



UNIVERSIDAD
DE GRANADA



Tecnologías Web

Grado en Ingeniería Informática

Tema 1 - Introducción

*Este documento está protegido por la Ley de Propiedad Intelectual (Real Decreto Ley 1/1996 de 12 de abril).
Queda expresamente prohibido su uso o distribución sin autorización del autor.*

© Javier Martínez Baena
jbaena@ugr.es

Departamento de Ciencias de la
Computación e Inteligencia Artificial
<http://decsai.ugr.es>



UNIVERSIDAD
DE GRANADA

Tecnologías Web

Grado en Ingeniería Informática

Introducción



1. Introducción

1. Software tradicional vs aplicaciones web
2. Evolución de apps en internet
3. Arquitectura usual de una aplicación web

2. Arquitectura cliente-servidor

1. Protocolo HTTP

3. Ingeniería del Software y desarrollo web

1. Modelo de desarrollo
2. Prototipado

4. Herramientas de desarrollo





Introducción

Software tradicional vs. Aplicaciones web

Software de escritorio

- Dependencia del sistema (SO, hardware)
- Limitadas por el hardware
- Instalación y actualizaciones: las hace el usuario, pueden ser complejas
- Eficiencia
- ...

Aplicaciones web

- Independencia del sistema (SO, hardware)
- Usables en móviles
- Escalabilidad del hardware
- Instalación y actualizaciones: transparentes al usuario
- Limitaciones de acceso al hardware
- Necesidad de conectividad a red (local/internet)
- ...

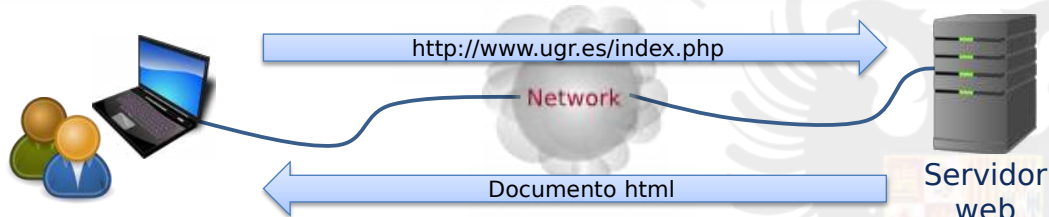


Introducción

Definiciones

Definiciones básicas

- URL: dirección de un recurso en internet
- Protocolo HTTP: comunicación entre cliente y servidor
- Página web: documento con hipertexto (normalmente en HTML)
- Sitio web: colección de páginas web relacionadas entre sí. Normalmente se alojan en un mismo servidor
- Aplicación web: aplicación cuya interfaz de usuario es un navegador web y que se ejecuta a través de internet en un servidor web. Normalmente el resultado de su ejecución son páginas web



Modelo cliente-servidor

Petición: URL, datos de formularios, ficheros, ...
 Respuesta: HTML, imágenes, ficheros, ...



Introducción

Origen del WWW

Tim Berners-Lee

1989-1990: crea la World Wide Web en el CERN (Consejo Europeo para la Investigación Nuclear).

- Primer navegador
- Primer servidor web
- Primera página web

<http://info.cern.ch/hypertext/WWW/TheProject.html>



WWW:

- HTTP: Protocolo para transmitir páginas
- HTML: Lenguaje para escribir páginas
- URL: dirección de las páginas

Aunque aplica el concepto de hiperenlace, este es anterior incluso a los ordenadores.

1994: Se traslada al MIT y pone en marcha el W3C (World Wide Web Consortium) para estandarizar tecnologías web.

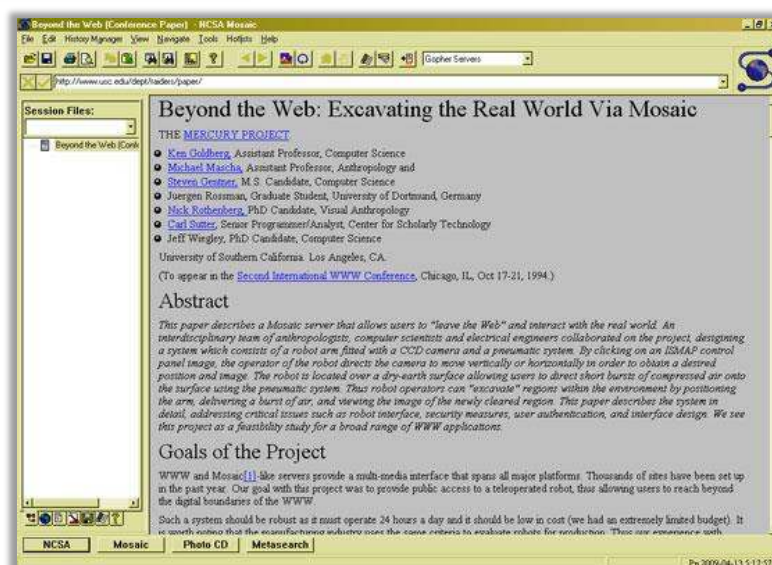


Introducción

Evolución de apps en internet

Inicio de los '90

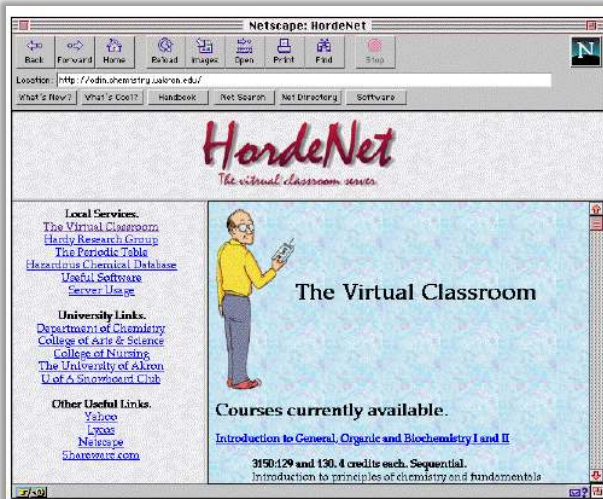
- Páginas basadas en texto
- Muy pocos gráficos
- Hipertexto



<https://www.webpagefx.com/blog/web-design/the-history-of-the-internet-in-a-nutshell/>

Inicio de los '90 - mediados de los '90

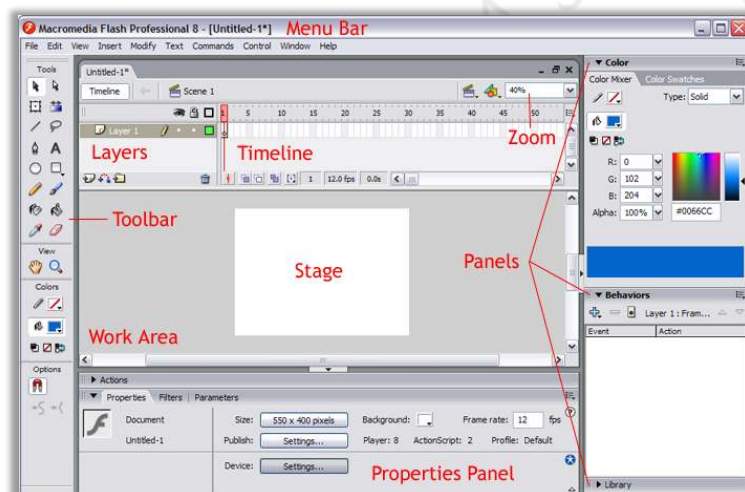
- Páginas basadas en tablas para dar formato (múltiples columnas)
- Uso de frames para incrustar unas páginas en otras (separar contenidos de menús de navegación, pies de página, etc)
- 1994 Nace el W3C (Objetivo: Estandarización)



<https://www.techwyse.com/blog/website-design/evolution-of-web-design/>
<https://www.webpagefx.com/blog/web-design/the-evolution-of-web-design/>

