



TECNOLÓGICO
NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO DE ACAPULCO
INGENIERIA EN SISTEMAS COMPUTACIONALES

INGENIERIA DE SOFTWARE

PROFESOR: MARIA NANCY GARCIA CASTRO

TAREA 3

2.3 DISEÑO DE DATOS

TAREA 4

2.4 DISEÑO DE INTERFAZ DE USUARIO.

ALUMNO: AGUILAR BARROSO JOSE MANUEL

NO.CONTROL: 18320791

HORARIO: 02:00pm a 03:00pm

CICLO ESCOLAR FEBRERO - JUNIO 2021

INDICE

INTRODUCCION	3
¿QUÉ ES EL DISEÑO DE DATOS?.....	5
PARA LLEVAR A CABO EL DISEÑO DE DATOS QUE SE DEBE TOMAR EN CUENTA EXPLIQUE Y DE UN EJEMPLO.....	6
DISEÑO DE LA INTERFAZ DE USUARIO.....	9
LAS REGLAS DORADAS.	9
MODELOS DE DISEÑO DE LA INTERFAZ	12
EL PROCESO DE DISEÑO DE LA INTERFAZ.....	13
ANALISIS Y MODELO DE DISEÑO.....	13
ACTIVIDADES DE DISEÑO DE INTERFAZ.....	14
PROBLEMAS DEL DISEÑO.....	14

INTRODUCCION

El diseño es la etapa, en la que se fomenta la calidad. Sin diseño se corre riesgo de construir un sistema inestable, y difícil de probar. El diseño deberá implementar todos los requisitos explícitos del modelo de análisis y deberá ajustarse a los requeridos por el cliente. El diseño tiene que ser una guía comprensible para los que implemente en software y consecuentemente, los que den soporte al mismo. Es tanto un proceso como un modelo. Es un proceso porque es una secuencia de pasos que el diseñador describe con el fin de construir el diseño del software. Es un modelo porque comienza con la totalidad, y luego refina. Nos enfocaremos en el diseño de datos, el activo más importante de una organización es la información por lo tanto en esta lección aprenderemos como realizar un diseño adecuado para que los datos no sean solo datos, sino que se conviertan en información que pueda ser consultado por todos y usada de forma adecuada.

Comenzaremos definiendo el diseño de interfaz de usuario, así como analizaremos los principios de usabilidad, describiendo la importancia de los principios para el desarrollo de un sistema de aplicación, ya que es de suma importancia en su buen funcionamiento para el desempeño de programas utilizados en las empresas El diseño de la interfaz de usuario crea un medio eficaz de comunicación entre los seres humanos y la computadora. Siguiendo un conjunto de principios de diseño de la interfaz, el diseño identifica los objetos y acciones de ésta y luego crea una plantilla de pantalla que constituye la base del prototipo de la interfaz de usuario.



TECNOLÓGICO
NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO DE ACAPULCO

INGENIERIA EN SISTEMAS COMPUTACIONALES

INGENIERIA DE SOFTWARE

PROFESOR: MARIA NANCY GARCIA CASTRO

TAREA 3

2.3 DISEÑO DE DATOS

ALUMNO: AGUILAR BARROSO JOSE MANUEL

NO.CONTROL: 18320791

HORARIO: 02:00pm a 03:00pm

CICLO ESCOLAR FEBRERO - JUNIO 2021

¿QUÉ ES EL DISEÑO DE DATOS?

