Sistemas de Información Basados en Web 2019/2020

Sergio Alonso [Zerjillo]

Dpto. de Lenguajes y Sistemas Informáticos
Universidad de Granada

Febrero - Junio de 2020

¿Qué vamos a cursar?

Sistemas de Información Basados en Web

¿Qué vamos a cursar?

- Introducción al desarrollo de software orientado a la web

Sistemas de Información Basados en Web

¿Qué vamos a cursar?

- Introducción al desarrollo de software orientado a la web
- Reafirmar fundamentos de desarrollo del software (adaptándolo a las peculiaridades de la Web)

Profesorado

- Sergio Alonso: Teoría (viernes) y prácticas (P2 y P3) (miércoles y viernes)

  zerjioi@ugr.es
  https://lsi.ugr.es/lsi/zerjioi

- José María Guirao: Prácticas (P1) (jueves)

  jmguirao@ugr.es
  https://lsi.ugr.es/lsi/jmguirao

Temario de teoría

- **Tema 1.** Introducción a los sistemas de información basados en web.
- **Tema 2.** Tecnologías de desarrollo web.
- **Tema 3.** Análisis y diseño de aplicaciones web.
- **Tema 4.** Gestión de la información.
- **Tema 5.** Estándares y normativas legales aplicables a los entornos web.

En total disponemos de **11** sesiones de teoría.

Temario de prácticas

- **Práctica 1.** `HTML 5 y CSS`.
- **Práctica 2.** `Javascript`.
- **Práctica 3.** Programación en el servidor (I): `PHP`.
- **Práctica 4.** Programación en el servidor (II): `PHP y Conexión a Bases de Datos`.
- **Práctica 5.** Comunicación asíncrona con el servidor: `AJAX`.

En total disponemos de **11** sesiones de prácticas.

Seguimiento de la asignatura

Haremos uso de la plataforma SWAD

Comunicación con los profesores

Preferentemente vía **e-mail**, poniendo en el asunto [SIBW]

Tutorías

¡Están para usarlas!

Material de trabajo

- Transparencias de clase: ¡Son solo un guión!
- Guiones de prácticas

Evaluación Continua (lo que os importa :-))

La asignatura consta de un fuerte componente práctico, por lo que la ponderación de la evaluación final será como sigue:

1. Evaluación trabajo autónomo teórico: 10%
2. Evaluación mediante examen teórico por escrito: 35% *
3. Evaluación trabajo autónomo práctico: 55% *
4. Trabajo de ampliación de conocimientos: 10%

* Para aprobar es requisito fundamental sacar al menos un 35% de la calificación máxima de los puntos marcados y que el total sea mayor o igual a 5.

Calendario prácticas

Grupo	P1		P2	P3			P4			P5	
P1 (Jueves)	20/02	27/02	05/03	12/03	19/03	26/03	16/04	23/04	30/04	07/05	14/05
P2 (Miércoles)	19/02	26/02	04/03	11/03	18/03	25/03	15/04	22/04	29/04	06/05	13/05
P3 (Viernes)	21/02	06/03	13/03	20/03	27/03	03/04	17/04	24/04	08/05	15/05	22/05

Desarrollo de las prácticas

Grupos

Se desarrollarán individualmente, aunque propiciamos que se trabaje "por parejas"

Temática

Se propone realizar una página de *"Gestión de Eventos"*

Entrega y defensa

- Las entregas de las prácticas serán **ANTES** de la siguiente práctica
- La defensa de las prácticas 1, 2 y 3 se hará después de Semana Santa
- La defensa de las prácticas 4 y 5 se hará tras finalizar la última clase práctica

Índice

- Conocer la historia y evolución de Internet y los Sistemas Web
- Identificar las particularidades del software desarrollado como Sistema Web
- Comprender la importancia de los SIBW

Internet

Definición

Internet es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP, lo cual garantiza que las redes físicas heterogéneas que la componen formen una red lógica única de alcance mundial.

Sus orígenes se remontan a 1969, cuando se estableció la primera conexión de computadoras, conocida como ARPANET, entre tres universidades en California (Estados Unidos).

Wikipedia

Enlaces de Interés (historia de Internet)

http://www.isoc-es.org/breve-historia-de-internet/
http://www.walthowe.com/navnet/history.html

Modelo de red de Internet

4 capas

- Capa de aplicación → `DNS` y `http`
- Capa de transporte
- Capa de internet → `IP`
- Capa de acceso a red

Enlaces de Interés

https://searchnetworking.techtarget.com/definition/TCP-IP

https://es.wikipedia.org/wiki/Modelo_TCP/IP

Protocolo IP

Asigna una dirección univoca a cada elemento conectado a la red.

IPv4

- Direcciones de 32 bits (4 números entre 0 y 255 separados por .)
- Fijas o dinámicas
- Se están quedando cortas
- Algunas direcciones reservadas a subredes privadas

IPv6

- Direcciones de 128 bits
- Compatible hacia atrás
- Se suelen expresar las direcciones en hexadecimal separado por :

¿Cuál es mi IP?

http://cualesmiip.com/

Servicio de DNS

Descripción

"Agenda" que asocia nombres inteligibles con direcciones IP.

Características

- Dominios estructurados en árbol (varios niveles, mínimo 2)
- Primer nivel (derecha): `.com`, `.es`, `.org`, `.net`
- Primer nivel de 2 letras: "paises" y "regiones"
- Primer nivel de 3 o más letras: otros usos ¿sabemos algunos?
- Siguientes niveles: divisiones o subdominios. El último (izquierda) suele corresponder a una máquina concreta

Enlaces de Interés

https://www.verisign.com/es_LA/website-presence/online/how-dns-works/index.xhtml

https://en.wikipedia.org/wiki/List_of_Internet_top-level_domains

Protocolo HTTP

HyperText Transfer Protocol

- Define un formato estandar de intercambio de recursos en la web
- Implementa un esquema cliente / servidor

Características

- No está orientado a contexto: cada petición es completamente independiente.
- `HTTP` es de tipo `pull`: solo se inicia la interaccción desde el cliente

Funcionamiento de HTTP

Esquema

1. Cliente quiere un recurso: introduce `URL` en navegador
2. Navegador crea una petición http y la envia al servidor web y puerto correspondientes
3. El servidor analiza la petición y obtiene el recurso solicitado
4. El servidor web construye una respuesta `http` con el recurso
5. El servidor web envía la respuesta al cliente
6. El cliente recibe y procesa la respuesta

Mensajes HTTP

Tipos

- **Peticiones:** Usan un verbo y especifican el recurso
- **Respuestas:** Incluye un código + el recuro solicitado

Verbos HTTP más comunes

- `GET`: Para obtener un recurso
- `POST`: Envia información (puede provocar cambios en el servidor)
- `PUT`: Para añadir recursos
- `DELETE`: Para borrar recursos
- `HEAD`: Igual que `GET` pero obtiene solo la cabecera.

Enlaces de interés

https://www.tutorialspoint.com/http/

https://developer.mozilla.org/es/docs/Web/HTTP/Methods

https://www.tutorialspoint.com/http/http_status_codes.htm

Algunas particularidades

GET

- Se almacenan en historial y caché
- Se pueden marcar como favoritos
- Toda la información de la petición se ve en la URL
- Tiene restricción de longitud

POST

NO se almacenan en historial y caché, **NO** se pueden marcar como favoritos, **NO** tiene restricción de longitud

¿Cual es más seguro?

NINGUNO DE LOS DOS (IGUAL)

Interesante

Herramientas de desarrollador de Firefox

Extensiones de firefox para hacer peticiones `http`: RESTED, RESTClient...

URL

Localizador Uniforme de Recursos

Secuencia de caracteres que permite nombrar recursos en Internet.

Sintaxis

`scheme://[user[:pass]@]host[:port][/directory[/.../]]/file`

[] $\Rightarrow$ opcional

Ejemplos

- http://lsi.ugr.es/lsi/postgrado/mgtpn
- https://webmail.ugr.es/

Servidores web

¿Qué es?

Un servidor web o servidor `HTTP` es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente.

Wikipedia

Ejemplos

- Apache
- Microsoft Internet Information Services
- Nginx
- node.js
- ...

