

Hada T3:

Introducción a las aplicaciones web

Aplicaciones web.

Objetivos

- Conocer la **arquitectura cliente/servidor**
- Comprender los conceptos de programación en el cliente y en el servidor
- Ventajas e inconvenientes de la arquitectura cliente/servidor
- Conocer las diferentes **tecnologías web**
- Conocer la arquitectura de n-capas
- Conocer los **servidores web** más utilizados
- Aprender una **metodología** en el desarrollo de las páginas web

Arquitectura cliente / servidor

- **Arquitectura cliente/servidor**
 - Arquitectura de n-capas
 - Aplicaciones web
 - Servidores web
 - Metodología de diseño
 - Ejercicios
-

Arquitectura cliente/servidor

- La arquitectura **cliente/servidor** es un modelo de diseño software en que los **servidores** comparten recursos o servicios, y los **clientes** solicitan dichos recursos
- Es una arquitectura típica de desarrollo de aplicaciones
- El **cliente** solicita y el **servidor** responde
- Es más ventajoso para **aplicaciones distribuidas** a través de una red de ordenadores
- La **capacidad de proceso** está dividida entre **clientes** y **servidores**.
- Facilita el diseño del sistema

Arquitectura cliente/servidor

- Diferentes requisitos para el **cliente** y el **servidor** (velocidad de procesamiento, memoria, capacidad de almacenamiento de datos, etc.)
- Un **servidor** puede dar servicio a miles de **clientes**
- **Cliente**
 - El software de la parte **cliente** se conoce como *front-end*
 - Inicia la petición
 - Espera y recibe la respuesta del **servidor**
 - Se puede conectar a un grupo de **servidores** de manera simultánea
 - Suele ofrecer un interfaz gráfico al usuario final

Arquitectura cliente/servidor

- **Servidor**

- El software de la parte del **servidor** se conoce como *back-end*
- Pasivo
- Espera las peticiones de los **clientes**
- Cuando recibe una petición, la procesa y devuelve la respuesta
- Normalmente acepta un gran número de conexiones simultáneas de los **clientes**

