

EI 1027 - Diseño e implementación de SI

Tema 0 - Qué es un SI

1- Definición de SI:

Conjunto interrelacionado de elementos que capturan, almacenan, tratan y distribuyen la información para dar apoyo a la toma de decisiones, el control, análisis y visión de la organización.

Formado por: Gente, HW, SF, redes de comunicaciones y recursos de datos.

2-Sistemas de información tradicionales:

Gestión de nóminas, inventario, marketing, atención al cliente.

Tecnología: Papel, calculadoras mecánicas, máquinas de escribir.

La información se guarda en archivos (documentos de papel)

3- Clasificación de SI:

Sistemas de información ejecutiva	Nivel estratégico	Data mining	Administradores de nivel superior
Sistema de apoyo de decisiones	Nivel de conocimientos	Data warehousing	Admins de nivel medio
Sistemas de apoyo a la gestión	Nivel administrativo	Sistemas OLTP	Trabajadores de conocimiento y datos
Sistema de procesamiento de transacciones	Nivel operacional	Sistemas OLTP	Administradores operativos

4-Enterprise Resource Planning (ERP)

Programas que pueden gestionar todo el conjunto de operaciones de negocio en una empresa.

Normalmente coordinan, planifican, controlan el inventario, la producción y las ventas.

5-Desarrollo del SI:

1. Sistema de investigación -> Entiende el problema
2. Sistema de análisis -> Entiende la solución
3. Sistema de diseño -> Selecciona y planea la mejor solución
4. Sistema de implementación -> Incorpora la solución
5. Sistema de mantenimiento y revisión -> Evalúa los resultados de la solución

Es un ciclo de 1 al 5 y vuelve al 1.

6-Requisitos de un SI empresarial

- Tienen que interactuar con la BBDD de la empresa o algún tipo de sistema de pago.
- El sistema tiene que tener en cuenta la lógica de negocio.
- El coste total (desarrollo + mantenimiento) es muy importante.
- Debe tener facilidad para adaptarse ante cambios.

A parte de esto los sistemas de proceso de transacciones deben:

- Tener un tiempo de respuesta inmediato.
- Tener alta disponibilidad.
- Una buena seguridad.
- Mantener la integridad de los datos.

7-Los SI se organizan en capas:

Arquitecturas de 1 capa: Aplicaciones monolíticas, de mantenimiento difícil (cualquier cambio implica tener que desarrollar de nuevo algunas partes de la aplicación)

Arquitecturas de 2 capas: Se separa la presentación y la lógica de negocio del acceso a los datos. Solo apta para aplicaciones simples. La no separación de la presentación y la lógica de negocio impide adaptarse a los cambios en los requisitos.

Arquitecturas de 3 capas: Separan presentación, lógica de negocio y capa de acceso a los datos. Es la mejor opción para aplicaciones empresariales en la mayor parte de los casos.

Arquitecturas de N capas: En el caso más general las capas se definen ad hoc (expresamente para algo). Caso típico: Separar el servidor web en una capa por motivos de rendimiento o seguridad.

8-Arquitectura de 3 capas:

Capa de presentación: Nivel superior de la aplicación es la interfaz de usuario. Su función principal es traducir las tareas y los resultados en algo que el usuario pueda entender.

Capa lógica: Esta capa coordina la aplicación, procesa órdenes, toma decisiones lógicas, hace cálculos, ... También mueve y procesa datos entre capas.

Capa de datos: Aquí la información se almacena y se recupera desde un sistema de BBDD o archivo.

9-Ventajas del uso de capas:

- Se puede cambiar una capa sin afectar a las otras (ejemplo: una nueva BBDD)
- Se pueden añadir nuevas reglas de negocio.
- Se puede dividir el trabajo de desarrollo en diferentes equipos.
- Se facilita enormemente el mantenimiento de la aplicación.
- Técnicas bien establecidas de diseño.

Tema 1 - Disseny Lògic

1- Repaso conceptos diseño lògico:

- Conjunto de relaciones o tablas.
- Claves primarias para cada tabla.
- Claves ajenas de las distintas tablas con sus reglas de modificación y borrado.
- Reglas de integridad necesarias.

2- Normalización:

- Primera Forma Normal (1FN)
 - Una relación està en 1FN si, y sólo si, todos sus dominios son atòmicos.
- Segunda Forma Normal (2FN)

- Una relación està en 2FN si, y sólo si, està en 1FN y, además, cada atributo no clave depende solo de la clave primària.

- Tercera Forma Normal (3FN)

- Una relación està en 3FN si, y sólo si, està en 2FN y, además, cada atributo no clave no depende transitivamente de la clave primària.

3- Transformación de UML a disseny lògic:

- Diagrama UML (los elementos deben tener su correspondiente en el diseño lògico)

- Clase → Tabla
- Atributo → Columnas de las tablas (no todos, cuidado con los calculados, como IMPORTE, que se calculará en la aplicación)
- Tipo → tipo de datos de las columnas.
- Asociaciones/Relaciones → Relaciones

- Pasos para el diseño lògico

1. Mapear las clases en tablas

- a. Cada clase UML es una tabla separada
- b. Cada atributo de clase es una columna de la tabla
- c. Atributos compuestos se dividen en columnas separadas
- d. Elegir uno o más atributos como clave primaria o crear uno artificial

2. Crear tablas por atributos multivaluados

- a. Crear una nueva tabla para cada atributo multivaluado
- b. Poner la clave primaria de la tabla a la que pertenece como a clave ajena

3. Representar relaciones de asociaciones

- a. (1:1)
 - i. Poner la clave primaria de una de las tablas como clave ajena de la otra tabla involucrada de la asociación
- b. (1:N)
 - i. Poner la clave primaria de la 1-tabla como ajena en la N-tabla
 - ii. Los atributos de la asociación se ponen en la n-tabla
- c. (N:M)

