Ingeniería en Software

**Trabajo Práctico N° 2**

“Usabilidad – Las ocho reglas doradas”

Integrantes:

García, Gabriel

Sastre, Yamila

Turno Noche

**Trabajo Práctico Nro. 2 –**

**“Usabilidad – Las ocho reglas doradas”**

* **Objetivo:** Evaluar usabilidad en páginas seleccionadas
* **Conocimiento Previo:** Lectura de la información brindada en el práctico
* **Premisas:** Analizar las página y colocar los puntajes según lo estipulado. Cada uno de los grupos expone los errores.
* **La realización del trabajo práctico “es grupal”** y hay revisiones cruzadas con otros equipos.

**Al finalizar deben entregar:**

* Informe del práctico.
* Informe de lecciones aprendidas.

* **Recursos para realizar el TP:** el sistema sobre el cual se aplicará el análisis del mismo, una notebook.

Ejercicio 1:

Debe evaluar las páginas:

* <http://www.fravega.com/>
* <http://www.launiondigital.com.ar/>

Cada grupo hará el test de dos sitios asignados  y deberá sumar puntos, según el tipo de error y los valores de la siguiente tabla:

|  |
| --- |
|  |
| Error |  | Puntos |  | Descripción |  |
|  |  |  |  |  |  |
| Fatales |  | 40 |  | Errores conceptuales, aplicación de un modelo mental erróneo o errores que impiden terminar la tarea exitosamente. |  |
|  |  |  |  |  |  |
| Graves |  | 15 |  | Errores que causan tal confusión en el usuario que ponen en peligro el éxito de la tarea. |  |
|  |  |  |  |  |  |
| Leves |  | 5 |  | Situaciones que pueden confundir momentáneamente al usuario pero muy raramente afectan el éxito de la tarea. |  |
|  |  |  |  |  |  |
| Cosméticos |  | 1 |  | Opiniones o cambios que no impactan en el diseño o performance del sitio sino en su apreciación. |  |
|  |  |  |  |  |  |

1. <http://www.fravega.com/>

|  |  |  |
| --- | --- | --- |
| Error |  | Puntos |
|  |  |  |
| Fatales |  | 0 |
|  |  |  |
| Graves |  | 0 |
|  |  |  |
| Leves |  | 5 |
|  |  |  |
| Cosméticos |  | 1 |
| Total |  | 6 |

Algunos errores y apreciaciones:

Opciones de menú repetidas en otros menús. (5)

Búsqueda de artículos compleja. (1)

1. <http://www.launiondigital.com.ar/>

|  |  |  |
| --- | --- | --- |
| Error |  | Puntos |
|  |  |  |
| Fatales |  | 0 |
|  |  |  |
| Graves |  | 15 |
|  |  |  |
| Leves |  | 0 |
|  |  |  |
| Cosméticos |  | 1 |
| Total |  | 16 |

Algunos errores y apreciaciones:

Link que no funciona. (15) (¿cuál?)

Presentación de imágenes de muy baja calidad. (1)

Al finalizar la clase se verá que errores encontró cada uno de los grupos

**Principios de Usabilidad**

**Que es la usabilidad?**

* Es la facilidad con que las personas pueden utilizar una herramienta particular o cualquier otro objeto fabricado por humanos con el fin de alcanzar un objeto concreto. En interacción persona-servidor: Se refiere a la claridad y la elegancia con que se diseña la interacción con un programa de ordenador o sitio web.
* Son técnicas que ayudan a los seres humanos a realizar tareas en entornos gráficos de un sistema.

**Conceptos muy importantes**.

Trabajamos para seres humanos, que quieren realizar una tarea de una forma sencilla y eficaz y en este caso particular, la deben realizar frente a un ordenador en un entorno gráfico, la web. La usabilidad ayuda a que esta tarea se realice de una forma sencilla analizando el comportamiento humano, y los pasos necesarios para ejecutar la tarea de una forma eficaz. Quizás suene un poco a trabajo industrial, pero al hacer referencia al término “tarea”, estamos haciendo referencia a buscar un programa que queremos ver en la tele, a ver información sobre mi equipo de fútbol favorito, a comprar un libro sobre artes marciales.

**¿Hace falta la usabilidad para hacer un sistema web?**

Sea tu web un portal o una página personal, no olvides que la gente que entra en tu web es porque básicamente busca algo. Hacerlo sencillo es decisión tuya, pero seguro que entre todos podemos hacer de internet un sitio mejor donde la gente no se sienta perdida o extraviada o sencillamente defraudada.

