

UNIDAD 1: INTRO AL DESARROLLO

ENTORNO

Un entorno de desarrollo es un conjunto de procedimientos y herramientas que se utilizan para desarrollar un código fuente o programa. Este término se utiliza a veces como sinónimo de entorno de desarrollo integrado (IDE), que es la herramienta de desarrollo de software utilizado para escribir, generar, probar y depurar un programa. También proporcionan a los desarrolladores una interfaz de usuario común (UI) para desarrollar y depurar en diferentes modos.

A la hora de elegir un entorno de desarrollo o IDE (Integrated Development Environment) es fundamental tener definido qué lenguaje de programación se va a utilizar tanto en el Frontend (la parte visible de la web) como en el Backend.

Niveles del entorno de desarrollo

El entorno de desarrollo normalmente tiene tres niveles de servidores, clasificados como desarrollo, montaje y producción. Los tres niveles juntos se denominan generalmente como el DSP.

- **Servidor de desarrollo:** Aquí es donde el desarrollador prueba el código y comprueba si la aplicación se ejecuta correctamente con ese código. Una vez que la implementación ha sido probada y el desarrollador considera que el código trabaja de forma correcta, la aplicación se mueve entonces al servidor intermedio de montaje.
- **Servidor de integración:** Este entorno se hace para que se vea exactamente igual que el entorno del servidor de producción. La aplicación se prueba en el servidor de ensayo para comprobar la fiabilidad y para asegurarse de que no falla en el servidor de producción real. Este tipo de pruebas en el servidor intermedio es el último paso antes de que la aplicación se despliegue en un servidor de producción. La aplicación tiene que ser aprobada con el fin de implementarla en el servidor de producción.
- **Servidor de producción:** Una vez realizada la aprobación, la aplicación se convierte en una parte de este servidor.

Características de los entornos de desarrollo

Los denominados IDE deben cumplir algunas características básicas para cumplir las expectativas del usuario. Una de ellas es que deben ser multiplataforma y tener una interfaz atractiva y práctica para trabajar. Deben facilitar el proceso integral de la programación y es importante además que cuenten con un asistente de ayuda y foros donde los usuarios puedan plasmar sus dudas.

Algunos de los entornos de desarrollo más utilizados son Eclipse o NetBeans. Eclipse, desarrollado inicialmente por IBM, es gestionado ahora por una fundación independiente sin ánimo de lucro que fomenta la creación de una comunidad de código abierto. Permite la instalación de extensiones y plugins que se pueden añadir en función de las necesidades del usuario.

Fuente: <https://www.arimetrics.com/glosario-digital/entorno-de-desarrollo>

INTERNET, WEB Y CLIENTE-SERVIDOR

Internet

Internet es una red de computadoras que se encuentran interconectadas a nivel mundial para compartir información. Se trata de una red de equipos de cálculo que se relacionan entre sí a través de la utilización de un lenguaje universal.

El concepto Internet tiene sus raíces en el idioma inglés y se encuentra conformado por el vocablo inter (que significa "entre") y net (proveniente de network que quiere decir "red electrónica"). Es un término que siempre debe ser escrito en mayúscula ya que hace referencia a "La Red" (que conecta a las computadoras mundialmente mediante el protocolo TCP/IP) y sin un artículo que lo acompañe (el/la) para hacerle referencia.

Fuente: <http://concepto.de/internet/#ixzz4zYUA5ee9>

WEB

Web es una palabra inglesa que significa red o telaraña. Se designa como 'la web' al sistema de gestión de información más popular para la transmisión de datos a través de internet. La web es el diminutivo de world wide web o www cuyas tecnologías para su funcionamiento (HTML, URL, HTTP) fueron desarrolladas en el año 1990 por Tim Berners Lee. Para usar la web es necesario tener acceso a internet y un navegador web, por la cual se solicita una página dinámica llamada también página web.

El navegador web como, por ejemplo, Google Chrome, se comunica con el servidor web mediante el protocolo web o HTTP (hypertext transfer protocol) para entregar la petición

deseada.

La web usa para la creación de sus páginas web el lenguaje HTML (hypertext markup language) que unificó la forma de buscar y crear información a través de internet. El conjunto de páginas web asociadas se sitúan en un sitio web como lo es, por ejemplo, Youtube.

