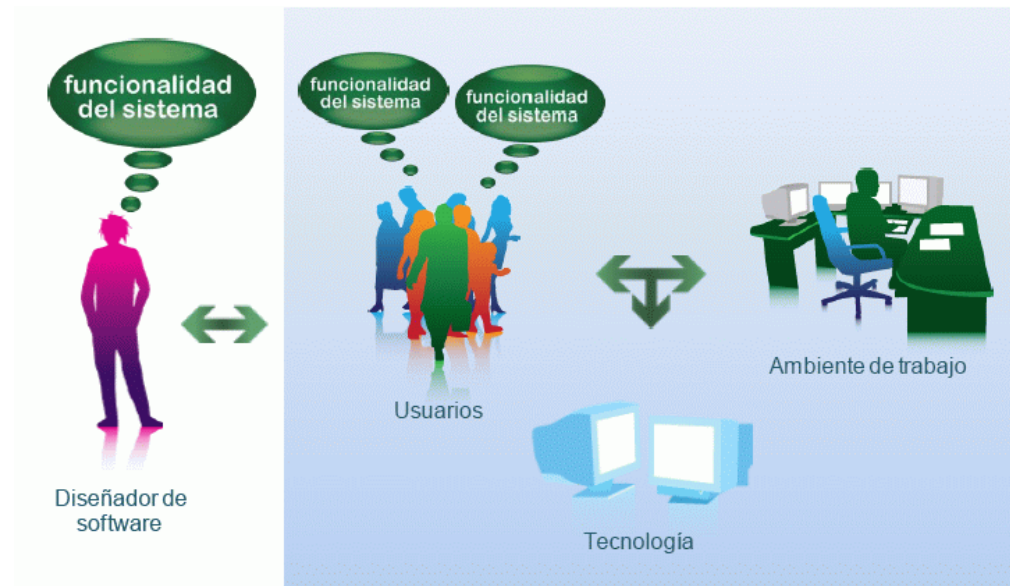

MODIHC

(Modelo para el diseño de la
IHC)

Abraham Sánchez López
Grupo MOVIS
FCC/BUAP

Formalización de MODIHC

- MODIHC es un modelo que permite diseñar todos los aspectos involucrados en la interacción entre un humano y una computadora cuando se están desarrollando productos de software.
- El modelo consta de cuatro componentes principales: usuarios, ambiente de trabajo, tecnología y funcionalidad del sistema, los cuales están relacionados entre sí y tienen como coordinador al diseñador.



Componentes de MODIHC

- El modelo está basado en la filosofía del DCU (diseño centrado en el usuario), así los usuarios cuentan con un papel principal en todo el proceso de diseño.
- El diseñador de software debe realizar la transición de la funcionalidad a la usabilidad, lo cual significa que el funcionamiento de un producto de software debe corresponder a los requisitos y habilidades del usuario en su ambiente natural de trabajo.
- Se proponen una serie de actividades que debe realizar el diseñador de software, las cuales están basadas en el proceso de diseño centrado en el ser humano.

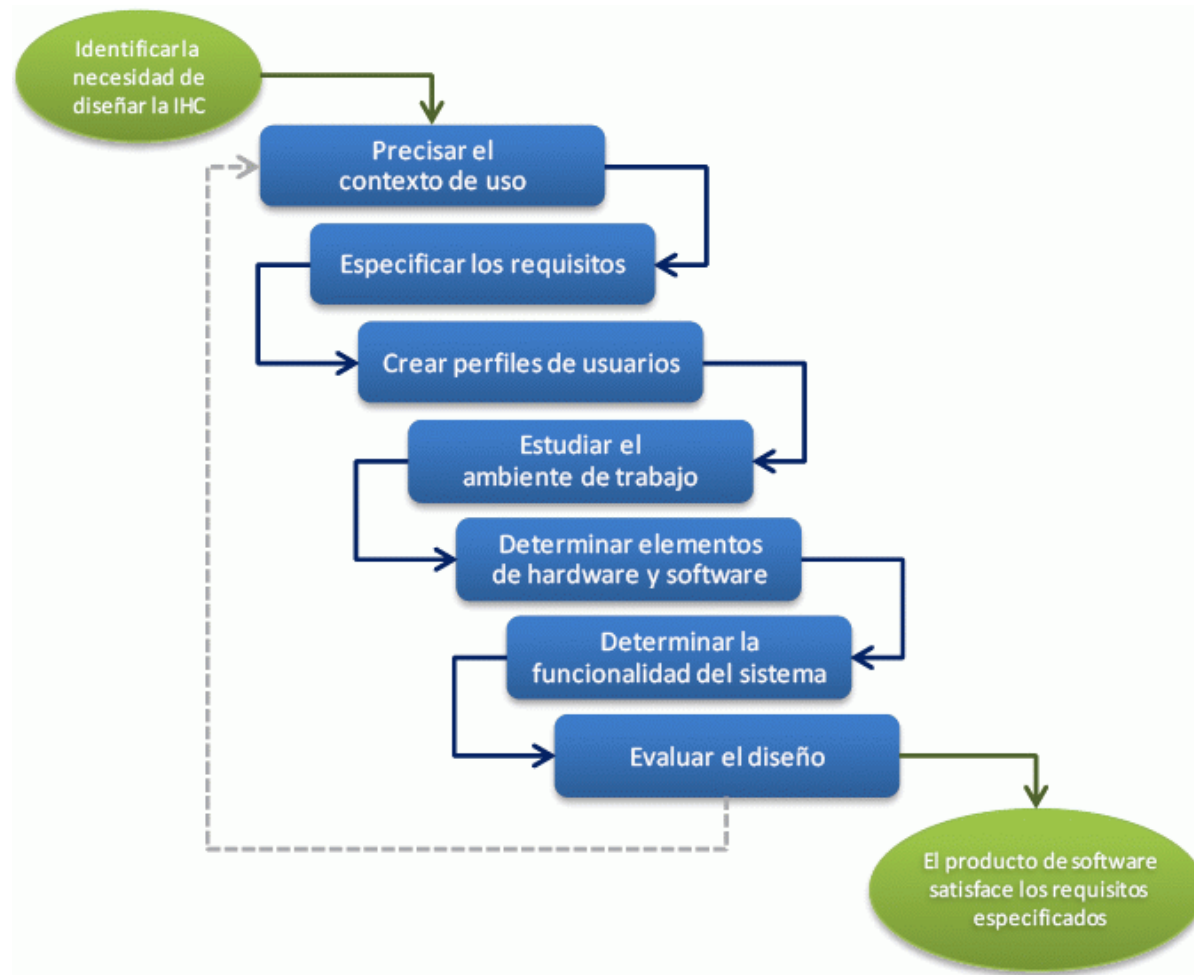
Actividades del diseñador de software

Componente de MODIHC		Actividad del diseñador de software
<div> <div>Usuarios</div> <div>Ambiente de trabajo</div> <div>Tecnología</div> <div>Funcionalidad del sistema</div> </div>		<p>Precisar el contexto de uso: Identificar a todos los usuarios del producto de software y clasificarlos según sus características particulares, conocer para qué y bajo qué condiciones usarán el producto de software.</p>
		<p>Especificar los requisitos: Identificar las necesidades de la organización y las metas de los usuarios que deben cumplirse para que el producto sea exitoso.</p>
		<p>Estudiar el ambiente de trabajo: Tomar en cuenta el ambiente en el cual se usará el sistema, tales como espacio y luz, aspectos sociales y organizacionales.</p>
		<p>Determinar elementos de hardware: Seleccionar los dispositivos de entrada/salida apropiados para la ejecución de las tareas del producto de software.</p>
		<p>Determinar elementos de software: Seleccionar cuál es el mejor estilo de interacción para diseñar la interfaz de usuario.</p>
		<p>Determinar la funcionalidad del sistema: Conocer el modelo mental de cada uno de los usuarios, mezclarlo con el modelo mental propio y diseñar la interfaz de usuario.</p>
		<p>Evaluar el diseño: Contrastar el diseño con los requisitos.</p>

Proceso para aplicar MODIHC I

- La tabla anterior muestra que el componente usuarios está presente en todas las actividades que realiza el diseñador de software.
- Para diseñar la IHC desde el punto de vista de la tecnología debe considerarse el ambiente de trabajo y los productos obtenidos del estudio del ambiente de trabajo y la selección de los elementos de hardware y software que influyen en el diseño de la funcionalidad del sistema.
- Entonces podemos proponer un proceso para aplicar MODIHC, el cual está basado en las actividades mencionadas con anterioridad.
- Mediante el proceso de diseño y desarrollo interactivo, tal como lo propone MODIHC, con la participación de los usuarios reales desde las etapas iniciales del proceso, se asegura la creación de productos de software fáciles de aprender y usar por los usuarios (usables), que son seguros, efectivos y facilitan las actividades que los usuarios quieren realizar.

