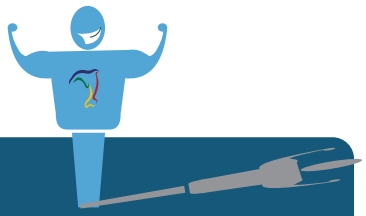
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

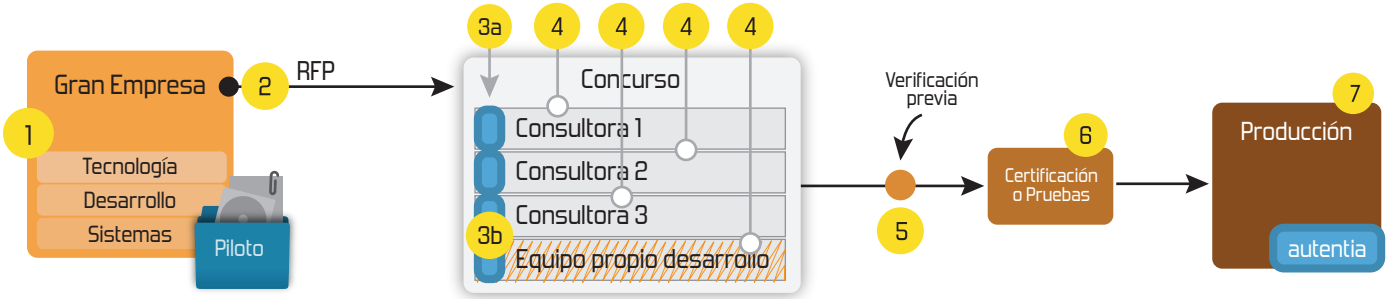
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

- Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery
- Gestor portales (Liferay)
- Gestor de contenidos (Alfresco)
- Aplicaciones híbridas
- Control de autenticación y acceso (Spring Security)
- UDDI
- Web Services
- Rest Services
- Social SSO
- SSO (Cas)
- JPA-Hibernate, MyBatis
- Motor de búsqueda empresarial (Solr)
- ETL (Talend)
- Dirección de Proyectos Informáticos.
- Metodologías ágiles
- Patrones de diseño
- TDD
- Tareas programadas (Quartz)
- Gestor documental (Alfresco)
- Inversión de control (Spring)
- BPM (jBPM o Bonita)
- Generación de informes (JasperReport)
- ESB (Open ESB)

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

Cristina Fernández Alvariño

Lincenciada en Arquitectura Superior en la ETSAV (Escuela Técnica Superior de Arquitectura de Valencia) de la UPV.

Master de Diseño Gráfico y Web en ESDIP (Madrid).

Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/JEE

Ver todos los tutoriales del autor

Fecha de publicación del tutorial: 2013-04-11

Tutorial visitado 1 veces Descargar en PDF

Descubriendo Responsive Web Design

En este tutorial vamos a construir una maqueta como ejemplo práctico aplicando los principios del Responsive Web Design que nos permitirá comprobar que nuestra web se ve correctamente en todos los dispositivos móviles, tablets y navegadores, y aprenderemos cómo hacerlo.

Índice

1. ¿Qué es el RWD (Responsive Web Design)?

Es un enfoque de diseño dirigido a la elaboración de sitios que nos proporciona una visualización y experiencia óptimas, tanto de navegación como en el cambio de tamaño, dependiendo del dispositivo desde donde accedamos a dicho sitio, velocidad, desplazamiento y por lo tanto de usabilidad. Se trata de adaptar nuestra web a cualquier dispositivo desde el que acceda el usuario ofreciendo una manera óptima de mostrarlo. Cuanto mayor sea la pantalla más elementos mostraremos, y cuanto más pequeña, el diseño variará ocultando los elementos prescindibles.