Mediados de los '90

- 1995 - Animaciones simples con JavaScript
- 1996 - Animaciones gráficas e interactividad con Macromedia Flash
- 1996 - Nace CSS (muy básico)
- Uso intensivo de animaciones GIF





Introducción

Evolución de apps en internet

Finales de los '90

- Uso intensivo de Flash
- 1998: PHP3 permite creación de sitios web dinámicos

2000-2005

- CSS se populariza: separación de contenido y aspecto
- 2002: JavaScript se populariza. Formularios interactivos, menús de navegación, etc.
- Web semántica: se incluye metainformación en las páginas para que las máquinas puedan "comprender" lo que contienen
- XML
- Web 2.0 (redes sociales)

2005 - 2010

- 2005 - AJAX
- 2006 - jQuery
- 2008 - Mobile web
- 2010 - HTML5
- 2012 - Responsive design



Introducción

Aplicaciones web

Desarrollo de sitios web

- CMS: Aplicaciones web genéricas para la creación de sitios web
 - Joomla!
 - Drupal
 - WordPress
 - Moodle
 - ...
- Programación del sitio
 - Programación
 - BBDD
 - Diseño
 - ...

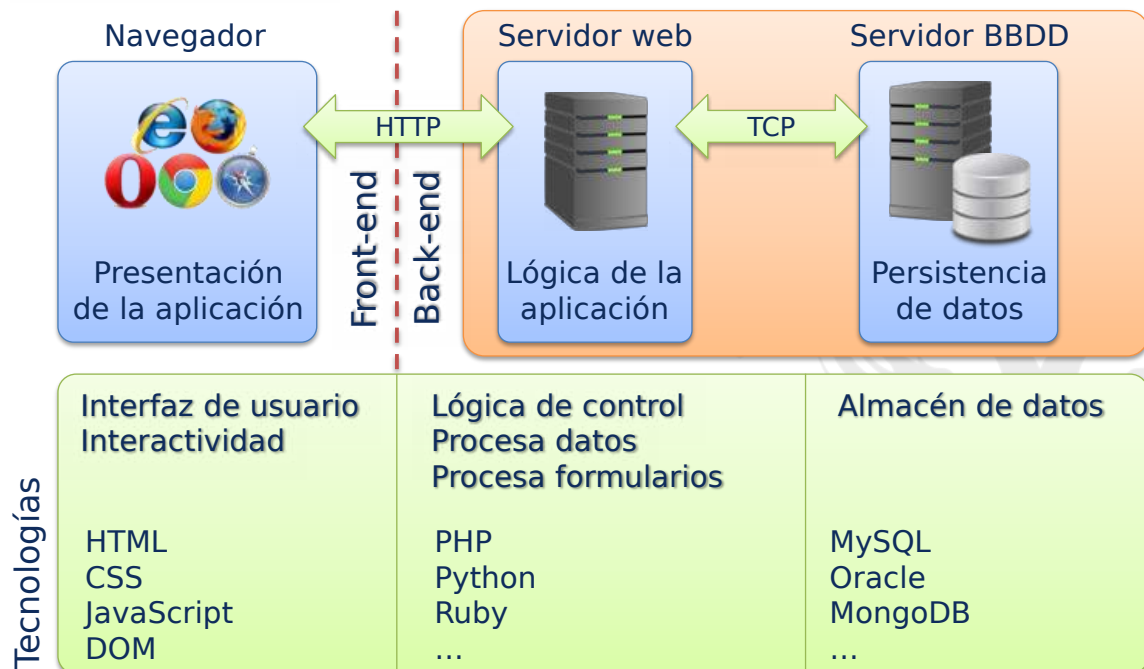


Introducción

Aplicaciones web

Arquitectura habitual de una aplicación web

Modelo de 3 capas



Full-stack developer: controla todas las tecnologías



Introducción

Aplicaciones web

Arquitectura habitual de una aplicación web

Modelo de 3 capas

Lenguaje	Frameworks/bibliotecas
CSS	Bootstrap 360 Foundation Skeleton ...
JavaScript	jQuery underscore.js AngularJS backbone.js React ...
PHP	Symphony Laravel CakePHP Zend ...

1. Introducción

1. Software tradicional vs aplicaciones web
2. Evolución de apps en internet
3. Arquitectura usual de una aplicación web

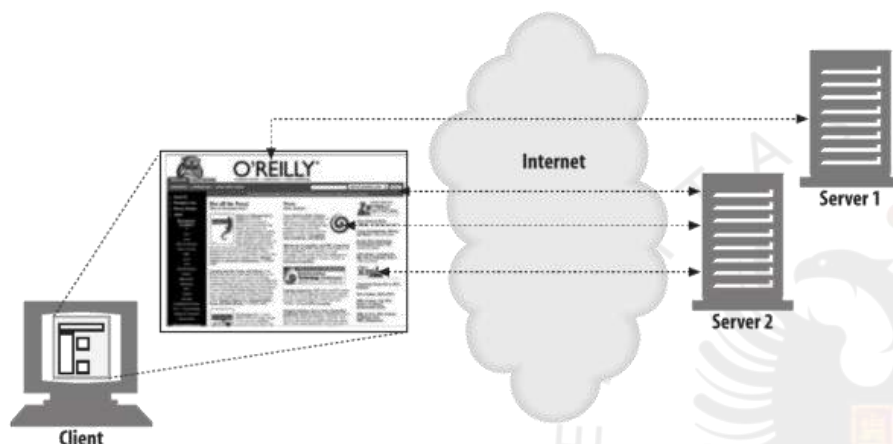
» 2. Arquitectura cliente-servidor

1. Protocolo HTTP
3. Ingeniería del Software y desarrollo web
 1. Modelo de desarrollo
 2. Prototipado
4. Herramientas de desarrollo

Modelo cliente-servidor

Cliente: navegador

Servidor: servidor web



HTTP: protocolo para intercambiar información en la web
Es un protocolo "stateless"

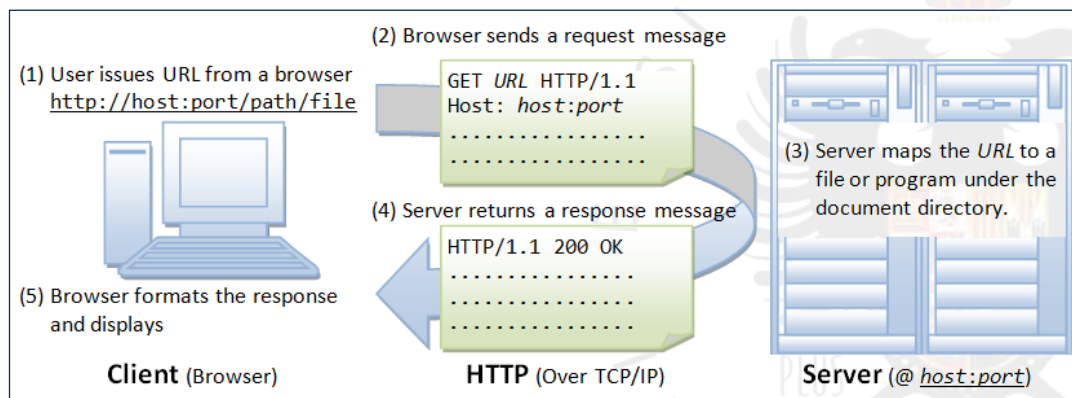


Arquitectura cliente-servidor

Protocolo HTTP

HTTP (HyperText Transfer Protocol)

- HTTP/0.9 en 1991 por Tim Berners-Lee
 - Solo soporta GET, no soporta MIME, no soporta HTTP headers, ...
- HTTP/1.0 (RFC 1945) en 1996
 - Añade headers, otros métodos (POST, ...), MIME, ...
- **HTTP/1.1 (RFC 2616)** en 1999 (<https://tools.ietf.org/html/rfc2616>)
 - Mejoras de rendimiento y correcciones de diseño
- HTTP/2.0 (varios RFC), borrador en 2012, se usa a partir de 2015
 - Mejoras de rendimiento, múltiples transferencias por conexión, ...



