

DESARROLLO WEB FULLSTACK -INTERMEDIO

Autor de contenido

Andrés Fernando Pineda Guerra



Tabla de Contenido



Presentación

En el curso de desarrollador Full Stack como componente intermedio, podrán adquirir las habilidades y lenguajes necesarios para el desarrollo web, enfocándose en sus grandes pilares, como lo son Front End, Back End, Diseño y modelamiento de aplicaciones y documentación de código.

El curso trata temas emergentes tales como, la seguridad informática, desarrollo de aplicaciones móviles, gestión de base de datos, todo esto basado en la metodología Scrum. De la misma manera, se hace énfasis en el manejo de proyectos tanto en los módulos de desarrollo como los módulos de gestión de proyectos de TI.

Objetivos del curso (competencias)



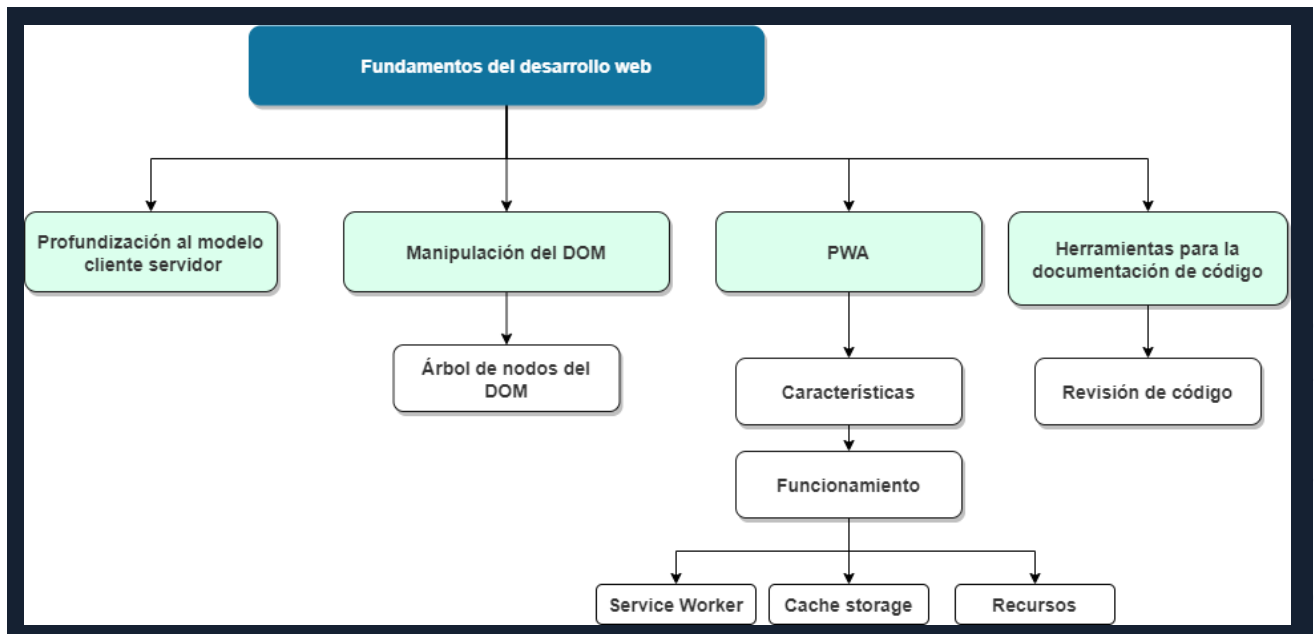
Objetivo general

Formar a los participantes en el desarrollo web en todo el ciclo de vida del software, en donde adquieran los conocimientos básicos para implementar soluciones web.

Objetivo específico

- Conocer los conceptos y teoría básica del desarrollo web.
- Identificar y conocer los diferentes lenguajes de programación y herramientas para el desarrollo web.
- Aplicar las diferentes tecnologías web, tendencias y herramientas en el desarrollo de soluciones web enfocadas a proyectos.
- Diseñar, desarrollar e implementar soluciones web básicas en donde se integren los componentes de Front End, Back End, seguridad, redes y buenas prácticas utilizando metodologías ágiles.
- Identificar y conocer los conceptos básicos para el desarrollo móvil, así como aplicar su desarrollo en aplicaciones básicas.

Mapa de contenido de la unidad



Módulo 2 Fundamentos del desarrollo web

Ideas clave

Modelo cliente - servidor, servidor, internet, clientes, funcionamiento en PHP.

DOM (Document object model), HTML, CSS, Java Script.

PWA (Progressive web application) contextualización general, aplicaciones web progresivas vs aplicaciones nativas, funcionamiento y componentes.