Ahora veremos cómo se hace :-)

2. Elementos que se utilizan

Un sitio diseñado con RWD utiliza:

- **CSS3 media queries**: Adapta el diseño al entorno de visualización. Permite utilizar distintos estilos CSS basándose en las características del dispositivo, normalmente se usa el ancho como referencia.
- **Cuadrícula fluida**: Se utilizarán unidades relativas como los porcentajes o ems, y no unidades absolutas como píxeles o puntos.
- **Imágenes flexibles**: También las usaremos en unidades relativas (hasta el 100%), así como para evitar que se muestren fuera de su elemento contenedor.

Algo muy interesante que conviene saber es que existen plantillas ya preparadas para ser usadas en nuestra web y de forma gratuita, aquí os muetsro 3 ejemplos:

- Skeleton
- Amazium
- Wiredframed

En estas webs puedes descargarte los archivos que ya vienen totalmente preparados para que los modifiques y completes con los estilos y contenido de tu propia web. Skeleton en concreto, te proporciona:

- index.html
- 404.html
- carpeta "stylesheets" que contiene:
  - skeleton.css
  - base.css
  - layout.css

El **index.html** y el **404.html** vienen con una estructura definida en **skeleton.css** que podrás modificar por tu cuenta y a tu gusto. Dicha estructura se basa en el ancho de columnas, es decir, las medidas de cada elemento se miden según el número de columnas que contiene, siendo el ancho de columna 40px. Este método es muy eficaz, pero a la vez limita su diseño dado que estamos componiendo nuestra web con anchos fijos y lo deseable es que sea lo más flexible posible

dada la cantidad de nuevos modelos de dispositivos y sus diferentes resoluciones que están saliendo en el mercado. Por lo que será más conveniente utilizar porcentajes.

La plantilla **base.css** contiene todos los estilos que se aplicarán a la web, podremos eliminar o añadir los que queramos.

Y por último, la plantilla que nos permite aplicar distintas clases CSS según el dispositivo desde donde accedamos es la **layout.css**, que utiliza las ya mencionadas **media queries**, y que ahora veremos cómo funciona.

3. Webs interesantes

He recopilado una serie de webs muy interesantes para ayudarnos a poner en práctica el RWD:

- **Para redimensionar el navegador**: Eliges el tamaño en el que deseas ver el navegador y éste te lo muestra de inmediato: resizemybrowser.com.
- **Para poner la url de tu web y ver cómo se ve en distintas resoluciones**: mattkersley.com/responsive.
- **Demo de cómo cambia el diseño según el tamaño de la pantalla/dispositivo**: finecitizens.com/define/responsive.
- **Imágenes flexibles**: Adapta tus imágenes dependiendo del tamaño del dispositivo adaptive-images.com.
- **Plugin jQuery para adaptar el tamaño de tu letra**: http://fittextjs.com/.

4. Ejemplo práctico

He "rediseñado" y adaptado la home de nuestra web **adictosaltrabajo** para ponerla como ejemplo de este tutorial. Vamos a ver paso a paso cómo construir una web según el RWD. Aquí os podéis descargar los recursos (html y css) que he creado como ejemplo para este tutorial.

+ Lo primero que tenemos que tener claro, es cómo queremos el diseño web, tendremos que pensar en distintos diseños, dependiendo del tamaño del dispositivo, para elegir qué elementos son importantes e imprescindibles, y qué elementos podrán desaparecer en los dispositivos de menor tamaño, así como qué haremos con las imágenes que tenemos, reducirlas o eliminarlas. Y pensar en el redimensionamiento de cada elemento que compone nuestra web.

+ Una vez lo tengamos claro. Nos ponemos a construir nuestro html.

HTML

+ Una de las cosas **importantes** a tener en cuenta es que para conseguir la correcta visualización en los dispositivos con tamaños diferentes tendremos que utilizar la etiqueta **meta name="viewport"** en la cabecera de nuestro html.

```
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
```

Esta etiqueta configura el área visible de nuestro navegador. Y tiene las siguientes propiedades:

- **width**: Define el ancho del área visible. El valor será el número de píxeles, o la constante que tomará el ancho del dispositivo: **device-width**.
- **height**: Define la altura del área visible. El valor será el número de píxeles, o la constante que tomará el ancho del dispositivo: **device-height**.
- **initial-scale**: Define la escala inicial del área visible. El valor será un número real que irá de 0.1 en adelante.
- **minimum-scale** : Define la escala mínima del área visible. El valor será un número real que irá de 0.1 en adelante.
- **maximum-scale** : Define la escala máxima del área visible. El valor será un número real que irá de 0.1 en adelante.
- **user-scalable** : Define los permisos de si se puede o no escalar el área visible. El valor será: yes/no.