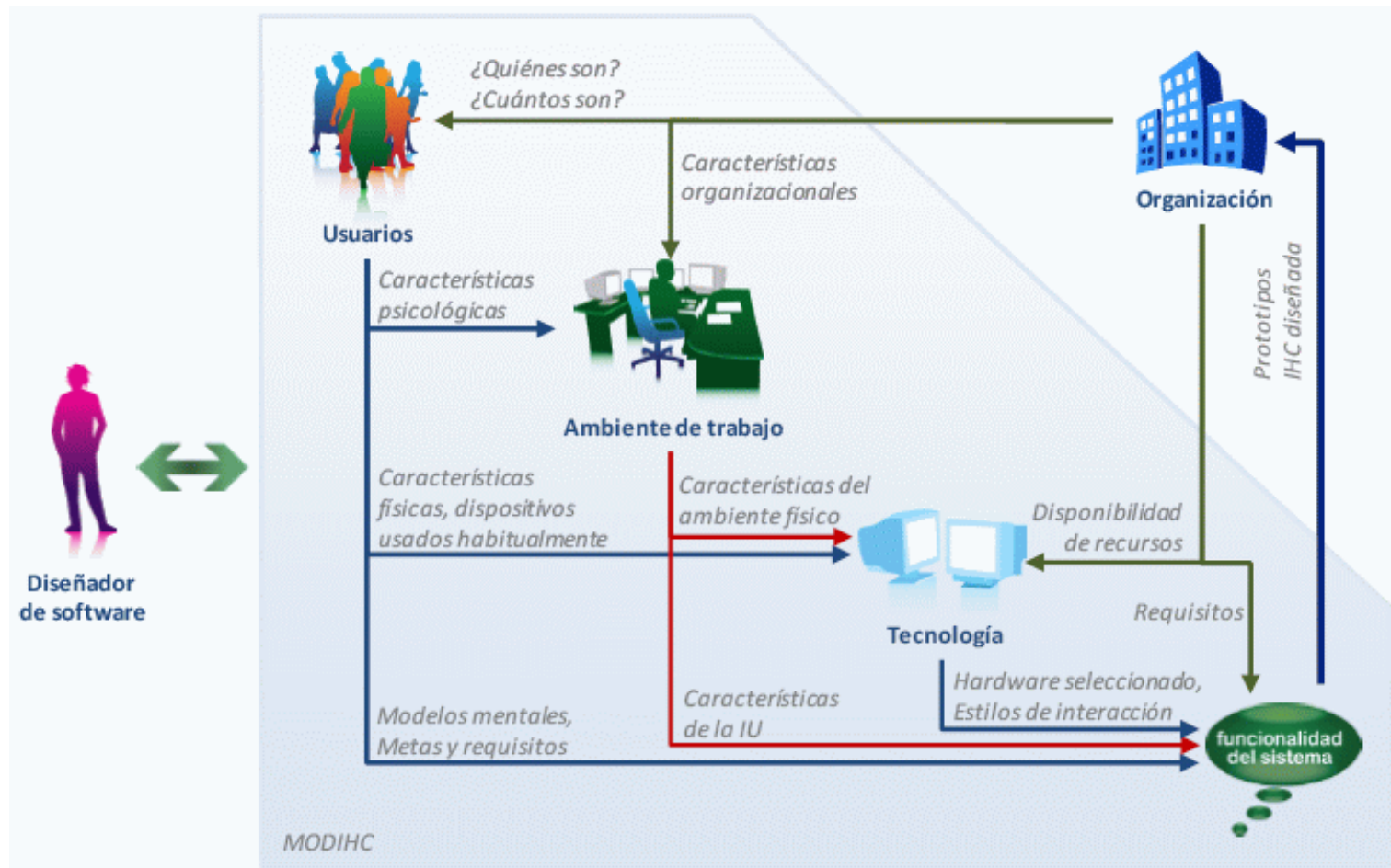
Proceso para aplicar MODIHC II



Relación de los componentes de MODIHC

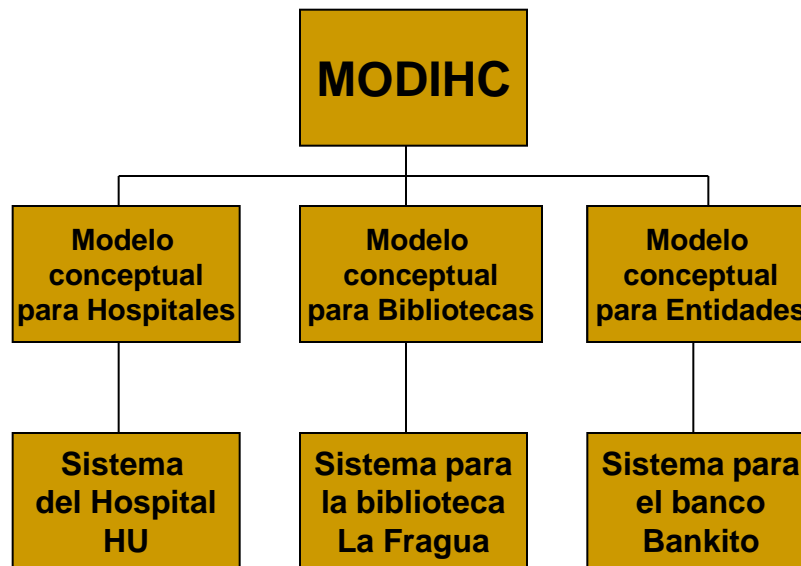
- En la relación entre los 4 componentes de MODIHC, se especifica la información que recibe como entrada cada componente y la que genera como salida.
- También se aprecia el elemento organización, este representa el ente que necesita o requiere el diseño de IHC del producto de software.
- La organización aporta la información inicial sobre el producto de software a desarrollar y datos para cada componente, como por ejemplo quiénes y cuántos son los usuarios del producto de software, cuáles son las características organizacionales, cuáles son los recursos disponibles y cuáles son los requisitos organizacionales del producto de software.
- El proceso para el diseño de la IHC funciona de manera similar al modelo en cascada, empezando por el componente usuarios, estudiando luego el ambiente de trabajo, definiendo posteriormente la tecnología a usar, culminando con el diseño de la funcionalidad del sistema.
- De este último componente surgen los prototipos del sistema que serán evaluados y el diseño de la IHC final que se entrega a la empresa.

Relación componentes y la organización



Importancia de MODIHC

- La importancia radica en que es genérico, dado que puede ser aplicado para el diseño de la IHC de los productos de software de cualquier organización, sin importar las características específicas de la misma ni de sus aplicaciones.
- MODIHC representa el nivel más alto de abstracción.



Especialistas para MODIHC

- Dependiendo de la naturaleza del producto de software a desarrollar, el diseñador de software puede solicitar la ayuda de profesionales en diferentes áreas para realizar sus actividades.

Especialista	Descripción	Actividades en la que intervienen
Psicólogo	Persona dotada de especial sensibilidad para el conocimiento del comportamiento humano.	<ul style="list-style-type: none">• Precisar el contexto de uso.• Crear el perfil de usuarios.• Identificar metas de los usuarios.
Estadístico	Especialista en el estudio de los datos cuantitativos de una población o de cualquier otra manifestación de las sociedades humanas.	<ul style="list-style-type: none">• Aplicar técnicas de recolección de datos.• Analizar datos recolectados de los usuarios.
Diseñador gráfico	Especialista en idear y proyectar mensajes visuales, contemplando necesidades específicas.	<ul style="list-style-type: none">• Establecer colores, formas y tamaños en la IU.
Sociólogo	Persona con especiales conocimientos en el funcionamiento de las sociedades humanas.	<ul style="list-style-type: none">• Estudio del ambiente social de trabajo.
Ergonomista	Especialista en el estudio de la adaptación de las máquinas, muebles y utensilios a la persona que los emplea habitualmente, para lograr una mayor comodidad y eficacia.	<ul style="list-style-type: none">• Estudio del ambiente físico de trabajo.• Selección del hardware adecuado.• Localización de equipos y material de oficina.
Ingeniero de software	Especialista en creación de software de calidad.	<ul style="list-style-type: none">• Determinar la funcionalidad del sistema.

Usuarios

- Para la IHC, entender el aspecto físico, intelectual y la personalidad de los diferentes usuarios es un factor fundamental.
- Conocer quiénes y cómo usaran el producto de software permitirá generar un diseño que posteriormente se traducirá en un sistema en operación eficiente y usable.
- Algunos factores relacionados con los usuarios determinan la selección del contexto, las interfaces de usuario y los dispositivos de E/S.
- La construcción del perfil de los usuarios es base para el diseño de productos de software eficientes.
- Involucrar desde el inicio a los usuarios en el diseño de un producto de software es esencial para que su desarrollo sea exitoso.
- La tarea de los diseñadores de software es diseñar el sistema en base a los requisitos de los usuarios, sin hacer especulaciones, para satisfacer las necesidades individuales de cada usuario y además satisfacer a una gran variedad de usuarios.

Relación diseñador de software-usuarios



Objetivos del componente usuarios

- Los objetivos que el diseñador de software debe alcanzar en este componente son los siguientes:
 - Identificar a todos los usuarios del producto de software.
 - Clasificar a los usuarios según sus características.
 - Conocer para qué y bajo qué condiciones se usará el producto de software.
 - Crear el perfil de usuarios.
 - Determinar características de la IU basadas en los usuarios.
 - Crear el documento de especificación de requisitos.
- Los productos que se deben obtener de este componente son el perfil de usuarios, el documento de especificación de requisitos y las características de la IU basadas en los usuarios.
- El perfil de usuarios es un documento donde se identifica y clasifica a todos los usuarios del producto de software.
- Se detalla igualmente para cada tipo de usuario, un conjunto de rasgos distintivos que lo caracterizan y que influyen de alguna manera en el estudio del ambiente de trabajo, dispositivos de E/S, etc.

