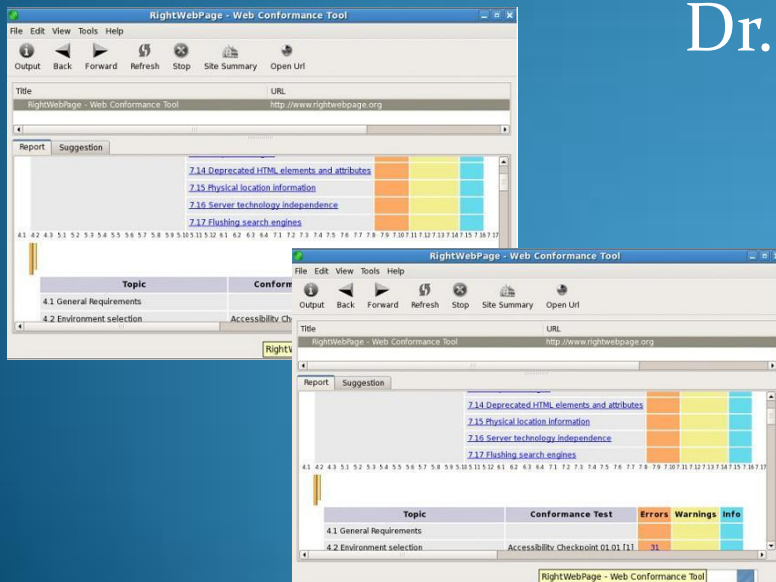


MODELADO DE ANALISIS PARA APLICACIONES WEB

Dr. Ing. Victor Paniagua Gallegos



¿Qué Es?

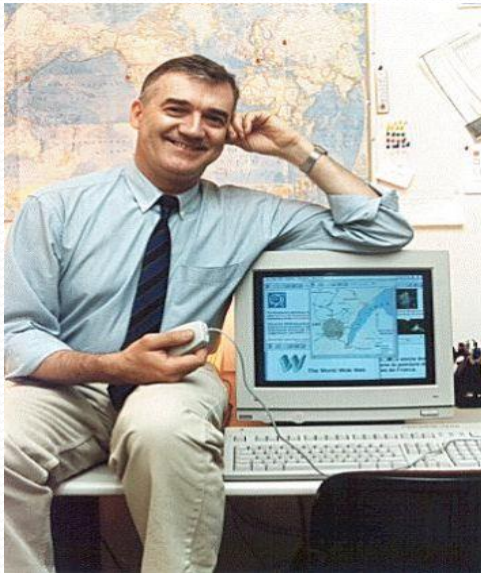
- El análisis de una potencial aplicación web se enfoca en tres preguntas importantes

- 1. ¿Qué información o contenido se presentara o manipulara?**
- 2. ¿Qué funciones realizara el usuario final?**
- 3. ¿Qué comportamientos exhibirá la Webapp conforme presente contenido y realice funciones?**

- Las respuestas se representan como parte de un modelo de análisis que abarca una diversidad de representaciones UML.

¿Quién lo hace?

- Los ingenieros web, los desarrolladores de contenido que no son técnicos y los clientes participan en la creación del modelo de análisis



¿Por qué es importante?

- El modelado de análisis es importante no por que permita que un equipo de ingeniería web desarrolle un modelo concreto de requisitos Webapp (las cosas cambian muy frecuentemente como para que esto sea una expectativa realista), sino que, mas bien, permite que un **ingeniero web** defina aspectos fundamentales del problema: elementos cuyo cambio no es probable (en un futuro cercano).
- El diseño y la construccion se facilitan cuando se comprenden el contenido, la función y el comportamiento fundamentales.

¿Cuáles son los pasos?

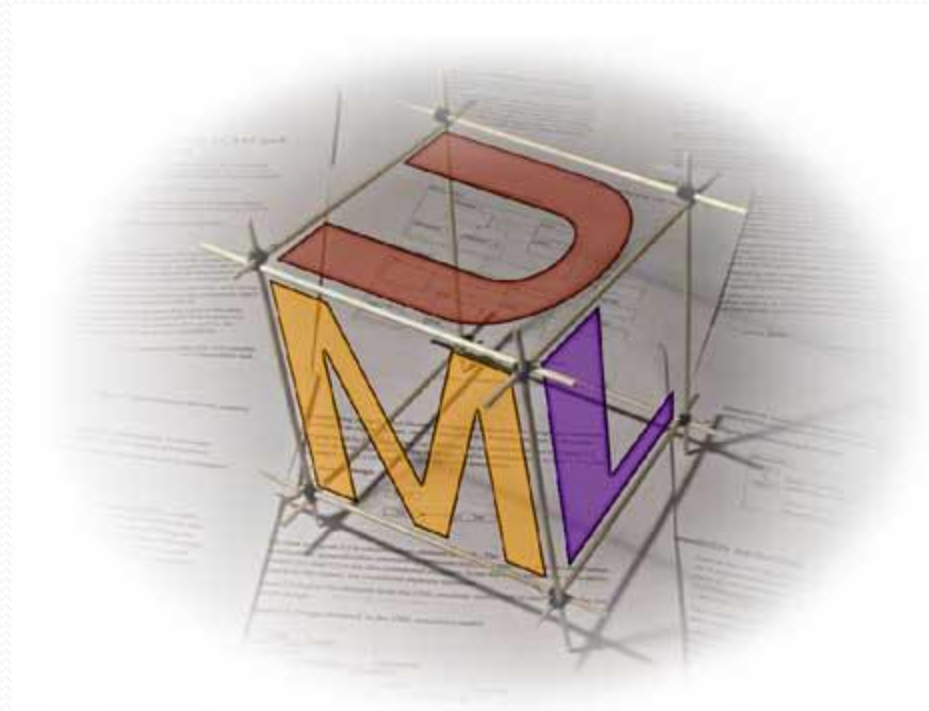
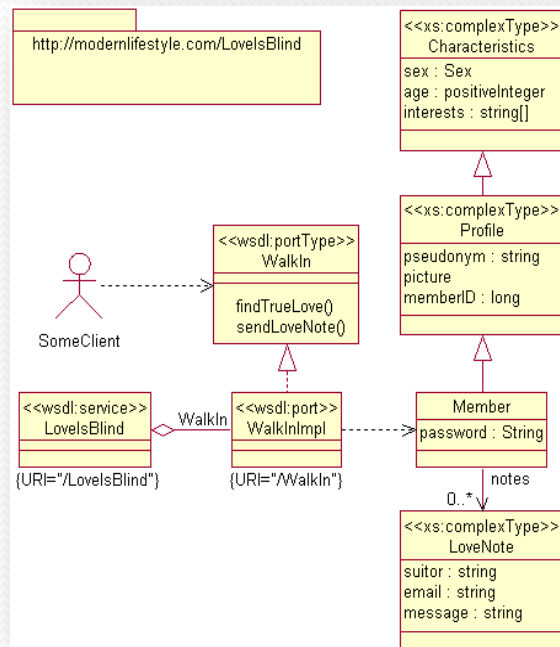
El modelado de análisis se enfoca en los aspectos fundamentales del problema: contenido, interacción, función y configuración.