Posible ejercicio de ampliación (subir nota)

Pequeño seminario (pocos minutos) sobre el servicio DNS

- Dominios de primer nivel nuevos, más usados, etc
- Sobre los organismos que se encargan de gestionar los dominios a nivel mundial o a nivel estatal (España, otro país o comparativas)
- Anécdotas de interés sobre dominios
- Consejos prácticos para el uso y contratación de dominios
- Resolución de conflictos en nombres de dominios (.es, .com, otros)
- ...

¿Voluntarios?

Índice

- Identificar los recursos necesarios para la puesta en marcha de un proyecto web
- Conocer tecnologías de desarrollo de SIW
- Conocer la importancia de seguir normas de estilos y estándares
- Ser capaz de decidir y justificar las tecnologías que se usarán

Elección del Dominio

- Facilidad de escritura y lectura
- ¿Caracteres ajenos al inglés?
- Evitar caracteres no alfabéticos
- Evitar palabras que puedan ser confundidas fácilmente
- Usar un *Top Level Domain* de acuerdo con la aplicación

Datos necesarios para el registro de dominio

IMPORTANTE

Al registrar el dominio es importante asegurarnos que aparecemos como propietarios del dominio. ¡Cuidado con los dominios regalados!

Datos a controlar

- **Propietario** (registrant)
- **Contacto administrativo** (administrative contact)
- **Contacto de pago** (billing contact)
- **Contacto técnico** (technical contact) → Mantiene el DNS

Alojamiento web

Cosas que necesitaremos

- Espacio de alojamiento
- Bases de datos
- E-mail
- Ancho de banda

Posibilidades

- Contratar alojamiento compartido
- Contratar un servidor dedicado (virtual o no)
- Usar un servidor propio y alojarlo en un centro de datos (housing)
- Usar un servidor propio y alojarlo en nuestras infraestructuras
- Usar un servidor gratuito

Discusión

¿Ventajas? ¿Inconvenientes?

Tecnologías Web del lado del Cliente

Tecnologías que se usan

- `HTML`: HyperText Markup Language: Estructura y contenido que vamos a mostrar
- `CSS`: Cascade Style Sheets: Estética y formato (independiente del contenido)
- `Javascript`: Código dinámico del lado del cliente

IMPORTANTE

Separar bien las tres componentes.

Javascript

Permite ejecutar código en el cliente (navegador).

Funciones

- Insertar o modificar contenido y estilos en la página web
- Recoger información del navegador y equipo cliente
- Reaccionar a eventos generados en el navegador
- Comunicarse con el servidor sin recargar la página Web (`Ajax`)

Ejemplos de Javascript

```
1  <!DOCTYPE html>
2  <html>
3   <head>
4    <title>Ejemplo</title>
5    </head>
6    <body>
7      <h1>Ejemplo de JavaScript.</h1>
8
9      <script type="text/javascript">
10       alert("Esto es una ventana emergente con mensaje.");
11     </script>
12   </body>
13 </html>
```

Ejemplos de Javascript

```
<!DOCTYPE html>
  <html>
  <head>
    <title>Ejemplo JavaScript 2</title>
  </head>
  <body>
  <h1>Ejemplo de JavaScript - Fecha</h1>

  <p>
    <script type="text/javascript">
      var fechaActual = new Date();
      var mensaje = "Hoy es ";
      document.write (mensaje + fechaActual.toDateString());
    </script>
  </p>
  </body>
</html>
```

Ejemplos de Javascript: Funciones

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo JavaScript - Funciones</title>
    <script>
      function getMensajeConFecha(mensaje) {
        var fechaActual = new Date();
        return mensaje + fechaActual.toDateString();
      }
    </script>
  </head>
  <body>
    <h1>Ejemplo de JavaScript - Funciones</h1>

    <p>
      <script type="text/javascript">
        document.write (getMensajeFecha("Hoy es "));
      </script>
    </p>
  </body>
</html>
```

Ejemplos de Javascript: En fichero aparte

```
1  <head>
2     ...
3
4     <script src="funciones.js"></script>
5
6     ...
7  </head>
```

Ejemplos de Javascript: Usando el DOM

DOM

Document Object Model

```
<body>
  <h1 id="encabezado1" class="grande bonito">Titulillo</h1>
  <p>...</p>


...

var objetoTitulo = document.getElementById("encabezado1");
objetoTitulo.innerHTML = "Bienvenidos todos";

console.log(objetoTitulo.className);
console.log(objetoTitulo.parentElement);
console.log(objetoTitulo.style);
console.log(objetoTitulo.tagName);
console.log(objetoTitulo.parentElement.children);
```

Ejemplos de Javascript: Eventos

```
1  <p> Un enlace <a href="#" onmouseover="alert('Yipi Ka Yei')">
     molesto</a></p>
```

Eventos "comunes"

- `onClick`
- `onMouseOver`
- `onMouseOut`
- `onKeyUp`: Elementos formulario, a
- `onChange`: Elementos seleccionables, cuadros texto
- `onFocus`: elementos formulario, a
- `onLoad`: `body` e `img`

Ejercicio de clase

Conseguir que un gato huya del cursor de nuestro ratón.

jQuery

Biblioteca de Javascript que simplifica la programación y la interoperatibilidad entre navegadores.

Facilita

- Seleccionar elementos del DOM
- Gestionar eventos
- Manipular el DOM
- Manipular CSS
- Hacer llamadas AJAX fácilmente

Posible ejercicio de ampliación (subir nota)

Pequeño seminario sobre `jQuery`

- "Instalación"
- Selectores
- Modificación del `DOM`
- Modificación del `CSS`
- Modificación del `CSS`
- No se hará (todavía) uso de `Ajax`
- ...

¿Voluntarios?

Asynchronous Javascript and XML

Técnica de desarrollo web para crear *Rich Inernet Applications*. Una parte muy importante de la aplicación se ejecuta en el **cliente**.

Características - ¿Ventajas? ¿Inconvenientes?

- Asíncrono
- Intercambio de información: `XML` (o `JSON`)
- Interfaz de petición: `XMLHttpRequest`
- Recomendación: Usar la biblioteca `jQuery`

Ejemplo AJAX: pruebaAjax.html

```
1  <!DOCTYPE html>
2    <html>
3    <head>
4      <title>Ejemplo AJAX</title>
5      <link rel="stylesheet" type="text/css" href="css/
       estiloEjemploAjax.css" />
6      <script src="https://ajax.googleapis.com/ajax/libs/jquery
       /3.3.1/jquery.min.js"></script>
7      <script src="js/ejemploAjax.js"></script>
8    </head>
9    <body>
10
11   <table id="tabla">
12     <thead><tr><th>Objeto</th><th>Cantidad</th></tr></thead>
13     <tbody><tr><td>...</td><td>...</td></tr></tbody>
14   </table>
15
16   <p><input type="text" id="numElems" value="4" />
17       <input type="button" id="boton" value="Obtener datos" /></p>
18
19   <div id="mensaje">...</div>
20   </body>
21 </html>
```

Ejemplo AJAX: css/estiloEjemploAjax.css

```
#tabla {
   font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
   border-collapse: collapse;
   width: 100%;
}
#tabla td, #tabla th {
   border: 1px solid #383;
   padding: 8px;
}
#tabla tr:nth-child(even){background-color: #f2f2f2;}
#tabla tr:hover {background-color: #ddd;}
#tabla th {
   padding-top: 12px;
   padding-bottom: 12px;
   background-color: #4CAF50;
   color: white;
}
#mensaje {
   background-color: #4CAF50;
   color: white;
   border: 1px solid #383;
   text-align: center;
   display:none;
}
```

Ejemplo AJAX: obtieneDatosAjax.php

```php
<?php
function randomString() {
    $characters = 'abcdefghijklmnopqrstuvwxyz';
    $randstring = '';
    for ($i = 0; $i < 10; $i++) {
        $randstring .= $characters[rand(0, strlen($characters))];
    }
    return $randstring;
}
    header('Content-Type: application/json'); // Devolvemos JSON

    sleep(rand(2, 6)); // Simulamos que la consulta lleva tiempo

    $numElems = (int)$_GET["num"];

    $datos = array();
    for ($i = 0 ; $i < $numElems ; $i++) {
      array_push($datos, ["obj" => randomString(), "cant" => rand
      (1,9)]);
    }
    echo(json_encode($datos));  // codificamos el objeto como JSON
?>
```

Ejemplo AJAX: ejemploAjax.js (1/2)

