

# Introducción a las aplicaciones para internet

## Historia y evolución

- **Tim Berners-Lee:**
  - Inventor de la World Wide Web.
  - Implementó HTTP y fundó W3C (World Wide Web Consortium).
- **Hitos:**
  - **1995:** JavaScript introducido en Netscape para enriquecer páginas estáticas.
  - **1996:** Flash popularizado, discontinuado en 2020.
  - **2008:** HTML5 mejora soporte multimedia.

## Definición y características

- **Aplicación web:** Software distribuido que usa tecnologías web y es accesible a través de navegadores.
- **Características clave:**
  - Multiusuario, multilenguaje, adaptabilidad a diferentes dispositivos.
  - Seguridad y confidencialidad.
  - Manejo de grandes volúmenes de datos en diversos formatos.

## Ventajas de las aplicaciones web

- Sin necesidad de instalación, multiplataforma, portabilidad.
- Actualizaciones instantáneas y ahorro de espacio.
- Alta disponibilidad y resistencia a virus.
- Ideales para la colaboración.

## Desventajas

- Menor funcionalidad comparada con aplicaciones de escritorio.
- Dependencia del servidor y del proveedor de red.
- Limitaciones del navegador y del protocolo HTTP.
- Vulnerabilidad a problemas de ancho de banda.

## Comparativas clave

- **Sitio web:**
  - Informativo, requiere poca interacción (ej.: USPCEU).
- **Aplicación web:**
  - Enfocada en la interacción y funciones complejas (ej.: Netflix, banca online).
- **Servicio web:**
  - Diseñado para comunicación entre máquinas, sin interfaz visual.

## Arquitectura

- **Modelos:**
  - **Básico:** Servidor único que responde a peticiones.

- **Separación de funciones:** Uso de servidores específicos para bases de datos o contenido dinámico.
- **Alto rendimiento:** Balanceo de carga y múltiples servidores.
- **Alta disponibilidad:** Duplicación de elementos para minimizar caídas.

## Front-end

- **Tareas clave:**
  - Optimización del rendimiento web (tiempos de carga, interactividad).
  - Diseño responsivo y pruebas en múltiples navegadores.
  - Usabilidad, accesibilidad y SEO.
- **Tecnologías comunes:**
  - HTML, CSS, JavaScript.
  - Frameworks: React, Angular, Bootstrap.

## Back-end

- **Tareas clave:**
  - Testing automatizado y administración de bases de datos.
  - Escalabilidad y alta disponibilidad.
  - Seguridad y transformación de datos (JSON, XML).
- **Tecnologías comunes:**
  - Scripting: PHP, Python, Node.js.
  - Compilados: Java, C#, Go.

## Consideraciones finales

- Importancia de una buena arquitectura y diseño pensando en escalabilidad y disponibilidad.
- Coste de caídas: Ejemplo de Facebook en 2021 con pérdidas de \$164,000/minuto.