

http://commons.wikimedia.org/wiki/File:Joy_Oil_gas_station_blueprints.jpg

Tema 5.2

Diseño conceptual

Interfaces Persona Computador
Depto. Sistemas Informáticos y Computación UPV

DOCENCIA VIRTUAL

Finalidad:

Prestación del servicio Público de educación superior (art. 1 LOU)

Responsable:

Universitat Politècnica de València.

Derechos de acceso, rectificación, supresión, portabilidad, limitación u oposición al tratamiento conforme a políticas de privacidad:

<http://www.upv.es/contenidos/DPD/>

Propiedad intelectual:

Uso exclusivo en el entorno de aula virtual.

Queda prohibida la difusión, distribución o divulgación de la grabación de las clases y particularmente su compartición en redes sociales o servicios dedicados a compartir apuntes.

La infracción de esta prohibición puede generar responsabilidad disciplinaria, administrativa o civil



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



En este vídeo aprenderemos a realizar el diseño conceptual de nuestra aplicación. Como resultado obtendremos un prototipo de baja fidelidad de nuestra aplicación.

Objetivos de aprendizaje

- ▶ Entender las dificultades de traducir los requisitos a una interfaz gráfica de usuario
- ▶ Aprender la importancia del diseño conceptual
- ▶ Estudiar una técnica para describir la estructura subyacente a una interfaz de usuario
- ▶ Ser capaz de generar un diagrama de contenidos a partir de un conjunto de casos de uso concretos

2

Antes de comenzar con la segunda parte 2 del tema 5, echemos la vista atrás. ¿Qué hemos visto hasta ahora? Hemos hablado de características de las buenas interfaces, hemos hablado de usabilidad y accesibilidad, hemos hablado de requisitos, insistiendo en que no podemos hacer una interfaz para una aplicación si no sabemos para qué se va a utilizar esa interfaz, sin saber que quién la va a utilizar, sin conocer cuáles son las necesidades del usuario. En tema 4 ya vimos algo de esto. Se habló de diferentes técnicas para obtener los requisitos de una aplicación. Vale, a partir de ahora suponemos que ya tenemos esa información. Ya sabemos cómo son nuestros usuarios y sabemos qué necesitan. ¿nos ponemos ya a dibujar, a hacer prototipos? ¿Podemos hacerlo? ¿Cuántas pantallas vamos a utilizar? ¿Qué objetos vamos a manejar en cada una de ellas? ¿Cómo vamos a agrupar la información? ¿Qué funcionalidad va a tener cada pantalla? ¿qué estilos de interacción vamos a utilizar? ¿Qué controles van a ser más adecuados? Demasiadas preguntas a las que hay que dar respuesta antes de empezar a dibujar esas pantallas.

Es cierto que el diseño visual de la ventanas de nuestra interfaz es uno de nuestros objetivos principales, pero no podemos dar un salto en el vacío. Para pasar de los requisitos a las ventanas finales hace falta una etapa intermedia, y esta etapa o proceso es el diseño conceptual.

Índice

- ▶ Introducción
- ▶ Diseño conceptual: el diagrama de contenidos
 1. Identificar los objetos de tarea primarios, los atributos y las acciones
 2. Identificar los contenedores y los objetos de tarea que van en cada uno de ellos
 3. Conectar los contenedores para mostrar el flujo de la navegación
- ▶ Referencias

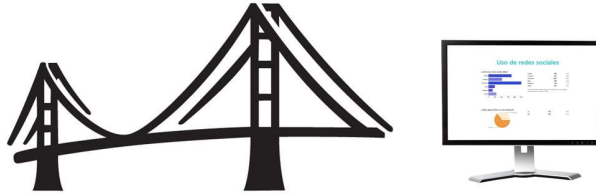
3

En este tema veremos una pequeña introducción y después pasaremos a ver los pasos necesarios para obtener el diagrama de contenidos, que como iremos viendo en las transparencias siguientes, es el objetivo de la fase de diseño conceptual

Introducción

- ▶ Este tema describe técnicas para pasar del análisis de requisitos al diseño físico de una interfaz de usuario

"El Sistema debe ser capaz de crear un informe de ventas mensual"



Introducción

▶ Reingeniería del trabajo:

- A menudo se desarrollará una nueva versión de un sistema existente
- Quizás los usuarios tengan que empezar a trabajar de forma diferente
 - Gestión sensible: involucrar a los usuarios en el proceso de desarrollo

▶ Objetivos

- Concienciarse de la potencia y eficiencia que hace posible la automatización
- Rediseñar el trabajo para dar un apoyo más efectivo a los objetivos del negocio
- Minimizar el reentrenamiento: aprovechar el conocimiento actual del usuario y tener en cuenta las limitaciones cognitivas y las capacidades del usuario al diseñar nuevas tarea

5

Durante esta etapa, entre el análisis de requisitos y el diseño físico de la interfaz, nos vamos a plantear una serie de cuestiones. Sabemos qué tareas necesitamos realizar con el nuevo sistema, es decir los requisitos funcionales, y conocemos cómo se han estado llevando a cabo estas tareas hasta la fecha, esta forma de llevar a cabo las tareas se plasmó en lo que llamamos escenarios de tarea, que ya vimos en el tema 4. Pero esa forma de hacer las tareas ¿era la más adecuada? Quizá no, quizá haya una forma mejor y más eficiente de llevar a cabo una determinada tarea. Es cierto que hay que aprovechar la experiencia del usuario, que hay que dejarle, en la medida de lo posible, libertad a la hora de realizar una tarea, pero también es posible que durante este periodo nos demos cuenta de que las tareas no se realizan de una forma adecuada ni eficiente y que, a pesar del usuario, hay que aprender a hacer las cosas de una manera diferente. Esto es lo que se llama Reingeniería del trabajo.

Ejemplo: biblioteca digital

► La fase de análisis ha producido: escenarios de tarea:

- Escenario de tarea. Buscar y pedir un recurso
 - Julia, una profesora del departamento, está buscando un CD-ROM que contiene ejemplos y ejercicios de Diseño y Análisis Orientado a Objetos. Sabe que Tom, otro profesor, imparte principalmente Diseño y Análisis Orientado a Objetos, así que llama a su puerta. Desafortunadamente no está en su despacho, así que le deja una nota en su puerta. Al regresar, Tom la busca y la encuentra en la cafetería. Le dice a Julia que Geoff tiene el CD-ROM. Desafortunadamente Geoff está de baja, así que Julia lo llama por teléfono, y él le promete que se lo mandará por correo.
- Escenario de tarea. Ver actualizaciones y pedir recursos.
 - Mark ha vuelto tras una estancia de 6 meses y quiere saber qué libros han comprado otros miembros del departamento durante su ausencia. Para ello, telefonea a cada miembro del departamento y concierta una cita. Debe hacerlo así porque cada profesor está en la universidad a distintas horas. Entonces se encuentra con cada uno y revisa su librería, pidiendo prestados libros que le interesan. Sólo pide un libro cada vez, ¡ya que lee despacio!

6

Antes de continuar con el diseño conceptual recordemos el ejemplo que se vio en el tema 4.

Recordamos que un escenario de tarea es una descripción narrativa de una tarea tal como se realiza en la actualidad, es decir con el sistema actual.

Aquí tenemos los dos escenarios de tarea que vimos.

Ejemplo: biblioteca digital

- ▶ La fase de análisis ha producido: escenarios de uso:
 - Escenario de uso. Buscar y pedir un recurso
 - Julia, está buscando un CD-ROM concreto que contiene ejemplos y ejercicios de Diseño y Análisis Orientado a Objetos. Accede a la biblioteca digital desde casa e introduce la cadena de búsqueda: 'Análisis orientado a objetos'. El sistema obtiene un resultado. Geoff tiene el correspondiente CD-ROM. Entonces Julia le envía un correo electrónico a Geoff pidiéndole que le preste el CD-ROM.
 - Escenario de uso. Ver actualizaciones y pedir recursos.
 - Mark ha vuelto recientemente de una estancia y quiere saber cuáles son los últimos recursos que se han añadido a la biblioteca digital. Selecciona 'Comprobar actualizaciones', identifica los libros en los que está interesado y envía un correo electrónico al propietario de libro que más le interesa.

