

MODELAMIENTO Y ARQUITECTURA WEB.

INTRODUCCIÓN

Anteriormente las Páginas Web se realizaban de manera estática, a través de un editor de html, usando herramientas practicas que facilitaran su desarrollo. Pero el crecimiento continuo del Internet y de los medios de comunicación exigían a su vez que ciertas paginas Web requirieran de otro tipo de desarrollo que pudieran soportar de manera dinámica. Con esto apareció el término aplicaciones Web, donde eran desarrollos que fueran robustos, escalables y fácilmente adaptables. Pero debido al gran éxito que obtuvo el desarrollo de estas aplicaciones Web en diferentes tecnologías (jsp, asp, javascript, etc) surgió la necesidad de que este trabajo necesitaba de más técnicas formales de diseño y recuperar ciertas metodologías de diseño que se realizan hoy en día para el desarrollo de un sistema de escritorio. Así fue como Jim Conallen presento sobre los diseños de UML modificados con estereotipos para las aplicaciones Web. A partir de allí, muchos investigadores se vieron en la tarea de buscar la manera de poder hacer un diseño de software orientado a Web.

1. QUE ES EL DISEÑO WEB?

El diseño web es una actividad que consiste en la planificación, diseño e implementación de sitios web y páginas web. No es simplemente una aplicación del diseño convencional sobre Internet ya que requiere tener en cuenta cuestiones tales como navegabilidad, interactividad, usabilidad, arquitectura de la información y la interacción de medios como el audio, texto, imagen y vídeo.

El diseño no es sólo la comunicación textual (contenidos) existentes en Internet de una manera visual, sino que obliga a pensar en una mejor estructuración de los mismos en este soporte. 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, que brinda posibilidades como el contacto directo entre el productor y el consumidor de contenidos, característica destacable del medio Internet.

El diseño de páginas Web es una amplia área de aplicación del diseño gráfico en la cual se integran conocimientos propios del diseño como son la composición, el uso de color y la tipografía con conocimientos técnicos del medio como son los lenguajes HTML (HiperText Markup Language) y CSS (Cascading Style Sheets), así como conocimientos sobre usabilidad, accesibilidad y organización de un sitio web.

1.2 Generaciones de Diseño Web:

Primera generación de diseño web: El primer diseño web de una página se realiza en 1993, la página web tenía el nombre Mosaic, y en menos de un año había recibido 2 millones de visitantes. El navegador web era capaz de mostrar tanto imágenes como textos, aunque con una limitación muy alta a la hora de diagramar la información de la página web. El diseño web de estas páginas era lineal y estaba orientado para científicos que eran los usuarios que compartían su información alrededor de todo el mundo mediante estas páginas web. La tecnología de los navegadores web era limitada y no disponía de la capacidad de transmitir información gráfica para la comunicación visual.

Las principales características de esta primera generación de diseño web eran las velocidades de transmisión de datos, ya que era conexión vía MODEM, lo que limitaba el peso de las páginas web. Otro detalle era el uso de monitores monocromos. Respecto al diseño web en particular, la estructuración era bastante desordenada con imágenes dispuestas horizontalmente y líneas de texto separadoras.

Debido a este caos en el diseño web, un año más tarde se estableció un consorcio para establecer unas normas y pautas para el futuro desarrollo de la web, el W3C. Se comenzaron entonces a desarrollar unos estándares de lenguaje html para una unificación del diseño web que trajo consigo la aparición de una serie de navegadores web con el constante desarrollo de nuevas funcionalidades y progresos en este ámbito.

Segunda generación de diseño web: El diseño web de esta generación está basado en los conceptos de la primera salvo por que empieza a utilizar iconos en lugar de algunas palabras, las páginas web comienzan a poseer imágenes de fondo, aparece el diseño y uso de botones con relieve para la navegabilidad, el uso de banners en lugar de cabeceras, la estructuración de texto de forma jerárquica mediante menús o listados, propiedades del código html Standard definido.

La aparición de estos elementos gráficos en el diseño web generó un deseo de todos los diseñadores de añadir estos archivos en sus páginas web, pero sin control, lo que generó una saturación en las páginas web, sin tener en cuenta la accesibilidad ni la usabilidad. Aun así seguían habiendo algunos diseñadores que utilizaban estos elementos de manera correcta para generar unos diseños innovadores y elegantes, llenos de elegancia, usando correctamente tanto etiquetas como el resto del código html. Se comenzaron a usar tablas para la organización de los contenidos, posicionamiento de los elementos y generación de diseño y maquetación similar a libros o revistas.

En esta generación está la aparición de monitores y tarjetas gráficas con mayores resoluciones y definición de color, lo cual generó la consecuente mejora en la calidad del diseño web. Pero apareció un problema, la diferencia en la adaptación de estándares de los 2 principales navegadores: Internet Explorer y Netscape Navigator.

Tercera generación de diseño web: En la tercera generación, el diseño web sigue teniendo muchas restricciones con el uso del lenguaje para los dos navegadores web. El diseño web se orienta en esta generación a los diseñadores, los cuales tienen mucho más dinamismo al aparecer el plugin de Macromedia Flash, el cual revolucionaría la concepción de diseño web. Es una era de enfocar las páginas web según el objetivo de las mismas: vender productos o servicios, comunidades, información, noticias. Para esta especialización del diseño web de acuerdo al objetivo de las páginas se necesita ayudar al usuario a encontrar la información, generando una navegabilidad estructurada e intuitiva.

La gran mayoría de páginas web que aparecen en esta generación son de publicidad y venta de productos y servicios, con lo que es este el diseño más utilizado. Conseguir acercarlos al producto, que deseen ver más páginas del sitio web.

Cuarta generación de diseño web: En la cuarta generación, el diseño web ya está enfocado totalmente a la multimedia, integrando en las páginas web los elementos multimedia de última generación. Con usuarios de todos los tipos, cualquiera tiene una página web a día de hoy y la variedad de diseño es enorme debido a todas las posibilidades que ofrecen las últimas tecnologías para los programadores. A esto le podemos añadir que las últimas versiones de

los navegadores soportan muchas más prestaciones y elementos en las páginas web.

