

Selección de Arquitecturas y Herramientas de Programación

Contenidos

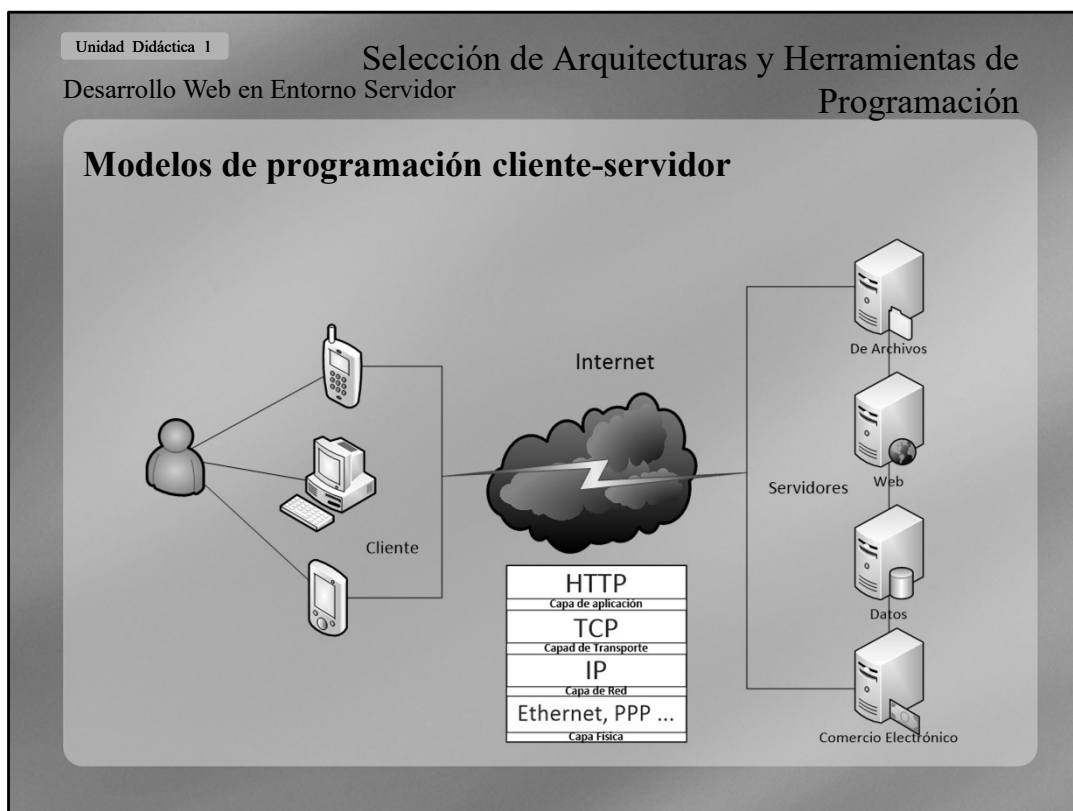
- Modelos de programación en entornos cliente/servidor
- Generación dinámica de páginas web
- Lenguajes de programación en entorno servidor
- Integración con los lenguajes de marcas
- Tecnologías asociadas
- Servidores de aplicaciones
- Integración con los servidores web
- Herramientas de programación
 - Editores y compiladores

Modelos de programación cliente-servidor

- Internet
 - Conjunto de recursos interconectados
 - Hardware: puentes, conmutadores, enrutadores
 - Software (protocolos de comunicación): TCP/IP, HTTP, FTP, etc.
 - Software (servicios): Sistema de nombres de dominio (DNS), etc.
- Configuración arquitectónica más habitual: *Cliente/Servidor*
 - Cliente: componente consumidor de servicios
 - Servidor: proceso proveedor de servicios

Para trabajar en red se necesitan tanto elementos físicos (cableado, elementos/equipos de interconexión (tarjetas de red, conmutadores, enrutadores, etc.), terminales, etc.) como elementos lógicos (protocolos (conjunto de normas a observar para llevar a buen puerto la comunicación), programas, aplicaciones, etc.)

