

Práctica 1.

Prototipado de interfaces de usuario

Índice

Introducción	1
¿Qué es un prototipo?	2
Prototipos de interfaz de usuario esenciales	2
Prototipo de interfaces de usuario tradicional	4
Diagramas de Flujo de Interfaces de Usuario	7
Usabilidad	8
Estrategias de diseño de interfaces de usuario	9
Desarrollo de la práctica	11
Bibliografía	13

Introducción

El prototipado es un término que no encontraremos en el Diccionario de la Real Academia Española (DRAE), no obstante, es una palabra de uso común en el ámbito de la Interacción Persona Ordenador (IPO) que se utiliza como traducción del anglicismo *prototyping*, que viene a ser un sustantivo que aglutina el significado de la palabra prototipo con las diferentes herramientas y técnicas que permiten la producción de dichos prototipos.

El prototipado es útil ya que, desde que se empieza el desarrollo de un sistema interactivo, necesitamos probar partes del mismo con multitud de objetivos: verificar funcionalidades, averiguar aspectos relacionados con la interfaz del sistema, validar la navegación, probar nuevas posibilidades técnicas, etc.

Independientemente de que la técnica de desarrollo de software que se utilice sea más o menos formal es impensable llegar al final del desarrollo sin haber realizado comprobaciones a lo largo del camino que conduce este desarrollo. En este sentido, los prototipos son precisamente el mecanismo que permite realizar estas comprobaciones.

En definitiva, los prototipos que son documentos, diseños o sistemas que simulan o tienen implementadas partes del sistema final, constituyen una herramienta muy útil para hacer participar al usuario en el desarrollo y poder evaluar el producto desde las primeras fases de desarrollo.

¿Qué es un prototipo?

Un prototipo, en sentido genérico, es una implementación parcial, pero concreta de un sistema o una parte del mismo que principalmente se crean para explorar cuestiones sobre aspectos muy diversos del sistema durante el desarrollo del mismo.

El uso de los prototipos en el desarrollo de sistemas software no se limita sólo a probar las interacciones que los usuarios deben realizar, sino que son útiles también para otras actividades que se realizan durante el proceso, como por ejemplo su gran utilidad en la fase de recogida o análisis de requisitos en cuanto que amplía y mejora y la información necesaria para el desarrollo del sistema.

Si desarrollas una interfaz de usuario confusa para un sistema entonces no importará lo bueno o malo que sea el resto del sistema: los usuarios terminaran odiando lo que les has construido. Los desarrolladores consecuentes entienden al menos, y luego aplican, los fundamentos del desarrollo de interfaces de usuario. Esto implica la aplicación de algunas de las técnicas resumidas en la Tabla 1 para ayudar a explorar la interfaz de usuario junto a todos los involucrados en el proceso de desarrollo, así analizarán sus necesidades y entonces diseñarán interfaces que encajen con ellas.

Artefacto	Descripción
Prototipo de interfaz de usuario esencial	Es un prototipo independiente de la tecnología creado utilizando papel y que puede ser usado para identificar requisitos de interfaz de usuario.
pantalla/informe sketch	Es un borrador que muestra el diseño de los elementos principales de la interfaz, es decir, se utilizan pantallas, páginas HTML, o informes.
Prototipo de interfaz de usuario tradicional	Una interfaz de usuario que “funciona” y que presenta el sistema o a parte del sistema a los usuarios; con frecuencia se crea para diseñar una interfaz de usuario en detalle.
Diagrama de flujo de interfaz de usuario	Un diagrama que muestra los elementos principales de la interfaz y cómo los usuarios pasan de unos a otros; se usa para explorar la usabilidad (nivel alto) del sistema y para documentar y validar la interfaz.

Tabla 1. Artefactos de desarrollo de interfaces de usuario.

Prototipos de interfaz de usuario esenciales

La interfaz de usuario es la parte del software con la que el usuario interactúa directamente. Los prototipos de interfaz de usuario esenciales (Contantine et al., 1999) son modelos de baja-fidelidad de la interfaz de usuario que ofrecerá un sistema. En estos prototipos se recogen las ideas que hay detrás de la interfaz de usuario, pero no los detalles exactos o concretos. Estos prototipos de interfaz de usuario sirven para recopilar e identificar requisitos de interfaz de usuario de una manera independiente de la tecnología, en este sentido, son a los requisitos de interfaz lo que los casos de uso esenciales son a los requisitos de comportamiento. Un prototipo de interfaz esencial es el punto inicial de los prototipos de interfaz para desarrollar un sistema.

Existen dos diferencias básicas entre el prototipado esencial y el tradicional. La primera es que el modelado esencial se centra en los usuarios y en el uso del sistema y no en las características del sistema. La segunda es que las herramientas de prototipado en el prototipado esencial son más simples, incluyéndose pizarras, papel y notas adhesivas. El uso de herramientas de prototipado electrónicas termina introduciendo condicionantes y prototipos que no son independientes de la tecnología como sucede con el prototipado esencial.