Igual que otras actividades de la ingeniería de software, el diseño de datos (en ocasiones denominado arquitectura de datos) crea un modelo de datos o información que se representa en un nivel de abstracción elevado (el punto de vista del usuario de los datos). Este modelo de los datos se refina después en forma progresiva hacia representaciones más específicas de la implementación que puedan ser procesadas por el sistema basado en computadora. En muchas aplicaciones de software, la arquitectura de los datos tendrá una influencia profunda en la arquitectura del software que debe procesarlo. La estructura de los datos siempre ha sido parte importante del diseño de software. En el nivel de componentes del programa, del diseño de las estructuras de datos y de los algoritmos requeridos para manipularlos, es esencial la creación de aplicaciones de alta calidad. En el nivel de la aplicación, la traducción de un modelo de datos (obtenido como parte de la ingeniería de los requerimientos) a una base de datos es crucial para lograr los objetivos de negocios de un sistema. En el nivel de negocios, el conjunto de información almacenada en bases de datos incompatibles y reorganizados en un “data Waterhouse” permite la minería de datos o descubrimiento de conocimiento que tiene un efecto en el éxito del negocio en sí. En cada caso, el diseño de los datos juega un papel importante.

PARA LLEVAR A CABO EL DISEÑO DE DATOS QUE SE DEBE TOMAR EN CUENTA EXPLIQUE Y DE UN EJEMPLO.

La actividad de diseño de datos traduce esos elementos del modelo de requisitos en estructuras #datos a nivel de los componentes del software y, cuando es necesario, a arquitectura de base de datos a nivel de aplicación

El diseño de datos a nivel de componentes se centra en la representación de estructuras de datos a las que se accede directamente a través de uno o más componentes del software. Wasserman [WASSO] ha propuesto un conjunto de principios que pueden emplearse para especificar y diseñar dicha estructura de datos. En realidad, el diseño de datos empieza durante la creación del modelo de análisis. Recordando que el análisis de requisitos y el diseño a menudo se solapan, consideramos el siguiente conjunto de principios [WASSO] para la especificación de datos:

- Los principios del análisis sistemático aplicado a la función y al comportamiento deberían aplicarse también a los datos. Invertimos mucho tiempo y esfuerzo en obtener, revisar y especificar los requisitos funcionales y el diseño preliminar. Las representaciones de flujo de datos y contenido deberían desarrollarse
- Todas las estructuras de datos y las operaciones a llevar a cabo en cada una de ellas deberían estar claramente identificadas. El diseño de una estructura de datos eficaz debe tener en cuenta las operaciones que se van a llevar a cabo sobre dicha estructura de datos (vea [AH083]). Por ejemplo, imagínese una estructura de datos hecha con un conjunto de elementos de datos diversos. La estructura de datos se va a manipular con varias funciones principales del software. Después de la evaluación de la operación realizada sobre la estructura de datos, se define un tipo de datos abstracto para usarlo en el diseño del software subsiguiente.
- Se debería establecer un diccionario de datos y usarlo para definir el diseño de los datos y del programa. Un diccionario de datos representa

explícitamente las relaciones entre los objetos de datos y las restricciones de los elementos de una estructura de datos. Los algoritmos que deben aprovecharse de estas relaciones específicas pueden definirse más fácilmente si existe una especificación de datos tipo diccionario.

- Las decisiones de diseño de datos de bajo nivel deberían dejarse para el final del proceso de diseño. Se puede usar un proceso de refinamiento paso a paso para el diseño de datos. Es decir, la organización general de datos puede definirse durante el análisis de requisitos; refinarse durante los trabajos de diseño de datos y especificarse en detalle durante el diseño a nivel de componentes.
- La representación de las estructuras de datos deberían conocerla sólo aquellos módulos que deban hacer uso directo de los datos contenidos dentro de la estructura
- Debería desarrollarse una biblioteca de estructuras de datos útiles y de las operaciones que se les pueden aplicar. Las estructuras de datos y las operaciones deberían considerarse como recursos para el diseño del software. Las estructuras de datos pueden diseñarse para que se puedan reutilizar.
- Un diseño del software y un lenguaje de programación debería soportar la especificación y realización de los tipos abstractos de datos. La implementación de una estructura de datos sofisticada puede hacerse excesivamente difícil si no existen los medios de especificación directa de la estructura en el lenguaje de programación escogido para dicha implementación.

BIBLIOGRAFIAS TAREA 3

- Ingeniería de software un enfoque práctico, 5ta. Edición
Roger pressman. Capítulo 13: conceptos y principios de
diseño, 219
- Ingeniería de software, 7ma. Edición Ian
Sommerville. Parte III, Diseño, 217.