El desarrollo de aplicaciones web que utilizan Internet debe tener en cuenta la distribución de los diferentes elementos y la función que realiza cada uno de ellos: Cliente o Servidor



En el modelo cliente/servidor es el cliente el que inicia la comunicación para demandar un servicio y el servidor responde y atiende la petición. El proceso es el siguiente:

1. **El equipo cliente solicita a un servidor web una página** con extensión .htm, .html o .xhtml. Su contenido define qué (información, datos) se debe mostrar en el navegador cliente y cómo (presentación, colores, disposición, etc.)
2. **El servidor busca esa página** en un almacén de páginas (cada una suele ser un fichero).
3. Si **el servidor encuentra esa página**, la recupera.
4. Y por último **se la envía al navegador** para que éste pueda mostrar su contenido.

Modelos de programación cliente-servidor

- Funcionalidades en los entornos cliente/servidor (*capas*)
 - **Capa de presentación:** con la que trabaja el usuario
 - **Capa lógica:** capa que conoce y gestiona las funcionalidades que esperamos del sistema o aplicación web
 - **Capa de persistencia o de datos:** capa donde residen los datos y la encargada de acceder a los mismos

Es importante decidir cómo se va a **organizar el código** de la aplicación. Muchas de las arquitecturas que se usan en la programación de aplicaciones web ayudan a estructurar el código de las aplicaciones en **capas o niveles**.

El motivo de dividir en capas el diseño de una aplicación es que se puedan **separar las funciones lógicas** de la misma, de tal forma que sea posible ejecutar cada una en un servidor distinto (en caso de que sea necesario).

Presentación: se encarga de dar formato a los datos para presentárselos al usuario final. Suele estar situada en el cliente. Presenta la interfaz gráfica con la que interactúan los usuarios. Su programación se centra en el formateo de la información enviada por el servidor y la captura de las acciones realizadas por el cliente.

Lógica: conoce y gestiona las funcionalidades que esperamos del sistema o aplicación web. Puede ubicarse tanto en el cliente como en el servidor. Recibe las peticiones del usuario y envía las respuestas apropiadas tras el procesamiento de la información proporcionada por el cliente (utiliza los datos para ejecutar un proceso y obtener un resultado).

Persistencia: mantiene los datos almacenados de forma organizada. También se encarga de acceder a los mismos para obtener e introducir los datos en el espacio de almacenamiento. Puede estar formada por uno o varios SGBD, éstos son los encargados de realizar todo el proceso de administración de datos, recibir las solicitudes de almacenamiento o recuperar la información desde la capa lógica.

- Clasificación de los modelos de programación Web:
 - Según el tamaño de los componentes:
 - *Fat Client (thin Server)*, el mayor peso de la aplicación se ejecuta en el cliente
 - *Fat Server (thin client)*, funcionalidad asociada al cliente limitada a la presentación de la información enviada por el servidor
 - Según la naturaleza del servicio ofrecido: en función de las capas ofrecidas por el servidor
 - Servidores de archivos, servidores de datos, servidores de transacciones, servidores de objetos, servidores web...
 - Reparto de funciones entre cliente y servidor: gestionar y distribuir las responsabilidades de cada una de las prestaciones funcionales entre cliente y el servidor
 - Modelo en dos capas
 - Modelo en tres capas

Tamaño de los componentes: atiende al criterio de qué elemento de la arquitectura web debe soportar más o menos carga de procesamiento. *Fat Client* El servidor se convierte en un mero administrador de datos. *Fat Server* el cliente se limita a presentar la información enviada por el servidor.

Reparto de funciones: Cada capa puede ocuparse de una o varias funciones (presentación, negocio, persistencia). Por ejemplo, en el modelo en tres capas:

- La capa cliente se encarga de la presentación, por tanto el código encargado de mostrar los datos al usuario se implementará en el cliente.
- La capa de aplicación o intermedia se encarga de la lógica de negocio, por tanto el código de la aplicación que decide cómo interactúa el usuario con la aplicación y qué resultados produce esa interacción se implementará en un servidor.
- La capa de acceso a datos se encargará de gestionar el acceso a los datos de la aplicación.