- **El análisis de contenido** identifica las clases y colaboración de contenido. El análisis de la interacción del usuario, la navegación y los comportamientos del sistema que ocurren.
- **El análisis de las funciones** define las funciones de la Webapp que realizara el usuario y la secuencia de procesamiento que ocurre como consecuencia.
- **El análisis de la configuración** identificada el ambiente(s) operativo en el cual reside la webapp



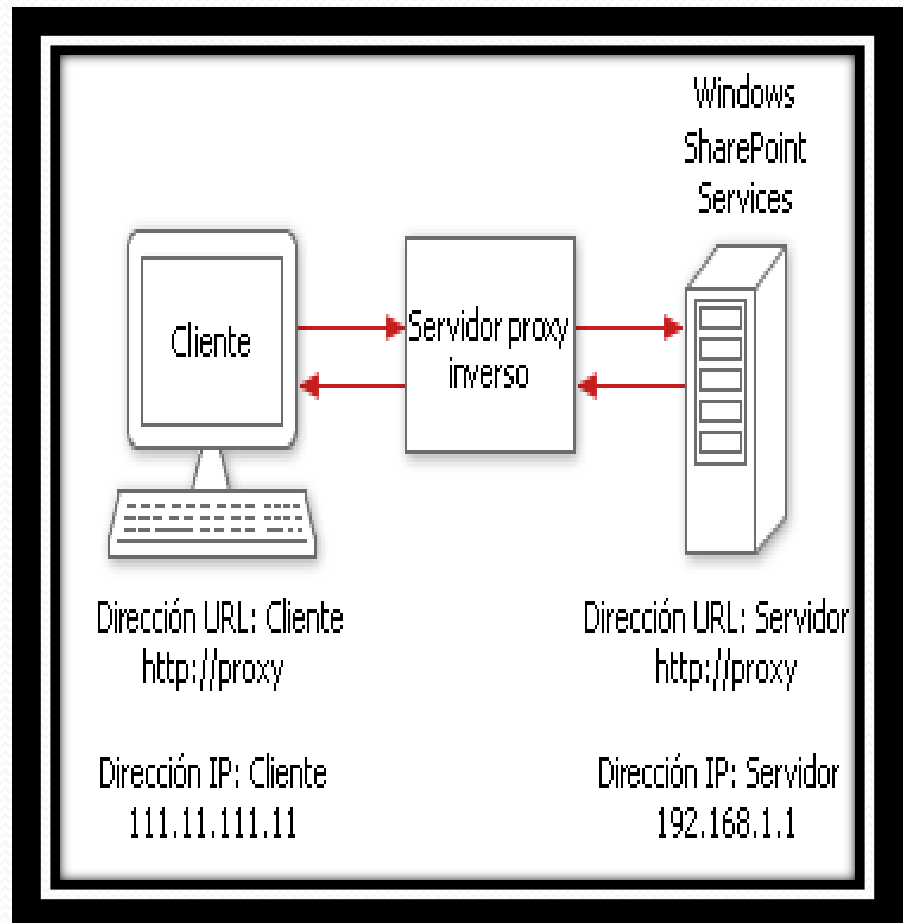
¿Cuál es el producto obtenido?

- El modelado lo integran un conjunto de diagramas y textos UML que describen el contenido, la interacción, la función y la configuración.



¿Cómo puedo estar seguro de que lo he hecho correctamente?

- Los productos obtenidos del modelado de análisis se deben reenviar para corregirlos, completarlos y darles consistencia.



- A primera vista. Existe una aparente contradicción cuando se considera modelado de análisis dentro del contexto de la ingeniería Web.
- Después de todo se ha notado que las WebApps tienen una inmediatez y una volatilidad contraria al modelado detallado, ya sea en la etapa del análisis o en la del diseño


Franklin advierte esta situación y escribe:

Los sitios web por lo general.

- Son complejos y enormemente dinámicos.
- Requieren fases de desarrollo cortas con la finalidad de tener listo el producto y ejecutarlo rápidamente.
- Con frecuencia los desarrolladores van directo a la fase de codificación respecto del servidor con frecuencia se hace *ad hoc*.
- Las tablas de bases de datos se agregan conforme se necesitan y la arquitectura evoluciona a veces en una forma a veces no intencional pero para alguna ingeniería de software no modelada y disciplina lograda que el proceso de desarrollo de software sea mucho mas suave y asegurada que el sistema Web sea mas sustentable en lo futuro.

- Un equipo de ingeniería web debe emprender el modelado de análisis cuando se cumplen la mayoría o todas las condiciones siguientes:

- La webapp que se construirá es grande o compleja.
- El numero de clientes es grande.
- El numero de ingenieros web y otros colaboradores es grande
- Las metas y objetivos (determinados durante la formulación) para la webapp afectaran la línea de referencia del negocio.
- El éxito de la webapp tendrá una fuente conexión con el del negocio.

- 
- Si las condiciones mencionadas no están presentes lo que le resta importancia al modelado de análisis, aprovechar la información obtenida durante la formulación y la recopilación de requisitos sirve como base para la creación de un modelo de diseño para la webapp.
 - En tales circunstancias, tal vez se obtenga un modelado de análisis limitado, pero que terminara incluido en el diseño.

Requisitos para el analisis en las WebApps

1. Requisitos para el análisis de las webapps

El análisis de requisitos para las webapps abarca tres grandes áreas:

- 1- Formulación,
- 2- Recopilación de requisitos y
- 3- Modelado de análisis.