https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html



Arquitectura cliente-servidor

Protocolo HTTP

Ejemplo de petición HTTP

```
curl -v http://decsai.ugr.es/~jbaena/holatw.html
```



`http://decsai.ugr.es/~jbaena/holatw.html`



1.- Establecer conexión TCP:

- * Trying 150.214.191.180...
- * Connected to decsai.ugr.es (150.214.191.180) port 80 (#0)

2.- Enviar petición HTTP:

```
GET /~jbaena/holatw.html HTTP/1.1
Host: decsai.ugr.es
User-Agent: curl/7.47.0
Accept: */*

```

} Petición
 } HTTP Headers
 ← (línea en blanco)
 } Cuerpo (vacío)



Arquitectura cliente-servidor

Protocolo HTTP



http://decsai.ugr.es/~jbaena/holatw.html

Devolución de página HTML

3.- Respuesta HTTP:

```

HTTP/1.1 200 OK
Date: Mon, 30 Oct 2017 07:42:31 GMT
Server: Apache
Last-Modified: Mon, 30 Oct 2017 07:42:04 GMT
Accept-Ranges: bytes
Content-Length: 145
Vary: Accept-Encoding
Connection: close
Content-Type: text/html
    
```

} Respuesta

} HTTP Headers

← (línea en blanco)

```


<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Hola Mundo</title>
</head>
<body>
<h1>iHola Tecnologías Web!</h1>
</body>
    
```

} Cuerpo

* Closing connection 0

4.- Cerrar conexión TCP/IP

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada
© Javier Martínez Baena
17



Arquitectura cliente-servidor

Protocolo HTTP

HTTP (HyperText Transfer Protocol)

Formato de los mensajes HTTP

Petición HTTP:

```

<método> <URL> <versión>
<encabezados>
<línea en blanco>
<cuerpo de la petición>
    
```

Cuerpo: otros datos

Método: acción solicitada

URL: URL completa o bien solo el path

Versión: versión del protocolo HTTP
HTTP/0.9, HTTP/1.0, HTTP/1.1, ...

Encabezados: uno por línea de la forma:
nombre: valor

Respuesta HTTP:

```

<versión> <estado> <motivo>
<encabezados>
<línea en blanco>
<cuerpo de la respuesta>
    
```

Estado: código de respuesta

Motivo: texto con explicación de la respuesta

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada
© Javier Martínez Baena
18



HTTP (HyperText Transfer Protocol)

Formato de los mensajes HTTP

Métodos:

- GET Petición de documento
- HEAD Solo petición de header de un documento
- POST Petición de documento (adjuntando datos)
- PUT Subir un fichero al servidor y almacenarlo
- DELETE Petición para borrar un documento
- ...

Estado: número de tres dígitos

- 200-299 (éxito en la petición)
 - 200 OK Éxito de la petición
 - 201 Created Éxito de PUT
 - 204 No Content Se envía solo header, sin cuerpo
- 300-399 (redirección)
- 400-499 (error del cliente)
 - 401 Unauthorized Se necesita autenticación
 - 403 Forbidden La petición se rechaza por el servidor
 - 404 Not Found No se encuentra la URL
- 500-599 (error del servidor)



HTTP (HyperText Transfer Protocol)

Formato de los mensajes HTTP

Encabezados genéricos (para peticiones y respuestas):

- Date Fecha de creación del mensaje
- ...

Encabezados de la petición:

- User-Agent Agente que hace la petición
- UA-Disp Características de la pantalla del cliente
- Host Host al que se dirige la petición
- Accept-Language Lenguaje que acepta el cliente
- Accept Tipos MIME aceptados
- ...

Encabezados de respuesta:

- Server Nombre y versión del servidor web

Encabezados de entidad (relativos al contenido/cuerpo):

- Content-Length Longitud en bytes del mensaje
- Content-Encoding Codificación del cuerpo del mensaje
- Content-Language Lenguaje en el que está escrito el cuerpo
- ...



Arquitectura cliente-servidor

Protocolo HTTP

Ejemplo de petición HTTP

Petición POST (con datos en el encabezado)

Simulamos el envío de datos de un formulario HTML

```
curl -v http://decsai.ugr.es/~jbaena/holatw.html --data "v=hola" --trace-ascii /dev/stdout
```

```
== Info: Trying 150.214.191.180...
== Info: Connected to decsai.ugr.es (150.214.191.180) port 80 (#0)
=> Send header, 165 bytes (0xa5)
0000: POST /~jbaena/holatw.html HTTP/1.1
0024: Host: decsai.ugr.es
0039: User-Agent: curl/7.47.0
0052: Accept: */*
005f: Content-Length: 6
0072: Content-Type: application/x-www-form-urlencoded
00a3:
=> Send data, 6 bytes (0x6)
0000: v=hola
== Info: upload completely sent off: 6 out of 6 bytes
```

Petición

HTTP Headers

← (línea en blanco)

Cuerpo



Arquitectura cliente-servidor

Protocolo HTTP

```
jbaena@void:~$ telnet www.senado.es 80
Trying 193.146.129.72...
Connected to www.senado.es.
Escape character is '^]'.
GET /web/index.html HTTP/1.0
```

```
HTTP/1.1 200 OK
Date: Tue, 31 Jan 2017 08:22:24 GMT
Cache-Control: no-cache
Set-Cookie: JSESSIONID=w5NHYQJQJh2tSCPbCCjWjKdz78PJ4HzfQJ29BrJQr1L7HCQCG75d!-1221
X-ORACLE-DMS-ECID: 8d5ad50f89e64e7a:-1e695803:159dedfdc33:-8000-000000000022163a
Content-Language: es-ES
X-Powered-By: Servlet/2.5 JSP/2.1
Vary: Accept-Encoding
Connection: close
Content-Type: text/html; charset=UTF-8
Set-Cookie: senpers=218540224.20480.0000; path=/
Set-Cookie: TS013cf280=019e5de7f1d17ce004b0d9645db316

<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//
f="/web/images/escudo.ico" rel="SHORTCUT ICON"/><titl
12016"/><link rel="stylesheet" type="text/css" media=
heet" type="text/css" media="screen"/><link href="/we
></script><script type="text/javascript" src="/web/js
t" src="/web/js/jquery-ui-skin.js"></script><script t
ript type="text/javascript" src="/web/js/agenda.js">
```



