
theme : "white" transition: "zoom" highlightTheme: "darkula" customTheme : "lola_theme"

EI1042 - Tecnologías y Aplicaciones Web

EI1036- Tecnologías Web para los Sistemas de Información

(2018/2019)

Professora: Dra. Dolores Mª Llidó Escrivá

[Universitat Jaume I.](#)

Índice

1. Accesibilidad
 2. Normativa Española
 3. Seguridad
 4. Seguridad en WP
 5. Responsive Web design
 6. Librerías JS para presentaciones
 7. Frameworks en el cliente
-

Accesibilidad

Guías accesibilidad

- UAAG 2.0: User Agent Accessibility Guidelines (for developers of Web browsers).
 - WCAG 2.1: Web Content Accessibility Guidelines (site designers).
 - ATAG 1.0 : Authoring Tool Accessibility Guidelines (HTML editors).
 - WAI-ARIA : Accessible Rich Internet Applications. (Dynamic content and advanced user interface).
-

Legislación Española

- Ley 34/2002, de 11 de julio, de Servicios de la Sociedad de la Información y de Comercio Electrónico. LSSI (B.O.E. de 12-7-02).
 - Fija por primera vez la obligación de que las páginas web de la Administración Pública española fueran accesibles
- Ley 9/2017, de 8 de noviembre, de Contratos del Sector Público, por la que se transponen al ordenamiento jurídico español las Directivas del Parlamento Europeo y del Consejo 2014/23/UE y 2014/24/UE, de 26 de febrero de 2014.

- Incluyen requisitos de *accesibilidad universal y diseño universal o diseño para todas las personas*.

--

Normativas accesibilidad Europea

Norma UNE 139803:2012 (PDF) , es equivalente a las WCAG 2.0

Es una Directiva del Parlamento Europeo sobre la accesibilidad de los sitios web y aplicaciones móviles.

- Establece los requisitos de accesibilidad de los sitios web y apps del sector público.
- Deberá ser transpuesta a la legislación española antes de septiembre de 2018.
- Estándar europeo que especifica los requisitos funcionales de accesibilidad de los productos y servicios TIC: equivalentes al nivel **AA de las WCAG 2.0** .

--

Leer :

- Esquema de los requisitos de la EN 301 549 (2018) aplicables a sitios web, documentos y apps nativas.
Correspondencia con las WCAG 2.1 (nuevo)
http://www.usableyaccesible.com/archivos/relacion_equivalencia_301549_WCAG21.xlsx

Ejercicio 1

-Por parejas intentar analizar un portal público y rellenar la hoja excel.

Seguridad

La seguridad supone un coste económico y de eficiencia.

- El riesgo cero no es práctico
- Hay diversas formas de mitigar el riesgo
- No se puede gastar un millón para proteger un céntimo

--

Tipos de seguridad:

1.Seguridad en el Cliente

2.Seguridad en el Servidor

3.Seguridad en la Comunicación

4.Seguridad en la Aplicación

Seguridad en el Cliente: Código móvil

- Código que circula por la red y se ejecuta en una máquina remota(cliente)
 - Aparece incrustado en un documento HTML. Un cliente de correo o un navegador que cargue el documento lo ejecutará en la máquina cliente.
 - Potencia la funcionalidad de los documentos HTML pero entraña riesgos de seguridad.
 - Un código móvil puede obtener información acerca de un sistema o un usuario y enviarla a una máquina remota.
 - Un método de protección: actualizar el software.
-

Seguridad en el Servidor

El desarrollo de una aplicación web requiere herramientas:

- servidores web,
- servidores de aplicaciones,
- servidores de bases de datos,
- lenguajes de servidor, etc.

Estas herramientas pueden plantear problemas:

- Vulnerabilidades debidas al uso de versiones no actualizadas
- Configuraciones por defecto inadecuadas
- Activación de cuentas por defecto
- Las herramientas deben estar actualizadas y bien configuradas para impedir ataques a la aplicación.

--

1.Servidor Web

- Establecer permisos adecuados para los ficheros del servidor.
- Definir un usuario y grupo especiales (web, www).
- Listado automático de directorios. Puede ser conveniente pero proporciona información sensible.
- Seguimiento de enlaces simbólicos. Peligroso si se enlazan ficheros externos al árbol de documentos.
- Revisar periódicamente los ficheros de log (access_log y error_log en Apache) para detectar posibles ataques.
- Configurar los servicios del servidor HTTP necesarios y deshabilitar los que no se utilizan.

