

1. Diseño Web.

El proceso de diseño de una página web no es un proceso que se limite únicamente a escribir y programar código, sino que se trata de una tarea mucho más compleja para el diseñador, ya que debe tener en cuenta muchas otras variables para conseguir que el sitio web a desarrollar presente unos requisitos mínimos de usabilidad y atractivo gráfico.

Para diseñar contenidos web competentes y atractivos para el usuario, es importante conocer:

- Los principios básicos del diseño web profesional.
- Los tipos y características de las principales orientaciones de diseño (objetivos, usuario e implementación).
- Las etapas que se pueden seguir para diseñar adecuadamente un sitio web:
 - Delimitar el tema.
 - Recolectar información.
 - Estudiar la agregación y descripción de contenidos.
 - Diseñar la estructura y navegabilidad del sitio web.
 - Definir la estructura y composición de las páginas.
- Efectuar el ensamble final del sitio.
- Testear el sitio y verificar que es compatible con los navegadores que más se suelen utilizar.

2. Lenguajes de marcado generales.

Un lenguaje de marcas (o de marcado) consiste en un lenguaje que marca textos de manera que el ordenador sea capaz de manipularlos. La mayoría de los lenguajes de marcas pueden ser interpretados gracias a que dichas marcas se establecen con una sintaxis determinada que las diferencia claramente del texto en sí.

HTML es el lenguaje de marcas con el que se construyen las páginas web. No obstante, este no es el único lenguaje de este tipo que existe.

En el estudio de los lenguajes de marcado generales, se hace fundamental conocer:

- Origen y evolución de los lenguajes de marcas.
- Definición y características de algunos de los lenguajes de marcas más conocidos (SGML, XML, HTML, XHTML, etc.)
- Características generales de los lenguajes de marcado.
- Principales elementos que pueden aparecer en un documento escrito con lenguaje de marcas (instrucciones de proceso, etiquetas, atributos, caracteres especiales, etc.).
- Validación XML: documentos válidos y bien formados.

A continuación, se enumeran algunos consejos que es muy recomendable tener en cuenta cuando se aplican estilos CSS a los elementos de cualquier sitio web:

- **Separar el contenido del diseño:** como se ha visto anteriormente, el código CSS puede escribirse en un archivo externo, enlazado en la cabecera del documento. Esto permite separar completamente el contenido (HTML) de la presentación (CSS).
- **Aprovechar los prefijos de proveedor:** a veces, los navegadores ofrecen compatibilidad con estándares que todavía no se han aprobado. Para usarlos, existen unos prefijos que, añadidos a la propiedad que corresponda, permiten visualizarla de manera confiable en el navegador de que se trate.
- **Utilizar preprocesadores:** este tipo de aplicaciones permiten escribir código con mayor rapidez y posibilidades, adaptándolo posteriormente a un formato óptimo para todo tipo de navegadores. Algunos ejemplos de preprocesadores son: *Sass*, *Stylus* y *Less*.
- **Depurar el código CSS:** existen herramientas que permiten depurar las hojas de estilo que se utilicen en un sitio web, eliminando (entre otras cosas) aquellos fragmentos de código que no se utilicen.
- **Evitar repeticiones:** también es importante evitar redundancias en las hojas de estilo. Esto permitirá, entre otras cosas, hacer que la carga del documento sea más liviana

3. Lenguajes de marcado para presentación de páginas web.

HTML es el lenguaje de marcas a partir del cual se desarrollan los documentos web. Cuando una página se visualiza en el navegador, realmente lo que se está viendo es una traducción visual del código HTML (y CSS) que el desarrollador ha implementado en el documento.

Para crear una página web, es imprescindible conocer el lenguaje HTML, que, gracias a su sencillez, puede ser dominado en muy poco tiempo. No obstante, también es importante saber cómo aplicar estilos utilizando CSS, ya que proporciona infinidad de posibilidades (algunas de ellas imposibles de conseguir con la única utilización de etiquetas HTML) relacionadas con la apariencia de los elementos que constituyen los documentos web.