Arquitectura cliente-servidor

Protocolo HTTP

→

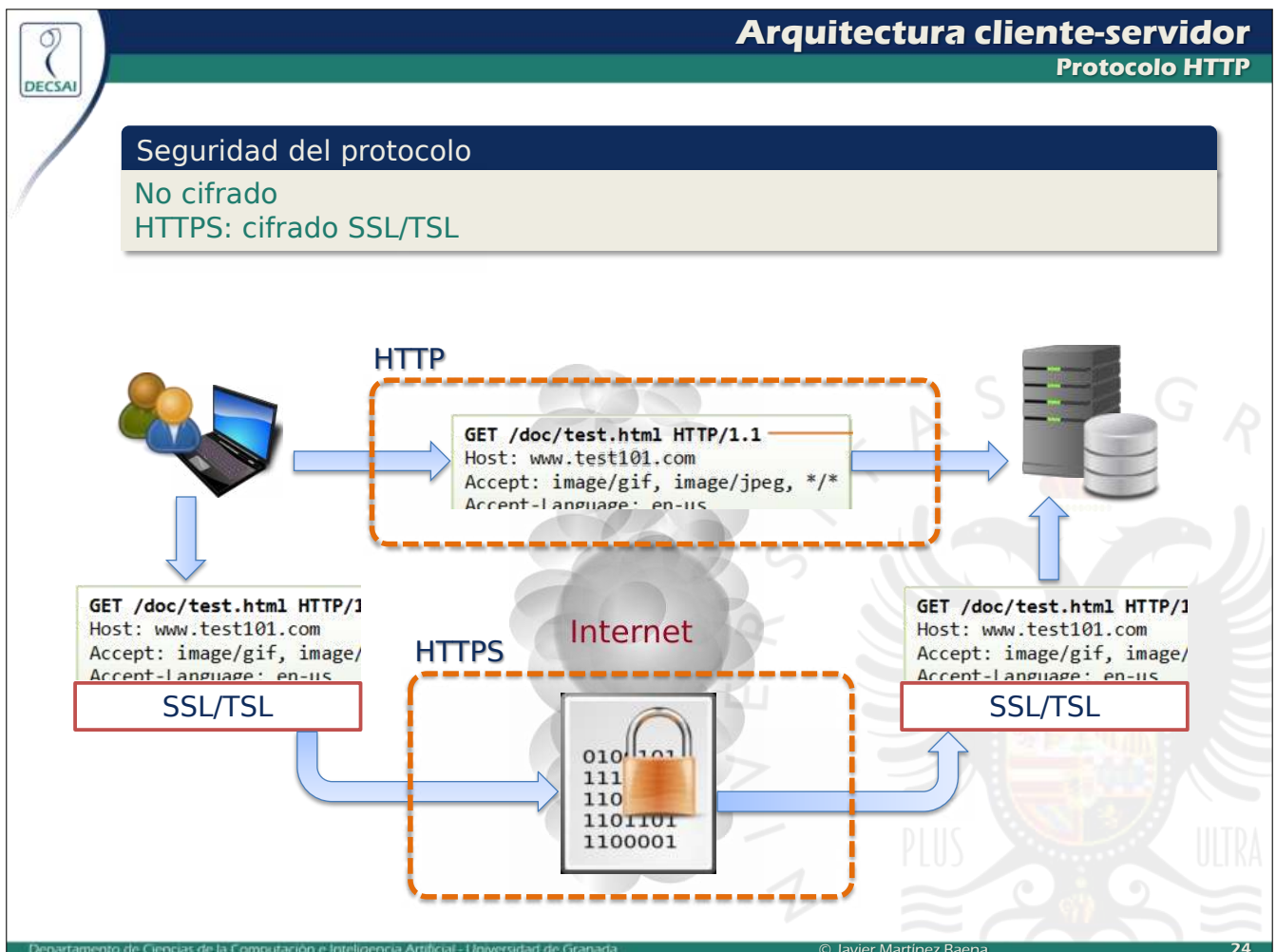
Redirección

```
jbaena@void:~$ telnet void.ugr.es 80
Trying 150.214.190.100...
Connected to void.ugr.es.
Escape character is '^]'.
GET /index.html HTTP/1.0
```

```
HTTP/1.1 302 Found
Date: Tue, 31 Jan 2017 08:18:56 GMT
Server: Apache/2.4.18 (Ubuntu)
Strict-Transport-Security: max-age=63072000; includeSubDomains
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Location: https://void.ugr.es/index.html
Content-Length: 291
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>302 Found</title>
</head><body>
<h1>Found</h1>
<p>The document has moved <a href="https://void.ugr.es/index.html">here</a>
</p>
<address>Apache/2.4.18 (Ubuntu) Server at void.ugr.es</address>
</body></html>
Connection closed by foreign host.
```

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada © Javier Martínez Baena 23





Arquitectura cliente-servidor

Protocolo HTTP

```
curl -v https://decsai.ugr.es/images/leon2.png
```

```
* Trying 150.214.191.180...
* Connected to decsai.ugr.es (150.214.191.180) port 443 (#0)
* found 150 certificates in /etc/ssl/certs/ca-certificates.crt
* found 602 certificates in /etc/ssl/certs
* ALPN, offering http/1.1
* SSL connection using TLS1.2 / ECDHE_RSA_A
* server certificate verification OK
* server certificate status verifica
* common name: decsai.ugr.es (matche
* server certificate expiration date
* server certificate activation date
* certificate public key: RSA
* certificate version: #3
* subject: C=ES,ST=Granada,L=Granada
Computacion,CN=decsai.ugr.es
* start date: Fri, 10 Feb 2017 00:00
* expire date: Wed, 19 Feb 2020 12:00
* issuer: C=NL,ST=Noord-Holland,L=Am
* compression: NULL
* ALPN, server accepted to use http/1.1

> GET /images/leon2.png HTTP/1.1
> Host: decsai.ugr.es
> User-Agent: curl/7.47.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Tue, 20 Feb 2018 09:19:13 GMT
< Server: Apache
< Last-Modified: Thu, 04 Sep 2014 13:00:42 GMT
< Accept-Ranges: bytes
< Content-Length: 2887
< Connection: close
< Content-Type: image/png
<
PNG
IHDR:9gAMA7
ImageRe
ID
w+J.P1
...
* Closing connection 0
```

Datos binarios de la imagen



Arquitectura cliente-servidor

Protocolo HTTP

Protocolo "stateless"

Sin "memoria" de lo que ocurrió con anterioridad
Solución: uso de cookies u otros métodos



El servidor puede almacenar "cookies" en el cliente
El cliente se las devuelve en cada nueva petición de forma automática

