



UNIVERSIDAD NACIONAL DE CAJAMARCA

FACULTAD DE INGENIERÍA



ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS

Tema:

Practica 01: Fundamentos de Desarrollo Web

Docente:

VÁSQUEZ FERNÁNDEZ LISI JANET

Alumnos:

Bolaños Atalaya Romer

Caruajulca Tiglla Alex Eli


Curso:

PROGRAMACION APLICADA II

Cajamarca- Perú

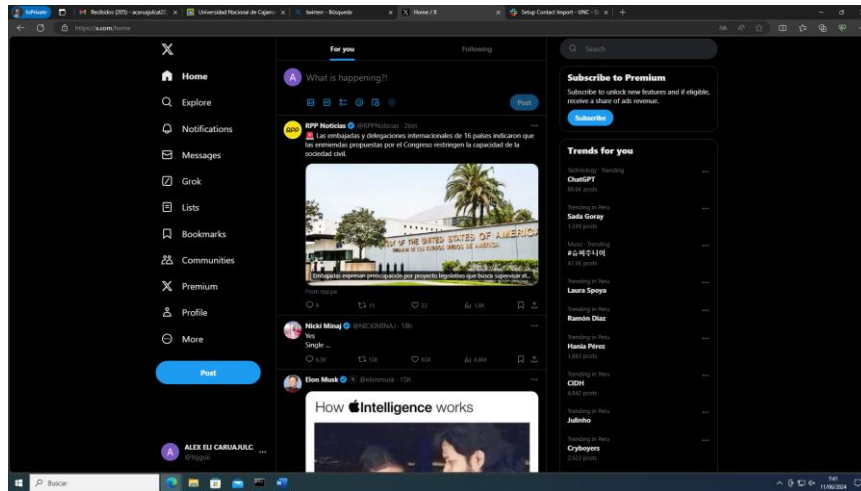
2024

1. CUADRO COMPARATIVO SITIO WEB VS APLICACIÓN WEB.

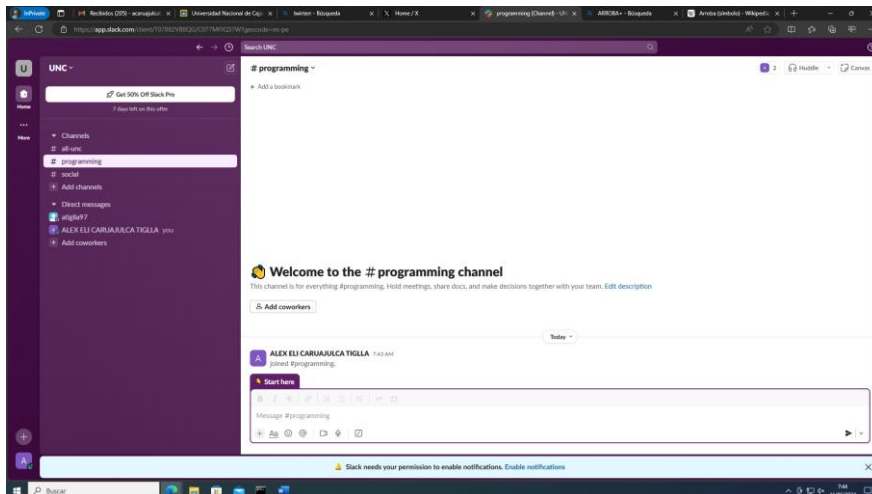
ASPECTO A COMPARAR	SITIO WEB	APLICACIÓN WEB
 Definición	Un sitio web es un conjunto de páginas web interrelacionadas que se acceden mediante un navegador. Generalmente estático o con interactividad limitada.	Una aplicación web es un software accesible a través de un navegador que ofrece funcionalidades y una experiencia de usuario más avanzadas, similar a las aplicaciones de escritorio o móviles.
Interactividad	Baja a moderada. Los usuarios principalmente consumen contenido.	Alta. Los usuarios interactúan intensamente con la aplicación, ingresando datos y ejecutando tareas complejas.
Tecnologías Usadas	HTML, CSS, JavaScript para la parte frontal. Puede usar PHP, ASP.NET, etc., para el backend.	HTML, CSS, JavaScript, frameworks front-end como React, Angular, o Vue.js, y back-end con Node.js, Django, Ruby on Rails, etc.
Actualización de Contenido	Fácil de actualizar con CMS (como WordPress, Joomla).	Requiere desarrollo y pruebas más complejas para actualizaciones, debido a la mayor interacción y lógica de negocio involucrada.
Experiencia de Usuario (UX)	Básica y orientada a la navegación y consumo de información.	Avanzada, con interfaces dinámicas y centradas en el usuario, a menudo usando AJAX para una experiencia sin recargas.
Acceso y Distribución	Accesible a través de cualquier navegador. Fácil de compartir a través de URL.	Accesible a través de un navegador, pero puede requerir autenticación y permisos. Algunas pueden ser empaquetadas como aplicaciones PWA (Progressive Web Apps).
Dependencia del Navegador	Depende completamente del navegador para funcionar.	También depende del navegador, pero algunas funcionalidades pueden ser independientes del navegador en el caso de las PWAs.
Rendimiento	Generalmente más rápido para carga inicial debido a su simplicidad.	Puede ser más lento al inicio debido a la mayor cantidad de recursos y lógica necesarios, pero más rápido y eficiente en la interacción continua.
Seguridad	Seguridad básica con HTTPS, autenticación básica.	Mayor enfoque en seguridad debido a la manipulación de datos y funcionalidades avanzadas, incluyendo autenticación, manejo de sesiones, y protección contra ataques XSS, CSRF, etc.
Escalabilidad	Menos complejo, generalmente fácil de escalar horizontalmente.	Requiere una arquitectura bien planificada para escalar, considerando balanceo de carga, base de datos y lógica de negocio.
Tiempo y Costo de Desarrollo	Menor tiempo y costo debido a su simplicidad y herramientas disponibles.	Mayor tiempo y costo debido a la complejidad y la necesidad de funcionalidades avanzadas y pruebas rigurosas.

