

Selección de arquitecturas herramientas de programación



Pablo Suárez García

1



2ºDAW

ÍNDICE

Diapositiva 1: Portada

Diapositiva 2: Índice

Diapositiva 3: Desarrollo de Software

Diapositiva 4: Desarrollo de Software

Diapositiva 5: Tipos de Aplicaciones

Diapositiva 6: Desarrollo Aplicaciones Web

Diapositiva 7: Lado Cliente/Servidor

Diapositiva 8: Arquitectura

Diapositiva 9: Arquitectura

Diapositiva 10: Tecnología del lado Cliente

Diapositiva 11: Tecnología del lado Cliente

Diapositiva 12: ¿Por qué JavaScript?

Diapositiva 13: Herramientas desarrollo cliente

Diapositiva 14: Navegador

Diapositiva 15: Editor de Código

Diapositiva 16: Editor de Código

Diapositiva 17: Editor de Código

Diapositiva 18: Editor de Código

Diapositiva 19: Interprete de JavaScript

Diapositiva 20: Otras herramientas útiles

Pablo Suárez García

2



2ºDAW

1.1 DESARROLLO DE SOFTWARE

El desarrollo de aplicaciones es el proceso de creación de un programa informático o un conjunto de programas para realizar las diferentes tareas que cubren una necesidad.

Esta necesidad puede haberse generado en cualquier parte de una organización: mostrar detalle del producto, aplicar descuentos, generar informes, etc.

Todas las necesidades comparten una necesidad común: automatización y aumento de la eficiencia



1.1 DESARROLLO DE SOFTWARE

Cada proceso de creación de aplicaciones siguen los mismos pasos

1. **Planificación y análisis de requisitos:** Es una de las etapas más importantes. La realizan los miembros más experimentados del equipo junto con el cliente.
2. **Definición de requisitos:** Definir y documentar claramente los requisitos y obtener la aprobación del cliente.
3. **Diseño de la arquitectura:** Se aborda la solución desde diferentes puntos de vista hasta encontrar el enfoque que se considera óptimo
4. **Construcción o implementación:** Si se realizan las fases anteriores correctamente, la generación debe lograrse. La elección del lenguaje de programación está vinculado al tipo de software que se está desarrollando.
5. **Pruebas:** Los defectos del producto se informan, rastrean, reparan y se vuelve a comprobar hasta que el producto llegue a los estándares de calidad definidos.
6. **Despliegue y mantenimiento:** Se despliega la aplicación en producción, se corrigen los defectos detectados por el contexto detectados por la ejecución.



1.1.1 TIPOS DE APLICACIONES

Las clasificaciones de los tipos de aplicaciones que se han venido utilizando atendía a los tipos de dispositivos para los que estaban diseñadas (escritorio, web y móvil).

1. **Desarrollo de aplicaciones personalizadas:** Se adaptan a requisitos específicos que el software comercial no cubre.
2. **Desarrollo rápido de aplicaciones:** Utiliza enfoques incrementales y desarrollo paralelo de módulos para acortar plazos.
3. **Desarrollo de aplicaciones low-code:** Plataformas con interfaces visuales que facilitan crear y modificar apps con poco código, incluso sin ser programador.
4. **Desarrollo de aplicaciones móviles:** Diseñadas para funcionar en cualquier dispositivo móvil.
5. **Desarrollo de aplicaciones de bases de datos:** Gestionan, organizan y analizan información con criterios, cálculos e informes.
6. **Desarrollo de aplicaciones empresariales:** Orientadas a grandes organizaciones, integran automatización y gestión de datos a gran escala.
7. **Desarrollo de aplicaciones web:** Alojadas en servidores, accesibles desde cualquier dispositivo mediante internet.



1.1.2 DESARROLLO DE APLICACIONES WEB

El desarrollo de aplicaciones es la construcción y el mantenimiento de sitios web. Por decirlo de una forma sencilla, es el trabajo que conduce a que un sitio web se vea bien, funcione rápido y con una experiencia de usuario satisfactoria

Los desarrolladores lo consiguen utilizando una enorme variedad de lenguajes de programación.