1. Introducción

1. Software tradicional vs aplicaciones web
2. Evolución de apps en internet
3. Arquitectura usual de una aplicación web

2. Arquitectura cliente-servidor

1. Protocolo HTTP

» 3. Ingeniería del Software y desarrollo web

1. Modelo de desarrollo
2. Prototipado

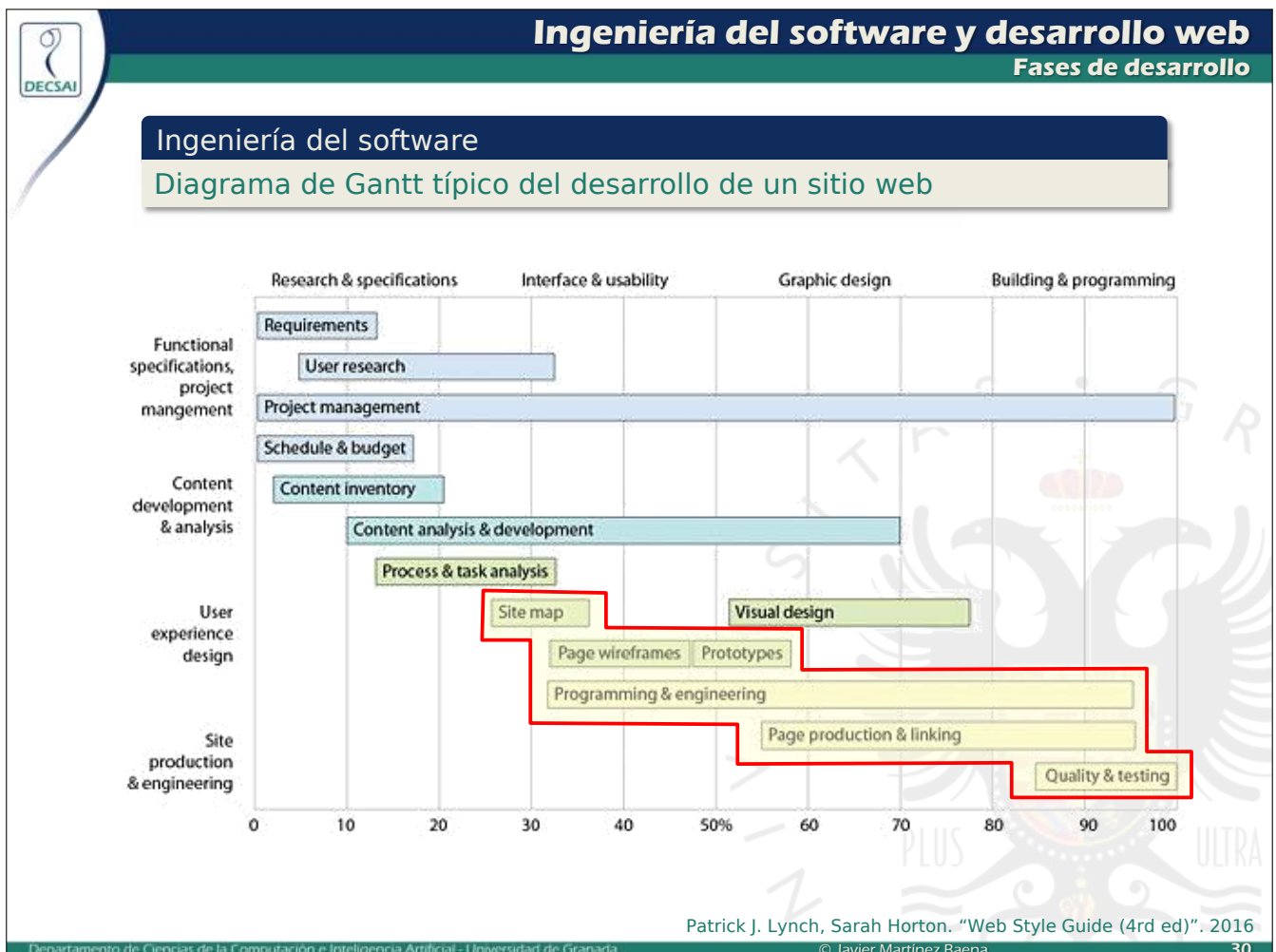
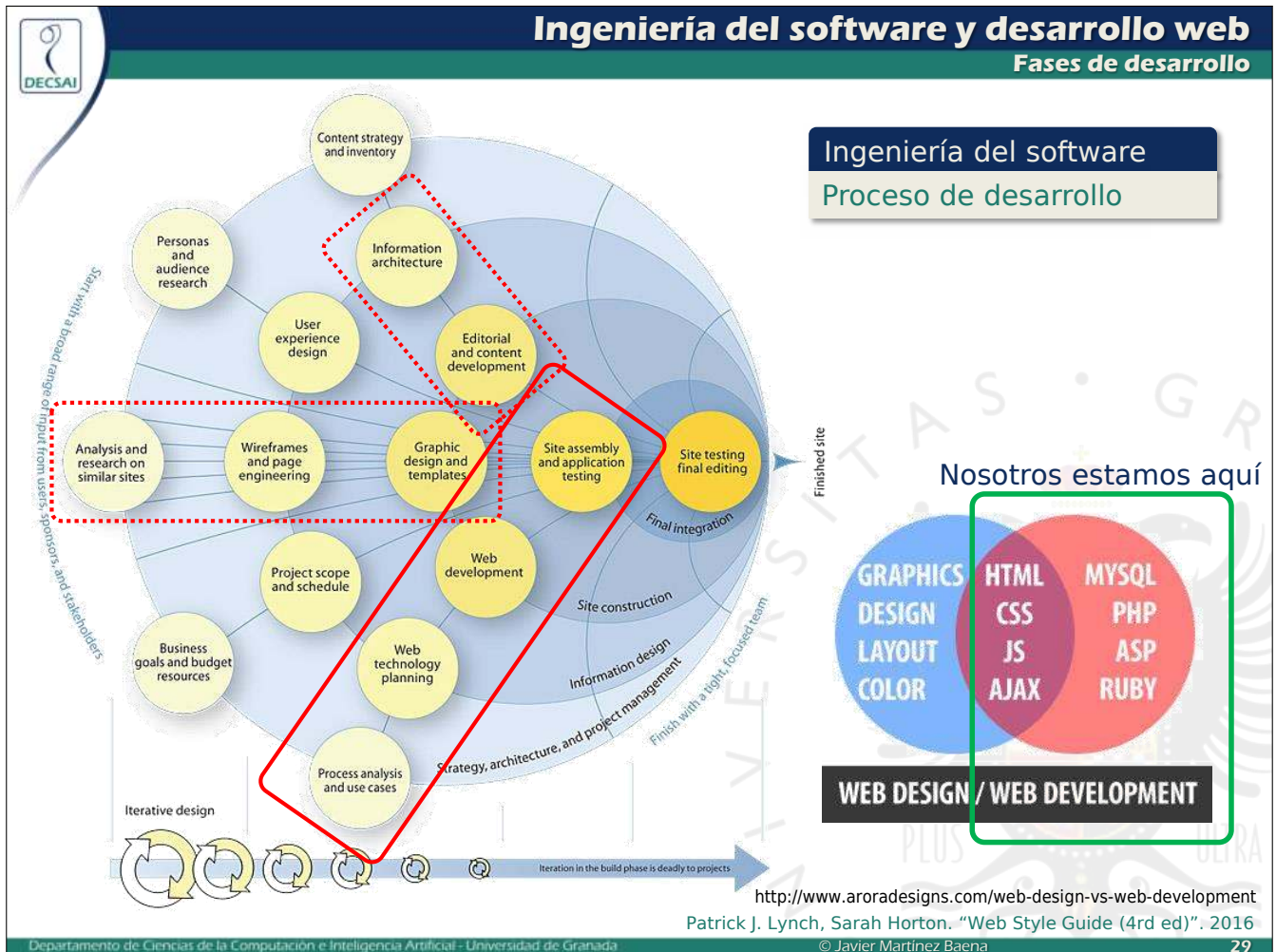
4. Herramientas de desarrollo

Modelo de desarrollo

1. Recopilación de información
Propósito, objetivos, público, contenido, análisis de requisitos
2. Planificación
Mapa del sitio
3. Diseño
Aspecto del sitio, prototipos, mockups, wireframes, HTML, CSS
4. Desarrollo
Creación del sitio, plantillas, interactividad, JavaScript, PHP, ...
Cargar contenidos
5. Pruebas
Servidor en producción y pruebas, Compatibilidad navegadores, ...
6. Mantenimiento

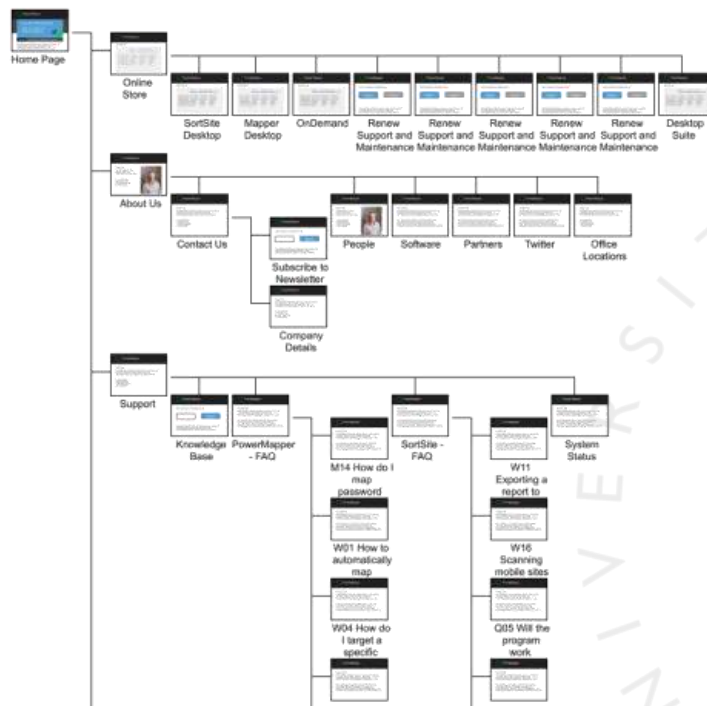
Disciplinas relacionadas

- Diseño gráfico, Interfaces de usuario
- **Programación Front-end**
- **Programación Back-end**
- Gestión de BBDD
- Posicionamiento web
- Administración de sistemas, Seguridad informática