- Durante la formulación se identifican la motivación (metas) y los objetivos básicos para la webapp, y también se definen las categorías de usuarios. Cuando comienza la recopilación de requisitos se intensifica la comunicación entre el equipo de ingeniería web y los accionistas (por ejemplo, clientes, usuarios finales)



1.1. La jerarquía de usuario

- La categoría de los usuarios finales que interactúan con la webapp se identifican como parte de las tareas de formulación y de recopilación de requisitos.
- En la mayoría de los casos, las categorías de usuario son relativamente limitadas y no necesitan una representación UML.
- Sin embargo, cuando crece el número de categorías de usuario, a veces es aconsejable desarrollar una *jerarquía de usuarios*

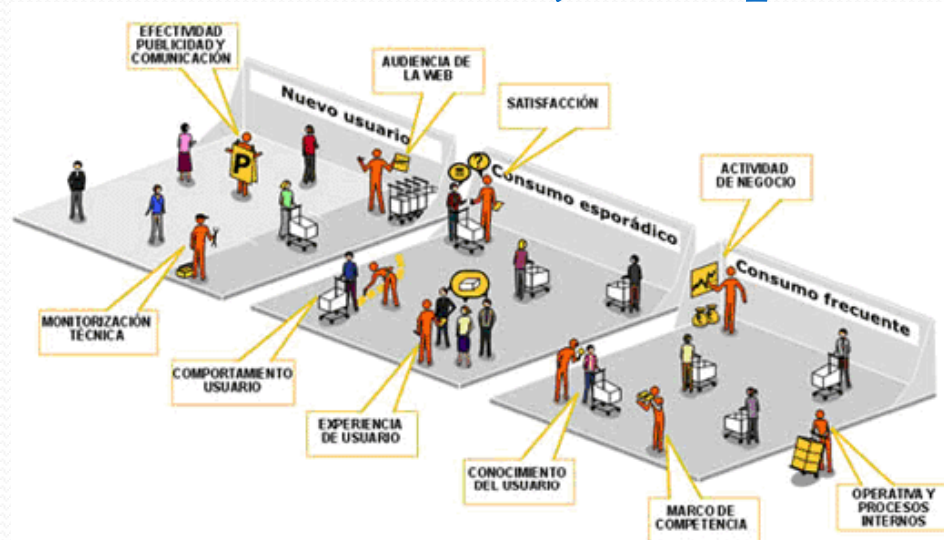
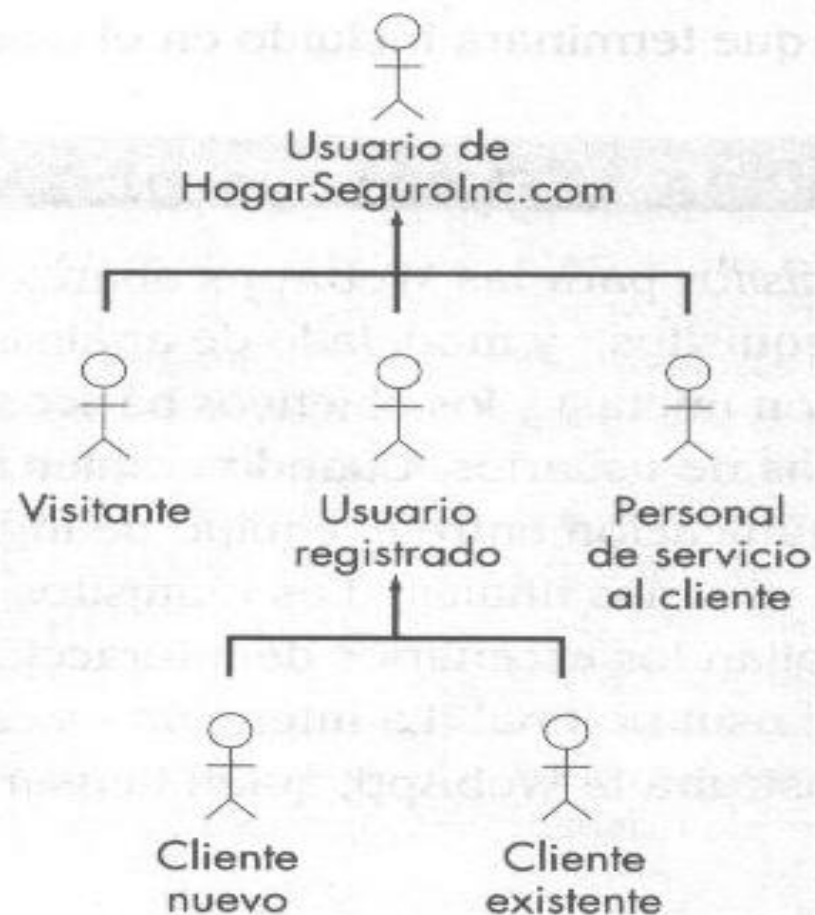


FIGURA 18.1

Jerarquía de usuarios para HogarSeguroInc.com.



Cliente nuevo

- Usuario registrado que quiere personalizar y luego comprar componentes (**y por lo tanto debe interactuar con la webapp de funcionalidad de comercio electrónico**)

Cliente existente

- Un usuario que ya posee componentes y usa la webapp para.
 1. Comprar componentes adicionales
 2. Adquirir información de soporte técnico ó
 3. Contactar con el soporte al cliente