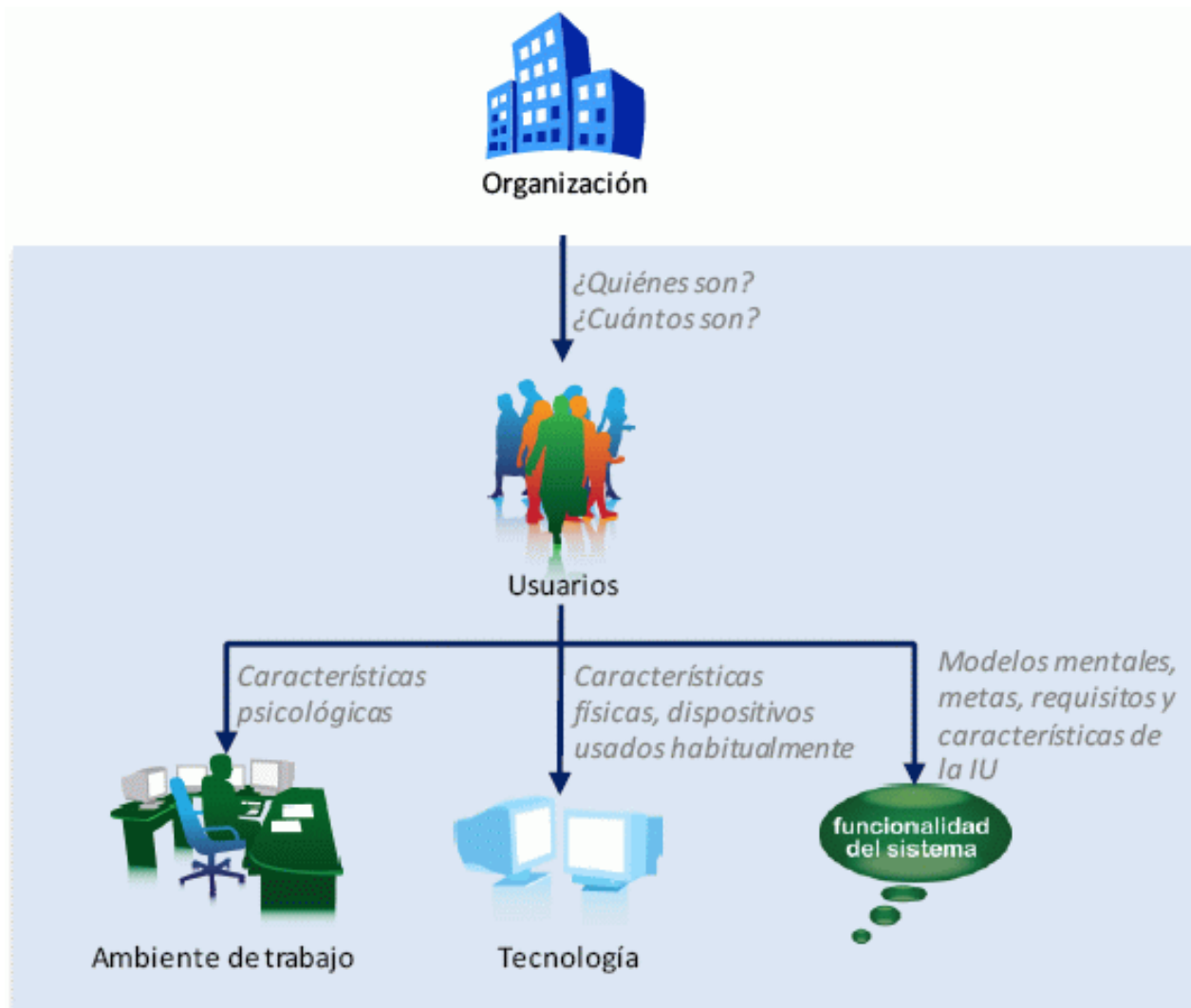
Perfil de usuarios

Elemento	Descripción	Importancia
Datos básicos	Se identifica el tipo de usuario, cargo y limitaciones en el uso del producto de software.	Permiten reconocer y clasificar a los usuarios.
Características físicas	Comprende variables como edad, sexo y limitaciones físicas.	Ayudan a localizar los equipos de oficina, establecer los estilos de interacción, definir los dispositivos de entrada/salida y establecer características de la IU.
Características psicológicas	Comprende el nivel de educación y conocimiento, carácter, habilidades y experiencia de los usuarios.	Ayudan a definir el ambiente social de trabajo y a establecer un diseño gráfico de la IU adaptada al usuario.
Dispositivos comúnmente usados	Elementos de hardware que los usuarios están acostumbrados a usar.	Permiten determinar dispositivos de entrada/salida que le resulten familiar a los usuarios.
Modelo mental del sistema	Breve explicación hecha por los usuarios de cómo funciona el sistema en el mundo real.	Permite predecir los tiempos de aprendizaje del producto de software, los tipos de errores y el grado de dificultad con el cual los usuarios pueden realizar sus tareas. Ayuda a obtener la IU apropiada.
Metas	Objetivos que los usuarios deben cumplir en el desempeño de sus funciones dentro de la organización.	Ayudan a diseñar la funcionalidad del sistema.
Requisitos	Necesidades de los usuarios sobre el contenido, forma o funcionalidad del producto de software.	Son base para el diseño de la funcionalidad del sistema. Deben satisfacerse para que el producto de software se considere útil.

Usuarios y las demás componentes

- La información contenida en el perfil de usuarios, el documento de especificación de requisitos y las características de la IU alimentan al resto de los componentes de MODIHC.
- Las características psicológicas de los usuarios se consideran para el estudio del ambiente social de trabajo.
- Las características físicas y los dispositivos que los usuarios están acostumbrados a utilizar son base para la selección de los equipos de E/S y los estilos de interacción.
- Los modelos mentales, las metas de los usuarios, los requisitos del producto de software y las características de la IU son usados para diseñar la funcionalidad del sistema.
- La organización aporta al diseñador de software los datos iniciales para el proceso de recolección de datos, como por ejemplo cuántos y quiénes son los usuarios.

Relación de los usuarios con los demás



Metodologías para el componente usuarios

- Para obtener los datos necesarios para la creación del perfil de usuarios, el documento de especificación de requisitos y las características de la IU, se deben aplicar una o varias técnicas de recolección de datos.
- El objetivo de estas técnicas es recolectar datos suficientes, relevantes y apropiados que permitan no sólo conocer las características clave de los usuarios, sino también cuál es el su modelo mental del sistema, qué características consideran que deben tener los dispositivos de E/S, los estilos de interacción y el ambiente del trabajo.
- Según Preece, se enumeran cuatro elementos que deben considerarse para realizar una recolección de datos exitosa:
 - **Las metas:** Es importante definir metas específicas de las actividades a realizar, las técnicas que serán usadas y el tipo de análisis que será realizado. Una vez establecidas las metas se definen qué datos buscar y qué hacer con ellos una vez que se tengan.

Más de la recolección de datos

- ❑ **La relación con los participantes:** La relación entre el diseñador de software y los usuarios debe ser clara y profesional. Los usuarios deben conocer el motivo del estudio y qué se hará con la información que suministren.
- ❑ **La triangulación:** Es una estrategia que implica usar más de una técnica de recolección de datos para abordar una meta, o usar más de un enfoque de análisis sobre los mismos datos. La triangulación proporciona perspectivas diferentes y corrobora los resultados de las diferentes técnicas usadas.
- ❑ **El estudio piloto:** Es una prueba del funcionamiento del estudio principal. Su objetivo es asegurar que el método propuesto para recolectar datos es viable antes de realizar el estudio real.
- Las técnicas de recolección de datos mas usadas son la entrevista, las sesiones de grupo, los cuestionarios y las observaciones directa e indirecta.
- La entrevista se utiliza para recabar información de forma verbal, a través de conversaciones entre el diseñador de software y cada uno de los usuarios.

Técnicas de recolección de datos

- Las sesiones de grupo son un tipo de entrevistas donde el diseñador de software realiza preguntas a un grupo de usuarios al mismo tiempo.
- Los cuestionarios son un conjunto de preguntas, formuladas por el diseñador de software, a las que los usuarios pueden responder oralmente o por escrito.
- La observación es una técnica que puede ser usada en cualquier fase del desarrollo de un producto de software.
- La elección de las técnicas de recolección de datos depende de varios factores, entre los que se encuentran factores relativos a los usuarios y al diseñador de software (ubicación y tiempo disponible), así como de los recursos disponibles por la organización para realiza el estudio.
- Es común combinar varias técnicas de recolección de datos con el fin de triangular los resultados.

Descripción y uso de la recolección de datos

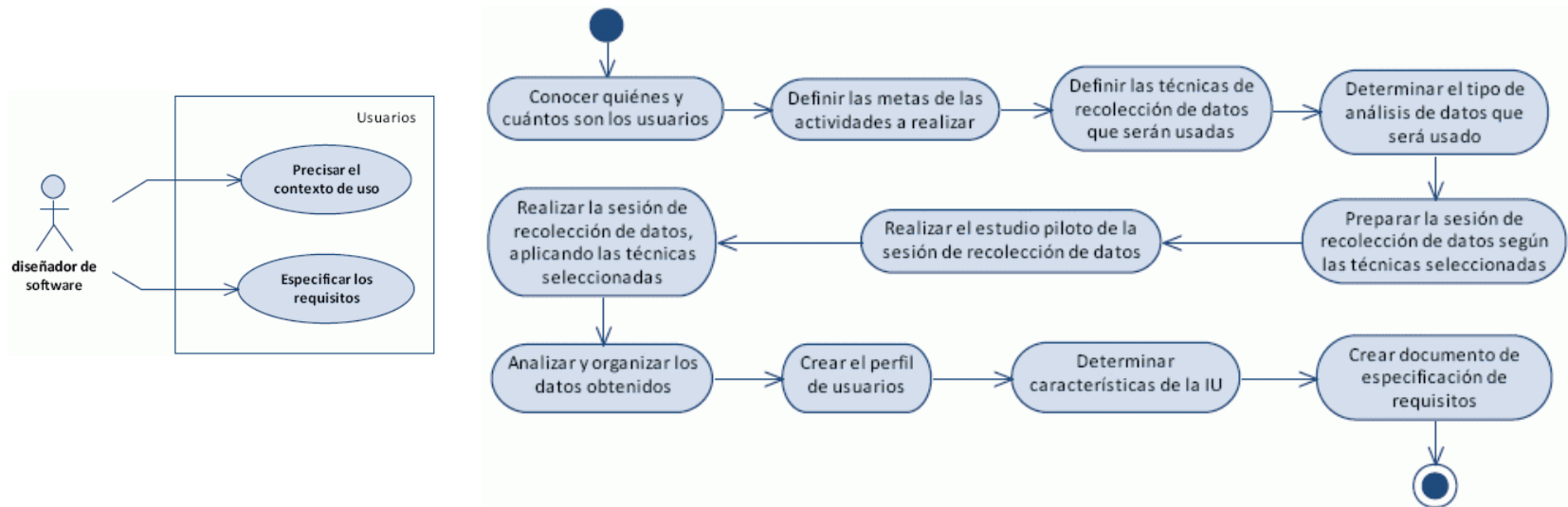
Técnica	Utilidad	Virtudes	Defectos
Entrevistas	Explorar temas.	El entrevistador puede guiar al entrevistado de ser necesario. Estimula el contacto entre los desarrolladores y los usuarios.	Consume tiempo. Puede intimidarse al entrevistado.
Sesiones de grupo	Recolectar varios puntos de vista.	Resalta las áreas de consenso y conflicto. Estimula el contacto entre los desarrolladores y los usuarios.	Posibilidad de existir personajes dominantes.
Cuestionarios	Responder preguntas específicas.	Con pocos recursos puede llegar a muchas personas.	El diseño es crucial. La tasa de respuesta puede ser baja. Las respuestas podrían ser inesperadas.
Observación directa en el campo	Entender actividades del contexto de uso.	Observar el trabajo real proporciona detalles que otras técnicas no hacen.	Consume mucho tiempo. Se recolectan grandes cantidades de datos.
Observación directa en ambiente controlado	Aprender sobre procedimientos, reglamentos y estándares.	Puede centrarse en la tarea sin interrupciones.	Los resultados pueden tener un uso limitado en el ambiente real.
Observación indirecta	Observar a los usuarios sin interrumpir sus actividades; los datos se capturan de forma automática.	Los usuarios no se distraen por la recolección de datos.	Se necesitan herramientas para analizar grandes cantidades de datos cuantitativos.

Diseño de la interfaz de usuario

Característica	Descripción	Valor observado	Recomendación para el diseño de la IU
Edad	Edad promedio de los usuarios del producto de software.	Niños (hasta los 12 años)	Usar en mayor cantidad imágenes e íconos. La cantidad de texto debe ser baja.
		Jóvenes-Adultos (de los 12 a los 60 años)	Soporta mayor cantidad de texto. El uso de imagen y texto es relativamente igual.
		Tercera edad (60 años o más)	Usar poco texto y tipografías de mayor tamaño. Definir instrucciones precisas. La interfaz debe ser intuitiva.
Habilidad en el uso de dispositivos	Conocimiento y experiencia de los usuarios en el uso de hardware y software.	Baja (Poco conocimiento sobre el uso de dispositivos)	Deben presentarse instrucciones claras sobre la funcionalidad. La interfaz debe ser intuitiva.
		Media (Los usuarios se encuentran familiarizados con los dispositivos)	Serán necesarias, como mínimo, las instrucciones de las actividades de mayor importancia que se realizan con el producto de software.
		Alta (Poseen alto grado de experiencia en el uso y manejo de dispositivos)	Las instrucciones sobre el uso de los dispositivos deben considerarse en segundo plano, de esta manera no se pierde tiempo mostrando cosas que ya saben.
Limitaciones físicas y/o psicomotoras	Usuarios con impedimentos en sus funciones intelectuales o físicas	Presentes (uno o más usuarios poseen)	La interfaz debe compensar las limitaciones encontradas, siempre que sea posible.
		Ausentes (ningún usuario posee)	Debe diseñarse la interfaz de tal manera que, de presentarse en el futuro cambios referentes a este tópico, pueda ser adaptada sin dificultad.