--

2. Servidor de Bases de Datos: Riesgos

- Descubrimiento de información acerca de los datos de conexión al servidor (usuario y clave), información sensible almacenada en la base de datos (tarjetas de crédito...) o información sobre la estructura de la base de datos

- Modificación de las instrucciones SQL enviadas al servidor, construidas de forma dinámica a partir de datos recibidos del usuario y por tanto potencialmente peligrosos (Inyección SQL)
- Acceso no autorizado a información restringida

--

Servidor de Bases de Datos: Protección

- Vigilar la configuración por defecto (evitar BD y usuarios predefinidos)
- No ejecutar el servidor BD como root.
- No dar a usuarios de Web acceso a la tabla de usuarios excepto al administrador.
- Asegurarse el administrador tiene un password seguro
- Restringir el acceso remoto al servidor
- No dar a un usuario más permisos que los estrictamente necesarios
- Almacenar los datos sensibles de forma encriptada

3. Seguridad en la Comunicación: HTTPS

- Proteger la información cuando se envía con protocolos seguros
- SSL (Secure Socket Layer) es el primer protocolo para asegurar el transporte de datos entre el cliente y el servidor web. Diseñado inicialmente por Netscape, hoy día es soportado por la mayoría de los servidores web.
- Podemos reconocer una conexión HTTP sobre SSL porque aparece el prefijo 'HTTPS' en lugar de 'HTTP' en la URL.
- Los datos utilizados con HTTPS son seguros vía **TSL** (Transport Layer Security protocol). Versión actualizada y más segura del SSL.
- HTTPS permite estas capas de protección:
 - encriptación: si algún atacante consigue interceptar esa información, no le servirá para nada ya que no sabrá descifrarla (pero tú sí).
 - integridad de datos: los atacantes no podrán "modificar" el contenido del mensaje enviado.
 - autenticación: se evitan los ataques de suplantación de identidad phishing o intermediarios ("man in the middle ") en el que tu usuario proporciona información a terceros cuando cree que te los está dando a ti.

--

PHP con HTTPS

```
<?php if  
  
( ! isset($_SERVER['HTTPS']) or $_SERVER['HTTPS'] == 'off' )  
  
{ $redirect_url = "https://". $_SERVER['HTTP_HOST'].  
$_SERVER['REQUEST_URI']; header("Location: $redirect_url");  
exit(); }  
  
?>
```

--

5. Lenguajes de programación en servidor/cliente

- Proteger el código fuente para evitar que pueda ser visualizado, especialmente cuando contiene información sensible como pueden ser los datos de conexión al servidor de bases de datos.
- Sacar el código fuente sensible fuera de la raíz de la web, y protegerlo contra lectura.
- Validar las instrucciones SQL antes de enviarlas al servidor.
- No revelar información sobre la base de datos en los mensajes de error (esquema, naturaleza de los datos almacenados, fragmentos SQL).

ClickJacking: robo de clics

- <http://www.elladodelmal.com/2015/08/el-ironframe-para-luchar-contra-ataques.html>
- Cuando el atacante quiere hacer un esquema de ClickJacking, inyecta un iframe en una web vulnerable y en ese iframe incluye la web de la que quiere robar los clics a la víctima.
- **Solución** proteger nuestros clientes: Evitar la inclusión de una web en un iframe por medio de HTTP Headers X-Frame-Options.
- **x-frame-options: DENY | SAMEORIGIN**
- Añadir en fichero .htaccess **Header always append X-Frame-Options SAMEORIGIN**
- Si esto está activo en la consola dará este error al incluir la página en otra web.: Refused to display 'https://www.google.es/maps/@38.9421251,-0.3578288,19z' in a frame because it set 'X-Frame-Options' to 'SAMEORIGIN'.

Seguridad: Top 10

The Open Web Application Security Project (OWASP).

https://www.owasp.org/images/b/b0/OWASP_Top_10_2017_RC2_Final.pdf

- A1-Injection
- A2-Broken Authentication and Session Management
- A3-Cross-Site Scripting (XSS)
- A4-Broken Access Control
- A5-Security Misconfiguration
- A6-Sensitive Data Exposure
- A7-Insufficient Attack Protection
- A8-Cross-Site Request Forgery (CSRF)
- A9-Using Components with Known Vulnerabilities
- A10-Underprotected APIs

--

Ejercicio 3

Define y pon un ejemplo de un ataques del top 10.