Hipertexto

El hipertexto, por otro lado, es un conjunto estructurado de textos, gráficos, imágenes o sonidos unidos entre sí por enlaces o vínculos (links) y conexiones lógicas. Actualmente, el hipertexto está siendo reemplazado por la hipermedia, que también es un conjunto estructurado de diversos medios (texto, imagen, sonido), pero usados conjunta y simultáneamente (multimedia) y unidos entre sí por enlaces y conexiones lógicas para la transmisión de información.

Página web

La página web es la unidad informativa de la web, es decir, son documentos compuestos por textos, imágenes, audios o videos a los que se puede acceder a través de la word wide web (o www) empleando un navegador.

Como ya se mencionó, la información de las páginas web está generalmente en formato HTML. Asimismo, contienen enlaces que las relacionan con otras páginas cuyos contenidos se relacionan.

Sitio web

Un sitio web es un conjunto de páginas web que se encuentran relacionadas por el contenido y por un dominio en internet, y que constituyen una gran red de información. Como ejemplo, actualmente muchas empresas tienen un sitio web que funciona como una tarjeta de presentación o bienvenida para el público. Dentro de este sitio web se encuentran diversas páginas web con funciones específicas que amplían la información de esta. De allí que no se deba confundir página web con sitio web.

Fuente: <https://www.significados.com/web/>

Navegador WEB

El navegador web o navegador de internet es el instrumento que permite a los usuarios de internet navegar o surfear entre las distintas páginas de sus sitios webs preferidos. Se trata de un software que posee una interfaz gráfica compuesta básicamente de: botones de navegación, una barra de dirección, una barra de estado (generalmente, en la parte inferior de la ventana) y la mayor parte, en el centro, que sirve para mostrar las páginas web a las que se accede. Los principales navegadores web del mercado son: Microsoft Internet Explorer (actualmente Microsoft Edge), Firefox, Google Chrome, Opera y Safari.



Fuente: <https://www.informatica-hoy.com.ar/aprender-informatica/Que-es-un-navegador-web.php>

■ Buscadores WEB

Son los programas dentro de un sitio o página web, los cuales al ingresar palabras clave, operan dentro de la base de datos del mismo buscador y recopilan todas las páginas posibles, que contengan información relacionada con lo que se esté buscando.

Estos motores de búsqueda funcionan mediante el envío de "arañas", las cuales son pequeños robots que se dedican a rastrear todos los sitios web a lo largo y ancho de Internet. Otro programa, llamado un indexador, a continuación, lee estos documentos y crea un índice basado en las palabras contenidas en cada documento. Cada motor de búsqueda utiliza un algoritmo propietario para crear sus índices tales que, idealmente, sólo los resultados significativos se devuelven para cada consulta.



Fuente:

[https://es.wikipedia.org/w/index.php?title=Motor de b%C3%BAsqueda&oldid=103402577](https://es.wikipedia.org/w/index.php?title=Motor_de_b%C3%BAsqueda&oldid=103402577)

FRONT-END

Front-end es la parte de una aplicación que interactúa con los usuarios, es conocida como el lado del cliente. Básicamente es todo lo que vemos en la pantalla cuando accedemos a un sitio web o aplicación: tipos de letra, colores, adaptación para distintas pantallas (RWD), los efectos del ratón, teclado, movimientos, desplazamientos, efectos visuales... y otros elementos que permiten navegar dentro de una página web. Este conjunto crea la experiencia del usuario.

Como hemos dicho, el desarrollador front-end se encarga de la experiencia del usuario, es decir, en el momento en el que este entra a una página web, debe ser capaz de navegar por ella, por lo que el usuario verá una interface sencilla de usar, atractiva y funcional.

Un desarrollador front-end debe conocer los siguientes lenguajes de programación: HTML5, CSS3, JavaScript, JQuery, Ajax.



BACK-END

Como hemos dicho antes, front-end es todo con lo que el usuario se encuentra directamente en la web o aplicación, entonces cuando hablamos de "back-end" nos referimos al interior de las aplicaciones que viven en el servidor y al que a menudo se le denomina "el lado del servidor".

El back-end del sitio web consiste en un servidor, una aplicación y una base de datos. Se toman los datos, se procesa la información y se envía al usuario. Los desarrolladores de front end y back-end suelen trabajar juntos para que todo funcione correctamente.

Un desarrollador back-end debe tener amplios conocimientos de los siguientes lenguajes: frameworks y los tipos de base de datos. No siendo necesario conocer todos los lenguajes pero sí entender y saber trabajar con algunos de ellos.

