

Contenido

1. Protocolos de comunicaciones.....	1
2. Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web.	1
3. Estudio sobre los métodos de petición HTTP /HTTPS más utilizados.	1
4. Estudio sobre el concepto de URI, estructura, utilidad y relación con el protocolo HTTP/HTTPS.....	1
5. Modelo de desarrollo de aplicaciones multicapa.	2
6. Modelo de división funcional front end y back end para aplicaciones web.	2
7. Página web estática, dinámica, aplicación web y mashup.	3
8. Componentes de una aplicación web.	3
9. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor.....	3
10. Lenguajes de programación utilizados en el lado servidor de una aplicación web.	4
11. Características y posibilidades de desarrollo de una plataforma XAMPP.....	4
12. IDE más utilizados.....	4
13. Servidores HTTP /HTTPS más utilizados.....	5
14. Apache HTTP vs Apache Tomcat	5
15. Navegadores HTTP /HTTPS más utilizados.....	6

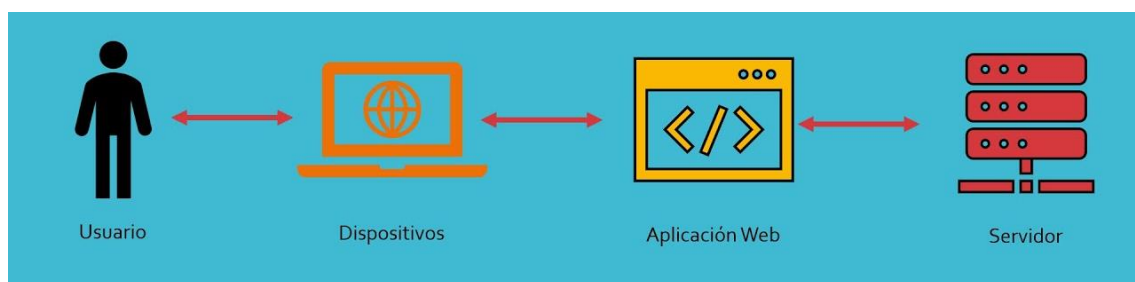
1. Protocolos de comunicaciones.

Conjunto de normas que establecen la forma de identificación, transmisión de datos y procesar datos entre ordenadores.

- **IP** protocolo cuya función es de origen y destino para transmitir datos.
- **TCP** protocolo que crea conexiones, distinguiendo aplicaciones a través del puerto.
- **HTTP** protocolo de transferencia de hipertexto a través de archivos.
- **HTTPS** protocolo basado en el HTTP destinado a la transferencia segura de hipertexto.

2. Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web.

Los clientes acceden a las aplicaciones web mediante un servidor y un navegador



3. Estudio sobre los métodos de petición HTTP /HTTPS más utilizados.

Los métodos de petición se utilizan para indicar una acción que se va a realizar en un recurso determinado. Los más comunes son:

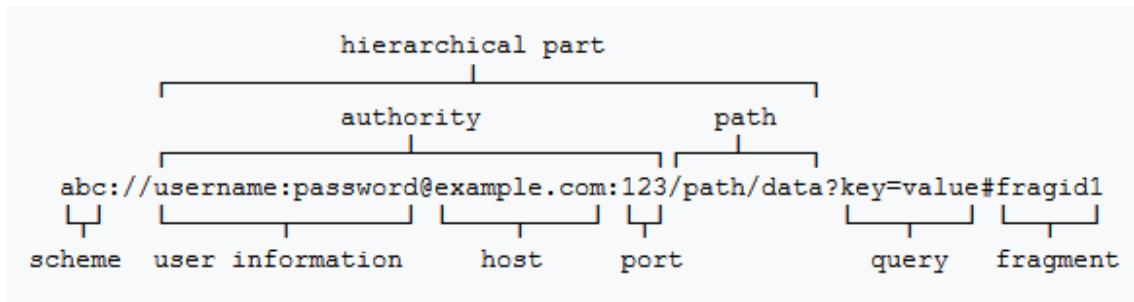
- **GET** solicita al servidor una información o recurso concreto.
- **HEAD** se recibe información de la cabecera de la página que se quiere cargar.
- **POST** envía datos al servidor.
- **PUT** actualizar información existente en el servidor.