En HTML, existe un gran número de etiquetas diferentes y algunas de ellas deben estar presentes en todos los documentos web, como son `<DOCTYPE>`, `<html>`, `<body>`, `<title>`, etc.

Otras directivas permiten establecer estilos a los textos de los documentos web, aunque no se recomienda su utilización. Es su lugar, la nueva especificación HTML5 aconseja utilizar hojas de estilo CSS.

Con HTML también es posible manejar y configurar otras muchas etiquetas, que permiten enriquecer los documentos web con multitud de elementos. Algunos de los más importantes son: enlaces, párrafos de texto, imágenes, tablas, formularios, etc.

4. Hojas de estilo web.

El aprendizaje de CSS es un paso fundamental para poder diseñar sitios web atractivos y competentes, ya que ofrece infinidad de posibilidades de diseño, muchas de las cuales son imposibles de lograr con la única utilización del lenguaje HTML.

Fundamentalmente, existen tres maneras diferentes de incluir estilos CSS en un documento web. Estas son:

- Mediante el atributo `style`.
- Mediante la etiqueta `<style>`.
- Mediante un archivo **“.css”** externo.

Para añadir estilos CSS con los dos últimos métodos se utilizan los denominados selectores, los cuales consisten en indicadores que informan acerca del elemento o elementos que van a ser los beneficiarios de los estilos que establezcan. Existen cuatro tipos de selectores básicos:

- Selectores de etiquetas.
- Selectores descendentes.
- Selectores de clase.
- Selectores id.

También es importante conocer las denominadas pseudo-clases, que permiten dotar de cierto dinamismo a los estilos CSS que se pueden definir para los elementos de un sitio web.

1. Arquitecturas de aplicaciones web.

Las aplicaciones web ofrecen servicios a los usuarios que acceden utilizando un navegador. A la hora de desarrollar una web hay que tener en cuenta la estructura de su contenido: la estructura **lineal** es aquella en la que todas las páginas están dispuestas de forma consecutiva; la estructura **jerárquica** es la típica estructura en árbol; en la estructura **en red**, todas las páginas pueden enlazarse unas con otras; mientras que la estructura **mixta** es una mezcla de la estructura jerárquica con enlaces cruzados del tipo estructura en red.

Para desarrollar el diseño conceptual, se debe llevar a cabo una serie de etapas, delimitando el tema del que constará la web, recolectando la información que contendrá, clasificando dicha información y diseñándola de acuerdo a un estilo, para posteriormente proceder al ensamblaje y testeo.

El modelo cliente-servidor tiene dos tipos de nodos en la red: los clientes y servidores. Estas arquitecturas genéricas son conocidas como arquitecturas en dos niveles o dos capas.

La arquitectura de tres capas dota a la arquitectura de dos capas de una capa intermedia donde cada proceso se ejecuta por separado en plataformas distintas.

El cliente es una aplicación informática o un ordenador que consume un servicio remoto en otro ordenador, conocido como servidor, normalmente a través de una red de telecomunicaciones.

2. Navegadores web.

La interfaz de usuario es el medio a través del cual un usuario va a interactuar con el navegador.

Existen numerosos navegadores web, los principales son: *Internet Explorer*, *Chrome*, *Firefox*, *Safari* y *Opera*.

Los motores de exploración son también conocidos como motores de búsqueda. Entre los más importantes se incluyen: *Google*, *Yahoo! Search*, *MSN*, y *Bing*.

Cada navegador, de forma interna, tiene sus propios mecanismos de seguridad, y estos pueden diferir de unos navegadores a otros.

Por defecto, cada navegador ofrece unos sistemas de seguridad que, si el usuario desea extender, debe hacerlo mediante una configuración avanzada.

Los *plugins* son pequeñas aplicaciones que, una vez añadidos, complementan la funcionalidad de los navegadores.

Firebug es una extensión creada y diseñada especialmente para los desarrolladores web.

El cumplimiento de los estándares es importante no solo para que el sitio sea mejor indexado por los motores de búsqueda, sino también para permitir que las páginas sean entendidas por usuarios con navegadores distintos a los usuales, ya se encuentren en un ordenador, en un smartphone o en una pantalla de televisión, sin olvidar los navegadores de voz que leen páginas web en voz alta a personas con dificultades visuales, los navegadores braille, etc.

