



UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA SUPERIOR EN INFORMÁTICA

**SITIO WEB SEMÁNTICO PARA PATRONES
DE DISEÑO HIPERMEDIA BASADO EN
ONTOLOGÍAS**

PROYECTO FIN DE CARRERA

José Miguel Sánchez Molinero

Marzo de 2009



UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA SUPERIOR EN INFORMÁTICA

**SITIO WEB SEMÁNTICO PARA PATRONES
DE DISEÑO HIPERMEDIA BASADO EN
ONTOLOGÍAS**

PROYECTO FIN DE CARRERA

Autor: José Miguel Sánchez Molinero

Tutora: Susana Montero Moreno

Marzo de 2009

Como ves, tu apoyo y, ante todo,
tu paciencia, no han sido en vano.
Gracias Ruth.

ÍNDICE GENERAL

ÍNDICE GENERAL	VII
ÍNDICE DE TABLAS	X
ÍNDICE DE FIGURAS.....	XI
1 INTRODUCCIÓN	13
1.1 Objetivos del proyecto.....	14
1.2 Fases.....	15
1.3 Organización de la memoria	16
2 ESTADO DEL ARTE.....	18
2.1 Sitios Web.....	19
2.1.1 ¿Qué son?	19
2.1.2 Tipología.....	21
2.2 Diseño Web.....	25
2.2.1 Introducción.....	25
2.2.2 Diseño centrado en el cliente.....	26
2.3 Patrones de diseño para interfaces Web	30
2.3.1 Introducción.....	30
2.3.2 <i>The Design of Sites</i> : Catálogo de patrones	32
2.4 Web Semántica	35
2.4.1 ¿Qué es la Web Semántica?.....	35
2.4.2 Tecnologías utilizadas	41
3 GESTIÓN DEL PROYECTO	46
3.1 Identificación de tareas	46
3.1.1 Ciclo de vida.....	46
3.1.2 Especificación de tareas.....	47
3.2 Especificación de recursos	48
3.3 Planificación de tareas	49
3.3.1 Planificación Inicial.....	50

3.3.2	Planificación Final	52
4	PLANTEAMIENTO DEL PROBLEMA Y SOLUCIÓN	54
4.1	Definición del problema	55
4.1.1	Interfaz de usuario	56
4.1.2	Aplicación Web	57
4.2	Solución propuesta	59
4.2.1	Interfaz de usuario	59
4.2.2	Aplicación Web	61
5	ANÁLISIS	63
5.1	Introducción al análisis	64
5.2	Evaluación de repositorios Web de patrones.....	66
5.2.1	Ajax Patterns	68
5.2.2	hcipatterns.org	71
5.2.3	Hypermedia Design Patterns Repository.....	72
5.2.4	Workflow Patterns.....	74
5.2.5	Yahoo! Design Pattern Library	76
5.2.6	Conclusiones.....	78
5.3	Análisis de los patrones del grupo A.....	81
5.3.1	Introducción.....	81
5.3.2	Patrón <i>Grassroots Information Sites</i> (A6).....	84
5.4	Especificación de requisitos	85
5.4.1	Introducción.....	85
5.4.2	Catálogo de requisitos	87
6	DISEÑO DE LA INTERFAZ DE USUARIO	92
6.1	Introducción al diseño de la interfaz de usuario	93
6.1.1	Construcción del lenguaje de patrones	94
6.1.2	Exploración del lenguaje de patrones.....	98
6.2	Aplicación de patrones.....	98
6.2.1	Patrón <i>Multiple Ways to Navigate</i> (B1).....	98
6.2.2	Patrón <i>Browsable Content</i> (B2)	100
6.2.3	Patrón <i>Hierarchical Organization</i> (B3)	102
6.2.4	Patrón <i>Alphabetical Organization</i> (B5).....	104
6.2.5	Patrón <i>Homepage Portal</i> (C1)	105
6.2.6	Patrón <i>Up-front Value Proposition</i> (C2).....	108
6.2.7	Patrón <i>Site Branding</i> (E1).....	109
6.2.8	Patrón <i>Page Templates</i> (D1).....	111
6.2.9	Patrón <i>Grid Layout</i> (I1)	114

6.2.10	Patrón <i>Separate Tables</i> (L3).....	115
6.2.11	Patrón <i>Above the Fold</i> (I2).....	116
6.2.12	Patrón <i>Fixed-Width Screen Size</i> (I5).....	117
6.2.13	Patrón <i>Content Modules</i> (D2).....	118
6.2.14	Patrón <i>Headlines and Blurbs</i> (D3).....	119
6.2.15	Patrón <i>External Links</i> (K8).....	120
6.2.16	Patrón <i>Pop-up Windows</i> (H6).....	121
6.2.17	Patrón <i>Descriptive, Longer Link Names</i> (K9).....	122
6.2.18	Patrón <i>Obvious Links</i> (K10).....	123
6.2.19	Patrón <i>Printable Pages</i> (D8).....	124
6.2.20	Patrón <i>Low Number of Files</i> (L1).....	125
6.2.21	Patrón <i>HTML Power</i> (L4).....	126
6.2.22	Patrón <i>Location Bread Crumbs</i> (K6).....	127
6.2.23	Patrón <i>Navigation Bar</i> (K2).....	128
6.2.24	Patrón <i>Consistent Sidebars of Related Content</i> (I6).....	130
6.2.25	Patrón <i>Search Action Module</i> (J1).....	131
6.2.26	Patrón <i>Straightforward Search Forms</i> (J2).....	132
6.2.27	Patrón <i>Organized Search Results</i> (J3).....	133
6.2.28	Patrón <i>Action Buttons</i> (K4).....	134
6.2.29	Patrón <i>Writing for Search Engines</i> (D6).....	135
6.2.30	Patrón <i>Distinctive HTML Titles</i> (D9).....	137
6.3	Resumen	138
7	DISEÑO DE LA APLICACIÓN WEB.....	141
7.1	Arquitectura.....	142
7.2	Diagramas de clases.....	143
8	IMPLEMENTACIÓN	164
8.1	Aspectos de implementación.....	165
8.1.1	Internacionalización.....	166
8.1.2	Gestión de errores.....	166
8.1.3	Especificaciones técnicas	166
8.2	Estructura de implementación.....	167
9	CONCLUSIONES	172
9.1	Conclusiones generales.....	172
9.2	Líneas de trabajo futuro	175
	APÉNDICE I: MANUAL DE USUARIO	176
	REFERENCIAS BIBLIOGRÁFICAS.....	190

ÍNDICE DE TABLAS

Tabla 1: Criterios de evaluación utilizados en nuestro análisis de sitios Web	67
Tabla 2: Conjunto de repositorios Web de patrones de diseño a evaluar.....	68
Tabla 3: Resumen del análisis de sitios Web	81
Tabla 4: Plantilla para la definición de los requisitos del sitio.....	86
Tabla 5: Grupos de patrones del catálogo de van Duyne	94
Tabla 6: Relación requisitos del sitio - patrones utilizados.....	140

ÍNDICE DE FIGURAS

Figura 1: Aplicación Web 3 capas.....	20
Figura 2: Factores clave del diseño centrado en el cliente	27
Figura 3: Ejemplo comprimido de uno de los patrones del catálogo de van Duyne	34
Figura 4: Arquitectura de la Web Semántica [Berners-Lee, 2000]	36
Figura 5: Mapa conceptual de la Web Semántica [Rodríguez <i>et al.</i> , 2005]	40
Figura 6: Ciclo de vida del proyecto	47
Figura 7: Modelo de representación del catálogo de patrones de [Montero, 2005]	58
Figura 8: Portada de nuestro libro de referencia “ <i>The Design of Sites</i> ”	60
Figura 9: Lenguaje de patrones para el sitio Web	97
Figura 10: Un primer esquema del <i>framework</i> de navegación.....	100
Figura 11: Se perfila el esquema final del <i>framework</i> de navegación.....	102
Figura 12: Organización jerárquica por categorías de nuestro repositorio de patrones	103
Figura 13: Listaremos en una sola página todos los patrones del repositorio	105
Figura 14: Página de inicio a nivel de bloques.....	107
Figura 15: La imagen de marca estará presente en todas las páginas.....	111
Figura 16: Aplicaremos la técnica <i>Grid Layout</i> en navegación y contenido.....	115
Figura 17: Tablas principales de las páginas de nuestro sitio Web.....	116
Figura 18: Emplearemos un ancho de página fijo con una resolución de 800x600	117
Figura 19: Emplearemos <i>headlines</i> y <i>blurbs</i> para listar patrones.....	119
Figura 20: Se captará una mayor atención al facilitar enlaces a sitios relacionados	120
Figura 21: Las descripciones de patrones dispondrán de una versión imprimible	125
Figura 22: Formato de los menús de miga de pan que emplearemos en nuestro sitio .	128
Figura 23: Una barra lateral izquierda para los accesos principales.....	129
Figura 24: Desde un patrón accederemos a su blog y a los patrones relacionados	130
Figura 25: Esquema de nuestro módulo de búsqueda rápida	131
Figura 26: Los resultados se listarán alfabéticamente con un resumen de búsqueda...	134
Figura 27: Arquitectura de la aplicación Web.....	142

Figura 28: Diagrama de paquetes de primer nivel.....	143
Figura 29: Diagrama de clases principal	145
Figura 30: Diagrama de clases del paquete CORE.....	150
Figura 31: Clases del paquete CORE.BEAN.....	151
Figura 32: Diagrama de clases del paquete ACTIONS	156
Figura 33: Diagrama de clases del paquete ACTIONS.CATALOG	156
Figura 34: Diagrama de clases del paquete ACTIONS.MISCELLANY	157
Figura 35: Diagrama de clases del paquete ACTIONS.RESOURCES	157
Figura 36: Diagrama de clases del paquete SESSION	162
Figura 37: Estructura del sitio Web en carpetas	167
Figura 38: Detalle de la carpeta jsp	169
Figura 39: Detalle de la carpeta con los fuentes de la aplicación.....	170
Figura 40: Página de inicio.....	177
Figura 41: Cabecera del sitio	177
Figura 42: Módulo de búsqueda rápida	178
Figura 43: Menú del sitio	178
Figura 44: Pie de página del sitio	178
Figura 45: Listado en orden alfabético de todos los patrones del repositorio	179
Figura 46: Listado resumen de los aspectos de diseño.....	180
Figura 47: Listado de patrones pertenecientes al aspecto <i>Presentation</i>	180
Figura 48: Listado resumen de los niveles de descripción	181
Figura 49: Listado de patrones pertenecientes al nivel <i>High</i>	181
Figura 50: Ejemplo de página de descripción de un patrón	182
Figura 51: Versión imprimible de la descripción de un patrón	183
Figura 52: Contenido del fichero RDF asociado a un patrón.....	184
Figura 53: Página de comentarios de un patrón	184
Figura 54: Formulario de búsqueda avanzada.....	185
Figura 55: Página de resultados de búsqueda.....	186
Figura 56: Página informativa “Cómo están organizados”	187
Figura 57: Página “Formato utilizado”	187
Figura 58: Página de enlaces a otros repositorios de patrones	188
Figura 59: Página de enlaces de descarga de documentos	189

1 INTRODUCCIÓN

Cada vez está más extendido el uso de patrones de diseño como medios para modelar soluciones a problemas comunes en el desarrollo de software, lo cual se refleja en la continua aparición de nuevas colecciones de patrones (véase por ejemplo [Voelter *et al.*, 2006], [Tidwell, 2006] o [Sanders *et al.*, 2007]). Dichas colecciones nos llegan presentadas de diferentes formas, normalmente en catálogos impresos y empleando principalmente lenguaje natural junto con ejemplos de código y representaciones diagramáticas. La cantidad de documentación impresa se ha incrementado de tal manera que se convierte en una información difícil de manejar.

Mientras que la mayor parte de la actividad se concentra en describir, validar y catalogar patrones, existen pocas iniciativas orientadas a estructurar y organizar este conocimiento de forma consistente o, al fin y al cabo, estudiar cómo los patrones deben ser entregados de modo que se conviertan en algo cómodo de utilizar para los ingenieros de software. Sin embargo, en los últimos años se está generalizando el empleo del formato Web como soporte para el acceso a los más diversos repositorios de patrones de diseño (véase por ejemplo [Reffel y Alpern, 2006], [Saponas *et al.*, 2006]).

Por su parte, las representaciones existentes de patrones no son todo lo efectivas o usables que se pudiera desear, fundamentalmente en lo que respecta a operaciones tan habituales como las búsquedas o a otros conceptos más abstractos como son la compartición y la gestión del conocimiento. En este contexto es donde comienzan a tomar protagonismo las representaciones ontológicas, y la Web Semántica (por extensión al universo digital) [Berners-Lee *et al.*, 2001], como solución a dichas deficiencias.

1. INTRODUCCIÓN

En este proyecto hemos desarrollado un sitio Web semántico como herramienta de acceso a una de estas colecciones, concretamente a la representación ontológica con OWL del repositorio de patrones de diseño hipermedia presentado en [Montero, 2005]. En los próximos apartados definiremos los objetivos generales del proyecto, expondremos resumidamente las fases en las que se ha dividido el desarrollo del sitio Web, y enumeraremos los capítulos que componen este documento.

1.1 Objetivos del proyecto

El objetivo principal del presente proyecto es dar un soporte a la visualización y consulta del repositorio de patrones de diseño hipermedia presentado en [Montero, 2005], mediante el diseño e implementación de un sitio Web basado en tecnologías para la Web semántica. Este repositorio viene definido por una ontología implementada con OWL (lenguaje para la definición e instanciación de ontologías Web) [W3C, 2004b], y cada uno de los patrones que lo componen están formalizados como instancias de dicha ontología por medio de RDF (marco de descripción de recursos) [W3C, 2004a].

Cualquier persona que lleve a cabo un desarrollo hipermedia podrá consultar el repositorio de patrones de diferentes modos, explorando los patrones de manera individual, o a través del lenguaje de patrones que representa, por aspecto de diseño, por nivel de abstracción, o haciendo búsquedas sobre su contenido semántico. De aquí se deduce lo que sería un listado de objetivos generales que deberá satisfacer el sitio Web que vamos a desarrollar:

1. Dar acceso a la visualización de las descripciones de los patrones. Se proporcionarán diferentes modos de explorar estas descripciones, basándonos en el aspecto de diseño y el nivel de abstracción de los patrones.
2. Posibilitar la realización de consultas o búsquedas en el repositorio. Aquí habrá que distinguir entre búsquedas sobre el contenido textual de los patrones y búsquedas sobre su contenido semántico.
3. Todo lo anterior debe ser soportado a través de una interfaz de usuario usable.

1. INTRODUCCIÓN

Para llevar a cabo los dos primeros objetivos será necesario, entre otras cosas, hacer uso de las tecnologías para la Web Semántica existentes en la actualidad. Aunque se trata de una tecnología relativamente nueva y que se puede considerar todavía en fase de experimentación, existe ya mucho software desarrollado y multitud de herramientas y aplicaciones basadas en Web Semántica.

Lo que queremos expresar con el tercero de los objetivos generales del sitio Web es que debemos ser conscientes que no servirá de mucho contar por un lado con un repositorio de patrones completo y bien estructurado, si por el otro nos limitamos a definir una serie de páginas que no atiendan a los principios básicos y deseables de un buen diseño Web. Veremos que para lograr una interfaz de usuario apropiada al sitio Web que desarrollaremos, nos hemos apoyado en el catálogo de patrones de diseño que definió van Duyne en su libro *The Design of Sites* [van Duyne *et al.*, 2002], uno de los más completos y aceptados en la actualidad por la comunidad de desarrolladores Web.

1.2 Fases

El desarrollo del sitio Web se llevará a cabo a través de una serie de fases que consideramos conveniente enumerar a continuación a modo de resumen:

- **Planteamiento del problema y solución:** se establece el punto de partida, se especifica qué buscamos y los problemas a los que nos enfrentamos, y se describen las soluciones que se dispondrán para llevarlo a efecto.
- **Análisis:** el análisis de los patrones pertenecientes al grupo A del catálogo de patrones recogido en [van Duyne *et al.*, 2002], y la evaluación de un conjunto representativo de sitios Web existentes de patrones de diseño, nos ayudará a definir el listado de requisitos (funcionales y no funcionales) que deberá satisfacer en general el sitio Web y en particular la interfaz de usuario.
- **Diseño de la interfaz de usuario:** seleccionaremos del catálogo recogido en [van Duyne *et al.*, 2002] un conjunto de patrones de diseño que consideremos aplicables a nuestro sitio Web, extraeremos un lenguaje de patrones de dicho conjunto, y atravesaremos el lenguaje definido para

1. INTRODUCCIÓN

aplicar cada uno de los patrones obteniendo un diseño concreto para nuestra interfaz de usuario.

- **Diseño de la aplicación Web:** definiremos la aplicación Web a desarrollar en términos de paquetes y clases y sus competencias.
- **Implementación del sitio:** con el diseño ya establecido, llevaremos a cabo la implementación del sitio a través de las tecnologías y abstracciones software seleccionadas en las fases anteriores.

1.3 Organización de la memoria

A continuación se procede a enumerar los capítulos en los que se estructura esta memoria, con un breve apunte de lo que versa cada uno de ellos:

1. **INTRODUCCIÓN;** justificamos la necesidad de desarrollar un sitio Web para dar acceso al repositorio de patrones de diseño hipertexto presentado en [Montero, 2005]. Exponemos los objetivos generales del proyecto y resumimos las fases en las que se divide el desarrollo del sitio Web.
2. **ESTADO DEL ARTE;** aquí realizamos un pequeño análisis de qué son los sitios Web y en qué consiste la Web Semántica.
3. **GESTIÓN DEL PROYECTO;** en este apartado desarrollamos la gestión del proyecto, incluyendo tanto la primera organización del trabajo, como la planificación final que realmente se ha llevado a cabo.
4. **PLANTEAMIENTO DEL PROBLEMA Y SOLUCIÓN;** dejando claro el punto de partida del proyecto, exponemos los problemas a los que nos debemos enfrentar para llevarlo a buen término, y proponemos las soluciones que consideramos más adecuadas para resolverlos.
5. **ANÁLISIS;** evaluamos un conjunto representativo de repositorios Web de patrones de diseño y analizamos los patrones del grupo A de [van Duyne *et al.*, 2002]. De estos dos procesos sacamos unas conclusiones que nos permitirán definir un listado de requisitos para el sitio Web.

1. INTRODUCCIÓN

6. **DISEÑO DE LA INTERFAZ DE USUARIO;** definimos los pasos seguidos para obtener el lenguaje de patrones que caracterizará el diseño de la interfaz de usuario y la aplicación de cada uno de ellos.
7. **DISEÑO DE LA APLICACIÓN WEB;** describe el software a desarrollar en forma de paquetes y clases.
8. **IMPLEMENTACIÓN;** exponemos el aspecto final del sitio Web a través de capturas de pantalla de las diferentes páginas que lo componen, y realizamos una breve descripción de los detalles de implementación.
9. **CONCLUSIONES;** enumeración de las conclusiones obtenidas durante el desarrollo del proyecto.

2 ESTADO DEL ARTE

No deja de ser sorprendente que en la actualidad existan millones y millones de sitios Web y de tan enorme diversidad. Esta diversidad se refleja no sólo en la temática de los sitios, también la observamos en la tipología y en la explosión de diseños con los que uno puede encontrarse navegando por la Web. Desde los inicios de la World Wide Web, tanto el diseño de las páginas como de los sitios Web, ha variado enormemente. De sencillos bloques de texto con unos pocos hiperenlaces, se ha pasado rápidamente a verdaderas obras de diseño gráfico, multimedia e ingeniería informática. Ya no importa sólo el contenido presentado, sino su diseño y su funcionalidad. Las últimas tendencias apuntan al diseño de sitios Web potentes que aporten un valor real y entreguen una experiencia positiva al usuario. En este contexto toman especial relevancia los patrones de diseño, cuyo uso está muy extendido en ámbitos como el diseño software y la orientación a objetos, no así para el diseño Web donde actualmente se puede ver como una disciplina relativamente nueva. También podemos considerar como algo nuevo la Web Semántica [Berners-Lee *et al.*, 2001], una Web extendida dotada de mayor significado que pretende solventar algunos de los problemas que presenta la Web como la conocemos hasta ahora.

Este capítulo pretende introducir brevemente al lector en las ideas expuestas en el párrafo anterior. Para ello, empezaremos exponiendo una clasificación tipológica de sitios Web (véase apartado 2.1). Seguiremos hablando de diseño Web y nos centraremos en una metodología basada en el acercamiento al modo de pensar del cliente o destinatario de un sitio Web (véase apartado 2.2). En el apartado 2.3, hablaremos del uso de patrones en el ámbito del diseño Web. Para finalizar, haremos una breve exposición de qué es la Web Semántica y de las tecnologías empleadas al respecto en el presente proyecto (véase apartado 2.4).

2.1 Sitios Web

2.1.1 ¿Qué son?

A lo largo de los últimos 15 años, la Web ha sido asimilada por millones de personas como un potentísimo medio de comunicar e intercambiar información. Hoy día, la Web es algo cotidiano para una gran parte de los más de mil millones de usuarios de Internet que hay en todo el mundo. Sus utilidades son diversas y su impacto en la economía mundial es apreciable. No sólo hay documentos de texto; hay imágenes, vídeos, música, se pueden comprar cosas, hacer reservas,...

Los sitios Web modernos permiten la captura, procesamiento, almacenamiento y transmisión de datos, para su uso inmediato. Y esto es conseguido a través aplicaciones Web. Páginas de logado, webmail, formularios de solicitud, carros de la compra, sistemas de gestión de contenido, etc..., dan forma a los actuales sitios Web. Estos son ejemplos comunes de aplicaciones Web.

Una **aplicación Web** es una aplicación informática que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet. Las aplicaciones Web son populares debido a la practicidad del navegador Web como cliente ligero. Un **navegador Web** es una aplicación informática que permite a los usuarios obtener información e interactuar con contenido alojado en páginas Web dentro de un sitio Web. La habilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad. Las aplicaciones Web permiten, por tanto, generar información dinámicamente que será presentada al usuario o visitante de un sitio Web, a través del navegador preferido por éste.

En los primeros tiempos de la computación cliente-servidor, cada aplicación tenía su propio programa cliente y su interfaz de usuario. Éstos tenían que ser instalados separadamente en cada estación de trabajo de los usuarios. Una mejora al servidor, como parte de la aplicación, requería típicamente una mejora de los clientes instalados en cada una de las estaciones de trabajo, añadiendo un costo de soporte técnico y disminuyendo la eficiencia del personal.

En contraste, y como ya se ha dicho, las aplicaciones Web generan dinámicamente una serie de páginas en un formato estándar (HTML o XHTML)

2. ESTADO DEL ARTE

soportado por todos los navegadores Web. Se utilizan lenguajes interpretados del lado del cliente, tales como JavaScript, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página Web individual es enviada al cliente como un documento estático, pero la secuencia de páginas proviene de una experiencia interactiva.

Una ventaja significativa en la construcción de aplicaciones Web que soporten las características de los navegadores estándar es que deberían funcionar igual independientemente de la versión del Sistema Operativo instalado en el cliente. En vez de crear clientes para Windows, Mac OS X, GNU/Linux, y otros Sistemas Operativos, la aplicación es escrita una vez y es mostrada casi en todos lados.

Aunque muchas variaciones son posibles, una aplicación Web está comúnmente estructurada como una aplicación de tres capas (véase la figura 1.1.1). En su forma más común, el navegador es la primera capa, un motor usando alguna tecnología Web dinámica (ejemplo: CGI, PHP, Java Servlets o ASP) es la capa de en medio, y una base de datos como última capa. El navegador Web manda peticiones a la capa media, que la entrega valiéndose de consultas y actualizaciones a la base de datos generando una interfaz de usuario.

A medida que crezca el número de empresas y particulares que se beneficien de las ventajas de la Web, de igual forma crecerá el uso de aplicaciones Web y otras tecnologías relacionadas.

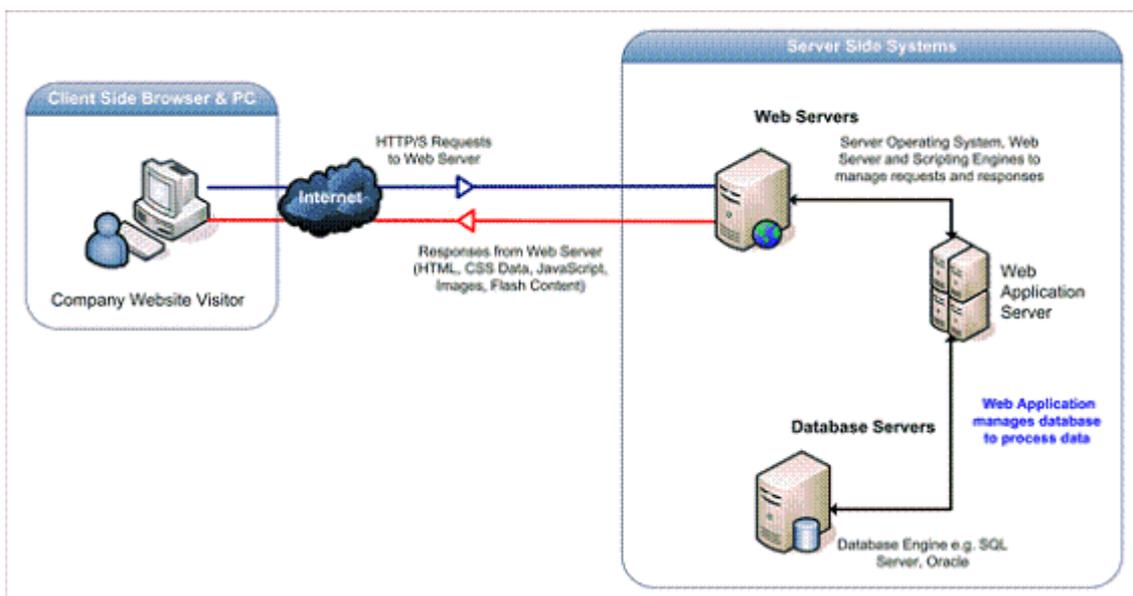


Figura 1: Aplicación Web 3 capas

2.1.2 Tipología

No es fácil establecer una tipología en Internet. Los modelos de sitios Web son muy variados y en muchos casos se mezclan y confunden. La clasificación que exponemos a continuación [Ferrerres, 2006] supone un buen acercamiento.

Webs corporativas

Ninguna compañía se plantea hoy en día prescindir de su página Web. Las Web permiten una comunicación directa con clientes, accionistas, medios y público interesado en cualquier lugar del mundo, las 24 horas del día. Las corporativas son una de las tipologías de Web más extendidas y estandarizadas que existen.

Las secciones que no deben faltar en una Web corporativa típica son: Quiénes somos (o Acerca de), Productos y Servicios, Sala de Prensa, Trabaja con Nosotros, Contacto y, en el caso de las empresas cotizadas, Información para Accionistas e Inversores, así como sobre Responsabilidad Corporativa. Adicionalmente, cada compañía puede personalizar su presencia en Internet para reflejar sus compromisos o su posicionamiento, por ejemplo, una organización respetuosa con el medio ambiente puede incluir una sección sobre dicho tema.

Catálogos Web

Cuando la mayor parte del espacio de una Web está dedicado a presentar la oferta de una compañía, estamos ante catálogos. Son Web en las que prima lo funcional, por lo que lograr una correcta organización de los contenidos es una cuestión clave. Los criterios para agrupar la información pueden ser muy variados, incluso convivir varios en un mismo website, de modo que los usuarios puedan buscar, por ejemplo, ordenadores por su tipología (portátiles, sobremesa, servidores...) o por el tipo de público al que van dirigidos (usuarios domésticos, pequeñas empresas, grandes corporaciones...).

La estructuración de un Catálogo Web se articula en torno a tres tipos de páginas. En la página de inicio o homepage se sitúan enlaces a las principales categorías de productos. Cada categoría, a su vez, tiene su propia portada, en la que se exhiben las últimas novedades o una relación de los artículos más vendidos. El tercer tipo de página es la ficha detallada; existe una por producto y se trata de una pieza clave, que debe haber sido diseñada cuidadosamente para ofrecer todas las características de los

2. ESTADO DEL ARTE

artículos de una forma estandarizada, de lo contrario, la consulta y actualización de un catálogo con miles de registros sería inviable.

Comercio electrónico

Si a los catálogos y escaparates les añadimos la posibilidad de comprar online nos adentramos en el territorio de los comercios electrónicos. El usuario que adquiere un traje online no lo puede tocar, pero quiere verlo con detalle y desde diferentes ángulos para no errar en su decisión. A continuación, entra en juego todo el proceso de pago (con sus implicaciones en materia de seguridad), logística y postventa. Deben ser Web atractivas, personalizadas, que proporcionen un sentido de comunidad, y que den confianza y seguridad.

Intranets y extranets

Internet se puede usar como red privada para compartir información, dando lugar a las intranets y a las extranets (resultantes de la interconexión de dos o más intranets). Una intranet típica es la que habilita una organización para sus empleados. Las extranets, por el contrario, facilitan las relaciones con proveedores, clientes u otros actores. El acceso a las intranets y extranets no es público, sino restringido.

La intranet es una excelente opción de bajo costo para las empresas. Las redes internas corporativas son unas potentes herramientas que permiten divulgar información de la compañía a los empleados con efectividad, consiguiendo que estos estén permanentemente informados con las últimas novedades y datos de la organización. Tienen gran valor como repositorio documental, convirtiéndose en un factor determinante para conseguir el objetivo de oficina sin papeles.

Medios de comunicación

Internet tiene un potencial incalculable como medio de comunicación. Junto a clásicos del periodismo, nuevos géneros han aparecido al calor de la Web, revolucionando el mapa informativo y los papeles tradicionalmente asignados a los medios. Tras unos comienzos tímidos, los medios están otorgando mayor peso a sus versiones digitales y apostando cada vez más por los contenidos audiovisuales. La participación de los usuarios cobra importancia. Los lectores ya no quieren ser meros receptores, sino comentar, valorar e incluso proponer noticias para que sean publicadas. Los medios de comunicación son la tipología de Web más activa y dinámica, siempre

2. ESTADO DEL ARTE

ensayando nuevas fórmulas de navegación, de disposición de los contenidos o de interactividad con los lectores.

Portales y buscadores

Los portales y buscadores nacieron con la Red ante una necesidad acuciante: permitir encontrar una Web o una información entre la ingente cantidad de sitios que se creaban por todos los lugares del mundo. Ambos, portales y buscadores, pretenden ser la puerta de entrada de los usuarios al ciberespacio, pero con una diferencia, al menos en sus orígenes. Mientras que el portal aspira a ofrecer al usuario todo lo que pueda necesitar y mantenerlo cautivo, el buscador sólo era un lugar de paso hacia la verdadera Web de destino.

La diferencia entre portal y buscador, sin embargo, se ha difuminado. Yahoo! fue inicialmente un directorio y actualmente utiliza tanto la clasificación manual como mediante robots (arañas o *spiders*). Google, por su parte, ha ido sumando servicios vertiginosamente hasta ofrecer todo lo que un usuario pueda imaginar: cuentas de correo gratuitas de capacidad ilimitada, mapas y callejeros, mensajería instantánea, noticias, herramientas de publicación...

En el fondo, el objetivo tanto de los portales como de los buscadores es ser la página de inicio de los usuarios. Lograr un hueco en la memoria del internauta, diferenciarse de los millones de webs que compiten por ser, si no la página de inicio, una de las 7-8 que visita diariamente. En esta filosofía se enmarcan iniciativas como las páginas de inicio personalizadas que han visto la luz en los últimos años.

El gran debate de fondo es, por tanto, si un buscador debe seguir siendo, por definición, una Web de paso cuando el negocio parece estar en las webs de destino, en las que el usuario pasa más tiempo y puede ser sometido, por tanto, a mayor número de impactos publicitarios, fuente de ingresos por excelencia de una Web.

Blogs

Un blog, también conocido como weblog o cuaderno de bitácora (listado de sucesos), es un sitio Web periódicamente actualizado que recopila cronológicamente textos o artículos de uno o varios autores, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente. Habitualmente, en cada artículo, los lectores pueden escribir sus comentarios y el autor darles respuesta, de forma que es posible establecer un diálogo. El uso o temática de

2. ESTADO DEL ARTE

cada blog es particular, los hay de tipo personal, periodístico, empresarial o corporativo, tecnológico, educativo, políticos, etc.

Comunidades virtuales

La World Wide Web permite a individuos y grupos comunicarse, relacionarse y construir cosas juntos. En definitiva, formar comunidades, espontáneas o formales. Se denomina comunidad virtual a aquella comunidad cuyos vínculos, interacciones y relaciones tienen lugar no en un espacio físico sino en un espacio virtual como Internet. Aunque se asocien por lo general con hobbies, lo cierto es que existen comunidades profesionales muy activas que realizan una importantísima tarea en la difusión de conocimiento e incluso como grupos de presión. Los foros y los Wikis son algunos de los entornos colaborativos más populares. Estos últimos requieren un estudio aparte.

Wikis

Un (o una) wiki es un sitio Web colaborativo que puede ser editado por varios usuarios. Los usuarios de una wiki pueden así crear, editar, borrar o modificar el contenido de una página web, de una forma interactiva, fácil y rápida; dichas facilidades hacen de una wiki una herramienta efectiva para la escritura colaborativa.

La tecnología wiki permite que páginas web alojadas en un servidor público (las páginas wiki) sean escritas de forma colaborativa a través de un navegador, utilizando una notación sencilla para dar formato, crear enlaces, etc, conservando un historial de cambios que permite recuperar de manera sencilla cualquier estado anterior de la página. Cuando alguien edita una página wiki, sus cambios aparecen inmediatamente en la web, sin pasar por ningún tipo de revisión previa. Wiki también se puede referir a una colección de páginas hipertexto, que pueden ser visitadas y editadas por cualquier persona.

La principal utilidad de un wiki es que permite crear y mejorar las páginas de forma instantánea, dando una gran libertad al usuario, y por medio de una interfaz muy simple. Esto hace que más gente participe en su edición, a diferencia de los sistemas tradicionales, donde resulta más difícil que los usuarios del sitio contribuyan a mejorarlo. Dada la gran rapidez con la que se actualizan los contenidos, la palabra «wiki» adopta todo su sentido. Un wiki permite que se escriban artículos colectivamente (co-autoría) por medio de un lenguaje de wikip texto editado mediante un navegador. Es mucho más sencillo y fácil de usar que una base de datos.

2.2 Diseño Web

2.2.1 Introducción

El diseño Web es un proceso que consiste en el diseño de sitios Web y páginas Web, con la particularidad de que no se trata simplemente de una puesta en práctica del diseño convencional en Internet o de una mera definición de las facetas visuales, sino que requiere tener en cuenta además una serie de aspectos bien identificados, tales como la navegabilidad, la interactividad y la usabilidad, entre otros.

Desde los inicios de la World Wide Web, tanto el diseño de las páginas como de los sitios Web, ha variado enormemente. Si al principio los hipertextos en la Web se limitaban a plasmar, sobre este nuevo medio, un texto impreso segmentándolo en pequeños bloques y colocando aquí y allá algunos enlaces, en la actualidad las páginas Web se han convertido en verdaderas obras de diseño gráfico, multimedia e ingeniería informática. Integración de bases de datos, servicios online, inclusión de sofisticadas herramientas de búsqueda y recuperación de información, dinamismo, usabilidad e interactividad, éstas son las claves de las Web actuales. Ya no importa sólo el contenido presentado, sino su diseño y su funcionalidad.

Hagamos una distinción en generaciones de lo que ha sido y es en la actualidad el diseño Web [van Duyne *et al.*, 2002]:

- **Primera generación;** hace algunos años la Web era algo nuevo y muy atractivo. La idea de lo que suponía construir una Web se resumía en “constrúyelo y la gente acudirá”. Importaba muy poco por tanto el contenido del sitio. Cualquiera podía crear un sitio Web. Por mucho talento que tuvieran las personas o los equipos que construían los sitios, una vez terminados sólo podían decir, sin más, que tenían un sitio Web.
- **Segunda generación;** coincide con la aparición de las primeras webs comerciales. La idea pasó a ser “anuncia que vendes online y la gente acudirá”. Se invirtieron enormes cantidades de dinero en anuncios y campañas de publicidad para invitar a la gente a visitar sus sitios de comercio electrónico. Incluso empresas consolidadas pusieron “.com” en sus cabeceras postales e impulsaron costosas campañas para advertir a la gente que no se habían quedado atrás. Todo esto no funcionó del todo,

puesto que el diseño Web era complejo y todavía incomprendido. Sin embargo fue a partir de este momento cuando la Web empezó a dar los primeros pasos hacia su verdadero crecimiento. Desde entonces, los diseñadores han explorado literalmente miles de ideas en un esfuerzo por comprender y hacer un uso eficiente de este nuevo medio.

- **Tercera generación;** en la actualidad, el foco ha cambiado hacia la construcción de sitios Web potentes que aporten un valor real y entreguen una experiencia positiva al usuario. El objetivo es que las personas que visitan un sitio Web den puntuaciones elevadas en cuanto a contenido, facilidad de uso, funcionamiento y confianza. En definitiva, que obtengan un alto grado de satisfacción. El desafío existe en todos los niveles; grandes multinacionales, administraciones públicas, servicios internos corporativos, pequeños negocios, organizaciones no gubernamentales,...

Ante el vertiginoso crecimiento de la Web, los diseñadores concentran gran parte de sus esfuerzos en atraer y mantener la atención de los usuarios. La unión de un buen diseño con una jerarquía bien elaborada de contenidos aumenta la eficiencia de la Web como canal de comunicación e intercambio de datos. De esta forma se consigue formar vínculos en el subconsciente del usuario y generar esquemas para captar y fidelizar a nuevos visitantes. Quizás haya triunfado la visión mercantil de la Web como un hipermercado global para hacer negocios, pero a la par, se han desarrollado otros servicios con un enfoque centrado en la transmisión de conocimientos en el desarrollo colaborativo para crear cultura, pensamiento, ciencia y, en definitiva, conocimiento.

2.2.2 Diseño centrado en el cliente

Como se verá en posteriores capítulos, el diseño de nuestro sitio Web se basa en la aplicación de los patrones recopilados en el catálogo definido por van Duyne, *The Design of Sites* [van Duyne *et al.*, 2002]. En esta obra, el autor establece como metodología de diseño Web lo que denomina como *Diseño centrado en el cliente*. En este apartado trataremos de resumir en qué consiste este paradigma basándonos en los propios textos del libro.

El diseño centrado en el cliente incrementa el valor de los sitios Web a través de un mejor diseño y evaluación. Se basa en la idea de acercarse al modo de pensar del cliente, tratando de comprender sus necesidades, las herramientas y tecnologías que usa,

2. ESTADO DEL ARTE

y su contexto social y organizacional. Indica cómo usar ese conocimiento para modelar los diseños, y posteriormente testear los mismos para asegurar que las necesidades del cliente son alcanzadas. Mediante esta metodología de diseño, uno puede estar seguro de que el sitio Web responderá a las características que los clientes necesitan, y de que se diseña de modo que lo comprendan con facilidad.

Es un método que a largo plazo acorta los tiempos y los gastos de desarrollo. Un sitio Web fácil de usar y centrado en el cliente, ayuda a realizar mejores revisiones, reducir el número de errores, disminuir el tiempo que se consume en encontrar cosas, y fundamentalmente incrementa el nivel de satisfacción del cliente. Todo esto supone a su vez, en la medida en que los clientes transmitan su experiencia dentro de un sitio Web a otras personas, que aumente el número de clientes potenciales.

Hay una fuerte correlación entre el mayor nivel de satisfacción y el incremento de beneficios para las Webs comerciales. Esto se deriva del hecho de que encontrar productos y servicios se vuelve más fácil, sumando esto a la reducción en los costes de soporte correspondiente (se reducen las llamadas, correos, devoluciones,...). Es muy probable que los visitantes de un sitio Web lo abandonen si:

- Piensan que es demasiado esfuerzo navegar por el sitio
- Piensan que no tienes el producto o servicio que buscan
- Piensan que tarda mucho en cargarse

El diseño centrado en el cliente trata de solventar estos problemas teniendo presentes ciertos factores que se resumen en forma de gráfico:

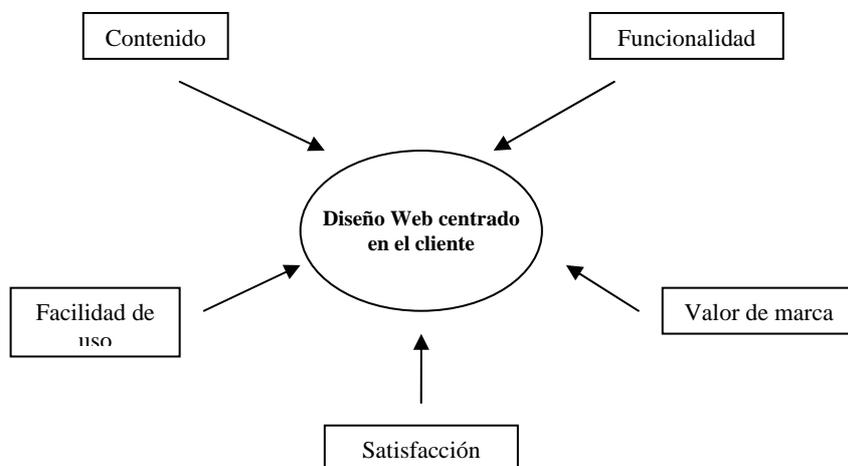


Figura 2: Factores clave del diseño centrado en el cliente

2. ESTADO DEL ARTE

Los autores identifican los diferentes tipos de diseño existentes con el fin de enmarcar el diseño centrado en el cliente y distanciarlo del resto. Son los siguientes:

- **Diseño centrado en el usuario:** el diseño centrado en el cliente está construido sobre la base del diseño centrado en el usuario, añadiendo aspectos o factores a tener en cuenta como son la facilidad de uso y la satisfacción. En particular este tipo de metodología de diseño se centra en la fusión de temas comerciales con temas de usabilidad.
- **Diseño centrado en la compañía:** las necesidades e intereses de la compañía dominan la estructura y el contenido del sitio Web. El error es pensar que lo que piensa la compañía que debería ser un sitio Web es necesariamente lo que los clientes quieren o necesitan.
- **Diseño centrado en la tecnología:** los sitios construidos en las bases de un diseño centrado en la tecnología están construidos a menudo con poca información sobre las necesidades del negocio y del cliente. El problema es que este enfoque da lugar a sitios Web que no son útiles, usables o deseables. En los primeros días de la Web este tipo de sitios eran habituales. En la actual época de madurez se van volviendo cada vez menos comunes.
- **Diseño centrado en el diseño:** es todavía popular en ciertos círculos; “lo que el cliente no comprende es que cuanto menos intervenga, mejor será. Nosotros conocemos lo que es mejor”. Normalmente son sitios lentos de cargar, difíciles de usar y no funcionan correctamente en todos los navegadores. Es un buen método para algunos sitios que tienen como temática algún tipo de arte, pero de ninguna manera para sitios comerciales o informativos.

Los diseños centrados en la compañía, en la tecnología y en el diseño, eran comprensibles en los primeros tiempos de la Web, cuando todavía se estaba en periodo de asimilar y entender esta nueva tecnología. No se consideraba lo que los clientes querían. En contraste con estos estilos el diseño centrado en el cliente da prioridad a los clientes y sus tareas por encima de todo. Actualmente, tener presente la importancia del papel del cliente en el proceso de diseño, ayudará a conseguir éxitos duraderos.

2. ESTADO DEL ARTE

Existen una serie de ideas establecidas y consideradas como ciertas por las compañías que son reacias a cambiar su metodología de diseño hacia el diseño centrado en el cliente, que es conveniente comenzar a dejar a un lado. Son las siguientes:

- ***Un buen diseño es sólo cuestión de sentido común.*** ¿Por qué hay entonces tantos sitios mal hechos? Los equipos de diseño tienen que tener en cuenta que ellos no son los clientes. No pueden ver las cosas desde el mismo punto de vista.
- ***Sólo los expertos crean buenos diseños.*** No es cierto, en realidad cualquier persona puede crear un buen diseño si aporta dedicación e interés.
- ***Las interfaces Web pueden ser rediseñadas justo antes de su lanzamiento.*** Esta idea asume que el sitio Web tiene las características correctas y que esas características están siendo correctamente construidas. Estos son dos riesgos evidentes. Lo conveniente es realimentar el proceso de diseño para que éste resulte correcto el día del lanzamiento.
- ***Un buen diseño supone mucho tiempo y dinero.*** Esto es muy relativo. Con un buen diseño se pueden reducir considerablemente los costes finales. Además de posibilitar la detección de problemas en las primeras etapas cuando aún son fáciles de reparar, previenen que surjan serios problemas que, al fin y al cabo, se traducen en tiempo y dinero.
- ***Un buen diseño es simplemente proporcionar unos buenos gráficos.*** Desgraciadamente, los gráficos son sólo una parte para conseguir comunicar lo que se quiere.
- ***Las guías de interfaces Web conducen a realizar buenos diseños.*** Pero sólo te indican cómo se debe implementar, no otros aspectos como organización, contenidos o navegación entre páginas.
- ***Los usuarios siempre pueden acudir a la ayuda y a la documentación.*** Los usuarios no son normalmente lo suficientemente pacientes como para tratar con una gran cantidad de documentación simplemente para usar un sitio.
- ***Las asesorías de calidad te aseguran el buen funcionamiento de un sitio.*** Las asesorías te aseguran que no hay errores y que el producto se ajusta a la especificación. Esto es muy diferente a los problemas reales que se encuentra el cliente.

2. ESTADO DEL ARTE

El autor distingue tres aspectos fundamentales que dan forma a un diseño centrado en el cliente. Son los siguientes:

- **Principios;** son conceptos a alto nivel que guían el proceso entero de diseño. Dos son los principios básicos: Conocer al cliente, e involucrar al cliente en el proceso de diseño e implementación.
- **Procesos;** se refieren a cómo poner los principios en práctica.
- **Patrones;** solucionan problemas de diseño. Los patrones son un lenguaje, un vocabulario común que te permite articular una infinidad de diseños Web.

Para el diseño de la interfaz de usuario de nuestro sitio Web nos hemos basado en el tercero de los aspectos del diseño centrado en el cliente. Más adelante veremos como hemos estudiado y analizado el catálogo de patrones de van Duyne para construir un lenguaje que guiará el diseño de la interfaz de nuestro sitio Web.