También tenemos que tener en cuenta que habrá elementos que aparezcan y otros que desaparezcan de nuestro sitio, así que también tendremos que definirlos en el html. En el que os adjunto, aparecen "Menú Peq" y "Menú Med", que hacen referencia al menú pequeño y al menú mediano que he definido para cuando los dispositivos tengan unas medidas concretas.

CSS

+ Una vez construido el html, vamos a definir los estilos CSS en una plantilla "base", yo la he llamado **base.css**. ;-) Aquí definiremos todos los estilos que tendrá nuestro diseño. En la CSS que os adjunto están ordenados por bloques, según el elemento al que hacen referencia.

La CSS que aquí cobra más importancia porque va a ser la que nos defina qué estilos cogerá según el tamaño, será la que he llamado **layout.css**. Dicha CSS está formada por las antes nombradas **Media Queries**

Y funcionan de manera, que al definir un rango de ancho (max y min), el comportamiento de la CSS será el que hayamos añadido dentro de ese rango. Es decir aplicará los estilos de la **base.css** pero los modificará a favor de los que estén dentro del rango definido. Veamos cómo:

```
/* Mayor de 960px (dispositivos y navegadores) */

@media only screen and (min-width: 959px) {
    .menuPeq {visibility: hidden; display:none;}
    .menuMed {visibility: hidden; display:none;}
    .secundario {visibility: hidden; display:none;}
    }
```

En este ejemplo, hemos elegido un ancho mínimo de 959px, esto quiere decir, que si el navegador es mayor que dicha cifra, va a aplicar los estilos definidos en la **base.css** y además los que contiene dentro, por lo que el menú pequeño, el mediano y el secundario desaparecerán.

Veamos otro ejemplo:

```
/* Tamaño para tablet y móvil, vertical y horizontal (dispositivos y navegadores) */

@media only screen and  (max-width: 767px) {
    #todo {padding:3px 10px;}
    #menu {visibility: hidden; display:none;}
    #menu ul li{padding:7px 10px;}
    .menuPeq {visibility: hidden; display:none;}
    .menuMed {visibility: visible; display:inline;}
    .column-right {visibility: hidden; display:none;}
    .column-right img {width:100%;}
    .column-left {width:100%;}
    .secundario {visibility: visible; display:inline;}
```

» Resolver problema LockTimeoutException en Spring Web Flow

» WebSockets con Stomp y ActiveMQ: ¿chateamos?

