

# INTRODUCCIÓN AL DESARROLLO FRONTEND CON REACT 2023



SECRETARÍA DE  
EXTENSIÓN  
UNIVERSITARIA  
UTN - FRC



\*UTN  
Facultad Regional Córdoba

Agencia  
CÓRDOBA  
JOVEN



CÓRDOBA  
entre todos

# Profesor

---



## RUBÉN ANÍBAL ROMERO

Ingeniero en Sistemas (UTN-FRC)  
Especialista en Ingeniería de Negocios (UTN-FRC)  
13 años de experiencia en la industria del software.  
Arquitecto de Software en NaranjaX  
Docente Universitario (UTN-FRC)  
Docente Investigador

# Objetivo del Curso

---

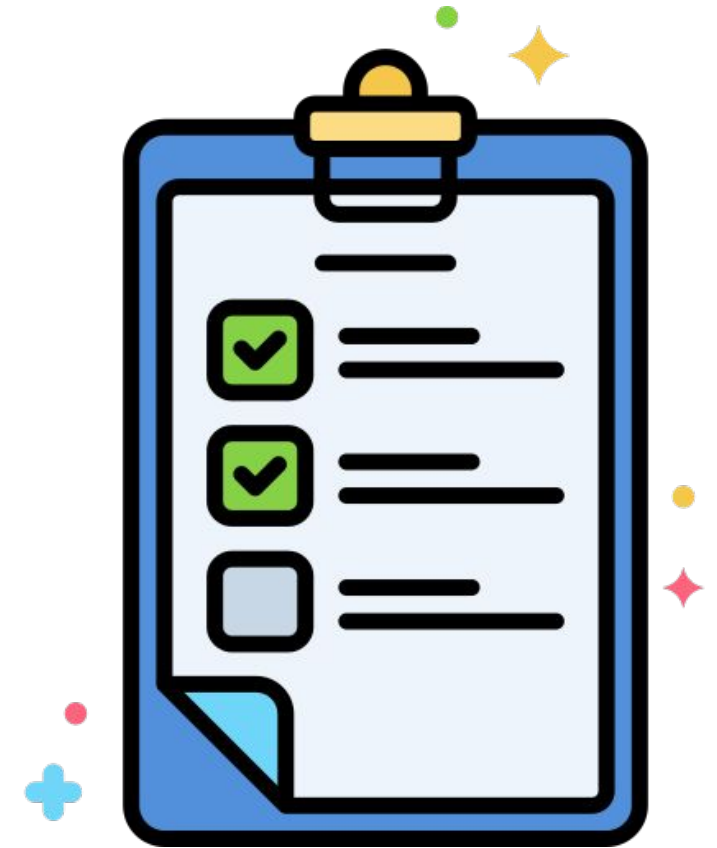
El objetivo principal de este curso es introducir a los estudiantes en el emocionante mundo del desarrollo de aplicaciones web con un enfoque en el FRONTEND, es decir, en la parte visible y la experiencia del usuario.



# Cursado

---

- 10 clases grabadas - 2 hs cada una.
- Cada clase tendrá contenidos con un ejemplo aplicado.
- Al finalizar la clase el alumno tendrá disponible actividad para realizar.
- Clase consulta una vez por semana. Horario y link de zoom en el aula virtual.



# Temas del curso

---



Clase 1. Introducción a la programación y al desarrollo web

Clase 2. Fundamentos de HTML y CSS

Clase 3. Introducción a JavaScript

Clase 4. Fundamentos de React

Clase 5. Desarrollo de aplicaciones web con React

Clase 6. Generación de código fuente asistida por IA con ChatGPT

Clase 7. Despliegue de una aplicación React

Clase 8. Desarrollo seguro en aplicaciones web

Clase 9. Integración con servicios de terceros

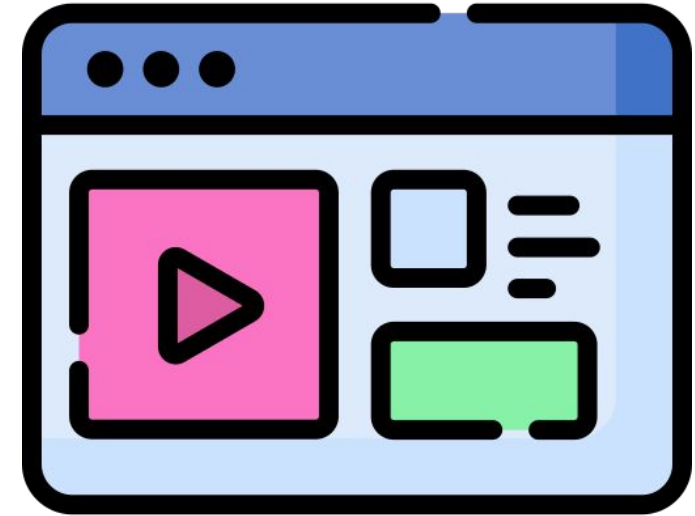
Clase 10. Desarrollo avanzado en React

# Clase 01 - Contenido

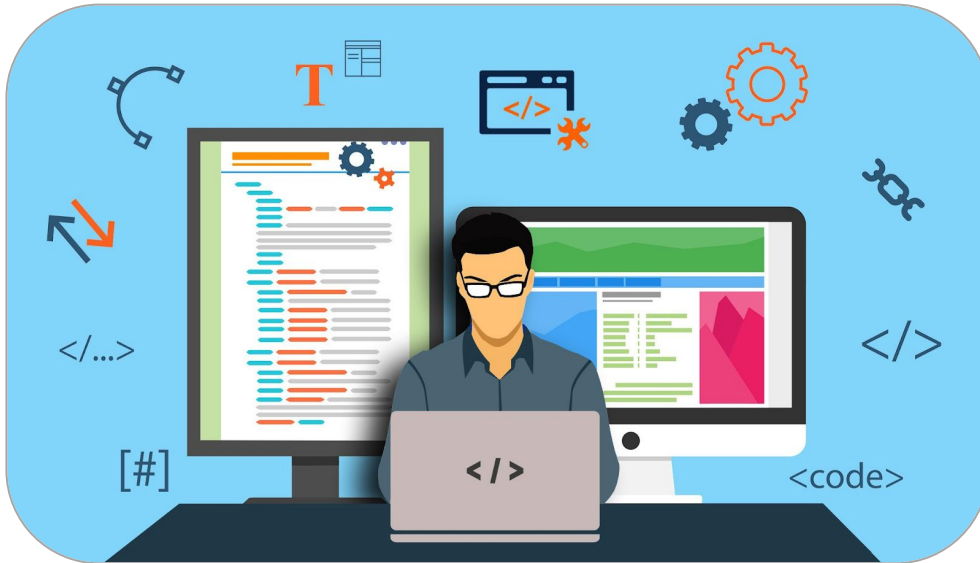
---

## Introducción a la programación y al desarrollo web

- Conceptos básicos de programación y desarrollo web
- Herramientas necesarias para el desarrollo web
- Entornos de desarrollo integrado (IDE)