TECNOLÓGICO
NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO DE ACAPULCO

INGENIERIA EN SISTEMAS COMPUTACIONALES

INGENIERIA DE SOFTWARE

PROFESOR: MARIA NANCY GARCIA CASTRO

TAREA 4

2.4 DISEÑO DE INTERFAZ DE USUARIO.

ALUMNO: AGUILAR BARROSO JOSE MANUEL

NO.CONTROL: 18320791

HORARIO: 02:00pm a 03:00pm

CICLO ESCOLAR FEBRERO - JUNIO 2021

DISEÑO DE LA INTERFAZ DE USUARIO.

El diseño de la interfaz de usuario crea un medio eficaz de comunicación entre los seres humanos y la computadora. Siguiendo un conjunto de principios de diseño de la interfaz, el diseño identifica los objetos y acciones de ésta y luego crea una pantalla que constituye la base del prototipo de la interfaz de usuario.

LAS REGLAS DORADAS.

Dejar el control al usuario

La mayor parte de las restricciones y limitaciones impuestas por el diseñador se han pensado para simplificar el modo de interacción. Pero, ¿para quienes? En muchos casos es posible que el diseñador introduzca limitaciones y restricciones para simplificar la implementación de la interfaz. Y el resultado puede ser una interfaz fácil de construir, pero frustrante de utilizar.

Se definirán una serie de principios de diseño que permiten que el usuario tenga el control:

- Definir los modos de interacción de manera que no obligue a que el usuario realice acciones innecesarias o no deseadas.
- Tener en consideración una interacción flexible.
- Permitir que la interacción del usuario sea interrumpible y también reversible.
- Facilitar la interacción a medida que aumenta la habilidad y permitir que aquella se personalice.
- Ocultar los tecnicismos internos al usuario ocasional.
- Diseñar la interacción directa con objetos que aparezcan en la pantalla.

Reducir la necesidad de que el usuario memorice

Cuanto más tenga que recordar un usuario, más propensa a errores será su interacción con el sistema. Esta es la razón por la que una interfaz de usuario bien diseñada no pondrá a prueba la memoria del usuario. Siempre que sea posible, el sistema deberá «recordar» la información pertinente y ayudar a que el usuario recuerde mediante un escenario de interacción, se definen los principios de diseño que hacen posible que una interfaz reduzca la carga de memoria del usuario:

- Reducir la demanda de memoria a corto plazo.

Cuando los usuarios se ven involucrados en tareas complejas, exigir una memoria a corto plazo puede ser significativo. La interfaz se deberá diseñar para reducir los requisitos y recordar acciones y resultados anteriores.

- Hacer que lo preestablecido sea significativo.

El conjunto inicial de valores por defecto tendrá que ser de utilidad para al usuario, pero un usuario también deberá tener la capacidad de especificar sus propias preferencias.

Definir atajos que sean intuitivos. Cuando para diseñar un sistema se utiliza la mnemónica (por ejemplo, alt-P para invocar la función de imprimir), ésta deberá ir unida a una acción que sea fácil de recordar (por ejemplo, la primera letra de la tarea que se invoca).

La distribución visual de la interfaz debe basarse en una metáfora del mundo real. Por ejemplo, en un sistema de pago de facturas se deberá utilizar la metáfora de la chequera y el registro del cheque para conducir al usuario por el proceso del pago de facturas.

- Revelar información de manera progresiva.

La interfaz debe estar organizada de manera jerárquica. Es decir, la información acerca de una tarea, objeto o comportamiento debe presentarse primero en un nivel de generalización elevado. Después de que con el ratón el usuario manifieste

interés, deben darse más detalles. Por ejemplo, para muchas aplicaciones de procesamiento de textos, se tiene la función de subrayar. La función en sí es una de varias en el menú estilo del texto. No obstante, no se enlista cada una de las herramientas para subrayar. El usuario debe hacer clic en la opción de subrayar; después se presentan todas las opciones para esta función (una raya, doble raya, línea punteada, etc.).