Quinta generación de diseño web: Se podría añadir una última generación de diseño web que representaría el auge de las páginas de televisión ip o televisión online. La integración de televisión en la web para empresas, ayuntamientos o asociaciones representa el movimiento más notable del diseño web en la actualidad. Un ejemplo de empresas de diseño que ya ofrecen la posibilidad de integración de televisión en web es Hooping.net mediante su plataforma de televisión IP Hooping.tv.

2. DISEÑO DE SITIOS WEB.

Según Forrester (1998), los costos de un mal diseño son extremadamente altos para una empresa, las perdidas mas importantes que se registran son:

- Perdidas de aproximadamente el 50% de las ventas potenciales, porque los usuarios no pueden encontrar el producto adecuado.
- Perdidas del 40% de segundas visitas de usuarios por la mala experiencia de la primera visita.

2.1 Principios generales para el diseño de sitios web (Tognazini, 1999).

1. Anticipación: el sitio web debe anticiparse a las necesidades del usuario.
2. Autonomía: los usuarios deben tener el control sobre la aplicación (entorno abarcable y finito)
3. Los colores: con precaución.
4. Consistencia: aplicaciones consistentes con las expectativas de los usuarios.
5. Eficiencia del usuario: las aplicaciones deben focalizarse en la productividad del usuario, no de la máquina.
6. Permitir la reversibilidad de las acciones.
7. Ley de Fitts, el tiempo para alcanzar un objetivo está en función de la distancia y el tamaño del objetivo. Por tanto, usando botones más grandes y situando los elementos en las esquinas de la pantalla reduciremos el tiempo.
8. Reducción del tiempo de latencia, es posible optimizar el tiempo disponible permitiendo la realización de otras tareas mientras se completa la previa.
9. Aprendizaje: los sitios web deben requerir un mínimo proceso de aprendizaje y deben poder ser usadas desde el primer momento.
10. El uso de metáforas, usadas adecuadamente facilitan el aprendizaje de un sitio web, en caso contrario pueden dificultarla.
11. La protección del trabajo de los usuarios (que no pierdan su trabajo como consecuencia de un error).
12. Legibilidad: contraste y tamaño de las fuentes del texto suficientemente grande.
13. Seguimiento de las acciones del usuario: conocerlo para volverlo al punto que estaba.
14. Interfaz visible: evitar elementos invisibles de navegación supuestos.

2.1.1 La escritura de los contenidos:

- a. Una página web no debe contener mucho **texto**, no más de unas pocas líneas. Los usuarios **no leen**, ojean velozmente (scan) en busca de la información que les interesa. “Las páginas deben ser ojeables”.
- b. Todo elemento de información presentado compite con el resto para captar la atención del usuario y por ello es crucial evitar presentar información superflua.

2.1.2 Características de la estructura de la información del texto:

- a. Texto estructurado: palabras resaltadas en negrita / color, listas numeradas...
- b. Contenidos estructurados con sumarios y tablas de contenidos.
- c. Títulos y subtítulos claros, simples y concisos.
- d. Párrafos cuyo contenido tenga una sola idea.
- e. Redacción en estilo de pirámide invertida: comenzando los textos por la conclusión y finalizando por los detalles.
- f. Usar la mitad de palabras que se usaría en la redacción de un texto.
- g. Lenguaje objetivo (sin adjetivos, palabras redundantes o afirmaciones fortuitas...).
- h. Buena combinación de colores de texto y fondo: texto claro sobre fondo oscuro o viceversa (evitando la fatiga ocular).
- i. No usar textos parpadeantes o deslizantes, dificultan la lectura e imposibilitan a prestar atención a otro punto de la página.
- j. El lenguaje simple e informal es preferido al académico (la lectura es más rápida).

2.1.3 La escritura de los contenidos para la red:

- a. La página inicial: es la primera que se visualiza y debido a la lentitud de la red, el efecto psicológico usual de primacía. El máximo de información contenido en el sitio web debe estar disponible en la página inicial. La página inicial debe ser un gran conjunto de links, cuanto más información se ofrezca al usuario, que recordemos ojea velozmente la página centrado en la búsqueda de su meta, más probabilidades existen de que encuentre lo que busca. Sin embargo, se debe evitar la saturación que disminuiría la visibilidad de la información.
- b. El orden temporal de visualización de la información, debe estar jerarquizado de mayor a menor utilidad y relevancia.
- c. Para reducir la experiencia subjetiva de tiempo en los usuarios se pueden utilizar varios elementos:
 - Un indicador de tiempo de espera (que le facilite su planificación).
 - Un indicador del buen funcionamiento del sistema.
 - Un indicador del tiempo transcurrido.
 - Para esperas más largas se debe captar la atención del usuario con elementos atractivos.
 - Indicar claramente al usuario cuando puede proceder.

2.1.4 El Refuerzo: una función de la velocidad, el tiempo y el orden de los elementos.

- a. Los buscadores: deben ser capaces de encontrar una información a pesar de escribir la palabra incorrectamente o confundirla con otra de pronunciación parecida.
- b. Es necesario un buscador de información interno cuando el “site” supere las 100 páginas y cuando supere las 200, la opción de “buscar” debe estar presente en cada página del sitio.
- c. El buscador debe buscar en todo el sitio web puesto que los usuarios raramente tienen una clara imagen de la estructura del sitio web que les permita reducir la búsqueda a determinados subsitios web.
- d. Las búsquedas booleanas o parciales deben ser restringidas a la sección de búsquedas avanzadas, puesto que usuarios sin experiencia pueden cometer errores.

2.1.5 La búsqueda de información:

- a. Las áreas de navegación solo deben contener un número reducido de links, es decir, solo los más útiles para los usuarios con un máximo de 6 o 7.
- b. Algunas páginas específicas no necesitan área de navegación de manera indispensable (disponer en el sitio de links relacionados más directamente con los temas tratados en esa página en lugar del área de navegación).
- c. Los mapas del sitio son muy útiles para facilitar una imagen global del sitio web.
- d. Se debe aplanar la estructura de la página, es decir, la información de un “site” no debe estar excesivamente jerarquizada. La obligatoriedad de navegar a través de muchas páginas antes de llegar al objetivo provocará la pérdida de muchos usuarios.

2.1.6 La Navegación:

Todas las páginas deben tener un área de navegación en la parte superior, con enlaces que permitan moverse dentro del “site”.(Debido a las características de los motores de búsqueda externos nunca sabemos en cuál de las páginas puede aterrizar un usuario).

