

PCW

PROGRAMACIÓN DEL CLIENTE WEB

Tema 01 - Introducción



Dept. de Ciència de la Computació i Intel·ligència *a*rtificial
Dpto. de Ciencia de la Computación e Inteligencia *a*rtificial



Universitat d'Alacant
Universidad de Alicante

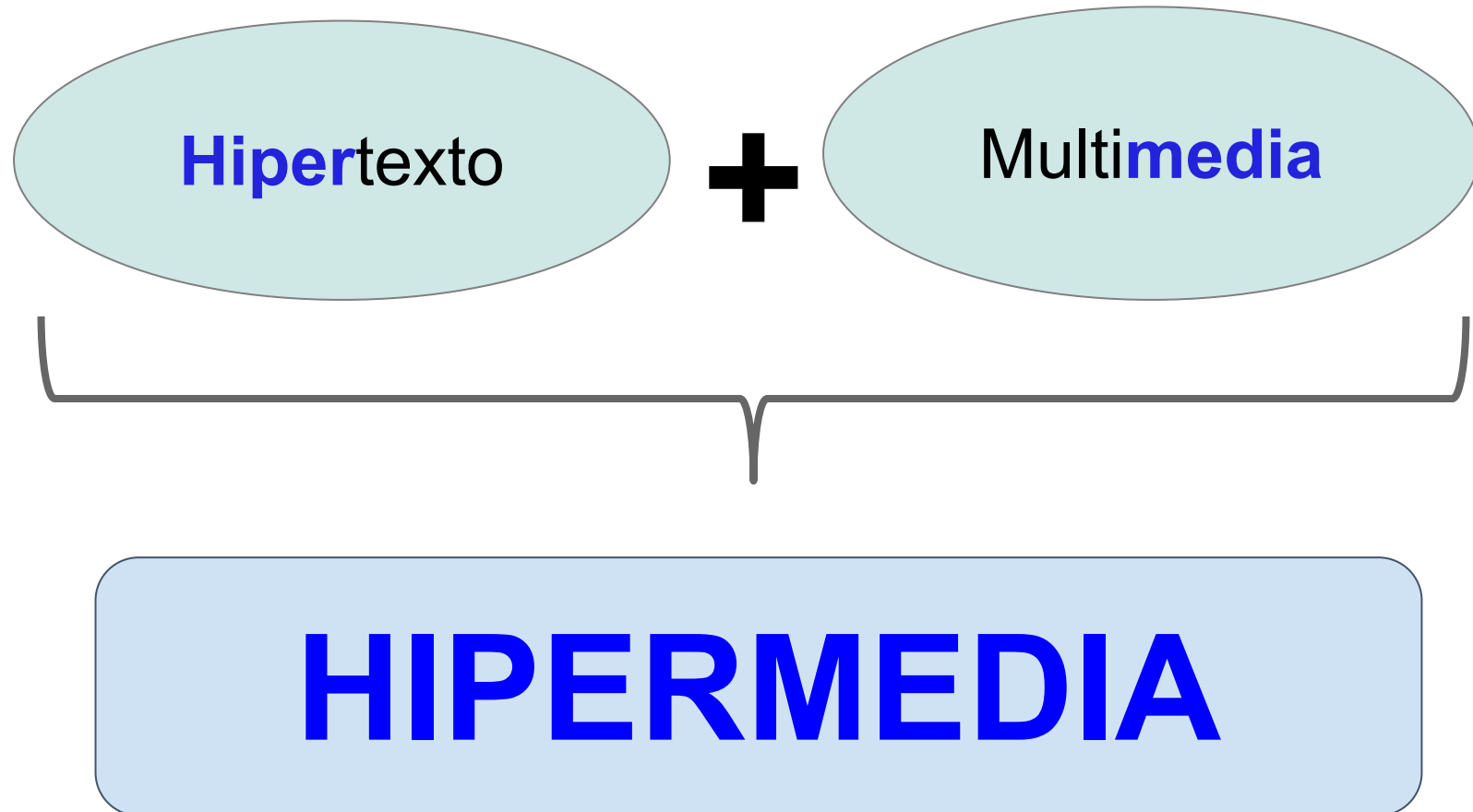
Tema 1. Introducción

- 1.1. Concepto de Hipermedia
- 1.2. Arquitectura cliente-servidor en una aplicación web
- 1.3. Ventajas y desventajas de una aplicación web
- 1.4. Características de una aplicación web
- 1.5. Aspectos importantes del diseño de una aplicación web
- 1.6. Desarrollo Frontend vs Backend