Mapa del sitio

Útil para: planificar el sitio, trato con cliente



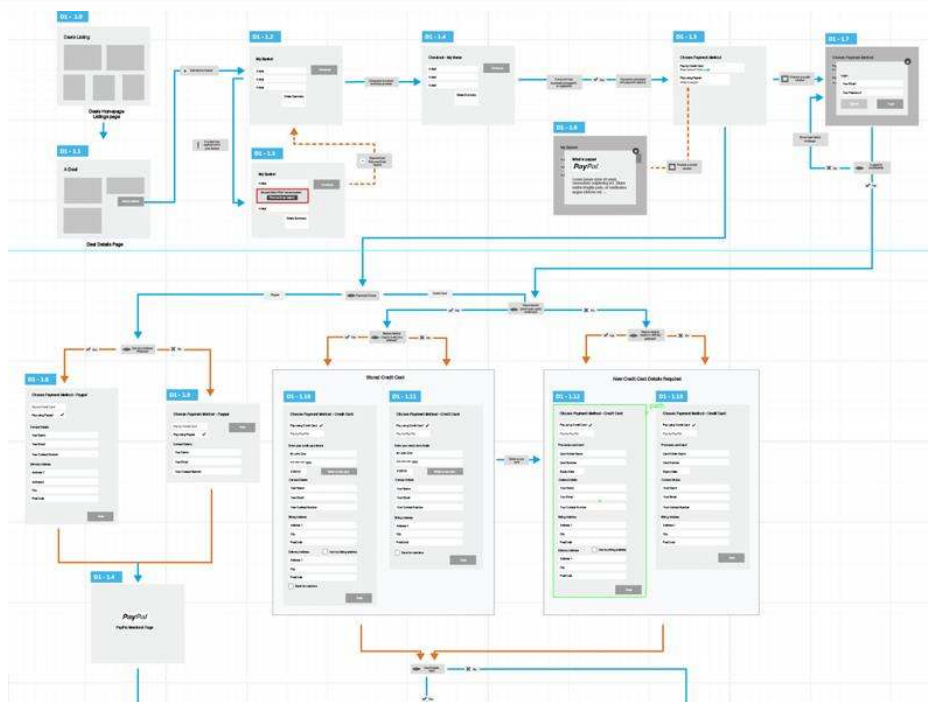
<https://www.powermapper.com/products/mapper/>

© Javier Martínez Baena

31

Diagramas de estado: representación del sitio web

Útil para: diseño web, programación web



<https://dribbble.com/shots/763887-Full-Transactional-Flows>

© Javier Martínez Baena

32



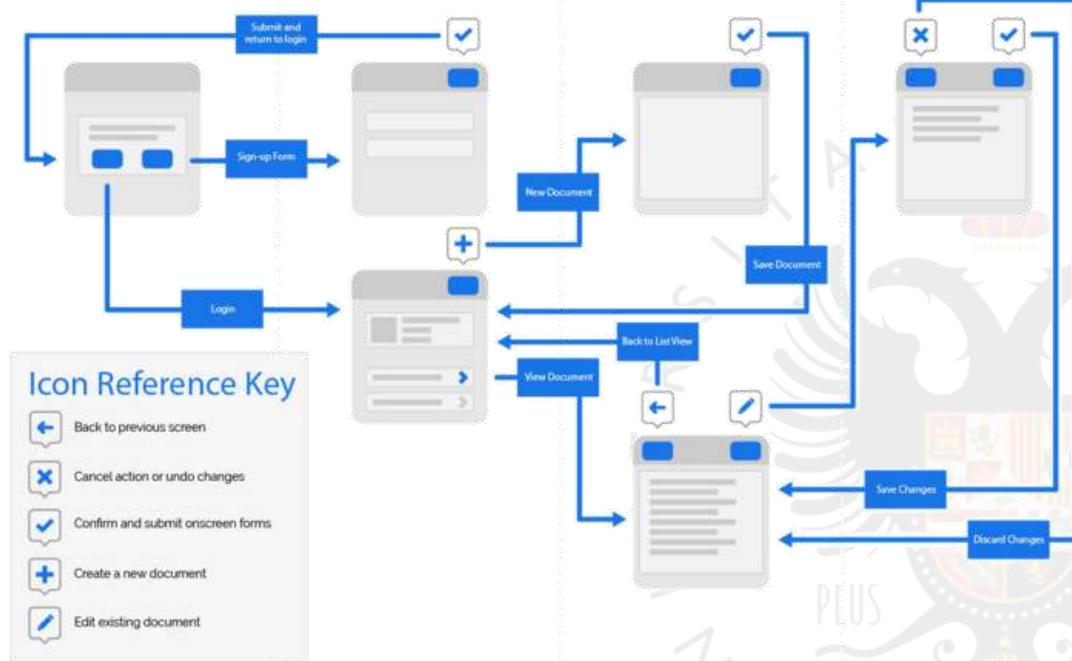
Ingeniería del software y desarrollo web

Diagramas de estado

Diagramas de estado: representación del sitio web

Útil para: diseño web, programación web

EXAMPLE Mobile Flowchart



<https://www.mediatopia.co.uk/blog/difference-between-mobile-site-and-responsive-design>

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada

© Javier Martínez Baena

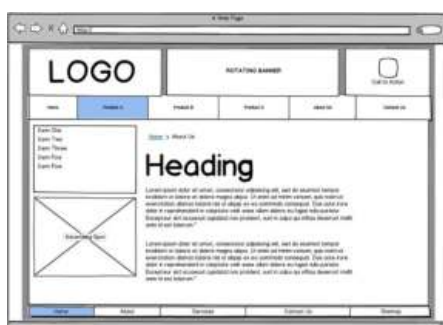
33



Ingeniería del software y desarrollo web

Prototipado

	Aspectos clave	Propósito	Apariencia
Wireframe Baja fidelidad	Representación básica de los elementos del diseño y estructura de la información	Documentación y comunicación	B/N/gris. Representación simple del interfaz
Prototipo Media/alta fidelidad	Interactividad Probar la usabilidad	Diseño de UI y prueba de la usabilidad e interactividad	Similar a Wireframe pero con interactividad
Mockup Media/alta fidelidad	Representa estructura de la información Visualización estática y marketing	Aceptación del diseño por parte del cliente	No interactivo aunque más rico que Wireframe



<https://designmodo.com/wireframing-prototyping-mockup/>

<http://thebridgesummit.co/wire-frames>

<http://techsini.com/category/web-development/website-mockup-generator/>

Departamento de Ciencias de la Computación e Inteligencia Artificial - Universidad de Granada

© Javier Martínez Baena

34

1. Introducción

1. Software tradicional vs aplicaciones web
2. Evolución de apps en internet
3. Arquitectura usual de una aplicación web

2. Arquitectura cliente-servidor

1. Protocolo HTTP

3. Ingeniería del Software y desarrollo web

1. Modelo de desarrollo
2. Prototipado

4. Herramientas de desarrollo

Herramientas de desarrollo

¿Qué se necesita?

Cliente Web

Navegador (browser)
Algún plugin (firebug, ...)



Servidor Web



Hardware y S.O.



