

Diseño Web Multimedia. Portales y tiendas virtuales. Internet del futuro

David Palomar Delgado ¹

¹ University Carlos III – Calle Madrid, 126, 28903 Getafe, Madrid, Spain
dpalomar@inf.uc3m.es

Resumen: A la hora de diseñar una tienda virtual, es necesario tener en cuenta que éste será un reflejo del negocio y que su organización, accesibilidad, facilidad de uso y su rendimiento, entre otras características tendrá una repercusión directa en el éxito de la ventas. Este capítulo se desarrollará agrupando un conjunto de conceptos relacionados con el desarrollo web y más concretamente, el diseño de portales y tiendas virtuales, sitios web orientados a la venta por internet y que dan soporte a la relación con el cliente. Una apartado a tener en cuenta es la influencia del “Internet del Futuro” y las tecnologías asociadas, en la configuración de un portal de comercio electrónico; y como una correcta explotación de las mismas pueden aportar una característica diferenciadora con respecto a la competencia.

Palabras clave: web; tienda online; Internet del Futuro

Abstract: When designing a virtual store, it is necessary to bear in mind that it will be a reflection of the business and that its organization, accessibility, ease of use and performance, among other features will have a direct impact on sales success. This chapter will be developed grouping together a set of concepts related to web development and more specifically, the design of portals and virtual stores, websites oriented to Internet sales and that support the relationship with the customer. A section to take into account is the influence of the "Internet of the Future" and associated technologies, in the configuration of an e-commerce portal, and as a proper exploitation of them can provide a differentiating feature with respect to the competition.

Keywords: web; online store; Future Internet

1 Introducción

Este capítulo se desarrollará agrupando un conjunto de conceptos relacionados con el desarrollo web y más concretamente, el diseño de portales y tiendas virtuales, sitios web orientados a la venta por internet y que dan soporte a la relación con el cliente.

A la hora de diseñar una tienda virtual, es necesario tener en cuenta que éste será un reflejo del negocio y que su organización, accesibilidad, facilidad de uso y su rendimiento, entre otras características tendrá una repercusión directa en el éxito de la ventas.

Una apartado a tener en cuenta, y que será tratado más extensamente en el tema, es la influencia del “Internet del Futuro” y las tecnologías asociadas, en la configuración de un portal de comercio electrónico; y como una correcta explotación de las mismas pueden aportar una característica diferenciadora con respecto a la competencia.

Con este tema se quiere reflejar la importancia del correcto diseño conceptual y funcional de un sitio web, y en especial de aquellos que están orientados a la venta por internet. Se ha dejado atrás la idea de que es suficiente mostrando un catálogo de productos y permitir la realización del pedido. Un portal de comercio electrónico se ha convertido en una herramienta completa para realizar el seguimiento del ciclo de vida de la relación con un cliente, incluyendo servicios con posterioridad a la recepción del producto/productos [1-5].

1.1 Diseño Web Multimedia

El diseño web es una actividad multidisciplinar y relativamente reciente como puede ser Internet. Está compuesto por diferentes disciplinas como son el diseño gráfico y las artes visuales, la programación de aplicaciones informáticas, el diseño de interfaces, la redacción de contenidos, la publicidad, el marketing; entre otras.

Se puede considerar que Diseñar es “un proceso creativo que combina arte y tecnología para comunicar ideas”.

Las cinco áreas que cubren las principales facetas del diseño web:

- 1 Contenidos
- 2 Aspecto visual
- 3 Programación
- 4 Almacenamiento y distribución por la red
- 5 Finalidad de la web.

Crear una página web para una negocio requiere profesionalidad, exactitud, creatividad y paciencia para cada aspecto del proceso de implementación. Es necesario combinar cada parte en el orden exacto para que el diseño gráfico y los objetivos de marketing se encuentren en el resultado final.

A la hora de realizar el diseño es recomendable seguir estándares como el del W3C (World Wide Web Consortium) para asegurarnos que la página es visible en cualquier navegador, PDA o teléfono móvil; además de permitir la compatibilidad con nuevas generaciones de dispositivos móviles, facilitando las actualizaciones y mantenimiento en el futuro.

Otro factor importante es la accesibilidad en el diseño web que permite que la información de la página pueda verse desde cualquier dispositivo (PDA, móvil, lectores de Braille, lectores de pantalla,...). Hay que tener en cuenta también el diseño para que el sitio web sea encontrado, es decir que la accesibilidad para los buscadores, requiere un diseño centrado en la información. Para ello se enriquece el texto en la página de inicio para que se dé una comunicación rápida y clara entre el sitio y el buscador. Por última la accesibilidad también significa “diseñado para la velocidad

de conexión del usuario” de modo que se diseñen las páginas para que los usuarios con conexiones más lentas puedan acceder a ellas.

Para atraer la atención de los usuarios es recomendable encontrar un equilibrio entre simplicidad y atractivo. El buen diseño se caracteriza por ser limpio, simple y fresco.

1.2 Portales y tiendas virtuales

La tienda virtual es un modelo engloba dentro del llamada B2C (Business to Consumer) que se caracteriza entre otras cosas por:

- Disponibilidad 24 horas
- Reducción de costes
- Nuevas relaciones con clientes y proveedores
- Mayor relación con el cliente
- ...

La tienda virtual se concibe como el intento de trasladar la actividad comercial de un comercio tradicional a Internet. Este concepto a evolucionado transformándose en el concepto de negocios electrónicos (e-business).

Dentro del concepto de tienda virtual se puede encontrar

- Tiendas virtuales especializadas
- Tiendas virtuales diversificadas
- Supermercados
- Portales

Algunos elementos principales dentro de las tiendas virtuales son los escaparates virtuales en los que es posible mostrar un producto o concepto y en el que no se limitará a incluir un fotografía o una descripción. También se puede destacar la utilidad de las visitas guiadas que son recorridos virtuales donde los usuarios pueden explorar en primera persona una espacio más o menos amplio; utilizando normalmente diseño 3D, video o fotografía panorámica.

1.3 Internet del Futuro

Con el concepto de Internet del Futuro se quiere englobar todos nuevos conceptos y tecnologías desde la infraestructura de red, dispositivos e interfaces, software y aplicaciones que en los próximos años se convertirán en algo cotidiano.

De esta forma surgen nuevas modelos e infraestructuras de red con mayor velocidad y accesibilidad, logrando una los principales objetivos de científicos y tecnólogos en el ámbito de las tecnologías de la información como es la ubicuidad. Es decir, que el usuario tenga acceso a determinados servicios en cualquier lugar independiente del dispositivo utilizado. Surgen también nuevas formas de interacción entre la persona y los sistemas, con especial hincapié en las interfaces 3D, lenguaje natural, interfaces hápticas,... Además se hace hincapié en la inteligencia de los sistemas y su adaptación a las necesidades concretas del usuario, logrando un alto grado de personalización.

Todos estos avances implicarán la innovación en los modelos de negocio actuales, entre los que se encuentra los modelos de comercio electrónico que se verán beneficiados en la mejora de la interacción con las aplicaciones informáticas, la mayor velocidad y seguridad en las redes, la

mejora en las búsquedas y la adaptación al perfil del usuario para lograr una experiencia de usuario más satisfactoria.

2 Tipos de portales y websites

Existen diferentes tipos de páginas web:

- Página Web Estática. Las páginas web estáticas son aquellas realizadas en HTML que pueden mostrar en alguna parte de la página objetos en movimiento tales como banners, GIFS animados, videos, etc.
- Página Web Dinámica. Existen muchos lenguajes de programación que son la base para la mayoría de páginas web dinámicas. Los más destacados son los lenguajes PHP, ASP y JSP. Estos lenguajes permiten la estructuración del contenido. Por una parte se crea la estructura de las páginas web, y por otra se almacena el contenido en determinados archivos. A partir de ahí, se crea el código de llamada que incluye el contenido en la propia página web estructurada. Este es el principio básico que siguen los lenguajes de programación. A partir de aquí se desarrollan aplicaciones para poder gestionar el contenido a través de un panel de control.
- Página Web Animada. Las páginas web animadas son aquellas que se realizan con la tecnología FLASH, ésta permite que una página web presente el contenido con ciertos efectos animados continuados. El uso de esta tecnología permite diseños más vanguardistas, modernos y creativos.
- Portal. Es un sitio web que desde su página principal permite el acceso a múltiples secciones que, por lo general, son foros, chats, cuentas de correo, buscador, acceso registrado para obtener ciertas funcionalidades, las últimas noticias de actualidad... un conjunto de recursos para brindar al usuario la mejor experiencia de búsqueda para una temática en particular o para una temática muy amplia. Se habla de dos tipos diferenciados de portal:
 - *Portal Horizontal*, se reconoce por su temática variada con el fin de captar el mayor tráfico de internautas posible. Son un ejemplo de esta modalidad, los portales de Yahoo, Terra, MSN, etc.
 - *Portal Vertical*, que se centra en ofrecer contenido de una temática en particular, de este modo podemos hablar de Portal Inmobiliario, Portal de Comercio, Portal Laboral, etc.

- Tienda virtual: se trata de un sitio web que publica los productos de una tienda en particular en Internet. Permite la compra on-line a través de VISA, domiciliación bancaria o transferencia por norma general. Ofrece al administrador un panel de gestión para poder subir los productos, actualizarlos, eliminarlos, etc. Es importante observar cuando realizamos una compra on-line, que la página donde realizamos la compra muestre un “candado” en la parte inferior derecha o bien en la parte derecha de la barra de navegación. Este detalle indica que la empresa propietaria de la tienda virtual ha habilitado una zona segura para que los datos no sean copiados por los hackers.
- Página Web con Gestor de Contenidos. Se trata de un sitio web cuyo contenido se actualiza a través de un panel de gestión por parte del administrador del sitio que se denomina gestor de contenidos. Este panel de gestión suele ser intuitivo y fácil de usar. En aquellas páginas web que requieran una actualización constante se suele incorporar este panel de gestión para que pueda ser mantenida diariamente por parte del cliente.

2.1 Página Web Estática

Son páginas que se construyen con el lenguaje HTML y se presentan con ausencia total de movimiento y sin funcionalidades más allá de los enlaces.

Son fáciles de construir pero muy difíciles de mantener. No poseen ningún tipo de interactividad con el usuario, pero constituyen la base fundamental e inicial de cualquier desarrollo web.

Estas páginas son muy sencillas de crear, aunque ofrecen pocas ventajas tanto a los desarrolladores como a los visitantes, ya que sólo se pueden presentar textos planos acompañados de imágenes y a lo sumo contenidos multimedia como pueden ser videos o sonidos.

Una página web estática presenta las siguientes características:

- Ausencia de movimiento y funcionalidades.
- Absoluta opacidad a los deseos o búsquedas del visitante a la página.
- Realizadas en XHTML o HTML.
- Para cambiar los contenidos de la página, es imprescindible acceder al servidor donde está alojada la página.
- El usuario no tiene ninguna posibilidad de seleccionar, ordenar o modificar los contenidos o el diseño de la página a su gusto.
- El proceso de actualización es lento, tedioso y esencialmente manual.
- No se pueden utilizar funcionalidades tales como bases de datos, foros, etc.

2.2 Página Web Dinámica

Una página es dinámica cuando se incluye cualquier efecto especial o funcionalidad y para ello es necesario utilizar otros lenguajes de programación, aparte del simple HTML.

Una página Web dinámica tiene las siguientes características:

- Gran número de posibilidades en su diseño y desarrollo.
- El visitante puede alterar el diseño, contenidos o presentación de la página a su gusto.
- En su realización se utilizan diversos lenguajes y técnicas de programación.
- El proceso de actualización es sumamente sencillo, sin necesidad de entrar en el servidor.
- Permite un gran número de funcionalidades tales como bases de datos, foros, contenido dinámico, etc.
- Pueden realizarse íntegramente con software de libre distribución.
- Existe una amplia comunidad de programadores que brinda apoyo desinteresado.
- Cuenta con un gran número de soluciones prediseñadas de libre disposición.

Mientras que las páginas estáticas todo el mundo se las puede imaginar y no merecen más explicaciones, las páginas dinámicas son más complejas y versátiles. Para aclarar este concepto, veremos con detalle a continuación cómo se generan las páginas dinámicas.

Como hemos visto, en realidad el HTML no es lenguaje de programación sino, más bien, se trata de un lenguaje descriptivo que tiene como objeto dar formato al texto y las imágenes que pretendemos visualizar en el navegador.

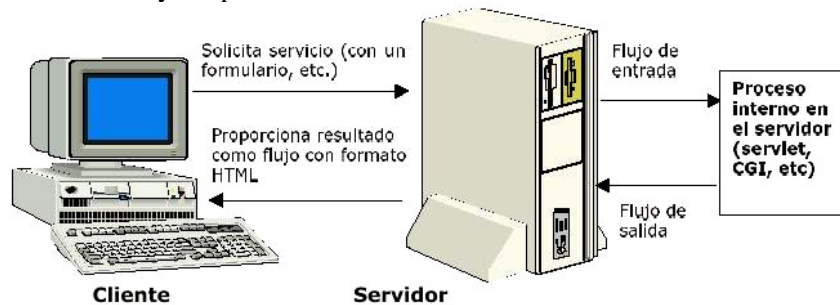
A partir de este lenguaje somos capaces de introducir enlaces, seleccionar el tamaño de las fuentes o intercalar imágenes, todo esto de una manera prefijada y en ningún caso inteligente. En efecto, el HTML no permite el realizar un simple cálculo matemático o crear una página de la nada a partir de una base de datos. A decir verdad, el HTML, aunque muy útil a pequeña escala, resulta bastante limitado a la hora de concebir grandes sitios o portales.

Es esta deficiencia del HTML la que ha hecho necesario el empleo de otros lenguajes accesorios mucho más versátiles y de un aprendizaje relativamente más complicado, capaces de responder de manera inteligente a las demandas del navegador y que permiten la automatización de determinadas tareas tediosas e irremediables como pueden ser las actualizaciones, el tratamiento de pedidos de una tienda virtual, etc.

Supongamos que hemos decidido realizar un portal de televisión donde una de las informaciones principales a proveer podría ser la programación semanal. Efectivamente, esta información suele ser dada por las televisiones con meses de antelación y podría ser muy fácilmente almacenada en una base de datos. Si trabajásemos con páginas HTML, tendríamos que construir una página independiente para cada semana en la cual introduciríamos "a mano" cada uno de los programas de cada una de las cadenas. Asimismo, cada semana nos tendríamos que acordar de descolgar la página de la semana pasada y colgar la de la anterior. Todo esto podría ser fácilmente resuelto mediante páginas dinámicas. En este caso, lo que haríamos sería crear un programa (solo uno) que se encargaría de recoger de la base de datos de la programación aquellos programas que son

retransmitidos en las fechas que nos interesan y de confeccionar una página donde aparecerían ordenados por cadena y por hora de retransmisión. De este modo, podemos automatizar un proceso y desentendernos de un aspecto de la página por unos meses.

La generación dinámica de contenidos para la web, requiere que el servidor realice algún tipo de procesamiento adicional sobre la petición HTTP iniciada por el cliente, con el fin de generar una respuesta personalizada y adaptada al usuario.



En definitiva, el concepto de página web dinámica se ha impuesto en el mundo del diseño y de la empresa en Internet. Páginas como Yahoo!, Google, Amazon son excelentes ejemplos de páginas Web dinámicas que permiten interactuar con el visitante y le ofrecen posibilidades realmente sorprendente: carritos de compra, posibilidad de incluir sus propias críticas en libros y discos, buscar en base a criterios determinados o participar en discusión.

Tenemos dos tipos de páginas dinámicas teniendo en cuenta donde se ejecutan:

- *Páginas dinámicas de Cliente:* son aquellas que se procesan en el cliente, esto es, se ejecutan en el navegador del usuario.
- *Páginas dinámicas de Servidor:* son aquellas que se ejecutan en el servidor.

Aunque las páginas dinámicas necesiten otros lenguajes aparte del HTML éste sigue siendo la base del desarrollo de cualquier página Web.

Generalmente una página dinámica tiene el código de los otros lenguajes de programación embebido dentro del código HTML. Muchos son los ejemplos de este tipo de páginas tales como horóscopos, tiendas virtuales.

2.2.1 Páginas dinámicas de Cliente

Toda la carga de procesamiento, de los efectos y funcionalidades la soporta el navegador. Su principal desventaja es que dependen de las características del navegador tales como el tipo, las versiones, etc. Sin embargo son muchas las ventajas debido a que las páginas descargan al servidor de tareas, ofrecen respuestas inmediatas a las acciones del usuario y permiten el uso recursos de la máquina local. El código necesario para crear los efectos y funcionalidades se incluye dentro del mismo archivo HTML y es llamado SCRIPT.

El navegador interpreta los scripts de cliente y los ejecuta para realizar los efectos y funcionalidades. Para escribir páginas dinámicas de cliente existen varios lenguajes, algunos de ellos son: Javascript, Visual Basic Script (VBScript), Dynamic HTML (DHTML), Cascade Style Sheets (CSS), Applets de Java. Las páginas dinámicas de cliente se escriben en dos lenguajes de programación principalmente Javascript y Visual Basic Script (VBScript) que con el uso de los CSS resulta una práctica generalizada en cualquier desarrollo actual.

El Javascript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página Web. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más

utilizado. Es un lenguaje de programación pensado para hacer las cosas con rapidez. Por un lado permite la generación de acciones de JavaScript para la creación efectos especiales sobre páginas web, tales como crear contenidos dinámicos y elementos de la página que tengan movimiento o cambien de color, etc. Por otro lado permite que se ejecuten las instrucciones como respuesta a las acciones del usuario, con lo que podemos dotar de interactividad cualquier página.

El VBScript es el lenguaje de scripts de Microsoft para la creación de páginas Web y sólo es compatible con Internet Explorer. Está basado en Visual Basic, un popular lenguaje para crear aplicaciones Windows, en su versión reducida. El modo de funcionamiento de Visual Basic Script para construir efectos especiales en páginas Web es muy similar al utilizado en Javascript. Los recursos a los que se puede acceder también son los mismos que con Javascript. No aconsejado en la Web que tiene como objetivo llegar a cualquier tipo de cliente/navegador, por tanto si se utiliza deberá ser para ciertos entornos limitados tales como una Intranet, donde sea conocida la tecnología utilizada.

DHTML o HTML dinámico, no es en sí mismo un lenguaje de programación sino que es una nueva capacidad de los navegadores modernos, para tener un mayor control sobre la página. Cualquier página que responde a las actividades del usuario y realiza efectos y funcionalidades se puede englobar dentro del DHTML (efectos en el navegador para mostrar y ocultar elementos de la página, modificar su posición, dimensiones, color, etc.). El DHTML nos da más control sobre la página, gracias a que los navegadores actuales incluyen una estructura para visualizar en páginas web denominada capa. Las capas se pueden ocultar, mostrar, desplazar, etc. Sin embargo para realizar cualquier acción en la página, como modificar la apariencia de una capa, necesitamos Javascript o VBScript. Dentro del concepto de DHTML se engloban también las Hojas de Estilo en Cascada o CSS (Cascade Style Sheets) ya que se descargan del lado del cliente o navegador que es quien interpreta los formatos y el estilo de los documentos para ser impresos, visualizados o leídos por un intérprete.

Los applets de Java son una manera de incluir programas complejos en el ámbito de una página Web. Se programan en Java (precompilados) y por tanto se benefician de la potencia de este lenguaje. La ventaja de utilizar applets consiste en que son mucho menos dependientes del navegador que los scripts en Javascript siendo además independientes del sistema operativo del ordenador donde se ejecutan. Como desventajas en relación con Javascript, los applets son más lentos de procesar y tienen espacio muy delimitado en la página donde se ejecutan ya que no se mezclan con todos los componentes de la página ni tienen acceso a ellos [6-10].

2.2.2 Páginas dinámicas del Servidor

Con este tipo de páginas dinámicas se puede hacer todo tipo de aplicaciones Web tales como agendas, foros, estadísticas, chats, etc. Son útiles en trabajos que acceden a información centralizada, situada en una base de datos en el servidor. Ejemplos habituales de este tipo de páginas son las de los bancos, la prensa electrónica, el comercio electrónico.

Las páginas dinámicas del servidor se suelen escribir en el mismo archivo HTML, mezclado con el código HTML y siguiendo el siguiente proceso:

- El cliente solicita una página.
- El servidor ejecuta los scripts de esa página y genera una página resultado (solamente contiene código HTML).
- El resultado se envía al cliente.

- El cliente puede interpretar el resultado sin lugar a errores ni incompatibilidades, puesto que sólo contiene HTML

Para escribir páginas dinámicas de servidor existen varios lenguajes, algunos de ellos son: *Common Gateway Interface (CGI)* comúnmente escritos en Perl, *Active Server Pages (ASP)*, *Hipertext Preprocesor (PHP)*, *Java Server Pages (JSP)*.

El CGI es el sistema más antiguo que existe para la programación de las páginas dinámicas de servidor. Los CGI se escriben habitualmente en el lenguaje Perl, aunque se pueden utilizar otros lenguajes: C, C++ o Visual Basic. Actualmente se encuentra un poco desfasado por la dificultad con la que se desarrollan los programas y la pesada carga que supone para el servidor que los ejecuta.

ASP (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página Web, utilizando el lenguaje *Visual Basic Script* o *Jscript* (Javascript de Microsoft). Se ejecuta en el servidor Web, justo antes de que se envíe la página través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la página ASP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. Los servidores que emplean este lenguaje son los que funcionan con sistema Windows. ASP permite acceso a bases de datos, sistema de archivos del servidor y en general a todos los recursos del servidor. Existen componentes ActiveX para múltiples usos, envío de correo, gráficas dinámicas, etc.

Otro lenguaje de programación del lado del servidor es PHP (HiperText Preprocesor), gratuito y de código abierto e independiente de plataforma. Es rápido y compatible con las bases de datos más comunes, como MySQL, Oracle, Informix, y ODBC. Posee una gran librería de funciones (matemáticas,...). Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. Finalmente *JSP (Java Server Pages)*, son páginas de Servidor Java. Es una tecnología orientada a crear páginas Web con programación en Java. Con JSP se puede crear aplicaciones Web multiplataforma (Java es multiplataforma). Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. El motor de las páginas JSP está basado en los servlets de Java (programas en Java destinados a ejecutarse en el servidor) pero JSP resulta mucho más sencillo aprender que los servlets.

Las ventajas de este tipo páginas Web dinámicas del lado del servidor son que el cliente no puede ver los scripts, ya que se ejecutan y transforman en HTML antes de enviarlos. Además son independientes del navegador del usuario, ya que el código que reciben es HTML fácilmente interpretable. Como desventajas se puede señalar que será necesario un servidor más potente y con más capacidades que el necesario para las páginas de cliente.

2.3 Página Web Animada

Las páginas web animadas son aquellas que se realizan con la tecnología FLASH, ésta permite que una página web presente el contenido con ciertos efectos animados continuados. El uso de esta tecnología permite diseños más vanguardistas, modernos y creativos, el aspecto es mejor en cuestión de diseño.

Esta Web es igual que las web estáticas, solo que esta totalmente animada, lo cual le agrega más interactividad a tu sitio y la hace más atractiva. Este tipo de sitios al ser estáticos, no están listos

para ser administrados por cualquier persona, por lo cual, una vez terminada la página es necesario que el mantenimiento y actualizaciones sean desarrollados por los desarrolladores.

Este tipo de páginas son frecuentemente utilizados como página publicitarias de eventos puntuales, ventas de productos concretos, etc., que no necesitan de una actualización frecuente.

Este tipo de sitios web han dejado de ser empleados debido a este factor, que limita su mantenimiento y supone un coste elevado y se suele combinar el uso de la tecnologías FLASH con páginas web dinámicas, tiendas virtuales para hacerlas más atractivas a los clientes y páginas basados en gestores de contenidos.

Algunos sitios web animados y muy interesantes son:



<http://saizenmedia.com/reinvented/>



<http://www.mojito.pl>



<http://www.gringo.nu/projects/cokelight-worldflavours/#/world>

2.4 Portal

Un portal web es un sitio web que ofrece el acceso a una serie de recursos y de servicios relacionados a un mismo tema. Incluye enlaces web, buscadores, foros, documentos, aplicaciones entre otras funcionalidades. Estas formado por diferentes páginas web enlazadas entre si que pueden ser meramente informativas o contar con funciones como recogida de datos, acceso a información, etc.

El término portal tiene como significado *puerta grande*, y precisamente su nombre hace referencia a su función u objetivo: es, por lo general, el punto de partida de un usuario que desea entrar y realizar búsquedas en la web u obtener información. Se puede decir que un portal ofrece servicios para la navegación en el Internet, logrando incrementar la intensidad de tráfico en el mismo.

Un portal web puede, por ejemplo, actuar como intermediario entre compradores y vendedores de productos o servicios. Un portal puede contar con herramientas de búsqueda de empresas con determinados productos, la visualización de anuncios publicitarios, acceso a correos electrónicos, etc.

El portal es considerado un *intermediario de información* que tiene como fuente de ingreso la publicidad de empresas que ahí se anuncian.

Un portal debe tener una temática de interés de un grupo importante - pero especializado - de internautas. Dado que un portal presenta la información de una forma fácil e integrada, debe tener una forma simple de acceder a toda la información referida de esa temática. Toda esta información no necesariamente está contenida dentro del mismo portal, porque el portal, normalmente, se encarga de centralizar *enlaces* en una forma fácil y organizada que facilite la navegación dentro de un tema específico en la red.

Un portal puede ofrecer diferentes servicios:

- Servicios de búsqueda que incluye mecanismos de búsqueda, directorios y páginas amarillas para localizar negocios o servicios.
- Contenidos, es decir, información de varios temas como noticias, deportes, pronósticos de clima, listas de eventos locales, mapas, opciones de entretenimiento, juegos, ligas a

estaciones de radio y a otros sitios con contenido especial en ciertas áreas de interés como coches, viajes y salud entre otros.

- Anuncios clasificados para trabajos, coches y casas; subastas, pequeños agregados de vendedores y ligas a otros sitios que también se dedican a la venta.

Existen tres modalidades de portales:

1. **Portales horizontales**, también llamados portales masivos o de propósito general, se dirigen a una audiencia amplia, tratando de llegar a toda la gente con muchas cosas. Como ejemplo de portales de esta categoría están Terra, AltaVista, Lycos, Yahoo, MSN, etc.
2. **Portales verticales**, se dirigen a usuarios para ofrecer contenido dentro de un tema específico como puede ser un portal de música, empleo, inmobiliario, un portal de finanzas personales, arte, educación o de deportes.
3. **Portales diagonales**, que es una mezcla entre el portal horizontal y el vertical. Se trataría de portales que utilizan redes sociales o aplicaciones generalistas como Facebook, LinkedIn, Flickr o YouTube, complementados con contenidos y/o utilidades dirigidas a un público muy concreto.

Los portales normalmente tienen programación que requiere muchos recursos computacionales y por su alto tráfico generalmente se hospedan en servidores de Internet dedicados.

2.5 Tienda Virtual

El concepto de tienda virtual representa el intento de trasladar la actividad comercial habitual de un comercio tradicional a internet. Sin embargo, el término ha llegado a englobar tanto significado que ahora ya debemos hablar de Negocios Electrónicos (e-business).

Las tiendas virtuales desarrollan, adoptan mecanismos y herramientas para alcanzar el prototipo de la tienda ideal. A continuación podremos ver esta distinción relacionada con los modelos que se pueden aplicar en diferentes empresas según el producto que comercializan.

Las tiendas virtuales se pueden clasificar en:

- *Tiendas virtuales especializadas*: estas tiendas se suelen inclinar hacia una estrategia de marketing en la que su ventaja competitiva recae en el enfoque de venta de determinado tipo de productos a un sector específico. En esta categoría entraría sobre todo aquellas empresas productoras que venden su propio producto.
- *Tiendas virtuales diversificadas*: suelen ser empresas que en sus inicios se consideraron tiendas con departamentos virtuales y que actualmente han modificado el enfoque de su negocio por una tendencia de extensión de línea. Se trata por ejemplo del caso de FNAC (<http://www.fnac.es>), donde se puede comprar tanto libros como música, videojuegos, aparatos electrónicos, etc.

- *Supermercados*: en la actualidad, la mayoría de los supermercados más conocidos tienen su tienda online a través de la cual las familias pueden realizar sus compras habituales. Normalmente, estos establecimientos establecen un importe mínimo de compra para poder realizar el pedido y otras, como es el caso de Mercadora, cobran una cantidad fija por el envío sin necesidad de ese mínimo. Pero lo más interesante de un proyecto como este es poder hacer compras en el extranjero y despachar los víveres a los familiares en otro lugar.
- *Portales*: algunos portales están empezando a contar con áreas dedicadas exclusivamente a las transferencias B2C a través de un esquema similar al de e-mail (centro comercial virtual), siendo un ejemplo de ello Yahoo! Shopping.

Una tienda virtual es un sitio en el que pretendemos vender algo, y como en la vida real, necesitamos publicidad, tanto para la tienda como para los productos. Para que la tienda venda, es necesario que tenga visitas y, por ello, los navegantes deben conocerla.

Empleando de forma eficiente las herramientas de marketing on line, se conseguirá hacer que la tienda destaque sobre las demás y por tanto el incremento de las ventas. Algunos factores que influyen en el comercio a través de Internet son:

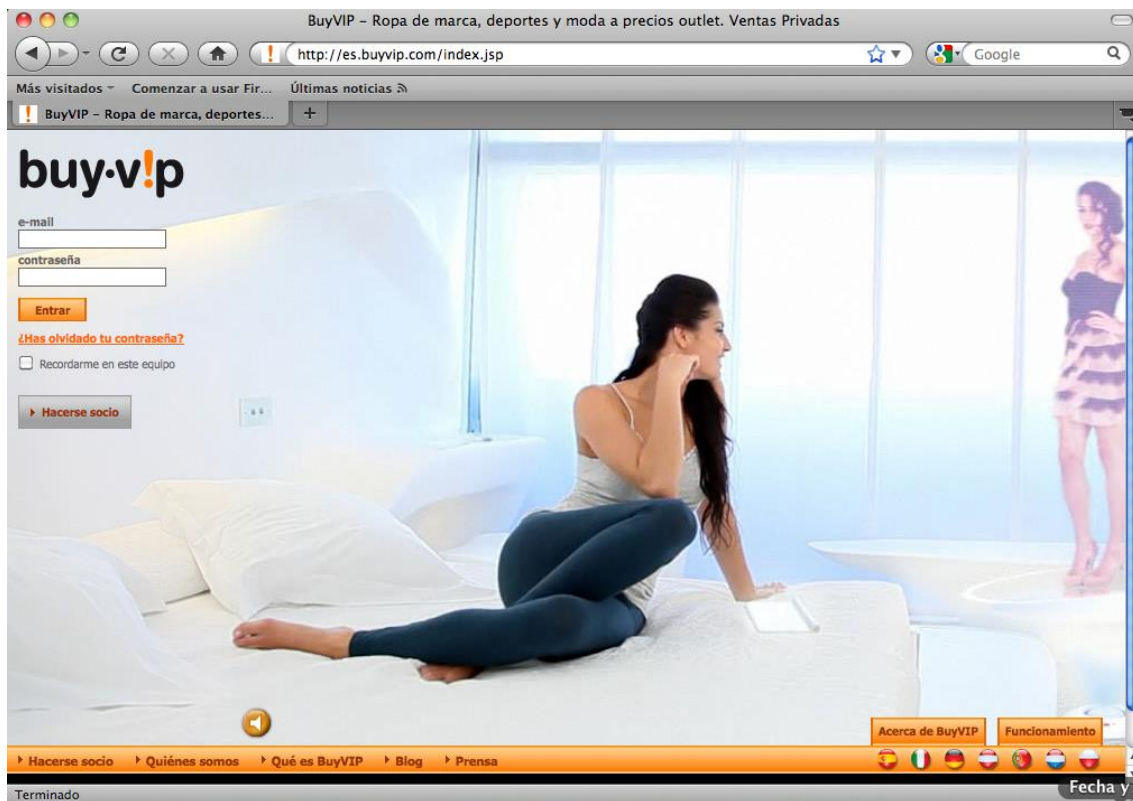
- *Tangibilidad del producto*: Internet es un medio electrónico, por lo que plantea algunos problemas a la hora de vender objetos del mundo real, especialmente, aquéllos que la gente está acostumbrada a ver y tocar antes de comprar. Por lo tanto el comprador debe tener la misma información que en un tienda real.
- *Distancia*: Internet es un mercado global, es decir, sus clientes potenciales pueden proceder de cualquier parte del mundo, por ello, la tienda virtual tiene que estar preparada para cumplir los plazos.
- *Confianza y seguridad*: La desconfianza, tanto en los medios de pago como en el propio sistema es de mayor obstáculo que ponen los navegantes para comprar en Internet. Para resolver este problema, una tienda virtual debe contar con medios de pago seguros, políticas de privacidad y un servicio postventa para presentar una posible reclamación.