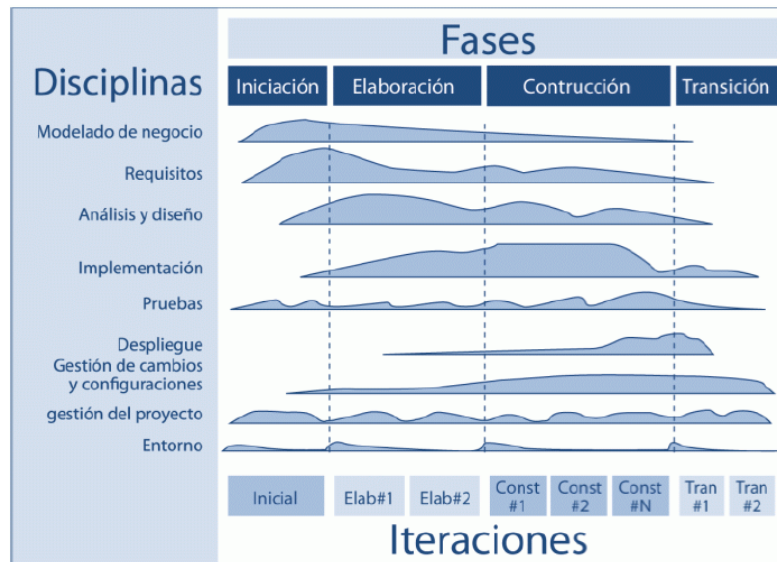
Flujo de trabajo del componente usuarios

- Con el objeto de formalizar el componente usuarios de MODIHC, se proponen ciertos pasos, los cuales deber seguirse por el diseñador de software para lograr los objetivos del componente.
- En un diagrama de casos de uso, se muestran las actividades del diseñador de software dentro del componente usuarios, y en un diagrama de actividades se describen los pasos a seguir para realizar dichas actividades.



Integración con la IS

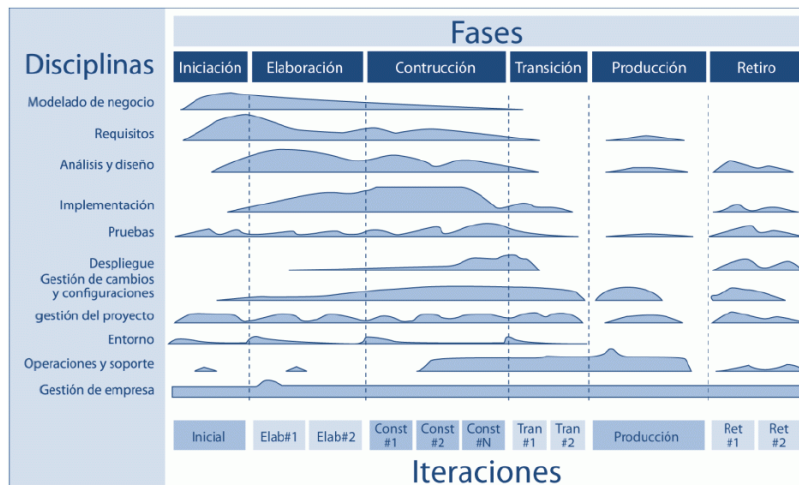
- En la siguiente tabla, se describen en que fase de los diferentes métodos y metodologías de desarrollo de software, se debe integrar el flujo de trabajo del componente usuarios.
- En las metodologías RUP (Proceso Unificado Racional) y EUP (Proceso Unificado para Empresas), las fases de iniciación, elaboración, construcción y transición son iguales, la integración de MODIHC se realiza de la misma manera en ambas tecnologías.



RUP de IBM

Es un proceso de desarrollo de software iterativo e incremental que permite asignar tareas y responsabilidades en una empresa de desarrollo.

Modelo EUP y WATCH



En la metodología EUP, la actividad de requisitos y la fase de iniciación son iguales que en el RUP.

La EUP contempla el proceso de software completo.



Es el resultado de integrar modelos de desarrollo (RUP, el método de Bruegge & Dutoit, estándar de la IEEE 1074).

Aplicaciones Web basadas en componentes

Integración

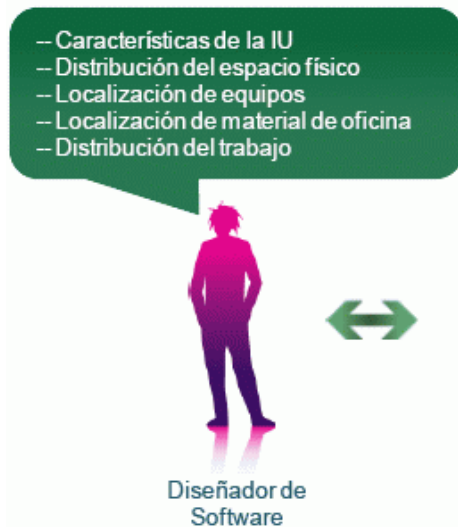
Ciclo de vida	Integración del flujo de trabajo del componente Usuarios
Cascada	<p>Debe integrarse en la fase de Análisis de requisitos.</p> <p>En este modelo, de la fase Análisis de requisitos, surge un documento de especificación de requisitos. Este documento, aparte de contener la especificación completa de lo que debe hacer el sistema, debe contener todos los datos necesarios para el diseño de la IHC.</p> <p>A los productos de esta fase deben sumarse el perfil de usuarios y las características de la IU.</p>
Espiral	<p>En la planificación inicial de este modelo se debe establecer en qué momento deben entrar las actividades de MODIHC en el ciclo de desarrollo.</p> <p>Las actividades del componente Usuarios deben realizarse antes de las del componente Ambiente de trabajo.</p> <p>Para la fase de Planificación ya se encuentran definidos los objetivos y productos, las restricciones dependerán de la situación actual del proyecto.</p> <p>En la fase de ingeniería se deben realizar las actividades definidas para este componente.</p>
RUP y EUP	<p>La creación del perfil de usuarios debe realizarse durante la disciplina Modelado de negocio en la fase de Iniciación. El proceso de identificación de actores del sistema puede completarse con la creación de este documento.</p> <p>El documento de análisis de requisitos se realiza durante la disciplina Requisitos en la fase de Iniciación, pudiendo extenderse hasta la fase de Elaboración.</p>
WATCH extendido	<p>El perfil de usuarios debe crearse en la fase de Modelado de negocio, durante la definición de los procesos estáticos.</p> <p>El documento de especificación de requisitos se elabora durante la fase de Definición y especificación de requisitos.</p>

Ambiente de trabajo

- Para el diseño de la IHC, es necesario realizar un estudio del ambiente en el cual va a operar el sistema computacional y para ello hay que tomar en cuenta tres aspectos: el organizacional, el físico y el social.
- La siguiente tabla muestra algunos de los factores que el diseñador debe tomar en consideración en cada uno de estos aspectos.
- La introducción del nuevo producto de software no debe causar impacto negativo dentro de la organización, además debe soportar las necesidades de los usuarios y proveer la funcionalidad requerida.
- Lo más importante de estos tres aspectos para MODIHC es que permite determinar ciertas características de las interfaces de usuario, la distribución del espacio, localización de equipos y material de oficina y la distribución del trabajo.
- También se muestra la relación entre el diseñador de software con los componentes usuarios y ambiente de trabajo.

Factores para el ambiente de trabajo

Aspecto	Factores
Organizacional	Estructura, misión y estrategias, cargos de trabajo, organización del trabajo, usuarios, clientes y proveedores, política y cultura.
Físico	Temperatura ambiental, iluminación natural y artificial, ruidos externos e internos y organización del espacio físico.
Social	Comunicación entre las personas e interacción con el medio que constituye su ambiente de trabajo.

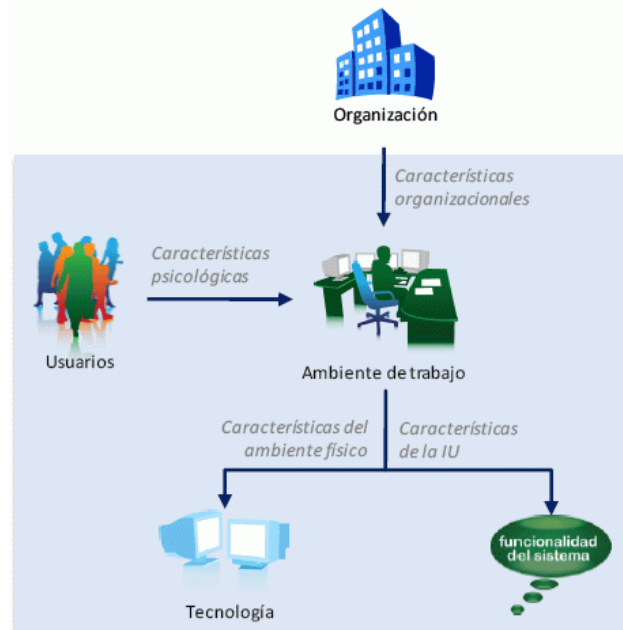


Objetivos del componente

- Los objetivos que el diseñador de software debe alcanzar en este componente son los siguientes:
 - Estudiar los aspectos organizaciones, físicos y sociales del ambiente de trabajo.
 - Conocer la distribución del espacio físico y del trabajo.
 - Localizar equipos y materiales de oficina.
 - Crear el documento de especificación del ambiente de trabajo.
 - Determinar características de la IU basadas en el ambiente de trabajo.
- Los productos que se obtienen de este componente son: un documento de especificación del ambiente de trabajo y la tabla de características de la IU basadas en ese ambiente.
- En el primer documento se describe el ambiente físico, social y organizacional donde se usará el producto de software.
- El documento igualmente contiene una descripción de cómo se distribuyen tanto el espacio físico como el trabajo y donde están localizados los equipos y materiales de oficina.

Componente ambiente de trabajo

- El contenido del documento de especificación del ambiente del trabajo y las características de la IU alimentan a los componentes tecnología y funcionalidad del sistema de MODIHC.
- Las características del ambiente físico influyen en la selección de los equipos de E/S y los estilos de interacción, las características de la IU definidas son base para el diseño de la funcionalidad del sistema.