1.1. Ejemplos de Aplicaciones Web

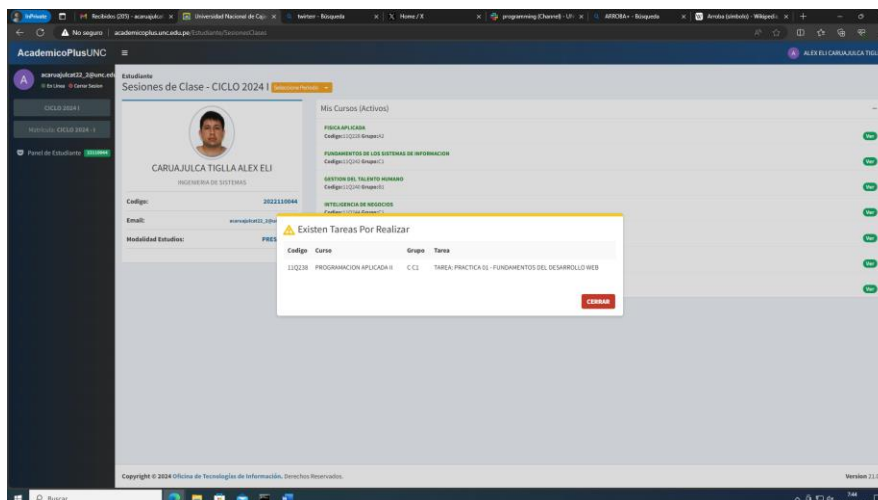
1.1.1. Twitter – Red Social



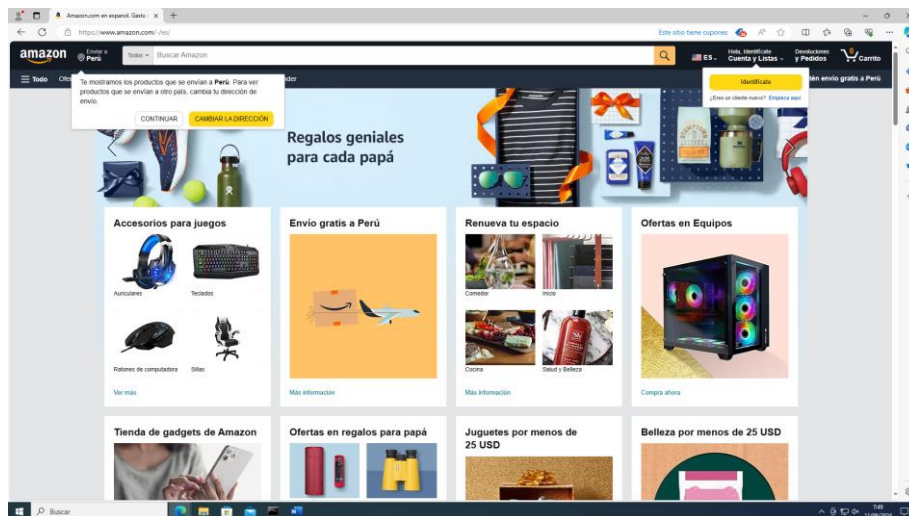
1.1.2. Slack – Chat App



1.1.3. Sistema Académico – Universidad Nacional de Cajamarca



1.1.4. Amazon



1.2. Ejemplos de Sitios Web

1.2.1. El peruano



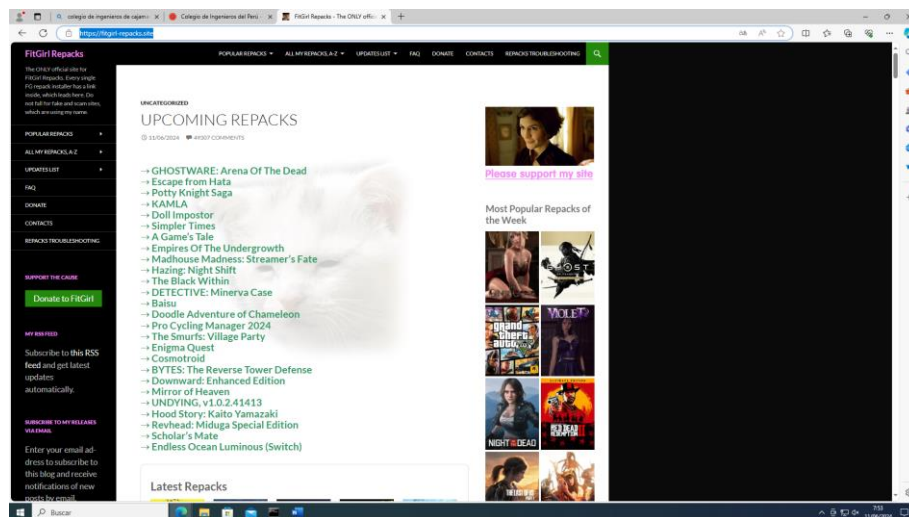
1.2.2. Portal – Universidad Nacional de Cajamarca