Servidor web



Lenguaje scripting



DBMS

- Editores de texto (vi, sublime, ...)
- IDE (Eclipse, Netbeans, Aptana, ...)
- Documentación
- Validación de código





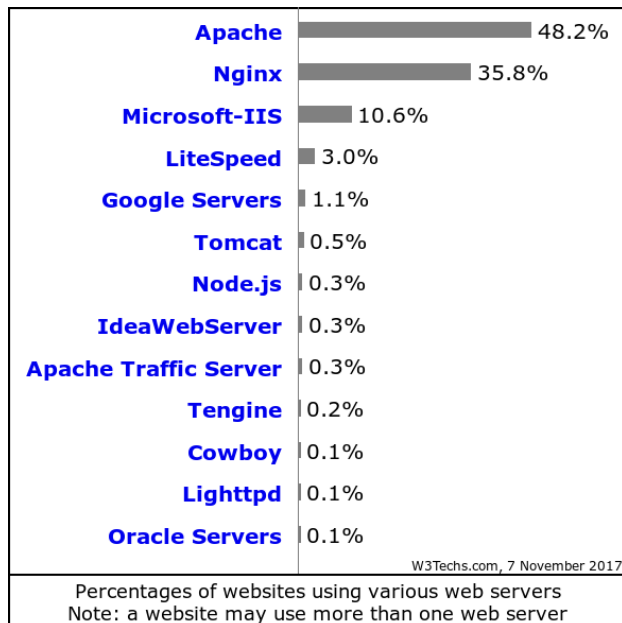
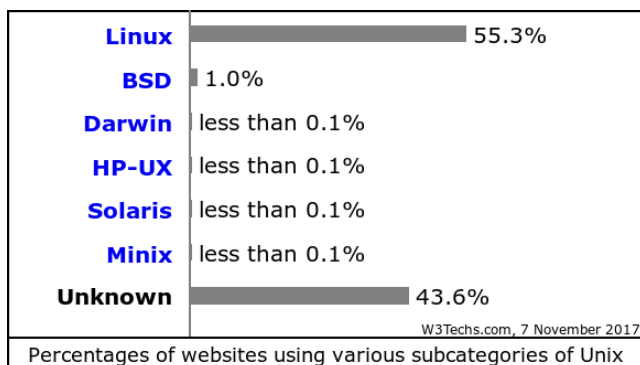
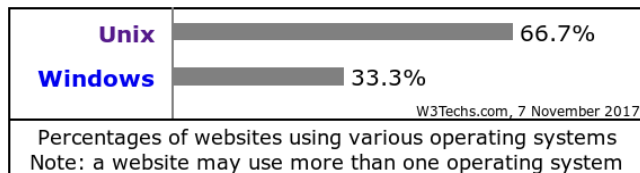
Herramientas de desarrollo

Sistema Operativo del servidor

W3Techs
Web Technology Surveys

<https://w3techs.com/>

Análisis de top 10.000.000 sites
Noviembre 2017

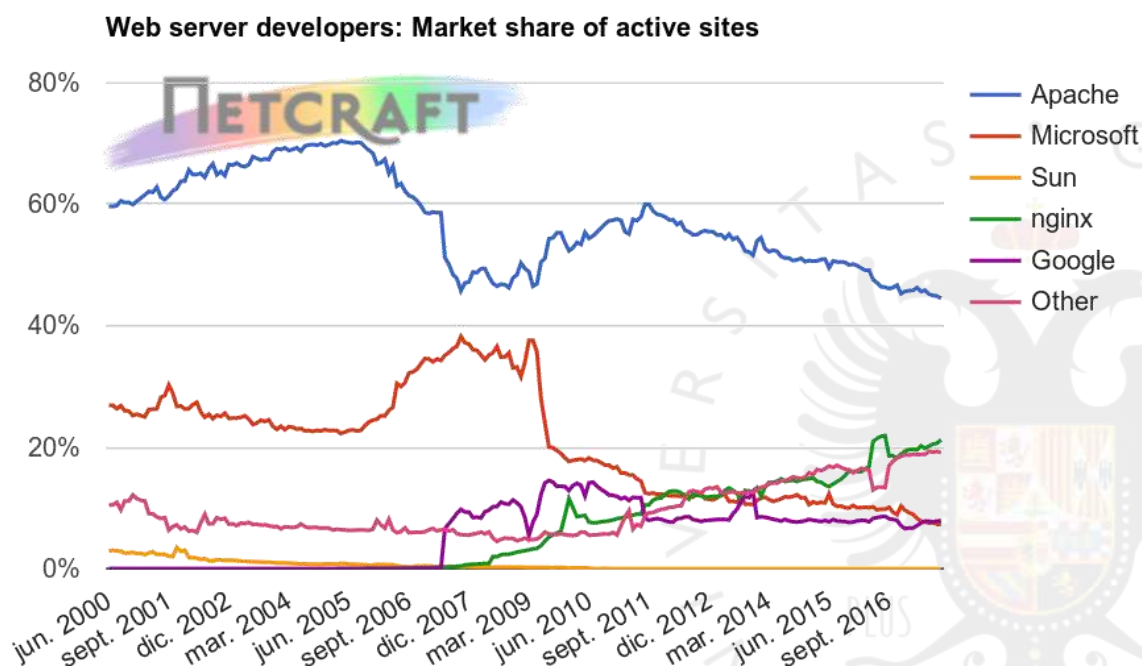


Herramientas de desarrollo

Servidor web

Evolución de servidores web
<https://news.netcraft.com>

Análisis de aprox 10M sites
Octubre 2017





Herramientas de desarrollo

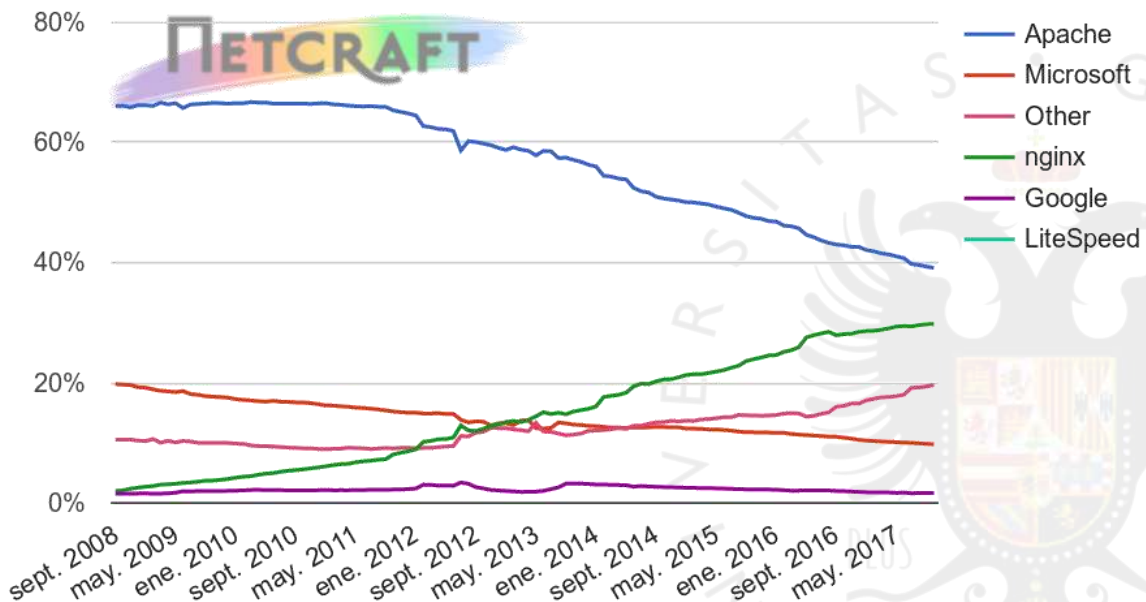
Servidor web

Evolución de servidores web de más tráfico

Análisis de aprox 1M sites
Octubre 2017

<https://news.netcraft.com>

Web server developers: Market share of the top million busiest sites



Herramientas de desarrollo

Servidor web

Comparación Apache vs NGINX

Apache (1995)

- Mejor soporte (documentación, comunidad, ...)
- Mayor cantidad de módulos para personalización
- Mejor soporte de carga dinámica de módulos
- Soporte completo en: UNIX, MacOS, Ms Windows
- Soporte de configuración por usuario (.htaccess)

NGINX (2002)

- Mejor rendimiento sirviendo contenido estático
- Mejor rendimiento con muchos accesos concurrentes
- Soporta menos variantes de UNIX
- El soporte de Ms Windows es pobre
- Más ligero, consume menos recursos
- No soporta configuración por usuario (.htaccess)

NGINX tiene más sentido cuando hay servidores dedicados a un mismo servicio y con gran volumen de tráfico.