ASP.NET , PHP, Python, Ruby, Node.js, Java, MySQL, SQL Server, PostgreSQL, Oracle, MongoDB



Full Stack

Por otro lado, un desarrollador Full Stack es el encargado de manejar cada uno de los aspectos relacionados con la creación y el mantenimiento de una aplicación web. Para ello es fundamental que el desarrollador Full Stack tenga conocimientos en desarrollo Front-End y Back-End además de manejar diferentes sistemas operativos y lenguajes de programación.

Fuente: <https://descubrecomunicacion.com/que-es-backend-y-frontend/>

SITIOS WEB ESTÁTICOS Y DINÁMICOS

Página web estática

Cuando hablamos de “estático” también podemos referirnos a que la página web tiene un número fijo de página, es decir, que tal como fue diseñada y almacenada en el servidor web, así mismo la recibe el navegador y la ve el usuario, como un número fijo de páginas HTML.

Una página web estática está compuesta por archivos HTML individuales por cada página que son pre-generados y presentados al usuario a través del navegador de la misma forma.

Como una página web estática básica está compuesta por elementos como títulos, cuadros de textos, etiquetas, imágenes y otros elementos multimedia, un usuario solo puede interactuar con una página web estática a través de lo que permiten los elementos HTML, por ejemplo haciendo clic en enlaces, botones o rellenando formularios como el clásico formulario de suscripción.

No son tan complejos técnicamente como un sitio web dinámico, pero tampoco son tan versátiles y efectivos cuando se trata de entregar funcionalidad. En pocas palabras, en una página web estática, verás la misma información, diseño y contenido cada vez que la visites, a menos que alguien aplique cambios al código fuente de forma manual.

Si quisieras crear una página web estática solo necesitas un editor de texto como el Bloc de notas y saber de HTML y CSS, no es necesario utilizar entornos de desarrollo complejos.

Ventajas de una página web estática

- El coste inicial de una página web estática puede ser mucho menor que al de una dinámica.
Por su naturaleza estática, la complejidad y tiempo de desarrollo es menor porque no requiere del uso de lenguajes de programación o bases de datos, y por ende su coste monetario es más bajo.
- Son muy flexibles cuando se trata del diseño.
Dado a su naturaleza independiente, cada página puede tener un diseño diferente. No es necesario un solo diseño para múltiples tipos de contenido, lo que en los sitios web dinámicos se le conoce como plantillas (templates).
- Los tiempos de carga son muy rápidos.
Ya que los sitios web estáticos son construidos previamente. No implica ejecución de scripts o secuencias de comandos complejas, bases de datos ni análisis de contenido a través de lenguajes de plantillas, etc.
Sin embargo, con la revolución del Jamstack, los generadores de sitios web estáticos como Jekyll, GatsbyJS o Eleventy, y los Headless CMS como Netlify CMS, Siteleaf o Forestry, y además la incorporación de CDN (Content Delivery Network en inglés) para gestionar los recursos multimedia, se puede generar un aumento en el coste de carga de

una página web estática dependiendo de sus características.

Desventajas de una página web estática

- Una página web estática puede ser más difícil de actualizar.
Para usuarios no técnicos, una vez la página es creada, hacer pequeños ajustes en el contenido puede representar un desafío a menos que estén familiarizados con HTML, CSS y el código del sitio web en general. Si no es así, es posible que deban pedirle al desarrollador que la creó originalmente, que realice los cambios que necesitan.
- Agregar contenido a la página web o realizar actualizaciones puede incurrir en costos adicionales.
Esto puede verse como una consecuencia de la desventaja anterior. Es decir que, con el tiempo, el mantenimiento de un sitio estático puede generar costos de mantenimiento continuo que podrían evitarse si tuvieras una página web dinámica.
- Agregar nuevas páginas o funcionalidades a una web estática puede ser más difícil que hacerlo para una web dinámica.
Por ejemplo, si creas una página web para promocionar productos de tecnología, cada vez que quieras agregar un producto, como un nuevo televisor o un nuevo portátil, tendrías que crear una nueva página específicamente para ese producto, lo que puede llevar mucho tiempo además del costo que puede llevar este proceso.

Ejemplos de páginas web estáticas

Un ejemplo sencillo de cómo es una página web estática, es el siguiente:

```
<head>
<title>Ejemplo página web estática</title>
</head>
<body>
La fecha de hoy es Enero 1, 2020
</body>
</html>
```

Página web dinámica

La palabra dinámica se refiere a elementos que cambian continuamente, son interactivos y funcionales, en lugar de ser simplemente informativos. Por supuesto, eso requiere utilizar más que solo código HTML y CSS.