7

Y aquí tenemos los dos escenarios de uso correspondientes a los dos escenarios de tarea anteriores. Recordamos que un escenario de uso es similar a un escenario de tarea, pero con el uso previsto en el sistema que se va a desarrollar.

Ejemplo: biblioteca digital

- ▶ La fase de análisis ha producido: casos de uso concretos

Acción del usuario	Respuesta del sistema
<i>El profesor introduce uno o más parámetros de búsqueda para el CD-ROM: título, año y plataforma</i>	<i>El sistema muestra los resultados de la búsqueda</i>
<i>El profesor selecciona un resultado</i>	<i>El sistema muestra los detalles del CD-ROM y los datos de contacto del dueño, que es un estudiante de doctorado</i>
<i>El profesor selecciona la dirección de correo</i>	<i>El sistema muestra un área de mensaje</i>
<i>El profesor escribe y manda la petición por e-mail</i>	<i>El sistema confirma el envío de la petición</i>

8

Y por último recordamos los casos de uso. Un caso de uso describe la funcionalidad del nuevo sistema en términos de la interacción entre el usuario y el sistema. El caso de uso se puede describir de diferentes formas, pero una forma habitual de hacerlo es en forma de tabla, como el caso de uso de la imagen, donde tenemos por un lado las acciones que realiza el usuario y por otro, la respuesta del sistema.

Ejemplo: biblioteca digital

- ▶ La fase de análisis ha producido: casos de uso concretos

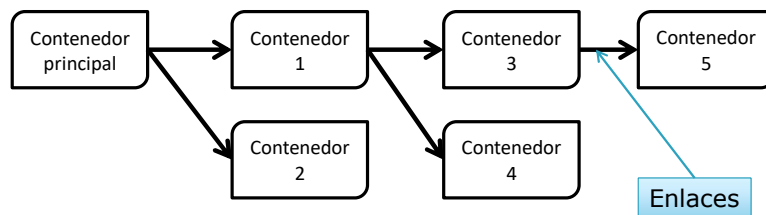
Acción del usuario	Respuesta del sistema
<i>El estudiante de doctorado solicita las últimas actualizaciones de la biblioteca digital</i>	<i>El sistema muestra la disponibilidad de los últimos libros, CD-ROMs, vídeos y revistas</i>
<i>El estudiante selecciona el año del libro de su autor favorito: J. Nielsen</i>	<i>El sistema muestra todos los detalles del libro y los datos de contacto del dueño, incluido su nombre, que es un profesor</i>
<i>El estudiante selecciona la dirección de correo del profesor</i>	<i>El sistema muestra un área de mensaje</i>
<i>El estudiante escribe y manda una petición del libro por e-mail</i>	<i>El sistema confirma el envío de la petición</i>

9

Hasta aquí es lo que hemos obtenido en la fase de análisis. Pero como ya he comentado al principio del tema, el paso de los casos de uso a las pantallas físicas (diseño visual) no es inmediato hay que tomar muchas decisiones, y por tanto necesitamos alguna metodología que nos ayude en la toma de estas decisiones.

Diseño conceptual

- ▶ El diseño conceptual es el proceso de establecer la organización y estructura subyacentes a la interfaz de usuario
 - Decidir que funcionalidad debe soportar cada pantalla
- ▶ Un diagrama de contenidos es un prototipo de baja fidelidad que representa la organización y la estructura de una interfaz desde el punto de vista del diseñador



10

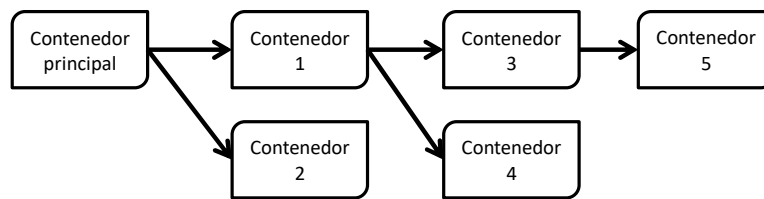
El diseño conceptual es el proceso que nos va a llevar de los casos de uso a al diseño de nuestra interfaz, es decir, el proceso que partiendo del resultado de la fase de análisis nos va a ayudar a decidir el diseño de la interfaz. Por ejemplo, en el caso de una interfaz basada en pantallas, nos ayudará a decidir la funcionalidad que va a soportar cada una de las pantallas, y como será la navegación del usuario entre ellas.

Así, el resultado de esta fase de Diseño Conceptual es la obtención de lo que llamamos diagrama de contenidos, el cual es un prototipo de baja fidelidad que representa la organización y la estructura subyacente de la interfaz.

Diseño conceptual

► Diagrama de contenidos

- Contenedor: representación abstracta de parte del trabajo del usuario y las funciones que lo soportan
 - Ejemplo: introducir los criterios de búsqueda de un libro
- Enlaces: representan la navegación del usuario entre las distintas áreas funcionales dentro de la interfaz



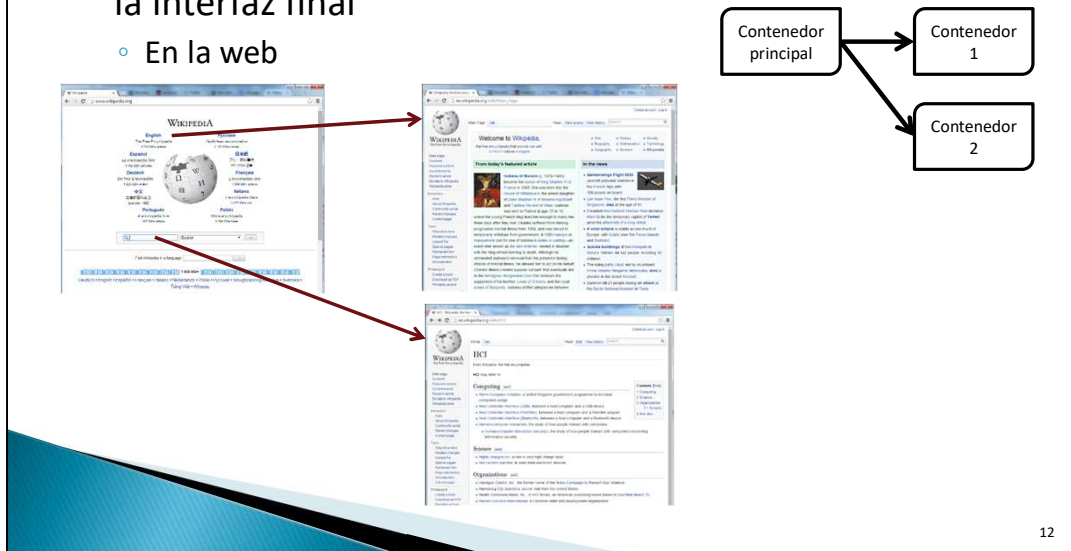
11

El diagrama de contenidos es independiente de dispositivo y de los estilos de interacción que se vayan a utilizar. Está formado por los siguientes elementos: Contenedores y enlaces. Los contenedores no representan pantallas, sino parte de la funcionalidad que necesita el usuario. Los enlaces representan como los usuarios pueden alcanzar un contenedor desde otro, es decir, como pueden navegar entre ellos.

Un diagrama de contenidos puede realizarse para toda la aplicación, o solo para parte de su funcionalidad.