3. Creación de contenido web dinámico

Se denomina metodología de programación al método riguroso y sistemático que utiliza aquella metodología necesaria para resolver los problemas mediante programas.

Un dato es un conjunto de posiciones en memoria, asociado a un nombre y un identificador. Los datos que maneja un programa pueden ser constantes y variables.

Constantes son objetos cuyo valor permanece invariable a lo largo de la ejecución de un programa, mientras que las variables pueden cambiar su valor en cualquier parte de la ejecución del programa.

Los datos se pueden clasificar en dos grandes tipos: simples (numérico, carácter, lógico) y compuestos (matriz, registro, ficheros).

El control de flujo puede ser ascendente, descendente, o se pueden ejecutar un conjunto de instrucciones una determinada cantidad de veces, mediante bucles que modifiquen algunos valores. Hasta que el valor no cumpla una condición, no se dejarán de ejecutar dichas instrucciones.

Si se divide un algoritmo en unidades de menor tamaño, en donde cada fragmento realiza una tarea específica y única, se está realizando una programación modular.

El tipo de parámetro es el conjunto de valores que puede tomar una variable. Es el conjunto de transformaciones y funciones internas y externas definidas sobre el conjunto de datos. Existen dos tipos de datos: simples, como los numéricos y alfanuméricos, y estructuras de datos, que pueden ser internas o externas.

La Programación Orientada a Objetos, o POO, requiere pensar de una forma distinta sobre cómo construir una aplicación. Los objetos permiten modelar en código las tareas reales, los procesos y las ideas controladas por la aplicación. En lugar de pensar en la aplicación como un subproceso de control que pasa fragmentos de datos de una función a otra, una solución POO permite modelar la aplicación como un conjunto de objetos en colaboración, que controlan de forma independiente determinadas actividades.

Los lenguajes de programación de guión constan de un archivo de órdenes o archivo de procesamiento por lotes, conocidos como *script*. Un *script* es un programa simple, que por lo regular se almacena en un archivo de texto plano.

Las aplicaciones de internet enriquecidas o RIA son aplicaciones web que tienen la mayoría de las características de las aplicaciones de escritorio tradicionales. Estas aplicaciones utilizan un navegador web para ejecutarse, y por medio de complementos, o mediante una máquina virtual, se agregan las características adicionales.

4. Lenguajes de gui3n de uso general.

JavaScript es un lenguaje de programaci3n que realiza acciones dentro de una p3gina web. Su compatibilidad con la mayor3a de los navegadores actuales lo posiciona como el lenguaje de programaci3n del lado del cliente mas utilizado.

Mediante el uso de JavaScript se puede conseguir que las p3ginas web sean m3s interactivas, ejecutando el c3digo en el ordenador del usuario (cliente). El navegador proporciona lo que se denomina *Modelos de Objetos del Documento (DOM)*, permitiendo acceder a cualquier elemento que se muestre en la p3gina, incluyendo al propio navegador.

En JavaScript se pueden anidar diferentes sentencias de control, como por ejemplo estructuras condicionales o repetitivas.

Para utilizar las variables, estas tienen que estar declaradas previamente, es decir, haberles dado un nombre y opcionalmente un valor. Para declarar una variable, se utiliza la palabra reservada “**VAR**”, seguida del nombre de la misma.

Los par3metros se utilizan para enviar valores a las funciones. Una funci3n utilizar3 los par3metros para realizar una serie de acciones. Los par3metros son los valores de entrada de una funci3n.

Los *array* en JavaScript siempre empiezan en la posici3n 0, por tanto un *array* que tenga 10 posiciones, tendr3 casillas de la 0 a la 9. Los *array* pueden ser tambi3n multidimensionales.

Para crear una clase en JavaScript se utilizan las funciones; estas pueden llevar o no constructor (para inicializar las propiedades del objeto) y se les puede aplicar la herencia solamente a clases individuales.