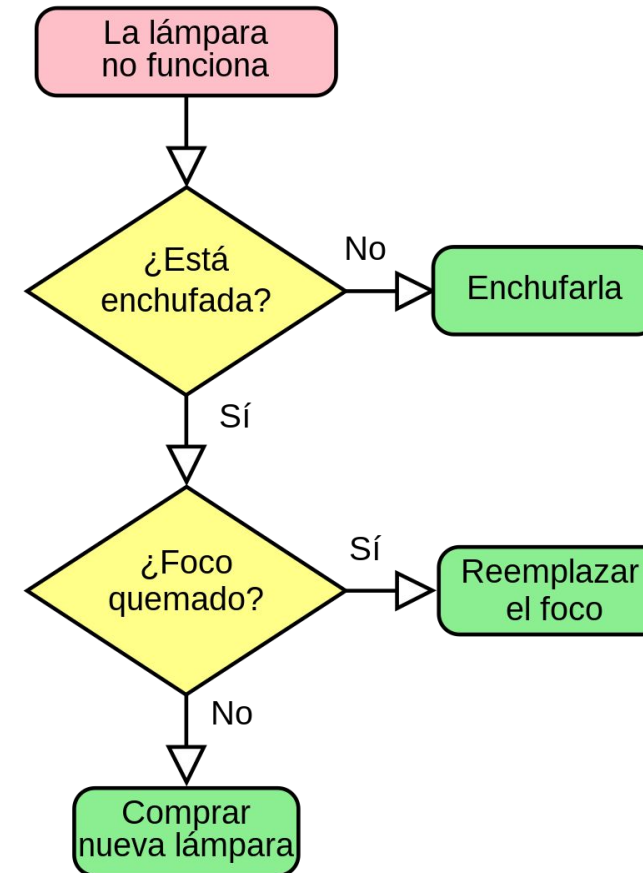
# Introducción a la programación



- En la programación se utilizan **algoritmos**, que son **secuencias de pasos lógicos** que permiten resolver un problema.
- Los **lenguajes de programación** son herramientas que permiten escribir **algoritmos** que podemos escribir en un **código fuente** y que serán ejecutadas por una computadora.
- La programación es el proceso de **diseñar, escribir, probar y mantener el código fuente** de una aplicación o software.
- La programación es una **habilidad muy demandada en la industria tecnológica** y puede ser utilizada para una gran variedad de aplicaciones.

# Un ejemplo de algoritmo

Un algoritmo es un conjunto de operaciones que busca resolver un problema determinado a través de secuencias lógicas.

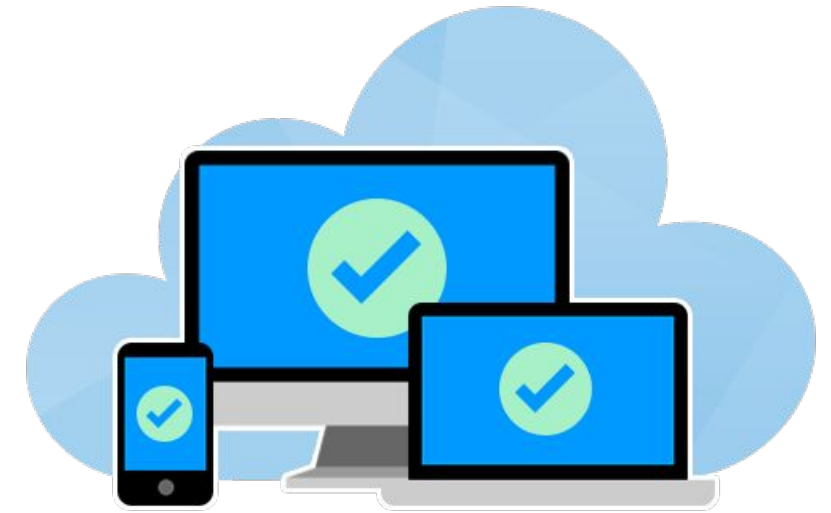




# ¿Qué es una aplicación?

---

- Una aplicación es un programa diseñado para ejecutar una tarea o un conjunto de tareas específicas.
- Las aplicaciones pueden ser de escritorio, móviles o web.
- En general, se espera que una aplicación sea fácil de usar, eficiente y resuelva problemas cotidianos.
- Las aplicaciones pueden ser desarrolladas por empresas, equipos de programadores o individuos.



**Ejemplo:**

Un juego de carreras, una aplicación de mensajería, un editor de texto, una aplicación para pedir comida a domicilio, entre otros.