En comparación con las páginas web estáticas, que son mayoritariamente informativas, una página web dinámica incluye aspectos que se caracterizan por la interactividad y la funcionalidad, por ejemplo, los usuarios pueden interactuar con la información que se presenta en la página gracias a las instrucciones creadas a través de los lenguajes de programación y la base de datos sobre la que está construida.

Los sitios web dinámicos basan su comportamiento y funcionalidad en dos tipos de programación, front-end (del lado del cliente) y back-end (del lado del servidor). Las instrucciones del lado del cliente es código JavaScript que se ejecuta en el navegador. Mientras que las instrucciones que se ejecutan del lado del servidor son instrucciones escritas en lenguajes de scripting o programación, como ASP.Net, PHP, Python, etc. y que son ejecutadas para crear lo que el usuario ha solicitado en su interacción con la página.

Una vez ejecutadas las instrucciones en el servidor, un nuevo HTTP response se envía al navegador del usuario para mostrarle lo que ha solicitado. El resultado final es el mismo que en un sitio web estático: una página HTML que el usuario ve desde el navegador.

Por resumir, una página web dinámica puede ser más compleja cuando hablamos de su diseño y desarrollo, pero también es más versátil cuando se trata de la funcionalidad que ofrece.

Ventajas de una página web dinámica

- Puede gestionar información a través de bases de datos.
Esto permite que el usuario pueda solicitar información fácilmente de una manera organizada y estructurada dentro de un catálogo, además de crear y mostrar contenido según el tipo de usuario que acceda a la página.
- El contenido se puede gestionar a través de un CMS.
El contenido almacenado en el CMS puede incluir una variedad de archivos, desde el texto hasta las imágenes que se muestran, diseños de página, configuraciones del sitio y más. Esto permite una flexibilidad extrema a la hora de crear el sitio y también permite que varios usuarios puedan manipular el contenido según sea necesario.
- El coste de mantenimiento es menor.
Si la página no necesita cambios en el diseño básico o en la funcionalidad definida al inicio de su desarrollo. Ya que se puede gestionar la información a través de un CMS, existe poco o nada de costes cuando se trata de su mantenimiento.

Desventajas de una página web dinámica

- Pueden existir limitaciones en el diseño.
Ya que el contenido está principalmente basado en la información contenida en la base de datos y la presentación al usuario se basa en la estructura de la misma. Esto puede hacer que el diseño sea complicado, ya que lo más sencillo es optar por un enfoque único para todas las páginas. Dependiendo del CMS, puede resultar difícil crear varios diseños o plantillas que permitan mostrar diferentes tipos de contenido de diferentes formas. ■ Puede involucrar altos costos de construcción iniciales.
Al coste del desarrollo de la página web se le suma el coste del desarrollo de las bases de datos donde se guardará el contenido a mostrar, etc. El desarrollo también puede costar más a medida que se agregan nuevas funcionalidades. Si bien los costos de mantenimiento pueden ser más bajos como fue mencionado en las ventajas, también puede involucrar costos de desarrollo iniciales mucho más altos que al desarrollar una

página web estática.

Ejemplos de páginas web dinámicas

Como ya hemos visto, es muy sencillo determinar si una página web es dinámica: por ejemplo, cuando puedes interactuar con ella, o si cada vez que la recargas, puedes ver contenido distinto.

Por lo tanto, la mayoría de las páginas que regularmente visitas es probable que sean dinámicas porque son interactivas. Por ejemplo, una página web dinámica te permite crear un perfil de usuario Facebook.com, comentar una publicación LinkedIn.com, pedir tu cena just-eat.es o hacer una reserva Booking.com.

Siguiendo el ejemplo de la página que muestra una fecha, si queremos convertirla en una página web dinámica, podemos cambiar la fecha escrita textualmente por una función que retorne la fecha actual, de esta forma:

```
<head>
<title>Página web dinámica</title>
</head>
<body>
La fecha de hoy es <%=Datetime.Now()%>
</body>
</html>
```

Aquí, cada vez que se recarga la página, se mostrará la fecha y hora actual, es decir será diferente en cada recarga de la página, ya que la instrucción `<%=Datetime.Now()%>` le indica al servidor que retorne la fecha del momento en que recibe la petición.