Una típica campaña de e-marketing debe seguir los siguientes pasos:

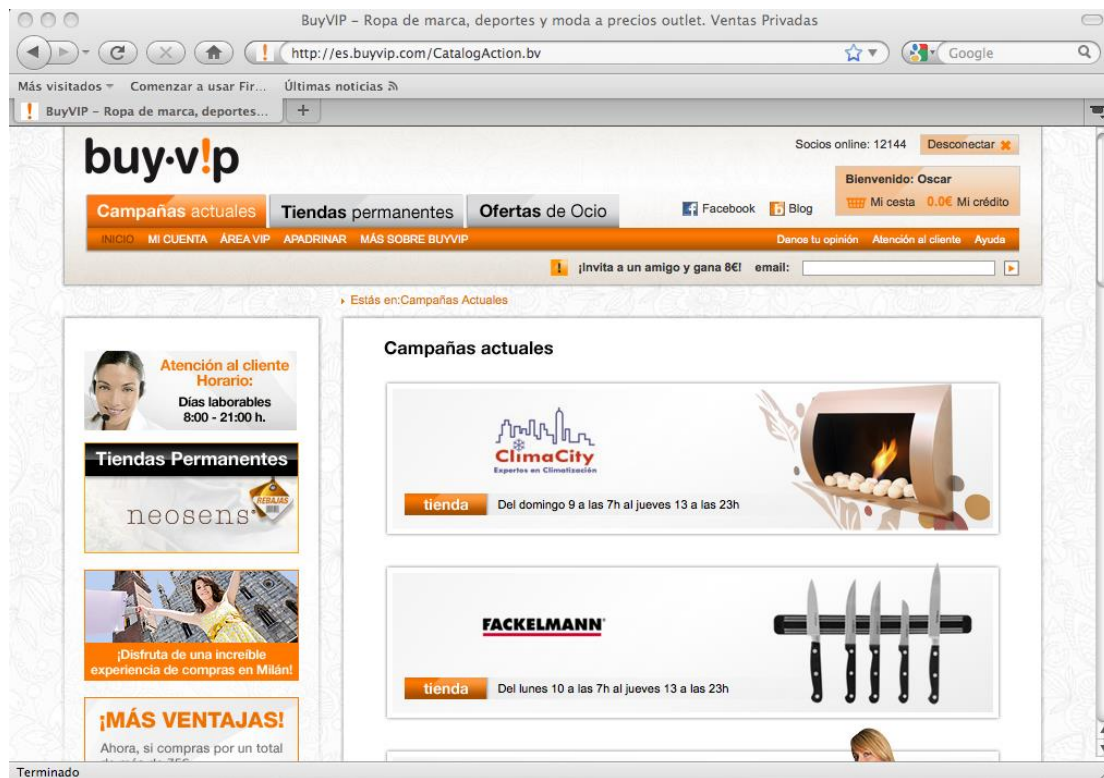
- *Analizar el producto o servicio que se pretende vender*: Las campañas tiene que dar información a los clientes acerca de quiénes somos, qué vendemos, cómo lo vendemos, los servicios proporcionados, etc.
- *Conocer a los clientes*: Una campaña de marketing debe permitirnos conocer con exactitud los posibles compradores. Para ello, la red nos proporciona herramientas muy útiles, por ejemplo el análisis de las visitas que tiene nuestra tienda, análisis que proporciona información interesante acerca de los países desde los que nos visitan.

- *Determinar el público objetivo:* Debemos identificar y segmentar fácilmente nuestro público objetivo.
- *Analizar la competencia.* Internet nos permite conocer y analizar fácilmente a la competencia. Ésta puede ser competencia directa, empresas que venden los mismos productos que nosotros y al mismo sector de mercado; competencia indirecta, empresas que ofrecen al mismo sector de clientes un productos distinto del nuestro; y competencia potencial, empresas que todavía no ofertan productos similares al nuestro, pero pueden hacerlo.
- *Marcar metas y objetivos:* A la hora de diseñar una campaña para promocionar nuestro sitio, es necesario marcar los objetivos que se pretenden conseguir. Solamente si fijamos objetivos, podremos valorar correctamente el éxito o fracaso de la campaña. Estos objetivos han de ser claros, realistas y medibles.
- *Diseñar la campaña:* El diseño de la campaña debe especificar los medios que se emplearán, el momento, la duración y el responsable de la misma.
- *Proporcionar retroalimentación:* Internet es un medio interactivo, y como tal, el navegante espera que sus sugerencias sean tenidas en cuenta. Por ello, la información que nos proporcionan los clientes y usuarios ha de formar parte de nuestra estrategia de marketing. Tenemos que poner a disposición de nuestros clientes herramientas para proporcionar la retroalimentación, como, por ejemplo, el correo electrónico, formularios, sistemas de respuesta automática o encuestas.

Un ejemplo práctico y muy conocida en la actualidad son las tiendas virtuales como por ejemplo *BuyVip* [11-15].



Estas tiendas tiene un modelo de negocio alternativo en el que venden en campañas, productos de diferente naturaleza (ropa, productos electrónicos, productos de belleza) que tienen una duración determinada y un stock limitado.



Es una aplicación muy atractiva visualmente e intuitiva con lo que los usuarios tienen facilidad para realizar sus compras. El éxito de esta página radica no sólo en los precios baratos que ofrecen sino también en la forma en la que se presentan los artículos y el seguimiento que permiten del pedido una vez realizado.

2.6 Página Web con Gestor de Contenidos

Los sistemas de gestión de contenidos (Content Management Systems CMS) son un software que se utiliza principalmente para facilitar la gestión de grandes sitios web, ya sea un sitio web en Internet o una intranet corporativa, y por ello también son conocidos como gestores de contenido web (Web Content Management Systems WCMS). Hay que tener en cuenta que la aplicación de los CMS no se limita sólo a las webs, sino también a la gestión de todo tipo de documentos y registros electrónicos.

Un CMS puede servir para realizar cuatro tareas fundamentales: creación, gestión, publicación y presentación de contenido.

- **Creación de contenido:** Un CMS aporta herramientas para que los creadores con escasos conocimientos técnicos en páginas web puedan incorporar contenido. Por lo general el CMS proporciona un editor de texto WYSIWYG (What You See Is What You Get) en el que el usuario ve el resultado final mientras escribe, similar a los editores comerciales, pero con un rango de formatos de texto limitado. Esta limitación tiene sentido, ya que el objetivo es que el creador pueda introducir diversos contenidos, pero sin modificar mucho

el estilo general del sitio web. También suelen integrarse otras herramientas como la edición de documentos en XML, utilización de aplicaciones ofimáticas incorporadas en el CMS, importación de documentos existentes y editores que permiten añadir marcas, habitualmente HTML, para indicar el formato y estructura de un documento.

Para la creación del sitio propiamente dicho, los CMS aportan herramientas para definir la estructura, el formato de las páginas, el aspecto visual, uso de patrones, y un sistema modular que permite incluir funciones no previstas originalmente.

- **Gestión de contenido:** Los documentos creados se depositan en una base de datos central donde también se guardan el resto de datos de la web, como son los datos relativos a los documentos (versiones hechas, autor, fecha de publicación y caducidad, etc.), datos y preferencia de los usuarios, la estructura de la web, etc.

La estructura de la web se puede configurar con una herramienta que, habitualmente, presenta una visión jerárquica del sitio y permite modificaciones. Mediante esta estructura se puede asignar un grupo a cada área, con responsables, editores, autores y usuarios con diferentes permisos. La determinación de perfiles es imprescindible para facilitar el ciclo de trabajo con un circuito de edición que va desde el autor hasta el responsable final de la publicación. El CMS permite la comunicación entre los miembros del grupo y hace un seguimiento del estado de cada paso del ciclo de trabajo.

- **Publicación:** Una página aprobada se publica automáticamente cuando llega la fecha de publicación y cuando caduca se archiva para futuras referencias. En su publicación se aplica el patrón definido para toda la web o para la sección concreta donde está situada, de forma que el resultado final es un sitio web con un aspecto consistente en todas sus páginas. Esta separación entre contenido y forma permite que se pueda modificar el aspecto visual de un sitio web sin afectar a los documentos ya creados. Por tanto, los autores de contenido no tienen que preocuparse por el diseño final de sus páginas.

2.6.1 Funcionalidades de los CMS

Un CMS puede gestionar automáticamente la accesibilidad de la web, con soporte de normas internacionales de accesibilidad como WAI y adaptarse a las preferencias o necesidades de cada usuario. También puede proporcionar compatibilidad con los diferentes navegadores disponibles en todas las plataformas (Windows, Linux, Mac, Palm, etc.) y su capacidad de internacionalización lo permite adaptarse al idioma, sistema de medidas y cultura del usuario.

El sistema se encarga de gestionar muchos otros aspectos como son los menús de navegación o la jerarquía de la página actual dentro de la web, añadiendo enlaces de forma automática.

En la actualidad, aparte de la ampliación de las funcionalidades de los CMS, uno de los campos más interesante es la incorporación de estándares que mejoran la compatibilidad de componentes, facilitan el aprendizaje al cambiar de sistema y aportan calidad y estabilidad al producto o aplicación resultante.

Algunos de estos estándares son:

- CSS, que permite la creación de hojas de estilo.

- XML, un lenguaje de marcas que permite estructuras un documento.
- XHTML, que es un subconjunto del anterior orientado a la presentación de documentos vía web.
- WAI, que asegura la accesibilidad del sistema.
- RSS, para syndicar contenidos de tipo noticia.

También las aplicaciones que rodean los CMS acostumbran a ser estándar (de facto), como los servidores web Apache e ISS; los lenguajes PHP, Perl y Python; y las bases de datos MySQL y PostgreSQL. La disponibilidad para los principales sistemas operativos de estas aplicaciones y módulos, permite que los CMS puedan funcionar en diversas plataformas sin muchas modificaciones.

2.6.2 Utilidades de los CMS

Algunos de los puntos más importantes que hacen útil y necesaria la utilización de un CMS:

- *Inclusión de nuevas funcionalidades en la web.* Esta operación puede implicar la revisión de multitud de páginas y la generación del código que aporta las funcionalidades. Con un CMS eso puede ser tan simple como incluir un módulo realizado por terceros, sin que eso suponga muchos cambios en la web. El sistema puede crecer y adaptarse a las necesidades futuras.
- *Mantenimiento de gran cantidad de páginas.* En una web con muchas páginas hace falta un sistema para distribuir los trabajos de creación, edición y mantenimiento con permisos de acceso a las diferentes áreas. También se tienen que gestionar los metadatos de cada documento, las versiones, la publicación y caducidad de páginas y los enlaces rotos, entre otros aspectos.
- *Reutilización de objetos o componentes.* Un CMS permite la recuperación y reutilización de páginas, documentos, y en general de cualquier objeto publicado o almacenado.
- *Páginas interactivas.* Las páginas estáticas llegan al usuario exactamente como están almacenadas en el servidor web. En cambio, las páginas dinámica no existen en el servidor tal como se reciben en los navegadores, sino que se generan según las peticiones de los usuarios. De esta manera cuando por ejemplo se utiliza un buscador, el sistema genera una página con los resultados que no existían antes de la petición. Para conseguir esta interacción, los CMS conectan con una base de datos que hace de repositorio central de todos los datos de la web.

- *Cambios de aspecto de la web.* Si no hay una buena separación entre contenido y presentación, un cambio de diseño puede comportar la revisión de muchas páginas para su adaptación. Los CMS facilitan los cambios con la utilización, por ejemplo, del estándar CSS (Cascading Style Sheets u hojas de estilo en cascada) con lo que se consigue la independencia de presentación y contenido.
- *Consistencia de la web.* La consistencia en un sitio web no quiere decir que todas las páginas sean iguales, sino que hay un orden (visual) en vez de caos. Un usuario nota enseguida cuando una página no es igual que el resto de las de la misma web por su aspecto, la disposición de los objetos o por los cambios en la forma de navegar. Estas diferencias provocan sensación de desorden y dan a entender que la web no la han diseñado profesionales. Los CMS puede aplicar un mismo estilo en todas las páginas con CSS, y aplicar una misma estructura mediante patrones de páginas.
- *Control de acceso.* Controlar el acceso a una web no consiste simplemente al permitir la entrada a la web, sino que comporta gestionar los diferentes permisos a cada área del sitio web aplicados a grupos o individuos.

Los gestores de contenidos más difundidos actualmente son:

- *Drupal:* es un sistema de gestión de contenido modular y código abierto con licencia GNU/GPL. Está escrito en PHP y desarrollado y mantenido por una activa comunidad de usuarios. Destaca por su respeto de los estándares de la web y en el énfasis en la usabilidad y consistencia del sistema. (<http://www.drupal.org>)
- *Joomla:* otro gestor de contenido web bastante extendido y con soporte de una comunidad de desarrolladores. El potencial de Joomla! Radica en su arquitectura que permite que desarrolladores de todo el mundo puedan crear add-ons y extensiones como generadores de formularios dinámicos, directorios de empresas u organizaciones, gestores de documentos, galerías de imágenes multimedia, etc. (<http://www.joomla.org>)

3 Estructura de un portal e Internet

3.1 Introducción

El comercio electrónico se aplica a la compra y venta de información, productos y servicios a través de Internet. Para que las transacciones sean correctas, este tipo de tiendas deben llevar asociadas unas características que den respuestas a las necesidades tanto de los consumidores, como de las propias empresas. Entre estas características se pueden destacar la reducción de costes, el tiempo de entrega, el aumento de calidad en los productos o la mejora de la comunicación con los clientes, entre otras [16-20].

Esta nueva forma de venta, se apoya en dos pilares, por un lado, el marco legal, el cuál, aunque está cada vez más desarrollado todavía no tiene los suficientes mecanismos para caracterizar adecuadamente la privacidad de la información, la seguridad de los sistemas o cómo proporcionar un adecuado acceso a la información. Y por otro lado, el marco tecnológico, que avanza rápido y con paso firme, éste define los estándares técnicos y los medios para la transmisión de información. Entre estos dos pilares, considerados como fundamentales, encontramos la necesidad de utilizar un conjunto de *códigos de uso*, o prácticas habituales. Entre ellos, destaca aspectos tecnológicos como la autenticación, la seguridad en transacciones, los medios de pago, etc.; o aspectos de uso propiamente, como el escaparate o el local de la tienda, el carrito de la compra, etc. Por ello, a lo largo de este apartado veremos algunas de las prácticas de uso más utilizadas, y que en la actualidad ya se han convertido en un estándar de facto en la industria de venta por Internet.

No obstante, antes de comenzar es importante resaltar características básicas que se presuponen, pero que no está por demás exponerlas: una tienda virtual no tiene escaparate, ya que su escaparate es la pantalla del ordenador; tampoco tiene vendedores, ya que el vendedor sigue siendo la pantalla del ordenador; por otro lado, tampoco tiene almacén (o no lo vemos), ya que se trabaja bajo pedido. Siguiendo esta línea se pueden seguir detallando innumerables características, no obstante, con independencia de las similitudes o diferencias con una tienda tradicional, o incluso si la tienda virtual ya existía como tienda física o no, es indudable que la forma de gestionar una tienda online o comercio electrónico cambia radicalmente respecto a un comercio tradicional con independencia del tamaño o características que éste posea.

3.2 Códigos de uso en un comercio electrónico

El propio concepto de Tienda virtual o tienda online representa el intento de trasladar la operativa comercial habitual de un comercio tradicional a Internet. Sin embargo, el cambio es tan radical que no sólo se habla de Comercio Electrónico sino que es necesario hablar de Negocios Electrónicos.

Pero, ¿Cuáles deben ser las características principales de una Tienda Virtual? Aunque pueda parecer una idea demasiado evidente, un error que se puede cometer con facilidad, sobretodo en nuevos proyectos, es confundir la construcción y explotación de la tienda virtual con el negocio. Supondremos, pues, que el equipo que está detrás de la tienda virtual tiene claro cuál es el proyecto de fondo, que ha preparado el plan de negocio y que tiene claro que la tienda virtual no es más que el medio para hacer negocios en Internet y no un fin en sí mismo.

3.2.1 El escaparate y el local

Internet es un gran escaparate para cualquier tipo de empresa, dando innumerables posibilidades de apoyo a cualquier negocio físico. Por un lado, se da la oportunidad de publicitar cada negocio en la red, con las ventajas que tiene exponerlo a millones de personas que navegan (y compran) cada día. Por otro lado, también supone un atractivo importante para un nuevo negocio que podrá ponerse en marcha con una inversión menor, disponiendo además de las mismas ventajas que cualquier otro portal o tienda online, con independencia de su tamaño y recursos.

Un sitio web suele constar de varias páginas, al entrar por primera vez, en cualquier tienda virtual el usuario se encuentra con un primer *escaparate*, es decir, la fachada del portal de ventas. El escaparate es el que hará que cualquier usuario, y por lo tanto cliente potencial “entre y vea” los productos que se ofrecen, por lo tanto es muy importante que le guste lo que ve y, que por lo tanto se quede. Cada escaparate tiene un espacio más o menos limitado, por lo tanto importa mucho la organización. La parte superior es la más importante, es la que primero se ve.

Otro punto a tener en cuenta es que el escaparate no es el lugar para destacar todos los productos y por lo tanto no tiene que abarcar todo un catálogo. Tan sólo es necesario exponer aquello que se considere más importantes. Es decir, el escaparate es el lugar dónde se expone los productos que son más recomendable e interesantes tanto para los clientes, como para los proveedores. Un escaparate puede cambiarse tantas veces como se quiera, pudiendo exponer una gran cantidad de productos sin que ello someta a una sobrecarga de información al cliente. Por otro lado, es muy recomendable poner el precio de los productos que se están exponiendo.

En la siguiente figura se observa un escaparate que en principio está bien organizado, ya que tiene un buen conjunto de productos (los principales), pero no existe una sobrecarga de información.



La primera página es la más importante, pero las siguientes no lo son menos, también hay que tenerlas en cuenta. Lo primero que buscarán los visitantes que entren en nuestra tienda virtual serán los productos o servicios que vendemos.

Los productos que aparecerán en las páginas web de nuestra tienda virtual se pueden presentar y gestionar de distintas formas. Los catálogos electrónicos permiten una gran organización de los productos (a través de familias y categorías) y no requiere crear miles de paginas web, sino que se utiliza un formato predefinido e información de las bases de datos para crear las paginas de productos de forma dinámica. Asimismo también facilita la incorporación de nuevos productos y mantenimiento de precios, cantidades y descripciones a través de un sistema administrativo.

3.2.2 El carrito de la compra

Uno de los conceptos que ha tomado mayor relevancia en el ámbito de las tiendas virtuales es el **carrito o cesta de la compra**. Éste simplemente es un módulo que asiste al cliente en su compra a través de un sistema de comercio electrónico. Anota los productos que se van indicando, calcula su importe final y le permite devolver productos en el caso de que al final no los necesite, etc.

Por lo tanto, el carrito de la compra es un elemento indispensable en todos los comercios electrónicos. Su importancia radica en la facilidad que ofrece al añadir, eliminar o modificar aquellos productos seleccionados previamente del índice general de productos o catálogo. Para que la funcionalidad del carrito sea completa se debe mostrar en todas las páginas de productos mediante un ícono representativo (motivo gráfico que puede recordar a un carrito o cesta). Un click en ese ícono debe permitir visualizar el pedido que cada cliente vaya acumulado e informarle del importe total de la factura con impuestos y portes de envío incluidos.



Desde la cesta de la compra el cliente debería tener la capacidad para añadir o modificar cantidades o eliminar cualquier producto que haya seleccionado previamente y además calcular automáticamente el coste final de la compra, incluyendo:

- **Ofertas y descuentos.** Las tiendas virtuales, como cualquier otra tienda, debe permitir definir ofertas puntuales, métodos de descuento personalizado, códigos promocionales, etc. Además cada uno de estas ofertas o promociones se deben poder asignar a un producto, familia de productos, grupo de clientes, etc.
- **Coste de envío.** El carrito debe calcular de forma automática cuáles van a ser los costes de envío antes de realizar la compra teniendo en cuenta una serie de parámetros como (peso, volumen, tipo de envío, tipo de producto, tipo de cliente, etc.).
- **Calculo automática de impuestos.** Se debe calcular de forma automática cualquier tipo de impuesto o gravamen que haya sobre los artículos que se vayan a comprar.

3.3 Procesos de la tienda virtual

Aunque tal y como ya se ha detallado a lo largo del apartado anterior, una tienda virtual puede (y debe) tener una gran cantidad de semejanzas con una tienda tradicional. Facilitando de este modo la adaptación del usuario al nuevo entorno y por lo tanto reducirá la resistencia al cambio que puede tener cualquier nuevo cliente de un negocio de estas características. Pero en la realidad, sin ir más allá, una tienda virtual es básicamente una aplicación informática destinada a realizar ventas por internet, que procesa una gran cantidad de complejos flujos de información entre clientes,

proveedores, comerciales, etc. Por tanto, es una aplicación de gestión de información, que puede ser básica o compleja en función del tamaño o de las características que se deseen implementar. A lo largo de este apartado se abordarán algunas de las características que debe poseer una tienda virtual, en tanto en cuanto a características funcionalidad. No obstante, no se pretende profundizar en aspectos relacionados con el Marketing, fundamentales en cualquier negocio de este tipo, ya que no es el principal interés de este curso.

3.3.1 Roles e interfaces

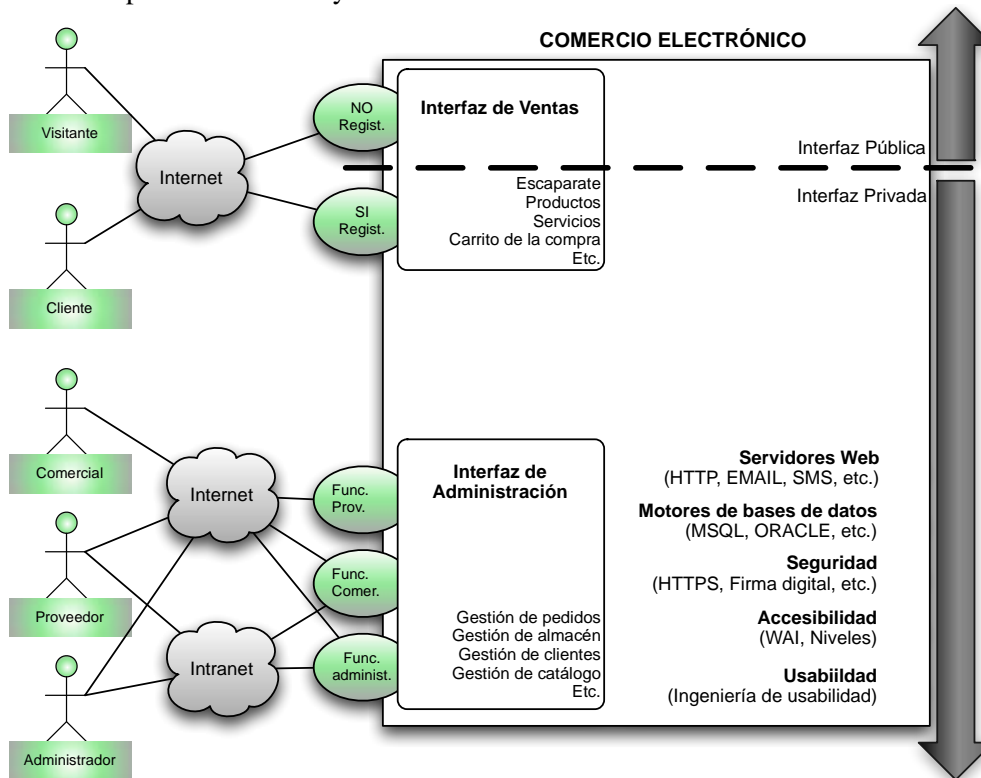
Cuando un cliente realiza una compra de un producto o un servicio, se inicia un flujo de información entre los usuarios que forman parte de la tienda virtual. Entre estos, en líneas general, se pueden destacar los siguientes roles:

- **Visitante.** Son los usuarios de Internet, que aunque visitan la tienda, no han realizado ninguna compra, ni están autenticados y por lo tanto no se les puede considerar clientes.
- **Cliente.** Usuarios registrados y autenticados, no es obligatorio que hayan realizado una compra.
- **Proveedor.** Este rol identifica a las empresas o personas que proporcionan servicios o productos que se venden a través del portal de Internet.
- **Comercial.** Trabajadores pertenecientes al establecimiento de la tienda online. Este es quizás el rol que mas se puede especializar dada la gran variedad de empleados que entran en juego cada vez que se realiza una compra por Internet (responsables de marketing, de ventas, de almacén, distribuidores, etc.).
- **Administrador.** Es el rol con mayor privilegios de la tienda online, tiene el control total y la capacidad de gestionar usuarios, contenidos y pedidos.

Este conjunto de roles, se puede considerar genérico, pero en cada comercio electrónico particular se pueden incluir o eliminar roles (o funcionalidades) en función de las necesidades específicas. Al contar con diferentes roles es necesario que la tienda virtual también disponga de diferentes interfaces para cada uno de ellos. No obstante, es una práctica habitual que varios roles compartan interfaz, pero con distinto nivel de privilegios.

- **Interfaz de ventas o de portada.** Donde los usuarios (roles Visitante y Cliente) pueden realizar las compras y transacciones.
- **Interfaz de administración.** Dónde el resto de roles (Proveedor, Comercial y Administrador) tienen acceso a su funcionalidad determinada en función de nivel de privilegios. Esta interfaz a su vez puede subdividirse en diferentes interfaces que difieren entre si en el tipo de funcionalidad o en el nivel de privilegios asignados:
 - Interfaz de proveedores.
 - Interfaz de comerciales.
 - Interfaz de administración.

Al hablar de diferentes interfaces, nos referimos únicamente a funcionalidad, ya que todas deben poseer el mismo diseño, acentuándose esta característica para la interfaz de compras o de portada que es utilizada por el rol Cliente y Visitante.



3.3.2 Funcionalidad y modularidad

Una comercio electrónico correctamente estructurado debe estar diseñado de forma modular, de forma que pueda ser fácilmente mantenible y escalable en función del crecimiento de las necesidades el mercado. Entre los módulos que cabe destacar dentro de un comercio electrónico tenemos:

- Módulo catálogo (Productos y servicios).
- Módulos de información general.
- Módulos carrito de la compra.
- Módulo encuestas.
- Módulo de gestión de usuarios.
- Módulo de gestión de contenidos.
- Módulo de gestión de pedidos.
- Módulo de estadísticas.
- Módulo de archivos.
- Módulo de publicidad y comunicados.

- Módulo de búsqueda inteligente.
- Módulo de sugerencias.

Al citar estos módulos, una vez más nos estamos refiriendo a funcionalidad, no a cómo los módulos están diseñados a nivel arquitectónico a bajo nivel y por ende como se comunican o relacionan entre sí.

3.3.3 Logística y gestión de pedidos

La venta a distancia, no es algo nuevo y revolucionario. No obstante, parte de los procesos sí que son nuevos y por lo tanto se han tenido que crear, lo que sí que ha supuesto una auténtica revolución. Dentro de estos nuevos procesos y sistemas revolucionarios cabe destacar la logística y la distribución de productos. Es decir, en las ventas tradicionales es el propio cliente quien recoge el producto, en cambio en las ventas realizadas a través de Internet es la tienda es la que le tiene que hacer llegar el producto o servicio al cliente. El envío en el momento adecuado y al lugar apropiado de un producto no es algo sencillo, especialmente cuando hablamos de B2C, es decir, cuando el destinatario es una persona y no una empresa.

En una tienda virtual, hay que considerar algunas premisas que condicionan la logística:

- El producto se ha de hacer llegar al cliente de la forma más rápida posible y menos molesta para el cliente.
- El coste del envío, en muchos casos, supone un freno a las ventas, por lo que es obligado buscar un compromiso entre el buen servicio y el coste.
- El momento del pago (transacción bancaria) también es un factor que puede jugar en contra de las ventas.
- El comprador debería conocer antes de realizar la compra el plazo de entrega de producto.
- Los errores pueden ser catastróficos en logística y por eso los controles de calidad y el seguimiento del proceso deben ser lo más exhaustivos posibles.

Una buena aplicación de comercio electrónico facilita el proceso de distribución, a través de la correcta difusión de la información a todas las partes implicadas (clientes, proveedores, comerciales, etc.). Además, no sólo es importante, proporcionar la información correcta sino el momento en el que se proporciona.

3.3.4 Estadísticas

El marketing es uno de los pilares sobre los que se sustenta cualquier comercio electrónico. El principal mecanismo para aplicar las técnicas de marketing a un comercio electrónico es recoger estadísticas sobre las acciones que realiza cada usuario en la tienda virtual. Por ello, disponer de un módulo de estadísticas completo ofrece una ventaja competitiva fundamental, ya que permite obtener la información necesaria para mejorar el proceso de fidelización de clientes y el comportamiento del sistema de comercio electrónico, en general [21-25].

Dentro del conjunto de estadísticas que se pueden recoger cabe destacar las siguientes:

- Obtención del número de visitas, el número de visitantes reales y su origen. Para ello, una de las herramientas más utilizadas y con los que mejores resultados se obtiene es Google Analytics (www.Google.com/Analytics), que además es gratuita.



- El número de cuentas de clientes, que debe ofrecer una información completa sobre el número de registros de usuarios, los usuarios que han realizado compras, cuáles son los mejores clientes, etc.
- Otros estadísticas a destacar son:
 - Visitantes en línea
 - Mejores clientes
 - Búsquedas de la tienda.
 - Ventas y pedido
 - Mejores proveedores.
 - Mejores productos
 - Mejores descuentos.

3.3.5 Seguridad

Como sucede en el comercio tradicional existe un riesgo, quizás más elevado, que en un sistema de comercio electrónico, al realizar una transacción por Internet, que el comprador sufra una desconfianza al facilitar sus datos personales (nombre completo, dirección postal, datos bancarios, etc.). La seguridad en el comercio electrónico y específicamente en las transacciones comerciales es un aspecto de suma importancia. Por ello, se hace necesario disponer de todas las *mecanismos* que permitan asegurar los siguientes servicios de seguridad:

- **Confidencialidad.** Información accesible únicamente a las partes autorizadas.
- **Autenticación.** Capacidad para asegurar la identificación del sujeto.
- **Integridad.** Asegurar la no manipulación de la información generada en el origen
- **No repudio.** De la operación de emisión o recepción de un mensaje por ninguna de las partes.
- **Control de acceso.** A la información y/o recursos administrados por un sistema.
- **Disponibilidad.** De los fondos o servicios, cuándo se necesiten.

No existe un único mecanismo para asegurar todos los servicios y que así de este modo el sistema no se vulnerable frente a ataques malintencionados. Por tales motivos, se han desarrollado sistemas de seguridad para la transmisión segura de información y para el aseguramiento de las transacciones realizadas a través de Internet. Entre estos sistemas cabe destacar:

- **La encriptación.** Que es un conjunto de técnicas que intentan hacer inaccesible la información a personas no autorizadas. SSL (Secure Sockets Layer) es un protocolo de propósito general para establecer comunicaciones seguras. Hoy constituye la solución de seguridad implantada en la mayoría de los servidores web que ofrecen servicios de comercio electrónico. Esta arquitectura no exige que el servidor disponga de capacidades especiales para el comercio. Basta con que se utilice como mínimo un canal seguro para transmitir la información de pago y el comerciante ya se ocupará manualmente de gestionar con su banco las compras. El canal seguro lo proporciona SSL. El sistema HTTPS utiliza un cifrado basado en las Secure Socket Layers (SSL) para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP. Cabe mencionar que el uso del protocolo HTTPS no impide que se pueda utilizar HTTP.
- **Firmas y certificados digitales.** Un Certificado Digital es un documento electrónico que contiene la identificación de una persona o entidad y la *clave pública* de la misma, haciéndose responsable de la autenticidad de los datos que figuran en el certificado otra persona o entidad de confianza, denominada Autoridad Certificadora. Las principales Autoridades Certificadoras actuales son *Verisign* (www.verisign.es) y *Thawte* (www.thawte.com). Así, se delega a un tercero la autenticidad de la firma. El procedimiento de *firma digital* obtiene un resumen de un documento o de un texto aleatorio y lo cifra con llave privada del propietario del certificado. Cuando nos llega un certificado, y su firma digital asociada, tan sólo debemos obtener nosotros el resumen el mismo, descifrar la firma con la llave pública del remitente y comprobar que ambos resúmenes coinciden, lo que nos hace estar totalmente seguros de la autenticidad del certificado.
- **Protocolo SET (*Secure Electronic Transactions*).** Es un junto de especificaciones diseñadas por VISA y MasterCard, con el apoyo de importantes empresas del sector (Microsoft, Verisign, IMG, etc.) que sigue vigente desde 1996. En el proceso SSL sólo intervienen dos entidades: el Comprador y el Vendedor; SET incluye otras entidades adicionales necesarias para la transacción:
 - *La Pasarela de Pago.* Que permite la comunicación directa a través de Internet entre el comerciante y las Redes Bancarias, con lo que el papel del vendedor queda limitado a un mero intermediario entre el cliente y su banco. Puede ser una entidad independiente o el mismo banco del comerciante.