Algunos consejos para la elaboración de links:

- a. El título del link debe informar del tipo de información que se encuentra en su destino.
- b. El título de un link no debe ser tan general que no llegue a despertar el interés en el usuario, pero tampoco tratar en ningún caso de embaucar al usuario con un título atractivo.
- c. El título de un link debe avisar acerca de posibles problemas en el sitio de destino (suscripción, registro o la existencia de frames).
- d. Los títulos de links deben ser de menos de 60 caracteres y cuando sea obvio el destino del link se suprimirá el texto.
- e. Son de utilidad para evitar equívocos los globos de texto con información (complementaria) concreta sobre el destino del link que se visualizan al pasar el ratón por encima.

- f. Los links cuyo destino esta en la misma página de origen deben distinguirse de los externos, el elemento “#” delante del título del link debe distinguir a los primeros.(Hyperterrorist Guide).
- g. No se debe repetir el mismo link.
- h. Se recomienda que la primera palabra sea una palabra clave sobre el tema tratado y relegar al final de la línea las palabras del título más comunes. Ej. “diseño de contenidos: los títulos”, sería preferible lo siguiente: “títulos en el diseño de contenidos”.

Es adecuado crear links a contenidos relacionados. Es útil enlazar a páginas de:

- Productos similares, un poco más caros o un poco más baratos.
- Productos relacionados en sus funciones o que generalmente se compran juntos.
- Productos que varían del actual en un aspecto distintivo.
- Diferentes versiones del producto actual, otros colores, tallas, etc.
- Versiones posteriores o anteriores del tema tratado en la página.
- Información contextual.
- Información del autor y listas de otros trabajos del mismo autor.
- Lista de discusión sobre el tema referido.
- Noticias de actualidad sobre el tema referido.

Todas las páginas de un sitio web deben tener el mismo diseño para facilitar su uso y evitar el aprendizaje en cada nueva página.

Las **direcciones internas** de las páginas deben ser legibles y no contener caracteres extraños (ej. “~”).

La estandarización de los **colores** de los links.

2.1.7 Consistencia: Reconocimiento y Memoria:

- Todas las imágenes deben utilizar el atributo tanto para facilitar la navegación de los usuarios que hayan suprimido la carga de imágenes como para favorecer la accesibilidad de discapacitados.
- Si se utilizan gráficos su utilización debe restringirse a la complementación de información del texto y no deben usarse para llamar la atención del usuario.
- Debido a su influencia sobre el tiempo de carga, su tamaño se recomienda que sea reducido.

2.1.8 Las imágenes:

El uso desmedido de imágenes a veces distrae a los visitantes y en otras ocasiones los incomoda si estas no son acordes con el sitio, haciendo que se tornen cansonas y el usuario termine por abandonar el sitio.

2.1.9 Errores más frecuentes, cometidos por los diseñadores

- Texto parpadeante: casi imposible prestar atención a nada más (como luces largas de un coche).
- Animaciones gratuitas: distraen la atención y relentizan la carga.
- Textos deslizantes: nada más incomodo que leer un texto que se mueve (marea la vista y dificulta la concentración y atención).
- Combinaciones ilegibles de texto y fondo: el contraste de luminancia.

- Encabezamientos decorados: imágenes publicitarias que tardan en visualizarse, relentizan y frustran al usuario.
- Uso de Marcos (Frames): no funcionan bien y muchos cuelgan a navegadores.
- Apertura automática de nuevas ventanas del navegador.
- Cualquier elemento que parezca publicidad (se ignora todo elemento que se parezca a la publicidad).
- Títulos que no tienen sentido fuera de su contexto: los usuarios no disfrutan yendo a donde no quieren, por ello debemos aportar máxima información sobre el destino de un link.
- Recomendaciones de cambio de resolución de un monitor.
- Verborrea corporativa.
- Páginas sin dirección de e-mail a la que contestar (ej. Revista Fortune, McDonald's,...): es una muestra de lo que estas empresas valoran las opiniones de sus clientes.
- Incluir cosas sin saber el porqué, solo porque lo hace la competencia.
- Confundir estudios de usabilidad con estudios de mercado.
- Creer que el diseño de un "site" es solo el aspecto que tiene.
- Pensar en la página Web como un medio secundario y tratarlo como un "tríptico colgado en la red".
- Diseñar el "site" para la autocomplacencia de los altos ejecutivos de la empresa (¿acaso son ellos la audiencia objetivo?).
- La estructura de la página web no tiene porque seguir la estructura organizativa de la propia compañía (no significa que una estructura organizativa adecuada deba ser fácilmente comprensible para navegar en ella).
- Olvidarse del presupuesto de mantenimiento cuando la página esta hecha (al menos de un 50% del coste inicial de la página).

3. MODELOS PARA EL DESARROLLO WEB

Existen varios modelos guías para el desarrollo de sitios Web, basados en ingeniería del software, entre ellos tenemos:

3.1 MODELO ESPIRAL: Esta clase trato sobre el modelo espiral, el cual es a base de una serie de ciclos los cuales se repiten en forma de espiral, cada vez que se avanza un ciclo se va alcanzando un nivel superior hasta concluir el proyecto.

Este modelo utiliza prototipos para un mejor desarrollo del producto.

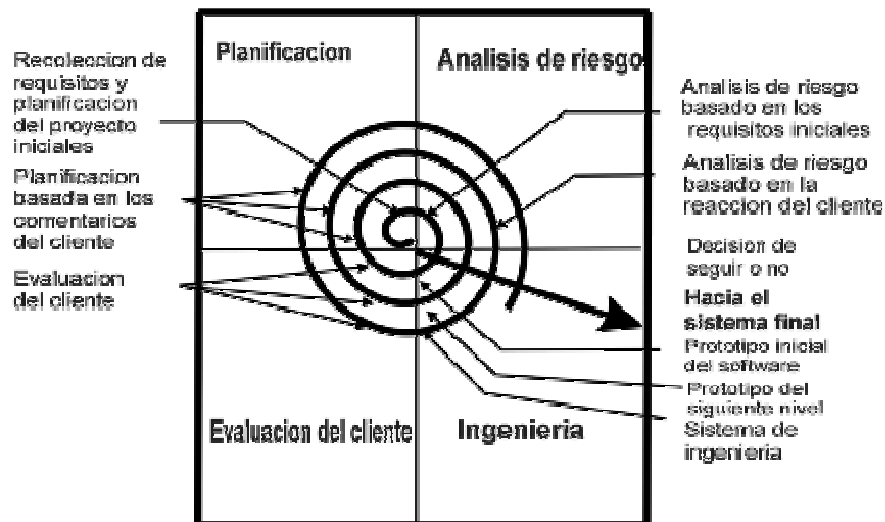
Lo característico de este modelo es que incluye un "análisis de riesgo" es decir que podemos analizar si el proyecto puede continuar o mejor lo suspendemos.