El m3todo *getElementById* de *document* es la forma m3s sencilla de acceder a un elemento concreto de una p3gina.

Existe un manejador de eventos asociado a cada evento reconocido en JavaScript. Así, existen: “onAbort”, “onBlur”, “onClick”, “onChange”, “onError”, “onFocus”, “onLoad”, “onMouseOut”, “onMouseOver”, “onSelect”, “onSubmit” y “onUnload”.

Usando funciones que se ejecuten repetidamente, es posible hacer que cualquier elemento posicionado parezca deslizarse de un punto a otro de la pantalla.

Es posible cambiar las propiedades CSS de cualquier objeto. Por ejemplo, se puede crear una función que use el ID de un objeto pasado como parámetro, y que cambie su estilo CSS.

Las aplicaciones AJAX añaden una capa entre el cliente y el servidor, para gestionar la comunicación entre ambos.

Para la creación de contenidos dinámicos es posible usar lo que se conoce como un CMS o *Content Management System*. Se refiere a sitios web que, mediante el uso de una administración de contenidos, permiten la fácil creación de contenidos, ya sean nuevas páginas, noticias, eventos, etc.

Es posible realizar aplicaciones para dispositivos móviles usando JavaScript. Existen una gran variedad de entornos de desarrollo, como *Titanium*, *Phonegap* o *Jquery Mobile*.

5. Contenidos multimedia

El término multimedia se utiliza para referirse a **cualquier objeto o sistema que utiliza múltiples medios de expresión (físicos o digitales para presentar o comunicar información**

Para poder reproducir contenido multimedia en un navegador web, es necesaria la utilización de adaptadores, como es el caso de *Adobe Flash Player* o el *Quick Time*.

MIME (*Multipurpose Internet Mail Extension*), son una serie de especificaciones dirigidas al intercambio a través de internet de todo tipo de archivos (audio, vídeo, texto, etc.), de forma transparente para el usuario.

A diferencia de la descarga progresiva, la descarga streaming se realiza a través de servidores especializados, que están optimizados para la transferencia de vídeo/audio digital.

El formato JPEG o JPG es uno de los más usados, ya que contiene cierto grado de compresión, pero mantiene una buena calidad en la imagen final.

Existen diferentes tipos de formato según la compresión del audio. Es importante distinguir entre archivo y *codecs*. El *codec* codifica y decodifica los datos del audio, mientras estos datos son archivados con un formato de audio específico

Youtube es un sitio web en el que los usuarios registrados pueden subir vídeos, y cualquier usuario puede compartir o buscar contenidos mediante su buscador.

Respecto a CSS3 existe una propuesta del W3C llamada “CSS3 Speech Module” que permitirá crear hojas de estilo que definan cómo reproducir mediante un lector de pantalla el contenido de una página web. Gracias a esto, los lectores de pantalla podrían interpretar las hojas de estilo para conseguir una reproducción más correcta.



Recuerde

Un sitio web accesible es un sitio web preparado para el futuro.

En cuanto al *hardware*, existen infinidad de investigaciones y propuestas que se pueden ver a diario de nuevas interfaces hombre máquina. Controlar ordenadores con la mente, con la voz, con gestos, es algo que se ve como algo común hoy en día y que seguirá evolucionando en los próximos años. Un claro ejemplo de esto es la cantidad de dispositivos de este tipo que está apareciendo en el campo de los videojuegos o la telefonía móvil.

1. Accesibilidad web

Gracias a la accesibilidad web, todas las personas, sin importar sus limitaciones, discapacidades, dificultades, y circunstancias, pueden acceder con facilidad a internet y sus contenidos.

Existen muchos tipos de discapacidades, cada una con sus características propias y con distintas barreras a las que se enfrentarse. Para superar las barreras se utilizan productos de apoyo y técnicas de diseño que permiten el acceso a la información a todo el mundo.

Aunque existen una serie de dificultades a la hora de implantar la accesibilidad, se compensa con las ventajas que aporta.