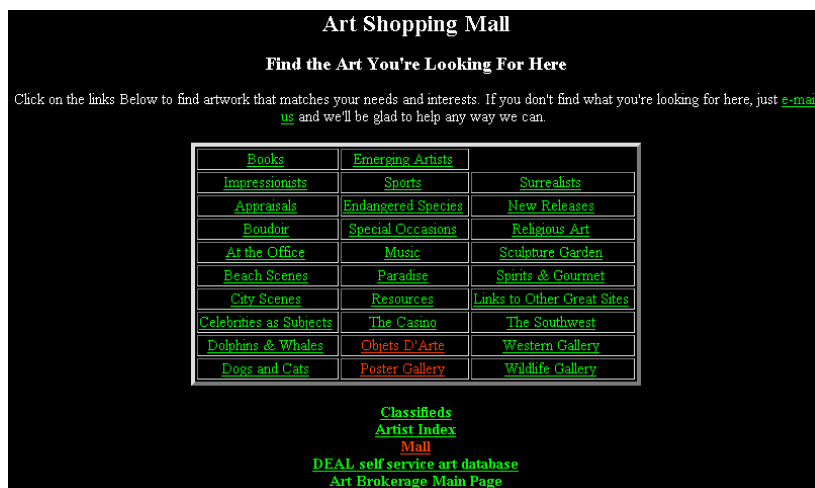
- *Autoridades de certificación.* Que emiten los certificados digitales usados como medio de autenticación de las entidades que intervienen directamente en la operación. Pueden ser entidades independientes autorizadas, bancos o los mismos propietarios de la marca de la tarjeta.

3.3.6 Interfaz, seguridad y confianza

Uno de los principales problemas que tiene el comercio electrónico es la desconfianza de los clientes y consumidores. Gran parte de esa desconfianza se basa en el desconocimiento existente, siendo *La Información* la mejor forma de combatirlo. En gran medida, el grado de desconfianza del cliente, viene dado por dos factores: seguridad e Interfaz.

Por un lado, para combatir el factor seguridad, la mejor herramienta es la Información. Para ello es necesario informar de qué es una zona segura y cómo se produce la certificación por una agencia externa de esa seguridad. Y por otro, con un fuerte servicio de atención al cliente. Si el consumidor sabe que una agencia externa ha auditado la seguridad de esa tienda y, por tanto, da fe de que la tienda es segura y de que esa tienda existe, también sabrá que el servicio de atención al cliente es totalmente fiable. A la tienda le tocará luego seguir una política de marketing que tenga como principal estrategia el servicio de atención al cliente, ya que el cliente es su mayor activo, y su principal objetivo debe ser fidelizar a éste.

Por otro lado, para mejorar la interfaz, es necesario que las páginas utilicen un lenguaje serio, orientado hacia el público, pero sin entrar en matices personales o nomenclatura propia de la empresa, hay que hablar el lenguaje del usuario, pero sin entrar en errores gramaticales ni de ortografía. Más adelante en el apartado de usabilidad se hablará de una forma mucho más detallada.



www.ARNGREN.net Teknologier & Gadgets [Index](#) [el-retur](#) **1909990** [Index](#) (Fra 07.11.2004) [Søk Arngren.net](#) [Slik Betaler du](#) **Bygg di 2-seters Kr. 398,-**

23 Butikker [Index](#) [Video](#) [Forbruker Elektronikk](#) [Ejernet](#) [Kjøpsloven](#) [Angreskjema](#)

Index
 Alvernum
 Alarm
 Alkolester
ATV (eL)
 Bate
 Bil (elektrisk/gas)
 Bilbane
 DAB-Radio
 Digital-Kikkert
Disko-Lys
 Dummy-Kamera
 DVD-Spiller
Elektronikk
 EL-Scouter & Bil
 Flaurer
 Golf-biler (møskit)
 Hobby & RC
 Hoverpog
 HP-Måler (Bil)
 Isbitmaskin
Kamera (drålest)
 Kino (bærbart)
 Kompass (bærb.)
 Laser-Jamer (Bil)
 Luft-Jekk
 Lykt (oppladbar)
 Mobiltelefon-1, 2
 Motorsykkel-Mini
 Omformer (110V)
 Oversetter (td.spek)
 PC-mini (9"), 2
Rakett-Fly
RC Produkter
 Robot-Hund
 Robot-Slavsuger
Roboter
 Spicelle-Flygassk
 Star Wars
 Sykkel-PC s.b.

Micro-Projektor (35 lum) m/Digital-avspiller & HD
Tilbud!
 131 cm lang
1999,-

Jagerfly Turbo-Jet (6ch) Imfelbare-Hjul
 kr. 1999,-

3 - 120" TV/Video-skjerm
 1398,-
 998,- kr. 4.998,-

44 Språk-Talende Oversetter m/Norsk fra kr. 398,-

Spion-Kamera fra kr. 499,-

RC Flybil VTOL (3ch) kr. 249,-

LANDAIR
 Alle Produktene på denne siden lagerføres hos ARNGREN i Oslo. Se Lagerkoden eller Prisen (lev. 2 - 5 dager):
 er på Lager
 kommer før 3 uker
 lengre enn 3 uker

RC Produkter
 120cm langt, Computer-styrt
Video-1 **Video-2**
 RC 4ch Helikopter fra kr. 798,- **Rakett-Fly**

Forbruker Elektronikk
Elektriske-Kjøretøy
Disko-Lys
Rakett-Fly
Roboter [Index](#) [Video](#)

Elektrisk Hummer barn & Ungdom fra 3
Nyhet!
Avatar-Gunship 299,- **Video**

Hummer el-Bil
Forbruker Elektronikk
Tank 72 cm Lang kr. 998,-

RC Produkter
RC kamp-Helikopter sett med Laser-Kanon (3ch) kr. 299,- (pr. sett)
Elektrisk-ATV 6000 watt kr. 59.998,- **EL-Scouter** 5000 watt 0 kr. 29.998,-

RC Fly-Båt 70 cm lang fra kr. 998,-

Elektronikk **Flysimulator** Gratis!
Airwolf-Helikopter m/Imfelbare-Hjul & Lys T-Drev 4ch/RC-12 (8 ch)

Elektronikk **Web-kam.** **Trådløst Mikro Farge-Kamera** sett kr. 499,- **Digital Fotoramme** kr. 199,- **Robot Kamera** med Wi-Fi til PC kr. 2998,-

RC Produkter
RC kamp-Helikopter sett med Laser-Kanon (3ch) kr. 299,- (pr. sett)
Elektrisk-ATV 6000 watt kr. 59.998,- **EL-Scouter** 5000 watt 0 kr. 29.998,-

RC Fly-Båt 70 cm lang fra kr. 998,-

RC Produkter
RC kamp-Helikopter sett med Laser-Kanon (3ch) kr. 299,- (pr. sett)
Elektrisk-ATV 6000 watt kr. 59.998,- **EL-Scouter** 5000 watt 0 kr. 29.998,-

RC Fly-Båt 70 cm lang fra kr. 998,-

RC Produkter
RC kamp-Helikopter sett med Laser-Kanon (3ch) kr. 299,- (pr. sett)
Elektrisk-ATV 6000 watt kr. 59.998,- **EL-Scouter** 5000 watt 0 kr. 29.998,-

RC Fly-Båt 70 cm lang fra kr. 998,-

4 Usabilidad Web

4.1 Introducción

Antes de comenzar este apartado, la primera tarea a realizar es preguntarse: ¿Qué es la usabilidad? El término todavía no forma parte del diccionario de la Real Academia de la Lengua Española, proviene del termino Inglés *Usability* o “able to be used”, es decir, capaz de ser utilizado.

Dos de las mejores definiciones del concepto son los estándares ISO 9241, por un lado, que dice que: *Usabilidad es la eficacia, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico*. Y por otro lado, el estándar ISO 9126 que la define como: *la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso* [26-30].

La primera definición se centra en el concepto de calidad de uso, es decir, la *eficiencia*, la *eficacia* y la *satisfacción*; teniendo en cuenta que esta calidad depende del contexto de uso, de los usuarios y de los objetivos de éstos. Según esta definición, la usabilidad se compone de dos atributos a conocer:

- Atributos cuantificables de forma objetiva
 - La *eficacia* está relacionada con el número de errores que comete el usuario al realizar una determinada tarea.
 - La *eficiencia* está relacionada con el tiempo que tarda un usuario para realizar un determinada tarea.
- Atributos cuantificables de forma subjetiva:
 - La *satisfacción* de uso medible a través de la interrogación al usuario.

La segunda definición, por su parte, se centra en los aspectos internos y externos del producto y señala al usuario como un baremo a considerar (... alcanzar objetivos específicos...). Teniendo en cuenta que no todos los usuarios son iguales, la usabilidad solo podrá ser medible para un contexto particular y un tipo de usuario bien definido. Por lo tanto ambas definiciones ponen al usuario y al contexto de uso como los dos pilares a valorar cuando se habla de usabilidad.

En general, se puede definir usabilidad como el grado con el que los usuarios puede realizar un determinado conjunto de tareas, teniendo en cuenta varios aspectos:

- **Funcionalmente correcto.** El sistema debe proporcionar la funcionalidad para la que ha sido diseñado. Un software que no permita a los usuarios realizar sus tareas no es usable.
- **Eficiente en el uso.** Tiempo o número de acciones requeridas para realizar una tarea.
- **Facilidad de aprendizaje.** Facilidad con la que nuevos usuarios pueden tener una interacción efectiva. Se tienen en cuenta características como predicibilidad, sintetización, familiaridad, la generalización y la consistencia. En general, cuantos menos pasos contenga un procedimiento, más fácil será de aprender.

- **Fácil de recordar.** La carga (grado de uso) de la memoria del usuario necesaria para una tarea, determina cómo de fácil es para los usuarios recordarlo. La interacción que permita alcanzar los objetivos no puede ser tan complicada que no pueda repetirse.
- **Tolerancia a errores.** La eficacia con la que se previenen los errores, la facilidad con la que se detectan e identifican cuando ocurren y por como se corrige la situación de error una vez identificada.
- **Subjetivamente agradable.** La usabilidad a menudo se determina por como los usuarios se sienten usando el sistema. Es necesario que todos los requisitos anteriores provoquen en el usuario la sensación de una experiencia satisfactoria.

Concluyendo, un buen diseño debe ser comprensible, fácil de usar, amigable, claro, intuitivo y de fácil aprendizaje para el usuario. Tal y como veremos, para aprender a hacer sitios Web usables no hace falta estudiar complejas especificaciones técnicas, ni aspectos tecnológicos muy avanzados. Simplemente muchas veces se basa en aplicar el sentido común, cosa que no es fácil, hasta que se descubre cómo. Éste es justamente el objetivo de este apartado: estudiar las distintas estrategias o directrices para lograr diseños de aplicaciones web con alta usabilidad.

4.1.1 ¿Por qué?

La usabilidad no es un término exclusivo de la Web, el término se puede (y debe) extrapolar a cualquier ámbito software, o incluso a cualquier producto o sistema, en general. Si en cualquier ámbito la usabilidad o sencillez de uso es una característica importante que hace que un producto triunfe o no, dentro de un entorno Web la facilidad de uso es una característica crucial e imprescindible. El propio Jakob Nielsen afirma: “*En el caso de productos y programas informáticos, los usuarios experimentan la usabilidad de un sitio antes de que se hayan comprometido a usarlo y antes de que hayan pagado nada*”. De esta afirmación podemos extraer dos conclusiones:

- En programas informáticos, los usuarios pagan por un software que posteriormente utilizarán.
- En el entorno Web, los usuarios experimentan y sólo pagan en el caso de que encuentren aquello que buscan.

Esta afirmación implica que si un usuario después de haber realizado una búsqueda y posteriormente haber accedido a un portal de ventas no encuentra en un determinado tiempo lo que buscaba, está claro que volverá atrás y tratará de buscar el artículo en la siguiente entrada proporcionada por el buscador. Es decir, nadie va a comprar un producto que no es capaz de encontrar. Por lo tanto y como conclusión, no se debe diseñar un sitio Web pensando que los usuarios van a esforzarse en aprender a utilizarlo.

El establecimiento de unos principios de diseño en ingeniería basados en la usabilidad han tenido como consecuencias probadas:

- **Una reducción de los costes de producción.** Los costes y tiempos de desarrollo totales pueden ser reducidos o evitando reduciendo el número de cambios posteriores requeridos en el producto.

- **Reducción de los costes de mantenimiento y apoyo.** Los sistemas que son fáciles de usar requieren menos entrenamiento, menos soporte para el usuario y menos mantenimiento.
- **Reducción de los costes de uso.** Los sistemas que mejor se ajustan a las necesidades del usuario mejoran la productividad, la calidad de las acciones y de las decisiones. Los sistemas más fáciles de utilizar reducen el esfuerzo y permiten a los trabajadores manejar una variedad más amplia de tareas.
- **Mejora en la calidad del producto.** El diseño centrado en el usuario da como resultado productos de mayor calidad de uso, y por lo tanto más competitivos en un mercado que demanda productos de fácil uso.

4.2 Principios de diseño de Nielsen

El mayor experto dentro de la Usabilidad en general y de la Usabilidad Web en particular es Jakob Nielsen, Nielsen definió un conjunto de principios de Usabilidad Web en 1990 que hoy sigue estando vigentes. Nielsen prefirió darles el nombre de *Heurísticas*, que ya se pueden asociar más a un conjunto de reglas de oro, que a un conjunto de directrices específicas para conseguir un diseño usable.

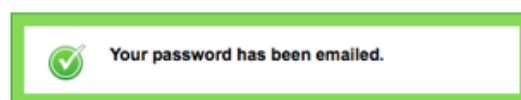
4.2.1 Visibilidad del estado del sistema

El sistema siempre debe proporcionar al usuario información (*feedback*) acerca de dónde está, lo que está ocurriendo y hacia dónde le va a conducir. Teniendo, además, que proporcionar esta información en un tiempo razonable.

Es decir, el usuario debe saber lo que el sistema está haciendo o procesando en cada momento. En diseño web esto se consigue ateniendo a dos principios básicos, por un lado, informar al usuario sobre las acciones que está realizando el sistema. Y por otro lado, informándole en cada momento dónde se encuentra, ya que la desorientación es uno de los mayores problemas existentes al navegar por la red.



Información acerca de lo que el sistema está realizando



Theresa Neil sign in

Información acerca de lo que ha sucedido



Proporcionar este tipo de información es esencial que ya el usuario siente la seguridad necesaria cómo para no preocuparse de la navegación y centrarse únicamente en la información ofrecida.

4.2.2 Correspondencia entre el sistema y el mundo real

Tal y como cita Nielsen, el sistema debe hablar el lenguaje de los usuarios, con palabras, frases y conceptos familiares al usuario, en lugar de utilizar términos orientados al sistema. También es necesario seguir las convenciones del mundo real, para ello se pueden utilizar metáforas. Finalmente, es necesario que la información aparezca en un orden natural y lógico.



Uso de metáforas

Cuando pensamos en los contenidos y la información que vamos a incluir en la web debemos saber quién es nuestro público. A partir de ahí, se evitarán aquellos iconos, palabras o contenidos que no sean fácilmente identificables por los usuarios finales de nuestra aplicación web.

4.2.3 Control del usuario y libertad

Este es uno de los principios que más relacionados está con la navegación del usuario y su adecuado uso contribuye a disminuir la sensación de desorientación a la que se enfrenta el usuario al navegar por Internet. El usuario siempre debe tener la sensación que tiene el control sobre el sistema (y no al revés), para ello, resulta adecuado seguir una serie de buenas prácticas a conocer:

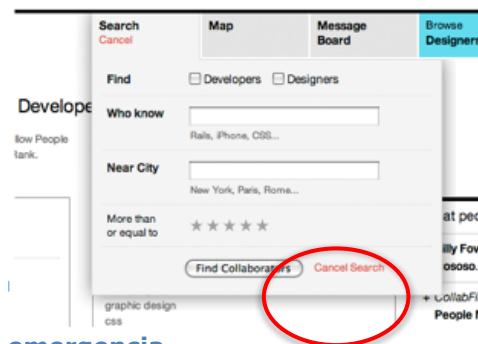
- En la medida de lo posible deben existir las opciones Deshacer y Rehacer.
- No se debe iniciar de manera automática una acción que el usuario no ha solicitado.
- Se deben proporcionar “salidas de emergencia”.
- Es recomendable utilizar un botón “Home” o “Página inicial”. El logo corporativo suele ser utilizado para este fin.

- No es recomendable utilizar ciertas características válidas para un único agente de usuario (navegador), restringiendo de este modo la libertad del usuario sobre el uso de este tipo de aplicaciones.
- Debe ser posible el control sobre el tamaño de la letra. Del mismo modo es necesario conseguir una independencia del tamaño y tipo de pantalla, nuestras páginas se tienen que visualizar correctamente tanto en ordenadores, como en teléfonos móviles o libros electrónicos.
- En procesos de varios pasos, se debe permitir volver siempre a pasos anteriores.

Paso 1. Datos de la empresa

1

Incorporación de pasos en formularios complejos



Uso de salidas de emergencia



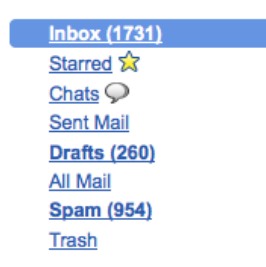
Control sobre el tamaño del texto



Lograr independencia del navegador

4.2.4 Consistencia y estándares

Los usuarios en ningún caso deben preguntarse si diferentes palabras, situaciones o acciones significan lo mismo. Es necesario seguir las convenciones de la plataforma en las que se está desplegando el sitio Web (Barras de navegación, scroll, botones, etc.).



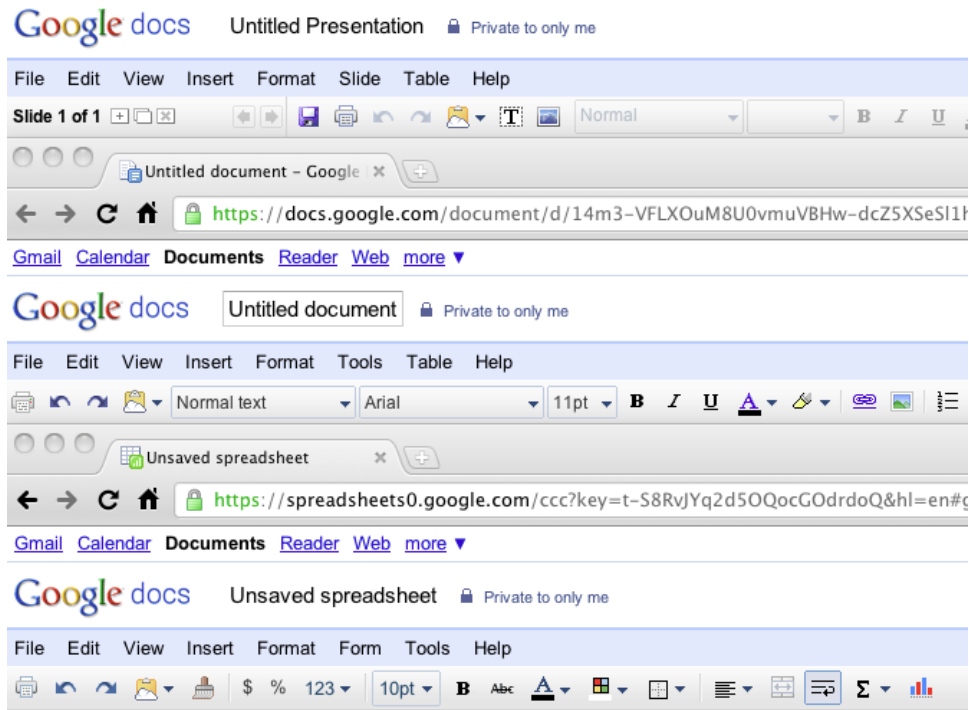
Similitud de Gmail con clientes de correo tradicionales

servicios InternetUn proveedor de (ISP, Internet Service Provider) es un proporciona acceso a Internet. Existen servicios Internet en todo el mundo. Por ejemplo, se marca un número de teléfono y el sistema remoto. Una vez conectado solo a Internet y a otros servicios, como por ejemplo, proporcionados por el ISP. Los ISP aplican cuotas para su uso



Error al utilizar una barra de scroll diferente

Es necesario definir un estilo claro para todo el portal Web, de forma que los mismos elementos realicen la misma acción, el lenguaje sea homogéneo entre todas las páginas del portal y el diseño sea lo más homogéneo posible en toda la Web.



Homogeneizar diseño, botones, pestañas, etc.

Dentro de este principio también se incluye el cumplimiento con las recomendaciones emitidas por el consorcio W3C (<http://www.w3.org/>) referentes a las principales tecnologías Web, entre las que cabe destacar:

- Lenguaje de marcado:

- xHTML 1.1: <http://www.w3.org/TR/2010/REC-xhtml-basic-20101123>
- HTML 4: <http://www.w3.org/TR/1999/REC-html401-19991224>
- HTML5 (En progreso): <http://www.w3.org/TR/2010/WD-html5-20101019/>
- Hojas de estilo:
 - CSS 2.1: <http://www.w3.org/TR/2010/WD-CSS2-20101207/>
- Accesibilidad Web:
 - WCAG 2.0: <http://www.w3.org/TR/WCAG20/>
 - WCAG 1.0: <http://www.w3.org/TR/WCAG10/>



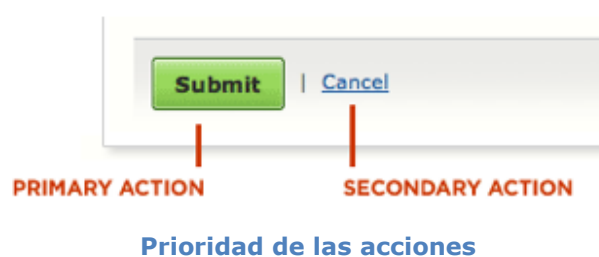
4.2.5 Prevención de errores

Siempre es preferible evitar la posibilidad de que aparezcan errores, a tener diseñados buenos mensajes de error. Existen dos tipos de errores, del sistema y del usuario, los errores del sistema son los que se produce por una mala programación y los errores del usuario se producen por una navegación no adecuada del usuario.

Para evitar errores derivados del uso que hace un usuario de una aplicación es recomendable incluir dentro del diseño toda la información necesaria para prevenir en la medida de lo posible estos errores. La información se debe incluir de forma clara y fácilmente visualizable, evitando cualquier tipo de contaminación visual.

Algunas de las recomendaciones más importantes para cumplir con este principio son:

- Desactivar los botones que no se necesiten usar, o desactivarlos una vez que ya se han utilizado.
- En formularios, situar el cursor (focus) en la primera acción a realizar.
- Resaltar la acción primaria frente a la acción secundaria.



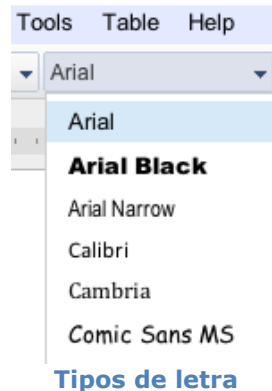
- Autorrelleno de formularios



Autorrelleno de formularios

4.2.6 Reconocer en vez de recordar

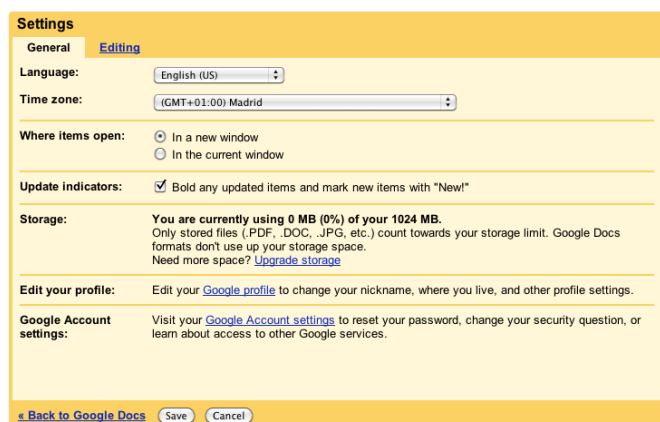
Es más fácil reconocer un objeto, que tener que recordarlo cada que se navega por un sitio Web. Por lo tanto, hay que minimizar la cantidad de memoria que un usuario tiene que utilizar para navegar por un sitio Web. Por ejemplo, el usuario no tiene que recordar información de una parte del diálogo mostrado anteriormente. La jerarquización, agrupación y presentación de contenidos ayudará, guiará y evitará la reiterada petición de información en la navegación del usuario.



Tipos de letra

Una norma para aplicar dentro de esta directriz de diseño es incluir formularios de búsqueda, ya que ningún usuario navegará hasta encontrar lo que busca, simplemente se irá de nuestro sitio en el caso de que no encuentre lo que busque en un tiempo determinado.

Las instrucciones de uso del sistema deben estar visibles y accesibles cuando el usuario las necesite.



Instrucciones de uso visibles

4.2.7 Flexibilidad y eficiencia de uso

Un portal Web debe estar diseñarse tanto para usuarios noveles, como para usuarios avanzados. En este sentido, el uso de aceleradores o atajos facilita al usuario experto la interacción con el sistema, automatizando con ello acciones frecuentes.

La tendencia suele ser centrarse en los usuarios noveles, complicando en exceso la navegación de usuarios avanzados o expertos. Es preciso buscar un equilibrio para que, sin dificultar la navegación novel, el usuario avanzado encuentre atajos que mejoren su experiencia de uso.

Atajos de teclado

Para facilitar la navegación sin ratón y el acceso directo a las principales secciones temáticas

- Estudios: 1
- Investigación: 2
- Internacional: 3
- La Universidad: 4
- Universidad Digital: 5
- Accesibilidad: a
- Ayuda: 6
- Directorio: 7
- Mapa web: 8

Atajos de teclado

Los sitios web se deben de cargar lo más rápidamente posible, independientemente del tipo de conexión a la red utilizada por el usuario. Existen extensiones para los navegadores que permiten medir el tiempo que tarda en cargarse una página, como por ejemplo la extensión YSlow (yslow.es/yslow) del agente de usuario Mozilla Firefox. Existen técnicas que permiten acelerar la carga de las páginas web, según Rich Skrenta se pueden resumir en 14 simples consejos:

- Realizar pocas peticiones http.
- Hacer uso de CDN (Content Delivery Network)
- Utilizar etiquetas para hacer expirar las cabeceras.
- Habilitar la compresión con Gzip.
- Situar las hojas de estilo (CSS) al principio del documento.
- Situar los scripts en javascript (JS) al final del documento.

- Poner tanto los scripts CSS como JS en documentos externos.
- Reducir las consultas DNS.
- Optimizar y minimizar el código Javascript.
- Evitar las redirecciones.
- Eliminar los scripts duplicados.
- Configurar los Etags.
- Hacer Ajax Cacheable.

Presencia en el Mundo | 01.07.11 TLVT \$27.18 +0.67% | Iniciar Sesión | Suscribirse | Clientes | English

Inicio | Nuestra Compañía | Sala de Prensa | Contacto | Áreas de Negocio | Inversores | Trabaja con Nosotros | Siguenos en: [t](#) [f](#) [in](#)

Información Inteligente para un Mundo Sostenible **TELVENT**

Consola HTML CSS Script DOM Red YSlow

Grade Components Statistics Tools Rulesets YSlow(V2) Edit Printable View Help

Grade **E** Overall performance score 73 Ruleset applied: YSlow(V2) URL: http://www.telvent.com/es/

ALL (23) FILTER BY: CONTENT (6) COOKIE (2) CSS (6) IMAGES (2) JAVASCRIPT (4) SERVER (6)

F Make fewer HTTP requests

F

A Use a Content Delivery Network (CDN)

A Avoid empty src or href

F Add Expires headers

F Compress components with gzip

B Put CSS at top

A Put JavaScript at bottom

A Avoid CSS expressions

Grade F on Make fewer HTTP requests

This page has 8 external Javascript scripts. Try combining them into one.
This page has 12 external stylesheets. Try combining them into one.

Decreasing the number of components on a page reduces the number of HTTP requests required to render the page, resulting in faster page loads. Some ways to reduce the number of components include: combine files, combine multiple scripts into one script, combine multiple CSS files into one style sheet, and use CSS Sprites and image maps.

[»Read More](#)

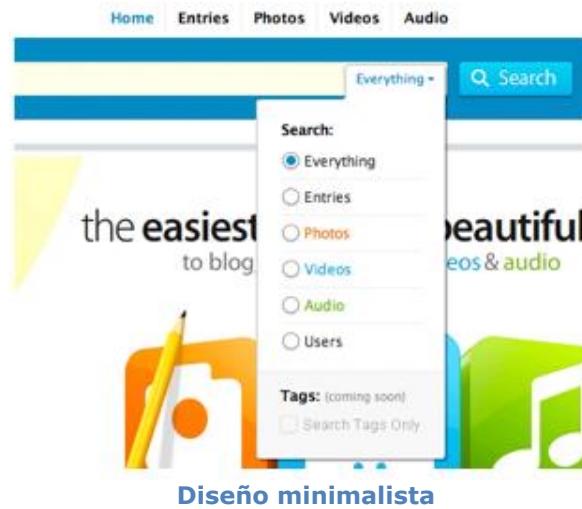
Copyright © 2010 Yahoo! Inc. All rights reserved.

Velocidad de carga www.telvent.com

4.2.8 Estética y diseño minimalista

Sólo se le debe mostrar al usuario la información que necesita en cada momento. Por lo tanto mostrar información irrelevante o no necesaria en un momento dado dificulta la navegación. Por otro lado, hace menos visible la información que realmente es relevante.

En este sentido, lo más complejo es conocer cuál es la información relevante y cuál no lo es. Además, también es necesario adaptarse al nivel de información requerido por cada tipo de usuario en cada momento. Para ello es importante jerarquizar la información de mayor a menor importancia, evitar el ruido visual generado por la acumulación de elementos y dosificar la información, evitando ofrecer todos los detalles de una sola vez [31-35].



Diseño minimalista

Es altamente recomendable que la información más importante sea colocada en la parte superior de la página, pues está es la región que siempre es visible en el navegador.

4.2.9 Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores

Los mensajes de error deben ser expresados como lenguaje natural, en ningún caso como códigos. Al mismo tiempo deben indicar el problema y sugerir una posible solución.

Or start a new account

Choose a username (no spaces)	<input type="text" value="bert"/>	▲ bert is already taken. Please choose a different username.
Choose a password	<input type="password" value="***"/>	▲ Passwords must be at least 6 characters and can only contain letters and numbers.
Retype password	<input type="password"/>	
Email address (must be real)	<input type="text" value="not an email"/>	▲ The email provided does not appear to be valid
<input checked="" type="checkbox"/> Send me occasional Digg updates.		

Mensajes de error apropiados y en lenguaje natural

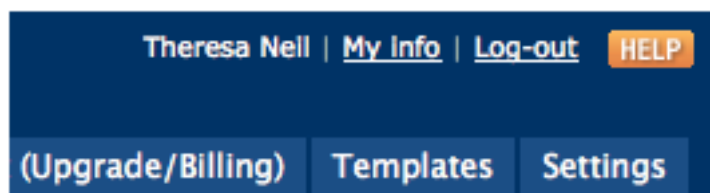
Por otro lado, también es recomendable hacer uso del mensaje de error por defecto que ofrecen los servidores web, los cuales se pueden personalizar.



Mensaje de error por defecto de Amazon

4.2.10 Ayuda y recuperación

Un sistema debe ser diseñado para ser usado sin que sea necesario utilizar el manual de usuario del mismo. En cuanto un sitio web ofrezca alguna característica fuera de la norma, o ligeramente complicada, será necesario ofrecer ayuda a los usuarios. No obstante, siempre es necesario proporcionar ayuda y documentación al usuario. La ayuda debe estar bien organizada mediante un listado de acciones posibles, centrada en las tareas que puede realizar el usuario, no ser demasiado extensa y llevar asociado un buscador.



Ayuda en un portal Web

4.3 Principios de diseño de interacción de Tognazzini

Una vez visto las “heurísticas” de Nielsen, a continuación se van a detallar una serie de directrices que el propio Nielsen recomienda seguir a la hora de diseñar interfaces Web.

Los siguientes principios son fundamentales para el diseño e implementación de interfaces gráficas efectivas, bien se trate de interfaces GUI de escritorio o de la web. Muchas de las aplicaciones web reflejan una falta de entendimiento de muchos de los siguientes principios. Estos principios no cambian aunque se trate de una aplicación web; es más, aplicar estos principios se vuelve más importante.