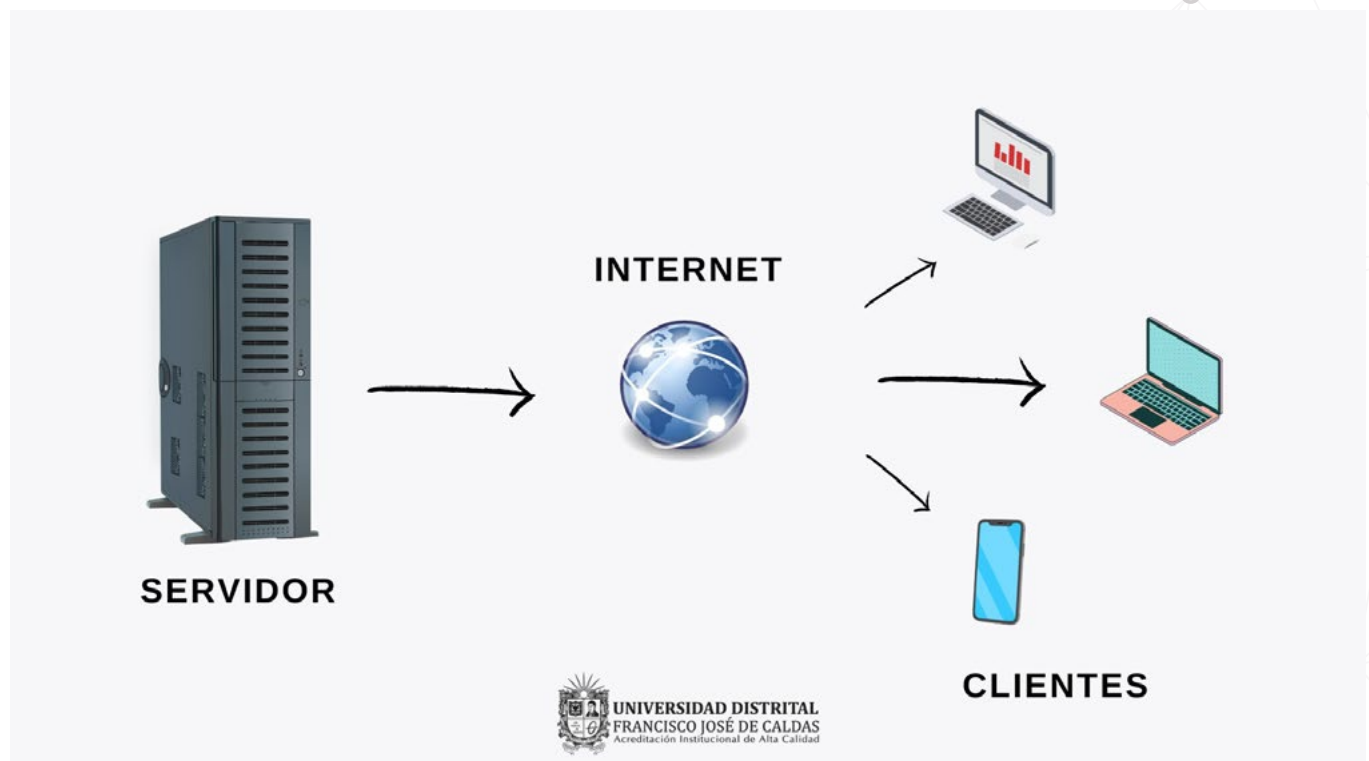
2.1. Profundización al modelo cliente servidor

Modelo cliente servidor

Es una de las principales arquitecturas utilizadas en los servicios web, este modelo de aplicación distribuida, permite que las tareas sean divididas entre los servidores y los clientes.

La red cliente-servidor es una red de comunicaciones en la cual los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta; y que los pone a disposición de los clientes cada vez que estos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor, de manera que en él se disponen los requerimientos provenientes de los clientes que tienen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son de sólo lectura y los que, por el contrario, pueden ser modificados, etc. Este tipo de red puede utilizarse conjuntamente en caso de que se este utilizando en una red mixta (Zafra et al., 2013).

Para entender mejor el modelo cliente - servidor, a continuación, se muestra una imagen de la arquitectura y sus componentes:



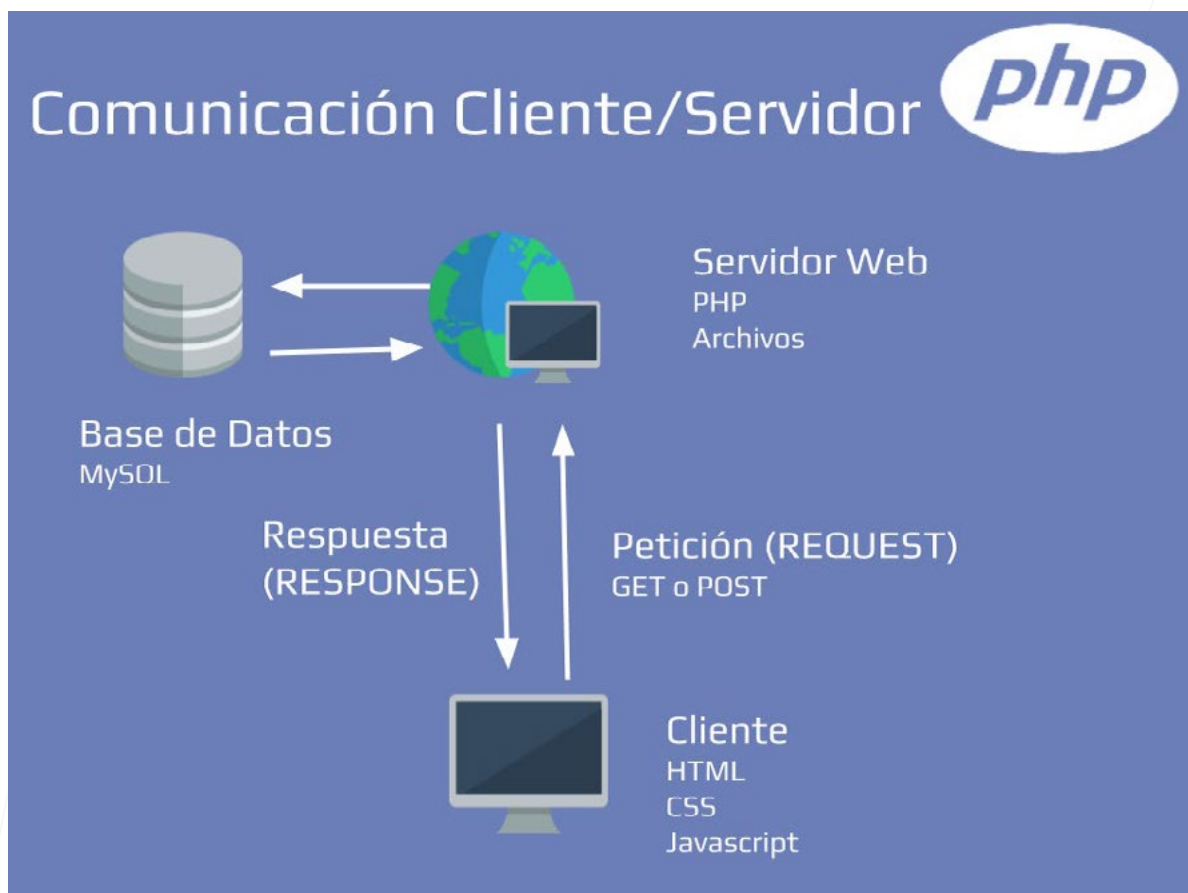
La arquitectura cliente servidor, permite centralizar la información y el control de los recursos e integridad de los datos que son controlados por el servidor, manteniendo separado lo afectado o realizado por algún usuario.

Ya que esta arquitectura “separa”, la parte del cliente con el servidor, facilita la escalabilidad y mantenimiento a futuro, ya que, si por ejemplo, se requiere de aumentar la capacidad o procesamiento a los datos, se puede realizar de manera independiente a la afectación de la capa del cliente.

Funcionamiento modelo cliente servidor PHP

PHP es uno de los lenguajes de programación más famosos y más utilizados para el desarrollo, debido a esto es importante saber que tipo de aplicación tiene este lenguaje en los diversos ambientes web, como por ejemplo en los Gestores de contenido, WordPress, Joomla, Drupal, osCommerce, Moodle, entre otros.

A continuación se presenta un modelo de comunicaciones cliente servidor en las aplicaciones web.



Programación web

El desarrollo o programación web se refiere a la creación, construcción y mantenimiento de sitios web. Es la creación de una aplicación que funciona a través de Internet, es decir, sitios web.

Lenguaje de programación

Un lenguaje de programación es un lenguaje informático que los programadores utilizan para desarrollar programas de software, scripts u otros conjuntos de instrucciones para que los ejecuten las computadoras.

Aunque muchos lenguajes comparten similitudes, cada uno tiene su propia sintaxis. Una vez que un programador aprende las reglas, la sintaxis y la estructura del lenguaje, escribe el código fuente en un editor de texto o IDE. Luego, el programador a menudo compila el código en un lenguaje de máquina que la computadora puede entender. Los lenguajes de secuencias de comandos, que no requieren un compilador, utilizan un intérprete para ejecutar la secuencia de comandos.

Historia

Los sistemas cliente-servidor empezaron a emerger en Estados Unidos cerca a los 80s cuando la informática pasó de grandes mainframes a procesamiento distribuido utilizando múltiples estaciones de trabajo o computadoras personales. Las empresas adoptaron rápidamente los sistemas cliente-servidor, que se convirtieron en la columna vertebral de su infraestructura de comunicaciones y automatización de oficinas.