# ¿Qué es una aplicación web?

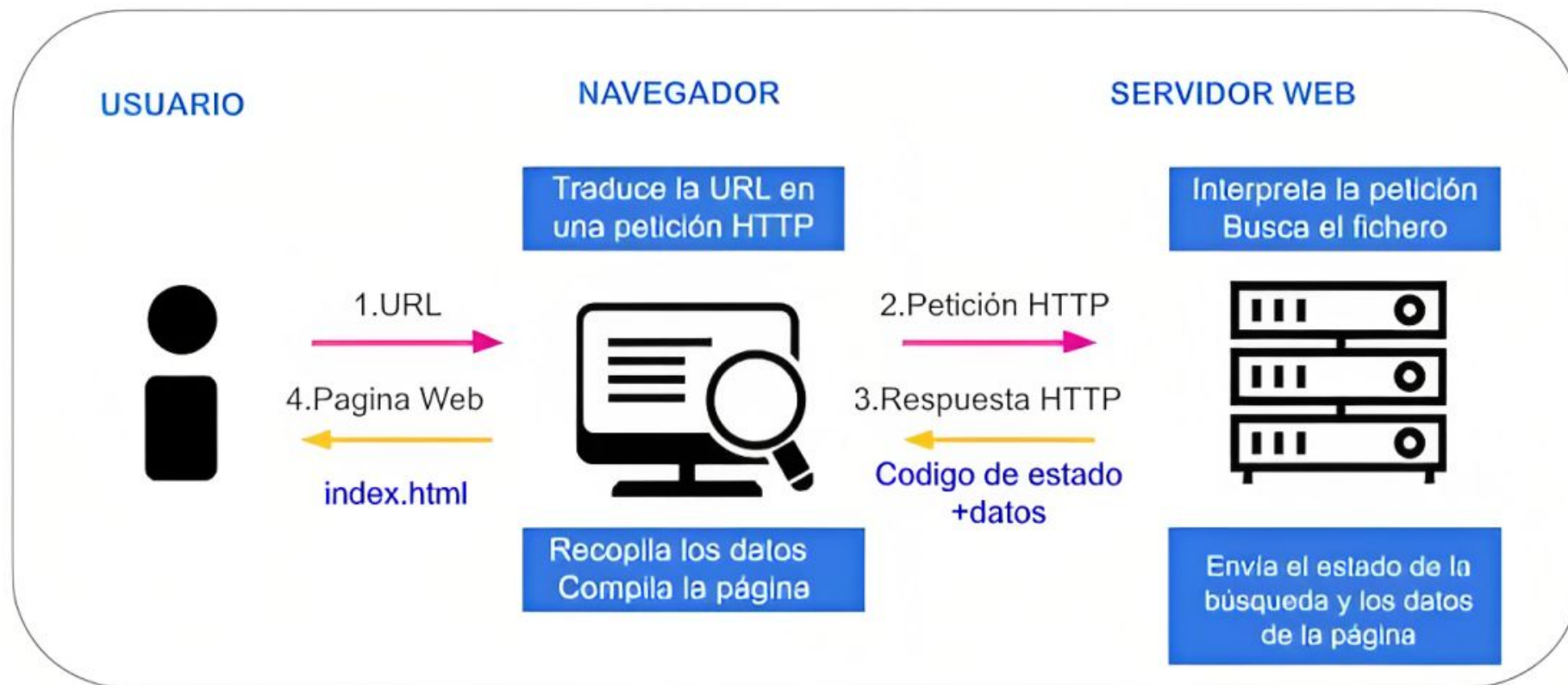


- Una aplicación web es una aplicación que se ejecuta en un navegador web y que permite realizar diversas tareas y actividades a través de internet.
- Las aplicaciones web son accesibles desde cualquier dispositivo con conexión a internet, lo que las hace muy convenientes para el usuario.

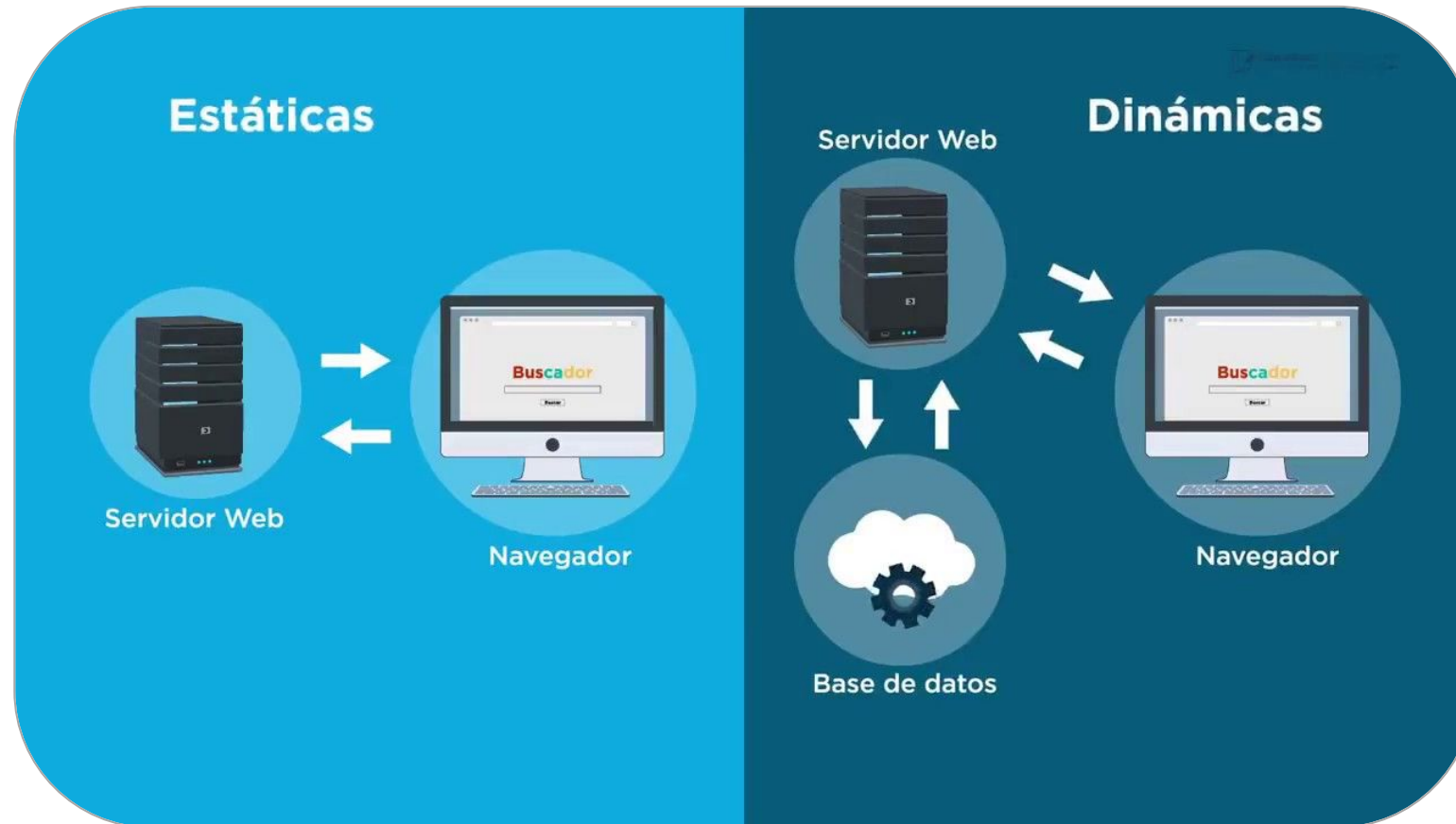
## Ejemplo:

Desde aplicaciones simples como una calculadora o una lista de tareas hasta aplicaciones más complejas como un sistema de comercio electrónico (MercadoLibre).

