**FRONTEND**

El desarrollo web front-end es la práctica de convertir datos a una interfaz gráfica, mediante el uso de HTML, CSS y JavaScript, para que los usuarios puedan ver e interactuar con esos datos.

**Herramientas utilizadas para el desarrollo front-end**

Hay varias herramientas y plataformas (WordPress, Magento, etc.) disponibles que se pueden usar para desarrollar el front-end de un sitio web, y comprender qué herramientas son las más adecuadas para tareas específicas marca la diferencia entre desarrollar un sitio pirateado y un sitio bien diseñado, sitio escalable.

**Lenguaje de marcado de hipertexto (HTML)**

Hyper Text Markup Language (HTML) es la columna vertebral de cualquier proceso de desarrollo de sitios web, sin el cual no existe una página web. El hipertexto significa que el texto tiene enlaces, denominados hipervínculos, incrustados en él. Cuando un usuario hace clic en una palabra o frase que tiene un hipervínculo, traerá otra página web. Un lenguaje de marcado indica que el texto puede convertirse en imágenes, tablas, enlaces y otras representaciones. Es el código HTML que proporciona un marco general de cómo se verá el sitio. HTML fue desarrollado por Tim Berners-Lee.

**Hojas de estilo en cascada (CSS)**

Las hojas de estilo en cascada (CSS) controlan el aspecto de presentación del sitio y le permiten a su sitio tener un aspecto único. Lo hace manteniendo hojas de estilo que se ubican por encima de otras reglas de estilo y se activan en función de otras entradas, como el tamaño y la resolución de la pantalla del dispositivo.

**JavaScript**

JavaScript es un lenguaje de programación imperativo basado en eventos (a diferencia del modelo de lenguaje declarativo de HTML) que se utiliza para transformar una página HTML estática en una interfaz dinámica. El código JavaScript puede usar el [Modelo de objeto de documento] (DOM), proporcionado por el estándar HTML, para manipular una página web en respuesta a eventos, como la entrada del usuario.

**WebAssembly**

WebAssembly, compatible con todos los principales navegadores (es decir, de los principales proveedores Google, Apple, Mozilla y Microsoft), es la única alternativa a JavaScript para ejecutar código en navegadores web (sin la ayuda de complementos, como Flash, Java o Silverlight; todo está descontinuado, ya que los navegadores dejan de admitir complementos). Antes de su adopción, había asm.js (un subconjunto de JavaScript; y, por lo tanto, funciona estrictamente en todos los navegadores), que también se utiliza como un compilador de destino con soporte eficiente en navegadores como Internet Explorer 11; y para los navegadores que no admiten WebAssembly directamente, se puede compilar en asm.js y esos navegadores se admiten de esa manera. En general, los programadores no programan directamente en WebAssembly (o asm.js), sino que usan lenguajes como Rust, C o C ++ o, en teoría, cualquier lenguaje que lo compile.

**Objetivos para el desarrollo**

El desarrollador del front-end mantiene estos puntos en mente, utilizando las herramientas y técnicas disponibles para llegar a este fin.

**Accesibilidad**

Con el desarrollo continuo de dispositivos móviles, como teléfonos inteligentes y tabletas, los diseñadores deben asegurarse de que su sitio aparezca correctamente en los navegadores de todos los dispositivos. Esto se puede hacer creando un diseño web receptivo usando hojas de estilo en CSS.

**Rendimiento**

Los objetivos de rendimiento tienen que ver principalmente con el tiempo de renderizado, manipulando el HTML, CSS y JavaScript para garantizar que el sitio se abra rápidamente.

**BACK END**

El Back end es la parte trasera de cualquier página web. Se trata de todo el conjunto del desarrollo que se encarga de que una página funcione y de que lo haga como lo hace, pero que al mismo tiempo es totalmente invisible para el usuario, que solo ve lo visual y gráfico.

Su contrapartida es el Front end, que es la mitad con la que interactúa cada visitante en una web. Ambas, juntas dan forma a todo el conjunto que construye una página web en la actualidad y lo que permite que, además de la usabilidad, también pueda hablarse de rendimiento y de comunicación en su interior.

Podemos encontrar muchos tipos de Back ends diferentes en la actualidad. Por ejemplo, cualquier página hecha con WordPress, que además lo utilice como CMS, tiene en él a gran parte de su backend, ya que es lo que se encarga del lado administrativo, de permitir que se elaboren contenidos y, al mismo tiempo, que se gestionen usuarios y se lleven a cabo otras tareas de carácter interno.