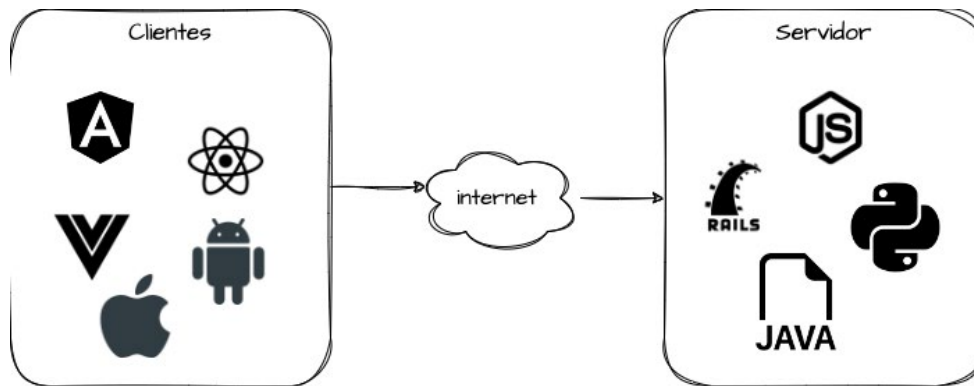
¿Qué es la arquitectura cliente servidor?

La arquitectura cliente-servidor define una aplicación distribuida que divide la carga de trabajo entre los proveedores de un recurso o servicio (servidores) y los solicitantes del servicio (clientes). Cuando el computador del cliente envía una solicitud de datos al servidor a través de internet, el servidor acepta el proceso solicitado y entrega los paquetes de datos solicitados al cliente. Los clientes no comparten ninguno de sus recursos. Ejemplos de modelo cliente-servidor son correo electrónico, World Wide Web, etc.

Backend y Frontend

Cuando se escucha hablar de backend, generalmente se refiere al lado del servidor. Se llama "backend" porque el usuario no ve esta parte, ya que solo ve la interfaz de usuario de la aplicación. Siempre que sirva una respuesta al dispositivo solicitante, se puede llamar como servidor. Lo contrario de es frontend (lado del cliente), esto es lo que ve el usuario, por ejemplo: la interfaz de usuario de la aplicación móvil o la página web que se muestra en su navegador web. Esta interfaz gráfica es implementada por medio de

tecnologías como: HTML, CSS y Javascript. La comunicación entre cliente-servidor se da por medio del protocolo HTTP (Hypertext Transfer Protocol) y estableciendo contratos en formato JSON (JavaScript Object Notation).

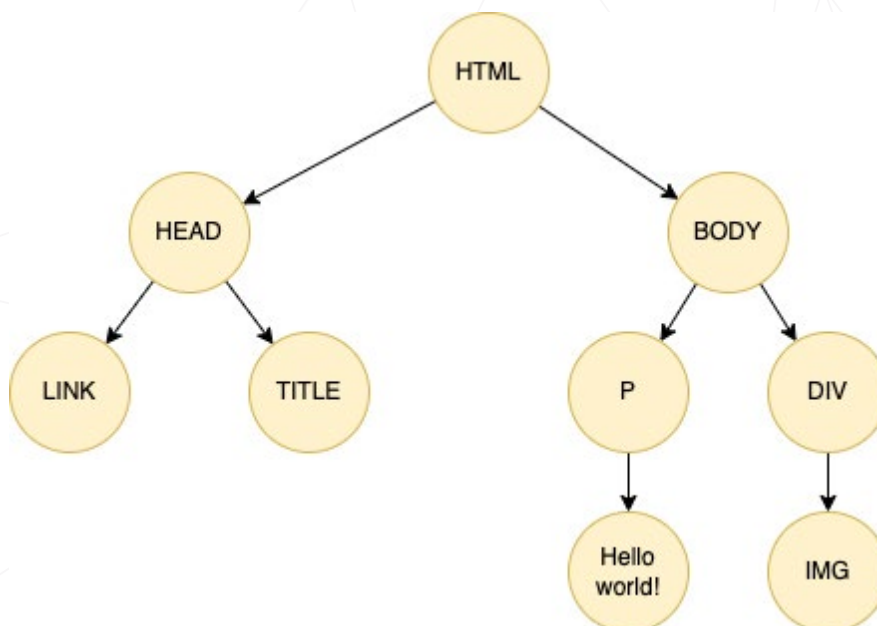


Arquitectura Cliente - Servidor

2.2. Manipulación del DOM

DOM (Document Object Model)

DOM es una forma de representar la página web de una manera jerárquica estructurada para que sea más fácil para los programadores y usuarios navegar por el documento. Con DOM, podemos acceder y manipular fácilmente etiquetas, ID, clases, atributos o elementos de HTML utilizando comandos o métodos proporcionados por el objeto del documento. Usando DOM, JavaScript obtiene acceso a HTML y CSS de la página web y también puede agregar comportamiento a los elementos HTML. Así que, básicamente, Document Object Model es una API que representa e interactúa con documentos HTML.