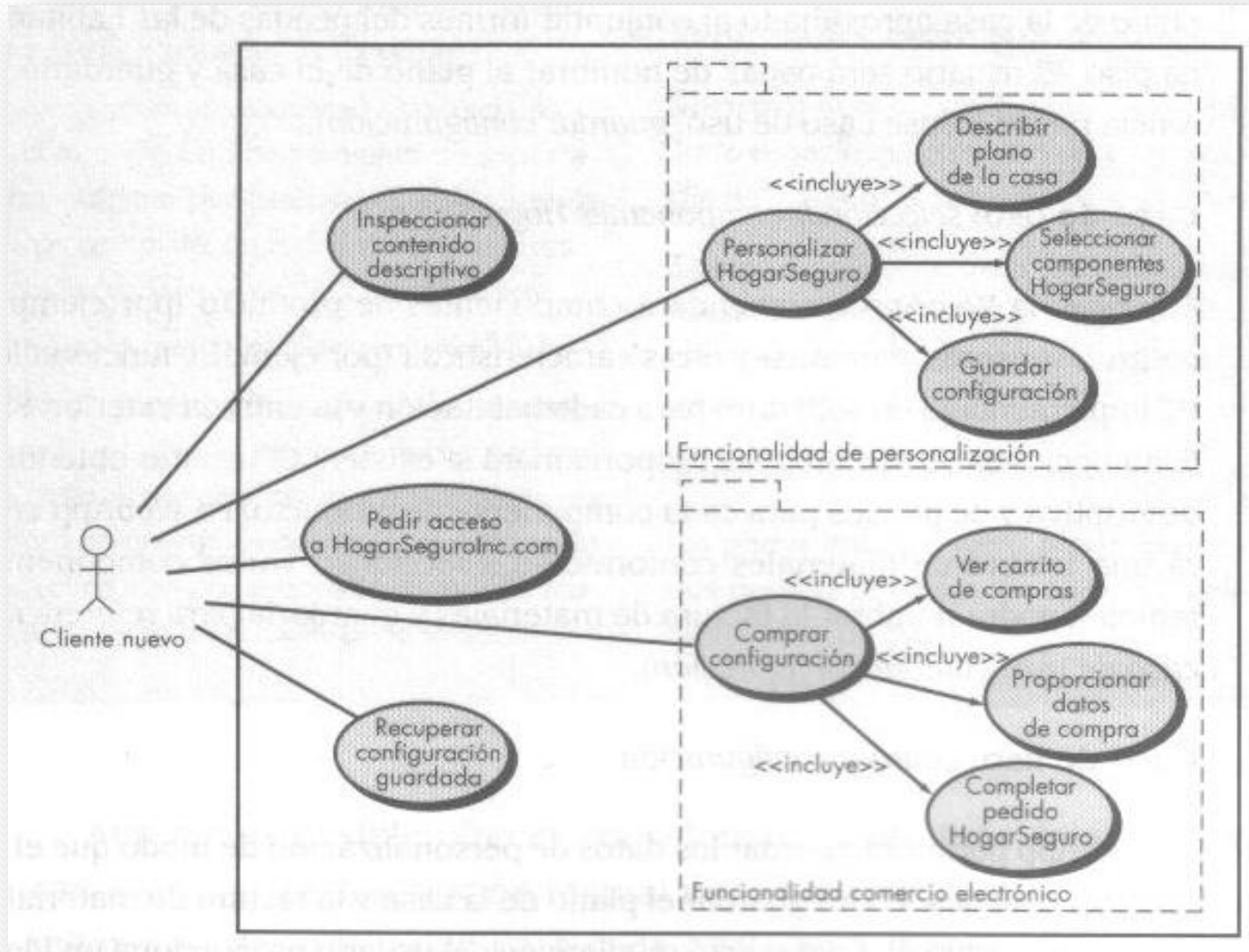
Importante!

Los miembros del **personal de servicio al cliente** son usuario especiales, que también pueden interactuar con el contenido y la pagina web. Conforme asisten a los clientes que han establecido contacto con el soporte técnico de la web.

1.2. Desarrollo de casos de uso

- Franklin se refiere a los casos de uso como “haces de funcionalidad”. Esta descripción captura la esencia de esta importante técnica de modelado de análisis.
- Los casos de uso se desarrollan para cada categoría de usuario descrita en la jerarquía de usuario.
- En el contexto de ingeniería web, el caso de uso en su mismo es relativamente informal: un párrafo narrativo que describe una interacción específica entre usuario y la webapp.

Ejm. Diagrama de caso de uso para Cliente Nuevo




1.3. Afinación del modelo de casos de uso

A la par se crean diagramas de caso de uso para cada categoría de usuario, se desarrolla una vista superior de los requisitos de la webapp observables de manera externa.

Los casos de uso se organizan en paquetes funcionales. Y cada paquete se valora para garantizar que es:

- **comprensible:** todos los clientes entienden el propósito del paquete
- **cohesivo:** el paquete aborda funciones relacionadas cercanamente una con otra.
- **Libremente acoplados:** las funciones o clases dentro del paquete colaboran una con otra, pero la colaboración exterior del paquete se mantiene en un mínimo
- **Jerárquicamente superficial:** las jerarquías funcionales profundas son difíciles de navegar y entender para los usuarios: en consecuencia, el número de niveles dentro de una jerarquía de casos de uso debe reducirse siempre que sea posible

- 
- Puesto que el análisis de requisitos y el modelado son actividades iterativas, es probable que se sumen nuevos casos de uso a los paquetes que se han definido que los casos de uso existentes sean refinados y que casos de uso específicos pueden reubicarse en paquetes diferentes.

2. El modelado de análisis para WebApps

- El modelado de análisis para una WebApp se basa en la información que contienen los casos de uso desarrollados para la aplicación.
- Las descripciones de los casos de uso se analizan gramaticalmente para identificar potenciales clases de análisis y las operaciones y atributos asociados son cada clase.
- Se identifica el contenido que presentara la WebApp y se extraen las funciones que se desarrollan a partir de las descripciones de caso de uso.
- Finalmente, los requisitos específicos de la implementación se deben desarrollar de modo que el ambiente y la infraestructura que apoyan la WebApp puedan construirse.

- Cuatro actividades de análisis, cada una con su aporte a la creación de un modelo de análisis completo, son:
 - **Análisis de contenido:** identifica todo el espectro del contenido que ofrecerá la webapp, el contenido incluye texto, graficas e imágenes, así como datos video y audio.
 - **Análisis de interacción:** describe como interactúa el usuario con la webapp.
 - **Análisis de funciones:** define las operaciones que se aplicaran al contenido de la webapp y describe otras funciones de procedimiento, independientes del contenido pero necesarias para el usuario final.
 - **Análisis de configuración:** describe el ambiente y la infraestructura en la que reside la webapp.