Este modelo se basa en que antes de hacer algo debemos analizarlo, también debemos buscar varias opciones de resolución de problemas para de allí escoger la opción más conveniente, y además analizar los riesgos que se pueda tener. Este modelo necesita de otro para poder desarrollarse. Se debe escoger el modelo cascada cuando se pierda el control del presupuesto o por el calendario; y el de prototipado cuando tengamos problemas con la interfaz.

El modelo espiral consta de 4 cuadrantes que son sus fases y se dividen de la siguiente forma:

- 1.- Planificación
- 2.- Análisis de Riesgos

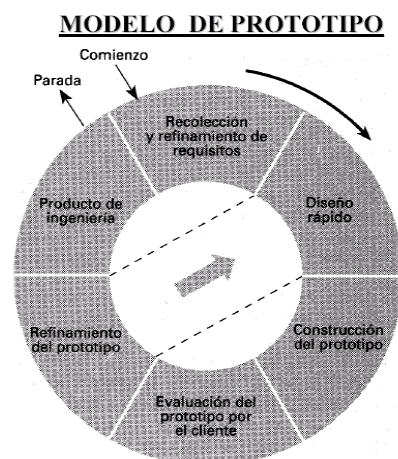
- 3.- Ingeniería
- 4.- Evaluación



3.2 MODELO DE PROTOTIPOS: Esta clase trata sobre el modelo de prototipos los cuales se utilizan para dar al usuario una vista preliminar de parte del software. Este modelo es básicamente prueba y error ya que si al usuario no le gusta una parte del prototipo significa que la prueba falló por lo cual se debe corregir el error que se tenga hasta que el usuario quede satisfecho.

Además el prototipo debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar mucho dinero pues a partir de que este sea aprobado nosotros podemos iniciar el verdadero desarrollo del software. Pero eso si al construir el prototipo nos asegura que nuestro software sea de mejor calidad, además de que su interfaz sea de agrado para el usuario. Un prototipo podrá ser construido solo si con el software es posible experimentar.

Sus desventajas son que debido a que el usuario ve que el prototipo funciona piensa que este es el producto terminado y no entienden que recién se va a desarrollar el software. Otro problema es que el prototipo debe ir acompañado de otro modelo para su desarrollo.



3.3 MODELO CASCADA Y V: Es un modelo base para los demás modelos. Fue definido por Royce y se trata principalmente de que se debe completar un paso correctamente sin ningún error para pasar al siguiente. Este modelo muestra de una forma básica el desarrollo de software, y representa en fases separadas procesos fundamentales.

Dice que se debe probar el software después de construirlo y antes de operarlo. Cada fase tiene como salida documentación.

Fases del Modelo Cascada:

Ingeniería y Análisis del Sistema: establece requisitos de los elementos del sistema.

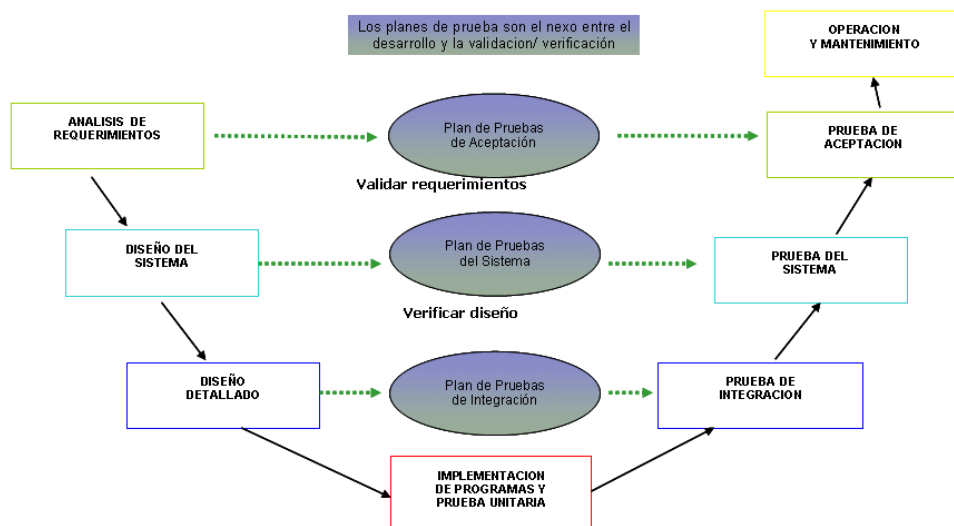
Análisis de los requisitos del software: identifica las funciones del software, el rendimiento, sus interfaces y la información.

Diseño: se basa en estructura de datos, arquitectura del software el detalle de los procedimientos y la caracterización de la interfaz. Además escoge las herramientas para la codificación.

Codificación: el diseño se traduce en lenguaje de máquina.

Pruebas: Aquí se comprueba si existe algún error con el software o si funciona correctamente. Hasta que sea aceptado por el usuario.

Mantenimiento: esta fase se da debido a que después de la entrega pudo haber errores en el software, o el software no se adapte al entorno externo o que el cliente requiera ampliaciones funcionales o de rendimiento.



1. ARQUITECTURA WEB.

Hasta el día de hoy, lo más importante en el desarrollo de aplicaciones Web han sido las herramientas. Poco se ha dicho sobre el proceso de desarrollo.

La fácil creación de hojas HTML y en general de sitios Web, usando herramientas simples, ha hecho que el desarrollo de este tipo de aplicaciones se haga sin un trabajo serio de análisis y diseño.

Cualquier sistema de complejidad no trivial, necesita ser analizado y modelado. Las aplicaciones Web, al igual que otras aplicaciones, necesitan métodos formales de análisis y diseño.