Existen normas de accesibilidad web a nivel nacional e internacional. Todas estas normas se basan en las recomendaciones de las guías desarrolladas por el grupo WAI perteneciente al W3C. La versión 2.0 de las Pautas de Accesibilidad al Contenido en la Web (WCAG 2.0 son las guías más importantes a seguir actualmente.

Para la construcción de sitios web accesibles se cuenta con herramientas de

desarrollo y evaluación que pueden ser servicios web externos, complementos para el navegador web o aplicaciones de escritorio.

para comprobar que los cambios han surtido efecto y el diseño se ha mejorado.

2. Usabilidad web

La usabilidad web hace referencia a la rapidez de aprendizaje, la eficiencia de uso y la satisfacción de los usuarios. Para un desarrollador web, tener unos conocimientos mínimos de usabilidad es algo imprescindible porque los sitios web son cada vez más complejos e interactivos. La accesibilidad y la usabilidad se complementan y deben trabajar juntas por el bien del usuario. Y aunque en principio su implantación puede suponer un mayor esfuerzo, los beneficios que se obtienen justifican su utilización.

Siempre se debe tener presente al usuario a la hora de diseñar. Se le debe incorporar en todas las fases del proyecto y él tendrá la última palabra para decidir si un sitio web es usable o no. Para conocer las necesidades de los usuarios no es suficiente con escucharlos, es imprescindible observar cómo interactúan. Existen muchos métodos para mejorar la usabilidad y utilizar uno u otro dependerá de los recursos disponibles y de la fase de desarrollo en la que se encuentre el proyecto. El desarrollo centrado en el usuario es cíclico y mejora con cada iteración. Los principios de diseño ayudan en el desarrollo y en la evaluación de un sitio web. A la hora de diseñar una interfaz, es mejor que sea sencilla con las opciones mínimas para que sea fácil de entender y utilizar.

1. El proceso del desarrollo de software

Durante el proceso de creación de un producto *software*, conviene adoptar un modelo de desarrollo, ya que marcará las pautas a seguir para obtener un *software* de calidad. Existen muchos modelos, surgidos de la necesidad de tener metodologías que encaminasen unos desarrollos *software* que iban creciendo en complejidad. La elección entre un modelo y otro depende de ciertos factores (límite de tiempo, necesidad de entregas parciales, recursos, complejidad del problema, etc.).

El ciclo de vida clásico, que sirvió como inspiración al resto de modelos, se divide en las siguientes fases:

- Análisis de requisitos: captura de los requisitos para la realización del sistema.
- Diseño: diseño arquitectónico, de componentes y de interfaz en base a los requisitos anteriores.
- Implementación y codificación: implementación del diseño en un lenguaje de programación. Normalmente, se puede seguir una metodología de programación estructurada o de programación orientada a objetos.
- Pruebas: existiendo un especial interés en el capítulo de implementación, al introducir unas nociones básicas de programación estructurada. También se ha tratado la calidad del *software* y la verificación y validación del sistema, para establecer lo que se espera de un *software* de calidad y los diferentes estándares de aplicación.
- Mantenimiento: diseño arquitectónico, de componentes y de interfaz en base a los requisitos anteriores.

Para llevar a cabo el desarrollo del *software*, generalmente englobado dentro de un proyecto, hay herramientas, tanto gratuitas como de pago, que ofrecen la funcionalidad necesaria sobre la que apoyar una correcta planificación y ejecución del desarrollo.

2. La orientación a objetos.

La programación orientada a objetos presenta muchas características que permiten un mayor acercamiento al mundo real, trabajando con clases que se instancian como objetos. Las clases especifican cómo tiene que ser un objeto y qué tiene que llevar para comportarse como tal. Esta especificación es independiente del lenguaje de programación utilizado.

Se puede establecer una jerarquía de clases cuando se observe que un concepto del mundo real con el que se pretende trabajar requiere un comportamiento ya definido en una clase, pero agregando funcionalidad propia. En ese caso, se creará una nueva clase que herede de la que se estime y especifique los nuevos atributos y métodos requeridos.

Los objetos representados presentan una serie de atributos que determinan su estado, los cuales, idealmente, no deben ser visibles al exterior de manera directa. Se utilizarán métodos especialmente definidos para ello, tanto para modificar como para acceder a su valor.