Cuando un equipo de trabajo crea un prototipo esencial de interfaces de usuario, éste itera a través de las siguientes tareas:

1. El equipo **explora el uso del sistema** y lo hará a través de diferentes medios. Primero, trabaja utilizando una pizarra sobre la que discutir las ideas, partirán de algún boceto o borrador inicial y se favorecerán de la naturaleza dinámica que ofrece la pizarra para hacer entender y presentar la porción del sistema que se esté discutiendo en cada momento.
2. **Modelará utilizando elementos de interfaz de alto nivel.** Los elementos de interfaz de alto nivel, tales como potenciales pantallas o informes, pueden ser modelados utilizando papel. Se emplea el atributo potencial al referirse a las pantallas o informes para recalcar el hecho de que finalmente se trate de una pantalla o un informe real dependerá de una decisión de diseño. A cada página de papel se le dará un nombre y contendrá los elementos de interfaz de usuario de menor rango. El uso del papel tiene varias ventajas: pueden colgarse en una pared, son buenas para trabajar en grupo y que todos sean partícipes de su contenido e interactúe con ellas, son suficientemente grandes como para poner notas adhesivas en ellas, se puede escribir en ellas y se pueden guardar de una sesión de trabajo a la siguiente.
3. **Modelar los elementos de interfaz de bajo nivel.** Los elementos de interfaz de menor nivel, tal como campos de entrada, listas y containers (elementos de interfaz inferior que albergan otros elementos de interfaz de orden inferior) se modelan utilizando notas adhesivas. (Constantine et al., 1999) recomienda utilizar diferentes colores para los diferentes tipos de componentes, por ejemplo, colores brillantes/vivos (amarillo o rojo) para los elementos de interfaz activos como son los campos de entrada y colores más suaves o apagados (blanco o canela) para los elementos pasivos como son los containers. La Figura 1 muestra un prototipo de interfaz de usuario esencial para facilitar la matricula de estudiantes en seminarios. La nota adhesiva etiquetada con Student name es un container que incluye cuatro elementos activos: First name, Surname, Middle y Title. La otra nota representa una lista de los seminarios que un estudiante ha elegido o está matriculado. Nótese como cada nota tiene un nombre que describe su propósito, pero no como se implementa. Mirando cada nota adhesiva puede conocerse inmediatamente como se usa. Se utilizan tamaños de nota que indican el tamaño relativo de cada elemento de interfaz y el orden relativo de cada elemento de interfaz se indica por orden en el que aparecen en las notas.
4. **Explorar la usabilidad de tu interfaz.** Los sistemas altamente usables son fáciles de aprender, fáciles de usar, fáciles de recordar, se comenten pocos errores y hay una

subjetiva sensación de satisfacción (Nielsen, 1993). Este punto relacionado con la usabilidad se tratará en un apartado posterior.

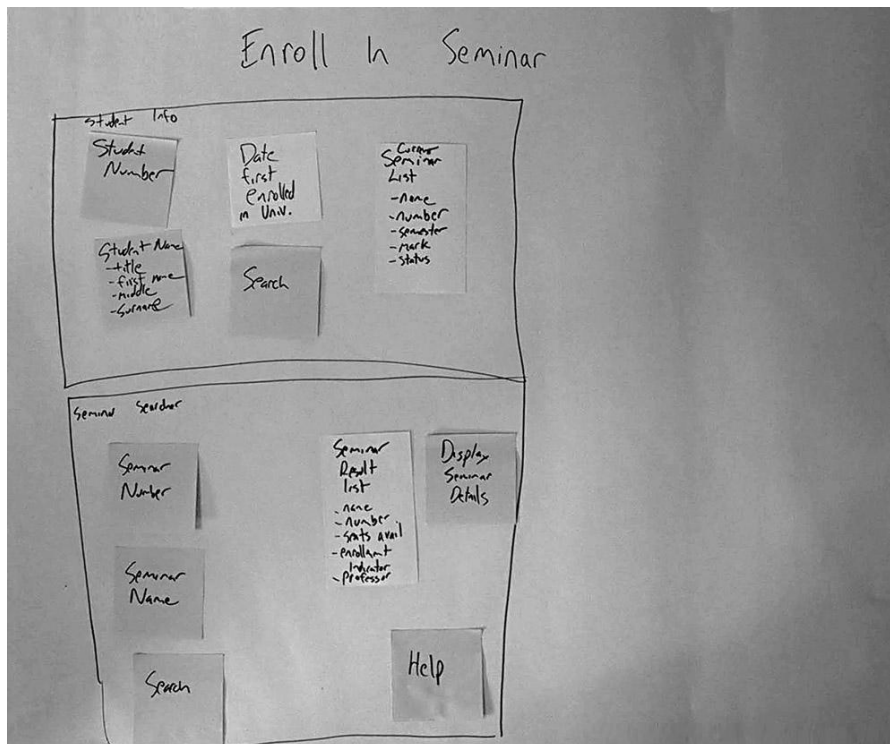


Figura 1. Prototipo de interfaz de usuario esencial para que un estudiante se matricule en un seminario.