Pablo Suárez García

6



2ºDAW

1.2 LADO CLIENTE Y LADO SERVIDOR

LADO CLIENTE (FRONT-END)	LADO SERVIDOR (BACK-END)
<p>Se trata de todo aquello que el usuario puede ver, lo que puede interactuar y experimentar.</p> <p>El objetivo de estos desarrolladores es programar las partes del sitio web que son visibles para el usuario.</p> <p>No puede funcionar de forma independiente exceptos sitios estáticos.</p> <p>Framework usado React.</p>	<p>Las acciones realizadas por el usuario son analizadas, recuperadas y devueltas a través de los programas almacenados y ejecutados en el servidor que los aloja.</p> <p>El trabajo de estos desarrolladores incluye vincular todos los aspectos entre sí con las bases de datos.</p> <p>Framework más usado Django</p>



1.2.1 ARQUITECTURA

Las aplicaciones web de hoy en día utilizan las tecnologías de ambos lados siguiendo una arquitectura de tres niveles.

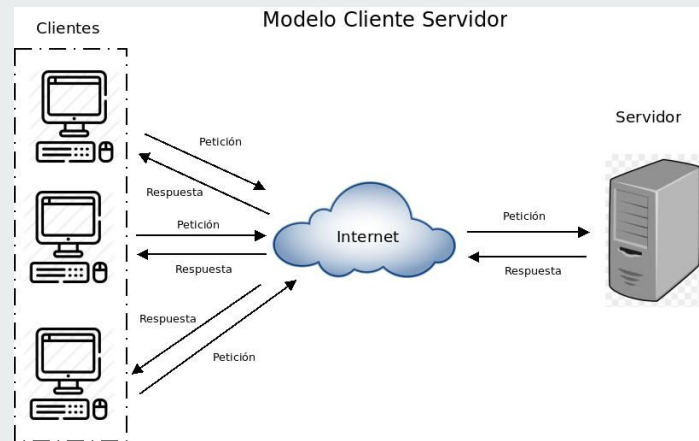
Se trata de la arquitectura de software predominante para las aplicaciones cliente-servidor.

Se organizan en tres niveles informáticos:

-Lógicos y físico: El nivel de presentación o interfaz de usuario.

-El nivel de aplicación: donde se procesan los datos

-El nivel de datos: donde se almacenan y gestionan los datos asociados a la aplicación



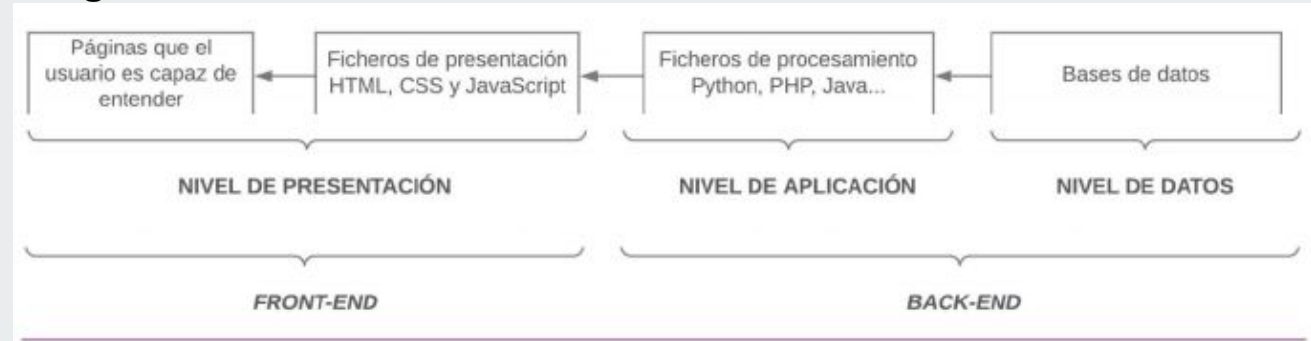
1.2.1 ARQUITECTURA

Cada nivel puede ser desarrollado por un equipo de desarrollo diferente:

-Nivel de presentación: Define la interfaz de usuario y la comunicación con la aplicación, donde el usuario final interactúa con la aplicación

-Nivel de aplicación: Es el corazón de la aplicación, tiene toda la información recopilada del nivel de presentación donde se procesa a veces con la colaboración del nivel de datos, utilizando la lógica.