Árbol de nodos del DOM


```
<html>
  <head>
    <link href="style.css" rel="stylesheet">
    <title>Title</title>
  </head>
  <body>
    <p>Hallo Welt!</p>
    <div></div>
  </body>
</html>
```

Documento HTML

```
paragraphs = document.getElementsByTagName ("p");
// paragraphs[0] es el primer elemento <p>
// paragraphs[1] es el segundo elemento <p>, etc.
alert (paragraphs [0].nodeName);
```

Acceso al DOM con javascript. (MDN)

¿Qué es el DOM en el diseño web?

En el diseño web, JavaScript se utiliza para controlar el comportamiento de las páginas. El DOM vincula JavaScript al código HTML fuente, lo que permite que JavaScript ejecute sus funciones en elementos HTML individuales. Esta interacción JavaScript-HTML crea la experiencia de un diseño web interactivo.

Los objetos no solo son buenos para organizar datos, sino que también permiten que los programas tengan fácil acceso a piezas específicas de información en ellas. De regreso a nuestro objeto casa, digamos que quieres editar el elemento «encimera» de una casa en particular. Para hacerlo, tu programa necesita encontrar la casa en cuestión, recorrer el objeto árbol hacia el objeto cocina, luego cambiar el elemento «encimera» de «cuarzo» a «mármol». Linda elección.

Del mismo modo, el DOM le da a JavaScript acceso a casi cualquier elemento en un documento HTML, desde la página entera hasta una pequeña línea de texto. Cuando algo ocurre en una página web después de que se cargue (por ejemplo, dar clic a un botón abre un formulario o hace que un elemento cambie de color o tamaño),

es el código JavaScript que apunta a un elemento específico a través del DOM y hace el cambio. El código JavaScript puede localizarse en su propio archivo .js o dentro del propio documento HTML.

Para un ejemplo básico del DOM en acción, prueba esta demostración de W3Schools. El panel de la izquierda contiene el código fuente, y el de la derecha muestra el código como se ve en la página web.

`document.getElementById('demo').style.display='block'`

En el código anterior:

`document` es la forma en que JavaScript apunta al objeto archivo HTML a través del DOM. Le dice al navegador «quiero ir al archivo HTML a realizar un cambio».

`.getElementById('demo')` apunta al elemento de la página con el ID «demo». Este es el elemento que JavaScript desea modificar. Para llegar ahí, JavaScript comienza en la etiqueta `<html>` más alta, luego se mueve a través del DOM hasta que llega al elemento con el texto «¡Hola, JavaScript!»

`.style.display='block'` es la acción. Una vez que el elemento al que se apunta es encontrado, revela el texto.

Las páginas web complejas ejecutan estos llamados cientos de veces (a medida que los usuarios interactúan con ellas). JavaScript utiliza el DOM en cada una de esas ocasiones.

2.3. PWA

Introducción a las PWA

En la actualidad, los dispositivos móviles se han convertido en la principal herramienta de comunicación e información del ser humano, permitiendo que las personas puedan tener acceso a diferentes aplicaciones y sitios web para llevar a cabo diferentes tipos de transacciones. Sin embargo, al momento de pensar en desarrollar un sitio web, los desarrolladores se encuentran con la problemática de adaptar la visualización, optimización de uso y correcta funcionalidad de los despliegues para los ambientes en dispositivos móviles (Responsive design). Adicionalmente, se presenta la problemática de poder brindar a los usuarios móviles una excelente experiencia de usuario al momento de acceder a los sitios web desplegados.

Una de las alternativas óptimas para que un sitio web funcione en todos los dispositivos y que además cuente con una excelente experiencia de usuario son las Aplicaciones web progresivas (PWA). PWA permite integrar desarrollos web, con aplicaciones nativas de dispositivos móviles.

Se han evidenciado los beneficios que se perciben al usar aplicaciones web progresivas en desarrollos web nativos, gestionados en WordPress, Joomla, drupal, node js, angular o react js; Dentro de los beneficios que se perciben se encuentran: la disminución del consumo de datos de los dispositivos, la optimizan la caché y el uso de recursos como notificaciones, cámara, localización, entre otros (Castell Ferreres, 2020).

Contextualización General

¿Qué son las aplicaciones web progresivas?

Las aplicaciones web progresivas son formas de ver los sitios web embebidos o con características de una aplicación nativa, es decir, son tecnologías que permiten mediante código web nativo (HTML, CSS y JavaScript), utilizar recursos propios de un dispositivo, como lo es: notificaciones push, geolocalización, cámara, micrófono, entre otras. Teniendo en cuenta esto, se puede identificar la similitud de las aplicaciones web progresivas con las aplicaciones móviles nativas (Alemu, s. f.).

Aplicaciones web progresivas VS Aplicaciones móviles nativas.

Las aplicaciones web progresivas han tomado mucha fuerza en los últimos años, ya que se han integrado por grandes empresas como Google, Apple, Facebook, Twitter, entre otros. En la Tabla 1 se presentan los aspectos más relevantes de la PWA en comparación con las aplicaciones nativas.

A continuación, se verán reflejados los aspectos principales y más esperados por los usuarios.