# ¿Cómo funciona una aplicación web?



# Aplicación Web Estático vs Dinámico

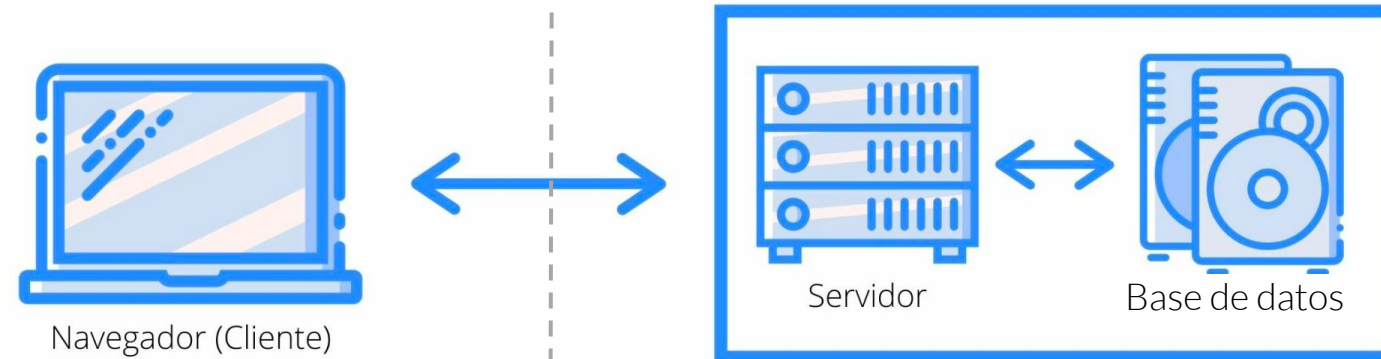


# Frontend vs Backend

- El **frontend** se refiere a la parte de una aplicación web que los usuarios ven y con la que interactúan directamente, como la **interfaz gráfica, diseño y experiencia del usuario**.
- El **backend** es la parte invisible de una aplicación web que maneja el **procesamiento de datos, la lógica de negocio y la comunicación con bases de datos**, permitiendo que la aplicación funcione de manera eficiente y segura.



# Desarrollador Aplicación Web



## Desarrollador FullStack

### Desarrollador Frontend

- Se ocupa de la interfaz de usuario de la aplicación.
- Se asegura de que la aplicación sea fácil de usar, accesible y atractiva visualmente.
- Usa herramientas como HTML, CSS y JavaScript para crear la interfaz de usuario de la aplicación.

### Desarrollador Backend:

- Se ocupa de la lógica de negocio de la aplicación y la gestión de datos.
- Crea y mantiene la API y se encarga de integrar la aplicación con otras bases de datos y servicios externos.
- Usa herramientas como Node.js y Express.js.



# HTML, CSS y JavaScript

---

## HTML (HyperText Markup Language):

- HTML es como el esqueleto de una casa. Se utiliza para estructurar el contenido de una página web, como títulos, párrafos, imágenes y enlaces.
- Pensemos en HTML como las piezas de Lego que se ensamblan para crear una estructura sólida.

## CSS (Cascading Style Sheets):

- CSS es como la pintura y la decoración de la casa. Se encarga de cómo se ven las cosas en la pantalla, como colores, fuentes, márgenes y bordes.
- Imaginemos que CSS es el diseñador de interiores que hace que todo se vea atractivo y ordenado.

## JavaScript:

- JavaScript es como el cerebro de la casa. Le da vida a las cosas, como hacer que un botón cambie de color cuando lo tocas o mostrar un mensaje cuando alguien hace algo.
- Puedes ver a JavaScript como el mago que hace que las cosas sucedan cuando las necesitas.



# Cómo Trabajan Juntos

---



- HTML proporciona la estructura y el contenido básico de la página.
- CSS agrega estilo y diseño, haciendo que la página sea visualmente atractiva.
- JavaScript agrega interactividad, permitiendo que la página responda a las acciones del usuario.



# HTML: Funciones

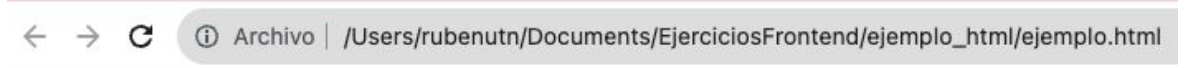
---



- **Estructura de la Página:** HTML se usa para dividir una página web en partes lógicas como encabezado, cuerpo y pie de página. Esto ayuda a los navegadores a entender cómo mostrar el contenido.
- **Elementos y Etiquetas:** En HTML, usamos "etiquetas" para indicar qué tipo de contenido está presente. Por ejemplo, `<h1>` se usa para encabezados grandes, `<p>` para párrafos y `<img>` para imágenes.
- **Enlaces y Navegación:** HTML permite crear enlaces para dirigir a los usuarios a otras páginas web o dentro de la misma página. Estos enlaces ayudan a la navegación.
- **Listas y Viñetas:** Puedes usar HTML para crear listas ordenadas (con números) y no ordenadas (con viñetas) para presentar información de manera organizada.

# HTML - Ejemplo

```
<!DOCTYPE html>
<html>
<head>
  <title>Título de la Pagina</title>
</head>
<body>
  <h1>¡Bienvenido a Mi Página Web!</h1>
  <p>Esta es una página de ejemplo creada con HTML.</p>
  <a href="https://www.ejemplo.com">Visita el Sitio de Ejemplo</a>
</body>
</html>
```



## ¡Bienvenido a Mi Página Web!

Esta es una página de ejemplo creada con HTML.

[Visita el Sitio de Ejemplo](https://www.ejemplo.com)

# CSS: Funciones

---



- **Estilo Visual:** CSS se encarga de cambiar el aspecto de los elementos HTML. Modificamos colores, fuentes, márgenes y tamaños para hacer que la página sea única.
- **Diseño y Formato:** Con CSS, podemos ajustar la disposición de los elementos en la página, como el espaciado entre párrafos y la alineación del texto.
- **Responsividad:** CSS nos permite crear diseños responsivos que se adaptan a diferentes tamaños de pantalla, desde computadoras de escritorio hasta teléfonos móviles.
- **Animaciones y Transiciones:** Con CSS, podemos añadir efectos de animación y transición a elementos en la página para hacerla más interactiva.