Arquitectura cliente/servidor

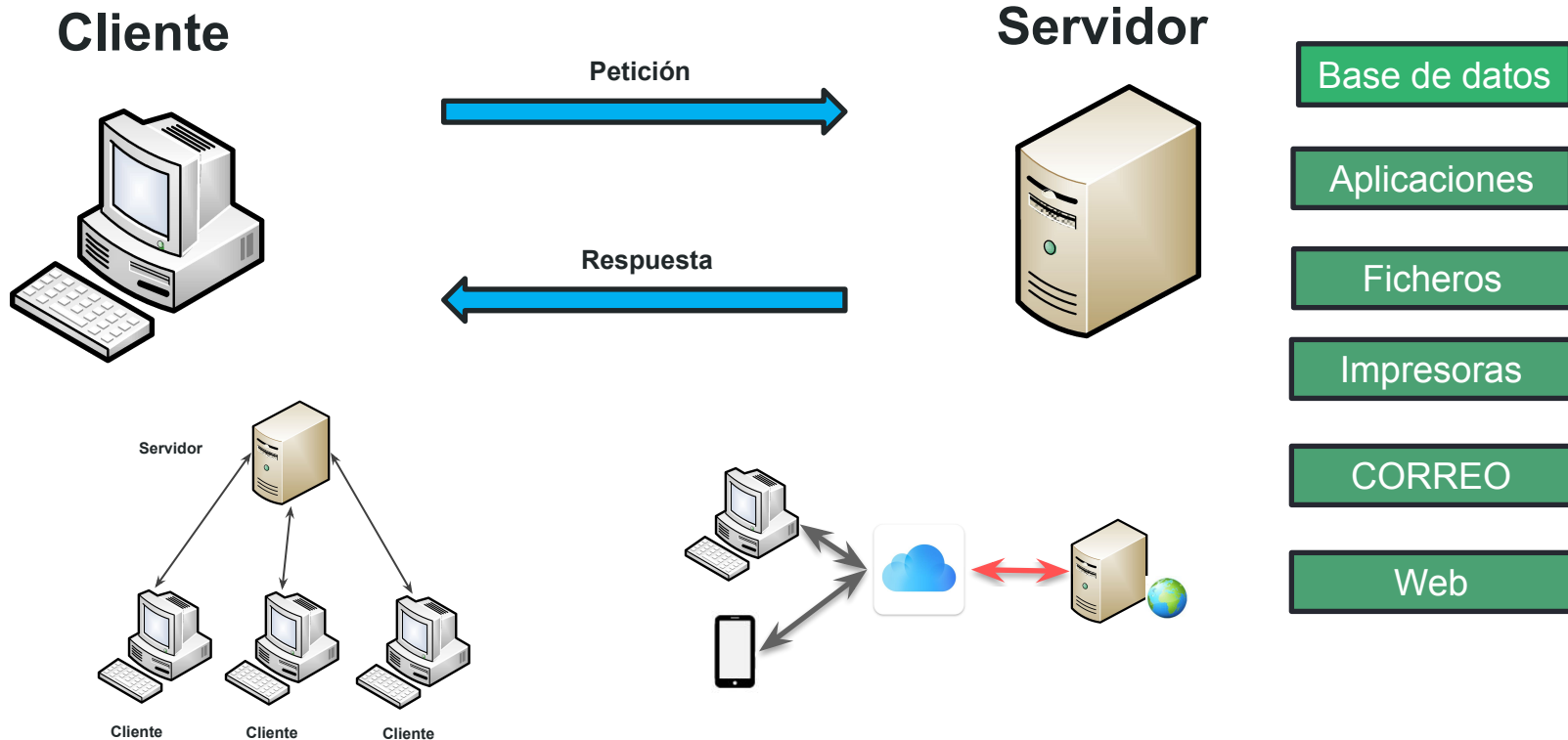
Ventajas

- + **Distribución de aplicaciones:**
conurrencia de procesos
- + **Estandarización.** Los recursos están publicados y otras aplicaciones pueden acceder a ellos
- + **Portabilidad**
- + **Escalabilidad**
- + Posibilidad de utilizar clientes ligeros con requisitos mínimos de instalación
- + Entornos heterogéneos. El hardware o el sistema operativo puede ser diferente

Desventajas

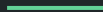
- Aumenta la comunicación.
Congestión tráfico de red
- **Falta de robustez.** Caídas del servidor

Arquitectura cliente/servidor



Arquitectura de n-capas

- Arquitectura cliente/servidor
- **Arquitectura de n-capas**
- Aplicaciones web
- Servidores web
- Metodología de diseño
- Ejercicios



Arquitectura de n-capas

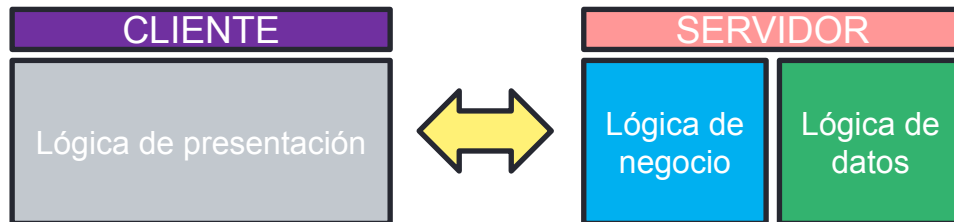
- **División de los componentes** de una aplicación en n niveles o capas lógicas.
- Pueden ser de 1 (aplicación individual), 2 o 3 o n capas.
- No implica separación física en distintos ordenadores de la red (una aplicación de 3 capas puede existir en un único ordenador).
- **Ventajas**
 - Permitir modificar una capa sin tener que modificar toda la aplicación.
 - Simplificación de la administración de los sistemas
 - Disponibilidad inmediata de cambios
 - Posibilidad de balanceo de carga de trabajo entre ordenadores