2.3 Patrones de diseño para interfaces Web

2.3.1 Introducción

Los patrones de diseño tienen su origen en el arquitecto Christopher Alexander. En su libro *A Pattern Language: Towns, Buildings, Construction* [Alexander *et al.*, 1977], defendía la creación de un lenguaje común basado en patrones para guiar el proceso de diseño y creación. Hacía hincapié en que se tratase de un lenguaje completo que conectase los patrones relacionados y que mostrase como interactúan unos con otros, ya que consideraba que los patrones aislados eran de poca utilidad. En su momento, fue una reacción contra el diseño de la arquitectura contemporánea, la cual no tenía suficiente consideración hacia las necesidades humanas, la naturaleza, el crecimiento y la comunidad. De acuerdo con Alexander:

"Un patrón describe un problema que sucede repetidamente y la solución a ese problema, de manera que esta solución pueda aplicarse millones de veces sin tener que hacerlo de la misma forma dos veces"

Por tanto, y dicho de una manera muy sencilla, un patrón de diseño es una descripción formal de un problema y su solución.

2. ESTADO DEL ARTE

Los patrones hacen frente a los problemas de diseño, capturando la esencia de los problemas y sus soluciones de una forma compacta. Describen el problema en profundidad, la solución más racional al mismo, cómo aplicar la solución, y algunos de los beneficios derivados de aplicarla. Si podemos aprender a reconocer la existencia de patrones conocidos en los nuevos problemas que puedan plantearse, podremos igualmente aplicar las mismas técnicas para resolverlos. Es decir, los patrones nos dan un medio para reconocer y hacer referencia a problemas conocidos, además de disponer de un medio para comparar soluciones alternativas siendo plenamente conscientes de las consecuencias de cada alternativa.

El primer intento por aplicar el concepto introducido por Christopher Alexander en el diseño de las interfaces de usuario se dio por Ward Cunningham y Kent Beck, quienes adaptaron la propuesta de Alexander y crearon cinco patrones de interfaz: *Window per task*, *Few panes*, *Standard panes*, *Nouns and verbs*, y *Short Menu* [Cunningham, 1987]. El interés por aplicar este paradigma a la Ingeniería de Software se manifiesta a principios de los años 90, donde los personajes más destacados son Erick Gamma, Richard Helm, Ralph Johnson y John Vlissides; en el 95 publicaron su famoso libro *Design Patterns: Elements of reusable Object Oriented Software* [Gamma et al., 1995]. En él recopilaron y documentaron 23 patrones de diseño aplicados usualmente por expertos diseñadores de software orientado a objetos. Los autores pretendían que el catálogo propuesto por ellos, fuera aceptado y compartido por la mayoría de los expertos de ésta área de investigación.

Los patrones, supusieron, por tanto, uno de los últimos avances de la tecnología orientada a objetos para el desarrollo de software. Por un lado, los diseñadores y analistas de software más experimentados trataban de aplicar sus conocimientos y experiencias para encontrar soluciones elegantes a problemas, en ocasiones, bastante similares. Por otra parte, las metodologías orientadas a objetos tienen como uno de sus principios básicos “no reinventar la rueda” para la resolución de éstos problemas. Por tanto, el concepto de patrón, se convirtió rápidamente en una parte muy importante de las tecnologías orientadas a objetos.

En contraste con el desarrollo de software, el diseño de interfaces de usuario, en particular dentro del ámbito Web, se sigue llevando a cabo en su mayor parte con un proceso de desarrollo creativo y poco estructurado. Esto está dando lugar a grandes dificultades de uso en una gran parte de la multitud de sitios Web existentes. Por tanto,

2. ESTADO DEL ARTE

se hace necesario que el diseño de interfaces Web abandone el estilo de desarrollo “artesanal”, para adoptar un proceso sistemático y bien definido. Muchos diseñadores Web llegan a utilizar modelos de otras tecnologías como UML para documentar todo el proceso de desarrollo, porque caen en el error de considerar que no existen modelos y métodos dentro de este ámbito.

La adopción de los patrones de diseño dentro de la comunidad de diseñadores Web ha permitido esa estructuración y sistematización del proceso de construcción de sitios Web. Los patrones de diseño contribuyen a la generación de modelos mentales respecto a elementos específicos o incluso a tipos de sitios en general. El uso de patrones garantiza que los usuarios se sientan cómodos y utilicen de un modo más apropiado un sitio Web.

En el siguiente apartado hablaremos del catálogo de patrones de diseño para interfaces Web recogido en *The Design of Sites* [van Duyne *et al.*, 2002], como ejemplo de uso de patrones en el ámbito del diseño Web.

2.3.2 *The Design of Sites: Catálogo de patrones*

El catálogo de patrones de diseño Web definido por van Duyne se centra en los clientes y sus necesidades, pretendiendo ser una reacción frente a la multitud de patrones de diseño implícitamente en uso que no adoptan un enfoque de diseño centrado en el cliente.

Muchos de los patrones recogidos en el catálogo reflejan cómo los clientes entienden e interactúan con los sitios Web. Las personas que navegan por la Web arrastran una experiencia y un conocimiento de cómo funcionan las cosas. Por ejemplo son capaces de reconocer cosas como los enlaces azules, los botones, los procesos de logado o de carrito de la compra, como medios potentes de hacer cualquier sitio Web sencillo fácil de usar.

Algunos de los patrones reflejan características abstractas que perfilan grandes sitios Web, como es la confianza, el valor percibido y la fiabilidad. A la hora de diseñar un sitio se deberán integrar tales características, reafirmarlas y reforzarlas en cada punto de contacto con los clientes. Estos patrones describen la esencia de esas cualidades y cómo éstas pueden ser incorporadas al sitio Web.

2. ESTADO DEL ARTE

Cada patrón del repositorio presenta los ingredientes esenciales de un patrón de diseño: una explicación del problema básico y la descripción de la solución a nivel general, una enumeración de los motivos que hacen necesario un diseño, y numerosas decisiones y compromisos que deben adoptarse si se utiliza el patrón. Además y lo que es más importante, hace referencia a otros patrones relacionados que afectan a cómo el patrón en cuestión será usado. Como hiciera Alexander dentro de su lenguaje de patrones [Alexander *et al.*, 1977], cada patrón del catálogo está conectado a diferentes patrones de más alto y de más bajo nivel de abstracción. Así, un patrón ayudará a completar los patrones de mayor nivel de abstracción que hay por encima de él, y será completado por los patrones de menor nivel de abstracción que tiene por debajo.

En lo referente al formato de los patrones, éstos presentan una estructura constituida por seis partes o secciones, que presentan muchas similitudes con el formato que presentan los patrones de otros dominios:

- **Nombre:** nombre dado a la solución. Cada patrón además queda identificado por un código formado por una letra que referencia el grupo al que pertenece el patrón, y un número.
- **Contexto:** perfila un contexto para el patrón, describiendo a otros patrones que te dirigen al mismo y cómo se relacionan, así como el alcance del patrón.
- **Problema:** se trata de una frase concisa para señalar el problema específico al que atiende el patrón.
- **Motivaciones:** se extiende en el problema, describiéndolo con más detalle, analizando cómo la gente, sus tareas, la tecnología, y la sociedad afectan al problema de diseño.
- **Solución:** se trata de una frase concisa acerca de cómo resolver el problema acompañada de un pequeño gráfico que permita visualizar la solución.
- **Otros patrones a considerar:** recomendaciones acerca de patrones más detallados que ayudan a completar el patrón en cuestión, que habrá que examinar y evaluar para seleccionar aquellos que se ajustan a las necesidades del problema.

2. ESTADO DEL ARTE



Figura 3: Ejemplo comprimido de uno de los patrones del catálogo de van Duyne

En su catálogo, van Duyne propone una serie de grupos organizados por letra y por nombre cada uno de los cuales contiene una colección de patrones relacionados temáticamente. Según el esquema propuesto los primeros grupos deberían ser los primeros en usarse en el proceso de diseño. Dentro de la descripción de un patrón se identifican patrones relacionados en varias de las secciones; en el contexto, en las motivaciones y en otros patrones a considerar. Esta red de patrones proporciona un modo de seleccionar rápidamente aquellos que se necesitan para completar un determinado diseño, y facilita la articulación de un gran número de diseños para escenarios diferentes.

2.4 Web Semántica

2.4.1 ¿Qué es la Web Semántica?

La **Web Semántica** vendría a ser una extensión de la Web actual dotada de significado, esto es, un espacio donde la información tendría un significado bien definido, de manera que pudiera ser interpretada tanto por agentes humanos como por agentes computerizados. La Web Semántica ha sido impulsada por Tim Berners-Lee, creador de la WWW, y otras personas relacionados con el W3C (World Wide Web Consortium). Según las propias palabras del W3C,

La Web Semántica es una Web extendida, dotada de mayor significado en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida. Al dotar a la Web de más significado y, por lo tanto, de más semántica, se pueden obtener soluciones a problemas habituales en la búsqueda de información gracias a la utilización de una infraestructura común, mediante la cual, es posible compartir, procesar y transferir información de forma sencilla. Esta Web extendida y basada en el significado, se apoya en lenguajes universales que resuelven los problemas ocasionados por una Web carente de semántica en la que, en ocasiones, el acceso a la información se convierte en una tarea difícil y frustrante” [W3C, 2008a].

En mayo de 2001, Tim Berners Lee, James Hendler y Ora Lassila popularizan la idea de la Web Semántica al publicar un artículo titulado "*The Semantic Web: a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities*" [Berners-Lee *et al.*, 2001], donde explican de forma sencilla su idea de la Web Semántica y los primeros pasos que hay que dar para llevarla a cabo.

Según Berners-Lee, la arquitectura de la Web Semántica se podría representar de la siguiente forma:

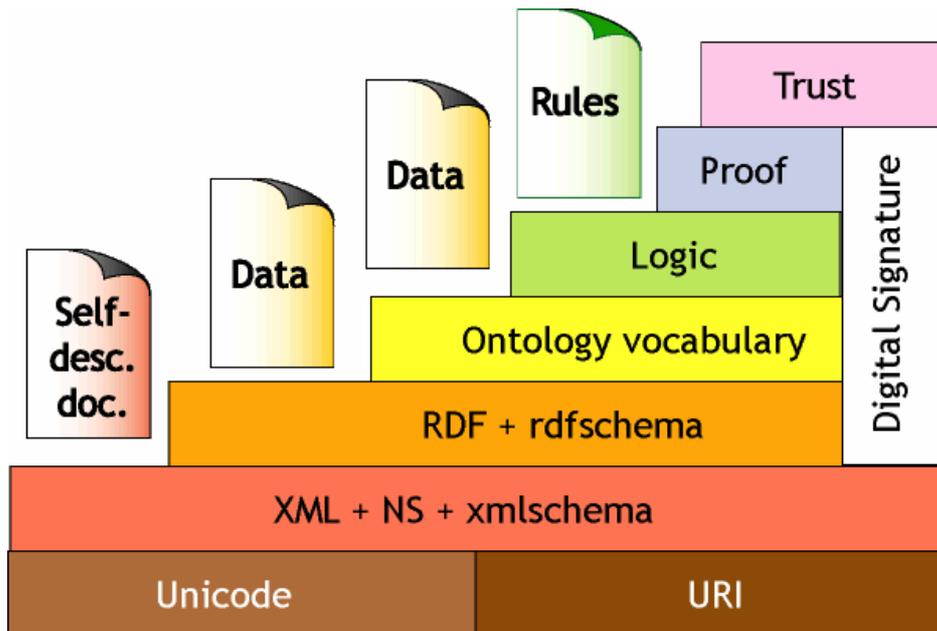


Figura 4: Arquitectura de la Web Semántica [Berners-Lee, 2000]

La mayoría de los sitios Web están contruidos en lenguaje HTML con marcas o etiquetas que se muestran cuando se visualiza el código fuente, pero que permanecen ocultas en la visualización normal de los navegadores y que contienen información sobre el contenido de la página, enlaces hacia otras páginas, formatos de letra, color, párrafos, imágenes, vídeos, etc. Los orígenes de la Web se basaron en el carácter abierto y universal de la base de la Web: el lenguaje **HTML**, y el empleo de archivos ASCII y los gráficos GIF y/o JPG. Esto permite a los buscadores clasificar los documentos HTML de la red y ponerlos en una página Web a modo de índice o catálogo, que se puede mostrar por medio del navegador. Gracias a que el lenguaje HTML se ajusta a unas normas estandarizadas, todos los ordenadores pueden reproducir correctamente esos documentos. Sin embargo, el lenguaje HTML se quedaba corto pues, orientado a la presentación de datos, la información que ofrece es muy limitada, no permite describir datos y no es extensible, esto es, únicamente ofrece un pequeño número de etiquetas. El sistema evolucionó y se realizaron algunas mejoras para hacer este lenguaje algo más dinámico con la introducción de otros elementos como **DHTML**, **Javascript**, **hojas de estilo** e, incluso, se añadieron a la Web otros lenguajes que permitieran ofrecer una información más estructurada, como el lenguaje XML, pero hacen falta otros lenguajes que permitan una descripción más detallada del documento y de su contenido, y que faciliten la comunicación entre los ordenadores. Y también hace falta una nueva

2. ESTADO DEL ARTE

generación de buscadores más inteligentes que puedan leer y evaluar rápidamente los documentos de Internet.

Así pues, el desarrollo de la Web Semántica requiere la utilización de otros lenguajes como el lenguaje estructurado **XML** (Extensible Markup Language) y el lenguaje **RDF** (Resource Description Framework) [W3C, 2004a] que puedan dotar a cada página, a cada archivo y a cada recursos o contenido de la red, de una lógica y un significado, y que permitan a los ordenadores conocer el significado de la información que manejan con el fin de que esta información pueda no sólo ser presentada en pantalla, sino también que pueda ser integrada y reutilizada.

Para poder explotar la Web Semántica, se necesitan lenguajes semánticos más potentes, esto es, lenguajes de marcado capaces de representar el conocimiento basándose en el uso de metadatos y ontologías. Utilizando anotaciones **RDF** y **RDF Schema** [W3C, 2004c] se pueden presentar algunas facetas sobre conceptos de un dominio del conocimiento y se puede, mediante relaciones taxonómicas, crear una jerarquía de conceptos. Pero se precisan lenguajes de marcado (basados en **RDF**) con una mayor expresividad y capacidad de razonamiento para representar los conocimientos que contienen las ontologías. Además, estos lenguajes deben ser estandarizados y formalizados para que su uso sea universal, reutilizable y compartido a lo largo y ancho de la Web. Se necesita un lenguaje común basado en Web, con suficiente capacidad expresiva y de razonamiento para representar la semántica de las ontologías. De esta forma, la utilización de lenguajes tales como **OWL** [W3C, 2004b] es un paso más en la consecución de la Web Semántica.

Es necesario, pues, crear una ontología o biblioteca de vocabularios descriptivos/semánticos, definidos en formato **RDF** y ubicados en la Web para determinar el significado contextual de una palabra por medio de la consulta a la ontología apropiada. De esta forma, agentes inteligentes y programas autónomos podrían rastrear la Web de forma automática y localizar, exclusivamente, las páginas que se refieran a la palabra buscada con el significado y concepto precisos con el que interpretemos ese término. Por lo tanto, para potenciar el uso de ontologías en la Web, se necesitan aplicaciones específicas de búsqueda de ontologías, que indiquen a los usuarios las ontologías existentes y sus características para utilizarlas en su sistema.

2. ESTADO DEL ARTE

Enumeramos a continuación las funciones que aportan las distintas capas de la Web Semántica en función de la arquitectura definida por Tim Berners-Lee (véase la Figura 3):

- **Unicode:** el alfabeto. Se trata de una codificación del texto que permite utilizar los símbolos de diferentes idiomas sin que aparezcan caracteres extraños. De esta forma, se puede expresar información en la Web Semántica en cualquier idioma.
- **URI (Uniform Resource Identifier):** las referencias. URI es el acrónimo de "" o Identificador Uniforme de Recursos, identificador único que permite la localización de un recurso que puede ser accedido vía Internet. Se trata del URL (descripción de la ubicación) más el URN (descripción del espacio de nombre).
- **XML + NS + xmlschema:** se trata de la capa más técnica de la Web Semántica. En esta capa se agrupan las diferentes tecnologías que hacen posible que los agentes puedan entenderse entre ellos. XML ofrece un formato común para intercambio de documentos, NL (namespaces) sirve para cualificar elementos y atributos de nombres usados en XML asociándolos con los espacios de nombre identificados por referencias URI y XML Schema ofrece una plantilla para elaborar documentos estándar. De esta forma, aunque se utilicen diferentes fuentes, se crean documentos uniformes en un formato común y no propietario.
- **RDF + rdfschema:** basada y apoyada en la capa anterior, esta capa define el *lenguaje universal* con el cual podemos expresar diferentes ideas en la Web Semántica. RDF es un lenguaje simple mediante el cual definimos sentencias en el formato de una 3-upla o triplete (sujeto: el recurso al que nos referimos; predicado: el recurso que indica qué es lo que estamos definiendo; y objeto: puede ser el recurso o un literal que podría considerarse el valor de lo que acabamos de definir). El modelo RDF o Resource Description Framework es un modelo común (Framework) que permite hacer afirmaciones sobre los recursos (Description) y que hace posible que estos recursos pueden ser nombrados por URIs (Resource). Por su parte RDF Schema provee un vocabulario definido sobre RDF que

permite el modelo de objetos con una semántica claramente definida. Esta capa no sólo ofrece descripción de los datos, sino también cierta información semántica. Tanto esta capa como la anterior corresponden a las anotaciones de la información (metadatos).

- **Lenguaje de Ontologías:** ofrece un criterio para catalogar y clasificar la información. El uso de ontologías permite describir objetos y sus relaciones con otros objetos ya que una ontología es la especificación formal de una conceptualización de un dominio concreto del conocimiento. Esta capa permite extender la funcionalidad de la Web Semántica, agregando nuevas clases y propiedades para describir los recursos.
- **Lógica:** además de ontologías se precisan también reglas de inferencia. Una ontología puede expresar la regla "Si un código de ciudad está asociado a un código de estado, y si una dirección es el código de ciudad, entonces esa dirección tiene el código de estado asociado". De esta forma, un programa podría deducir que una dirección de la Universidad Complutense, al estar en la ciudad de Madrid, debe estar situada en España, y debería por lo tanto estar formateado según los estándares españoles. El ordenador no "entiende" nada de lo que está procesando, pero puede manipular los términos de modo mucho más eficiente beneficiando la inteligibilidad humana.
- **Pruebas:** será necesario el intercambio de "pruebas" escritas en el lenguaje unificador (se trata del lenguaje que hace posible las inferencias lógicas hecha posibles a través del uso de reglas de inferencia tal como es especificado por las ontologías) de la Web Semántica.
- **Confianza:** los agentes deberían ser muy escépticos acerca de lo que leen en la Web Semántica hasta que hayan podido comprobar de forma exhaustiva las fuentes de información.
- **Firma digital:** bloque encriptado de datos que serán utilizados por los ordenadores y los agentes para verificar que la información adjunta ha sido ofrecida por una fuente específica confiable.

2. ESTADO DEL ARTE

En suma, el objetivo de la Web Semántica es que la Web pase de ser una colección de documentos a convertirse en una base de conocimiento.

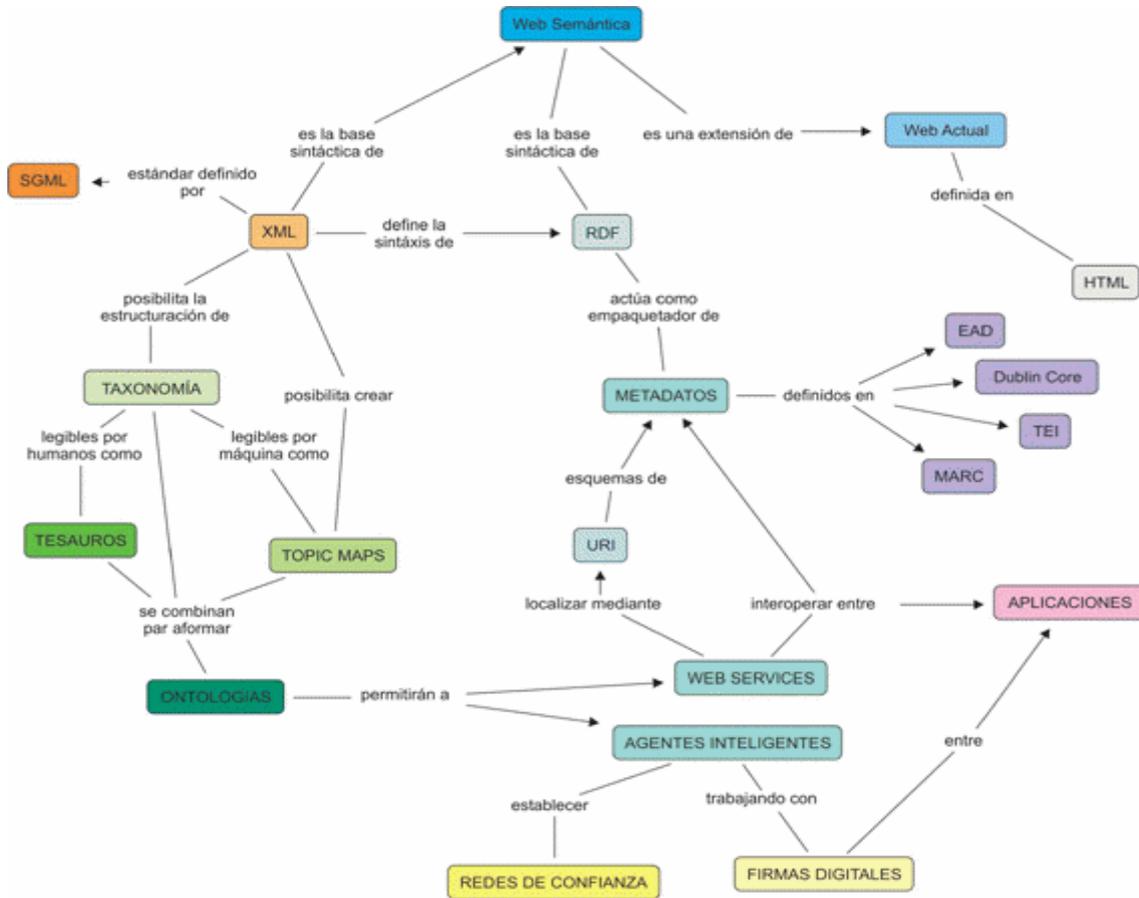


Figura 5: Mapa conceptual de la Web Semántica [Rodríguez *et al.*, 2005]

La principal fuente de información sobre la Web Semántica es el W3 Consortium, una de cuyas actividades actuales es desarrollar y estandarizar un marco común que permita compartir y reutilizar datos entre distintas aplicaciones y plataformas para hacer posible la Web Semántica.

Los proyectos en marcha son numerosos y variados. Sin embargo, la Web Semántica no goza de excesiva popularidad entre los creadores de sitios web debido a que no está pensada para seres humanos, sino para agentes o programas que recorran la World Wide Web e infieran nuevos datos. En los aspectos técnicos ha habido grandes logros ya que existen librerías para lenguajes como Java, PHP, Perl, Python, etc; y también existen distintos instrumentos para inferir información como Jena, CWM, EARL, etc. pero la realidad es que existen muy pocos agentes que humanicen la información de la Web Semántica.

2. ESTADO DEL ARTE

Así pues, la Web Semántica cuenta ya con una nutrida infraestructura de lenguajes y tecnologías para hacerla posible. Sin embargo, para que la antigua tarea humana e intelectual de la identificación, indización y clasificación de documentos pueda hacerse mediante ordenadores sólo podrá llevarse a la práctica si colaboran de forma interdisciplinaria documentalistas, lingüistas, informáticos, arquitectos de la información, ingenieros y expertos en inteligencia artificial, etc.

2.4.2 Tecnologías utilizadas

En este apartado vamos a ver de forma resumida en qué consisten **RDF** y **OWL**, para después pasar a exponer brevemente las dos tecnologías que hemos utilizado en nuestro proyecto para el acceso y manipulación del contenido semántico del sitio Web a desarrollar; Jena y SPARQL. **Jena** [Jena] es un framework de desarrollo para construir aplicaciones basadas en Web Semántica. Por su parte **SPARQL** [W3C, 2008b] se trata de un lenguaje de consulta para documentos RDF.

RDF

El **Marco de Descripción de Recursos** (RDF) es un framework para metadatos en la World Wide Web, desarrollado por el W3C.

Este modelo se basa en la idea de convertir las declaraciones de los recursos en expresiones con la forma **sujeto-predicado-objeto** (conocidas en términos RDF como tripletes). El sujeto es el recurso, es decir aquello que se está describiendo. El predicado es la propiedad o relación que se desea establecer acerca del recurso. Por último, el objeto es el valor de la propiedad o el otro recurso con el que se establece la relación. El modelo RDF se basa en principios perfectamente establecidos de varias comunidades de representación de datos. Las propiedades RDF pueden recordar a atributos de recursos y en este sentido corresponden con los tradicionales pares de atributo-valor. Las propiedades RDF representan también la relación entre recursos y por lo tanto, un modelo RDF puede parecer un diagrama entidad-relación. En la terminología del diseño orientado a objetos, los recursos corresponden con objetos y las propiedades corresponden con objetos específicos y variables de una categoría.

El modelo de datos básico consiste en tres tipos de objetos:

- **Recursos.** Todas las cosas descritas por expresiones RDF se denominan recursos. Un recurso puede ser una página Web completa. Un recurso

puede ser una parte de una página Web; por ejemplo un elemento HTML o XML específico dentro del documento fuente. Un recurso puede ser también una colección completa de páginas; por ejemplo un sitio Web completo. Un recurso puede ser también un objeto que no sea directamente accesible vía Web; por ejemplo un libro impreso. Los recursos se designan siempre por **URIs** más identificadores de anclas opcionales. Cualquier cosa puede tener un URI; la extensibilidad de URIs permite la introducción de identificadores para cualquier entidad imaginable.

- **Propiedades.** Una propiedad es un aspecto específico, característica, atributo, o relación utilizado para describir un recurso. Cada propiedad tiene un significado específico, define sus valores permitidos, los tipos de recursos que puede describir, y sus relaciones con otras propiedades.
- **Sentencias (declaraciones, enunciados).** Un recurso específico junto con una propiedad denominada, más el valor de dicha propiedad para ese recurso es una sentencia RDF. Estas tres partes individuales de una sentencia se denominan, respectivamente, sujeto, predicado y objeto. El objeto de una sentencia (es decir, el valor de la propiedad) puede ser otro recurso o puede ser un literal; es decir, un recurso (especificado por un URI) o una cadena simple de caracteres u otros tipos de datos primitivos definidos por XML. En términos RDF, un literal puede comprender en su contenido marcado XML pero ya no puede valorarse más por un procesador RDF. Existen varias restricciones sintácticas en cómo se puede expresar el marcado en literales.

OWL

El **Lenguaje de Ontologías Web (OWL)** está diseñado para ser usado en aplicaciones que necesitan procesar el contenido de la información en lugar de únicamente representar información para los humanos. OWL puede ser usado para representar explícitamente el significado de términos en vocabularios y las relaciones entre esos términos OWL. Esta representación de términos y sus interrelaciones se denomina **ontología**. OWL tiene mayor capacidad para expresar significado y semántica que XML, RDF, y RDF-S, y, de este modo, OWL va más allá de estos

2. ESTADO DEL ARTE

lenguajes en su capacidad para representar contenido interpretable por un ordenador en la Web. OWL tiene tres sublenguajes, con un nivel de expresividad creciente: *OWL Lite*, *OWL DL*, y *OWL Full*.

La Web Semántica se basará en la capacidad de XML para definir esquemas de etiquetas a medida y en la aproximación flexible de RDF para representar datos. El primer nivel requerido por encima de RDF para la Web Semántica es un lenguaje de ontologías que pueda describir formalmente el significado de la terminología usada en los documentos Web. Si se espera que las máquinas hagan tareas útiles de razonamiento sobre estos documentos, el lenguaje debe ir más allá de las semánticas básicas del RDF Schema. OWL ha sido diseñado para cubrir esta necesidad de un lenguaje de ontologías Web. OWL forma parte de un conjunto creciente de recomendaciones del W3C relacionadas con la Web Semántica:

- **XML** proporciona una sintaxis superficial para documentos estructurados, pero no impone restricciones semánticas en el significado de estos documentos.
- **XML Schema** es un lenguaje que se utiliza para restringir la estructura de los documentos XML, además de para ampliar XML con tipos de datos.
- **RDF** es un modelo de datos para objetos ("recursos") y relaciones entre ellos, proporcionando una semántica simple para éste. Este tipo de modelo de datos puede ser representado en una sintaxis XML.
- **RDF Schema** es un vocabulario utilizado para describir propiedades y clases de recursos RDF, con una semántica para la generalización y jerarquización tanto de propiedades como de clases.
- **OWL** añade más vocabulario para describir propiedades y clases: entre otros, relaciones entre clases (por ejemplo, desunión), cardinalidad (por ejemplo, "uno exacto"), igualdad, más tipos de propiedades, características de propiedades (por ejemplo, simetría), y clases enumeradas.

Jena

Como hemos dicho, Jena es un framework de desarrollo para construir aplicaciones basadas en tecnologías de la Web Semántica. Jena proporciona un entorno de programación para RDF, RDFS, OWL y SPARQL. Es un proyecto *open source*

2. ESTADO DEL ARTE

desarrollado por HP Labs Semantic Web Programme. Este marco de desarrollo incluye, entre otras cosas, lo siguiente:

Una API para RDF

Es más sencillo pensar en RDF en forma de grafos con nodos y arcos, donde los recursos identificados por un URI poseen unas propiedades que toman unos determinados valores, que en ocasiones serán literales y en otras serán otros recursos. El API Jena para RDF es un API implementado en **Java** que se puede utilizar para crear y manipular grafos RDF de este tipo. Jena define clases para representar grafos, recursos, propiedades y literales.

En Jena, un grafo se denomina modelo, y un modelo está representado como un conjunto de **sentencias o tripletes**, que no son otra cosa que cada uno de los arcos que componen el grafo. Cada triplete constata un hecho sobre un recurso, y se compone de; el sujeto (recurso del que parte el arco), el predicado (la propiedad que etiqueta el arco), y el objeto (literal o recurso apuntado por el arco).

Además de poder construir modelos mediante la creación de sentencias, Jena permite la lectura y escritura de RDF como tecnología **XML**, lo cual permite salvar un modelo RDF a un archivo para posteriormente cargarlo de nuevo en memoria. Una vez construido o cargado un modelo, Jena proporciona métodos para poder acceder a información del mismo a partir de las URIs de los recursos. Para la realización de búsquedas, lo más potente es utilizar las facilidades que da Jena para lanzar consultas con **SPARQL**.

Una API para Ontologías

Existen varios lenguajes para representar ontologías en la Web Semántica. A través de su API para ontologías, Jena pretende proporcionar un entorno de programación consistente para el desarrollo de ontologías, independientemente del lenguaje ontológico que se esté usando en un programa. Se trata de un API neutral, en el sentido de que los nombres de las clases no hacen mención al lenguaje subyacente. Para establecer las diferencias entre las diferentes representaciones, cada uno de los lenguajes de ontología tiene un perfil propio.

El perfil está unido a un “modelo ontológico”, que se trata de de una extensión del modelo base de Jena. Mientras que el modelo base permite el acceso a las relaciones

2. ESTADO DEL ARTE

sujeto-predicado-objeto en un RDF, el modelo ontológico añade soporte para los tipos de objetos esperados en una ontología: clases (en una jerarquía de clases), propiedades (en una jerarquía de propiedades), e individuos. Cuando se trabaja con una ontología en Jena, toda la información de estado queda codificada como tripletas RDF almacenadas en un modelo RDF. La API para ontologías no cambia la representación RDF de ontologías, sino que lo que hace es añadir un conjunto de clases y métodos convenientes que facilita la escritura de programas que manipulan las tripletas RDF.

Un motor de búsqueda para SPARQL: ARQ

ARQ es un motor de búsqueda incluido en Jena que soporta el lenguaje SPARQL RDF Query, el lenguaje de consulta diseñado por el W3C RDF Data Access Working Group. ARQ también puede ejecutar consultas RDQL (lenguaje precursor de SPARQL).

SPARQL

SPARQL (SPARQL Protocol and RDF Query Language) es un lenguaje de consulta a la vez que un protocolo, para acceder a RDF, diseñado por el W3C RDF Data Access Working Group. Como lenguaje de consulta que es, SPARQL está orientado a los datos en el sentido de que sólo consulta la información contenida en los modelos; no existe ningún tipo de inferencia. No hace más que describir lo que la aplicación quiere y recuperar dicha información.

Esta tecnología de consulta permite que las personas puedan centrarse en la información que quieren, sin tener en cuenta la tecnología subyacente o el formato utilizado para almacenar esos datos. Debido a que las consultas en el lenguaje SPARQL expresan objetivos a alto nivel, es fácil extenderlos a orígenes de datos inesperados, o incluso transferirlos a nuevas aplicaciones.

El lenguaje de recuperación SPARQL ha sido diseñado para un uso a escala de la Web, así permite hacer consultas sobre orígenes de datos distribuidos, independientemente del formato. A la hora de recuperar la información la creación de una sólo consulta a través de diferentes almacenes es mejor que múltiples consultas, además de tener un coste menor y de ofrecer unos resultados mejores.

3 GESTIÓN DEL PROYECTO

Este capítulo pretende mostrar cómo se ha organizado el proyecto y qué planificación temporal ha sido estimada para su realización. Veremos una visión global sobre estos dos aspectos relativos a la gestión del proyecto, con el propósito de realizar un acercamiento lo más aproximado posible al coste en términos de tiempo. La organización del proyecto se traducirá en identificar y definir brevemente las tareas a llevar a cabo, siendo coherentes con el ciclo de vida software que seleccionemos. Por su parte la planificación se traducirá en asignar el número de horas estimado para cada tarea, y por extensión deduciremos la duración total del proyecto.

3.1 Identificación de tareas

En este apartado vamos a identificar y describir las tareas o actividades que es necesario acometer para llevar a cabo el proyecto. Para ello contaremos con la definición de los objetivos del proyecto, y con la especificación de las fases en las que se agruparán de forma global las tareas. Estas fases saldrán de la identificación del ciclo de vida más adecuado para el desarrollo del software del proyecto.

3.1.1 Ciclo de vida

La definición del ciclo de vida facilitará la clasificación de tareas y el control y planificación temporal de las mismas, ya que establecerá la sucesión de fases que compondrán el proyecto. Un ciclo de vida lineal será apropiado teniendo en cuenta que el sitio Web no soportará una gran carga funcional, y que tan sólo se prevén situaciones de realimentación entre fases debido a posibles correcciones. En la siguiente figura identificamos las fases a través de su relación en ciclo lineal:

3. GESTIÓN DEL PROYECTO

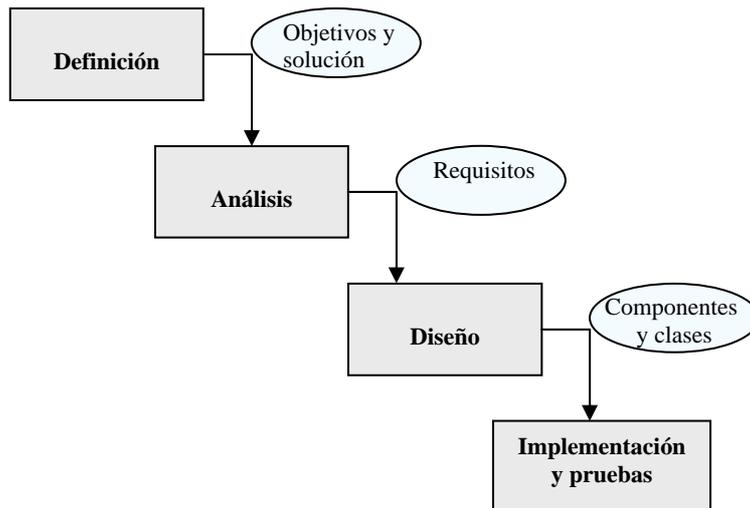


Figura 6: Ciclo de vida del proyecto

3.1.2 Especificación de tareas

Basándonos en las fases definidas en el apartado anterior trataremos ahora de establecer las tareas que caracterizarán cada una de ellas.

- **Fase de definición;** durante esta fase deberán identificarse los problemas a los que habremos de enfrentarnos, traduciéndolos en objetivos más concretos. Así mismo se definirá la solución a dichos problemas, traducida en las herramientas y tecnologías que dispondremos para sacar adelante los objetivos del proyecto. Tareas:
 - Planteamiento del problema; se detectarán cuáles son los problemas y objetivos generales a los que dar solución.
 - Definición de la solución; se identificarán las soluciones a los problemas planteados, y de entre todas se seleccionarán las que se consideren más convenientes.
- **Fase de análisis;** la fase de análisis cubrirá tanto lo que es el estudio del contexto, necesario para poder afrontar el proyecto con garantías, como los pasos a seguir para concretar en un listado de requisitos más específicos los objetivos generales del proyecto. Tareas:
 - Estudio del contexto; esta tarea se dividirá a su vez en función de las diferentes ramas de estudio que entren en juego a la hora de poner en práctica la solución definida en la fase de definición.

3. GESTIÓN DEL PROYECTO

- Extracción de requisitos; en esta tarea se aplicarán técnicas viables, a nuestro caso particular, de captura y definición de requisitos, y como resultado se definirá una lista de requisitos para el sitio Web.
- **Fase de diseño;** esta fase tendrá como punto de partida los requisitos establecidos en la fase de análisis. Considerando las soluciones identificadas durante la fase de definición, al final de esta fase tendremos definida la arquitectura de la aplicación, los diagramas de las clases y componentes software principales, y el diseño a nivel de bloques de las principales páginas Web que formarán la interfaz de usuario. Tareas:
 - Diseño de la interfaz Web; esta tarea comprenderá los pasos necesarios que darán lugar a la identificación de las principales páginas Web del sitio y a su diseño a nivel de bloques.
 - Diseño de la aplicación; esta tarea comprenderá las tareas de diseño software para deducir y definir la arquitectura y la división en clases y componentes del software del sitio Web.
- **Implementación y pruebas;** esta fase cubrirá las tareas de programación de la aplicación y de maquetación de la interfaz, por medio de las tecnologías dispuestas durante la fase de definición y atendiendo a las abstracciones resultantes de la fase de diseño. Tareas:
 - Maquetación de la interfaz; traducción del diseño a nivel de bloques de las páginas de la interfaz a código HTML.
 - Programación de la aplicación; programación de las clases y componentes identificados y definidos en la fase de diseño.
 - Pruebas; pruebas y depuración de la aplicación.

3.2 Especificación de recursos

Los recursos humanos de este proyecto, por la naturaleza del mismo, se limitarán a una sola persona, que será el autor de este trabajo, y puntualmente a las aportaciones de la tutora del proyecto.

3. GESTIÓN DEL PROYECTO

Por otra parte, debemos identificar los recursos materiales que serán necesarios para sacar adelante el proyecto. Al tratarse de un proyecto software, nos centraremos en definir las herramientas software que vamos a utilizar:

- Eclipse, como entorno de desarrollo (IDE) de la aplicación.
- Internet Explorer, como navegador para la recopilación de información y para la realización de pruebas de la aplicación.
- Firefox, como navegador para la realización de pruebas de la aplicación.
- Microsoft Word; como procesador de texto para la realización de la memoria del proyecto.
- Microsoft Project; para la construcción del diagrama de GANTT en la planificación del proyecto.
- Microsoft Power Point; para diseñar las diapositivas a mostrar durante la exposición del proyecto.
- Omondo EclipseUML; para la realización de los diagramas UML de la fase de diseño.
- Snapshot Maker; para realizar las capturas de pantalla a incluir en la memoria del proyecto.
- Tomcat; como servidor de aplicaciones sobre el que se desplegará la aplicación Web durante la fase de desarrollo y pruebas.
- UltraEdit; como herramienta para la edición de ficheros XML, RDF y OWL.

3.3 Planificación de tareas

Una vez identificadas las tareas en las que vamos a organizar el trabajo, el siguiente paso consistirá en establecer el tiempo estimado para cubrirlas. Para ello vamos a apoyarnos en diagramas de GANTT, en los que quedará fijado el número de horas laborales y un calendario laboral sobre el cuál desarrollar la planificación. Se realizarán estimaciones considerando jornadas laborales de 8 horas para una semana laboral de 20 horas.

3. GESTIÓN DEL PROYECTO

3.3.1 Planificación Inicial

En la página siguiente se muestra el diagrama de GANTT donde se refleja la planificación temporal inicial de este proyecto.

3. GESTIÓN DEL PROYECTO

3.3.2 Planificación Final

Incluimos en este punto el diagrama de GANTT que refleja el coste temporal real a la finalización del proyecto (ver página siguiente). Si se compara con el diagrama de GANTT correspondiente a la planificación inicial se pueden identificar fácilmente los puntos que han sufrido retrasos o adelantos.

4 PLANTEAMIENTO DEL PROBLEMA Y SOLUCIÓN

El uso de la Web como medio para difundir colecciones de patrones de diseño es un hecho cada vez más generalizado. En la actualidad uno puede dar con sitios Web para los más diversos catálogos. Por lo general, se remiten a una serie de páginas estáticas en las que se incrustan las descripciones de los patrones, y a través de las cuales puedes navegar de una forma más o menos sencilla ([AjaxPatterns], [Workflow Patterns]). Como las descripciones de los patrones normalmente vienen dadas en lenguaje natural, estos sitios se pueden ver como meras transcripciones del papel a código HTML. Además, lo común es encontrarlos con sitios que presentan una escasa preocupación por cumplir con los principios básicos de un buen diseño Web, basados en conceptos tales como la usabilidad, aunque como en todos los ámbitos existen honrosas excepciones ([Web Patterns], [Yahoo! DPL]).

A medida que el empleo de patrones de diseño se ha ido extendiendo, se ha ido avanzando en el estudio de sus posibles representaciones, buscando alternativas a la representación basada en plantillas de descripción y lenguaje natural. Las ontologías y su aplicación al mundo de la Web Semántica surge como una de las posibilidades más nuevas y ventajosas. Así se consideró al definirse el catálogo de patrones que sirve de base a este proyecto [Montero, 2005].

En el presente proyecto se plantea el desarrollo de un sitio Web que sirva como punto de acceso al contenido de dicho catálogo. Según lo expuesto antes, quedan planteados a su vez dos retos. Por un lado construir para el sitio Web una interfaz de usuario apropiada y de acuerdo a principios bien conocidos y aceptados por la comunidad de diseñadores Web. Por otro lado, conocer y emplear alguna de las herramientas existentes para la explotación de ontologías en el ámbito de la Web Semántica.

4.1 Definición del problema

Como ya hemos indicado, nuestro punto de partida es el repositorio semántico de patrones de diseño hipermedia realizado en [Montero, 2005]. Empleando OWL se definió un catálogo de patrones de diseño aplicados al dominio de la hipermedia y se estableció las relaciones entre los mismos. OWL [W3C, 2004b] es un lenguaje para la definición e instanciación de ontologías Web codificado en XML y construido sobre RDF [W3C, 2004a]. Como tecnología basada en XML que es, una representación con OWL por sí sola no es la forma más adecuada para explorar el dominio que representa. Necesita de otros soportes tecnológicos que se encarguen de facilitar estas tareas.

De aquí parte la idea de diseñar y construir un sitio Web que dé acceso a dicho repositorio de patrones de diseño. El uso de la Web como medio para difundir catálogos de patrones de diseño, incluidos aquellos orientados al universo hipermedia, es algo que se está dando cada vez con mayor frecuencia. Sin embargo, nuestro sitio Web tendrá el valor añadido de tratar con un repositorio semántico, lo cual representará una aportación atractiva y novedosa. Por tanto deberemos emplear alguna de las plataformas de Web Semántica existentes para el manejo de ontologías definidas con OWL.

Por otra parte, el diseño y construcción de un sitio Web no se debería reducir únicamente a la implementación de un conjunto de páginas Web enlazadas entre sí, sino que además en el proceso siempre se deberían tener presentes otros conceptos tan importantes como son la usabilidad, la interacción con el usuario o el rendimiento, conceptos que van más allá de lo que es la simple codificación de las páginas. Deberemos establecer por tanto de qué modo nos guiaremos para conseguir diseñar un sitio Web que resulte útil y usable.

Una buena forma de comenzar a afrontar el desarrollo de un sitio Web es considerando por separado la interfaz de usuario y el procesamiento de la información, y así se procederá en este proyecto. Entre ambas partes se debe establecer un contrato que defina qué es lo que la interfaz de usuario necesita que le proporcione la parte procesadora (a partir de ahora nos referiremos a ella como “**aplicación Web**”), y qué es lo que esta última necesita de la interfaz de usuario para proporcionárselo. Al dividir el desarrollo en dos partes, habrá que especificar cuáles son los problemas que competen de forma exclusiva a cada una de ellas, y cuáles son las soluciones que se aplicarán para resolverlos.

4.1.1 Interfaz de usuario

El diseño de una interfaz Web debe realizarse de manera que se obtenga un producto que resulte equilibrado, usable, y útil. Conseguirlo no es una tarea trivial, sino que por el contrario es un proceso complejo en el que no sólo hay que considerar aspectos de diseño gráfico. Los sitios Web pobremente diseñados frustran a los usuarios, minan la fidelidad de éstos, y hacen perder el tiempo a todas las personas implicadas en su desarrollo. Es muy probable que, entre otras cosas, los visitantes de un sitio Web lo abandonen si consideran que es demasiado esfuerzo navegar por el mismo, piensan que no abarca el producto o servicio que buscan, o tienen la sensación de que tarda mucho en cargarse. Conceptos como usabilidad, navegabilidad o interactividad, son elementos importantes a la hora de que un sitio Web triunfe en Internet, ya que son los elementos que más valora el usuario final [van Duyne *et al.*, 2002].

Conforme esta idea se ha ido imponiendo en el área del desarrollo Web, también se ha ido imponiendo la figura del diseñador Web. Un diseñador Web debe ser capaz de combinar las tareas de programación y diseño gráfico, con aquellos principios generales de diseño establecidos y consensuados por los expertos. Estos principios se tienden a plasmar en reglas de diseño que guían al diseñador Web para conseguir aplicaciones usables (véase [Nielsen, 2002] y [Tognazzini, 1998]).

Resumimos en los siguientes puntos los principales problemas a los que nos enfrentaremos a la hora de diseñar nuestra interfaz de usuario:

1. Buscar un diseño apropiado o adecuado al tipo de información a la que daremos acceso (patrones de diseño).
2. Proporcionar varios modos de navegación para explorar el repositorio de patrones.
3. Diseñar un sitio Web ligero. El tiempo que consumirá el sitio en procesar la información deberá ser compensado con un diseño ligero que mejore la experiencia del usuario.

En el apartado 4.2.1 veremos en qué medida los patrones de diseño se presentan como la solución a todo lo expuesto anteriormente, y propondremos una colección de patrones en particular como guía y referencia del proceso de diseño de la interfaz de usuario de nuestro sitio Web.