Prototipo de interfaces de usuario tradicional

El prototipado de interfaces de usuario es una técnica de análisis iterativa en la que los usuarios están involucrados de forma activa en el desarrollo de bocetos de interfaz para un sistema. Los prototipos de interfaces de usuario tienen varios propósitos:

- Como artefacto de análisis permite explorar el espacio del problema con el resto de involucrados en el desarrollo;
- Como artefacto de diseño permite explorar el espacio de la solución que ofrece el sistema;
- Como vehículo de comunicación facilita el intercambio de opiniones y propuestas entre los involucrados en el desarrollo; y
- Como cimiento o base potencial permite sentar las bases sobre las que continuar el desarrollo del sistema.

En el diagrama de actividad mostrado en la Figura 2 hay cuatro pasos en el proceso de prototipado de cualquier interfaz de usuario. Esos cuatro pasos se corresponden con la estructura básica de cualquier propuesta de Ingeniería de Requisitos. El primer paso pasaría por determinar las necesidades del sistema, y no es un estado diferente al de elicitación de requisitos en el que nos preguntamos quién y para qué se utilizará el sistema. En esta fase, el modelado de interfaces se mueve desde la definición de requisitos al análisis, punto en el que

se decide evolucionar todo o parte de un prototipo esencial a un prototipo de interfaz tradicional. Esto implica convertir los borradores, elaborados a mano, a algo más sustancial.

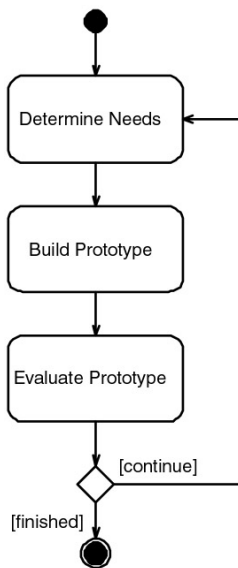
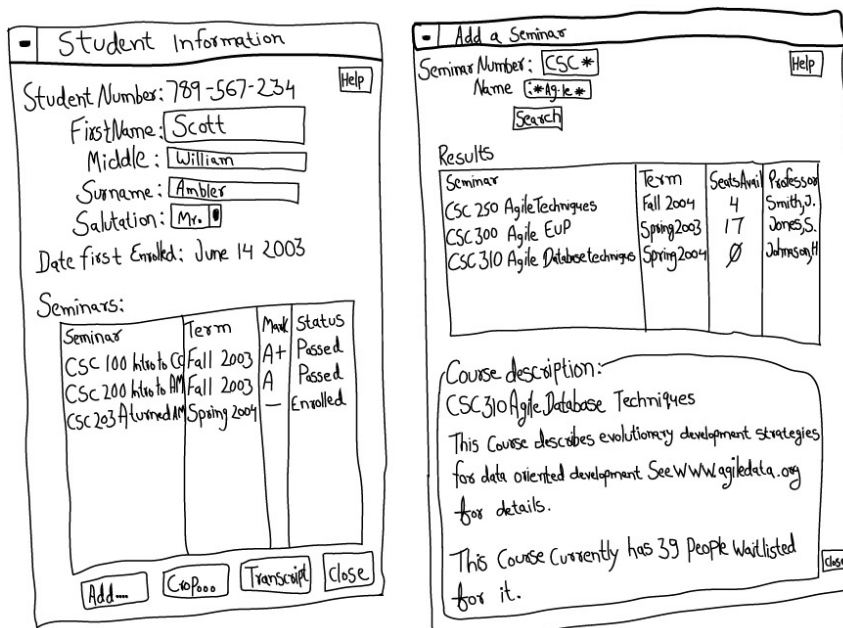


Figura 2. Ciclo iterativo del prototipado

Mientras se identifican y determinan las necesidades de los distintos involucrados en el desarrollo se pueden transformar los prototipos de interfaz de usuario esenciales, si es que fueron creados, y empezar a desarrollar *sketches*. La Figura 3 muestra un sketch de dos pantallas potenciales o páginas HTML basadas en los prototipos de interfaz de usuario de la Figura 1. Transformación entre prototipo esencial y tradicional, realmente no sería la palabra más adecuada ya que unos y otros no tienen porqué tener una asociación directa, en muchos casos unos y otros prototipos serán completamente diferentes.



Student Information

Student Number: 789-567-234 Help

First Name:

Middle:

Surname:

Salutation: ☐

Date first Enrolled: June 14 2003

Seminars:

Seminar	Term	Mark	Status
CSC 100 Intro to CS	Fall 2003	A+	Passed
CSC 200 Intro to AI	Fall 2003	A	Passed
CSC 203 AI Fundamentals	Spring 2004	-	Enrolled

Add a Seminar

Seminar Number: Help

Name: Search

Results

Seminar	Term	Seats Avail	Professor
CSC 250 Agile Techniques	Fall 2004	4	Smithy J.
CSC 300 Agile EUP	Spring 2003	17	Jones S.
CSC 310 Agile Database Techniques	Spring 2004	0	Johnsont H.