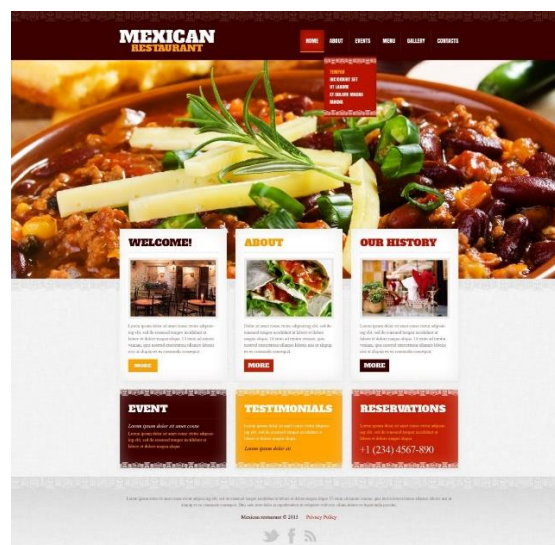
1.2.3. Colegio de Ingenieros de Cajamarca



1.2.4. Fitgirl – Sitio de Repacks



1.2.5. Mexican Restaurant



1.3. Referencias

<https://ed.team/blog/cuales-son-las-diferencias-entre-pagina-web-sitio-web-y-aplicacion-web>

<https://worldcampus.saintleo.edu/noticias/cuales-son-las-diferencias-entre-sitio-web-y-aplicacion-web>