1.1

Concepto de Hipermedia

1.1 Concepto de Hipermedia



1.1 Concepto de Hipermedia (y II)

HIPERTEXTO

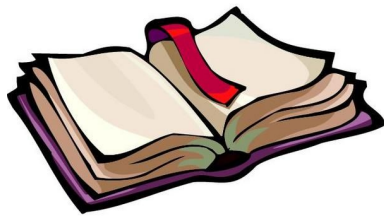
“Consiste en piezas de texto enlazadas entre sí de manera no secuencial”

Jakob Nielsen, 1990

1.1 Concepto de Hipermmedia (y III)

HIPERTEXTO (y II)

- Texto normal:

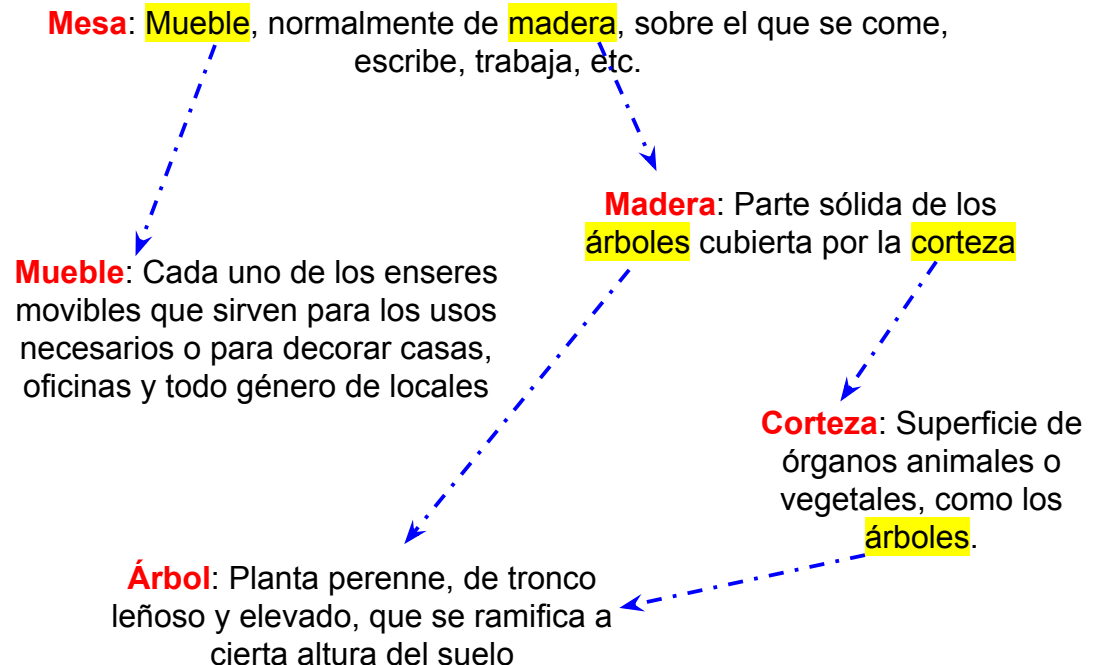


Libro



Texto secuencial

- Hipertexto:



1.1 Concepto de Hipermedia (y IV)

MULTIMEDIA

Información
consumida, sea cual
sea el formato:
texto, audio, vídeo,
fotografía, etc.

+

Fuentes que sirven
la información que
se consume

1.1 Concepto de Hipermedia (y V)

HIPERMEDIA

“Conjunto de fuentes y contenidos multimedia, interconectados entre sí, que permiten al usuario su exploración de una forma no secuencial, así como también le ofrecen la posibilidad de interactuar con ellos.”