```
1  $(document).ready(function(){
2    boton.onclick = hacerPeticionAjax;
3  });
4
5  function hacerPeticionAjax() {
6    num = $("#numElems").val();
7
8    $.ajax({
9      data:  {num},
10     url:   'obtieneDatosAjax.php',
11     type:  'get',
12     beforeSend: function () {
13       $("#mensaje").html("Obteniendo datos, espere por favor...");
14       $("#mensaje").show();
15     },
16     success:  function (respuesta) {  // Se ejecuta asincronamente
17       $("#mensaje").hide();           // (no sabemos cuando en el
       futuro)
18       procesaRespuestaAjax(respuesta);
19     }
20   });
21 }
```

Ejemplo AJAX: ejemploAjax.js (2/2)

```
1 function procesaRespuestaAjax(respuesta) {
2   res = "";
3   for (i = 0 ; i < respuesta.length ; i++) {
4     res += "<tr><td>" + respuesta[i].obj + "</td><td>" + respuesta[
      i].cant + "</td></tr>\n";
5   }
6
7   $("#tabla > tbody").html(res);
8 }
```

Tecnologías Web del lado del Servidor

Evolución

Web estática (en su origen)

↓

Web dinámica (generar contenidos dinámicamente)

Cuestión

¿Cómo podemos generar contenido dinámicamente?

Generación dinámica de contenidos

Posibilidades

- Programas ejecutables en el servidor
- Usar lenguajes de scripting

Idea principal

Componer *(controlador)* las páginas web *(vista)* dinámicamente extrayendo la información de una fuente de datos *(modelo)*

A riesgo de ser repetitivos: Hay que implementar bien el esquema **MVC**

Ejecución de programas en el servidor

CGI

Common Gateway Interface

Puesta en funcionamiento

- Se configura el servidor web para que ante una petición se lance un ejecutable
- La salida del ejecutable se manda como respuesta a la petición

¿Ventajas? ¿Inconvenientes?

Ejecución de programas en el servidor

CGI

Common Gateway Interface

Puesta en funcionamiento

- Se configura el servidor web para que ante una petición se lance un ejecutable
- La salida del ejecutable se manda como respuesta a la petición

¿Ventajas? ¿Inconvenientes?

- Ejecutables pueden ser dependientes de la máquina
- Eficiencia si hay que hacer cálculos intensivos
- "Dificultad" de programación ("bajo nivel")

Ejecución en el servidor: Python, Java Servlets...

Pequeño seminario sobre `Python` (para web), `Java Servlets`...

- Características
- Algo de sintaxis
- "Instalación"
- Comparativa con otras opciones
- Ventajas, inconvenientes
- ...

¿Voluntarios?

Ejecución en el servidor: Python

Python

Este lenguaje se está usando cada vez más para desarrollo web, usualmente usando algún framework:

- `Flask`
- `Django`
- ...

¿Quieres saber más?

Cursad la asignatura

DAI - Desarrollo de Aplicaciones para Internet

Ejecución en el servidor: Java Servlets

Java Servlets

Programas en Java que se ejecutan en el servidor y atienden peticiones http.

Características

- Heredan de la clase HttpServlet e implementan doGet y/o doPost
- Devuelven como respuesta lo que se envie a un Writer (out.println)
- Usualmente servidos con el servidor Tomcat

Lenguajes de Scripting: PHP, ASP, Perl, JSP

Características

- Código interpretado
- Se puede intercalar en la plantilla de `HTML`
- Por defecto el servidor tiene los scripts en su directorio de páginas web pero si los identifica como scripts (por su extensión) ejecuta el intérprete

¿Ventajas? ¿Inconvenientes?

Lenguajes de Scripting: PHP, ASP, Perl, JSP

Características

- Código interpretado
- Se puede intercalar en la plantilla de `HTML`
- Por defecto el servidor tiene los scripts en su directorio de páginas web pero si los identifica como scripts (por su extensión) ejecuta el intérprete

¿Ventajas? ¿Inconvenientes?

- Más a alto nivel
- Desarrollo "rápido"
- Puede ser dificil implementar bien el **MVC**

PHP: Hypertext Preprocesor

PHP

- Nace (1994) como un preprocesador de páginas web para darles algo de contenido dinámico
- Se ha convertido en uno de los estándares de facto para la web
- Existen cantidad de módulos que le aportan funcionalidad (como enlaces a casi todas las bases de datos usuales)

PHP: Primer ejemplo

```
1  <!doctype html>
2  <html>
3    <head>
4      <title>Ejemplo PHP</title>
5    </head>
6    <body>
7      <?php
8        echo("<h1>Hola Mundo</h1>");
9      ?>
10   </body>
11 </html>
```

ACLARACIÓN

El código `PHP` se ejecuta en el **SERVIDOR**. El `HTML` se interpreta en el **CLIENTE**. El código `Javascript` se ejecuta en el **CLIENTE**.

PHP: Sintaxis y otros

Características

- Código PHP siempre incrustado en la etiqueta `<?php ... ?>`
- Variables precedidas de $
- Variables sin tipo preestablecido: ¿Bueno? ¿Malo?
- ¿Sensible a las mayúsculas?: **SI** para variables y constantes. **NO** para funciones, clases...
- Existen algunas variables globales importantes (sobre el entorno o la petición HTTP): `$_GET`, `$_POST`, `$_SESSION`, , `$_ENV`...
- `include()` y `require()`
- **Matrices:** Son en realidad un mapa ordenado que puede ser usado como matriz, lista (vector), pila, cola , diccionario...

```php
$aux = array(
   clave => valor,
   clave2 => valor2,
   ...)
```

```php
$aux = [
"foo" => "bar",
"bar" => "foo",
];
```

```php
echo($aux[0]);

echo($aux["bar"]);
```

PHP: Pistas para desarrollo eficiente con PHP

Separación MVC

Al ser un lenguaje que se puede intercalar en el `HTML` puede ser difícil implementar MVC

Buenas prácticas

- Separar completamente la **Vista** del **Controlador**:
  - Los ficheros de la vista (los que contienen `HTML` deben tener el mínimo `PHP` posible)
  - Los ficheros del controlador NO DEBEN TENER nada de `HTML`
- Separar completamente el **Modelo** del **Controlador**:
  - El modelo encapsula toda la interacción con la fuente de datos (BD)
  - En el controlador no debería aparecer ni una llamada directa a la fuente de datos
  - Una buena posibilidad puede ser usar un ORM (o al menos un acceso a BD orientado a objetos)

Uso de un motor de plantillas (Twig)

Twig

Para una mejor separación de la **Vista** y el **Controlador** es muy recomendable usar un motor de plantillas como Twig

Características

- Lenguaje de plantillas reducido y simple (apto para diseñadores)
- Fácil aprendizaje
- Previene malas prácticas
- Libre (y gratuito)

Ejemplo básico de Twig

templates/pruebaTemplate.html

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <title>Ejemplo Twig</title>
  </head>
  <body>
    <p>{{nombre}} tiene edad {{edad}}.</p>
  </body>
</html>
```

```
<?php
require_once 'vendor/autoload.php'; # Carga la biblioteca Twig

$loader = new \Twig\Loader\FilesystemLoader('templates');
$twig = new \Twig\Environment($loader,[ ]);

echo $twig->render('pruebaTemplate.html',
          ['nombre' => 'Espinete', 'edad' => 'Indefinida']);
?>
```

Twig: filtros

```
1  {{ 'TEXTO MAL FORMATEADO'|title }}
2  {# Salida: Texto Mal Formateado #}
3
4
5  {{ ['Pepe', 'Maria', 'Juanjo']|join(', ', ' y ') }}
6  {# Salida: Pepe, Maria y Juanjo #}
7
8
9  {{ "Un flotante: %.2f"|format(123.4567) }}
10 {# Salida: Un flotante 123.46 #}
```

Twig: herencia

Herencia

- Permite reutilizar templates y partes de templates
- Creamos una estructura de "bloques" que podemos redefinir en los hijos
- Con la función `parent()` podemos renderizar el bloque padre

Twig: herencia (ejemplo)

templates/templatePadre.html:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <title>{% block titulo %}XXXX{% endblock %} - Ejemplo Twig</title>
  </head>
  <body>
    <header>
      {% block cabecera %}
        <h1>Chorizos de Cantimpalo</h1>
      {% endblock %}
    </header>

    <main>
      {% block principal %}
        <p>Se venden <del>politicos</del> chorizos.</p>
      {% endblock %}
    </main>