El conjunto de directrices para el diseño de interfaces Web son las siguientes:

- **Anticipación.** Las aplicaciones deberían intentar anticiparse a las necesidades y deseos del usuario. Muestra al usuario toda la información y herramientas necesarias para cada etapa en su trabajo.
- **Autonomía.** Los usuarios aprenden rápido y ganan confianza cuando se sienten que tienen el control del sistema. Pese a lo que pueda parecer, sin fronteras o restricciones el usuario no se siente libre.

Es necesario mantener informado al usuario del estado del sistema, además este estado debe ser mostrado de manera visible y actualizada.

- **Daltonismo.** Cuando se utiliza color para transmitir información se debe utilizar otros elementos complementarios para la gente con daltonismo. Las pistas secundarias pueden consistir en distintos tonos de gris, gráficos complementarios o etiquetas de texto.
- **Consistencia.** La consistencia más importante es aquella que espera el usuario, por lo tanto, es necesario probar con ellos. Se debe mantener una consistencia estricta. En la siguiente lista aparecen los elementos de la interfaz ordenados por su necesidad de consistencia, de mayor a menor.
 - Interpretación del comportamiento del usuario. Los atajos de teclado deben funcionar siempre del mismo modo.
 - Estructuras visibles, las estructuras invisibles se refieren a propiedades no visibles a primera vista, como el menú contextual, que se obtiene haciendo clic en el botón derecho del ratón.
 - Estructuras visibles pequeñas, como por ejemplo las flechas de desplazamiento.
 - El aspecto general de una aplicación.
 - Una suite de productos
 - Consistencia interna
 - Consistencia con la plataforma
- **Valores por defecto.** Se deben utilizar valores por defectos, no obstante, estos deben ser descartados con facilidad y rapidez.
- **Eficacia del usuario.** En primer lugar hay que tener en cuenta que los saltos cualitativos en eficiencia se encuentran en la arquitectura del sistema, no en su superficie, es decir en el diseño visual de la interfaz. No obstante, es recomendable perseguir los siguientes principios:
 - Buscar la productividad del usuario, no del ordenador.
 - Mantener ocupado al usuario.
 - Maximizar la eficacia de la organización a través de maximizar la eficacia de todos los involucrados.
 - Escribir mensajes de ayuda concisos y que ayuden a resolver problemas.
 - Los menús y etiquetas de botones deben comenzar por la palabra más importante.

- **Interfaces explorables.** Es importante proporcionar al usuario caminos de navegación bien señalizados, una vez que tiene claro cuál es el camino, es interesante proporcionar atajos que faciliten la navegación.
 - No es necesario encerrar a un usuario en un único camino, no obstante, se le puede ofrecer la ruta más simple.
 - Darle a los usuarios nociones estables de cómo llegar al inicio.
 - Hacer que todas las acciones sean reversibles.
 - Incluir en la medida de lo posible la acción deshacer, sino es posible, avisar al usuario de los posibles riesgos.
 - No atrapar a los usuarios en una página Web, dejar siempre una salida clara.
 - Ley de Fitt, el tiempo necesario para alcanzar un objeto es una función de la distancia y el tamaño del objeto en si.
- **Reducción de latencia.** Algunas de las técnicas para reducir o minimizar el tiempo de espera son las siguientes:
 - Comunicar el clic de los botones.
 - Mostrar un reloj de arena para acciones de más de medio segundo y menos de 2 segundos.
 - Mostrar un mensaje comunicando acciones de más de 2 segundos.
 - Comunicar el tamaño y progreso de acciones mediante barras de estado.
 - Hacer que el usuario esté entretenido mientras se produzca una espera.
 - Utilizar técnicas de programación para aumentar la velocidad (Multithreat).
- **Aprendizaje.** Lo ideal es que no exista periodo de aprendizaje, pero en la realidad siempre existe un periodo de aprendizaje. Es necesario acortar este tiempo en la medida de lo posible.
- **Uso de metáforas.** Es una buena práctica escoger aquellas metáforas que permitan al usuario comprender los detalles del modelo conceptual
- **Protege el trabajo del usuario.** Es necesario asegurarse de que el usuario nunca pierde el trabajo por un error, ya sea suyo, del sistema o del contexto (Internet, corte de suministro eléctrico, etc.).
- **Legibilidad.** Es necesario utilizar texto con alto contraste (negro sobre blanco), evitar los fondos grises cuando haya texto. Utilizar un tamaño de letra adecuado y permitir su modificación.

- **Guardar estado.** La Web utiliza un protocolo sin estado, es necesario utilizar los métodos que permitan guardar el estado de un usuario.

La información del estado debe almacenarse en una cookie durante la sesión en el ordenador cliente, luego debe almacenarse en un servidor, de forma que el usuario siempre disponga de la misma interfaz con independencia del dispositivo desde el que se conecte.

- **Navegación visible.** Se debe evitar la navegación invisible, la mayoría de usuarios no puede mantener mapas mentales complejos. Aunque la Web es, en si misma, un espacio de navegación invisible. Una vez que el usuario llega a un portal, debemos procurar reducir al máximo la navegación y ofrecer de forma clara y natural el contenido. Hay que procurar que el usuario tenga la sensación de estar siempre en el mismo sitio.

4.4 Ingeniería de la Usabilidad

La ingeniería del software es una disciplina que trata de todos los aspectos de la producción de software. Existen varias definiciones que resultan interesantes:

Ingeniería del software es el establecimiento y uso de principios sólidos de ingeniería, orientados a obtener software económico que sea fiable y trabaje de manera eficiente en máquinas reales.

Bauer, 1972

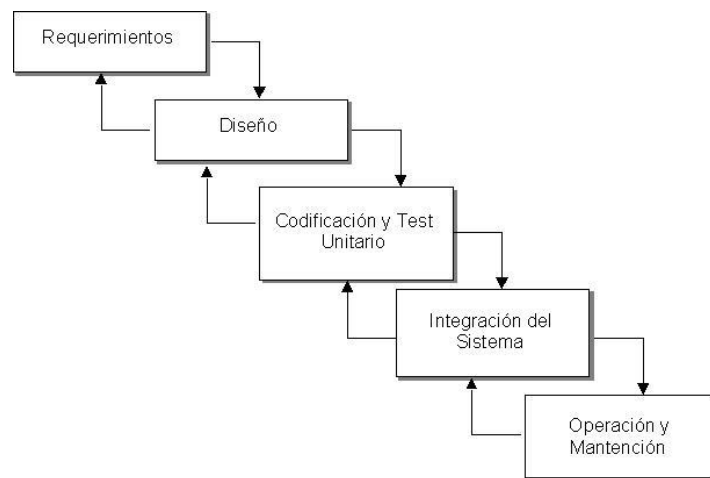
Tratamiento sistemático de todas las fases del ciclo de vida del software. Se refiere a la aplicación de metodologías para el desarrollo del sistema software

AECC, 1986

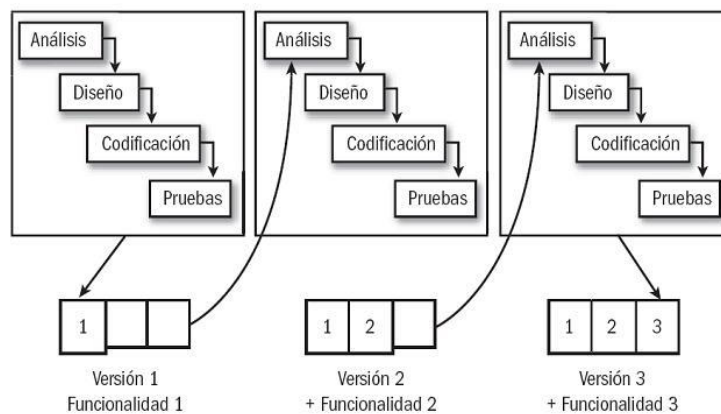
La aplicación disciplinada de principios, métodos y herramientas de ingeniería, ciencia y matemáticas para la producción económica de software de calidad

Humphrey, 1989

Los ingenieros de software deben entonces adoptar un enfoque sistemático y organizado de su trabajo y usar herramientas y técnicas apropiadas dependiendo del problema que van a solucionar, las restricciones de desarrollo y los recursos disponibles. Uno de los modelos estructurados más básicos, y que sirve como bloque de construcción para los demás modelos de ciclo de vida, fue el modelo en cascada que se puede apreciar en la figura siguiente.

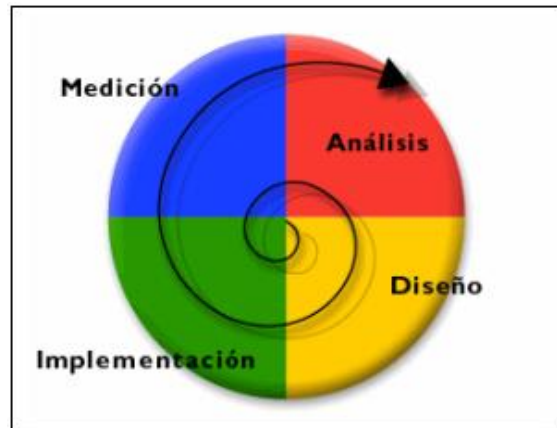


Actualmente ya está en desuso y se tiende utilizar a modelos de desarrollo rápido de forma iterativa e incremental.



Estos cambios de paradigmas en los modelos y procesos de desarrollo de software imponen también un cambio de paradigma en la interacción usuario-sistema en sus diferentes niveles, lo que ha provocado el origen de una metodología orientada a asegurar una interacción usuario-sistema más natural (cómoda) y eficiente (productiva), así como facilitar la comprensión del sistema por parte de nuevos usuarios, eliminando inconsistencias en la interacción. Desde un enfoque de diseño y evaluación de aplicaciones Web, hablamos de Ingeniería de la Usabilidad como un área incluida en el campo de la IPO (Interacción Persona Ordenador) que se define como un conjunto de fundamentos teóricos y metodológicos que aseguran el cumplimiento de los niveles de usabilidad requeridos.

La Ingeniería de la Usabilidad es multidisciplinar; se nutre de la informática, de la psicología, de la lingüística, de la sociología, de la antropología y del diseño industrial.



El Modelo de Proceso de la Ingeniería de la Usabilidad especifica una metodología que guía al equipo de desarrollo de aplicaciones interactivas con altos niveles de usabilidad. Este modelo, tal y como ya se ha avanzado tiene sus cimientos por una parte en la Ingeniería del Software y por otra en la Interacción Persona-Ordenador

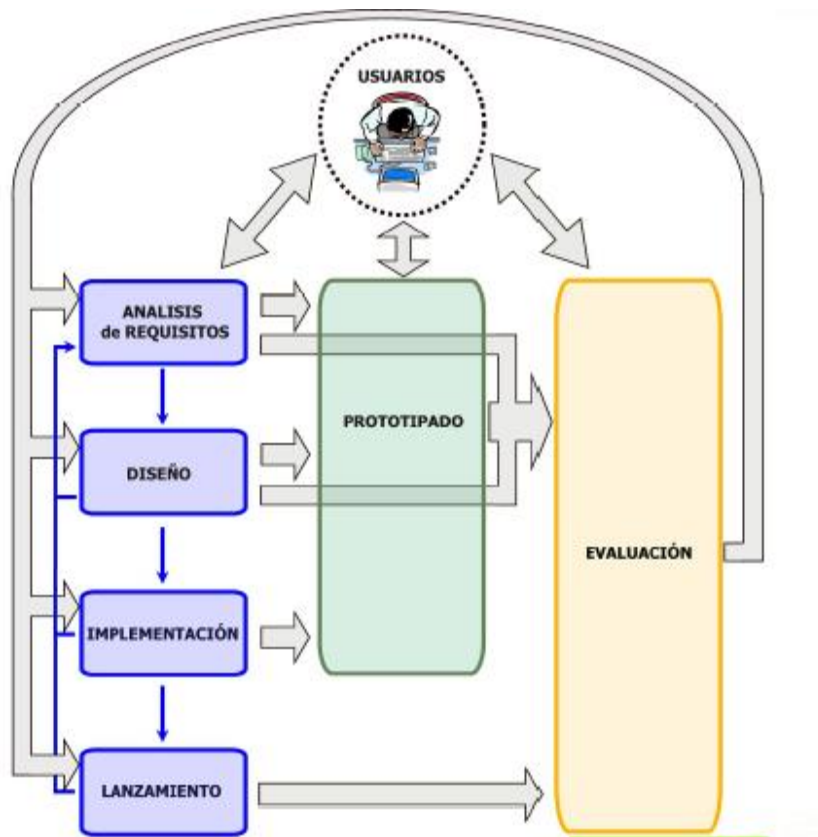
Pretende ser una herramienta de trabajo para ayudar metodológicamente a los equipos de desarrollo. No especifica ni el uso de un determinado lenguaje de programación, ni ninguna tecnología específica, ni ningún factor que pueda determinar la aplicación, sino todo lo contrario, está pensado para todo tipo de aplicaciones y tecnologías, en definitiva, es independiente de los dispositivos y la tecnología [36-40].

4.4.1 Modelo del proceso de diseño de usabilidad

Un método basado en la usabilidad debe disponer de un esquema claro y consistente, que permita ver claramente al usuario de este método y que sirva como guía en todo momento para saber en que fase del desarrollo se encuentra y que posibilidades se tienen a partir de la fase actual de continuar su desarrollo. Tal y como se aprecia en la siguiente figura, el modelo se encuentra estructurado en tres pilares básicos:

- Ingeniería del Software. En el proceso de Ingeniería de la usabilidad se definen cuatro fases principales: Análisis de requisitos, Diseño, Implementación, Lanzamiento.
 - En color azul en el esquema (izquierda).
- Prototipado. Engloba técnicas que permitirán la posterior fase de evaluación.
 - En color verde en el esquema (centro).
- Evaluación. Engloba y caracteriza los métodos evaluación existentes.
 - En color amarillo en el esquema (derecha).

Así mismo, dentro de este esquema hay que tener en cuenta por un lado al Usuario, dejando claro que este proceso se centra en él. Y por otro lado al método iterativo que se sigue de forma que mediante un conjunto de iteraciones sobre el producto objetivo se consigue el producto final, en el esquema se aprecian las flechas que guían el proceso del software.



A continuación en los siguientes apartados se desarrollarán más detalladamente cada uno de estos subprocesos dentro del diseño centrado en el usuario.

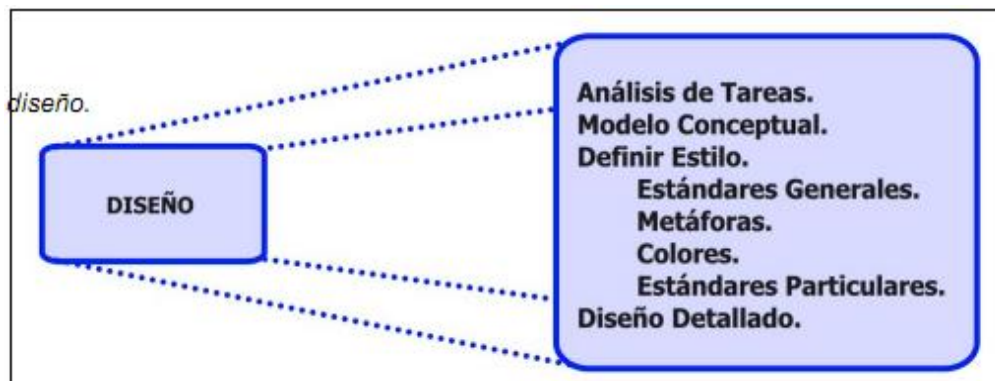
4.4.1.1 Ingeniería del software

En este apartado se detallan las fases que componen el modelo de Ingeniería del software:

- **Análisis de requisitos.** De los resultados de esta fase depende el resto de proyecto, por lo tanto es necesario formular adecuadamente el problema de diseño: determinando concretamente la audiencia y las plataformas destino, las metas del usuario y los requisitos técnicos, así como las necesidades de los usuarios y los requisitos específicos de usabilidad.



- Diseño conceptual.** Como resultado de esta fase se debe alcanzar una idea clara de cómo será la interfaz de usuario y las relaciones con ésta para desarrollar las especificaciones funcionales que sirvan de guía al diseño posterior. Cada tipo de interfaz tienen particularidades específicas, que hay que tener en cuenta en el momento de desarrollar los prototipos, pero que también hay que tenerlas en cuenta en esta etapa (de diseño) porque pueden afectar a las funcionalidades de la interfaz y pueden condicionar el resto de requisitos. Métodos a utilizar esta etapa: Casos de uso, análisis de tareas y sobre la Arquitectura de la Información, que se refiere al diseño, organización, etiquetado, navegación y sistemas de búsqueda que ayudan a los usuarios a encontrar y gestionar información de manera efectiva.



- Implementación.** En esta fase se crea el producto final. El modelo de esta etapa puede utilizar cualquier proceso o modelo de la ingeniería del software clásica (cascada, espiral, iterativo, desarrollos rápidos, etc.).

Lanzamiento. Es una de las fases más críticas, ya que es cuando se comprueba si el esfuerzo del desarrollo ha obtenido el éxito buscado inicialmente. La percepción que el usuario final tiene un peso específico enorme a la hora de indicar si el producto será aceptado o no. El proceso seguido mediante la Ingeniería de la Usabilidad asegura que tanto los usuarios, como los responsables de éstos se sientan satisfechos con el producto creado. En esta fase también resulta importante recoger la retroalimentación o impresiones proporcionadas por el usuario del sistema (*user feedback*). Esta información recogida se puede utilizar para:

- Proporcionar una entrada para el mantenimiento y posibles mejoras del producto.

- Proporcionar una entrada para la implementación de futuras revisiones del producto.
- Proporcionar una entrada para el diseño y desarrollo de productos relacionados que serán utilizados por los mismos usuarios o de características similares.
- Incrementar el autoaprendizaje en cuanto a la usabilidad

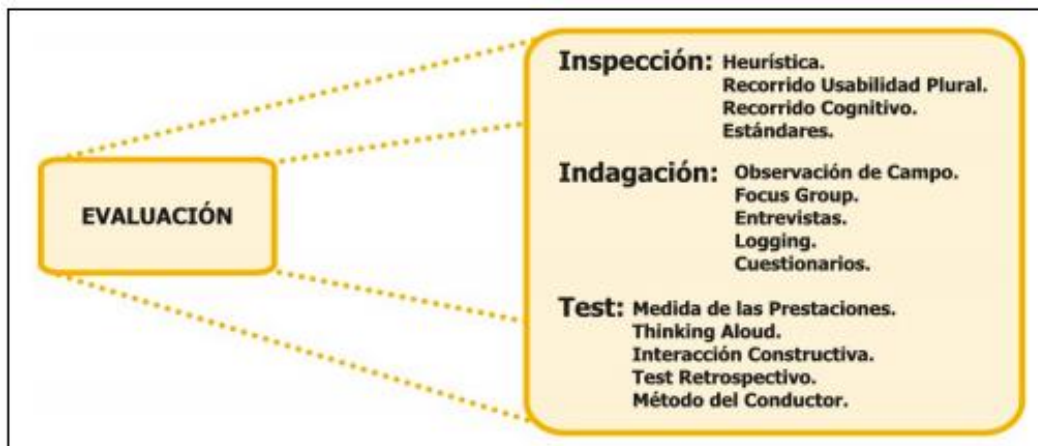
4.4.1.2 Prototipado

Los prototipos son diseños, documentos o sistemas que simulan o tiene implementadas partes del sistema final a desarrollar. Los prototipos son muy importantes para diseñar un buen sitio Web, ya que facilitan la planificación del proceso de creación, reducen el coste de la evaluación, aumentando su efectividad, reduciendo de este modo los errores de diseño.

Un prototipo se realiza para evaluar un diseño prematuramente, Por lo que el objetivo es producir estos prototipos rápidamente y evaluarlos eficazmente para que puedan ser refinados, elaborados y reevaluados antes del producto final.

4.4.1.3 Evaluación de la usabilidad

Todas las fases de un proyecto de este tipo, que hemos visto en el apartado de Ingeniería del software, llevan asociada una realimentación que verifica de forma empírica si se alcanzan los objetivos deseados.



Tal y como se aprecia en la figura anterior, habitualmente los métodos existentes se pueden clasificar en tres categorías:

- **Inspección.** Aglutina un conjunto de métodos orientados a la evaluación de la usabilidad mediante el conocimiento experto, es decir, haciendo uso de expertos para evaluar el grado de usabilidad. Los métodos más importantes dentro de esta categoría son:
 - Heurística. El método se basa en evaluar las 10 heurísticas propuestas por Nielsen a partir de 249 problemas de usabilidad.
 - Recorrido de usabilidad plural. Evaluación del modelo en papel por varios expertos y posterior debate final entre ellos.

- Recorrido cognitivo. Se centra en la evaluación de la facilidad de aprendizaje del sistema, el método de aprendizaje más utilizado suele ser la exploración. Por lo que los revisores exploran la interfaz y evalúan este proceso para un conjunto de tareas concretas.
- Estándares. Un experto comprueba el cumplimiento de un conjunto de estándares.
- **Indagación.** Los métodos de evaluación realizados por indagación hay un trabajo de interrogación y observación a los usuarios que hacen uso de un determinado sistema. Dentro de los métodos más utilizados tenemos:
 - Observación de campo. El principal objetivo consiste en observar a los usuarios para entender cómo realizan las tareas y saber cuál es el modelo mental que tienen sobre ellas. Esta información se completa con preguntas o entrevistas personales.
 - Focus Group. El método de Grupo de Discusión Dirigido es una técnica de recolección de datos dónde se reúnen de 6 a 9 usuarios para discutir aspectos sobre el sistema. Un experto hace de moderador y recoge la información relevante de la discusión.
 - Entrevistas. Las entrevistas con los usuarios es una forma directa y estructura de recoger abundante información sobre un sistema interactivo. Las entrevistas suelen aportar información muy valiosa sobre aspectos que a veces no se han tenido en cuenta. Las entrevistas se pueden complementar con los cuestionarios.
 - Logging. Esta técnica consiste en guardar toda la información sobre las actividades que realiza un usuario con el sistema para su posterior análisis.
 - Cuestionarios. Un cuestionario es el complemento perfecto de una entrevista, pero es menos flexible. La principal ventaja es que se puede llegar a un mayor número de usuario en un tiempo menor y se pueden evaluar con mayor rigor.
- **Test.** Para la evaluación se utilizan los datos producidos mediante del uso que hacen un conjunto de usuarios en la interfaz de un determinado sistema. Los métodos más utilizados son:
 - Medida de las prestaciones. El objetivo es mejorar el producto a través del análisis de usuarios reales, en tareas también reales.
 - Thinking aloud. Método introducido por Nielsen, en él se les pide a los usuarios que describan en voz alta sus pensamientos, sentimientos y opiniones sobre un sistema mientras lo están utilizando.

- Interacción constructiva. Es una variante del método anterior, con la diferencia que se realiza el test con un grupo de usuarios al mismo tiempo.
- Método del conductor. En los métodos anteriores el usuario es el que tiene el libre albedrío de realizar las tareas que desee, en esta variante, existe un conductor que dictamina cuáles son las tareas a realizar.
- Test retrospectivo. Este método es complementario a los anterior, ya que consiste en grabar otros métodos en video y analizar posteriormente la grabación.

5 La elección de la tecnología, como condiciona el futuro del negocio

5.1 Introducción

A la hora de crear un portal de ventas en Internet, uno de los problemas iniciales consiste en decantarse por la tecnología a utilizar. Una de las opciones disponibles puede ser decantarse por una solución prefabricada, lo que nos aportará una gran rapidez en el desarrollo, pero tendremos que ajustar nuestras necesidades a las características de la solución elegida. Entre estas soluciones tenemos:

- Os-commerce - <http://www.oscommerce.com/>
- Drupal - <http://drupal.org/>
- Joomla - <http://www.joomla.org/>

Por el contrario, si necesitamos un grado de personalización mayor, tendremos que comenzar un desarrollo desde cero. El primer paso será elegir por el tipo de tecnología a utilizar a lo largo del desarrollo, algo que siempre estará ligado a las necesidades del portal de ventas que se desee construir. Entre las características que tenemos que tener en cuenta se encuentran:

- Tipo de portal (Estático o dinámico).
- Tipo de licencia (Abierta o propietario).
- Tamaño del portal (Pequeño, Mediano, Grande)
- Crecimiento del negocio (escalabilidad)
- Necesidad de cambios (Mantenibilidad y actualización).
- Número estimado de clientes.
- Tipo de soporte de la tecnología (Comunidad de usuarios, empresa, etc.).
- Necesidad de interactividad.
- Seguridad (Frecuencia de actualizaciones de seguridad).
- Accesibilidad para usuarios discapacitados.
- Indexación en buscadores.
- Usabilidad de los componentes proporcionados.

- Velocidad de carga.
- Tiempo de desarrollo.
- Capacidad de procesamiento.
- Etc.

Además de esta relación de características, existen otras características específicas de cada tipo de portal o negocio, incluso también puede influir la línea corporativa de la propia empresa.

A lo largo de este apartado, se presentará una revisión del estado del arte de las principales tecnologías Web, con el objetivo de dar a conocer cuáles son, sus ventajas y debilidades, facilitando de este modo la elección tecnológica a la hora de desarrollar un portal de ventas a través de Internet [41-45].

5.2 Alternativas tecnológicas

5.2.1 Lenguajes

5.2.1.1 El lenguaje HTML y CSS

HTML (*HyperText Markup Language*) es un lenguaje muy sencillo que permite describir hipertexto, es decir, *texto presentado de forma estructurada* y agradable, con vínculos o enlaces (*hyperlinks*) que conducen a otros documentos o fuentes de información relacionadas y con inserciones multimedia (gráficos, sonido, etc.).

El lenguaje HTML debe funcionar en diferentes plataformas y navegadores. Se ha desarrollado con la premisa de que cualquier tipo de dispositivo debería ser capaz de usar información de la Web (ordenadores, teléfonos móviles, PDA's, etc). Por ello, todas las versiones de este lenguaje están ampliamente estandarizadas por el organismo W3C (*World Wide Web Consortium*).

Por su parte el lenguaje CSS es un lenguaje que permite definir, mediante una sintaxis especial la presentación de un documento HTML o bien de un sitio Web, facilitando la consistencia de estilo en todas las páginas del sitio.

Ambos lenguajes, HTML y CSS, tienen características similares por lo que es posible su agrupación. También se incluye dentro de este grupo el lenguaje XHTML ya que su única diferencia con HTML es que es éste más restrictivo respecto al uso de etiquetas.

Lenguaje HTML y CSS	
Ventajas	Inconvenientes
<ul style="list-style-type: none"> • Sencillo • Fácil de aprender • Despliegue rápido • Archivos pequeños, carga rápida. 	<ul style="list-style-type: none"> • Estático

<ul style="list-style-type: none"> • Es admitido por todos los navegadores • Desarrollo mediante editor WYSIWYG (<i>What You See Is What Your Get</i>) 	<ul style="list-style-type: none"> • Cada navegador interpreta de forma diferente el lenguaje, es decir, no todos los navegadores implementan a la perfección los estándares de la W3C. • Mantenimiento complejo, tedioso y lento. • Conjunto de etiquetas muy limitadas. • Diseño simple.
--	--

```
<html>
  <head>
    <title>Hola Mundo</title>
  </head>
  <body>
    <p>Hola mundo!</p>
```

Hola Mundo en HTML

5.2.1.2 Lenguaje Javascript

El lenguaje Javascript es un lenguaje interpretado (el ordenador va leyendo cada instrucción del programa, la traduce y la ejecuta). Los programas escritos con este lenguaje son conocidos como *script*. Con Javascript no se pueden construirse programas independientes, sólo pueden escribirse scripts que funcionarán en el entorno de una página Web, interpretándose por un explorador, teniendo en cuenta que no todos los exploradores integran en la misma forma los guiones JavaScript, pese a estar estandarizados.

El uso de Javascript se justifica en que con un buen programa editor podemos obtener una página para publicar en la red, pero esa página Web consistiría en: texto, imágenes e hipervínculos, con los atributos como colores, tipos de letra y poco más sobre los que el autor puede actuar. Si queremos ir más allá de la simple presentación de un documento en pantalla y queremos controlar al explorador no hay más remedio que utilizar “algo más”, en este caso Javascript.

En Javascript, lo normal es que la ejecución se realice de forma automática cuando el navegador carga una página, o cuando el usuario pasa el ratón por una imagen, etc. Estos cambios provocan los llamados eventos que son recibidos por el navegador que reaccionará en la forma adecuada: si haces clic en un hipervínculo se genera un evento y el navegador abre una nueva página. Esos eventos son los que se aprovechan para que se ejecuten las instrucciones que nosotros escribimos en código Javascript. A cada evento se le puede asociar una función para que haga algo predeterminado por nosotros. Por ejemplo, cuando pasamos el ratón por una enlace se produce otro evento que puede aprovecharse para llamar a una función que modifique el color en que se muestra el enlace, o cuando el usuario pulsa una tecla.

Lenguaje JavaScript	
Ventajas	Inconvenientes

<ul style="list-style-type: none"> • El código Javascript se ejecuta del lado del cliente, por lo que se descentraliza las necesidades de cálculo. • Fiable si se siguen unas mínimas normas de uso. 	<ul style="list-style-type: none"> • Código es visible por cualquier usuario. • El código debe descargarse, aunque no se vaya a ejecutar. • Inmune frente ataques de tipo XSS (<i>Cross Site Scripting</i>).
--	---

```

<html>
  <head>
    <title>Hola Mundo</title>
    <script type="text/javascript">
      alert("¡Hola mundo!");
    </script>
  </head>

```

Hola mundo en Javascript

5.2.1.3 El lenguaje PHP

El lenguaje PHP (*PHP: Hipertext Preprocessor*) es un lenguaje interpretado de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Es un lenguaje gratuito por lo que puede utilizarse en cualquier desarrollo y ser desplegado en la mayoría de los servidor Web. Su estructura es similar a lenguajes estructurados como C, y sobre todo Perl, lo que permite a la mayoría de los usuarios crear aplicaciones complejas en un tiempo muy corto, debido a la facilidad de aprendizaje.

Dos de sus principales características son, por un lado, soporte para la mayoría de bases de datos existentes en el mercado. Y por otro lado, la gran cantidad de funciones existentes a través de módulos (o ext.), destacando también la gran documentación de todas estas funcionalidades a través de su página Web (<http://www.php.net>).

Lenguaje PHP	
Ventajas	Inconvenientes
<ul style="list-style-type: none"> • Fácil de aprender • Desarrollo muy rápido. • Es orientado a objetos (cada vez más). • Multiplataforma (Unix, Windows, Mac, etc.) • Conexión con múltiples bases de datos (MySQL, PostgreSQL, Oracle, etc.). 	<ul style="list-style-type: none"> • La potencia de cálculo se centraliza en el servidor. • El código suele ser bastante desestructurado e ilegible, por lo que es difícil de escalar. • La programación orientada a objetos es todavía deficiente. • El diseño arquitectónico es complejo debido a las dificultades de modularización

<ul style="list-style-type: none"> • Extensibilidad de sus funcionalidades mediante el uso de módulos (Ext. o Extensiones) • Documentación amplia y bien organizada. • Es libre y gratuito 	<p>y organización en capas de cada aplicación.</p> <ul style="list-style-type: none"> • PHP es un lenguaje tradicionalmente inseguro.
---	--

```

<!DOCTYPE HTML PUBLIC "-//
//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Ejemplo php</title>
  </head>
  <body>
    <?php
        echo "Hola mundo!";
    ?>
  </body>
</html>

```

Hola mundo en PHP

5.2.1.4 Lenguaje ASP y ASP.NET

ASP es el acrónimo en inglés de *Active Server Pages*, es la apuesta de Microsoft para la creación de páginas Web dinámicas del lado del servidor. Dado su gran vinculación con esta gran empresa, este lenguaje está muy ligado al desarrollo tecnológico de Microsoft y por ello se programa siguiendo la filosofía de otros lenguajes como Visual Basic o C#. Este modelo de desarrollo tiene una gran cantidad de limitaciones, como por ejemplo, el excesivo código necesario para funcionalidades simples, código desorganización, etc. Pese a la gran cantidad de limitaciones que tiene esta filosofía en entornos Web, también existen algunas ventajas como el uso de controles ActiveX, así como componentes del lado del servidor.

Para dar solución a la mayoría de los inconvenientes de esta tecnología, Microsoft desarrolló el Framework para aplicaciones Web ASP.NET, que es la tecnología que ha sucedido a la antigua y anticuada ASP. La ventaja que tiene ASP.NET es que permite el uso de cualquier lenguaje admitido por el Framework .NET gracias a que está construido sobre lo que se denomina Common Language Runtime. ASP.Net introduce un nuevo nivel de abstracción en la construcción de sitios Web, siguiendo la filosofía de inclusión de controles incluidos en el Framework u otros gratuitos, ocultando el código de los mismos y facilitando por tanto su uso.