***Existen tres principios básicos de Usabilidad que nunca deben ser olvidados en el diseño de una interfaz:***

**Primer Principio: Reconocer la Diversidad**

Es importantísimo siempre tener en mente que cada interfaz está hecha para diferentes tareas, no puede haber una interfaz que funcione efectivamente para todas las tareas. Además de que cada interfaz debe estar orientada a una tarea en especial, también se debe tener en cuenta que no hay usuarios iguales.

Teniendo estos factores en cuenta existen una cantidad exorbitante de posibilidades para la creación de interfaces. La creación de interfaces para cada individuo es simplemente imposible, es por tanto, que los usuarios se deben categorizar. Las categorías pueden estar basadas en muchos criterios, por ejemplo el nivel de destreza o conocimientos de un programa o tema (novicios, intermedios o expertos) cada uno con diferentes necesidades en las interfaces. Otro criterio puede ser el tipo de trabajo desempeñado, por ejemplo, una interfaz para una industria puede abarcar interfaces especiales para obreros, supervisor, gerente, etc. Es necesario analizar a fondo la forma en la cual se llevará a cabo la interacción entre el usuario y la interfaz. Se debe determinar el total de las tareas que pueden llevarse a cabo mediante la interfaz y cada una de estas separarse en acciones más pequeñas, y estás a su vez en otras aún más básicas.

Se debe tener un control del nivel de granularidad de las tareas más básicas, pues tener pocas causa problemas al tener que realizar muchas tareas básicas para realizar una tarea. Así como tener muy pocas tareas básicas lleva a tener muchas tareas de alto nivel, confundiendo al usuario con el gran número de posibles acciones.

En el afán de crear la mejor interfaz posible, es necesario estudiar las formas en las cuales puede llevarse a cabo la interacción, teniendo en cuenta los requerimientos y posibilidades (físicas y mentales) de los usuarios. Así, podemos reconocer las siguientes formas de interacción:

* **Manipulación Directa.** Cuando el usuario interactúa con la representación visual del entorno sobre el que trabaja. Este tipo de interacción es sumamente sencilla para el usuario y extraordinariamente intuitiva. Requiere poco, o nulo, entrenamiento previo para el uso de la interfaz.
* **Selección de Menús.** Los usuarios seleccionan los elementos con los que se trabaja de una lista con posibilidades. Este tipo de interacción nos permite utilizar muy poca memoria para realizar los trabajos. Requiere mucho esfuerzo por parte de los diseñadores para asegurar que los elementos se encuentren donde tengan un significado para los usuarios.
* **Llenado de Formas**. Usado cuando es necesario ingresar datos a los sistemas. Es generalmente una forma en la cual los usuarios pueden introducir gran cantidad de errores. Es necesario ser claros en los tipos de datos pedidos, así como permitir a los usuarios muchas libertades como moverse a través de los cambios y que no todos los campos de la forma sean obligatorios.
* **Lenguajes de Comandos**. Los lenguajes de comandos son utilizados frecuentemente por usuarios de nivel avanzado. Los comandos dan gran flexibilidad y libertad de movimiento, así como dan la capacidad al usuario de mejorar las características del sistema. Son por lo general lentos de aprender, pero sumamente rápidos una vez dominados.
* **Lenguaje Natural.** La interacción mediante lenguaje natural entre una computadora y un humano es un tema que aún no ha sido resuelto. Es verdad que abre una gran cantidad de posibilidades, pues un usuario no tendría por qué tomar cursos de entrenamiento para utilizar el sistema (solo conocimiento previo de las tareas a realizar, lenguaje técnico, etc.). Hasta el momento, aunque existen muchos intentos de interfaces de lenguaje natural, el panorama actual no es muy alentador en dominios muy generales. Para dominios específicos de conocimiento y gramáticas reducidas, es posible encontrar buenos sistemas, aunque siempre faltando los elementos humanos por esenciales.

**Segundo Principio:** [**8 reglas de Oro sobre la usabilidad**](http://anisgottcreativo.wordpress.com/2011/08/19/8-reglas-de-oro-sobre-la-usabilidad/)

1. **Esforzarse para conseguir consistencia**

Deberíamos exigirnos secuencias de acciones consistentes, utilizarse terminología consistente en los mensajes, menús y pantallas de ayuda, emplearse de forma consistente el color[[1]](#endnote-1), composición, mayúsculas, fuentes, etc.