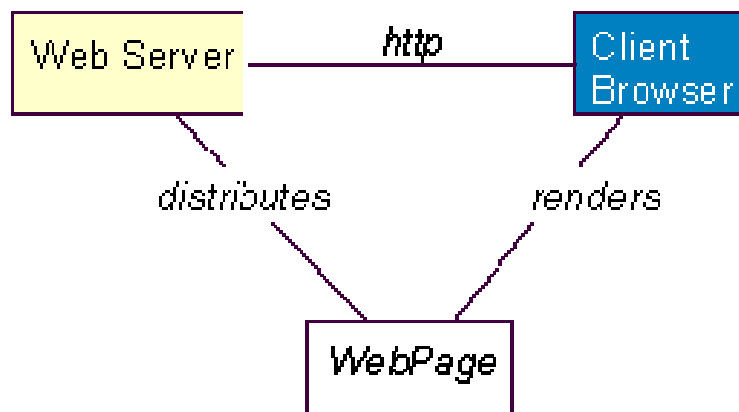
¿Cuál es la diferencia entre un sitio Web y una aplicación Web?

“Una aplicación Web es un sitio Web donde la navegación a través del sitio, y la entrada de datos por parte de un usuario, afectan el estado de la lógica del negocio. En esencia, una aplicación Web usa un sitio Web como entrada (front-end) a una aplicación típica.

Si no existe lógica del negocio en el servidor, el sistema no puede ser llamado aplicación Web.” [Conallen 99]

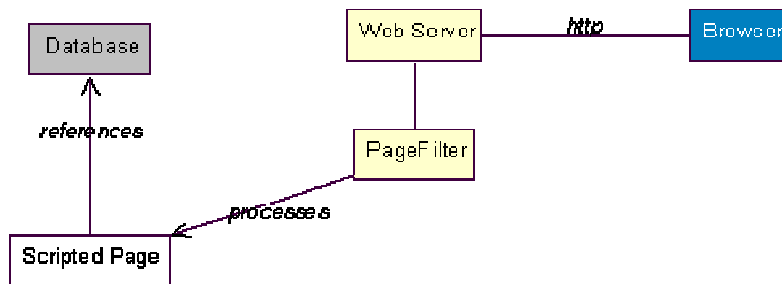
La arquitectura de un sitio Web tiene tres componentes principales: un servidor Web, una conexión de red, y uno o más clientes (browsers).

El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP.



4.1 Arquitectura básica de una aplicación/sitio Web: La información mostrada en las páginas está típicamente almacenada en archivos. Sin embargo, muchas veces esta información está almacenada en una base de datos, y las páginas son creadas dinámicamente. Los sitios Web que usan este esquema, son llamados **sitios dinámicos**.

Mientras que los **sitios estáticos** son aquellos que su contenido no varía cada vez que se visita, es decir siempre es el mismo. Además el usuario no tiene ninguna libertad de cambiar su contenido lo único que puede cambiar es el orden de navegabilidad.



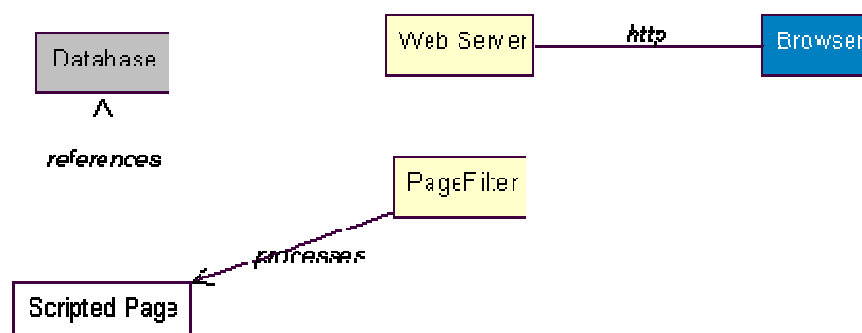
Páginas Web: Las páginas Web son el componente principal de una aplicación o sitio Web. Los browsers piden páginas (almacenadas o creadas dinámicamente) con información a los servidores Web. En algunos ambientes de desarrollo de aplicaciones Web, las páginas contienen código HTML y scripts dinámicos, que son ejecutados por el servidor antes de entregar la página. Una vez que se entrega una página, la conexión entre el browser y el servidor Web se rompe (a diferencia de otros esquemas tipo cliente/servidor). Es decir que la lógica del negocio en el servidor solamente se activa por la ejecución de los scripts de las páginas solicitadas por el browser (en el servidor, no en el cliente).

Scripts en el cliente Cuando el browser ejecuta un script en el cliente, éste no tiene acceso directo a los recursos del servidor. Hay otros componentes que no son scripts, como los applets o los componentes ActiveX. Los scripts del cliente son por lo general código JavaScript o VBScript, mezclados con código HTML.

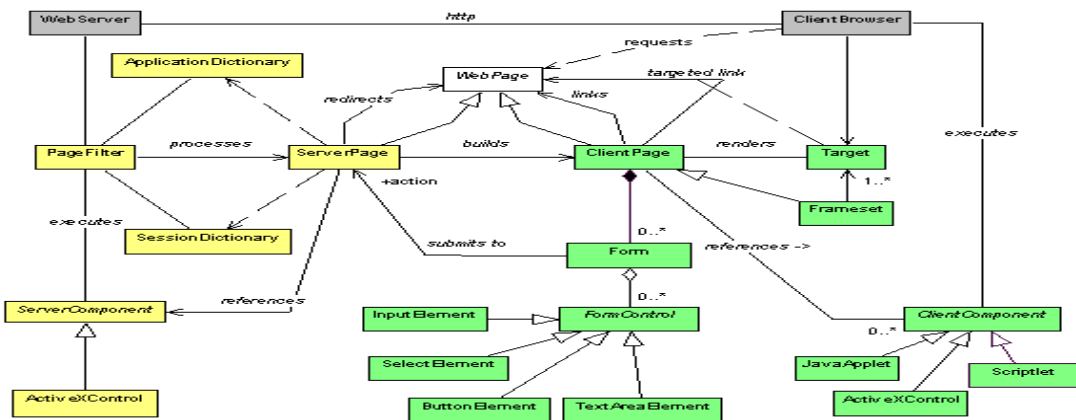
Formularios La forma más común de capturar la información dada por el usuario, es a través de formularios. Un *formulario* (form) es una colección de campos de entrada: textbox, text area, checkbox, radio button group, button y selection list.