Lenguaje ASP	
Ventajas	Inconvenientes

<ul style="list-style-type: none"> • Es fácil de programar. • Uso óptimo de todos los lenguajes y herramientas de Microsoft (JScript). 	<ul style="list-style-type: none"> • Código desorganizado. • Es necesario mucho código para funciones complejas. • Tecnología propietaria. • Hospedaje en caro.
Lenguaje ASP.NET	
Ventajas	Inconvenientes
<ul style="list-style-type: none"> • Orientado a Objetos • Diseño arquitectónico en capas de aplicación y diseño. • Código mantenible. • Velocidad • Seguridad 	<ul style="list-style-type: none"> • Gran consumo de recursos.
<pre> <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"> <html> <head> <title>Ejemplo php</title> </head> <body> <% Response.Write("Hola Mundo") %> </body> </pre>	

Figure 1 - Hola mundo en ASP

5.2.1.5 Lenguaje Java – JSP

JSP es una tecnología Java que permite generar código dinámico en documentos HTML o XHTML. El lenguaje JSP incrusta código Java en scripts dentro de páginas HTML. Fue desarrollado por Sun Microsystems y actualmente es mantenido por Oracle.

El principal avance de las páginas JSP frente a otras tecnologías es que éste utiliza Java, que es un lenguaje de propósito general y por lo tanto es más simple separar la lógica de negocio (Java) de la visualización del usuario en la Web (HTML+CSS+JSP).

Utilizando la tecnología JSP el servidor interpreta el código de la página JSP, construyendo un servlet de Java. Posteriormente, será este *servlet* el que genere el documento HTML que recibirá el usuario en su navegador.

Lenguaje Java (JSP - Java Server Pages)	
Ventajas	Inconvenientes
<ul style="list-style-type: none"> • Se ejecutan del lado del servidor • Multiplataforma • Código bien estructurado. • Integración con módulos y componentes de Java. • Habitualmente es un lenguaje seguro. 	<ul style="list-style-type: none"> • Complejidad de aprendizaje. • Es lento.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Ejemplo php</title>
  </head>
  <body>
    <%
      out.println("Hola mundo!<br/>");
    %>
  </body>
</html>

```

Figure 2 - Hola mundo en JSP/Java

5.2.1.6 Ruby on Rails (RoR)

El lenguaje Ruby on Rails es un framework de programación de aplicaciones Web escrito en Ruby. Sigue la filosofía *Don't Repeat Yourself*, lo que viene a decir que *No te repitas*. Es decir, significa que las definiciones deben hacerse una única vez, los componentes están integrados, por lo que no es necesario establecer conexiones con ellos, además también existen convenciones sobre configuración, lo que implica que el programado sólo necesita definir aquella configuración que no es convencional.

Lenguaje RoR	
Ventajas	Inconvenientes
<ul style="list-style-type: none"> • Software libre y multiplataforma • Es un lenguaje orientado a objetos • Sigue el patrón Modelo-Vista-Controlador. • Sintaxis clara, código elegante. 	<ul style="list-style-type: none"> • No soporta algunas características de la orientación a objetos • Es lento, al ser interpretado.

```

class PrincipalController < ApplicationController
  def index
    mensaje = "Hola Mundo!!!"
  end
end

```

Figure 3 - Hola mundo en RoR

5.2.1.7 Otras lenguajes

Además de los lenguajes que hemos visto y que, en general, son los más utilizados en el desarrollo Web, también tenemos otros lenguajes como Python, Perl, CGI, etc. A continuación vamos a ver otro par de lenguajes muy interesantes pero que no se pueden considerar lenguajes como tal.

Por un lado Ajax (*Asynchronous JavaScript And XML*), que no se puede considerar un lenguaje propiamente dicho ya que es un compendio de tecnologías y lenguajes entre los que destaca Javascript asíncrono y XML. Actualmente se está imponiendo en el desarrollo Web, ya que es una tecnología totalmente asíncrona. Mientras que Javascript tenía que descargarse totalmente en el navegador para poderse interpretar (y ejecutarse). Ajax por su parte, permite alojar la información o datos adicionales en el servidor y cargarla en segundo plano sólo cuando se necesitan, si interferir con el comportamiento de la página y sin necesidad de la actualización de toda la página. No obstante, todo no iban a ser ventajas, entre los inconvenientes, cabe destacar:

- La dificultad en el desarrollo.
- Gran uso de recursos, al depender tanto del navegador del cliente, como del servidor.
- Al actualizarse una pequeña porción de la página y no toda la página en si, el seguimiento de la trazabilidad es mucho más compleja que en las páginas dinámicas tradicionales.
- El contenido Ajax no se indexa por parte de los motores de búsqueda.

Y por otro lado, tenemos Flash que utilizan gráficos vectoriales e imágenes de alta calidad junto con sonido y video para la creación de entornos altamente interactivos. Los entornos creados en Flash, se almacenan habitualmente con la extensión SWF y pueden cargarse en una página Web y ser vistos desde un navegador. Actualmente, Flash ya no se utiliza para crear páginas Web completas, pero antes era habitual ver portales Web contruidos totalmente en Flash, lo cuál tiende a ser considerado como un error. No obstante, si que resulta atractivo a la hora de completar y complementar un portal Web dado su gran nivel de interactividad, dado que permite mostrar videos, sonidos, banners publicitarios con un nivel mayor de interactividad, etc.

La tecnología Flash siempre ha tenido dos problemas básicos, por un lado, excesiva lentitud de carga, ya que era necesario descargar todo el entorno de interacción que habitualmente es pesado, dado que se compone de imágenes, sonido y video de alta calidad. Y por otro lado, el hecho de que no se indexa por parte de los buscadores, esto hoy en día es incierto, ya que una buena página Web desarrollada únicamente en Flash si que se va a poder indexar, aunque no con la misma facilidad que cualquier otra página Web de similares características implementada en cualquier de los otros lenguajes existentes [46-50].

Actualmente, se le puede considerar una tecnología en declive para crear páginas Web completas, dadas sus dificultades a la hora de integrarse en dispositivos móviles y sobre todo al nacimiento de HTML5, que estandariza una gran cantidad de funcionalidad que antes sólo se podía implementar utilizando Flash, como por ejemplo la reproducción de videos.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
  <html>
    <head>
      <title>Hola Mundo con AJAX</title>
      <script type="text/javascript">
        function descargaArchivo() {
          if(window.XMLHttpRequest) {
            peticion_http = new XMLHttpRequest();
          }
          else if(window.ActiveXObject) {
            peticion_http = new ActiveXObject("Mi-
crosoft.XMLHTTP");
          }
          peticion_http.onreadystatechange = muestraCon-
tenido;
          peticion_http.open('GET', 'http://localhost/hola-
mundo.txt', true);
          peticion_http.send(null);
          function muestraContenido() {
            if(peticion_http.readyState == 4) {
              if(peticion_http.status == 200) {
                alert(peticion_http.respon-
seText);
              }
            }
          }
          window.onload = descargaArchivo;
        }
      </script>
    </head>
  </html>

```

Hola mundo en RoR

5.3 Frameworks

Un framework conceptualmente es una estructura previamente definida para resolver de una forma elegante y rápida problemas que a priori son complejos. Los frameworks proporcionan una serie de funcionalidades previamente implementadas que liberan al diseñador de tareas de bajo nivel y así poder dedicarle un mayor tiempo al diseño correcto de los aspectos propios del sistema concreto. En el sentido de reusabilidad se puede decir que son como librerías de software, pero los framework van un paso más adelante, ya que habitualmente son componentes estructurales de la herramienta que se desarrolla. Es decir, se utilizan los Frameworks para “no tener que reinventar la rueda”, no tener que implementar la misma funcionalidad en cada desarrollo.

Se obtiene diferentes ventajas de la utilización de un framework:

- Desarrollo rápido de aplicaciones, se libera al desarrollador de tener que programar funcionalidad a bajo nivel.
- Alta reutilización de componentes software,

- El uso y la programación de componentes software que siguen una política de diseño uniforme.

Por otro lado, también existe algunas desventajas, como por ejemplo:

- Mayor demanda de recursos computacionales, debido a la mayor funcionalidad y capas de abstracción.
- La dependencia del código fuente de la aplicación con respecto al framework.

En el mercado existe una gran cantidad de Frameworks. Si ya es complicado, decantarnos por un lenguaje de programación, una vez que lo hemos hecho será necesario decantarnos por un framework de desarrollo dentro de los existentes para el lenguaje elegido. A la hora de decirse con un Framework hay que tener en cuenta los siguientes aspectos:

- **Tipo de licencia.** Es un factor que limita en gran medida el desarrollo de un proyecto. En general hay dos tipos, software libre o propietario.
- **El nivel de complejidad.** Un framework debe ser simple, pero sin que la simplicidad quite importancia a la funcionalidad.
- **Los patrones de diseño.** Un framework debe estar basado en un conjunto de patrones de diseño que son los que debe ayudar a implementar. Dentro de estos patrones cabe destacar MVC, DAO, Abstract Layer, Facade, etc.
- **Documentación.** Es una parte fundamental de cualquier framework y facilita el desarrollo y el aprendizaje.
- **Soporte.** Hay que quién da soporte al framework elegido, el tipo de soporte y la duración media del mismo.
- **Estándares.** Incluyendo tanto estándares formales, como de facto.
- **Soporte de componentes.** Hay que tener en cuenta si el framework soporta una librería de componentes complementaria.
- **Plantillas o transformaciones.** Existen frameworks que utilizan plantillas y otros que utilizan transformaciones, en el primer caso se utilizan lenguajes de marcado, mientras que en el segundo caso se controla la interfaz a través de cambios en la salida del archivo xml producto, como por ejemplo XSLT (Extensible Stylesheet Language Transformations).

5.3.1 Frameworks existentes

5.3.1.1 Frameworks Java

A continuación vamos a presentar algunos de los Frameworks más destacados dentro del entorno Java:

- **Struts.** Es software abierto y forma parte del proyecto Jakarta de Apache Software Foundation. Este Framework implementa el patrón de diseño MVC que facilita la creación de aplicaciones Web. La pieza central de Struts es el controlador, que actúa como controlador central para la capa de presentación de la aplicación Web sobre la que se está aplicando.
- **Struts 2.** Constituye una nueva generación de frameworks que ofrecen una mejor productividad gracias que usan menos configuraciones XML, más convenios inteligentes y una arquitectura modular con un grado muy bajo de acoplamiento. Entre otras características cabe destacar:
 - Las configuraciones del framework se puede realizar como anotaciones de Java.
 - Permite la inyección de dependencias (patrones de diseño) lo que posibilita la integración con otros patrones.
 - No necesita extender de ninguna clase para implementar la funcionalidad.
- **Spring MVC.** Es un framework basado en software abierto. SMVC permite usar cualquier objeto Java como comando u objeto formulario, no hay necesidad de implementar ninguna interfaz. El acoplamiento de datos es muy flexible y los problemas de concordancia de tipos se tratan como errores de validaciones. En otras características cabe destacar:
 - Fácil integración con otros sistemas.
 - Configuración directa y potente.
 - Gran adaptabilidad
 - Código de la capa de negocio reutilizable.
 - Integrable con cualquier tecnología de la vista.
- **JSF (Java Server Faces).** Es un Framework basado en componentes, diseñado por Sun Microsystems. Es un intento de proveer un framework parecido a Swing pero como GUI Web, es decir utilizando widgets, que son componibles, orientados a eventos y pueden cambiar su apariencia (*look-and-feel*). La importancia de JSF es que se puede considerar como la tecnología oficial de presentación en la capa Web de JEE.

5.3.1.2 Frameworks PHP

PHP ha evolucionado mucho en los últimos años, ha pasado de ser un lenguaje completamente caótico donde se podía hacer cualquier cosa mezclando mucho código distinto, a ser un lenguaje con un elevado soporte para objetos.

PHP dispone de una gran cantidad de Frameworks, cada uno con sus características particulares. A continuación, se presentan algunos de los más importantes junto a sus características generales.

	PHP4	PHP5	MVC	ORM	Template	Validation	Ajax	Modules
CodeIgniter	X	X	X		X	X		
CakePHP	X	X	X	X		X	X	X
Yii		X	X	X	X	X	X	X
Zend		X	X	X		X	X	X
Symfony		X	X	X		X	X	X
PHP-DevShell		X			X		X	X
Prado		X	X	X	X	X	X	X
Akelos	X	X	X	X	X	X	X	X
ZooP	X	X	X		X	X	X	
QPHP	X	X	X		X	X	X	X

Como ejemplo, vamos a detallar un poco el Framework Zend, ya que además de ser uno de los más utilizados, también es uno de los que más soporte tiene por parte de grandes empresas como Google o Microsoft.

Zend Framework es de código abierto para desarrollar aplicaciones web y servicios web con PHP5. Es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de Zend Framework es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como *use-at-will* (uso a voluntad).

Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar conforman un potente y extensible framework de aplicaciones web al combinarse. Ofrecen un gran rendimiento y una robusta implementación MVC, una abstracción de base de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos. Otros componentes, proveen autenticación de usuarios y autorización diferentes a las tiendas de certificados comunes. También existen componentes que implementan bibliotecas de cliente para acceder de forma sencilla a los Web Services más populares.

5.3.1.3 Framework .NET

.NET es un framework creado por Microsoft que se centra en la transparencia de redes, con independencia del hardware, permitiendo un rápido desarrollo de aplicaciones. En cierta medida Microsoft lanzó este nuevo software para dar una respuesta a los frameworks tanto de la plataforma Java, como PHP. Su propuesta es ofrecer una manera rápida, segura y robusta de desarrollar aplicaciones (denominadas soluciones) permitiendo una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo.



5.4 Herramientas de Mapeo Objeto-Relacional

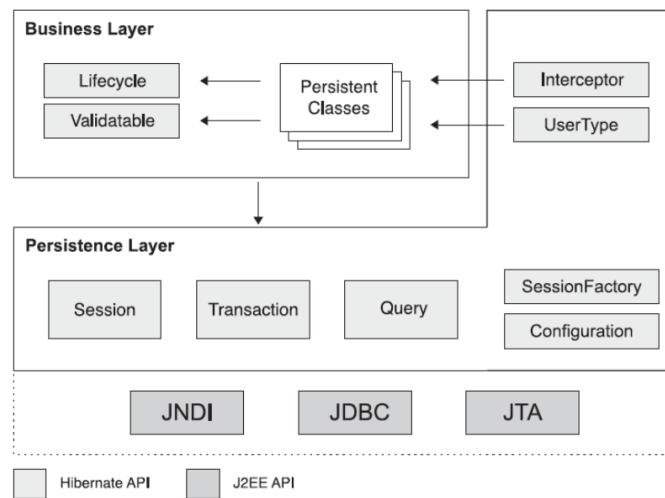
El paradigma de la programación orientada a objetos es muy diferente al de las bases de datos relacionales. Mientras que el modelo relacional trata con relaciones, conjuntos, etc., el paradigma orientado a objetos trata con objetos, su estado y sus operaciones. Por ello el desarrollador opta con frecuencia a utilizar soluciones de compromiso que le hacen perder de vista el mundo objetual y la aplicación pierde la arquitectura limpia y reutilizable que proporciona el paradigma orientado a objetos.

Para solucionar este problema, lo que se necesita es una herramienta ORM (*Object Relational Mapping*), básicamente lo que intenta hacer esta herramienta es realizar la tarea de mapeo de objetos a relaciones de una manera transparente al desarrollador.

Para este proceso sólo es necesario definir la forma en que se establecen las relaciones entre clases y tablas una única vez (es decir, indicar que propiedad corresponde, con que tabla). Después de esto tan sólo tendremos que utilizar nuestros *Value Objects* o *Pojo's* (*Plain Old Java Objects*) de nuestra aplicación y decirle expresamente a la herramienta ORM que los haga persistentes o que los devuelva a la memoria, para alterar el estado del objeto.

Hibernate es un entorno de trabajo que tiene como objeto facilitar la persistencia de objetos Java en bases de datos relacionales y al mismo tiempo la consulta de estas bases de datos para obtener objetos. Es una herramienta ORM de libre distribución, que además, es de las más maduras y completas. Actualmente, su uso está muy extendido y su desarrollo es muy activo.

En la siguiente Figura se muestran los roles de las interfaces Hibernate más importantes en las capas de persistencia y de negocio de una aplicación J2EE. La capa de negocio está situada sobre la capa de persistencia, ya que la capa de negocio actúa como cliente de la capa de persistencia.



- **La interfaz Session.** Es una de las interfaces primarias en cualquier aplicación Hibernate. Se puede interpretar una sesión como una colección de objetos cargados relacionados con una única unidad de trabajo. Hibernate puede detectar cambios en los objetos pertenecientes a una unidad de Trabajo.
- **Session Factory.** Permite obtener instancias de Session, típicamente hay una instancia por aplicación y por cada base de datos de esta aplicación.
- **La interfaz Configuración** se utiliza para configurar y “arrancar” Hibernate. Es aquí donde se debe indicar la correspondencia entre los objetos y las tablas de la base de datos.
- **La interfaz Query,** que permite realizar peticiones a la base de datos y controlar cómo se ejecuta dicha petición (*query*). Existe un lenguaje específico para Hibernate denominado HQL, que no es más que un dialecto de SQL.

Para indicar la correspondencia entre los objetos de nuestra aplicación y las tablas de la base de datos, tradicionalmente se utilizaban fichero XML, lo que producía actualizaciones muy complicadas ya que había que actualizar el objeto y la base de datos para que el sistema siguiera funcionando correctamente.

En la actualidad, existe un sistema mucho más simple, denominado Hibernate annotations, consistente en la utilización de anotaciones de Java directamente sobre las clases *Value Object*, indicando el nombre de tablas, columnas, relaciones, etc. Este método es mucho más sencillo y nos libera de la tarea de tener que mantener dos ficheros para mapear un mismo objeto, este enfoque cada vez empieza a ser más utilizado, siendo mucho más recomendable.

6 Desarrollo de una Web: Metodología de Desarrollo

Una metodología de desarrollo software se fundamenta sobre tres pilares básicos:

1. *Qué hay que hacer y en qué orden*: etapas, actividades y tareas que se deben llevar a cabo
2. *Cómo deben realizarse las tareas*: técnicas que hay que emplear para realizar estas actividades
3. *Con qué pueden llevarse a cabo*: herramientas software a utilizar en cada caso.

Al igual que en el desarrollo de cualquier aplicación software, es necesario hacer uso de metodologías en el desarrollo de una Web, para que ésta satisfaga los requisitos de usuarios y clientes. El problema que surge es que no existe una metodología universalmente aceptada para el desarrollo de aplicaciones web.

Fraternali (2000) propone que una posible solución en el desarrollo de aplicaciones web es el de aplicar metodologías tradicionales, con las propuestas de diseño para el desarrollo de aplicaciones hipermedia. Los métodos clásicos de desarrollo que se basan en modelos iterativos e incrementales (con los que se logran varias versiones del mismo producto) no son apropiadas para el desarrollo de aplicaciones web porque no contemplan características particulares de este tipo de aplicaciones. Además las metodologías tradicionales implican demasiados procesos y documentación, que no permiten un desarrollo ágil y rápido, más apropiado para la Web.

El desarrollo de un aplicación web implica un conjunto de requisitos que incrementa probabilidad de éxito en su implantación:

- **Portabilidad**: con frecuencia es necesario implantar la misma aplicación en distintas plataformas, con distintas arquitecturas, con distintas tecnologías y atendiendo a distintos dispositivos de acceso; lo que hace necesario independizar al máximo el desarrollo de la aplicación.
- **Rapidez de implantación**: el desarrollo de aplicaciones web requiere de un período de implantación más reducido.
- **Creación de contenidos como parte de la fase de ingeniería de la aplicación**: es una tarea que debe realizarse de forma paralela al diseño de la propia aplicación.
- **Integración de fuentes heterogéneas de información**: consiste en el manejo integrado de contenido estructurado y no estructurado, almacenado en distintos formatos (bases de datos, sistemas de archivos, dispositivos multimedia) y accesibles de forma distribuida mediante múltiples aplicaciones.

Por su parte los requisitos más destacables de una aplicación web son:

- **Evolución orgánica**: Es fundamental en la web, donde el contenido y los requisitos de las aplicaciones evolucionan a gran velocidad. Esto es debido a que los clientes de este tipo de aplicaciones suelen tener un conocimiento reducido de sus necesidades y de las posibilidades del sistema.

- **Seguridad en la comunicación:** Dado que las aplicaciones web están disponibles a través de una red, es difícil limitar el grupo de usuarios finales que pueden acceder a ella. Por ello son necesarios mecanismos para proteger información sensible y proporcionar modos seguros de transmisión de datos.
- **Calidad:** los usuarios se muestran poco permisivos ante los errores en aplicaciones Web (robustez, facilidad de uso o rendimiento) como enlaces erróneos o información desactualizada; provocando la pérdida de usuarios de la aplicación. Por ello es necesario disponer de mecanismos exhaustivos de control de calidad que minimicen las posibilidades de fracaso de la aplicación.
- **Velocidad:** la elección de protocolos de comunicación y el mantenimiento de una velocidad de acceso adecuada es una parte clave de diseño de dichas aplicaciones.
- **Importancia de la interfaz.** Es necesario implementar interfaces de usuario más intuitivas, capaces de capturar la atención del usuario y facilitar el acceso a la información a aquéllos que poseen una habilidad limitada en el uso de aplicaciones informáticas.
- **Necesidad de personalización.** la personalización es un elemento significativo del diseño, y da valor añadido a un contenido que debe además ser accesible y estar actualizado.
- A estos requisitos se añaden los de seguridad de la propia aplicación, escalabilidad, disponibilidad, interoperabilidad con sistemas propietarios, etc.

Los principales problemas que surgen en las aplicaciones web son la falta de fiabilidad, seguridad, escalabilidad, mantenimiento, integración y la dependencia en el desarrollo y la implantación así como la falta de estándares.

Por ello se pretende proporcionar un proceso sistemático orientado a la mejora de la calidad de la aplicación final. De esta forma, las necesidades de evolución, mantenimiento, la adaptación a nuevos dispositivos de acceso y la migración a nuevas plataformas y entornos de desarrollo deben dirigir el proceso del ciclo de vida de una aplicación web.

Para todo ello se han desarrollado metodologías que permiten estructurar, comunicar, simplificar y formalizar tanto el dominio como las decisiones de diseño, así como disponer de documentación detallada para posibles cambios del software.

Algunos ejemplos de metodologías de desarrollo web son:

1. **EORM:** es una metodología de relación entre objetos (Enhanced Object Relationship Methodology). Se define mediante un proceso iterativo centrado en el modelado orientado a objetos a través de la representación de relaciones entre estos objetos (acoplamientos) como objetos. Se caracteriza por la flexibilidad, múltiples formas de relaciones entre nodos, y la reutilización, por disponer de una librería de clases de enlaces. Existe una herramienta que automatiza el uso de esta metodología llamada *ODMTool*, que funciona junto a un generador de interfaces gráficas de usuario llamado *ONTOSTudio* y un sistema de

gestión de base de datos orientado a objetivos, SGBDOO. Estas herramientas permiten el diseño interactivo de esquemas EORM y la generación de código fuente en C++, de las clases del esquema.

2. **OOHDM:** es un método de diseño en hipertexto orientado a objetos (Object-Oriented Hypermedia Design Method) que incluye cuatro actividades: modelo conceptual, diseño navegacional, diseño abstracto de interfaz y la puesta en práctica. Los modelos orientados a objetos se construyen en sucesivos pasos que mejoran modelos diseñados en interacciones anteriores.
3. **SOHDM:** es un método que desarrolla diseño en panoramas (escenario) orientada a objetos en hipertexto (Scenario – based Object-oriented Hypermedia Design Methodology). Los escenarios permiten captar las necesidades del sistemas. Se caracteriza porque su ciclo de vida comienza con la aplicación de escenarios como técnica de definición de requisitos.
4. **WSDM:** es un método de diseño para sitios web (Web Site Design Method), que permite un acercamiento al usuario que define los objetos de información basado en sus requisitos de información para el uso de la Web. En este método se definen una aplicación Web a partir de los diferentes grupos de usuarios que vayan a utilizar el sistema. Propone cuatro etapas: modelo de usuario, diseño conceptual, diseño de la implementación e implementación. En la primera fase se identifican y clasifican los usuarios que van a hacer uso de la aplicación web.
5. **RNA:** método de navegación relacional (Relationship Navigational Analysis) que define una secuencia de pasos que se utilizarán para el desarrollo de la Web. Este método incluye: análisis del entorno, que estudia las características de los usuarios; definiciones de elemento de interés; el análisis del conocimiento y navegación y la implementación de los análisis realizados.
6. **UWE:** ingeniería Web basada en UML (Koch 2000) (UML Web Engineering= es una metodología detallada con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso es iterativo e incremental, incluye flujos de trabajo y puntos de control y sus fases coinciden con las propuesta en el Proceso Unificado de Modelado (UML – Unified Modelling Language). UWE está especializada en la especificación de aplicaciones adaptativas y por tanto hace especial hincapié en las características de personalización, como es la definición de un modelo de usuario o una etapa de definición de

características adaptativas de la navegación en función de las preferencias, conocimiento o tareas del usuario.

Ante estas metodologías más pesadas, que combinan conceptos de las metodologías clásicas de desarrollo software, ha surgido las llamadas metodologías ágiles, con un proceso de desarrollo más ligero pero suficiente para lograr aplicaciones eficientes.

6.1 Metodologías pesadas vs Metodologías ágiles

Las metodologías pesadas o tradicionales tienen como objetivo conseguir un software más eficiente y predecible, imponiendo un proceso disciplinado en el desarrollo de software. Para ello se refuerza el concepto de la planificación total del trabajo a realizar y una vez hecha esta planificación, comienza el ciclo de desarrollo del software. Esta planificación está influenciada por otras disciplinas de la ingeniería, por lo que estas metodologías también reciben el nombre de metodologías de ingeniería.

La principal crítica que se hacen a las metodologías de desarrollo es que son demasiado burocráticas, implicando tantas tareas que se ralentiza mucho el desarrollo. Además las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son adecuadas en entornos con requisitos cambiantes o no predecibles.

Ante estas metodologías tan complejas han surgido las metodologías ágiles que presentan un equilibrio entre los procesos de la metodología y el esfuerzo que implican.

Como principales diferencias entre ambas metodologías están:

- Las metodologías ágiles son adaptativas y proponen procesos que se adaptan y progresan con el cambio hasta llegar a cambiar ellos mismos. Además están orientadas al programador para que el desarrollo se convierta en algo interesante y actúen simplemente como apoyo al equipo de desarrollo en su trabajo.
- Las metodologías tradicionales o pesadas apoyan una planificación detallada del desarrollo software a largo plazo, de modo que si se produce un cambio, la planificación se verá afectada en gran medida.

6.2 Desarrollo de una web con metodologías ágiles

Entre las metodologías ágiles, existen características comunes pero también diferencias significativas. Las metodologías más comunes son:

1. **DSDM**: se considera como el método original de desarrollo ágil porque está basado en todos los principios de éste. Se denomina método de desarrollo de sistemas dinámicos (Dynamic Systems Development Method) y se basa en la continua implicación del usuario en un desarrollo iterativo y creciente que acepta requisitos cambiantes. Se desarrolló en los años 90 en el Reino Unido, combinando la experiencia con las mejoras prácticas de un consorcio de proveedores y de expertos en desarrollo de sistemas de información.

2. **XP:** La programación extrema (Extreme Programming) da mucha importancia al testeo de software. XP toma las pruebas de software como base del desarrollo y el mismo programa que escribe el código, escribe los casos de prueba. Estas pruebas forman parte de la generación de código y se integra con él. Sobre la plataforma estable definida, se genera un proceso de diseño evolutivo basado en iteraciones. Esta metodología dirige las fases de análisis, desarrollo y pruebas para lograr la calidad en el producto final.
3. **Open Source:** se basa en la distribución de trabajo entre diferentes equipos como en el resto de procesos adaptativos. En este tipo de proyectos, los supervisores de código son los únicos autorizados para realizar un cambio en el repositorio del código fuente; además de ser el responsable de coordinar y mantener la consistencia del diseño de software. El resto del personal puede hacer cambios en el código base. Se caracteriza porque se puede paralelizar la depuración y puede involucrar a un gran número de personas. Al localizar el error, se envía la supervisor de código para que éste lo corrija.
4. **SCRUM:** se basa en un proceso iterativo e incremental, que se concentra en cómo administrar tareas dentro de un ambiente de desarrollo basado en equipo. Define un conjunto de prácticas y roles que aplican en el proceso de desarrollo. Los roles principales son el *ScrumMaster*, que mantiene los procesos y se asemeja al director de proyecto, el *ProductOwner*, que representa a los clientes externos y el *Team* que incluye a los desarrolladores. En cada Sprint, de entre 15 y 30 días, el equipo crea un incremento de software entregable. Cada sprint está definido por el *ProductBacklog*, conjunto de requisitos de alto nivel que definen el trabajo a realizar.

6.3 Metodología SCRUM

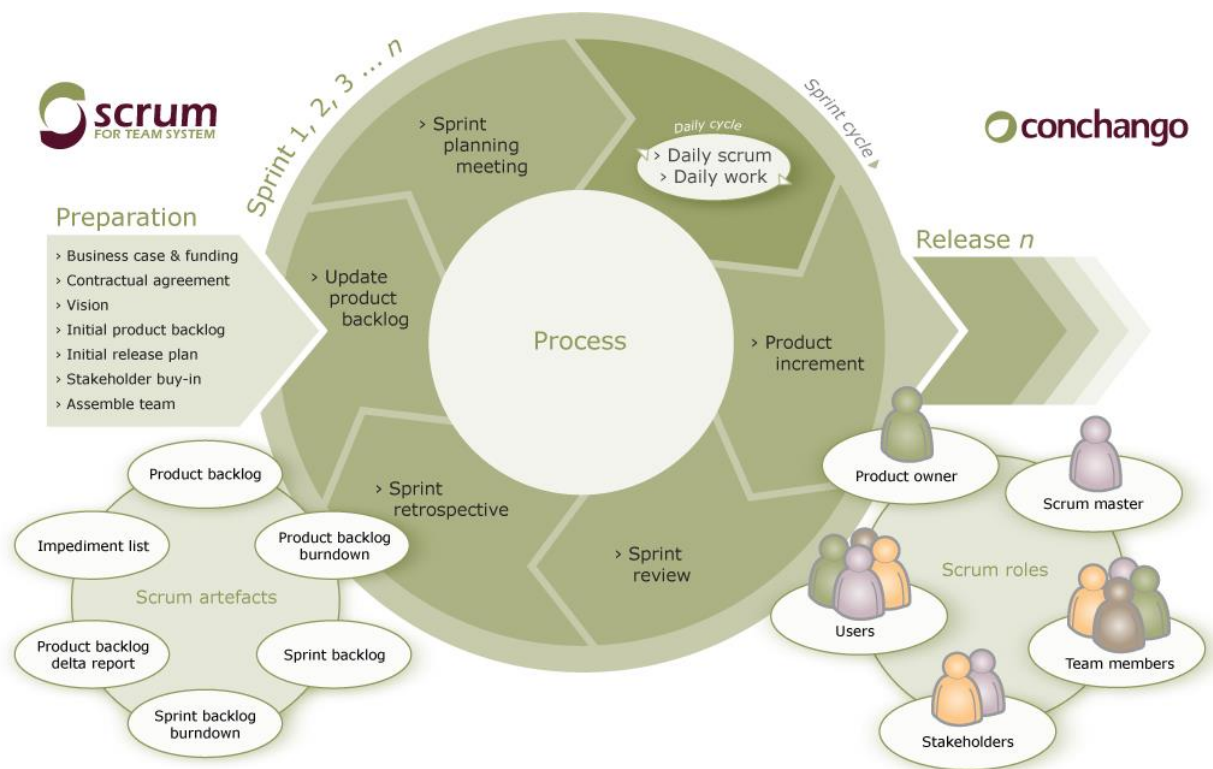
SCRUM tiene en cuenta la complejidad del desarrollo de software, en especial las aplicaciones web, y que éste se encuentra sujeto a multitud de variaciones e incertidumbres, sobre todo al comienzo. Además cada problema es diferente y no existe una solución general para todo ellos. Por ello trata de tener en cuenta la incertidumbre y los riesgos desde el modelo de desarrollo.

Otra característica importante de SCRUM es la visibilidad. Esta metodología se basa en que las personas involucradas en el proyecto conozcan en qué punto del desarrollo se encuentra el proyecto y qué falta por hacer. Esto se consigue a través de las reuniones de control, que no tratan de informar al jefe de proyecto o los clientes, sino a TODAS las personas involucradas en el proyecto. Con ello se intenta también que el equipo esté cohesionado y colaboren unos miembros con otros, se solapen y se apoyen mutuamente.

Se puede considerar que SCRUM es un marco de posibles estrategias, respetando unas reglas predefinidas. Se puede acoplar distintas técnicas de planificación, gestión, control, estimación, análisis, diseño, implementación y pruebas. Destaca también que es una metodología flexible y desde el punto de vista de desarrollo, se trata de un proceso iterativo.