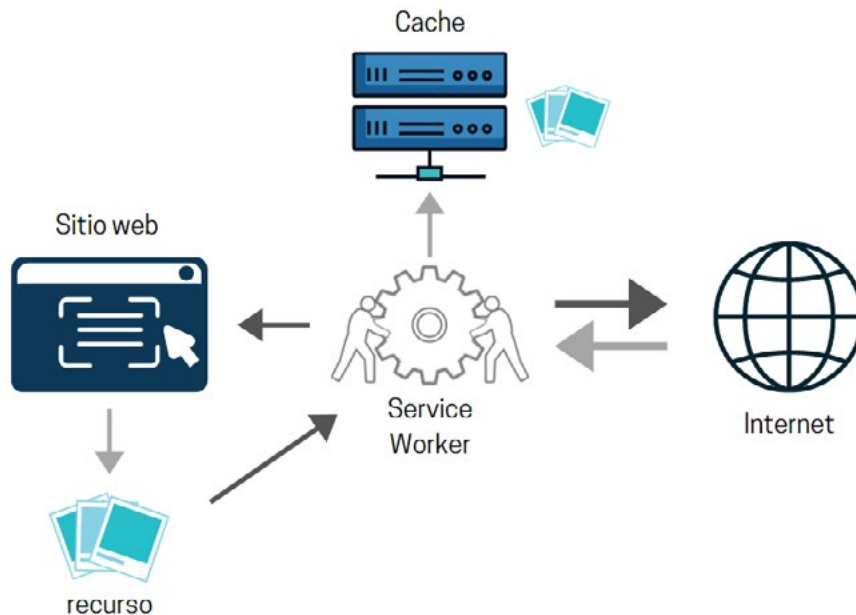
Aspecto	Aplicaciones web progresivas	Aplicaciones nativas
Tiempos de desarrollo (TD)	TD menores debido a que usa tecnologías de desarrollo web en lugar de lenguajes estructurados y complejos.	Desarrollos específicos para dispositivos móviles, conllevando el uso de tecnologías como java o kotlin, incrementando los tiempos de desarrollo comparado con las PWA. Si se requiere el desarrollo para diversos sistemas operativos en algunos casos se tiene que hacer un doble desarrollo para obtener la compatibilidad.
Icono	Ambos desarrollos generan un icono propio en el dispositivo del usuario, generando mejor experiencia de uso.	
Consumo de recursos Sitio de descarga.	<p>Utilizan recursos del navegador, optimizando el espacio al evitar su descarga directa al dispositivo. Mediante el service worker se almacena datos en la caché, proporcionando acceso al momento de utilizar la aplicación.</p> <p>Se realiza directamente del navegador en el url del sitio web.</p>	<p>Requieren de una instalación obligatoria para su uso, ocupando espacio en el dispositivo y el uso de recursos es mucho más alto.</p> <p>Se hace desde un sitio de descarga de aplicaciones como play (Android) store o App Store (Apple). Su instalación consume recursos de almacenamiento en el dispositivo.</p>

Mantenimiento	Es más sencillo, debido a que, si se actualiza el sitio web principal, este realiza una actualización en tiempo real de todos los servicios que le pertenezcan.	Requiere de compilar las actualizaciones autorizadas por los usuarios para poder verse reflejada.
Utilización de recursos	PWA no tiene acceso a todos los sensores, aunque por su gran impacto en el mercado se tiene la posibilidad de ampliar el acceso de recursos. Actualmente se tiene acceso a: notificaciones push, geolocalización, cámara, micrófono y sensor de movimiento.	Con las aplicaciones nativas se tiene acceso a todos los recursos del dispositivo (teniendo en cuenta los permisos que da el usuario) como lo son: contactos, gps, cámara, notificaciones push, en general a todos los sensores del dispositivo.
Costos	Los costos tanto de desarrollo, diseño y mantenimiento son bajos comparados con las aplicaciones nativas debido a que se hace en el servidor directamente, y a nivel de conocimientos es más costoso los lenguajes de alto nivel como java, comparados con HTML, css y JavaScript.	Varían dependiendo de la complejidad de la actualización, en su mayoría son costosos por personal, pruebas y mantenimiento.

¿Cómo funcionan las aplicaciones web progresivas?