Cuando un formulario es llenado, se envía al servidor usando una operación *submit* solicitada por el usuario típicamente al hacer click en un botón.

Servidor Web En muchas aplicaciones Web hay una capa intermedia, compuesta por un conjunto de componentes, que se ejecutan no necesariamente en el servidor Web, sino en otros servidores de aplicaciones. Esta capa encapsula la lógica del negocio, y, al ser componentes compilados puede contener objetos, con sus métodos y atributos (llamados *business objects*).

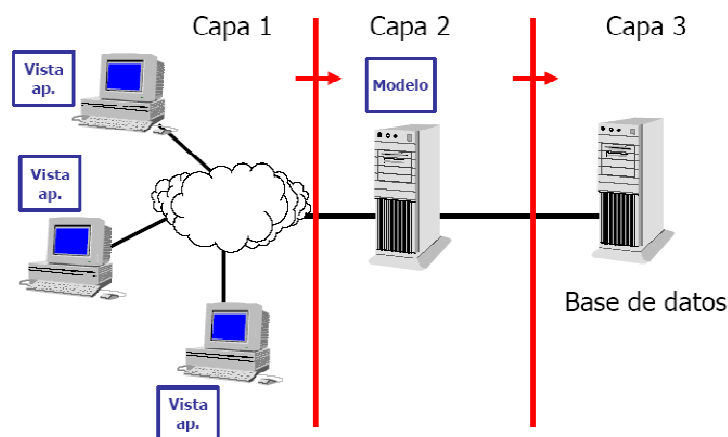


Arquitectura generalizada de una aplicación Web:



4.2 Aplicaciones en capas: La estrategia tradicional de utilizar aplicaciones compactas causa gran cantidad de problemas de integración en sistemas software complejos como pueden ser los sistemas de gestión de una empresa o los sistemas de información integrados consistentes en más de una aplicación. Estas aplicaciones suelen encontrarse con importantes problemas de escalabilidad, disponibilidad, seguridad, integración, entre otros.

Para solventar estos problemas se ha generalizado la división de las aplicaciones en capas que normalmente serán tres: una capa que servirá para guardar los datos (base de datos), una capa para centralizar la lógica de negocio (modelo) y por último una interfaz gráfica que facilite al usuario el uso del sistema.

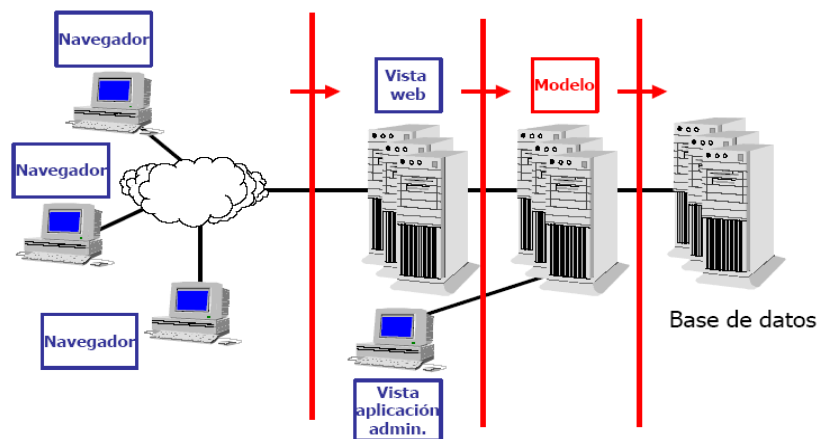


Si establecemos una separación entre la capa de interfaz gráfica (cliente), replicada en cada uno de los entornos de usuario, y la capa modelo, que quedaría centralizada en un servidor de aplicaciones, según el diagrama anterior, esta arquitectura que nos otorga algunas ventajas:

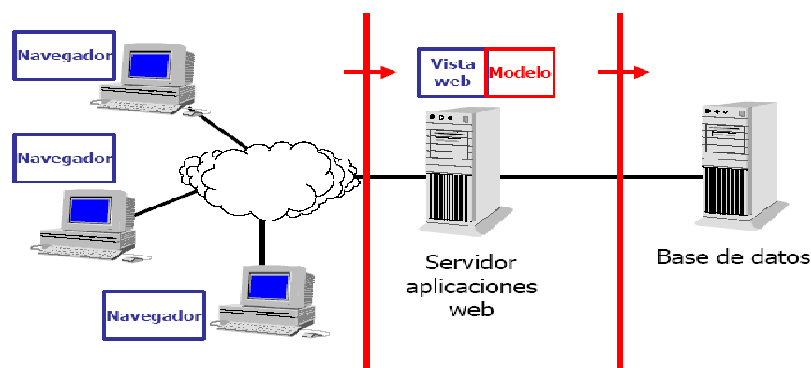
- Centralización de los aspectos de seguridad y transaccionalidad, que serían responsabilidad del modelo.
- No replicación de lógica de negocio en los clientes: esto permite que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios, reduciendo los costes de mantenimiento.
- Mayor sencillez de los clientes.

Si intentamos aplicar esto a las aplicaciones web, debido a la obligatoria sencillez del software cliente que será un navegador web, nos encontramos con una doble posibilidad:

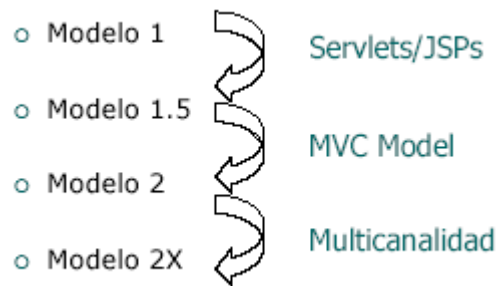
- Crear un modelo de 4 capas, tal y como puede verse en la siguiente figura, separando cliente, servidor web, modelo y almacén de datos. Esto nos permite una mayor extensibilidad en caso de que existan también clientes no web en el sistema, que trabajarían directamente contra el servidor del modelo.



- Mantener el número de capas en 3, como se ve en la siguiente figura, integrando interfaz web y modelo en un mismo servidor aunque conservando su independencia funcional. Ésta es la distribución en capas más común en las aplicaciones web.