# CSS - Ejemplo

```
/* Estilo para los encabezados */
h1 {
  color: #007bff;
  font-size: 24px;
}

/* Estilo para los párrafos */
p {
  color: #333;
  font-size: 16px;
}

/* Estilo para los enlaces */
a {
  color: #ff9900;
  text-decoration: none;
}
```

← → ↻ ⓘ Archivo | /Users/rubenutn/Documents/EjerciciosFrontend/ejemplo\_html/ejemplo.html

**¡Bienvenido a Mi Página Web!**

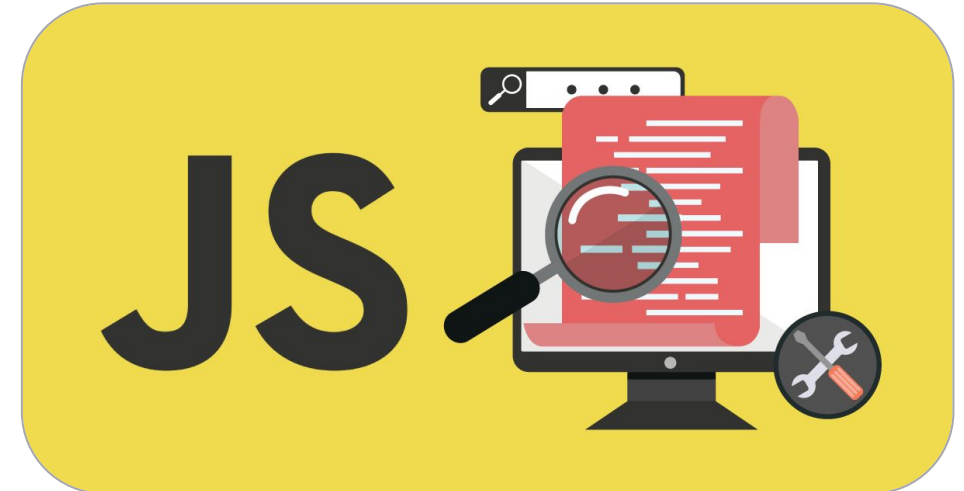
Esta es una página de ejemplo creada con HTML.

[Visita el Sitio de Ejemplo](#)

# ¿Qué es JavaScript (js)?

---

- JavaScript es un **lenguaje de programación** que se utiliza para conectar el frontend con el backend.
- JavaScript **funciona en el navegador** web del usuario y puede interactuar con el contenido de una página web.



# ¿Por qué es importante JavaScript?

- Lenguaje de programación versátil y dinámico.
- Ampliamente utilizado en el desarrollo de aplicaciones web.
- **Permite conectar frontend con backend y crear efectos visuales en el frontend.**
- Utilizado tanto en el frontend como en el backend.



# Sintaxis básica de JavaScript

- Javascript utiliza **variables** para almacenar datos y operadores para realizar operaciones matemáticas y lógicas.
- Las declaraciones de control de flujo, como "if" y "else", permiten tomar decisiones y ejecutar diferentes acciones en función de las condiciones.



La **sintaxis** se refiere a las reglas y convenciones que se utilizan para escribir un código en un lenguaje de programación.



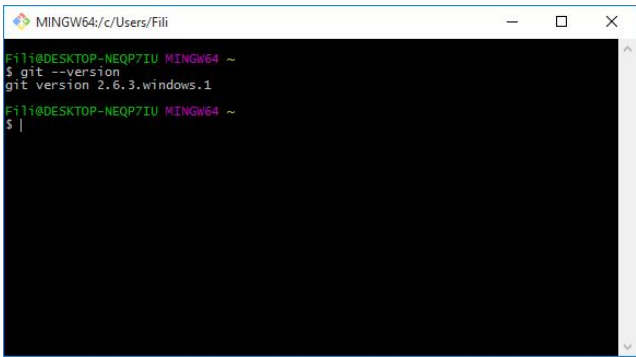
# Ejemplo: JavaScript



```
1  const calcularPromedio = (numeros) => {
2    let suma = 0;
3
4    for (let i = 0; i < numeros.length; i++) {
5      suma += numeros[i];
6    }
7
8    let promedio = suma / numeros.length;
9    return promedio;
10 }
11
12 // Prueba de la función
13 let numeros = [5, 8, 2, 10, 4];
14 let resultado = calcularPromedio(numeros);
15 console.log(resultado);
```



# Herramientas de Desarrollo

Editor de Código Fuente	Control de Versiones	Linea de Comando
 Visual Studio Code		

# Visual Studio Code (VSCode)

---

VSCode es un editor de código altamente popular y potente, además de uso libre.

Características destacadas de VSCode:

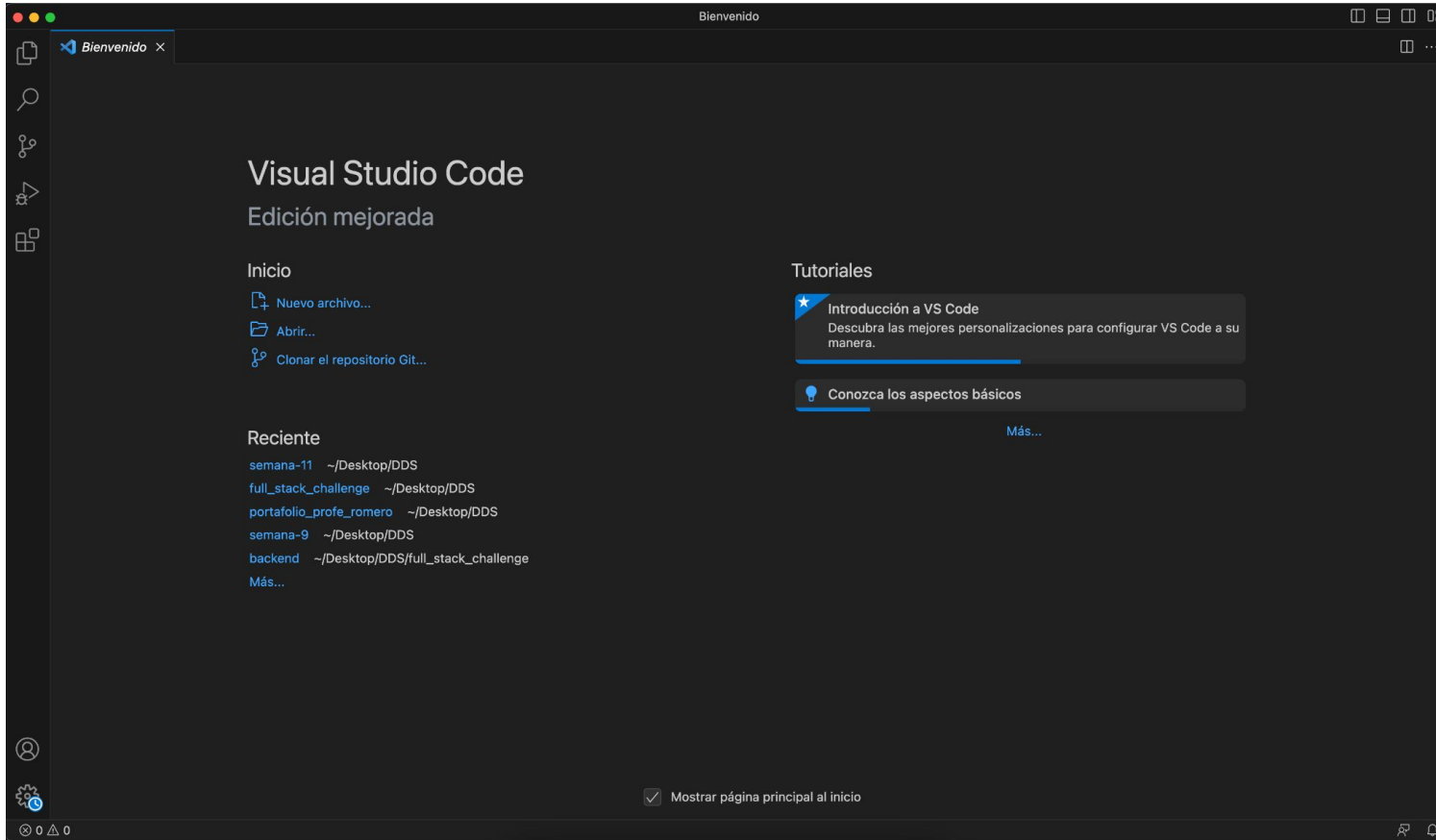
- **Resaltado de sintaxis:** facilita la lectura y comprensión del código.
- **Autocompletado inteligente:** ahorra tiempo al sugerir automáticamente código mientras escribes.
- **Depuración integrada:** permite identificar y solucionar errores de manera más eficiente.
- **Integración con control de versiones:** facilita la colaboración y el seguimiento de cambios.
- **Extensiones personalizables:** amplía la funcionalidad de VSCode según tus necesidades.



## Instalar VSCode

- Descargar archivo de instalación usando el siguiente link:
  - <https://code.visualstudio.com/>
- Seguir las instrucciones de instalación.

# Un paseo por Visual Studio Code (VSCode)



# Instalar Extensiones

---



- **ESLint:** es un linter que analiza estáticamente su código para encontrar problemas en función de un conjunto de reglas preconfiguradas.
- **JavaScript Booster:** El asistente de "solución rápida" predeterminado en VS Code.
- **Prettier:** es un formateador de código y aplica un estilo coherente que ha preconfigurado en su proyecto.
- **Auto Rename Tag:** Con esta extensión, cuando cambias el nombre de una etiqueta HTML o XML, se actualizará automáticamente la etiqueta de cierre correspondiente, lo que evita errores de estructura.
- **HTML CSS Support:** Esta extensión agrega funciones de autocompletado para HTML y CSS, lo que agiliza la escritura de código y reduce los errores tipográficos.
- **CSS Peek:** CSS Peek permite navegar desde el código HTML a las reglas CSS correspondientes y viceversa. Esto facilita la edición y el seguimiento de estilos en tu proyecto.
- **Auto Rename Tag:** Con esta extensión, cuando cambias el nombre de una etiqueta HTML o XML, se actualizará automáticamente la etiqueta de cierre correspondiente, lo que evita errores de estructura.
- **HTML Preview:** permite ver una vista previa en tiempo real de tus archivos HTML. Muestra cómo se verá tu página web en un navegador sin tener que cambiar de ventana. Esta extensión agiliza el proceso de desarrollo frontend al permitirte visualizar los cambios de HTML al instante.

# Control de versiones con Git



## Instalar Git

- Descargar archivo de instalación según el sistema operativo:
  - <https://git-scm.com/downloads>
- Seguir las instrucciones de instalación.

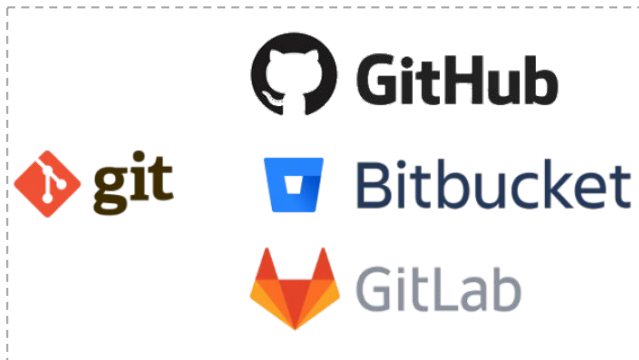
El control de versiones es esencial para el desarrollo colaborativo y el seguimiento de cambios en el código.

Permite a los desarrolladores **rastrear y gestionar los cambios en el código fuente** de manera eficiente.

Conceptos básicos de Git:

- **Repositorio:** almacena los archivos y su historial de cambios.
- **Commit:** registra los cambios realizados en el repositorio en un momento específico.
- **Branch:** crea ramas para trabajar en diferentes líneas de desarrollo.
- **Merge:** combina los cambios de diferentes ramas en una sola.

# Git y Plataformas de Alojamiento



## Resumiendo

- Git es la herramienta de control de versiones que permite el seguimiento de cambios en el código fuente.
- Las plataformas de alojamiento (GitLab, GitHub y Bitbucket) son lugares donde se pueden almacenar, colaborar y administrar los repositorios Git de manera más conveniente y efectiva.

GitLab, GitHub y Bitbucket son plataformas de alojamiento de repositorios basadas en Git.