Para implementar este paradigma de programación existen muchos lenguajes, siendo los más utilizados Java y C++. Como entorno de desarrollo se recomienda *Eclipse*, aunque existen alternativas.

Para modelar y representar un sistema orientado a objetos existen los diagramas de UML, siendo todos perfectamente válidos, con mención especial a los diagramas de clases y a los de objetos.

3. Arquitectura web

Una aplicación web es una aplicación disponible en un servidor, la cual se encuentra a la espera, pendiente de recibir peticiones por parte de un cliente para generar una respuesta y enviarla. Este cliente accede a la aplicación del servidor de diversas maneras, siendo la más común a través de un navegador web.

Las interconexiones entre equipos se basan en un modelo de siete capas (llamado modelo OSI, que especifica las funciones y protocolos que dan lugar a la comunicación.

La preparación de un servidor es un proceso que requiere la instalación de varios programas que proporcionan la funcionalidad básica (como puede ser un servidor de base de datos, por ejemplo, disponiéndose también de módulos accesorios (inicialmente deshabilitados o de terceros que amplían las posibilidades y características.

Para llevar a cabo un desarrollo, es muy normal tener instalado un servidor de manera local, trabajando codo con codo con una base de datos (también instalada en local. Para acceder a dicho servidor local, hay que escribir `http://localhost` en la barra de direcciones del navegador.

4. Lenguajes de programación de aplicaciones web en el lado servidor

Una aplicación web, por regla general, es el resultado de la combinación de varios lenguajes de programación, tanto del lado del cliente como del lado del servidor. Los primeros proporcionan interactividad, mientras que los segundos ofrecen funcionalidad gracias a la potencia de los *scripts* ejecutados en el servidor. Existen también tecnologías como AJAX, que permiten invocar *scripts* del servidor desde el lado del cliente, sin necesidad de realizar una petición que suponga un refresco completo de página.

Los applets son un ejemplo de código ejecutado en el cliente, mientras que los servlets son su equivalente en el servidor. Ambos se programan en Java, necesitando los servlets de un contenedor especial, como puede ser Apache Tomcat.

Las posibilidades de una aplicación web son enormes, permitiendo acceder tanto a ficheros almacenados en el servidor como a bases de datos para realizar consulta y modificación.

Cuando se programa una aplicación web, hay que tener en cuenta aspectos como la seguridad y gestión de errores. Para la primera, es de utilidad la gestión de sesiones, aparte de obligar al usuario a realizar una autenticación ante el sistema (cotejando una base de datos, por ejemplo. Para el segundo, se recomienda el uso de excepciones, con el fin de tratar los errores de una manera elegante y amigable con el usuario.

Para la creación de aplicaciones web (y, por extensión, la creación de cualquier programa existen varios patrones de diseño. El modelo vista controlador, por ejemplo, divide el programa en tres componentes claramente diferenciados: vista, modelo y controlador, siendo este último el eje central y coordinador entre los dos anteriores.

Resulta sumamente recomendable documentar el código con el fin de facilitar tareas de mantenimiento y extensibilidad. Java tiene una herramienta llamada Javadoc, que permite generación automática de documentación en formato HTML.

1. Modelos de datos

Cuando conozca los requisitos de una aplicación informática, debe seguir una serie de procedimientos para poder realizar una base de datos que permita desarrollar esa aplicación deseada.

El primer paso consistirá en dibujar un modelo conceptual utilizando diagramas entidad/relación. Esta será una representación gráfica del problema a resolver. Deberá identificar las entidades y las relaciones entre ellas.

Una entidad puede relacionarse con cero o con muchas entidades. Puede incluso relacionarse consigo misma.

Los diagramas entidad/relación pueden ser dibujados a mano o utilizando algún software informático, lo cual facilitará mucho el trabajo. *MySQL Work-Bench* es una buena opción para esto.

Una vez tenga el modelo conceptual, se pasará al modelo lógico, en el cual habrán de seguirse una serie de reglas de transformación para convertir un diagrama entidad/relación en un esquema relacional, el cual se compone de tablas y atributos.