-Nivel de datos: Es donde se almacena y administra la información procesada por la aplicación. Este puede ser un sistema de gestión de base de datos relacional como MySQL o un servidor NOSQL como MongoDB.



1.2.2 TECNOLOGÍAS DEL LADO CLIENTE

Las aplicaciones web es el marco de ejecución en el navegador.

Para el desarrollo web se necesita dominar:

-HTML: Es un lenguaje que se utiliza en la construcción de páginas web. Presenta elementos que proporcionan el diseño básico para un sitio web

-CSS: Es el lenguaje que se utiliza para modificar la estética de los elementos HTML. Incluye soporte para múltiples navegadores

-JavaScript: Es el lenguaje de desarrollo del lado cliente más importante y popular. Proporciona flexibilidad y capacidad de respuesta rápida al sitio web.



1.2.2 TECNOLOGÍAS DEL LADO CLIENTE

Existen tecnologías para completar, acelerar y optimizar el desarrollo como:

-SASS: Es un procesador de CSS, es una tecnología con la que pueden generarse de forma automática con hojas de estilo que contienen elementos propios del lenguaje

-jQuery: Mejora el procesamiento del código HTML, el manejo de eventos y animaciones

-vue.js: Es un framework creado para aplicaciones web compactas.

-AngularJS: Es un framework lanzado por Google que proporciona elementos más atractivos a las plantillas HTML y aumenta su rendimiento.

-React: Es otro framework lanzado por Facebook, que mejora los componentes de la interfaz de usuario y provee de mucho más dinamismo a las aplicaciones web.



1.2.3 ¿POR QUÉ JAVASCRIPT?

JavaScript ates algunos de los méritos de este lenguaje:

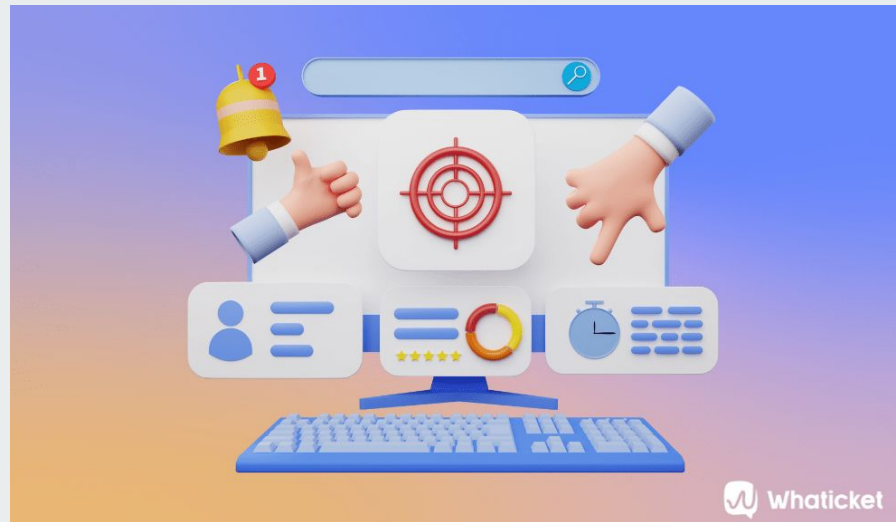
- Es muy fácil de implementar:** tan solo es necesario colocar el código en un documento HTML
- Funciona en todos los navegadores** que usan los usuarios de la web incluso cuando están desconectados
- Permite crear interfaces** altamente amigables que mejoran la experiencia del usuario y no tenga que esperar el servidor reaccionar y muestre otra página.
- Puede cargar contenido de forma asíncrona** en el documento si el usuario lo necesitas.
- Puede comprobar** lo que es posible hacer en un navegador y reaccionar en consecuencia
- Puede ayudar a solucionar** problemas de incompatibilidades entre navegadores y corregir problemas de diseño con CSS