Metodologías para el ambiente de trabajo

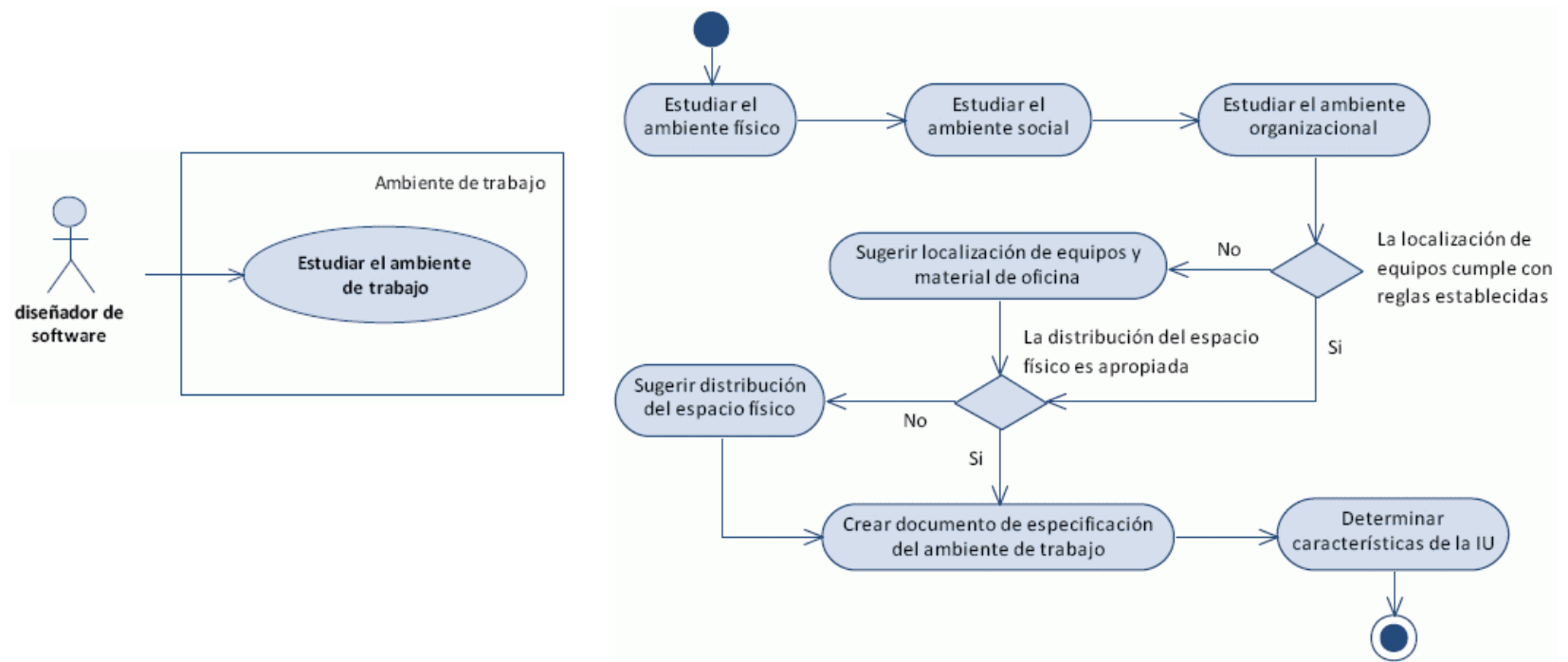
- Los datos necesarios para la creación del documento de especificación del ambiente de trabajo y las características de la IU se obtienen mediante el uso de una o varias de las técnicas de recolección de datos discutidas anteriormente.
- Se recomienda para el estudio del ambiente organizacional, el uso de la entrevista y para el estudio del ambiente físico, la observación directa en el campo, finalmente para el estudio del ambiente social, el uso tanto de la entrevista como de la observación directa en el campo.
- Existen reglas para la localización de equipos y la distribución de espacio físico que previenen riesgos laborales y favorecen la salud y seguridad de los usuarios [Steelcase, 2007].
- El diseñador debe evaluar el cumplimiento de estas reglas y advertir si no se están aplicando de manera correcta.
- Se puede incluso sugerir, en el caso de que sea necesario, la modificación del ambiente de trabajo.

Recomendaciones para la interfaz

Característica	Descripción	Valor observado	Recomendación para el diseño de la IU
Elementos e individuos presentes	Cantidad de elementos y personas que se encuentran en el lugar.	Muy recargado	Usar fuentes tipográficas de morfología simple y diseño de estructura limpia. Usar espacios en blanco para obtener descanso visual.
		Poco recargado	Pueden usarse tipografías y diseños más elaborados.
Acústica	Sonidos presentes en el ambiente de trabajo.	Altos	Debe evitarse el uso de sonidos dentro del producto de software. Los sonidos pueden no ser escuchados o resultar incómodos para el usuario.
		Bajos	Se permite el uso de sonidos dentro del software (Avisos, música de fondo, tecleo) puesto que este mantendrá despierto al usuario durante la jornada de uso. No debe exagerarse en el uso de sonidos para no aturdir o alterar al usuario.
Luz y colores	Tipo de iluminación y colores presentes en el ambiente.	Incandescente	Usar alto contraste en los colores de la IU y evitar el uso de objetos pequeños. De esta manera se esfuerza menos la vista del usuario.
		Natural	Usar colores que se adapten al ambiente y a las actividades realizadas por los usuarios.
Temperatura	Temperatura del ambiente.	Cálida	Los colores de la IU deben contrarrestar los efectos causados por la temperatura. Usar en este caso colores fríos.
		Fría	Hacer uso de colores cálidos.

Flujo de trabajo del componente

- Con el fin de formalizar el componente ambiente de trabajo, el diseñador debe cumplir con la actividad presentada en el diagrama de casos de usos. El diagrama de actividad igualmente, muestra los pasos a seguir para realizar dicha actividad.

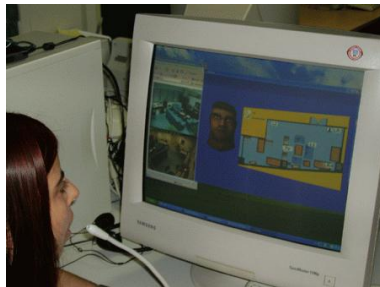


Integración con la IS

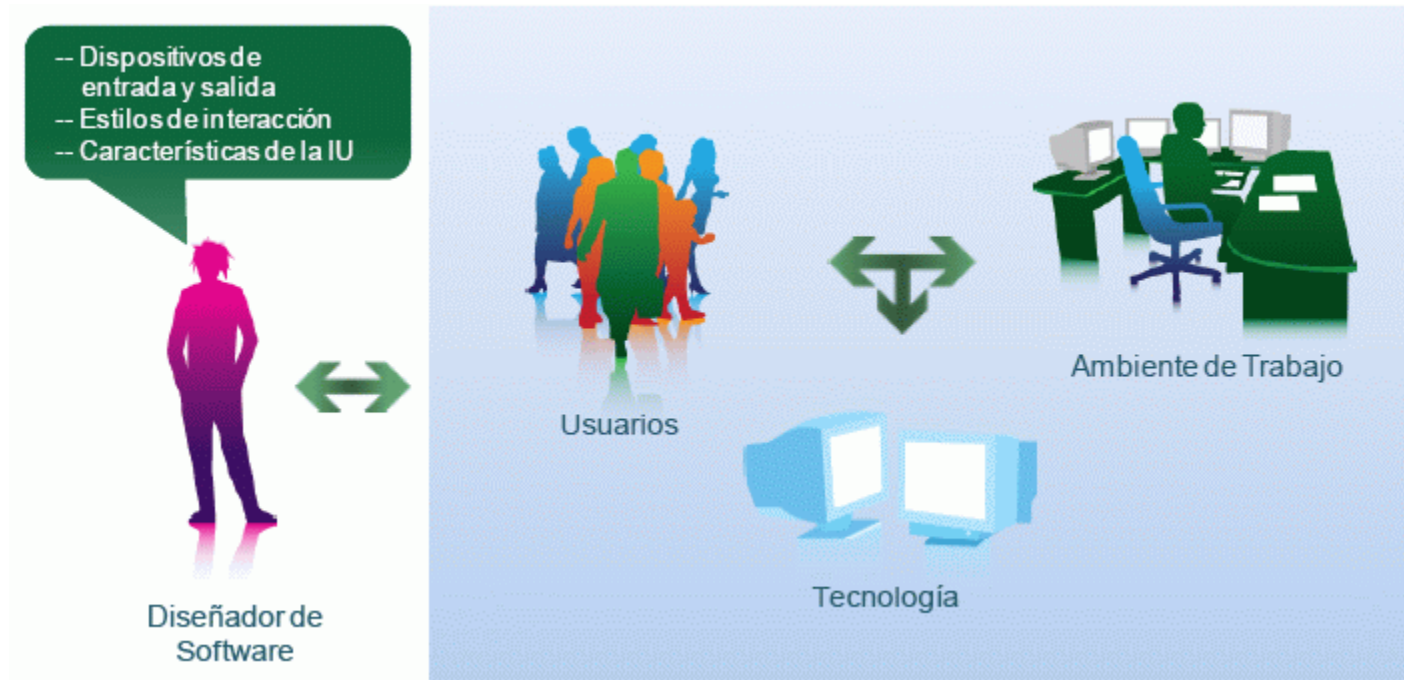
Ciclo de vida	Integración del flujo de trabajo del componente Ambiente de trabajo
Cascada	Debe integrarse en la fase de Análisis de requisitos. Las características de la IU y el documento de especificación del Ambiente de trabajo deben definirse antes de empezar la fase de diseño.
Espiral	Las actividades del componente Ambiente de trabajo deben realizarse después de las del componente Usuarios y antes de las del componente Tecnología. Para la fase de Planificación ya se encuentran definidos los objetivos y productos, las restricciones dependerán de la situación actual del proyecto. En la fase de ingeniería se deben realizar las actividades definidas para este componente.
RUP y EUP	El documento de especificación del Ambiente de trabajo debe realizarse en la fase de Iniciación, durante la actividad de Modelado de negocio. Las características de la IU se definen durante la actividad de Requisitos.
WATCH extendido	El documento de especificación del Ambiente de trabajo debe realizarse en la fase de Modelado de negocio, durante la definición de procesos estáticos. La localización de equipos y material de oficina se realiza en la fase de Diseño arquitectural de la aplicación, en el paso de Diseño de la arquitectura del sistema.