4. Estudio sobre el concepto de URI, estructura, utilidad y relación con el protocolo HTTP/HTTPS.

El URI (uniform resource identifier) o identificador uniforme de recursos es una cadena de caracteres utilizada para nombrar recursos en la web, proporcionando una forma estandarizada de localizar recursos.

Su estructura se divide en 5 partes principales.

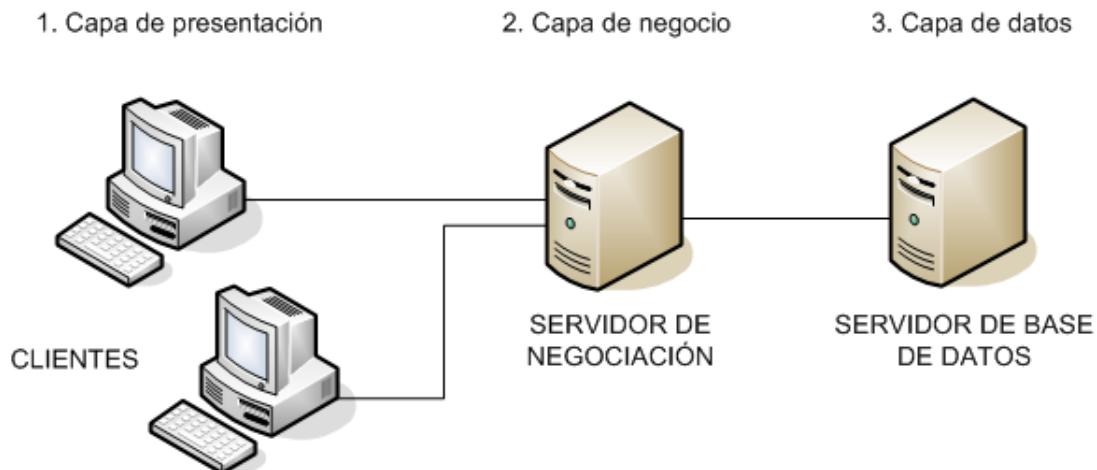
- **Scheme** o **esquema**, especifica el protocolo para acceder al recurso: http, https, ftp, etc.
- **Authority** o **autoridad**, nombre del dominio del recurso.
- **Path** o **camino**, estructura interna de la autoridad.
- **Query** o **consulta**, componente opcional repetible para pasar datos y parámetros al recurso; en caso de ser varios se separan con un &.
- **Fragment** o **fragmento**, componente opcional que apunta a una sección específica del recurso.



