

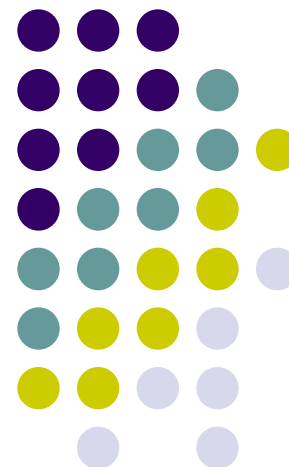
Desarrollo de Aplicaciones Web

Desarrollo Web en Entornos de Servidor



Tema 1

Generación dinámica de páginas Web. Arquitecturas y herramientas de desarrollo



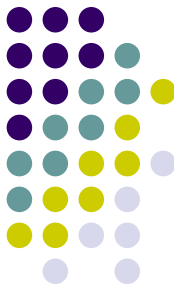
Vicente J. Aracil Miralles

vj.aracilmiralles@edu.gva.es

03/09/2024

Tema 1

Generación dinámica de páginas Web. Arquitecturas y herramientas de desarrollo

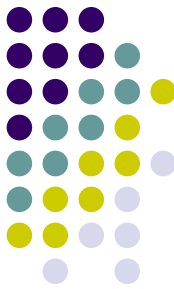


Objetivos

- Identificar el modelo de arquitectura del software Cliente/Servidor.
- Considerar la arquitectura del software de una aplicación Web como un caso particular de la arquitectura Cliente/Servidor.
- Conocer la agrupación de funcionalidades del modelo de desarrollo del software de tres capas en los entornos basados en una arquitectura del software Cliente/Servidor.
- Reconocer las ventajas de la generación dinámica de páginas Web.
- Comprender los mecanismos de ejecución de código en los entornos basados en una arquitectura del software Cliente/Servidor, considerando las diversas tecnologías de desarrollo Web existentes.
- Identificar y caracterizar las distintas tecnologías, *frameworks* y herramientas de desarrollo existentes para el desarrollo de aplicaciones Web con procesamiento en el lado del servidor.
- Comprender los aspectos básicos de la tecnología de desarrollo ASP.NET de Microsoft.
- Conocer las características de uso del entorno de desarrollo Microsoft Visual Studio.

Tema 1

Generación dinámica de páginas Web. Arquitecturas y herramientas de desarrollo



Contenidos

- 1.1 Aspectos previos: el servicio de información Web
- 1.2 Modelos de desarrollo en entorno cliente/servidor
- 1.3 Generación de páginas Web de forma dinámica
- 1.4 Tecnologías de desarrollo Web en el servidor
- 1.5 Integración con servidores Web
- 1.6 Herramientas de desarrollo
- 1.7 Aspectos básicos de ASP.NET
- 1.8 Uso de Microsoft Visual Studio

1.1 Aspectos básicos: el servicio de información Web



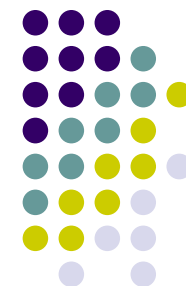
El servicio de información Web (*World Wide Web*)

- Representa un universo de información accesible a través de Internet
 - Está formado por un conjunto de recursos de información interconectados que conforman buena parte del conocimiento humano actual
- Su funcionamiento es posible debido a la coexistencia de una serie de elementos:
 - Lógicos (software)
 - Físicos (hardware)
- Dentro de un contexto orientado hacia el desarrollo Web en entorno servidor se consideran exclusivamente los elementos lógicos intervinientes



[Esta foto](#) de Autor desconocido está bajo licencia [CC BY-NC](#)

1.1 Aspectos básicos: el servicio de información Web

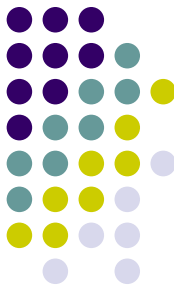


El servicio de información Web (*World Wide Web*)

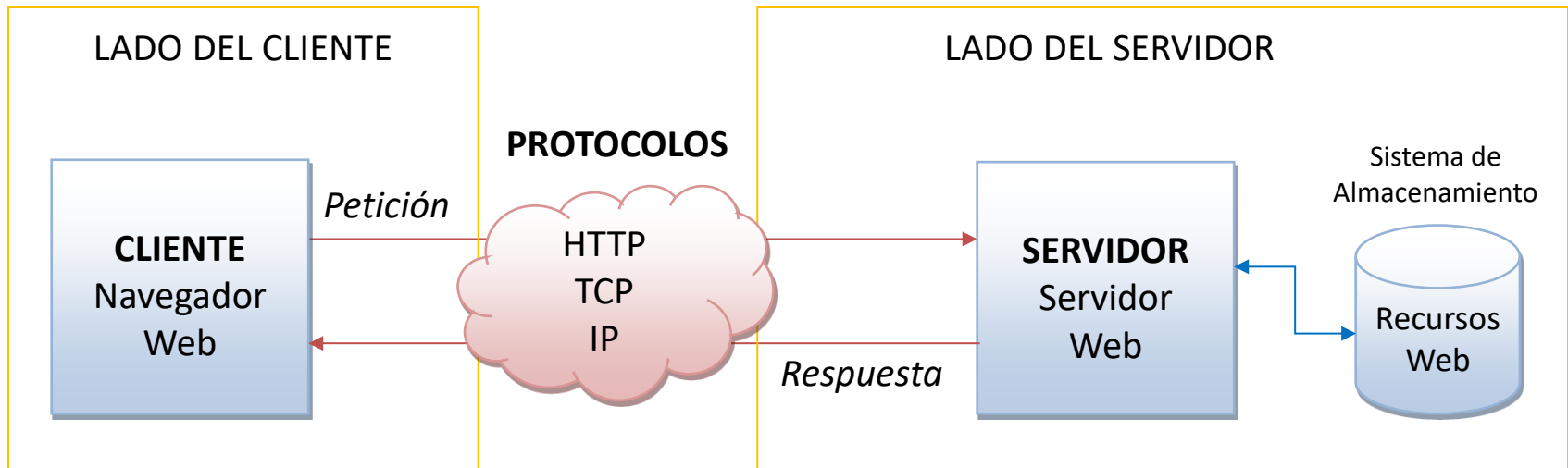
- Es un **servicio hipermedia o multimedia de acceso a la información a través de Internet** que está basado en:
 - El uso de la **infraestructura de Internet** basada en la familia de protocolos TCP/IP
 - La utilización de documentos **HIPERTEXTO** para facilitar el acceso y la navegación entre los recursos solicitados mediante enlaces o *links*
 - El modelo de arquitectura del software **CLIENTE/SERVIDOR** que incorpora un esquema de **Petición-Respuesta** (Solicitud-Envío) para el intercambio de información entre ambos
 - El sistema **URI** (*Uniform Resource Identifier*) para la identificación genérica de los recursos en la red. En el caso particular de referencias a direcciones Web se emplea el sistema URL (*Uniform Resource Locator*)

URI	esquema://autoridad/ruta/recurso	ftp://ftp.iesmarenostrum.com/pub/Tema1.pdf
URL	http://autoridad/ruta/recurso	http://www.iesmarenostrum.com/default.htm
- El **protocolo de comunicaciones HTTP** usado como mecanismo de transmisión de los recursos solicitados. Especifica la sintaxis, semántica y sincronización que utilizan el cliente y el servidor para comunicarse e intercambiar información entre ambos

1.1 Aspectos básicos: el servicio de información Web

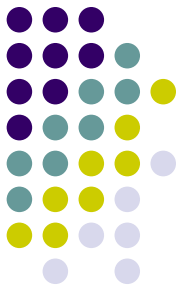


Dinámica de funcionamiento del servicio Web



- 1) El navegador Web (cliente) utiliza una URL, o mejor una URI, para conectarse con el servidor Web (servidor) y solicitarle un recurso que contiene la información requerida
- 2) Establecida la conexión, el servidor Web localiza el recurso solicitado dentro del sistema de almacenamiento asociado y lo envía al cliente como respuesta
- 3) Conforme se va recibiendo el contenido del recurso solicitado, éste va siendo interpretado por el cliente, reconstruyéndose en la ventana de visualización del navegador Web

1.2 Modelos de desarrollo en entorno Cliente/Servidor



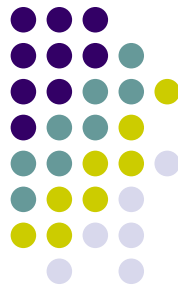
Modelo de arquitectura del software Cliente/Servidor

- El modelo de desarrollo de software en entornos Web se fundamenta en el modelo de arquitectura del software Cliente/Servidor



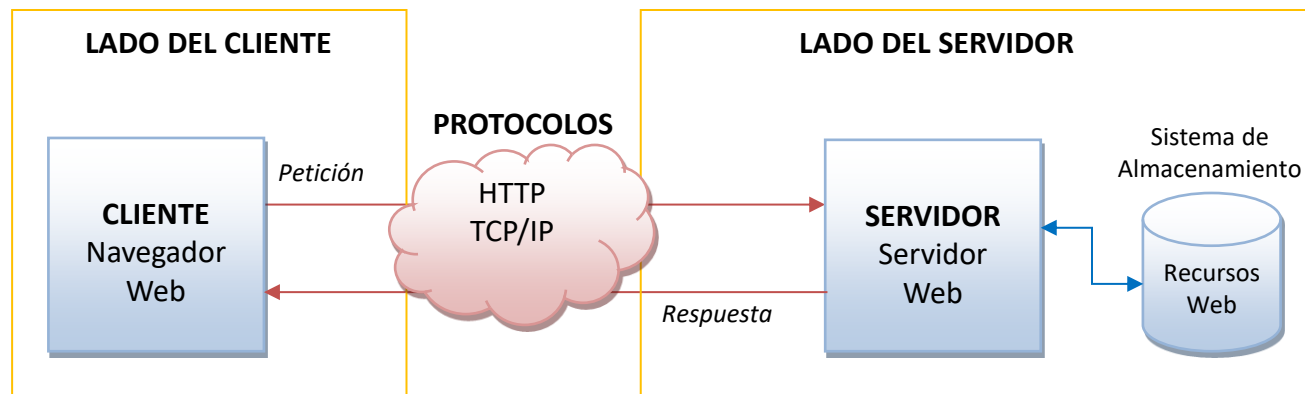
- Está basado en la idea de servicio: el cliente es un proceso consumidor de servicios y el servidor es un proceso proveedor de servicios
- La relación entre el cliente y el servidor se establece en base al intercambio de mensajes a través de un protocolo de comunicaciones, como el único elemento de acoplamiento entre ambos
- Elementos intervinientes: el cliente, el servidor y el protocolo de comunicación
 - En un proceso habitual, el cliente es el que inicia el intercambio de información, solicitando datos al servidor (Petición). El servidor responde enviando un flujo de datos al cliente (Respuesta)
 - Este intercambio de información suele incluir información administrativa adicional: el nombre del recurso a transferir, las direcciones de red y el nombre de los dispositivos, etc.