3. El modelo de contenido

- El modelo de contenido contiene elementos estructurales que proporcionan una importante visión de los requisitos de contenido para una webapp. Dichos elementos estructurales incluyen objetos como contenido (por ejemplo: texto imágenes graficas, fotografías, imágenes de video, audio) que se presentan como parte como parte de la webapp además, el modelo de contenido incluye atributos que la describen, operaciones que afectan el comportamiento requerido de la clase y colaboraciones que permiten la comunicación de la clases con otras clases.

3.1. Definición de objetos de contenido

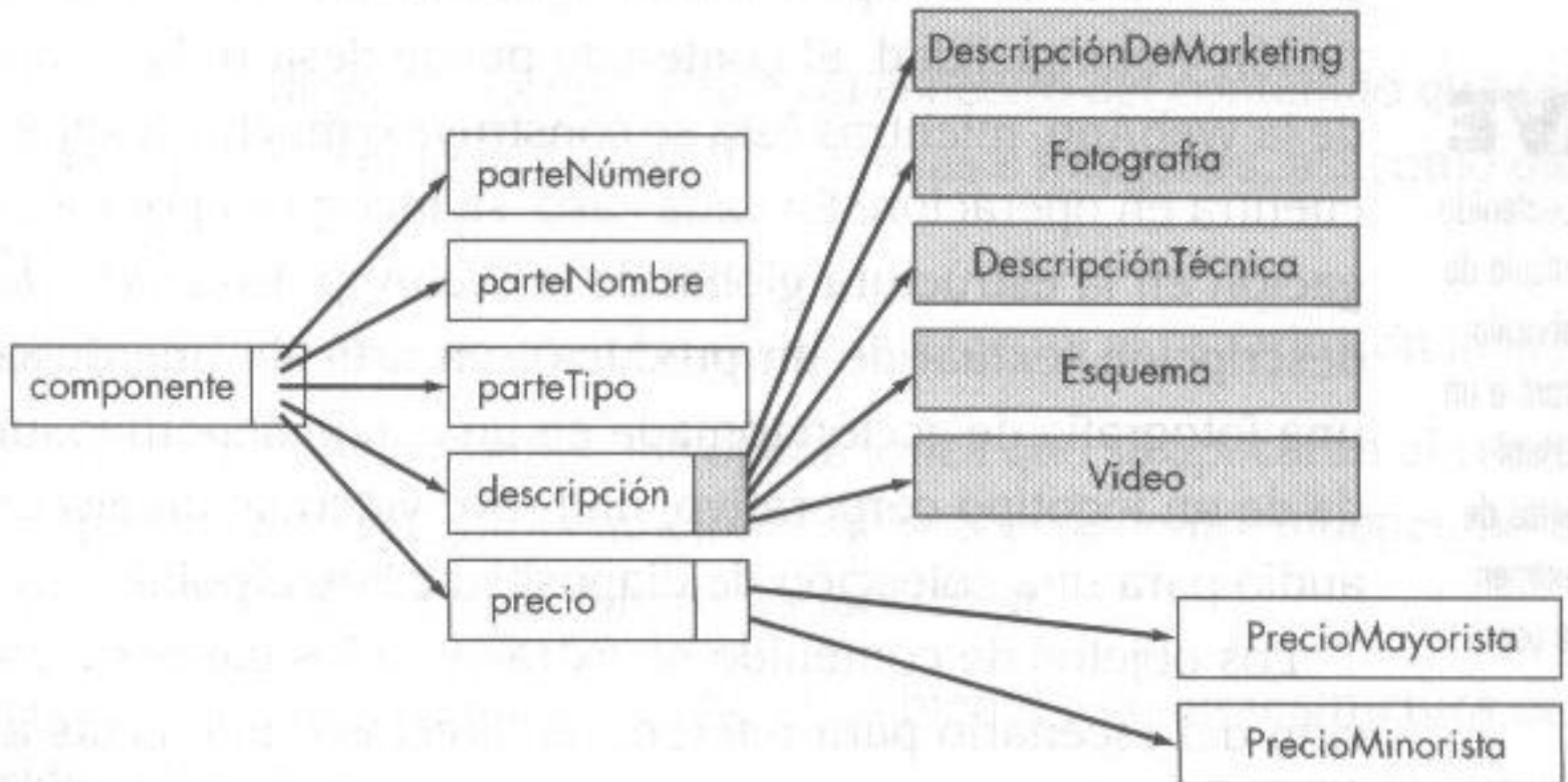
- Las aplicaciones web presentan información preexistente – llamada contenido – a un usuario final.
- El tipo y forma del contenido abarca un amplio espectro de elaboración y complejidad.
- El contenido puede desarrollarse antes de la implementación de la webapp, mientras esta se construye o mucho después de que la webapp se encuentra en operación.
- En cada caso, se incorpora por medio de referencias de navegación en la estructura global de la webapp un objeto de contenido puede ser una descripción textual de un producto, un artículo que describa un evento noticioso una fotografía de acción tomada en un cotejo deportivo, una representación animada de un logotipo corporativo.

- Es importante advertir que cada uno de estos objetos de contenido debe desarrollarse (**con frecuencia a través de desarrolladores de contenido que no son ingenieros web**) o adquirirse para integrarlo en la arquitectura de la webapp

3.2. Relaciones y jerarquía de contenido.

- En muchas instancias, una simple lista de objetos de contenido, pareadas con una breve descripción de cada objeto, es suficiente para definir los requisitos para el contenido que deben diseñarse e implementarse.
- Sin embargo, en algunos casos el modelo del contenido puede contener diagramas de relación de entidades o *arboles de datos* que bosquejan las relaciones entre los objetos de contenido o la jerarquía de este que mantiene una webapp.