Arquitectura de n-capas

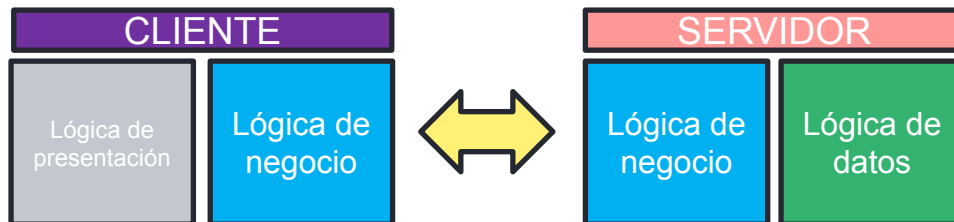
- Una arquitectura de 3 capas está dividida en:
 - **Interfaz de usuario (o lógica de presentación)**, componentes que interactúan con el usuario final. Interfaz gráfico (GUI) o basado en texto.
 - **Lógica de negocio**, contienen las reglas de negocio (funcionalidades) de nuestra aplicación. Corresponde con el núcleo de la aplicación.
 - **Persistencia (o lógica) de datos**, permite el acceso y almacenamiento de los datos

Modelos de distribución por capas

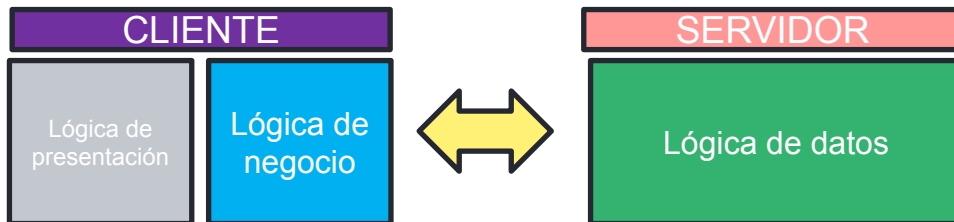
- Presentación distribuida
(una interfaz con validación de datos)



- Aplicación distribuida
(máxima flexibilidad)

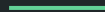


- Datos distribuidos
(máxima carga en cliente y mayor ancho de banda)



Aplicaciones web

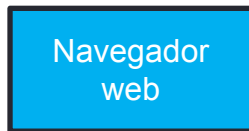
- Arquitectura cliente/servidor
- Arquitectura de n-capas
- **Aplicaciones web**
- Servidores web
- Metodología de diseño
- Ejercicios



Introducción

- Una aplicación web es una aplicación informática que se ejecuta en el entorno web
- Basada en la arquitectura **cliente/servidor**.

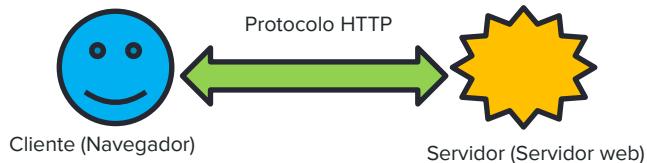
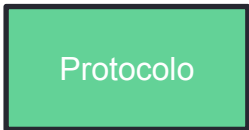
- **Cliente**