Generación dinámica de páginas Web

- Clasificación de las aplicaciones Web:
 - **Aplicaciones web estáticas**
 - El usuario recibe una página web cuya interacción no conlleva ningún tipo de acción
 - **Aplicaciones web dinámicas**
 - La interacción del cliente con el recurso recibido por parte del servidor (página web) produce algún tipo de cambio en la visualización del mismo
 - **Aplicaciones web interactivas**
 - La interacción del usuario hace que se genere un diálogo entre el cliente y el servidor

Estáticas: Utiliza HTML para organizar visualmente la información. Su contenido no cambia. Para cambiar el contenido, el programador debe modificar o actualizar su contenido

Dinámicas. DHTML. Del lado cliente. HTML, CSS, JavaScripts y otros lenguajes de guionado (scripting). Ej. Longitud de contraseña incorrecta, Campo obligatorio no informado, etc.

Interactivas. La programación puede estar en el lado cliente, en el lado servidor o en ambos. Ej. Correo web.

- Lenguajes del lado cliente: HTML (formularios), controles ActiveX (ejecución de comportamientos asociados a certificados), Flash (objetos embebidos con animaciones interactivas), applets, AJAX (carga dinámica de contenidos), etc.
- Lenguajes del lado servidor: lenguajes de guionado como PHP, ASP o Java cuyo código se embebe en el código HTML; lenguajes de propósito general como C cuyo código se compila y se enlaza a través del protocolo CGI; Java que produce objetos Servlets o ASP .Net que separa la lógica de negocio de la presentación. El contenido que se almacena en el servidor no es el mismo que se envía al navegador: el código HTML que se envía al cliente es el resultado de la ejecución de un código en el servidor.

Generación dinámica de páginas Web

- Una página web interactiva puede ejecutar código
 - En el entorno del servidor
 - y en el cliente
- Estas dos tecnologías se complementan
 - La programación del lado cliente ayuda a reducir el tráfico de red, p. e.: avisando del olvido de cumplimentar un campo obligatorio de un formulario
 - La aparición de AJAX fomenta la inclusión de código en el cliente pues acelera las consultas al servidor
- Muchas aplicaciones web actuales utilizan la combinación de estas dos tecnologías: ejecución de código en el servidor y en el cliente

En el **entorno del servidor** para generar la página web que se enviará al cliente

En el **entorno del cliente**, normalmente mediante JavaScript, para realizar animaciones, comprobación de datos introducidos en formularios e incluso modificar el contenido de la página

Una página de correo web con código AJAX en el cliente puede producir que cuando se va a pulsar con el ratón encima de un correo que se desea leer, el código AJAX detecte incluso la posibilidad de la acción y, en ese instante, consultar a través de Internet el texto que contiene ese correo y mostrarlo en la misma página, modificando su estructura en caso de que sea necesario. Es decir, sin salir de una página poder modificar su contenido en base a la información que se almacena en un servidor de Internet.

Generación dinámica de páginas Web

- Esquema de funcionamiento de una página web interactiva
 1. **El cliente web solicita** a un servidor web una **página web**
 2. El **servidor busca** la página y la recupera
 3. Si la **página es interactiva**, el **servidor web** la envía al módulo responsable de ejecutar el código de la página para que produzca el código HTML que se enviará al cliente
 - Como parte del proceso de ejecución, puede ser necesario obtener información de algún **repositorio**
 4. El **resultado** de la ejecución es una página HTML
 5. El **servidor web envía el resultado** obtenido al navegador cliente, que la procesa y muestra en pantalla

Consulta al repositorio, para, por ejemplo, consultar registros almacenados en una base de datos.

Ejemplo de página web interactiva: correo web Google.

1º Se introduce el nombre de usuario y contraseña

- Si faltan elementos la página informa de ello
- Si el usuario y/o contraseña no son correctos, la página informa de ello

2º Se muestra la bandeja de entrada del usuario con los correos recibidos.

Esta página es diferente para todos los usuarios y se generó de forma dinámica en el servidor con los datos propios del usuario.