1.2 Modelos de desarrollo en entorno Cliente/Servidor



Modelo de arquitectura del software en entornos Web

- La arquitectura del software en entornos Web puede ser considerado como un caso particular de un modelo de arquitectura del software Cliente/Servidor, en el cual los elementos software intervinientes están constituidos, fundamentalmente, por software estándar.
- Cliente.** Cualquier navegador Web (Edge, Chrome, Firefox, Safari, Opera, etc.) ejecutado desde cualquier dispositivo con acceso a internet
- Servidor.** Cualquier servidor Web (Internet Information Services (IIS), Apache, Nginx, etc.
- Protocolos de comunicación.** HTTP (servicio Web) y TCP/IP (Internet)

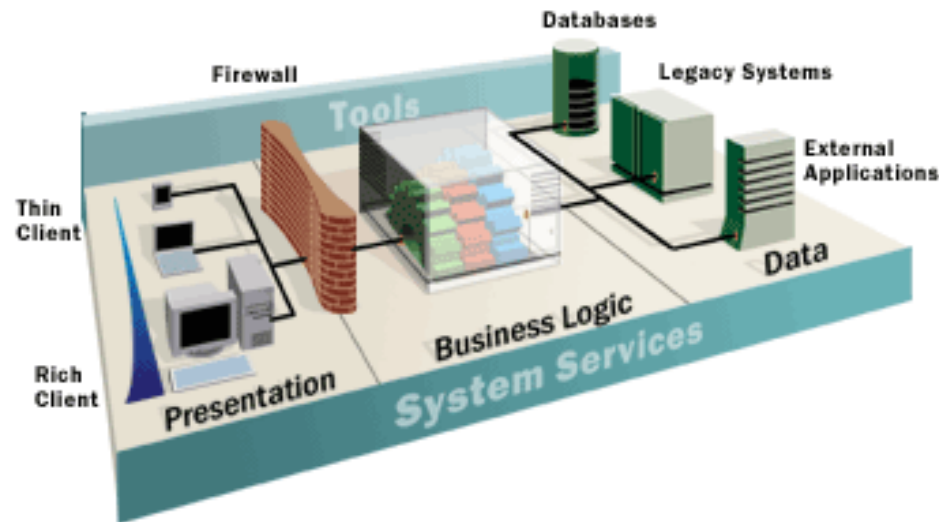


1.2 Modelos de desarrollo en entorno Cliente/Servidor

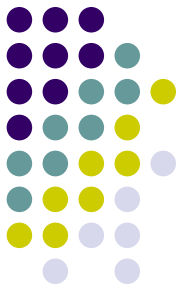


Modelo de desarrollo de software de tres capas

- Cuando se desarrolla una aplicación para la Web, como caso particular de una arquitectura Cliente/Servidor, las funcionalidades del software suelen quedar agrupadas en diferentes capas, cada una especializada en la gestión de un aspecto concreto del servicio



Capa	Presentación	Lógica del negocio	Persistencia o de datos
Tareas	Interfaz de usuario	Lógica del procesamiento	Acceso a datos
Software	Navegador Web (Cliente)	Navegador Web (Cliente) y/o Servidor Web (Servidor)	Servidor de Base de Datos



✂ Modelo de desarrollo de software de tres capas

- **Capa de presentación**

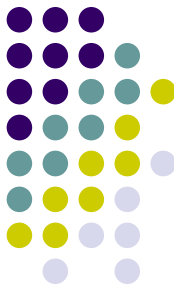
- Corresponde con la interfaz de usuario que es procesada por el cliente
- El desarrollo de esta capa se centra en el formateo de la información enviada por el servidor y la captura de las acciones interactivas realizadas por el usuario en el cliente
- Un navegador Web (cliente) únicamente es capaz de interpretar documentos HTML

- **Capa de lógica del negocio**

- Soporta y gestiona las funcionalidades que se esperan de la aplicación Web. Representa la lógica o reglas de negocio que implementa la solución software
- El desarrollo de esta capa consiste en recibir las peticiones del usuario y enviar las respuestas apropiadas tras el procesamiento de la información
- Puede ser procesada por el cliente (navegador Web) y/o por el servidor (servidor Web)

- **Capa de persistencia o datos**

- Es la capa donde residen los datos que maneja la aplicación y se encarga de su acceso
- Está formada por uno o más servidores de BD que reciben solicitudes de manipulación (almacenamiento o recuperación) de información desde la capa de negocio. El resultado del acceso a la información solicitado es enviado hacia la capa de negocio



Modelo de desarrollo de software de tres capas

- Las propuestas de arquitecturas para el desarrollo de aplicaciones Web se suelen basar en un modelo de desarrollo de software en tres capas
 - Dependiendo de la especialización de las capas, de las tecnologías empleadas y del reparto de las funcionalidades agrupadas en capas, pueden encontrarse otras configuraciones con más o menos capas. El diseño de las capas determina el número de capas que se utilizarán. Por este motivo, generalizando, este modelo de desarrollo suele denominarse **modelo de desarrollo de software en capas**, o también **modelo de desarrollo de software de n capas**
- Las tecnologías de desarrollo Web actuales permiten gestionar y distribuir cada una de las funcionalidades de las capas del software entre el cliente y el servidor, todo ello utilizando un modelo de desarrollo de software en capas
 - El diseño y desarrollo de cada una de las capas influye a la hora de seleccionar el elemento de la arquitectura Web que soportará más o menos carga de procesamiento. Así se suele hablar de:
 - Configuraciones de cliente rico (*Rich Client*). La mayor carga de procesamiento la soporta el cliente.
 - Configuraciones de cliente ligero (*Thin client*). El mayor peso del procesamiento recae en el servidor y la funcionalidad asociada al cliente está limitada a la presentación de la información enviada por el servidor

En el ámbito del desarrollo de aplicaciones Web en el entorno del servidor, se emplean configuraciones de cliente ligero

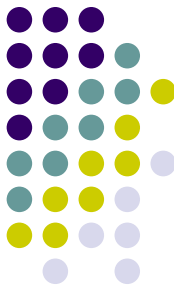




Tecnologías de desarrollo de aplicaciones Web

- **En el lado del cliente**

- Se suelen emplear, entre otras, las siguientes tecnologías:
 - Lenguaje de marcas **HTML** para estructurar la presentación de la información
 - Hojas de estilo en cascada **CSS** para definir la presentación visual de los elementos de la página
 - **Objetos embebidos** que muestran diversos tipos de contenido
 - **Scripts de cliente** para incluir código procedural mediante un lenguaje de script interpretado por el cliente (Javascript)
 - Además, los scripts de cliente permiten la creación de **aplicaciones Web interactivas desde el cliente a través conexiones asíncronas** con el servidor Web para realizar cargas dinámicas de contenidos



Tecnologías de desarrollo de aplicaciones Web

- **En el lado del servidor**

- Fundamentalmente, se suelen emplear las siguientes tecnologías:
 - **Scripts de servidor** embebido en el código HTML (ASP, PHP, JSP, etc.)
 - Enlaces a **ejecutables externos** (CGI)
 - **Componentes en objeto** (EJB de Oracle) que encapsulan la lógica del negocio de la aplicación Web en componentes precompilados.
 - **Tecnologías de código subyacente** (ASP.NET de Microsoft) que se basa en separar la presentación de la página de la lógica de negocio.
- El lado del servidor suele incorporar otras capacidades, además del servidor Web, como pueden ser: servidor de base de datos, servidor de aplicaciones, servidor de objetos o componentes, servidor de transacciones, etc.
 - Se denomina **entorno del servidor Web** al conjunto de estas capacidades que forman parte del lado del servidor, incluido el propio servidor Web



Concepto de aplicación Web



- Definición de **Aplicaciones Web** o **Aplicaciones para la Web**:

Aquellas aplicaciones informáticas cuya **interfaz de usuario se construye utilizando páginas Web que son interpretadas por un navegador Web**, en lugar de construirse sobre ventanas y controles específicos de un sistema operativo concreto

- Como cualquier otro tipo de aplicación informática, están constituidas por:
 - **Un conjunto de procesos**, representados por páginas Web que pueden incluir procesamiento en el cliente o en el servidor (capas de presentación y capa de lógica del negocio)
 - **Un conjunto de datos**, almacenados en Bases de Datos (capa de datos)
- Clasificación funcional de las aplicaciones Web, considerando la capacidad de interacción de los usuarios con la interfaz Web que se les presenta:
 - **Estáticas**
 - **Dinámicas**
 - **Interactivas**