- **Servidor**



- **Comunicación**



Introducción



Otros servicios

Base de datos

Aplicaciones

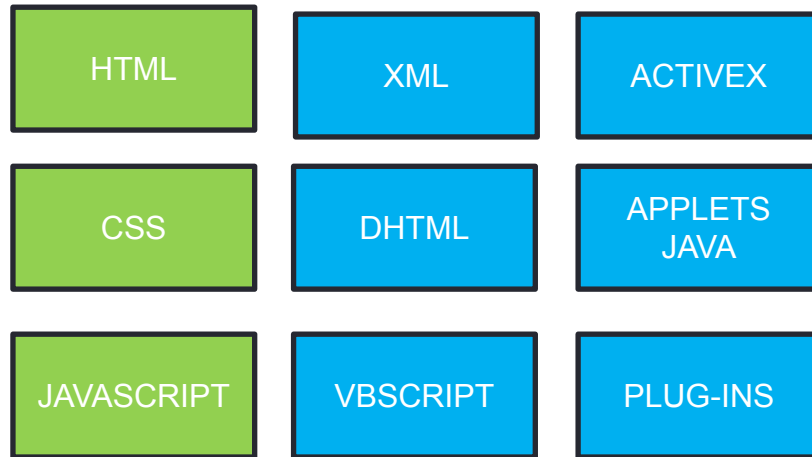
Ficheros

Impresoras

CORREO

Cliente

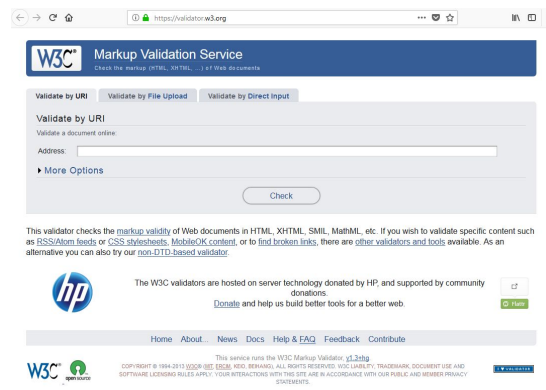
- Se encarga de gestionar las peticiones del usuario y la recepción del contenido de las páginas que provienen del servidor web
- Es capaz de interpretar los documentos HTML y otros recursos
- Tecnologías:



Tecnologías en cliente



- **HyperText Markup Language** (lenguaje de marcas de hipertexto)
- Lenguaje de programación para definir el contenido de una página web
- Versión actual: HTML 5.2
- **Define el formato del texto, posición, colores, tamaños**
- **Enlaces:** Permite añadir elementos externos a partir de referencias a la ubicación de dicho elementos (no se incrusta directamente en el código)
- Algunas **etiquetas:** <html>, <script>, <head>, <div>, , , , etc.
- Editores: Notepad, Atom, jedit, Visual Studio Code, Visual Studio, etc.
- W3C validator: <http://validator.w3.org>
 - Revisa el código fuente
 - Busca errores de sintaxis
 - Devuelve el resultado de la prueba de validación



Tecnologías en cliente

CSS

- **Cascading Style Sheet** (hojas de estilo en cascada)
- Es un lenguaje de diseño gráfico que **define la presentación** de un documento estructurado en un lenguaje de marcado como HTML
- Versión actual: CSS3
- **El estilo define la manera de mostrar los elementos HTML**
- El estilo se puede incluir en la misma página o en un fichero externo
- Permite establecer una separación entre la presentación (estilo) y el contenido de una página web
- **Ventajas:**
 - Posibilidad de mayor control en la presentación
 - Diferentes presentaciones para un mismo contenido
 - Menor carga de las páginas web
 - Programación independiente