<https://blog.ida.cl/estrategia-digital/diferencias-aplicacion-web-sitio-web/>

2. NAVEGADORES EN LOS ÚLTIMO AÑOS

2.1. Microsoft Edge

- 2.1.1. **Historia:** Microsoft Edge fue lanzado el 29 de julio de 2015 junto con Windows 10, como el sucesor de Internet Explorer. Originalmente utilizaba el motor de renderizado EdgeHTML, pero en enero de 2020 se relanzó con el motor Chromium para mejorar su compatibilidad y rendimiento.
- 2.1.2. **Características:** Edge ofrece integración con servicios de Microsoft, como Cortana y OneDrive. Con el cambio a Chromium, adquirió soporte para extensiones de Chrome, mejorando su funcionalidad y adoptabilidad.
- 2.1.3. **Participación de Mercado:** En 2024, Edge tiene una cuota de mercado global del 5.21%, con una participación significativa en el mercado de escritorio del 11.89%. La adopción móvil es menor, con solo el 0.24%.
- 2.1.4. **Privacidad:** Edge ha mejorado sus características de privacidad en las versiones recientes, incluyendo controles de rastreo y protección contra phishing y malware.

2.2. Brave

- 2.2.1. **Historia:** Brave fue lanzado en enero de 2016 por Brendan Eich, cofundador de Mozilla y creador de JavaScript. Brave se enfoca en la privacidad y seguridad, bloqueando anuncios y rastreadores por defecto.
- 2.2.2. **Características:** Brave utiliza el motor Chromium, lo que le permite soportar extensiones de Chrome. Ofrece Brave Rewards, que permite a los usuarios ganar criptomonedas (BAT) por ver anuncios opcionales.
- 2.2.3. **Participación de Mercado:** Aunque Brave tiene una cuota de mercado más pequeña en comparación con otros navegadores, ha ganado popularidad entre los usuarios preocupados por la privacidad y ha reportado millones de descargas y usuarios activos.
- 2.2.4. **Privacidad:** Brave es conocido por su enfoque en la privacidad, bloqueando rastreadores de terceros y anuncios intrusivos. También ofrece la opción de Tor integrado para navegación anónima.

2.3. Vivaldi

- 2.3.1. **Historial:** Vivaldi fue lanzado en abril de 2016 por Jon Stephenson von Tetzchner, cofundador de Opera. Está diseñado para usuarios avanzados que buscan un alto nivel de personalización.
- 2.3.2. **Características:** Vivaldi permite una personalización extensa de la interfaz, atajos de teclado y gestos del ratón. Ofrece características únicas como paneles web, notas integradas y capturas de pantalla.
- 2.3.3. **Participación de Mercado:** Vivaldi tiene una cuota de mercado pequeña pero cuenta con una base de usuarios leales que valoran su personalización y características avanzadas.

- 2.3.4. **Privacidad:** Vivaldi se enfoca en la privacidad, no rastreando a sus usuarios y ofreciendo herramientas para bloquear rastreadores y anuncios.

2.4. Samsung Internet

- 2.4.1. **Historia:** Samsung Internet fue lanzado en 2015 como el navegador predeterminado para dispositivos Samsung Galaxy. Está basado en Chromium y ha sido optimizado para dispositivos móviles.
- 2.4.2. **Características:** Ofrece características como un modo de alto contraste, bloqueadores de contenido, y sincronización con cuentas Samsung. También incluye un modo secreto y soporte para realidad virtual con Samsung Gear VR.
- 2.4.3. **Participación de Mercado:** En 2024, Samsung Internet tiene una participación del 2.66% en el mercado global y es especialmente popular en el mercado móvil, con una cuota del 4.48%.
- 2.4.4. **Privacidad:** Samsung Internet incluye características de privacidad como un bloqueador de rastreadores y la opción de navegar en modo incógnito.