Diseño conceptual

- Posibles relaciones entre el diagrama de contenidos y la interfaz final
 - En la web



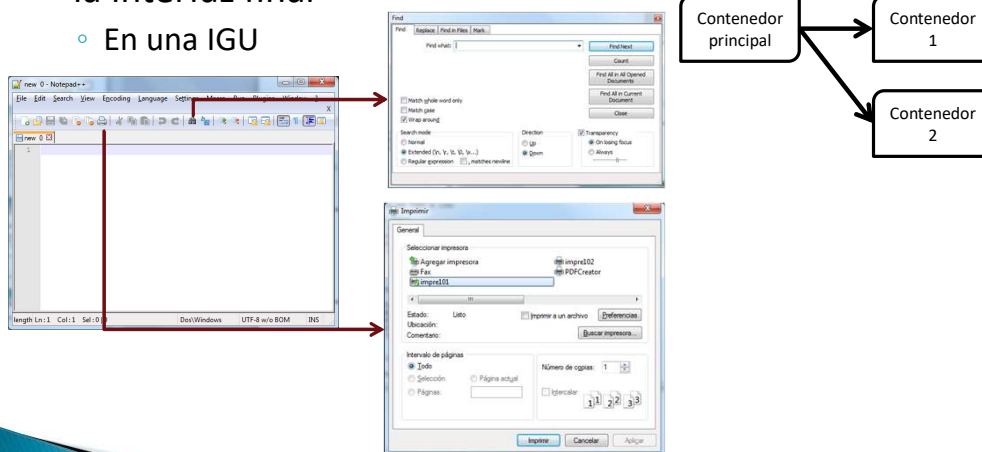
Habitualmente el diagrama de contenidos comienza por el contenedor principal, que representa el punto de inicio de la aplicación, o de la tarea representada.

En este ejemplo podemos ver cual podría ser el diagrama de contenidos asociado a una aplicación web (a una parte de ella).

Así, tendríamos un contenedor principal desde el que se podría acceder a cada una de las posibles entradas de la web. En este caso, cada contenedor se ha traducido finalmente en una vista, pero veremos que no siempre es así.

Diseño conceptual

- Posibles relaciones entre el diagrama de contenidos y la interfaz final
 - En una IGU

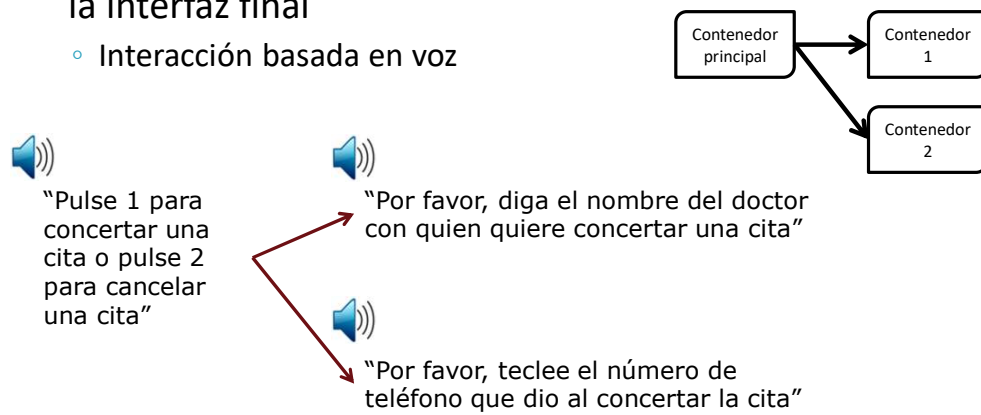


13

Aquí podemos ver el diagrama asociado a una aplicación de escritorio. Hay que tener en cuenta que en el ejemplo solo se muestra una mínima parte de lo que sería el diagrama de la aplicación.

Diseño conceptual

- Posibles relaciones entre el diagrama de contenidos y la interfaz final
 - Interacción basada en voz



14

Aquí se muestra cómo podría ser el diagrama para una interfaz de audio. Donde la locución inicial estaría representada por el contenedor principal y cada uno de los contenedores representa la respuesta del sistema dependiendo de la opción elegida.

Diseño conceptual

- ▶ Posibles relaciones entre el diagrama de contenidos y la interfaz final
 - Cuando se diseñan el contenido y la estructura de la interfaz el diagrama de contenidos debe servir como una guía, no como una receta estricta (por ejemplo, dos contenedores se pueden combinar en una sola pantalla)
- ▶ El diagrama de contenidos se crea a partir de información obtenida durante la recopilación de requisitos y a partir de los casos de uso concretos
 - El diagrama será probablemente incompleto, pero es útil para identificar las áreas funcionales principales y las relaciones entre ellas

15

Vamos a insistir en que el diagrama de contenidos que vamos a obtener durante la fase de diseño conceptual es un prototipo que representa la estructura de la interfaz, por tanto no hay una solución única. Aunque cualquier resultado que obtengamos tiene que ser consistente con los casos de uso descritos en la fase de análisis, el nivel de detalle y la forma de agrupar las funciones y objetos puede variar.

Por otra parte, hay que tener en cuenta, que dado que el diagrama de contenidos es un prototipo, la relación entre los contenedores y las pantallas finales no es biyectiva, es decir, un contenedor no tiene por qué corresponderse con una única pantalla, y en una pantalla pueden mostrarse varios contenedores. Del mismo modo, un contenedor puede dividirse en varias pantallas.

Diseño conceptual: procedimiento

- ▶ Pasos para crear un diagrama de contenidos
 1. Identificar los objetos de tarea primarios, los atributos y las acciones
 2. Identificar los contenedores y los objetos de tarea que van en cada uno de ellos
 3. Conectar los contenedores para mostrar el flujo de la navegación
- ▶ Es un proceso creativo y debe mejorarse iterativamente
 - Utilizar post-its y pósters



<http://think-ux.com/2012/01/09/usability-study-lds-org-true-intent-study/>

16

Los pasos para crear un diagrama de contenidos son los siguientes.

En primer lugar hay que identificar los objetos de tarea que llamamos primarios, sus atributos y las acciones que se pueden realizar sobre estos objetos. Los objetos primarios son los objetos sobre los que trabaja la aplicación, los que consulta, modifica, etc. Por ejemplo, a partir de los casos de uso de las transparencias 8 y 9, podríamos extraer un objeto de tarea que sería Recurso, algunos de sus atributos, como el título y los datos de contacto del dueño... y Búsqueda sería una de las acciones que se pueden realizar sobre el objeto Recurso.

Una vez hemos obtenido todos los objetos con sus atributos y acciones, el siguiente paso será obtener los contenedores. Los contenedores representan una parte de los pasos necesarios para completar una tarea. Más adelante volveremos a hablar sobre los contenedores.

Y como último paso, hay que conectar los contenedores. Estas conexiones van a representar el flujo de navegación entre contenedores que nos va a llevar a completar la tarea.

Como ya hemos dicho, el resultado de este proceso no es único, pero sí que hay algunas soluciones mejores que otras. Por tanto, para obtener una solución óptima, va a ser necesario realizar un proceso iterativo en el que sucesivamente

se vaya refinando la solución.

Diseño conceptual: procedimiento

Identificar objetos de tarea, atributos y acciones

- ▶ Paso 1: Identificar los objetos de tarea primarios, los atributos y las acciones
 - Se usan para decidir qué va en cada contenedor y para establecer los enlaces entre contenedores
 - Similar al diseño orientado a objetos:
 - Objetos de tarea primarios: entidades con información o datos con las que interactúa el usuario para llevar a cabo sus tareas
 - Atributos: las propiedades de los objetos de tarea o enlaces a otros objetos
 - Acciones: funciones que puede realizar el usuario sobre los objetos de tarea

17

El primer paso en la obtención del diagrama es extraer todos los objetos que interviene en las tareas, con sus atributos y acciones.

El concepto de objeto es similar al del diseño orientado objetos. Cualquier objeto debe tener algún atributo o propiedad y alguna acción que el usuario pueda realizar sobre él.