4.1.2 Aplicación Web

Si olvidamos por un momento todo lo que implica el desarrollo clásico de una aplicación Web, la principal cuestión que nos debemos plantear se centra en seleccionar la mejor forma de explotar la ontología de patrones de diseño hipermedia definida en [Montero, 2005]. Esta forma de representar un patrón de diseño hipermedia combina el conocimiento de dos dominios diferentes, el dominio que define cómo se describe un patrón, es decir, su formato; y el dominio implicado en qué se describe en el patrón, es decir, los componentes de un sistema hipermedia. Esto implica la existencia de tres representaciones ontológicas, cada una de las cuales está realizada en lenguaje **OWL** y que enumeramos a continuación:

- **Patrón;** los componentes de esta ontología representan el formato a seguir para describir un patrón.
- **Hipermedia;** representa el dominio sobre el cual se capturan los problemas y soluciones que especifican los patrones, es decir, esta representación está asociada al diseño de las aplicaciones hipermedia.
- **Patrón Hipermedia;** representa explícitamente cómo se han de combinar los elementos de cada una de las representaciones anteriores dando lugar al armazón subyacente a partir del cual los patrones de diseño hipermedia pueden ser formalizados.

Los patrones de diseño hipermedia que componen el catálogo vendrán dados como instancias individuales de esta representación ontológica. Cada una de estas instancias individuales vienen descritas en el lenguaje **RDF**. Los patrones del repositorio quedan descritos por los siguientes componentes :

- **Nombre:** nombra y enumera el patrón de manera inequívoca.
- **Aspecto de diseño;** Navegación, Estructura, Presentación, Interacción, Personalización o Seguridad.
- **Nivel de descripción;** Alto, Medio o Bajo.
- **Contexto:** introduce el contexto para el patrón, explicando cómo va a ayudar a completar ciertos patrones de mayor escala y hace referencia a aquellos patrones que deberían haber sido aplicados anteriormente.

4. PLANTEAMIENTO DEL PROBLEMA Y SOLUCIÓN

- **Problema:** describe en una o dos frases la esencia del problema.
- **Discusión:** describe los antecedentes empíricos del patrón, la evidencia para su validez y el rango de los diferentes modos en que el patrón se puede manifestar.
- **Solución:** describe el campo de las relaciones físicas y sociales requeridas para resolver el problema indicado, en el contexto indicado.
- **Diagrama:** muestra la solución en forma de diagrama, con etiquetas que indican sus principales componentes.
- **Relacionado con:** enlaza el patrón a aquellos patrones de menor escala que son necesarios para completar a este patrón, adornarlo o rellenarlo.

El Aspecto de Diseño y el Nivel de Descripción representan los dos componentes que categorizan y que permiten clasificar los patrones de diseño dentro del dominio de la hipermedia. Esto será importante a la hora de estructurar la navegación del sitio Web, por lo que influirá especialmente en lo que respecta al diseño de la interfaz.

Los componentes que implican un desarrollo textual más o menos extenso, como el Contexto o la Solución, no contienen el texto completo sino una serie de **Anotaciones** que sintetizan las ideas más significativas del mismo, ya sean en forma de secciones descriptivas de texto o en forma de referencias a otros patrones. Las descripciones textuales completas de los patrones vienen definidas en ficheros **XML** enlazados con los correspondientes RDF a partir de un identificador de patrón.

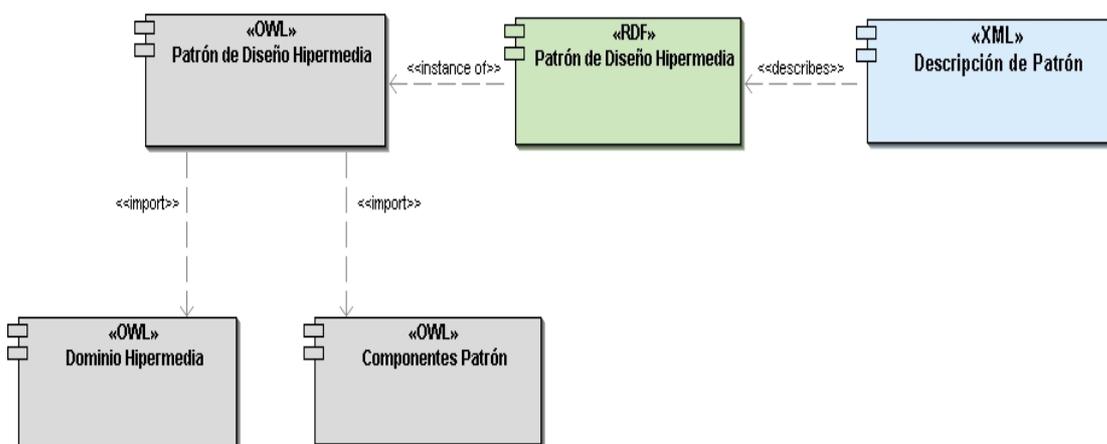


Figura 7: Modelo de representación del catálogo de patrones de [Montero, 2005]

4. PLANTEAMIENTO DEL PROBLEMA Y SOLUCIÓN

Identificada la información a la que se dará acceso y conocido el modelo de representación se plantean las siguientes cuestiones básicas para la construcción del sitio Web:

1. Cómo acceder a la información estructural de los patrones atacando los ficheros OWL que contienen la definición de la ontología.
2. Cómo acceder a la descripción de los patrones atacando los ficheros RDF y realizar consultas sobre los mismos.
3. Cómo acceder a las descripciones completas de los patrones contenidas en los XML y realizar consultas sobre los mismos.

4.2 Solución propuesta

De la misma forma que hemos diferenciado la definición del problema en lo que respecta al diseño de la interfaz de usuario y en lo que respecta al diseño e implementación de la aplicación Web, igualmente procederemos al exponer la solución que proponemos.

4.2.1 Interfaz de usuario

La solución que proponemos es acudir a la experiencia de diseñadores Web contrastados plasmada en forma de **patrones de diseño**. Los patrones de diseño están presentes en nuestros proyectos de forma cotidiana, por ejemplo cuando proponemos un diseño que contenga un "directorio tipo Yahoo!", o una "navegación por *tabs* como la de Amazon", nos estamos refiriendo a modelos de conocimiento probados que podemos utilizar como ayuda a nuestros problemas. Si bien es cierto que no todo puede solucionarse con patrones, el mismo Jakob Nielsen defiende que debemos producir y prestar atención a patrones de diseño para las convenciones más ampliamente utilizadas en el diseño Web [Nielsen, 2004]. Por su parte, Jared Spool sostiene la conveniencia del uso de patrones de diseño frente a plantillas y reglas de diseño [Spool, 2003].

El catálogo de patrones que hemos seleccionado para guiar nuestro diseño se corresponde con el recogido en la obra de referencia *The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience* [van Duyne *et al.*, 2002]. En él, los autores defienden el "diseño centrado en el cliente" como camino para incrementar el valor de los sitios Web y alcanzar un grado óptimo de

4. PLANTEAMIENTO DEL PROBLEMA Y SOLUCIÓN

satisfacción por parte de los “clientes” de los mismos. En base a sus experiencias, investigaciones y discusiones con otros expertos, han estructurado en tres partes el modo en que aplicar este particular enfoque de diseño Web. La tercera de las partes, en la cual centraremos nuestros esfuerzos de estudio y análisis, se corresponde con el catálogo de patrones de diseño, los cuales proveen un lenguaje común para articular un número indefinido de diseños Web.

En la fase de Análisis y fundamentalmente en la fase de Diseño veremos con más detalle de qué forma vamos a manipular este catálogo. Resumidamente, la idea será explorar el catálogo extrayendo a la vez los patrones que consideremos aplicables a nuestro diseño en particular. Con el conjunto resultante, generaremos un lenguaje propio que refleje las relaciones entre los patrones que lo componen, el cual utilizaremos posteriormente para conducir la aplicación de los mismos.

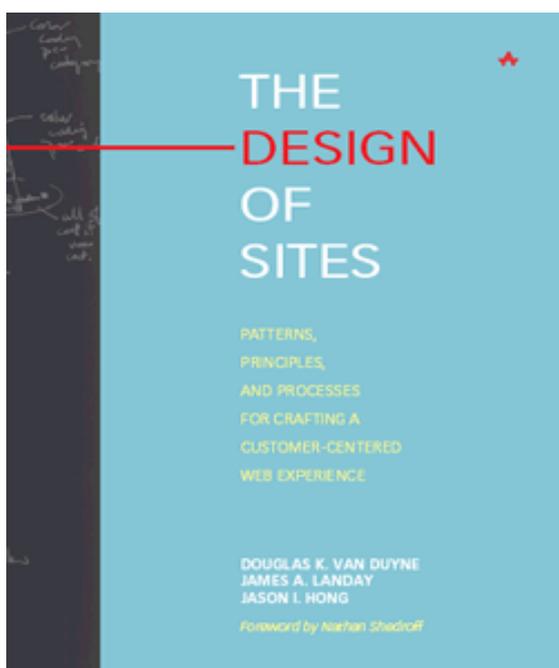


Figura 8: Portada de nuestro libro de referencia “*The Design of Sites*”

Sin embargo, para poder seleccionar los patrones del catálogo más apropiados necesitaremos trabajar sobre una base de requisitos detallada que nos ayude a concretar lo que esperamos del sitio Web. Estos requisitos los extraeremos de la evaluación de otros sitios Web de patrones de diseño. Mediante esta evaluación podremos identificar qué cosas incorporar, pero también lo que debemos evitar y aquellos aspectos susceptibles de mejorar.

4.2.2 Aplicación Web

El patrón arquitectural **MVC** (Modelo Vista Controlador) [Burbeck, 1992] se adapta perfectamente a la hora de plantear el diseño de la aplicación Web en conjunto. El patrón MVC está ampliamente extendido a la hora de desarrollar aplicaciones Web y existen un buen puñado de *frameworks* que lo implementan. Básicamente describe la separación entre los datos de una aplicación, la interfaz del usuario, y la lógica de control en tres componentes distintos. La idea es desacoplar el acceso a los datos y la lógica de negocio de la presentación de los datos y la interacción con el usuario, introduciendo un componente intermedio llamado controlador. De forma más detallada:

- **Modelo:** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
- **Vista:** Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- **Controlador:** Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista.

En nuestro caso el Modelo abarcará fundamentalmente todas las funciones de acceso y manipulación de la ontología OWL definida en [Montero, 2005]. Precisamente, el principal problema que hemos planteado a la hora de afrontar el desarrollo de nuestra aplicación Web se resumía en decidir el modo en que se explotará el acceso a los ficheros OWL que definen la ontología del repositorio de patrones, y el acceso y consulta de los ficheros RDF que representan las instancias concretas de los patrones. Necesitamos de librerías software que nos permitan realizar estas tareas de manipulación de ficheros OWL y RDF, y de algún lenguaje que nos permita definir consultas para realizar búsquedas en su contenido.

Para lo primero haremos uso del marco de desarrollo **Jena**. Se trata de un *framework open source* escrito en Java para la construcción de aplicaciones basadas en Web Semántica. Proporciona un entorno de programación para trabajar con RDF, RDFS, OWL y SPARQL. Por tanto, Jena nos permitirá cargar las ontologías escritas en

4. PLANTEAMIENTO DEL PROBLEMA Y SOLUCIÓN

OWL para poder trabajar sobre ellas. De igual forma podremos cargar las descripciones de los patrones escritas en RDF y acceder a su contenido.

A la hora de realizar búsquedas semántica en el repositorio nos bastaría con acceder secuencialmente por medio de Jena a cada uno de los ficheros RDF y seleccionar aquellos que cumplan con los criterios establecidos. Sin embargo, una opción más interesante puede ser la de utilizar el lenguaje de consulta **SPARQL**. Se trata de una recomendación para crear un lenguaje de consulta dentro de la Web Semántica. En palabras de Tim Berners-Lee, director del W3C, “pretender usar la Web Semántica sin SPARQL es como pretender usar una base de datos relacional sin SQL”. Jena soporta el parseo y ejecución de consultas escritas en SPARQL a través de un módulo llamado ARQ.

Pero como vimos al plantear el problema, no toda la información de un patrón se encuentra en su fichero RDF. El contenido textual de su descripción se almacena en ficheros XML a los que habrá que acceder. Para la realización de las búsquedas a partir de texto libre, atacaremos estos ficheros XML apoyándonos en el API de indexación y recuperación de información **Lucene** [Lucene]. A través de este API indexaremos las descripciones textuales de los patrones, almacenando esta información en un directorio de búsqueda sobre el que se lanzarán las consultas de patrones.

La elección de Jena como plataforma para la manipulación de la información semántica de nuestro sitio Web, implicará el uso de tecnología **Java** para la construcción del mismo. Esto implicará a su vez el uso de las especificaciones definidas por la plataforma **J2EE** en lo que respecta a aplicaciones Web, fundamentalmente la especificación **Servlets** y la especificación **JavaServer Pages**.

5 ANÁLISIS

El presente capítulo abarca la descripción de los pasos que conforman la fase de análisis de nuestro sitio Web. El resultado final esperado de esta fase será la obtención de un listado de requisitos que especificarán de forma más detallada lo que debe cumplir nuestro sitio Web, y que nos servirán como punto de partida para poder afrontar con garantías la posterior fase de diseño.

Hay que decidir el modo en que realizaremos esta extracción de requisitos, y se sabe que existen diferentes técnicas para llevarlo a cabo. Nosotros emplearemos una técnica de estudio que abordará dos fuentes de información de diferente naturaleza. Por un lado evaluaremos un conjunto representativo de repositorios Web de patrones de diseño, con el objetivo de extraer aspectos positivos que puedan incorporarse a nuestro propio diseño, pero también hacer una labor crítica para identificar aquellos aspectos que podamos mejorar o directamente evitar (véase el apartado 5.2).

Por otro lado, haremos una primera incursión en el catálogo de patrones recogidos en [van Duyne *et al.*, 2002], concretamente para realizar un análisis de los patrones enmarcados dentro del grupo A (véase el apartado 5.3). Cada patrón de este grupo define un marco de trabajo a muy alto nivel para un tipo o género de sitio Web determinado. Debemos decidir cuál o cuáles de estos géneros se ajustan más a nuestras necesidades, para extraer las propiedades y características generales que deberá tener nuestro sitio Web.

Finalizamos el capítulo mostrando el listado de requisitos resultante del análisis de ambas fuentes (véase el apartado 5.4).

5.1 Introducción al análisis

Hasta ahora sabemos que lo que se pretende, a muy grandes rasgos, es construir un sitio Web que dé acceso al repositorio semántico de patrones de diseño hipermedia definidos en [Montero, 2005]. En la Introducción al proyecto (capítulo 1), llegamos a dar un listado mínimo de objetivos generales que debía satisfacer el sitio Web, y que en este punto volvemos a recordar:

1. Dar acceso a la visualización de las descripciones de los patrones.
2. Posibilitar la realización de consultas o búsquedas en el repositorio.
3. Todo lo anterior debe ser soportado a través de una interfaz de usuario usable.

Sin embargo estos objetivos únicamente representan una visión general de lo que esperamos del sitio Web. Por sí solos no nos garantizan la consecución de un diseño efectivo, ya que partiríamos de ideas que aún presentan un nivel de abstracción demasiado alto. Debemos hilar un poco más fino y obtener un conjunto de requisitos que representen unos objetivos más específicos a partir de los cuales abordar con garantías la fase de diseño.

La captura de requisitos es la actividad mediante la cual se extraen las necesidades de un sistema, y existen diferentes técnicas para llevarlo a cabo. Una de las posibles técnicas a emplear es la basada en entrevistas, ya sea con entrevistas a través de cuestionarios de preguntas aplicadas al cliente, entrevistas individuales guiadas por el ingeniero software o entrevistas en grupos de trabajo. Otra técnica muy utilizada es la basada en casos de uso, la cual parte de cada una de las funciones del sistema y en base a éstas especifica los requisitos del mismo. Hay varios tipos más, pero nosotros emplearemos la técnica del estudio, la cual se basa en el análisis de documentación y en la evaluación de otros desarrollos similares que puedan ayudar a determinar qué incorporar al propio desarrollo y cómo hacerlo de forma adecuada.

Nuestro estudio reposará en dos procesos independientes que tratarán con fuentes de información distintas, cada uno de los cuales nos hará llegar a una serie de conclusiones que se traducirán posteriormente en forma de requisitos. Resumimos a continuación en qué consistirá cada uno de estos procesos:

Evaluación de repositorios Web de patrones. Un buen modo de decidir qué vamos a ofrecer a nuestros usuarios y de qué modo vamos a ofrecérselo, es observando qué ofrecen y cómo lo ofrecen otros sitios Web de características similares a lo que es nuestro propósito.

Hemos navegado por Internet buscando diferentes repositorios Web de patrones de diseño para hacer una selección representativa de los mismos, resultando un conjunto con los siguientes sitios Web: Ajax Patterns [AjaxPatterns], hcipatterns.org [hcipatterns], Hipermedia Design Patterns Repository [HDPR], Workflow Patterns [Workflow Patterns], y Yahoo! Design Pattern Library [Yahoo! DPL]. Cada uno de ellos será evaluado basándonos en una serie de criterios bien definidos.

Con lo que resulte de las cinco evaluaciones, extraeremos unas conclusiones acerca del estado actual de este tipo de Web. Con ello buscamos determinar por un lado cuáles son las mejores prácticas o aspectos más positivos de estos sitios que pueden o deben ser incorporados en nuestro desarrollo, y por otro lado identificar aquellos aspectos más negativos que son susceptibles de ser mejorados o directamente descartados.

Análisis de los patrones del grupo A. En capítulos anteriores, ya comentamos que el catálogo de patrones de [van Duyne *et al.*, 2002] se organiza o divide en una serie de grupos que representan diferentes niveles de detalle en el diseño. El primero de ellos es el denominado grupo de patrones A o “*Site Genres*”. En este grupo los autores han categorizado los sitios Web en tipos o géneros. Cada patrón incluido se identifica con uno de ellos, y define un marco de trabajo que servirá como punto de partida para el resto del diseño del sitio Web.

Por tanto, el propósito de este proceso será buscar la correspondencia del sitio Web que queremos construir con uno o varios de estos patrones, de los cuales extraeremos características o propiedades de carácter general que deberemos incluir o tener en cuenta a la hora de realizar nuestro diseño.

5.2 Evaluación de repositorios Web de patrones

Antes de tomar cualquier tipo de decisión de diseño acerca de los contenidos, de la organización o del aspecto que presentará nuestro sitio Web, consideramos conveniente conocer y analizar los contenidos, la organización y el aspecto que presentan los repositorios Web de patrones de diseño que podemos encontrar en Internet actualmente. La evaluación que podamos hacer de estos sitios es útil en la medida en que las conclusiones que se saquen del mismo puedan ayudar al posterior proceso de diseño de nuestro sitio Web, puesto que nos ayudarán a saber un poco más acerca de lo que queremos y cómo lo queremos, y por supuesto, acerca de lo que debemos evitar, descartar o de lo que es susceptible de mejorar. Los objetivos que buscamos conseguir con este paso se pueden concretar en:

- Extraer las funcionalidades de mayor interés.
- Realizar una síntesis de las mejores prácticas.
- Comprender la estructura que presentan este tipo de sitios.
- Identificar los puntos débiles y carencias.

A la hora de evaluar un sitio Web existen tantos aspectos en juego que es difícil encontrar una aproximación suficientemente metódica y convincente. Existen diferentes métodos de evaluación de sitios Web que varían en su rigurosidad, costes y conocimientos necesarios para llevarlos a cabo [Hom, 1998]. La estrategia de evaluación más aconsejable es combinar una evaluación heurística con un test de usuarios posterior, posiblemente las dos técnicas más eficientes, prácticas y asequibles. El test con usuarios es una prueba basada en la observación y análisis de cómo un grupo de usuarios reales utiliza el sitio Web, anotando los problemas con los que se encuentran. Es una prueba complementaria a la evaluación heurística, con el handicap de que es bastante costoso y poco viable para la gran mayoría de casos.

La **evaluación heurística** [Nielsen *et al.*, 1994] es un tipo de método de inspección, que tiene como ventaja la facilidad y rapidez con la que se puede llevar a cabo. Normalmente es llevada a cabo por un grupo reducido de evaluadores que, en base a su propia experiencia, fundamentándose en reconocidos principios de usabilidad o heurísticos, y apoyándose en guías elaboradas para tal fin, evalúan de forma

5. ANÁLISIS

independiente el sitio Web, contrastando finalmente los resultados con el resto de evaluadores. Diversos autores han propuesto diferentes conjuntos de heurísticos o principios de usabilidad para realizar las evaluaciones, pero en algunos casos pueden resultar demasiado generales por lo que normalmente se suelen disponer de sub-heurísticos más detallados.

Vamos a intentar aplicar una evaluación heurística en nuestro proceso de evaluación de repositorios Web de patrones. En este caso el equipo de evaluadores se reducirá a un solo miembro, lo cual se aleja de las recomendaciones hechas en [Nielsen *et al.*, 1994]. Además, debido a la proliferación de heurísticos hemos realizado una síntesis de entre todas las propuestas que hemos encontrado [Hassan Montero *et al.*, 2003] adaptándolas al caso concreto de repositorios Web de patrones de diseño, cuyo resumen queda especificado en la Tabla 2.

Criterio	Resumen
Búsqueda	Capacidad de realizar búsquedas de patrones.
Contenido	Cómo aborda cada sitio Web la descripción de los patrones y las relaciones entre los mismos.
Estructura	Organización lógica del sitio Web. Es decir, el modo en que se particiona en nodos y como están organizados estos nodos.
Navegabilidad	Facilidad con la que el usuario puede desplazarse por todas las páginas del sitio Web.
Nivel de colaboración del usuario	En qué modo permite el sitio Web la participación y colaboración de sus visitantes.
Página de inicio	Qué aspecto presentan las páginas de inicio de estos sitios, y en qué medida resultan equilibradas y atractivas al usuario.

Tabla 1: Criterios de evaluación utilizados en nuestro análisis de sitios Web

En la Tabla 3 se listan los sitios Web que hemos seleccionado para su evaluación. Consideraremos este pequeño conjunto como válido y representativo del conjunto total de repositorios Web de patrones de diseño existentes. El caso de `hcipatterns.org` es una excepción ya que no es un verdadero repositorio de patrones, sino que simplemente recopila referencias a información relacionada con patrones de diseño de interacción. Sin embargo, lo hemos incluido en el conjunto porque a priori puede ser

5. ANÁLISIS

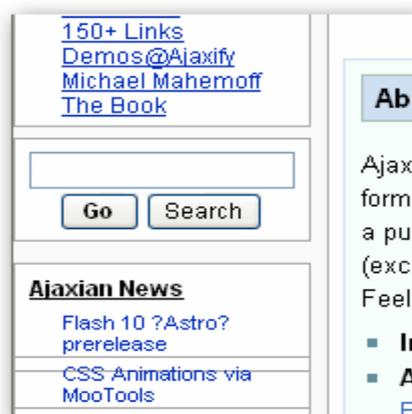
interesante ver cómo estructura dicha información. En los siguientes subapartados describimos en detalle la evaluación de cada sitio, y al final exponemos las conclusiones extraídas de todas ellas.

Nombre	URL	Contenido
Ajax Patterns	http://ajaxpatterns.org/wiki/index.php	Repositorio editable de todo lo relacionado con Ajax, incluido un completo catálogo de patrones de diseño aplicables al uso de Ajax.
hcipatterns.org	http://www.hcipatterns.org/tiki-index.php	No dispone de ningún repositorio. Reúne todo tipo de información relacionada con patrones de diseño de interacción.
Hypermedia Design Patterns Repository	http://www.designpattern.lu.unisi.ch	Colección de patrones de diseño Hipermedia creada por ACM-SIGWEB en colaboración con la <i>University of Italian Switzerland</i> .
Workflow Patterns	http://www.workflowpatterns.com/index.php	Colección de patrones de diseño de sistemas de control de flujo.
Yahoo! Design Pattern Library	http://developer.yahoo.com/ypatterns/	Librería de patrones de diseño elaborada por <i>Yahoo!</i> .

Tabla 2: Conjunto de repositorios Web de patrones de diseño a evaluar

5.2.1 Ajax Patterns

Búsqueda. Habilita una herramienta de búsqueda rápida en la barra de navegación izquierda del sitio, lo que permite que esté constantemente accesible y utilizable por el usuario. No busca sólo patrones, sino que posibilita buscar coincidencias del texto introducido por el usuario en el contenido de todas y cada una de las páginas del sitio. Por tanto, cada resultado hace referencia a una página

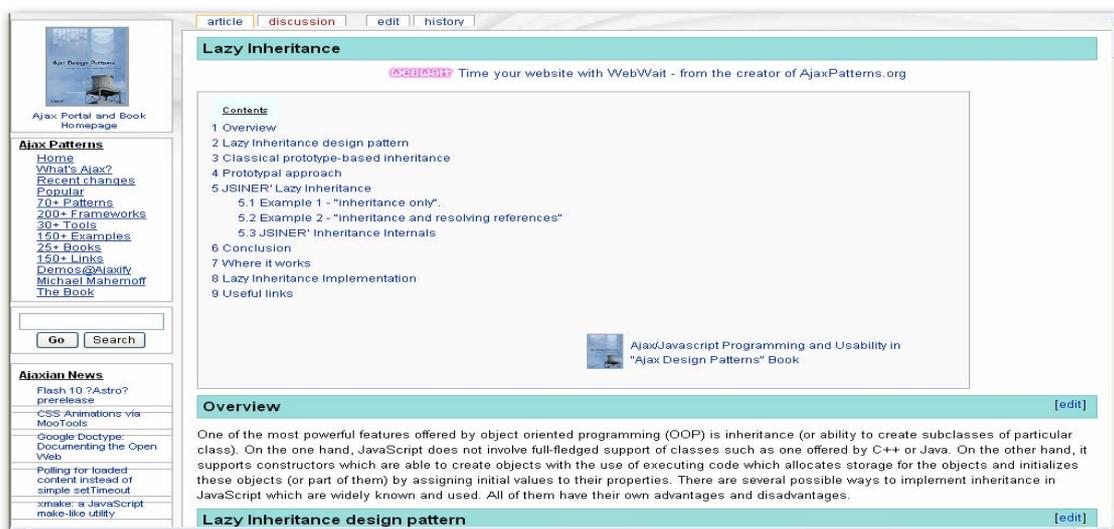


en la que aparece una o más veces el texto de búsqueda. Los resultados son mostrados en forma de lista, diferenciando entre páginas con coincidencias en su título (“*Article title matches*”) y páginas con coincidencias en el resto de su contenido (“*Page text matches*”). Cada resultado está formado por un enlace a la página que referencia, seguido de los extractos de línea donde han sido halladas las coincidencias. Los resultados están además paginados, con accesos a las diferentes páginas tanto en la

5. ANÁLISIS

cabecera como en el pie. Además al pie de página podemos ver una serie de *checks* que permiten acotar la búsqueda en función de las secciones del sitio donde uno quiera buscar.

Contenido. El nombre del patrón encabeza la descripción seguido de un índice destacado en un recuadro con enlaces a las diferentes secciones y subsecciones, lo que facilita la exploración del patrón. El nombre de cada sección destaca con respecto al texto asociado. Los accesos a los patrones relacionados y a los ejemplos de aplicación representan secciones de la descripción que se ubican en la parte final de la misma, por lo que requiere dar un salto que se podría evitar fácilmente.



Estructura. En general es una Web muy completa, ya que además de un completo repositorio de patrones de diseño para la técnica de desarrollo *Ajax*, presenta multitud de información relacionada con esta técnica y de muy diversos tipos, además de permitir la participación y la colaboración del usuario de forma muy efectiva. En lo que respecta concretamente al repositorio, éste queda presentado a través de una organización jerárquica de tres niveles; categorías, subcategorías y patrones. Sin embargo las diferentes categorías y subcategorías no poseen páginas independientes. Por el contrario, todo el árbol jerárquico queda recogido en una única página, sin tener correspondencia en ninguna de las herramientas de navegación.

Navegabilidad. Emplea dos herramientas para navegar con dos propósitos diferentes; una barra de navegación en la parte izquierda idéntica para todas las páginas que da acceso a las secciones principales del sitio, y una barra de navegación en la parte

5. ANÁLISIS

superior de la página que da acceso a las diferentes subsecciones que componen cada página del sitio (esto es, el propio cometido de la página, el blog de opinión asociado, la página de edición, y el historial de la misma). No guarda memoria de lo que se va visitando, por lo que el único medio de regresar a una página es empleando los botones del navegador. La navegación en el sitio se ve reforzada por la posibilidad de realizar búsquedas rápidas desde la barra de navegación izquierda, opción que ha sido tratada antes en la evaluación del criterio *Búsqueda*. En la parte inferior de todas las páginas del sitio se emplea una barra con enlaces externos de carácter publicitario a sitios que nada tienen que ver con la tecnología *Ajax*.

Nivel de colaboración del usuario. En gran medida el sitio es una *Wiki* que permite editar y realizar cambios sobre las diferentes secciones de la descripción de un patrón, de una forma rápida y sencilla. Si existe la posibilidad de añadir nuevos patrones, ésta no debe hallarse muy a la vista. También facilita un modo de aportar opiniones a través de una sección particular de cada página que denomina *discussion* y que se accede a través de un *tab* en la parte superior.

Página de inicio. La página resulta una maraña de enlaces que no invita a su exploración, compuesta de una sección de bienvenida, y continuada de una serie de grupos de enlaces encabezados por un título. Haciéndolo extensivo a todo el sitio, resulta en conjunto una interfaz poco amigable y saturada de enlaces.

The screenshot shows the AjaxPatterns.org homepage. At the top, there's a navigation bar with tabs for 'article', 'discussion', 'view source', and 'history'. Below this is a 'Main Page' section with a welcome message and a list of categories: Design Patterns, Websites, Design Tips, Frameworks+Libs, Links, Gotchas, Tools, and What's Ajax?. A prominent advertisement for 'Solar Energy Charity' is displayed in the center, with the text 'Helping Relieve Poverty Through The Provision of Solar Energy. Join Us!' and the website 'www.Solar-Aid.org'. To the left, there is a sidebar with a search bar and a list of links under 'Ajax Patterns', including 'Home', 'What's Ajax?', 'Recent changes', 'Popular', '70+ Patterns', '200+ Frameworks', '30+ Tools', '150+ Examples', '25+ Books', '150+ Links', 'Demos@Ajaxify', 'Michael Mahemoff', and 'The Book'. Below the search bar is a section for 'Ajaxian News' with various news items. On the right side, there are sections for 'All About Ajax' and 'Ajax Design Patterns', both containing lists of links and resources. The page also includes a footer with a search bar and a 'Go' button.

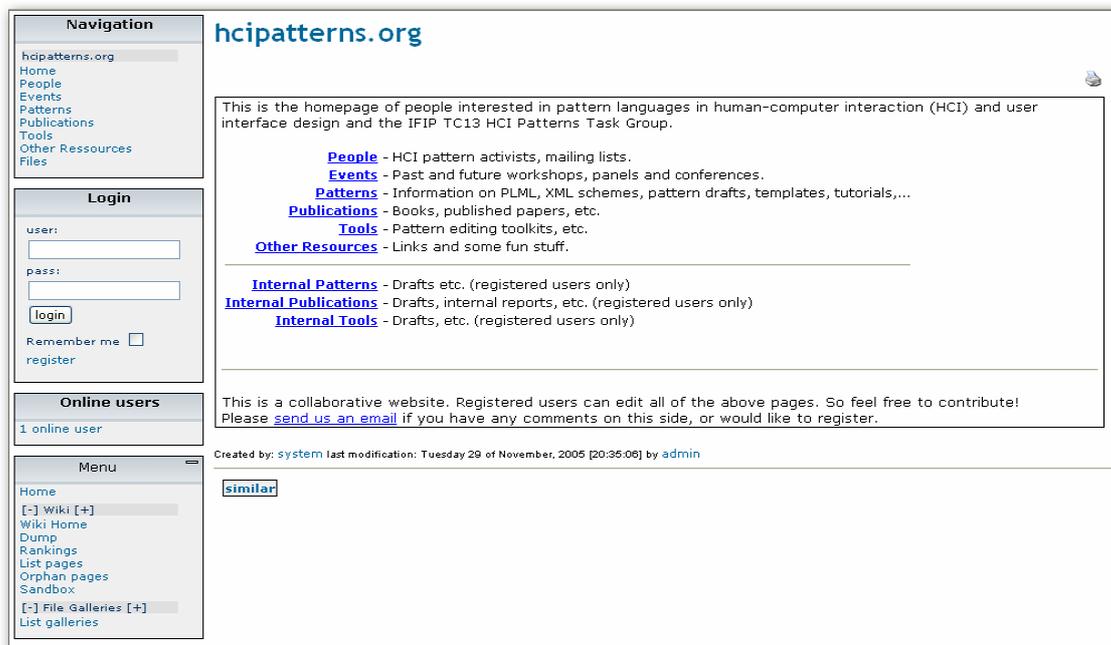
5. ANÁLISIS

5.2.2 hcipatterns.org

Búsqueda. No procede su evaluación, no es un verdadero repositorio de patrones.

Contenido. No procede su evaluación, no es un verdadero repositorio de patrones.

Estructura. Se trata de un sitio que versa sobre patrones de diseño, pero no es un repositorio como tal. Simplemente es un sitio de referencia de todo tipo de información externa relacionada con patrones de diseño de interacción en particular, y de diseño de interfaces de usuario en general. Esta información la diferencia en secciones independientes con acceso directo desde cualquier lugar del sitio; eventos, publicaciones, herramientas, archivos y documentos.



The screenshot shows the homepage of hcipatterns.org. On the left side, there is a navigation menu with categories: Navigation (listing Home, People, Events, Patterns, Publications, Tools, Other Ressources, Files), Login (with fields for user and pass, a login button, and a remember me checkbox), Online users (showing 1 online user), and Menu (listing Home, Wiki, Dump, Rankings, List pages, Orphan pages, Sandbox, File Galleries). The main content area features the site title 'hcipatterns.org' and a welcome message: 'This is the homepage of people interested in pattern languages in human-computer interaction (HCI) and user interface design and the IFIP TC13 HCI Patterns Task Group.' Below this, there are several links with descriptions: People (HCI pattern activists), Events (workshops and conferences), Patterns (PLML, XML schemes), Publications (books and papers), Tools (editing toolkits), and Other Resources (links and fun stuff). There are also links for Internal Patterns, Internal Publications, and Internal Tools, all noted as being for registered users only. At the bottom, there is a note about the collaborative nature of the site and a 'similar' link.

Navegabilidad. Con una barra de navegación en la parte izquierda de la página, cubre toda la necesidad de navegación del sitio debido a la sencillez de su estructura.

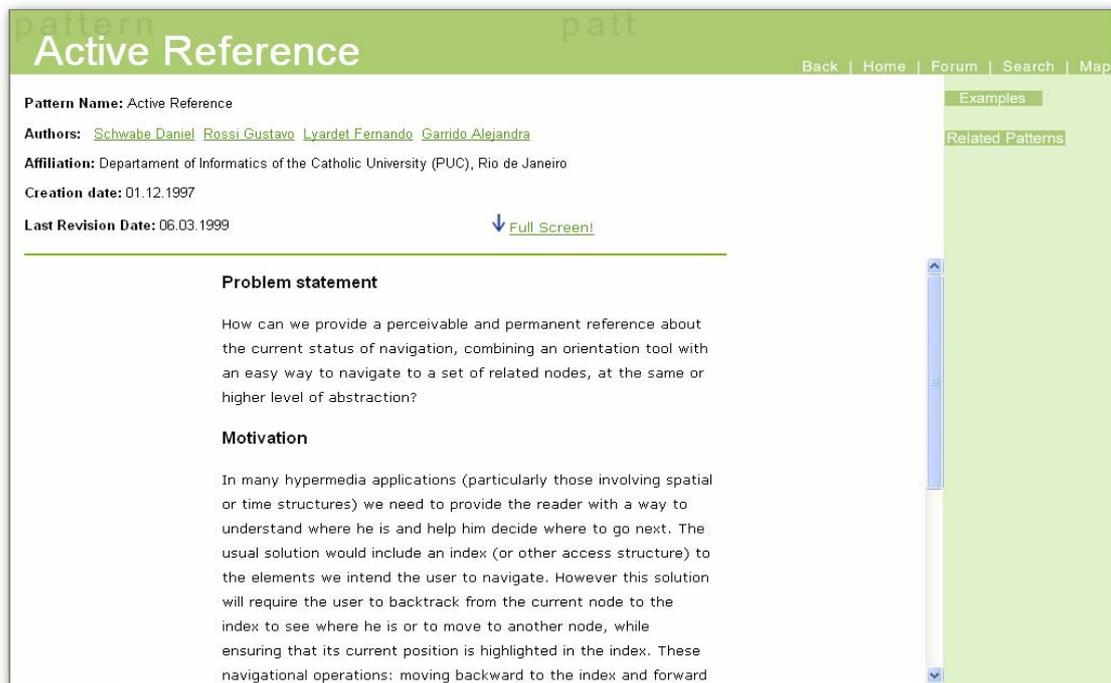
Nivel de colaboración del usuario. En realidad el sitio es una *Wiki*, en la que cualquier usuario registrado está en disposición de añadir nuevos enlaces a las páginas del sitio.

Página de inicio. La página de inicio no es más que una especie de copia parcial de la barra de navegación. El sitio no requiere mucho más.

5.2.3 Hypermedia Design Patterns Repository

Búsqueda. Dispone de una página para realizar búsquedas rápidas, pero aparentemente no está operativa, por lo que no podemos evaluar el modo en el que muestra los resultados.

Contenido. El nombre del patrón encabeza la página de descripción. En el cuerpo se distinguen dos partes: una parte fija con datos sobre la autoría del patrón (autores, fecha de creación, de revisión,...), y una parte *scrollable* en forma de marco donde se muestra el propio contenido del patrón. La ausencia de enlaces internos a las diferentes secciones de la descripción queda solventada en parte por el uso de un marco para este contenido, que te permite desplazarte verticalmente por el mismo sin perder de vista las herramientas de navegación. En la parte derecha hay una barra lateral donde se ubican accesos a patrones relacionados y ejemplos de sitios Web donde se ha utilizado el patrón. Resulta muy cómodo el disponer de estos accesos ya que agilizan la navegación entre patrones. A destacar la presencia de un acceso a la versión imprimible de la descripción.



Estructura. El repositorio presenta por un lado una organización jerárquica muy sencilla de dos niveles, el primero se corresponde a tres grandes categorías (“interface/layout”, “structure/navigation” y “content oriented”) y el segundo a los propios patrones. Por otro lado, además de esta organización jerárquica existe la

5. ANÁLISIS

posibilidad de acceder de una sola vez a la totalidad de los patrones del repositorio a través de un listado de los mismos en orden alfabético. Aunque se asimila rápidamente, también es cierto que desde la página de inicio no hay ninguna pista acerca de cómo está organizado el repositorio. El repositorio de patrones se complementa con una sección de enlaces externos.

Navegabilidad. Posiblemente es el que más opciones de navegación ofrece de los cinco sitios. También es el único sitio de los cinco que no emplea una barra de navegación en la parte izquierda de la página. Emplea un menú en la parte derecha que varía o desaparece en función de la página en la que nos encontremos, lo cual llega a confundir en ciertos momentos. De hecho se apoya en un menú en la parte inferior para intentar suplir las funciones que no cubre el menú de la derecha, ya sea por ausencia o por haber variado. Hay otra barra de navegación en la parte superior derecha de las páginas que ofrece otros accesos complementarios (a la página de inicio, a la página de búsqueda o al mapa del sitio, entre otros). La posibilidad de dar marcha atrás durante la exploración del sitio la basa en el juego de variaciones que realiza con las diferentes barras de navegación, en lugar de ofrecer herramientas específicas para este cometido (botón “Atrás”, *breadcrumbs*).

Patterns Classification Home | Forum | Search | Map

What is a Hypermedia Design Pattern?

"A **Pattern**", in the original definition of architect Christopher Alexander, "describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use that solution a million of times over..."

Although since many years used in software engineering design, specific patterns for hypermedia applications are being studied now only.

Design Patterns have been classified in **three categories**:
Patterns concerning Interface and Lay-out aspects; Patterns involving Structure of information and Navigation dynamics; Patterns more Content Oriented, i.e. Patterns that pay more attention to the message strategy and, in general, to the Content of the application.

Interface/Lay-out
Structure/Navigation
Content Oriented
Pattern Applications
Browse Patterns

HYPERMEDIA
PATTERN
REPOSITORY

Patterns | Authors Index | Papers | Submit Pattern

5. ANÁLISIS

Nivel de colaboración del usuario. Si se desea, se puede enviar un correo electrónico al administrador de la Web con descripciones completas de nuevos patrones siempre y cuando el envío se ajuste a un determinado formato. Por otro lado existe una sección de foros de conversación con una serie de temas propuestos, aunque en todos ellos apenas ha habido intervenciones, lo que refleja el estado de abandono en que se encuentra esta sección. También te dan la posibilidad de unir tu dirección de correo en una *mailing-list*.

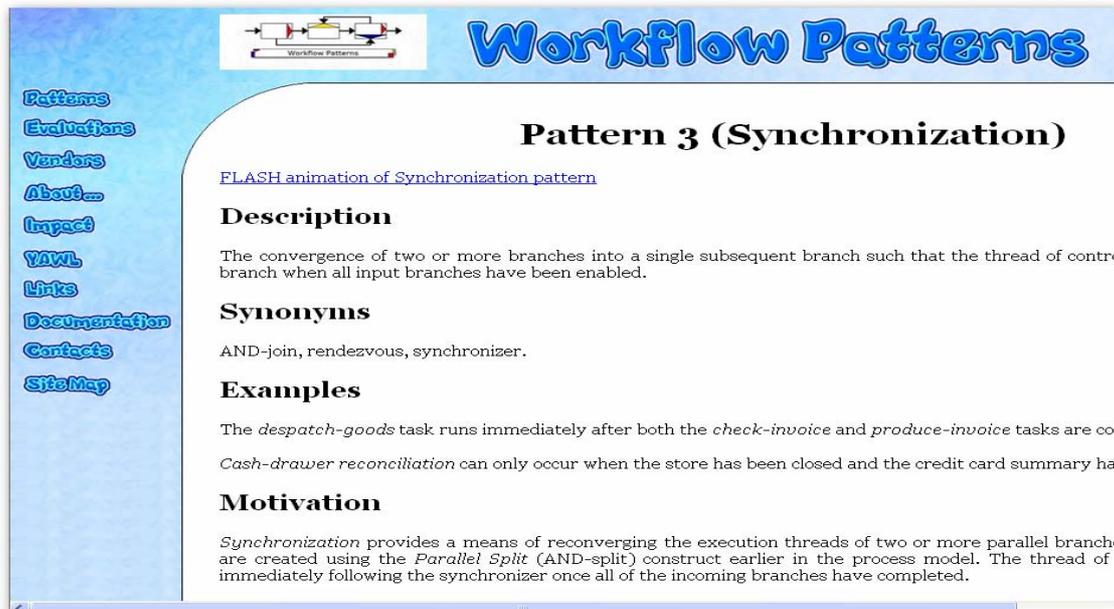


Página de inicio. La presentación es muy adecuada y atractiva visualmente, al igual que en el resto del sitio. Sin embargo la página de inicio no está muy aprovechada, tan sólo hay una pequeña sección de bienvenida, así como alguna información complementaria como la última fecha de revisión o la dirección de correo electrónico del administrador.

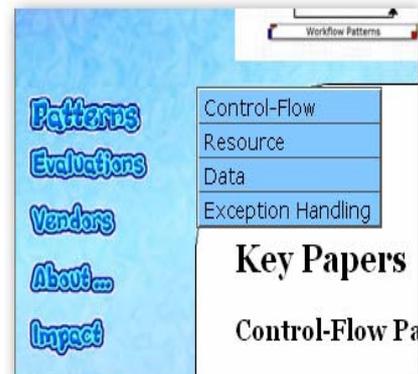
5.2.4 Workflow Patterns

Búsqueda. No posibilita la búsqueda de patrones.

Contenido. La descripción de los patrones se encabeza por el número de patrón seguido del nombre del patrón entre paréntesis. En cualquier caso el número de patrón mostrado va en función de la categoría, por lo que existen patrones de diferente categoría con el mismo número de patrón, lo cual puede resultar confuso si el usuario se guía por el número y no por el nombre. El nombre de cada sección destaca con respecto al texto asociado. La sección de patrones relacionados forma parte de la descripción y se ubica al principio de la misma, lo cual evita tener que ir página abajo para tener acceso. La única manera de leer el patrón es desplazando la barra de *scroll* verticalmente, lo que resulta tedioso en páginas largas. No hay uniformidad en cuanto a la resolución de pantalla mínima necesaria para no realizar *scrolling* horizontal en la lectura del patrón.



Estructura. El sitio presenta una sencilla organización jerárquica de tres niveles. El primer nivel lo conforman cuatro grandes categorías, las cuales se dividen en subcategorías que representan el segundo nivel, y por último tenemos los propios patrones como tercer nivel. A pesar de su sencillez, la estructura no resulta evidente puesto que los accesos a las subcategorías y a los patrones requiere obligatoriamente el recorrido completo de la página dedicada a la categoría correspondiente, a lo cual se suma que las subcategorías no poseen página propia. Por otro lado, el sitio dispone de un gran número de secciones complementarias al repositorio que suponen uno de los puntos fuertes del sitio, como son la sección de descargas de documentación, la sección de enlaces a sitios externos, la sección de referencias bibliográficas, etc.



Navegabilidad. La capacidad de navegación por el sitio se reduce a un menú en la parte izquierda de la página que da acceso a las principales secciones, y a una copia de este menú en la parte inferior de la página. Ambos aparecen en todas las páginas del menú sin ningún tipo de variación. En lo que respecta a la navegación específica por el repositorio de patrones, el menú sólo da acceso a las páginas de categorías principales, y es en el contenido de éstas donde reside el único modo de acceder a las descripciones de

5. ANÁLISIS

los patrones. La única posibilidad de deshacer el camino recorrido durante la exploración del sitio es a través de las herramientas del navegador.