Las excepciones, como la necesaria confirmación de la orden de borrado o no mostrar los caracteres de las contraseñas, deberían ser comprensibles y limitadas en número.

1. Todos los componentes de la interfaz deben ser uniformes y seguir un conjunto de reglas simples.
2. Ayuda a generalizar conocimiento de un aspecto del sistema a los otros.
3. Prever la frustración inducida cuando el sistema no se comporta de la manera esperada y lógica.
4. Ejemplos:
5. Mismas palabras o códigos utilizados
6. Posición y orden de controles y botones (ayuda, status, salida, etc.)
7. Ítems de un menú colocados siempre en la misma posición (salir = 0)
8. Comandos como Ayuda, Status, Copiar, etc. deben estar siempre disponibles
9. Consistencia con el sistema operativo y otros programas
10. **Atender la usabilidad universal.**

Reconocer las necesidades de los diversos usuarios, facilitando la transformación del contenido.

Diferencias entre segmentos: Principiante/experto, edades, discapacidades y diversidad tecnológica.

Ejemplo:

1. Usuarios Casuales
2. Tutoriales
3. Wizards
4. Prompts (qué hacer ahora)
5. Ayuda
6. En línea
7. ToolTips
8. Ventanas (sensibles a contexto)
9. Modo Simple: Los comandos más complejos están escondidos
10. Manipulación Directa
11. Uso de Valores por Defecto
12. Usuarios Expertos
13. Shortcuts de teclado
14. Línea de Comandos
15. Modo Experto
16. Eliminación de Prompts y Warnings
17. Interfaz Extensible y Customizable
18. **Ofrecer realimentación informativa**

Para cada acción del sistema debería haber una realimentación por parte del sistema.

1. Aspectos:
2. Qué está haciendo el sistema.
3. Cómo se interpretan los comandos del Usuario.
4. El usuario debe saber en todo momento que está pasando.
5. Tipos de Feedback:
6. Respuesta a un comando del usuario: Movimiento del cursor, selección de un Menú, etc.
7. Estado Actual: Brocha seleccionada, color, posición, dirección de la carpeta, scroll bars.
8. Procesamiento por parte de la máquina: instantáneos, cursores de espera, diálogos explicativos, barras de progreso.
9. **Diseñar diálogos para conducir a la finalización**

Crear sistemas claros de conducción hasta el final del proceso de las secuencias de acción del usuario.

Ej.: Al final de una compra aparece una ventana o un texto que te indica que tú compra ha finalizado.

1. La interacción debe ser sencilla y comportarse de la manera en que el usuario lo espera.
2. Minimizar elementos de la interfaz
3. Menos para aprender, para equivocarse, distraerse.
4. Orden Natural de la Información:
5. Agrupar gráficamente información relacionada.
6. El orden de acceso a la información debe ser como el usuario la espera
7. Esconder o eliminar información irrelevante o usada ocasionalmente
8. Puede distraer al usuario de las informaciones importantes.
9. Utilizar el Lenguaje del Usuario:
10. Usar terminologías e iconografías familiares para el Usuario
11. Traducir mensajes de error al lenguaje del Usuario
12. **Prevenir errores**

Diseñar el sistema de forma que los usuarios no puedan cometer errores serios. Ej.: deshabilitar elementos necesarios, etc…

Si fuera así el sistema debe de indicarle al usuario el error por medio de diferentes interfaces textuales o metafóricos.

1. Nunca ofrecer un comando que lleve a un mensaje como: "Comando Ilegal": Desactivarlo
2. Borrar un objeto que no puede ser borrado
3. Aplicar estilo de letra a un objeto que no sea texto
4. Utilizar controles que impidan introducir datos erróneos:
5. Campos de Texto numéricos
6. Fechas
7. Scroll Bars
8. ListBoxes
9. Los comandos que ofrece el sistema deben ser válidos en el contexto
10. Esto permite simplificar la interfaz, ofreciendo únicamente los comandos relevantes al contexto (ej.: Diálogo de Selección, barra de Dibujos)
11. Advertir al Usuario
12. Es menos deseable, puesto que interrumpe el flujo de la interacción.
13. **Permitir deshacer acciones de forma fácil**