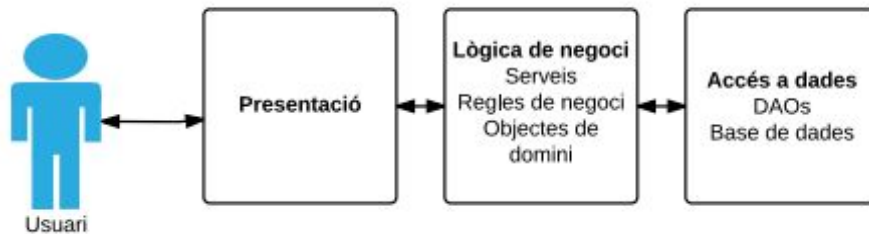
- i. Crear una nueva tabla que tendrá como clave primaria como mínimo la composición de las claves primarias de las dos tablas relacionadas
 - ii. Los atributos de la asociación irán en la nueva tabla
4. Representar relaciones de composición
 - a. Se representan igual que una relación de asociación normal
 - b. Cada clase componente se transforma en una tabla que tiene como clave ajena la clave primaria de la tabla compuesta
5. Representar relaciones de especialización
 - a. Añadir un nuevo atributo a la clase general para indicar la clase especial de cada fila
 - b. Si las clases especiales definen nuevos atributos, estos crean nuevas tablas que tienen como clave primaria y clave ajena la clave primaria de la clase general.

Tema 2 - Arquitectura y patrones para sistemas de información

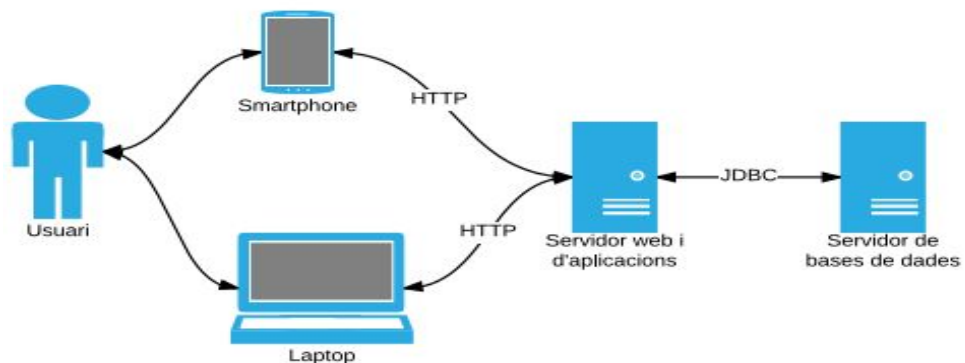
1-Arquitectura de un SI:

- Es el conjunto de las **principales** decisiones de diseño sobre el sistema
- “Principal” se refiere a decisiones globales:
 - Los componentes a partir de los cuales se construye el sistema
 - La interacción entre los componentes
 - Los componentes que gobernarán la composición de los componentes
 - Las restricciones a la composición
- Arquitectura Lógica (componentes, subsistemas y programas) VS físicas (PC)
 - Lógica:
 - 3 capas: interfaz + lógica de dominio + fuente de datos
 - 5 capas
 - Interficie
 - Lógica de la interficie
 - Lógica del dominio
 - Mapeado de datos (interficie con fuentes de datos)
 - Almacenamiento de datos

- Arquitectura lógica



- Arquitectura física



- Las capas lógicas, cuando se hace su implantación física, se traducen en una arquitectura de 1, 2, 3, ... n capas físicas:
 - 1 capa: no frecuente en entornos empresariales con aplicaciones más complejas
 - 2 o 3 capas es el más habitual → 2 escenarios:
 - Entorno web
 - Arquitectura cliente/servidor
- Entorno web
 - Beneficios:
 - Acceso universal con navegadores de internet
 - Estandarización de protocolos de comunicación (HTTP, SOAP)
 - Estandarización de lenguajes de representación de interfaces gráficas (HTML)
 - Disminución de costos al eliminar pérdidas asociadas al mantenimiento del programario en el PC cliente.

2. Patrons de disseny software

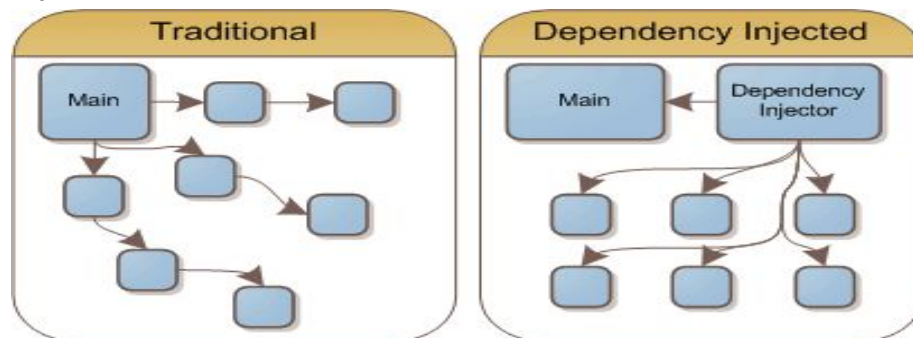
- Un patrón describe una posible solución correcta para un problema de diseño dentro de un **contexto específico**, describiendo las cualidades invariantes de todas las soluciones.
- Los patrones de diseño software son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software
- Solución a problemas con las siguientes características:
 - Se debe comprobar la efectividad dando solución a problemas similares.
 - Debe de ser rentable.
- Los patrones de diseño pretenden:
 - Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
 - Evitar la reiteración en la búsqueda de soluciones
 - Formalizar un vocabulario común entre diseñadores
 - Estandarizar la manera en que se realiza el diseño
 - Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente
- No pretenden:
 - Eliminar la creatividad inherente en el proceso de diseño
 - Imponer alternativas de diseño frente a otras