Fuente: <https://openwebinars.net/blog/paginas-web-estaticas-vs-paginas-web-dinamicas/>

APLICACIONES WEB

Las aplicaciones web son una herramienta que se codifica en un lenguaje que es soportado por los navegadores y que se ejecuta en los mismos. Es decir, son una clase de software que puedes usar accediendo a cierto servidor web a través de Internet o de una Intranet por medio de un navegador que ejecutará la aplicación. Pero, si quieres saber yos, quédate con nosotros. En Einatec te mostramos cómo funcionan las aplicaciones web en detalle.

Funcionamiento de las aplicaciones web

Lo primero que debes tener en cuenta es que gracias a que las aplicaciones web se ejecutan en un servidor web todo lo que haces en ellas se procesa y almacena en el interior de una base de datos por medio de un navegador. De esta forma, no es necesario que las instales en tu ordenador o dispositivo móvil.

Se puede utilizar una web para que el usuario pueda acceder a la información que contiene de manera interactiva. De hecho, las aplicaciones web están especialmente diseñadas para almacenar datos en la nube. Esta información se mantiene almacenada en los servidores web y cuando necesitas utilizarla la aplicación te envía esos datos a tu ordenador o a tus dispositivos móviles. Para tal fin, hace copias temporales en el ordenador o dispositivo que estás usando.

Las aplicaciones web están compuestas por tres partes:

- Una base de datos: en la base de datos se almacenan precisamente eso, todos los datos, permisos, usuarios, contenidos, información de interés...
- El código de la aplicación: la propia aplicación se aloja en un servidor en la nube de aplicaciones, en algunos casos puede almacenarse en un servidor local.
- El acceso a través del navegador: puedes acceder utilizando un ordenador o un dispositivo móvil por medio de un navegador. Se incluye el administrador o el gestor que es el usuario final. Podrá acceder a distintas secciones dependiendo de los permisos con los que cuente.

Para desarrollar las aplicaciones web se suele utilizar alguna de las siguientes tecnologías:

- Para la base de datos se emplea SQL Server o MySQL.
- Para la aplicación se usa ASP.NET o PHP.
- Tu navegador recibe la información en formato HTML5.

¿Para qué se utilizan las aplicaciones web?

- Gestión interna: se emplean para controlar el stock, los clientes, los usuarios; para gestionar el personal, las facturas o la contabilidad en general; o para que fichen los empleados.
- Servicios a los usuarios: gestionan todo lo que tiene que ver con los servicios que se le prestan al usuario, como los permisos para acceder a diferentes contenidos, la gestión de incidencias o de espacios, por ejemplo.
- Herramientas de trabajo: herramientas para gestión documental, accesos diferenciados, Intranets...
- Herramientas para el control de calidad: herramientas que facilitan el control de calidad.
- Herramientas para la comunicación: comunicaciones con los usuarios personalizadas,

mailings o boletines digitales.

- Herramientas web: gestión de ventas online, buscadores, tiendas virtuales, webs...

Fuente: <https://einatec.com/como-funcionan-las-aplicaciones-web/>

LENGUAJES DE LA WEB

En la actualidad, existe una gran cantidad de lenguajes de programación enfocados al desarrollo web. En sus orígenes, estos lenguajes tenían un formato estático. Sin embargo, con el paso de los años y la evolución a la hora de crear páginas web han evolucionado en lenguajes dinámicos. Estos nuevos avances permiten al usuario interaccionar más con la página y mejorar la experiencia en la navegación, con lo que ya no es un mero tablón que muestra información.

Como hemos dicho, existen muchos lenguajes. Para conocer un poco mejor qué tipos de lenguaje de programación web existen, es necesario hacer un recopilatorio de los más utilizados y explicar sus ventajas e inconvenientes, en comparación a los demás.

■ Lenguaje JavaScript

Se utiliza principalmente del lado del cliente aunque se puede utilizar del lado del servidor. Actualmente y gracias a tecnologías como AJAX es utilizado para enviar y recibir información del servidor.

Como principales ventajas, tenemos que destacar que es un lenguaje de scripting seguro y fiable, cuyos scripts tienen capacidades limitadas, debido a la seguridad.

Como desventajas, podríamos mencionar que el código debe descargarse por completo y es visible por cualquier usuario.

■ Lenguaje PHP

Es un lenguaje enfocado en la creación de webs dinámicas. Sus scripts son interpretados por el servidor y genera código HTML. Requiere Apache o IIS con librerías de PHP. Hereda su sintaxis de C, Java y Perl.