Course description:

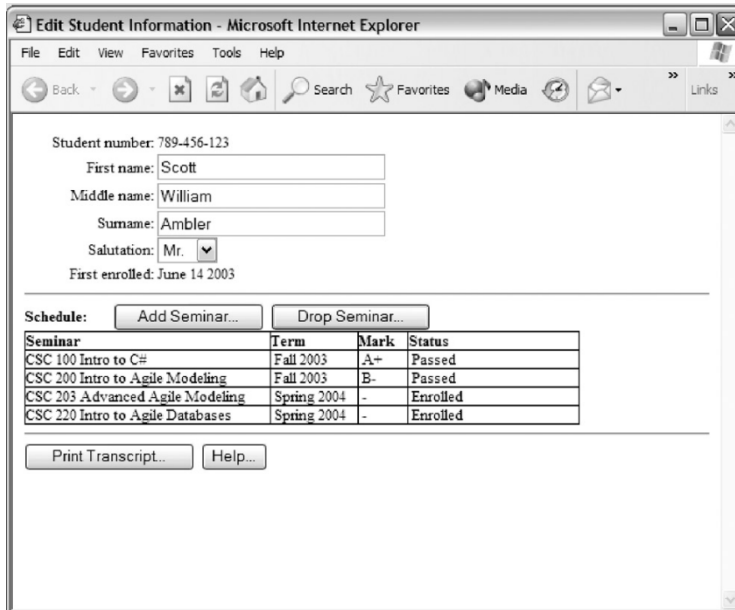
CSC 310 Agile Database Techniques

This Course describes evolutionary development strategies for data oriented development. See www.agiledata.org for details.

This Course currently has 39 People Waitlisted for it. Close

Figura 3. Pantallas sketch para matricularse en un seminario.

Como se puede observar en la Figura 3 hay decisiones de diseño y consideraciones relacionadas con aspectos tecnológicos y por motivos de cohesión el prototipo esencial original se ha dividido en dos pantallas.



Student number: 789-456-123

First name:

Middle name:

Surname:

Salutation: ▼

First enrolled: June 14 2003

Schedule:

Seminar	Term	Mark	Status
CSC 100 Intro to C#	Fall 2003	A+	Passed
CSC 200 Intro to Agile Modeling	Fall 2003	B-	Passed
CSC 203 Advanced Agile Modeling	Spring 2004	-	Enrolled
CSC 220 Intro to Agile Databases	Spring 2004	-	Enrolled

Figura 4. Una posible página web para matricularse en unos seminarios.

Una vez que se entienden las necesidades de interfaz de los involucrados en el uso y desarrollo del sistema, el siguiente paso es construir un prototipo. Usando una herramienta de prototipado o un lenguaje de alto nivel, se puede desarrollar pantallas, páginas e informes que los usuarios necesitan. Con la plataforma de interfaz seleccionada, puede empezar el proceso de convertir aspectos individuales recogidos en los prototipos esenciales a los prototipos tradicionales. En este punto se puede pasar por la realización de sketches como el mostrado en la Figura 3 o pasar directamente a la confección de interfaces como los mostrados en la Figura 4. La realización de sketches como los mostrados en la Figura 3 involucra más a los usuarios.

Es importante tener claro que no se necesita crear un prototipo para el sistema completo. Es muy habitual limitarse a prototipar una pequeña porción de la interfaz de usuario, quizás una simple pantalla o página HTML antes de proceder con la implementación. Debe tenerse también en cuenta que los desarrolladores trabajan siguiendo una filosofía evolutiva. Eso no quita para que en algunos casos, y puntualmente, se tenga que prototipar una porción importante del sistema, se tratará de ejercicios de visionado o para ayudar a definir el alcance del proyecto y justificar los recursos económicos que se solicitan.

Una vez que el prototipo ha sido construido, debe ser evaluado o validado por los involucrados en el proceso de desarrollo con el fin de verificar que se contemplan todas las necesidades. Algunas veces esto es tan fácil como pedirle a alguien que gaste algo de tiempo mirando lo que otro ha construido y otras veces es tan complicado como fijar una reunión en la que discutir y presentar el desarrollo al grupo de gente involucrada en el desarrollo. Cuando se evalúa un prototipo a nivel personal es útil hacerse preguntas como las siguientes:

- ¿qué destacarías del prototipo elaborado?
- ¿qué identificas como malo o controvertido?
- ¿qué decisiones de diseño has considerado y qué se ha perdido en el camino?

Después de evaluar el prototipo, se puede necesitar partir, modificar o incluso enriquecer. También se puede dar por finalizado el proceso de evaluación de prototipo cuando no surjan nuevas ideas que considerar. En cualquier otro caso habrá que volver a elicitar requisitos, volviendo a la primera etapa mostrada en la Figura 2.