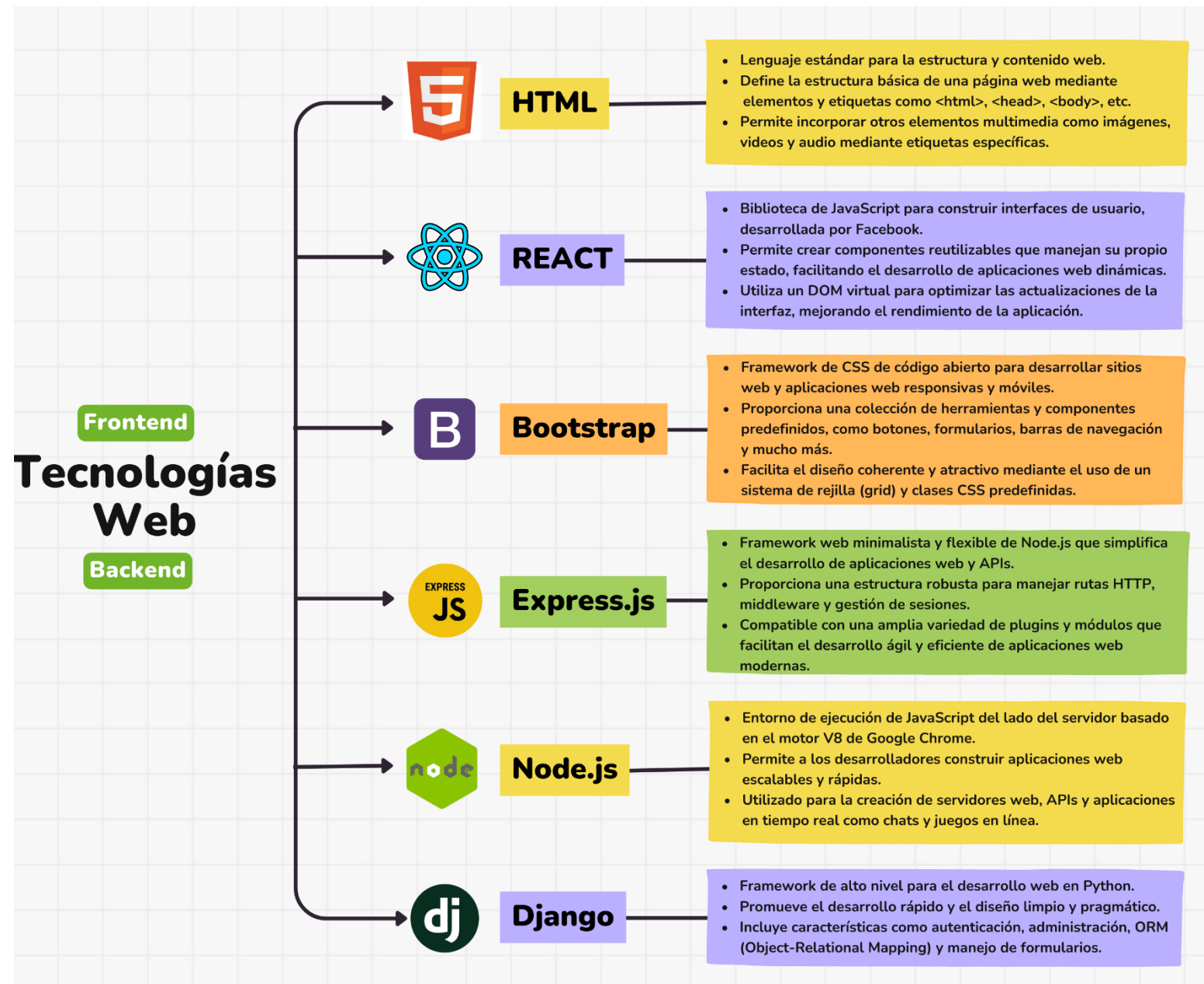
2.5. Referencias

<https://www.arsys.es/blog/navegadores-web>

<https://www.telefonica.com/es/sala-comunicacion/blog/historia-evolucion-navegadores-web/>

<https://www.stackscale.com/es/blog/top-navegadores-caracteristicas-comparativa-estadisticas/>

3. TECNOLOGÍAS DEL DESARROLLO WEB



3.1. Referencias

- <https://www.hostinger.es/tutoriales/que-es-html>
- <https://www.hostinger.es/tutoriales/que-es-react>
- <https://rockcontent.com/es/blog/bootstrap/>
- https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction
- https://www.w3schools.com/nodejs/nodejs_intro.asp
- <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>

TEMA: Evolución de las tecnologías web

1. Años 90: Los Primeros Pasos

HTML (HyperText Markup Language): Introducido por Tim Berners-Lee en 1991, permitió la creación de las primeras páginas web estáticas.