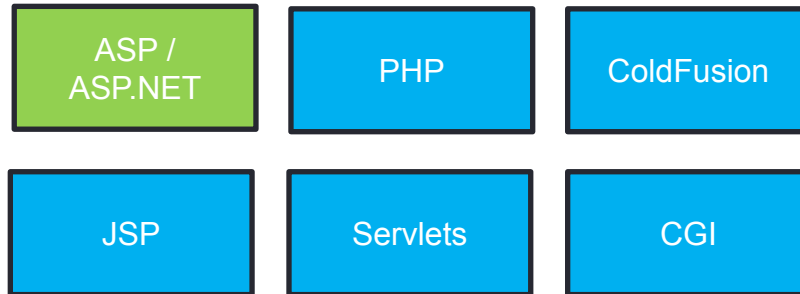
Tecnologías en cliente

JAVASCRIPT

- **Javascript** es un lenguaje de programación que está embebido dentro de una página web, para crear contenido e interactividad dinámica
- Lenguaje más estándar para **script** (más compatible)
- Durante muchos años fue el único método para incorporar dinamismo a las páginas web. En la actualidad hay otras tecnologías
- Sintaxis parecida a JAVA y C++
- Case sensitive (sensible a minúsculas y a mayúsculas)
- Incorpora el *Document Object model (DOM)*, que se utiliza para interactuar con la página web
- **Ejemplos de uso:** validar los datos de entrada de un usuario (comprobar datos correctos), proporcionar dinamismo en respuesta a las acciones del usuario (desplegar un menú al pasar el ratón), etc.

Servidor

- Software que espera las peticiones de los clientes
- En la aplicación del servidor habrá:
 - Conjunto de páginas estáticas (HTML)
 - Recursos multimedia (imágenes, documentos, videos , etc.)
 - Programas que se ejecutan y proporcionan páginas web dinámicas
- Tecnologías:



Tecnologías en servidor

The PHP logo is a blue square with a black border, containing the letters 'PHP' in white, bold, sans-serif font.

- **Hipertext Preprocessor** es un lenguaje de Script, de **programación en el servidor** para desarrollo web **de contenido dinámico**
- Se suele combinar con el sistema gestor de base de datos MySQL
- El código de PHP se puede mezclar con el código HTML de la página web
- El código escrito en PHP es invisible al cliente, ya que es el servidor quien se encarga de ejecutar el código, y enviar el resultado HTML al cliente (navegador web)
- Es libre (licencia GPL), multiplataforma y tiene gran velocidad de respuesta
- Funciona con distintos servidores web: Apache, Nginx
- Manejo de excepciones (desde PHP5)
- Muy popular el desarrollo con la herramienta XAMPP (Apache, MariaDB, PHP y Perl)

Tecnologías en servidor

ASP

- **Active Server Pages (ASP):** Lenguaje de Script, con un intérprete (no compilado)
 - Es una tecnología de Microsoft, en el lado del servidor, para crear páginas web dinámicas
 - Utiliza el servidor IIS (Internet Information Server)
 - El código de ASP se puede mezclar con el código HTML de la página web
 - El código ASP se ejecuta en el servidor, y produce en su salida un código HTML (compatibilidad con navegadores)