Aquí encontramos lenguajes de desarrollo bastante diferentes a los anteriores, como pueden ser PHP, Java o .Net. En el Back end se llevan a cabo tareas como la lógica de la web, las conexiones con bases de datos en el servidor, el uso de librerías o incluso las operaciones llevadas a cabo por las distintas funciones disponibles.

**Para qué sirve el Back end**

Sirve para que el portal funcione como lo hace, que permita desde publicar contenidos hasta habilitar un botón para que el usuario se registre. En función de cómo esté construido, puede hacer que una web sea más rápida o no, que sea más completa o menos, que el usuario quede más o menos satisfecho. El Back end es una parte fundamental de cualquier web, aunque debe estar siempre junto con el Front end.

Debido a esto, es algo que influye incluso a ojos de Google y el posicionamiento web. Tiempos de carga elevados no son buenos para el buscador, y mucho menos la presencia de código inútil o que realiza operaciones fallidas. Cuidar esta parte trasera de una web es crucial para el crecimiento de una empresa en internet.

**Herramientas**

Los lenguajes de programación que se utilizan en el Back-end en la actualidad son PHP, JavaScript, Python y Ruby. Además de HTML y CSS, que, aunque básico, también deben saber al menos un poco. Aquí me gustaría hablar de 2 cosas, primero que nada, yo he mencionado 4 lenguajes de programación, pero no son todos los que existen y no es obligatorio utilizar todos y cada uno de ellos, hay programadores back-end que solo saben algunos, no es necesario saber todos, depende de lo que se quiera programar y de las capacidades del lenguaje. Como segunda cosa que me gustaría comentar es que, aunque el Back-End también debe saber de HTML y CSS, no es necesario saber a un nivel alto como lo haría un Front-End, es simplemente para poder crear una estructura básica en la que trabajar.

Las herramientas que se utilizan en el Back-end son editores de código, compiladores, algunos debuggeadores para revisar errores y seguridad, gestores de bases de datos y algunas otras cosas.

**ADMINISTRADOR DE SERVIDORES**

Una carrera como administrador de servidores puede ser desafiante y gratificante. Un administrador de servidores es responsable de la operación y mantenimiento de un servidor o de una red de la oficina. En la mayoría de las empresas, un administrador de servidores trabaja en el Departamento (Dirección General) de Tecnología de la Información (TI) y tiene muchas funciones y responsabilidades.

**Requisitos generales**

Un administrador de servidores tiene muchas funciones de trabajo que varían de una organización a otra. Generalmente, sin embargo, los administradores de servidores son responsables de la Planeación, Instalación, Soporte y Mantenimiento de un Sistema o Servidor informático. Los servidores centralizados son fuentes de datos para una empresa y asegurarse de que funcionen sin problemas es fundamental. Evitar el tiempo de inactividad del servidor a través de un mantenimiento programado, garantizando la seguridad del servidor y ayudar al personal en la conexión con el servidor son todas las tareas importantes de un administrador de servidores. Hacer una copia de seguridad de los datos del servidor es también un deber del personal de administración de servicios.

**Gestión de proyectos y supervisión**

Otras funciones laborales pueden incluir la gestión de proyectos con sistemas relacionados, la supervisión y la capacitación del personal que trabaja con computadoras, la reparación del mal funcionamiento del servidor y las consultas sobre problemas demasiado complejos para el soporte técnico. Esto puede implicar la interfaz con los usuarios novatos del equipo y con el soporte técnico avanzado. Por lo tanto, aquellas personas en la administración de servidores deben ser capaces de explicar el uso de la computadora y las funciones en un nivel en que un laico pueda entender. También debe hacer frente a los aspectos más técnicos de la administración de servidores.

**Funciones de gestión**

Los administradores de servidores a menudo deben gestionar el sistema operativo del servidor, mantener la integridad del rendimiento del servidor, instalar y configurar el software nuevo y las actualizaciones, solucionar problemas y actualizar la información de cuentas de usuario (añadir/eliminar usuarios y restablecer contraseñas). También deben realizar copias de seguridad de rutina; integrar las nuevas tecnologías; gestionar redes; y aplicar las actualizaciones, revisiones y cambios de configuración para el sistema operativo.

En otras palabras, un Administrador de Servidores es la persona responsable de establecer y mantener la Plataforma de datos y comunicaciones de una organización / institución. También llamados o conocidos como Administradores de Sistemas o SysAdmin, y muchas veces abarca a los Administradores de Redes y Especialistas de Soporte avanzados (Nivel II o III) duchos en múltiples Sistemas Operativos privativos y libres.