Tecnología

- Para diseñar la IHC, desde el punto de vista de la tecnología, es necesario determinar en primer lugar los dispositivos de E/S apropiados para la interacción con dicho sistema, tomando en cuenta la disponibilidad o posibilidad de adquisición de los mismos dentro de la organización, y en segundo lugar los diferentes estilos de interacción (elementos de software), teniendo siempre en mente a los usuarios, la funcionalidad del sistema y el ambiente de trabajo.
- La conjunción de todos estos factores permitirá determinar, entre otras cosas, la forma cómo se van a visualizar las tareas y objetos del sistema, si es necesario o no el uso de sonido, texto, gráficos, visualización 2D o 3D, animación, realidad virtual, etc.



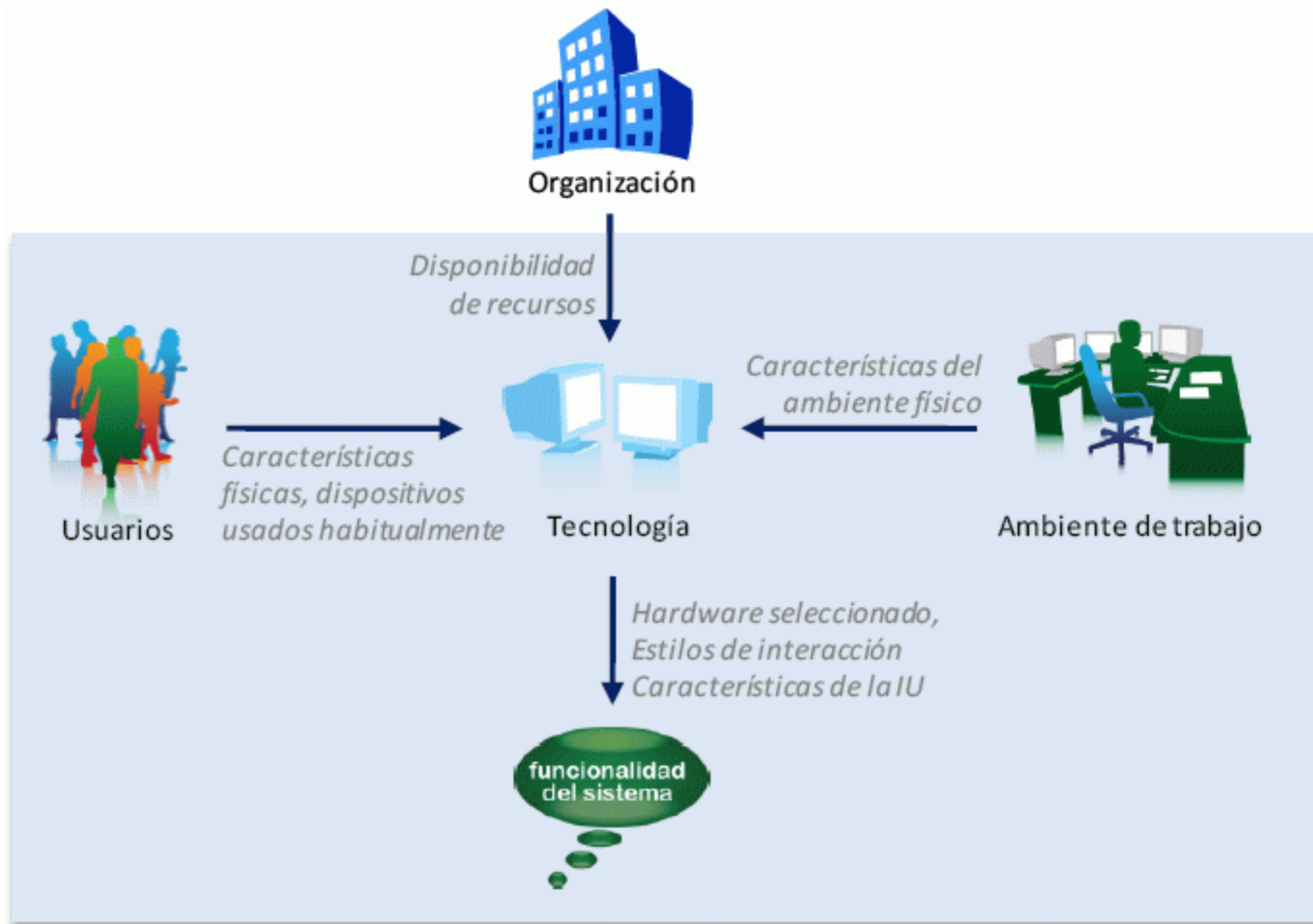
Relación diseñador-tecnología



Objetivos del componente tecnología

- Los objetivos que el diseñador debe alcanzar son:
 - Seleccionar los dispositivos de E/S que mejor se adapten al usuario, al ambiente de trabajo y a la disponibilidad de recursos de la organización.
 - Seleccionar los estilos de interacción apropiados para el producto de software a desarrollar.
 - Determinar características de la IU basadas en la tecnología.
- Los productos que el diseñador de software deben obtener de este componente son, una lista con los dispositivos de E/S elegidos, los estilos de interacción seleccionados y la tabla de características de la IU basadas en la tecnología.
- Los dispositivos de E/S seleccionados van acompañados de una breve descripción del uso que tendrán en el manejo del producto de software.
- La tabla de características de la IU presenta los dispositivos de E/S y los estilos de interacción seleccionados, acompañados de las condiciones que debe cumplir la IU en su diseño.

Relación tecnología con MODIHC



Dispositivos de E, S y E/S

Tipo de dispositivo	Descripción	Ejemplos
Entrada	Introducen datos externos a la computadora para su posterior tratamiento por parte del CPU. La principal fuente de estos datos son los usuarios del producto de software.	<ul style="list-style-type: none">- Ratón- Teclado- Escáner- Cámara web- Lápiz óptico- Palanca de mando o <i>Joystick</i>- Micrófono- Conversor analógico-digital- Lector de código de barras
Salida	Muestran o proyectan información hacia el exterior del ordenador. La mayoría son para informar, alertar, comunicar, proyectar o dar al usuario cierta información. Se encargan de convertir los impulsos eléctricos en información legible para el usuario.	<ul style="list-style-type: none">- Monitor- Altavoz- Impresora- Bocinas- Auriculares- Trazador de gráficos o <i>Plotter</i>
Entrada/salida	Cumplen las funciones de los dispositivos de entrada y los dispositivos de salida.	<ul style="list-style-type: none">- Fax- Módem- Unidades de almacenamiento (CD, DVD, Pendrive)- Pantalla táctil

Metodologías para el componente tecnología

- Para la selección de los dispositivos apropiados para la interacción con un sistema computacional, el diseñador debe tomar en cuenta:
 - Ayudar al usuario a realizar sus tareas de manera segura, efectiva, eficiente y agradable.
 - Determinar las manipulaciones particulares que deben llevarse a cabo para realizar las tareas requeridas.
 - Establecer una correspondencia natural entre la forma en que se manipula el dispositivo, la retroalimentación dada por el sistema y el significado del resultado en términos del modelo mental del usuario.
 - Permitir, cuando sea necesario, la interacción con usuarios discapacitados.
 - Tomar en cuenta ambientes de trabajo con necesidades especiales.
- Estilos de interacción es un término genérico que agrupa las diferentes maneras en que los usuarios se comunican o interaccionan con una computadora.

Estilos de interacción I

- Las interfaces de usuario pueden estar formadas por uno o varios estilos de interacción. Los principales estilos de interacción:
 - **Sistema de comandos:** Es un pequeño lenguaje que permite activar las diferentes partes del producto de software. Actúan como interpretes en tiempo real durante la interacción con el usuario.
 - **Formularios:** Se utilizan para la entrada de datos o recuperación de información. Consiste en una serie de campos que el usuario debe completar ya sea con texto o seleccionando entre múltiples opciones.
 - **Menús:** Es un conjunto de opciones visualizadas en pantalla que se pueden seleccionar y llevan a la ejecución de una acción asociada. Existen dos tipos: menú tipo índice (describen mediante palabras o frases muy cortas lo que la opción puede hacer) y menú mediante iconos (describen las acciones del producto de software mediante pequeños gráficos o iconos).
 - **Lenguaje natural:** La comunicación de los usuarios hacia las computadoras se realiza mediante el uso de la voz o mediante la escritura, donde no es necesario el uso de sintaxis específicas para realizar acciones.

Estilos de interacción II

- **Manipulación directa:** Representan los objetos y las acciones de manera directa y en muchos casos realista. Permite la manipulación directa y la simulación de los objetos de interés quienes remplazan los comandos.
- Los estilos de interacción compuestos, son aquellos formados por uno o más estilos de interacción, podemos distinguir:
 - **Menú con comandos:** Se basa en una combinación de menús con instrucciones en lenguaje de comando, que permite a usuarios expertos acceder rápidamente a las funcionalidades del sistema.
 - **Interfaz gráfica de usuarios (IGU):** Como se puede de su nombre, este tipo de interacción utiliza una representación gráfica en la pantalla de la aplicación, datos y objetos a utilizar. Por lo tanto, se fundamenta en una combinación de elementos usados para la interacción simple, además de apuntadores que permiten manipular los objetos.
 - **Interfaz orientada a objetos:** Su centro es la manipulación de objetos, es decir, no dependen de la aplicación como el caso de las interfaces gráficas de usuarios.