    <footer>
      {% block pie %}
        <p>&copy; 2019 Todos los chorizos reservados</p>
      {% endblock %}
    </footer>
  </body>
</html>
```

Twig: herencia (continuación ejemplo)

templates/templateHijo.html:

```
1 {% extends "templatePadre.html" %}
2
3 {% block titulo %}Pagina hija{% endblock %}
4
5 {% block principal %}
6   {{ parent() }}
7
8   <p>Y otros embutidos.</p>
9 {% endblock %}
```

Jugando un poco con el servidor web

Por defecto `Apache` estará sirviendo todos los ficheros y directorios en nuestra carpeta base. Por ej. si queremos evitar servir la carpeta `templates`:
Añadimos a `/etc/apache2/sites-available/000-default.conf`:

```
<Directory /vagrant/html>
    AllowOverride all
</Directory>
```

```
MV> sudo service apache2 restart
```

Ya podemos crear ficheros de configuración en cada carpeta de nuestro sitio web. Por ejemplo `/vagrant/html/.htaccess`:

```
RedirectMatch 404 /templates
```

Redirigimos cualquier acceso a la carpeta `/templates` a un error `404 - Not Found`.

Cambiando el sistema de gestión de URLs

Activamos el módulo `Rewrite`:

```
MV> sudo a2enmod rewrite
MV> sudo service apache2 restart
```

En el `.htaccess`:

```
RewriteEngine On
RewriteBase /
Options -Indexes
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ /index.php [L]
```

Con eso cualquier URL que no sea un fichero en el servidor se redirige a `index.php`. Allí:

```
$uri = $_SERVER['REQUEST_URI'];
if (strcmp($uri, "/evento") == 0) {
  // ejecutamos el codigo de evento
}
```

Otros lenguajes de Scripting: ASP, ASP.net

Active Server Pages

Tecnología propietaria de Microsoft. Pensado originalmente para correr sobre IIS. Originalmente basado en Visual Basic. Actualmente cualquier lenguaje .NET.

Características

- Se compila el lenguaje a un lenguaje intermedio (Intermediate Language - IL).
- De IL se compila a código máquina con un compilador JIT.
- El código intermedio impone una serie de restricciones a los lenguajes .NET que se usen.

Otros lenguajes de Scripting: JSP

Java Server Pages

Permite insertar código `Java` en el `HTML` usando la etiqueta <% ... %>.

Características

- Usa internamente `servlets`
- El servidor `Tomcat` transforma la página `JSP` a un `servlet` y lo ejecuta.
- El `servlet` compilado se cachea.

Otros lenguajes de Scripting: ASP, ASP.net, Java Server Pages

Pequeño seminario sobre `ASP`, `ASP.net`, `Java Server Pages` (`JSP`)

- Características
- Algo de sintaxis
- "Instalación"
- Comparativa con otras opciones
- Ventajas, inconvenientes
- ...

¿Voluntarios?

SOA: Arquitecturas Orientadas a Servicios

Estandar para publicar y usar servicios

Características

- Los servicios intercambian mensajes XML sobre http
- Busca mínimo acoplamiento
- SOAP: Simple Object Access Protocol. 3 actores: proveedor, consumidor y publicador

REST: Representational State Transfer

REST: Representations State Transfer

Tipo de arquitectura de software.

Características

- Protocolo cliente / servidor **sin estado**
- Sus operaciones están bien definidas (CRUD sobre POST, GET, PUT, DELETE explícitamente)
- Sintaxis universal a través de la URI
- Tipo de información enviado: usualmente XML o JSON.

Índice

- Conocer las peculiaridades de los sistemas web y adaptar las metodologías de desarrollo a los mismos
- Ser capaz de conceptualizar en diagramas y documentos los requisitos de un SIBW
- Conocer distintas arquitecturas y paradigmas de diseño de sistemas web y aplicarlos en ejemplos concretos

Ingeniería de Requisitos (IR)

Actividades que llevan a la especificación de las necesidades de usuarios y departamentos interesados así como las restricciones que recaen sobre dicho sistema.

¡Importante!

Normalemente el sistema web tendrá que interaccionar con otros sistemas (otro software), o influirá en la manera en que trabajarán los empleados / departamentos...

Leyes no escritas (pero reales) de la IR

- La especificación de requisitos es un problema de comunicación
- Los interesados y el equipo de desarrollo no escriben correctamente lo que entienden como requisitos
- La validación de los requisitos se produce tarde
- El usuario ve el producto final tarde
- Los requisitos suelen ser inestables y variables (muy malo para el desarrollador)

Requisitos funcionales (1/2)

Requisitos funcionales (RF)

Aquellos que determinan que debe hacer el software, los servicios que debe prestar y a que datos debe reaccionar.

Por ejemplo

- Elementos de entrada que tendrá el interfaz web
- Que funcionalidad tendrá
- Como se vinculan las páginas de información entre sí

Requisitos funcionales (2/2)

En general pueden referirse a:

- *Requisitos de la organización*: Diferentes puntos de vista de la organización o entorno donde se implantará la solución
- *Requisitos del dominio de la aplicación*: Funcionalidad en sí de la aplicación: Contenido de la información, flujo de la información, estructura de la información
- *Requisitos de navegación*: Como se pasa de entre distintos elementos de información
- *Requisitos de interacción*: Lo relativo a la interfaz del usuario

Requisitos **no** funcionales

Requisitos no funcionales (RNF)

Restricciones o condiciones que se imponen al sistema que no tienen que ver con la funcionalidad (también llamados *Atributos de calidad*).

Por ejemplo

- *Requisitos del producto*: memoria que se puede usar o plataforma sobre la que debe correr, navegadores soportados, resoluciones de pantalla...
- *Requisitos organizacionales*: impuestos por el desarrollador, como por ejemplo lenguajes a usar, estándares de desarrollo, metodologías...
- *Requisitos externos*: normativas legales aplicables, interacción con sistemas externos...

Especificación de requisitos: ejemplos

Clasificar si es RF o RNF:

- "En las oficinas tenemos W7, W8 y W10 y los usuarios usan tanto Firefox como Chrome"
- "La gestión de pagos con tarjeta se hace con varios bancos distintos"
- "En 4 meses empieza la temporada fuerte. Necesitamos el producto para ayer"
- "Esta es una muestra de los listados que tiene que ofrecer la aplicación"
- "Debe funcionar bien en móviles, ordenadores y tabletas"
- "Cuando el cliente hace un pedido debe llegar un mensaje al departamento de pedidos y otro al de cobros"
- "Los productos deben poder clasificarse en familias y sub-familias"

Extracción de requisitos, algunos mecanismos (1/2)

Estudio de viabilidad

Indicado para sistemas novedosos (las posibilidades de fracaso son mas altos). Imprescindible contar con un especialista que conozca el negocio (clientes, canales de dsitribución, flujo de la información, interacción de los clientes y el prestador de servicios...)

Entrevistas

Fundamentales. No solo con el dueño o jefe, sino con personal de todos los estamentos e incluso con clientes (top-down). Hay que prepararlas y son mejores las preguntas abiertas.

Extracción de requisitos, algunos mecanismos (2/2)

Desarrollo conjunto de aplicaciones

Sesiones conjuntas entre usuarios y analistas. Aprovechar dinámicas de grupo ara obtener requisitos conjuntamente.

Observación directa - etnografía

Si ya hay un sistema funcionando observar directamente su funcionamiento para entender como se llevan a cabo las tareas en la actualidad. Se analiza no solo el programa sino la interacción con las personas que lo usan y las interacciones entre las personas. ¡Importante!: Dejar constancia de que no es una evaluación del personal para que el trabajo se desarrolle como normalmente.

Elaboración del documento de requisitos

Una vez obtenida la información hay que clasificar, ordenar, priorizar, documentar y especificar los requisitos.
¡Ojo!: No es trivial y lleva tiempo, pero es importante.

Posible estructura del documento

- Resumen, Autor
- Descripción de escenarios comunes
- Especificaciones detalladas pantalla a pantalla
- Requisitos no incluidos y cuestiones por resolver
- Ideas para el diseño
- Trabajos futuros
- Requisitos funcionales del usuario
- Requisitos funcionales del dominio del problema
- Requisitos no funcionales

Negociación y validación de requisitos

- Es fundamental **negociar** y **validar** los requisitos con el cliente. Normalmente querrá que se satisfagan todos, pero a un coste limitado.

- Puede ser necesaria la firma de un contrato con los mismos: se resalta la importancia de los mismos y se concreta exactamente el alcance del proyecto. Se evitan malentendidos por ambas partes

- Puede ser interesante ofrecer información sobre precio y obligatoriedad o no de cada requisito.

Ejemplos de dificultades en la extracción de requisitos

Diferenciar políticas de empresa con requisitos

"los prestamos se conceden a periodos entre 6 meses y 30 años"

$\Downarrow$

"el sistema debe permitir el periodo de amortización de prestamo, que será de duración finita pero variable"

Diferenciar entre requisito y opciones de implementación

"Los tipos de vehículo aparecerán en un desplegable"

$\Downarrow$

"Se podrá seleccionar el tipo de vehículo"

Más dificultades en la especificación de requisitos

- *Límites difusos del sistema*: El cliente irá añadiendo requisitos según describe el problema
- *Diferencias de dominio entre el cliente y el desarrollador*: Cada uno es experto en su ámbito: dificultad en la comprensión
- *Volatilidad*: Los requisitos cambian. Hay que aprender a gestionar esos cambios
- *Problemas no tecnológicos*: Cambios de interlocutor, políticas de empresa, de prioridades...

Discusión: ¿Cómo calcular el precio del sistema web a desarrollar?

?¿?¿?¿?¿?

Como curiosidad (¡no le hagáis caso!): How Much Does a Website Cost?

Diseño de Aplicaciones Web

Una vez realizada una especificación de requisitos hay que refinar las abstracciones identificadas en la fase previa para organizar los datos, la navegación, la presentación y la arquitectura de la aplicación.

Peculiaridades de las aplicaciones web

- Mayor accesibilidad de la información y servicios (+usuarios simultaneos)
- Interfaz orientada al documento
- Variedad de tecnologías de gestión, acceso y procesamiento de datos
- Variedad de tecnologías y motores de visualización
- Aquitectura compleja (sistema distribuido, balanceo de carga...)

Diseño de flujos de trabajo

Usaremos herramientas conocidas de ingeniería del software:

Lenguaje Unificado de Modelado (UML)

- Diagramas de flujo
- Diagramas de secuencia

Diseño de datos

Diagrama de clases - UML

- En el caso de basos de datos relacionales la conversión es directa
- En otros tipos de bases de datos hay que trabajar la conversión

Diseño de la navegación

Estructurar las rutas de navegación a través de la información y servicios ofrecidos por nuestro sistema de información web.

Aspectos fundamentales

- La estructura del sitio
- Comportamiento del usuario al navegar (acciones del usuario y eventos que desencadena)

Diseño de la estructura del sitio: IFML

Componentes de cualquier lenguaje modelado del sitio

- *Ítems atómicos:* Elementos de información con instancias de entidades de datos
- *Items compuestos:* Estructuras compuestas de varios ítems atómicos
- *Estructuras contextuales:* estructuras de navegación para acceder a los items (menús, índices, metaetiquetas...)

Interaction Flow Modeling Language - IFML

Estandar para llevar a cabo el diseño estructural de aplicaciones (no necesariamente web).
Si seguimos el esquema *Modelo, Vista, Controlador* fundamentalmente se encarga de la parte de la *Vista*, pero está claro que depende y hace referencia al modelo y al controlador.

Modelado con IFML

Abordamos las siguientes perspectivas:

- Especificación de la estructura de la vista (contenedores, relaciones entre ellos...)
- Especificación del contenido de la vista (contenido de las vistas)
- Especificación de eventos (de los que pueden afectar al estado del interfaz)
- Especificación de transición de eventos (efectos de los eventos sobre el interfaz)
- Especificación de los parametros de conexión (dependencias de entrada/salida entre los componentes de la vista y las acciones)

Diagramas IFML

- Consisten en uno o más contenedores de vista de alto nivel
- Internamente puede estructurarse como una jerarquía de sub-contenedores
- Los contenedores pueden contener componentes de vista (contenido o elementos de entrada de información)
- Los componentes tienen parametros de entrada y salida. Por ejemplo:
  - Un componente que muestra las propiedades de un objeto tiene como entrada el ID del objeto
  - Un formulario tiene como parámetros de salida los valores introducidos
- Los contenedores y componentes se puede asociar con eventos (indica la interacción con el usuario)

Más sobre IFML

Para más información y ejemplos consultar:

- Extracto sobre IFML en Prado
- IFML Examples
- IFML - The interaction flow modeling language, the OMG standard for UI modeling. An intro
- IFML Quick Reference Card

Diseño de la Adaptación

Las aplicaciones web serán usadas probablemente desde diversas partes del mundo y bajo muy distintos dispositivos. La adaptabilidad de las mismas es por tanto fundamental. Normalmente la adaptabilidad se refiere a tres características:

- *Localización* e *internacionalización*: Idioma, moneda, horas y fechas, aspectos culturales
- *Personalización* y *adaptación*: Ajustes para cada uno de los usuarios
- *Accesibilidad*: Ajustes para las personas con discapacidades.

Localización e Internacionalización

Locale

Región geográfica en la que sus habitantes comparten idioma y valores comunes.

Internacionalización (i18n)

Identificación y separación de todos los elementos específicos que componen los `locale`.

Localización (l10n)

Adaptación concreta de los elementos identificados en la `i18n` para un `locale` concreto.

Ejemplos de aspectos a tener en cuenta para la `l10n`

- Idioma (incluso dialectos)
- Unidades de medida
- Moneda
- Fechas
- Cantidades
- Direcciones
- Metáforas
- Colores
- Referencias históricas
- Gestos, aspecto de las personas...

Localización e Internacionalización en PHP

Varias posibilidades:

- Crear varias versiones de las páginas con los textos traducidos (**¡NO!**)
- Usar ficheros de texto con las traducciones y cargar el oportuno
- Utilizar `gettext`: Manual `gettext` y `gettext en PHP`
- No reinventar la rueda y usar un motor de plantillas que soporte la internacionalización: `i18n extension for Twig`

Personalización y adaptación

Adaptaciones que se suelen encontrar en aplicaciones web

- Adaptación de contenidos
- Acciones de navegación automática
- Adaptación de la estructura de navegación
- Adaptación del *layout*

Índice

- Conocer diversas tecnologías para almacenar la información en plataformas Web
- Ser capaz de interpretar documentos `XML` y valorar su uso en soluciones Web
- Conocer los aspectos básicos de la Web Semántica y su influencia en los sistemas de información basados en Web

Tipos de información en la WWW

Tipos de Información

- **Datos desestructurados:** Archivos de texto, vídeo, audio, imágenes, presentaciones... Dificil tratamiento y almacenaje en BD
- **Datos semi-estructurados:** Con cierta estructura, pero no muy rígida. Ejemplo: Curriculum Vitae
- **Datos estructurados:** Todos los registros con una estructura fija y fuertemente tipada

Gestión de Datos Estructurados

Opción típica

Modelo relacional: Los *Sistemas de Gestión de Bases de Datos* (SGBD) relacionales usualmente implementan la mayoría de las funcionalidades necesarias.

Alternativas existentes

- `MySQL`
- `PostgreSQL`
- `SQLite`
- `Oracle`

Algunos detalles interesantes sobre MySQL, seguridad y PHP

Seguridad en bases de datos

- PHP: Database Security
- PHP: Inyección de SQL

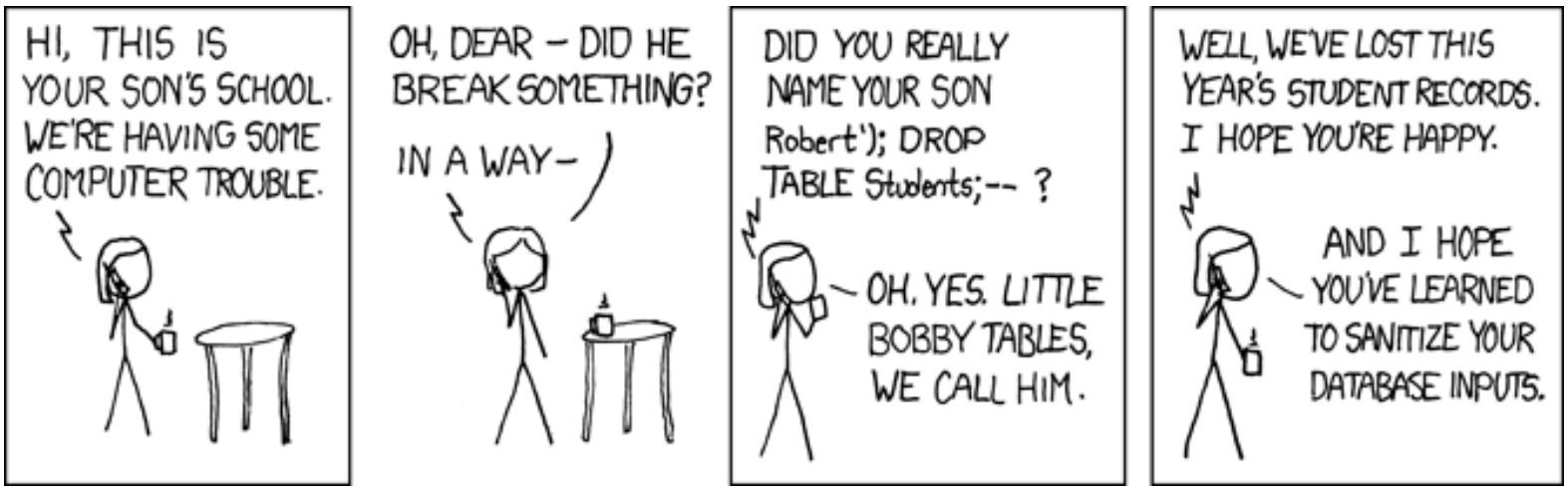

Imagen de xkcd

Conexiones persistentes... ¿si o no?

- PHP: Conexiones persistentes a bases de datos

Gestión de datos semiestructurados

Dificultad

La gestión de datos semiestructurados puede ser más compleja que los datos estructurados.

Posibles formatos para datos semiestructurados

- `json`
- `XML`

SGBD para datos desestructurados: `noSQL`

- `json` → `MongoDB`
- tipo clave-valor → `Cassandra`, `BigTable`...
- ...

Definición

Metalenguaje de marcas para definir lenguajes de marcas, simplificación de SGML.

Lenguajes derivados de XML

- XHTML
- Google Site Map
- NewsML: News Markup Language
- OOXML: Office Open XML → (¡muy criticado!)
- ...

Sintaxis de XML (1/5)

Se forma como una jerarquía de contenidos (árbol):

- etiquetado,
- ilimitado y
- ordenado

Elementos y texto

Elemento: Etiqueta de apertura, contenido textual y etiqueta de fin:

```
1  <etiqueta> contenido textual </etiqueta>
2  <etiqueta />
```

Atributos: En la etiqueta de apertura pueden aparecer pares clave-valor:

```
1  <persona nombre="Zerjillo" apellidos="Alonso" />
```

Importante: Los atributos no están ordenados, el contenido si. El contenido pueden ser sub-árboles. Los atributos pueden ser complejos.

Sintaxis de XML (2/5)

Cabeceras y Entidades

Prologo: Versión de XML, codificación, localización de recursos externos:

```
1 <?xml version="1.0" encoding="utf-8" ?>
```

Entidades: Referencias o contenidos asignados a constantes:

```
1 <!ENTITY tema1 "Tema 1: Intro " >
2 <!ENTITY tema2 SYSTEM "tema2.xml" >
3 <report> &tema1; &tema2; </report>
```

Sintaxis de XML (3/5)

Document Type Definition: DTD

Mecanismo simple de definición de documentos XML. DTD Tutorial

Ejemplo

```
<?xml version="1.0"?>

<!DOCTYPE email [
  <!ELEMENT email (header, body)>
  <!ELEMENT header (from, to, cc?)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT cc (#PCDATA)>
  <!ELEMENT body (paragraph*)>
  <!ELEMENT paragraph (#PCDATA)>
]>

<email>
  <header>
    <from>...</from>
    <to>...</to>
  </header>
  <body>
    <paragraph>Hola</paragraph>
    <paragraph>Caracola</paragraph>
  </body>
</email>
```

Sintaxis de XML (4/5)

XML-Schema

Lenguaje de descripción de tipos XML propuesto por el W3C. Muy versatil pero bastante más complicado que los DTDs. XML Schema tutorial

Ejemplo

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="book">
    <xs:complexType>
    <xs:sequence>
      <xs:element name="title" type="xs:string"/>
      <xs:element name="author" type="xs:string"/>
      <xs:element name="character" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType> <xs:sequence>
          <xs:element name="name" type="xs:string"/>
          <xs:element name="age" type="xs:integer"/>
          <xs:element name="since" type="xs:date"/>
          <xs:element name="qualification" type="xs:string"/>
        </xs:sequence> </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="isbn" type="xs:string"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Sintaxis de XML (5/5)

Continuación del ejemplo XML Schema

Permiten definición de tipos:

```
1  <name>Yo</name>
2  <age>34</age>
3  <since>1968-03-27</since>
```

Permiten la definición de atributos:

```
1  <book isbn="314-2322-22">...</book>
```

Un último ejemplo de XML: MathML

MathML

Lenguaje para describir fórmulas matemáticas. Wikipedia. Editor Online

Ejemplo

$$x^2 + 4 \cdot x + 7$$

