



# Arquitectura Front

## End



# Contenido

- Arquitectura
- Librerías
- Frameworks
- Ejemplos



# Arquitectura Front-end

## ¿Qué Sabemos ?

- HTML construye la estructura y contenido de un sitio web (títulos, párrafos, enlaces, imágenes, videos).
- CSS: define los estilos, colores, posición y animaciones para cada elemento en HTML.
- JavaScript: nos permite crear interacciones personalizadas con los usuarios, recibir sus solicitudes y entregarles inmediatamente el nuevo contenido que están buscando.

# Arquitectura Front-end

## ¿Qué más existe?

Estas 3 tecnologías son los fundamentos del desarrollo web. Pero existen herramientas especializadas para construir aplicaciones mucho más complejas de forma mucho más fácil:

- **Librerías:** son bloques de código que cumplen una función en concreto, con ellas puedes solucionar problemas muy específicos.
- **Frameworks:** son herramientas todo en uno, suelen imponer una forma de programar muy estricta, pero te permiten solucionar casi cualquier problema de tu aplicación web muy fácilmente.

# Arquitectura Front-end

## ¿Qué necesito?

Todas estas tecnologías hacen parte del desarrollo frontend. Pero una arquitecto(a) frontend profesional, también debe entender el desarrollo backend. Aprender backend con Node.js te convertirá en un mejor frontend developer.

- Backend: es la parte de un sitio web que no pueden ver los usuarios, donde interactuamos con las bases de datos para procesar, guardar y encontrar la información del negocio.
- Node.js: es JavaScript para desarrollar en el lado del backend, tiene muchas ventajas para interactuar con el frontend desde el servidor (porque comparte JavaScript tanto en el frontend como en el backend).

# Librerías

## ¿Qué son?

Para explicarlas mejor, comparémosla con la librería de la universidad, si necesito saber algo sobre un tema o solucionar un problema, recurro a ella.

Es una herramienta para hacer que la programación **lleve menos tiempo** porque puedes obtener piezas de código JavaScript y ponerlo en tu aplicación web. **Generalmente, los conceptos de frameworks y librerías son sinónimos, pero las librerías son normalmente más pequeñas.**



# Librerías

## ¿Qué son?

Para explicarlas mejor, comparémosla con la librería de la universidad, si necesito saber algo sobre un tema o solucionar un problema, recurro a ella.

Es una herramienta para hacer que la programación **lleve menos tiempo** porque puedes obtener piezas de código JavaScript y ponerlo en tu aplicación web. **Generalmente, los conceptos de frameworks y librerías son sinónimos, pero las librerías son normalmente más pequeñas.**

# Librerías - Donde se encuentran

1.- Incluir por medio de un CDN (Content Delivery Network o Red de Distribución de Contenidos), cuando solo necesita incluir el CSS o JS compilado.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dis" data-bbox="344 429 643 450"/>
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/di" data-bbox="344 519 643 539"/>
```

2.- Por un gestor de paquetes como de paquetes como **npm**

```
$ npm install bootstrap@5.3.0-alpha3
```

**npm** es parte esencial de Node.js, el entorno de ejecución de JavaScript en el lado del servidor basado en el motor V8 de Google.

3.- Descargando directamente el compilado de la librería.





# Algunas Librerías css

**hover.css** Permite utilizar distintos tipos de estilos al posicionarnos sobre un elemento.  
<https://www.chartjs.org/docs/latest/getting-started/>

**animate.css** Permite animar elementos del sitio con distintos efectos preestablecidos  
<https://ianlunn.co.uk/articles/hover-css-tutorial-introduction>

**threedots.css** Permite insertar de manera sencilla tres puntos para simular una carga de la pagina (estado loader) <https://nzbin.github.io/three-dots/>

**lightgalleryjs** Permite crear galerías de videos e imágenes elegantes y modernas, funciona con JQuery <https://www.lightgalleryjs.com/demos/thumbnails/>

# Algunas Librerías css / js

**chart.js** Librería dedicada a la creación de gráficos y/o uso de datos estadísticos.

<https://www.chartjs.org/docs/latest/getting-started/>

<https://ordinarycoders.com/blog/article/11-chart-js-examples>

**moment.js** Ayuda al trabajo con fechas (operaciones y/o formatos) <https://momentjs.com/>

**axios.js** peticiones HTTP a un endpoint, para consumir una API por ejemplo <https://axios-http.com/es/docs/example>

**video.js** permita crear un reproductor de video propio <https://videojs.com/getting-started>

**bootstrap** Es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. <https://getbootstrap.com/docs/5.3/examples/>

# Actividad Uso de librerías

- Consumir una API de indicadores económicos. (<https://mindicador.cl/>) con **axios.js**.
- Crear un sitio con la ayuda de la librería de **bootstrap**.
- Listar en un selector los tipos de indicadores que tiene la API (ver documentación api).
- Cuando seleccione un tipo de indicador, la pagina debe mostrar el valor de esa moneda con toda su información, de manera ordenada y bien visible para el usuario.
- Graficar el movimiento del indicador en el último año (ver documentación api) usar libreria **chart.js**.
- Debe mantener buenas practicas de desarrollo, que se vea bien y funcional en cualquier tipo de vistas o tamaños de página (responsivo). si utiliza cualquier otra libreria revisada podra tener puntaje extra