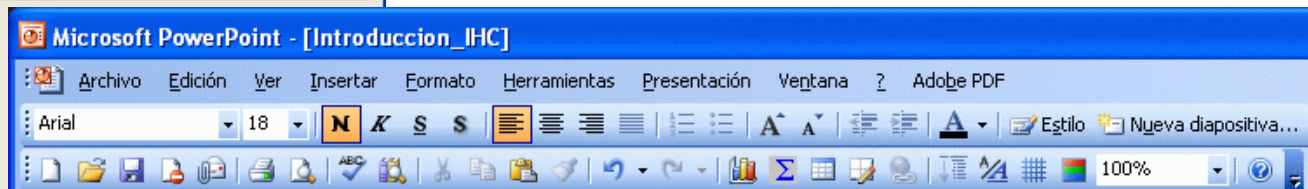
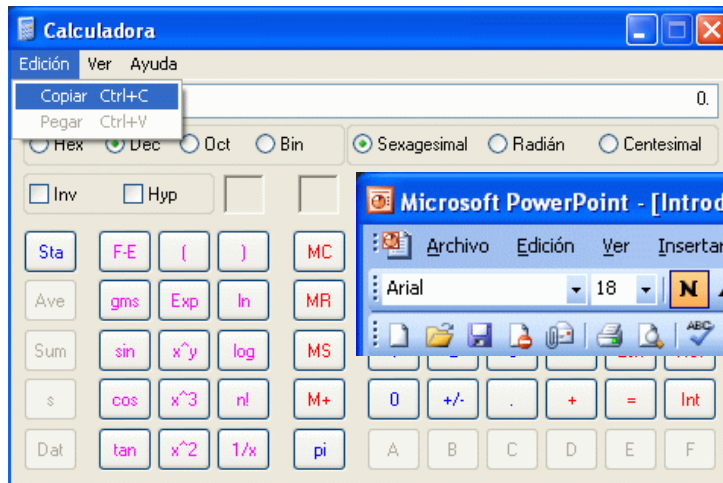
Fortalezas y debilidades de los estilos

Estilo de Interacción	Fortalezas	Debilidades
Sistema de comandos	<ul style="list-style-type: none">• Flexibilidad, la orden puede ser aplicada a varios objetos a la vez.• Permite la iniciativa del usuario.• Es atractivo para usuarios expertos.• Potencialmente rápido para tareas complejas.• Capacidad para hacer macros.	<ul style="list-style-type: none">• Dificultad de aprendizaje.• Requiere una memorización y entrenamiento importante.• El software no presenta las acciones disponibles.• Las mismas acciones utilizan diferentes órdenes en sistemas distintos.
Formulario	<ul style="list-style-type: none">• Simplifica la entrada de datos.• Guía al usuario al seguir reglas predefinidas.• Permite la navegación entre campos y corrección de datos.• No necesitan aprendizaje por parte del usuario.	<ul style="list-style-type: none">• Limita las acciones del usuario.• Ocupan mucho espacio.• Por lo general el proceso se debe resumir en el llenado del formulario.
Menús	<ul style="list-style-type: none">• Entrenamiento reducido, menos tecleo.• Selección rápida de las opciones que se presentan todas de forma mnemónica y sinóptica.• Toma de decisión estructurada.	<ul style="list-style-type: none">• No se pueden programar secuencias de acciones.• Son muy inflexibles.• No se pueden parametrizar.• Puede resultar lento para usuarios expertos.
Lenguaje natural	<ul style="list-style-type: none">• Conocimiento del propio lenguaje.• Uso de la voz, por tanto manos libres.	<ul style="list-style-type: none">• Diferencias en lenguajes y voces.• Interfaces poco inteligentes.

Más de los estilos de interacción

Manipulación directa

- Los nuevos usuarios aprenden más rápido.
- Los usuarios expertos trabajan de manera rápida.
- Se ve en menor tiempo el resultado de las acciones.
- Las acciones son reversibles.
- Se necesitan más recursos.
- No todas las tareas pueden ser modeladas en objetos concretos.
- No todas las tareas se pueden hacer directamente.
- Poca accesibilidad, principalmente para personas con discapacidad visual.



Diseño de la interfaz

Estilo de Interacción	Recomendación para el diseño de la IU
Sistema de comandos	<ul style="list-style-type: none">• El lenguaje debe ser conciso y mnemónico.• Si invoca el lenguaje natural será más fácil y cómodo para el usuario.• Debe prever estructuras de control y un pequeño conjunto de tipos de datos para especificar parámetros.• Debe considerar las características contextuales, culturales del usuario, por ejemplo si la aplicación es para niños, gerentes, etc.
Formularios	<ul style="list-style-type: none">• Las instrucciones han de ser breves y comprensibles.• Las etiquetas de los campos deben usar terminología familiar.• Indicar, en los casos que se pueda y mediante mensajes, cuando un usuario ha introducido valores incorrectos.
Menús	<ul style="list-style-type: none">• No deben ser muy largos ni muy anidados.• El estilo, tipo y tamaño del texto debe ayudar a la presentación del menú.• El menú puede contener al lado de su texto la clave o tecla a pisar para una selección rápida desde el teclado, para usuarios expertos.• El menú no debe tapar los elementos presentes en la pantalla que son necesarios para tomar una decisión.
Lenguaje natural	<ul style="list-style-type: none">• Debe considerar el uso, por parte del usuario, de cualquier frase dicha o escrita.• Debe responder acorde a lo preguntado, sin caer en malos entendidos.• Indicar cuando una frase no está relacionada con el sistema.
Manipulación directa	<ul style="list-style-type: none">• Asegurarse que el usuario conozca las opciones y limitaciones en el manejo de objetos de la interfaz.• Utilizarlo con otros estilos de interacción, como el uso de comandos a través de menús o teclado, para mejorar la usabilidad del producto de software.

Mensajería I

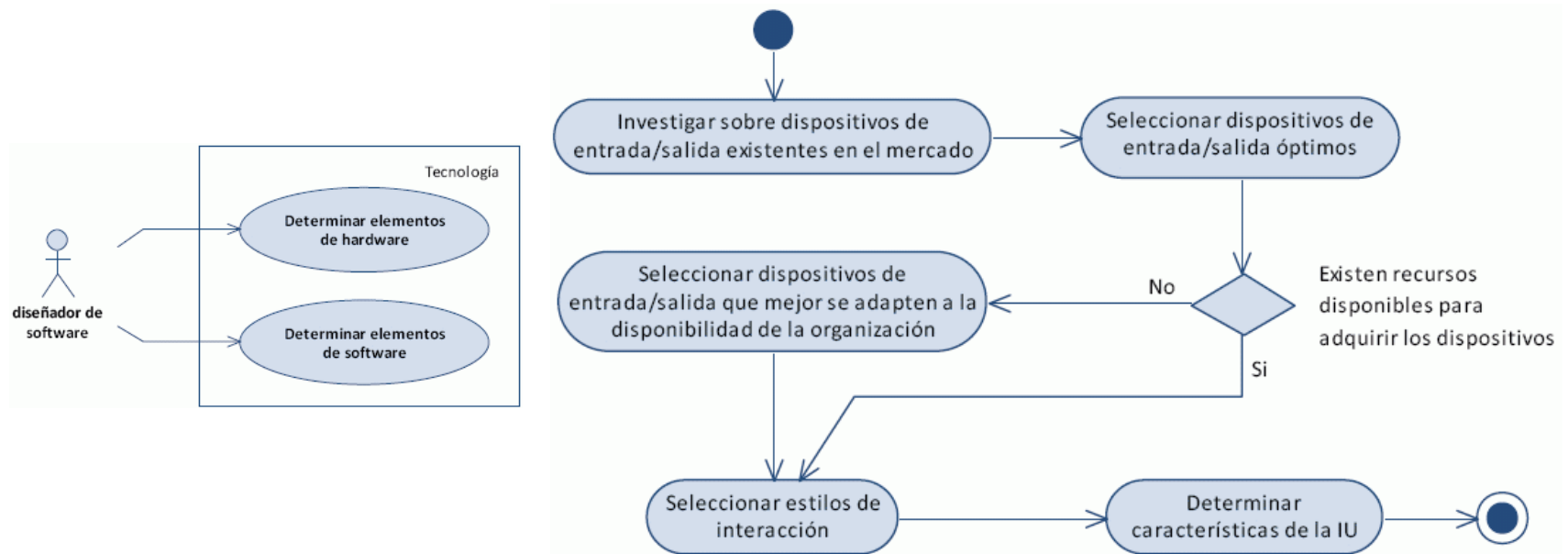
- La mensajería en un producto de software consiste en notificaciones que se muestran a los usuarios cuando se presentan situaciones especiales o excepcionales durante su uso.
- Los mensajes se despliegan principalmente para notificar errores en el sistema, advertencias, información o preguntas a los usuarios.
- El contenido de estos mensajes se obtiene del estudio de la documentación obtenida de la fase de análisis de la metodología de desarrollo que se esté aplicando.
- Todos los mensajes de error o advertencia generados por el producto de software deben tener las siguientes características:
 - Debe describir el problema en un lenguaje que puedan entender los usuarios.
 - Debe proporcionar información constructiva para recuperarse del error.
 - Debe indicar cualquier consecuencia del error de manera que los usuarios puedan comprobar si ha ocurrido.

Mensajería II

- ❑ Debe ir acompañado por una señal audible o visible, es decir, se puede emitir un sonido para acompañar a la visualización del mensaje, o el mensaje puede parpadear momentáneamente o mostrarse en un color (ya sea en el fondo, texto o ícono) fácilmente reconocible como el “color del error”.
- ❑ El mensaje no debe hacer juicios. Es decir, el texto nunca debe culpar a los usuarios.
- El utilizar códigos de color en mensajes para los usuarios reduce enormemente la mala interpretación y las respuesta incorrectas.
- El rojo es un buen color para alertar a los usuarios hacia un error.
- El amarillo es apropiado para un mensaje de advertencia y el verde para mostrar un progreso positivo.
- El usar verde para mensajes de error o el rojo para mensajes de estado positivo sólo conduce a los usuarios a malas interpretaciones y a frustraciones.

Flujo de trabajo del componente tecnología

- Con la finalidad de formalizar el componente tecnología de MODIHC, se utiliza un diagrama de casos de uso con las actividades del diseñador y en un diagrama de actividad los pasos a seguir para realizar dichas actividades.



Integración con la IS

Ciclo de vida	Integración del flujo de trabajo del componente Tecnología
Cascada	<p>En la fase de Análisis de requisitos se debe conocer la disponibilidad de recursos que tenga la organización.</p> <p>La selección de los dispositivos de entrada/salida y los estilos de interacción se debe realizar en las primeras actividades de la fase de Diseño.</p>
Espiral	<p>Las actividades del componente Tecnología deben realizarse después de las del componente Ambiente de trabajo y antes de las del componente Funcionalidad del sistema.</p> <p>Para la fase de Planificación ya se encuentran definidos los objetivos y productos, las restricciones dependerán de la situación actual del proyecto.</p> <p>En la fase de ingeniería se deben realizar las actividades definidas para este componente.</p>
RUP y EUP	<p>La selección de los dispositivos de entrada/salida y los estilos de interacción se deben realizar durante la actividad de requisitos en la Fase de Elaboración.</p>
WATCH extendido	<p>En la fase de Diseño arquitectural de la aplicación, dentro de las actividades del Diseño de la IU se definen los estilos de interacción y dentro de las actividades del Diseño de la arquitectura del sistema se seleccionan los dispositivos de entrada/salida.</p>