1.1 Concepto de Hypermedia (y VI)

HIPERMEDIA (y II)

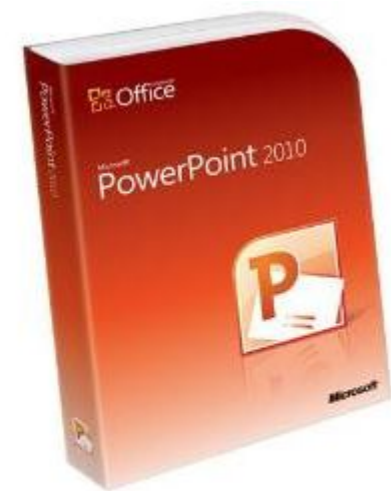
Ejemplos de sistemas hipermedia



World Wide Web



Película de DVD



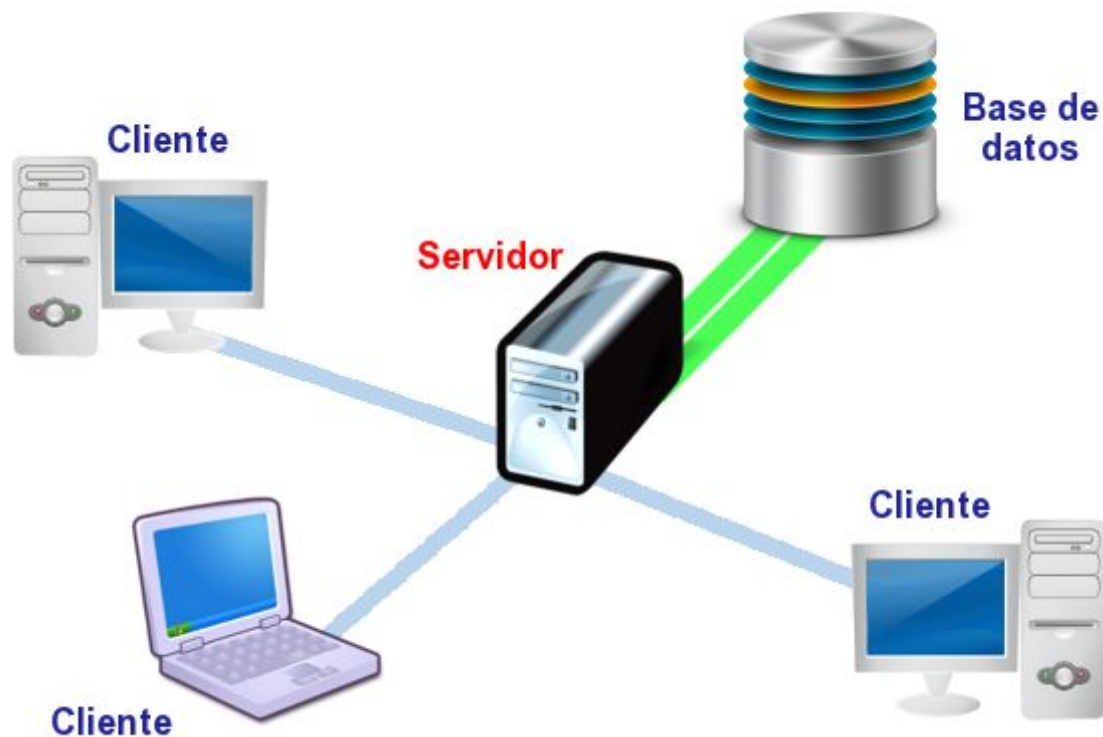
Presentación de Power Point

1.2

Arquitectura cliente-servidor en una aplicación web