Últimos Tutoriales del Autor

» Webs que ofrecen plantillas CSS gratuitas

» Creación de presentaciones con PowToon

» Creación de un gif animado con Adobe Photoshop

» Maquetación de un libro con Adobe InDesign

» Maquetación de un documento con Adobe InDesign

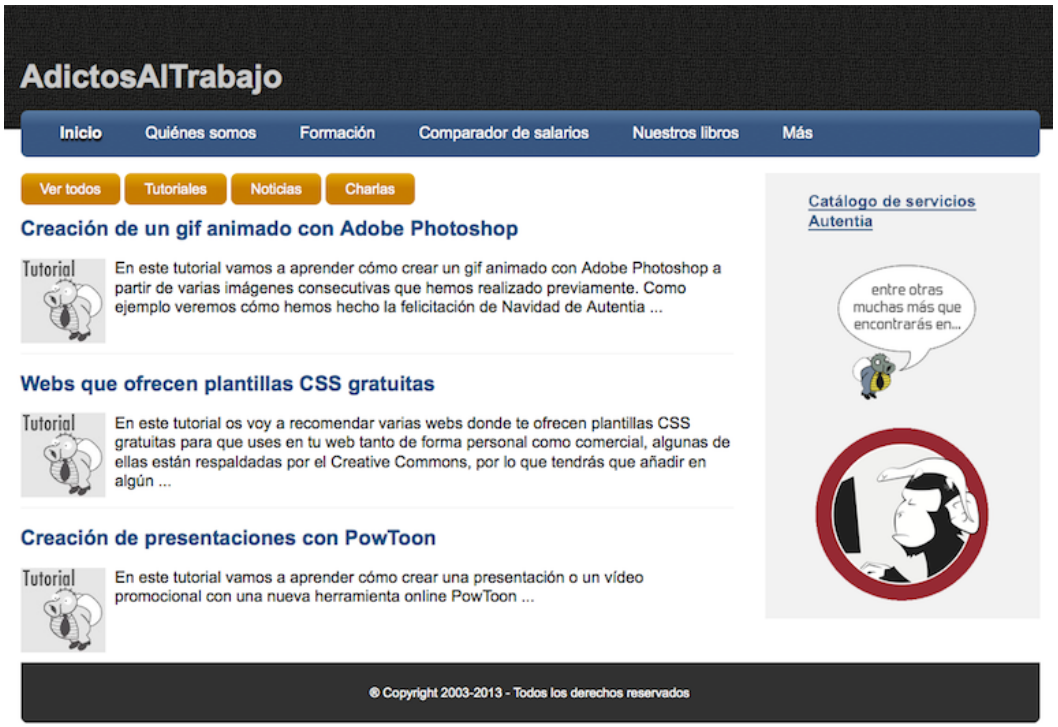
```
13      .sub-menu ul {margin:16px 0 0 0;}
14      .texto h2 {font-size:1.1em;}
15      img.imagen-tuto {width:12%;}
16      }
```

Aquí el tamaño máximo es de 767px, lo que quiere decir, que todas las clases que tiene dentro las aplicará siempre y cuando el tamaño del navegador sea inferior a 767px. Como anteriormente hemos explicado, primero cogerá los estilos de la plantilla **base.css** y después aplicará o modificará los que se encuentren dentro de su rango.

Ejemplo

Una vez hemos construido nuestra web, podemos ir probándola en la web que mencioné antes, donde podréis comprobar cómo va quedando, simplemente introduciendo la url de vuestra web: mattkersley.com/responsive

Y si queréis ver el ejemplo que he diseñado para este tutorial, podéis encontrarlo aquí y podéis ir viendo cómo cambia el diseño según el tamaño del navegador. También podéis acceder **a través del móvil** y comprobar el cambio en el diseño según el dispositivo que tengáis ;-)



5. Conclusiones

Estás serán las ventajas del diseño web adaptativo:

- **Mejora la experiencia de usuario**: la web se ve de la manera más óptima en cada dispositivo para todos los usuarios.
- **Costes más bajos:** se reducen los costes puesto que ya no necesitamos hacer un diseño para cada versión móvil.
- **Actualizaciones más eficientes**: cualquier modificación o actualización se verá reflejada en el diseño de manera automática, lo que reduce el tiempo y aumenta la eficacia.
- **Búsquedas**: sólo es necesario una url, lo que evitará errores o redirecciones.
- **Mejora en el posicionamiento SEO**: Google descubre así mejor tu contenido.
- **Accesibilidad**: se incluye a todas las personas con visión reducida.

También tenemos que asegurarnos de que nuestra CSS y nuestro HTML son válidos y no contienen errores, esto lo haremos con el validator CSS y el validator html5:

- CSS validation Service
- Validator HTML5

Espero que os haya servido este tutorial, y si tenéis alguna duda, sólo tenéis que escribirme :-)

A continuación puedes evaluarlo:

Regístrate para evaluarlo

Por favor, vota +1 o compártelo si te pareció interesante

Anímate y coméntanos lo que pienses sobre este **TUTORIAL:**

» Registrate y accede a esta y otras ventajas **«**