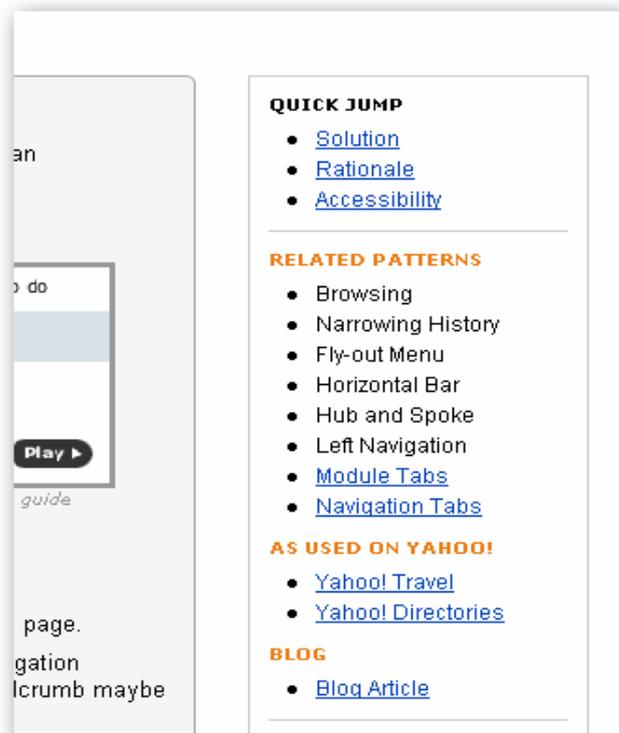
Nivel de colaboración del usuario. No hay foros de discusión, ni blogs, ni ningún otro mecanismo de participación. Tampoco da la posibilidad de colaborar con el contenido del repositorio.

Página de inicio. No es en absoluto una página atractiva visualmente, lo cual es extensivo al resto del sitio. Se limita a una sección de bienvenida que la ocupa por completo.

5.2.5 Yahoo! Design Pattern Library

Búsqueda. No posibilita la búsqueda de patrones.

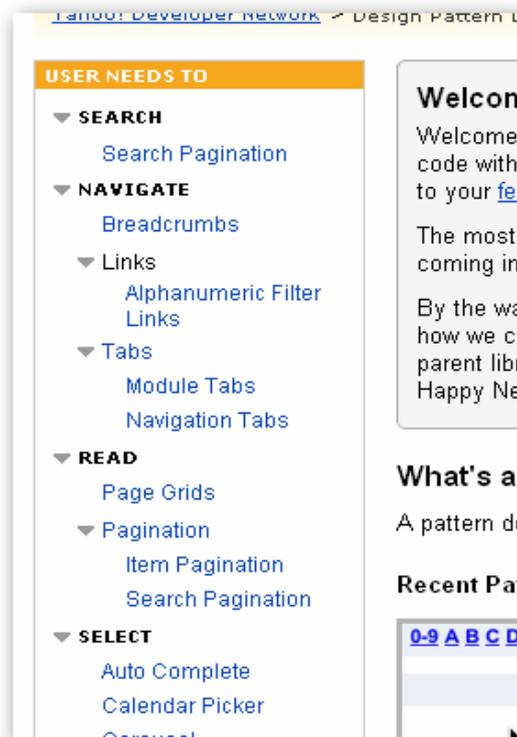
Contenido. Las descripciones de los patrones se presentan con una diferenciación visual entre las secciones que forman parte del planteamiento del problema y las secciones que forman parte de la solución. El nombre del patrón encabeza su descripción, y el nombre de cada sección destaca con respecto al texto asociado. En estas páginas desaparece el menú de la izquierda y se muestra una barra en la parte derecha con enlaces estrechamente relacionados con el patrón visualizado; blog asociado al patrón, patrones relacionados y enlaces directos a las distintas secciones de la descripción. Resulta muy cómodo el disponer de estos accesos ya que agilizan la navegación entre patrones. También hay un enlace externo al sitio de *Yahoo! Movies*, lo cual de alguna manera rompe de manera desafortunada con el carácter informativo del sitio.



5. ANÁLISIS

Estructura. La estructura del sitio es fácil de comprender. Presenta una organización jerárquica de tres niveles; un primer nivel correspondiente a las categorías de patrones (“búsqueda”, “navegación”, “interacción”,...), un segundo nivel correspondiente a subcategorías de patrones (por ejemplo, para la categoría “navegación” tenemos las subcategorías “links” y “tabs”) o a patrones sin enmarcar en ninguna subcategoría, y un tercer nivel correspondiente a patrones incluidos en subcategorías. Además es un repositorio de patrones puro, es decir, no hay otra cosa en el sitio más que patrones, lo cual simplifica la estructura y organización que pueda presentar.

Navegabilidad. La navegación por el repositorio se basa principalmente en un menú en la parte izquierda donde se despliegan los enlaces a todos los patrones agrupados por categorías y subcategorías, las cuales también se muestran como enlaces que apuntan a su propia página y al lado de las cuales hay un icono para plegar/desplegar el árbol particular de cada una. Según se vayan incluyendo patrones el menú se irá alargando, lo que puede resultar contraproducente a la hora de encontrar un patrón si están todas las categorías desplegadas. En las páginas de descripción de patrones, desaparece el menú izquierda, de modo que sólo se puede navegar a través de los *breadcrumbs*. Los accesos a los elementos relacionados con un patrón están muy bien cubiertos a través de la barra lateral derecha de la página, la cual agiliza el proceso.



Nivel de colaboración del usuario. El repositorio cuenta con un blog de opinión por cada patrón para habilitar los comentarios que puedan y quieran hacer los usuarios, y se accede desde la página de descripción del patrón correspondiente. Sin embargo estos blogs emplean una infraestructura externa al sitio, que al tener un aspecto distinto a éste, transmite una cierta sensación de impersonalidad. No da la posibilidad de colaborar con el contenido del repositorio.

5. ANÁLISIS

Página de inicio. En general es una más que efectiva página de inicio, con un estilo muy acorde a lo que se puede esperar de un sitio de esta naturaleza. En general, todas las secciones del sitio presentan un buen equilibrio entre funcionalidad y aspecto visual. En la página de inicio se distinguen tres módulos de contenido. El primero es un texto de bienvenida que encabeza la página. El segundo te enlaza a una descripción formal de patrón y al proceso de definición de patrones seguido por el equipo de desarrolladores de *Yahoo!*. Es una buena manera para hacer entrar en materia a los nuevos usuarios. Por último, una sección con un listado de los últimos patrones publicados. Es una extraordinaria opción para los habituales del sitio, para que tengan una referencia rápida de lo que ha variado el repositorio desde que accedieron la última vez.

The screenshot shows the 'Design Pattern Library' page on the Yahoo! Developer Network. The page has a yellow header with the 'YAHOO! DEVELOPER NETWORK' logo and the title 'Design Pattern Library'. Below the header is a navigation breadcrumb: 'Yahoo! Developer Network > Design Pattern Library'. On the left side, there is a 'USER NEEDS TO' sidebar with a tree view containing categories like SEARCH, NAVIGATE, READ, SELECT, and INTERACT, each with sub-items. The main content area is titled 'Welcome' and contains a paragraph of text, a link to 'feedback', and a section titled 'What's a Pattern?' with a definition and a 'more...' link. Below this is a 'Recent Patterns' section with three items: 'Alphanumeric Filter Links' (with a screenshot of a search bar), 'Animate Transition' (with a screenshot of a stock market quote), and 'Calendar Picker' (with a screenshot of a calendar). Each item includes a brief description of the user need or designer requirement.

5.2.6 Conclusiones

Siempre resulta de utilidad la posibilidad de realizar búsquedas o consultas en los contenidos de cualquier sitio Web. De esa idea partimos a la hora de incluir como criterio de evaluación aquel que etiquetamos como “Búsqueda”. Con ello pretendíamos observar de qué modo se habilitaban las herramientas de búsqueda de patrones en los distintos repositorios, y de qué modo se mostraban los resultados correspondientes. Sorprendentemente, hemos podido observar que no es una funcionalidad tenida en cuenta en este tipo de sitios, ya que del conjunto de sitios evaluados tan sólo era cubierta por **Ajax Patterns**. No podemos extraer por tanto comportamientos comunes al respecto, ni conclusiones en lo que respecta fundamentalmente a la

5. ANÁLISIS

presentación de los resultados de búsqueda. Nos conformaremos con destacar la ventaja de disponer en cualquier página de un módulo de búsqueda rápida al estilo de **Ajax Patterns**, para hacer uso de él en cualquier momento.

Sí encontramos un patrón común de comportamiento en todos los sitios en el modo en que organizan las páginas de sus catálogos de patrones. Todos acuden a una organización jerárquica basada en categorías de patrones con una profundidad máxima de tres niveles. Aunque esta sea la forma más natural de organizar información de este tipo, siempre será un valor añadido el hecho de estructurarla considerando otros puntos de vista con el fin de dar mayor flexibilidad al sitio, como sucede con el listado en orden alfabético del catálogo incluido en **Hypermedia Design Patterns Repository**.

Otra norma casi generalizada es la de complementar la información que representan los propios repositorios de patrones, con referencias a otras fuentes de información relacionadas. Parece coherente que una persona que esté interesada en un determinado catálogo de patrones de diseño accesible desde un determinado sitio Web, agradezca y considere positivamente el hecho de tener acceso a esta información complementaria sin necesidad de salir del sitio. Estas referencias son presentadas normalmente en forma de secciones independientes con enlaces a sitios Web externos y en forma de secciones de descargas de documentos. **Ajax Patterns** y **hcipatterns.org** disponen de varias secciones bien categorizadas de enlaces a información relacionada.

Uno de los criterios de evaluación que hemos definido es el de comprobar el nivel de participación y colaboración que permite cada sitio a través de las herramientas que facilita para ello. Podemos decir que en general sí incorporan mecanismos de participación, pero también que tienen poco éxito si atendemos a la escasez de intervenciones en la mayoría de ellos. Durante el análisis de los patrones del grupo A (véase el apartado 5.3) veremos qué medidas se pueden tomar para fomentar la participación. Hemos distinguido concretamente dos enfoques para recoger las intervenciones de los usuarios: blogs de opinión individuales a un patrón y foros de discusión con propuestas de temas más generales. Nos gusta esta división por ser complementaria, y encontramos en **Yahoo! Design Pattern Library** un ejemplo de sitio donde conviven ambos mecanismos. En contra de este último extraemos el hecho de que emplea infraestructuras externas al sitio, lo cual entendemos que da una

5. ANÁLISIS

sensación más impersonal y por tanto menos cercana al usuario, y que por tanto se deben utilizar infraestructuras personalizadas e internas al sitio (como por ejemplo en **Ajax Patterns**).

En lo que respecta a la capacidad de colaboración del usuario en los contenidos, a pesar de ser una opción muy válida hacer algo en forma de Wiki de forma similar a **AjaxPatterns**, preferimos para nuestro caso emplear formularios de entrada de datos para usuarios registrados. Sin embargo, la opción de facilitar un modo de enviar sugerencias de nuevos patrones al modo de **Hypermedia Design Patterns Repository** para usuarios que no estén registrados parece una buena práctica para recopilar conocimiento potencialmente válido de otras personas.

Otra de las conclusiones que extraemos de la evaluación, es que el hecho de habilitar muchas herramientas de navegación no es sinónimo de una mejor navegabilidad, tal y como hemos observado en **Hypermedia Design Patterns Repository**. Obviamente, tampoco es de desear lo contrario, como ocurre en el caso de **Workflow Patterns**. Lo ideal sería encontrar un equilibrio entre los contenidos y las variantes de navegación con las que trabajar, empleando tan sólo las justas y necesarias en función del volumen de información del sitio y la naturaleza de esta información. El uso de una barra de navegación en la parte izquierda de la página es la práctica más extendida para acceder a las secciones principales del sitio.

La disposición de un índice encabezando la descripción de cada patrón de diseño con enlaces internos a las diferentes secciones de la misma, tal y como encontramos en **Ajax Patterns** y en **Yahoo! Design Pattern Library**, es una buena opción para mejorar la navegabilidad interna de estas páginas. También parece una buena práctica para agilizar la exploración del sitio, el uso de un espacio de la página para alojar accesos a elementos relacionados con un patrón, tales como patrones relacionados o ejemplos de aplicación, como ocurre en **Hypermedia Design Patterns Repository** y **Yahoo! Design Pattern Library**. Una norma común es la de encabezar las páginas de descripción de patrones con el nombre del patrón correspondiente.

A priori, el fin último de un usuario de un repositorio Web de patrones es el de acceder a las descripciones de los mismos. De hecho es muy lógico que quieran acceder a una versión imprimible de las mismas. Sin embargo, tan sólo en el caso de

5. ANÁLISIS

Hypermedia Design Patterns Repository hemos encontrado la posibilidad de visualizar la descripción de los patrones aislada del resto de componentes de la página.

En cuanto a las páginas de inicio nos ha quedado la sensación de que en general no están bien aprovechadas, ya sea por escasez de contenidos (**Workflow Patterns**) o por exceso de información (**Ajax Patterns**). Tan sólo en **Yahoo! Design Pattern Library** hemos encontrado una página de inicio atractiva y efectiva, de la que destacamos la inclusión de los últimos patrones publicados en el repositorio como parte de la misma.

	Búsqueda	Contenido	Estructura	Navegabilidad	Nivel de colaboración del usuario	Página de inicio
Ajax Patterns	✓	✓	✓	✗	✓	✗
hcipatterns.org	--	--	✓	✓	✓	✓
Hypermedia Design Patterns Repository	✗	✓	✓	✗	✗	✗
Workflow Patterns	✗	✗	✗	✗	✗	✗
Yahoo! Design Pattern Library	✗	✓	✓	✓	✗	✓

Tabla 3: Resumen del análisis de sitios Web

5.3 Análisis de los patrones del grupo A

5.3.1 Introducción

El primer grupo de patrones que nos encontramos dentro del catálogo de patrones de [van Duyne *et al.*, 2002] es el denominado grupo A o “*Site Genres*”. En él los autores categorizan los sitios Web en tipos o géneros, de forma que cada género posee sus propios contenidos, necesidades y audiencia. Básicamente este grupo proporciona el marco de trabajo necesario para construir varios tipos de sitios diferentes. Los patrones aquí incluidos se mueven dentro de un nivel de abstracción alto, describiendo

5. ANÁLISIS

características y propiedades de carácter general que deben presentar determinados tipos de sitios Web. De estos patrones parten muchas referencias a otros patrones con un mayor nivel de detalle que contendrán las pautas para diseñar e implementar las ideas presentadas en los primeros, así como referencias a otros patrones relacionados.

La exploración de este grupo de patrones nos ayudará a determinar y clarificar el alcance del proyecto y los objetivos y necesidades del mismo, ya que el identificar nuestro sitio Web con uno o varios de los géneros de sitios Web propuestos nos permitirá centrar el diseño hacia lo que cabe esperar de un tipo de sitio de características similares a las que marcan nuestros propósitos u objetivos generales.

En la introducción al análisis (véase el apartado 5.1), recordábamos en una serie de puntos los objetivos generales que debe cumplir nuestro sitio Web. Partiendo de estos objetivos podemos enfrentarnos a la búsqueda del patrón (o patrones) de diseño de los enmarcados dentro del grupo A cuyas premisas más se ajusten a los mismos. No es nuestra intención explicar y desglosar aquí los once patrones que componen este grupo, aunque sí puede ser procedente resumir el propósito de cada uno de ellos con el fin de que queden enumerados los diferentes géneros de sitios Web que identifican los autores del libro:

Patrón A1 - *Personal E-Commerce* → Marca las pautas para construir un sitio de compras online fácil de usar, transparente y atractivo al comprador potencial.

Patrón A2 - *News Mosaic* → Define el marco de trabajo para crear portales Web de noticias que resulten útiles y que llamen la atención de los usuarios.

Patrón A3 - *Community Conference* → Ayuda a conseguir una comunidad virtual próspera, dinámica y de confianza, promocionando una gestión responsable de la misma.

Patrón A4 - *Self-Service Government* → Describe cómo construir un entorno Web donde los servicios gubernamentales resulten más accesibles, más cercanos a las personas y menos burocráticos.

Patrón A5 - *Nonprofits as Networks of Help* → Define el marco de trabajo para crear sitios de organizaciones no lucrativas que aprovechen al máximo los beneficios de la Web.

5. ANÁLISIS

Patrón A6 - *Grassroots Information Sites* → Ayuda a construir sitios Web de referencia informativa acerca de un tema determinado que sean capaces de dar respuestas efectivas a las inquietudes del usuario.

Patrón A7 - *Valuable Company Sites* → Describe cómo conseguir un portal Web empresarial que distribuya adecuada y proporcionalmente las necesidades de las diferentes audiencias del mismo.

Patrón A8 - *Educational Forums* → Describe las claves para lograr foros educacionales que consigan reunir de forma armónica a las diferentes partes implicadas.

Patrón A9 - *Stimulating Arts and Entertainment* → Define el marco de trabajo de cualquier sitio especializado en actividades de entretenimiento y disfrute, para hacerlo más agradable y atractivo.

Patrón A10 - *Web Apps that Work* → Muestra qué debe haber detrás de una aplicación Web, partiendo del hecho de que su uso difiere de la forma en que se hace con las aplicaciones de escritorio.

Patrón A11 - *Enabling Intranets* → Ayuda a construir intranets que permitan obtener una mayor productividad.

Leídos y analizados todos ellos, llegamos a la conclusión de que el género de sitio Web que más se acerca a lo que planteamos en nuestros objetivos generales es el que se denomina como *Grassroots Information Sites*. A pesar de que no es nuestra intención que el sitio Web se convierta en una guía especializada de enlaces a información externa, sí es cierto que existe un tema único o tópico sobre el que girará todo el contenido como son los Patrones de Diseño Hipermedia, y que se proporcionarán descripciones detalladas de los mismos que pueden ser perfectamente complementadas con aportaciones externas, ya sean en forma de documentos o en forma de *links* externos.

A continuación, entramos más en detalle en este patrón para poner de relieve los consejos y planteamientos que exponen e ir extrayendo de esta manera las conclusiones que consideremos oportunas en relación a nuestro sitio Web, y que a modo de diferenciación del resto de los contenidos destacaremos en cuadros de texto.

5. ANÁLISIS

5.3.2 Patrón *Grassroots Information Sites* (A6)

Este patrón pretende formalizar la base de todos los sitios de referencia de información, en los que existe un profundo interés en alguna temática en particular.

Problema:

Clasificar cientos de resultados sobre un determinado tópico requiere dedicar más tiempo de lo que uno puede desear. Sin un guía o referencia, los usuarios terminan por desanimarse y se dan por vencidos o bien trabajan con sólo una parte de la información disponible.

Cómo responder preguntas

Cuando alguien accede a un sitio Web de información, espera obtener respuestas a sus preguntas o inquietudes de forma rápida y fácil. Las respuestas pueden ofrecerse de tres formas:

- Como una Web de conocimientos propios o recoleccionados.
- Como un directorio de los mejores recursos de Internet.
- Como una combinación de las dos anteriores.

Adoptaremos el tercer enfoque: un sitio con descripciones pormenorizadas de patrones de diseño hipermedia, complementadas con referencias a recursos externos.

Credibilidad y confianza

Cuando alguien confía en un sitio Web, confía en que la información que en él encuentra sea veraz. Hay que proporcionar por tanto, información veraz para que crezca la credibilidad y confianza en el sitio. Para ello se aconseja buscar la exactitud de la información y huir de las ambigüedades. Una buena práctica es fechar cada una de las páginas para que los usuarios del portal conozcan la antigüedad de la información que leen y si, en función a la misma, la consideran relevante para sus propósitos.

Recursos externos

Es conveniente usar principios básicos de organización y navegación. Hay que evitar largas listas de recursos porque los visitantes no tienen forma de saber qué es interesante y qué no lo es. Añadir pequeños comentarios sobre cada enlace ayuda a los visitantes a encontrar más rápidamente lo que buscan.

5. ANÁLISIS

Si el número de enlaces es especialmente largo, es recomendable agruparlos por contenido y etiquetar los grupos resultantes, evitando además un excesivo *scrolling*. Cada categoría se corresponderá con una página con 15 o 20 enlaces como máximo, y se incluirá en un nivel superior un directorio de las categorías creadas.

Escribiendo para buscadores

La regla básica a la hora de escribir en un sitio Web se reduce a conocer tu audiencia. En este caso hay dos audiencias a tener en cuenta. Por un lado están los propios lectores del sitio. Y por otro lado están los buscadores, porque lo normal es que mucha de la gente dé con el sitio Web a través de uno de ellos.

Crecimiento del sitio

Con el tiempo, lo que se espera de este tipo de sitios es que crezcan con nuevas páginas y recursos. Una forma de conseguirlo es procurando que los usuarios participen, a través de sus opiniones o incluso proporcionando nuevos contenidos. Este paso representa un cambio fundamental en el modo en que el sitio opera, ya que el sitio pasa de ser mantenido por un pequeño grupo de administradores a estar pendiente también de las aportaciones que puedan hacer los usuarios.

Buscaremos en qué modo permitiremos la participación de los usuarios con los contenidos de nuestro sitio Web.

5.4 Especificación de requisitos

5.4.1 Introducción

En base a las conclusiones extraídas en los apartados anteriores, pasamos a elaborar el listado o catálogo de requisitos que describirán las necesidades que debe cubrir el sitio Web a desarrollar, y en particular su interfaz de usuario. Al igual que sucede con el proceso de captura de requisitos, en el proceso de definición de requisitos existen un gran número de técnicas propuestas. El uso del lenguaje natural, la definición de diagramas de casos de uso, o la cumplimentación de plantillas estructuradas, son algunas de las técnicas más relevantes, y a menudo se usan en combinación. Para nuestro proceso de definición de requisitos nos limitaremos a emplear una plantilla muy sencilla compuesta por una serie de campos que describimos en la Tabla 4.

5. ANÁLISIS

Hay que tener en cuenta que el desarrollo de sistemas Web agrupa una serie de características que lo hacen diferente del desarrollo de otros sistemas, y que por tanto estas características también hay que tenerlas en cuenta a la hora de especificar los requisitos [Escalona *et al.*, 2002]. Para el entorno Web nos encontramos, como es habitual, con varias propuestas, en las que además no siempre se emplea la misma terminología. De forma sintetizada se nos presenta en [Escalona *et al.*, 2002] una clasificación de requisitos relevantes en sistemas Web:

- *Requisitos de datos.* Estos requisitos responden a la pregunta de qué información debe almacenar y administrar el sistema.
- *Requisitos de interacción.* Responden a la pregunta de cómo va a interactuar el usuario con el sistema.
- *Requisitos de navegación.* Necesidades de navegación del usuario.
- *Requisitos de personalización.* Describen cómo debe adaptarse el sistema en función de qué usuario interactúe con él.
- *Requisitos funcionales internos.* Recogen qué debe hacer el sistema de forma interna, sin incluir aspectos de interfaz o interacción.
- *Requisitos no funcionales.* Son por ejemplo los requisitos de portabilidad, de entorno de desarrollo, de usabilidad, de disponibilidad, etc.

Nuestra intención es la de adoptar esta clasificación de requisitos para nuestro proceso de definición particular. En la Tabla 4, el campo “Tipo” se encargará de recoger el valor que identifica el grupo clasificatorio al que pertenece el requisito.

Campo	Descripción
Identificador	Clave única que nos servirá para, en ocasiones posteriores, poder referenciar al requisito que identifica.
Tipo	Grupo clasificatorio en el que se enmarca el requisito. Posibles valores: Datos, Interacción, Navegación, Personalización, Funcional, No Funcional.
Origen	Indica de dónde viene el requisito. Tres posibilidades; de la evaluación de los sitios, del análisis de los patrones del grupo A, o de necesidades deducidas de los objetivos generales del proyecto.
Descripción	Descripción en lenguaje natural del propósito

Tabla 4: Plantilla para la definición de los requisitos del sitio

5.4.2 Catálogo de requisitos

Identificador: R1

Tipo: Funcional
Origen: Objetivos generales
Descripción: El sitio posibilitará la realización de búsquedas rápidas de patrones a partir de un texto libre.

Identificador: R2

Tipo: Funcional
Origen: Objetivos generales
Descripción: El sitio posibilitará la realización de búsquedas avanzadas a través del contenido semántico de los patrones del repositorio.

Identificador: R3

Tipo: Interacción
Origen: Evaluación de sitios
Descripción: La búsqueda rápida de patrones estará accesible en todas las páginas del sitio.

Identificador: R4

Tipo: Navegación
Origen: Evaluación de sitios
Descripción: Se presentará una estructura de navegación jerárquica basada en categorías para la exploración del repositorio de patrones.

Identificador: R5

Tipo: Funcional
Origen: Evaluación de sitios
Descripción: El sitio mostrará el contenido de cada categoría de la estructura jerárquica del repositorio en páginas independientes.

Identificador: R6

Tipo: No Funcional
Origen: Patrones grupo A
Descripción: Se paginará el contenido de cada categoría si éste se compone de más de 15 enlaces.

Identificador: R7

Tipo: Navegación
Origen: Evaluación de sitios
Descripción: Se presentará una estructura de navegación en orden alfabético para la exploración del repositorio de patrones.

Identificador: R8

Tipo: Funcional
Origen: Objetivos generales
Descripción: El sitio dará acceso a las descripciones completas de cada patrón del repositorio.

Identificador: R9

Tipo: Navegación
Origen: Evaluación de sitios
Descripción: En la página de descripción de cada patrón se habilitará un índice con enlaces internos a las diferentes secciones que componen la descripción.

Identificador: R10

Tipo: Funcional
Origen: Evaluación de sitios
Descripción: El sitio facilitará una versión imprimible de la descripción de cada patrón.

Identificador: R11

Tipo: Navegación
Origen: Evaluación de sitios
Descripción: En la página de descripción de cada patrón se destinará un espacio de la misma para agilizar el acceso a los elementos relacionados con el patrón: blog, patrones relacionados y versión imprimible.

Identificador: R12

Tipo: Funcional
Origen: Patrones grupo A
Descripción: El sitio incluirá información acerca de los patrones de diseño a nivel general, y más en particular del formato y organización de nuestro repositorio de patrones de diseño hipermedia.

Identificador: R13

Tipo: Funcional
Origen: Evaluación de sitios / Patrones grupo A
Descripción: El sitio complementará el repositorio de patrones con referencias externas a información relacionada en forma de enlaces.

Identificador: R14

Tipo: Navegación
Origen: Evaluación de sitios
Descripción: En función del tipo de información referenciada, las referencias externas a información relacionada se categorizarán en cuatro secciones independientes y accesibles de forma directa: Bibliografía, Descargas, Noticias y Otros repositorios.

Identificador: R15

Tipo: No Funcional
Origen: Patrones grupo A
Descripción: Se paginará el contenido de cada sección de referencias externas si éste se compone de más de 15 enlaces.

Identificador: R16

Tipo: Funcional
Origen: Evaluación de sitios / Patrones grupo A
Descripción: El sitio dará acceso a un blog de opinión o sección de comentarios independiente por cada patrón del repositorio.

Identificador: R17

Tipo: Funcional
Origen: Evaluación de sitios / Patrones grupo A
Descripción: El sitio permitirá introducir al usuario nuevos comentarios en los blogs de opinión asociados a cada uno de los patrones.

Identificador: R18

Tipo: No Funcional
Origen: Evaluación de sitios
Descripción: Se emplearán infraestructuras propias e internas al sitio para los blogs de opinión asociados a cada uno de los patrones.

Identificador: R19

Tipo: No Funcional
Origen: Objetivos generales
Descripción: La apariencia del sitio y de los diferentes componentes de la página irá acorde a la funcionalidad y propósito del mismo.

Identificador: R20

Tipo: No Funcional
Origen: Patrones grupo A
Descripción: Se tendrán en consideración los motores de búsqueda de Internet a la hora de implementar el sitio.

Identificador: R21

Tipo: No Funcional
Origen: Objetivos generales
Descripción: El sitio deberá ser accesible durante las 24 horas del día, todos los días del año.

Identificador: R22

Tipo: No Funcional
Origen: Objetivos generales
Descripción: El sitio estará optimizado para cualquier navegador desde el que se acceda.

Identificador: R23

Tipo: No Funcional
Origen: Objetivos generales
Descripción: El sitio se adaptará a diferentes idiomas sin necesidad de realizar cambios de código.

6 DISEÑO DE LA INTERFAZ DE USUARIO

Durante el planteamiento del problema, identificamos el catálogo de patrones de diseño recogido en [van Duyne *et al.*, 2002] como la herramienta en la que basamos la solución a la interfaz de usuario de nuestro sitio Web. Esto lo vamos a concretar en la fase de diseño, de la cual se ocupa este capítulo. Los requisitos especificados al final del capítulo anterior marcan el punto de partida de esta fase, ya que gracias a ellos sabemos con mayor proximidad lo que queremos y por tanto tenemos un criterio más definido a la hora de seleccionar los patrones de diseño del catálogo que nos han de ayudar a construir la interfaz de usuario de nuestro sitio Web.

El proceso de exploración y selección de los patrones debe realizarse en base a un orden. El formato en el que éstos son expuestos en el libro ayudará a formalizar este proceso en una serie de pasos a seguir similares a los que propuso Alexander [Alexander *et al.*, 1977]. La selección de los patrones irá dando lugar a un lenguaje de patrones propio válido para nuestro sitio Web. El objetivo final de todo esto es pasar a aplicar dichos patrones para empezar a dar forma al sitio, pero antes será conveniente establecer un orden que rija el proceso de aplicación, por lo que definiremos unas pautas centradas en el lenguaje de patrones extraído previamente, que nos guíen de forma adecuada por el mismo. En la “Introducción al diseño de la interfaz de usuario” (véase el apartado 6.1) se explicarán todos estos asuntos más detalladamente.

La mayor parte del capítulo se dedicará a profundizar en la aplicación individual de cada patrón seleccionado (véase el apartado 6.2). Para finalizar mostraremos una tabla que asocie los requisitos deducidos del análisis y los requisitos que deduzcamos del diseño con los patrones que se han utilizado para cubrirlos (véase el apartado 6.3).

6.1 Introducción al diseño de la interfaz de usuario

Culminábamos la fase de análisis con una especificación de requisitos, y con ellos se lograba establecer la funcionalidad y los límites del sitio Web, y sentar las bases de la estructuración y contenido deseados de la interfaz de usuario. Lo que conseguimos con esto es, al fin y al cabo, formarnos un criterio a partir del cual poder decidir qué patrones del catálogo recogido en [van Duyne *et al.*, 2002] debemos seleccionar para aplicarlos en el diseño de la interfaz de usuario y cuáles debemos descartar.

Si hablamos de fase de diseño de forma estricta, ésta se referirá fundamentalmente a la aplicación de los patrones. Sin embargo no hay que olvidar que para poder aplicarlos antes hay que seleccionarlos y que esto conforma por tanto un proceso previo al de aplicación. Como para cualquier otro proceso, es de desear el establecer una serie de pautas ordenadas que lo caractericen. El formato de descripción de los patrones de van Duyne se adapta muy bien a los pasos propuestos en su día por Alexander para llevar a cabo la selección del conjunto de patrones que resuelven un determinado problema [Alexander *et al.*, 1977]. Conforme se avance en el proceso de selección se irá dando forma a un lenguaje de patrones propio que será el que utilicemos finalmente en el proceso de aplicación.

Por tanto, distinguimos aquí el concepto de catálogo de patrones del concepto de lenguaje de patrones. Un catálogo se puede definir como un grupo de patrones clasificados por algún criterio, mientras que un lenguaje, en palabras de van Duyne, es una colección de patrones interrelacionados y organizados como un todo que proporciona una solución detallada a un problema de diseño a gran escala [van Duyne *et al.*, 2002]. Veremos más adelante qué criterios de clasificación emplea van Duyne para agrupar sus patrones, y veremos también que éstos están relacionados unos con otros de tal modo que podemos obtener múltiples combinaciones en forma de lenguajes.

También nos hemos referido a la aplicación de los patrones como un proceso, y como tal también trataremos de definir una mecánica que lo guíe. A grandes rasgos consistirá en atravesar el lenguaje de patrones construido en el proceso de selección, lo cual generará un diseño concreto. También habrá que tener en cuenta que a medida que apliquemos cada patrón, surgirán nuevas necesidades o características no recogidas en nuestro catálogo de requisitos que queramos incluir en el mismo.

6.1.1 Construcción del lenguaje de patrones

El arquitecto Christopher Alexander propuso por primera vez un lenguaje de patrones para poder diseñar ciudades, barrios, grupos de edificios, plazas, edificios, habitaciones, etc. Este orden en que describimos los elementos posibles de ser construidos con los patrones propuestos por Alexander es el que el propio autor recomienda para su utilización. Alexander subraya que lo más importante de la secuencia es que está basada en las conexiones entre los patrones. Cada uno de ellos está conectado con algún otro más amplio y que está por encima en el lenguaje definido por los patrones, y además con otros más reducidos que están por debajo en el lenguaje. Cada nivel de patrones ayuda a completar un nivel superior y a su vez está completado por un nivel inferior [Alexander *et al.*, 1977].

Letra	Nombre	Resumen
A	Site Genres	Explica los aspectos de cada <i>género</i> de sitio, y ayuda a escoger patrones más detallados.
B	Creating a Navigation Framework	Ayuda a maximizar la flexibilidad del sitio para acomodar los diferentes hábitos de navegación del cliente.
C	Creating a Powerful Homepage	Describe cómo diseñar una página de inicio potente y ajustada a las necesidades del cliente.
D	Writing and Managing Content	Presenta una visión general para ayudar a crear y gestionar el contenido de forma efectiva.
E	Building Trust and Credibility	Trata temas relacionados con la confianza y la credibilidad del sitio Web.
F	Basic E-Commerce	Describe cómo mejorar la experiencia del cliente en un sitio de comercio electrónico.
G	Advanced E-Commerce	Describe características avanzadas y opcionales que se puede desear incluir en un sitio de comercio electrónico.
H	Helping Customers Complete Tasks	Describe las formas de estructurar el sitio para minimizar los problemas e incrementar el ratio de tareas realizadas.
I	Designing Effective Page Layouts	Describe cómo crear composiciones que los clientes encuentren claras, predecibles y fáciles de entender.
J	Making Site Search Fast and Relevant	Trata el diseño de la interacción de los clientes, de modo que las búsquedas de los clientes sean más efectivas.
K	Making Navigation Easy	Describe las técnicas para organizar y mostrar los elementos de navegación para hacerlos fáciles de encontrar y comprender.
L	Speeding Up Your Site	Describe técnicas para hacer que un sitio Web sea más rápido.

Tabla 5: Grupos de patrones del catálogo de van Duyne

El catálogo de patrones de van Duyne se ajusta fielmente a esta idea trasladándola al mundo del diseño Web. Éste propone una serie de grupos organizados por letra y por nombre (véase la Tabla 5), cada uno de los cuales contiene una colección de patrones relacionados temáticamente. Según el esquema propuesto los primeros grupos deberían ser los primeros en usarse en el proceso de diseño. Por su parte, cada patrón presenta un formato compuesto de seis secciones: el nombre; el contexto y alcance del patrón; el problema que se intenta resolver; el cuerpo o desarrollo del problema; la solución al problema; y para finalizar se señalan los otros patrones a considerar, concretamente aquellos patrones a más bajo nivel que ayudan a completar el patrón. Si observamos las descripciones de cada patrón vemos que se identifican patrones relacionados en varias de las secciones; en el contexto, en el cuerpo y en otros patrones a considerar. Esta red de patrones proporciona un modo de seleccionar rápidamente aquellos que se necesitan para completar un determinado diseño, y facilita la articulación de un gran número de diseños para escenarios diferentes.

Por tanto, el proceso de selección de los patrones del catálogo de van Duyne que han de ayudarnos a resolver el diseño de nuestra interfaz de usuario se puede basar en los pasos que propuso Alexander. El seguimiento de estos pasos irá modelando el lenguaje de patrones que determinará el diseño concreto para nuestro escenario:

1. Elegir el patrón que mejor describe el alcance global del proyecto.
2. Ir al final del patrón, donde se referencian los patrones de menor escala que soportan ese patrón, y hacer una lista con los que parecen que se pueden aplicar al proyecto.
3. Para cada patrón seleccionado en el paso 2, repetir el paso anterior y también examinar los patrones de mayor escala que aparecen al comienzo de cada patrón, añadiendo todos los que sean relevantes a la lista.
4. Repetir los pasos 2 y 3 hasta conseguir una lista de patrones.
5. Ajustar la lista de patrones añadiendo material propio, modificando los patrones existentes para que sean más relevantes a la situación actual o creando nuevos patrones.

El primer paso nos indica que hay que seleccionar el patrón que mejor describe el alcance global del proyecto. Éste lo deberemos buscar de entre los patrones del grupo A de modo que hagamos corresponder el sitio que queremos diseñar con alguno de los géneros de sitios Web que se identifican en este grupo. Si recordamos, este paso lo cubrimos durante la fase de Análisis (véase el Capítulo 5) ya que consideramos que estos patrones nos serían de utilidad para definir mejor las características generales de nuestro sitio Web. Vimos que el patrón A6 - *Grassroots Information Sites* era el que más se ajustaba a la idea de un sitio Web de referencia informativa respecto a una temática concreta, como es en nuestro caso los patrones de diseño hipermedia.

El segundo paso por tanto consistiría en seleccionar los patrones de menor escala relacionados con los que hemos seleccionado en primera instancia y que, bajo el criterio que nos hemos creado durante la fase de Análisis, consideremos aplicables a nuestro sitio. Por ejemplo, comprobamos que el patrón A6 nos emplaza a diseñar un contenido explorable mediante herramientas de navegación siguiendo las indicaciones del patrón B2 - *Browsable Content*. Conforme al tercer paso, tendremos que repetir esta operación para este patrón. Pero además hay que examinar los patrones de mayor escala que aparecen al comienzo de su descripción, por lo que observamos que el patrón B2 - *Browsable Content* es en realidad un requisito del patrón B1 - *Multiple Ways of Navigate*, el cual añadiremos a la lista para tenerlo en consideración.

Así se procede repetidamente hasta que el proceso va descansando paulatinamente en patrones de menor escala. Como ya dijimos previamente, con esto en realidad lo que conseguimos es modelar un lenguaje de patrones interrelacionados que en conjunto caracterizarán el diseño del sitio Web. Una manera intuitiva de representar este lenguaje es empleando un diagrama en forma árbol con flechas que representen las relaciones entre los patrones. En la página siguiente se muestra el diagrama con el que queda definido el lenguaje de patrones que aplicaremos a nuestro diseño.

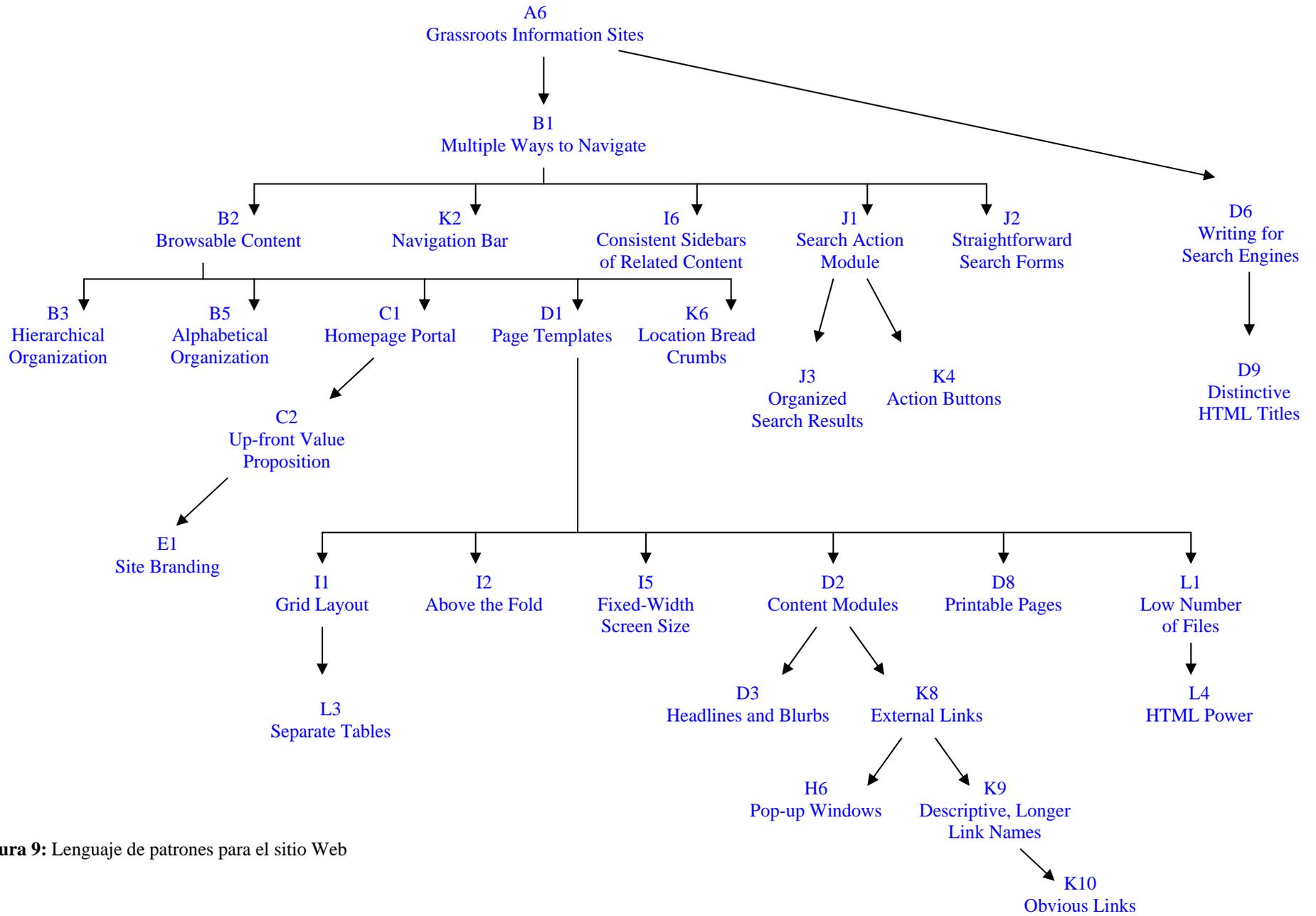


Figura 9: Lenguaje de patrones para el sitio Web

6.1.2 Exploración del lenguaje de patrones

Definido el lenguaje, se procederá a aplicar los patrones que lo componen a medida que se va explorando. Se ha procurado definir un proceso ordenado para la exploración basado en la estructura en árbol mostrada en la figura anterior, y que queda resumido en los siguientes pasos:

1. Aplicar el patrón raíz del árbol
2. De los patrones hijos, aplicar el que represente un mayor nivel de abstracción.
3. Si hemos llegado a un patrón hoja del árbol, regresar al patrón padre del mismo.
4. Repetir 2 y 3, hasta haber aplicado todos los patrones.

6.2 Aplicación de patrones

6.2.1 Patrón *Multiple Ways to Navigate* (B1)

Este patrón identifica los diferentes esquemas de navegación que determinan cómo los usuarios navegan por los sitios.

Problema:

Estamos acostumbrados a navegar por los sitios Web de diversas maneras. Es por ello por lo que deberemos ofrecer diferentes tipos de navegación que cubran por completo las necesidades de movilidad de los usuarios.

Intención e impulso

Hay que tener presentes las dos formas en las que un usuario se comporta cuando navega por un sitio Web; intención e impulso. Por “intención” se entiende el hecho de dirigirse a un sitio Web con un objetivo concreto, como puede ser el realizar una compra o buscar información sobre un determinado tema. Por “impulso” consideramos aquellas acciones conducidas por las cosas que llaman nuestra atención, como puede ser dirigirse a un elemento relacionado con el que visualizas en un momento dado.

6. DISEÑO DE LA INTERFAZ DE USUARIO

Estilos de navegación

En el comportamiento dirigido por una intención, se enmarcan las típicas opciones de navegación basadas en la **búsqueda** (*search*), la **exploración** (*browse*) y en el **paso a paso** (*wizard*).

Proporcionaremos en todas las páginas una opción de búsqueda rápida de patrones
→ patrón *Search Action Module* (J1)
Proporcionaremos acceso a las principales partes de nuestro sitio Web a través de barras de navegación → patrón *Navigation Bar* (K2)

Por su parte, dentro del comportamiento impulsivo se identifican los estilos de navegación de **relación** (*relate*) y **promoción** (*promote*).

Los patrones de diseño relacionados con un determinado patrón y el Blog de opinión que asociaremos a cada uno de ellos, son ejemplos claros de relación. Diseñaremos una barra lateral adecuada para dar acceso a elementos relacionados donde sea necesario → patrón *Consistent Sidebars of Related Content* (I6)

Ubicación de las herramientas de navegación

No sólo es cuestión de tener diferentes opciones de navegación, sino también de ubicarlas correctamente. Proporcionar una buena localización es una de las decisiones más importantes para hacer la navegación más fácil:

- Las herramientas que permitan al usuario alcanzar sus objetivos e intenciones, se sitúan cerca de donde se comienza a leer. Es decir, a la izquierda y en la parte superior de la página (*search, browse, wizard*). Esto asegura que estas herramientas serán encontradas y usadas.
- Aquellas otras que permiten dar una continuidad o completar algo, se deben situar donde se continúa leyendo. Es decir, en la parte superior derecha de la página y en la parte inferior (*relate, promote*).

Además, por su propia naturaleza, las reacciones impulsivas no se pueden garantizar, por lo que el espacio de página dedicado a las herramientas de navegación impulsiva debe ser menor.

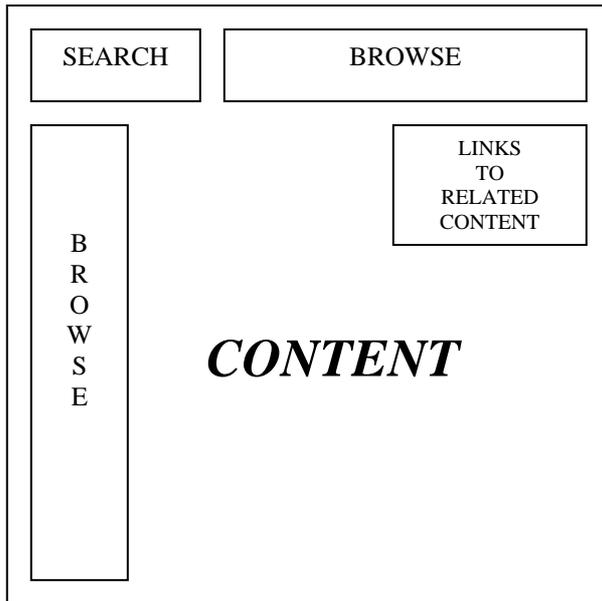


Figura 10: Un primer esquema del *framework* de navegación

6.2.2 Patrón *Browsable Content* (B2)

Este patrón hace explorable el contenido a través de una combinación de sugerencias organizacionales y de navegación.

Problema:

Explorar el contenido de un sitio Web puede resultar difícil si la información no está organizada o si no hay un plan de navegación adecuado para encontrar lo que se busca y regresar al mismo punto posteriormente si así se desea.

Usar un esquema organizacional

Hay que aplicar el mejor esquema de organización posible para nuestro sitio Web. La elección dependerá del tipo de información involucrada, del volumen de ésta, y de las tareas que pueda llevar a cabo el usuario. Se proponen diferentes patrones para diferentes esquemas organizacionales:

- Organización jerárquica.
- Organización basada en tareas.
- Organización alfabética.
- Organización cronológica.
- Organización basada en la popularidad.