Clasificación funcional de las aplicaciones Web

- **Aplicaciones Web estáticas**

- Son aquellas aplicaciones Web en las que el usuario recibe solo páginas Web pasivas, donde la interacción del usuario con la página Web o recurso recibido no produce ningún tipo de acción, ni en las propias páginas, ni genera respuesta alguna por parte del servidor
- Son procesadas (interpretadas) solo por el cliente (navegador Web)
- Este tipo de aplicaciones suele realizarse utilizando los lenguajes **HTML** y **CSS** para especificar la presentación de la información y la organización visual
- Un ejemplo típico de este tipo de aplicaciones Web:
 - Sitio Web corporativo de cualquier organización, en el que los contenidos presentados se modifican a través de tareas de mantenimiento de las páginas Web que conforman el sitio Web



Clasificación funcional de las aplicaciones Web

- **Aplicaciones Web dinámicas**

- Son aquellas aplicaciones Web en las que la interacción del usuario con el recurso recibido (página Web) produce algún cambio en relación con la visualización del mismo, es decir, cambios en la capa de presentación
- Estos cambios dinámicos, o a lo largo del tiempo, se refieren a:
 - Modificaciones de formato
 - Ocultación de partes del documento
 - Creación de elementos nuevos, etc.que pueden ser ejecutados tanto en el cliente como en el servidor, o incluso en ambos conjuntamente
- Las tecnologías Web más significativas involucradas en el desarrollo de este tipo de aplicaciones Web son: **HTML, CSS, scripts de cliente (JavaScript), objetos Flash u otros objetos embebidos, controles ActiveX, controles de servidor, etc.**



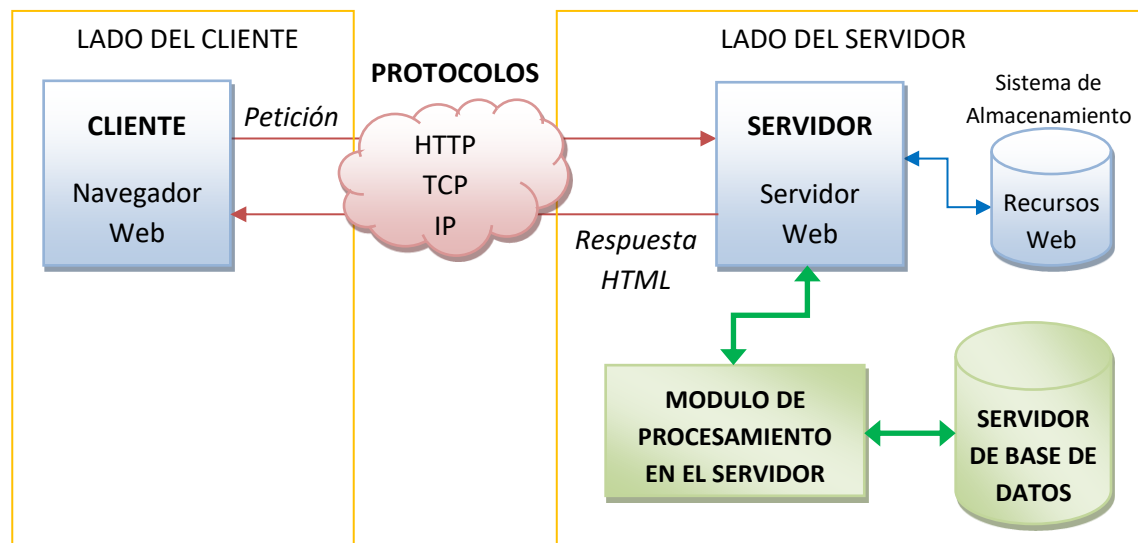
Clasificación funcional de las aplicaciones Web

- **Aplicaciones Web interactivas**

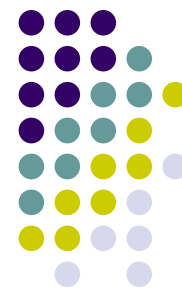
- Son aplicaciones Web en las que **la interacción del usuario produce algún cambio en la información contenida en el recurso recibido**
- Se basan en que la interacción del usuario hace que se genere un intercambio de información o diálogo entre el cliente y el servidor
 - La lógica del procesamiento asociada al inicio y a la gestión de dicho diálogo puede ser ejecutada en el entorno del cliente, en el entorno servidor, o en ambos conjuntamente
- El interés actual por el desarrollo de aplicaciones Web interactivas se relaciona con la necesidad de integrar en la Web los datos almacenados en bases de datos
- Permiten **representar funcionalidades complejas** relacionadas con:
 - **Presentar información de naturaleza cambiante a lo largo del tiempo**, y/o para
 - **Ajustarse a las necesidades de información propias de cada usuario**
- Gran diversidad de tecnologías de desarrollo involucradas
- El **ámbito de estudio** de este módulo se refiere al desarrollo de aplicaciones Web interactivas usando tecnologías de procesamiento en el entorno del servidor Web

1.3 Generación de páginas Web de forma dinámica

Dinámica del procesamiento de código en el servidor Web



- 1) El navegador Web solicita un recurso que contiene código de procesamiento en el servidor
- 2) Establecida la conexión con el servidor Web, al detectar que incluye código escrito a ejecutar por el servidor Web, éste realiza una llamada al módulo de procesamiento en el servidor
- 3) El módulo de procesamiento en el entorno del servidor ejecuta el procesamiento solicitado
- 4) Finalizada la ejecución, el servidor Web envía el resultado del procesamiento solicitado al cliente. **El resultado de un procesamiento en el servidor Web es un documento HTML válido** que incorpora dinámicamente en su contenido la información resultante de la ejecución
- 5) Conforme va llegando el contenido del documento HTML resultante del procesamiento en el entorno del servidor (página de respuesta), éste va siendo interpretado por el navegador Web



Generalidades

- Definición de **Tecnología de desarrollo Web en el entorno del servidor**:
 - Es aquella **tecnología de desarrollo Web cuyo código**, ya sea como código interpretado o bien como código objeto precompilado, **es ejecutado por un componente específico de software en el entorno del servidor Web**
- Se aplican solo para construir aplicaciones Web interactivas
- El código ejecutado en el servidor:
 - Representa la capa de lógica del negocio
 - Proporciona funcionalidades complejas a la aplicación Web, como el acceso a datos
- Existe una **gran variedad de tecnologías** que facilitan la ejecución de código en el lado del servidor Web. Se pueden clasificar en tres grupos:
 - Tecnologías basadas en lenguajes de script de servidor
 - Tecnologías basadas en enlaces a programas y componentes ejecutables
 - Tecnologías de código subyacente

1.4 Tecnologías de desarrollo Web en el servidor



Tecnologías basadas en lenguajes de script de servidor

- Se basan en utilizar un lenguaje de script que es **interpretado** por el servidor Web
- El **código escrito utilizando un lenguaje de script de servidor se entremezcla con el código HTML estático**, para conformar una página de respuesta que incluya el resultado del procesamiento en el servidor
 - El código HTML especifica la estructura básica de página de respuesta
 - Las porciones de código escrito en un lenguaje de script son ejecutadas por el servidor Web para obtener y presentar los resultados del procesamiento en el servidor sobre la página de respuesta que se envía al cliente
 - Para que las porciones de código script sean ejecutadas, el servidor Web debe tener instalado un módulo de procesamiento específico, denominado *Scripting engine*. Este módulo es el encargado de reconocer e interpretar el lenguaje de script utilizado para escribir las porciones de código que se han insertado en la página Web y que especifican el procesamiento a ejecutar por el servidor
- Existe una gran diversidad de lenguajes de script de servidor Web, así como de Frameworks que están basados en esos mismos lenguajes de script
 - Por razones de productividad, es habitual abordar el desarrollo de las aplicaciones Web interactivas partiendo de un *Framework*

1.4 Tecnologías de desarrollo Web en el servidor



Tecnologías basadas en lenguajes de script de servidor

Lenguajes de script de servidor	Frameworks	Fabricante
C#, Basic, etc.	Active Server Pages (ASP)	Microsoft
PHP	Laravel Symfony CakePHP CodeIgniter Yii	Taylor Otwell Symfony SAS Cake Software Foundation, Inc Ellislab Yii Software, LLC
Java	JavaServer Pages (JSP) JavaServer Faces (JSF)	Oracle Corporation Java Community Process
Perl	Catalyst Dancer Jifty	Catalyst Foundation Alexis Sukrieh Best Practical Solutions, LLC
Ruby	Ruby on Rails Sinatra	Rails Core Team Blake Mizerany
CFML	ColdFusion	Adobe
Python	Django Flask	Django Software Foundation Pallets Organization
Javascript	Node.js Express	OpenJS Foundation OpenJS Foundation