Diseño conceptual: procedimiento

Identificar objetos de tarea, atributos y acciones

- ▶ Objetos de tarea primarios
 - Objetos de alto nivel; suele haber unos pocos
 - Ejemplos: libro, CD-ROM, vídeo, revista, profesor, investigador, estudiante
 - Dónde buscar objetos de tarea: en la documentación de los requisitos y en los casos de uso concretos
 - Consejo: encontrar unidades de información que se buscan o modifican de alguna manera; también artefactos como formularios, documentos, papeles, listas, etc.

18

Los objetos se pueden clasificar en Primarios y secundarios. Los objetos primarios son los objetos sobre los que se actúa directamente en la realización de una tarea. Por ejemplo si la tarea es “buscar un recurso”, Recurso sería un objeto de alto nivel o primario. Por otra parte, los objetos secundarios son objetos que a su vez actúan como atributos de algún objeto primario, es decir . Por ejemplo, un atributo del recurso podría ser el dueño. Pero Dueño a su vez puede ser un objeto, con sus propios atributos y acciones, como por ejemplo su nombre, su datos de contacto, etc.

Para obtener los objetos principalmente nos basaremos en la documentación generada en la fase de análisis, pero en ocasiones también podemos recurrir a nuestra experiencia o conocimiento del dominio de la aplicación.

Diseño conceptual: procedimiento

Identificar objetos de tarea, atributos y acciones

- ▶ **Atributos: son propiedades de un objeto**
 - Un objeto de tarea tiene que tener atributos, si no es un atributo de otro objeto
 - Dos tipos de atributos:
 - Propiedades: datos que pertenecen exclusivamente al objeto
 - Objetos hijo: son objetos de tarea por sí mismos, pero a su vez atributos de otro u otros objetos
 - En una interfaz el objeto hijo aparece cuando se muestra el padre, normalmente dentro de él con todos sus atributos o con una parte de ellos
 - En la interfaz los objetos de tarea y sus atributos se traducen en una combinación de elementos como listas, *combo boxes*, botones de radio, cajas de texto, etc.

19

Respecto a los atributos, representan las propiedades de los objetos. Los describen o definen. Todos los objetos deben tener algún atributo, puesto que un objeto sin atributos no es un objeto.

Los atributos pueden ser propiedades del objeto, como el título de un recurso, o puede ser a su vez un objeto, objeto hijo, como el caso del dueño del recurso.

Posteriormente, durante la fase de diseño físico, estos objetos y sus atributos se asociarán a controles como listas, *combo boxes*, etc.

Diseño conceptual: procedimiento

Identificar objetos de tarea, atributos y acciones

► Acciones

- Los usuarios realizan acciones sobre los objetos de tarea (reservar un vuelo, hacer un pedido, buscar un libro, etc.)
- Además de las acciones específicas de los objetos de tarea, se deben considerar acciones estándar como ver, crear, borrar, copiar, guardar, editar, imprimir, etc.
- En la interfaz, las acciones se representan mediante opciones de menú, botones en la barra de herramientas, un hipervínculo en una página web, etc.

20

Las acciones representan qué se puede hacer con los objetos de tarea. Por ejemplo, un objeto se puede crear, borrar, buscar, imprimir.... Algunas de estas acciones se pueden considerar estándar, pero otras son propias del objeto. Por ejemplo, cualquier objeto se tiene que poder crear y borrar, pero no todos los objetos se imprimen, o se copian.

Muchas de las acciones tienen asociado un caso de uso, pero hay otras acciones de más bajo nivel que se encuentran dentro de otro caso de uso más genérico. Por ejemplo, el caso de uso concreto mostrado en la diapositiva 8 directamente describe la acción buscar del objeto primario recurso. Pero también podríamos considerar como acción seleccionar, indicando que un recurso puede ser seleccionado.

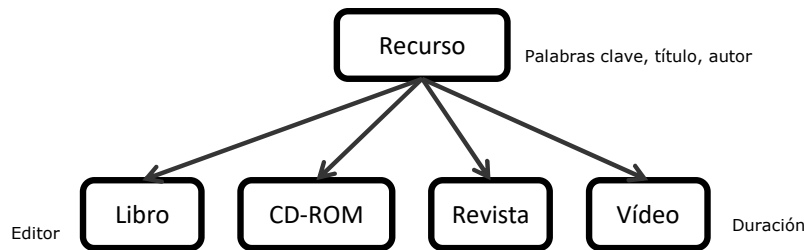
En la fase de diseño físico, estas acciones se asociarán a controles como botones, opciones de menú, etc.

Diseño conceptual: procedimiento

Identificar objetos de tarea, atributos y acciones

► Objetos de tarea primarios

- A menudo los objetos de tarea se pueden agrupar en clases, que abstraen los atributos comunes



21

Vamos con el primer paso para obtener el diagrama de contenidos: la extracción de los objetos de tarea primarios. Pero antes, veamos que los objetos se pueden especializar (o generalizar), de manera podemos tener un objeto padre con los atributos y acciones comunes a los objetos especializados (hijos) y cada uno de los hijos puede tener sus propios atributos y acciones. Es decir, agrupamos los objetos primarios en clases, de forma que todo lo común forma parte de la clase padre, mientras que los atributos específicos forman parte de las clases hijas.

Así, en nuestro ejemplo, podemos agrupar todos los objetos que podemos buscar en la base de datos bajo una clase común o padre llamada "Recurso", que es descrita por los atributos palabras clave, título y autor. Estos atributos son compartidos y describen también a sus clases hijas libro, CD-ROM, etc. Además, cada clase hija tiene además datos propios, como por ejemplo "editor" en el caso del Libro.

Diseño conceptual: procedimiento

Identificar objetos de tarea, atributos y acciones

- ▶ Cómo usar casos de uso concretos para identificar los objetos de tarea y sus atributos
 - Subrayado simple para nombres que pueden corresponder a objetos de tarea
 - Subrayado doble para los atributos de esos objetos de tarea
- ▶ Normalmente los verbos corresponden a acciones, pero no los marcamos porque las relaciones entre ellos y los objetos suelen ser menos directas

22

La documentación y artefactos generados en la fase de análisis puede ser muy extensa, por tanto extraer los objetos, con sus atributos y acciones, puede ser un trabajo muy costoso. Además, obtener una lista completa y sin redundancias se puede volver imposible. Por ello es necesario utilizar algún tipo de metodología.

Aquí mostramos una metodología muy simple, pero a la vez efectiva.

Consiste en recorrer los casos de uso generados e ir marcando los elementos que sean objetos y los que sean atributos. Los objetos los marcaremos subrayándolos con una línea simple, y los atributos con una línea doble. Hay que llevar cuidado de no marcar cada objeto ni cada atributo más de una vez. El objetivo no es detectar cuántas veces aparece un objeto o un atributo, si no simplemente si aparece.

Puede ser útil trabajar con lápices de colores, para poder utilizar un color para subrayar a cada objeto y a sus atributos en un caso de uso concreto, para no mezclar atributos de objetos distintos.

Las acciones (normalmente las detectaremos porque se utilizan verbos) no las marcaremos ya que la relación entra las acciones y los objetos no siempre es directa y en muchos casos se utilizan diferentes verbos para referirse a la misma acción.

Según se vaya extrayendo la información, rellenaremos una tabla por objeto, en la que se muestre esta relación entre el objeto, sus atributos y sus acciones.