1.3 HERRAMIENTAS PARA EL DESARROLLO EN EL LADO CLIENTE

Antes de iniciar el recorrido en el aprendizaje de JavaScript es necesario tener bien establecido un entorno de trabajo.

Existen docenas de herramientas que correctamente configuradas pueden llevar a otro nivel el desarrollo de aplicaciones web en entorno cliente. Cuando sea necesario aumentar la productividad, gestionar cantidad de ficheros o trabajar en equipos de desarrollo.

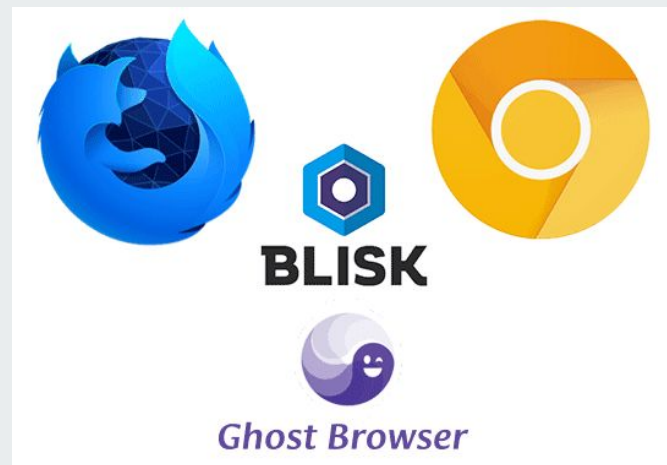


1.3.1 NAVEGADOR

El navegador web es de las principales herramientas, puesto que sin él no pueden verse las aplicaciones ejecutándose en su contexto.

De la misma manera, optimizar las aplicaciones para todos los navegadores, es muy complicado, ya que no compensa la cantidad de esfuerzo y recursos.

También es fundamental probar diferentes configuraciones del navegador para ajustarlo al mayor número de tamaños de pantalla posibles.



1.3.2 EDITOR DE CÓDIGO

Un editor de código tiene muchas funciones personalizadas especialmente diseñadas para satisfacer las necesidades de un desarrollador de software:

- Comprobación de errores** es lo que proporciona un primer nivel de corrección.
- Sugerencias de autocompletado** agiliza la escritura y acorta los tiempos de desarrollo
- Fragmentos de código** se configuran con aquellas porciones utilizadas con mucha frecuencia y que evitan reescribirlos constantemente
- Resultados de sintaxis** ayuda a identificar cada uno de los elementos de los programas.
- Navegación entre archivos y recursos** se ayuda a tener localizados todos los ficheros en sus ubicaciones correctas
- Funcionalidad extendida** a través de complementos específicos de las tecnologías con las que se trabaja en un proyecto en concreto



1.3.2 EDITOR DE CÓDIGO

Características de **Un editor de código** que pueden marcar la diferencia como:

- Funciones básicas imprescindibles:** resaltado de sintaxis, sangría automática, finalización automática, coincidencia de bloques y visualización de números de línea.
- Experiencia del usuario:** Reducir todo lo posibles las dificultades para escribir código y aumentar así la eficiencia en la producción de código final.
- Facilidad de aprendizaje:** Invertir tiempo en entender el editor cuando no se domina el lenguaje.
- Extensibilidad:** Proporcionar vías para extender las capacidades del editor con más funciones y herramientas adicionales.
- Velocidad:** La rapidez es clave a la hora de codificar y no todos los editores responden bien cuando se realizan ciertas tareas en equipos cortos de hardware.
- Sistema operativo:** Mucho mejor si el editor proviene de versiones de Windows, Mac y Linux.
- Compatibilidad con Git:** Sería deseable incorporar la funcionalidad de forma nativa como extensión
- Soporte de la comunidad:** Debe tener un lugar para hacer preguntas y obtener soporte técnico.