5. Modelo de desarrollo de aplicaciones multicapa.

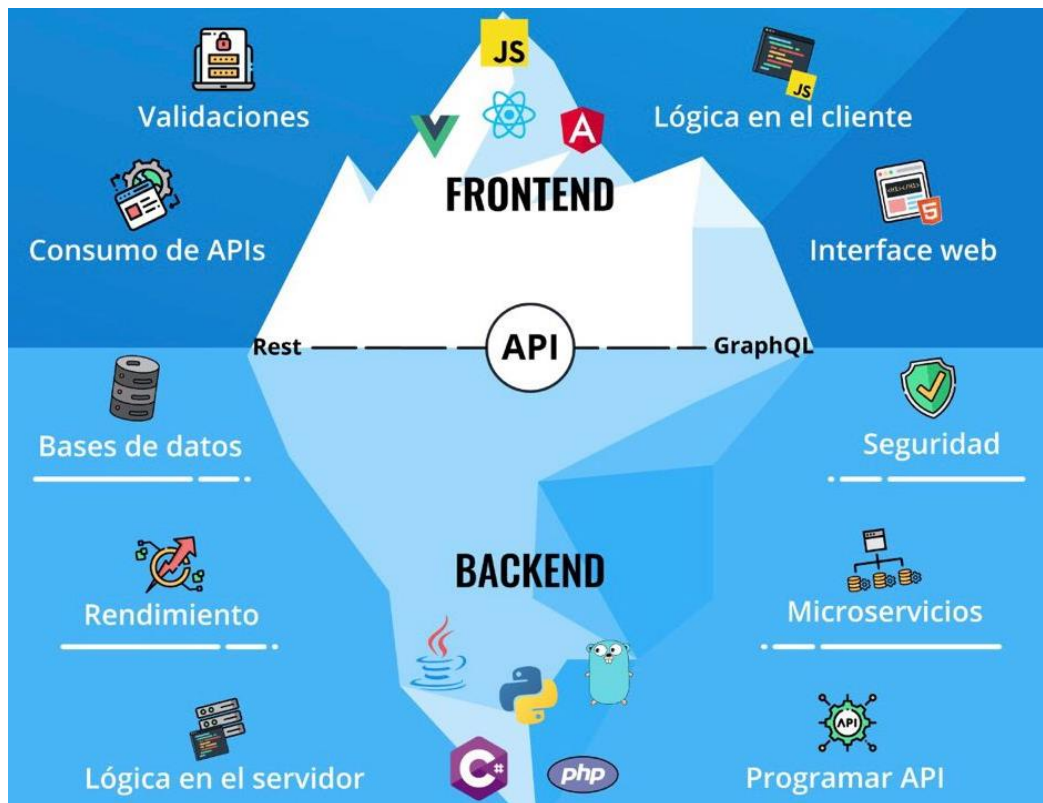
Una aplicación multicapa es aquella que tiene separadas físicamente las funciones de presentación, lógica de negocios y gestión de datos.

- **Capa de presentación** presenta al sistema y comunica y captura información del usuario.
- **Capa de negocios** controla la funcionalidad de la aplicación mediante procesamientos detallados.
- **Capa de datos** almacena los datos.



6. Modelo de división funcional front end y back end para aplicaciones web.

La funcionalidad front end se encarga de la capa de presentación, mientras que back end de la capa de acceso a datos.



7. Página web estática, dinámica, aplicación web y mashup.

- **Página web estática** archivos individuales generados y presentados al usuario de la misma forma.
- **Página web dinámica** elementos en cambio constante, interactivos y funcionales.
- **Aplicación web** programa almacenado en un servidor remoto al que se accede a través de un navegador web.
- **Mashup** una aplicación web es llamada desde otra con el fin de utilizar su contenido o funcionalidad.

8. Componentes de una aplicación web.

- **Servidor web** procesa la aplicación en el lado servidor generando respuestas para el lado cliente.
- **Almacenamiento de datos.**
- **Trabajos** servicios de menor importancia que suceden en segundo plano.
- **Interfaz front end** aspecto visual de la aplicación.
- **Lógica empresarial** realiza operaciones sencillas con rapidez en el lado cliente.

9. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor.

Lado del cliente	Lado del servidor
React	ASP.NET
JavaScript	PERL
CSS	PHP
HTML	Python
Angular	NodeJS

10. Lenguajes de programación utilizados en el lado servidor de una aplicación web.

- **ASP.NET** permite el control de usuario personalizado y una división entre capa de diseño y código.
- **PERL** útil en el procesamiento de textos y archivos.
- **PHP** ofrece una gran facilidad para establecer conexiones con bases de datos.
- **Python** lenguaje multiplataforma con una gran cantidad de soporte, recursos y bibliotecas.
- **NodeJS** tienen una gran agilidad al trabajar con varios procesos en marcha.