Tecnologías basadas en enlaces a programas y componentes ejecutables

- Se basan en desarrollar **unidades de software (programas o componentes) externas que son ejecutados de forma independiente del servidor Web**
 - Una alternativa a los lenguajes de script en el servidor es utilizar enlaces a programas y componentes ejecutables, que reciban ciertos parámetros de entrada y devuelvan como resultado el contenido HTML que se enviará al cliente como respuesta
 - Tecnologías más representativas:
 - **CGI (Common Gateway Interface)**. Se basa en delegar la generación de las páginas Web de forma dinámica en el servidor a un programa ejecutable externo. Pueden utilizarse diversos lenguajes para especificar la lógica del procesamiento. La localización del programa externo a ejecutar se indica en la URL que forma la petición HTTP del cliente. La principal desventaja es su bajo rendimiento, puesto que se crea un proceso en el servidor para cada petición CGI
 - **Componentes precompilados**. Se basan en utilizar componentes precompilados en objeto que son ejecutados por un servidor de aplicaciones independiente que forma parte del entorno del servidor Web. Los componentes expresan la lógica de negocio de la aplicación Web. Existen diversas tecnologías que utilizan el lenguaje Java y la plataforma **J2EE (Java™ 2 Platform, Enterprise Edition)** de Oracle para proveer los componentes al servidor Web: **EJB (Enterprise Java Beans)** de Oracle, **Struts** de Apache Software Foundation y **Spring** de Rob Johnson



Tecnologías de código subyacente

- Se basan en **separar la lógica del procesamiento en el servidor del código de presentación**, pero manteniendo la lógica de procesamiento en librerías independientes que residen en el propio servidor Web
 - Se trata de una tecnología intermedia entre el procesamiento en el servidor basado en scripts de servidor, que puede sobrecargar los servidores Web, y el procesamiento de componentes precompilados en servidor independiente, que puede producir pérdidas de rendimiento y de fiabilidad
 - La tecnología más representativa es **ASP.NET (Active Server Pages .NET) de Microsoft** que emplea la plataforma de desarrollo .NET
 - Una aplicación Web se precompila al lenguaje nativo de .NET Framework. Posteriormente, en cada solicitud se ejecuta el código nativo a través de un módulo de software específico, denominado *Runtime*, lo que permite mejorar el rendimiento
 - Se puede utilizar cualquier lenguaje de programación orientado a objetos y mapeado por eventos que sea compatible con .NET, lo que facilita aumentar la productividad en el desarrollo
- Las tecnologías más avanzadas para construir aplicaciones Web interactivas de altas prestaciones son: ASP.NET y EJB o Struts con la plataforma J2EE



Concepto de servidor Web

- Pueden utilizarse diferentes modelos, tecnologías y lenguajes para el desarrollo de aplicaciones Web interactivas
 - En cualquier caso, la posibilidad de responder a una petición hecha por un cliente depende de las capacidades que tenga el servidor Web
- Servidor Web
 - Es un **programa cuya misión es servir información en forma de documentos HTML**
 - La función básica de un servidor Web es proveer de contenido a un cliente (navegador Web) que ha realizado una petición
 - Para poder cumplir con esta funcionalidad es necesario que el servidor Web pueda entender la petición del navegador Web. La petición debe ser correcta formalmente.
 - Las direcciones de la peticiones deben cumplir con los requisitos del sistema de identificación de recursos en la red (URI) que define los elementos que debe contener la petición: el protocolo (HTTP, FTP, etc.), la dirección del servidor Web y la descripción del recurso solicitado en forma de ruta



Interacción entre el cliente y el servidor Web

- El **desarrollo de aplicaciones Web interactivas** se basa:
 - En que la interacción del usuario pueda producir un cambio en la información contenida en el recurso recibido para adaptarse a la realidad del momento como consecuencia del procesamiento en el servidor de la información enviada desde el cliente

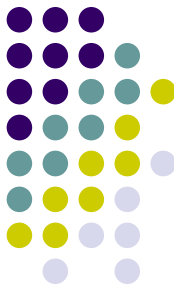
Recordando: El resultado de un procesamiento en el servidor Web es una página HTML de respuesta

- Esta interacción se consigue a través de un intercambio de información o diálogo entre el cliente y el servidor a través de mensajes de solicitud y de respuesta
 - La solicitud proporciona la descripción del recurso que debe procesar el servidor, así como la información necesaria para ello
 - La respuesta provee al cliente de la información solicitada en formato HTML
- Para poder establecer este diálogo es fundamental la información que se envía desde el cliente hacia al servidor Web a través de **formularios Web** junto con la petición HTTP



Interacción entre el cliente y el servidor Web

- **Métodos de envío** de la información contenida en los campos de un formulario Web:
 - **Método GET**
 - La información que se envía forma parte de la cadena que define la propia dirección URI. Ejemplos:
<http://www.morea.com?nombre=juan&dni=21212121-H&direccion=C/cisne,3>
<http://www.servidor3.com?nombre=jose&email=jose@infor.es>
 - Pone en riesgo la protección de la información de tipo personal
 - Tiene limitado el tamaño del envío a 256 caracteres
 - **Método POST**
 - Es el método recomendado, por razones de seguridad y protección de la información
 - Los identificadores y valores de los campos del formulario son enviados desde el cliente hacia el servidor de forma transparente al usuario y al proceso de interacción durante el envío
 - El envío se realiza empleando un sistema de encriptación
- Independientemente del método de envío utilizado, el servidor Web:
 1. Recibirá la información enviada desde el cliente a través de un formulario
 2. Realizará el procesamiento considerando la información recibida
 3. Devolverá al cliente el resultado del procesamiento, que ha realizado tras esa petición, en formato HTML



Infraestructura del servicio en el entorno del servidor Web

- El desarrollo y la explotación de las aplicaciones Web interactivas
 - Se fundamenta en la utilización de una serie de tecnologías que son ejecutadas por software especializado en uno o más servidores, entre ellos el servidor Web
- La forma en la cual se distribuye la infraestructura del servicio en el entorno del servidor Web se basa en una separación de elementos atendiendo a la funcionalidad que aporta cada uno de ellos
 - Pueden crearse tantos niveles como funcionalidades hayan involucradas en el procesamiento y el acceso a la información almacenada
- La gestión del flujo de trabajo y la **arquitectura del entorno del servidor depende:**
 - **Principalmente, de la tecnología de desarrollo que se utilice** para construir la aplicación Web interactiva
 - La distribución de la lógica del negocio de la aplicación Web
 - La forma de interactuar con la información que proviene del cliente

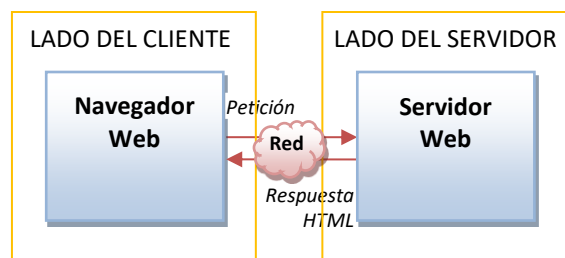
Cada tecnología de desarrollo requiere de una arquitectura y configuración determinada del entorno del servidor y, en ocasiones, de la instalación de componentes software específicos

1.5 Integración con servidores Web

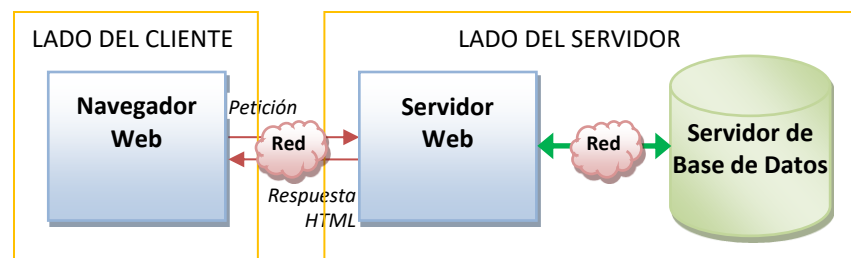


Infraestructura del servicio en el entorno del servidor Web

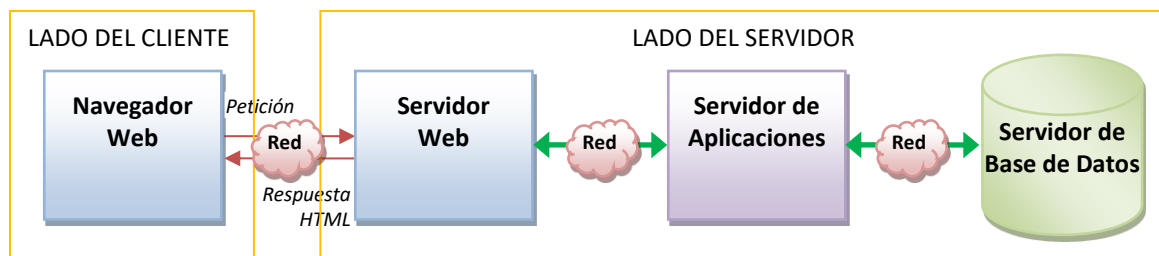
- En los entornos del servidor Web, existe una separación de elementos atendiendo a la funcionalidad que aporta cada uno de ellos



Arquitectura de 2 niveles (tradicional C/S)



Arquitectura de 3 niveles (más habitual)



Arquitectura de 4 niveles (J2EE)

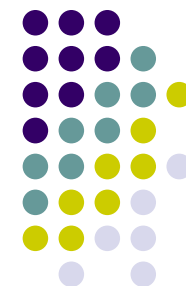
La elección y configuración del entorno del servidor Web vienen determinadas por la tecnología de desarrollo a utilizar



Entorno de ejecución de desarrollo y de explotación

- Por razones de rentabilidad, el proceso de desarrollo de una aplicación Web suele realizarse en un entorno de ejecución diferente al que se llevará a cabo la ejecución efectiva de la aplicación Web por parte de los usuarios finales. Se diferencia entre:
 - **Entorno de desarrollo.** Es un entorno de ejecución que permite realizar las tareas de construcción de la aplicación Web. Incorpora únicamente las herramientas de desarrollo necesarias para implementar de la Aplicación Web
 - **Entorno de pruebas o preproducción.** Es un entorno de ejecución que se utiliza para realizar las pruebas finales de integración de la aplicación Web, una vez que se ha finalizado su desarrollo. Debe recrear unas condiciones idénticas al entorno de explotación
 - **Entorno de explotación o de producción.** Es el entorno de ejecución final que usan los usuarios finales para la ejecución y utilización efectiva de la aplicación Web. Incorpora todas las prestaciones y características necesarias para que la ejecución de la aplicación web se realice de manera adecuada, una vez ésta que haya sido construida y probada