## Alojamiento de Repositorios:

- Proporcionan un lugar centralizado donde los desarrolladores pueden almacenar y gestionar sus repositorios Git.
- Los repositorios alojados contienen el historial completo de cambios, ramas y etiquetas del proyecto.

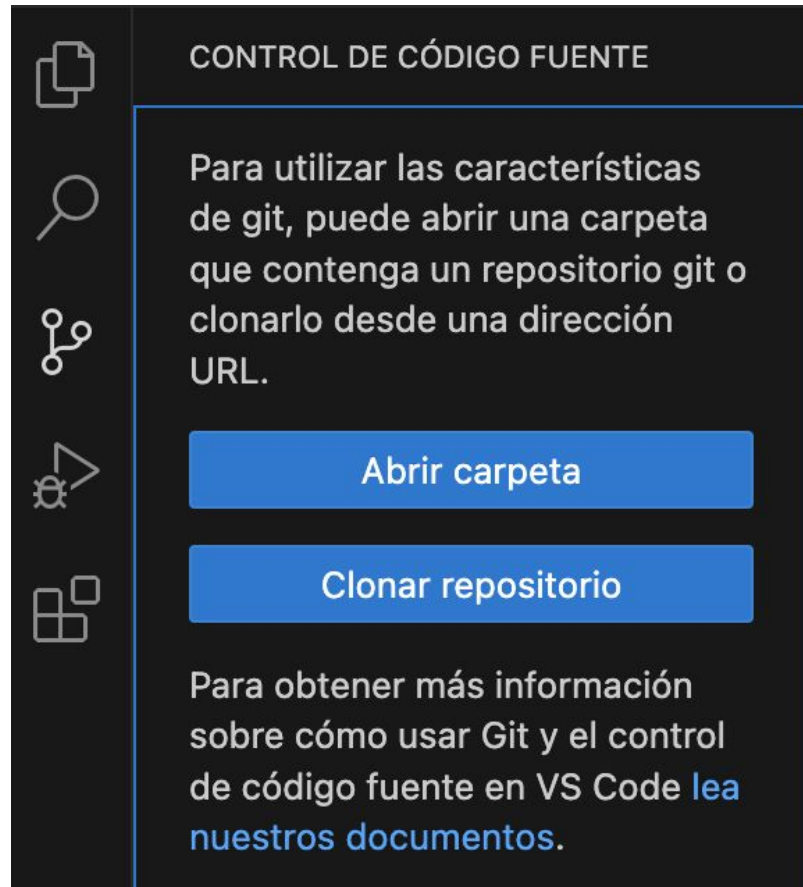
## Colaboración y Gestión de Proyectos:

- Permiten a los equipos de desarrollo colaborar en los proyectos de forma más efectiva.
- Ofrecen características como gestión de problemas, seguimiento de tareas, asignación de tareas y comentarios en línea.

## Integración Continua y Entrega Continua (CI/CD):

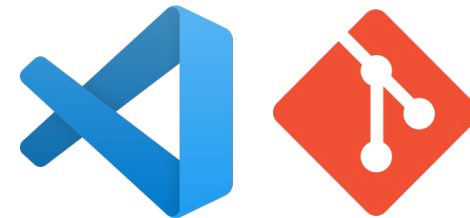
- Proporcionan herramientas para automatizar el proceso de construcción, pruebas y entrega de software.
- Facilitan la implementación rápida y confiable de nuevas versiones del software.

# Git en VSCode



- **Extensiones VSCode**

- **GitLens:** Proporciona una visualización avanzada de información de Git, incluyendo quién modificó cada línea de código.
- **Git History:** Permite ver el historial de cambios de Git dentro de VSCode y realizar un seguimiento detallado de los commits.
- **Git Graph:** Muestra el historial de commits y las ramas de Git en un gráfico interactivo.



# Línea de Comandos

---

- Las herramientas de línea de comandos son interfaces de texto que permiten **interactuar con el sistema operativo** y ejecutar comandos mediante instrucciones escritas en forma de texto.
- Estas herramientas ofrecen una forma poderosa y eficiente de interactuar con el sistema operativo, proporcionando una amplia variedad de funcionalidades y capacidades.
- Algunos ejemplos de herramientas de línea de comandos populares incluyen el :
  - **Terminal** en **macOS** y **Linux**.
  - **PowerShell** en **Windows**.