(Ambler, 2004) sugiere los siguientes trucos y técnicas que le han sido útiles en su trabajo realizando prototipos:

1. Trabaja con usuarios reales.
2. Haz que los involucrados en el proceso de desarrollo trabajen con los prototipos.
3. Entiende el proceso real de negocio al que el sistema dará soporte.
4. Prototipa utilizando características que estés capacitado para desarrollar.
5. Consigue que alguien experto en interfaces te ayude a diseñar.
6. Explica qué es un prototipo y su utilidad.
7. Deja las decisiones de implementación para tan tarde como sea posible.

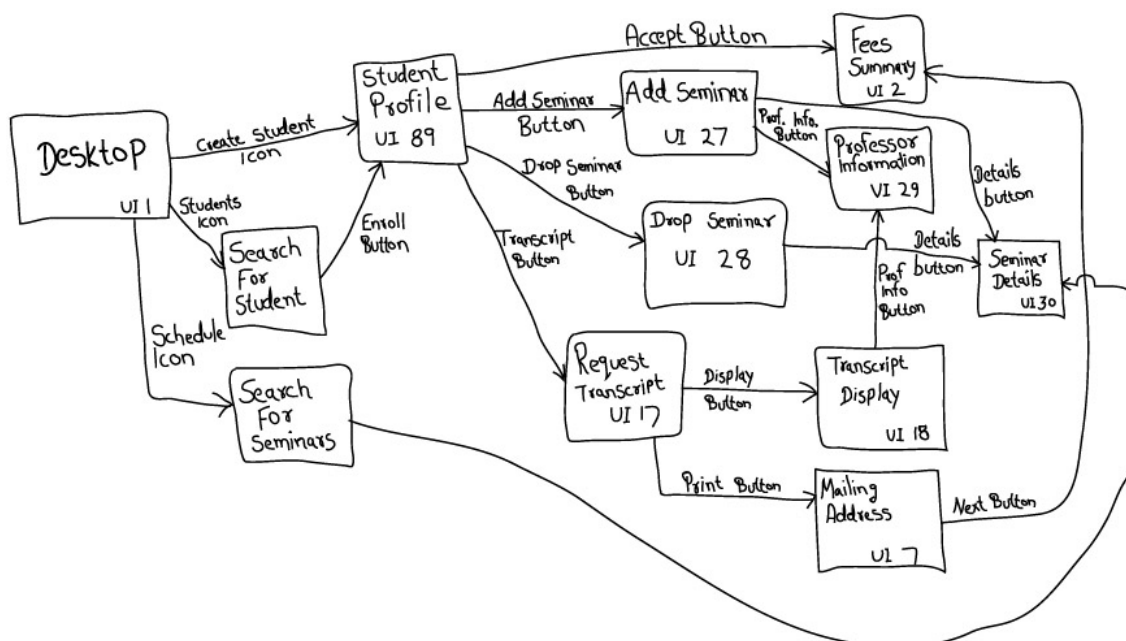


Figura 5. Diagrama de flujo de interfaz de usuario para un sistema de gestión de una universidad.

Diagramas de Flujo de Interfaces de Usuario

Los prototipos de interfaz son un medio excelente para explorar la interfaz, pero desafortunadamente no es fácil hacerse una idea de “todo el cuadro” y muchas veces nos quedamos relegados a cuestiones de diseño y grano más fino. Los diagramas de flujo de interfaz de usuario –también denominados storyboards, diagramas de navegación entre

ventanas (Page-Jones, 2000) o mapas de navegación contextual (Constantine et al., 1999)- permiten modelar y reflejar las relaciones entre los elementos de interfaz de usuario.

En la Figura 5 se visualiza un diagrama de flujo para un sistema gestor de una universidad. Las cajas representan los elementos de interfaz de nivel superior, modelados como instancias, y las flechas representan los posibles flujos entre ellos, modelados como transiciones en los diagramas de actividad.

Los diagramas de flujo de interfaz de usuario son útiles para dos propósitos. Primero, se usan para modelar las interacciones que los usuarios tienen con su software, como se define en los casos de uso. Por ejemplo, un caso de uso no sólo se referirá a varias pantallas también facilita señales o indicios como estas pantallas se usan. Basándose en esta información, los desarrolladores de diagramas de flujo de interfaz de usuario reflejan la visión de comportamiento de los casos de uso. Segundo, como se muestra en la Figura 5, permiten tomar conciencia del comportamiento de la interfaz de usuario a nivel alto. Esta visión es la vista conjunta de todos los comportamientos reflejados en los distintos casos de uso identificados y es lo que se denomina visión arquitectónica de la interfaz de usuario (Constantine et al., 1999)

Puesto que los diagramas de flujo de interfaz de usuario ofrecen una visión de alto nivel de la interfaz de usuario del sistema, con ellos se puede conseguir entender cómo se espera que el sistema funcione. Además, el flujo de interfaz de usuario también sirve para determinar si la interfaz de usuario será usable, si hay muchas cajas y muchas conexiones entre ellas es que el sistema probablemente será más complejo de aprender y entender.