Seguridad en WP

Cualquier software es susceptible de ser hackeado, y los plugins de WordPress no son una excepción, por eso es importante desarrollar el software teniendo en cuenta los estándares de seguridad en la programación web, y los que nos ofrece el propio CMS, así como mantener un seguimiento del software para lanzar nuevas versiones en caso de vulnerabilidades.

--









Seguridad en WP

- **Nivel 1. Javascript desde el navegador:** En este nivel validamos los campos que el usuario debe rellenar, formatos numéricos, etc. La validación se hace cuando aún no se ha enviado ningún dato al servidor. Mediante el uso de Javascript en el propio navegador del usuario.
- **Nivel 2. Funciones PHP:** El propio lenguaje de programación PHP nos proporciona herramientas para comprobar la legitimidad de los datos, con funciones como *isset()*, *empty()*, *preg_match()*, etc.
- **Nivel 3: API de WordPress:** Una vez que tenemos los datos del usuario, podemos llevar a cabo acciones de chequeo de usuarios, comprobación de taxonomías, sanitización de cadenas, a través de la familia de funciones *exists()*, *validate()*, *is()*, *sanitize()*, *nonce*()*, etc.

Responsive Web design

Es un diseño web que tiene que diseñarse para:

- todos los navegadores y sistemas,
- todas las resoluciones de pantalla,
- todas las velocidades de conexión.

 <p>Bootstrap </p> <p>Front-End Frameworks</p>	 <p>Foundation  </p> <p>Front-End Frameworks</p>	 <p>Semantic UI  </p> <p>Front-End Frameworks</p>
---	--	---

<https://stackshare.io/stackups/bootstrap-vs-foundation-vs-semantic-ui>

<https://www.slant.co/topics/3522/~fully-featured-responsive-css-frameworks>

--

Responsive Web design

- **Mejora progresiva** : Estrategia que acentúa la accesibilidad, que permite que cada uno tenga acceso al contenido y a la funcionalidad básica de una página web, usando cualquier navegador web o conexión a Internet, mientras que también permite a otros con un mayor ancho de banda o un navegador web más avanzado experimentar una versión mejorada de la página.

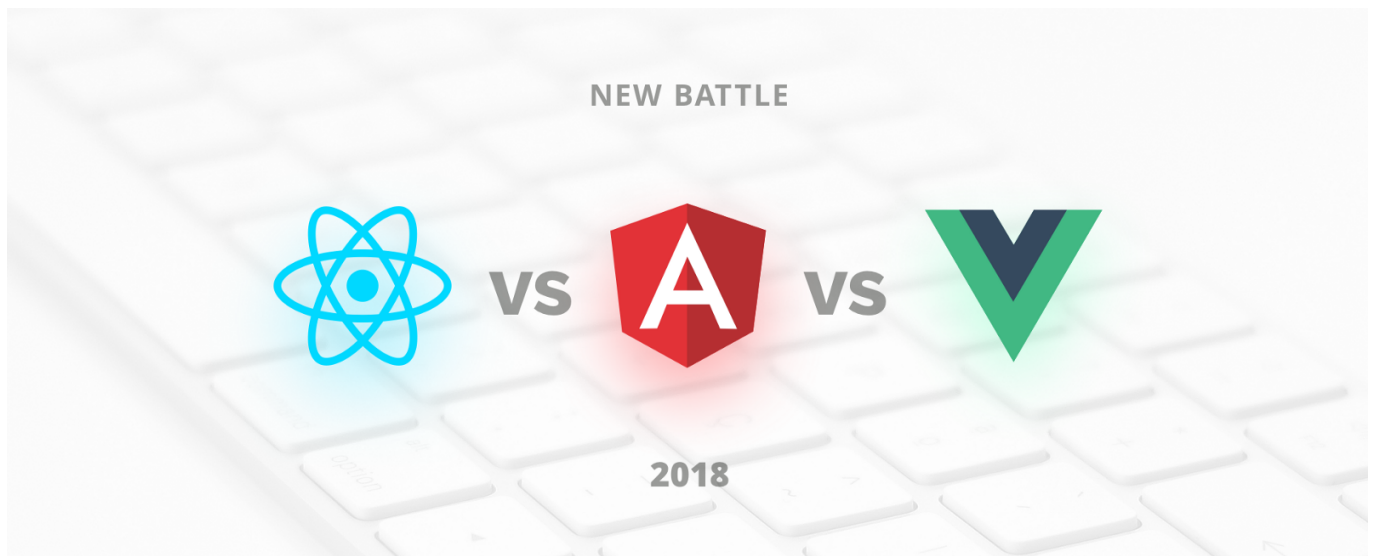
- **Mobile First:** diseñar primero para móviles y después para ordenadores de sobremesa.