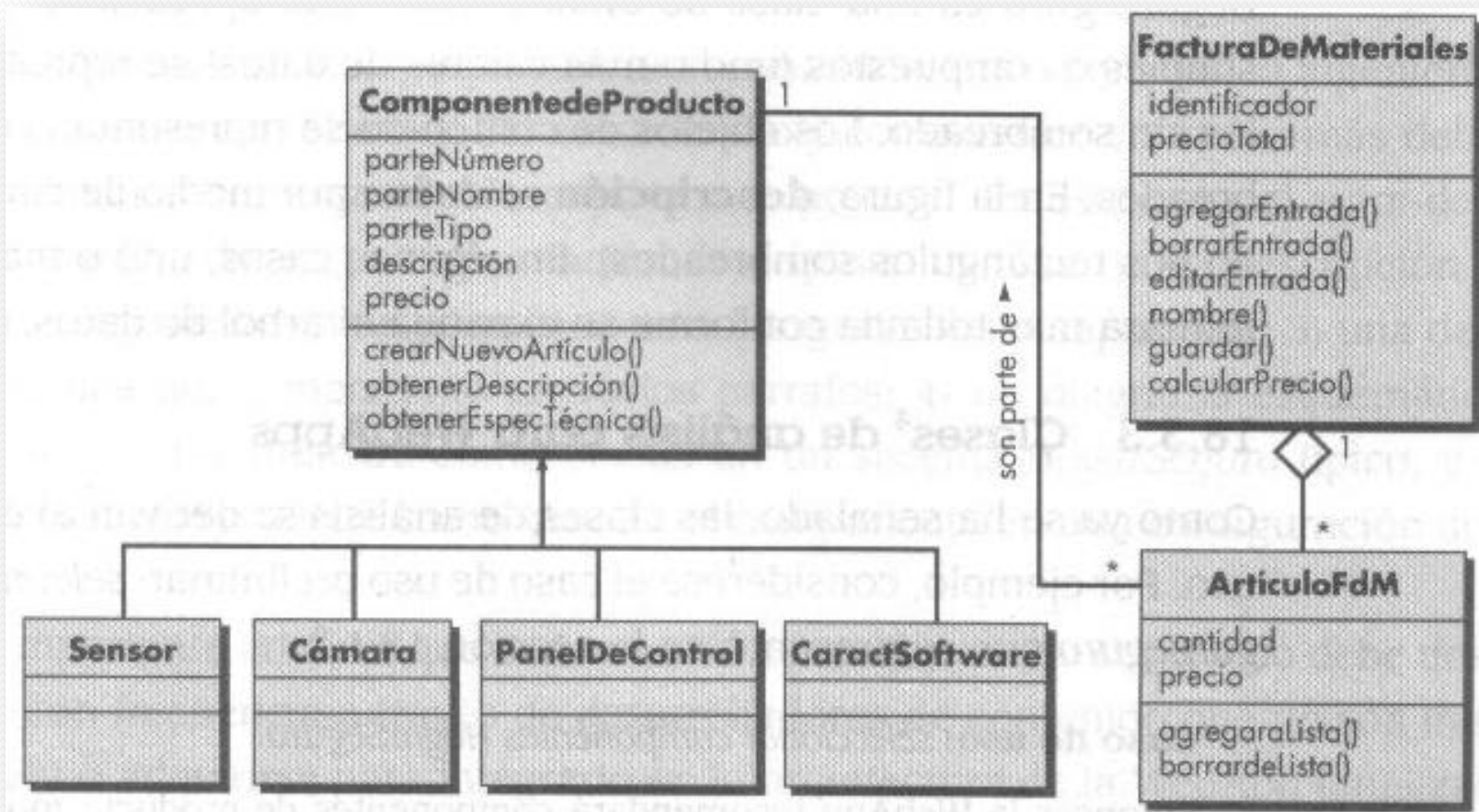
Ejm. Arbol de datos para un componente hogar seguro



3.3. Clases de analisis para webapps

- Como ya se ha señalado, las clases de análisis se derivan al examinar cada caso de uso.
 - **Caso de uso:** *seleccionar componentes*
entonces la webapp recomendará componentes de producto (por ejemplo paneles de control sensores cámaras) y otras características (por ejemplo funcionalidad basada en PC implementada en software) para cada habitación y la entrada exterior. Si el usuario solicita opciones, la webapp las proporcionará si existen

Ejm. Clases de análisis para el caso de uso seleccionar componente Hogar seguro



4. El modelo de interacción

- La gran mayoría de las WebApps permiten una “conversación” entre un usuario final y la funcionalidad, el contenido y el comportamiento de una aplicación.
- Este modelo de interacción lo componen cuatro elementos:
 - 1. Casos de uso,
 - 2. Diagrama de secuencia
 - 3. Diagramas de estado
 - 4. Un diagrama de interfaz de usuario

Casos de uso

- Los casos de uso son el elemento dominante del modelo de interacción para las webapps. No es raro descubrir 100 o mas casos de uso cuando se analizan, diseñan y construyen grandes y complejas webapps. Sin embargo, un porcentaje relativamente bajo de estos casos de uso describe las principales interacciones entre las categorías de usuario final (actores) y el sistema.

Diagrama de secuencia

- Los diagramas de secuencia UML ofrecen una representación abreviada de la forma en la cual las acciones de usuario (los elementos dinámicos de un sistema que definen los casos de uso) colaboran con clases de análisis (elementos estructurales de un sistema que definen los diagramas de clase).
- Dado que las clases de análisis se extraen de las descripciones de caso de uso existe la necesidad de garantizar que hay una forma de realizar un seguimiento entre las clases definidas y los casos de uso que describen la interacción del sistema.