3. Patrones de disseny de SI

- Recordamos los requisitos específicos del SI empresariales:
 - Sistemas basados en componentes reutilizables
 - Tener en cuenta la lógica de negocio, que cambia a menudo.
 - El mantenimiento y la facilidad de adaptar la aplicación a los cambios que se van produciendo es fundamental.
- Se utilizan patrones para conseguir cumplir estos requisitos
- Algunos de los principales patrones son:
 - Inversión de control.
 - Inyección de dependèncias
 - MVC.
- Inversión de control (idC) y inyección de dependencias (ID)
 - Problema de diseño:
 - Las dependencias entre partes del código provoca problemas de modularidad y extensibilidad.
 - En código, las dependencias se traducen en referencias entre clases
 - Ejemplo:
 - El acceso a datos està programado contra el API propietario de una BD concreta
 - Solución:
 - Desacoplar las dependencias de código
 - Delegar la resolución de referencias a un componente genérico dedicado

- Inversión de control
 - Es una técnica que invierte el flujo tradicional.
 - Flujo tradicional: el código llama a las librerías
 - Inversión de control: Las librerías son las que llaman al código
 - Por ejemplo, en el framework Spring, la inversión de control consiste en dar control a una entidad externa a la aplicación, llamada “contenedor”, que gestiona las instancias (así como a su creación y destrucción)
 - Muchos patrones de hoy en día se basan en esta técnica
 - Diferentes formas de IdC:
 - Listener/observer
 - De construcción: Factoría, factoría abstracta o builder
 - Inyección de dependencias

- Inyección de dependencias



- Tipos ID:
 - Por interficie
 - Por propiedad
 - Por constructor
 - Spring soporta inyección de dependencias por constructor y por propiedad, pero se aconseja hacerlo a través de los métodos set (por propiedad)
- Modelo Vista Controlador (MVC)
 - Es un patrón software que separa los datos de la aplicación, la interfaz de usuario y la lógica de presentación en 3 componentes diferentes.
 - Es frecuente en aplicaciones web.
 - El flujo que sigue generalmente es:
 - El usuario interactúa con la interfaz de alguna manera.
 - El controlador recibe la notificación de la acción solicitada por el usuario y gestiona el evento con el gestor de eventos (handler) o callback.

- El controlador accede al modelo, actualizándolo, modificándolo a la manera apropiada para el usuario.
- El controlador delega a los objetos de la vista la tarea a desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo. El modelo no debe tener conocimientos directos sobre la vista.

Tema 3. Introducción a la usabilidad y diseño de interfaces

1-Definición de usabilidad:

Eficacia, eficiencia y satisfacción con la que un producto permite conseguir una serie de objetivos a un usuario.

2-Elementos de la usabilidad (Nielsen and Schniedermann):

- Facilidad de aprendizaje: Es fácil para los usuarios realizar las tareas básicas la primera vez que se encuentran ante el diseño?
- Eficiencia: Una vez el usuario ha aprendido el diseño con qué rapidez puede realizar las tareas.
- Recordable (memoria): Cuando se vuelve usar después de un tiempo con qué facilidad te puedes adaptar a su uso de nuevo.
- Errores: Cuántos errores cometen los usuarios, cómo son de graves y con qué facilidad se pueden recuperar de los errores.
- Satisfacción: Como es de agradable el uso de este diseño.

3-Medidas de usabilidad:

Definir la comunidad de usuarios objetivos y las clases de tareas asociadas a la interfaz.

- Tiempo de aprendizaje: El tiempo que necesita un usuario para aprender una tarea relevante
- Rendimiento: Cuanto tiempo cuesta realizar tareas relevantes.
- Tasa de errores de los usuarios: Cuantos y que tipos de errores se producen durante la realización de tareas relevantes.
- Retención en el tiempo: Hace referencia a que la frecuencia y uso ayuda a la retención de usuarios.
- Satisfacción subjetiva: Retroalimentación de usuarios mediante entrevistas, encuestas y comentarios informales.

4-Usabilidad universal:

Diversidad cultural e internacionalización

- Caracteres, números, caracteres especiales y diacríticos.
- Escritura de izquierda a derecha, de derecha a izquierda o vertical.
- Formatos de fecha y hora
- Formatos numéricos y de moneda.
- Pesos y medidas.
- Números de teléfono y direcciones.
- Nombres y títulos (Mr., Sra., Mme)
- Números de la seguridad social, identificación nacional y números de pasaporte.
- Capitalización y puntuación.
- Ordenación de secuencias.
- Iconos, botones, colores.
- Pluralización, gramática, ortografía.
- Etiqueta, política, tono, formalidad, metáforas.

5-Usabilidad universal (USUARIOS):

- Usuarios con discapacidades físicas (Menos costoso si se planifica desde el principio, las empresas han de cumplir con los requisitos legales.
- Gente mayor (Relativamente fácil de incluir, ayuda la configuración del sonido, los colores, el nivel de brillo, el tamaño de la fuente, etc...)
- Usuarios jóvenes (niños)

6-Proceso de diseño centrado en el usuario:

Determinar las capacidades de los usuarios

- Conoce a tu usuario.
- Edad, sexo, habilidad físicas y cognitivas, educación, origen cultural o étnico, formación, motivación, objetivos y personalidad.
- Diseñar objetivos basados en el nivel de capacidad:
 - Usuarios novatos.
 - Usuarios intermitentes y con suficientes conocimientos.
 - Usuarios frecuentes y expertos.
- Diseño multicapa

Identificar tareas

- Suele requerir muchas horas y hacer entrevistas a los usuarios.
- Hace falta descomponer tareas de alto nivel
- Analizar la frecuencia relativa de las tareas.