Herramientas de desarrollo

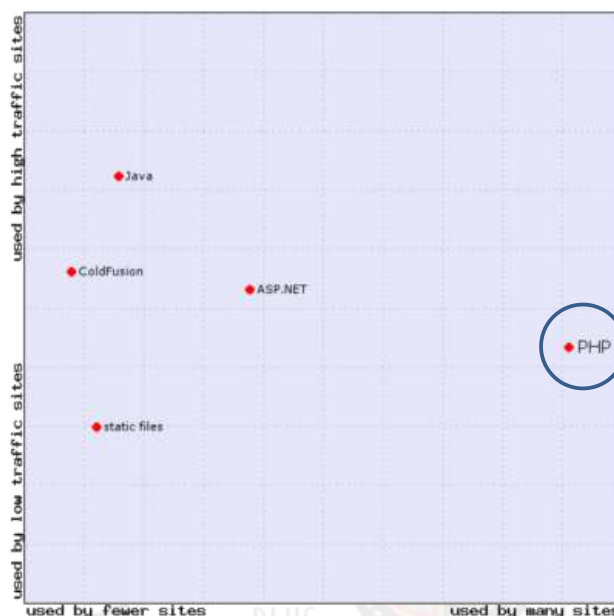
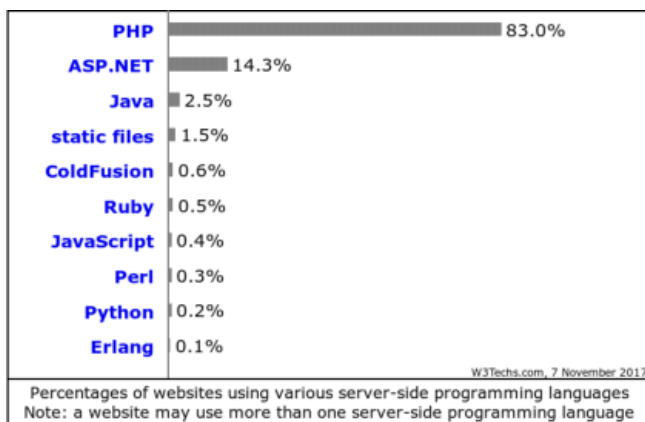
Lenguaje de scripting en el servidor

W3Techs

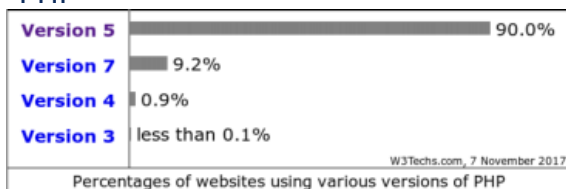
Web Technology Surveys

<https://w3techs.com/>

Análisis de top 10.000.000 sites
Noviembre 2017



PHP



Herramientas de desarrollo

Persistencia de la información en el servidor

DB-ENGINES

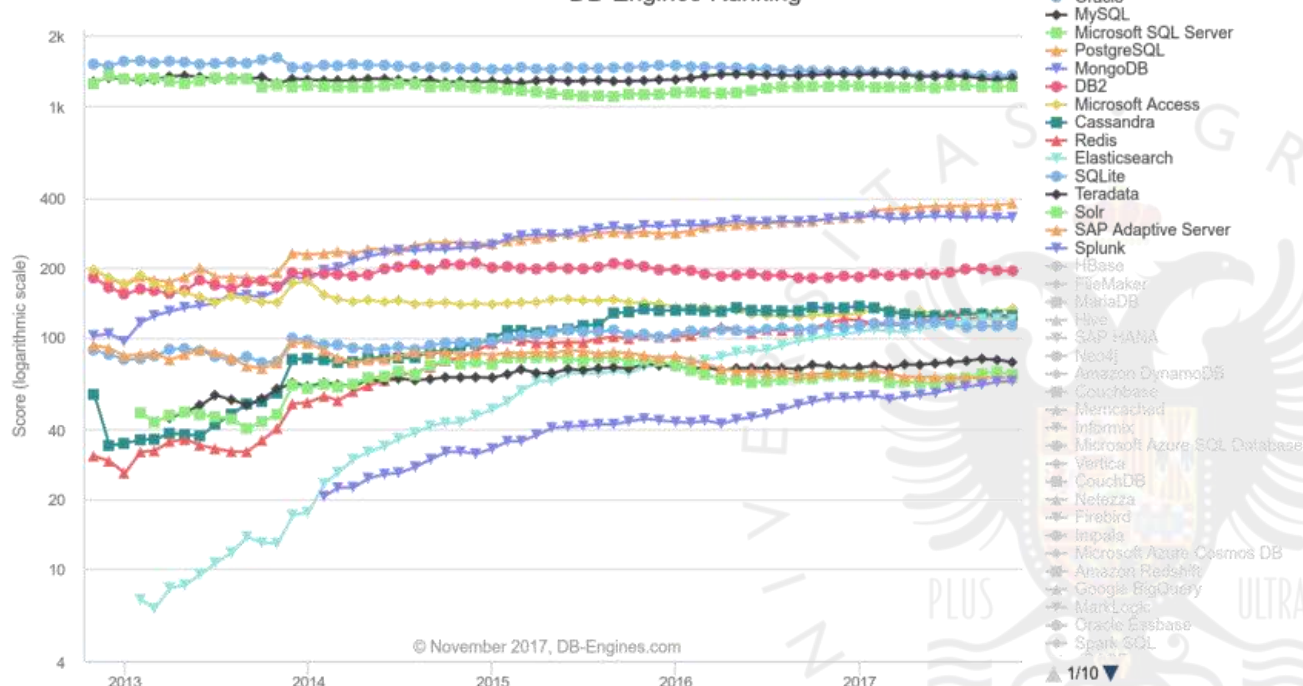
<http://db-engines.com>

Noviembre 2017

Número de menciones en websites, Google trends, discusiones en stackoverflow y DBA stack Exchange, ofertas de trabajo, ...

Ranking de popularidad de DBMS

DB-Engines Ranking





Herramientas de desarrollo

¿Qué usaremos nosotros?

¿Qué usaremos nosotros?

- Servidor web Apache (GNU/Linux / Windows / Mac)
- Servidor de BBDD MySQL
- Programación en el cliente
 - HTML, CSS, JavaScript
- Programación en el servidor
 - PHP
- Recomendaciones:
 - Navegador: Chrome / Firefox
 - Editor de texto: Sublime
 - WireFrame/Mockup: Pencil
 - Diseño BBDD: MySQL Workbench / PHPMyAdmin
- Documentación online:
 - <https://www.w3c.org>: estándares (HTML, CSS, ...), validación
 - <https://developer.mozilla.org>: HTTP, HTML, CSS, JavaScript, DOM
 - <https://php.net>: PHP
 - <https://www.w3schools.com>: HTML, CSS, JavaScript, SQL, PHP, DOM (+ Bootstrap, jQuery, AJAX, ...)
 - <https://stackoverflow.com>: Dudas puntuales
- Bibliografía de la asignatura



Herramientas de desarrollo

Bibliografía



David Gourley & Brian Totty
HTTP The definitive guide
 O'Reilly. 2002



Patrick J. Lynch, Sarah Horton
Web Style Guide (3rd online, 4th book)
 2009 / 2016
<http://webstyleguide.com/>

- https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html
- https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Authentication.html
- https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_StateManagement.html