



# HTML5 semántico y estructura base

Ing. Mauricio Quezada

# Introducción al Desarrollo Frontend

- ¿Qué es el desarrollo Frontend?
- Rol del Frontend en aplicaciones web modernas
- Tecnologías base del Frontend
- Buenas prácticas iniciales



# ¿Qué es el Desarrollo Frontend?

---

- **Definición básica**
- Área del desarrollo web encargada de:
  - La interfaz visual
  - La interacción con el usuario
  - La experiencia de uso
- Todo lo que el usuario **ve, toca y usa** en una aplicación web





# ¿Por qué es tan importante?

Es el **primer contacto** del usuario con el sistema

Influye directamente en:

Un sistema puede funcionar bien internamente, pero fracasar si el Frontend es deficiente

Usabilidad

Accesibilidad

Experiencia de usuario (UX)



# Frontend en el mundo real



**Ejemplos de  
aplicaciones  
modernas**



Sistemas  
académicos



Tiendas en línea



Plataformas  
bancarias



Redes sociales



Aplicaciones  
empresariales

An abstract graphic on the left side of the slide. It features a light gray background with a network of thin gray lines and dots. Overlaid on this are various colored squares: black, blue, purple, orange, green, and pink. Some squares are solid, while others are outlined. They are arranged in a scattered, interconnected manner, suggesting a digital or network theme.

# Tecnologías base del Frontend

## El trío fundamental

- **HTML** → Estructura del contenido
- **CSS** → Apariencia y diseño visual
- **JavaScript** → Interactividad y comportamiento



# Analogía sencilla

## Para entender mejor...

- HTML → Esqueleto de una casa
- CSS → Pintura, decoración y estilo
- JavaScript → Puertas, luces y comportamiento



# Estructura general de una aplicación web

## Componentes principales

- Estructura del documento (HTML)
- Estilos y diseño (CSS)
- Lógica e interacción (JavaScript)
- Navegador como entorno de ejecución





# Flujo básico de funcionamiento

## ¿Qué pasa cuando abres una página web?

- El navegador carga el HTML
- Aplica los estilos CSS
- Ejecuta el JavaScript
- Se muestra la interfaz al usuario
- El usuario interactúa con la aplicación



# HTML (HyperText Markup Language)

Define la **estructura** del contenido

Organiza la información

Permite que el navegador entienda qué es cada elemento

Base de cualquier aplicación web



# ¿Qué es HTML5?

**HTML5** es el lenguaje que define la **estructura** de una página web

Le dice al navegador:

- Qué contenido hay
- Qué es un título, un texto, un menú, una imagen, etc.

No da diseño

No da comportamiento

**Da significado y estructura**



# ¿Para qué sirve HTML5?

HTML5 sirve para:

Organizar la información de una página web

Dar sentido al contenido

Facilitar:

- La lectura del navegador
- La accesibilidad
- El mantenimiento del código

👉 Es la **base de todo Frontend**.

# ¿Qué es una etiqueta HTML?

- HTML funciona mediante **etiquetas**
- Una etiqueta:
  - Marca el inicio y fin de un elemento
  - Define qué tipo de contenido es
- Ejemplo conceptual:
  - “Esto es un título”
  - “Esto es un párrafo”
  - “Esto es un menú”





# Tipos de etiquetas en HTML5

Etiquetas de estructura

Etiquetas de contenido

Etiquetas semánticas

Etiquetas genéricas



# Estructura mínima de una página

Un documento HTML tiene:

- Un inicio
- Un contenido
- Un final

Todo lo que se ve va dentro del **body**

👉 El body es la “hoja” donde se escribe.

# Tipos básicos de elementos HTML



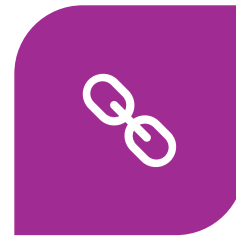
TÍTULOS



TEXTOS



IMÁGENES



ENLACES



CONTENEDORES

# Estructura base de un documento HTML

## Etiquetas obligatorias

`<!DOCTYPE html>` → indica HTML5

`<html>` → contiene toda la página

`<head>` → información para el navegador

`<body>` → contenido visible



Sin estas etiquetas, no hay página web.