```
1  <?xml version="1.0" ?>
2  <math xmlns="http://www.w3.org/1998/Math/MathML">
3    <apply>
4      <plus/>
5      <apply>
6        <power/>
7        <ci>x</ci>
8        <cn>2</cn>
9      </apply>
10     <apply>
11       <times/>
12       <cn>4</cn>
13       <ci>x</ci>
14     </apply>
15     <cn>7</cn>
16   </apply>
17 </math>
```

¿Cómo sería $x^{a+2wz} + y$?

eXtensible Stylesheet Language: XSL

XSL

Para darle estilo a los documentos XML

Partes de las que consta

- XSLT: Lenguaje para transformar XML
- Xpath: Para navegador por los documentos XML
- XSL-FO: Para formatear los documentos XML

La web semántica

Definición

La *web semántica* es un conjunto de actividades desarrolladas en el seno de W3C con tendencia a la creación de tecnologías para publicar datos legibles por aplicaciones informáticas. Se basa en la idea de añadir metadatos semánticos y ontológicos a la WWW. Esas informaciones adicionales -que describen el contenido, el significado y la relación de los datos- se deben proporcionar de manera formal, para que así sea posible evaluarlas automáticamente por máquinas de procesamiento. Wikipedia

Ontología

Descripción formal que proporciona a los usuarios humanos un conocimiento compartido sobre un dominio concreto. Es una definición formal de tipos, propiedades, y relaciones entre entidades que realmente o fundamentalmente existen para un dominio de discurso en particular. Wikipedia

Ontologías

Útiles para

- Organizar datos
- Mejorar las búsquedas
- Integrar información

Ejemplo: Ontología "Universidad"

- **Clases:** `:Profesor`, `:Alumno`, `:Asignatura`, `:Departamento`
- **Instancias de clases:** `:Zerjillo` es instancia de `:Profesor`
- **Relaciones:** `:Imparte(:Zerjillo, :SIBW)`
- **Herencia:** `:Profesor` es subclase de `:Personal`
- **Restricciones:** no `:Imparte(:Alumno, :Asignatura)`
- **Restric. de cardinalidad:** `:Departamento` solo tiene un `:Director`

Lenguajes para la web semántica

Resource Description Framework: RDF

Familia de especificaciones de la W3C originalmente diseñado como un modelo de datos para metadatos. Permite describir anotaciones sobre recursos web (asociados a una URI) Wikipedia

RDF Schema

Extensión semántica de RDF. Un lenguaje primitivo de ontologías que proporciona los elementos básicos para la descripción de vocabularios. Wikipedia

Web Ontology Language: OWL

Lenguaje de marcado para publicar y compartir datos usando ontologías en la WWW. OWL tiene como objetivo facilitar un modelo de marcado construido sobre RDF y codificado en XML. Wikipedia

Gestión de datos desestructurados: búsqueda de información

Problemática

Es dificil gestionar fuentes de información tan variada como la que hay en general en la web.

Buscadores

¿Cómo se las apañan los buscadores?

Rastreadores (*Crawlers* o *Spiders*)

Definición

Es un robot que navega por las webs indexando y clasificando los contenidos de las mismas. En el fondo hacen una búsqueda en un grafo (por anchura, por profundidad o esquemas mixtos).

Mapas del sitio (*sitemap*)

Archivos `XML` que describen de manera jerárquica la estructura del sitio. Facilitan la vida a los buscadores. Google: sobre los *sitemaps*

Limitaciones a los buscadores: `robots.txt`

Se puede pedir a los buscadores que partes de nuestro sitio (o completo) no sea indexado. Google: sobre `robots.txt`

Buscadores: procesamiento de texto

Fases habituales

- **Tokenización:** Extraer palabras
- **Limpieza:** Eliminar tokens no útiles
- **Análisis semántico:** se relacionan términos similares
- **Indexación:** Asociar términos de búsqueda y términos en el artículo
- Evaluar la relevancia del artículo frente a las palabras de búsqueda

Técnicas SEO (Search Engine Optimization)

Conjunto de acciones orientadas a mejorar el posicionamiento de un sitio web en la lista de resultados de los buscadores de internet. Trabaja aspectos técnicos como la optimización de la estructura y los metadatos de una web, pero también se aplica a nivel de contenidos, con el objetivo de volverlos más útiles y relevantes para los usuarios. Wikipedia

¿Opiniones?

Índice

- Conocer el marco legal aplicable a los Sistemas de Información Basados en Web

- Comprender la necesidad del seguimiento de las recomendaciones y normativas de accesibilidad y usabilidad

Normativas de accesibilidad web (WCAG)

Accesibilidad web

Conseguir que todo tipo de usuarios (incluidos los que tengan alguna discapacidad) puedan percibir, entender, navegar e interactuar con el sistema web.

En España

- **Real Decreto 1494/2007 (12 noviembre):** Reglamento sobre las condiciones básicas para el acceso a las personas con discapacidad a las tecnologías, productos y servicios relacionados con la sociedad de la información y medios de comunicación social
- **Ley 56/2007 (28 diciembre):** Medidas de Impulso de la Sociedad de la Información

Pautas de Accesibilidad para el Contenido Web (WCAG 2.0)

- Promovidas por un subgrupo del W3C.
- Evolución de la versión 1.0 donde la principal diferencia es la "norma" de neutralidad tecnológica, es decir, que estas pautas se puedan (y deban) aplicar a cualqueir tecnología (incluso totalmente ajenas al W3C).
- Los criterios que se presentan son verificables (bien automáticamente, bien manualmente).

Principios de accesibilidad

Cualquier usuario debe obtener un contenido:

1. **Perceptible:** Por al menos un sentido
2. **Operable:** Que se pueda interaccionar con el contenido
3. **Comprensible:** Legible, entendible, predecible
4. **Robusto:** Contenido suficientemente descrito para poder ser leido con distintos lectores y tecnologías de asistencia

Pautas generales de accesibilidad (1/2)

Perceptibilidad

- **Pauta 1.1: Alternativas textuales** para cualquier contenido no textual
- **Pauta 1.2: Alternativas para multimedia tempo-dependientes**
- **Pauta 1.3: Adaptable,** pero sin perder información o estructura
- **Pauta 1.4: Distinguible (vista y oído)** incluyendo distinción entre lo más y menos importante

Operatibilidad

- **Pauta 2.1: Acceso mediante teclado**
- **Pauta 2.2: Suficiente tiempo**
- **Pauta 2.3: No destellos** (evitar ataques epilépticos)
- **Pauta 2.4: Navegable:** Medios que ayuden a navegar, localizar el contenido y determinar dónde se encuentran

Pautas generales de accesibilidad (2/2)

Comprensibilidad

- **Pauta 3.1: Legible e inteligible**
- **Pauta 3.2: Predecible** (apariencia y operabilidad)
- **Pauta 3.3: Ayuda a la entrada de datos**

Robustez

- **Pauta 4.1: Compatible.** La compatibilidad con los agentes de usuario debe ser máxima (tanto actuales como futuros).

Técnicas para cumplir las pautas

Cada pauta tiene asociadas recomendaciones o técnicas para cumplirlas:

WCAG 2.0 Guidelines

Niveles WCAG 2.0 (1/3)

Criterios de éxito

Se han definido 60 criterios de éxito o puntos de comprobación y verificación.

Nivel de conformidad "A"

Se cumplen los puntos de verificación de prioridad 1.

- no especifican cómo se representa la información
- son razonablemente aplicables a cualquier sitio web
- son comprobables de forma automática. Algunos requieren la evaluación de forma manual. Los criterios de cumplimiento que requieren comprobación manual producen resultados consistentes bajo múltiples verificaciones por personas distintas

Niveles WCAG 2.0 (2/3)

Nivel de conformidad "AA"

Se cumplen los puntos de verificación de prioridad 1 y 2.

- puede requerir que se presente el contenido de una cierta manera
- son razonablemente aplicables a cualquier sitio web
- son comprobables de forma automática. Algunos requieren la evaluación de forma manual. Los criterios de cumplimiento que requieren comprobación manual producen resultados consistentes bajo múltiples verificaciones por personas distintas.

Nivel de conformidad "AAA"

Se cumplen los puntos de verificación de prioridad 1, 2 y 3.

- son criterios que van más allá de los niveles 1 y 2 y pueden aplicarse para hacer sitios accesible a más personas con cualquier discapacidad o un tipo concreto de discapacidad

Niveles WCAG 2.0 (3/3)

Si cumplimos algún nivel WCAG 2.0

Incluiremos una declaración al respecto, incluyendo el logotipo (+ info)
También debemos indicar:

- fecha en que se revisó el cumplimiento
- título, versión y URI de las pautas WCAG 2.0
- nivel de conformidad alcanzado
- alcance (enumeración de las páginas que lo cumplen)
- listado de las tecnologías de las que depende el contenido

Evaluación de la accesibilidad

Pasos a seguir

1. Determinar el alcance de la evaluación
2. Establecer la muestra representativa de las páginas que se van a analizar
3. Evaluación automática
4. Evaluación manual
5. Resumir los problemas. Realizar informe.

Evaluación *manual* de la accesibilidad

- Aplicar un listado de puntos de comprobación de las pautas
- Probar múltiples configuraciones de distintos navegadores existentes
- Técnicas de filtrado:
  - Desactivar imágenes y comprobar textos alternativos
  - Desactivar el sonido y comprobar que el contenido del audio está disponible a través de texto equivalente (subtitulado, trascripción)
  - Aumentar el tamaño de fuente: diseño sigue preciso y usable
  - No es necesario el desplazamiento horizontal con diferentes resoluciones de pantalla y/o con diferentes tamaños de ventana
  - Visualizar en escala de grises y observar si el contraste es suficiente
  - Acceso en la navegación y funcionalidad sólo con teclado
  - Navegar a través de los enlaces y controles de formulario, además comprobar que los vínculos indican claramente el destino
  - Acceso en la navegación y funcionalidad desactivando plugins, scripts...
- Acceder y examinar las páginas con un lector de pantalla y navegadores especiales como sólo texto (toda la información disponible y en un orden lógico significativo)
- Leer y evaluar los contenidos: texto claro, sencillo y adecuado

La protección de datos. Preliminares

Art. 18.4 de la Constitución Española

La ley limitará el uso de la informática para garantizar el honor y la intimidad personal y familiar de los ciudadanos y el pleno ejercicio de sus derechos.

¿Qué entendemos por "protección de datos"?