1.2 Arquitectura cliente-servidor en una aplicación web

Arquitectura cliente-servidor en red local

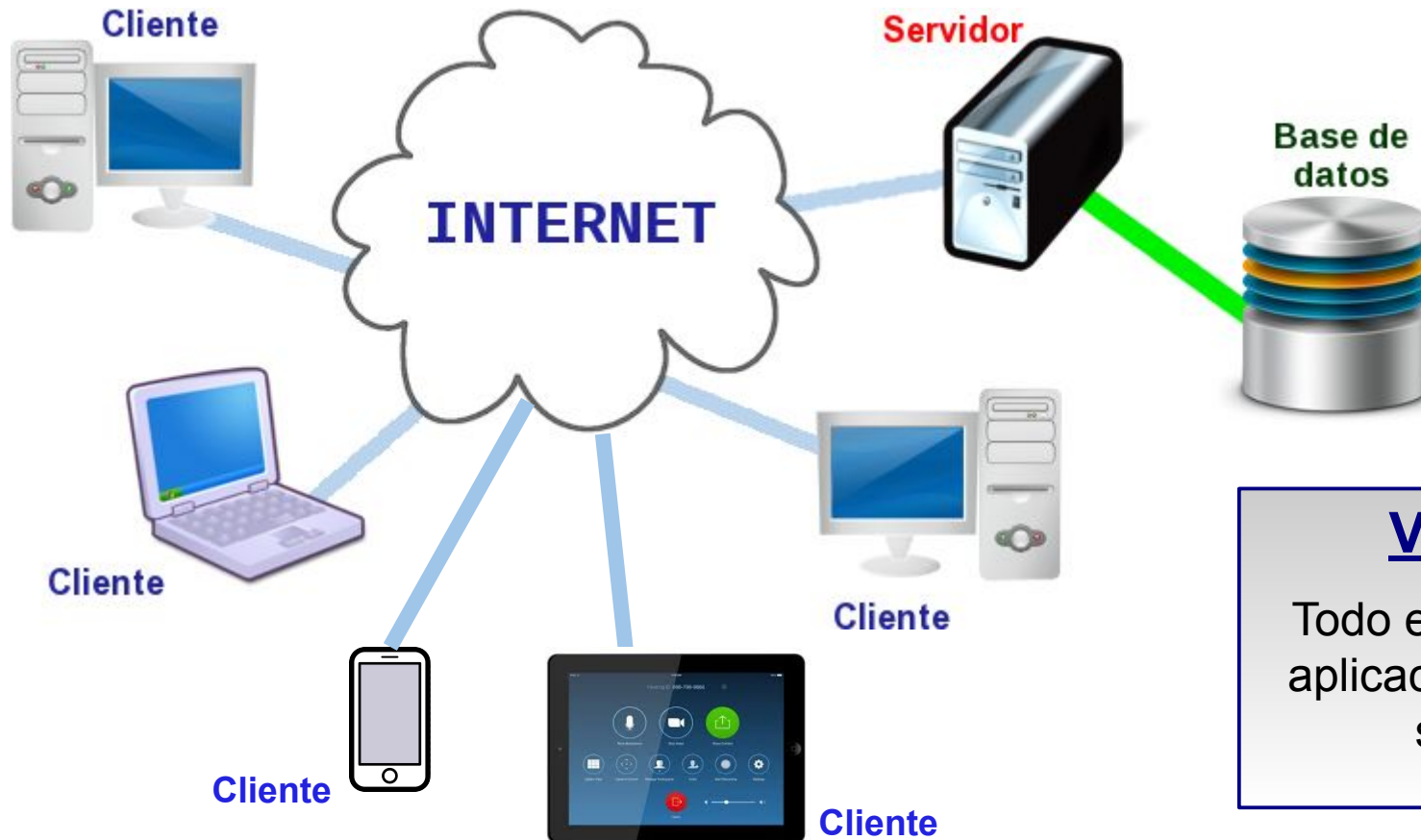


Desventajas:

- Es necesario instalar la aplicación cliente en todas las máquinas cliente.
- Cualquier cambio en la aplicación cliente implica una reinstalación de la misma en todas las máquinas cliente.