Diseño conceptual: procedimiento

Identificar objetos de tarea, atributos y acciones

► Ejemplo: Caso de uso “Buscar un recurso”

Acción del usuario	Respuesta del sistema
El profesor introduce uno o más parámetros de búsqueda para el CD-ROM: título, año y plataforma	El sistema muestra los resultados de la búsqueda
El profesor selecciona un resultado	El sistema muestra los detalles del CD-ROM y los datos de contacto del dueño, que es un estudiante de doctorado
El profesor selecciona la dirección de correo	El sistema muestra un área de mensaje
El profesor escribe y manda la petición por e-mail	El sistema confirma el envío de la petición

23

De este caso de uso ¿qué podríamos extraer

Diseño conceptual: procedimiento

Identificar objetos de tarea, atributos y acciones

► Ejemplo: Caso de uso “Buscar un recurso”

Acción del usuario	Respuesta del sistema
El <u>profesor</u> introduce uno o más parámetros de búsqueda para el CD-ROM: <u>título</u> , <u>año</u> y <u>plataforma</u>	El sistema muestra los resultados de la búsqueda
El profesor selecciona un resultado	El sistema muestra los detalles del CD-ROM y los datos de contacto del dueño, que es un <u>estudiante de doctorado</u>
El profesor selecciona la <u>dirección de correo</u>	El sistema muestra un <u>área de mensaje</u>
El profesor escribe y manda la petición por e-mail	El sistema confirma el envío de la petición

24

Leyendo el caso de uso nos damos cuenta de que “profesor”, “DC-Rom” y “Estudiante” son objetos. ¿Cómo detectamos que son objetos? Porque tienen propiedades. Algunas aparecen explícitamente, en ese caso sería la dirección de correo del profesor, o el título año y plataforma del CD-Rom, aunque para obtener otros atributos tendríamos que analizar otros casos de uso o basarnos en nuestra experiencia.

Diseño conceptual: procedimiento

Identificar objetos de tarea, atributos y acciones

► Otro ejemplo: Caso de uso “Ver actualizaciones”

Acción del usuario	Respuesta del sistema
<i>El estudiante de doctorado solicita las últimas actualizaciones de la biblioteca digital</i>	<i>El sistema muestra la disponibilidad de los últimos libros, CD-ROMs, vídeos y revistas</i>
<i>El estudiante selecciona el año del libro de su autor favorito: J. Nielsen</i>	<i>El sistema muestra todos los detalles del libro y los datos de contacto del dueño, incluido su nombre, que es un profesor</i>
<i>El estudiante selecciona la dirección de correo del profesor</i>	<i>El sistema muestra un área de mensaje</i>
<i>El estudiante escribe y manda una petición del libro por e-mail</i>	<i>El sistema confirma el envío de la petición</i>

25

Del caso de uso “Ver actualizaciones” podríamos obtener otros objetos,

Diseño conceptual: procedimiento

Identificar objetos de tarea, atributos y acciones

► Otro ejemplo: Caso de uso “Ver actualizaciones”

Acción del usuario	Respuesta del sistema
El <u>estudiante de doctorado</u> solicita las últimas actualizaciones de la biblioteca digital	El sistema muestra la disponibilidad de los últimos <u>libros</u> , <u>CD-ROMs</u> , <u>vídeos</u> y <u>revistas</u>
El estudiante selecciona el <u>año</u> del libro de su <u>autor</u> favorito: J. Nielsen	El sistema muestra todos los detalles del libro y los datos de contacto del <u>dueño</u> , incluido su nombre, que es un <u>profesor</u>
El estudiante selecciona la <u>dirección de correo</u> del profesor	El sistema muestra un <u>área de mensaje</u>
El estudiante escribe y manda una petición del libro por e-mail	El sistema confirma el envío de la petición

26

Por ejemplo, nos damos cuenta de que además de CD-Rom hay otros objetos como libro, vídeo o revista. Todos ellos son especializaciones de Recurso.

Por otra parte aparece “dueño”. ¿Qué es dueño , un objeto o un atributo? Analizando los casos de uso de los ejemplos, vemos que tanto los profesores como los estudiantes pueden ser dueños. Por tanto, podríamos ver dueño, como un objeto padre de Profesor y Estudiante, al igual que pasa con Recurso. Sin embargo, en el ejemplo está marcado con el doble subrayado propio de los atributos. Como ya hemos dicho, en ocasiones los objetos hacen el papel de atributos, como en este caso. ¿Podríamos haberlo subrayado con una línea simple? Sí. Tened en cuenta que en ocasiones no nos damos cuenta de que un atributo también es un objeto dentro del propio caso de uso en el que nos encontramos, puede ser que haga falta haber analizado varios casos de uso para reconocerlo como objeto.

Diseño conceptual: procedimiento

Identificar objetos de tarea, atributos y acciones

- ▶ Tras identificar los objetos de tarea y los atributos, se agrupan en una tabla por objeto de tarea

Objeto de tarea	Atributos	Acciones	
CD-ROM	Palabras clave	Reservar	→ Específica
Del análisis del dominio	Autor	Ver	
	Título	Añadir	
Del caso de uso concreto	Año	Imprimir	
	Plataforma	Borrar	→ Estándar
	Propietario (profesor, investigador o alumno de doctorado)	Buscar	
		Editar	

27

Por cada uno de los objetos extraídos vamos a construir una tabla. Una única tabla por objeto, que iremos rellenando a partir del análisis de **todos** los casos de uso.

La tabla que aquí se muestra es la correspondiente al objeto CD-ROM. Fijaos que los atributos son los que aparecen en los dos casos de uso anteriores. Pero podríamos añadir algunos más que extraeríamos del resto de casos de uso de la aplicación y que no hemos visto, y de nuestro conocimiento del dominio. Por ejemplo podríamos añadir una descripción o la duración.

Respecto a las acciones ocurre lo mismo. Además de las que son específicas de la aplicación hay otras se pueden considerar estándar, que suelen realizarse sobre la mayoría de los objetos de tarea, como añadir, borrar, editar... Por tanto, las podemos añadir aunque no aparezcan en los casos de uso, siempre que nos parezcan lógicas.

Diseño conceptual: procedimiento

Identificar objetos de tarea, atributos y acciones

- ▶ Tras identificar los objetos de tarea y los atributos, se agrupan en una tabla por objeto de tarea

Objeto de tarea	Atributos	Acciones
Profesor	Nombre	Ver
	Número de teléfono	Añadir
	Despacho	Editar
	Dirección de correo	Imprimir
		Guardar
		Borrar

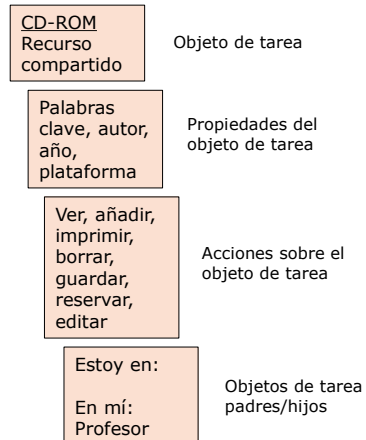
28

Aquí vemos la tabla Objeto-Atributo-Acción correspondiente al objeto Profesor

Diseño conceptual: procedimiento

Identificar objetos de tarea, atributos y acciones

- ▶ Como es difícil capturar todos los objetos de tarea, atributos y acciones de los casos de uso concretos, se debe utilizar una estrategia iterativa
 - Buscar información en los casos concretos de uso, pero también en el conocimiento del usuario del dominio y en nuestro propio análisis del dominio
 - Prototipa tus ideas y evalúalas con los usuarios



29

En general, una única pasada no es suficiente para obtener el conjunto de tablas adecuado. Necesitaremos repetir este proceso para completar lo que se nos puede haber pasado en iteraciones anteriores o incluso detectar redundancias. En ocasiones, y dependiendo de la complejidad de la aplicación, no llegaremos a tener la garantía de que las tablas obtenidas son las mejores, pero el proceso se parará cuando estemos razonablemente seguros de que no nos hemos dejado nada importante.

Si es posible, trate de evaluar el resultado con los usuarios, a fin de asegurarse que se han identificado bien los objetos sobre los que actuará la aplicación y las acciones que desean hacer los usuarios con ellos.