"el amparo debido a los ciudadanos contra la posible utilización por terceros, en forma no autorizada, de sus datos personales susceptibles de tratamiento, para, de esta forma, confeccionar una información que, identificable con él, afecte a su entorno personal, social o profesional, en los límites de su intimidad"

La Agencia de Protección de Datos

La Agencia de Protección de Datos

Es un ente de Derecho Público, con personalidad jurídica propia y plena capacidad pública y privada. Su finalidad principal es velar por el cumplimiento de la legislación sobre protección de datos personales y controlar su aplicación, en especial en lo relativo a los derechos de información, acceso, rectificación y cancelación de datos.

Funciones:

- Atender las peticiones y reclamaciones presentadas por los afectados
- Proporcionar información acerca de sus derechos
- Ejercer la potestad sancionadora
- Ordenar el cese o inmovilización de los ficheros que proceda
- Inspeccionar los ficheros
- Ejercer el control y adoptar las autorizaciones que procedan para los movimientos internacionales de datos

Ley Orgánica 15/1999. de 13 de diciembre, de Protección de Datos de Carácter Personal (LOPD)

LOPD

Pretende garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor e intimidad personal y familiar.

Definiciones

- Datos de caracter personal
- Tratamiento de datos
- Fichero
- Responsable del fichero
- Encargado del tratamiento
- Afectado o interesado
- Consentimiento del interesado
- Cancelación
- Cesión o comunicación de datos
- Procedimiento de disociación
- Fuentes accesibles al público
- Responsable de seguridad

Clasificación de los datos personales

Datos Personales

- **Públicos:** Conocidos por mucha gente y de fuente o difusión no reconocible: *nombre, apellidos, edad, profesión...*
- **Privados:** Los que en ocasiones concretas estamos obligados a proporcionar
  - **Íntimos:** El individuo puede proteger de su difusión frente a cualquiera pero que, de acuerdo con un fin determinado, esté obligado –por mandato legal- a dar periódica o regularmente en cumplimiento de sus obligaciones cívicas
  - **Secretos (sensibles, sensibilísimos y sensibilidad especial):** El ciudadano no estará obligado a dar a nadie salvo casos excepcionales, expresamente regulados en las leyes
    - **Reservados:** Bajo ningún concepto, ni por ningún motivo, está obligado el titular a darlos a conocer a terceros si no es así su voluntad
    - **Profundos:** Como los reservados pero con algunas excepciones

Medidas de seguridad en función del tipo de datos (1/3)

Nivel alto

- Ideología, afiliación sindical, religión, creencias, origen racial, salud o vida sexual
- Datos recabados para fines policiales sin consentimiento de las personas afectadas
- Derivados de actos de violencia de género

Medidas de seguridad en función del tipo de datos (2/3)

Nivel medio

- Relativos a la comisión de infracciones administrativas o penales.
- Regulados por el artículo 29 de la Ley Orgánica 15/1999 (13 diciembre) (Solvencia Patrimonial y Crédito)
- Aquellos de los que sean responsables las Administraciones Tributarias y se relacionen con el ejercicio de sus potestades tributarias
- Aquellos de los que sean responsables las entidades financieras para finalidades relacionadas con la prestación de servicios financieros
- Aquellos de los que sean responsables las Entidades Gestoras y Servicios Comunes de la Seguridad Social y Mutuas.
- Aquellos de los que sean responsables los operadores que presten servicios de comunicaciones electrónicas disponibles al público o exploten redes públicas de comunicaciones electrónicas respecto a los datos tráfico y localización.
- Aquellos que contengan un conjunto de datos de carácter personal que ofrezcan una definición de las características o de la personalidad de los ciudadanos y que permitan evaluar determinados aspectos de la personalidad o del comportamiento de los mismos.

Medidas de seguridad en función del tipo de datos (3/3)

Nivel bajo

Cualquier otro

Excepciones

Nivel básico para ficheros con ideología, afiliación sindical, religión, creencias, origen racial, salud o vida sexual:

- Única finalidad de realizar una transferencia dineraria a las entidades de las que los afectados sean asociados o miembros
- Se trate de ficheros o tratamientos no automatizados en los que de forma incidental o accesoria se contengan aquellos datos sin guardar relación con su finalidad
- Se trate de ficheros o tratamientos que contengan datos relativos a la salud, referentes exclusivamente al grado de discapacidad o la simple declaración de la condición de discapacidad o invalidez del afectado, con motivo del cumplimiento de deberes públicos

El documento de seguridad

El documento de seguridad

El "documento de seguridad" es un documento de carácter interno que debe reflejar por escrito todo lo relacionado con las medidas, normas, procedimientos de actuación, reglas y estándares encaminados a garantizar la seguridad de los datos en una organización determinada. Dicho documento debe ser elaborado por el responsable del fichero y, en su caso, por el encargado del tratamiento, y es de obligado cumplimiento para todo el personal que tenga acceso a los sistemas de información.

+ Info

Inscripción de ficheros

Siempre que se proceda al tratamiento de datos personales, definidos en el art. 3,a) de la Ley Orgánica 15/1999, como "cualquier información concerniente a personas físicas identificadas o identificables," que suponga la inclusión de dichos datos en un fichero, considerado por la propia norma (artículo 3.b).), como "conjunto organizado de datos de carácter personal, cualquiera que fuere la forma o modalidad de su creación, almacenamiento, organización y acceso," el fichero se encontrará sometido a la Ley, siendo obligatoria su inscripción en el Registro General de Protección de Datos, conforme dispone el artículo 26.

Sanciones por incumplimiento de la LOPD

¡Infórmate! ¡Te interesa!

Aviso Legal

Documento que recoge las cuestiones que la Ley de Servicios de la Información (LSSI) obliga a incluir prácticamente en todas las webs, concretamente en aquellos "prestadores de servicios de la sociedad de la información", es decir personas físicas o jurídicas, que realicen actividades económicas por internet u otros medios telemáticos siempre que la dirección y gestión de su negocio esté centralizada en España o posea una sucursal, oficina o cualquier otro tipo de establecimiento permanente situado en España. Por ejemplo

- Web corporativa de una empresa
- Tienda ecommerce
- Autónomo con una web corporativa (página informativa sobre sus negocios o como tienda online)
- Blog particular si incluye publicidad

Política de privacidad

Hay que informar a los usuarios del procedimiento llevado a cabo por la Web para recoger los datos personales, permitiendo ver a los usuario el uso que se les da. Esta Política de Privacidad será aceptada por los usuarios de manera previa en los formularios de recogida de datos, y deberán de ser informados de manera inequívoca, según el artículo 5 de la LOPD:

- De la existencia de un fichero o tratamiento de datos de carácter personal, de la finalidad de la recogida de éstos y de los destinatarios
- Del carácter obligatorio o facultativo de su respuesta a las preguntas que les sean planteadas
- De las consecuencias de la obtención de los datos o de la negativa a suministrarlos
- De la posibilidad de ejercitar los derechos de acceso, rectificación, cancelación y oposición
- De la identidad y dirección del responsable del tratamiento

Condiciones Generales de Contratación y/o Uso

Si tenemos algún tipo de herramienta, asesoramiento online o comercio electrónico, debemos mostrar este texto legal que el usuario deberá aceptar previo a la formalización de la compra, donde se indique:

- Información clara y detallada de los precios, con mención expresa de si incluyen los impuestos correspondientes y gastos de envío y su importe
- Descripción del proceso de compra
- Obligaciones tanto para el vendedor y el comprador
- Condiciones de la compra, plazos, forma de entrega, forma de pago...
- Soluciones en el caso de que el pedido sea defectuoso
- Idioma en el que se va a celebrar el contrato

Además, la LSSICE obliga a confirmar al comprador la realización de la operación, puede ser expuesta por dos vías:

- Mediante email (máximo de 24 horas después de la compra)
- Mediante una pantalla de confirmación al finalizar la compra

Política de Cookies (1/2)

Extracto del apartado 2 del artículo 22 de la LSSI

Los prestadores de servicios podrán utilizar dispositivos de almacenamiento y recuperación de datos en equipos terminales de los destinatarios, a condición de que los mismos hayan dado su consentimiento después de que se les haya facilitado información clara y completa sobre su utilización...
Cuando sea técnicamente posible y eficaz, el consentimiento del destinatario para aceptar el tratamiento de los datos podrá facilitarse mediante el uso de los parámetros adecuados del navegador o de otras aplicaciones, siempre que aquél deba proceder a su configuración durante su instalación o actualización mediante una acción expresa a tal efecto.
Lo anterior no impedirá el posible almacenamiento o acceso de índole técnica al solo fin de efectuar la transmisión de una comunicación por una red de comunicaciones electrónicas o, en la medida que resulte estrictamente necesario, para la prestación de un servicio de la sociedad de la información expresamente solicitado por el destinatario.

Política de Cookies (2/2)

Cookies exentas

- Cookies estrictamente necesarias para prestar un servicio expresamente solicitado por el usuario
- Cookies necesarias únicamente para permitir la comunicación entre el equipo del usuario y la red

Ejemplos cookies exentas

- de entrada del usuario
- de autenticación de usuario
- de seguridad del usuario
- de sesión de repr. multimedia
- de carga
- de personalización de la interfaz
- de cesta de la compra
- para rellenar un formulario

¡Cuidado con!

- Publicidad de terceros
- Servicios externos (contadores de visitas, analíticas, mapas...)