Tecnologías en servidor

ASP.NET

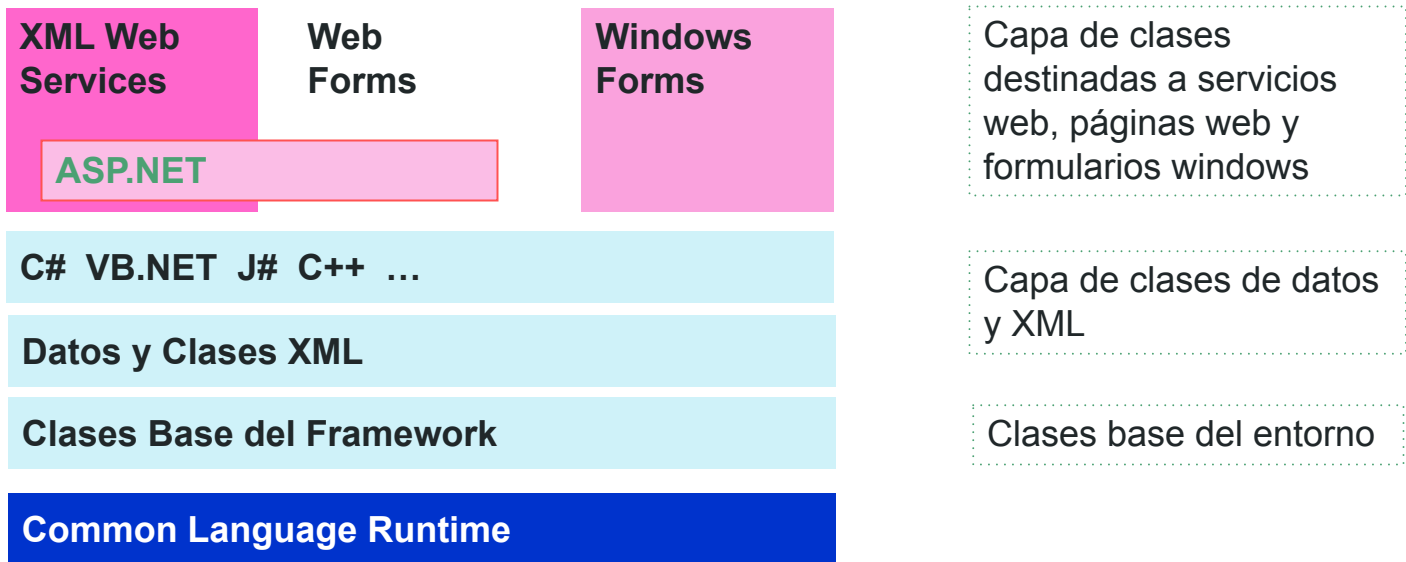
- **ASP. NET:** plataforma para construir aplicaciones Web.
- Forma parte del .NET Framework y contiene un conjunto de librerías
- Soporte de múltiples lenguajes: *Orientados a Objetos, compilados y dirigidos por Eventos. Utilizaremos C#*
- Para desarrollar aplicaciones ASP.NET necesitamos:
 - Editor o entorno de desarrollo (Microsoft Visual Studio)
 - **.NET Framework instalado**
 - Un servidor web con IIS