**Deberes de un administrador del sistema**

Los Administradores de Servidores (Sysadmins) suelen ser los encargados de la instalación, soporte y el mantenimiento de los servidores u otros sistemas informáticos, así como la planificación y respuesta a interrupciones del servicio y otros problemas. Otras tareas pueden incluir la programación de secuencias de comandos o programación ligera, para los sistemas de gestión relacionados con los proyectos.

**El administrador del sistema es responsable de lo siguiente:**

Poner en marcha nuevos servidores: con la compra de un nuevo equipo habrá que instalar y configurar todo el sistema operativo y demás paquetes de software que sean necesarios según la funcionalidad que vaya a tener ese equipo, configurarlo en la red y resto de tareas que veremos también a través de los siguientes puntos. No es una tarea cotidiana, pero si es normal mantener un entorno de prueba y analizar su funcionamiento antes de integrarlo en producción.

* **Realizar backups:** empezamos por lo que se suele dejar para lo último. Es una de las tareas primordiales y quizás menos querida por parte de los administradores de sistemas, por lo que tiene de monótona y repetitiva, aunque pueda automatizarse en gran medida. Aquí la labor del administrador es fundamental, primero diseñando una política de backups con frecuencias y niveles (totales, parciales, incrementales, por sistemas, por bbdd, ficheros usuarios…) que salvaguarde adecuadamente la información ante posibles fallos. Además, programar una ejecución automatizada, supervisión de los medios sobre los que se vuelca la información, asegurar que haya copias a buen recaudo en otras ubicaciones, etc… Dependiendo la complejidad del sistema que administra, como ya dijimos, puede ser imprescindible tener una política de recovery bien diseñada ante fallos graves.
* **Cambiar la configuración hardware de los equipos:** si añadimos un nuevo sistema de almacenamiento, una tarjeta de red o una impresora compartida, este nuevo hardware debe ser reconocido por el sistema con una adecuada configuración, hallar posibles incompatibilidades con determinadas versiones de software que tenga que ser actualizado o la necesidad de buscar controladores específicos para ese dispositivo. Con la virtualización, las ‘reglas de juego’ se pueden complicar para que ese hardware se instale y comparta correctamente y con seguridad.
* **Instalar nuevo software y actualizar el existente:** otra tarea que requiere dedicación constante, sobre todo en lo relativo a mantener las versiones adecuadas de software y aplicar los distintos ‘parches’ de seguridad que protejan a los sistemas de vulnerabilidades que se vayan identificando. Para software nuevo, lo normal es probarlos en esos entornos de prueba que comentábamos en el primer punto antes de que entren en producción.
* **Gestión de cuentas de usuarios:** esto incluye por supuesto altas, bajas o modificaciones de usuarios existentes o la configuración de privilegios de acceso. Todas estas acciones llevan aparejadas tareas a realizar, muchas de las cuales acaban siendo automatizadas y normalmente una planificación previa del procedimiento. Por ejemplo, en un alta, en qué máquina estará o cuál será el directorio home de ese usuario; en una baja, aparte de deshabilitarlo para impedir el acceso posiblemente deba hacer backup de todos sus ficheros y pasarlos a algún formato de almacenaje y liberar espacio. En este capítulo también podríamos incluir la virtualización de puestos de trabajo, cada vez más en boga, y que lleva más allá el proceso de configuración para crear el entorno específico a cada usuario, aunque posteriormente facilite la gestión.
* **Monitorizar el rendimiento del sistema:** un buen administrador es proactivo y se adelanta a los fallos, o lo que es lo mismo, prevenir es mejor que curar. La degradación del rendimiento de los sistemas o la aparición de problemas incipientes conviene que sean detectados cuanto antes y no por el reporte de los usuarios., que pueden esperar hasta que sus síntomas sean evidentes y ya sea demasiado tarde. La vigilancia de los distintos sistemas de logs y alertas del sistema, su correcta configuración y automatización mediante scripts o herramientas software específicas que incluso avisen por email o sms, es una tarea preventiva que proporciona buenos dividendos.
* **Seguridad:** otro aspecto crítico y con similitudes al anterior en cuanto a vigilancia continuada. No hay ningún sistema infalible ante un acceso no autorizado o un ataque, pero está en su mano tener las políticas adecuadas en marcha incluyendo la concienciación a los usuarios, las configuraciones en regla, así como las versiones y parches de software actualizados, la instalación de sistema de protección y detección ‘proporcionados’ al rango del sistema.
* **Fallos y caídas del sistema:** si no supimos detectarlo, o si el fallo se produce súbitamente, es necesaria una intervención rápida para restaurar el sistema a su funcionamiento normal. En fallos graves se suele dar la máxima de que es más difícil diagnosticar el problema que solucionarlo, y es labor del administrador determinar qué camino seguir, incluyendo la llamada a un mantenimiento externo contratado. Los administradores veteranos suelen decir que un Administrador que se aprecie no es tal si no ha pasado un fin de semana entero sin dormir en la Sala de Servidores (Centro de Datos / DataCenter), intentando levantar un sistema que no funciona.
* **Atención usuarios:** quizás la labor más discutida y la que ocupe la mayor parte de su tiempo. El Administrador no debería ser el help-desk, pero en organizaciones pequeñas puede ser casi el único recurso técnico al que pueden acudir los usuarios en busca de ayuda. Esta labor de soporte podría aligerarse en muchas ocasiones con una formación adecuada de los usuarios en los sistemas y programas que manejan en la Compañía, pero puede ser clamar en el desierto. Hay que guardar un difícil equilibrio entre estar continuamente resolviendo emergencias o atendiendo peticiones adhoc y el poner una barrera infranqueable haciendo caso omiso de lo que no viene por el canal reglamentario y está suficientemente documentado y justificado. Ser diplomático y empatizar con los usuarios no es algo que se enseñe habitualmente en los cursos de administración de sistemas, pero es condición imprescindible para llegar a ser un apreciado y valioso personal de TI.
* **Reporte a dirección/organización:** es algo que no se suele tener en cuenta, pero como decíamos en el anterior punto, el administrador ha de saber moverse entre sus colegas y ‘vender’ su trabajo. Como los porteros de fútbol, los fallos se notan más que en el resto porque una caída de un servidor crítico puede afectar a toda la organización o una incidencia resuelta tarde puede perjudicar mucho a un usuario. El administrador (si se han hecho razonablemente bien las cosas) debe transmitir la visión global y frente al 1% de fallos, resaltar el 99% de normalidad o ante un retraso considerable en resolver un problema puntual, mostrar el volumen total atendido y la media en la respuesta. En resumen, saber fijar y medir los indicadores adecuados de su desempeño y cómo transmitir esos resultados a la organización.
* **Documentación del sistema:** para finalizar, lo que siempre se deja para lo último y rara vez se hace. Uno no está solo y, sobre todo, debe poder irse de vacaciones. Si la configuración del sistema solo está en nuestra cabeza, dado que no se ha descubierto ningún Administrador que pueda separarse de ella y seguir llevando una vida normal, es conveniente que documentemos todas las particularidades de nuestro sistema, procedimientos, políticas o rutinas que se siguen, información (incluyendo passwords, ¡lógicamente manteniendo la confidencialidad!) y cualquier otro aspecto que ayude a un compañero o sustituto a llevar a cabo nuestras tareas. Hacerse el imprescindible porque nadie más sepa lo que hacemos puede ser una mala política.