Consejos

- Dar consejo con un texto equivalente para cada elemento no textual.

- Para cada presentación multimedia que dependa de un temporizador, sincronizar alternativas equivalentes.
- La información proporcionada mediante un color también hay que proporcionarla sin color.
- Titular cada marco para facilitar la identificación y la navegación.

Organizar la pantalla (5 objetivos)

- Consistencia a la hora de mostrar los datos.
- Asimilación eficiente de la información por el usuario.
- Carga de memoria mínima por parte del usuario.
- Compatibilidad en la forma de mostrar los datos de entrada.
- Flexibilidad de los datos mostrados, que el usuario los pueda controlar.

Conseguir la atención del usuario

- Intensidad
- Marcas
- Medida
- Selección de fuentes
- Color
- Audio

Principios

- Más fundamentales y generalmente aplicables a las guías
- Necesitan más clarificación
- Cinco estilos de interacción fundamentales
 - Línea de comandos: Usuarios expertos, sentimiento de control e iniciativa, tasa de errores alta, entrenamiento necesario, memorización alta, mensajes de error y asistencia escasa.
 - Menús: Fácil de usar, de aprender y de programar. Lento al usarlo en grandes sistemas, número limitado de opciones en cada menú.
 - Formularios: Simplifica la entrada de dato, requiere un entrenamiento modesto, genera la ayuda conveniente. Consume espacio de la pantalla.
 - Manipulación directa: Representación continua de objetos de interés. Acciones rápidas, incrementales y reversibles que provocan un efecto visible inmediato en el objeto seleccionado. VENTAJAS: Nuevos usuarios aprenden más rápido, los expertos pueden trabajar rápidamente, los usuarios ven rápidamente el resultado de sus acciones, las acciones son reversibles. DESVENTAJAS: Hacen falta más recursos y es más difícil de programar, no todas las tareas pueden ser descritas

por objetos y no todas las acciones se pueden hacer directamente

- Interacción asistida: Metáfora de un asistente personal, el usuario no dirige la interacción, trabaja de forma cooperativa con el agente, se reduce el esfuerzo del usuario.
- Prevenir errores:
 - Los mensajes de error tienen que ser específicos, con tono positivo y constructivos
 - Acciones para prevenir:
 - Desactivar acciones inapropiadas.
 - Selección es la mejor manera de entrada libre.
 - Autocompletar.
- Automatización y control humano.
- Ocho reglas de oro del diseño de interfaces.

7-Las 8 reglas de oro en el diseño de interfaces:

1-Conseguir consistencia:

- Acciones similares deben tener consecuencias similares
- Visual, usar el mismo diseño de color, fuente, etc en todo momento.
- Relativa a las acciones(icono papelera para borrar)
- Limitar las excepciones al máximo.

2-Conseguir usabilidad universal:

- Tener en cuenta las diferencias entre las capacidades de los usuarios.

3-Ofrecer feedback informativo:

- El sistema tiene que responder a cada acción del usuario.
- La respuesta será ligera para actividades rutinarias.
- La respuesta será más elaborada para acciones infrecuentes e importantes.

4-Diseñar diálogos que indiquen que se ha completado la tarea:

- Las secuencias de acciones deben tener un inicio, parte media y un final.
- Al final se ha de dar una información clara de que ha acabado la tarea.

5-Prevenir errores:

- Desactivar menús o botones inadecuados.
- Utilizar autocompletar
- Si se comete un error el sistema a de detectarlo y ofrecer instrucciones simples, específicas y constructivas para la recuperación.
- Un error no tiene que cambiar el estado del sistema.

6-Permitir deshacer acciones:

- Se pueden deshacer acciones individuales o incluso grupos de acciones.

7-Dar sensación de control al usuario:

- Se puede permitir personalizar la aplicación o diseñar aplicaciones con un modo experto y un modo básico.

8-Reducir la carga de memoria a corto plazo:

- Los usuarios pueden tener en memoria a corto plazo 7+-2 cosas.
- No se deben hacer interfaces en las que el usuario tengan que recordar información entre pantallas.

8-Uso de prototipos:

- Necesario ya que lo que piensa el usuario no suele coincidir con lo que expresamos.
- **Prototipo:** Documento, diseño o sistema que simula partes del sistema final. La calidad de la representación puede variar entre media-alta. Es una herramienta útil para hacer partícipe al usuario en el diseño.
 - Tiene un tiempo de vida corto y ha de ser construido de forma rápida y eficiente.
 - Tipos:
 - Maqueta “para tirar”
 - Incremental -> Se construye con componentes separados.
 - Evolutivo -> Continúa utilizándose en el proceso evolutivo.
 - Prototipado en papel -> Una hoja para cada escenario.

9-Prototipado en papel:

- Ventajas:
 - Coste muy reducido.
 - Los cambios se pueden realizar muy rápidamente y sobre la marcha.
 - Los usuarios se sienten más cómodos para realizar críticas al diseño.
- Problemas:
 - Trabajar con prototipos requiere tiempo y experiencia en la planificación.
 - Las características más importantes del sistema se sacrifican en el prototipo (seguridad y fiabilidad)

Tema 4. Disseny d'interfícies Web

1-Errors Típics (10):

- “Enterrar” la información muy profundamente en la web
- Sobrecargar las páginas con demasiado material

- Crear una navegación confusa
- Poner información en lugares no esperados en la página
- No hacer links obvios y claros
- Presentar la información en tablas “no coherentes”
- Escribir el texto muy pequeño(dificulta la lectura)
- Hacer uso de combinaciones de colores para el texto que dificulte su lectura
- Utilizar formularios incorrectos
- Ocultar,o no dar, características-pistas que puedan ayudar a los usuarios