6. DISEÑO DE LA INTERFAZ DE USUARIO

La utilización de una de ellas no excluye el empleo de otras. Cada sección o subsitio del sitio Web puede ajustarse a esquemas de organización que difieran del resto. Por tanto pueden aplicarse de forma aislada o combinada.

Se aplicará una organización jerárquica basada en categorías para el repositorio de patrones → patrón *Hierarchical Organization* (B3)

Será de utilidad dar acceso a la lista completa de los patrones del repositorio ordenados alfabéticamente → patrón *Alphabetical Organization* (B5)

Para los mensajes de foros y blogs puede que nos resulte más adecuada una organización cronológica → patrón *Chronological Organization* (B6)

Ayudar a encontrar el camino de vuelta

Es importante que un usuario logre encontrar con facilidad el camino de vuelta durante la navegación por el sitio Web para evitar que se sienta perdido y que como consecuencia decida no aventurarse por el mismo.

Ayudaremos al usuario a no perder el camino recorrido apoyándonos en “menús de miga de pan” → patrón *Location Bread Crumbs* (K6)

Habilitaremos en todas las páginas un enlace para regresar a la página de inicio sobre la “imagen de marca” de nuestro sitio → patrón *Site Branding* (E1)

Sobrecarga de contenido

Es imprescindible no sobrecargar las páginas con mucha información, de modo que el usuario no reciba una sensación de saturación que podría hacerle ceder en su empeño por leer el contenido mostrado. Hay muchos factores que contribuyen a alimentar esta sensación de sobrecarga, desde el tamaño de fuente seleccionado hasta la ausencia de una organización de contenido.

Se puede mitigar la sobrecarga de información mediante técnicas que permitan a los usuarios explorar una página y encontrar con rapidez lo que buscan. Presentar cada página mediante una disposición tabulada de los diferentes elementos, con una fuerte jerarquía visual, con áreas consistentes de contenido y de navegación, y emplear una fuente fácilmente legible, son algunas de ellas.

En todo esto se profundizará cuando organicemos el contenido de las páginas → patrones *Homepage Portal (C1)*, *Page Template (D1)*, *Content Modules (D2)*, *Grid Layout (I1)*

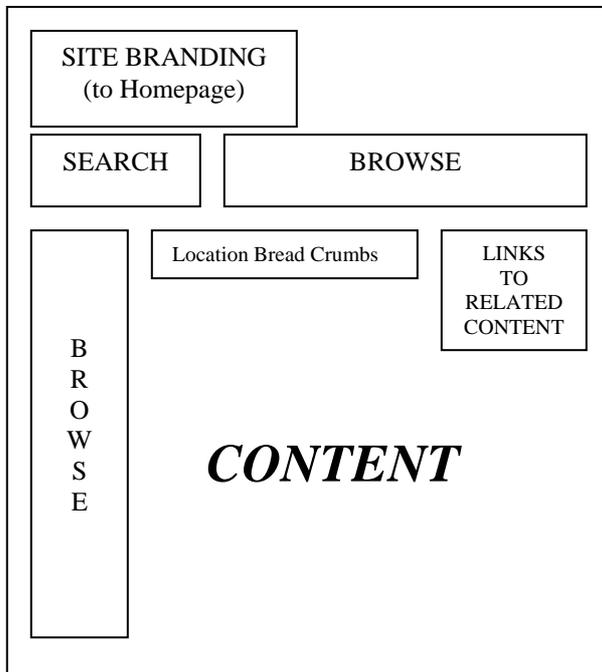


Figura 11: Se perfila el esquema final del framework de navegación

6.2.3 Patrón *Hierarchical Organization (B3)*

Este patrón proporciona una forma de organizar grandes cantidades de contenido cuando éste puede ser estructurado jerárquicamente.

Problema:

Organizar la información en una jerarquía de categorías puede ayudar a los usuarios a encontrar cosas. Construir una jerarquía efectiva no es fácil.

Nombres descriptivos y distintivos de categorías

Los términos que un diseñador piensa como los mejores para describir una categoría pueden no ser los mismos que aquellos que la mayoría de los usuarios elegirían.

Las etiquetas que representen las categorías deben ser descriptivas acerca de lo que cada una de ellas abarca. Hay que evitar por tanto nombres ambiguos (como “miscelánea” u “otros”) que no ayudan al usuario a determinar exactamente qué contienen en realidad. De la misma forma hay que procurar emplear etiquetas que sean

6. DISEÑO DE LA INTERFAZ DE USUARIO

distintivas unas de otras, de modo que no haya lugar a la confusión entre lo que contiene cada una de ellas.

En nuestro caso no hará falta ninguna entrevista ni ningún tipo de estudio previo. Las categorías en las que se dividen los patrones ya nos vienen dadas. Al final de este punto se muestran las mismas.

Proporcionar ejemplos en cada categoría

De forma que los usuarios puedan comprender más fácilmente lo que contiene cada categoría y así estar más seguros de hacia donde se dirigen, es aconsejable adjuntar una pequeña descripción al enlace de cada categoría o pequeñas muestras en forma de ejemplos de lo que contienen.

Mantener el número de subcategorías por debajo de 50

Cuando las listas de subcategorías son muy largas, puede acabar en algo incómodo de leer. Manteniendo el número de subcategorías por debajo de las 50, se logrará que las páginas sean más fáciles de leer y una mayor rapidez de navegación. Si se tienen más de 50, hay que intentar definir nuevas subcategorías para cada subcategoría, o bien combinar subcategorías similares.

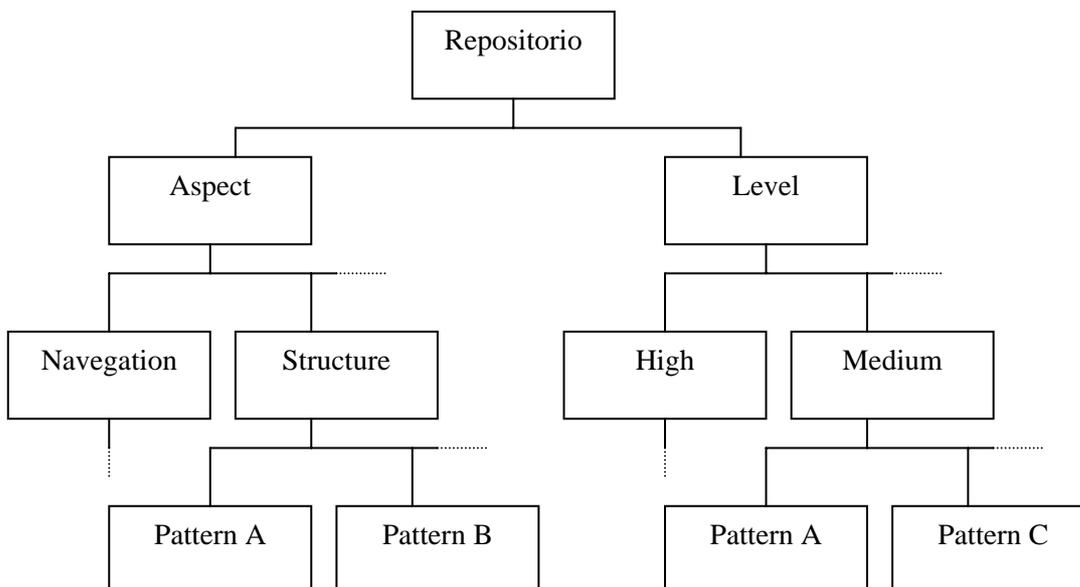


Figura 12: Organización jerárquica por categorías de nuestro repositorio de patrones

6.2.4 Patrón *Alphabetical Organization* (B5)

Este patrón proporciona una forma de organizar un volumen de contenido relativamente pequeño cuando éste no está relacionado o se compone de elementos de nombres bien conocidos.

Problema:

Ordenar una lista alfabéticamente parece una forma evidente de organizar el contenido. Sin embargo, en un sitio Web las listas alfabéticas demasiado largas son incómodas de usar.

Cuándo emplear una lista alfabética

Es una opción muy efectiva cuando cada elemento de información es conocido por el nombre más que por lo que representa. Cuanto más conozcan los usuarios el nombre de aquello que buscan, más rápido les resultará encontrarlo en una lista ordenada alfabéticamente. Aquí se listan algunas de las ocasiones en las que es útil emplear este tipo de organización:

- Listas de artículos conocidos por su nombre.
- Listado de todos los documentos de un sitio Web.
- Listado de todos los clientes de una compañía.
- Listado de todos los productos que ofrece una compañía.

Intentar mantener la lista en una página

Si una lista alfabética está dividida en varias páginas Web, los usuarios deben pulsar en la letra que da comienzo a la primera palabra de lo que buscan, y esperar a que se cargue la página asociada a dicha letra. Esto puede resultar frustrante si no se garantiza que lo que se busca comienza realmente por esa palabra. Por ello puede ser más fácil encontrar algo si la lista se despliega por completo en una misma página, apoyada por un índice en la parte superior de la página que enlace con cada grupo de elementos de una misma letra.

Emplearemos una sola página apoyándonos en un índice en la parte superior con enlaces internos al comienzo de cada grupo de patrones asociado a una misma letra.

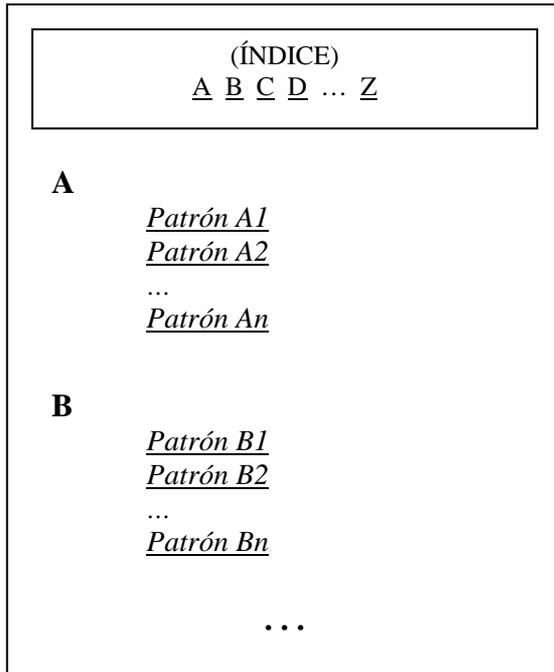


Figura 13: Listaremos en una sola página todos los patrones del repositorio

6.2.5 Patrón *Homepage Portal* (C1)

Este patrón ayuda a perfilar el grueso del diseño de la página de inicio de un sitio Web.

Problema:

Las páginas de inicio deben seducir a los visitantes mediante un estilo adecuado a la vez que deben guardar un equilibrio con otros aspectos como pueden ser la navegación, el contenido, la marca e identidad del sitio, y la rapidez de descarga.

Marca e identidad del sitio

La marca de un sitio Web es una promesa de lo que éste ofrece, y que tiene como objetivo atraer a los usuarios para que naveguen por el sitio. Una vez conseguida la atención, hay que intentar satisfacer continuamente la promesa para ganar su respeto y confianza.

La identidad del sitio dependerá de la definición de dos aspectos:

1. La idea o proposición fundamental que se quiere transmitir al usuario → patrón *Up-front Value Proposition* (C2)
2. La imagen de marca → patrón *Site Branding* (E1)

6. DISEÑO DE LA INTERFAZ DE USUARIO

Lograr una primera impresión positiva del estilo

Si no se aplica un estilo adecuado a la naturaleza y propósito del sitio Web, se corre el riesgo de que los usuarios piensen que no es lo que estaban buscando. Por tanto hay que diseñar pensando en la audiencia potencial del sitio.

El núcleo central de nuestro sitio Web es el repositorio de patrones de diseño hipermedia. Por tanto la finalidad principal es la de ser un sitio informativo, didáctico. Por ello consideramos que el estilo debe ser sobrio pero evitando caer en una sensación de frialdad.

Seducir con el contenido

Pasados sólo unos segundos tras entrar a un sitio Web, los usuarios hacen un juicio de valor del mismo. Ese es el momento en el que se debe captar y retener su atención. Además de una escritura y unos recursos visuales vivos, es importante determinar los contenidos apropiados a mostrar en la página. Hay que procurar organizar estos contenidos en títulos, resúmenes y cuerpo.

Como una buena práctica extraída de la evaluación del portal Yahoo! Design Pattern Library, destinaremos una parte de la página de inicio a presentar resumidamente los últimos patrones incorporados a nuestro repositorio. A los usuarios habituales del sitio les será de gran utilidad, y en general dará una idea de lo que se encontrarán en otras páginas, además de fomentar la impresión de que se trata de un sitio Web dinámico y actualizado.

Muchas veces un visitante puede no saber por donde empezar. Por ello destinaremos otra zona de la página de inicio, para sugerirles la mejor manera de comenzar a explorar los contenidos de nuestro sitio Web.

Hacer una página de inicio ligera

Si un usuario se dirige a un sitio y la página de inicio tarda demasiado en cargarse, lo más probable es que desista y dé marcha atrás. Estrategias para agilizar las descargas:

- La primera vez que se accede a un sitio Web, las imágenes de la página de inicio no están aún cacheadas, lo cual ralentiza la descarga. Para combatirlo,

6. DISEÑO DE LA INTERFAZ DE USUARIO

lo mejor es aprovechar las capacidades del HTML y emplear texto siempre que sea posible.

- Usar imágenes rápidas de descargar.
- Usar tablas HTML separadas en lugar de una gran tabla, ya que una única gran tabla fuerza al usuario a esperar a que todas las imágenes se hayan cargado antes de ver algo.
- Evitar contenido de descarga lenta como sonido, animaciones *flash* o *applets*, no sólo por la lentitud sino porque se corre el riesgo de que los usuarios no dispongan de las últimas tecnologías instaladas en sus ordenadores.

Algunos de estos aspectos se tratarán con más profundidad cuando analicemos algunos de los patrones del Grupo L, los cuales nos ayudarán a mejorar el rendimiento del sitio.

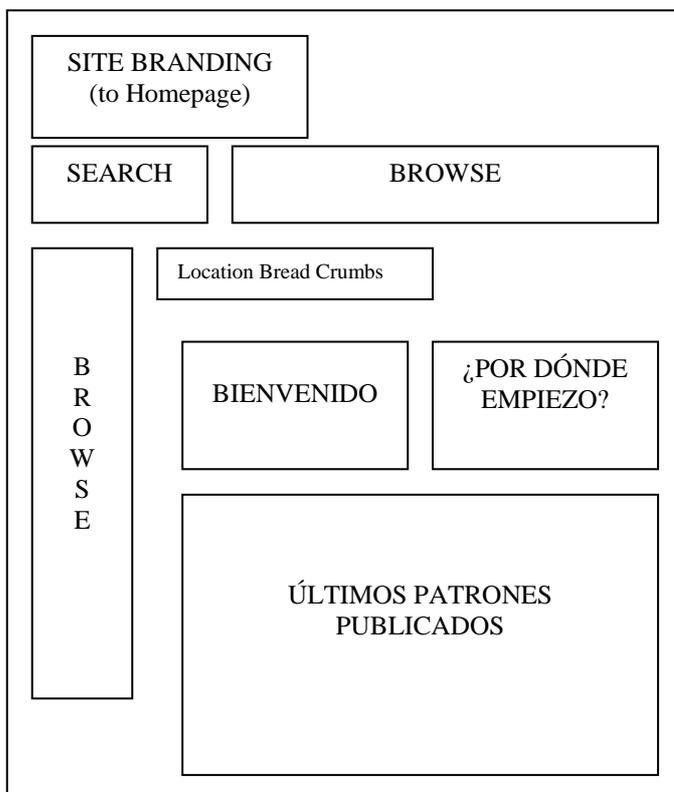


Figura 14: Página de inicio a nivel de bloques

6.2.6 Patrón *Up-front Value Proposition* (C2)

El presente patrón de diseño se centra en ofrecer una manera contrastada de articular una promesa de gran alcance.

Problema:

En muchos sitios Web, a menudo los visitantes no pueden asegurar con certeza lo que éstos ofrecen o no llegan a comprenderlo bien, a pesar incluso de que el sitio determinado esté intentando comunicarlo.

El objetivo reside en explicar a los visitantes lo que el sitio ofrece de forma convincente, con una única frase o promesa. Esta frase será el tema que dé una unidad al sitio. En cualquier caso no hay que olvidar que la promesa por sí sola no termina de ser suficiente. Ésta tiene que ser satisfecha a lo largo de todas las páginas del sitio y reforzada a través de buenas prácticas.

A pesar de que nuestro sitio Web será algo más que un repositorio de patrones de diseño hipermedia, lo que es cierto es que todo girará en torno al repositorio y por tanto la promesa o mensaje que ideemos debe hacer referencia al mismo.

Requisitos

La página de inicio es una especie de anuncio para el resto del sitio, y es donde más relevancia tendrá el mensaje o idea que se quiere transmitir. Esto es lo que se necesita para conseguir un mensaje efectivo:

- Una promesa persuasiva.
- Un único ofrecimiento.
- Imágenes y texto descriptivos que sean fáciles de comprender.

Se resume en una serie de pasos el proceso (a realizar en equipo) recomendado para poder crear un mensaje único y persuasivo:

- Desarrollar una lista inicial de proposiciones. En un principio, toda sentencia propuesta es igual de válida.

6. DISEÑO DE LA INTERFAZ DE USUARIO

- Cada miembro del equipo seleccionará los candidatos que considere más fuertes y tendrá que defender con argumentos su elección. De una reunión tendrán que salir un pequeño conjunto con los mejores mensajes.
- Este conjunto deberá ser evaluado por potenciales usuarios del sitio. Si no surgen calificaciones muy positivas para ninguno de los candidatos, se repetirá el proceso.
- De los mejor calificados, se vuelve a preguntar al mismo grupo de usuarios. De aquí saldrá el mensaje definitivo.

No vendemos productos, ni queremos sacar provecho económico del sitio Web. El sitio es puramente informativo, por lo que el mensaje debe ser sencillo y funcional:

“Repositorio de Patrones de Diseño Hipermedia”

Integrar el mensaje en el sitio

Habrá que desarrollar diferentes diseños para que acompañen al mensaje. En general, lo mejor es que esté próximo o integrado en la marca o logo del sitio Web.

Nos apoyaremos en otro patrón para realizar esta integración convenientemente → patrón *Site Branding* (E1)

6.2.7 Patrón *Site Branding* (E1)

Este patrón de diseño ayuda a definir la imagen de marca del sitio.

Problema:

Una marca es algo más que una imagen. Es lo que la gente recuerda. Depende de la audiencia a la que se espera alcanzar y los valores que ésta estima importantes.

Diferenciando una marca

Los usuarios intentarán discernir las diferencias entre un sitio Web y sus competidores, y en particular las promesas que hacen y su habilidad para cumplirlas. A pesar de que un sitio Web se haga fuerte en diversas áreas u ofrezca más de un servicio, si se establece para el mismo un único e importante valor de marca, se conseguirá que los usuarios recuerden con mayor probabilidad ese valor de marca cuando les sea

6. DISEÑO DE LA INTERFAZ DE USUARIO

necesario. Sea lo que sea lo que diferencia un sitio Web de su competencia, debe traducirse en un mensaje que las personas recuerden.

Hemos podido comprobar cómo se está extendiendo el uso de la Web como soporte para publicar repositorios de patrones de diferente naturaleza. Un aspecto diferenciador con los mismos puede ser el carácter semántico del repositorio.

Incorporaremos este concepto distintivo al mensaje o promesa de nuestro sitio, que ya quedó establecido en el punto anterior y que quedará modificado como sigue:

“Repositorio Semántico de Patrones de Diseño Hipermedia”

Imagen de marca

Los usuarios identifican un sitio Web por la imagen de marca. Si un usuario accede a un sitio y no sabe bien dónde se encuentra, seguramente es porque la marca no es suficientemente prominente o quizá está relegada a un segundo plano por parte de otros elementos. Veamos algunas consideraciones fundamentales acerca de la imagen de marca:

- Usar exactamente las mismas fuentes, colores, gráficos, posiciones relativas, y proporciones en cualquier lugar que aparezca.
- Emplear un tamaño lo suficientemente grande para ser el segundo o tercer elemento que sea leído en la página.
- La ubicación es algo ya estandarizado; la esquina superior izquierda de la página.

A falta de ajustar el formato conforme al estilo que posteriormente se aplique al conjunto, esto es lo que será nuestra “imagen de marca”, con el mensaje del sitio Web ya integrado a la misma:

SEMANTIC HyPATTERNS

Repositorio Semántico de Patrones de Diseño Hipermedia

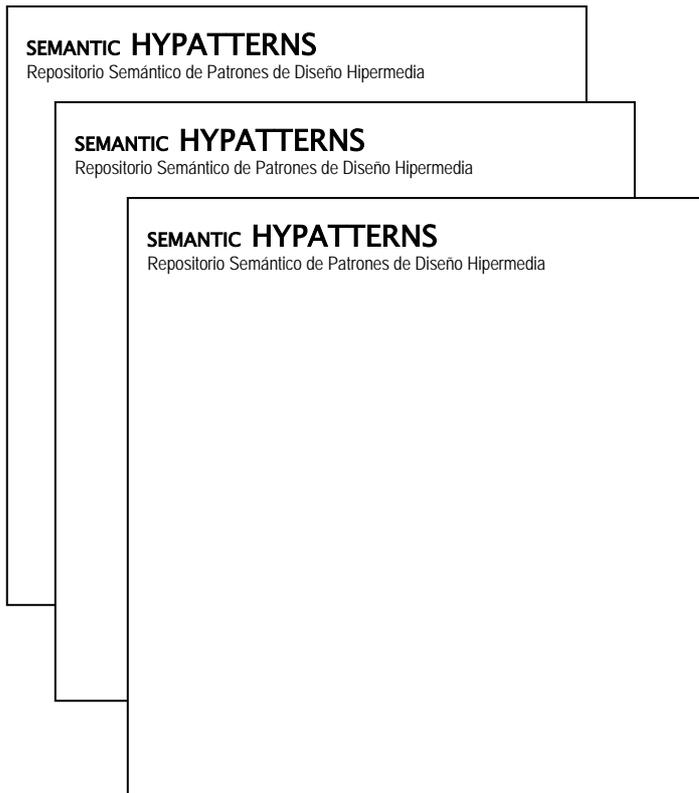


Figura 15: La imagen de marca estará presente en todas las páginas

6.2.8 Patrón *Page Templates* (D1)

Este patrón se encarga de sentar las bases para definir plantillas de contenido que proporcionen consistencia a las páginas de un sitio Web.

Problema:

Un sitio que no sea consistente entre página y página será difícil de explorar por parte de los usuarios y de mantener por parte de los diseñadores. Por tanto, cada una de estas páginas requiere de una plantilla que describa su contenido.

Construir plantillas tabuladas

Las personas leemos a lo largo de líneas verticales y horizontales. Usando líneas implícitas en el diseño que alineen los objetos y textos de las páginas, se agiliza la lectura de las mismas.

Seguiremos este modo de diseñar páginas haciendo de él la base de todas las plantillas del sitio → patrón *Grid Layout* (I1)

Definir plantillas globales e individuales

Hay que mantener la misma estructura de diseño gráfica base en todo el sitio. Así los usuarios recordarán donde aparecen las herramientas de navegación y el contenido en cada página, lo que da lugar a una mayor facilidad de uso.

Las plantillas globales describen la estructura y diseño general para todas las páginas del sitio. Esto incluiría cosas como la marca del sitio, las barras de navegación o los módulos de búsqueda.

Con la definición del *framework* de navegación y de la imagen de marca, podemos dar por definida la plantilla global para las páginas de nuestro sitio.

Las plantillas individuales se construyen sobre las plantillas globales, describiendo tipos específicos de páginas. Por ejemplo, se podrían tener plantillas individuales para cada sección, o para las páginas de descripción de productos.

En nuestro sitio podemos distinguir a priori las siguientes plantillas individuales:

- Categorías de patrones
- Listado de patrones por categoría
- Listado alfabético de patrones
- Listado de resultados de búsqueda de patrones
- Descripción de un patrón
- Enlaces externos a sitios relacionados
- Blogs o sección de comentarios

Se debe ser consistente en cómo se abordará el redimensionamiento de las páginas. Dos enfoques:

- Ajustar el ancho de la página al ancho de la pantalla

Esto se consigue empleando **porcentajes**. De esta forma el ancho de la página se convierte en algo relativo al porcentaje empleado en el diseño y al ancho de pantalla que emplee cada visitante.

La principal ventaja es la posibilidad de diseñar para todas las resoluciones sin necesidad de hacer diferentes versiones de la Web. Además son mejores para imprimir.

Entre las desventajas nos encontramos con que son más lentas de cargar. También con que se pierde el control sobre el ancho de línea, produciéndose así un salto de línea mayor en resoluciones altas, y por tanto líneas más largas, lo cual implica una lectura más lenta.

Por todo ello, este enfoque se ajusta mejor a sitios Web con muchos elementos de navegación y una amplia variedad de contenidos, ya que las líneas no serán largas y por tanto el ensanchamiento de la página tendrá un menor impacto.

- Emplear un ancho de página fijo

Esto se consigue estableciendo un **número de píxeles fijo** para el ancho de página.

La principal ventaja es que se consigue controlar a la perfección el ancho de línea para cualquier resolución.

Entre las desventajas, diremos que no se adaptan a las resoluciones de todos los usuarios, por lo que con resoluciones grandes se verá la Web más pequeña. Además suelen dar problemas a la hora de imprimir.

Al contrario que en el anterior enfoque, éste se adapta mejor a sitios Web con pocos elementos de navegación y donde en la mayor parte de las páginas se dedica a presentar un único texto, y donde por lo tanto interesa controlar lo más posible el ancho de línea.

Las páginas más importantes de nuestro sitio Web serán las dedicadas a la descripción de cada patrón del repositorio. Estas descripciones son textos relativamente grandes que ocuparán la totalidad del espacio de la página dedicado al contenido, para los que nos interesará mantener el ancho de línea. Además el resto de páginas tipo, tampoco mostrarán una gran variedad de contenidos.

Optaremos por ello por emplear un **ancho de página fijo**. Concretamente diseñaremos el sitio Web para una resolución de 800x600, de forma que probablemente abarcaremos un mayor número de usuarios. Para los que naveguen con una resolución de 1024x768 tampoco les resultará muy incómodo que las páginas no ocupen toda la pantalla.

6. DISEÑO DE LA INTERFAZ DE USUARIO

Los elementos clave deberán ubicarse en la mitad superior de las páginas.

Aplicaremos el patrón dedicado a este punto → patrón *Above the Fold* (I2).

Es conveniente proporcionar un camino para que los usuarios previsualicen las páginas de forma que les sea más fácil imprimirlas si ellos quieren.

Tal y como indicamos en uno de nuestros requisitos, proporcionaremos una opción de previsualización para las páginas de descripción de patrones → patrón *Printable Page* (D8)

6.2.9 Patrón *Grid Layout* (II)

Este patrón describe en qué consisten los diseños “cuadrículados” y cómo crear uno.

Problema:

No es fácil organizar de una forma cohesiva la gran variedad de elementos de una página Web.

“*Grid Layout*” es una técnica de diseño gráfico empleado en periódicos, revistas y otros documentos, que se puede usar para organizar el contenido de las páginas Web. Según esta técnica una página se divide en filas y columnas, y cualquier elemento se construye ajustándolo a esta estructura tabular. Prácticamente todas las páginas usan tablas HTML para implementar esta técnica. Ventajas del uso de *grids*:

- Da una estructura visual coherente, facilitando la búsqueda e identificación de elementos dentro de la página.
- Reduce la sensación de desorden, facilitando y agilizando la lectura de los contenidos.
- Una vez diseñado, el *Grid* puede ser reutilizado, lo cual permite centrarse más en los contenidos.

Usaremos tablas HTML para alinear y ajustar el contenido y las herramientas de navegación de nuestro sitio Web.

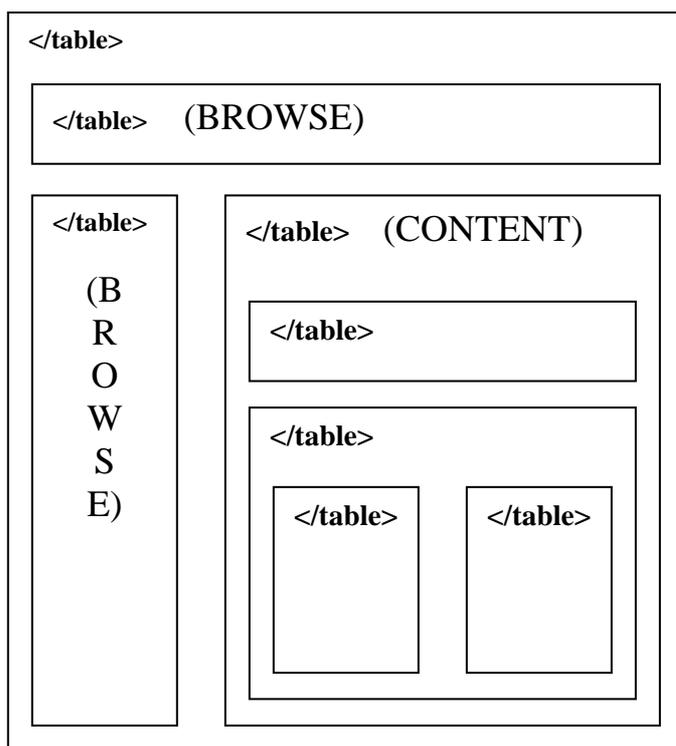


Figura 16: Aplicaremos la técnica *Grid Layout* en navegación y contenido

6.2.10 Patrón *Separate Tables* (L3)

Este patrón describe maneras de simplificar tablas y dividir las en otras más pequeñas para hacerlas más rápidas de visualizar en los navegadores.

Problema:

Las páginas Web con tablas HTML grandes y complejas son más lentas de visualizar en los navegadores Web.

El navegador debe calcular la altura y el ancho de todos los textos, imágenes, etc., antes de que algo sea visualizado. Como resultado se obtiene la impresión de que la página Web no está activa.

Por ello es recomendable dividir la página en tablas separadas en lugar de usar una única tabla para todos los elementos. De este modo se puede hacer que primero se visualicen ciertos elementos cercanos a la parte superior de la página mientras el resto de los elementos terminan de cargarse. Lo más importante a considerar al emplear tablas separadas es lo que debería ubicarse en la parte superior. Conviene tener en cuenta que no todo tiene que ir dentro de una tabla HTML. Por lo tanto puede ser beneficioso evaluar si se pueden sacar elementos de las tablas siempre que no afecte demasiado a la alineación de la página.

Mantendremos la imagen de marca y las herramientas de navegación superiores en tablas separadas de la tabla principal.

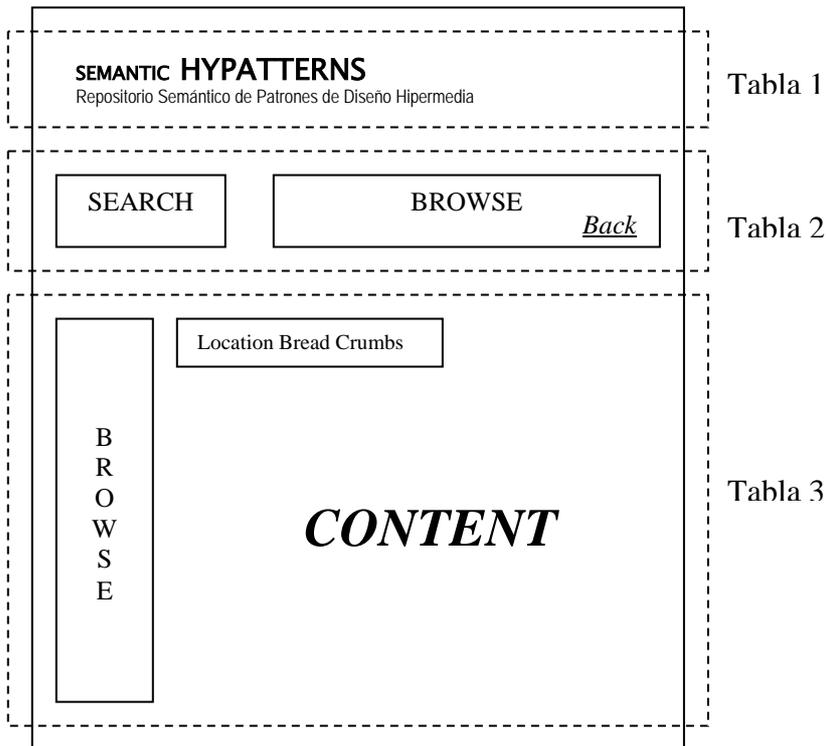


Figura 17: Tablas principales de las páginas de nuestro sitio Web

6.2.11 Patrón *Above the Fold* (I2)

Este patrón pretende explicar por qué la información de mayor relevancia debe ubicarse en la parte superior de una página Web.

Problema:

A menudo se puede perder información importante si ésta no está bien posicionada.

La expresión "*Above the fold*" se refiere a aquello que los usuarios de una página Web pueden ver sin necesidad de tener que hacer *scroll*. Como no se puede saber qué resolución de pantalla emplearán todos los usuarios, no se puede saber que verán.

Por ello los elementos más importantes se posicionan en la parte superior izquierda de la página, haciendo que estos elementos sean lo primero que verán todos los visitantes. Es un error grave, por tanto, el situar demasiada información de poca relevancia en esa zona de la página.

6. DISEÑO DE LA INTERFAZ DE USUARIO

A causa de esto, en la vida real las diferentes partes involucradas en el diseño de un sitio Web pretenden ubicar sus anuncios, principales productos, funcionalidades, o habilidades gráficas en dicha zona. El desafío consiste en combinar todas las necesidades y encontrar un diseño que satisfaga a todas las partes.

Dos decisiones a tomar:

- Elegir la mínima resolución de pantalla que soportará el sitio Web.
- Determinar los elementos que se ubicarán en la susodicha parte superior izquierda.

Para la resolución mínima de pantalla que seleccionemos, mantendremos todos los elementos de navegación visibles sin necesidad de hacer *scroll*. Los módulos de contenido siempre visibles dependerán de la plantilla individual a la que pertenezca la página.

6.2.12 Patrón *Fixed-Width Screen Size* (15)

En el apartado 6.2.8 referente al patrón *Page Templates*, se incluyeron las notas procedentes al presente patrón.

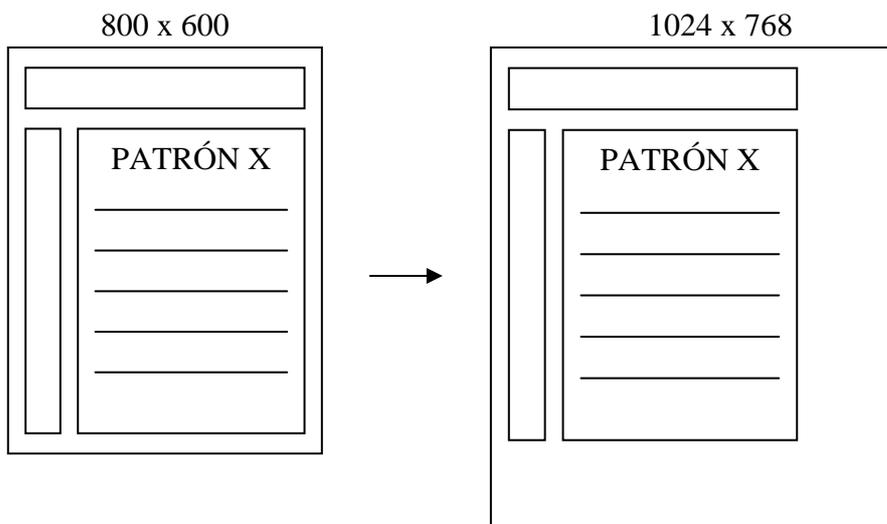


Figura 18: Emplearemos un ancho de página fijo con una resolución de 800x600

6.2.13 Patrón *Content Modules* (D2)

Este patrón describe los módulos de contenido, un componente clave de toda plantilla y una manera de gestionar el proceso de publicación.

Problema:

Sin un buen sistema, publicar y gestionar grandes volúmenes de contenido son procesos costosos en tiempo y proclives a errores.

Definir dónde posicionar el contenido en una página

Crear una plantilla con módulos de contenido hace más fácil añadir nuevas páginas e incluir contenido nuevo, evitando tener que idear continuamente cómo diseñar el documento.

Para dar a los usuarios información potencialmente útil que esté relacionado a lo que se está visualizando, es conveniente destinar un espacio en un lateral a modo de barras laterales de contenido relacionado. Para enganchar a los usuarios por medio de estos contenidos relacionados, se puede emplear las cabeceras de los mismos con pequeñas descripciones a modo de introducción.

Organizar el contenido con archivos o con bases de datos

Usar una base de datos para publicar contenido, evita tener que recodificar cada página en la que aparecen o se modifican contenidos. Puede emplearse un sistema de ficheros o una base de datos en sí misma, el efecto es el mismo. Todo depende de la cantidad de contenido del sitio o de las características que se quieran proporcionar a los clientes.

El modo en que publicaremos el contenido viene marcado por el carácter semántico de nuestro repositorio: el contenido se almacena en ficheros OWL y RDF.
--

Crear una página de administración

Construir una herramienta de publicación de contenido reduce tiempo y esfuerzo. Todo lo que se necesita es un formulario Web que incluya todos los campos de contenido, más la fecha de publicación y su ubicación dentro del sitio.

6.2.14 Patrón *Headlines and Blurbs* (D3)

Este patrón explica la necesidad de emplear cabeceras acompañadas de pequeñas descripciones, para introducir páginas más detalladas y atraer la atención del usuario.

Problema:

Las páginas de contenido necesitan cabeceras y resúmenes adecuados para inducir a los usuarios a que profundicen en los contenidos de un sitio.

Los usuarios están acostumbrados a ver cabeceras acompañadas de pequeños resúmenes, llamados *blurbs*, como forma de introducir elementos de información más detallada.

Las cabeceras son textos de no más de 10 palabras, de tal forma que pueden presentarse con un tamaño grande en poco espacio. Usar el título HTML de la página que referencia puede ser una buena solución si el título es suficientemente descriptivo.

Un *blurb* es una continuación de la cabecera, que proporciona información acerca de lo que es importante y exclusivo del contenido que referencia. Tienen que ser textos cortos y precisos. Pueden construirse a partir de las primeras líneas de un texto mayor, o bien ser resúmenes del mismo.

El empleo de cabeceras y *blurbs*, es una buena elección para emplear en muchas partes de un sitio Web, desde los módulos de contenido de la página de inicio hasta en barras laterales de información relacionada.

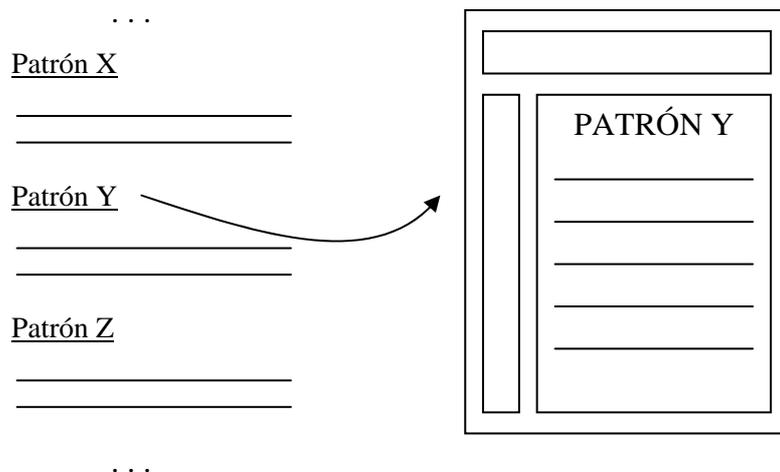


Figura 19: Emplearemos *headlines* y *blurbs* para listar patrones

6.2.15 Patrón *External Links* (K8)

Este patrón explica algunas técnicas para organizar los enlaces a sitios externos.

Problema:

Los enlaces a sitios externos necesitan ser tratados de forma especial de manera que los usuarios comprendan que éstos te dirigen a otros sitios Web que no son competencia del actual.

Considerar tener enlaces a sitios Web externos

Proporcionar a los usuarios información útil adicional que no se dispone en el propio sitio, ayuda a fortalecer la confianza y credibilidad respecto al mismo. Los usuarios apreciarán la existencia de referencias a otros sitios que les mantengan mejor informados.

Dedicaremos una sección de nuestro sitio Web a facilitar enlaces a información relacionada con el diseño hipermedia y los patrones.

Abrir nuevas ventanas para enlaces a sitios Web externos

Hay que evaluar el hecho de abrir una nueva ventana cada vez que el usuario pulse un enlace a un sitio externo. Las ventanas de *pop-up* pueden dar problemas, pero usadas convenientemente pueden resultar muy apropiadas.

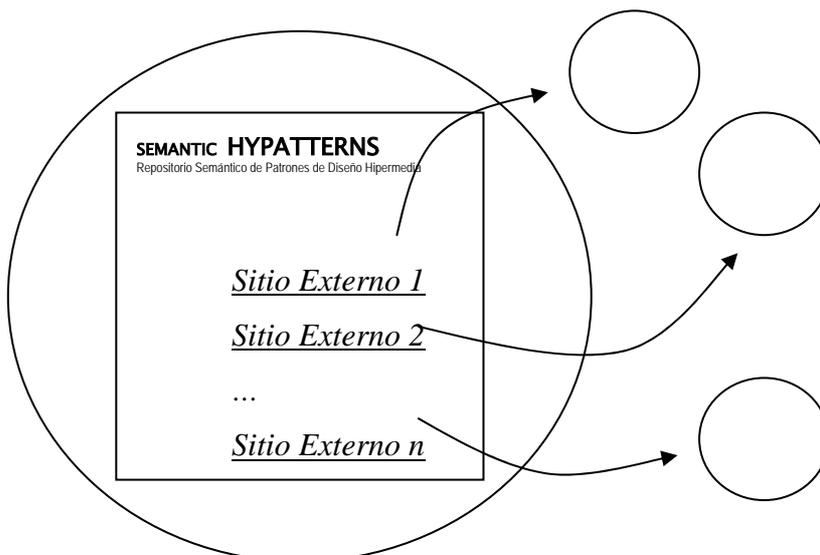


Figura 20: Se captará una mayor atención al facilitar enlaces a sitios relacionados

6.2.16 Patrón *Pop-up Windows* (H6)

Este patrón describe cómo las ventanas de *pop-up* pueden ser empleadas para determinados propósitos.

Problema:

Se puede necesitar mostrar al usuario cierta información adicional, mientras se mantiene el contexto y el navegador del usuario sobre la misma página.

Ventanas de *pop-up* automáticas

Las ventanas de *pop-up* automáticas aparecen como el resultado de llegar o salir de un determinado sitio Web. Este tipo de ventanas de *pop-up* se suelen emplear para mostrar información no relacionada, como pueden ser anuncios publicitarios o encuestas para mejorar la calidad del sitio Web.

Ventanas de *pop-up* basadas en enlaces

Una ventana de *pop-up* basada en un enlace aparece como resultado del pulsado de un enlace por parte de un usuario. Este tipo de ventanas de *pop-up* se suelen emplear para mostrar información relacionada en una nueva ventana, desde enlaces embebidos en texto, enlaces a sitios externos y procesos en cadena.

En un principio, en nuestro sitio tan sólo tendrá sentido el uso de ventanas de *pop-up* basadas en enlaces, fundamentalmente para la sección de enlaces a sitios externos relacionados. También para mostrar ventanas de ayuda en el caso de necesitarlas, o para dar a conocer las políticas y normas del sitio.

Usar ventanas de *pop-up* de forma moderada

Es conveniente minimizar el número de ventanas de *pop-up* de un sitio Web para evitar que inunden la pantalla y frustren a los usuarios.

Una idea puede ser limitar el uso de *pop-up* automáticos sólo cuando un usuario llegue al sitio y cuando lo abandone. En el caso de los *pop-up* basados en enlaces, una idea es reutilizar *pop-up* abiertos anteriormente. Esto reduce a dos ventanas de explorador abiertas como máximo en un momento dado.

6.2.17 Patrón *Descriptive, Longer Link Names* (K9)

Este patrón describe cómo los nombres largos y descriptivos pueden guiar a los usuarios al lugar deseado fácilmente.

Problema:

Los hiperenlaces de texto deben ser predecibles y comprensibles en cuanto a las páginas a las que enlazan. En otro caso, los usuarios constantemente se dirigirán a páginas que no les interesan, obligándoles a regresar e incrementando su frustración.

Términos genéricos dan lugar a nombres de enlaces pobres

Hay que evitar emplear términos genéricos para los enlaces, tales como “*click here*” o “*download*”. Lo ideal es que un usuario que elija un enlace aleatoriamente pueda intuir que tipo de cosas verá si lo pulsa.

Nombres largos de enlaces mejoran la accesibilidad del sitio

El uso de nombres descriptivos y largos para los enlaces pueden fortalecer la accesibilidad del sitio de tres formas:

- Son más fáciles de pulsar que los nombres cortos.
- Son más útiles para las personas ciegas, ya que a menudo saltan el contenido de un sitio y escuchan sólo los enlaces de una página Web.
- Ayudan a aumentar la confianza del usuario de que la página que van a descargar es la que ellos esperan.

Nombres largos de enlaces hacen una página más fácil de hojear

Los nombres largos de enlaces hacen que textos extensos sean más fáciles de hojear. A menudo los usuarios exploran un texto en busca de enlaces embebidos. Nombres más descriptivos para los enlaces pueden ayudarles a comprender de qué trata el texto y encontrar el enlace adecuado.

Usar lenguaje familiar

Hay que evitar los tecnicismos, porque un usuario no tiene por qué saber qué es lo que significan. De aquí se concluye la utilidad de usar nombres de enlaces que todo el

6. DISEÑO DE LA INTERFAZ DE USUARIO

mundo usa, como por ejemplo “*home*” para la página de inicio, o “*search*” o “*find*” para la página de búsqueda.

Diferenciar o eliminar los enlaces con nombres similares

Enlaces que suenan similares, pueden resultar confusos al usuario. Esto se hace más patente en sitios Web grandes, en los cuales a menudo nombran la misma cosa de forma diferente en diferentes páginas.

Separar enlaces con nombres cortados

En ocasiones nos podemos encontrar con nombres de enlaces cortados, dando la impresión de que un único enlace son varios. Por tanto hay que asegurarse que cualquier grupo de enlaces con nombres largos están claramente diferenciados los unos de los otros.

Dentro de las secciones de contenido de nuestro sitio nos podremos esperar dos grupos principales de enlaces; aquellos que servirán para navegar por el repositorio de patrones, y los empleados para dar acceso a sitios externos de información relacionada. Para los primeros lo más intuitivo es emplear los nombres de los patrones y los de sus categorías. Por tanto, el uso de nombres largos y descriptivos tan sólo cobrará sentido para los enlaces a sitios externos.