En la medida que podamos diseñar nuestro interfaz, el usuario puede tener la posibilidad de remediar acciones erróneas, por lo tanto construir un sistema reversible. Este sistema suaviza la ansiedad que se puede producir en el usuario. Ej.: control + Z

1. Ejemplos: Deshacer (Undo): Para comandos completados, Abortar: Para comando en proceso y Cancelar: Para formas o conjunto de comandos.
2. **Dar soporte al control interno**

Los operadores experimentados desean firmemente tener la sensación de que están al mando de la interfaz y que la interfaz responde a sus acciones. Las acciones sorprendentes por parte de la interfaz, las secuencias de entrada de datos tediosas, la incapacidad o dificultad para obtener la información necesaria y la incapacidad para producir acciones deseadas, crea ansiedad e insatisfacción. EJ: panel de control de Windows

1. **Reducir la carga de la memoria a corto plazo**

Diseñar sistemas de visualización simples, la frecuencia de movimiento de ventana se reduzca y que se asigne suficiente tiempo de entrenamiento para códigos y secuencias de acciones.

\* La regla general que los humanos pueden recordar “siete más / menos dos elementos” de información)

*Nota: Escribir bien para internet es todo un arte. Pero siguiendo las reglas básicas de: (1) poner las conclusiones al principio y (2) escribir como un 25% de lo que pondrías en un papel, se puede llegar muy lejos. Leer en pantalla cuesta mucho, por lo que, en el caso de textos para internet, reduce y simplifica todo lo que puedas.*

**Tercer Principio: Prevenir Errores**

Es necesario partir del hecho que los usuarios son altamente propensos a cometer errores, por grandes o pequeños que los errores sean. Es necesario por tanto prevenir o ayudar a corregir los errores mediante el uso de mensajes altamente descriptivos sobre el origen de los errores y posibilidades de corrección. Una manera muy simple de prevenir en editores es proveer de comandos de muy alto nivel que realicen mediante una sola orden acciones completas. Otra manera de corregir o prevenir errores es corregir “en línea” los posibles errores, como por ejemplo en lenguajes de comandos antes de ejecutar una orden el intérprete de órdenes puede verificar la existencia de los comandos y que los argumentos sean correctos.

Link Recomendado:

* http://materias.fi.uba.ar/7507/content/20101/lecturas/usabilidad.pdf

1. Manejo del Color

   La tonalidad, distribución y contraste de los colores son aspectos básicos sino se quiere tener un diseño pesado o ausente de coherencia.

   Los colores tienen un profundo impacto en el comportamiento de las personas. Colores como el amarillo, el naranja y el rojo dan una sensación de calidez, además de ser colores que llaman la atención. Por otra parte, el azul o el verde son colores fríos, colores que trasmiten una sensación de calma, de quietud. Esta percepción emotiva y subjetiva de los colores puede influir no sólo en la armonía visual de la interfaz, sino determinar qué tanto las personas se identifiquen con la aplicación, y como consecuencia, en el grado de interacción entre ellas a través del sistema

   Innovar es un asunto delicado cuando no se respetan ciertos principios esenciales, como la necesidad de emplear contrastes fuertes entre las fuentes de texto y el fondo. Algunas personas consideran que los fondos claros y los textos oscuros son los más adecuados cuando se necesita trabajar mucho tiempo frente a una interfaz. Además se debe evitar el empleo de muchos colores diferentes, y de imágenes distractoras en el fondo.

   Escoger los colores y sus combinaciones debe tener como punto de partida la observación que se haga del grupo y de ciertos elementos que permitan identificar los colores que pueden hacer parte de su identidad

   Recomendaciones:

   1. No usar más de 5 a 7 colores por despliegue.
   2. Si se usan rojo y verde, localizarlos en el centro de la pantalla y no en la periferia.
   3. Usar el azul para fondos y áreas grandes.
   4. Es más difícil mantener el foco con colores saturados adyacentes, que sólo difieran en la cantidad de azul que hay en ellos.
   5. Usar colores cálidos para los elementos que deban aparentar estar más cerca.
   6. No dejar que sea sólo el color la única característica que diferencie información en áreas pequeñas.
   7. Evitar las combinaciones rojo/verde, rojo/azul, rojo/amarillo y azul/amarillo.
   8. Recordar que el significado de los colores es afectado por factores culturales y por experiencias personales

   [↑](#endnote-ref-1)