Como principales ventajas, hemos de decir que es un lenguaje fácil de aprender y muy rápido. Soporta la orientación a objetos y utiliza un lenguaje multiplataforma. Además, puede conectarse con una gran cantidad de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server.. No necesita que se definan los tipos de variables. Uno de sus aspectos más llamativos es que está diseñado con el fin de ser un lenguaje muy seguro para escribir CGI, más que Perl o C.

Es el lenguaje base que utilizan la mayoría de CMS o gestores de contenidos más extendidos como WordPress, PrestaShop, Drupal o Joomla.

■ Lenguaje Python

Considerado por muchos el lenguaje más limpio a la hora de programar. El código, al igual que JavaScript, es interpretado y no compilado.

Algo curioso en este lenguaje es que permite a los programadores elegir un estilo de programación concreto (objetos, estructurado, funcional...), debido a que es un lenguaje de programación multiplataforma.

Como ventajas de Python, destacamos que es libre y de fuente abierta, de propósito general. Cuenta con muchas funciones y librerías y es multiplataforma y fácil de programar. Por otro lado, su principal desventaja es que, al ser un lenguaje interpretado, es bastante lento.

📖 Lenguaje Ruby

Como el anterior, es un lenguaje interpretado y está orientado a objetos. Hereda su sintaxis de Python y Perl. El lenguaje puede cargar librerías de extensiones dinámicamente si el sistema operativo lo permite. Además, es un lenguaje portátil.

Otra ventaja que ofrece es que cualquiera puede encontrar en su [página web](#) gran cantidad de información y tutoriales.

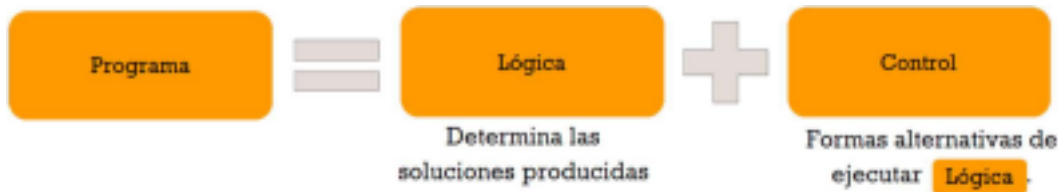
El desarrollo web es una tarea que requiere mucho tiempo, por lo que elegir un lenguaje para tu [hosting](#) con el que sentirse cómodo es crucial.

Fuente: <https://www.piensasolutions.com/blog/principales-lenguajes-programacion-web/>

PROGRAMACIÓN LÓGICA

Paradigma de programación basado en la lógica de primer orden. La programación lógica estudia el uso de la lógica para el planteamiento de problemas y el control sobre las reglas de inferencia para alcanzar la solución automática.

La programación lógica, junto con la funcional, forma parte de lo que se conoce como Programación Declarativa, es decir la programación consiste en indicar como resolver un problema mediante sentencias, en la Programación Lógica, se trabaja en una forma descriptiva, estableciendo relaciones entre entidades, indicando no como, sino que hacer, entonces se dice que la idea esencial de la programación lógica es



Se puede ver como

una deducción controlada.

Lógica (programador): hechos y reglas para representar conocimiento.

Control (interprete): deducción lógica para dar respuestas (soluciones).

¿Qué trata de resolver?

Dado un problema S , saber si la afirmación A es solución o no del problema o en cuáles casos lo es. Además, queremos que los métodos sean implantados en máquinas de forma que la resolución del problema se haga de forma automática

La programación lógica construye base de conocimientos mediante reglas y hechos.

Características del Paradigma

- Unificación de términos.
- Mecanismos de inferencia automática.
- Recursión como estructura de control básica.
- Visión lógica de la computación.

Fuente: https://ferestrepoca.github.io/paradigmas-de-programacion/proglogica/logica_teoría/proglogica.html

MAQUETADO Y ESTILADO

La maquetación, maquetar, puede entenderse como el proceso de tomar un conjunto aleatorio de textos, fotos, títulos, ilustraciones, pies de foto, nombres de secciones, etc. y construir con ellos un diseño global estructurado, de modo que se pueda leer de modo secuencial (libro) o aleatorio (revista), manteniendo una coherencia y siguiendo un orden lógico, y utilizando recursos gráficos para distribuir los diferentes elementos en función de su importancia.