6.2.18 Patrón *Obvious Links* (K10)

Este patrón describe las ventajas de emplear enlaces de color azul y subrayados, y discute algunos de los inconvenientes.

Problema:

No siempre queda claro qué trozos de texto son enlaces.

Los enlaces azules han sido parte de la Web desde sus comienzos, y los usuarios esperan que los enlaces que no han visitado sean azules y subrayados. La mayoría de los sitios todavía siguen esta convención.

Este tipo de enlaces no son los más atractivos visualmente, pero la experiencia muestra que la facilidad de uso es un factor crítico. Esto no quiere decir que sólo haya que emplear enlaces azules en un sitio Web, sino que lo ideal es buscar un equilibrio adecuado entre éstos y el resto de enlaces. Se listan a continuación una serie de consejos relacionados:

6. DISEÑO DE LA INTERFAZ DE USUARIO

- Hay que evitar usar texto de color azul para otra cosa que no sean enlaces Web.
- Hay que evitar usar subrayado para otra cosa que no sean enlaces Web.
- Se debe jugar con el tamaño y estilo de las fuentes para hacer más atractivos los enlaces.
- Hay que considerar el uso del atributo TITLE de HTML con los enlaces Web, para anotar los mismos con una descripción.
- Se debe evitar emplear colores relacionados a deficiencias visuales, tales como el rojo o el verde.

6.2.19 Patrón *Printable Pages* (D8)

Este patrón proporciona la solución para imprimir cualquier página Web.

Problema:

En ocasiones los usuarios quieren imprimir lo que ven en sus pantallas. Éstos se verán frustrados si la impresión de una página Web corta contenido, saca páginas y páginas de contenido irrelevante, o no ofrece una versión imprimible.

Modificar cualquier página del sitio Web para adecuarla a su posible impresión requiere numerosos cambios que además restringen las opciones de diseño. Por ello, resulta mucho mejor crear una página alternativa con el mismo contenido pero más apropiada para imprimir.

Eliminar los elementos de navegación y contenido extraños

Hay que eliminar de la plantilla de impresión, todos los marcos, barras laterales de contenido relacionado, y barras de navegación laterales, porque no son en absoluto útiles en un formato imprimible.

Si nos encontramos con artículos que se extienden a lo largo de varias páginas, conviene que la versión imprimible reúna todas esas páginas en una sola, ya que en caso contrario su impresión se convertiría en algo tedioso.

Añadir etiquetas que ayuden a encontrar de nuevo la página online

Las versiones imprimibles de las páginas son una buena oportunidad para transmitir información de utilidad a los usuarios, como puede ser el título, el autor, la fecha y la URL de la página. También es común incluir uno o dos anuncios publicitarios.

Extraer el contenido principal de las tablas HTML

Las tablas HTML y los diseños tabulares son una de las principales fuentes de problemas a la hora de imprimir una página. A menos que el contenido tenga que estar formateado de forma tabular, hay que asegurarse que el contenido no está ubicado dentro de una tabla en la versión imprimible de la página.

Facilitaremos versiones imprimibles exclusivamente para las páginas de descripción de patrones. Este patrón cubre así uno de los requisitos marcados durante la fase de análisis. A la versión imprimible de un patrón le añadiremos la URL del sitio y la fecha de publicación del mismo.

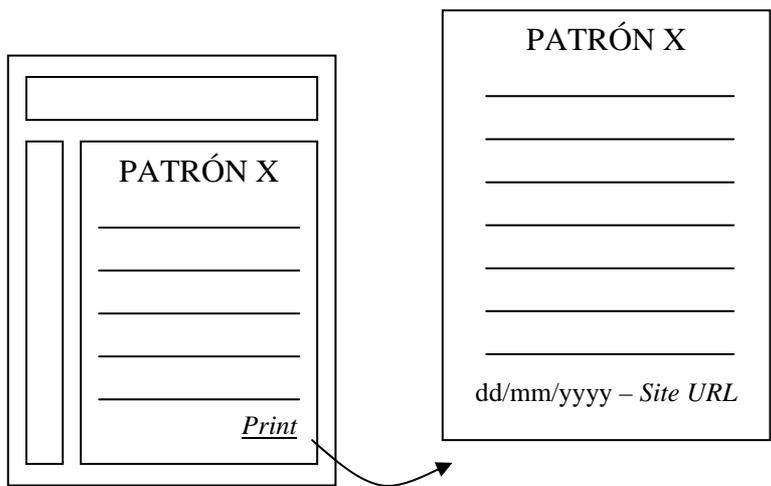


Figura 21: Las descripciones de patrones dispondrán de una versión imprimible

6.2.20 Patrón *Low Number of Files (L1)*

Este patrón puede usarse para agilizar la descarga de las páginas de un sitio Web.

Problema:

Las páginas Web con muchas imágenes, archivos de audio, *applets*, y *plug-ins* son lentas de descargar.

Como diseñador se plantea un dilema: por un lado se quiere jugar con muchos elementos visuales como pueden ser el logo, las barras de navegación, un fondo de página atractivo, uno o dos anuncios, algunas imágenes,... Sin embargo esto choca frontalmente con la velocidad de descarga de la página, con lo que ello supone. La idea es minimizar el número de archivos que deben transferirse. Se dan cuatro enfoques:

1. Eliminar imágenes, archivos de audio, applets, etc., que sean innecesarios: si la ausencia de un determinado archivo de la página no hiere al sitio, lo mejor es suprimirlo de la misma. Todo elemento extra aleja la atención de lo realmente importante y ralentiza la descarga de la página.
2. Usar las posibilidades de HTML en vez de imágenes, donde sea posible: esta idea se basa en que con HTML podemos definir botones, líneas y fondos, por lo que en estos casos no es imprescindible emplear imágenes, siendo HTML mucho más rápido en la descarga.
3. Reutilizar imágenes: los navegadores Web cachean en archivos temporales lo que ya ha sido descargado. Las imágenes que sean reutilizadas no tendrán que cargarse de nuevo.
4. Redireccionar los ficheros más pesados de las páginas que necesitan ser rápidas: los ficheros más pesados se pueden redireccionar a páginas independientes enlazadas. De esta forma los usuarios previsualizan dichos ficheros, y es decisión suya si quieren o no verlo con detalle.

Es importante priorizar las páginas del sitio Web, para centrarse en incrementar la rapidez de carga de las páginas más relevantes.

6.2.21 Patrón *HTML Power* (L4)

Este patrón describe características de HTML que ayudan a reducir el número de imágenes de una página.

Problema:

Las imágenes son elementos críticos dentro de un buen diseño de un sitio Web ya que actúan como signos visuales sobre la forma de interactuar y sobre el

6. DISEÑO DE LA INTERFAZ DE USUARIO

modo en que la página está organizada. Sin embargo, las páginas Web con muchas imágenes son lentas en la descarga.

HTML tiene muchas características que pueden emplearse en lugar de imágenes. Por ejemplo, se podría aplicar cualquiera de los siguientes puntos:

- El atributo BGCOLOR en el cuerpo de la página para establecer un color de fondo en lugar de una imagen. Esto es extensible a cualquier tabla, logrando combinar colores que distinguan las diferentes partes de la página.
- Emplear texto en las barras de navegación.
- Con el *tag* UL podemos crear los *bullets* de las listas.
- Usar botones HTML en lugar de imágenes con enlaces.
- Con el *tag* HR se pueden crear líneas horizontales.

Con todo esto no se quiere decir que se prescindiera siempre de las imágenes, sino que es preferible aprovechar las capacidades del HTML en lugares que funcionalmente y estéticamente lo aceptan.

Debido al carácter y objetivos de nuestro sitio Web priorizaremos el uso de código HTML, prescindiendo en general de las imágenes.
--

6.2.22 Patrón *Location Bread Crumbs* (K6)

Este patrón describe las maneras de emplear los menús de miga de pan de forma efectiva.

Problema:

No es difícil que los usuarios se sientan en ocasiones perdidos durante la exploración de un sitio Web sin conocer en qué punto se encuentran en relación con otras páginas del sitio.

Los menús de miga de pan o rastreadores proporcionan localizadores que los usuarios pueden emplear como forma de navegación. Concretamente muestran el rastro de páginas por las que un usuario ha navegado desde la página de inicio hasta la actual, dando información acerca de la estructura del sitio Web.

6. DISEÑO DE LA INTERFAZ DE USUARIO

Cada página dentro del rastreador está hiperenlazada permitiendo a los usuarios dar marcha atrás rápidamente. Así mismo, permiten ver a los usuarios dónde se encuentran en relación con la página de inicio.

Para separar cada etiqueta se utilizan símbolos como '>', '/' y '|'. El más recomendable es '>', puesto que sugiere que un localizador apunta al siguiente nivel de detalle.

Ubicaremos un menú de miga de pan en todas las páginas en la esquina superior izquierda de la zona reservada para el contenido, empleando el separador sugerido por el patrón ('>').

Home > Page 1 > Page 1-1 > Page 1-1-1

Figura 22: Formato de los menús de miga de pan que emplearemos en nuestro sitio

6.2.23 Patrón *Navigation Bar* (K2)

Este patrón describe cómo implementar las barras de navegación de forma que los usuarios las encuentren útiles.

Problema:

Los usuarios necesitan una forma estructurada y organizada de llegar a las partes más importantes de un sitio Web, que además sea fácil de comprender y usar.

Las barras de navegación son la forma más común de proporcionar acceso a las principales partes de un sitio Web. Hay tres tipos de barras de navegación:

- *Top-running*; se ubica a lo largo de la parte superior de una página Web. A menudo estas barras de navegación actúan como acceso de alto nivel, es decir, enlazando directamente con los diferentes subsitios o categorías.
- *Side-running*; se ubica a lo largo de la parte izquierda de una página Web. Es extremadamente raro ver una barra de este tipo a la derecha. Estas barras de navegación disponen de más espacio que las *top-running*. Normalmente muestran más categorías, a menudo actuando como acceso de segundo nivel, es decir, proporcionando enlaces dentro de un subsitio.

6. DISEÑO DE LA INTERFAZ DE USUARIO

- *Top-and-left*; se extiende a través de la parte superior y a lo largo de la parte izquierda de una página Web. A menudo la parte superior da acceso a los subsitios y la parte lateral profundiza en la navegación del sitio en concreto.

Los enlaces incluidos en las barras de navegación vienen dados en forma de texto o con iconos. Los iconos empleados de forma aislada no son efectivos porque no siempre son reconocidos de forma global a través de naciones o de diferentes regiones geográficas. Si se usan iconos es conveniente acompañarlos de una descripción textual.

Los contenidos de nuestro sitio Web no requieren el uso de dos barras de navegación. Emplearemos tan sólo una barra lateral izquierda para ubicar los accesos a las principales secciones de nuestro sitio Web. Si algunas de las secciones poseen subcategorías, se les dará también acceso desde esta barra hasta dos niveles

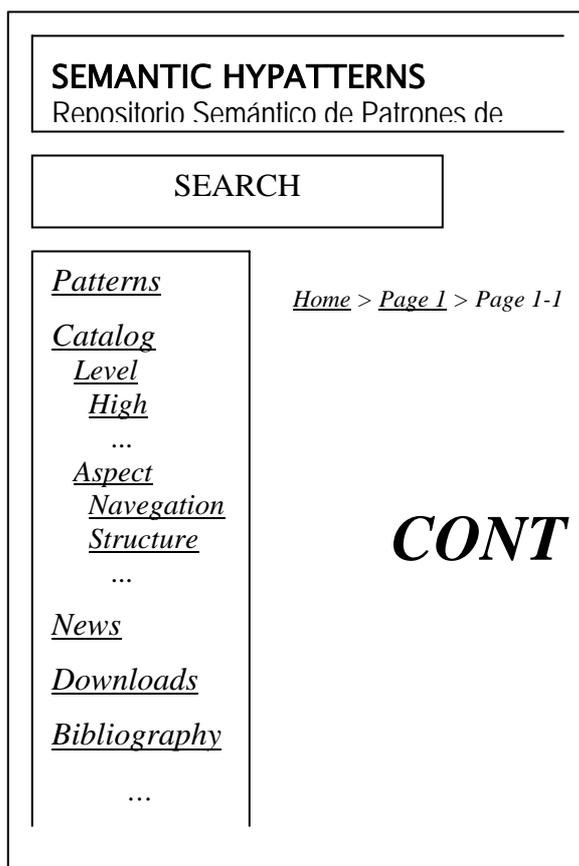


Figura 23: Una barra lateral izquierda para los accesos principales

6.2.24 Patrón *Consistent Sidebars of Related Content* (16)

Este patrón discute cómo usar una barra lateral para poner de relieve los contenidos relacionados en una determinada página Web.

Problema:

Una vez que un usuario explora el contenido de una página Web, a menudo termina queriendo acceder a elementos relacionados con el mismo.

Hay que encontrar el modo en el que presentar los enlaces a estos elementos de forma clara y evidente, logrando a la vez no distraer la atención sobre el contenido principal y sin ceder demasiado espacio. Esto se puede conseguir ubicando los enlaces relacionados en la parte inferior de la página. Sin embargo existe el riesgo de que algunos usuarios no lleguen a verlos, sobre todo si para llegar al final de la página hay que hacer demasiado *scrolling*. La mejor solución es el uso de barras laterales de contenido relacionado, las cuales se ubican en un lado cerca de la parte superior, normalmente en la parte derecha de la página, ya que las barras de navegación generalmente se ubican a lo largo de la parte superior e izquierda, y el contenido está generalmente ubicado en el centro. De esta forma los enlaces son fáciles de ver y se distinguen visualmente del contenido.

Ubicaremos la barra lateral de acceso a contenido relacionado en la parte superior derecha del espacio destinado al contenido, únicamente en aquellas páginas en las que realmente sean necesarias.

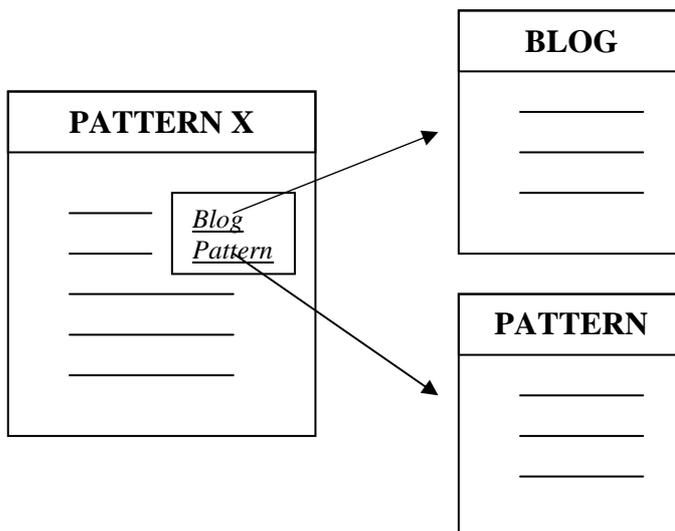


Figura 24: Desde un patrón accederemos a su blog y a los patrones relacionados

6.2.25 Patrón *Search Action Module* (J1)

Este patrón explica el resultado de combinar en Web los conceptos de “búsqueda” y “exploración”.

Problema:

A menudo el usuario desea saltar rápidamente desde una página a otra, pero los formularios de búsqueda son demasiado complejos para ello. Además, en ocasiones los usuarios conocen con exactitud lo que están buscando.

Una sencilla y rápida funcionalidad de búsqueda en la página de inicio consigue satisfacer al visitante. Esto se puede hacer extensible a cualquier página, ya que durante la exploración de un sitio Web puede surgir la necesidad de realizar una búsqueda determinada en cualquier momento.

Un simple texto que identifique el módulo de búsqueda y un botón que lance la búsqueda es suficiente para cubrir esta necesidad. Después habrá que proporcionarle una ubicación visible, en la mitad superior de la página, ya sea a la izquierda, en el medio o a la derecha.

En sitios Web complejos y ricos en contenidos puede resultar útil apoyarse en un combo de diferentes temas que acoten la búsqueda, para evitar así tener que enfrentarse a una página de resultados demasiado larga.

Los usuarios esperan encontrar la herramienta de búsqueda en el mismo lugar en todas las páginas. Por ello el módulo de búsqueda se debe ubicar en alguna zona de la mitad superior de las mismas, y mantenerlo ahí en todas las páginas.

Nuestro módulo de búsqueda rápida aparecerá en todas las páginas del sitio Web en una ubicación fija a lo largo de la parte superior. Se compondrá de una etiqueta identificativa, una caja de texto y un botón.

Además de esta facilidad, daremos la posibilidad de realizar búsquedas más detalladas → patrón *Straightforward Search Forms* (J2).

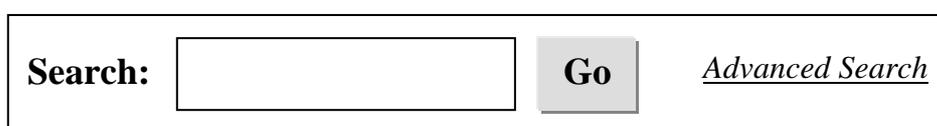


Figura 25: Esquema de nuestro módulo de búsqueda rápida

6.2.26 Patrón *Straightforward Search Forms* (J2)

Este patrón describe cómo evitar errores comunes en los formularios de búsqueda.

Problema:

Los formularios de búsqueda a menudo no resultan lo suficientemente efectivos debido a que requieren demasiada precisión. Están pensados para facilitar el procesamiento de la búsqueda más que para facilitar la búsqueda al usuario.

Las expresiones booleanas son difíciles de comprender

Las búsquedas booleanas usan los términos *and* y *or*. A pesar de que es fácil añadir estas cláusulas a un motor de búsqueda, también ha quedado demostrado que los usuarios las encuentran difíciles de comprender.

Hay que ayudar al usuario proporcionando un motor de búsqueda que no abuse de los operadores booleanos, o proporcionando formularios de búsqueda que tengan frases más explícitas.

Demasiada precisión resulta contraproducente

Algunos motores de búsqueda requieren que los parámetros de búsqueda sean exactos a lo que se quiere encontrar. De tal modo que ante la búsqueda sobre una palabra y sobre su plural da resultados diferentes.

Los nombres de productos también son fuente común de búsquedas frustradas, ya que los múltiples usuarios del sitio emplearán términos diferentes para referirse a la misma cosa, por lo que los sinónimos son un aspecto a tener en consideración.

La utilidad de búsquedas por defecto

Algunos formularios de búsqueda precisan que el usuario especifique una categoría sobre la que realizar la búsqueda. Si no la especificas puedes encontrarte con búsquedas que no dan resultados.

Para asegurar que los usuarios visualizarán algo, resulta de utilidad establecer la búsqueda sobre todas las categorías como acción por defecto.

Construiremos un mecanismo de búsqueda basada en la información semántica de los patrones para cada uno de los campos que los describen.

6.2.27 Patrón *Organized Search Results* (J3)

Este patrón da pautas para estructurar y organizar convenientemente los resultados de una búsqueda de forma que se tornen más fáciles de asimilar para los usuarios.

Problema:

Puede resultar difícil para los usuarios de un sitio comprender los resultados de una búsqueda si hay demasiadas o muy pocas entradas.

Proporcionar resúmenes relevantes

Los resultados de una búsqueda deben ir acompañados de un resumen relevante acerca de aspectos comunes a todos ellos. No hay una norma estricta para confeccionar este resumen, sino que dependerá de la naturaleza de dichos resultados el identificar los datos globales que puedan resultar de mayor interés.

A la hora de presentar los resultados de búsqueda se incluirá la siguiente información de cabecera:

- un resumen de los criterios de búsqueda utilizados por el usuario.
- el número total de entradas resultantes.

Organizar los resultados

No resulta muy apetecible el tener que recorrer una larga lista de resultados hasta dar con lo que se busca. Habrá que encontrar una manera lógica y coherente de agrupar los resultados, que facilite esta tarea.

Mostraremos los resultados de búsqueda en orden alfabético por nombre de patrón.

Proporcionar títulos hiperenlazados por cada resultado

Cada resultado tiene asociado un título hiperenlazado que te dirige a la página que contiene la información. Es importante seleccionar buenos títulos. Una buena práctica es la de la mayoría de los motores de búsqueda, que emplean el título HTML de la página como nombre del resultado de búsqueda.

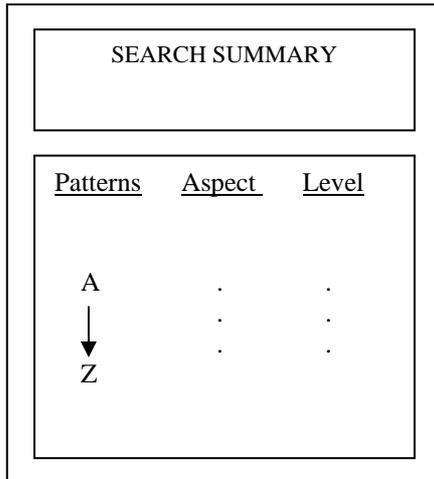


Figura 26: Los resultados se listarán alfabéticamente con un resumen de búsqueda

6.2.28 Patrón *Action Buttons* (K4)

Este patrón procura establecer una diferenciación entre enlaces ordinarios y enlaces de acción a través del uso de botones.

Problema:

Los hiperenlaces de texto son buenos para moverse entre páginas, pero no resultan muy representativos para las acciones que implican un procesamiento de información por parte del servidor Web.

Botones de acción gráficos

Este tipo de botones están contruidos con imágenes modificadas mediante recursos visuales que hacen parecer que pueden ser presionadas. Para ayudar a explicar su propósito, estos botones deberían estar acompañados de texto o etiquetas en la medida de lo posible. No deberían ser demasiado pequeños, o resultarían difíciles de ver y pulsar. Si son demasiado grandes se corre el riesgo de ocupar demasiado espacio.

Botones de acción HTML

Este tipo de botones codificados en HTML, son más rápidos de descargar, pero no proporcionan mucho control sobre el diseño y apariencia respecto al texto incrustado en el botón. La mejor opción es minimizar el texto de forma que el botón no se extienda más de lo deseado y poder posicionarlo con mayor libertad.

Optaremos por botones HTML, jugando con el tamaño del texto, el color del fondo, y el color de los bordes para proporcionar al botón una apariencia de 3D.

6.2.29 Patrón *Writing for Search Engines* (D6)

Este patrón proporciona la solución para escribir páginas Web que obtengan un *ranking* elevado por parte de los motores de búsqueda.

Problema:

Cuando se busca información en Internet relacionada con el sitio Web que se está diseñando, es fundamental que éste aparezca en las primeras posiciones de la lista de resultados, si queremos que los navegantes lo encuentren rápida y regularmente.

Los motores de búsqueda son las herramientas más populares para buscar páginas Web en Internet. Presentan listas de resultados de búsquedas que los usuarios exploran hasta que dan con el sitio Web que se ajusta a sus expectativas. Como lo normal es realizar la exploración partiendo de la cabeza de la lista, un sitio Web necesitará aparecer en las primeras posiciones. Los motores de búsqueda indexan archivos almacenados en servidores Web. Estos programas, llamados *crawlers* o *spiders*, parten de un pequeño conjunto de sitios Web, y recorren las páginas de enlace a enlace, recopilando e indexando información sobre los contenidos de las mismas. Cuando se busca información en los motores, éstos consultan su base de datos y presentan resultados clasificados por su relevancia o por la frecuencia de apariciones de palabras en una página o sitio. En cualquier caso, las palabras empleadas en el contenido de un sitio Web deben ajustarse a las palabras clave que los buscadores recogen del formulario de búsqueda.

Escribir títulos distintivos para todas las páginas

Emplear títulos HTML distintivos es importante de cara a los motores de búsqueda por dos razones. La primera es que muchos motores evalúan mejor las páginas Web si uno de los términos de búsqueda aparece en el título. La segunda razón es que a menudo los diseñadores de los buscadores toman los títulos HTML como la descripción de cada entrada mostrada en la lista de resultados de búsqueda. Si una página Web no dispone de título HTML, el motor tendrá que tirar de la etiqueta de un enlace o en el peor de los casos, de la URL de la página como descripción de la entrada.

Aplicaremos el patrón que se encarga de este aspecto → patrón *Distinctive HTML Titles* (D9)

Escribir texto descriptivo con palabras clave en la parte superior de cada página

Algunos motores de búsqueda asignan al texto situado en la parte superior de una página más peso que al texto situado cerca de la parte inferior. Por tanto, habrá que procurar emplear las palabras claves de un sitio Web en la mitad superior de las páginas.

Emplear el *tag* META

Los *tags* META son marcadores que indican a las aplicaciones software, incluyendo los motores de búsqueda, lo que un sitio o página Web contiene. Se aconseja usar estos *tags* en todas las páginas de un sitio Web para proporcionar palabras clave adicionales que no estén incluidas en los contenidos, incluyendo sinónimos, frases, y traducciones en otros idiomas.

Especificar el contenido que no se quiere que sea inspeccionado

El archivo estándar de exclusión “*robots*”, es una convención empleada por los motores de búsqueda para especificarles lo que se puede inspeccionar y lo que no. Es un archivo que se puede encontrar en la raíz de un sitio Web. El archivo “*robots*” es una forma adecuada de indicar a los spiders que eviten páginas tendentes a cambiar frecuentemente, páginas que requieren un logado previo, y archivos multimedia.

Estrategias ineficaces

La mayoría de los motores de búsqueda guardan secreto sobre el algoritmo que emplean para ponderar los sitios Web, para evitar el plagio y para evitar los trucos usados por algunos diseñadores en sus sitios. Uno de los trucos es añadir palabras claves muy populares pero para nada representativas del sitio, tales como “sexo” y “MP3”, que dirijan el tráfico al sitio al tener más probabilidad de aparecer en resultados de búsqueda. Otro truco es presentar un conjunto de páginas falsas a los motores, mientras se presentan otras a los visitantes del sitio. Este tipo de estrategias para mejorar el ranking en las búsquedas no es ético e incluso son perseguidas por los gestores de los servicios de búsqueda. Aquí se muestra una lista de estrategias a evitar:

- Palabras clave repetidas.
- Palabras clave que no describen el contenido del sitio.
- Palabras clave propiedad de otros sitios.
- Páginas falsas para los motores de búsqueda.

Hay otra serie de elecciones de diseño de otra naturaleza que pueden afectar a la hora de obtener un buen ranking:

- Páginas con el mismo título HTML.
- Páginas con el mismo contenido pero con URLs diferentes.
- Contenido oculto en CGI, Java o JavaScript.
- Contenido oculto tras formularios de logado.
- Contenido oculto en bases de datos.
- Contenido pensado para ser generado dinámicamente (ASP, JSP), o páginas que tienen URLs con caracteres especiales.

6.2.30 Patrón *Distinctive HTML Titles* (D9)

Este patrón describe las bases para construir un título HTML efectivo.

Problema:

Entre otros usos, los títulos HTML son usados por los motores de búsqueda cuando muestran resultados de búsqueda. Sin embargo, es frecuente que los títulos de página no informen adecuadamente de los contenidos de la página.

Los títulos de página HTML se usan como marcadores en los navegadores Web, como accesos directos de escritorio, y por algunos motores de búsqueda al mostrar los resultados de búsquedas. Sin embargo, a menudo se emplean nombres ambiguos que no representan o que no dan evidencias del contenido de la página.

Es por ello que se han de emplear nombres distintivos para los títulos HTML, que hagan más fácil a los usuarios el encontrar la página adecuada. Además proporcionan nombres de enlaces muy apropiados para usar en otras páginas.

Lo más fácil sería dar el mismo título a todas las páginas de un mismo sitio Web. Sin embargo los visitantes no sabrían qué página elegir o en qué página estuvieron, por lo que hay que variar los títulos entre página y página.

Una buena práctica es emplear un sistema similar a los menús de miga de pan (*location bread crumbs*) para construir el título HTML de la página, basándolo en el *path* que un usuario tiene que seguir para llegar a la página.

6.3 Resumen

A medida que hemos ido aplicando las recomendaciones de los patrones de nuestro lenguaje, hemos ido extrayendo conclusiones que han terminado por convertirse en exigencias que requerimos para nuestro sitio Web, puesto que hemos sabido reconocer las ventajas y beneficios que supondrán para el mismo.

En la siguiente tabla se relaciona cada requisito del sitio Web con el patrón que se ha utilizado para resolverlo. A los requisitos definidos en la fase de análisis uniremos los requisitos extraídos durante la aplicación de los patrones.

Identificador	Descripción	Patrón que lo resuelve
R1	El sitio posibilitará la realización de búsquedas rápidas de patrones a partir de un texto libre.	J1 Search Action Module
R2	El sitio posibilitará la realización de búsquedas avanzadas a través del contenido semántico de los patrones del repositorio.	J2 Straightforward Search Forms
R3	La búsqueda rápida de patrones estará accesible en todas las páginas del sitio.	J1 Search Action Module
R4	Se presentará una estructura de navegación jerárquica basada en categorías para la exploración del repositorio de patrones.	B3 Hierarchical Organization
R5	El sitio mostrará el contenido de cada categoría de la estructura jerárquica del repositorio en páginas independientes.	B3 Hierarchical Organization
R6	Se paginará el contenido de cada categoría si éste se compone de más de 15 enlaces.	B3 Hierarchical Organization
R7	Se presentará una estructura de navegación en orden alfabético para la exploración del repositorio de patrones.	B5 Alphabetical Organization
R8	El sitio dará acceso a las descripciones completas de cada patrón del repositorio.	A6 Grassroots Information Sites
R10	El sitio facilitará una versión imprimible de la descripción de cada patrón.	D8 Printable Pages

6. DISEÑO DE LA INTERFAZ DE USUARIO

Identificador	Descripción	Patrón que lo resuelve
R11	En la página de descripción de cada patrón se destinará un espacio de la misma para agilizar el acceso a los elementos relacionados con el patrón: blog, patrones relacionados y versión imprimible.	I6 Consistent Sidebars of Related Content
R13	El sitio complementará el repositorio de patrones con referencias externas a información relacionada en forma de enlaces.	A6 Grassroots Information Sites
R14	En función del tipo de información referenciada, las referencias externas a información relacionada se categorizarán en cuatro secciones independientes y accesibles de forma directa: Bibliografía, Descargas, Noticias y Otros repositorios.	A6 Grassroots Information Sites
R15	Se paginará el contenido de cada sección de referencias externas si éste se compone de más de 15 enlaces.	A6 Grassroots Information Sites
R20	Se tendrán en consideración los motores de búsqueda de Internet a la hora de implementar el sitio.	D6 Writing for Search Engines
R24	El sitio posibilitará una navegación fluida mediante el uso de diferentes barras y herramientas situadas convenientemente dentro del esqueleto de las páginas.	B1 Multiple Ways to Navigate
R25	Se combinarán los esquemas organizacionales más adecuados a las diferentes secciones del sitio.	B2 Browsable Content
R26	La página de inicio deberá ser capaz de transmitir claramente al usuario lo que el sitio puede ofrecerle.	C1 Homepage Portal
R27	Se diseñará la página de inicio con contenidos que llamen la atención del usuario y la retengan.	C1 Homepage Portal
R28	El sitio quedará ligado a un mensaje que logre transmitir cuál es el cometido del mismo.	C2 Up-front Value Proposition
R29	Se diseñará una imagen de marca que actúe como elemento unificador del sitio y que logre comunicar el valor diferenciador del mismo.	E1 Site Branding
R30	Hay que familiarizar al usuario con la exploración del sitio a	D1

6. DISEÑO DE LA INTERFAZ DE USUARIO

Identificador	Descripción	Patrón que lo resuelve
	través de una estructura y un diseño gráfico consistentes para todas las páginas del sitio.	Page Templates
R31	Los módulos de las páginas se dispondrán de forma alineada siguiendo un diseño tabular.	I1 Grid Layout
R32	La carga de las páginas será progresivo, de forma que el usuario obtenga respuesta sin tener que esperar a disponer de toda la página de una vez.	L3 Separate Tables
R33	El redimensionamiento de las páginas priorizará el mantenimiento del ancho de línea.	I5 Fixed-Width Screen Size
R34	Los enlaces a patrones o enlaces a información relacionada en el contenido irán acompañados de una pequeña descripción que ayude al usuario a saber dónde conducen.	D3 Headline and Blurbs
R35	Los enlaces a sitios externos se abrirán en ventanas independientes.	H6 Pop-up Windows
R36	Los enlaces a sitios externos estarán formados por nombres descriptivos.	K9 Descriptive, Longer Link Names
R37	Se procurará minimizar el tiempo de descarga del sitio en general y la página de inicio en particular.	L1 Low Number of Files
R38	El usuario debe poder saber dónde se encuentra en cada momento respecto de la página inicio, y debe poder regresar con facilidad a páginas previas.	K6 Location Bread Crumbs

Tabla 6: Relación requisitos del sitio - patrones utilizados

7 DISEÑO DE LA APLICACIÓN WEB

La fase de análisis finalizaba con la especificación de requisitos del sitio Web, a partir de la cual hemos afrontado el diseño de la interfaz de usuario. Los requisitos serán también el punto de partida de la fase de diseño de la aplicación Web, la cuál dará respuesta a las acciones del usuario y alimentará las páginas que componen la interfaz de usuario del sitio Web con la información requerida. El presente capítulo trata de explicar esta fase, exponiendo la arquitectura de la aplicación y la distribución del negocio en clases y paquetes.

Durante el planteamiento del problema propusimos aplicar las líneas marcadas por el patrón arquitectural MVC para definir la arquitectura global de nuestra aplicación Web. Además la decisión de utilizar el framework de desarrollo Jena para realizar el acceso a la información semántica del sitio nos llevó a decidir usar tecnología Java para la construcción de la aplicación, y esto a su vez, debido a la naturaleza Web de la misma, nos conduce al uso de las especificaciones definidas por la plataforma J2EE tales como *Servlets* y *JavaServer Pages*. En el apartado 7.1 exponemos gráficamente la arquitectura resultante en forma de diagrama de bloques, y resumimos cada una de las partes que la componen.

El apartado 7.2 lo dedicaremos a exponer la parte del Modelo y la parte del Controlador en forma de clases. Explicaremos la función de cada una de las clases participantes y nos apoyaremos en diagramas para exponer visualmente las relaciones entre las mismas. Hemos tratado de distribuir las clases en paquetes en función de la naturaleza y cometido de cada una de ellas. Por ejemplo, los componentes Web (como los servlets) irán todos incluidos en un mismo paquete, y las clases encargadas de acceder al repositorio de patrones en otro diferente.

7.1 Arquitectura

Teniendo en cuenta la tecnología seleccionada para implementar la aplicación, conocido el modelo de datos, y aplicando el patrón MVC, la arquitectura final del sitio quedará tal y como vemos en la siguiente figura.

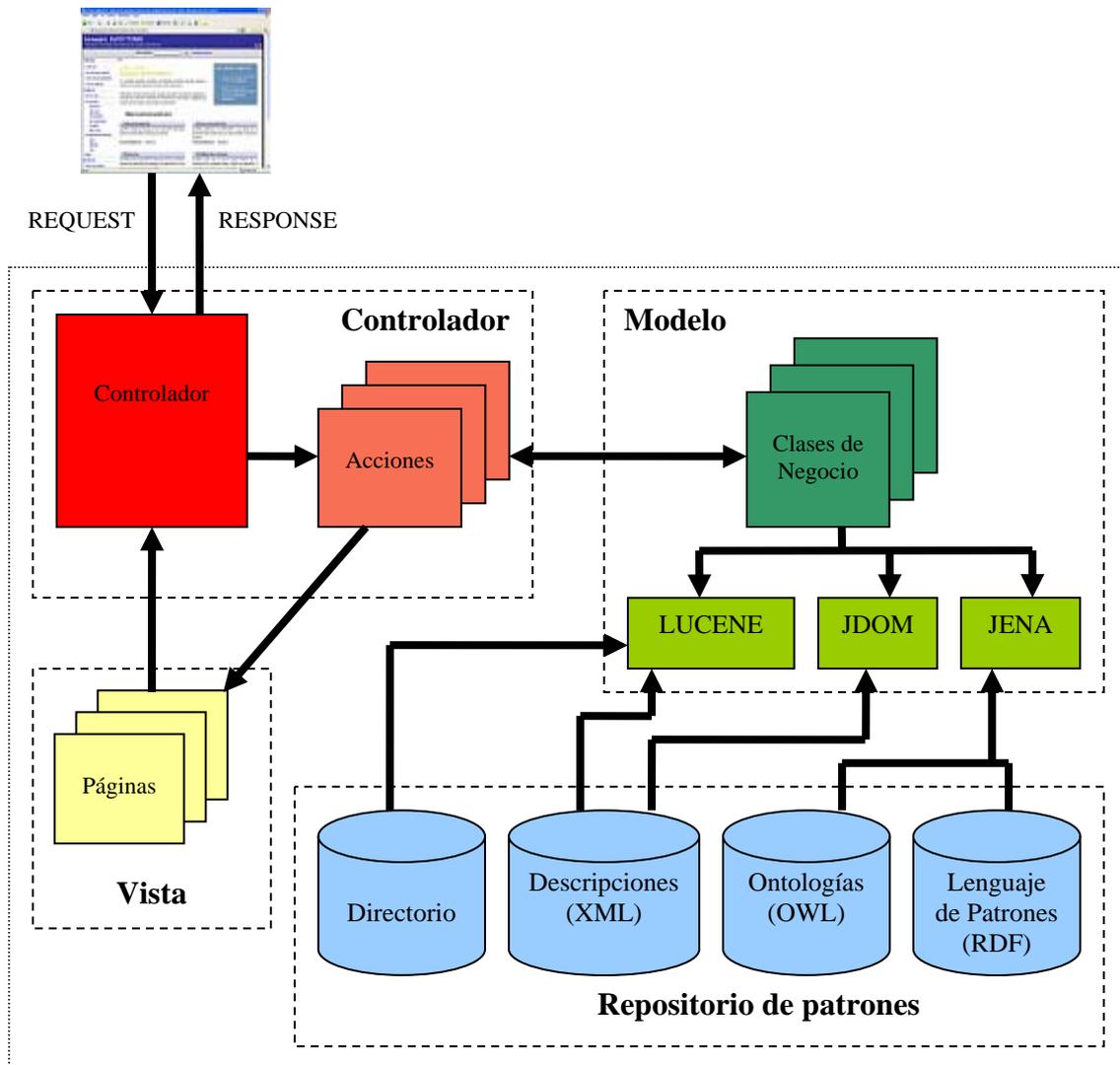


Figura 27: Arquitectura de la aplicación Web

Como se puede apreciar, cada vez que el usuario del sitio Web realiza una petición, la “clase controlador” recoge la misma e inicia la cadena de llamadas que darán lugar a la respuesta. Cada una de las diferentes peticiones que puede realizar el usuario se ligará a una “clase acción”. El controlador conocerá a qué acción se corresponde la petición que le llega, y le delegará la responsabilidad de tratar con las clases de negocio. Es decir, el controlador será el punto único de acceso a la aplicación, y será el encargado de distribuir el tráfico. La mayor parte de las acciones requerirán la

intervención de las clases de negocio para extraer información del repositorio, y llamarán a los métodos adecuados para obtenerla.

Las clases de negocio serán las encargadas, entre otras cosas, de acceder al repositorio de patrones y extraer la información requerida en cada momento. Para acceder a la información semántica del repositorio, es decir, para acceder a los ficheros RDF y OWL, estas clases utilizarán los métodos ofrecidos por el API Jena, y para tratar con los ficheros XML de las descripciones, emplearán los métodos del API JDOM. Por último, para la realización de consultas de patrones a partir de texto libre, se emplearán los métodos proporcionados por el API Lucene.

Una vez la acción reciba los datos, seleccionará el componente de la Vista que preparará la respuesta o la página adecuada conforme a lo definido en la interfaz de usuario. Por último, el controlador redirigirá la petición a la página generada, la cuál formateará debidamente la respuesta que el navegador mostrará al usuario.

7.2 Diagramas de clases

Definida la arquitectura, en este apartado nos centraremos en identificar y exponer las clases de la parte controladora y el negocio de la aplicación Web. El primer paso será distinguir los paquetes de primer nivel en los que se distribuirán las clases, para posteriormente ir aumentando el nivel de detalle e ir mostrando las clases contenidas. El siguiente diagrama muestra estos paquetes y sus dependencias.

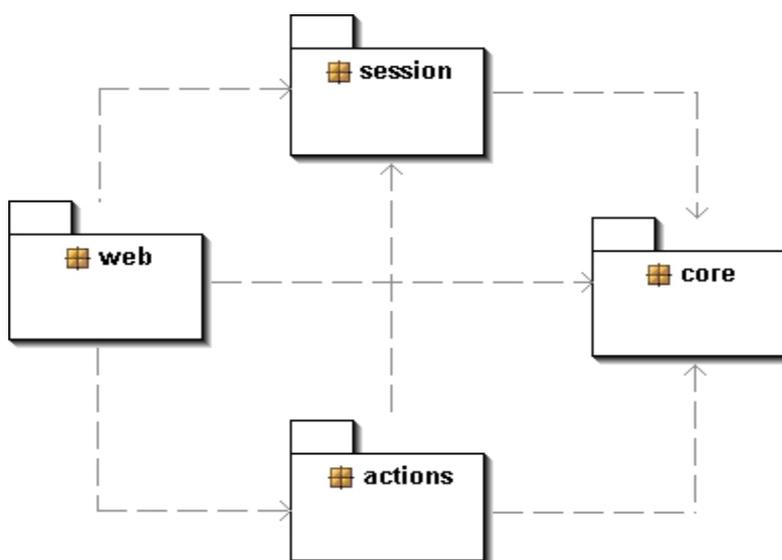


Figura 28: Diagrama de paquetes de primer nivel

7. DISEÑO DE LA APLICACIÓN WEB

A continuación analizamos con más detalle cada uno de los paquetes del diagrama anterior.

PAQUETE WEB

En este paquete se agruparán todas las clases de naturaleza Web (servlets, listeners,...) que sean necesarias. Aquí por ejemplo, quedaría incluida la “clase controlador” de la aplicación. Las clases de este paquete serán las que inicien cualquier diálogo con el resto de la aplicación, normalmente a partir de una petición del usuario y en ocasiones por una invocación por parte del servidor de aplicaciones (expiración de sesión, por ejemplo).

PAQUETE ACTIONS

Aquí agruparemos todas las clases “acción” de la aplicación invocadas por el la “clase controlador”. Como se dijo en el apartado anterior, existirá una acción por cada una de las posibles diferentes peticiones iniciadas por el usuario. Dentro podremos necesitar de un segundo nivel de agrupación para diferenciar las acciones en función del ámbito funcional de las mismas. Las acciones invocarán al método o métodos de las clases de negocio (paquete CORE) necesarios para preparar la respuesta que se mostrará al usuario.

PAQUETE SESSION

Será necesario almacenar información relativa a cada sesión de usuario. En este paquete incluiremos las clases que gestionen las diferentes sesiones de usuario abiertas en un momento dado. A la información de sesión podrán necesitar acceder las clases del paquete WEB y las del paquete ACTIONS.

PAQUETE CORE

Como su propio nombre indica, este paquete será el corazón de la aplicación en la medida en que la mayor parte del negocio estará aglutinado en las clases contenidas y en que será el punto de acceso a los datos del sitio Web. Aquí accederán tanto las clases del paquete SESSION para recabar información que se necesite asociar a las sesiones de usuario, a los componentes Web del paquete WEB, y a las clases del paquete ACTIONS para recuperar la información necesaria para satisfacer las peticiones del usuario.

7. DISEÑO DE LA APLICACIÓN WEB

Una vez identificados los paquetes de primer nivel en los que quedarán distribuidas todas las clases de la aplicación, vamos a empezar a aumentar el nivel de detalle. Para ello lo primero será identificar todas aquellas clases que puedan iniciar de algún modo una comunicación completa con el resto de la aplicación que, por lo visto anteriormente, serán las contenidas en el paquete WEB.

Igualmente en este paso será conveniente definir las clases que conformen el contrato que establecerá el negocio de la aplicación, atendiendo a los requisitos funcionales recogidos en la fase de análisis. En este punto aplicaremos el patrón Facade [Gamma *et al.*, 1995], de tal modo que identificaremos una interfaz unificada sencilla como único punto de acceso al negocio de la aplicación. Todas las llamadas al negocio de la aplicación pasarán primero por una llamada a alguno de los métodos ofrecidos por esta interfaz. Todo esto queda reflejado en el diagrama de clases de la figura.

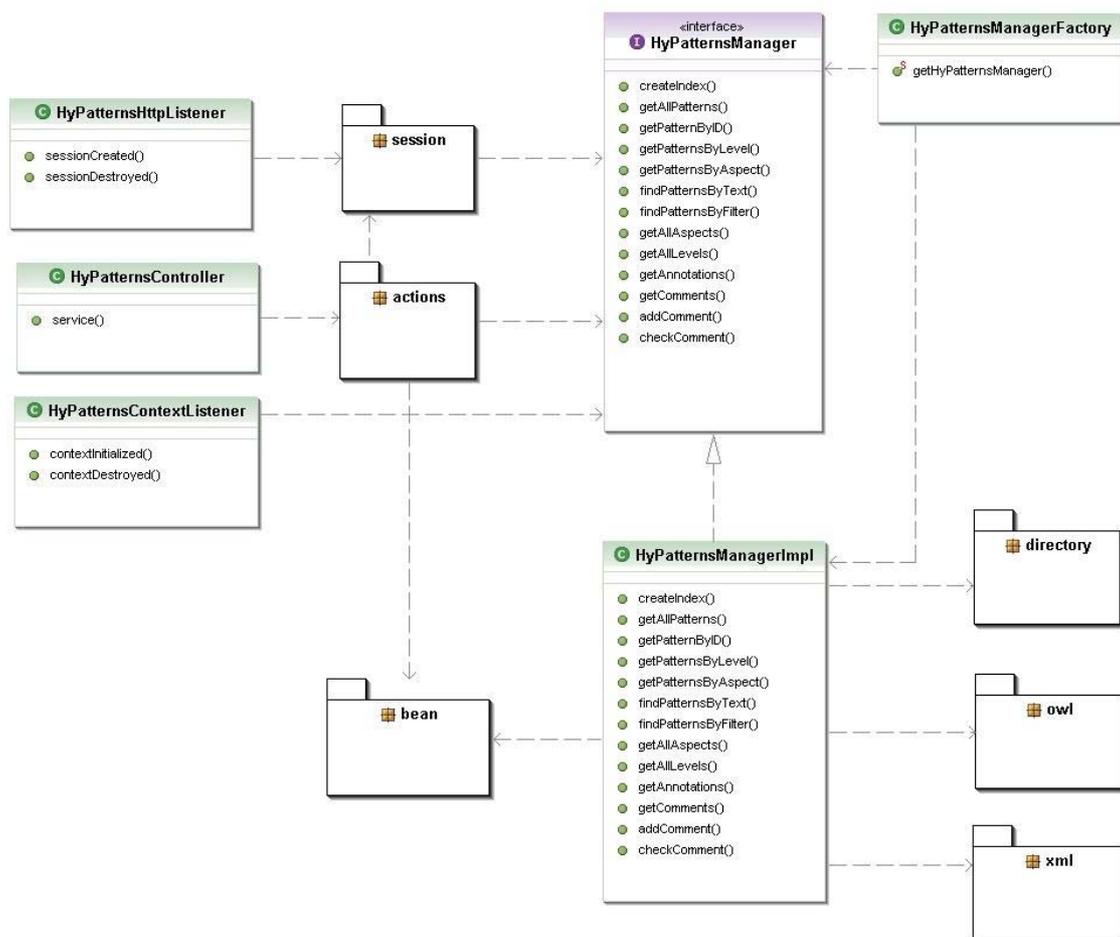


Figura 29: Diagrama de clases principal

7. DISEÑO DE LA APLICACIÓN WEB

A continuación analizamos con más detalle cada una de las clases y los nuevos paquetes del diagrama anterior.