Tecnologías en servidor

ASP.NET

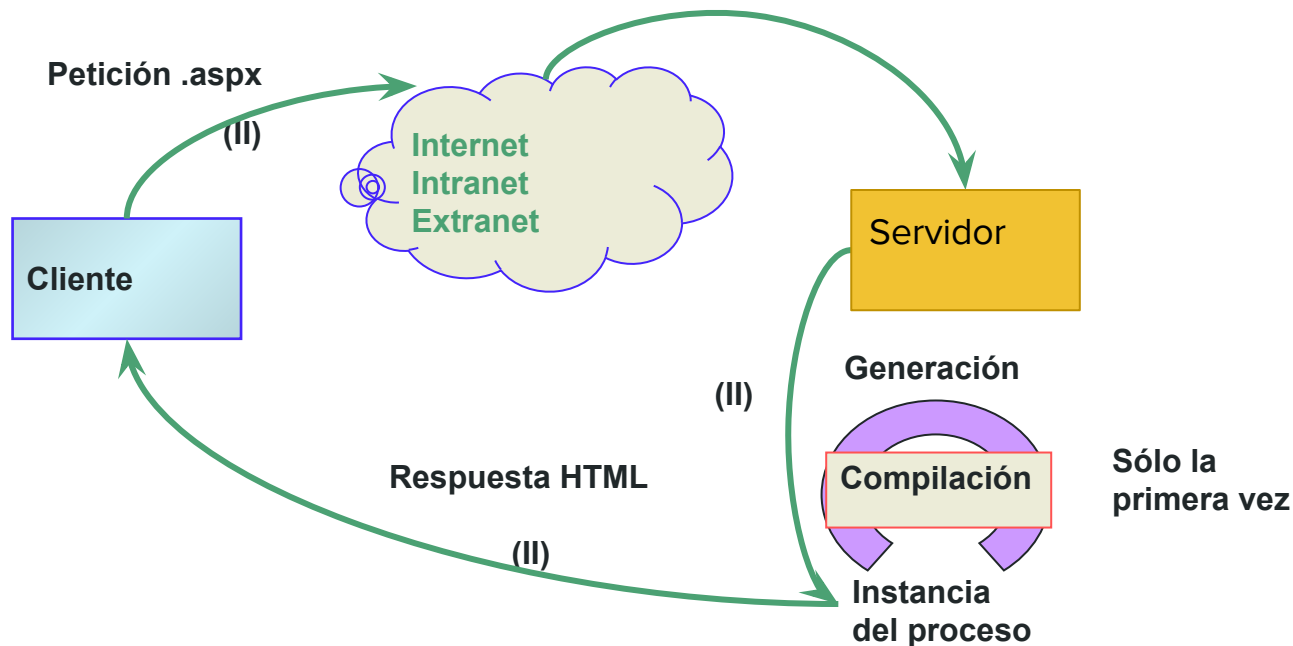
Framework de Microsoft .NET



Tecnologías en servidor

ASP.NET

Llamada a una página ASP .NET



Servidores web

- Arquitectura cliente/servidor
 - Arquitectura de n-capas
 - Aplicaciones web
 - **Servidores web**
 - Metodología de diseño
 - Ejercicios
-

Internet Information Server (IIS)



- IIS es un servidor web
- IIS es un conjunto de servicios de Microsoft
- IIS es una plataforma web unificada que integra:
 - el servidor web (HTTP/HTTPS),
 - ASP.NET, y
 - otros servicios como FTP, SMTP, NNTP
- También puede incluir PHP o Perl

Internet Information Server (IIS)



Ventajas

- + Ejecución de aplicaciones web en ASP, ASP.NET, y PHP en un mismo servidor
- + Confiable, seguro y fácil de utilizar
- + Posibilidad de agregar o eliminar componentes IIS integrados e incluso reemplazarlos por módulos personalizados
- + Aumenta la velocidad del sitio web mediante el almacenamiento en caché dinámico integrado y mejora de la compresión
- + Soporte técnico por parte de Microsoft

Desventajas

- No se recomienda su uso para desplegar aplicaciones en PHP, Python, Perl o Ruby (que se ejecutan de forma óptima en Linux y UNIX)
- Su licencia no es gratuita
- No es multiplataforma (sólo Windows)
- Código fuente propietario

APACHE

- Es un servidor web de código abierto multiplataforma
- Es el más utilizado en el mercado del hosting



Ventajas

- + Software de código abierto (instalación y configuración adaptable mediante módulos)
- + Sin coste de licencia
- + Buen soporte debido a que es el más utilizado por lo que muchos programadores contribuyen con mejoras
- + Multiplataforma (Windows, Linux y MacOS)
- + Buena interacción con PHP y MYSQL

Desventajas

- No posee interfaz gráfica
- No contiene una configuración estándar. Puede ser adaptada a cada aplicación
- No hay soporte técnico
- Difícil de administrar y configurar
- Pocas actualizaciones

Metodología de diseño de aplicaciones web

- Arquitectura cliente/servidor
- Arquitectura de n-capas
- Aplicaciones web
- Servidores web
- **Metodología de diseño**
- Ejercicios



Metodología de diseño

1 Análisis de requisitos

Es un punto muy importante porque hay que analizar todo lo que nos han pedido y los objetivos