Hacer consistente la interfaz.

La interfaz debe presentar y obtener información en forma consistente. Esto implica: 1) que toda la información se organice de acuerdo con reglas de diseño que se respeten en todas las pantallas desplegadas, 2) que los mecanismos de entrada se limiten a un conjunto pequeño usado en forma consistente en toda la aplicación, y 3) que los mecanismos para pasar de una tarea a otra se definan e implementen de modo consistente.

Estas reglas de oro forman en realidad la base para los principios del diseño de la interfaz de usuario que servirán de guía para esta actividad de diseño de software tan importante.

MODELOS DE DISEÑO DE LA INTERFAZ

Cuando se va a diseñar la interfaz de usuario entran en juego cuatro modelos diferentes. El ingeniero del software crea un modelo de diseño; cualquier otro ingeniero (o el mismo ingeniero del software) establece un modelo de usuario, el usuario final desarrolla una imagen mental que se suele llamar modelo de usuario o percepción del usuario, y los que implementan el sistema crean una imagen de sistema [RUBSS].

Un modelo de diseño de un sistema completo incorpora las representaciones del software en función de los datos, arquitectura, interfaz y procedimiento. La especificación de los requisitos puede que establezca ciertas limitaciones que ayudarán a definir al usuario del sistema, pero el diseño de la interfaz suele ser un único tema secundario de modelo de interfaz'.

EL PROCESO DE DISEÑO DE LA INTERFAZ

El proceso de diseño de las interfaces de usuario es iterativo y se puede representar mediante un modelo espiral similar. En la Figura 15.1 se puede observar que el proceso de diseño de la interfaz de usuario acompaña cuatro actividades distintas de marco de trabajo [MAN97]:

- Análisis y modelado de usuarios, tareas y entornos.
- Diseño de la interfaz
- Implementación de la interfaz
- Validación de la interfaz

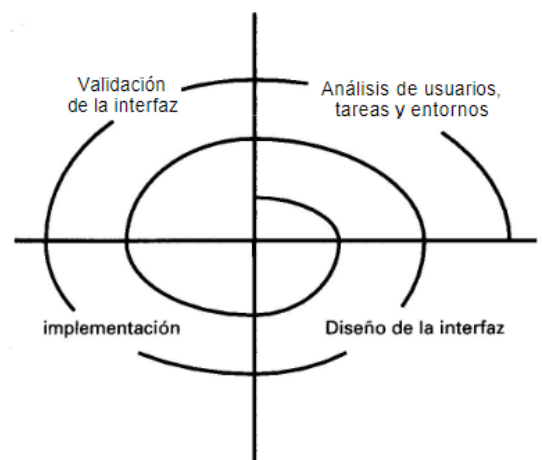


FIGURA 15.1. El proceso de diseño de la interfaz de usuario.

El objetivo del diseño de la interfaz es definir un conjunto de objetos y acciones de interfaz (y sus representaciones en la pantalla) que posibiliten al usuario llevar a cabo todas las tareas definidas de forma que cumplan todos los objetivos de usabilidad definidos por el sistema.

ANÁLISIS Y MODELO DE DISEÑO

El análisis de tareas para el diseño de la interfaz o bien utiliza un enfoque deliberativo o bien orientado a objetos, pero aplicando este enfoque a las actividades humanas.

El análisis de tareas se puede aplicar de dos maneras. Como ya hemos destacado anteriormente, un sistema interactivo basado en computadora se suele utilizar para reemplazar una actividad manual o semi-manual. Para comprender las tareas que se han de llevar a cabo para lograr el objetivo de la actividad, un ingeniero4 deberá entender las tareas que realizan los hombres actualmente (cuando se

utiliza un enfoque manual) y hacer corresponder estas tareas con un conjunto de tareas similar (aun- que no necesariamente idénticas) que se implementan en el contexto de la interfaz de usuario