Generación dinámica de páginas Web

- Ventajas de las aplicaciones web
 - **No** es necesario instalarlas en los equipos en que se vayan a utilizar (aplicaciones distribuidas)
 - Es muy sencillo gestionarlas
 - Se pueden utilizar en todos aquellos sistemas que dispongan de un navegador web
 - Se pueden utilizar desde cualquier lugar en el que dispongamos de conexión con el servidor

1º .Se instalan y se ejecutan solamente en un equipo, en el servidor, y esto es suficiente para que se puedan utilizar de forma simultánea desde muchos equipos.

2º Dado que sólo se encuentran instaladas en un equipo es fácil hacer copias de seguridad de sus datos, corregir errores, actualizarlas, etc.

3º Independientemente de sus características (no es necesario un equipo potente) o de su sistema operativo.

4º En muchos casos esto hace posible que se pueda acceder a las aplicaciones desde sistemas no convencionales, como por ejemplo teléfonos móviles.

Generación dinámica de páginas Web

- Inconvenientes de las aplicaciones web:
 - La interfaz de usuario de las aplicaciones web es la página que se muestra en el navegador
 - Dependemos de una conexión con el servidor para poder utilizarlas
 - La información que se muestra en el navegador debe transmitirse y procesarse desde el servidor

1º Restringe las características de la interfaz a aquellas de una página web

2º Si nos falla la conexión, no podremos acceder a la aplicación web

3º Esto hace que cierto tipo de aplicaciones no sean adecuadas para su implementación como aplicación web (por ejemplo, las aplicaciones de edición de vídeo).

Lenguajes de programación en entorno servidor

- Se caracterizan porque el código producido mediante estos lenguajes es ejecutado por un software específico en el servidor
 - El resultado de la ejecución es la página HTML que se envía al cliente
 - El código fuente puede ser interpretado o puede haber sido previamente compilado
- Tipos:
 - Lenguajes de scripting
 - Lenguajes compilados a código nativo
 - Lenguajes compilados a código intermedio

Lenguajes de programación en entorno servidor

- Lenguajes de scripting
 - Su código se ejecuta directamente a partir del código fuente original
 - Normalmente se almacenan en texto plano
- Ventajas
 - Portabilidad
 - Se pueden realizar modificaciones en el momento
- Inconvenientes
 - Menor rendimiento debido al proceso de interpretación

Cuando un cliente solicita una página con código de scripting, el servidor envía la página al **intérprete** para que traduzca y procese el programa y genere el código HTML de la página web

Portabilidad, al no ser necesario traducir el código fuente original para ser ejecutado. Es la máquina que alberga el servidor web y el módulo del lenguaje la encargada de traducir el código cuando el cliente solicita la página. Esta página puede haber sido codificada en un sistema totalmente incompatible con la máquina que hospeda el servidor web.

Inconveniente: la página se debe interpretar antes de enviar su código HTML al cliente. El proceso de traducción del código (la interpretación) es previo al envío de la página, lo que ralentiza el proceso de obtención por parte del cliente del recurso solicitado.

Lenguajes de programación en entorno servidor

- Principales lenguajes de programación
 - **PHP** (*Hypertext Processor*)
 - **ASP** (*Active Server Pages*)
 - **Perl**
 - **Python**
 - **Java - JSP** (*Java Server Pages*)

PHP, lenguaje imperativo de tipos dinámicos con posibilidad de utilizar construcciones orientadas a objetos. De código abierto (muchos desarrolladores diferentes aportan elementos al lenguaje), portable (se ejecuta en diferentes plataformas) y gratuito

ASP, propietario (Microsoft), diseñado para ser ejecutado en los servidores de Microsoft (IIS) aunque puede ejecutarse sobre otros servidores. Utiliza VBScript y Jscript como lenguajes de construcción de los guiones (scripts). En desuso.

Perl, lenguaje imperativo. De código abierto.

Python, orientado a objetos. Independiente de la plataforma. Permite realizar cualquier tipo de programa (de propósito general), incluido aplicaciones web. Posee una extensa biblioteca, variedad de tipos de datos y funciones. Es gratuito.

Java – JSP, en principio las páginas JSP son similares al resto de páginas producidas con otros lenguajes, sin embargo, cuando el cliente solicita una página con este tipo de código, el código Java intercalado en el código HTML es traducido y transformado en un Servlet que queda almacenado en la memoria del servidor. Esto aumenta su eficiencia, ya que en las sucesivas peticiones de la misma página, ésta está ya traducida.