Este esquema relacional debe ser normalizado, para asegurarse que posteriormente, al explotar la base de datos, no aparezcan deficiencias, anomalías o problemas de rendimiento.

Para ello, se seguirán las convenciones de normalización, las seis formas normales.

Una vez hecho esto, puede considerarse completado el diseño lógico. Este diseño lógico ya está listo para ser implementado en cualquier soporte físico que se desee. Típicamente será un Sistema Gestor de Bases de Datos, pero si el modelo de datos no es muy complejo, es posible que se desee implementarlo utilizando un sencillo sistema de archivos de texto. La utilización de uno u otro dependerá de la complejidad de la aplicación y de las limitaciones técnicas que se tengan.

2. Sistemas de Gestión de Bases de Datos (SGBD)

Un SGBD es un conjunto de aplicaciones informáticas altamente complejas que sirven para facilitar las tareas de mantenimiento de los datos de una aplicación. Proporciona un sistema fiable y escalable en el que guardar los datos.

Son una evolución de los ficheros de datos, pero con gestión de restricciones de integridad, opciones de concurrencia, transacciones y lenguajes propios para comunicarse con ellos.

Con los SGBD surge una nueva figura que es la del DBA (administrador de bases de datos: una persona especializada y con experiencia trabajando con el SGBD que utilice la empresa. El DBA se encargará de crear las bases de datos, administrirlas y de asegurarse de que son escalables.

Dos aspectos básicos de los SGBD son las transacciones y la concurrencia. Una transacción es un conjunto de operaciones que se envían a la base de datos y que se gestionan como una sola. La concurrencia es la habilidad que tiene un SGBD para manejar transacciones simultáneas que trabajen sobre los mismos datos.

Elegir un SGBD es una decisión técnica de gran transcendencia en una empresa. Determinará la manera de escalar la base de datos, su rendimiento y la manera de usarla. Siempre hay que saber por qué se elige un sistema u otro.

3. Lenguajes de gestión de bases de datos. El estándar SQL

El SQL (*Structured Query Language*) es un lenguaje inspirado en el lenguaje natural en inglés que se utiliza para interactuar con sistemas gestores de bases de datos. Es un estándar. Todos los SGBD lo implementan y adaptan para satisfacer sus requisitos propios. Este libro se centra en el SQL orientado hacia *MySQL* porque el SQL estándar por sí solo no tiene una aplicación directa. No hay ningún sistema que lo implemente como forma única de funcionamiento.

Con el SQL puede definirse un modelo de datos, creando bases de datos con `CREATE DATABASE` y definiendo las tablas mediante `CREATE`, `ALTER` y `DROP TABLE`.

Una vez definido el modelo de datos puede procederse a manipular sus datos. Para ello se cuenta con las sentencias `INSERT`, `UPDATE` y `DELETE`, que permiten insertar, modificar y eliminar filas.

Cuando las tablas contienen información es posible lanzar consultas `SELECT` sobre la base de datos para obtener los datos que se deseen. Es habitual realizar uniones entre tablas y filtraciones para obtener las filas y columnas que se necesiten en la aplicación.

Además, el SQL proporciona métodos más avanzados para obtener datos, como son las subconsultas, que permiten anidar unas consultas dentro de otras para aplicar filtraciones complejas u obtener datos de otras tablas. También pueden realizarse agrupaciones de datos, muy útiles para generar informes que de otra manera resultarían bastante complejos de realizar.

El SQL también cuenta con una sintaxis que permite gestionar los permisos de usuarios de la base de datos, dando acceso solo a determinadas áreas de la misma.

4. Lenguajes de marcas de uso común en el lado servidor.

Los lenguajes de marcado permiten compartir información entre diversos sistemas utilizando documentos de texto que, además de ser fáciles de leer por computadores, son fáciles de leer por humanos.

Los lenguajes de marcado por excelencia son el HTML y el XML. Este capítulo se ha centrado en XML. Sus grandes bazas son el soporte de caracteres internacionales, su gran difusión y su gran legibilidad.