SCRUM es aplicable a cualquier tipo y tamaño de proyectos, pero es muy útil en proyectos medios y grandes en cuanto a complejidad o a tamaño, o bien en aquellos proyectos donde la incertidumbre es grande, bien por el campo en el cual se desarrolla, o bien porque los requisitos no están bien definidos desde el principio y son muy susceptibles de cambiar durante el desarrollo. Las principales ventajas de SCRUM son:

- Permite entregar al cliente un producto que le satisface, que cumple mejor los requisitos que él pedía y con la calidad adecuada.
- El equipo es productivo y trabaja más a gusto, se compromete con el proyecto y la organización haciendo lo que más le gusta.
- Permite reducir la estimación temporal hasta en un 40%, debido al aumento en la productividad, fruto de los rápidos desarrollos en los *Sprints* y los compromisos adquiridos por los miembros del equipo.



6.3.1 Conceptos

La metodología utiliza una serie de conceptos como son:

- **Product Backlog:** es el documento en el cual se detallan los requisitos del sistema asignándoles prioridad.

- **Sprint:** es cada una de las iteraciones de que se compone el desarrollo. Durante el desarrollo del Spring se puede considerar necesario detenerlo y comenzar uno nuevo, por una razón justificada.
- **Sprint Backlog:** conjuntos de requisitos que serán implementados en el sprint (iteración) actual.

En cuanto a los roles existentes se distinguen:

- **Product Owner:** Propietario del producto o cliente, que definirá los requisitos del software o web a desarrollar. Es fundamental que la persona que desempeña este rol se implique al máximo en el proyecto, para que éste tenga garantías de éxito.
- **Scrum Master:** es el gestor del proyecto encargado de motivar al equipo. Crea un clima en el cual el equipo trabaje a gusto y sin interrupciones: protege a su equipo. Sin embargo, le deja trabajar libremente. Siempre estará ahí, para atender las necesidades del equipo. Es más un *líder* que un *gestor* al uso.
- **Team:** Equipo de desarrollo. Cada equipo estará compuesto por 7 +-2 personas (es decir: entre 5 y 9 personas, preferiblemente 7). El *Scrum Master* jamás intervendrá aquí. El equipo debe estar cohesionado y debe promoverse la filosofía de “todos ayudan a todos”. El equipo se autogestiona a sí mismo. Los miembros del equipo no tienen roles: colaboran entre ellos y “todos hacen de todo”: análisis, diseño, implementación, pruebas y documentación. De este modo no se rompe la continua colaboración y desarrollo de las funcionalidades del proyecto.



En esta metodología existen dos roles con nombres graciosos y que dan forma a SCRUM: cerdos (pigs) y gallinas (chicken). La denominación viene de un chiste: *Un cerdo y una gallina se encuentran en la calle. La gallina mira al cerdo y dice: “Hey, ¿por qué no abrimos un restaurante?” El cerdo mira a la gallina y le dice: “Buena idea, ¿cómo se llamaría el restaurante?” La gallina piensa un poco y contesta: “¿Por qué no lo llamamos “Huevos con jamón?” “Lo siento pero no”, dice el cerdo, “Yo estaría comprometido pero tú solamente estarías involucrada”.*

En SCRUM, los *cerdos* (aquellos que están comprometidos de forma seria con el proyecto) son desempeñados por los roles *Team*, *Scrum Master* y *Product Owner*. El rol *gallina* está constituido por otros clientes y *stakeholders* (ejecutivos, etc.).

6.3.2 Fases de la metodología

Se distingue tres fases fundamentales en SCRUM:

- *fase de planificación*, en la cual se realizan las labores básicas de una planificación breve: visión general del proyecto (estimación muy general, viabilidad del sistema) y construcción del *Backlog* por un lado y por otro el desarrollo de la arquitectura al detalle
- *fase de desarrollo*, en la cual tienen lugar los *Sprints*
- *fase de entrega y balance* de los éxitos y fracasos logrados.

La fase fundamental de la metodología, es la fase de *Sprint* puesto que las otras dos no difieren mucho de otras metodologías. El desarrollo en la fase *Sprint* es iterativo, en uno o más *Sprints*, hasta que el proyecto se da por finalizado por el *Product Owner*. De este modo se hará frente al problema de los requisitos variables; de modo que aunque los requisitos probablemente cambiarán, la metodología está preparada para ello.

Existen cuatro tipos de reuniones durante el desarrollo de un proyecto con SCRUM:

- **Encuentro de Planificación (4 horas)**: al comienzo de un *Sprint* se decide qué parte del *Product Backlog* se implementará en este *Sprint*. Una vez decididas las funcionalidades a implementar, basándose en estimaciones de tamaño, tiempo, esfuerzo, etc., el *Sprint Backlog* no se toca durante todo el *Sprint*, bajo ninguna circunstancia. Si algo falla, el *Scrum Master* podrá cancelar el *Sprint* y comenzar otro.
- **Encuentro Diario (15 minutos)**: diariamente el equipo se reúne unos 15 minutos, para responder, individualmente, a 3 preguntas básicas: *¿qué hiciste ayer? ¿qué vas a hacer hoy? ¿qué te está impidiendo alcanzar tus objetivos?* De este modo se realiza el control del proyecto y el seguimiento de los posibles riesgos.
- **Encuentro de Revisión (4 horas)**: al final del *Sprint*, se realizará una reunión con el *Product Owner* y otros clientes (gallinas) para exponer la funcionalidad desarrollada junto con las posibles preguntas y ampliaciones del *Backlog* que se les pueda ocurrir a los diferentes *stakeholders*. Con esto se logra un *feedback* continuo con el cliente, que está al tanto del progreso del proyecto

- **Encuentro Retrospectivo (4 horas):** reunión del *Scrum Master* con el Team para revisar cómo fue el *Sprint*: qué se consiguió realizar bien y cómo se podría mejorar. Con ello se pretende obtener información histórica del propio proyecto, útil para futuras estimaciones y *Sprints*.

6.3.3 Desarrollo de la fase Sprint

Cada Sprint tendrá una duración aproximada de **30 días**; que es un tiempo adecuado para desarrollar un producto entregable y funcional al *Product Owner*. Se considera una jornada laboral de 8 horas. No se contempla, bajo ningún concepto, la realización de horas extra al considerarlo poco productivo [51-57].

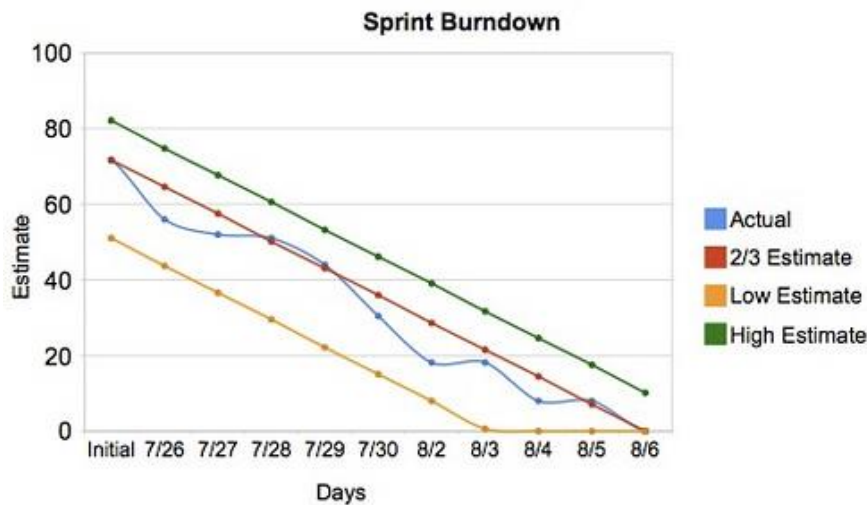
El Sprint comienza con un Encuentro de Planificación en el que se decide qué se va a desarrollar del *Product Backlog* y se elabora el *Sprint Backlog*. A la hora de seleccionar las funcionalidades a desarrollar se tendrá en cuenta la prioridad de los requisitos, establecida por el *Product Owner*. Se descomponen los requisitos en tareas simples, realizables en poco más de una jornada laboral de 8 horas como máximo. Una vez decidido el *Sprint Backlog* éste no se toca bajo ninguna circunstancia durante el *Sprint*. Las funcionalidades seleccionadas NO se asignan a los miembros del *Team*, sino que cada miembro **elige** qué desea desarrollar.

Una vez comenzado el *Sprint*, el *Scrum Master* no se implicará en el *Team*; realizará sus labores de planificación, estimación, control y gestión y solamente contactará con el *Team* en las reuniones diarias. Aparte de esto solamente tratará con algún miembro del *Team* cuando éste desee preguntarle algo o solicitar algún tipo de ayuda.

Hay libertad de desarrollo para el equipo, lo que no significa ausencia de control o anarquía. El *Scrum Master* tratará de favorecer el flujo o máxima productividad de su *Team* y tratará de crear un clima en el cual se pueda trabajar sin interrupciones.

Cada miembro se responsabilizará de entregar la parte que se ha comprometido a realizar en el plazo fijado. Si es incapaz de ello hablará con el *Scrum Master* para solicitar ayuda o detener el *Sprint*, de forma muy excepcional. Si el *Sprint Backlog* se termina antes de tiempo, se puede dialogar con el *Product Owner* para añadir más funcionalidad al Sprint.

El avance *Sprint a Sprint* se visualiza mediante un diagrama de *burn down* que representa la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint. Dibujando una línea que conecte los puntos de todos los *Sprints* completados, podremos ver el progreso del proyecto. Lo ideal es que esta línea sea descendente (en casos en que todo va bien en el sentido de que los requisitos están bien definidos desde el principio y no varían nunca) hasta llegar al eje horizontal, cuando el proyecto se ha terminado (no hay más requisitos pendientes de ser completados en el *Backlog*). Si durante el proceso se añaden nuevos requisitos la recta tendrá pendiente ascendente en determinados segmentos, y si se modifican algunos requisitos la pendiente variará o incluso valdrá cero en algunos tramos.



Cuando en un *Sprint* hay más de un SCRUM, entra en juego un añadido a la metodología SCRUM, que es el concepto de SCRUM of SCRUMS, que consiste en realizar un SCRUM diario con un representante o portavoz de cada equipo, el cual explicará al resto los avances diarios que está realizando su grupo (responderá a las 3 preguntas antes mencionadas).

7 De la Web 2.0 a la Web 3.0

7.1 Introducción

En los últimos años se ha producido una revolución en la Web: los portales y las páginas en las que el administrador del portal publica contenido para que los usuarios los consuman ha dejado paso a espacios colaborativos en los cuales los usuarios son *prosumidores*, productores y consumidores de contenidos. Ejemplos como *Wikipedia*, *YouTube* o *Flickr* representan de la mejor manera la nueva generación de la Web denominada Web 2.0.

La evolución de la Web ha llevado a un nuevo auge la Web Semántica, nacida en 1999 gracias a la investigación de Tim Berners-Lee que escribió: “*Tengo un sueño para la Web, en el cual los ordenadores llegan a ser capaces de analizar los datos de la Web, el contenido, los enlaces, las transacciones entre personas y ordenadores*”. Se refiere a la Web Semántica como evolución de la Web clásica en la cual las aplicaciones orientadas a la recuperación de la información carecen de la capacidad de análisis de relaciones complejas entre los datos impidiendo su reutilización por personas u otras aplicaciones. En la Web actual, los individuos utilizan conocimiento implícito a la hora de acceder a la información, mientras los ordenadores necesitan conocimiento explícito para acceder y razonar sobre los datos.

Interés de Búsqueda en la Web: "web 2.0"

Todo el mundo, 2004 - hoy

Categorías: [Noticias y eventos actuales \(25-50%\)](#), [Internet \(10-25%\)](#), [más...](#)

Evolución del término Web 2.0

A finales del mismo año, *Tim O'Reilly* (O'Reilly, 2005) en su célebre artículo "What is Web 2.0?" formalizó la definición de esta nueva tendencia. Según *O'Reilly* la Web 2.0 es un paradigma que rompe con la anterior generación de la Web manteniendo pero unos límites difíciles de definir y distinguir "aplicaciones Web 2.0" de aplicaciones que no lo son no es una tarea trivial. La "Web 2.0" es un "núcleo gravitacional", usando las palabras de *O'Reilly*, de principios y prácticas alrededor del cual giran las aplicaciones a una distancia menor o mayor de centro.

Los dos conceptos que forman el núcleo de la Web 2.0 se pueden resumir en:

- *Posicionamiento Estratégico*: la Web es la base para nuevos desarrollos, no solo tecnológicos, sino también sociales, comunicacionales y organizacionales.
- *Posicionamiento del Usuario*: se considera al usuario (cómo individuo y cómo parte de una comunidad) como el motor principal de esta nueva filosofía. Es gracias el usuario y a la forma en la cual éste utiliza esta plataforma.

Es el aspecto "social" de la Web es el nuevo protagonista. A las interacciones on-line se añade un significado social y humano. Las relaciones sociales se extienden a Internet para ser expandidas y amplificadas por el enorme número de participantes y por la naturaleza viral del flujo de información en Internet. Los usuarios adquieren creciente importancia tanto que Time Magazine les dedica la portada de "Personaje del año 2006".

Otras definiciones en la misma línea son la de *Batelle* (2004) y la de *Xavier Ribes* (2007) para los cuales la Web 2.0 consiste en todas aquellas herramientas y servicios basados en una fuente de datos que pueden ser modificados por los usuarios tanto a nivel de contenidos cómo a nivel de presentación.

Tres prácticas son rompedoras respecto al enfoque de las primeras aplicaciones en Internet:

- *Web 2.0 cómo actitud y no cómo Tecnología*: los efectos de la Web 2.0 han provocado profundos impactos en los aspectos sociales, tecnológicos y de negocio.
- *La Larga Cola*: si antes el enfoque de mercado se basaba en el alto rendimiento de pocos productos ahora es necesario considerar la suma o acumulación de todas las pequeñas

ventas de muchos productos, que puede igualar o superar al primero. Son el antiguo mercado de masas y el nuevo nicho de mercados, representados por la cabeza y la cola de una distribución estadística.

- *El Beta Perpetuo*: el producto se desarrolla continuamente, incorporando nuevas funcionalidades mensuales, semanales e incluso a diario. Asimismo, el software es distribuido como un servicio y no como un producto.

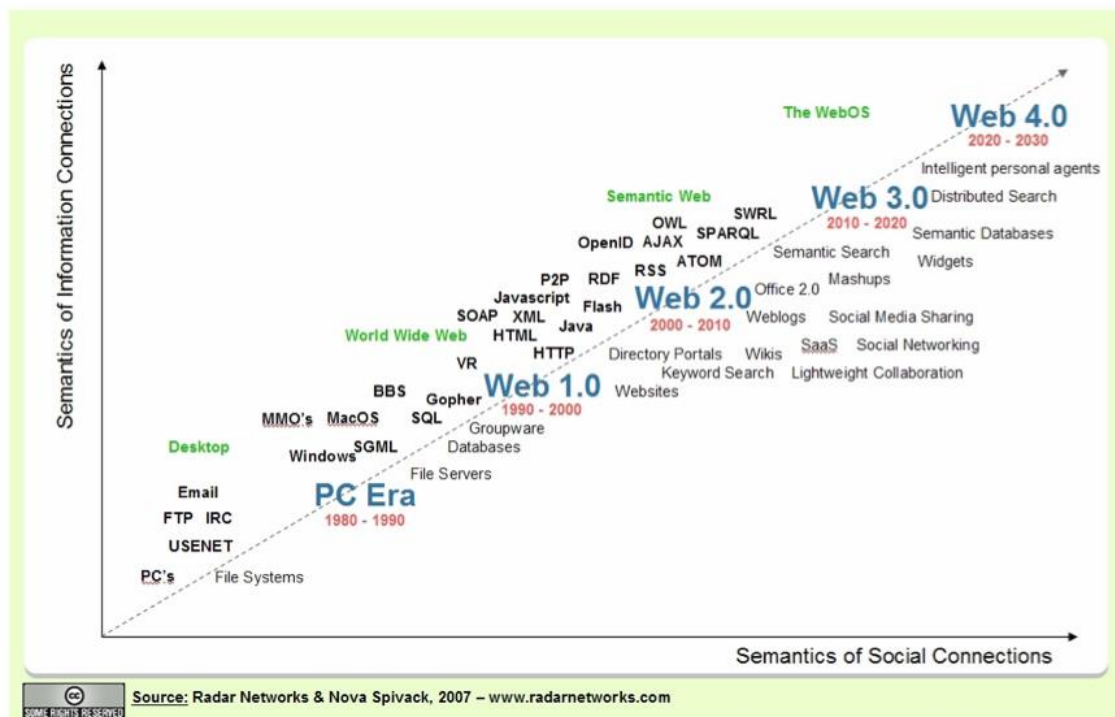
7.1.2 Desde la Web 2.0 a la Web 3.0... y más allá

En la Web 2.0 el usuario es el nuevo protagonista de las aplicaciones on-line y se aprovecha la inteligencia colectiva para la creación y gestión de los contenidos. Las diferencias entre Web 1.0 y Web 2.0 se pueden resumir en la siguiente tabla:

Web 1.0	Web 2.0
Lectura	Escritura
Compañías	Comunidades
Homepages	Blogs
Portales	RSS
Taxonomía	Tags y Folksonomies
Cableado	Inalámbrico
Propiedad	Compartir
Netscape	Google
Propiedad	Compartir
Conexión Telefónica	Banda Ancha
Costes Hardware	Costes en Ancho de Banda

Se podría opinar que permitir al usuario modificar la información presente en bases de datos no es una novedad ya que, por ejemplo, las aplicaciones que permiten operaciones bancarias a través de Internet por definición dejan que el usuario cambie no sólo los datos (por ejemplo haciendo una transferencia) si no también la presentación de los mismos (el saldo varía al hacer operaciones). Las Wikis ya existían en los años 90: la primera wiki, “*Wiki Wiki Web*” creado por *Ward Cunningham*, fue publicado el 25 de marzo de 1995 y todavía se encuentra on-line (<http://c2.com/cgi/wiki>) (*Ribes 2007*). Según *Ribes* la novedad reside en los metadatos: *en la explotación de los mismos y en la facilidad que la tecnología otorga para su gestión de forma colaborativa*. Además la facilidad de publicación de información ha hecho que la cantidad de datos crezca rápidamente y una eficiente gestión de un volumen tan grande pasa por la buena gestión de los metadatos: *“la manera fundamental de abordar esta hiperinflación es la indexación y el tratamiento automatizado de los datos de esta indexación”* (*Quijada 2006*). Gracias a los metadatos es posible describir, recuperar, organizar la información a la que se refieren. Los mismos usuarios crean metadatos y clasifican el contenido que crean (cómo hemos visto definiendo las folksonomías). Los usuarios se convierten entonces en “indexadores”, relacionando los datos

con los tags que definen: millones de personas diariamente trabajan de forma continua, voluntaria y gratuita en la clasificación de contenidos. “Clasificar” implica “añadir significado”, es decir, “semántica”. La Web 2.0 ha llevado así a una nueva auge el concepto de Web Semántica planteado por *Tim Berners-Lee* (1998). Si por un lado es más fácil crear contenidos, por el otro es necesario dar significado a los mismos para su gestión y los mismos usuarios pueden contribuir a la definición de este significado. Según *Nova Spivak* (2007) la evolución de la Web va desde la era “prehistórica” de un escritorio local hasta un futuro no muy lejano de un sistema operativo Web. *Spivak* cree que en 2007 ya se está en la cola de la Web 2.0 empezando a crear los fundamentos de la próxima generación la Web 3.0: *una Web en la cual la semántica tiene mucho que decir*.



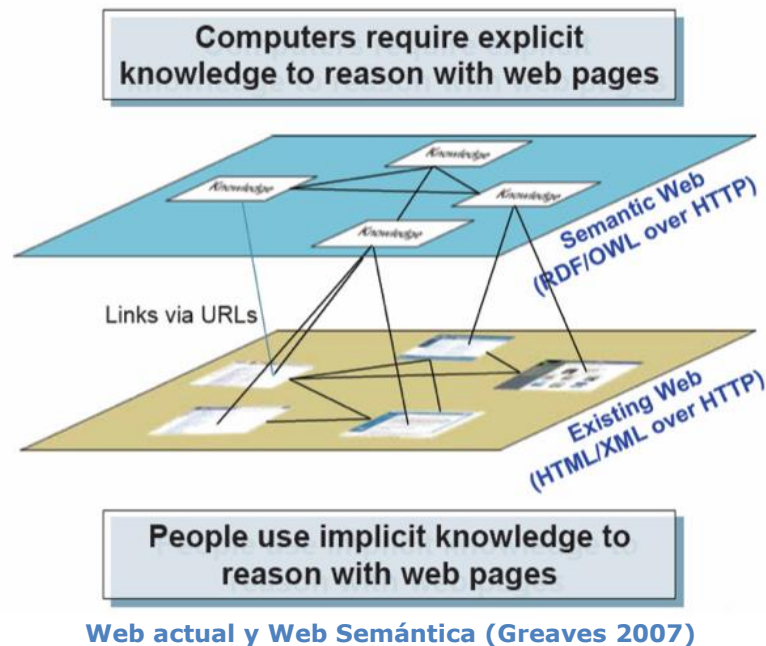
Evolución de la Web (Radar Networks)

7.1.3 Pasos hacia la Web Semántica



Desde la última década del siglo pasado, la Web es una herramienta al alcance de millones de personas en todo el mundo. El desarrollo del primer navegador Web (*Mosaic*) en 1993 (*Schatz and Hardin 1994*), y la evolución de las interfaces gráficas de usuario en los sistemas de búsqueda y recuperación de información han permitido el desarrollo de una arquitectura basada en protocolos, lenguajes de descripción de forma y contenido que constituyen la Web actual.

La escasa estructuración de la información presente en bases de datos y sitios Web y la necesidad de un esquema de información normalizado, organizado e interoperable para una efectiva recuperación de los contenidos ha abierto una nueva rama de investigación: especialistas de distintas áreas del conocimiento (lingüística, inteligencia artificial, psicología y documentación, entre otras) han integrado sus experiencias con el fin de crear una Web más organizada a favor del mejoramiento de los sistemas de gestión de contenidos. El resultado es la que *Tim Berners-Lee* (1998, 2001) ha bautizado como Web Semántica. Según *Berners-Lee* la Web es un medio para el intercambio de datos, información y conocimiento. **El objetivo de la Web Semántica es facilitar este intercambio, entre personas y entre aplicaciones** (*Herman 2008*).



Tim Berners-Lee (1999) pone el acento sobre la interoperabilidad de la Web del futuro: *Tengo un sueño para la Web, en el cual los ordenadores llegan a ser capaces de analizar los datos de la Web, el contenido, los enlaces, las transacciones entre personas y ordenadores.* *Berners-Lee* sueña la Web Semántica como evolución de la Web clásica en la cual las aplicaciones orientadas a la recuperación de la información carecen de la capacidad de análisis de relaciones complejas entre los datos, impidiendo su reutilización por personas u otras aplicaciones. En la Web actual los individuos utilizan conocimiento implícito a la hora de acceder a la información, mientras los ordenadores necesitan conocimiento explícito para acceder y razonar sobre los datos. Los obstáculos que dificultan el camino hacia la Web Semántica consisten la falta de mecanismos automáticos para el procesamiento de la información y de mecanismos de interoperabilidad sintáctica, semántica y estructural de los sistemas de información basados en la Web. Por interoperabilidad sintáctica entendemos la capacidad de los sistemas de información para leer datos procedentes de diferentes fuentes y manejar una representación compatible. La interoperabilidad semántica es la capacidad de los sistemas de información para intercambiar información basándose en un significado común de los términos y expresiones que se usan.

7.1.4 Web 2.0 y Web Semántica: hacia la Web 3.0

Si la Web 2.0 representa la nueva generación de la Web desde el punto de vista del usuario, la Web Semántica es la evolución hacia una Web orientada a los ordenadores. No obstante la semántica es ya parte integrante de la Web 2.0: *el creciente uso de folksonomías y clasificación colaborativa por medio de etiquetas simples en un espacio de nombres sin jerarquías, son el primer acercamiento a la Web Semántica.* Las folksonomías son sistemas de etiquetas basados en un entendimiento común entre los usuarios, así como las ontologías son una especificación formal y explícita de una conceptualización compartida. De la misma manera, los *mash-ups* necesitan una referencia común para el intercambio de información como geolocalizaciones, tiempo o nombres. Además aplicaciones como la *Wikipedia* se basan en una organización jerárquica y semántica de su contenido.

La Web Semántica puede contribuir al desarrollo de la Web 2.0 proporcionando modelos formales entendibles por las máquinas de simple fragmentos de texto: “geográficamente localizado” y “lugar geográfico contiene” son propiedades inversas, “genero” puede ser “masculino” o “femenino”. La infraestructura de tipos y jerarquías de las ontologías permite representar información como “discoteca” es un “local nocturno”. Es además posible definir simples reglas de inferencia y clasificación basadas en semántica (Greaves 2007). Esto significaría un mejor acceso a los contenidos a través de expansión semántica de las consultas y filtrado semántico de los resultados y se beneficiaría la interoperabilidad de los *mash-ups*. Sin tecnología semántica, la interoperabilidad entre aplicaciones 2.0 es muy limitada y provocando aislamiento (*walled gardens*)

Por otro lado hemos visto como herramientas típicas de la Web 2.0, las wikis, representan posibilidades efectivas para la definición colaborativa de ontologías.

La integración de Web 2.0 y tecnología semántica permite:

- *Mejorar la gestión de los datos*: cuando los contenidos provienen de diferentes fuentes, son necesarias aplicaciones para gestionarlos. La relación entre los datos además no siempre es explícita y las fuentes pueden ser heterogéneas. La Web Semántica propone resolver este problema mediante la descripción formal de la relación entre los datos, proporcionando mecanismos de automatización para la integración de los mismos.
- *Fomentar colaboración*: la posibilidad de integrar fuentes de información heterogéneas facilita y fomenta la interacción en un entorno colaborativo (las aplicaciones 2.0) haciendo este proceso más eficiente incrementando la accesibilidad a los datos y sus creadores.

En definitiva, la Web 2.0 permite la creación fácil de contenidos, que cuando alcanzan una masa crítica se vuelven rápidamente inaccesibles. Las tecnologías semánticas se presentan como herramientas válidas para estructurar y manejar este contenido proporcionando un acceso eficaz a la información y garantizando escalabilidad.

La convergencia entre la Web 2.0 y la Web Semántica ha tomado el nombre de "Web 3.0". Este término "Web 3.0" se da a conocer masivamente el 2006 en un artículo de *John Markoff* en el New York Times. La primera conferencia sobre Web 3.0 se ha celebrado en Octubre 2008 en Santa Clara, California (<http://www.web3event.com/>).

Interés de Búsqueda en la Web: "web 3.0"

Todo el mundo, 2004 - hoy

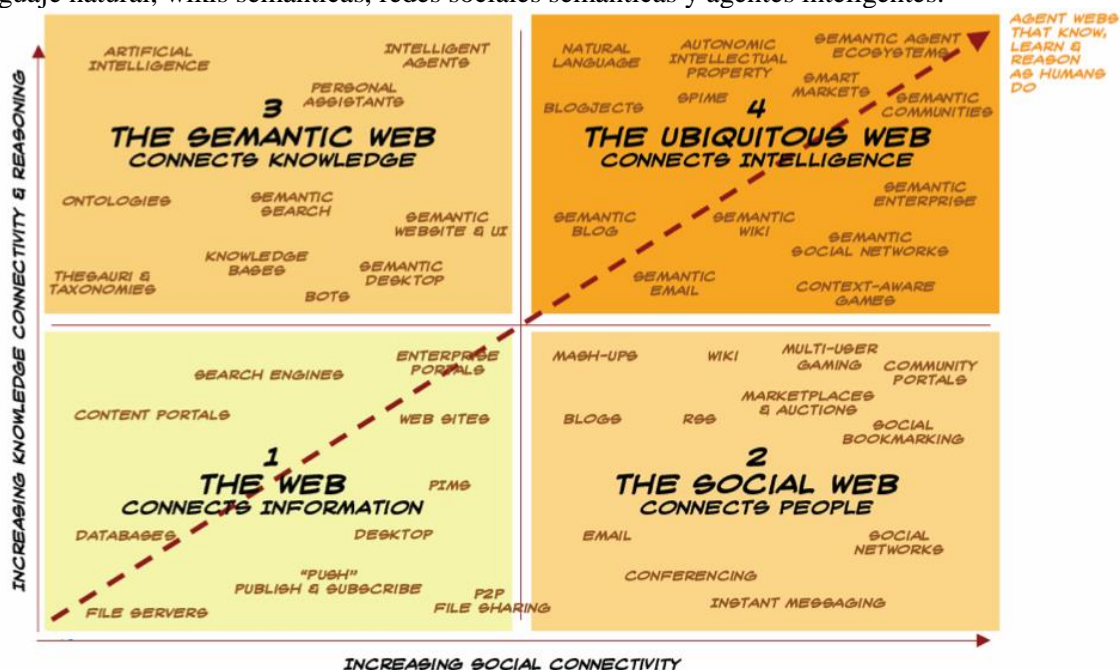
Categorías: [Noticias y eventos actuales \(10-25%\)](#), [Informática y electrónica \(10-25%\)](#), [más...](#)

Totales
"web 3.0" 50



Evolución de la popularidad del término "Web 3.0"

En la figura inferior se representa la evolución de la Web hasta el 2020. La Web clásica es la que detiene el menor nivel de conectividad social y conectividad de conocimiento. La Web 2.0 pone el acento en la interacción social, mientras la Web Semántica se orienta al conocimiento gracias a técnicas de inteligencia artificial, bases de conocimiento, taxonomías y ontologías. La Web del futuro reúne estos dos aspectos en la Web Inteligente formada por aplicaciones basadas en lenguaje natural, wikis semánticas, redes sociales semánticas y agentes inteligentes.



Evolución de la web hasta el 2020

7.2 La evolución de las nuevos portales. Internet como un medio audiovisual

7.2.1 Web 2.0 e inteligencia colectiva

Ruilifson y Engelbart en 1999 afirman que problemas complejos se pueden resolver de manera más eficaz por un grupo de personas que trabaja en específicas condiciones que por un único individuo de la comunidad, acuñando el concepto de “Inteligencia colectiva”. Según sus teorías las condiciones necesarias para que un grupo sea “sabio” se pueden resumir en cuatro puntos:

- Diversidad de opinión entre los miembros del grupo, causada por diversidad de información inicial
- Independencia a la hora de tomar la decisión o formular la opinión
- Descentralización
- Mecanismos de combinación de las diferentes opiniones de los participantes.

7.2.2 Motores de búsqueda

Desde los principios de la Web y cada vez con más fuerza las aplicaciones de éxito se han construido sobre las potencialidades de la inteligencia colectiva: es suficiente pensar que el fundamento de la misma Web es el *hyperlinking* (la técnica que permite enlazar documentos entre ellos). Nuevo contenido creado para los usuarios se hace accesible a través de los enlaces presentes en otras páginas o documentos que a su vez han sido añadidos por otros usuarios que lo han descubierto y encontrado interesante. *Yahoo!* (<http://www.yahoo.com>) nació como listado de páginas Web organizadas por categorías (Labrou Y., Finin T. 1999) y en su historia y evolución ha mantenido su enfoque hacia el trabajo colaborativo como signo de distinción y valor añadido. De la inteligencia colectiva se beneficia también *Google* (<http://www.google.com>) que basa los resultados de las búsquedas en un algoritmo (PageRank) que tiene en cuenta los enlaces entre páginas que los mismos usuarios han creado. Resumiendo brevemente el funcionamiento del algoritmo, una vez identificadas páginas Web relacionadas con la búsqueda del usuario se considera el número de enlaces que cada página recibe para calcular su relevancia y ordenar así los resultados de la búsqueda (Rogers 2002).

7.2.3 Reputación y feedback

Otras empresas también se benefician del trabajo colaborativo de sus usuarios: el valor añadido de *eBay* (www.ebay.com), sitio Web de subastas on-line, es la actividad colectiva de sus usuarios (Resnick & Zeckhauser, 2002). *eBay* es el medio gracias al cual los usuarios interactúan, compran, venden y crean contenido. La ventaja competitiva de eBay está en la masa crítica de usuarios que cada día participan a las subastas como vendedores o compradores.