Desafortunadamente, UML no ofrece esta clase de diagramas. Esperemos que la Object Management Group (OMG) defina un perfil para modelar el flujo de interfaces de un producto software a la vista de las ventajas que podrían obtenerse con su disponibilidad.

Usabilidad

Un aspecto importante que debe considerarse cuando se desarrolla un prototipo de interfaz de usuario esencial o tradicional para un sistema, es el de la usabilidad. Un sistema informático es usable cuando puede utilizarse para realizar una determinada tarea de forma eficiente.

Centrándonos, por ejemplo, en las aplicaciones para Internet, se podría decir que una web es usable cuando (véase Figura 6):

- **aporta información** relevante y frecuentemente actualizada
- resulta **fácil de aprender y recordar**
- permite **realizar las tareas** para las que fue diseñado de forma rápida y sencilla
- genera **pocos errores**
- proporciona una experiencia subjetivamente **agradable**



Figura 6. Descomposición y caracterización de la usabilidad de un sistema (Nielsen, 1993)

¿Por qué la usabilidad es importante? Primero, al centrarse inicialmente en el uso y en la usabilidad en lugar de en las características o la funcionalidad del usuario, y en sus usuarios y en sus necesidades más que en las interfaces, los sistemas se pueden convertir en entes menos complejas, simples y menos caras. Segundo, los mejores sistemas son aquellos placenteros, satisfactorios y la gente se siente bien al utilizarlos, en definitiva son usables. Tercero, los sistemas que son difíciles de utilizar suelen ser difíciles de aprender y más proclives al cambio y a la necesidad de mantenimiento. Cuarto, conforme los usuarios se acostumbran a utilizar productos software se hacen más entendidos y exigentes y menos tolerantes a productos diseñados y elaborados de forma pobre.

Estrategias de diseño de interfaces de usuario

Cuando desarrollas interfaces de usuario debes tener presente y considerar técnicas y principios básicos de diseño de interfaces de usuario. La experiencia demuestra que al utilizar dichas recomendaciones los resultados son mucho más factibles. Otra recomendación es que dentro del proyecto de desarrollo del sistema haya alguien que juegue el papel de experto en diseños de interfaz de usuario. De otra forma esa labor deben realizarla desarrolladores habituales que no siempre tienen conocimientos suficientes sobre el tema.

En (Constantine e al, 1999) se describe una colección de principios básicos para mejorar la calidad del diseño de las interfaces de usuario. Estos principios son:

1. **El principio de estructura.** Los diseños deberían organizarse adecuadamente, modelos claros, consistentes que sean aparentes y fáciles de reconocer por los usuarios, donde se pongan las cosas relacionadas juntas y separadas aquellas otras que no tienen nada que ver.
2. **El principio de simplicidad.** El diseño debería ser simple, las tareas habituales deben ser fáciles de hacer, presentarse de forma clara y simple, utilizando el lenguaje del usuario, y proporcionando una asistencia significativa.
3. **El principio de visibilidad.** Los diseños realizados deberían facilitar todas las opciones y material necesario para que una tarea pueda ser realizada sin distracciones o

información redundante o innecesaria. Los buenos diseños no saturan o confunden al usuario con muchas alternativas o información superflua.

4. **El principio de feedback.** El diseño elaborado debería informar al usuario de las acciones, cambios de estado o condición, de los errores que surjan o de las excepciones relevantes y de interés para el usuario a través de mensajes elaborados utilizando lenguaje claro, conciso, correcto y familiar para los usuarios.
5. **El principio de tolerancia.** Tus diseños deberían ser flexibles y tolerantes al error, reduciendo los errores y permitiendo operaciones de hacer-y-deshacer.
6. **El principio de reutilización.** Tus diseños deberían reutilizar componentes internos y externos y sus comportamientos. La consistencia debe primar en tus diseños eso facilita aprender y recordar los sistemas.

Además de los principios listados arriba, se recogen seguidamente trucos, recomendaciones y técnicas que se han demostrado válidas:

1. **Consistencia, consistencia, consistencia.**
2. **Considerar y seguir los estándares.**
3. **Ser consciente y estar convencido de la necesidad de consideración de los estándares relacionados con las interfaces de usuario.**
4. **Explica las reglas.**
5. **La navegación entre los elementos de interfaz de primer nivel es importante.**
6. **La navegación en las pantallas es importante.**
7. **Cuida tus mensajes, etiquetas y rótulos.**
8. **Entiende las componentes de interfaz disponibles (widgets).**
9. **Ten en cuenta las aplicaciones similares que haya disponibles, pero no lo hagas a pie juntillas en lo que se refiere a diseño de interfaz: cuestiona y aprovecha.**
10. **Usa colores adecuados.**
11. **Sigue la regla del contraste.**
12. **Alinea los campos de forma efectiva.**
13. **Ten por seguro que tus usuarios cometerán errores.**
14. **Justifica la información de forma adecuada.**
15. **Tus diseños deberían ser intuitivos (Raskin, 1994).**
16. **No crees interfaces de usuario difíciles de entender o de utilizar.**
17. **Agrupar las cosas de forma efectiva.**

Desarrollo de la práctica

En las siguientes referencias puedes encontrar y, en algunos casos, descargar diferentes herramientas desarrolladas para la elaboración de prototipos de diferente tipo.