La operación por la cual se realiza la transferencia de la aplicación Web a un entorno de explotación, o de pruebas, una vez finalizado completamente el proceso de desarrollo, se denomina **despliegue de la aplicación Web**



Características de los servidores Web más utilizados

- **Apache HTTP Server** de Apache Foundation
 - Está diseñado para ser utilizado en múltiples plataformas y sistemas operativos
 - Se trata de un servidor robusto, que implementa los últimos estándares y protocolos de red y que cuenta con capacidad de modularización
 - Es un servidor Web de código abierto y cuenta con una comunidad de desarrolladores amplia que proporcionan documentación sobre su uso y configuración para diferentes propósitos
 - Su diseño modular, flexible y configurable permite elegir las características que van a ser soportadas por el servidor Web y se pueden seleccionar los módulos que van a estar disponibles cuando se ejecute el servidor Web
 - Gestionar el acceso a bases de datos, establecer páginas protegidas por contraseña, personalizar las páginas de error devueltas por el servidor, generar registros de actividad en múltiples formatos, etc.
 - Ofrece soporte a diferentes lenguajes y tecnologías de desarrollo como pueden ser: Perl, Python, PHP, Ruby, componentes precompilados basados en la tecnología J2EE, ASP.NET, Node.js, etc.



Características de los servidores Web más utilizados

- **Internet Information Services (IIS)** de Microsoft
 - Es un servidor Web de alto rendimiento, flexible, robusto, altamente seguro, fiable y fácil de gestionar que permite alojar cualquier tipo de contenido
 - Destaca, principalmente, por dar soporte nativo a lenguajes y tecnologías de desarrollo como ASP y **ASP.NET de forma nativa**
 - Dado que el modelo de desarrollo de aplicaciones .NET es independiente del lenguaje de programación empleado para la construcción de la aplicación Web, permite utilizar cualquier lenguaje de programación, como puede ser: PHP, Java, Perl, Python, C#, VisualBasic, etc.
 - Incorpora servicios de despliegue de aplicaciones multimedia, acceso a bases de datos o administración remota del servidor
 - Ofrece soporte a diferentes lenguajes y tecnologías de desarrollo como pueden ser: ASP.NET, Node.js, componentes precompilados basados en la tecnología J2EE, Ruby, etc.



Características de los servidores Web más utilizados

- **Nginx** de Nginx, Inc
 - En la actualidad es ampliamente utilizado
 - Se trata de un servidor Web de código abierto y cuenta con una arquitectura modular de alto rendimiento, además de permitir funcionar como servidor proxy para otros protocolos de Internet como IMAP o POP3 (correo electrónico)
 - Sus principales características son la estabilidad, la facilidad de configuración y la capacidad para ser ejecutado en múltiples plataformas, consumiendo relativamente pocos recursos
 - Soporta el lenguaje de programación **PHP de forma nativa**, sin necesidad de tener que instalar un módulo como es el caso de Apache
 - Es el servidor Web utilizado por compañías como Wordpress, Netflix, GitHub, SourceForge, etc.
 - Ofrece soporte a diversos lenguajes y tecnologías de desarrollo como pueden ser: PHP, JSP, ASP.NET, componentes precompilados basados en la tecnología J2EE, Node.js, etc.



Características de los servidores Web más utilizados

- **Particularidades de los servidores Web**

- El soporte nativo de los lenguajes y tecnologías de desarrollo Web suele proporcionar una mejora significativa de las prestaciones del servicio en cuanto a rendimiento, fiabilidad y seguridad, por lo que suele ser la opción recomendada
 - En otros casos, se precisa de la instalación y configuración de módulos, extensiones específicas para poder dar soporte a determinadas tecnologías y lenguajes de programación, como es el caso del módulo de soporte de PHP para el servidor Web de Apache.
- La tecnología basada en componentes J2EE precisa, además del servidor Web, de la participación de un servidor de aplicaciones para poder ejecutar componentes precompilados de los estándares EJB, Struts, Spring, etc.
 - Algunos de los servidores de aplicaciones Web para J2EE que actúan en combinación con Apache son los siguientes: Oracle WebLogic Server (Oracle), JBoss (Redhat), WildFly (RedHat), Glassfish (Oracle), WebSphere (IBM).
- El servidor **Web Google Web Server (GWS)** de Google LLC es utilizado por la compañía Google para dar servicio a su propia infraestructura Web
 - A pesar de que es utilizado exclusivamente por Google para el acceso a sus propios dominios y sitios Web, es uno de los servidores Web que recibe más peticiones HTTP recibe actualmente



Servicios de despliegue de aplicaciones Web en la nube (*Cloud Computing*)

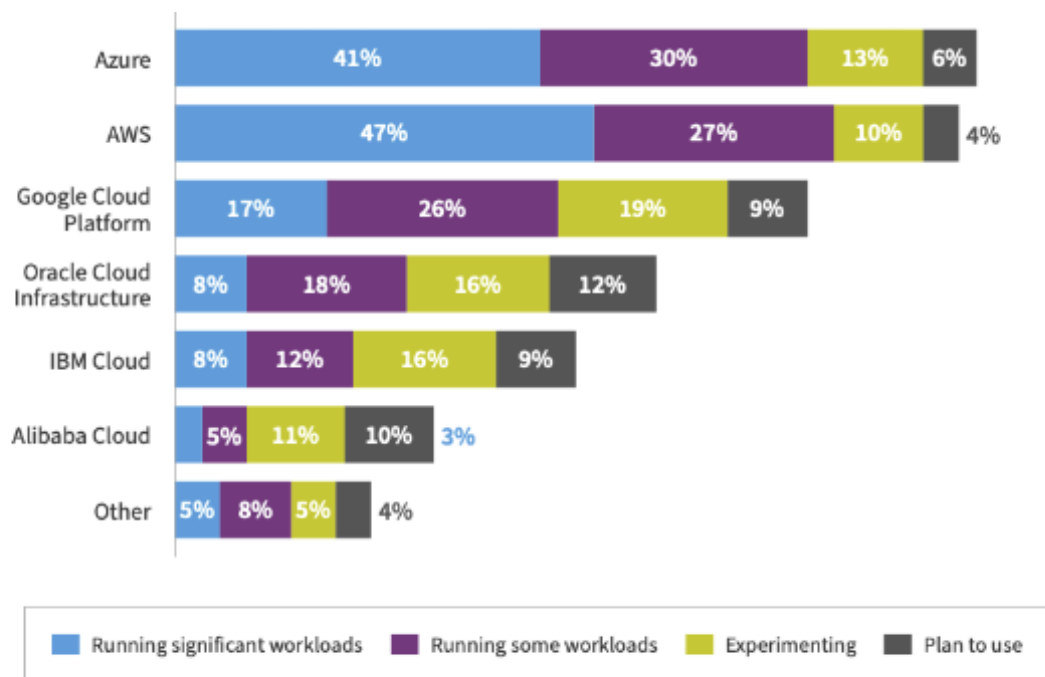
- La tendencia actual, en cuanto al uso de infraestructura de procesamiento, **es trasladar las necesidades de procesamiento a la nube (*Cloud Computing*)**
- El procesamiento en la nube se basa en la idea de utilizar **la infraestructura de soporte a las aplicaciones como un servicio** (*IaaS, Infrastructure as a Service*)
 - La razón principal de este cambio de planteamiento que se ha producido en las empresas y organizaciones viene provocado por la elevada reducción de costes que supone el uso de servicios de procesamiento en la nube frente al uso de infraestructura física propia
 - Esta tendencia que afecta, en general, a todo el software, también está afectando, de manera muy importante, al uso de la infraestructura tecnológica del entorno del servidor Web
- Actualmente, se suelen contratar **servicios de despliegue de aplicaciones Web en la nube** para implementar el entorno de explotación o producción de las aplicaciones Web interactivas, así como el entorno de pruebas. Este tipo de servicios:
 - Proveen de servidores Web virtuales y otros tipos de servicios que se puedan necesitar como, por ejemplo:: servidores virtuales de acceso a bases de datos, servidores de aplicaciones, etc..
 - Proporcionan de servicios de gestión del despliegue de aplicaciones Web, tales como **Docker** o **Kubernetes**, que automatizan todas las tareas de virtualización necesarias

1.5 Integración con servidores Web



Servicios de despliegue de aplicaciones Web en la nube (Cloud Computing)

- Los proveedores de servicios en la nube más utilizados actualmente son: **Microsoft Azure, Amazon Web Services (AWS) y Google Cloud Platform**



N=750

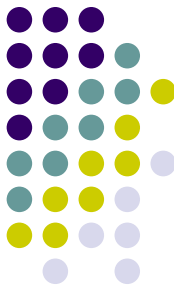
Source: Flexera 2023 State of the Cloud Report

Fuente: <https://www.flexera.com/blog/cloud/cloud-computing-trends-flexera-2023-state-of-the-cloud-report/>
Informe Flexera 2023 sobre Grado de adopción de nube pública por empresas (% de empresas encuestadas)