1.2 Arquitectura cliente-servidor en una aplicación web (y II)

Arquitectura cliente-servidor en la Web



Ventaja:

Todo el código de la aplicación está en el servidor

1.2 Arquitectura cliente-servidor en una aplicación web (y III)

El cliente de una aplicación web

- Recibe órdenes del servidor, fundamentalmente, en forma de código HTML, que interpreta para mostrar el resultado: una página web estática.
- Se puede aportar dinamismo mediante **lenguajes interpretados** en el lado del cliente de dos maneras:
 - Interpretado *directamente* por el navegador: JavaScript
 - Interpretado *indirectamente*, mediante *plugins* instalados en el navegador: flash, applets java.

1.2 Arquitectura cliente-servidor en una aplicación web (y IV)

Lenguaje interpretado en el cliente

Ventajas

- Es independiente del servidor.
- Libera de trabajo al servidor.
- Facilita la interactividad con el usuario.
- Un mismo código funciona igual en cualquier plataforma: Windows, Linux, Mac, etc.

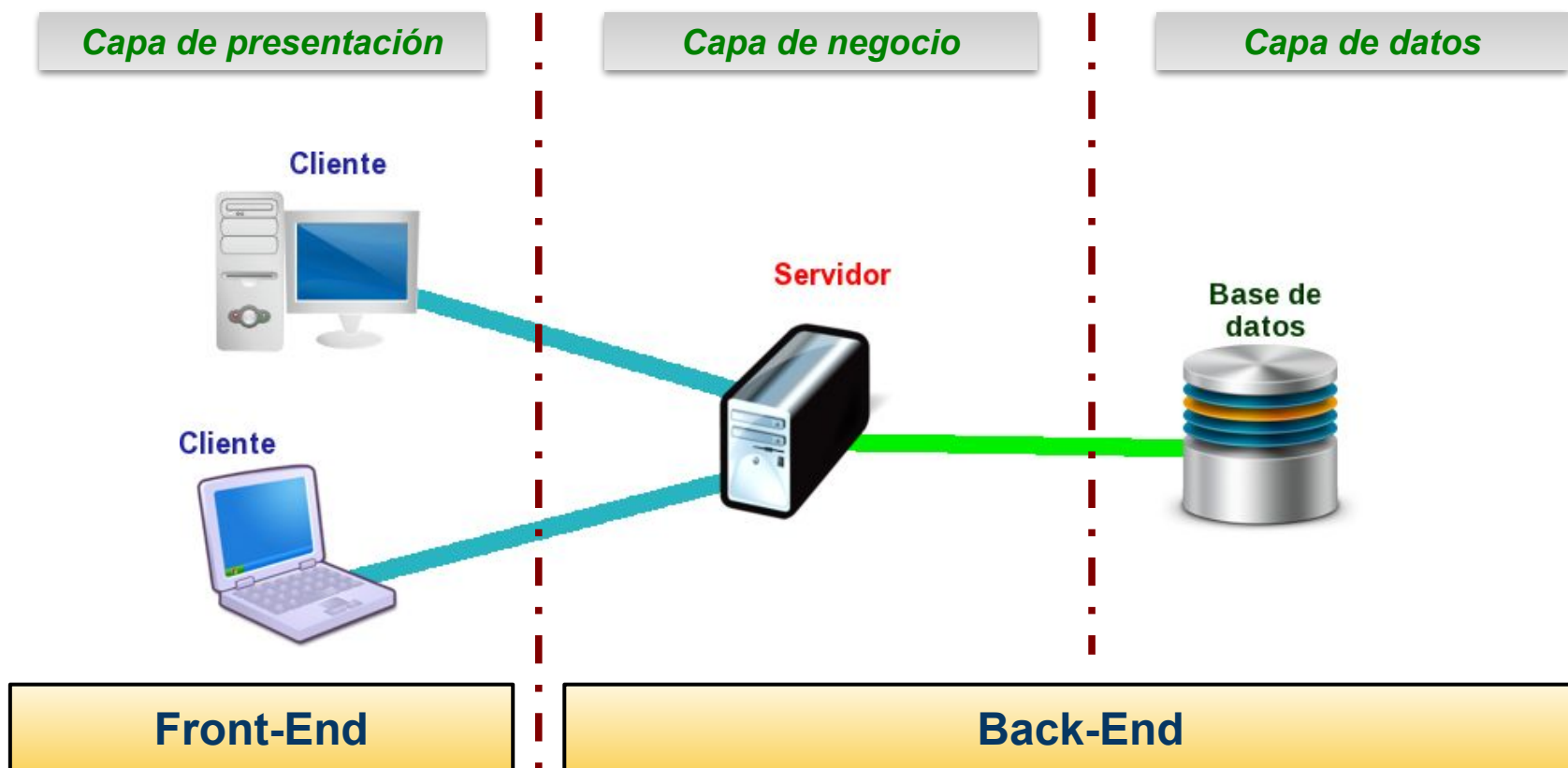
Desventajas

- Tienen capacidades limitadas por razones de seguridad.
- El código es visible por cualquier usuario.
- El código debe descargarse completamente para funcionar.

1.2 Arquitectura cliente-servidor en una aplicación web (y V)

Estructura de una aplicación web

Arquitectura en tres capas:



1.3

Ventajas y desventajas de una aplicación web

1.3 Ventajas y desventajas de una aplicación web

Ventajas

- Coste
- Compatibilidad
- Actualizaciones rápidas
- Bajo consumo de recursos
- Fácil acceso
- Evolución de las tecnologías

Desventajas

- Menos potentes y rápidas que las de escritorio
- Dependencia de internet
- Requieren clientes (navegadores) compatibles o, en su defecto, la instalación de plugins
- Seguridad

1.4

Características de una aplicación web

1.4 Características de las aplicaciones web

- Diseño orientado a dispositivo móvil (*Mobile-First design*)
- Proporcionan una gran experiencia de usuario (UX)
- Basada en la nube, escalable y fácilmente actualizable
- Modular y descentralizada/distribuida
- Compatible con distintas plataformas
- Combinan el comportamiento de una aplicación web tradicional y el de webs SPA (*Single Page Application*)