2-Introducción:

- La usabilidad universal en web es muy importante
- La presentación del contenido en una web se puede dividir en diferentes niveles:
 - Nivel de site
 - Estilos que se deben mantener a lo largo de todo el site y no solo en una página(uso de marcos,profundidad versus anchura del site)
 - Nivel de página
 - Componentes de las páginas individuales(tablas,gráficas,formularios,controles, diseño de página,presentación de links)
 - Nivel de información especial:
 - Como sitemaps, funciones de cerca , asistencia a el usuario,feedback..
- Hay muchas guías para diseñadores web para asegurar consistencia y adherencia a los estándares

3-Guia W3C

- El W3C tiene la iniciativa de accesibilidad Web(WAI) patrocinada por varias organizaciones.
- Las guías juegan un papel importante para crear webs accesibles.
- WAI ofrece 3 guías diferentes:
 - *Web Content Accessibility Guidelines (WCAG)*
 - Principios de diseño para sitios web accesibles.Estudian escenarios que pueden ocasionar problemas a usuarios discapacitados
 - *Authoring Tool Accessibility Guidelines (ATAG)*
 - Ayudan a los desarrolladores de herramientas de creación de contenido web para que estos sean accesibles
 - *User Agent Accessibility Guidelines (UAAG)*
 - Explican las características de las interfaces que

benefician a las personas con discapacidad(navegación por teclado,opciones de configuración,documentación,comunicación por voz..)

4-Usabilidad universal

- El mayor problema en la web es el tiempo de descarga
- El segundo mayor problema que hace que la gente no compre en una Web es la dificultad de navegación del site
- Hay que evitar que el usuario/cliente
 - Tenga problemas para encontrar el producto que quieran
 - Abandone un site por el bajo rendimiento,especialmente por una pobre navegación o descarga lenta
 - Abandone la búsqueda de un producto online
- Si se mejora la usabilidad,las ventas pueden mejorar
 - IBM incrementó ventas en un 400% después de rediseñar su site

5-Retos en el diseño de interfaces de usuario web

- Tiempo de descarga corto
- Opciones de interacción limitadas
- Amplia diversidad de usuarios/destinatarios
- Usuarios impacientes
- Muchas opciones en la aplicación
- Hardware y software de interacción diverso
- Múltiples lenguajes
- Usuarios discapacitados

6-Procés de disseny d'interfícies web centrat en l'usuari

- Marco metodológico para diseño web centrado en el usuario basado en el marco metodológico
 - Todo el proceso tiene que estar conducido por el usuario,sus necesidades, características y objetivos(involucrar a los usuarios en el proceso desde el inicio:como son,que necesitan,para que harán uso del web,testear el web con los usuarios,investigar cómo reaccionan con el diseño..)
- Lo que se diseña ha de estar evaluado constantemente haciendo uso del prototipo

6.1-Planificación

- Identificar objetivos del sitio web , y de los usuarios potenciales sus: necesidades,requisitos y objetivos
 - Objetivos técnicos (back-end,front-end)
 - Equipos de trabajo
 - Presupuesto

- Con entrevistas y reuniones se obtiene información sobre necesidades y objetivos del cliente. Además, hace falta obtener:
 - Como se comporta el usuario
 - Como sera el contexto de uso y cómo afecta a la interacción
 - Experiencia y conocimientos previos..como?
 - Indagando:Aproximación contextual,estudios de campo,métodos de aproximación..
- Fase de recogida,análisis, y ordenación de toda la información posible

6.2-Diseño

1. Modelado del usuario

- Toda la información analizada de los usuarios en planificación hay que resumir-la y sintetizar-la
 - Definir clases o perfiles de usuario según atributos comunes
 - El diseñador sabe para quien diseña,que espera del usuario y de qué forma
- Problema:Si audencia es muy extensa y heterogénea la categorización total NO es viable → Método “Persona”(Cooper 1999)

- Metodo Persona:
 - Definición de arquetipos de usuario que representan patrones de conducta,objetivos y necesidades.Son usuarios inventados pero con atributos, características y necesidades basadas en información real obtenida a partir de análisis.
 - Se debe definir “escenarios”
 - Sirve de soporte para tomar decisiones en el diseño centrándonos en el usuario “real”(tiene limitaciones, habilidades y necesidades reales)

6.3-Diseño Conceptual

- Objetivos:definir la organización, funcionamiento y navegación del sitio web
- No se centra en apariencia,si no en el concepto: arquitectura de información
- Se trata de crear:
 - La estructura del sitio (conexiones entre páginas)
 - La topología de la red de páginas
 - La navegación a las posibilidades y formas de cada página presenta las opciones de desplazamiento a otras páginas
- 2 formas de definir estructura:

- Descendiente (del “todo” a las partes)
- Ascendiente (De los bloques mínimos de información a las páginas)
- La estructura hace falta documentarla con **¿CON grafos?**, esquemas, diagramas, que hagan rápida su comprensión por parte de todo el equipo
- Una técnica de diseño conceptual es el “**Card Sorting**”:
 - Observar cómo los usuarios agrupan y asocian entre sí un número predeterminado de tarjetas etiquetadas con diferentes categorías temáticas del sitio web.