2 Elección de las tecnologías web a usar

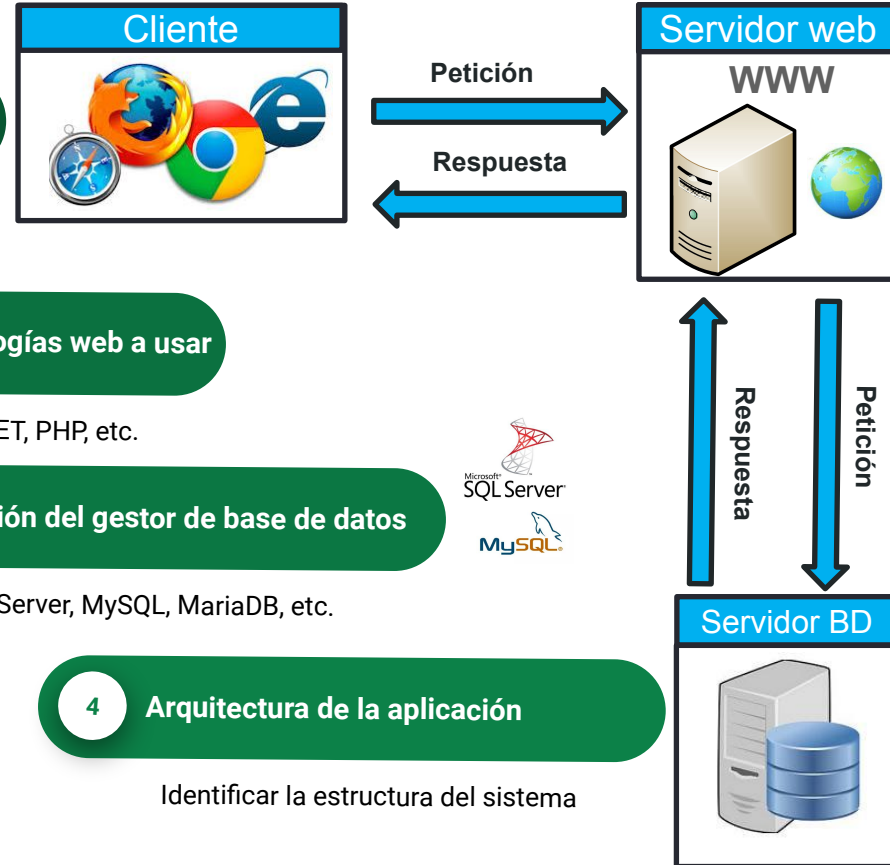
HTML, CSS, ASP, ASP.NET, PHP, etc.

3 Elección del gestor de base de datos

SQL Server, MySQL, MariaDB, etc.

4 Arquitectura de la aplicación

Identificar la estructura del sistema



- Es recomendable seguir una metodología en el diseño de una aplicación web.
- Existen diferentes metodologías.
- Nosotros proponemos la siguiente metodología:

Metodología de diseño

5

Diseño de estructura y mapa de navegación web

Se definen las páginas web que se van a utilizar y sus navegación en el sitio web



Páginas de contenido secundario



6

Creación del contenido de las páginas

Se implementan las páginas estáticas de la web



Páginas de contenido secundario



7

Diseño gráfico

Se añade la maquetación mediante CSS



8

Generación de scripts para controles en el cliente

Se implementa el código Javascript



JavaScript

Metodología de diseño

9

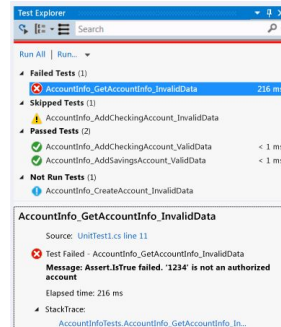
Diseño y desarrollo de páginas dinámicas

ASP /
ASP.NET

PHP

10

Validación y pruebas



Pruebas
funcionales

Pruebas de
Integración

Pruebas
Unitarias

11

Despliegue de la solución



Ejercicios

- Arquitectura cliente/servidor
 - Arquitectura de n-capas
 - Aplicaciones web
 - Servidores web
 - Metodología de diseño
 - **Ejercicios**
-

Ejercicios

1. ¿Qué son las páginas con extensión .htm? ¿Es lo mismo .html que .htm? Razona la respuesta.
2. ¿Cuál es el lenguaje estándar para aplicar estilos de presentación a nuestras páginas web?
3. ¿En qué lugar se ejecuta el código JavaScript?
4. ¿Es mejor utilizar PHP o ASP? ¿Por qué?