Cliente nuevo

: Habitación

: Plano de la casa

: Componente de Producto

: Factura De Materiales

Almacén de plano de la casa

Almacén de FdM

Describir habitación*

Colocar habitación en plano de la casa

Guardar configuración plano de la planta

Seleccionar componente de producto

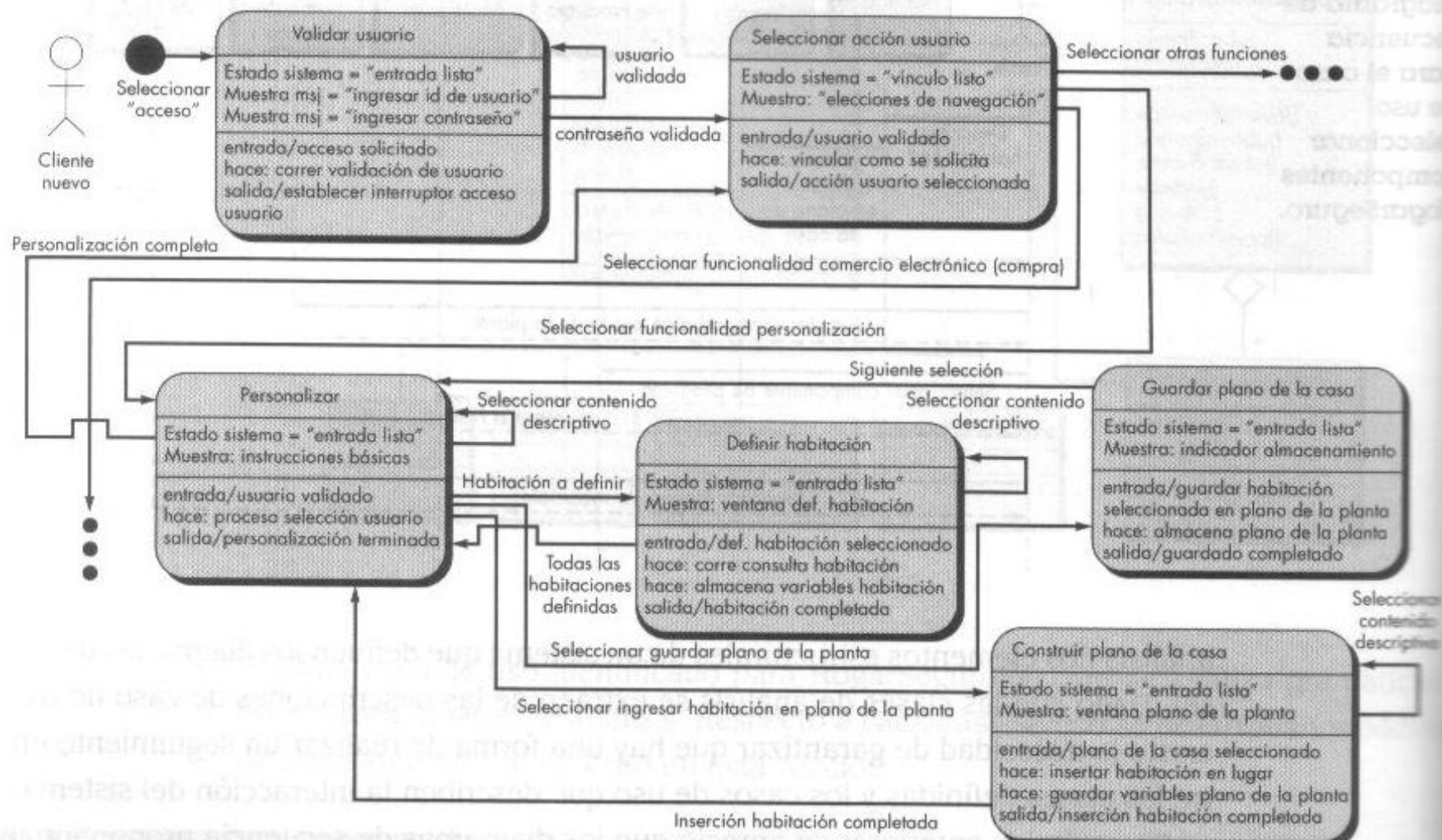
agregar a FdM

Guardar factura de materiales

Diagramas de estado

- El diagrama de estado de UML ofrece otra representación del comportamiento dinámico de la webapp conforme sucede una interacción.
- Al igual que la mayoría de las representaciones de modelado utilizadas en la ingeniería web (o ingeniería del software), el diagrama de estado puede representarse en diferentes grados de abstracción.

FIGURA 18.6 Diagrama de estado parcial para interacción con cliente nuevo.



Prototipo de la interfaz del usuario

- La plantilla de la interfaz de usuario, el contenido que presenta , los mecanismos de interacción que implementa y la estética global de las conexiones usuario-webapp, tienen mucho con la satisfacción del usuario y la aceptación global de la webapp.
- Aunque se puede argumentar que la creación de un prototipo de interfaz de usuario es una actividad de diseño, es una buena idea realizar durante la creación del modelo de análisis mientras mas rápido se pueda revisar la representación física de una interfaz de usuario mayor será la probabilidad de que los usuarios finales obtengan lo que quieren.

5. Modelo funcional

- El *modelo funcional* aborda dos elementos de procesamiento de la webapp y cada uno representa un grado diferente de la abstracción de procedimiento:
 - 1. funcionalidad observable respecto al usuario y que entrega al usuario final la webapp y
 - 2. las operaciones dentro de las clases de análisis que complementan comportamientos asociados con la clases

- **La funcionalidad observable** para el usuario comprende cualesquiera funciones de procesamiento que este inicia directamente. Por ejemplo un sitio web financiero de procesamiento que este inicia directamente por ejemplo, un sitio web financiero puede implementar una variedad de funciones financieras (como una calculadora para fondo de matricula universitaria o una calculadora para fondo de retiro).
- Dichas funciones en realidad pueden implementarse mediante **operaciones dentro de las clases de análisis**, pero desde el punto de vista del usuario final, la función (mas precisamente los datos que proporciona la función es el resultado visible)

6. El modelo de configuración

- Las WebApps se deben diseñar e implementar de forma que se acomoden a una diversidad de ambientes tanto en lado del **servidor** como en la del **cliente**.
- La WebApp puede residir en un servidor que proporcione acceso vía internet, una intranet o una extranet.
- Se deben especificar el hardware del servidor y el ambiente del SO.
- Además, se deben considerar aspectos de interoperabilidad en el lado del servidor. Si la WebApp debe tener acceso a una gran base de datos o interoperar con las aplicaciones corporativas existentes en el lado del servidor, se deben especificar las interfaces apropiadas, los protocolos de comunicación y la información complementaria necesaria.

- En algunos casos, el modelo de configuración no es mas que una lista de atributos tanto del lado del servidor como del cliente.
- Sin embargo, para webapps mas elaboradas, varias complejidades de configuración (por ejemplo: distribución de carga entre múltiples servidores, arquitecturas de cache, base de datos remotas, múltiples servidores que sirven a varios objetos en una misma pagina web) pueden impactar el análisis y el diseño. Es factible aprovechar el diagrama de despliegue UML en situaciones e las cuales se deban considerar arquitecturas de configuración complejas.