Para el uso de las aplicaciones web progresivas se debe contar con buenas prácticas de programación, Responsive design definido y control total de los archivos del sitio web (Bjørn-Hansen et al., 2018). Las aplicaciones web progresivas se basan en modelos como el descrito en la siguiente imagen

Figura 1 - Modelo de funcionamiento PWA



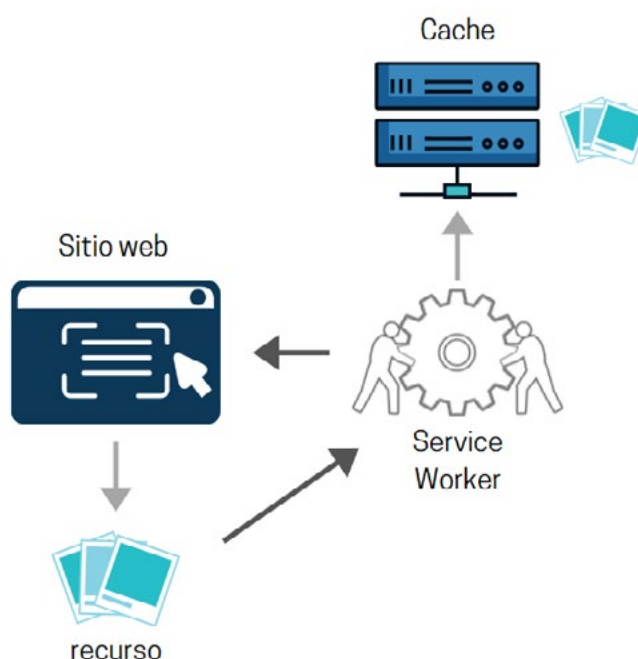
En este modelo, en primera instancia el sitio web debe contener unos recursos fijos y/o estáticos, como, por ejemplo: videos, textos, imágenes, componentes, entre otros.

Figura 2 - Especificación de sitio web y recursos



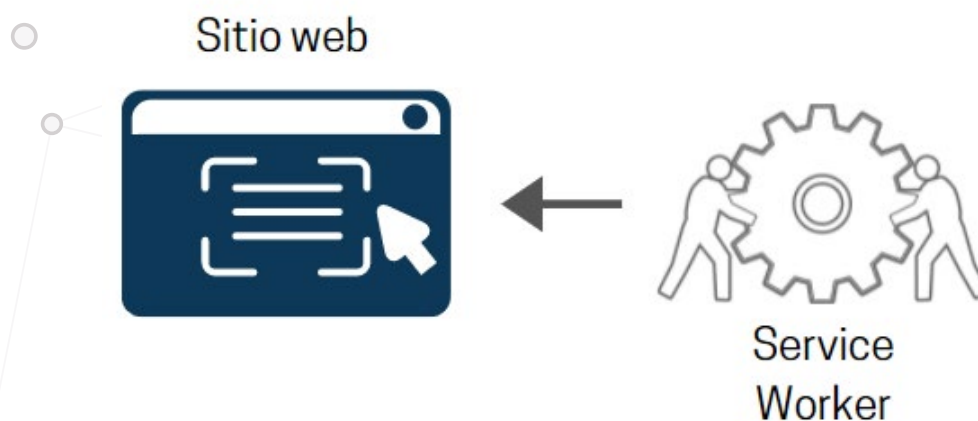
Seguidamente, el recurso se dirige al service worker, el cual permite direccionar estos recursos a la caché (convirtiéndola en cache storage al ser particionada), dicho recurso se almacena en la caché, tal como se muestra en la siguiente figura .

Figura 3 - Funcionamiento del service worker



Posteriormente, el service worker verifica la conexión e interpreta esa conexión como flujo de datos y devuelve los servicios al sitio web. En el caso de que no se tenga una conexión a internet, el service worker redirige a la memoria caché y devuelve aquellos recursos almacenados allí y permiten tener una previsualización del sitio web, lo cual mejora la experiencia de usuario. Por último, el service worker devuelve los datos al sitio web obtenidos, bien sea de la memoria caché o de internet (servidor en donde se tenga alojado el sitio web) directamente.

Figura 4 - Interpretación service worker y navegador



2.4. Herramientas para la documentación de código

Herramientas para la documentación de código

Con las herramientas para la documentación de código podemos evaluar cualquier nuevo código con el fin de detectar errores, posibles fallos, entre otros.

Tener habilidades de documentación del código permite realizar revisión de código, y fomenta el trabajo colaborativo ó en equipo.

¿Por qué es importante la revisión del código?

El proceso de revisión del código es crítico porque nunca forma parte del plan de estudios formal en las escuelas. Puedes aprender los matices de un lenguaje de programación y la gestión de proyectos, pero la revisión de código es un proceso que evoluciona a medida que una organización progresa.

La revisión del código es crítica por las siguientes razones:

- Asegúrate de no tener errores en el código.
- Minimiza tus posibilidades de tener problemas.
- Confirma que el nuevo código se adhiere a las directrices.
- Aumentar la eficiencia del nuevo código.

Las revisiones de los códigos conducen a mejorar la experiencia de otros miembros del equipo. Como un desarrollador senior suele realizar una revisión de código, un desarrollador junior puede utilizar esta información para mejorar su propia codificación.

¿Por qué es importante el uso de herramienta de revisión de código?

El principal resultado de un proceso de revisión del código es aumentar la eficiencia. Aunque estos métodos tradicionales de revisión de códigos han funcionado en el pasado, es posible que se esté perdiendo eficiencia si no se ha cambiado a una herramienta de revisión de códigos. Una herramienta de revisión de códigos automatiza el proceso de revisión de códigos para que un revisor se centre únicamente en el código.