Diseño conceptual: procedimiento

Identifica los contenedores y los objetos de tarea que van dentro de cada uno

- ▶ Paso 2: Identifica los contenedores y los objetos de tarea que van en cada uno de ellos
 - Cada contenedor ayuda al usuario a realizar un cierto trabajo agrupando las funciones y los objetos de tarea necesarios
 - Más adelante se convertirán en pantallas, ventanas o cuadros de diálogo
 - En el paso 3 conectaremos los contenedores con enlaces

30

En el segundo paso vamos a identificar y definir los contenedores. Es frecuente que en un contenedor se maneje un solo objeto, pero no siempre es así. Por otro lado hay algunos contenedores especiales que no utilizan ningún objeto de tarea.

Cada contenedor va a representar uno o varios pasos en la realización de una tarea. Por ejemplo, en una aplicación de escritorio, estos contenedores se convertirán en pantallas, aunque como ya hemos dicho anteriormente, la relación con las pantallas no es 1 a 1, es decir, puede darse el caso de que un contenedor se represente en varias ventanas de la interfaz, o al contrario, que varios contenedores se representen en una única ventana.

Por último, conectaremos los contenedores mediante enlaces completando el diagrama.

Diseño conceptual: procedimiento

Identifica los contenedores y los objetos de tarea que van dentro de cada uno

► Elementos en un contenedor:

- Nombre: el nombre del contenedor
- Propósito: una frase indicando cómo va a soportar la tarea del usuario
- Funciones: ● invocadas por el usuario, ■ invocadas por el sistema
- Enlaces: nombres de contenedores con los que está conectado: ► el nuevo contenedor sustituye al actual, ►► los dos contenedores trabajan en paralelo
- Objetos: los objetos de tarea cuyos atributos y acciones son necesarios para el contenedor
- Restricciones: cualquier restricción para el contenedor, como velocidad, fiabilidad y disponibilidad

31

Para la definición de los contenedores utilizaremos una plantilla que incluirá los siguientes elementos:

El **nombre** del contenedor

El **propósito** u objetivo de ese contenedor.

Las **funciones** que se van a realizar en ese contenedor. Estas funciones pueden ser de dos tipos, las que invoca el usuario (se marcan con un círculo), por ejemplo “seleccionar recurso”, o las que realiza el sistema (marcadas con un cuadrado), por ejemplo “mostrar detalles del recurso”. Es decir, el usuario selecciona un recurso y el sistema como respuesta a esa acción de usuario muestra los detalles del recurso seleccionado. En lo que respecta a las funciones hay que tener en cuenta que no se pueden utilizar acciones físicas de usuario, como pulsar un botón, marcar un check box, etc. Recordemos que estamos en el diseño Conceptual y todavía no sabemos qué pantallas tendremos, ni qué estilos de interacción tendremos, ni por tanto los controles que utilizaremos. Esas decisiones no se toman ahora.

Los **enlaces** que representan la navegación entre contenedores. El enlace que utiliza una flecha simple indica que el contenedor al que se accede va a sustituir al contenedor actual. El enlace representado por una flecha doble indica que los dos contenedores pueden estar activos simultáneamente (es decir, no hace falta “cerrar” un contenedor para pasar al siguiente). El nombre de los enlaces coincidirá con el nombre de los contenedores a los que nos dirigen. De esta forma, si tenemos dos enlaces con el mismo nombre, indicará que nos llevan al

mismo contenedor.

Los **Objetos** que se manejan en el contenedor, ya sean primarios o secundarios. Es necesario indicar todos los objetos necesarios para realizar las funciones indicadas.

Y, por último, las **restricciones**, en caso de haberlas. Normalmente las restricciones se toman de los requisitos no funcionales que hayamos detectado. También puede haber restricciones de acceso, por ejemplo, que el usuario esté logueado.

Diseño conceptual: procedimiento

Identifica los contenedores y los objetos de tarea que van dentro de cada uno

► Plantilla para contenedores

Nombre
Propósito
Funciones
● {realizadas por el usuario}
■ {realizadas por el sistema}
Enlaces
► {enlace sencillo}
►► {enlace doble}
Objetos
Restricciones

32

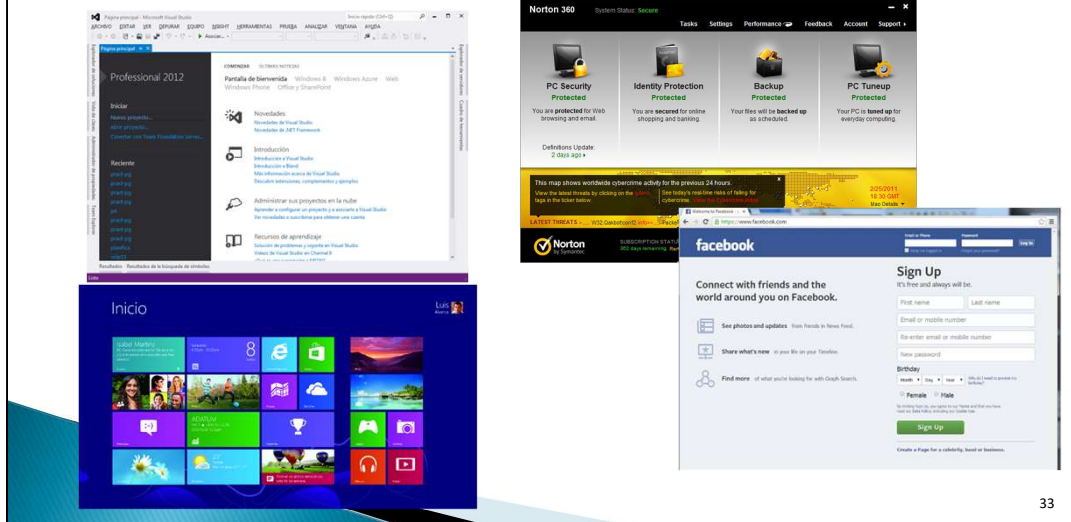
Aquí vemos la plantilla a utilizar para definir los contenedores. Se debe seguir el orden indicado. Además, recuerda marcar con un pequeño círculo las acciones del usuario, y con un pequeño cuadrado las del sistema. Es mejor añadirlas en el orden secuencial en el que se producen.

En cuanto a los enlaces, recuerda, **un** símbolo de flecha si el nuevo contenedor reemplaza al actual, solo está visible **uno**. **Dos**, si tenemos los **dos** contenedores al mismo tiempo.

Diseño conceptual: procedimiento

Identifica los contenedores y los objetos de tarea que van dentro de cada uno

- El contenedor principal representa la primera pantalla que se encuentra el usuario



Los diagramas de contenido van a tener un contenedor principal que representa el punto de entrada a nuestro sistema o tarea. Así, en una aplicación de escritorio, web, etc., se corresponde con la primera pantalla que se encontrará el usuario al realizar la tarea (si el diagrama de contenidos solo se ha realizado para una tarea), o al iniciar la aplicación (si es un diagrama completo).

Diseño conceptual: procedimiento

Identifica los contenedores y los objetos de tarea que van dentro de cada uno

- ▶ El contenedor principal tendrá enlaces a:
 - Tareas vitales: el usuario tendrá que realizar estas tareas rápidamente, aun en condiciones de estrés
 - Tareas frecuentes: tareas en las que el usuario emplea la mayor parte del tiempo: debe ser rápido acceder a ellas
 - Ayudas de navegación: facilitan al usuario encontrar aquello que es capaz de hacer la aplicación
- ▶ El contenedor principal no realiza ninguna de esas acciones, sólo proporciona enlaces a los contenedores que las realizarán

34

El contenedor principal puede ser un contenedor tipo “lanzadera” donde únicamente se puede seleccionar la acción a realizar por el usuario, en este caso el contenedor no incluirá funciones de sistema, solo de usuario, y no tendrá objetos, o puede ser un contenedor que incluya las tareas que nos interesa encontrar en primer lugar al arrancar la aplicación, como las que son más importantes, las que se realizan con más frecuencia o la ayuda.