Navegadores: El primer navegador web, WorldWideWeb (posteriormente renombrado Nexus), fue seguido por otros como Mosaic (1993) y Netscape Navigator (1994), que popularizaron la web.

2. Finales de los 90 y Principios de los 2000: Interactividad y Expansión

JavaScript: Introducido en 1995 por Netscape, permitió la creación de contenido dinámico en las páginas web.

CSS (Cascading Style Sheets): Introducido en 1996, permitió la separación del contenido y la presentación visual, facilitando el diseño web.

Flash: Utilizado ampliamente para animaciones, juegos y videos interactivos.

Applets de Java: Permitían ejecutar aplicaciones dentro de los navegadores.

3. Mediados de los 2000: Web 2.0 y la Nube

Web 2.0

AJAX (Asynchronous JavaScript and XML): Permitió la creación de aplicaciones web más interactivas y rápidas mediante la actualización de partes de la página sin recargarla por completo.

Blogs y Redes Sociales: Surgieron plataformas como Blogger, MySpace y más tarde Facebook y Twitter, facilitando la participación de los usuarios y la creación de contenido.

Servicios en la Nube

Software as a Service (SaaS): Herramientas como Google Docs y Gmail comenzaron a ofrecer aplicaciones web complejas que reemplazaron a las aplicaciones de escritorio tradicionales.

4. Finales de los 2000 y 2010: La Web Móvil y HTML5

HTML5: Lanzado en 2014, incluyó nuevas características para soportar mejor multimedia y gráficos sin la necesidad de plugins externos como Flash.

CSS3: Introdujo nuevas capacidades de diseño y animación, mejorando significativamente la presentación visual de las páginas web.

Responsive Design: Metodologías y frameworks como Bootstrap permitieron la creación de sitios web que se adaptan a diferentes tamaños de pantalla.

Aplicaciones Web Progresivas (PWA): Combinan lo mejor de las aplicaciones web y móviles, ofreciendo experiencias rápidas y fiables.

5. 2020 y más allá: Blockchain, Realidad Aumentada y Virtual e Inteligencia artificial.

Blockchain

Blockchain y DApps (Aplicaciones Descentralizadas): Introducción de tecnologías que permiten aplicaciones descentralizadas, aumentando la seguridad y la transparencia.

Realidad Aumentada (AR) y Realidad Virtual (VR)

WebXR: Un API que permite la creación de experiencias de realidad aumentada y virtual directamente en el navegador.

RDF, OWL y SPARQL: Tecnologías que permiten a los datos ser compartidos y reutilizados a través de aplicaciones, comunidades y fronteras empresariales, haciendo que la web sea más "inteligente".

Inteligencia Artificial y Machine Learning

IA y ML: Integración de IA y ML en las tecnologías web para mejorar la personalización, la búsqueda y la automatización de tareas.

Referencias

[Evolución de las tecnologías web a lo largo del tiempo \(keepcoding.io\):](https://keepcoding.io/blog/evolucion-de-las-tecnologias-web/)

<https://keepcoding.io/blog/evolucion-de-las-tecnologias-web/>

<https://eva.fing.edu.uy/mod/resource/view.php?id=81130>

<https://www.ceupe.mx/blog/evolucion-de-la-web.html>

Mougayar, W. (2016). *El blockchain en los negocios: Promesas, prácticas y aplicaciones de la próxima tecnología de Internet*. Wiley, traducido al español.

O'Reilly, T. (2005). *¿Qué es Web 2.0?: Patrones de diseño y modelos de negocio para la próxima generación de software*. Disponible en línea en la página de O'Reilly.

TEMA: Evolución de los frameworks para desarrollo web

ASP (Active Server Pages) - 1996

- **Descripción:** Introducido por Microsoft, permitió la creación de aplicaciones web dinámicas en el servidor usando lenguajes de script como VBScript y JScript.
- **Impacto:** Marcó el inicio de las aplicaciones web dinámicas y fue ampliamente adoptado en entornos empresariales.