Librerías JS para presentaciones

- Flowtime.js
- impress.js
- FormidableLabs/spectacle: utiliza ReactJS
- Reveal.js

Frameworks en el cliente Web

- Angular (TypeScript).
- ReactJS es una librería Javascript desarrollada por Facebook.
- Vue: Esta en fase inicial, parecido al angular. Librería pequeña 20KB.



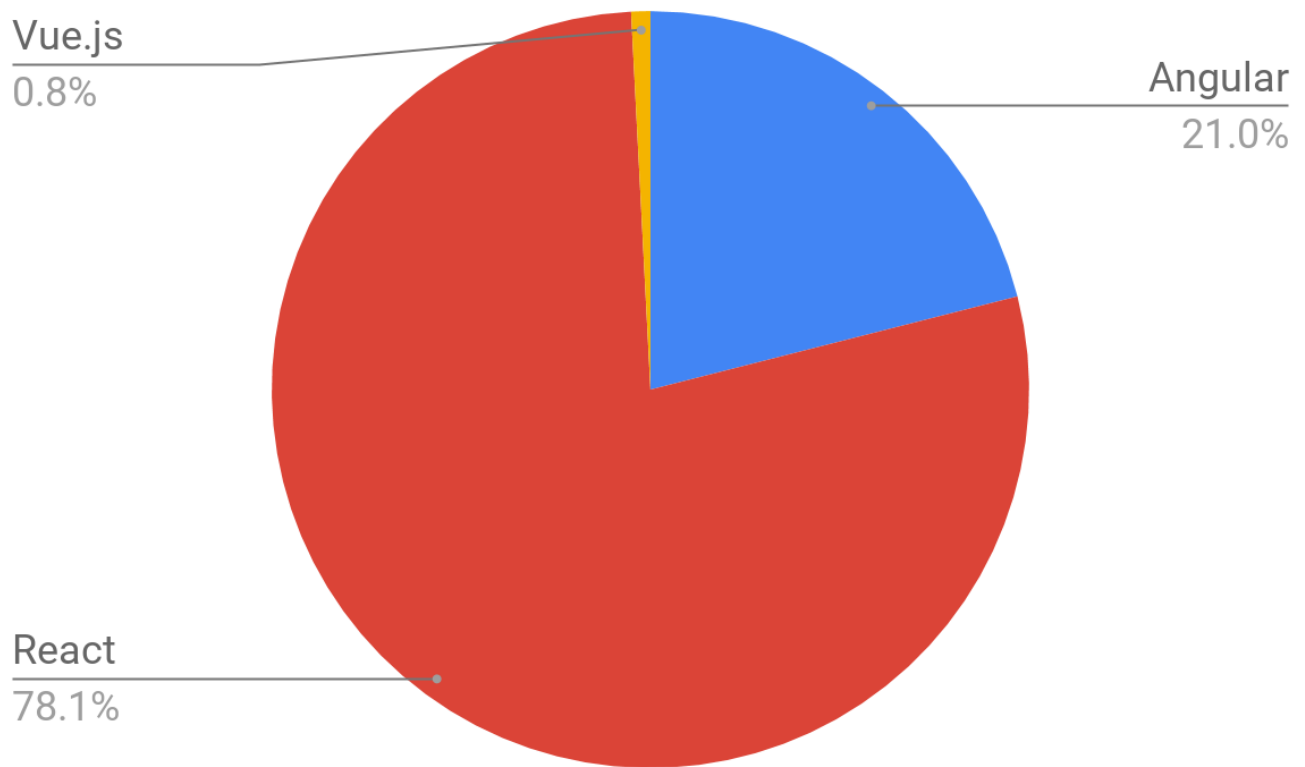
--

Typescript

- TypeScript es un lenguaje de programación libre y de código abierto desarrollado por Microsoft.
- Es un **superconjunto de JavaScript**, que esencialmente añade tipado estático y objetos basados en clases.
- Los navegadores no convertirán TypeScript a Javascript.
- Mediante un Transpiler que convierte de código TypeScript a JavaScript.
- Se puede usar para programar tanto el cliente como el servidor (nodejs).

--

Comparativa:



<https://medium.com/@TechMagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d>

--

¿Dudas?