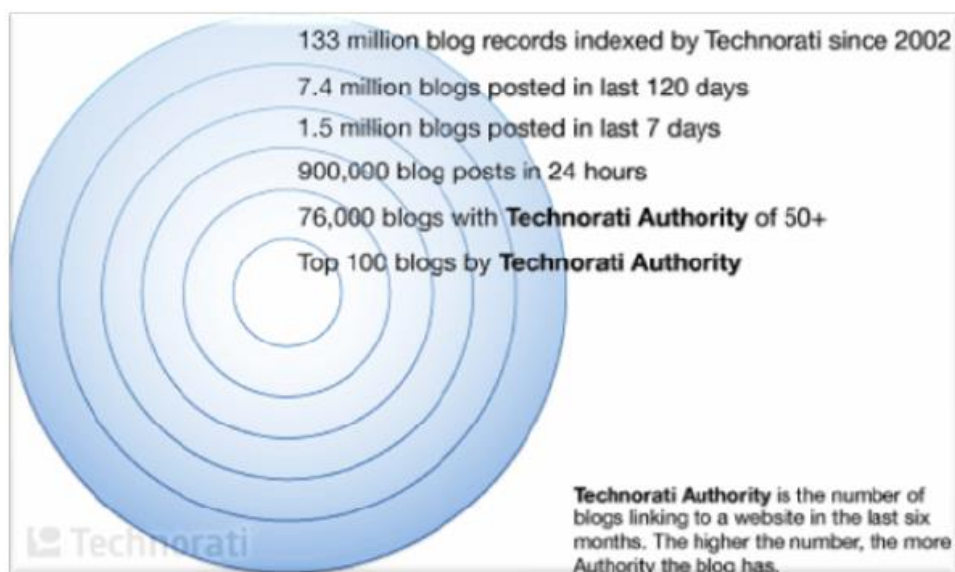
El caso de *Amazon* (www.amazon.com) es parecido: aunque el modelo de venta on-line sea fácilmente replicable, Amazon posee una ventaja competitiva en el compromiso que sus usuarios ponen a la hora de escribir reseñas de los productos a la venta o de alimentar el sistema de reputación relativo a los vendedores. Otro valor añadido que proviene directamente de la colaboración entre usuarios es la capacidad de ofrecer a los compradores productos “a la moda” gracias a índices de popularidad calculados en tiempo real a partir de las compras o de las valoraciones de los mismos usuarios.

7.2.4 Creación de contenido

Flickr (www.flickr.com) o *You Tube* (www.youtube.com) permiten interactividad en la creación y modificación de contenidos (fotos y video respectivamente).

Además de crear contenido los usuarios pueden clasificarlo, asociando tags y términos a lo que están compartiendo. De esta manera se crean folksonomías, vocabularios de términos creados por los mismos usuarios, en oposición a la taxonomía tradicional donde las categorías de clasificación están previamente establecidas. En las folksonomías se basa también *del.icio.us* (www.delicious.com), una aplicación de *social bookmarkings* gracias a la cual es posible compartir y etiquetar enlaces a páginas

interesantes para construir un directorio de manera social y colaborativa.



Otro fenómeno que caracteriza la nueva era de la Web es el *blogging*. Un blog (o Weblog) es un espacio muy parecido a una página Web personal del estilo 1.0. La mayor diferencia entre un blog y una página 1.0 es el formato de diario que el primero tiene. Un blog es un diario con una característica peculiar: la posibilidad de suscribirse para recibir actualizaciones por medio de un tipo de documento llamado RSS. En Marzo 2007, *Technorati* (www.technorati.com) había indexado 70 millones de blog.

En 2008 los blogs ya eran 133 millones. Si ampliamos el concepto de blog desde espacio para la fácil creación de contenido personal a espacio para fácil creación de contenido de una comunidad obtenemos una Wiki. El ejemplo de más éxito de una Wiki es la Wikipedia (www.wikipedia.org) que en pocos años ha conseguido superar el número de entradas de la Enciclopedia Británica y cuyo nombre aparece en más de 30 millones de búsquedas diarias en Google (fuente Google Trends - Nov. 2008).

7.2.5 Usuarios cómo valor añadido

Según *O'Reilly* (2005) los *network effects* (efectos que se produce en un mercado cuando se agregan nuevos participantes que atraen vendedores y compradores) debidos a las contribución de los usuarios son la clave para dominar el mercado en la era de la Web 2.0. Es la colaboración que añade valor a los servicios de búsqueda de *Google*, a *del.icio.us* o *Flickr*. Es la comunidad la que decide el contenido y la importancia que tiene a través de un sistema basado en el prestigio, la reputación, la confianza que la misma comunidad otorga a contenidos o a otros usuarios. La fuerza de Wikipedia es la ausencia de privilegios o categorías de control: cualquier usuario puede introducir nuevos contenidos y ampliar o corregir artículos escritos por otros. Las informaciones erróneas serán antes o después eliminadas, según la ley de *Linus* (*Linus Torvalds*): “dados suficientes ojos, todos los errores serán evidentes” (*Raymond 2000*). *Morville* afirma que la autoridad deriva de la arquitectura de la información y en el poder de la inteligencia colectiva. (*Morville 2005*). El aspecto arquitectural es fundamental también en la opinión de *O'Reilly*. La participación voluntaria a un proyecto colectivo reside en el enfoque y en la organización que el proyecto tiene. Es la clave del éxito de los sistemas P2P en los cuales cuantos más datos los usuarios comparten, más datos podrán obtener. Desde el punto de vista de la teoría de los juegos un sistema P2P

cumple las características de modelo estudiado en teoría de los juegos que premia los participantes que comparten más recursos con el fin de minimizar los posibles abusos por uno de los jugadores. Como consecuencia, sólo una estrategia colaborativa resulta ganadora: el sistema tiende a un punto de equilibrio de Nash. El modelo de creación de datos basado en una arquitectura que fomenta la colaboración en estilo P2P se opone al primer modelo introducido por *Yahoo!* (se paga gente para que recopile información) y al segundo de *Open Directory Project* basado en el trabajo voluntario (*Bricklin 2001*).

7.2.6 Internet como medio de comunicación

Las ventajas de Internet frente a otros medios radican en su alta capacidad de afinidad (sobre todo para *targets* comerciales), el poder de microsegmentación, la interactividad y la medición post-campaña, así como en la posibilidad de calcular el retorno de la inversión (ROI) de una forma inmediata. Por su parte, la cobertura todavía no alcanza la de otros medios, si bien en los últimos años Internet ha tenido un crecimiento espectacularmente rápido. Según el “Estudio de Calidad de los Medios en España 2007”, los tres aspectos más importantes en la planificación en Internet, frente a otros medios, son la afinidad con el *target*, el coste y la audiencia/cobertura.

Según un estudio de ámbito europeo, en España superamos la media europea en cuanto a acceso a redes sociales (un 47% frente al dato europeo de 41%), ver videos y televisión por Internet (32% frente al porcentaje europeo de 27%) y el uso de la mensajería instantánea (41% frente al 36% de usuarios europeos), entre otros.

Las cifras de penetración del medio en la población y de inversión publicitaria online corroboran la validez de Internet como medio de comunicación y soporte publicitario, respectivamente. Según datos de Red.es, a final de 2009 el 60,5% de la población son internautas (24,3 millones de individuos), y los españoles con edades comprendidas entre los 16 y los 24 años o los 25 y los 34 años superan la media de acceso europeo con un 90,3% y un 78,3%, respectivamente.

La inversión publicitaria online ha ido creciendo progresivamente, en consonancia con el mayor papel que las empresas han ido concediendo a Internet dentro del plan de marketing. Según datos de *InfoAdex*, en 2008 la inversión publicitaria descendió en todos los medios con respecto a los datos de 2007, excepto en Internet, donde presentó un ascenso del 26,5%.

Existen un conjunto de características que hacen de Internet un instrumento especialmente idóneo en la tendencia actual del marketing: la precisión y su máxima expresión: el *Marketing one to one*, de forma masiva, incrementando el valor ofrecido, sin incrementar costes.

- **Bajo coste relativo.** Por un lado, puede evitar con mayor facilidad impactos inútiles (los realizados sobre público no objetivo) y por otro, el precio del impacto útil suele ser menor que, incluso, el correo convencional, mejorando sensiblemente sus índices de respuesta (del 2% característico del correo convencional, se eleva al 12 o 18%, según estudios).
- **Internacionalización.** Se trata de un medio que permite superar fronteras geográficas, política y radioeléctricas, con costes locales y obstáculos lingüísticos e idiomáticos, muy fáciles de superar, lo que facilita el acceso a una gran audiencia potencial.
- **Amplitud horaria.** Disponible todos los días del año, las 24 horas del día (lo asemejaría a medios permanentes (ej: bayas; paneles; etc.), pero que requieren una mayor inversión y el desplazamiento físico del usuario al punto de exhibición).

- **Automatización.** Posibilidad de automatizar gran parte de los servicios prestados a través de la red.
- **Interactividad.** Permite una estrecha relación usuario-empresa, ya que da al usuario la oportunidad de poder interactuar, dándole acceso directo a mensajes, propuestas, productos, servicios y puntos de venta:
 - Se interactúa con el medio. La primera relación entre el emisor y el receptor no es entre ellos sino con el medio con el que interactúa.
 - Se interactúan a través del medio, por lo que intervienen aspectos propios del entorno tecnológico: velocidad de acceso, características del hardware, etc.

La bidireccionalidad en tiempo real, en que se basa la interactividad, posibilita la retroalimentación inmediata; la influencia recíproca o feedback (respuesta que proporciona el sujeto, ordenador / programa), convirtiéndose en estímulo para el otro sujeto / individuo. Los desarrollos tecnológicos de la interactividad van buscando poder reproducir las conversaciones cara a cara, a las que se pueden unir la combinación multimodal de los datos (imagen, sonido, movimiento, color, etc.).

Desde el punto de vista de la teoría de la comunicación se debe acudir al análisis transaccional que trata a la comunicación de términos de transacciones, de intercambios, de acción y reacción.

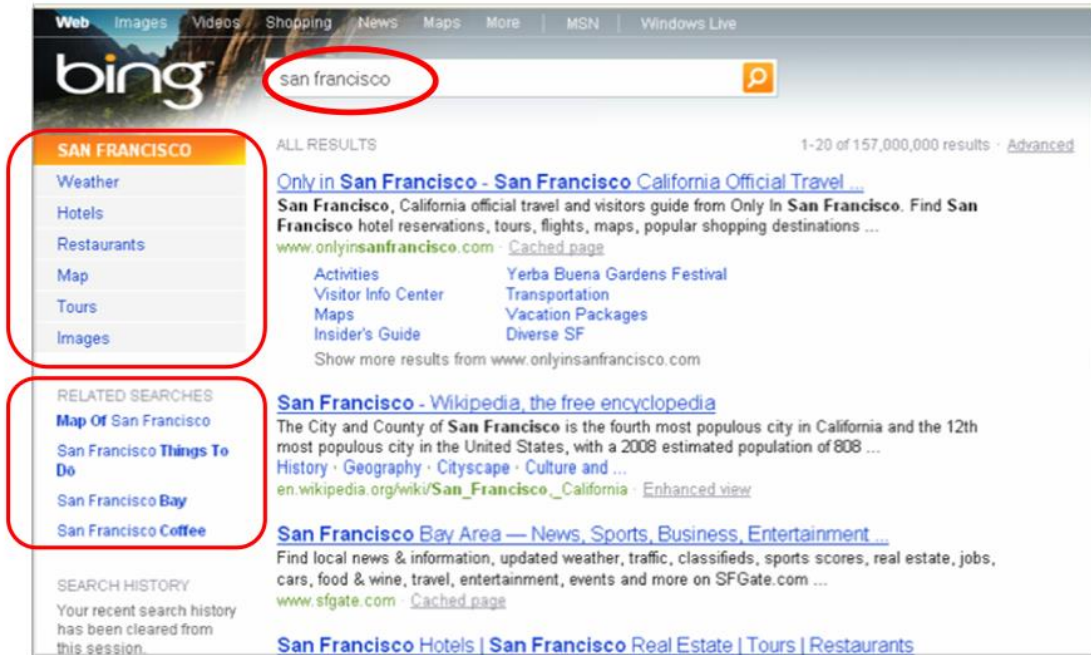
Además de la idea de doble dirección, la interactividad sugiere la “acción” misma que un sujeto realiza, cuya consecuencia es una respuesta por parte del otro “sujeto” ordenador / programa, posibilitando, por ejemplo, la compraventa directa de productos y contratación de servicios online.

- **Personalización.** Hacia y desde el usuario, pudiendo difundirse tantos Mix de Marketing, como usuarios se identifiquen y posibilitar, a su vez, que el visitante escoja lo que desea ver y en qué ítems desea profundizar. Posibilita por ejemplo, la publicidad segmentable de contenido dinámico.
- **Pre y postest de campaña en el universo.** Permite realizar el pre y post-test de campaña sobre el universo poblacional y no sólo con muestras de ese universo (normalmente de representatividad desconocida y conclusiones difíciles de generalizar).
- **Control en tiempo real.** Permite conocer resultados en tiempo real y con una precisión no alcanzada por ningún otro medio (en Internet se puede conocer el número exacto de personas que ven la publicidad –y su perfil- facilitando una gestión más oportuna de la campañas. Se trata de información exacta y disponible inmediatamente, que no se pueden obtener en la publicidad que se realiza en medios convencionales.

basado en ontologías RDF que permite la generación automática de etiquetas, descripciones y sumarios para páginas web.

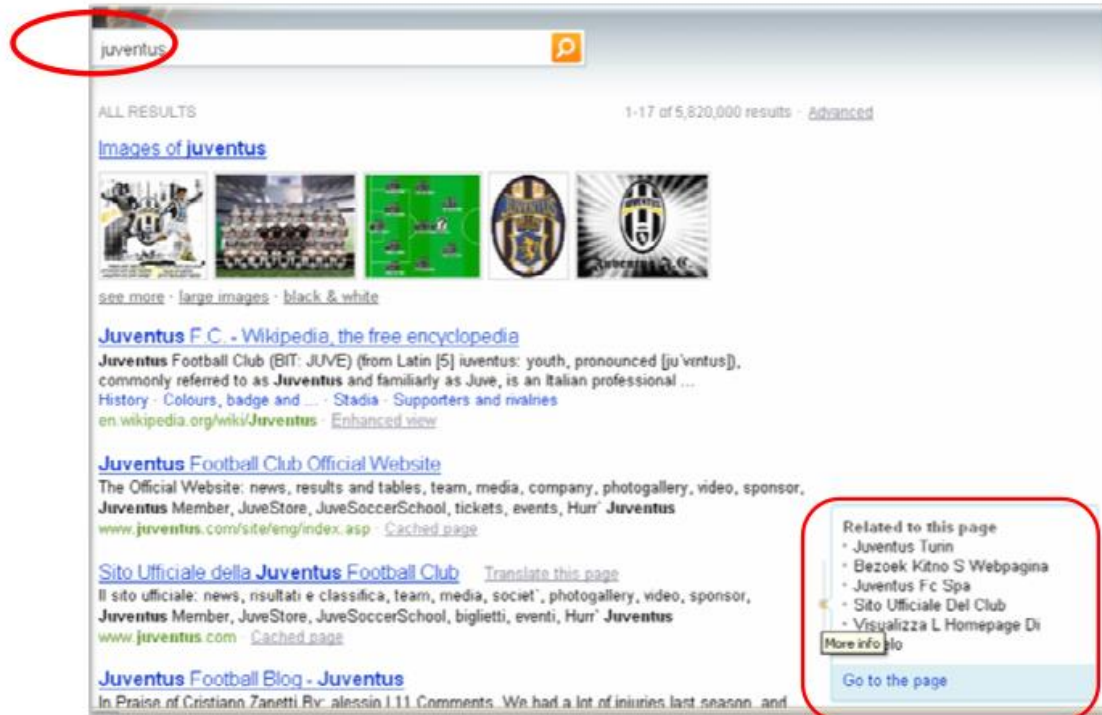
Bing

Bing (<http://www.bing.com/>) es el nuevo motor de búsqueda con el cual la casa de *Redmont* intenta atacar la hegemonía de Google.



Links asociados al término de búsqueda y búsquedas relacionadas al mismo

El motor se basa en ofrecer datos de utilidad inmediata para las necesidades prácticas: según los términos introducidos por el usuario, *Bing* "entiende" el contexto de búsqueda y proporciona información relacionada. Por ejemplo, si introducimos el nombre de una localidad turística, *Bing* nos pide si queremos ver fotos u obtener información sobre hoteles. Además, en contraste con la gráfica escueta de Google, *Bing* tiene varias secciones (Salud, Turismo, Deporte..) en las cuales el usuario puede navegar en la búsqueda de la información que necesita.

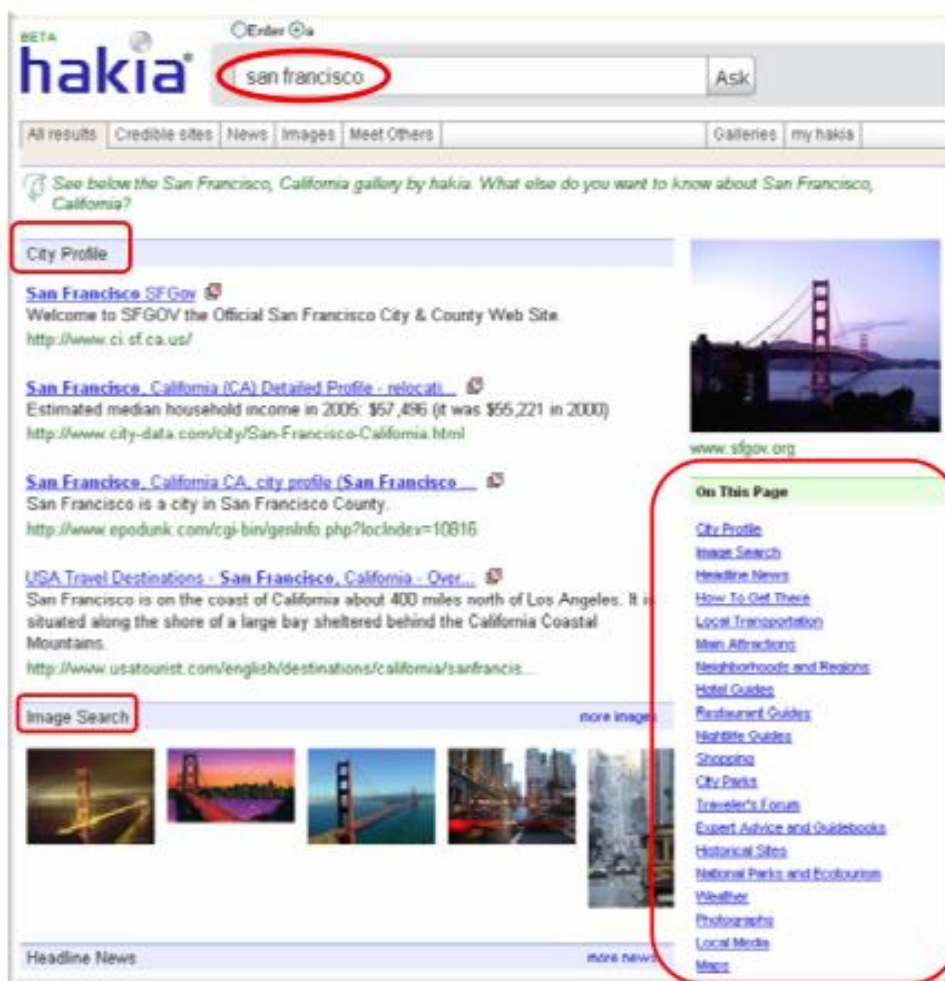


Páginas relacionadas con un resultado de búsqueda

Hakia

Hakia (<http://www.hakia.com/>), como Bing, ofrece una navegación contextual de los resultados. Para la búsqueda "San Francisco" proporciona listas de enlaces organizados por tipo: unos muy genéricos como "Búsqueda de imágenes", "Noticias", otros específicamente relacionados a la búsqueda, como "Perfil de la ciudad", "Como llegar" etc.

De nuevo el valor añadido es la capacidad de gestionar el contexto y la capacidad de presentar resultados de forma diferente según el tipo de búsqueda efectuada.



Resultados de Hakia ordenados según contexto de la búsqueda

Buscadores clásicos: Google y Yahoo hacía búsquedas semánticas

Aunque Google no incorpore explícitamente búsquedas semánticas, hay funcionalidades que nos indican que también el buscador está adoptando tecnología semántica para mejorar sus resultados. Buscando "citizens" Google ofrece búsquedas relacionadas cómo Citizens Bank, relojes Citizen y si buscamos "citizen" al singular, sugiere búsquedas relativas a la ciudadanía.

Google tiene la capacidad de responder a ciertas preguntas con un dato, en lugar de un enlace a una página que contenga los términos buscados. Tecleando "Cual es la capital del estado de Oregon", Google contesta "Salem" demostrando la capacidad de obtener datos estructurados de una fuente semi-estructurada como una página web.

Además recientemente Google ha introducido los que ha llamado *Rich Snippets: información adicional que se presenta debajo de un resultado de búsqueda*. Esta información se muestra para aquellas páginas que tengan contengan metadatos RDFa (un conjuntos de etiquetas que extienden XHTML) o microformatos (otras etiquetas que introducen semántica en las páginas web).

```

<p xmlns:dc="http://purl.org/dc/elements/1.1/"
  about="http://www.example.com/books/wikinomics">
  In his latest book
  <em property="dc:title">Wikinomics</em>,
  <span property="dc:creator">Don Tapscott</span>
  explains deep changes in technology,
  demographics and business.
  The book is due to be published in
  <span property="dc:date" content="2006-10-01">October 2006</span>.
</p>

```

Ejemplo documento con RDFa

En RDFa se basa también *SearchMonkey* (<http://developer.yahoo.com/searchmonkey/>) de Yahoo, una plataforma que permite la creación de páginas web con contenido enriquecido.

Los desarrolladores pueden crear plugins para modificar la presentación de su página en los resultados de Yahoo. Por ejemplo *Last.fm* ha desarrollado un plugin gracias al cual (para los usuarios de Yahoo que decidan activarlo) cada vez que aparezca un resultado proveniente de esa web se vea información adicional con enlaces a la biografía, música y videos de determinado artista.

Portales semánticos

La tecnología semántica no es solamente un instrumento para enriquecer la experiencia de usuario a la hora de efectuar una búsqueda en la web: unos de los objetivos de la web semántica consiste, sobre todo, en crear la capacidad de compartir datos entre aplicaciones. Para esto es fundamental la publicación de datos en formatos estándar que puedan ser reutilizados por terceros.

El proyecto *BBC Music Beta* es una iniciativa de la BBC para crear páginas Web enlazadas y anotadas semánticamente sobre artistas cuya música se emite por las frecuencias de su radio. La información publicada en su página contienen metadatos semánticos que permiten a los apasionados de música explorar conexiones entre artistas que quizás no conocían.

La fuente de datos explotada y enriquecida procede de la *Wikipedia* y de *MusicBrainz*, uno de los partners del proyecto *Linked Data*. *MusicBrainz* es una “meta-base de datos” de contenido musical abierta que contiene información sobre más de 400.000 artistas.

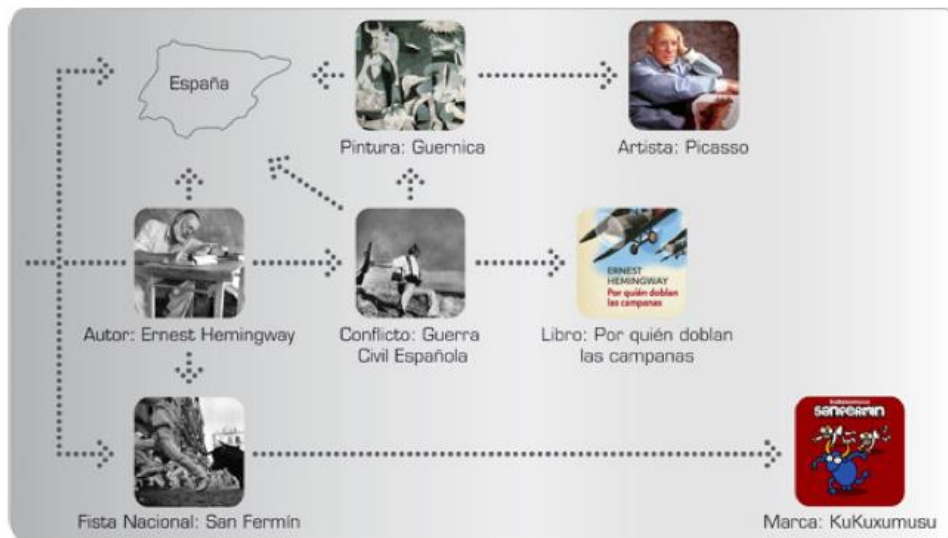
La potencia del proyecto *BBC Music* consiste en la explotación e integración de la información de *MusicBrainz* y *Wikipedia* gracias a un identificador único (una *URI - Uniform Resource Identifier*) que permite enlazar datos de fuentes heterogéneas. Todos los datos publicados están además disponibles en diferentes formatos (RDF, XML, JSON, YAML) para que otras aplicaciones puedan a su vez explotarlos.

Linked Open Data

Una de las grandes ventajas de las herramientas semánticas es la posibilidad de intercambiar datos fácilmente. *Tim Berners-Lee* propuso la creación de “datos vinculados” (linked data) a través de URIs, identificadores únicos de recursos en la Web.

Tim Berners-Lee definió cuatro principios que caracterizan los datos vinculados en su ponencia de presentación para el W3C.[1] Estos principios se pueden parafrasear de la siguiente forma:

- Utilizar URIs para identificar los recursos publicados en la Web.
- Aprovechar el HTTP de la URI para que la gente pueda localizar y consultar estos recursos.
- Proporcionar información útil acerca del recurso
- Incluir enlaces a otras URI relacionadas con los datos contenidos en el recurso, de forma que se potencie el descubrimiento de información en la Web.



El objetivo del proyecto *Linking Open Data* (<http://linkeddata.org/>) desarrollado por el grupo de la W3C encargado de divulgar y explicar la Web semántica (*Semantic Web Education and Outreach*) es ampliar la web con una base de datos común mediante la publicación en la Web de bases de datos en RDF y mediante el establecimiento de enlaces RDF entre datos de diferentes fuentes.

Freebase (<http://www.freebase.com/>) por ejemplo, es una enorme base de conocimiento a la cual cualquier usuario puede participar. Es una colección online de datos estructurados obtenidos desde varias fuentes, incluyendo contribuciones de individuos particulares. El objetivo de *Freebase* es crear un recurso global que permita, tanto a las personas como a las máquinas, acceder a la información de manera más eficiente. *Bing*, *Zemanta* y el *News Timeline* de Google presentan al usuario información enriquecida obtenida desde *Freebase*. La edición on-line del *Wall Street Journal* integra información sobre películas gracias a los datos obtenidos desde *Freebase*, enriqueciendo el contenido y proponiendo contenido relacionado.

Zemanta

La tecnología semántica puede ayudar ofreciendo información contextual al usuario que no la ha pedido explícitamente: mientras se redacta el texto de un blog o un email, *Zemanta* (<http://www.zemanta.com/>) ayuda ofreciendo imágenes semánticamente relacionadas, identificando nombres de personas y empresas y proponiendo añadir enlaces a sus páginas personales/corporativas o a la *Wikipedia*. Además busca noticias relacionadas para que se puedan leer o añadir cómo información interesante a esta entrada. Y esto es aplicable a cualquier tipo de texto: desde un documento relativo a un proceso de negocio a páginas HTML. *Zemanta* permite a todos los "creadores de contenidos" (*bloggers*, periodistas o tan solo personas que escriben un email) reducir el tiempo que normalmente se emplea para enriquecer el texto con enlaces, imágenes y noticias relacionadas.

La capacidad de identificar el contenido (o más bien la semántica) de un texto abre nuevos escenarios. Se puede ofrecer contenido interesante al usuario, con el fin de ayudarle en su redacción con datos relacionados o se podría mostrar publicidad contextual, seleccionada de manera más precisa respecto a una selección por palabras clave.

OpenCalais

OpenCalais (<http://www.opencalais.com/>) es una aplicación que a través de un API que permite etiquetar de forma automática textos que no contienen información semántica.

La plataforma permite incorporar un nivel alto de funcionalidad semántica dentro de un blog, sistema de contenido, sitio web o aplicación. El texto plano se envía a *OpenCalais* para ser enriquecido con metadatos semánticos. Gracias a técnicas de procesamiento de lenguaje natural, algoritmos de aprendizaje automático, *OpenCalais* reconoce las entidades que aparecen en un texto y las relacionas con datos y eventos implícitos en el documento inicial.

Una vez enriquecido un texto con metadatos semánticos es más sencillo crear aplicaciones para:

- Integrar datos,
- Agrupar documentos de forma automática según su contenido,
- Crear nubes de conceptos,
- Sugerir artículos y otros contenidos relacionados como vídeo o fotos

Siri - El asistente virtual

Siri (<http://www.siri.com/>), cuyas orígenes se encuentran en el proyecto militar *CALO* (<http://caloproject.sri.com/>) es un asistente virtual para móviles cuyo objetivo es ayudar al usuario a llevar a cabo tareas como reservar un vuelo o un restaurante o planificar las actividades del fin de semana. *Siri* prevé una forma nueva para que las personas interactúen con los servicios disponibles en internet: su orientación a las tareas le hace destacar respecto a los clásicos motores de búsqueda. Un usuario puede interactuar con *Siri* escribiendo o pronunciando simples frases.

La aplicación interpreta las intenciones y gracias a una potente gestión del contexto es capaz de sorprender con un comportamiento "inteligente". Es posible por ejemplo reservar un vuelo indicando tan solo el destino y la fecha. *Siri* supondrá que el aeropuerto de salida es el más próximo al usuario y si se quiere reservar también un restaurante, las propuestas de *Siri* incluirán resultados cercano a donde se encontrará el usuario en aquel momento, consultando su agenda.



Asistente SIRI

La medicina es un claro ejemplo de producto de alto valor y de gran mercado. Quedan todos aquellos productos que se sitúan en la parte alta a la izquierda en la matriz, donde el valor es grande, pero el mercado es pequeño.

Aplicando la matriz a productos basados en tecnología semántica, obtenemos los resultados que podemos visualizar en la siguiente ilustración: si las aplicaciones como gestión semántica de datos, integración y control de flujo representan un buen balance entre valor y dimensión de mercado, *Tague* cree que la tecnología semántica aplicada a la "Web social" (intercambio de links, minería de datos sobre redes sociales, intercambio de noticias) no constituyen una opción atractiva.

La mejor aplicación de tecnología semántica, según *Tague*, es en el mundo de la publicidad: el valor añadido está en la capacidad de proponer a quien lo necesita, el anuncio que necesita y en el momento adecuado gracias a herramientas capaces de "entender" el contenido, más allá de los mecanismos clásicos como *Google AdWords*.

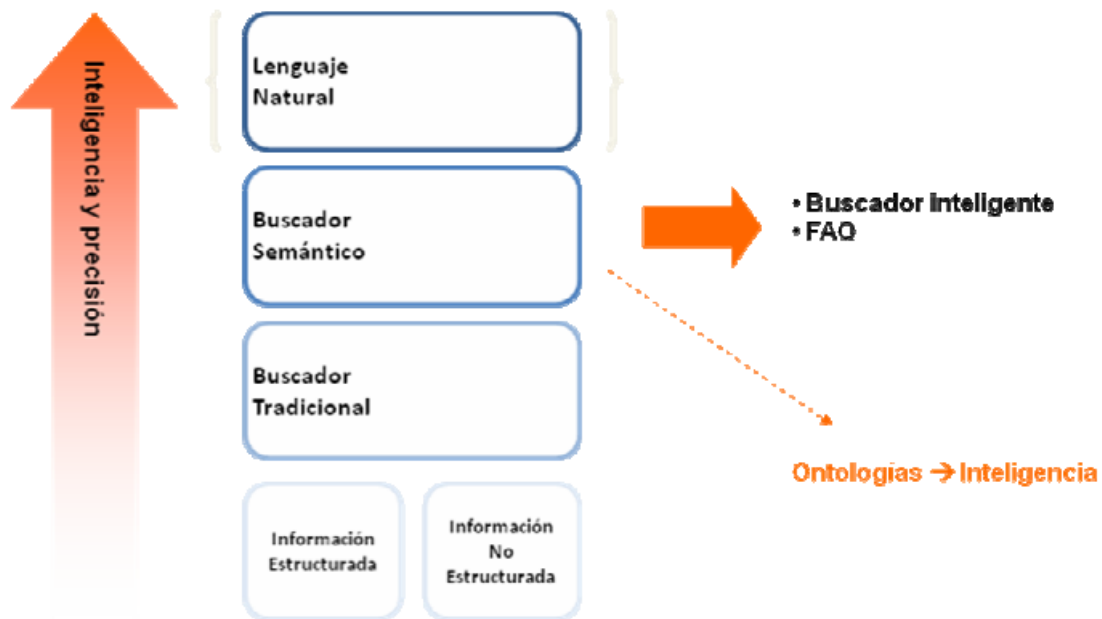
La búsqueda semántica aplicada a un dominio concreto, aunque limite la dimensión del mercado puede ofrecer un alto valor añadido, mientras todas las tareas de edición y creación (y gestión) de contenidos pueden beneficiarse de la Web Semántica, como hemos visto en el caso de *Zemanta* o del *Wall Street Journal* y *Freebase*.

Acceso inteligente a la información

Uno de las posibles aplicaciones de la Web Semántica es sin duda en el campo del acceso inteligente al contenido y en concreto en los buscadores.