6.4-Diseño Visual y definición de estilos

- Fase especializada en aspectos visuales web:
 - Composición de cada tipo de página, aspecto, comportamiento de elementos de interacción, presentación de elementos multimedia.
- Evitar sobrecarga informativa, hace falta:
 - Considerar sitios con más importancia visual
 - Considerar espacio y tamaño de cada elemento
 - Utilización de contraste de color para discriminar y distribuir información
 - uso de efectos tipográficos para enfatizar contenidos,...
- Evitar sobrecarga memorística, hace falta:
 - Definir menús de navegación con un número de opciones reducido
- Accesibilidad:
 - Color: bastante contraste entre texto y fondo
 - Colores aptos para todos
 - Si hay imágenes: Resolución y tamaño adecuados
 - Uso de metáforas (papelera, escritorio, ...)
- Mantener consistencia visual en todo el sitio web
 - Útil crear un libro o guía para todo el equipo

6.5-Diseño de contenidos

- Diseñar contenidos interrelacionados y vinculados, manteniendo la coherencia informativa, comunicacional y organizativa.
- Hace falta ser precisos, creativo y estructurado a la hora de redactar y adaptar el lenguaje, tono y vocabulario al usuario objetivo.
- Consejos:
 - Seguir estructura piramidal
 - Permitir una exploración fácil
 - Un párrafo una idea
 - Ser conciso y preciso (no gusta leer en pantalla)
 - Tono familiar y cercano, pero no irrespetuoso

- Confianza: Permitir el diálogo y donar información del autor

6.6 Prototipos

Como resultado del diseño tendremos un prototipo que:

- En las primeras etapas en papel es barato y acelera el procés.
- Son reproducciones, no implementan la interfaz (se descartan).
- Aseguran que el sitio es **usable**.

6.7 Evaluación

Evaluar la usabilidad:

- Por expertos (heurística)
- Por usuarios finales
- Eres una miqueta borinotet/a*

Tema 5- Evaluación de sitios web

1-Introducció:

Avaluació: part bàsica d'un disseny d'un sistema centrat en l'usuari.

Sense avaluar, és impossible conèixer o no si un disseny o un sistema compleix les expectatives dels usuaris i si s'adapta al seu context social, físic i organitzatiu.

Usabilitat: mesura en la qual un producte pot ser usat per usuaris específics amb efectivitat, eficiència i satisfacció en un context d'us especificat.

2-Avaluació:

Def: diverses metodologies i tècniques que estudien la usabilitat d'un sistema interactiu en diferents etapes del cicle de vida.

Quan avaluar? Durant tot el cicle de vida del producte.

Aplicar mètodes d'avaluació de la usabilitat permet crear millors productes i ajudar als usuaris a realitzar tasques més productivament.

Avantatges:

1. Reducció dels costos de producció.
2. Reducció dels costos de manteniment i suport.
3. Reducció dels costos d'ús.
4. Millora en la qualitat de producte.

Aspectes a considerar:

1. Cost.
2. Persones que la faran.

3. Etapes del cicle de vida.

3-Mètodes d'avaluació de llocs web:

1. Avaluació heurística per experts avaluadors.
2. Test d'usuaris.
3. A través de simulació cognitiva o recorregut cognitiu (cognitive walkthrough). Variant: Simulació per equip multidisciplinari (pluralistic walkthrough).
4. De usabilitat intercultural.
5. De l'accessibilitat.

Alguns mètodes d'avaluació poc adequats per avaluar la usabilitat són els estudis de mercat i els estudis automatitzats, ja que tenen seriosos inconvenients.

Per a avaluar llocs web podem trobar diferents mètodes, no obstant, l'estratègia més aconsellable es combinar una avaluació heurística amb un test d'usuaris posterior.

4-Test d'usuari:

Def: Prova d'usabilitat basada en l'observació i anàlisi de com un grup d'usuaris reals utilitza el lloc web, anotant els problemes d'ús amb que aquests es troben per poder solucionar-los posteriorment.

Duta a terme en 'laboratori', no podem confondre-la amb un estudi de camp.

És la manera més propera d'aproximar-se a l'ús real d'aquest.

Primordial en l'anàlisi d'un lloc web. Si es realitza correctament, complementa perfectament a l'avaluació heurística.

Tipus de test d'usuari:

- Test d'usuaris formal:
 - Comptar amb un local (laboratori) adequat.
 - Comptar amb la infraestructura òptima.
 - Contractar avaluadors especialitzats.
 - Delegar en alguna empresa la selecció i reclutament dels participants de la prova.
 - Bastant costós i poc viable per a la gran majoria de casos.
- Proves informals o 'tests de guerrilla':
 - Fer les proves en el client.
 - Ús de llapis i paper.
 - Només un avaluador especialitzat.
 - Reclutar usuaris voluntaris: coneguts, familiars...
 - Fer la prova en diverses fases (un usuari, un avaluador).
 - És més econòmic.
- Tests d'usuaris en remot:
 - Realitzar una investigació online sobre una àmplia mostra d'usuaris que participen en una prova o test des del seu

- context real (casa o oficina).
- Exemple test A/B.
- Avantatges:
 - Informació rica i detallada.
 - Participació des d'entorn real.
 - Mostres geogràficament disperses.
 - Complementa altres mètodes.
 - Capacitat de segmentar resultats.
 - Major rapidesa de resultats.
 - Major eficiència de costos.

Prova de test d'usuari en remot - Test A/B:

- Problema:
 - Els usuaris són complexes i la nostra intuïció moltes vegades s'equivoca.
 - Desplegar una característica, a tots els usuaris a la vegada, és perillós.
- Propòsit test A/B:
 - Conèixer el que els usuaris volen inconscientment o d'una altra manera.
 - Ajuda a fallar ràpid i seguir endavant.

Les proves de cinc segons són senzilles proves d'usabilitat que l'ajuden a mesurar l'eficàcia dels dissenys.

Realitzar una prova de cinc segons us permet esbrinar quines parts dels dissenys són els més destacats. Podeu fer-ho de dues maneres:

Prova de memòria

Dóna als usuaris cinc segons per mirar el disseny i després demanar-los que recordin elements específics.

Feu clic a Prova

Doneu als usuaris cinc segons per localitzar i fer clic en elements específics del vostre disseny.