7. Análisis Relacion-Navegacion

- Los elementos del modelo de análisis descritos en las secciones previas identifican los elementos de contenido y funcionalidad, junto con la forma en que se utilizan para implementar la interacción con el usuario. Conforme el análisis evoluciona en diseño, dichos elementos se vuelven parte de la arquitectura de la webapp. En el contexto de las aplicaciones web, cada elemento arquitectónico tiene el potencial de vincularse con todos los otros elementos arquitectónicos. Pero conforme aumenta el numero de vínculos, la complejidad de navegación a través de la webapp también crece. Entonces, la pregunta es como establecer los vínculos apropiados entre los objetos de contenido y las funciones que proporcionan las capacidades que requiere el usuario

- El *análisis relacion-navegacion* ARN proporciona a los analistas de sistemas una técnica sistemática para determinar la estructura de relación de una aplicación, lo que les ayuda a descubrir las relaciones potencialmente útiles en los dominios de la aplicación y que se pueden implementar como vínculos mas adelante. El ARN también ayuda a determinar las estructuras de navegación apropiadas sobre estos vínculos. El ARN fomenta la comprensión de los desarrolladores de sistemas en torno a los dominós de la aplicación en ampliar y profundizar su modelo conceptual del dominio. Entonces los desarrolladores pueden mejorar su implementación al incluir vínculos, meta informáticos y navegación adicionales

El enfoque ARN se organiza en 5 pasos:

- **Análisis de los participantes:** identifica las diversas categorías de usuario y establece una apropiada jerarquía de participante
- **Análisis de elementos:** identifica los objetos de contenido y los elementos funcionales de interés para los usuarios finales
- **Análisis de relaciones:** describe las relaciones entre los elementos de la webapp.
- **Análisis de navegación:** examina como los usuarios pueden acceder a elementos individuales o grupos de elementos.
- **Análisis de evaluación:** considera temas pragmáticos (por ejemplo costo beneficio) asociados con la implementación de las relaciones definidas con anterioridad.

- 
- Los primeros dos pasos en el enfoque ARN se trataron en párrafos anteriores de este capítulo. En las siguientes secciones se consideran métodos para establecer las relaciones entre los objetos de contenido y las funciones.

7.1. Análisis de relaciones: preguntas clave.

- Yoo y Bieber sugieren una lista de preguntas que un ingeniero web o analista de sistemas deben responder acerca de cada elemento (objeto de contenido o función) identificado dentro del modelo de análisis. La siguiente lista adaptada para webapps, es representativa.
 - ¿el elemento es miembro de una categoría de elemento mas amplia?
 - ¿Qué atributos o parámetros se han identificado para el elemento?

- ¿ya existe información descriptiva acerca del elemento? Si es así ¿Dónde esta?
- ¿El elemento aparece en diferentes ubicaciones dentro de la webapp? Si es así ¿donde?
- ¿el elemento lo componen otros pequeños elementos? Si es así ¿Cuáles son?
- ¿el elemento es miembro de una colección de elementos mayor? Si es así ¿Cuál es y cual es su estructura?
- ¿al elemento lo describe una clase de análisis?
- ¿otros elementos son similares al elemento considerado? Si es así ¿es posible que pudieran combinarse en un elemento?

- ¿el elemento se usa en un ordenamiento específico de otros elementos? ¿su aparición depende de otros elementos?
- ¿otro elemento siempre sigue a la aparición del elemento considerado?
- ¿Qué condiciones previas y posteriores se deben satisfacer para utilizar el elemento?
- ¿categorías de usuario específicas aprovechan al elemento? ¿las diferentes categorías de usuario emplean de manera diferente al elemento? Si es así ¿Cómo?
- ¿el elemento puede estar asociado con una meta u objetivo de formulación específico? ¿con un requisito webapp específico?

- ¿este elemento siempre aparece al mismo tiempo que aparecen otros elementos? Si es así ¿Cuáles son los otros elementos?
- ¿este elemento siempre aparece en el mismo lugar (por ejemplo misma ubicación de la pantalla o pagina) que otros elementos?

las respuestas a estas otras preguntas ayudan al ingeniero web a posicionar el elemento en cuestión dentro de la webapp y a establecer relaciones entre elementos.

Es posible desarrollar una relación taxonómica y categorizar cada relación identificada debido a las preguntas anotadas. El lector interesado debe remitirse a para mas allá.

7.2. Analisis de navegacion

- Una vez que entre los elementos se han desarrollado relaciones definidas dentro del modelo de análisis, el ingeniero web debe considerar los requisitos que dictan como navegara cada categoría de usuario de un elemento (por ejemplo el contenido) a otro. Los mecanismos de navegación se definen como parte del diseño.
- **En esta etapa los desarrolladores deben considerar requisitos de navegación globales las siguientes preguntas se deben plantear y responder:**



- ¿ciertos elementos deben ser mas fáciles de alcanzar (es decir, requieren menos pasos de navegación) que otros? ¿Cuál es la prioridad de presentación?
- ¿ciertos elementos deben resaltarse para forzar a los usuarios a navegar en su dirección?
- ¿Cómo se manejaran los errores de navegación?
- ¿la navegación hacia grupos de elementos relacionados debe ser prioritaria sobre la navegación hacia un elemento específico?
- ¿la navegación se debe lograr por medio de vínculos, de acceso basado en búsqueda de otros medios?

- ¿ciertos elementos deben presentar a los usuarios con base en el contexto de acciones de navegación previas?
- ¿el acceso a la navegación debe mantenerse para los usuarios?
- ¿en cada punto de la interacción del usuario debe estar disponible un mapa o menú de navegación completo (en oposición a un simple vínculo de retroceso o puntero dirigido)?
- ¿el diseño de la navegación debe nutrirse de los comportamientos de usuario mas comúnmente esperados o mediante la importancia percibida de los elementos webapp definidos?

- ¿un usuario puede “almacenar” su navegación previa a través de la webapp para un uso futuro expedito?
- ¿para que categoría de usuario se debe diseñar una navegación optima?
- ¿Cómo se manejaran los vínculos externos a la webapp? ¿superponiendo la ventana de navegador existente? ¿Cómo nueva ventana de navegador como un marco separado?

- Estas y muchas preguntas se deben plantear y responder como parte del análisis de navegación
 - El equipo de ingeniería web y sus participantes también deben determinar los requisitos globales para la navegación. Por ejemplo ¿se proporciona una mapa del sitio para brindar a los usuarios un panorama integral de la estructura de la webapp? ¿el usuario puede realizar un recorrido que subraye los elementos mas importantes (objetos de contenido y funciones) disponibles?

Fin ...