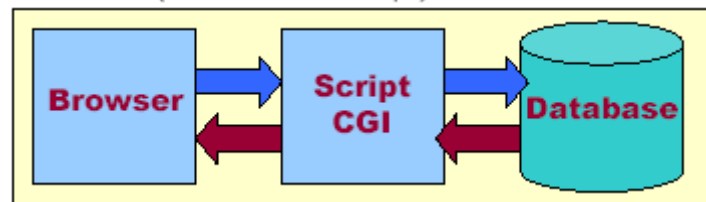


4.3 Evolución de Modelos Arquitectónicos



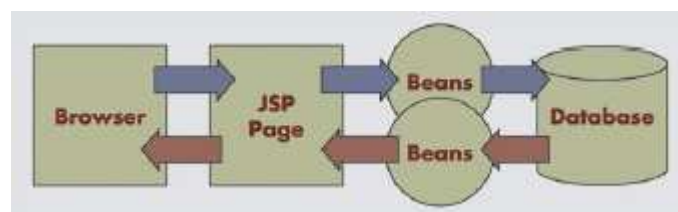
Modelo de Arquitectura 1 Aplicaciones CGI:

- ¡ Las más primitivas
- ¡ Aplicaciones Web CGI
- ¡ Presentación, negocio y persistencia mezclados
- ¡ El estado se almacena en el cliente y cada petición supone una ejecución completa independiente de estado (Transaction Script)



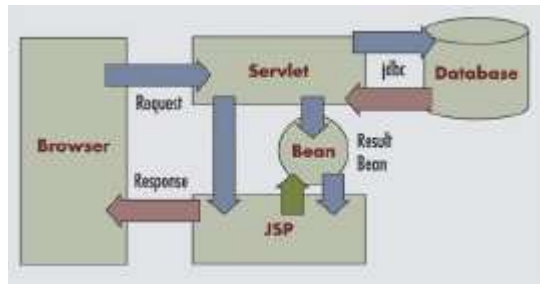
Modelo de Arquitectura 1.5 JSP y Servlets:

- ¡ Separación de responsabilidades
- ¡ JSPs llevan la lógica de presentación (navegabilidad, visualización, etc.)
- ¡ Beans incrustados asumen las responsabilidades de negocio y datos



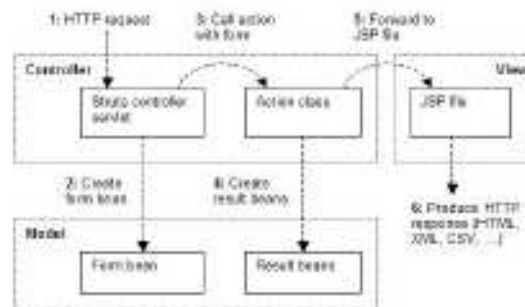
Modelo de Arquitectura 2 MVC:

- ¡ Evolución del modelo 1.5
- ¡ Incorporación del patrón de diseño MVC.
- ¡ Controlador: Navegación
- ¡ Negocio y Datos: Beans
- ¡ Presentación: JSPs



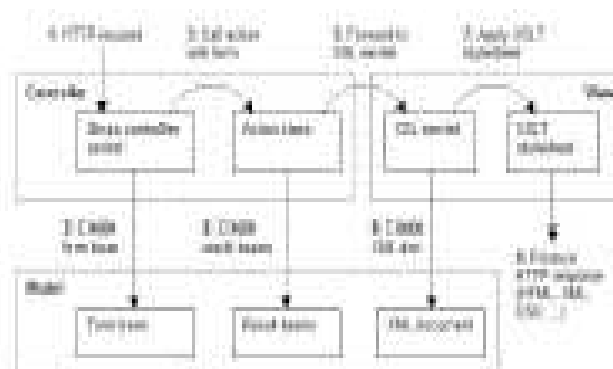
Modelo de Arquitectura 2 MVC con Struts:

- ¡ Struts es la implementación del MVC que aporta Jakarta para aplicaciones web java.
- ¡ <http://jakarta.apache.org/struts>



Modelo de Arquitectura 2X Aplicaciones Multicanal:

- ¡ Evolución del modelo 2 para construir aplicaciones multicanal.
- ¡ Implementación de referencia STXX (extiende Struts)
- ¡ <http://stxx.sourceforge.net/>
- ¡ Soluciones basadas en XML y XSLTs.



4.4 La arquitectura MVC y Model 2

La arquitectura Model-View-Controller surgió como patrón arquitectónico para el desarrollo de interfaces gráficas de usuario en entornos Smalltalk. Su concepto se basaba en separar el modelo de datos de la aplicación de su representación de cara al usuario y de la interacción de éste con la aplicación, mediante la división de la aplicación en tres partes fundamentales:

- El modelo, que contiene la lógica de negocio de la aplicación.
- La vista, que muestra al usuario la información que éste necesita.
- El controlador, que recibe e interpreta la interacción del usuario, actuando sobre modelo y vista de manera adecuada para provocar cambios de estado en la representación interna de los datos, así como en su visualización.

Esta arquitectura ha demostrado ser muy apropiada para las aplicaciones web y especialmente adaptarse bien a las tecnologías proporcionadas por la plataforma J2EE, de manera que:

- El modelo, conteniendo lógica de negocio, sería modelado por un conjunto de clases Java, existiendo dos claras alternativas de implementación, utilizando objetos java tradicionales llamados POJOs (*Plain Old Java Objects*) o bien utilizando EJB (*Enterprise JavaBeans*) en sistemas con unas mayores necesidades de concurrencia o distribución.
- La vista proporcionará una serie de páginas web dinámicamente al cliente, siendo para él simples páginas HTML. Existen múltiples frameworks que generan estas páginas web a partir de distintos formatos, siendo el más extendido el de páginas JSP (*JavaServer Pages*), que mediante un conjunto de tags XML proporcionan un interfaz sencillo y adecuado a clases Java y objetos proporcionados por el servidor de aplicaciones. Esto permite que sean sencillas de desarrollar por personas con conocimientos de HTML. Entre estos tags tienen mención especial la librería estándar JSTL (*JavaServer Pages Standard Tag Library*) que proporciona una gran funcionalidad y versatilidad.
- El controlador en la plataforma J2EE se desarrolla mediante servlets, que hacen de intermediarios entre la vista y el modelo, más versátiles que los JSP para esta función al estar escritos como clases Java normales, evitando mezclar código visual (HTML, XML...) con código Java. Para facilitar la implementación de estos servlets también existe una serie de frameworks que proporcionan soporte a los desarrolladores, entre los que cabe destacar Struts, que con una amplia comunidad de usuarios se ha convertido en el estándar *de facto* en este rol.