**Un administrador también puede ser:**

* Un Administrador de Bases de Datos
* Un Administrador de Seguridad Informática
* Un Administrador de Redes
* Un Administrador de Sistemas Operativos (Privativos o Libres)

Los administradores de sistema no son Ingenieros de software o Desarrolladores. Por lo general no está dentro de las funciones de un Ingeniero de redes el diseño de nuevas aplicaciones de software. Sin embargo, debe comprender el comportamiento del software con el fin de implementar y solucionar problemas, y en general, debe ser bueno en varios lenguajes de programación utilizados para secuencias de comandos o la automatización de tareas rutinarias como Shell, Awk, Perl, Python, etc.

Resolver problemas, y punto. Cuando un servidor o estación de trabajo deja de funcionar, es a nosotros a quien llaman para resolver el problema. Un Administrador de Servidores debe poder diagnosticar rápida y correctamente el problema. Usted debe saber lo que está mal y cómo puede ser reparado en el menor tiempo posible.

**Conocimientos esenciales y habilidades de un Administrador de Servidores**

Muchas tareas para la Administración de los Servidores y Sistemas se pueden automatizar usando Perl o Scripts de Shell.

**Para realizar acciones tales como:**

* Crear nuevos usuarios
* La restauración de contraseñas de usuario
* Bloqueo / desbloqueo de cuentas de usuario
* Monitor de la seguridad del servidor
* Monitor de servicios especiales, etc

En realidad, un SysAdmin de alto nivel automatiza todo lo que puede.