Normalmente, aunque no siempre, el proceso también supone dar al conjunto una estructura que pueda ser publicable en un volumen físico, impreso y encuadernado, por lo que la maquetación tiene una relación muy grande con las artes gráficas. Además, en los últimos años también se relaciona, y cada vez más, con los soportes de páginas web y libros electrónicos.

A diferencia de la maquetación editorial, que termina cuando el diseño está estructurado para su impresión, la maquetación web implica separar los elementos de la imagen y, dependiendo de la función que deba desempeñar cada elemento, desarrollar y estructurar el código que le permita a cada uno tener su orden y su aplicación.

Los elementos visuales que percibe el usuario al entrar a una página web, se conoce como la "interfaz de usuario", y se puede diseñar por un profesional en programas de edición de imagen como Photoshop o Illustrator. Cuando se separa la imagen final del diseño por elementos, se otorga a cada uno su función y se le da un orden por medio de un lenguaje de programación, el resultado será una página maquetada para web.

Para lograr que una página web maquetada sea óptima, se debe contar con un diseño separado por capas para que cada elemento pueda realizar su función. Estos elementos se colocarán en la web por medio de HTML, mientras que la parte que reúne dichos elementos para dar el orden y la función correspondientes al diseño se lleva a cabo por medio de CSS (hojas de estilo en cascada por sus siglas en inglés).

La maquetación web combina el diseño con la programación de una página, ya que lo que se busca es brindarle al usuario una experiencia agradable a la vista, contenido comprensible y buena usabilidad del sitio, según sus necesidades.

Al combinar la estructura de un código (HTML) y sus estilos aplicados (CSS), se hace la función de maquetar. Los elementos del sitio se definen por medio de contenedores (divs) que se posicionan a través de hojas de estilo. Se suelen emplear diseños específicos a las necesidades del sitio, así como a su tipo de usuario, distribuyendo los distintos elementos visuales o interactivos en un espacio determinado dentro de la página.

Anteriormente la maquetación se llevaba a cabo por medio de tablas, resultando mucho menos dinámico y limitando las opciones tanto de diseño como de desarrollo. Ahora la manera más funcional de crear una web visualmente atractiva es por medio de hojas de estilo. Es además la manera más acertada de llevar la idea específica del cliente sobre el diseño visual de una página web a la realidad, ya que las posibilidades son ilimitadas.

En Sector Web pasamos cualquier diseño desarrollado en Photoshop o Illustrator a código, construyendo la estructura en HTML5, asignando estilos con CSS3 y programando el contenido multimedia con jQuery, asegurándonos de que el diseño sea responsivo para que funcione en todos los dispositivos.

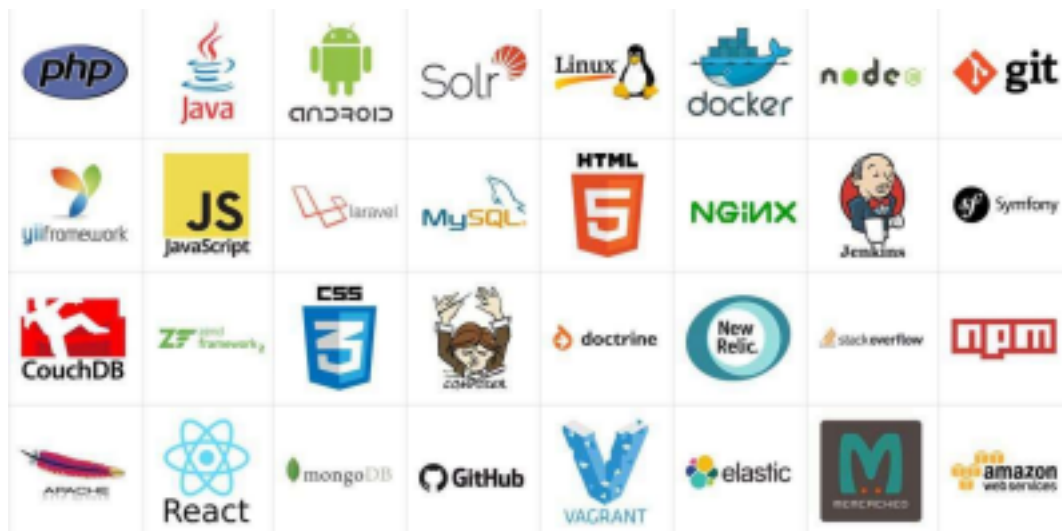
Fuentes:

<https://www.calamoycran.com/blog/maquetacion-de-que-estamos-hablando-i>

<https://sectorweb.mx/blog/que-es-la-maquetacion-web>

HERRAMIENTAS DE DESARROLLO

Una herramienta de programación o una herramienta de desarrollo de software es un programa informático que los desarrolladores de software utilizan para crear, depurar, mantener o dar soporte a otros programas y aplicaciones. El término generalmente se refiere a programas relativamente simples, que se pueden combinar para realizar una tarea, de la misma manera que se pueden usar varias manos para arreglar un objeto físico. Las herramientas más básicas son un editor de código fuente y un compilador o intérprete, que se utilizan de forma ubicua y continua. Otras herramientas se utilizan más o menos según el lenguaje, la metodología de desarrollo y el ingeniero individual, que a menudo se utilizan para una tarea discreta, como un depurador o un generador de perfiles. Las herramientas pueden ser programas discretos, ejecutados por separado, a menudo desde la línea de comandos, o pueden ser partes de un único programa grande, llamado entorno de desarrollo integrado (IDE). En muchos casos, particularmente para un uso más simple, se utilizan técnicas ad hoc simples en lugar de una herramienta, como depuración de impresión en lugar de usar un depurador, cronometraje manual (del programa general o sección de código) en lugar de un generador de perfiles, o rastrear errores en un archivo de texto o una hoja de cálculo en lugar de un sistema de seguimiento de errores. La distinción entre herramientas y aplicaciones es confusa. Por ejemplo, los desarrolladores usan bases de datos simples (como un archivo que contiene una lista de valores importantes) todo el tiempo como herramientas. [Dudoso - discutir] Sin embargo, una base de datos completa generalmente se considera una aplicación o software por derecho propio. Durante muchos años, se buscaron herramientas de ingeniería de software asistida por computadora (CASE). Las herramientas exitosas han resultado difíciles de alcanzar. En cierto sentido, las herramientas CASE enfatizaron el soporte de diseño y arquitectura, como para UML. Pero las más exitosas de estas herramientas son los IDE.



Usos de herramientas de programación

📖 Traducir del lenguaje humano al informático

Las computadoras modernas son muy complejas y para programarlas productivamente, se necesitan varias abstracciones. Por ejemplo, en lugar de escribir la representación binaria de un programa, un programador escribirá un programa en un lenguaje de programación como C, Java o Python. Las herramientas de programación como los ensambladores, compiladores y enlazadores traducen un programa de un lenguaje fuente legible y con capacidad de escritura humana a bits y bytes que pueden ser ejecutados por una computadora. Los intérpretes interpretan el programa sobre la marcha para producir el comportamiento deseado. Estos programas realizan muchas tareas bien definidas y repetitivas que, sin embargo, llevarían mucho tiempo y serían propensas a errores cuando las realiza un humano, como colocar partes de un programa en la memoria y arreglar las referencias entre partes de un programa como lo hace un enlazador. La optimización de los compiladores, por otro lado, puede realizar transformaciones complejas en el código fuente para mejorar la velocidad de ejecución u otras características de un programa. Esto permite que un programador se concentre más en los aspectos conceptuales de un programa de nivel superior sin preocuparse por los detalles de la máquina en la que se está ejecutando.

📖 Hacer que la información del programa esté disponible para los humanos

Debido a la alta complejidad del software, no es posible comprender la mayoría de los programas de un solo vistazo, incluso para el desarrollador de software más experimentado. Las abstracciones proporcionadas por los lenguajes de programación de alto nivel también dificultan la comprensión de la conexión entre el código fuente escrito por un programador y el comportamiento real del programa. Para encontrar errores en los programas y evitar la creación de nuevos errores al extender un programa, un desarrollador de software utiliza algunas herramientas de programación para visualizar todo tipo de información sobre los programas. Por ejemplo, un depurador permite a un programador extraer información sobre un programa en ejecución en términos del lenguaje fuente utilizado para programarlo. El depurador puede calcular el valor de una variable en el programa fuente a partir del estado de la máquina concreta utilizando información almacenada por el compilador. Los depuradores de memoria pueden señalar directamente accesos a la memoria cuestionables o completamente incorrectos de programas en ejecución que, de lo contrario, podrían pasar desapercibidos y son una fuente común de fallas de programas.

Fuente: https://en.wikipedia.org/wiki/Programming_tool