Generalidades

- Factores a tener en cuenta para la **selección de un entorno de desarrollo** en que se llevará a cabo la construcción de una aplicación Web interactiva
 - Las diferentes tecnologías de desarrollo Web que pueden intervenir en la construcción de la aplicación Web interactiva: HTML, CSS, frameworks de diseño Web, scripts, programas escritos en diferentes lenguajes, componentes precompilados, bases de datos, recursos de diferentes formatos, acceso a servicios de datos, etc.
 - El tipo de servidor Web
 - Los módulos y extensiones del servidor necesarios
 - Los permisos de acceso y ejecución
 - La localización de los datos utilizados por la aplicación Web
 - Especialmente importante es la selección de las **herramientas de desarrollo** que se van a emplear para la construcción de la aplicación Web interactiva



Selección de herramientas de desarrollo

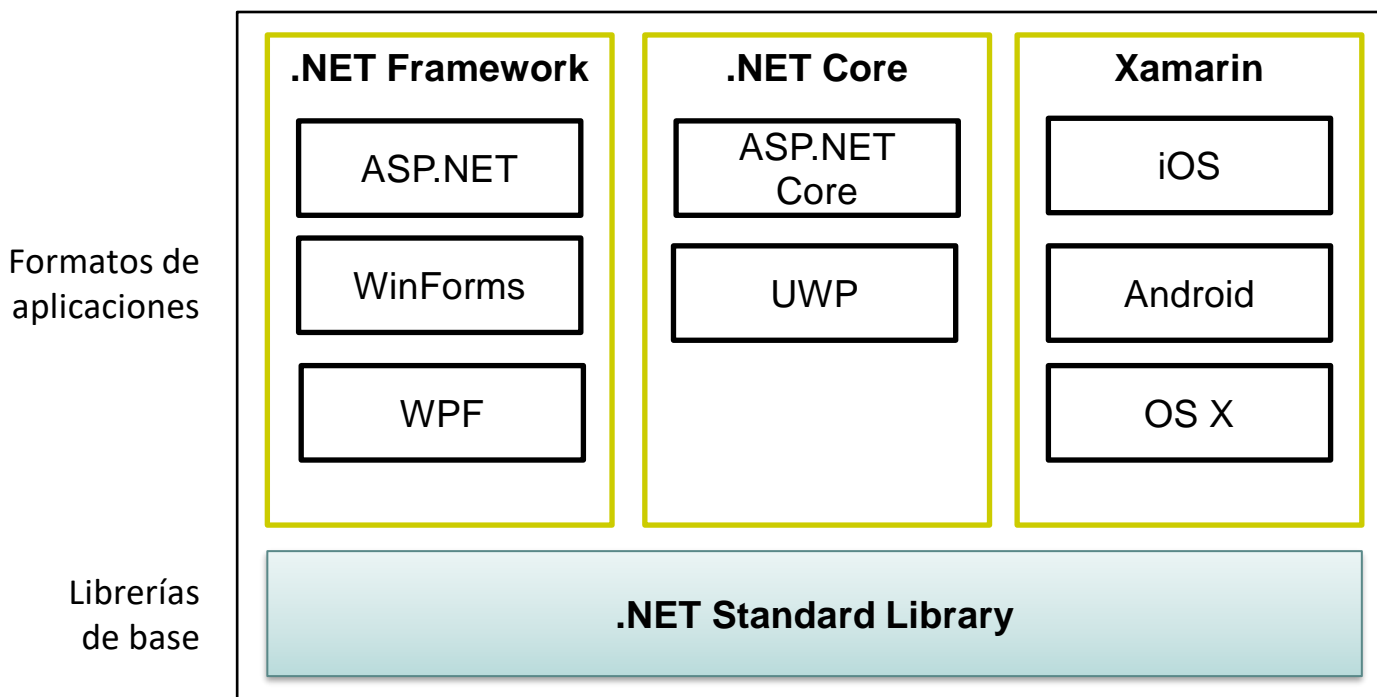
- Clasificación de herramientas de desarrollo en función de sus capacidades
 - **Navegadores.** Son los programas que permiten capturar las interacciones del usuario y presentar la información resultante en forma de documentos escritos en lenguaje HTML
 - **Editores de texto.** Son editores de texto que permiten escribir código HTML, CSS, Javascript, PHP, etc. Algunos de los editores de código más conocidos son: Notepad++, UltraEdit, Kompozer, Sublime Text, Brackets, Atom, Visual Studio Code, etc.
 - **Herramientas específicas de diseño Web.** Son herramientas visuales de ayuda a creación de páginas y sitios Web sobre entorno gráfico. Algunas de las herramientas más conocidas son: Adobe Dreamweaver, Google Web Designer, etc.
 - **Entornos integrados de desarrollo (IDE).** Permiten editar, compilar, ejecutar y probar la aplicación Web. Algunos de los IDE más utilizados son: Visual Studio de Microsoft, NetBeans de Oracle, Eclipse de Eclipse Foundation, etc.
 - **Herramientas de edición de contenidos multimedia.** Facilitan el tratamiento de los contenidos multimedia (imágenes, vídeos, sonidos, etc.) que utiliza la aplicación Web
 - **Herramientas para la creación, manipulación y administración de bases de datos.** Se trata de herramientas para la definición, manipulación y control de la información almacenada en las bases de datos que utilizará la aplicación Web

1.7 Aspectos básicos de ASP.NET



Modelo de desarrollo .NET

- El modelo de desarrollo de software .NET de Microsoft constituye una plataforma para desarrollar: **aplicaciones de escritorio para Windows, aplicaciones Web, Servicios Web (Web Services) y aplicaciones para dispositivos móviles**
 - La infraestructura básica de la tecnología .NET se denomina .NET Standard que unifica las tres implementaciones específicas que la componen: .NET Framework, .NET Core y Xamarin





Common Language Runtime (CLR)

- Un componente básico del modelo de desarrollo .NET es ***Common Language Runtime (CLR)***
 - Proporciona el entorno de ejecución administrado del código nativo de .NET Framework
 - Ejecuta el código precompilado, que se conoce como código administrado, lo que facilita la integración con una gran variedad de lenguajes de programación
 - Los compiladores que son compatibles con .NET traducen el código fuente escrito utilizando un lenguaje de programación al código administrado que ejecuta el componente CLR de .NET
 - Hace posible que la tecnología de desarrollo .NET sea **independiente del lenguaje de programación empleado para el desarrollo de las aplicaciones**
 - En la actualidad existe una gran variedad de compiladores compatibles con .NET disponibles para multitud de lenguajes de programación.



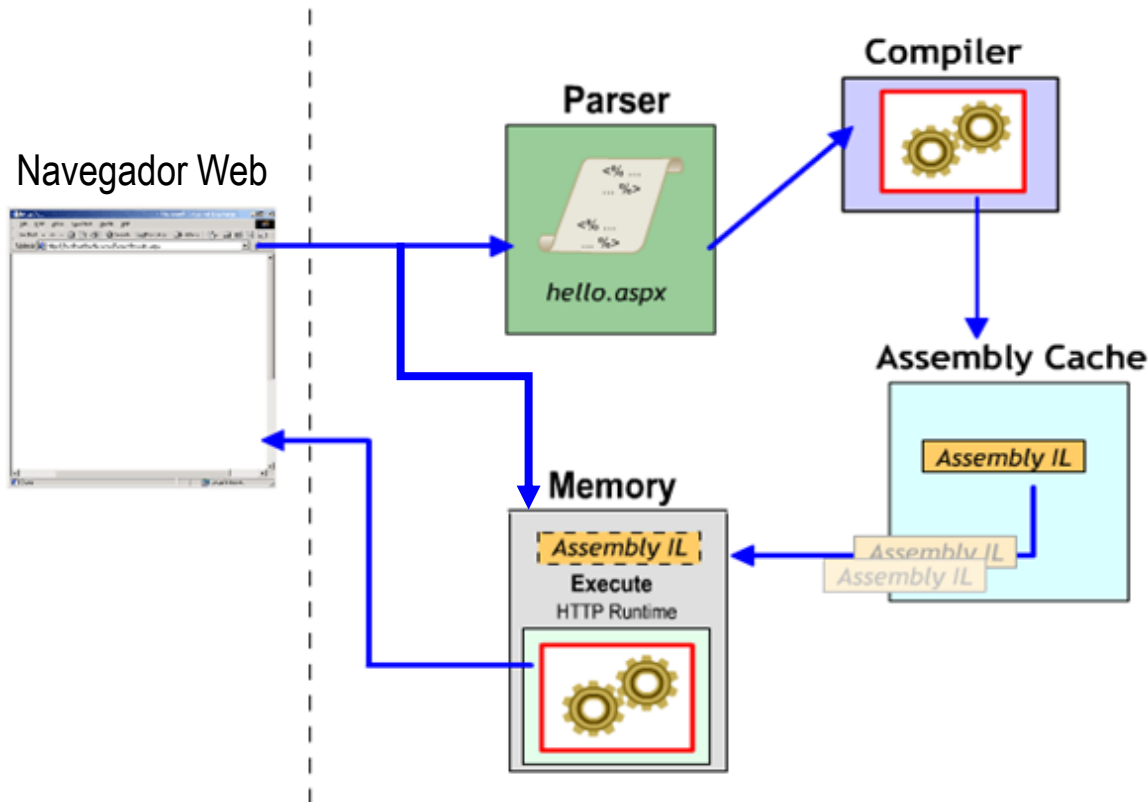
La tecnología de desarrollo ASP.NET

- **ASP.NET configura un modelo de desarrollo** unificado que incluye los servicios necesarios **para crear aplicaciones Web empresariales de altas prestaciones**
- Sus características más significativas son los siguientes:
 - Proporciona un modelo de programación orientado a objetos y manejado por eventos de aplicaciones Web basado en la plataforma .NET
 - Es independiente del lenguaje de programación empleado para expresar la lógica del negocio. Se pueden utilizar lenguajes compatibles con .NET: C#, Visual Basic, J#, etc.
 - Es independiente del navegador Web empleado como cliente
 - Es independiente de los dispositivos empleados: smartphone, tablet, PC, etc.
 - Garantiza la separación de la capa de presentación de la capa de lógica del negocio, a través de la utilización de los archivos de código subyacente (code-behind).
 - Utiliza todas las funcionalidades que proporciona la librería de clases .NET para el desarrollo las aplicaciones Web, como la librería ADO.NET para acceso a datos
 - Se trata de una tecnología de desarrollo compilada, no interpretada
 - Conserva el contenido de los formularios (Web Forms) después de que sean enviados al servidor. Esta funcionalidad recibe el nombre de mantenimiento del estado de la página