Diseño conceptual: procedimiento

Identifica los contenedores y los objetos de tarea que van dentro de cada uno

► Ejemplo del contenedor principal de la biblioteca

Principal

Soporta las tareas más frecuentes

Funciones

- Buscar recursos
- Ver actualizaciones recientes
- Contactar con soporte técnico

Enlaces

- Introducir detalles de búsqueda
- Visualizar actualizaciones
- Contactar

Objetos

Restricciones

35

Aquí vemos el que sería el contenedor principal correspondiente a la aplicación de la biblioteca de recursos digitales.

Como vemos es un contenedor de tipo “lanzadera”. Solo hay funciones de usuario, y cada una de estas funciones, al ser seleccionada, activaría uno de los enlaces.

Cada uno de los enlaces se corresponderá con un contenedor.

Además, observar que ningún objeto está asociado a este contenedor. Recordar añadir las secciones objetos y restricciones aunque no tengan valores.

Diseño conceptual: procedimiento

Identifica los contenedores y los objetos de tarea que van dentro de cada uno

► Otros contenedores

- Derivan de los casos de uso concretos (cada caso de uso muestra la secuencia de pasos necesarios para realizar una tarea particular)
- La funcionalidad de la tarea puede dividirse entre uno o más contenedores

36

El resto de contenedores, lo que no son el contenedor principal, los vamos a extraer principalmente de los casos de uso, pero también podemos apoyarnos en la descomposición jerárquica de las tareas realizada en la fase de análisis.

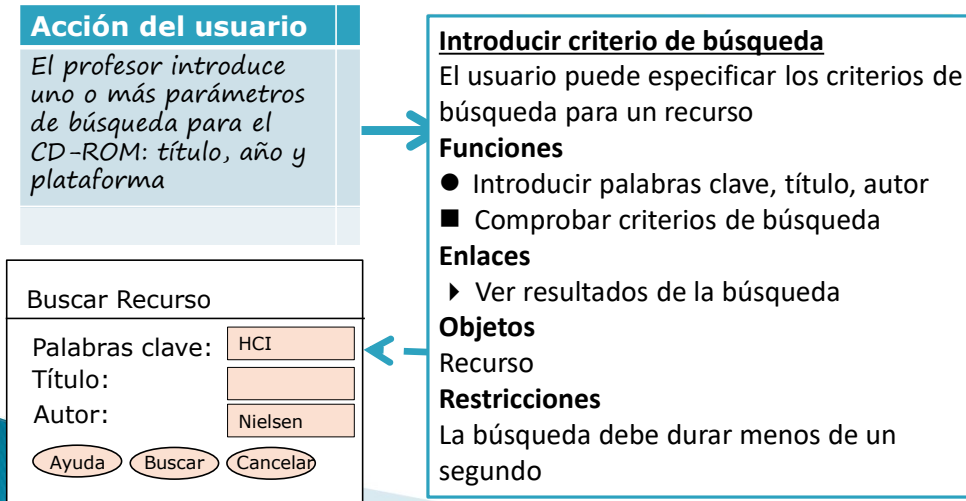
En las siguientes transparencias vamos a ver los contenedores correspondientes al caso de uso “Buscar y solicitar recurso” de la aplicación de Biblioteca.

Diseño conceptual: procedimiento

Identifica los contenedores y los objetos de tarea que van dentro de cada uno

► Ejemplo:

- Si utilizamos una sola página para buscar todos los recursos



Partiendo del caso de uso, y analizando el primer paso realizado por el usuario, podríamos obtener el contenedor que se muestra. Hay que tener en cuenta que una celda del caso de uso se puede corresponder con un contenedor, ¡pero no siempre es así!

Veamos cuál es la interpretación de este contenedor:

El usuario proporciona al sistema los parámetros necesarios para la búsqueda de los recursos. El sistema comprueba si los parámetros son correctos, realiza la búsqueda y nos redirecciona al contenedor “Ver resultados de la búsqueda” donde se mostrarán los resultados.

El objeto a utilizar es el Recurso (los atributos y operaciones a realizar son de este objeto), y en este caso, hemos añadido una restricción en la que decimos que el sistema tiene que ser capaz de realizar la búsqueda en menos de 1 segundo.

¿Hay una única forma de diseñar este contenedor? No. Por ejemplo, podríamos haber puesto una función inicial de sistema “Solicitar parámetros de búsqueda”. Hubiera sido igualmente correcto.

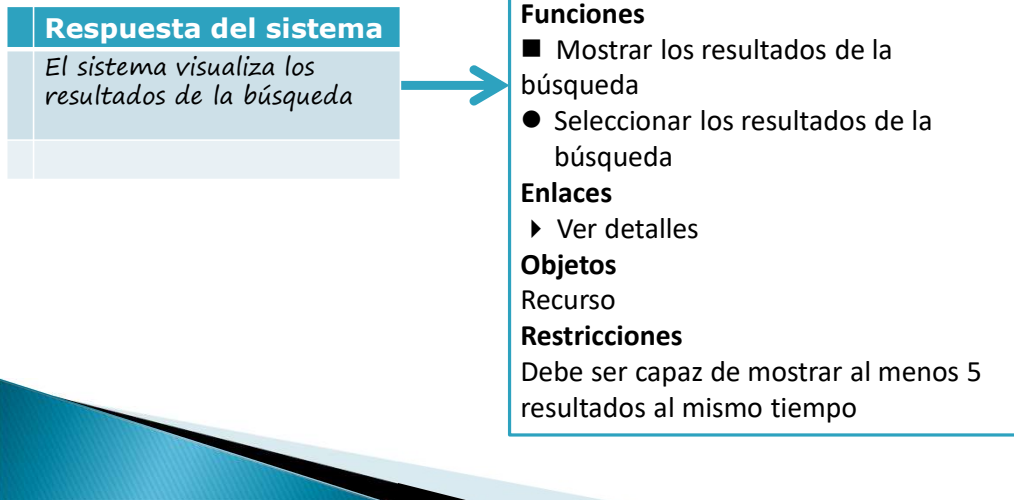
Una vez tenemos el contenedor, podríamos obtener un prototipo para la pantalla correspondiente a este contenedor, como se muestra en la imagen, aunque no

olvidemos que la traducción entre contenedor y pantalla no siempre es inmediata.

Diseño conceptual: procedimiento

Identifica los contenedores y los objetos de tarea que van dentro de cada uno

► Otro ejemplo



Del contenedor anterior pasaríamos a este contenedor, donde se mostrarán los resultados de la búsqueda realizada.

La interpretación sería la siguiente:

El sistema muestra los resultados.

El usuario selecciona el recurso que le interesa y pasamos al contenedor "Mostrar detalles" donde se nos mostraran los detalles del recurso seleccionado.

Seguimos trabajando con el objeto Recurso y ponemos la restricción de que al menos nos muestre 5 resultados simultáneamente.

Este proceso lo repetiríamos hasta que tengamos los contenedores que consideremos adecuados para nuestra tarea.