Lenguajes de programación en entorno servidor

- Lenguajes compilados a código nativo
 - El código fuente se traduce a código máquina antes de ser ejecutado
 - El ejecutable recibe ciertos parámetros de entrada y devuelve como resultado el contenido que debe visualizar al cliente
 - El método principal para ejecutar programas binarios desde un servidor web es CGI (*Common Gateway Interface*)

Lenguajes de programación en entorno servidor

- Ventajas
 - Son los de mayor velocidad de ejecución
- Inconvenientes
 - Presentan problemas de integración con el servidor web
 - No son portables
 - Escaso rendimiento a la hora de responder a múltiples peticiones CGI simultáneamente pues no se reutilizan los procesos para atender las nuevas peticiones
 - Alternativas:
 - *FastCGI*
 - *Servlets*
 - *EJB/JavaBeans*

Velocidad, al no necesitar traducción, ya están traducidos, se ejecutan más rápido
Integración, los programas se construyen, generalmente, con lenguajes de propósito general que no están pensados para ejecutarse en el entorno de un servidor web.
Portabilidad, el código fuente ha de ser traducido en la máquina que va a albergar el programa, lo que a veces puede requerir incluso alguna modificación en el código.
Reutilización, cada nueva petición de una misma página crea un nuevo proceso en el servidor.

Lenguajes de programación en entorno servidor

- Lenguajes compilados
 - Perl
 - Python
 - C
 - C++
 - Lenguajes de propósito general capaces de producir archivos HTML a través de sentencias o instrucciones tipo `echo`

Perl: es un potente lenguaje imperativo de código libre creado originalmente para la administración de servidores que devuelve páginas web (HTML) como resultado de su ejecución. Cuenta con una amplia comunidad de usuarios y mucho código libre disponible.

Lenguajes de programación en entorno servidor

- Lenguajes compilados a código intermedio
 - El código fuente se traduce a un código intermedio, independiente del procesador, antes de ser ejecutado
 - Este código intermedio es el que se almacena en el servidor
 - Ofrecen un equilibrio entre los lenguajes interpretados y los compilados y permiten obtener aplicaciones híbridas de código repartido
- Ventajas
 - Buen rendimiento
 - Son portables

Estos lenguajes utilizan un procedimiento de compilación JIT (Just In Time, en el momento o cuando se necesita). Este término hace referencia a la forma en que se convierte el código intermedio a código binario para ser ejecutado por el procesador. Para acelerar la ejecución, el compilador puede traducir todo o parte del código intermedio a código nativo cuando se invoca al programa. El código nativo obtenido suele almacenarse para ser utilizado de nuevo cuando sea necesario y así acelerar el proceso ante una nueva petición de la página.

Rendimiento, si bien el código intermedio debe ser traducido al lenguaje máquina del equipo que hospeda al servidor (JIT), esta traducción es más rápida, y una vez traducido se deja en memoria para nuevas posibles peticiones de la página.

Portable, el código intermedio es independiente de la máquina, solo se necesita que la máquina que hospeda el servidor tenga una implementación de la arquitectura (como un contenedor de servlets)

Unidad Didáctica 1

Selección de Arquitecturas y Herramientas de
Desarrollo Web en Entorno ServidorProgramación

Lenguajes de programación en entorno servidor

- Principales lenguajes
 - C#
 - Visual Basic.Net
 - Java (Servlets)

C#, Visual Basic.Net: Son lenguajes de la arquitectura ASP.Net. Los formularios web están contenidos en archivos con una extensión ASPX que son los que el cliente solicita a través de una URL al servidor. Estos archivos ASPX contienen código HTML o estático y también etiquetas propias de la plataforma .Net. Con las etiquetas se definen *Controles Web* que se procesan del lado del servidor y *Controles de Usuario* donde los desarrolladores colocan todo el código estático y dinámico requerido por la página web