Ruby on Rails (2004)

- **Descripción:** Introducido por David Heinemeier Hansson, popularizó el concepto de "convención sobre configuración" y "DRY" (Don't Repeat Yourself).
- **Impacto:** Facilitó el desarrollo rápido de aplicaciones web y tuvo una gran influencia en futuros frameworks.

Django (2005)

- **Descripción:** Framework de alto nivel para Python, diseñado para construir aplicaciones web rápidamente con un enfoque en la simplicidad y la reutilización de código.
- **Impacto:** Introdujo características como el ORM y un potente sistema de administración, influyendo en otros frameworks.

ASP.NET (2002)

- **Descripción:** Evolución de ASP, ASP.NET es una plataforma de desarrollo web unificada con herramientas y bibliotecas para aplicaciones web robustas.
- **Impacto:** Introdujo la separación de código y diseño con "code-behind" y el uso de controles de servidor.

jQuery (2006)

- **Descripción:** Biblioteca de JavaScript que simplificó la manipulación del DOM, el manejo de eventos y las animaciones.
- **Impacto:** Se convirtió en una herramienta estándar debido a su facilidad de uso y compatibilidad con múltiples navegadores.

ASP.NET MVC (2009)

- **Descripción:** Basado en el patrón Model-View-Controller (MVC), proporcionó una mayor separación de preocupaciones y flexibilidad en ASP.NET.
- **Impacto:** Adoptó prácticas modernas como la inyección de dependencias y mejor control sobre el HTML.

AngularJS (2010)

- **Descripción:** Framework de JavaScript desarrollado por Google que permitió la creación de aplicaciones web dinámicas mediante el uso de plantillas declarativas y enlace de datos bidireccional.
- **Impacto:** Popularizó el desarrollo de aplicaciones de una sola página (SPA).

React (2013)

- **Descripción:** Biblioteca de JavaScript desarrollada por Facebook para construir interfaces de usuario, introdujo componentes y un DOM virtual.
- **Impacto:** Revolucionó la construcción de interfaces de usuario, adoptando un enfoque basado en componentes.

Vue.js (2014)

- **Descripción:** Framework progresivo de JavaScript creado por Evan You, centrado en la integración gradual y una curva de aprendizaje suave.
- **Impacto:** Se hizo popular por su flexibilidad, rendimiento y facilidad de integración.

ASP.NET Core (2016)

- **Descripción:** Reescritura completa de ASP.NET, es un framework de código abierto y multiplataforma para aplicaciones web modernas.
- **Impacto:** Introdujo un pipeline de middleware modular, unificó ASP.NET MVC y Web API, y soporta el desarrollo en múltiples sistemas operativos.

Svelte (2016)

- **Descripción:** Framework de JavaScript creado por Rich Harris, enfocado en la compilación en lugar de la ejecución en el navegador.
- **Impacto:** Mejoras significativas en rendimiento y experiencia de desarrollo simplificada.

Next.js (2016)

- **Descripción:** Framework basado en React desarrollado por Vercel, facilita el desarrollo de aplicaciones React con renderizado del lado del servidor y generación de sitios estáticos.
- **Impacto:** Herramienta popular para crear aplicaciones web rápidas y optimizadas para SEO.

Nuxt.js (2016)

- **Descripción:** Framework basado en Vue.js para la creación de aplicaciones universales (isomórficas) con renderizado del lado del servidor.
- **Impacto:** Mejora el rendimiento y la experiencia de usuario similar a Next.js.

Blazor (2018)

- **Descripción:** Parte de ASP.NET Core, permite la creación de aplicaciones web interactivas usando C# y .NET en lugar de JavaScript, ejecutándose en el navegador con WebAssembly.
- **Impacto:** Permite a los desarrolladores .NET construir aplicaciones web completas utilizando el mismo lenguaje y ecosistema.

REFERENCIAS

Ruby on Rails. <https://rubyonrails.org/>

Django Project. <https://www.djangoproject.com/>

ASP.NET Core. <https://docs.microsoft.com/en-us/aspnet/core/>

React. <https://reactjs.org/>