1.5

Aspectos importantes en el diseño de una aplicación web

1.5 Aspectos importantes en el diseño de una aplicación web

- 1) **Minimizar el contenido superfluo** de la página web todo lo posible, incluido el número de opciones y controles disponibles para el usuario, para ayudarlo a completar la tarea.
- 2) **Navegación clara e intuitiva**, que permita al usuario encontrar la información buscada de forma rápida.
- 3) Diseñar la **navegación en torno a la parte principal** de la aplicación, eliminando los típicos elementos de navegación web (botón *Home*, etc), pero manteniendo los **elementos principales accesibles** todo el tiempo.
- 4) Proporcionar **una experiencia y un diseño consistente**, el usuario siempre encontrará las mismas opciones en el mismo sitio y con la misma apariencia, sin importar la parte de la aplicación en la que se encuentre.
- 5) **Utilizar las capacidades del dispositivo** siempre que sea posible.

1.6

Desarrollo Frontend vs Backend

1.6 Desarrollo Frontend vs Backend

¿Qué es el desarrollo Frontend?

- El desarrollo **Frontend** es el realizado para crear la **parte cliente** de una aplicación o sitio web.
- Consiste en crear la **interfaz de usuario** de sitios o aplicaciones web para que los usuarios puedan ver su contenido e interactuar con él.
- Las tecnologías web utilizadas son:
 - Lenguaje de marcas: **HMTL**
 - Lenguaje de presentación: **CSS**
 - Modelo de objetos del documento: **DOM**
 - Lenguaje de programación: **JavaScript**

1.6 Desarrollo Frontend vs Backend

¿Qué es el desarrollo Backend?

- El desarrollo **Backend** es realizado para crear la **parte servidor** de una aplicación o sitio web.
- Consiste en **producir el código de la capa de negocio y la capa de datos**. Es decir, la parte de la aplicación que se alojará en el servidor y que, por lo general, se encargará de acceder a una base de datos en la que guardará, y de la que extraerá, información.
- Por tanto, **requiere conocimientos de bases de datos** (capa de datos): SQL Server, MySQL, MongoDB, etc.
- Las tecnologías web utilizadas para la capa de negocio son más numerosas que las del lado del cliente: **Java, PHP, Python, .Net, JavaScript (Node.js)**, etc.