XML consiste en etiquetas que se colocan entre signos de menor que y mayor que (<>), las cuales pueden contener otras etiquetas dentro o bien contener valores textuales. Cada etiqueta puede tener una serie de atributos. Toda etiqueta abierta debe ser cerrada.

XML está pensado para ser utilizado en servicios de la World Wide Web, como pueden ser las páginas web o servicios web.

El W3C es el órgano que regula la especificación oficial de XML y de la mayoría de sus estándares. Algunos de los estándares más usados son los de las plantillas DTD para restringir qué elementos deben aparecer en un documento XML y cómo. También son comunes las XSLT (hojas de estilos de XML, que permiten transformar un documento XML simple en otro documento como un documento HTML, LaTeX, etc. Cualquier documento cuyo formato interno sea texto.

XML proporciona diversos estándares para comunicarse con sus documentos como el DOM, XPath y XQuery. Todos ellos son estándares que buscan reflejar el árbol de nodos.

Dada la gran difusión de XML existen muchas utilidades que pueden ayudar para trabajar con documentos en este formato. Se cuenta con validadores, editores visuales, formateadores, etc.

Una manera muy habitual de leer y manipular documentos XML es mediante **lenguajes de servidor** como PHP o JSP. Es posible crear documentos XML dinámicamente cuyos contenidos provengan de diversas fuentes de datos como una base de datos o una API. También es posible leer documentos ya creados de tal forma que puedan utilizarse como una fuente de datos más.

1. Arquitecturas distribuidas orientadas a servicios

Un sistema distribuido se define como un conjunto de ordenadores separados físicamente y conectados entre sí por una determinada red de comunicaciones. Cada ordenador posee sus propios componentes *hardware* y *software*, aunque el usuario lo ve como un solo sistema (al fin y al cabo no necesita saber donde se ubica cada entidad

o recurso de dicho sistema distribuido, sino usarlo y que funcione adecuadamente en el momento justo).

En la historia de la informática han aparecido los siguientes sistemas:

- Sistemas por lotes.
- Sistemas centralizados de tiempo compartido.
- Sistemas de teleproceso (red telefónica).
- Sistemas personales (PC).
- Sistemas en red: Cliente/Servidor, protocolos.
- Sistemas distribuidos.

Sin embargo, cuando se piensa en un sistema distribuido se tiene que tener en cuenta una serie de ventajas y desventajas.

Como ventajas destacan:

- Bajo coste: se puede implementar con PC normales.
- Escalabilidad.
- Flexibilidad.
- Disponibilidad.
- Paralelismo.
- Acceso a recursos remotos.
- Transparencia.
- Eficiencia.

Como desventajas cabe citar:

- Mantenimiento costoso.
- Distribución inadecuada de recursos: mala gestión.
- Gestión de seguridades.
- Red de interconexión que no depende del desarrollador.

2. Programación de servicios web en entornos distribuidos

Las diferentes formas de acceso a un servicio web pueden ser:

Publicación/Suscripción. Gracias al servicio de Publicación/Suscripción se consigue que las aplicaciones no estén íntimamente unidas a una determinada tecnología. Este servicio se va a encargar del manejo de los datos como si fueran mensajes y libera del proveedor de dichos datos.

Repositorios. Sistema *software* encargado de almacenar recursos y metadatos y el cual va a proporcionar una interfaz para poder interactuar con ellos, o inclusive con otros repositorios.

Agentes de usuario. La interfaz que se tiene con el usuario del servicio. Los agentes de usuario van a ser los encargados de interactuar con el usuario a través de un conjunto de elementos gráficos. El mecanismo es simple: se recibe una consulta del usuario en el agente de usuario y este le devolverá los resultados obtenidos de dicha consulta.

Proveedores y consumidores. El proveedor de servicios es un nodo de la red que va a proveer el acceso a las interfaces de un determinado servicio *software* para realizar una tarea específica. Por el contrario, el consumidor de servicios es un nodo de la red el cual se va a poder conectar a un proveedor de servicios y utilizarlo para implementar una determinada solución a un problema (los proveedores pueden ser gratuitos o bien de previo pago por uso de sus servicios).