Con todo lo anterior, el funcionamiento de una aplicación web J2EE que utilice el patrón arquitectural MVC se puede descomponer en una serie de pasos:

1. El usuario realiza una acción en su navegador, que llega al servidor mediante una petición HTTP y es recibida por un servlet (controlador).

- Esa petición es interpretada y se transforma en la ejecución de código java que delegará al modelo la ejecución de una acción de éste.
2. El modelo recibe las peticiones del controlador, a través de un interfaz o fachada que encapsulará y ocultará la complejidad del modelo al controlador. El resultado de esa petición será devuelto al controlador.
 3. El controlador recibe del modelo el resultado, y en función de éste, selecciona la vista que será mostrada al usuario, y le proporcionará los datos recibidos del modelo y otros datos necesarios para su transformación a HTML. Una vez hecho esto el control pasa a la vista para la realización de esa transformación.
 4. En la vista se realiza la transformación tras recibir los datos del controlador, elaborando la respuesta HTML adecuada para que el usuario la visualice.

Esta arquitectura de aplicaciones otorga varias ventajas clave al desarrollo de aplicaciones web, destacando:

- Al separar de manera clara la lógica de negocio (modelo) de la vista permite la reusabilidad del modelo, de modo que la misma implementación de la lógica de negocio que maneja una aplicación pueda ser usado en otras aplicaciones, sean éstas web o no.
- Permite una sencilla división de roles, dejando que sean diseñadores gráficos sin conocimientos de programación o desarrollo de aplicaciones los que se encarguen de la realización de la capa vista, sin necesidad de mezclar código Java entre el código visual que desarrollen (tan sólo utilizando algunos tags, no muy diferentes de los usados en el código HTML).

Model 2

Sun Microsystems, creadora de la plataforma Java, acuñó el término Model 2 para referirse al modelo arquitectural recomendado para las aplicaciones web desarrolladas sobre J2EE.

Dicha arquitectura consiste en el desarrollo de una aplicación según el patrón Model-View-Controller, pero especificando que el controlador debe estar formado por un único servlet, que centralice el control de todas las peticiones al sistema, y que basándose en la URL de la petición HTTP y en el estado actual del sistema, derive la gestión y control de la petición a una determinada acción de entre las registradas en la capa controlador. Esta centralización del controlador en un único punto de acceso se conoce como patrón front controller.

Las ventajas que este patrón ofrece provienen de la capacidad de gestionar en un único punto la aplicación de filtros a las peticiones, las comprobaciones de seguridad, la realización de logs, etc.

5. TIPOS DE SITIOS WEB:

Todas los sitios Web no sirven al mismo propósito, podemos dividir los sitios en tres grandes grupos: institucionales, comerciales o corporativas, y personales.

Sitios Institucionales: Los sitios institucionales tienen el deber de servir a un grupo de personas, por lo general, diverso y de características variadas.

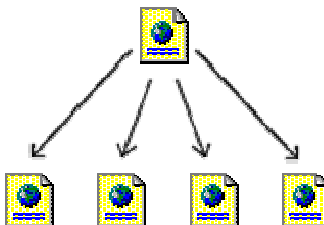
Sitios Comerciales o Corporativos: Los sitios comerciales o corporativos, la información es importante, pero la presentación también: se trata de negocios que buscan vender productos o servicios, por lo que, al mismo tiempo que presentan la información de forma clara, deben hacerlo de forma atractiva. Es en este entorno en el que suelen usarse tecnologías como Flash, Quicktime o Real.

Sitios Personales: En los sitios personales, hay una gran variedad ya que pueden pertenecer tanto a principiantes como a auténticos expertos. Son una forma muy interesante de experimentar y mejorar ya que no están sujetas a las presiones que tienen las páginas institucionales y las comerciales.

6. TIPOS DE ESTRUCTURAS WEB:

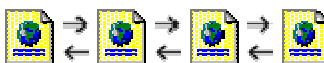
6.1 Jerárquica: La estructura jerárquica, es la típica estructura de árbol, en el que la raíz es la hoja de bienvenida, esta hoja se puede también sustituir por la hoja de contenido, en la que se exponen las diferentes secciones que contendrá nuestro sitio. La selección de una sección nos conduce asimismo a una lista de subtemas que pueden o no dividirse.

Este tipo de organización permite al lector conocer en qué lugar de la estructura se encuentra, además de saber que, con forme se adentra en la estructura obtiene información más específica y que la información más general se encuentra en los niveles superiores.



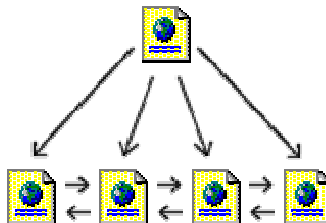
6.2 Lineal: La estructura lineal es la más simple de todas, la manera de recorrerla es la misma que si estuviésemos leyendo un libro, de manera que estando en una página, podemos ir a la siguiente página o a la anterior.

Esta estructura es muy útil cuando queremos que el lector siga un camino fijo y guiado, además impedimos que se distraiga con enlaces a otras páginas. Por otra parte podemos causar a lector la sensación de estar encerrado si el camino es muy largo o poco interesante. Muy utilizado para diseño de tutoriales



6.3 Lineal con jerarquía: Este tipo de estructura es una mezcla de la dos anteriores, los temas y subtemas están organizados de una forma jerárquica, pero uno puede leer todo el contenido de una forma lineal si se desea.

Esto permite tener el contenido organizado jerárquicamente y simultáneamente poder acceder a toda la información de una manera lineal como si estuviésemos leyendo un libro. Esta guía sigue básicamente este tipo de estructura.



6.4 Red: La estructura de red es una organización en la que aparentemente no hay ningún orden establecido, las páginas pueden apuntarse unas a otras sin ningún orden aparente.

Este tipo de organización es la más libre, pero también es la más peligrosa ya que si no se informa al lector de en dónde se encuentra, puede perderse o puede no encontrar lo que anda buscando o no llegar a ver lo que le queremos mostrar. Por eso es muy recomendable asociar la estructura de las páginas con alguna estructura conocida, como por ejemplo la de una ciudad.