Diseño conceptual: procedimiento

Conectar los contenedores para mostrar el flujo de navegación

- ▶ Paso 3: Conectar los contenedores para mostrar el flujo de la navegación
 - Los enlaces reflejan el orden de las acciones que sigue el usuario para realizar una tarea determinada
 - Normalmente la identificación y los enlaces de los contenedores se hacen en paralelo
 - Recuerda, los enlaces pueden ser:
 - > con una flecha (el nuevo contenedor sustituye al actual)
 - >> con dos flechas (los dos contenedores trabajan en paralelo)
 - Los enlaces se pueden etiquetar mediante condiciones de interacción, que indican condiciones en que el flujo puede atravesar el enlace

39

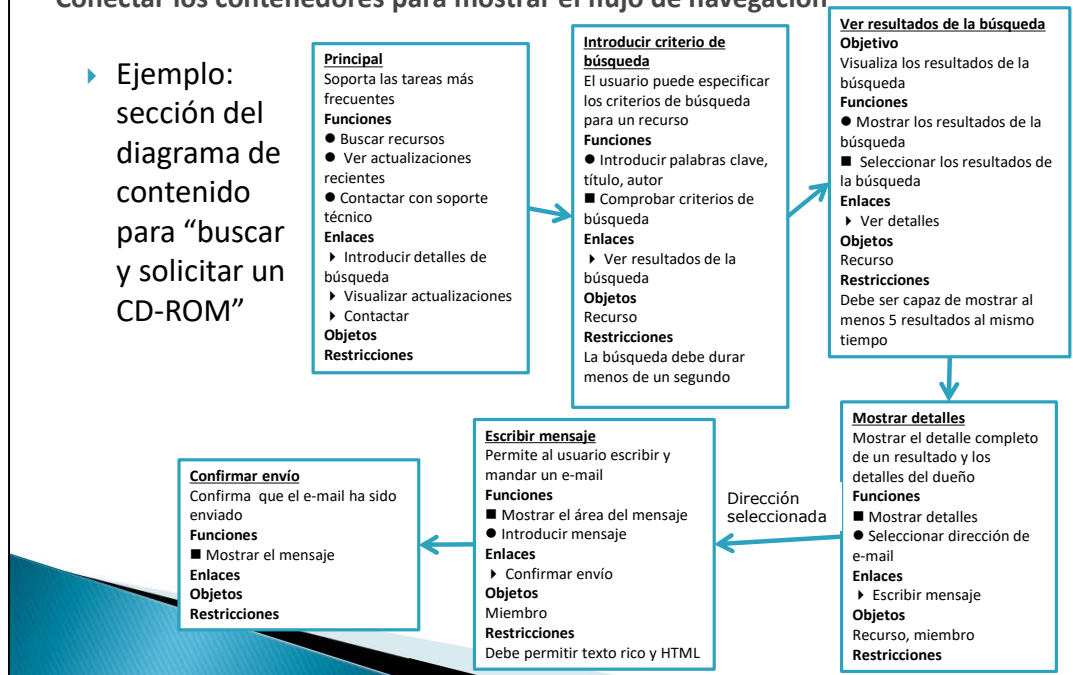
El tercer paso es la conexión de los contenedores. Si los contenedores están bien diseñados este paso será rápido. Si no están bien diseñados nos servirá para detectar, corregir errores y mejorar el diagrama.

Los enlaces reflejan el orden en el que el usuario interacciona con el sistema cuando realiza sus tareas. Posteriormente, en la fase de diseño físico, los enlaces pasarán a ser botones, opciones de menú o hipervínculos, entre otros tipos de objeto.

Diseño conceptual: procedimiento

Conectar los contenedores para mostrar el flujo de navegación

- ▶ Ejemplo: sección del diagrama de contenido para “buscar y solicitar un CD-ROM”

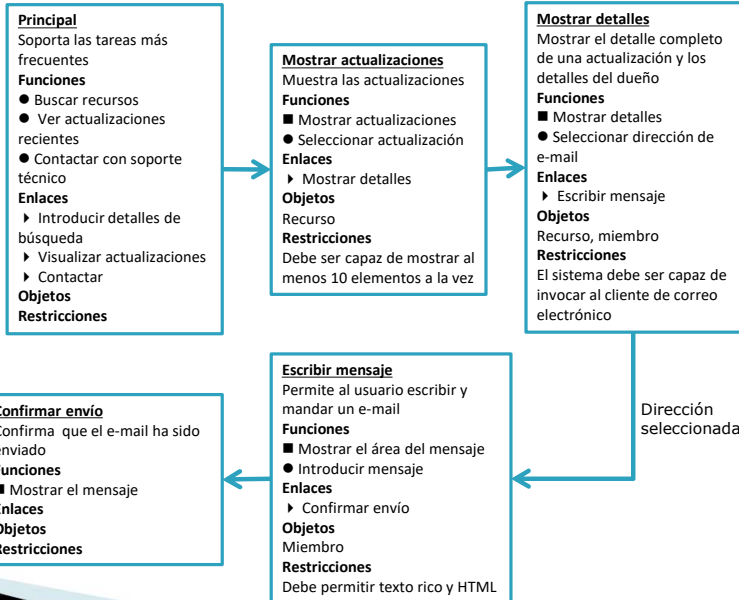


Aquí vemos el diagrama completo para la tarea “Buscar y solicitar un recurso”

Diseño conceptual: procedimiento

Conectar los contenedores para mostrar el flujo de navegación

- ▶ Ejemplo: sección del diagrama de contenido para “Ver actualizaciones y solicitar un libro”



41

Y aquí vemos el diagrama para el caso de uso “Ver actualizaciones y solicitar un recurso”. Como veis los contenedores “Mostrar detalles”, “Escribir mensaje” y “Confirmar envío”, son los mismos en los dos diagramas. Lo que quiere decir que podremos llegar a esos contenedores desde dos lugares distintos.

Diseño conceptual: procedimiento

Conectar los contenedores para mostrar el flujo de navegación

- ▶ Crear un diagrama de contenidos para una aplicación compleja es difícil, por lo tanto se deben hacer varias iteraciones sobre los prototipos correspondientes
- ▶ Para evaluar el diagrama de contenidos, se deben hacer varias pasadas por los casos de uso concretos
 - De este modo se garantiza que los contenedores soportan la funcionalidad necesaria y que los enlaces permiten al usuario acceder a ellos

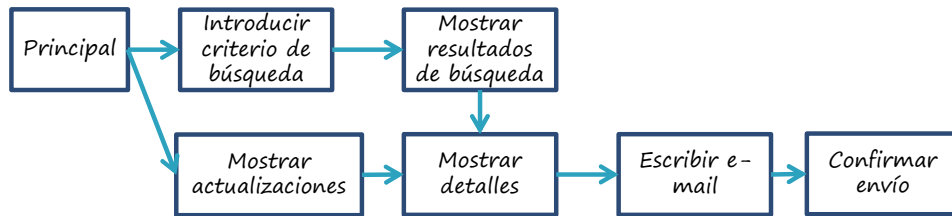
42

Como conclusión podemos decir que la fase de Diseño Conceptual no es una fase sencilla, pero es fundamental para poder hacer el diseño físico de la interfaz. Podemos verla como una fase de “aproximación” a las ventanas finales.

Como veis la solución no siempre es única, ni siquiera vamos a poder asegurar siempre que haya una solución que objetivamente sea mejor que las demás y la hayamos alcanzado. En cualquier caso recordad que este es un proceso iterativo, en el que en cada iteración tendremos que evaluar el diagrama obtenido para comprobar si los contenedores soportan la funcionalidad necesaria, y si se van a poder completar las tareas de forma satisfactoria.

Diseño conceptual: Resultado

► Diagrama de contenidos



- La traducción de contenedores a elementos del interfaz no es directa. Un diagrama de contenidos puede ocupar varias pantallas o varios contenedores pueden combinarse en una sola pantalla.
 - Todavía quedan muchas decisiones por tomar.

Aquí vemos el diagrama de contenidos para las tareas “Buscar y solicitar recurso” y “Ver actualizaciones”. Los contenedores aparecen "comprimidos" de forma que solo aparece su nombre. Como veis, los contenedores que son el mismo no se repiten en el diagrama.

El tipo de diagramas que obtenemos puede ser comprendido por muchos tipos de personas, por lo que pueden ser utilizados para evaluar nuestra propuesta de flujo de interacción con el usuario, y saber si hemos comprendido bien los requisitos. De hecho, al final del proceso de diseño conceptual, el diagrama de contenidos obtenido es un prototipo de baja fidelidad de nuestro sistema.

Con esto acabamos la explicación del tema.

Referencias

- ▶ D. Stone, C. Jarrett, M. Woodroffe. User Interface Design and Evaluation. Morgan Kaufmann, 2005.
 - Capítulo 3