GUILayout: <http://www.bitfolge.de/pubs/thesis/>

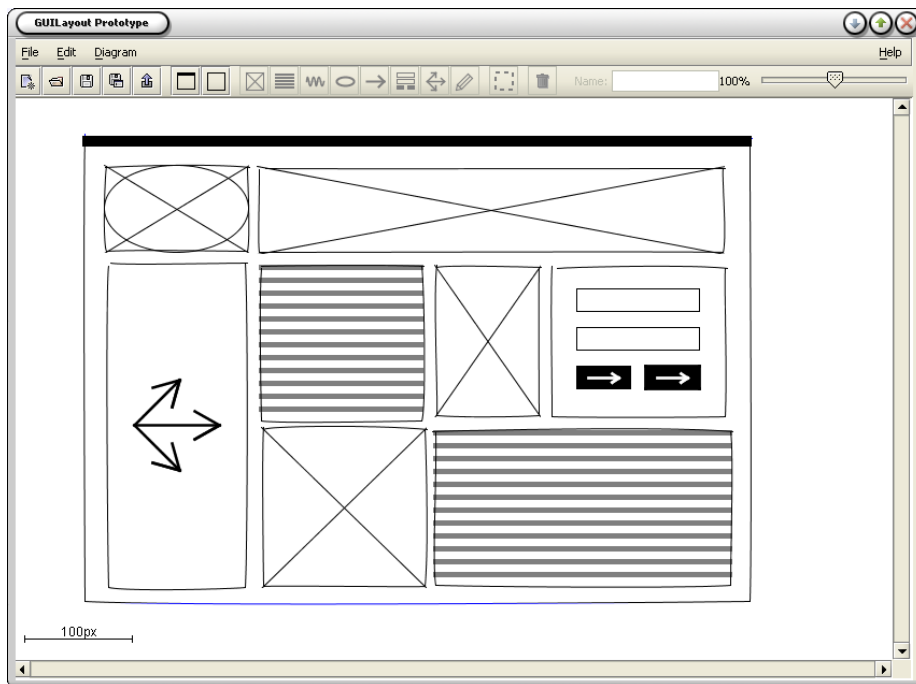


Figura 7. Ventana principal de la aplicación GUILayout

Denim: <http://dub.washington.edu/projects/denim/>

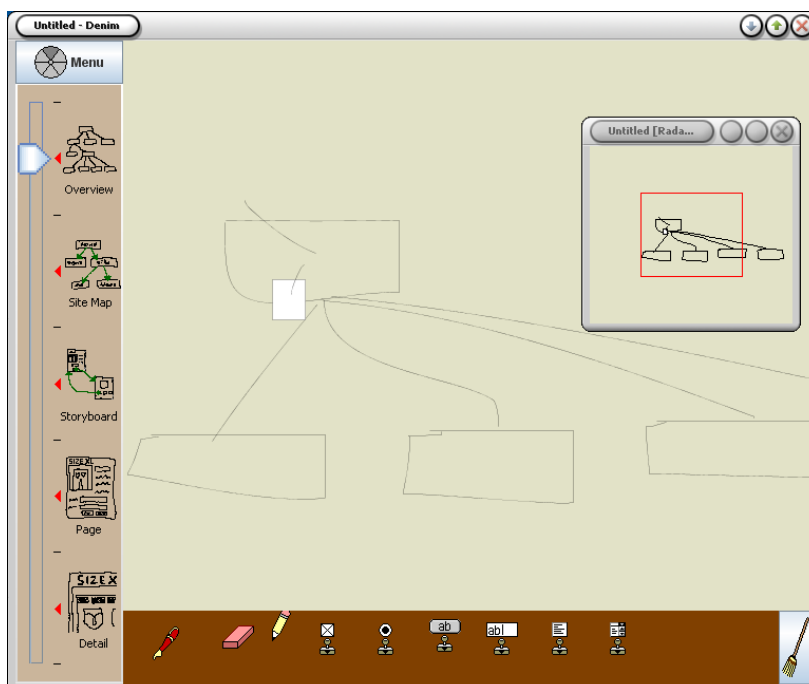


Figura 8. Ventana principal de la aplicación DENIM

CanonSketch: <http://dme.uma.pt/projects/canonsketch/>

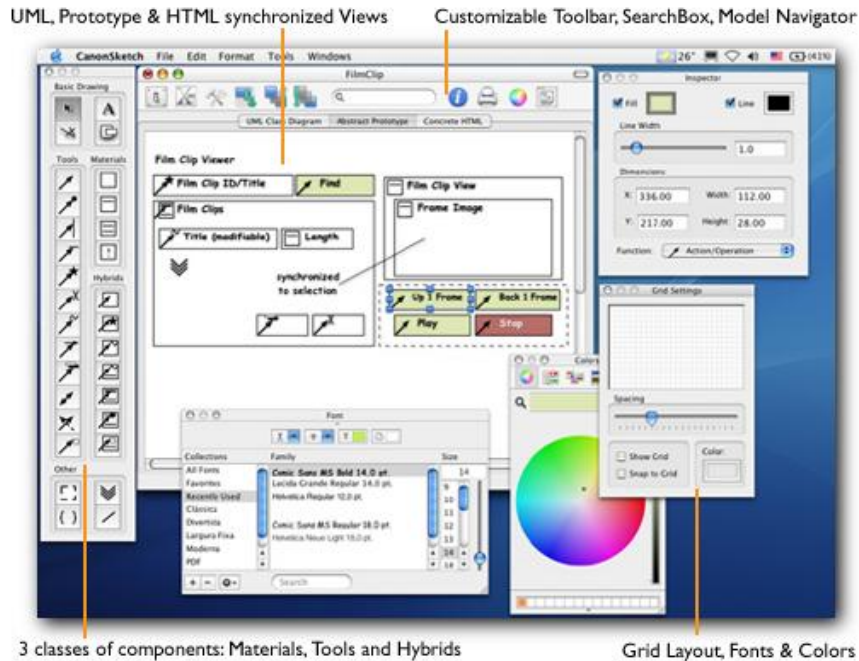


Figura 9. Entorno de trabajo definido en CanonSketch para Mac OS

SketchiXML: <http://www.usixml.org>

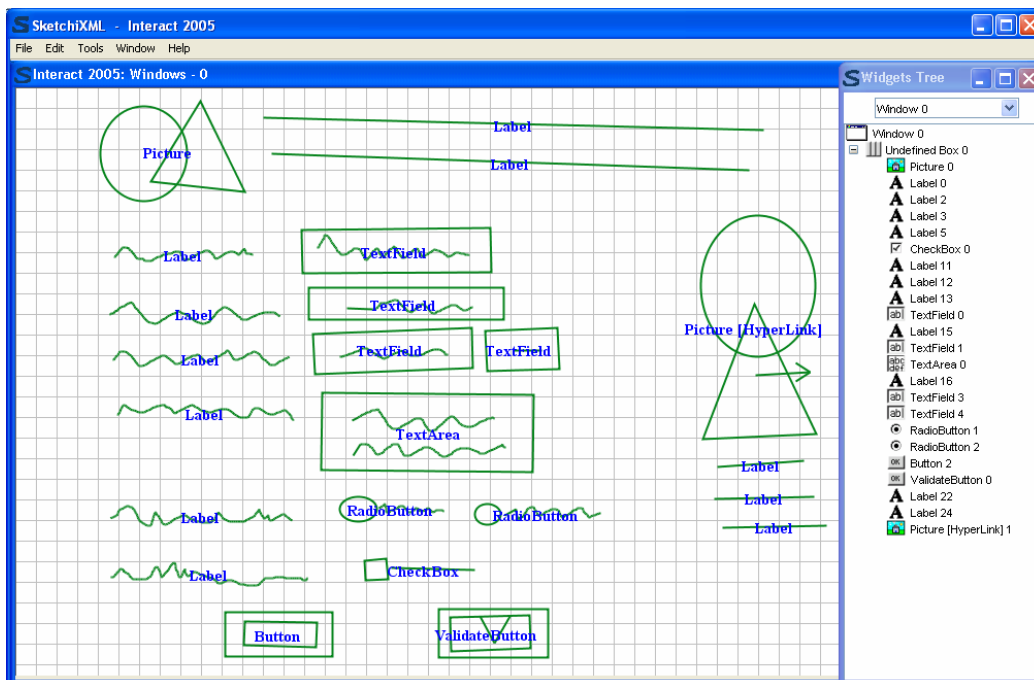


Figura 10. Pantalla de trabajo de SketchiXML

Elije una de las herramientas anteriores, alguna que puedas descargar (GUILayout o DENIM) y trata de esbozar una página web con la que estés familiarizado o aprovecha para diseñar tu

propia página web personal. Comenta tus impresiones utilizando las herramienta elegida y sugiere cambios o modificaciones que piensas que podrían ser útiles.

Bibliografía

Ambler, S. (2004). *The Object Primer*, Third Edition. Douglas Baldwin and Greg W. Scragg. Cambridge University Press.

Constantine, L. L., and Lockwood, L. A. D. (1999). *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*. New York: ACM Press.

Mayhew, D. J. (1992). *Principles and Guidelines in Software User Interface Design*. Englewood Cliffs, NJ: Prentice Hall.

Nielsen, J. (1993). *Usability engineering*. Boston, AP Professional.

Page-Jones, M. (2000). *Fundamentals of Object-Oriented Design in UML*. New York: Dorset-House.

Raskin, J. (1994). Intuitive equals familiar. *Communications of the ACM*, 37(9), 17–18.