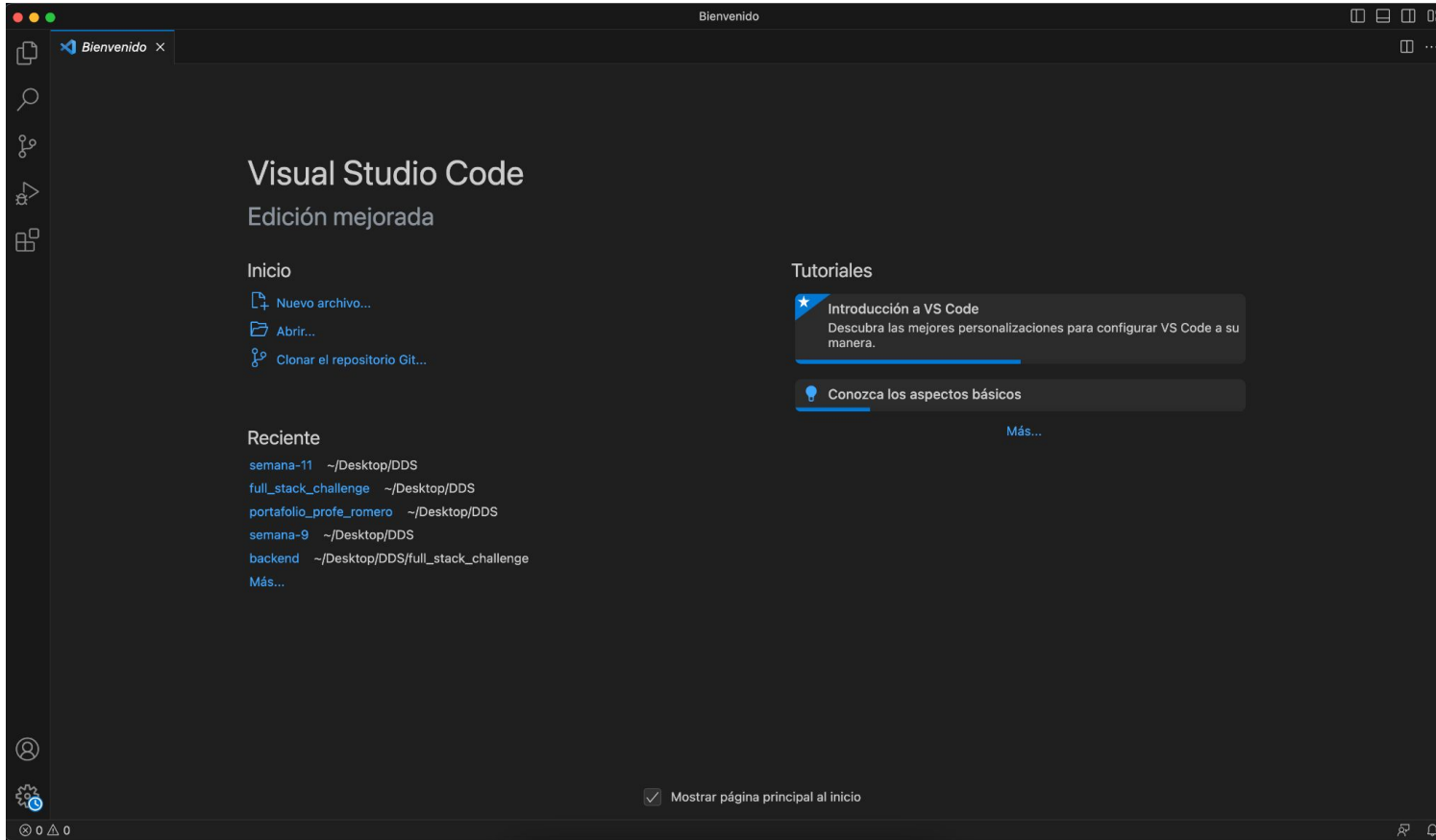
# Comandos más usados

---

- **cd**: Cambia el directorio actual.
  - Ejemplo 1: **cd documentos** cambia al directorio "documentos".
  - Ejemplo 2: **cd ..** cambia al directorio padre.
- **ls (en Linux/Mac) o dir (en Windows)**: Lista los archivos y directorios en el directorio actual.
  - Ejemplo: **ls** muestra los archivos y directorios en el directorio actual.
- **mkdir**: Crea un nuevo directorio.
  - Ejemplo: **mkdir nueva\_capeta** crea un directorio llamado "nueva\_capeta".
- **pwd**: Muestra la ruta completa del directorio actual (present working directory).
  - Ejemplo: **pwd** muestra la ruta completa del directorio actual en la línea de comandos.



# Línea de Comandos en VSCode



# Bueno, Vamo a Codea!!!!

¡Es hora de poner en práctica lo que hemos aprendido! Vamos a crear juntos una página web básica utilizando HTML y CSS. Este ejercicio nos ayudará a comprender cómo funciona la estructura y el estilo de una página web.

1. **Creamos un Archivo HTML:** Abrimos nuestro editor de código y creamos un nuevo archivo con extensión .html. Este es nuestro lienzo en blanco.
2. **Agregamos Estructura Básica:** Usamos la etiqueta `<!DOCTYPE html>` para definir el tipo de documento. Luego, añadimos las etiquetas `<html>`, `<head>` y `<body>` para estructurar la página.
3. **Añadimos Contenido:** En el `<body>`, añadimos un encabezado `<h1>` con un título para nuestra página. A continuación, creamos algunos párrafos `<p>` para el contenido.
4. **Aplicamos Estilos:** Creamos un archivo separado con extensión .css y lo vinculamos al archivo HTML mediante la etiqueta `<link>`. Dentro del archivo CSS, empezamos a añadir estilos para cambiar los colores y el diseño de la página.
5. **Usamos HTML Preview:** Utilizamos la extensión "HTML Preview" para ver cómo se ve nuestra página web en tiempo real mientras realizamos cambios en el código.
6. **Probamos en el Navegador:** También podemos abrir el archivo HTML en un navegador o utilizar la extensión Live Server para ver cómo se ve nuestra página. Hacemos ajustes en los estilos hasta que estamos satisfechos con el resultado.
7. **Exploramos y Jugamos:** Jugamos con diferentes propiedades CSS, como colores, fuentes y márgenes, para personalizar aún más nuestra página. Si queremos, agregamos imágenes.



# Actividad 1: Paso a Paso Aplicación Nodejs

- Seguir las instrucciones de la actividad publicada en la UVE.

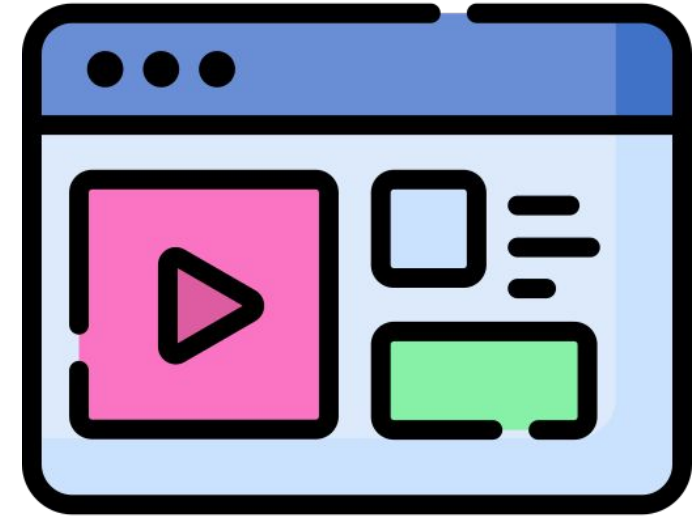


# Próxima Clase

---

## Fundamentos de HTML y CSS

- Estructura básica de un archivo HTML.
- Etiquetas HTML y sus atributos más comunes.
- Introducción a CSS y su sintaxis.
- Selectores y propiedades CSS básicas.



# MUCHAS TOTALES

---

**ANDÉN**  
Centro de Innovación  
y Emprendimientos Tecnológicos

SECRETARÍA DE  
EXTENSIÓN  
UNIVERSITARIA  
UTN - FRC

**SEU**

**\*UTN**  
Facultad Regional Córdoba

Agencia  
**CÓRDOBA  
JOVEN**

 **CÓRDOBA**  
entre todos