Integración con los lenguajes de marcas

- En la construcción de aplicaciones web, las etiquetas HTML
 - Pueden estar integradas en el código de los lenguajes compilados o interpretados
 - Siempre en entornos CGI
 - Posible con lenguajes como PHP, aunque no conveniente
 - Pueden integrar el código del programa
 - El código que puede variar el contenido de la página se incrusta directamente en las etiquetas HTML
 - Pueden estar separadas del código del lenguaje de programación
 - Arquitectura ASP.Net

1º Un programa realizado por ejemplo en C, incluye dentro de su código sentencias `echo` cuya ejecución produce el código HTML de la página web. De esta forma se programan, por ejemplo, los guiones CGI y los servlets. Los lenguajes interpretados también pueden integrar todo el código HTML dentro del código del lenguaje de programación, sin embargo no es conveniente dado que disminuye su eficiencia (es más rápido `<p> algo </p>` que tener que interpretar `echo "<p> algo </p>"`;))

2º Esta metodología de programación es la que se emplea en los lenguajes ASP, PHP y con las páginas JSP de Java EE

3º Cada página se divide en dos ficheros:

- Uno contiene las etiquetas HTML
- Otro el código en el lenguaje de programación utilizado

De esta forma se logra cierta independencia entre el aspecto de la aplicación y la gestión del contenido dinámico. A partir de esos ficheros se obtiene un código intermedio (CIL, Common Intermediate Language, antiguamente llamado MSIL) que es lo que almacena el servidor

Tecnologías asociadas

- Los componentes principales que participan en la ejecución de aplicaciones web en un servidor son:
 - Un servidor web
 - El módulo encargado de ejecutar el código o programa y generar la página web resultante
 - Una aplicación de base de datos
 - El lenguaje de programación

1º Recibe las peticiones de los clientes web (normalmente navegadores) y envía la página que solicitan una vez generada. El servidor web debe conocer el procedimiento a seguir para generar la página web:

- qué módulo se encargará de la ejecución del código y
- cómo se debe comunicar con él.

2º Este módulo debe integrarse de alguna forma con el servidor web, y dependerá del lenguaje y tecnología que utilicemos para programar la aplicación web.

3º Normalmente también será un servidor. Este módulo no es estrictamente necesario pero en la práctica se utiliza en todas las aplicaciones web que utilizan grandes cantidades de datos para almacenarlos.

4º Con el que desarrollar las aplicaciones.

Tecnologías asociadas

- En el desarrollo web se utilizan herramientas para
 - el diseño
 - la maquetación
 - la programación y
 - la depuración

Tecnologías asociadas

- Diseño
 - Consiste en crear esbozos de la web final mediante una herramienta gráfica, como Photoshop, GIMP o Inkscape
- Maquetación HTML/CSS
 - Consiste en convertir los esbozos creados en la fase anterior en plantillas HTML, su respectiva hoja de estilos, y las imágenes usadas
 - Es posible saltarse la fase anterior para comenzar directamente con esta fase, dependiendo de la disponibilidad y dominio de herramientas como Photoshop

Tecnologías asociadas

- Herramientas para diseño y maquetación
 - Adobe Photoshop
 - Sólo disponible para Windows, cara, la más usada (por lo que existe una gran comunidad de usuarios)
 - GIMP
 - Equivalente a Photoshop, herramienta de software libre disponible para Linux y Windows, gratuita e incluida en las distribuciones GNU/Linux
- Se utilizan para crear el diseño, así como las imágenes de la maquetación HTML

Tecnologías asociadas

- Herramientas para maquetación
 - Inkscape
 - Excelente editor gráfico vectorial, gratuito, muy fácil de usar y sencillo, multiplataforma
Linux/Unix/Windows/Mac
 - Adobe Fireworks
 - Editor de gráficos vectoriales, caro, muy fácil de usar, solo Windows
- Se usan para crear iconos, banners y demás dibujos que se vayan a utilizar en la web

Tecnologías asociadas

- Programación cliente
 - Se utilizan lenguajes como JavaScript, VBScript, Java (applets), Flash, etc.
- Programación servidor
 - Esta fase se desarrolla a la par de la anterior
 - Crea la aplicación web en un lenguaje de servidor, como PHP, ASP .NET, Python, Perl, etc.