© W3Techs.com	usage
1. PHP	75.7%
2. Ruby	6.0%
3. ASP.NET	5.7%
4. Java	5.0%
5. JavaScript	3.6%

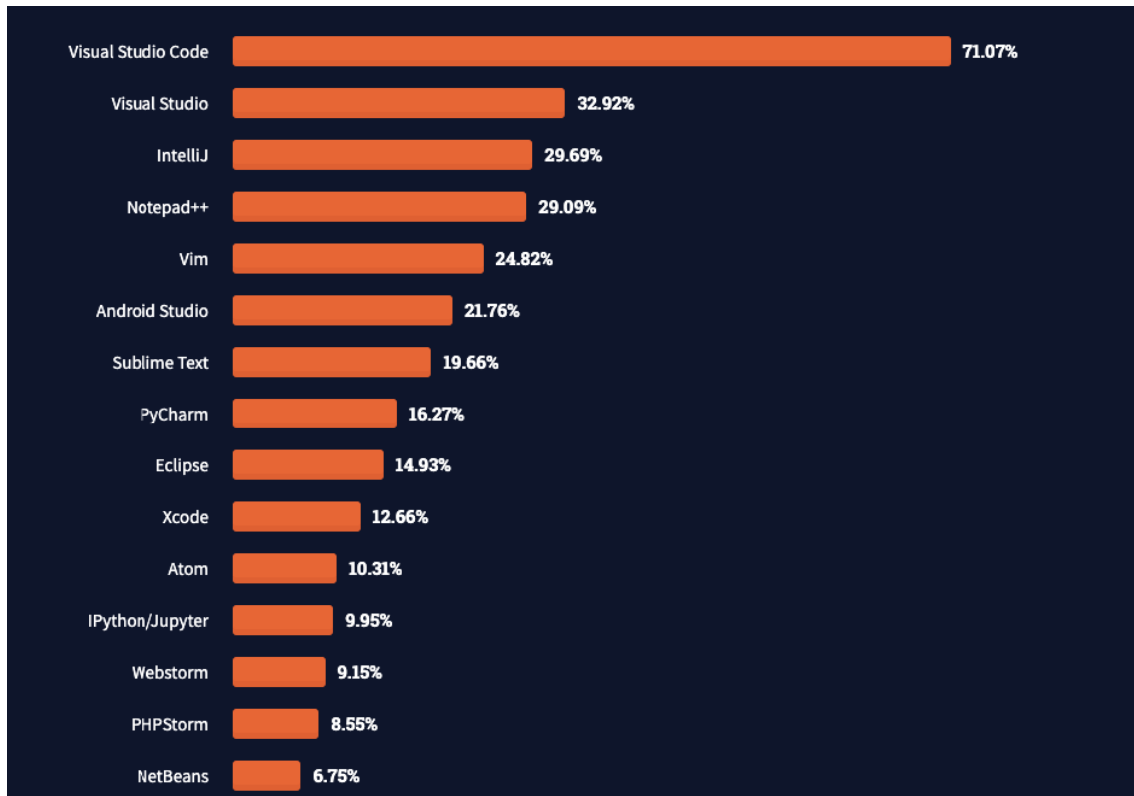
11. Características y posibilidades de desarrollo de una plataforma XAMPP.

XAMPP es un paquete de software que ofrece todos los componentes necesarios para desarrollar aplicaciones web.

Sus características principales son su facilidad de uso y configuración, la portabilidad y compatibilidad multiplataforma y su soporte para múltiples lenguajes de programación y bases de datos.

12. IDE más utilizados.

- **Visual Studio Code**
- **IntelliJ**
- **Vim**
- **Eclipse**
- **NetBeans**



13. Servidores HTTP /HTTPS más utilizados.

- **Nginx**
- **Apache**
- **Cloudflare**
- **LiteSpeed**
- **Microsoft-IIS**

	usage
1. Nginx	33.7%
2. Apache	28.7%
3. Cloudflare Server	22.8%
4. LiteSpeed	14.0%
5. Microsoft-IIS	4.4%

14. Apache HTTP vs Apache Tomcat

Apache HTTP almacena tanto página estáticas, mientras que Apache Tomcat está especializado en las dinámicas.

15. Navegadores HTTP /HTTPS más utilizados.

