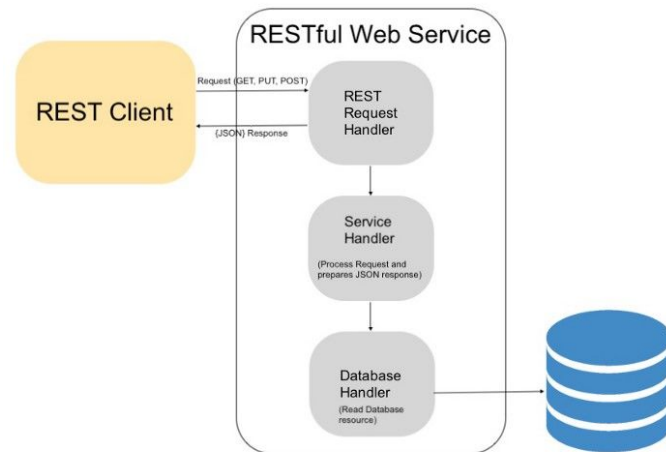


Imagen tomada de [Medium/@subhangdxt](https://medium.com/@subhangdxt)



Interfaz de programación de aplicaciones (API)

- Por último tenemos la estructura de una URL, como mencionó previamente:

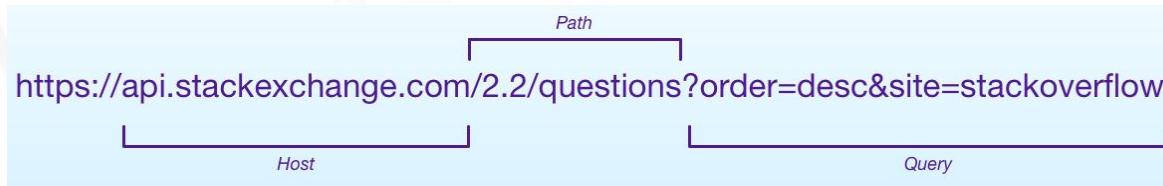


Imagen tomada de [Medium/@subhangdxt](#)

- **Host:** Representa la máquina a la que queremos acceder.
- **Path:** Representa el recurso que queremos consumir de la API.
- **Query:** Representa parámetros adicionales a nuestra consulta, permitidos por la API.



El futuro digital
es de todos

MinTIC

UN UNIVERSIDAD
DEL NORTE

Vigilada Mineducación

Ejercicios de práctica

Mision
TIC2022



Referencias

- <https://betterprogramming.pub/beginners-guide-to-web-development-d46dadfdd049>
- <https://medium.com/@subhangdxt/beginners-guide-to-client-server-communication-8099cf0ac3af>
- <https://medium.com/@larong2122/what-is-back-end-development-d93732c129e9>
- <https://roadmap.sh/frontend>
- <https://medium.com/@kamranahmedse/modern-frontend-developer-in-2018-4c2072fa2b9c>
- <https://infinijith.medium.com/everything-you-want-to-know-about-full-stack-development-all-about-full-stack-development-9f79ffdf8d09>
- <https://medium.com/@perrysetgo/what-exactly-is-an-api-69f36968a41f>
- <https://medium.com/extend/what-is-rest-a-simple-explanation-for-beginners-part-1-introduction-b4a072f8740f>
- <https://medium.com/extend/what-is-rest-a-simple-explanation-for-beginners-part-2-rest-constraints-129a4b69a582>



El futuro digital
es de todos

MinTIC

UN UNIVERSIDAD
DEL NORTE

Vigilada Mineducación

¡GRACIAS
POR SER PARTE DE
ESTA EXPERIENCIA
DE APRENDIZAJE!



Misión
TIC 2022