HYPATTERNSCONTROLLER

Se trata de la “clase controlador” de la aplicación. Por esta clase pasarán todas las peticiones iniciadas por un usuario que navegue por el sitio Web, actuando como un controlador de tráfico que delega el foco a la acción apropiada en función de la petición.

A partir de un parámetro de petición “acción” que viajará en la llamada al controlador, éste identificará qué es lo que el usuario ha pulsado o solicitado. Tras esto, instanciará la acción que corresponda, e invocará su ejecución. Finalmente redirigirá la respuesta devuelta por la acción a una página de la Vista o a otra acción, o a la página de error en caso de fallo.

HYPATTERNSHTTPLISTENER

Clase “listener” para detectar el momento en que se crea/destruye una sesión de usuario. Esto nos servirá para realizar determinadas tareas que convenga hacer antes de que el usuario empiece a navegar por el sitio o una vez el usuario ha decidido abandonarlo.

Por ejemplo, será conveniente liberar la memoria ocupada por la información de sesión de un usuario cuando éste termine la misma o bien cuando expire el *timeout* definido en el descriptor de despliegue de la aplicación.

HYPATTERNSCONTEXTLISTENER

En la aplicación existirá determinada información que tendrá que estar disponible para todos los usuarios, desde el comienzo de su ejecución. La preparación de esta información la realizará esta clase “listener”, para detectar cambios en el contexto de la aplicación.

Por ejemplo, al arrancar el servidor de aplicaciones será necesario regenerar los índices de búsqueda de los XML de descripción de patrones, por si existe alguno nuevo o se han modificado los existentes.

HYPATTERNSMANAGER

Interfaz que representa el punto único de acceso a los métodos de negocio o modelo de la aplicación. Toda clase externa al paquete CORE que quiera obtener cualquier tipo de información almacenada en OWL, RDF, XML o en el directorio de búsqueda, deberá primero llamar a alguno de los métodos declarados en esta interfaz. Listamos a continuación cada uno de estos métodos:

- *createIndex()*; crea los índices de búsqueda invocando al método correspondiente de la clase `HyPatternsDirectory`.
- *getAllPatterns()*; retorna una colección con todos los patrones del repositorio. Para ello recorrerá cada uno de los RDF de patrones del repositorio y extraerá la información básica que describe un patrón con los métodos proporcionados por el API Jena.
- *getPatternByID()*; retorna la descripción completa de un patrón a partir del identificador del mismo. Para ello extraerá del RDF correspondiente la información básica del patrón y del XML de descripción asociado el contenido completo. Para el parseo del XML invocará al método correspondiente de la clase `PatternDescription`.
- *getPatternsByLevel()*; retorna una colección con todos los patrones que pertenecen a un determinado nivel de descripción.
- *getPatternsByAspect()*; retorna una colección con todos los patrones que pertenecen a un determinado aspecto de diseño.
- *findPatternsByText()*; retorna una colección con todos los patrones resultantes de hacer una búsqueda de un texto cualquiera sobre el directorio de búsqueda. Para ello invocará al método correspondiente de la clase `HyPatternsDirectory`.
- *findPatternsByFilter()*; lanza una búsqueda semántica sobre el repositorio para retornar la colección con los patrones que se ajustan a unas determinadas condiciones de búsqueda. Para ello construirá una consulta SPARQL y lanzará su ejecución por medio del motor ARQ del API Jena.

- *getAllAspects()*; retorna una colección con todos los aspectos de diseño en los que se clasifican los patrones del repositorio. Esta información queda recogida en la propiedad *Aspect* definida en la ontología, por lo que será necesario acceder a la misma a través de los métodos de acceso a OWL del API Jena. Posteriormente se invocará al método de la clase `AspectDescription` para obtener la descripción textual de cada aspecto.
- *getAllLevels()*; retorna una colección con todos los niveles de descripción en los que se clasifican los patrones del repositorio. Esta información queda recogida en la propiedad *Level* definida en la ontología, por lo que será necesario acceder a la misma a través de los métodos de acceso a OWL del API Jena. Posteriormente se invocará al método de la clase `LevelDescription` para obtener la descripción textual de cada nivel de descripción.
- *getAnnotations()*; carga las anotaciones encontradas en el repositorio para un campo determinado. Para ello recorrerá cada uno de los RDF de patrones del repositorio extrayendo el contenido de la propiedad correspondiente al campo.
- *getComments()*; retorna una colección con todos los comentarios “aceptados” asociados a un patrón. Para ello se apoya en el método correspondiente de la clase `Comments`.
- *addComment()*; añadirá un nuevo comentario a un patrón a través del método correspondiente de la clase `Comments`, y enviará un correo electrónico al administrador del sitio para informarle.
- *checkComment()*; aceptará o rechazará un comentario invocando el método correspondiente de la clase `Comments`, y enviará un correo electrónico al autor del comentario para comunicárselo.

HyPatternsManagerImpl

Esta es la clase que implementa la interfaz `HyPatternsManager`, y que contiene la mayor parte del negocio de la aplicación. La mayor parte del código que manipulará los ficheros OWL y RDF se realizará aquí. Otras tareas, como la

7. DISEÑO DE LA APLICACIÓN WEB

gestión del directorio de búsqueda o el parseo de los ficheros XML, serán delegadas a otras clases que veremos posteriormente.

HYPATTERNSMANAGERFACTORY

Clase factoría, invocada por alguna de las clases del paquete WEB para obtener instancias de la interfaz `HyPatternsManager`, para iniciar el diálogo con las clases de negocio.

PAQUETE BEAN

En este paquete se localizarán las clases de “estado”, que se corresponderán con las entidades del modelo identificadas en la aplicación. Por ejemplo, será necesario una clase que represente un patrón de diseño. De esta forma, cualquier método de la aplicación que retorne información acerca de un patrón del repositorio, deberá retornar una instancia de la clase “Patrón” con dicha información. O si se trata de devolver una lista de patrones, se deberá retornar una lista de objetos “Patrón”.

Dicho de otro modo, las instancias de las clases contenidas en este paquete serán los componentes de intercambio entre las distintas capas de la aplicación (exceptuando cuando se trabaje con tipos de datos primitivos). Por tanto, la estructura de las clases contenidas en este paquete serán conocidas por todas las clases de la aplicación, desde las clases del modelo que se encargarán de instanciarlas a la hora de preparar la información, hasta las páginas de la Vista para extraer la información contenida y presentarla de manera adecuada.

PAQUETE DIRECTORY

En este paquete se ubicarán las clases encargadas de gestionar y manipular el directorio de búsqueda de patrones. Será por tanto el punto único de diálogo con las clases del API Lucene.

PAQUETE OWL

En este paquete incluiremos clases de apoyo para la recuperación de datos incluidos en ficheros OWL y RDF. Cada una de las clases incluidas, representará una ontología o abstracción semántica empleada en la aplicación.

PAQUETE XML

Este paquete agrupará todas las clases de parseo y construcción de ficheros XML, por lo que será el punto único de diálogo con las clases del API JDOM. Habrá una clase por cada tipo de fichero XML que tome parte en el sitio Web.

Acabamos de identificar las tres clases principales del paquete CORE, y hemos definido para el mismo cuatro nuevos subpaquetes. A continuación vamos a entrar más en detalle en este paquete con el diagrama de la siguiente figura.

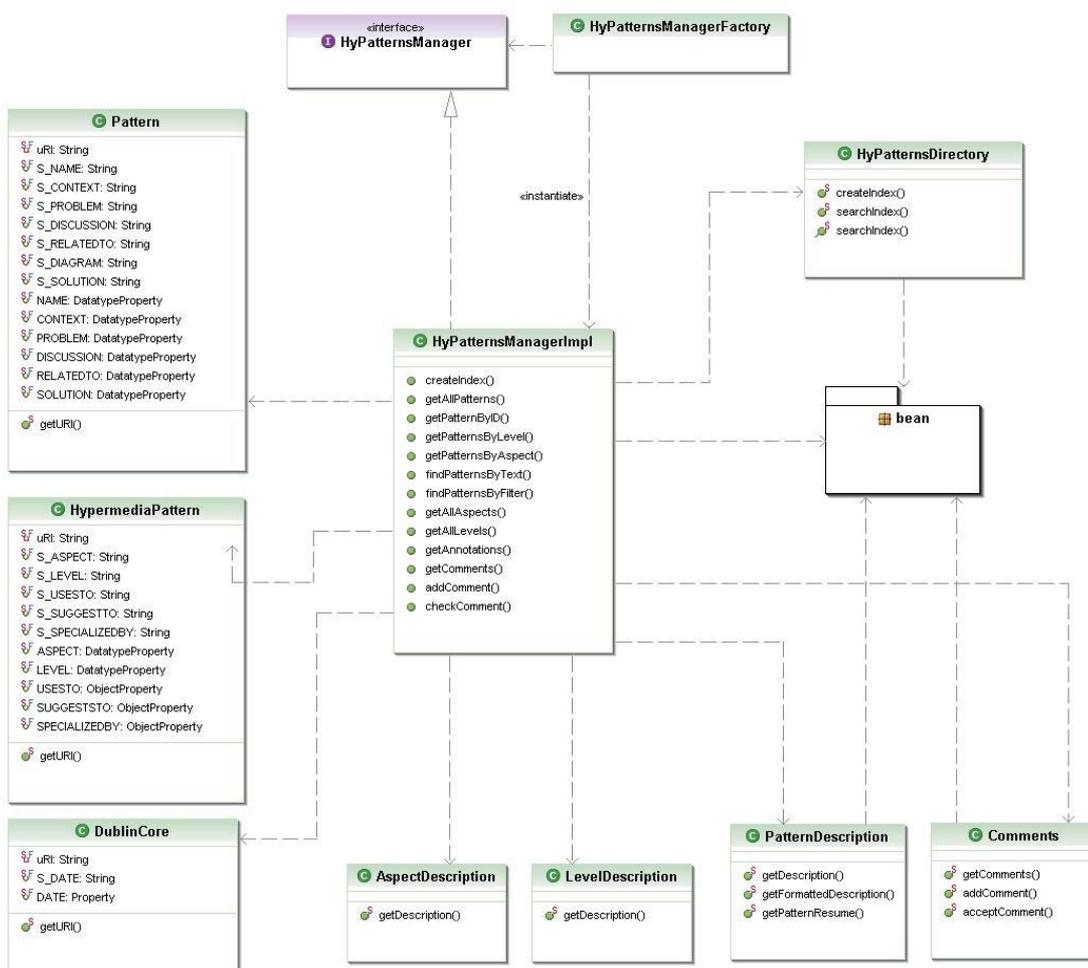


Figura 30: Diagrama de clases del paquete CORE

En el diagrama anterior, para no recargarlo demasiado no hemos mostrado las clases que componen el paquete BEAN. Las mostramos, ahora sí, en la siguiente figura.

7. DISEÑO DE LA APLICACIÓN WEB



Figura 31: Clases del paquete CORE.BEAN

A continuación analizamos con más detalle cada una de las nuevas clases incluidas en el paquete CORE.

HYPATTERNSDIRECTORY

Esta es la clase de gestión y explotación del directorio de índices de búsqueda de patrones. Los métodos públicos de esta clase serán los dos siguientes:

- *createIndex()*; este método será invocado para crear los índices de búsqueda a partir de las descripciones XML de los patrones del repositorio. Este método será invocado por el “listener” `HyPatternsContextListener` al detectar el arranque del contexto de la aplicación. Es decir, sólo se ejecutará al arrancar la aplicación. Si se quiere renovar el contenido del directorio, ya sea porque se ha eliminado algún fichero XML, se ha añadido uno nuevo, o se ha modificado alguno existente, será necesario tirar y levantar la aplicación.

Para crear los índices, el primer paso será instanciar un directorio en disco en la ruta que tengamos definida para ello. Posteriormente se indexa cada uno de los XML de descripción a partir de una instancia

7. DISEÑO DE LA APLICACIÓN WEB

de la clase `org.apache.lucene.document.Document` de Lucene, indicando no sólo el contenido a indexar sino también aquellos campos del patrón que nos interesará mostrar cada vez que se realice una búsqueda.

- `searchIndex()`; este será el método invocado por la clase `HyPatternsManagerImpl` cada vez que sea necesario lanzar una búsqueda de una cadena de texto sobre el directorio de búsqueda. Para ello hará uso del motor de búsqueda incluido en la clase `org.apache.lucene.search.IndexSearcher` de Lucene.

PATTERN

Esta clase representa la ontología definida por el fichero “Pattern.owl”. Será un reflejo de la ontología en memoria, para centralizar cierta información de utilidad para los métodos de la clase `HyPatternsManagerImpl` que necesiten acceder a través de Jena a la información contenida en esta ontología o a la información de las instancias de la ontología (es decir a la información contenida en los ficheros RDF de cada patrón).

HYPERMEDIAPATTERN

Esta clase representa la ontología definida por el fichero “HypermediaPattern.owl”. Será un reflejo de la ontología en memoria, para centralizar cierta información de utilidad para los métodos de la clase `HyPatternsManagerImpl` que necesiten acceder a través de Jena a la información contenida en esta ontología o a la información de las instancias de la ontología (es decir a la información contenida en los ficheros RDF de cada patrón).

DUBLINCORE

Esta clase representa el RDF correspondiente a los Dublin Core Elements. Dublin Core es un sistema de definiciones diseñado específicamente para proporcionar un vocabulario de características “base”. Será un reflejo del RDF en memoria, para centralizar cierta información de utilidad para los métodos de la clase `HyPatternsManagerImpl` que necesiten acceder a través de Jena a la

información de esta abstracción semántica que esté presente en los RDF de patrones, como es el caso de la fecha de publicación de un patrón.

ASPECTDESCRIPTION

Las descripciones de los diferentes “aspectos de diseño” en los que se clasifican los patrones del repositorio, se encuentran almacenadas en un fichero XML. Esta clase contendrá el método encargado de parsear este XML, para retornar la descripción asociada a un determinado aspecto.

LEVELDESCRIPTION

Las descripciones de los diferentes “niveles de descripción” en los que se clasifican los patrones del repositorio, se encuentran almacenadas en un fichero XML. Esta clase contendrá el método encargado de parsear este XML, para retornar la descripción asociada a un determinado nivel de descripción.

PATTERNDESCRIPTION

Como ya se ha dicho, los ficheros RDF de patrones del repositorio no contienen las descripciones textuales completas de los campos que describen el patrón, sino anotaciones con información semántica extraídas de las mismas. Las descripciones se almacenan por separado en ficheros XML. Esta clase se encargará de parsear estos XML para extraer las descripciones. Se identifican dos métodos principales:

- *getDescription()*; extraerá el contenido textual de cada uno de los campos que describen un patrón tal cual viaje en el XML.
- *getFormattedDescription()*; extraerá el contenido textual de cada uno de los campos que describen un patrón formateando convenientemente aquellos campos donde se hacen referencia a otros patrones de modo que cuando se vaya a presentar la información estas referencias se localicen fácilmente para tratar su presentación de modo diferente.

COMMENTS

Por cada patrón del repositorio, existirá un fichero XML conteniendo los comentarios que los usuarios añadan desde la sección de comentarios del sitio Web. Esta clase contendrá los métodos encargados de parsear y manipular el contenido de estos XML. Se identifican tres métodos principales:

- *getComments()*; retornará una lista con todos los comentarios “aceptados” asociados a un patrón.
- *addComment()*; añadirá un nuevo comentario al XML de comentarios del patrón correspondiente, a falta de ser aceptado o rechazado.
- *acceptComment()*; este método se invocará cuando el administrador del sistema decida marcar un comentario como aceptado o como rechazado.

PATTERNBEAN

Clase que representa a la entidad “Patrón” dentro de la aplicación. Listamos los campos que compondrán una instancia de un Patrón en memoria:

- *name*; nombre del patrón.
- *ID*; identificador del patrón.
- *date*; fecha de publicación.
- *level*; nivel de descripción.
- *aspect*; aspecto de diseño.
- *context*; contenido textual del campo *context*.
- *problem*; contenido textual del campo *problem*.
- *discussion*; contenido textual del campo *discussion*.
- *relatedTo*; contenido textual del campo *relatedTo*.
- *usesTo*; lista de patrones que mantienen una relación de tipo *usesTo* con el patrón.
- *suggestsTo*; lista de patrones que mantienen una relación de tipo *suggestsTo* con el patrón.
- *specializedBy*; lista de patrones que mantienen una relación de tipo *specializedBy* con el patrón.
- *image*; nombre del fichero de la imagen asociada.
- *diagram*; nombre del fichero del diagrama asociado.

- *solution*; contenido textual del campo *solution*.
- *resume*; texto que actuará como resumen del patrón.

ASPECTBEAN

Clase que representa a la entidad “Aspecto de diseño” dentro de la aplicación. Listamos los campos que compondrán una instancia de un Aspecto de diseño en memoria:

- *nombre*; nombre identificativo del aspecto de diseño.
- *descripcion*; texto descriptivo del aspecto de diseño.

LEVELBEAN

Clase que representa a la entidad “Nivel de descripción” dentro de la aplicación. Listamos los campos que compondrán una instancia de un Nivel de descripción en memoria:

- *nombre*; nombre identificativo del nivel de descripción.
- *descripcion*; texto descriptivo del nivel de descripción.

ANNOTATIONBEAN

Clase que representa a la entidad “Anotación” dentro de la aplicación. Listamos los campos que compondrán una instancia de una Anotación en memoria:

- *ID*; identificador de la anotación.
- *texto*; contenido textual de la anotación.

COMMENTBEAN

Clase que representa a la entidad “Comentario” dentro de la aplicación. Listamos los campos que compondrán una instancia de un Comentario en memoria:

- *nick*; nombre nick identificativo del autor del comentario.
- *date*; fecha de publicación del comentario.
- *email*; correo electrónico del autor del comentario.
- *comment*; contenido textual del comentario.
- *iaccepted*; flag que indica si el comentario está aceptado o no.

7. DISEÑO DE LA APLICACIÓN WEB

Con la descripción de estas últimas clases, queda cubierta la parte central de la aplicación Web. En los siguientes diagramas veremos el contenido de los paquetes restantes. El de esta figura corresponde al del paquete ACTIONS.

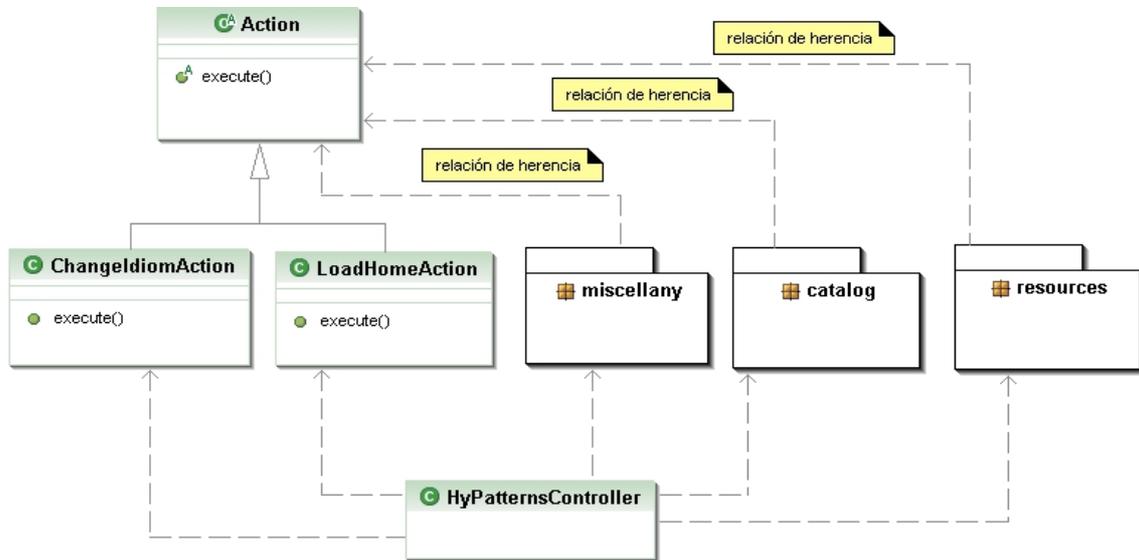


Figura 32: Diagrama de clases del paquete ACTIONS

Las clases pertenecientes a este paquete se corresponden con las acciones invocadas por la “clase controlador”. Todas ellas heredan de una misma clase ACTION, y se han definido tres paquetes de segundo nivel para distinguir a cuál de los tres ámbitos funcionales del sitio Web pertenece cada acción. Vamos a ver los tres diagramas de clases de estos tres nuevos paquetes y después detallamos cada componente.

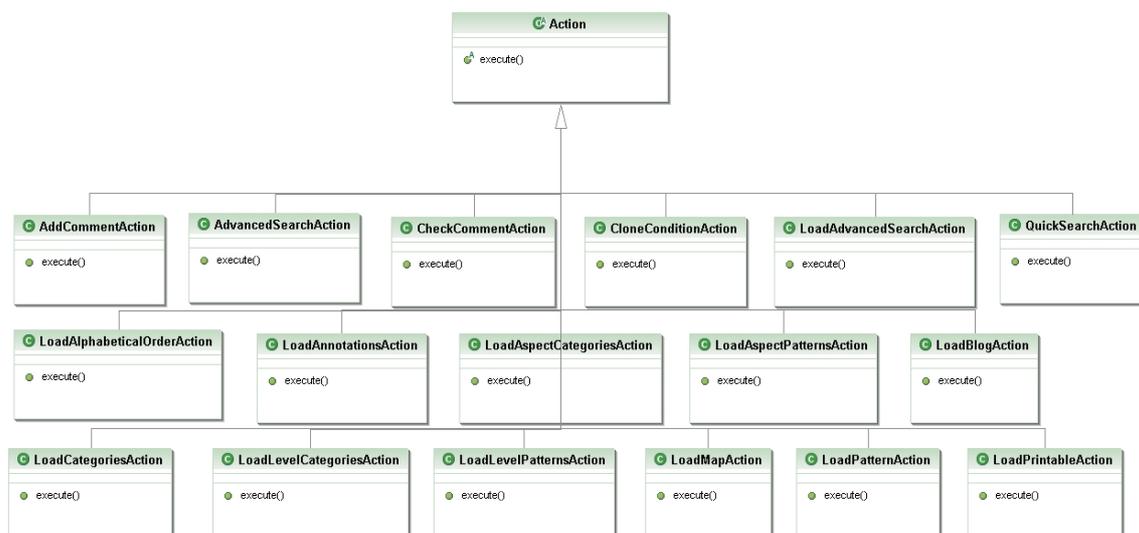


Figura 33: Diagrama de clases del paquete ACTIONS.CATALOG

7. DISEÑO DE LA APLICACIÓN WEB

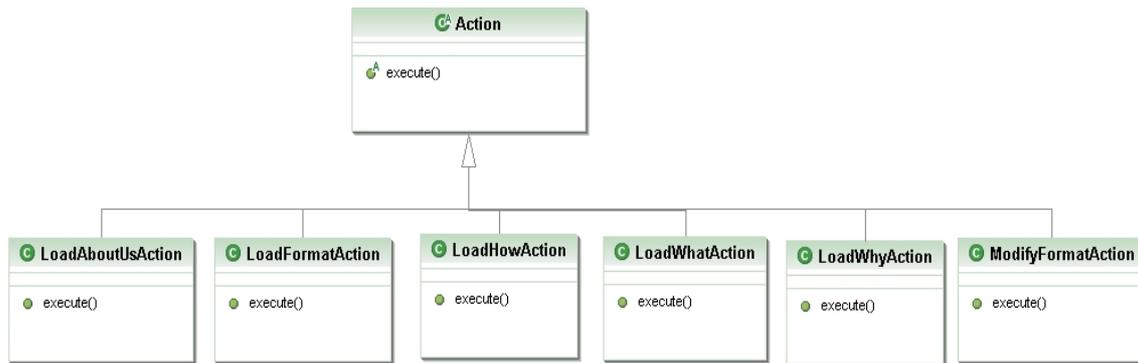


Figura 34: Diagrama de clases del paquete ACTIONS.MISCELLANY

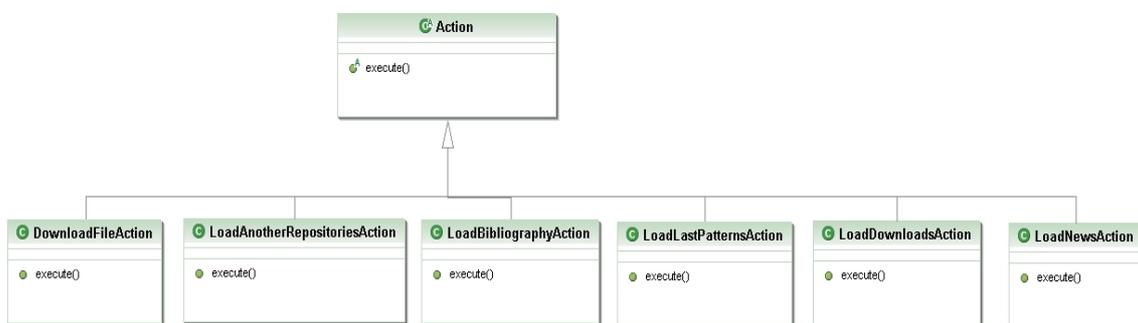


Figura 35: Diagrama de clases del paquete ACTIONS.RESOURCES

A continuación analizamos con más detalle cada una de las clases incluidas en el paquete ACTIONS y en los paquetes de segundo nivel contenidos en éste.

ACTION

Clase abstracta padre de todas las “clases acción” del paquete, declara el método `execute()` invocado por la “clase controlador” `HyPatternsController` para lanzar la acción asociada a una determinada petición del usuario, y que deberá ser implementado en cada una de las “clases acción” hijas.

LOADHOMEACTION

Carga la página de inicio de la aplicación. En la página de inicio se mostrarán los últimos patrones publicados. Para obtenerlos, invocará el método `getAllPatterns()` de la interfaz `HyPatternsManager`, y ordenará la colección resultante por fecha.

CHANGEIDIOMACTION

Acción invocada cuando un usuario cambia de idioma. Actualizará la información de sesión que se vea afectada por este cambio, como es el *locale* de la sesión.

ADDCOMMENTACTION

Acción invocada cuando un usuario añade un nuevo comentario. Para ello invocará al método *addComment()* de la interfaz `HyPatternsManager` pasándole los valores introducidos por el usuario.

ADVANCEDSEARCHACTION

Acción invocada cuando un usuario realiza una búsqueda avanzada. La acción se encargará de invocar el método *findPatternsByFilter()* de la interfaz `HyPatternsManager` proporcionándole los filtros o condiciones de búsqueda seleccionados por el usuario. Como resultado se cargará la página de resultados de búsqueda con los patrones que cumplan con las condiciones establecidas.

CHECKCOMMENTACTION

Acción invocada por el administrador del sistema para aceptar o rechazar un determinado comentario, valiéndose del método *checkComment()* de la interfaz `HyPatternsManager`.

CLONECONDITIONACTION

Escribe el código HTML necesario para clonar los campos de una nueva condición en la pantalla de búsqueda avanzada, como resultado de añadir una nueva condición de búsqueda por parte del usuario.

LOADADVANCEDSEARCHACTION

Carga la página para realizar búsquedas avanzadas de patrones.

LOADALPHABETICALORDERACTION

Carga la página con el listado en orden alfabético de todos los patrones del repositorio. Para obtenerlos, invocará el método *getAllPatterns()* de la interfaz `HyPatternsManager`, y ordenará la colección resultante por nombre.

LOADANNOTATIONS ACTION

Carga un combo con las anotaciones encontradas para un campo determinado. Puede haber dos tipos de anotaciones; referencias a otros patrones o literales de texto.

LOADASPECTCATEGORIES ACTION

Carga la página con el listado de los aspectos de diseño del repositorio. Esta página se valdrá de la información de sesión para pintar los aspectos.

LOADASPECTPATTERNS ACTION

Carga la página con el listado de los patrones que pertenecen a un determinado aspecto de diseño. Para obtenerlos invoca el método *getPatternsByAspect()* de la interfaz `HyPatternsManager`.

LOADBLOG ACTION

Carga la página de comentarios asociados a un determinado patrón. Para obtener el listado de comentarios invoca el método *getComments()* pasándole el identificador del patrón.

LOADCATEGORIES ACTION

Carga la página que lista los diferentes tipos de categorizaciones del repositorio que facilita el sitio Web.

LOADLEVELCATEGORIES ACTION

Carga la página con el listado de los niveles de descripción del repositorio. Esta página se valdrá de la información de sesión para pintar los niveles.

LOADLEVELPATTERNS ACTION

Carga la página con el listado de los patrones que pertenecen a un determinado nivel de descripción. Para obtenerlos invoca el método *getPatternsByLevel()* de la interfaz `HyPatternsManager`.

LOADMAP ACTION

Carga la página que contiene la imagen con el mapa del repositorio.

LOADPATTERNACTION

Carga la página de descripción de un patrón. Para conseguir la información a mostrar en la página invocará el método *getPatternByID()* de la interfaz `HyPatternsManager`, y posteriormente convertirá las referencias a otros patrones en enlaces a las páginas de descripción de los mismos.

LOADPRINTABLEACTION

Carga la página de impresión de un patrón. Al igual que al cargar la página de descripción, obtiene la información del método *getPatternByID()* de la interfaz `HyPatternsManager`, pero en lugar de crear enlaces las referencias a otros patrones las marca en negrita.

QUICKSEARCHACTION

Esta acción se invoca cuando el usuario realiza una búsqueda rápida de un texto sobre el repositorio. En este caso la búsqueda se realizará sobre el contenido indexado en el directorio a través del API Lucene, por lo que la acción invocará el método *findPatternsByText()* de la interfaz `HyPatternsManager` para obtener la colección de patrones que cumplan con la búsqueda.

LOADABOUTUSACTION

Carga la página de contenido informativo estático “Acerca De”.

LOADFORMATACTION

Carga la página de contenido informativo estático “Formato utilizado”.

LOADHOWACTION

Carga la página de contenido informativo estático “Cómo están organizados”.

LOADWHATACTION

Carga la página de contenido informativo estático “¿Qué son?”.

LOADWHYACTION

Carga la página de contenido informativo estático “¿Por qué y para quién?”.

MODIFYFORMATACTION

Acción invocada para establecer los campos que se visualizarán al entrar en la página de descripción de un patrón. Esta información queda almacenada como información de sesión.

DOWNLOADFILEACTION

Esta acción prepara el diálogo de descarga con el fichero que el usuario haya seleccionado para descargar.

LOADANOTHERREPOSITORIESACTION

Carga la página de contenido estático con los enlaces externos a otros repositorios de patrones.

LOADBIBLIOGRAPHYACTION

Carga la página de contenido estático con la bibliografía recomendada.

LOADDOWNLOADSACTION

Carga la página de contenido estático con los enlaces de descarga de documentación relacionada.

LOADNEWSACTION

Carga la página de contenido estático de noticias relacionadas.

Para terminar, pasamos a identificar las clases contenidas en el paquete `SESSION`. Si recordamos, este paquete mantenía una relación de dependencia con el paquete `CORE`, ya que la sesión del usuario puede requerir el disponer de determinada información proporcionada por el Modelo. Por otra parte indicamos anteriormente que las clases del paquete `ACTIONS` pueden requerir información almacenada en la sesión para la preparación de una determinada respuesta, y que el *listener* de sesión se encargaba de invalidar las sesiones al vencer el timeout establecido. Estas relaciones de dependencia se recuerdan en el siguiente diagrama.

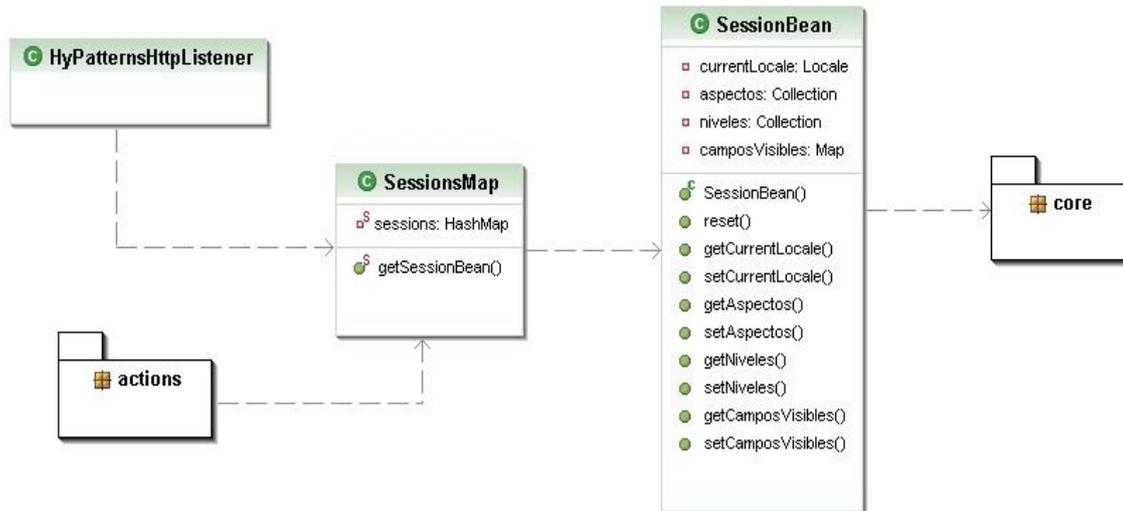


Figura 36: Diagrama de clases del paquete SESSION

A continuación analizamos con más detalle las dos clases que componen el paquete SESSION.

SESSIONSMAP

No es recomendable hacer que los datos de sesión viajen en cada petición que realice el usuario, sino que es mejor mantener esta información en algún tipo de estructura en memoria de modo que se acceda a ella a partir del identificador de sesión. Aplicando el patrón Singleton [Gamma *et al.*, 1995], esta clase representará el punto único de acceso a la información de las sesiones abiertas de los usuarios. Para ello nos apoyaremos en una estructura de tipo *HashMap*, donde cada elemento de la misma se corresponderá con un bean que contendrá la información asociada a un identificador de sesión.

SESSIONBEAN

Cada instancia de esta clase representará la información en memoria de una sesión abierta en un momento dado. Listamos a continuación los campos que representarán la información de sesión que interesará tener accesible durante toda la sesión de un usuario:

- *currentLocale*; idioma seleccionado por el usuario.
- *aspectos*; lista con los aspectos de diseño en el idioma seleccionado.

7. DISEÑO DE LA APLICACIÓN WEB

- *niveles*; lista con los niveles de descripción en el idioma seleccionado.
- *camposVisibles*; lista de los campos que serán visibles en la pantalla de descripción de un patrón. Esta lista dependerá de lo seleccionado por el usuario.

8 IMPLEMENTACIÓN

Llegados a este punto tenemos definido por un lado el diseño de las páginas de la interfaz de usuario, y por otro lado la arquitectura y la estructuración en clases y paquetes de la aplicación Web. El siguiente paso consistirá en concretar e integrar el diseño de ambas partes a través de su codificación y su estructuración en ficheros y carpetas. Esto se traduce en la programación de las clases haciendo uso de los métodos de las diferentes APIs que entrarán en juego en la aplicación (Jena, Lucene, JDOM), en la implementación de las páginas del sitio, en la definición de hojas de estilos y controles Javascript para dichas páginas, y en la definición de los ficheros de apoyo que sean necesarios, como pueden ser los ficheros de idiomáticos para la internacionalización del sitio, o el fichero de propiedades de la aplicación para hacer configurables ciertos valores de utilidad. Una vez implementado y definido todo esto, se empaquetará y se preparará la estructura de carpetas y ficheros requeridos para poder realizar el despliegue en un servidor de aplicaciones Web.

La elección de Jena como librería de acceso al contenido semántico del sitio, implicó a su vez la utilización de tecnología Java para la construcción de la aplicación. En el apartado 8.1 exponemos resumidamente los aspectos de implementación más relevantes, motivados fundamentalmente por el uso de Java como plataforma de desarrollo.

El siguiente objetivo es exponer el aspecto final del sitio Web tal y como se verá una vez desplegado en el servidor de aplicaciones, y explicar los aspectos no recogidos durante la fase de diseño que consideremos de mayor interés (véase el apartado 8.2). El aspecto final de las páginas que compondrán el sitio Web puede verse en el manual de usuario que sirve como apéndice de este documento (véase Apéndice I).

8.1 Aspectos de implementación

Cuando decidimos emplear Jena como solución para acceder al contenido semántico del sitio Web, a la vez estábamos seleccionando Java como plataforma tecnológica de implementación. Entre otras cosas esto implica el uso de las especificaciones definidas por la plataforma J2EE en lo que respecta a aplicaciones Web. Veamos las implicaciones de forma más detallada, atendiendo a la división arquitectural MVC definida en la fase de diseño:

- **Modelo:** las clases de negocio serán simples **clases Java** que interaccionarán con las diferentes librerías Java de acceso al repositorio de patrones; Jena, JDOM y Lucene.
- **Vista:** las páginas que integrarán la Vista de la aplicación, se traducirán en **páginas JSP** encargadas de transformar convenientemente los datos devueltos por las clases de negocio en las páginas HTML que finalmente visualizarán los usuarios en su navegador.
- **Controlador:** la “clase controlador” de la aplicación se traducirá en una clase **Servlet**, que tratará en primer lugar cualquier petición iniciada por los usuarios del sitio Web. El Servlet recibirá los parámetros de la petición a través de un objeto `javax.servlet.http.HttpServletRequest` y escribirá la respuesta (normalmente delegando en una página JSP) en un objeto `javax.servlet.http.HttpServletResponse`.

Entre los parámetros de la petición, viaja el valor que identifica la acción que debe ejecutarse. Cada una de estas acciones será una clase Java, que conocerá los métodos de las clases de negocio que deberá invocar para satisfacer la petición de origen.

Además, durante la fase de diseño identificamos dos clases “listeners”, una para detectar cambios en el contexto de la aplicación, y otra para detectar cambios en una sesión de usuario. Estas clases implementarán respectivamente la interfaz `javax.servlet.ServletContextListener` y la interfaz `javax.servlet.http.HttpSessionListener`.

8. IMPLEMENTACIÓN

8.1.1 Internacionalización

Uno de los requisitos no funcionales definidos al final de la fase de análisis, establecía que el sitio Web se adaptará a cualquier idioma sin realizar cambios de código. Esto abreviadamente se denomina internacionalización. Para llevarlo a cabo nos hemos apoyado en dos tipos de objetos del paquete `java.util`:

- `Locale`; un objeto `Locale` es un identificador para una combinación particular de idioma y región.
- `ResourceBundle`; estos objetos contienen objetos específicos del `Locale` del usuario.

Cada usuario tendrá asociado a su sesión un `Locale` que identificará el idioma en el que está visualizando el sitio Web. En función de este valor, el objeto `ResourceBundle` buscará en el fichero del idioma seleccionado por el usuario.

8.1.2 Gestión de errores

Ante cualquier excepción o error no esperado de la aplicación, ésta redirigirá la salida a una página genérica de error que visualizará el usuario.

Además, la aplicación dejará constancia del mismo en forma de trazas en un fichero de log que el administrador del sitio podrá consultar en orden a diagnosticar las causas. La escritura en el fichero de log se realizará empleando la biblioteca **Log4j** de Apache. Para configurar la localización del fichero y el formato de las trazas, se hará uso de un fichero de propiedades (del tipo “clave=valor”) que se ubicará dentro del contexto de la aplicación.

8.1.3 Especificaciones técnicas

Al no existir limitaciones previas que pudieran establecer algún tipo de incompatibilidad bloqueante, se ha decidido utilizar la versión de **JDK 1.6** para la compilación y ejecución de la aplicación. Las versiones de las principales librerías que usa el sitio Web son; la versión **2.5** para **Jena**, y la versión **2.3** para **Lucene**.

Durante la fase de desarrollo y pruebas, se empleará un servidor **Apache Tomcat versión 6.0**.

8.2 Estructura de implementación

En la figura 35 podemos observar la estructura en directorios del sitio Web tal y como quedaría desplegado en el servidor de aplicaciones.

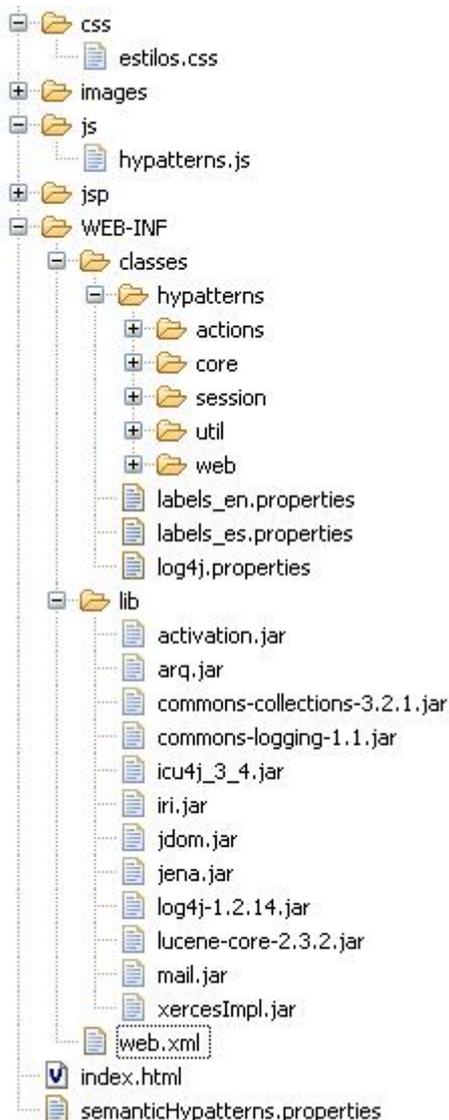


Figura 37: Estructura del sitio Web en carpetas

El subdirectorio **WEB-INF** es el más importante de todos, ya que contiene el fichero descriptor de despliegue de la aplicación (fichero **web.xml**), todas las clases compiladas de la aplicación (directorio **classes**), y las librerías de clases en forma de ficheros **.jar** que la aplicación Web usa. En el fichero **web.xml** declararemos el servlet controlador y los dos listeners que definimos en la etapa de diseño, así como la página de bienvenida de la aplicación (**index.html**), y definiremos el timeout de sesión.

8. IMPLEMENTACIÓN

La página de bienvenida de un sitio Web normalmente cuelga de la raíz, y generalmente se identifica con el nombre *index*. Nuestra página de bienvenida **index.html** se limita a redirigir la llamada directamente al servlet controlador.

También colgando directamente de la raíz hemos situado el fichero de propiedades de la aplicación **semanticHypattern.properties**. Este fichero contiene pares clave-valor para hacer configurables determinados valores de utilidad que pueden variar en función de donde se realice la implantación de la aplicación, de forma que se gane en flexibilidad. Enumeramos a continuación las propiedades contenidas en este fichero:

- **patternOWL**; URL del fichero pattern.owl.
- **hypermediaOWL**; URL del fichero hypermedia.owl.
- **hypermediaPatternOWL**; URL del fichero hypermediaPattern.owl.
- **repository**; ruta física de los ficheros RDF y los XML de descripción.
- **directory**; ruta física del directorio de índices de búsqueda.
- **downloads**; ruta física de los documentos de descarga.
- **comments**; ruta física con los XML de comentarios.
- **smtpServer**; servidor SMTP para el envío de correo electrónico.
- **emailComments**; dirección de correo electrónico del administrador del sitio.

Además del subdirectorio WEB-INF, de la raíz del sitio cuelgan otros cuatro subdirectorios más. El primero de ellos es el subdirectorio **css**, donde se ubicará la hoja de estilos o fichero CSS **estilos.css** donde se describirá el aspecto visual de los componentes de las páginas del sitio Web. En el subdirectorio **images** se localizarán todas las imágenes referenciadas desde las páginas del sitio, incluidas aquellas que formen parte de las descripciones de los patrones. Algunos navegadores no muestran las imágenes si éstas se referencian empleando rutas no relativas al contexto del sitio Web, de ahí que no se haya hecho configurable la ruta donde ubicar las imágenes. El subdirectorio **js** contendrá el fichero **hypatterns.js** en el que se definirán las funciones javascript comunes a todas las páginas del sitio. Por último nos encontramos con el subdirectorio **jsp**, que contendrá todas las páginas JSP del sitio Web. En la figura 36 vemos en detalle el contenido de este subdirectorio.

8. IMPLEMENTACIÓN

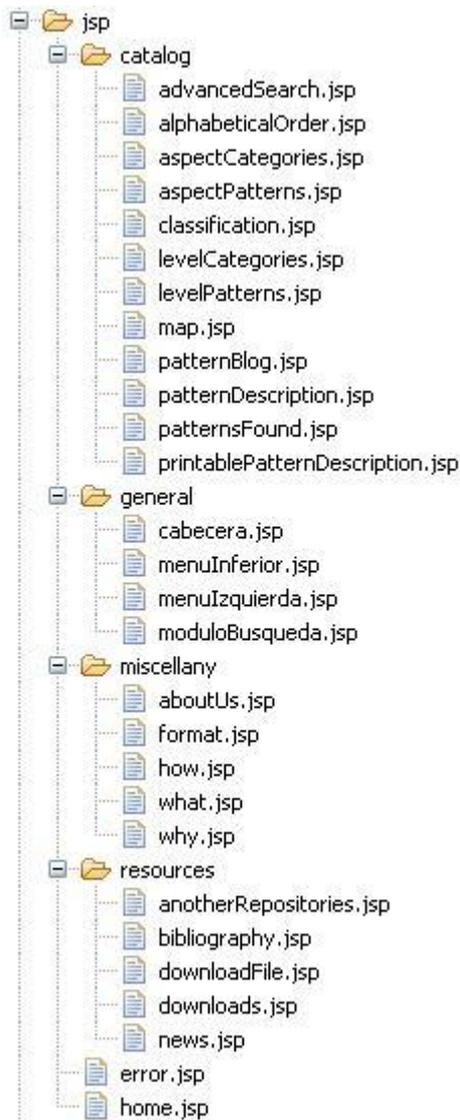


Figura 38: Detalle de la carpeta jsp

Como se puede observar en esta figura, las páginas JSP se han estructurado de una forma casi idéntica a las clases de acción definidas durante la fase de diseño. Debido a que la mayoría de acciones asocian su respuesta a una página JSP, hemos decidido estructurarlo de la misma manera. De este modo las páginas se reparten entre cuatro directorios, a excepción de la página de inicio y la página de error que cuelgan directamente de **jsp**. Los listamos a continuación:

- **catalog;** páginas relacionadas directamente con el repositorio de patrones
- **general;** páginas de uso general en el resto de páginas.
- **miscellany;** páginas de información general de la temática del sitio.
- **resources;** páginas de referencia informativa.