Tot el que necessites fer és carregar un disseny que vulguis provar i triar el tipus de prova que vols executar. Se't donarà un enllaç únic que podràs compartir amb amics i companys i fer que facin la teva prova. Per descomptat, sempre podeu deixar que els usuaris aleatoris facin la vostra prova per rebre comentaris realment ràpids.

Quan realitzar un test d'usuari?

“Com més tard, pitjor.”

- A causa del cost elevat de reparar els errors trobats.
- Millor fer-ho des de fases de prototipatge.

“Sempre, després d'una avaluació heurística.”

- Ja que elimina els errors més bàsics.

Per què són realment útils?

“És una demostració amb fets”

- Millor tècnica per acabar amb discussions entre l’equip de desenvolupament sobre quina decisió de disseny és la més adequada.

“Cada lloc web és un món”

- Les característiques i peculiaritats de cada disseny, així com de cada audiència, faran necessària una avaluació “específica” que assegure la usabilitat del lloc.

Cóm fer un test d’usuari?

- Reclutar als participants.
 - El nom òptim de participants en la prova ha de ser almenys de 5.
 - Es pot comptar amb amics, familiars i persones no relacionades amb el projecte.
 - Els usuaris han de tenir perfils acords amb la naturalesa del projecte.
- Aconseguir local i materials.
 - Aconseguir un espai on realitzar les proves (on ningú moleste).
 - Bloc de notes per fer el registre de les accions de l’usuari.
 - Càmera per registrar la navegació i la cara de l’usuari.
 - Programari de registre de les accions de l’usuari.
- Planificació de la prova.
 - Planificació previa de les accions a realitzar:
 - Selecció de les tasques a realitzar.
 - Descripció d’escenaris.
 - Redacció de narratives per a la prova.
 - Elaboració d’un guió de treball:
 - Què se li va a dir a cada usuari.
 - Què se li va a demanar que faci.
 - Com ha de fer-ho.
 - Quánt de temps té per a realitzar la prova.
 - Acord de confidencialitat i privacitat.
- Abans de començar.
 - Establir un ambient amigable i confortable.
 - Explicar molt bé l’objectiu de la prova: no s’avalua al participant.
 - Explicar que si es comet algun error, no és culpa de l’usuari, és culpa del disseny.
 - Convidar l’usuari a pensar en veu alta mentre realitza les tasques.
 - L’usuari ha de saber que l’avaluador no l’ajudarà a realitzar les tasques.

- No s'ha d'explicar el lloc web a l'usuari: aquest ha de ser entès de forma de forma autònoma per aquell.
- Desenvolupar la prova.
 - L'avaluador obri el navegador amb el lloc web carregat.
 - Es demana a l'usuari que observe la interfície i diga en quin lloc web està i quina és la seua funció i objectiu.
 - Fer atenció al que diga l'usuari, a les seues expressions i als seus gestos.
 - üs de “aha” per a respondre a l'usuari.
 - Si l'usuari s'encalla en una tasca, se li demanarà que iniciï la següent i deixi l'anterior.
 - Els usuaris ens indiquen problemes de disseny, no la seva solució.
- Elaboració d'un informe al finalitzar.
 - Registrar les tasques realitzades pels usuaris.
 - Registrar els resultats obtinguts pels usuaris en la realització de tasques.
 - Registrar els problemes d'usabilitat detectats en la realització de les tasques per part dels usuaris.
 - Reflectir algunes solucions o indicacions per solucionar els problemes detectats.

5-Avaluació heurística:

Def: Avaluació de la interfície, per un conjunt reduït d'avaluadors (mínim 5 per detectar el 75%), tenint en compte els principis reconeguts d'usabilitat heurística.

S'utilitzen experts per validar la interfície, ja que és difícil que un avaluador pugui trobar tots els problemes d'usabilitat en una interfície a partir d'uns criteris definits.

Guia, en forma de checklist, per a l'avaluació dels principis d'usabilitat d'una web sobre la base de dimensions com:

- Identitat.
- Llenguatge i redacció.
- Accessibilitat.
- Layout.
- Elements multimèdia.

Molts experts es basen en les 8 regles d'or del disseny d'interfícies per avaluar-les.

Cada avaluador realitza individualment una revisió de la interfície.

En acabar les avaluacions, es permet als avaluadors comunicar els

resultats i sintetitzar-los.

Aquest procediment és important per assegurar avaluacions independents i imparcials de cada avaluador.

Els resultats de l'avaluació es poden registrar amb:

- Informes escrits de cada avaluador.
- Els avaluadors comuniquen verbalment els seus comentaris a un observador mentre realitzen l'avaluació.

10 regles heurístiques:

1. L'estat del sistema ha d'estar sempre visible.
2. S'ha d'utilitzar el llenguatge dels usuaris.
3. L'usuari té control i llibertat.
4. Hi ha consistència i es segueixen estàndards.
5. Hi ha prevenció d'errors.
6. Es minimitza la càrrega de la memòria de l'usuari.
7. Hi ha flexibilitat i eficiència d'ús.
8. Els diàlegs són estètics i de disseny minimalista.
9. Amb l'ajuda, es reconeixen, diagnostiquen i es recuperen.
10. Hi ha ajuda i documentació.

Criteris per a l'avaluació heurística:

- Generals.
- Identitat i informació.
- Llenguatge i redacció.
- Retolat.
- Estructura i navegació.
- Layout de la pàgina.
- Cerca (si cal).
- Elements multimèdia.
- Ajuda.
- Accessibilitat.
- Control i retroalimentació.

Segons estudis, l'experiència dels avaluadors influeix la quantitat de problemes trobats, de manera que quant més nivell, menys avaluadors fan falta per trobar errors.