En un dominio acotado en el cual es posible definir un modelo formal de los conceptos que se manejan y sus relaciones, un buscador semántico puede aportar un valor añadido importante. A partir de datos más o menos estructurada un buscador tradicional permite encontrar documentos que contengan los términos de una consulta. Buscadores más avanzados son capaces de distinguir variaciones morfológicas y correcciones ortográficas (como en Google, el "Quiso decir...").

Los buscadores semánticos garantizan mas inteligencia y precisión: gracias a la explotación de la información presente en una ontología, el buscador efectúa la búsqueda no solo usando los términos de la consulta, sino también el "contexto semántico", constituido por todos aquellos conceptos relacionados en el dominio de aplicación.



Del buscador del presente al buscador del futuro

Es posible identificar diez características que convierten un buscador en un buscador de éxito:

1. El usuario encuentra lo que busca
2. Si no lo encuentra, sabe que no está
3. Tiene información relacionada que le podría servir
4. Si encuentra demasiado, enseguida filtra lo que necesita
5. Todo con una experiencia visual agradable y sencilla
6. Con un sistema que aprende de sus gustos
7. Se anticipa a sus necesidades
8. Aprende de las necesidades de sus compañeros
9. De manera no intrusiva
10. Integrada en sus aplicaciones habituales

El usuario encuentra lo que busca y si no lo encuentra sabe que no está

Aunque las dos primeras características puedan parecer obvias, un buscador que solo devuelva documentos que contengan los términos introducidos por el usuario es frecuentemente insuficiente.



Los documentos se devuelven por orden de relevancia, resaltando los términos que se han encontrado. Además el buscador es capaz de gestionar las variaciones morfológicas (singulares y plurales, masculino y femenino y conjugación de verbos) y efectúa correcciones ortográficas.

Un buscador semántico que dispone de una ontología puede además expandir la consulta del usuario, usando sinónimos y términos relacionados: buscando por ejemplo "registrar mi pitbull" sería posible encontrar un documento que contenga: "Censo canino" ya que "Registrar" y "Censo" son términos semánticamente cercanos, así como "Pitbull" y "Canino".

Tiene información relacionada que le podría servir

Cuando el usuario no encuentra lo que está buscando es importante ayudarle a reformular su búsqueda. En el ejemplo que se muestra en la imagen siguiente, al no encontrar documentos que contengan "prevención de riesgos laborales" el buscador sugiere documentos que contienen "Medidas de protección", concepto que en la ontología se encuentra cercano al que ha buscado el usuario.

La expansión de los términos de búsqueda se puede ampliar, encontrando documentos que contengan "Guantes", término referido a "Medidas de protección" y "Gloves" traducción de "Guantes"

Si encuentra demasiado, enseguida filtra lo que necesita

Si la expansión de los términos de búsqueda genera un número de excesivo de resultados, es importante introducir mecanismos que permitan de forma fácil e intuitiva filtrar los resultados.



Todo con una experiencia visual agradable y sencilla

Todos los buscadores actuales tienen un interfaz sencilla. A través de una única caja de búsqueda es posible acceder a toda la información necesaria.

Con un sistema que aprende de sus gustos

Gracias a la identificación de usuarios, el buscador puede aprender cuales son los resultados "preferidos" por un usuario.

Se anticipa a sus necesidades

La búsqueda no se tiene que limitar a los documentos, puede incluir Personas (autores de documentos relacionados) y Procesos (cuya documentación tiene relevancia para la búsqueda).

Aprende de las necesidades de sus compañeros

Los resultados de búsqueda se pueden enriquecer a través de la actividad de otros usuarios: el buscador aprende del comportamiento de todos los usuarios.

prevención de riesgos laborales

(por) [Ley de Prevención de Riesgos Laborales \(L.P.R.L.\)](#)
La normativa sobre **prevención de riesgos laborales** está constituida por la (Modificado por Ley 54/2003) La **prevención de riesgos laborales** deberá ...

[Prevención de riesgos laborales](#)
Prevención de riesgos laborales y seguridad industrial. Un punto de referencia sobre higiene en el trabajo, **salud laboral** y **prevenciones** de accidentes ...

[Portal Prevención de Riesgos Laborales y Salud Laboral](#)
Portal de la Junta de Castilla y León que le ofrece toda la información sobre: **riesgos laborales**, **salud laboral**, legislación de **prevención**.

[Servicios de Prevención de Riesgos Laborales, seguridad y salud...](#)
Ofrecemos servicios de **prevención de riesgos laborales** y **Cheques Laborales** en Clínicas y Unidades Móviles de Reconocimientos Médicos, Madrid, Construcción, ...

Usuarios que han buscador "prevención de riesgos laborales" han apreciado:

[Salud laboral - Wikipedia, la enciclopedia libre](#)
Ir a [Prevención de riesgos laborales](#): Los **sinistros laborales** pueden deberse a condiciones medioambientales del centro de trabajo, ...

[Master en Prevención de Riesgos Laborales](#)
Este **Master** figura en el ranking publicado por el diario "El Mundo" el 19 de junio de 2009 como uno de los mejores en el área de "Trabajo, **riesgos laborales** ...

Filtrado y sugerencia social

De manera no intrusiva e Integrada en sus aplicaciones habituales

El acceso a la información no necesariamente tiene que pasar a través de la caja de un buscador: la integración de la tecnología de búsqueda en las aplicaciones habituales del usuario es clave para ofrecer contenido en el momento adecuado, sin una petición explícita por parte del usuario. Hablamos en este caso de **serendipia**: *un descubrimiento o un hallazgo afortunado e inesperado*. Mientras el usuario está redactando un documento es posible ofrecerle de manera automática los documentos que le pueden ayudar, tanto porque están semánticamente relacionados o porque algún otro usuario los ha utilizado redactando un documento similar. Igualmente es posible mostrarle las personas que podrían ayudarle en la redacción del documento (porque se han demostrado expertos) o los procesos involucrados en lo que se está describiendo.

7.3.3 Innovación Abierta

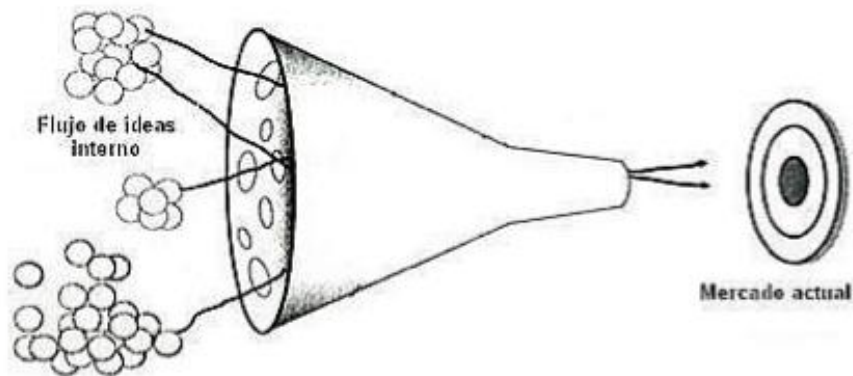
La innovación es un aspecto crucial en toda estrategia de negocio. La innovación permite a las empresas alcanzar y defender la ventaja competitiva, optimizar los procesos de producción, mejorar y crear nuevos productos y de esta manera mejorar los dividendos. Para las grandes compañías innovar es una necesidad imprescindible para evitar que sus productos se conviertan en "commodities" y el mercado se sature rápidamente.

Según de la definición del autor, Henry Chesbrough, la innovación abierta (Open Innovation) es el modelo para el aprovechamiento del conocimiento tanto dentro como fuera del departamento de I+D (y de la compañía en general) acelerando el proceso de innovación interna y expandiendo los mercados para la aplicación de la innovación. "Open Innovation" es un paradigma que asume que las compañías pueden y deben usar ideas tanto internas como externas como posibles vías hacia el mercado. El flujo de conocimiento desde dentro y desde fuera se considera en tres diferentes niveles:

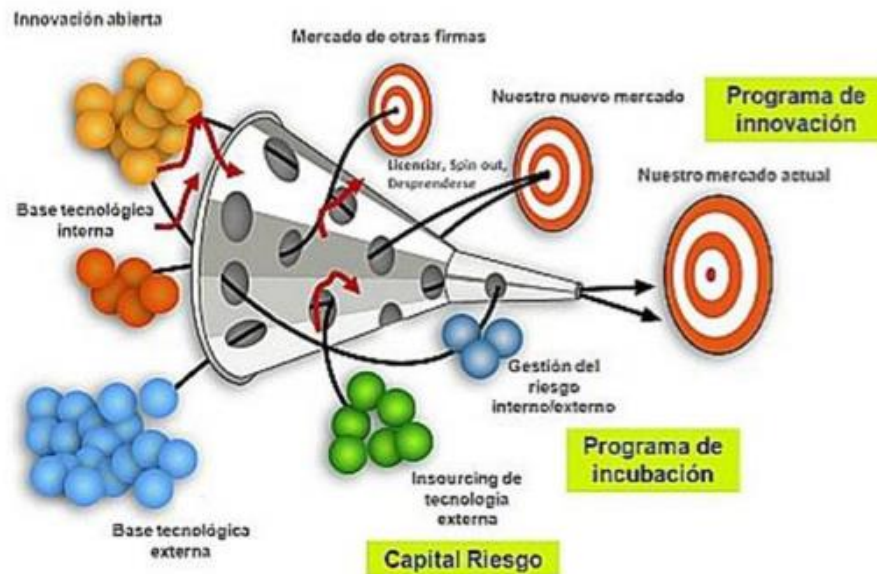
- Interdepartamental

- Entre empresas
- Con el entorno institucional (instituciones gubernamentales, universidades, etc.)

El modelo de innovación abierta representa un cambio radical respecto al modelo tradicional: en el modelo tradicional los productos se desarrollan internamente para luego distribuirse. En el nuevo paradigma el modelo de negocio asume un rol central con el fin de crear y capturar valor gracias a ideas internas y externas que permiten desarrollar nuevas vías hacia el mercado. En un entorno en el cual el conocimiento y la información son ampliamente distribuidos, el modelo tradicional de innovación cerrada presenta sus límites. El embudo cerrado representa la innovación como se entiende a partir de los años '90 (modelo vertical): las ideas y propuestas innovadoras fluyen a través del embudo donde se filtran y refinan antes de salir hacia la unidad de negocio.



El embudo así concebido se revela inadecuado frente a las necesidades y oportunidades de explotación de conocimiento e información externas a la compañía: hay que abrir el embudo para que ideas, conocimiento y tecnología puedan entrar y salir del departamento de investigación.



La colaboración entre miembros de la compañía es especialmente importante desde la revolución que ha implicado la adopción a gran escala de nuevas tecnologías (sobre todo, Internet) que permiten una más fácil comunicación e intercambio de información: el conocimiento se crea y evoluciona más rápidamente que nunca.

La adopción de Innovación Abierta requiere la introducción de un soporte tecnológico que ayude a los usuarios y a los responsables de este proceso a llevar a cabo las tareas que componen el ciclo de vida de la innovación.



En la fase de creación los actores que participan a este proceso redactan ideas, individualmente o en grupos. Las ideas se recogen en un espacio público y las interacciones entre miembros de la

comunidad permiten refinarlas y mejorarlas. En todo momento todos los miembros de la comunidad tienen la posibilidad de visualizar una idea y trabajar sobre ella.

La inteligencia colectiva expresada por la comunidad (votación) ejerce de primer filtro sobre las ideas. Es posible introducir un segundo filtro constituido por expertos designados por la dirección (o por los responsables del departamento de I+D) con el fin de tener una segunda evaluación que tenga en cuenta aspectos específicos como la viabilidad técnica o comercial de una idea.

La decisión sobre las ideas que se implementarán o se hibernarán deberá ser tomada por un organismo formado por miembros directivos de la compañía.

Una vez seleccionadas las ideas "ganadoras" será necesario comunicar a la comunidad la decisión, premiar el/los autor/autores y dar visibilidad a su trabajo con el fin de otorgar reconocimiento social.

La fase siguiente consiste en la implementación de la idea y su consecuente explotación.

En primer lugar es preciso proporcionar a los usuarios un espacio virtual: un portal a través del cual se podrán escribir, modificar, votar y comentar ideas, tanto de forma proactiva o como respuestas a retos del departamento de I+D, suscribirse a RSS y ver estadísticas y nube de conceptos relativos a la actividad de la comunidad. Votos y comentarios podrán ser anónimos o no.

El portal sigue la filosofía de la Web 2.0 para fomentar la colaboración y explotar las ventajas de la inteligencia colectiva.

Gracias a tecnología semántica se enriquece la experiencia del usuario a través de funcionalidades cuales:

- Identificación de términos relevantes de una idea a partir de un diccionario del dominio, considerando variaciones morfológicas, correcciones ortográficas, sinónimos;
- Identificación de ideas con contenido semánticamente parecido (ej. las ideas relativas a "fondos de inversión" están de alguna manera relacionadas a ideas sobre "productos de riesgo");
- Identificación de conocimientos e intereses personales a partir de la actividad de un usuario en el portal;
- Identificación de ideas potencialmente interesantes para un usuario;
- Identificación de usuarios que podrían colaborar en la mejora de la idea;
- Identificación de usuarios con intereses y conocimientos semánticamente cercanos a los intereses de un dado usuario;
- Identificación de los expertos que pueden evaluar una idea
- Identificación de nuevos expertos entre los miembros de la comunidad

Estudios académicos confirman la eficacia de la tecnología semántica como soporte al proceso de innovación:

- El proceso es más eficiente: se reducen los tiempos de evaluación y decisión.
- El proceso es más eficaz: se produce un mayor número de ideas y de mayor calidad.
- Se genera una cultura de innovación a varios niveles en la organización.
- Se crea un repositorio de ideas explotable en el tiempo.
- Se recopila información sobre las tendencias, los intereses y los conocimientos reales de los miembros de la comunidad

7.3.4 El futuro del comercio electrónico

Las nuevas tendencias en el comercio electrónico se trataron en el Experto D.S.C.E. A continuación se incluye un resumen.

El mayor impacto de Internet en las empresas se ha detectado en la mejora de la eficiencia de los procesos y en el perfeccionamiento de sus relaciones con clientes y proveedores. El mayor impacto se produce en los procesos internos, y el menor en los productos y servicios.

Se puede definir el comercio colaborativo como “las relaciones comerciales entre organizaciones que definen unos objetivos comunes y trabajan de modo colaborativo, participando conjuntamente en procesos de negocio previamente acordados”. Esto incluye la colaboración con proveedores y clientes, pero también con competidores. Algunos de los procesos más proclives a formar parte del comercio colaborativo son: planificación y estimaciones, gestión de inventario y de la cadena de aprovisionamiento, diseño y desarrollo de productos, fabricación (capacidad productiva y medios de producción) y los procesos logísticos.

El comercio electrónico ubicuo (uComercio) surge a partir de la convergencia entre el comercio electrónico (eComercio) y el comercio electrónico móvil (mComercio). El uComercio significa poder tener acceso a la Red en cualquier momento, en cualquier lugar y con, prácticamente, cualquier dispositivo de acceso. El número de usuarios de teléfonos móviles que se descargarán ficheros musicales, harán pequeñas compras, accederán a la Intranet de su empresa o se comunicarán sus amigos; se incrementa exponencialmente. En este sentido el mComercio se considera que evolucionará hacia el uComercio.

Para que el mComercio tenga éxito, es necesario que se tengan en cuenta las necesidades de los usuarios:

1. Existe una gran diferencia entre lo que la tecnología puede ofrecer y las expectativas de los consumidores.
2. Los usuarios piden que aumente la velocidad de acceso a Internet a través de dispositivos móviles. También piden que sea más fácil poder teclear texto y la navegación dentro de Internet, utilizando el teléfono móvil.
3. Los usuarios están exigiendo que se reduzcan los costes iniciales y que la tarifa de acceso a Internet, utilizando el teléfono móvil, sea una tarifa plana.
4. Existe gran preocupación respecto a todos los aspectos relacionados con la seguridad y la privacidad de los datos.

Una de las ventajas del mComercio es que los usuarios llevan sus teléfonos móviles prácticamente en todo momento y en cualquier lugar. Esto permite ofrecer servicios que son impensables en el caso del eComercio:

- Servicios que varían según la localización geográfica del usuario.
- Servicios en los que la confidencialidad es muy importante, como el pago electrónico a través del móvil.
- Servicios que permiten unir el comercio tradicional y el electrónico, como la posibilidad de comparar los productos de una tienda con aquellos que están disponibles en Internet; y todo mientras el comprador está físicamente en la tienda.
- Servicios que permiten la creación de comunidades virtuales, favoreciendo la comunicación entre los usuarios e incluyendo aplicaciones de entretenimiento, de un modo mucho más avanzado que con el acceso a Internet a través del ordenador o TV.

Los principales problemas del mComercio son los problemas derivados del tamaño de los dispositivos móviles de acceso a Internet. El pequeño tamaño de la pantalla y del teclado aumenta las dificultades de acceso a Internet, y es una traba añadida que no existía en el eComercio.

La importancia del mComercio es evidentemente grande, sin embargo la verdadera revolución en los negocios reside en el comercio electrónico ubicuo (uComercio). Para poder llegar la uComercio es necesario que haya sensores en todos los lugares, incluyendo la oficina, el coche y el hogar. Estos sensores deben poder intercambiar información con otros dispositivos, para lo que utilizarán ondas de radio de baja frecuencia.

Las tecnología inalámbrica permite alcanzar unos niveles elevados de personalización en el contenido y en los servicios, de un modo integrado y casi instantáneo. Esta tecnología es lo que en el futuro dará lugar al uComercio, que se integrará en la vida cotidiana de las personas. Las personas estarán haciendo transacciones comerciales mientras trabajan, al caminar, al descansar, en las reuniones familiares, etc.

Gracias al uComercio, se podrá conocer la demanda real en todo momento, se podrá influir sobre la demanda y se podrán utilizar esos datos para redefinir las relaciones entre los miembros de la cadena de valor. Para conseguir que el uComercio sea una realidad es necesario que cada dispositivo de acceso a Internet (ordenador, teléfono móvil, lavadoras, etc.) tenga una dirección IP única, por lo que será necesario la utilización de IPv6. Además es fundamental el desarrollo del sistema GALILEO, que sustituirá al actual GPS, para poder conocer la posición geográfica exacta de cada uno de los dispositivos de acceso a Internet.

References

1. Adrián Sánchez-Carmona, Sergi Robles, Carlos Borrego (2015). Improving Podcast Distribution on Gwanda using PrivHab: a Multiagent Secure Georouting Protocol. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* (ISSN: 2255-2863), Salamanca, v. 4, n. 1
2. Baruque, B., Corchado, E., Mata, A., & Corchado, J. M. (2010). A forecasting solution to the oil spill problem based on a hybrid intelligent system. *Information Sciences*, 180(10), 2029–2043. <https://doi.org/10.1016/j.ins.2009.12.032>
3. Bogdan Okresa Durik. (2017) Organisational Metamodel for Large-Scale Multi-Agent Systems: First Steps Towards Modelling Organisation Dynamics. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* (ISSN: 2255-2863), Salamanca, v. 6, n. 3
4. Casado-Vara, R., Chamoso, P., De la Prieta, F., Prieto J., & Corchado J.M. (2019). Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management. *Information Fusion*.
5. Casado-Vara, R., de la Prieta, F., Prieto, J., & Corchado, J. M. (2018, November). Blockchain framework for IoT data quality via edge computing. In *Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems* (pp. 19-24). ACM.
6. Casado-Vara, R., Novais, P., Gil, A. B., Prieto, J., & Corchado, J. M. (2019). Distributed continuous-time fault estimation control for multiple devices in IoT networks. *IEEE Access*.
7. Casado-Vara, R., Vale, Z., Prieto, J., & Corchado, J. (2018). Fault-tolerant temperature control algorithm for IoT networks in smart buildings. *Energies*, 11(12), 3430.
8. Casado-Vara, R., Prieto-Castrillo, F., & Corchado, J. M. (2018). A game theory approach for cooperative control to improve data quality and false data detection in WSN. *International Journal of Robust and Nonlinear Control*, 28(16), 5087-5102.
9. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado J.M. (2019). Social computing in currency exchange. *Knowledge and Information Systems*.
10. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado, J. M. (2019). Social computing in currency exchange. *Knowledge and Information Systems*, 1-21.
11. Chamoso, P., González-Briones, A., Rodríguez, S., & Corchado, J. M. (2018). Tendencies of technologies and platforms in smart cities: A state-of-the-art review. *Wireless Communications and Mobile Computing*, 2018.
12. Chamoso, P., Rivas, A., Martín-Limorti, J. J., & Rodríguez, S. (2018). A Hash Based Image Matching Algorithm for Social Networks. In *Advances in Intelligent Systems and Computing* (Vol. 619, pp. 183–190). https://doi.org/10.1007/978-3-319-61578-3_18
13. Chamoso, P., Rodríguez, S., de la Prieta, F., & Bajo, J. (2018). Classification of retinal vessels using a collaborative agent-based architecture. *AI Communications*, (Preprint), 1-18.
14. Choon, Y. W., Mohamad, M. S., Deris, S., Illias, R. M., Chong, C. K., Chai, L. E., ... Corchado, J. M. (2014). Differential bees flux balance analysis with OptKnock for in silico microbial strains optimization. *PLoS ONE*, 9(7). <https://doi.org/10.1371/journal.pone.0102744>
15. Corchado, J. A., Aiken, J., Corchado, E. S., Lefevre, N., & Smyth, T. (2004). Quantifying the Ocean's CO2 budget with a CoHeL-IBR system. In *Advances in Case-Based Reasoning, Proceedings* (Vol. 3155, pp. 533–546).
16. Corchado, J. M., & Fyfe, C. (1999). Unsupervised neural method for temperature forecasting. *Artificial Intelligence in Engineering*, 13(4), 351–357. [https://doi.org/10.1016/S0954-1810\(99\)00007-2](https://doi.org/10.1016/S0954-1810(99)00007-2)
17. Corchado, J. M., Borrajo, M. L., Pellicer, M. A., & Yáñez, J. C. (2004). Neuro-symbolic System for Business Internal Control. In *Industrial Conference on Data Mining* (pp. 1–10). https://doi.org/10.1007/978-3-540-30185-1_1
18. Corchado, J. M., Corchado, E. S., Aiken, J., Fyfe, C., Fernandez, F., & Gonzalez, M. (2003). Maximum likelihood hebbian learning based retrieval method for CBR systems. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 2689, pp. 107–121). https://doi.org/10.1007/3-540-45006-8_11
19. Corchado, J. M., Pavón, J., Corchado, E. S., & Castillo, L. F. (2004). Development of CBR-BDI agents: A tourist guide application. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 3155, pp. 547–559). <https://doi.org/10.1007/978-3-540-28631-8>
20. Corchado, J., Fyfe, C., & Lees, B. (1998). Unsupervised learning for financial forecasting. In *Proceedings of the IEEE/IAFE/INFORMS 1998 Conference on Computational Intelligence for Financial Engineering (CIFER)* (Cat. No.98TH8367) (pp. 259–263). <https://doi.org/10.1109/CIFER.1998.690316>
21. Costa, Á., Novais, P., Corchado, J. M., & Neves, J. (2012). Increased performance and better patient attendance in an hospital with the use of smart agendas. *Logic Journal of the IGPL*, 20(4), 689–698. <https://doi.org/10.1093/jigpal/jzr021>

22. Cristian Peñaranda, Jorge Agüero, Carlos Carrascosa, Miguel Rebollo, Vicente Julián (2016). An Agent-Based Approach for a Smart Transport System. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* (ISSN: 2255-2863), Salamanca, v. 5, n. 2
23. David Griol, Jose Manuel Molina, Araceli Sanchís De Miguel (2014). Developing multimodal conversational agents for an enhanced e-learning experience. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* (ISSN: 2255-2863), Salamanca, v. 3, n. 1
24. Di Mascio, T., Vittorini, P., Gennari, R., Melonio, A., De La Prieta, F., & Alrifai, M. (2012, July). The Learners' User Classes in the TERENCE Adaptive Learning System. In *2012 IEEE 12th International Conference on Advanced Learning Technologies* (pp. 572-576). IEEE.
25. Fdez-Riverola, F., & Corchado, J. M. (2003). CBR based system for forecasting red tides. *Knowledge-Based Systems*, 16(5–6 SPEC.), 321–328. [https://doi.org/10.1016/S0950-7051\(03\)00034-0](https://doi.org/10.1016/S0950-7051(03)00034-0)
26. Fernández-Riverola, F., Díaz, F., & Corchado, J. M. (2007). Reducing the memory size of a Fuzzy case-based reasoning system applying rough set techniques. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 37(1), 138–146. <https://doi.org/10.1109/TSMCC.2006.876058>
27. Fyfe, C., & Corchado, J. (2002). A comparison of Kernel methods for instantiating case based reasoning systems. *Advanced Engineering Informatics*, 16(3), 165–178. [https://doi.org/10.1016/S1474-0346\(02\)00008-3](https://doi.org/10.1016/S1474-0346(02)00008-3)
28. Fyfe, C., & Corchado, J. M. (2001). Automating the construction of CBR systems using kernel methods. *International Journal of Intelligent Systems*, 16(4), 571–586. <https://doi.org/10.1002/int.1024>
29. Gabriel Santos, Tiago Pinto, Zita Vale, Isabel Praça, Hugo Morais (2016). Enabling Communications in Heterogeneous Multi-Agent Systems: Electricity Markets Ontology. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* (ISSN: 2255-2863), Salamanca, v. 5, n. 2
30. García Coria, J. A., Castellanos-Garzón, J. A., & Corchado, J. M. (2014). Intelligent business processes composition based on multi-agent systems. *Expert Systems with Applications*, 41(4 PART 1), 1189–1205. <https://doi.org/10.1016/j.eswa.2013.08.003>
31. García, O., Chamoso, P., Prieto, J., Rodríguez, S., & De La Prieta, F. (2017). A serious game to reduce consumption in smart buildings. In *Communications in Computer and Information Science* (Vol. 722, pp. 481–493). https://doi.org/10.1007/978-3-319-60285-1_41
32. Glez-Bedia, M., Corchado, J. M., Corchado, E. S., & Fyfe, C. (2002). Analytical model for constructing deliberative agents. *International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications*, 10(3).
33. Glez-Peña, D., Díaz, F., Hernández, J. M., Corchado, J. M., & Fdez-Riverola, F. (2009). geneCBR: A translational tool for multiple-microarray analysis and integrative information retrieval for aiding diagnosis in cancer research. *BMC Bioinformatics*, 10. <https://doi.org/10.1186/1471-2105-10-187>
34. Gonzalez-Briones, A., Chamoso, P., De La Prieta, F., Demazeau, Y., & Corchado, J. M. (2018). Agreement Technologies for Energy Optimization at Home. *Sensors* (Basel), 18(5), 1633-1633. doi:10.3390/s18051633
35. González-Briones, A., Chamoso, P., Yoe, H., & Corchado, J. M. (2018). GreenVMAS: virtual organization-based platform for heating greenhouses using waste energy from power plants. *Sensors*, 18(3), 861.
36. Gonzalez-Briones, A., Prieto, J., De La Prieta, F., Herrera-Viedma, E., & Corchado, J. M. (2018). Energy Optimization Using a Case-Based Reasoning Strategy. *Sensors* (Basel), 18(3), 865-865. doi:10.3390/s18030865
37. Hafewa Bargaoui, Olfa Belkahl Driss (2014). Multi-Agent Model based on Tabu Search for the Permutation Flow Shop Scheduling Problem. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* (ISSN: 2255-2863), Salamanca, v. 3, n. 1
38. Jesús Ángel Román Gallego, Sara Rodríguez González (2015). Improvement in the distribution of services in multi-agent systems with SCODA. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* (ISSN: 2255-2863), Salamanca, v. 4, n. 3
39. Jörg Bremer, Sebastian Lehnhoff. (2017) Decentralized Coalition Formation with Agent-based Combinatorial Heuristics. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* (ISSN: 2255-2863), Salamanca, v. 6, n. 3
40. José Alemany, Stella Heras, Javier Palanca, Vicente Julián (2016). Bargaining agents based system for automatic classification of potential allergens in recipes. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* (ISSN: 2255-2863), Salamanca, v. 5, n. 2
41. Laza, R., Pavn, R., & Corchado, J. M. (2004). A reasoning model for CBR_BDI agents using an adaptable fuzzy inference system. In *Lecture Notes in Computer Science* (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 3040, pp. 96–106). Springer, Berlin, Heidelberg.

42. Leonor Becerra-Bonache, M. Dolores Jiménez López (2014). Linguistic Models at the Crossroads of Agents, Learning and Formal Languages. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* (ISSN: 2255-2863), Salamanca, v. 3, n. 4
43. Li, T., Sun, S., Corchado, J. M., & Siyau, M. F. (2014). A particle dyeing approach for track continuity for the SMC-PHD filter. In *FUSION 2014 - 17th International Conference on Information Fusion*. Retrieved from <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84910637583&partnerID=40&md5=709eb4815eaf544ce01a2c21aa749d8f>
44. Li, T., Sun, S., Corchado, J. M., & Siyau, M. F. (2014). Random finite set-based Bayesian filters using magnitude-adaptive target birth intensity. In *FUSION 2014 - 17th International Conference on Information Fusion*. Retrieved from <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84910637788&partnerID=40&md5=bd8602d6146b014266cf07dc35a681e0>
45. Lima, A. C. E. S., De Castro, L. N., & Corchado, J. M. (2015). A polarity analysis framework for Twitter messages. *Applied Mathematics and Computation*, 270, 756–767. <https://doi.org/10.1016/j.amc.2015.08.059>
46. Mata, A., & Corchado, J. M. (2009). Forecasting the probability of finding oil slicks using a CBR system. *Expert Systems with Applications*, 36(4), 8239–8246. <https://doi.org/10.1016/j.eswa.2008.10.003>
47. Méndez, J. R., Fdez-Riverola, F., Díaz, F., Iglesias, E. L., & Corchado, J. M. (2006). A comparative performance study of feature selection methods for the anti-spam filtering domain. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4065 LNAI, 106–120. Retrieved from <https://www.scopus.com/inward/record.uri?eid=2-s2.0-33746435792&partnerID=40&md5=25345ac884f61c182680241828d448c5>
48. Omar Jassim, Moamin Mahmoud, Mohd Sharifuddin Ahmad (2014). Research Supervision Management Via A Multi-Agent Framework. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* (ISSN: 2255-2863), Salamanca, v. 3, n. 4
49. Pablo Chamoso, Fernando De La Prieta (2015). Simulation environment for algorithms and agents evaluation. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* (ISSN: 2255-2863), Salamanca, v. 4, n. 3
50. Paula Andrea Rodríguez Marín, Néstor Duque, Demetrio Ovalle (2015). Multi-agent system for Knowledge-based recommendation of Learning Objects. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* (ISSN: 2255-2863), Salamanca, v. 4, n. 1
51. Ricardo Silveira, Guilherme Klein Da Silva Bitencourt, Thiago Ângelo Gelaim, Jerusa Marchi, Fernando De La Prieta (2015). Towards a Model of Open and Reliable Cognitive Multiagent Systems: Dealing with Trust and Emotions. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* (ISSN: 2255-2863), Salamanca, v. 4, n. 3
52. Rodríguez-Fernandez J., Pinto T., Silva F., Praça I., Vale Z., Corchado J.M. (2018) Reputation Computational Model to Support Electricity Market Players Energy Contracts Negotiation. In: Bajo J. et al. (eds) *Highlights of Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection*. PAAMS 2018. *Communications in Computer and Information Science*, vol 887. Springer, Cham
53. Rodríguez, S., De La Prieta, F., Tapia, D. I., & Corchado, J. M. (2010). Agents and computer vision for processing stereoscopic images. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 6077 LNAI). https://doi.org/10.1007/978-3-642-13803-4_12
54. Rodríguez, S., Gil, O., De La Prieta, F., Zato, C., Corchado, J. M., Vega, P., & Francisco, M. (2010). People detection and stereoscopic analysis using MAS. In *INES 2010 - 14th International Conference on Intelligent Engineering Systems, Proceedings*. <https://doi.org/10.1109/INES.2010.5483855>
55. Rodríguez, S., Tapia, D. I., Sanz, E., Zato, C., De La Prieta, F., & Gil, O. (2010). Cloud computing integrated into service-oriented multi-agent architecture. *IFIP Advances in Information and Communication Technology* (Vol. 322 AICT). https://doi.org/10.1007/978-3-642-14341-0_29
56. Sigeru Omatu, Tatsuyuki Wada, Pablo Chamoso (2013). Odor Classification using Agent Technology. *DCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* (ISSN: 2255-2863), Salamanca, v. 2, n. 4
57. Sittón, I., & Rodríguez, S. (2017). Pattern Extraction for the Design of Predictive Models in Industry 4.0. In *International Conference on Practical Applications of Agents and Multi-Agent Systems* (pp. 258–261).