8. IMPLEMENTACIÓN

Como hemos dicho, en el subdirectorio **WEB-INF/classes** se ubican los compilables (.class) de las clases de la aplicación. De este subdirectorio cuelgan además los ficheros de idiomáticos **labels_en.properties** y **labels_es.properties** donde se ubican las traducciones en inglés y en español respectivamente, de las etiquetas de las páginas del sitio.

Por otro lado, como se indicó en el apartado anterior, la aplicación utilizará la biblioteca **Log4j** para dirigir la salida, el nivel de granularidad y para establecer el formato de los mensajes o “logs” en tiempo de ejecución. Para configurar esta librería definimos un fichero **log4j.properties** también localizado en el subdirectorio **classes**.

En la figura 37 mostramos un detalle con la estructura en subdirectorios de los fuentes de la aplicación. En ella podemos observar que se ha respetado la organización de las clases en paquetes realizado en la fase de diseño de forma que cada paquete de diseño se corresponderá con un paquete en java. El paquete HYPATTERNS actuará como paquete raíz de todos los fuentes.

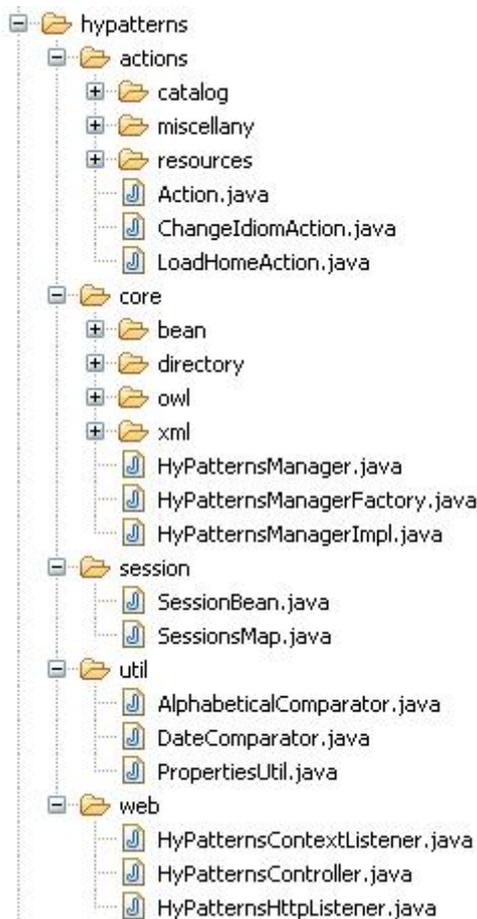


Figura 39: Detalle de la carpeta con los fuentes de la aplicación

8. IMPLEMENTACIÓN

Para finalizar vamos a enumerar brevemente las librerías en formato JAR del subdirectorio **WEB-INF/lib** utilizadas por la aplicación. Algunas de ellas sólo son necesarias en tiempo de compilación, pero creemos conveniente mantenerlas todas en una misma localización:

- **activation**; contiene las clases que componen el framework JavaBeans Activation.
- **arq**; contiene las clases que componen el motor de consultas SPARQL de Jena.
- **commons-collections**; framework para el manejo de estructuras de almacenamiento Java.
- **icu4j**; librería para dar soporte a la internacionalización y globalización de aplicaciones
- **iri**; es una implementación del proyecto Jena de las recomendaciones RFC 3987 (IRI) y RFC 3986 (URL).
- **jdom**; conjunto de clases Java para el acceso, manipulación y creación de ficheros XML.
- **jena**; framework para la construcción de aplicaciones basadas en tecnologías de la Web Semántica.
- **log4j**; librería de clases Java que permite elegir la salida y el nivel de granularidad de los “logs” en tiempo de ejecución.
- **lucene-core**; librería Java para el uso en aplicaciones que requieren indexado y realizar búsquedas sobre texto completo.
- **mail**; framework para la construcción de aplicaciones de mensajería y correo electrónico.
- **xercesImpl**; librería de parseo de ficheros XML.

9 CONCLUSIONES

9.1 Conclusiones generales

A lo largo de este proyecto hemos desarrollado un sitio Web para dar acceso al repositorio semántico de patrones de diseño hipermedia presentado en [Montero, 2005], y con ello se ha conseguido cubrir con éxito el principal objetivo que nos marcamos al principio. A partir de este momento el conocimiento recogido en este repositorio es accesible de forma instantánea y cómoda por cualquier usuario de Internet, a través de un diseño homogéneo y bien estructurado y una navegación intuitiva que facilita la exploración y lectura de los patrones que lo componen.

El estudio y análisis de otras Webs de patrones de diseño como medio para la captura y posterior definición de requisitos nos ha parecido acertada y suficiente en función de los objetivos generales, muy abstractos, de los que partíamos. Así, a modo de ejemplo, vimos la utilidad de disponer de un índice de secciones en la cabecera de la descripción de un patrón, la necesidad de proporcionar una versión imprimible para estas descripciones, y valoramos el enriquecimiento que supone para el sitio Web el dedicar un espacio para los comentarios de los usuarios, así como la inclusión de información complementaria en forma de enlaces externos a sitios relacionados, documentación, bibliografía y similares. De la misma forma, echamos de menos la inclusión de páginas o herramientas de búsqueda, advertimos una cierta despreocupación estética en la presentación de las páginas, especialmente en las páginas de inicio, y pudimos comprobar que en la mayor parte de los casos las herramientas de navegación son insuficientes o no están bien aprovechadas.

Uno de los objetivos generales que nos marcamos fue el de proporcionar una interfaz Web usable. Como solución se decidió construir un lenguaje de patrones de diseño que guiara el diseño de la interfaz, a partir del catálogo de patrones

9. CONCLUSIONES

que van Duyne recopila en su libro *The Design of Sites*. La conclusión más positiva que se puede extraer de esta solución, es la de haber ponderado debidamente y haber intentado estructurar un proceso que tradicionalmente carece de un orden y al que no se le otorga la importancia que merece.

Por otro lado, todos conocemos a estas alturas las ventajas que proporcionan los patrones de diseño como herramientas para la creación de sistemas robustos, escalables, fácilmente adaptables,... No es una excepción el catálogo de van Duyne, muy completo, atendiendo a diferentes niveles de detalle, permitiendo diseños muy flexibles y acompañando las explicaciones de ejemplos de aplicación en el mundo real. Sin embargo, también podemos sacar algunas conclusiones acerca de los problemas que puede acarrear su uso:

- Existe el riesgo de crear un diseño demasiado complejo o sofisticado para un sitio Web que por sus características requiera una solución muy simple. Si no se tiene en cuenta esto a la hora de afrontar el problema podría incluso convertirse en algo contraproducente, retrasando el proceso y dando como resultado diseños sobrecargados.
- Desconocer el catálogo y emplearlo por primera vez, como ha sido nuestro caso, puede dar lugar a algunos problemas. Por ejemplo es probable que en ocasiones se use un patrón en algún lugar donde habría sido mejor utilizar otro diferente, o se haga un uso inadecuado de patrones si se tiende a ponerlos en práctica de una manera rápida y sin un análisis previo detallado de los mismos.
- En su catálogo, van Duyne relaciona los patrones unos con otros de modo que permite muchas posibilidades y la creación de diseños muy flexibles. Sin embargo el hecho de estar en soporte de papel dificulta mucho su navegación convirtiéndose en un proceso algo tedioso.
- Aunque se trata de un catálogo muy completo que abarca casi todos los géneros de sitios Web, también es cierto que tiende a centrarse bastante en las Webs comerciales, fundamentalmente a la hora de dar ejemplos.

Por su parte, los objetivos de carácter funcional del sitio se resumían en los siguientes puntos:

9. CONCLUSIONES

1. Dar acceso a la visualización de las descripciones de los patrones.
2. Posibilitar la realización de consultas o búsquedas en el repositorio.

Debido a la naturaleza semántica del repositorio de patrones al que da acceso nuestro sitio Web, la consecución de dichos objetivos pasaba por seleccionar tecnologías para Web Semántica de entre las disponibles actualmente adecuadas a nuestras necesidades. El framework para desarrollo de aplicaciones de Web Semántica **Jena** ha resultado más que suficiente para la exploración, lectura y consulta de nuestro repositorio de patrones, puesto que da soporte a RDF y al manejo de ontologías y OWL. Estas son algunas de las reflexiones que podemos extraer tras su uso:

- La impresión general obtenida después de la utilización de Jena es positiva. Ofrece un buen rendimiento y su uso es sencillo. Un aspecto importante es que no requiere un conocimiento previo de la sintaxis de las tecnologías de Web Semántica a las que da acceso, como es el caso de OWL o RDF. Sin embargo sí es recomendable familiarizarse con dichas tecnologías antes de empezar a trabajar para asimilar con más garantías la filosofía de funcionamiento de la librería, sobre todo si atendemos a la escasa documentación que hay disponible.
- Es difícil extraer alguna desventaja o algún aspecto negativo del uso de Jena, probablemente en parte porque las funcionalidades de acceso y consulta requeridas por nuestro sitio Web han quedado cubiertas con las posibilidades más básicas de esta plataforma. De hecho Jena ofrece muchas más características de las que han hecho falta utilizar.
- Para la realización de búsquedas semánticas de patrones se decidió explotar las posibilidades del lenguaje de consulta SPARQL, gracias a que Jena ofrece el motor de búsqueda ARQ para la explotación de consultas en dicho lenguaje. Por otro lado, la navegación por el repositorio se decidió hacer por medio de los métodos de iteración, exploración y lectura básicos de Jena. La comparativa entre ambos métodos de acceso nos lleva a concluir que es mejor el uso de consultas SPARQL, en la medida en que clarifica el código y el rendimiento es similar en los dos casos.

9.2 Líneas de trabajo futuro

Para finalizar vamos a procurar definir algunos posibles trabajos o ampliaciones futuras que podrían llevarse a cabo sobre el sitio Web que hemos desarrollado:

- A pesar de que el sitio Web se ha preparado para poder visualizarse en cualquier idioma sin necesidad de modificar código, ésto por el momento únicamente es efectivo en lo que concierne al contenido estático de las páginas. Los patrones del repositorio sólo están definidos en castellano, tanto la información semántica como las descripciones textuales. De poco sirve presentar las herramientas de navegación o cualquier otro elemento de una página en un determinado idioma, si no se dispone en ese mismo idioma del contenido que realmente importa. Por tanto, la primera línea de trabajo debería ir orientada en este sentido, aprovechando además las capacidades de Jena en lo que a internacionalización se refiere.
- En un principio las necesidades de acceso al repositorio han quedado cubiertas completamente, ya sea a través de la exploración del sitio o a través de la realización de búsquedas, semánticas o a partir de un texto libre. Las futuras ampliaciones del sitio deberían ir encaminadas fundamentalmente a la gestión o administración del repositorio. Así, sería francamente interesante disponer de una sección en la que el administrador o administradores del sitio Web pudieran dar de alta nuevos patrones o modificar los ya existentes. En este sentido, incluso podría plantearse la posibilidad de hacer del sitio una especie de Wiki en el que cualquier persona registrada pudiera hacer cambios sobre la descripción del patrón.
- En otra línea de ampliación menos ambiciosa a la anterior, sería positivo enriquecer el sitio a través de nuevas herramientas o iniciativas de participación y colaboración.

APÉNDICE I: MANUAL DE USUARIO

Introducción

Este breve manual pretende mostrar el aspecto final del sitio Web desarrollado en este proyecto a través de una serie de capturas de pantalla, sirviendo a su vez como guía de iniciación para el uso del mismo.

En el sitio Web se distinguen tres grandes secciones:

- **Catálogo:** esta sección es la razón de ser del sitio, y por tanto la más importante. Da acceso a los patrones de diseño hipermedia del repositorio, por medio de diferentes caminos.
- **Sobre patrones de diseño hipermedia:** las páginas que conforman esta sección, reúnen información acerca de los patrones de diseño hipermedia en general, y de nuestro repositorio en particular.
- **Recursos:** en esta sección se ha procurado reunir información de interés relacionada con los patrones de diseño hipermedia, ya sea a través de enlaces a sitios externos, bibliografía recomendada, noticias, o por medio de documentos disponibles para su descarga.

Nos basaremos en la división por secciones anterior para ir navegando a través de las diferentes páginas del sitio. Antes de pasar a ver cada una de ellas, veremos y explicaremos primero la página de inicio, y a continuación los componentes comunes a todas las páginas.

Página de inicio

En la página de inicio (ver figura 39) se distinguen tres partes bien diferenciadas. La primera de ellas consiste en un pequeño mensaje de bienvenida. A su derecha, en un recuadro azul, se dedica un espacio pensado para el usuario que visita el sitio por primera vez, con una serie sugerencias para empezar a navegar. El resto de la página se dedica a listar los últimos patrones de diseño incorporados al repositorio.



Figura 40: Página de inicio

Componentes comunes

Hay una serie de componentes que son comunes e invariantes a todas las páginas del sitio Web. El primero de ellos corresponde a la cabecera (ver figura 40), ubicada a lo largo de la parte superior de cada página. En la cabecera queda ubicada la “imagen de marca” o nombre del sitio Web, y en su extremo derecho se localizan los enlaces para cambio de idioma en forma de pequeños iconos de banderas.



Figura 41: Cabecera del sitio

Justo debajo de la cabecera se encuentra el módulo de búsqueda rápida de patrones (ver figura 41). En cualquier momento el usuario podrá realizar búsquedas sobre el repositorio de patrones a partir de un texto libre que escribirá en la caja de texto destinada para ello y pulsando el botón . Además, desde este módulo se podrá acceder a la página de búsqueda avanzada (o búsqueda semántica), por medio del enlace ubicado en la parte derecha.

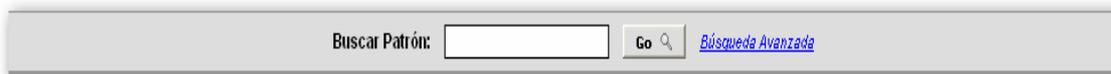


Figura 42: Módulo de búsqueda rápida

A lo largo de la parte izquierda de las páginas se ubica el menú o elemento de navegación principal del sitio Web (ver figura 42). Si exceptuamos lo que respecta a las operaciones de búsqueda (accesibles desde el módulo de búsqueda rápida), desde este menú tendremos acceso a todas las secciones que componen el sitio. Cada uno de los enlaces que componen el menú, se destacará en color azul cuando el cursor del ratón se sitúe sobre ellos. De alguna manera este menú refleja de un simple vistazo, la estructura global del sitio Web en forma de árbol de hasta dos niveles máximo. Así, a lo largo del menú se distinguen tres zonas principales correspondientes con las tres secciones principales del sitio (ver la introducción de este manual), cada una de las cuales se ramifica en las diferentes subsecciones que la componen.



Figura 43: Menú del sitio

Por último, en el pie de página se destina un espacio donde el usuario podrá encontrar enlaces de contacto o relacionados al propietario del sitio Web (ver figura 43).



Figura 44: Pie de página del sitio

Catálogo

El catálogo de patrones de diseño hipermedia del sitio puede ser explorado inicialmente de tres formas diferentes:

- Accediendo directamente al contenido completo del repositorio a través de un listado en orden alfabético.
- En función del aspecto de diseño de los patrones.
- En función del nivel de descripción de los patrones.

Desde la opción **De la A a la Z** del menú del sitio se accede directamente al listado alfabético de patrones (ver figura 44). El orden del listado va en función del nombre del patrón y está encabezado por un abecedario, en el que cada letra actúa como un enlace interno al inicio del grupo de patrones cuyo nombre empieza por la letra correspondiente. Cada uno de estos grupos se encabeza por la letra que lo representa, a cuyo margen derecho se ubica un enlace interno (*go top*) que te permite regresar al comienzo de la página sin necesidad de realizar *scroll*.

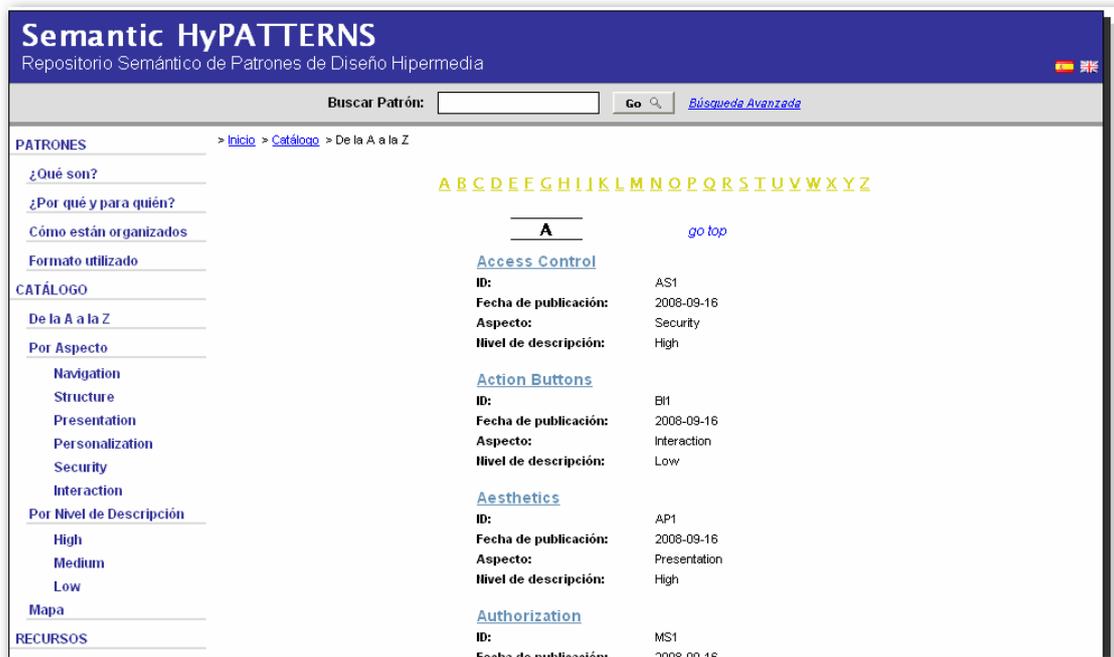


Figura 45: Listado en orden alfabético de todos los patrones del repositorio

El usuario tiene la posibilidad de explorar el catálogo de patrones en función del aspecto de diseño en el que se encuadren pulsando la opción del menú que corresponda;

Navigation, Structure, Presentation, Personalization, Security, ó Interaction. Por otra parte, desde la opción **Por Aspecto** del menú, el usuario puede acceder a una página donde se hace un breve resumen de los diferentes aspectos de diseño que hemos enumerado antes (ver figura 45). En la figura 46, se muestra un ejemplo de listado de patrones por aspecto de diseño, en concreto para el aspecto *Presentation*, accesible directamente desde el menú o desde la página de la figura 45.



Figura 46: Listado resumen de los aspectos de diseño

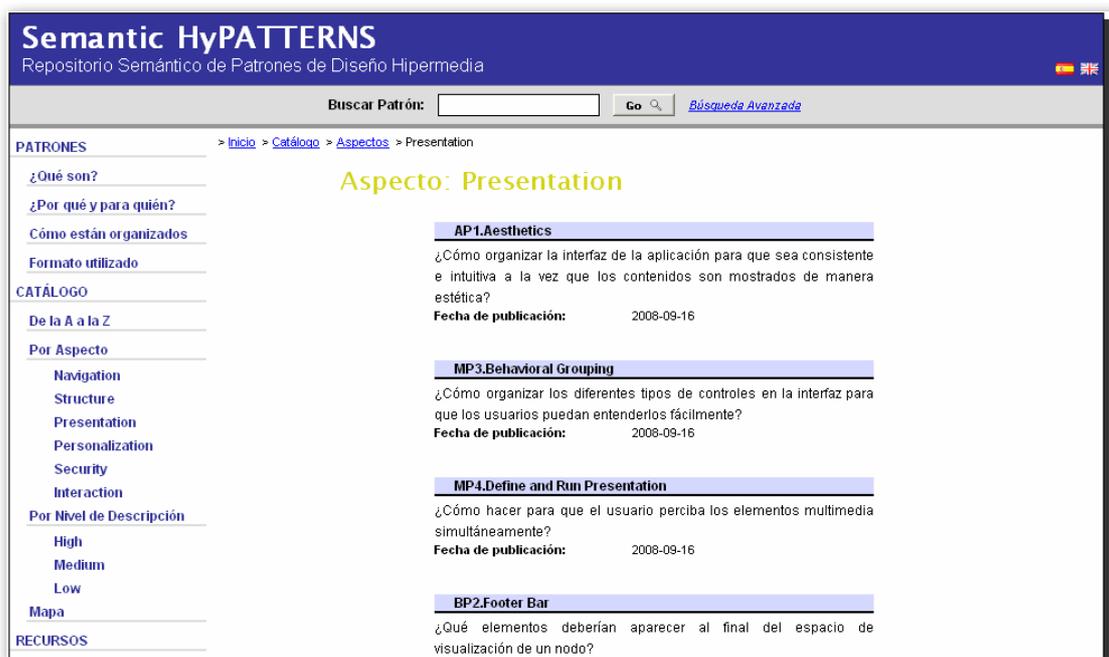


Figura 47: Listado de patrones pertenecientes al aspecto *Presentation*

De la misma forma el usuario puede explorar el catálogo en función del nivel de descripción de los patrones; **High**, **Medium**, o **Low**. Así, desde la opción **Por Nivel de Descripción** del menú del sitio el usuario puede acceder a la página resumen de los diferentes niveles de descripción (ver figura 47). En la figura 48, se muestra un ejemplo de listado de patrones por nivel de descripción, en concreto para el nivel *High*, accesible directamente desde el menú o desde la página de la figura 47.



Figura 48: Listado resumen de los niveles de descripción



Figura 49: Listado de patrones pertenecientes al nivel *High*

Cada una de las entradas de los listados de patrones que acabamos de ver, se trata de un enlace a la página de descripción del patrón al que hace referencia. En la figura 49 se muestra un ejemplo de página descripción de patrón.



Figura 50: Ejemplo de página de descripción de un patrón

La página de descripción de un patrón está encabezada por el nombre del patrón en cuestión. Justo debajo del nombre del patrón, se puede ver un recuadro con dos funciones:

- Contenedor de los campos identificativos del patrón. Estos son; el identificador o ID, la fecha de su publicación, el aspecto de diseño en que se enmarca, y el nivel de descripción al que pertenece.
- Índice del resto de campos que describen al patrón. Cada entrada del índice es un enlace interno al comienzo del texto del campo correspondiente, de forma que se evita tener que hacer *scroll* hasta el campo que se desea visualizar.

A continuación del índice y a lo largo del resto de la página se muestra la descripción propiamente dicha del patrón en forma de campos bien diferenciados. Cada uno de estos campos dispone de un enlace interno (*go top*) que permite regresar al comienzo de la página sin necesidad de hacer *scroll*.

Si lo que se quiere es acceder a una versión imprimible de la descripción del patrón, se deberá pulsar el botón  situado a la derecha del recuadro del índice. Como resultado se abrirá una nueva ventana en forma de *pop-up* (ver figura 50), donde se visualizará la descripción en formato texto. El mismo botón de antes  aparece en la parte superior de esta nueva ventana. Pulsándolo se abrirá el cuadro de diálogo estándar para la impresión de la página.



Figura 51: Versión imprimible de la descripción de un patrón

Como se puede ver en la página de ejemplo de descripción de patrón (ver figura 49), a lo largo de la parte derecha de la misma queda localizada una barra lateral con enlaces a contenido relacionado con el patrón visualizado. Dentro de esta barra lateral hay tres partes bien diferenciadas. La primera de ellas, identificada por el encabezado **Patrones Relacionados**, lista los enlaces a las páginas de descripción de todos los patrones de diseño del repositorio que mantienen algún tipo de relación con el patrón visualizado. Seguidamente se muestra una pequeña sección llamada **RDF** que contiene un enlace con la URL donde se localiza el fichero RDF asociado al patrón, el cuál se visualizará en una ventana independiente a modo de *pop-up* (ver figura 51).

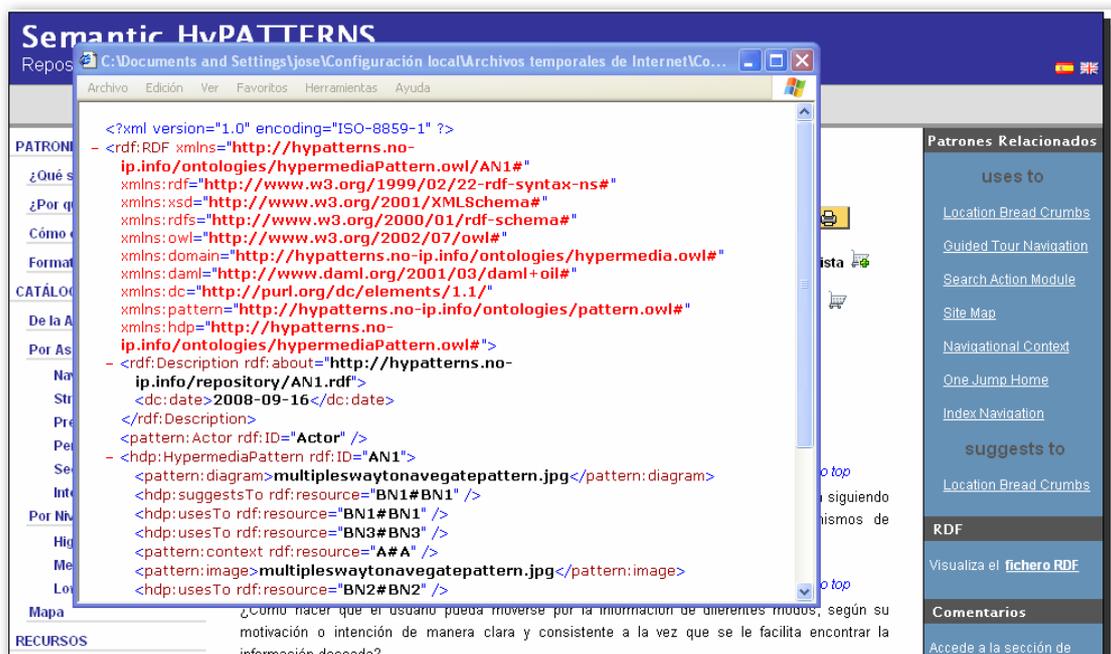


Figura 52: Contenido del fichero RDF asociado a un patrón

La tercera de las partes de la barra lateral de contenido relacionado (**Comentarios**), contiene el enlace a la sección de comentarios del patrón (ver figura 52). En esta nueva página el usuario puede escribir su propio comentario. Para ello tan sólo deberá acompañarlo de un nombre identificativo y de un correo electrónico válido, y pulsar el botón . Si todo va bien, se mostrará un mensaje confirmando que el comentario se ha enviado correctamente.



Figura 53: Página de comentarios de un patrón

Hasta el momento hemos resumido cómo explorar el repositorio de patrones a través de la navegación. La otra alternativa consiste en la realización de búsquedas. Para ello existen dos posibilidades. Una de ellas, la búsqueda a partir de texto libre, ya se mostró cuando se explicó el módulo de búsqueda, presente en todas las páginas del sitio. La búsqueda avanzada de patrones o búsqueda semántica, es la otra posibilidad. Desde el enlace [Búsqueda Avanzada](#) del módulo de búsqueda accederemos a la página con el formulario de búsqueda avanzada (ver figura 53).



Figura 54: Formulario de búsqueda avanzada

La búsqueda avanzada consiste en ir definiendo condiciones de búsqueda sobre la información semántica de los diferentes atributos de un patrón. Inicialmente la página se mostrará sin ninguna condición de búsqueda. Pulsando el botón  se añadirán todas las condiciones de búsqueda que se deseen. Por cada condición primeramente se tendrá que seleccionar de un *combo* el atributo sobre el que se definirá la misma. Una vez seleccionado, a su derecha aparecerá un nuevo *combo* con los valores que conforman la información semántica encerrada en todos los patrones del repositorio para dicho atributo. Si se desea, existe la posibilidad de eliminar condiciones ya definidas a través del icono  ubicado en el margen derecho de cada condición. Finalmente, pulsando el  botón se lanzará la búsqueda.

Tanto la búsqueda por texto libre como la búsqueda avanzada, mostrarán el resultado de la ejecución en una página de resultados de búsqueda (ver figura 54). Esta página estará encabezada por un pequeño recuadro resumen en el que se informará del número de patrones encontrados y en el que se enumerarán las condiciones de búsqueda

definidas por el usuario. A continuación se mostrará el listado de los patrones que se ajustan a dichas condiciones.

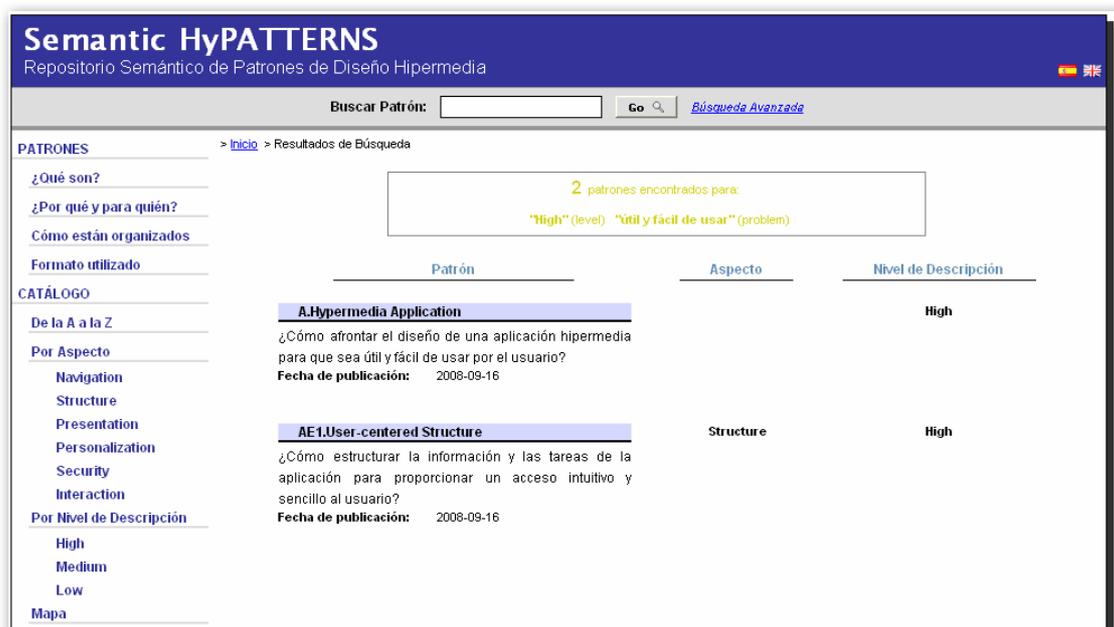


Figura 55: Página de resultados de búsqueda

Sobre patrones de diseño hipermedia

Esta sección, en su casi totalidad puramente informativa, la conforman cuatro páginas accesibles desde el menú del sitio con los siguientes identificadores:

- **¿Qué son?**; introducción general al concepto de patrón de diseño y su aplicación al contexto del diseño hipermedia.
- **¿Por qué y para quién?**; justifica la creación del repositorio e identifica los usuarios potenciales del mismo.
- **Cómo están organizados**; explica cómo está organizado el repositorio y argumenta el por qué de dicha organización.
- **Formato utilizado**; enumera y define brevemente los atributos que componen la descripción de un patrón.

Las tres primeras de las páginas enumeradas presentan un formato común. En la figura 55, mostramos una de ellas a modo de ejemplo.



Figura 56: Página informativa “Cómo están organizados”

La página de **Formato utilizado** (ver figura 56) además sirve para que cada usuario pueda personalizar la visualización de las descripciones de los patrones, mediante la selección de aquellos atributos que se deseen hacer visibles. Para ello se deberán marcar los atributos que interesen y pulsar el botón  .

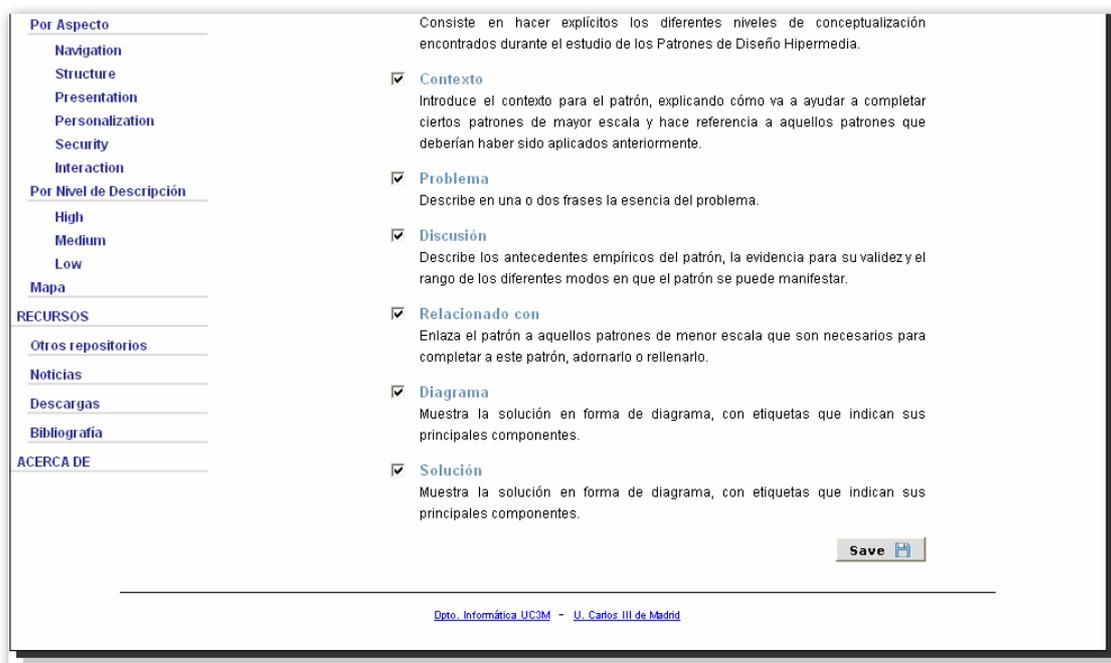


Figura 57: Página “Formato utilizado”

Recursos

Esta sección reúne cuatro páginas con listas de entradas o enlaces a diferentes fuentes de información relacionada con los patrones de diseño hipermedia. Éstas son accesibles desde el menú del sitio con los siguientes identificadores:

- **Otros repositorios**; lista de enlaces a sitios Web externos de información relacionada, en concreto a otras colecciones de patrones.
- **Descargas**; página con enlaces para la descarga de documentos en diferentes formatos de información relacionada
- **Noticias**; esta página reunirá aquellas noticias, avisos o eventos de interés relacionados al propio sitio Web o a otras entidades.
- **Bibliografía**; aquí se muestra una colección de referencias bibliográficas recomendadas.

Como ejemplos de esta sección en las figuras 57 y 58 se muestra respectivamente la página de enlaces a otros repositorios y la página de descargas.

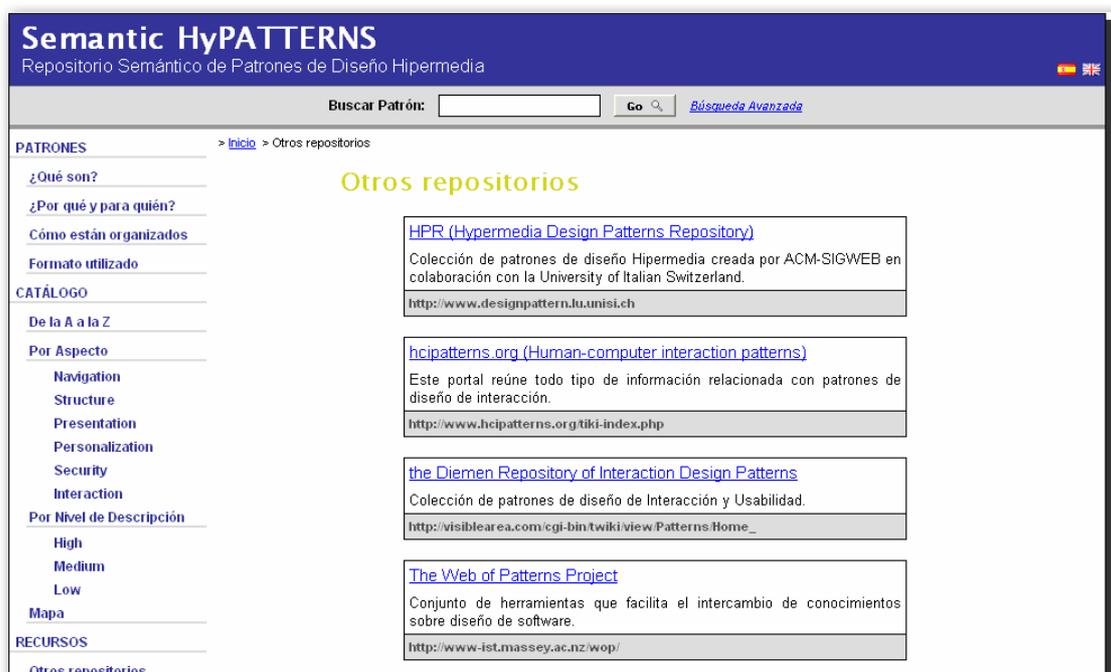


Figura 58: Página de enlaces a otros repositorios de patrones

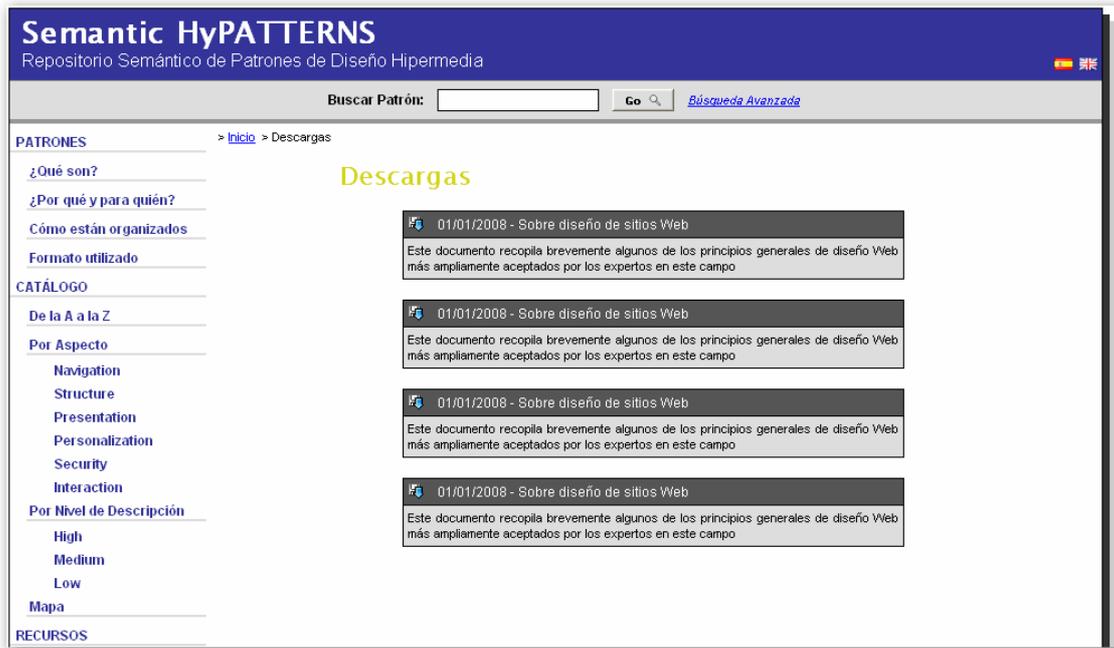


Figura 59: Página de enlaces de descarga de documentos

REFERENCIAS BIBLIOGRÁFICAS

- [Alexander *et al.*, 1977] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, y S. Angel. *A pattern language: Towns, Buildings, Construction*. Oxford University Press, New York, 1977.
- [AjaxPatterns] <http://ajaxpatterns.org/wiki/index.php>
- [Berners-Lee, 2000] T. Berners-Lee. *Semantic Web -XML2000. Architecture*. <Disponible en: <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide11-0.html>>
- [Berners-Lee *et al.*, 2001] T. Berners-Lee, J. Hendler, y O. Lassila. *The semantic web*. Scientific American, May 2001.
- [Burbeck, 1992] S. Burbeck. *Applications Programming in Smalltalk- 80: How to Use Model-View-Controller (MVC)*. University of Illinois in Urbana-Champaign (UIUC) Smalltalk Archive, 1992. <Disponible en: <http://www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>>
- [Cunningham, 1987] W. Cunningham, y K. Beck. *Using Pattern Languages for Object-Oriented Programs*. OOPSLA, 1987.
- [Escalona *et al.*, 2002] M. J. Escalona, y N. Koch. *Ingeniería de Requisitos en Aplicaciones para la Web – Un estudio comparativo*. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla, Diciembre 2002.
- [Ferrerres, 2006] G. Ferreres. *Tipología de sitios web*. <Disponible en: <http://mosaic.uoc.edu/articulos/gferrerres1106.html>>
- [Gamma *et al.*, 1995] E. Gamma, R. Helm, R. Johnson, y J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.

REFERENCIAS BIBLIOGRÁFICAS

- [hcipatterns] <http://www.hcipatterns.org/tiki-index.php>
- [HDPR] <http://www.designpattern.lu.unisi.ch>
- [Hom, 1998] J. Hom. *The Usability Methods Toolbox*. Junio 1998. <Disponible en: <http://jthom.best.vwh.net/usability/>>
- [Hassan Montero *et al.*, 2003] Y. Hassan Montero, y F. J. Martín. *Guía de Evaluación Heurística de Sitios Web*. Marzo 2003. <Disponible en: <http://www.nosolousabilidad.com/articulos/heuristica.htm>>
- [Jena] <http://jena.sourceforge.net/>
- [Lucene] <http://lucene.apache.org/>
- [Mahemoff, 2006] M. Mahemoff. *Ajax Design Patterns*. O'Reilly, 2007.
- [Montero, 2005] S. Montero. *Integración de Patrones en el proceso de Diseño de Sistemas Hipermedia mediante el uso de ontologías*. Departamento DEI de la Universidad Carlos III de Madrid, 2005.
- [Nielsen *et al.*, 1994] J. Nielsen, y R. L. Mack. *Usability Inspection Methods*. John Wiley & Sons, Inc., New York, 1994.
- [Nielsen, 2002] J. Nielsen. *Ten Usability Heuristics*. 2002. <Disponible en: http://www.useit.com/papers/heuristic/heuristic_list.html>
- [Nielsen, 2004] J. Nielsen. *The Need for Web Design Standards*. Septiembre 2004. <Disponible en: <http://www.useit.com/alertbox/20040913.html>>
- [Reffel *et al.*, 2006] J. Reffel, y M. Alpern. *Design Patterns in the Real World*. In Proceedings of the 7th Information Architecture, 2006.
- [Rodríguez *et al.*, 2005] K. Rodríguez Perojo y R. Ronda León. *Mapa conceptual de la Web Semántica. Web Semántica: un nuevo enfoque para la organización y recuperación de información en la web*. Acimed, vol. 13, núm. 6, Noviembre 2005.
- [Rosengard *et al.*, 2004] J. Rosengard, y M. Ursu. *Ontological Representations of Software Patterns*. University of London, 2004.

REFERENCIAS BIBLIOGRÁFICAS

- [Sanders *et al.*, 2007] W. Sanders, y C. Cumaranatunge. *ActionScript 3.0 Design Patterns*. O'Reilly, 2007.
- [Saponas *et al.*, 2006] T. S. Saponas, M. K. Prabaker, G. D. Abowd, y J. A. Landay. *The impact of pre-patterns on the design of digital home applications*. ACM Press, New York, 2006.
- [Spool, 2003] J. M. Spool. *Design Patterns: An Evolutionary Step to Managing Complex Sites*. 2003. <Disponible en: http://www.uie.com/articles/design_patterns/>
- [Tidwell, 2006] J. Tidwell. *Designing Interfaces*. O'Reilly, 2006.
- [Tognazzini, 1998] B. Tognazzini. *First Principles of Interaction Design*. Noviembre 1998. <Disponible en: <http://www.asktog.com/basics/firstPrinciples.html>>
- [van Duyne *et al.*, 2002] D. K. van Duyne, J. A. Landay, y J. I. Hong. *The Design of Sites: Patterns, Principles, and Processes for Crafting a Customer-Centered Web Experience*. Addison-Wesley, 2002.
- [van Welie *et al.*, 2000] M. van Welie, G. C. van der Veer, y A. Eliëns. *Patterns as Tools for User Interface Design*. In: International Workshop on Tools for Working with Guidelines, October 2000.
- [Voelter *et al.*, 2006] M. Voelter, D. Manolescu, J. Noble. *Pattern Languages of Program Design*. Addison-Wesley, 2006.
- [W3C, 2004a] The World Wide Web Consortium (W3C). *Resource Description Framework (RDF): Concepts and Abstract Syntax*. W3C Recommendation 10 February 2004. <Disponible en: <http://www.w3.org/TR/rdf-concepts/>>
- [W3C, 2004b] The World Wide Web Consortium (W3C). *OWL Web Ontology Language Overview*. W3C Recommendation 10 February 2004. <Disponible en: <http://www.w3.org/TR/owl-features/>>
- [W3C, 2004c] The World Wide Web Consortium (W3C). *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C Recommendation 10 February 2004. <Disponible en: <http://www.w3.org/TR/rdf-schema/>>

REFERENCIAS BIBLIOGRÁFICAS

[W3C, 2008a] The World Wide Web Consortium (W3C). *Guía breve de Web Semántica*. W3C Oficina Española, 2008. <Disponible en: <http://www.w3c.es/Divulgacion/Guiasbreves/WebSemantica>>

[W3C, 2008b] The World Wide Web Consortium (W3C). *SPARQL Query Language for RDF*. W3C Recommendation 15 January 2008. <Disponible en: <http://www.w3.org/TR/rdf-sparql-query/>>

[Web Patterns] http://groups.ischool.berkeley.edu/ui_designpatterns/webpatterns2/webpatterns

[Workflow Patterns] <http://www.workflowpatterns.com/index.php>

[Yahoo! DPL] <http://developer.yahoo.com/ypatterns/>