Tecnologías asociadas

- Herramientas para maquetación y programación cliente
 - Dreamweaver, Aptana, Amaya
 - Se utilizan para crear el código HTML/CSS (maquetación), siendo Aptana la opción más profesional para la programación cliente (Javascript)

Tecnologías asociadas

- Depuración
 - Esta fase enlaza la anterior con la siguiente
 - Se realizan pruebas unitarias, aserciones, trazas, etc.
- Pruebas en local
 - Se realizan todas las pruebas posibles en un servidor local
- Subida de archivos al hosting
 - Una vez la web está terminada y correctamente comprobada se sube al servidor del hosting de producción mediante FTP, SFTP (SSH), WebDAV o Subversion
 - Subversion es rápido y cómodo y además permite control de versiones

Tecnologías asociadas

- Pruebas en hosting
 - Se realizan las últimas pruebas en el servidor del hosting para comprobar que el cambio de servidor no ha afectado a nada
 - Para evitar problemas, el servidor local se debe configurar exactamente igual que el servidor del hosting

Servidores de aplicaciones

- Es el contendedor de los componentes que definen la lógica de negocio y de acceso a los datos de la aplicación distribuida
 - Proporcionan servicios que soportan la ejecución y disponibilidad de las aplicaciones desplegadas
 - Ofrecen un entorno integrado para implementar y ejecutar aplicaciones empresariales personalizadas basadas en servidor
- Mejoran tres aspectos de las aplicaciones
 - Alta disponibilidad
 - Escalabilidad
 - Mantenimiento

El concepto de servidor de aplicaciones está relacionado con el concepto de sistema distribuido. Un sistema distribuido, en oposición a un sistema monolítico, permite mejorar tres aspectos fundamentales en una aplicación:

Alta disponibilidad: un sistema debe estar funcionando las 24 horas del día los 365 días al año. Para poder alcanzar esta característica es necesario el uso de técnicas de balanceo de carga y de recuperación ante fallos (*failover*).

Escalabilidad: hacer crecer un sistema cuando se incrementa la carga de trabajo (el número de peticiones). Cada máquina tiene una capacidad finita de recursos y por lo tanto sólo puede servir un número limitado de peticiones.

Mantenimiento: versatilidad a la hora de actualizar, depurar fallos y mantener un sistema. La solución al mantenimiento es la construcción de la lógica de negocio en unidades reusables y modulares.

Servidores de aplicaciones

- Proporcionan una estructura en tres capas
- Suministran los componentes para interactuar con los servicios propios del servidor
 - Middleware
 - Manejo de transacciones
 - Seguridad
 - Multihilo
 - API
 - Soporte de estándares como HTML, XML, IIOP, JDBC, SSL, etc.

Los servidores de aplicación también brindan soporte a una gran variedad de estándares, tales como HTML, XML, IIOP, JDBC, SSL, etc., que les permiten su funcionamiento en ambientes web (como Internet) y la conexión a una gran variedad de fuentes de datos, sistemas y dispositivos.

No todas las aplicaciones web o de empresa necesitan un servidor de aplicaciones para funcionar. Una pequeña aplicación que acceda a una base de datos no muy compleja y que no sea distribuida probablemente no necesitará un servidor de aplicaciones, tan solo con un servidor web (usando servlets y jsp) sea suficiente.

Servidores de aplicaciones

- Ventajas
 - Centralización
 - Disminución de la complejidad en el desarrollo de aplicaciones

Servidores de aplicaciones

- El término servidor de aplicaciones nace con la tecnología Java EE (anteriormente J2EE)
 - Java EE no es un producto, es un estándar: son las empresas de software las encargadas de implantar las soluciones
- Los servidores Java EE trabajan con tecnología Java
 - Los desarrollos pueden ser multiplataforma sin necesidad de realizar cambios importantes

Servidores de aplicaciones

- Servidores de aplicaciones Java EE
 - WebLogic
 - Glassfish
 - Websphere
 - JBoss AS
 - JOnAS
 - Geronimo
- La portabilidad de Java hace que estos servidores de aplicaciones sean multiplataforma: Unix/Linux, Windows, etc.