Tipus d'avaluadors:

- Novells: sense coneixement d'usabilitat ni del domini.
- Experts: en usabilitat, però no del domini.
- Especialistes: en usabilitat i amb coneixement del domini.

Trobar errors requereix combinar avaluadors de diferents tipus.

Avantatges:

- Econòmic.
- Intuïtiu.
- Utilitzable en les primeres fases del procés de desenvolupament.

- Pot fer-se en qualsevol sistema i fins i tot remotament.
- Troba molts problemes.
- Troba tant problemes majors com menors.

Desavantatges:

- Pot ignorar problemes específics del domini.
- No es recullen dades quantitatives.

6-Anàlisi i interpretació dels resultats:

- Elaborar **fitxes d'activitat**: registrar totes les accions dels usuaris, per escrit, en realitzar les tasques.
- Elaborar **matriu de tasques i accions**: que reflecteixi les tasques completades i les que no.
- Elaborar **matriu d'incidències**: incidències en cada pas de les tasques realitzades.
- Llistat de **conclusions** agrupades per tasques.
- **Matriu de propostes** de solució.

7-Guió de tasques:

- Es divideix en conjunts de preguntes i series de tasques, cadascuna vinculada a objectius. Contingut mínim:
 - Introducció
 - Qüestionari previ al test
 - Descripció d'un escenari
 - Tasques
 - Preguntes
 - Qüestionari posterior al test

8-Errors freqüents en realitzar un test d'usuaris:

- Comparacions entre versions de llocs web:
 - Una comparació pot ser útil per a convèncer a un client dels beneficis de la usabilitat, però no es el mes adequat per a millorar la usabilitat del lloc
 - Comparar es malgastar recursos
 - No te sentit testar la antiga versió del lloc per a tindre una referència en la comparació, la antiga sol tindre errors y funcionalitats diferents el que fa que la comparació no aporte informació
 - Comparar és inadequat desde el punt de vista de la validesa dels resultats, en la majoria dels casos es comparen llocs molt diferents.
 - Comparar com a metode per a obtindre conclusions té la seua fortaleza quan els llocs son identics y es diferencien sols de la característica que estem comparant (ineficient porque crear dos llocs iguals on canvie una característica es llarg i

- costós)
- **Unic us adequat de la comparacio com a ferramenta de la usabilitat es durant la creació de prototips de baix cost**
- El mes aconsellable es definir la creació d'un lloc web com un procés iteratiu on sols es deu seguir un camí, es a dir, enfocar tots els recursos a desenvolupar una unica versió que es va testejant i refinant.
- Manca d'una avaluació heurística prèvia
 - Un professional pot realitzar la evaluació de manera rapida i detectar a baix cost els errors de la usabilitat
 - Un lloc pot estar plagat d'errors i en el test d'usuaris es detecten sols errors de bulto en el diseny
 - En un formulari de registre un professional pot detectar molts problemes de usabilitat rapidament i un usuari pot quedar-se bloquejat en un pas i no es podrà evaluar completament
- Ús inadequat de qüestionaris
 - Preguntes totalment inadequades (Preguntes sesgades que no aporten informació valiosa i es una herencia dels estudis de mercat):
 - Li ha paregut humà el lloc web?
 - Li ha paregut bonic el lloc web?
 - En internet el procesament d'informació esta per damunt dels aspectes emocionals. En internet es mes important "fer" que "sentir" (ej:que et sembla una imatge es una pregunta escasa)
- Poca definició de tasques i mesures
 - Un video es apropiat per a confirmar els resultats o ilustrarlos graphicament, però el anàlisi de un video requereix molt de temps i no es pot considerar una base del test, sino sols un complement.
 - Per obtindre mesures fiables es deu demanar als usuaris diferents tasques que simulen un comportament real
 - Mesures objectives d'usabilitat son la tasa de finalització de la tasca, el temps, el nombre de errors, la memòria, etc.

9-Mesures de prestacions:

Cóm mesurar la usabilitat?

- Considerarem cómo planificar les observacions i mesures per a un test d'usabilitat.
- En primer lloc tractarem de comprendre què és el que es pot mesurar:
 - Mesures de rendiment: comptar les accions i els comportaments que es pugen veure.
 - Son quantitatives, es poden comptar quantes persones hi ha, quants errors fan, quantes vegades es repeteixen els mateixos errors.

- La major part de les mesures de rendiment requereixen observacions acurades.
- Mesures subjectives: percepcions de les persones, opinions i judicis.
 - Poden ser quantitatives o qualitatives.
 - Preferències o raons de la preferència:
 - D'una versió prèvia.
 - Sobre un producte de la competència.
 - De la manera com estem fent les tasques ara.
 - Prediccions de comportament de l'usuari.

Mesurar una tasca:

Criteris que podem definir per a mesures de prestacions:

- Excel·lent.
- Acceptable.
- Inacceptable.

Resultats del test:

Un test de prestacions genera una quantitat important de dades:

- L'objectiu és trobar problemes reals amb el producte i amb el procés que s'ha d'utilitzar per desenvolupar el producte.
- Una llista de problemes que han anat creixent durant la realització del test.
- Dades quantitatives de temps, errors i altres mesures de rendiment.
- Dades quantitatives de valoracions subjectives i altres qüestions de qüestionaris post-tasca i post-test.
- Comentaris dels participants dels enregistraments i dels qüestionaris.
- Ye, no seràs tu un màquina llegint-ho tot*
- Les notes escrites de l'equip de test o els seus comentaris que poden ser gravats.
- Dades generals dels participants, dels seus perfils o de qüestionaris de pre-test.
- Els enregistraments de vídeo, presentant diferents vistes del test.
- Vale, si te creus que vas a poder aprobar per estar llegint hasta ací, vol dir que no tens ni idea del temari, ale va, tira cap a dalt i torna a començar tira...*