1.3.2 EDITOR DE CÓDIGO

Un editor de código en el peor momento y sin capacidad de soporte puede marcar diferencia entre un proyecto exitoso y otro fallido

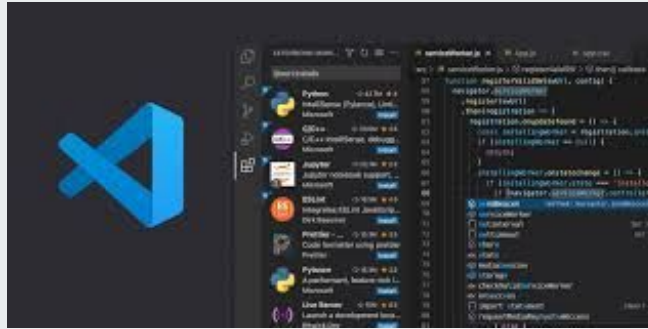


- Compatibilidad con lenguajes de programación:** Es prácticamente seguro que será necesario escribir ficheros con distintos lenguajes en un mismo fichero.
- Precio:** Existen muchos editores de alta calidad que son gratis, otro aportan funcionalidades premium con planes de pago.
- Atajos de teclado** ayudan mucho a los desarrolladores a crear fragmentados de código sin necesidad de usar el ratón constante
- Ventana de vista previa** puede ahorrar mucho tiempo evitando cambiar entre editor y navegador.
- Gestión amigable de errores:** los editores incorporan un sistema de coloreado e información ampliada de los errores.



1.3.2 EDITOR DE CÓDIGO

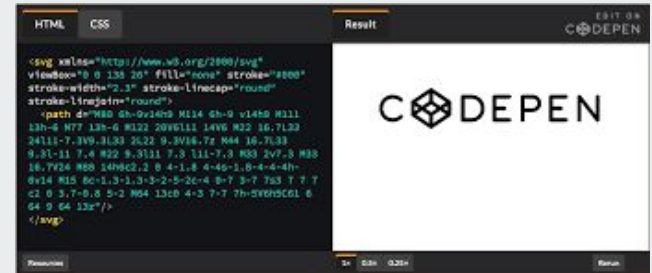
Editores de desarrollo integrados



Editores de código



Editores online



Pablo Suárez García

18



2ºDAW

1.3.3 INTÉRPRETE DE JAVASCRIPT

Para escribir código **JavaScript** es necesario tener el propio intérprete del lenguaje para saber si lo que se ha escrito es correcto. En algunos navegadores viene integrado, se debe comprobar si está habilitado porque en muchas ocasiones está deshabilitado. A continuación se explica como habilitarlo en los 4 navegadores:

-Chrome: Configuración -> Seguridad y privacidad -> Configuración de los sitios -> JavaScript.

-Safari: Herramientas -> Preferencias -> Seguridad -> JavaScript

-Edge: Opciones -> Mostrar opciones avanzadas -> Privacidad -> Configuración de contenido -> JavaScript -> Permitir que todos los sitios ejecuten JavaScript

-Firefox: Herramientas -> Opciones -> Contenido -> JavaScript



1.3.4 OTRAS HERRAMIENTAS ÚTILES

Node.js

Plataforma construida sobre el motor V8 de Google Chrome para el desarrollo fácil escalable de aplicaciones web



Pablo Suárez García

Git

Es el sistema de control de versiones más actualizado. Es útil para cualquier profesional que escriba código o necesite hacer un seguimiento de los cambios que se producen en los archivos



20



2ºDAW