1.7 Aspectos básicos de ASP.NET



Proceso de compilación y ejecución



Las modificaciones de la aplicación Web se realizan en caliente sobre el entorno de producción, sin necesidad de detener el servidor Web

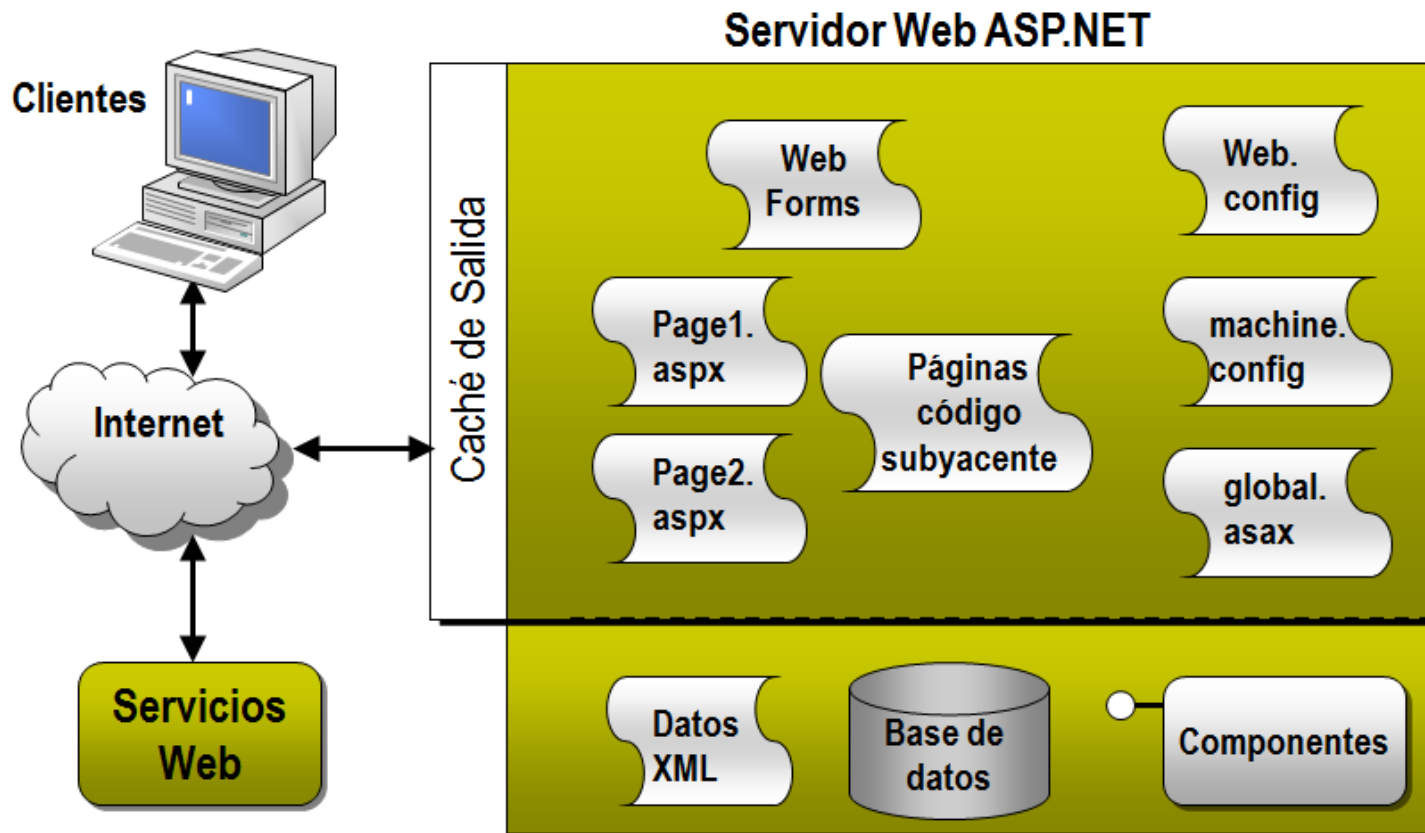
El servidor Web analiza el código de la página solicitada para comprobar si ha sido modificado respecto de la última compilación:

- Si el código fuente ha sido modificado, inicia el proceso de compilación, que incluye las llamadas al analizador (*parser*), al compilador (*compiler*) y al ensamblador (*assembly cache*) para generar el código administrado
- Si el código fuente no ha sido modificado desde la última compilación de la página, el entorno de ejecución CLR del servidor carga y ejecuta el código administrado. Después, envía la respuesta al cliente

1.7 Aspectos básicos de ASP.NET



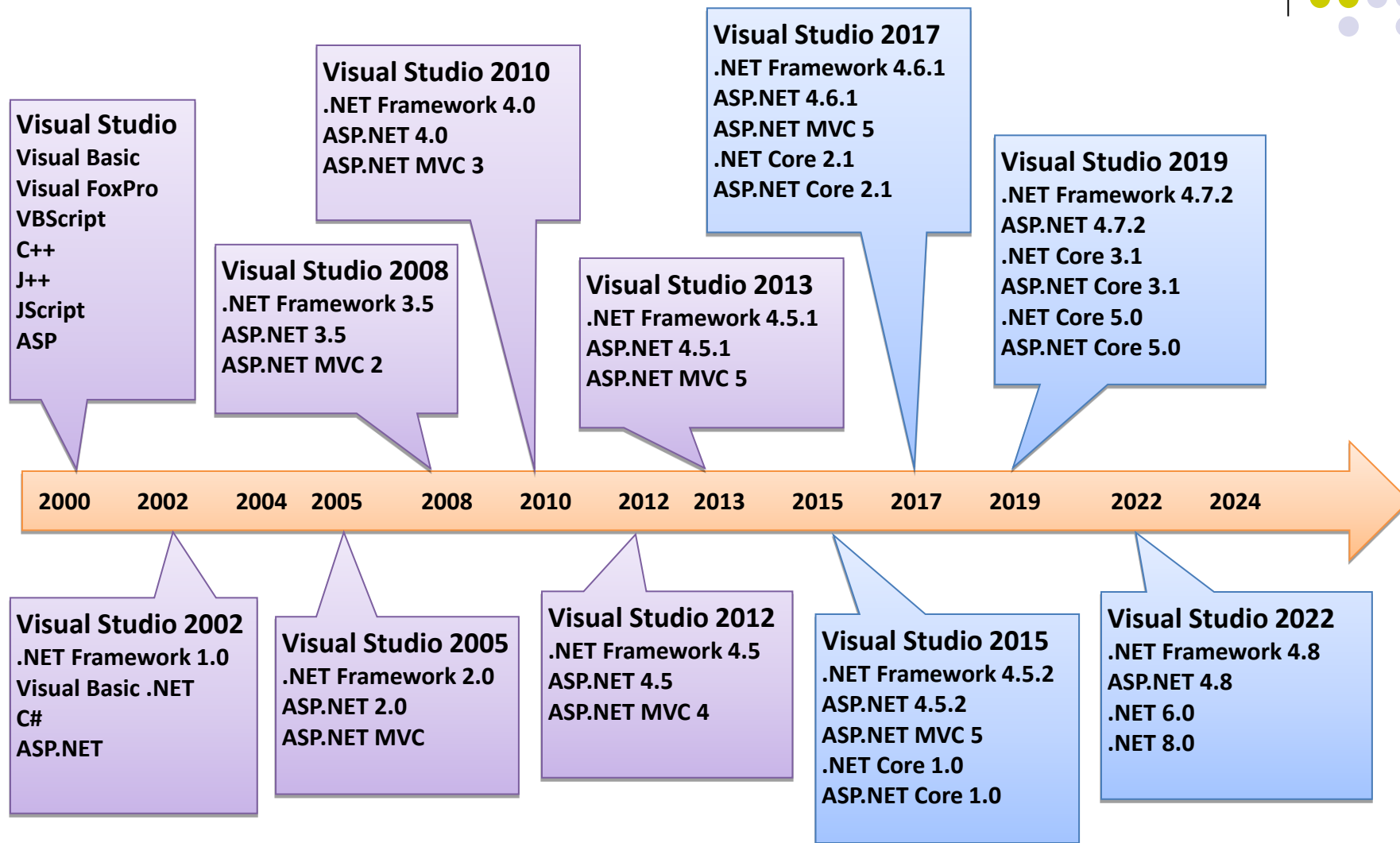
Elementos que forman una aplicación Web basada en ASP.NET