<head>

¿Para qué sirve?

No se muestra al usuario

Contiene:

- Título de la pestaña
- Metadatos
- Enlaces a estilos

Es información **sobre** la página

`<body>`

---

**El cuerpo de la página**

---

Todo lo que el usuario ve está aquí

---

Textos, imágenes, botones, enlaces

---

Es la parte visible del sitio

Títulos:  
<h1> a  
<h6>

---

## Jerarquía del contenido

---

<h1> → título principal

---

<h2> → subtítulos

---

<h3> → subniveles

---

Hasta <h6>

---



No se usan por tamaño, se usan por **orden**.



Texto: <p>

---

# Párrafos

---

Contiene texto normal

---

Organiza la lectura

---

No es solo texto suelto

## Salto y separadores

---

`<br>` → salto de línea

---

`<hr>` → separación  
temática



Se usan con  
moderación.

# Enlaces:

`<a>`

## Navegación

Permite ir a:

- Otras páginas
- Otros sitios

Esencial para la web



Sin enlaces, no hay web.



# Imágenes:

## <img>

---

### Mostrar imágenes

---

Inserta imágenes en la página

---

No tiene cierre

---

Atributo importante: texto alternativo

---



Fundamental para accesibilidad.

# Listas

## Organización de información

`<ul>` → lista sin orden

`<ol>` → lista ordenada

`<li>` → elemento de la lista

👉 Muy usadas en menús.

# Contenedores

---

## Agrupar contenido

---

`<div>` → contenedor genérico

---

`<span>` → contenedor en línea

---

 No agregan significado, solo agrupan.



# ¿Qué es un contenedor?

Un contenedor es un elemento que:

- **Agrupar** contenido
- Ordena elementos
- Ayuda al diseño y organización



No muestra nada especial por sí solo.

El `<div>`  
explicado  
fácil

---

¿Qué es un `<div>`?

---

`<div>` significa **division**

---

Es un **contenedor genérico**

---

No tiene significado semántico

---

Solo sirve para **agrupar cosas**

---



Es como una **caja invisible.**

# ¿Para qué se usa un `<div>`?

## Usos comunes

Agrupar textos

Agrupar imágenes

Separar secciones visuales

Aplicar estilos con CSS

Manipular grupos con JavaScript

👉 El `<div>` no dice qué es el contenido, solo lo agrupa.



# Ejemplo mental de <div>

Una caja de cartón:

- Puede contener libros
- Puede contener ropa
- Puede contener cables

La caja no define el contenido

Solo lo **contiene**



Eso es un <div>.

¿Qué pasa  
si uso solo  
<div>?

El navegador **no sabe** qué es cada cosa

Los lectores de pantalla se confunden

El código se vuelve difícil de leer

Todo se vuelve “una caja dentro de otra caja”

# <div> vs etiquetas semánticas

<div> → caja genérica

<header> → encabezado

<nav> → navegación

<main> → contenido principal

👉 El <div> **no explica**, las semánticas **sí**.

# ¿Cuándo Sí usar <div>?

---

Cuando no existe una  
etiqueta semántica adecuada

---

Para agrupar elementos solo  
por diseño

---

Para aplicar estilos o lógica  
específica



`<span>`

---

**Texto en línea**

---

Agrupar partes pequeñas de texto

---

No rompe líneas

---

Muy usado para estilos puntuales

Formularios:  
<form>

Interacción

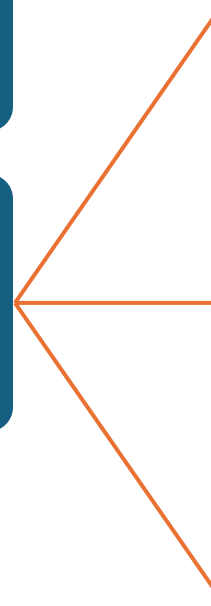
Permite  
enviar datos

Base de:

Login

Registro

Contacto



# Campos de formulario

## Elementos más usados

`<input>` → campo de entrada

`<label>` → etiqueta descriptiva

`<textarea>` → texto largo

`<button>` → botón

# Orden básico de una página

---

Contenedor principal

---

Secciones internas

---

Contenido dentro de secciones

---

👉 Todo se organiza por  
**bloques.**