Una herramienta de revisión de código se integra con su ciclo de desarrollo para iniciar una revisión de código antes de que el nuevo código se fusione en la base de código principal. Puedes elegir una herramienta que sea compatible con tu pila tecnológica para integrar sin problemas en tu flujo de trabajo.

Documentación en línea

- Funciona como un sistema de almacenamiento central: Puede proteger toda la información relacionada con el producto y crear un acceso central para los clientes y los equipos de soporte al que pueden consultar cada vez que tengan dudas sobre una característica o cómo funciona.
- Ahorra tiempo y energía: Cuando crea un depósito central de información de productos, ayuda tanto a los clientes como a los representantes de soporte a ahorrar tiempo y energía en la búsqueda de la solución adecuada. Incluso puede evitar agotamientos frecuentes y ayudar a los equipos de soporte a enfocarse en problemas urgentes de soporte.
- Promueve el compromiso y la retención: Las herramientas de documentación de software también ayudan a mantener a los clientes comprometidos con la marca a lo largo de su viaje. Los clientes pueden evitar buscar respuestas sobre el producto o su funcionalidad en Internet. Regresarán al recurso que han desarrollado y permanecerán conectados con la marca. Esta es una excelente manera de mantenerlos comprometidos con la marca y retenerlos al mismo tiempo.
- Ofrece un conocimiento profundo del producto: Desde la configuración hasta todas las funciones, puede encontrarlas en la documentación en línea. Incluso puede encontrar cuándo se lanzó la nueva versión y en qué se diferencia de la anterior. Ayuda a los prospectos en el proceso de toma de decisiones, ya que también incluyen casos de uso y estudios de casos de diferentes industrias.
- Mejora la experiencia de atención al cliente: Uno de los mayores beneficios de tener documentación en línea para su negocio es mejorar la experiencia de servicio al cliente. Puede ayudar a su equipo a ahorrar tiempo y dirigir a los clientes a una solución que responda a sus preguntas sin esperar en colas o anticipar un correo electrónico.

Document360 - de pago

Ofrece una solución completa de documentación en línea que incluye, entre otros, búsqueda simple, administrador de categorías, administración de etiquetas, plantillas de artículos y más. Su interfaz de usuario de arrastrar y soltar lo ayuda a enumerar todas las categorías y sus subcategorías en el orden que desee.

Viene con búsqueda impulsada por IA, alojamiento privado, capacidades de control de versiones, copia de seguridad y restauración, etc. El editor enriquecido es algo que debe consultar. Puede elegir entre un Markdown y un WYSIWYG editor.

Otros materiales para profundizar

Recursos de video



Ing. René Domínguez (Director). (2020, junio 10). PWA Web App Básica Cordova | HTML | CSS | Bootstrap | JQuery | Ajax | PHP | MySQL. <https://www.youtube.com/watch?v=g5tXye4nb18>

UskoKruM2010 (Director). (2021, mayo 26). Modelo Cliente - Servidor ¿Qué es? Explicación Sencilla <https://www.youtube.com/watch?v=KA7Ngcgth0Q>

yacklyon (Director). (2021, julio 27). Curso de JAVASCRIPT MANIPULACION DEL DOM - Que es el DOM. <https://www.youtube.com/watch?v=BwFygAN0iok>

Referencias bibliográficas de la unidad



Jensen, S. H., Madsen, M., & Møller, A. (2011). Modeling the HTML DOM and browser API in static analysis of JavaScript web applications. Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering, 59-69. <https://doi.org/10.1145/2025113.2025125>

Mariano, T., & Tomás, V. (2012). Encriptación y compresión de archivos en un modelo cliente-servidor. <https://www.uaeh.edu.mx/investigacion/producto.php?producto=5554>

Zafra, A., Gibaja, E., Luque, M., & Ventura, S. (2013). Diseño de aplicaciones cliente/servidor para el aprendizaje de las tecnologías de comunicación. Iniciación a la Investigación. <https://revistaselectronicas.ujaen.es/index.php/ininv/article/view/1748>

Alemu, Y. (s. f.). Analysis of deploying a React PWA on Google Play store using Trusted Web Activity. 61.

Biørn-Hansen, A., Majchrzak, T. A., & Grønli, T.-M. (2018). Progressive Web Apps for the Unified Development of Mobile Applications. En T. A. Majchrzak, P. Traverso, K.-H. Krempels, & V. Monfort (Eds.), Web Information Systems and Technologies (pp. 64-86). Springer International Publishing. https://doi.org/10.1007/978-3-319-93527-0_4

Castell Ferreres, G. (2020). Desarrollo e implementación de una aplicación web progresiva (PWA). <https://upcommons.upc.edu/handle/2117/192433>



UNIVERSIDAD DISTRICTAL
FRANCISCO JOSÉ DE CALDAS
Acreditación Institucional de Alta Calidad