Servidores de aplicaciones

- Microsoft posee también su solución propietaria a través de IIS
 - No es portable, está diseñado para la arquitectura Windows
 - La construcción de aplicaciones se apoyan en la tecnología .Net y las soluciones no serán muy portables
 - Los lenguajes para crear las aplicaciones dinámicas son C# y Visual Basic .Net principalmente

Integración con los servidores Web

- La funcionalidad básica de un servidor web es:
 - Proveer de contenido estático a un cliente que ha realizado una petición a través de un navegador
- Es frecuente que el servidor web se encuentre íntimamente relacionado con el servidor de aplicaciones
 - El servidor web se encarga de realizar la labor de presentar los datos que ha devuelto el servidor de aplicaciones para enviarlo al cliente web
 - El servidor de aplicaciones es el contendedor de los componentes que definen la lógica de negocio del programa distribuido

Integración con los servidores Web

- El servidor IIS integra soporte de forma modular para manejar peticiones de páginas dinámicas ASP.Net
 - La utilidad de administración del servidor web incluye funciones de administración de las aplicaciones web instaladas
- Para integrar los servidores Java EE con Apache es necesario una serie de módulos (*mod_proxy.so* y *mod_proxy_http.so*) además de realizar una serie de configuraciones

Herramientas de programación

- Instrumentos involucrados en el desarrollo Web:
 - Navegadores
 - Editores de documentos
 - Entornos de programación
 - Entornos de desarrollo integrados
 - Herramientas de tratamiento de imágenes
 - Herramientas para la creación y administración de bases de datos
- *Entornos de programación*: programas, aplicaciones o simples utilidades destinadas a la programación

Herramientas de programación

- Los entornos de desarrollo integrados (IDE) agrupan en un único programa muchas de las herramientas necesarias para crear aplicaciones web (editores de texto, de imágenes, etc.)
- Algunos entornos son específicos de la plataforma
 - Visual Studio, de Microsoft, para desarrollar aplicaciones en lenguaje C# o Visual Basic para la plataforma .Net
- Otros son libres, multiplataforma, gratuitos que permiten personalizar el entorno para trabajar con diferentes lenguajes (Java, PHP, Python, C, etc.)
 - Eclipse
 - NetBeans

Herramientas de programación

- Características IDE
 - Generación automática de código
 - Resaltado de texto
 - Completado automático
 - Navegación por el código
 - Comprobación de errores durante la edición
 - Ejecución y depuración
 - Gestión de versiones
- Algunas de estas características son compartidas por muchos editores de código generalistas
 - Notepad++,
- O específicos
 - CoffeeCup, Amaya, etc.

Generación automática de código: Crean automáticamente la estructura básica de la aplicación o de sus elementos individuales (clases, funciones, etc.)

Resaltado de texto: Ayudan a identificar mejor la sintaxis del lenguaje, cambiando de color las etiquetas, realizando tabulaciones en el texto, etc.

Completado automático: Muestra distintas opciones para completar el texto que se está escribiendo.

Navegación en el código: Permite la búsqueda rápida de elementos de la aplicación así como escribir notas de elementos por completar del programa.

Comprobación de errores al editar: Reconoce la sintaxis del lenguaje y revisa el código en busca de errores mientras se escribe el código.

Ejecución y depuración: El IDE se puede encarga de ejecutar el programa para probar su funcionamiento. Cuando algo no funciona correctamente permite depurar el programa con herramientas como la ejecución paso a paso, el establecimiento de puntos de ruptura o la inspección del valor que almacenan las variables, pruebas unitarias, etc.

Gestión de versiones: Junto con un sistema de control de versiones, el IDE ayuda a guardar copias del estado del proyecto a lo largo del tiempo, para en caso necesario revertir el proyecto a una versión estable anterior.

Trabajos

- Servidores de aplicaciones Java EE y Microsoft IIS: arquitectura básica y mecanismos de extensión para el soporte de diferentes tecnologías de programación en entorno servidor
- Servlets, JSP (Java Server Pages) y EJB (Enterprise JavaBeans)
 - Qué son, para qué se utilizan, en qué se diferencian
 - Características de cada uno
 - Para qué tipos de aplicaciones web es mejor cada uno de ellos ¿es indiferente? ¿por qué? ¿qué característica diferenciadora tiene cada uno de ellos?