1.7 Aspectos básicos de ASP.NET



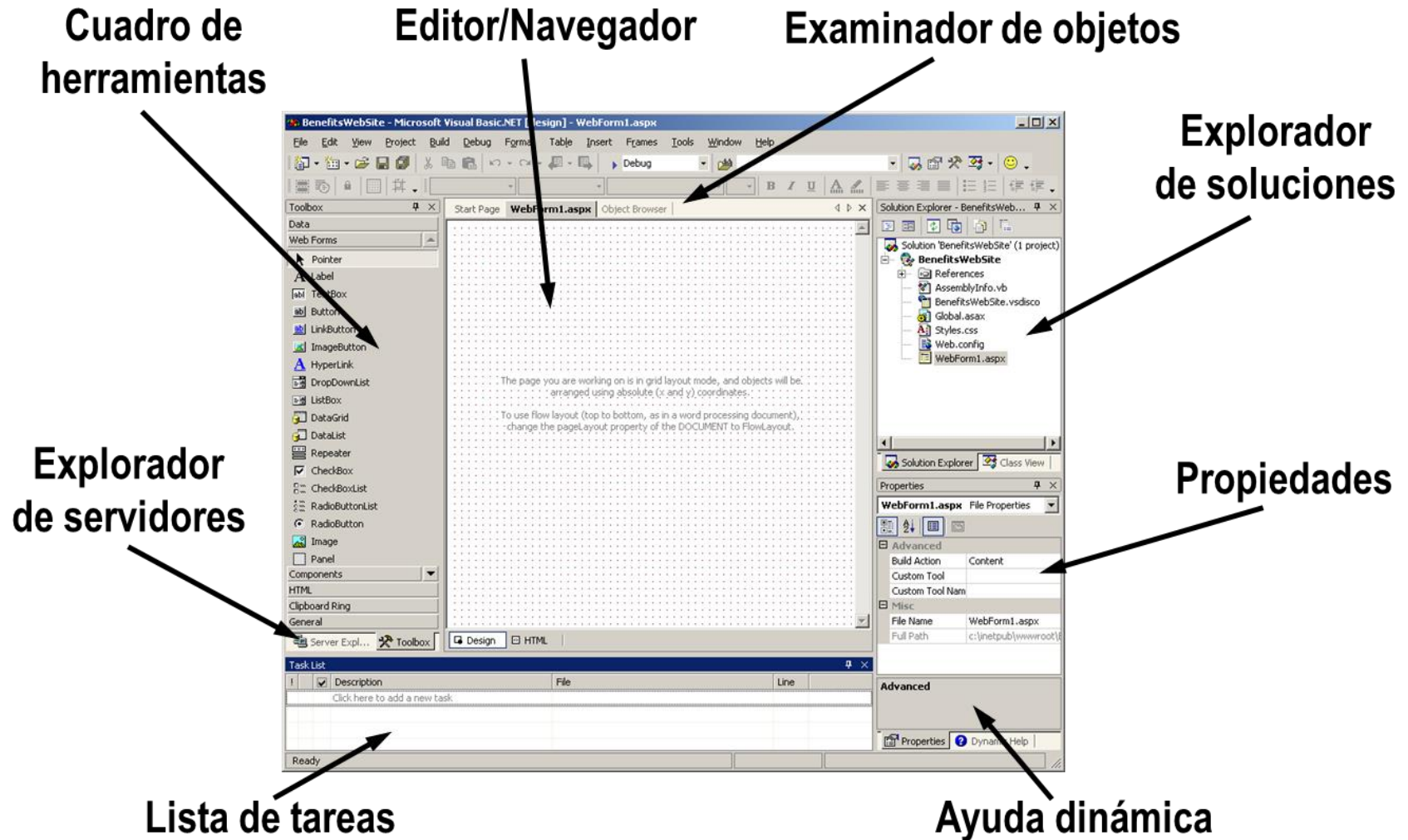
Línea de tiempo de la tecnología ASP.NET

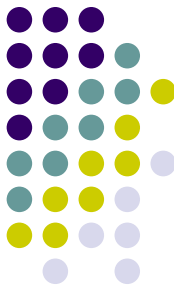


1.8 Uso de Microsoft Visual Studio



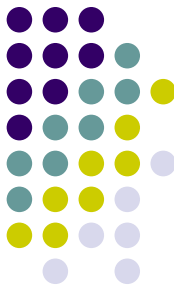
Entorno de Desarrollo Integrado (IDE)





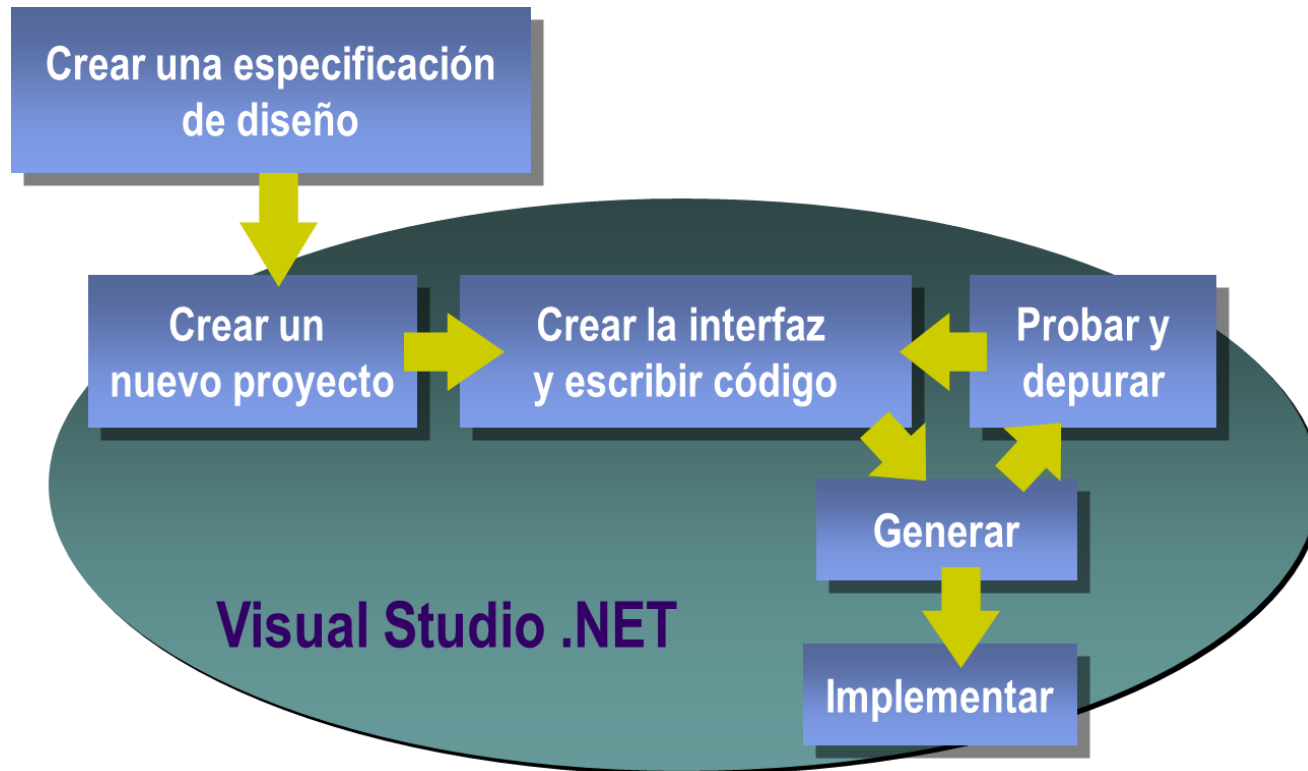
Generalidades y características

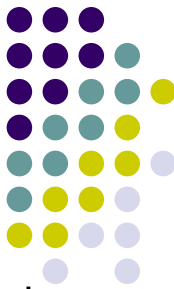
- Visual Studio es la herramienta de desarrollo de Microsoft empleada para el desarrollo de software basado en el modelo de desarrollo .NET
- Características principales en relación con el desarrollo de aplicaciones Web
 - Soporta múltiples tipos de proyectos de desarrollo de aplicaciones de escritorio, aplicaciones Web, Servicios Web y aplicaciones para dispositivos móviles
 - Pueden emplearse múltiples lenguajes de programación en un mismo proyecto
 - Incluye herramientas para gestionar y administrar el acceso a datos de las aplicaciones
 - Incorpora navegador Web, servidor Web de desarrollo y servidor de bases de datos SQL Server integrados y adaptados al entorno de desarrollo, lo que configura un entorno de desarrollo completo para el desarrollo de aplicaciones Web
 - Incorpora herramientas y utilidades que facilitan la compilación de los Proyectos de aplicación Web basadas en ASP.NET, así como el despliegue en el entorno de producción
 - Da soporte a la realización pruebas unitarias y conjuntas e incorpora herramientas de depuración
 - Dispone de editores visuales y de código para el desarrollo de aplicaciones Web



Fases del proceso de desarrollo de software

- Visual Studio incorpora herramientas y utilidades que facilitan las tareas a realizar por desarrollador durante todas las fases del proceso de desarrollo de software





Sistema de archivos de una aplicación Web de ASP.NET

- Las Soluciones y Proyectos de Visual Studio forman una estructura de contenedores que facilita la organización de los elementos que se están desarrollando
 - Una **Solución** es un contenedor que encapsula uno o varios Proyectoso
 - Un **Proyecto** es un contenedor que encapsula el conjunto de elementos que conforman la aplicación que se está desarrollando
- La estructura de archivos de una aplicación Web basada en ASP.NET de .NET Framework puede contener los siguientes elementos:
 - Archivos de Solución (.sln, .suo)
 - Archivos de Proyecto (.vbproj, .csproj)
 - Web Forms o páginas de ASP.NET (.aspx)
 - Archivos de código subyacente (aspx.cs, aspx.cs, etc.)
 - Archivos de definición de hojas de estilo en cascada (.ccs)
 - Archivo de configuración de la aplicación (Web.config)
 - Archivo de clases globales de aplicación (global.asax)
 - Archivo de ensamblado del proyecto (.dll)