

# Interacción y dinámicas

Juan Morales del Olmo

20 de marzo de 2014

# Interacción hombre-máquina

Según el ISO 9241 “Ergonomics of Human-System Interaction”, los objetivos son maximizar:

- ▶ Efectividad
- ▶ Eficiencia
- ▶ Satisfacción

Si nos centramos más en la práctica. Se puede medir y evaluar:

- ▶ Tiempo en aprender a realizar la tarea
- ▶ Velocidad de realización (de la tarea de *benchmark*)
- ▶ Ratios de error, ya sea de número como de tiempo en solucionarlos
- ▶ Tiempo que los usuarios recuerdan cómo usar la interfaz
- ▶ Satisfacción subjetiva

# Diseño centrado en usuario

- ▶ **El usuario está involucrado en todas las etapas de diseño**
  - ▶ Minimiza los riesgos más importantes:
    - ▶ No conocer al **usuario**
    - ▶ No conocer la **tarea**

- ▶ Marida mal con el modelo desarrollo software *en cascada*
  - ▶ Se nos dan unos requisitos. . . que pueden estar equivocados
  - ▶ Lo que entregamos al final del proceso no tiene nada que ver con lo que el usuario esperaba

- ▶ Hay que utilizar un desarrollo iterativo
  - ▶ Se va en espiral:
    - ▶ Requisitos -> Diseño -> Implementación -> Pruebas -> ...
  - ▶ Basado en prototipos y/o funcionalidades

- ▶ Metodologías ágiles de desarrollo (ej: *SCRUM*) se adaptan muy bien
  - ▶ Invitar al usuario a las reuniones de diseño
  - ▶ Ciclos cortos (sprints de pocas semanas)



# Conocer al usuario

- ▶ Tipificar al usuarios esperados según:
  - ▶ Usuario novato
  - ▶ Con conocimiento pero uso intermitente
  - ▶ Experto y con uso frecuente del sistema (*Power user*)
- ▶ Experiencia con ordenadores (u otras interfaces similares)
- ▶ Discapacidades motoras y sensoriales
- ▶ Rangos de edad
- ▶ Entorno de trabajo

# Conocer la tarea

Engloba lo que en educación de requisitos se llama: *Requisitos funcionales y no funcionales*

- ▶ Identificar las tareas de Alto nivel y descomponerlas en tareas de bajo nivel
- ▶ Frecuencia de cada tarea (ej: poco usadas a menús)
- ▶ Lenguaje del dominio de aplicación (menús, colores)
- ▶ Tiempos que hay que satisfacer (ej: tiempo real, batch)
- ▶ Sesión típica de trabajo (ej: uso esporádico, 8 horas al día)
- ▶ Tolerancia a errores (ej: tarea crítica en central nuclear)

## Estilos de interacción

# Manipulación directa

- ▶ + Representa visualmente conceptos visuales
- ▶ + Fácil aprendizaje
- ▶ + Fácil memorización
- ▶ + Permite evitar errores
- ▶ + Alienta la exploración
- ▶ + Da mejor satisfacción subjetiva
- ▶ - Puede ser más difícil de programar
- ▶ - Más requisitos (ej: pantalla gráfica y dispositivos de apuntado)

# Selección en menús

- ▶ + El aprendizaje más corto
- ▶ + Reduce las pulsaciones de teclado
- ▶ + Estructura la toma de decisión
- ▶ + Restringe los posibles errores introducidos
- ▶ - Riesgo de demasiados menús
- ▶ - Los usuarios frecuentes van lento
- ▶ - Utilizan mucho espacio en pantalla
- ▶ - Requieren refrescos rápidos de pantalla

# Completado de formularios

- ▶ + Simplifica la entrada de datos
- ▶ + Poco entrenamiento
- ▶ + Guía la interacción del usuario
- ▶ - Consume mucho espacio en pantalla

# Lenguaje de mandatos

- ▶ + Flexible
- ▶ + Suele gustar a *power users*
- ▶ + Da mucha sensación de control
- ▶ + Historiales y creación de macros
- ▶ - El manejo de errores pobre, limitado a mensajes de error
- ▶ - Requiere mucho entrenamiento y memorización

# Lenguaje natural

- ▶ + Quita la barrera de aprender una sintaxis nueva
- ▶ - Requiere diálogos aclaratorios
- ▶ - No muestran el contexto (ej: ¿qué ofrecen?)
- ▶ - La interacción puede ser más lenta



## Guía de diseño

# Las 8 reglas de oro

- 1.- Persigue la coherencia   ▶ Para conseguir efectos similares, la secuencias de acciones a realizar debe ser similar.
  - ▶ Idéntica terminología para los mismos conceptos, es decir, evita usar sinónimos.
- 2.- Permite a los usuarios frecuentes utilizar accesos directos   ▶

A medida que la frecuencia de uso aumenta, también lo hacen los deseos del usuario para reducir el número de acciones y aumentar el ritmo de interacción.

  - ▶ Acrónimos y abreviaturas, las teclas de función, los comandos ocultos, y macro instalaciones son muy útiles para un usuario experto.

3.- Ofrece comentarios informativos ▶ Cada acción del usuario debe tener algún sistema de retroalimentación.

- ▶ Para acciones frecuentes y/o de poco efecto, la respuesta puede ser modesta.
- ▶ Para las poco frecuentes y/o las acciones principales, la respuesta debería ser más sustancial.
- ▶ Representar los objetos sobre los que se actúa (Direct manipulation) es una manera explícita de dar feedback.

4.- Diseña los diálogos para producir la consecución de la tarea ▶

Acciones secuenciales debe organizarse en grupos con un comienzo, un nudo y un final.

- ▶ Cuando se terminan la acción se tiene que informar adecuadamente al usuario.
- ▶ Dichos diálogos fomentan la sensación de logro, generando alivio y sitio en las mentes para otro grupo de acciones.
- ▶ Por ejemplo: compras por internet, instaladores.





# Principios destacados

- ▶ **Prevenir errores**
- ▶ **Asegurar el control del usuario y a la vez aumentar la automatización**

# Tiempos de respuesta

- ▶ Acciones percibidas como instantáneas: **50 - 150 ms** ( )
- ▶ Acciones frecuentes y simples, percibidas como inmediatas: **menos de 1 segundo**
- ▶ Acciones comunes: **2-4 segundos**
- ▶ Acciones complejas: **8-12 segundos**
- ▶ Más de **15 segundos** pierdes la atención del usuario

# Tiempos de respuesta (recomendaciones)

- ▶ **Es un elemento más de diseño:**
  - ▶ Lo normal es: mejor cuanto menor sean los tiempos de espera
  - ▶ Pero tiempos cortos llevan al usuario a pensar menos sus acciones
- ▶ Los tiempos de espera largos tiene que ser avisados
- ▶ Para esperas de más de 2 segundos se necesita enseñar algún feedback
- ▶ Para esperas de más de 10 segundos, enseñar el proceso
- ▶ **Lo peor son retardos inesperadas.** Hay que controlar la varianza de los tiempos de espera para la misma acción