ACTIVIDADES DE DISEÑO DE INTERFAZ

Una vez finalizado el análisis de tareas, quedan definidas detalladamente todas (objetos y acciones) las que requiere el usuario final y comienza la actividad del diseño de la interfaz. Mediante el enfoque que se muestra a continuación se podrán llevar a cabo los primeros pasos del diseño de la interfaz [NOR86]:

- ❖ Establecer los objetivos e intenciones para cada tarea.
- ❖ Hacer corresponder cada objetivo/intención con una secuencia de acciones específicas
- ❖ Especificar la secuencia de acciones de tareas y sub- tareas, también llamado escenario del usuario, de la manera en que se ejecutarán a nivel de la interfaz.
- ❖ Indicar el estado del sistema, esto es, el aspecto que tiene la interfaz cuando se está llevando a cabo el escenario del usuario.
- ❖ Definir el mecanismo de control, esto es, los objetos y acciones disponibles para que el usuario altere el estado del sistema.
- ❖ Mostrar la forma en que los mecanismos de control afectan al estado del sistema.
- ❖ Indicar la forma en que el usuario interpreta el estado del sistema a partir de la información proporcionada gracias a la interfaz.

PROBLEMAS DEL DISEÑO

A medida que la interfaz de usuario va evolucionando casi siempre afloran cuatro temas comunes de diseño: el tiempo de respuesta del sistema, los servicios de ayuda al usuario, la manipulación de información de errores y el etiquetado de órdenes. Desgraciadamente, muchos diseñadores no abordan estos temas dentro del proceso de diseño hasta que es relativamente tarde (algunas veces no se siente la aparición de un error hasta que se dispone del prototipo operativo). El

resultado suele ser una iteración innecesaria, demoras de proyecto y frustración del usuario. Es infinitamente mejor establecer el tema de diseño que se vaya a tener en cuenta al iniciar el diseño del software, es decir cuando los cambios son fáciles y los costes más reducidos.

Cuando ha salido algo mal, los mensajes de error y las sugerencias son «malas noticias» para los usuarios de sistemas interactivos. En el peor de los casos, estos mensajes imparten información sin utilizar o engañosa y sirven solo para incrementar la frustración del usuario. Existen muy pocos usuarios que puedan decir que no se han encontrado con un error del tipo:

FALLO GRAVE DEL SISTEMA - - 14A

En algún lugar debe existir una explicación del error 14A, o sino ¿por qué habrá incluido el diseñador esta identificación? A pesar de esto, el mensaje de error no proporciona una indicación verdadera de lo que va mal o de dónde mirar para obtener más información. Un mensaje de error como el que se ha presentado anteriormente no hace nada por aliviar la ansiedad del usuario o por ayudar a solucionar el problema. En general, todos los mensajes de error o sugerencias de un sistema interactivo deberán tener las características siguientes:

- El mensaje deberá describir el problema en una jerga que el usuario pueda entender.
- El mensaje deberá proporcionar consejos constructivos para recuperarse de un error.
- El mensaje deberá indicar cualquier consecuencia negativa del error (por ejemplo, los archivos de datos posiblemente deteriorados) para que el usuario pueda comprobar y garantizar que no hay ninguno (y corregirlos si existen).

BIBLIOGRAFIAS TAREA 4

- Ingeniería de software un enfoque práctico, 5ta. Edición **Roger pressman**.Capítulo 13: conceptos y principios de diseño, 219
- Ingeniería de software, 7ma. Edición **Ian Sommerville**.Parte III, Diseño, 217.
- https://virtual.itca.edu.sv/Mediadores/stis/35___diseo_de_la_interfaz_de_usuario.html

BIBLIOGRAFIA GENERAL

- Ingeniería de software un enfoque práctico, 5ta. Edición **Roger pressman**.Capítulo 13: conceptos y principios de diseño, 219
- Ingeniería de software, 7ma. Edición **Ian Sommerville**.Parte III, Diseño, 217.
- https://virtual.itca.edu.sv/Mediadores/stis/35___diseo_de_la_interfaz_de_usuario.html