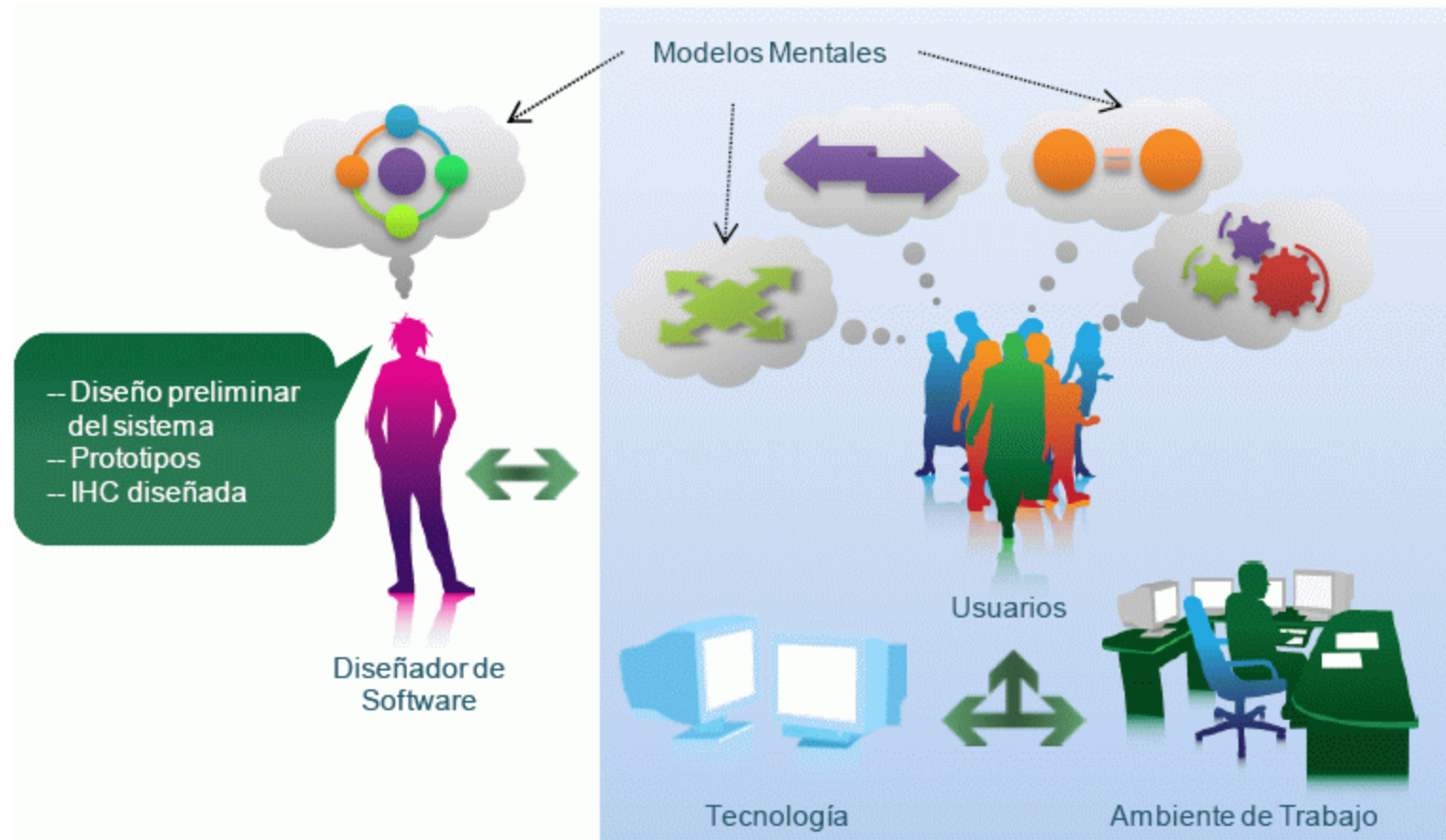
Funcionalidad del sistema I

- El uso de productos de software para la realización de diferentes actividades, ha aumentado debido a que estos permiten a los humanos realizar las tareas de una manera más rápida y eficiente.
- El siguiente paso de los diseñadores es definir una IU que exprese lo que hará el producto de software en función del dominio de las tareas o actividades que realizan los usuarios, las condiciones del ambiente en que va a ser usado y las restricciones bajo las cuales dicho software debe operar para asegurar la construcción de un sistema correcto.
- El número de objetos y tareas definidos debe satisfacer tanto los requisitos del usuario como la funcionalidad adecuada y eficiente del sistema.
- El diseñador de software debe estudiar el modelo mental que cada uno de los usuarios tiene sobre el sistema y cómo razonar con respecto a sus funciones.

Funcionalidad del sistema II

- Debe igualmente, mezclar estos diferentes modelos mentales para así determinar, mediante la IU, la funcionalidad correcta del sistema.
- Además, debe conocer las preferencias de los usuarios para determinar lo que ellos encontrarán aceptable como un sistema usable.
- Es importante para el diseño de la IU, estudiar la documentación obtenida de la fase de análisis de la metodología de desarrollo utilizada.
- La documentación puede contener, diagramas de casos de uso, especificación de actores y actividades, componentes del sistema, especificación de requisitos de la IU, etc.
- Esto permite al diseñador de software determinar y organizar los datos que se deben solicitar al usuario y la información que se debe mostrar al usuario y en qué momento.

Relación diseñador-funcionalidad

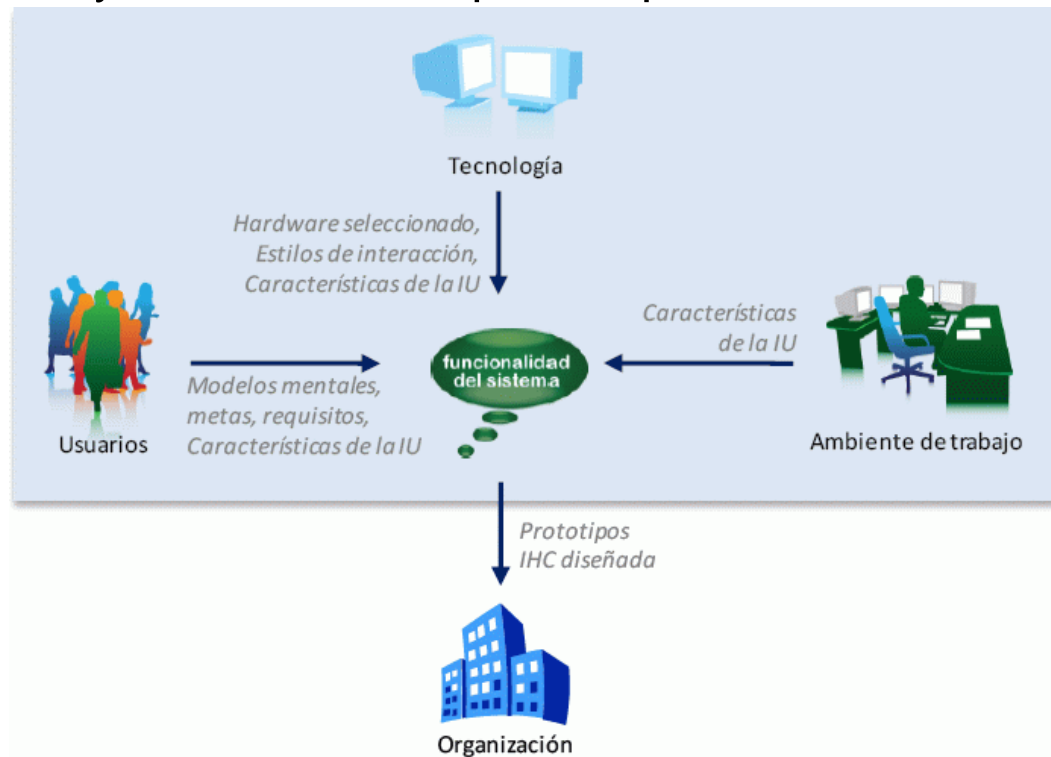


Objetivos de la funcionalidad

- Los objetivos que el diseñador de software debe alcanzar son:
 - Identificar las necesidades de la organización que debe satisfacer el producto de software.
 - Identificar las metas y requisitos de los usuarios que deben cumplirse mediante el uso del producto de software.
 - Estudiar el modelo mental que cada usuario tiene del sistema.
 - Estudiar las características de la IU definidas en los componentes anteriores.
 - Estudiar la información obtenida de la fase de análisis del método o metodología de desarrollo de software que se aplique.
 - Diseñar una IU que represente de manera apropiada la funcionalidad del sistema.
- El producto de este componente es el diseño de la IHC del producto de software que satisface los requisitos de los usuarios y los requisitos organizacionales, cumple con la filosofía del desarrollo centrado en el usuario, etc.

Productos del componente funcionalidad

- El diseño de la IHC se entrega a la organización en la forma de un documento (informe final) donde se detalla el perfil de usuarios, la especificación de requisitos, el ambiente de trabajo, la tecnología recomendada y la IU diseñada para el producto de software.



Metodologías para el componente

- El diseño de la IU de un producto de software va evolucionando de manera iterativa a través de ciclos de diseño-evaluación-rediseño en los que los usuarios siempre están presentes.
- Existen dos tipos de diseño: conceptual (elaboración de un modelo conceptual que representa lo que el producto de software va a hacer y cómo se comportará) y físico (se refiere a los detalles del diseño, como pantallas, estructuras de menú, iconos y gráficos).
- Para que los usuarios y la organización puedan ir evaluando de manera efectiva los diseños de la IU que se van creando, el diseñador debe crear prototipos.
- A medida que los prototipos se van corrigiendo, la IU diseñada se hace más firme.
- Un prototipo es una representación limitada de un diseño que permite a los usuarios interactuar con él y explorar su idoneidad.

Diferentes tipos de prototipos

- Es un medio entre los diseñadores de software y los usuarios para discutir ideas sobre el funcionamiento del sistema y evaluar si los modelos mentales anteriormente definidos se encuentran reflejados en el diseño.
- Los prototipos se clasifican en dos tipos:
 - Prototipos de baja fidelidad y
 - Prototipos de alta fidelidad
- El uso de prototipos de alta fidelidad es útil para vender ideas a la gente y realizar pruebas de nivel técnico.
- Sin embargo, el uso de prototipos de baja fidelidad permite corregir problemas de contenido o estructura.
- Obviamente, existen ventajas y desventajas de usar uno u otro tipo de prototipo.

Ejemplos de diferentes tipos de prototipos

Tipo	Descripción	Ejemplos	
Baja fidelidad	Estos prototipos no se parecen mucho al producto final. Su utilidad radica en el hecho de ser simples, económicos y de fácil desarrollo.	Descripción gráfica (<i>storyboarding</i>)	Consiste en una serie de bocetos de la UI que representan la secuencia en que serán mostrados.
		Bosquejos (<i>sketching</i>)	Consiste en modelar mediante bocetos simples los elementos del sistema y la relación entre estos.
		Tarjetas (<i>index cards</i>)	El uso de tarjetas permite observar el comportamiento de la UI de manera dinámica. Es usado comúnmente en el diseño de interfaces de sitios web.
		Mago de Oz	Hace uso de la computadora, donde el usuario interactúa directamente con la UI pero es el diseñador quien simula las respuestas que debe generar el producto de software.
Alta fidelidad	Son prototipos que lucen como el producto final y utilizan elementos que se esperan estén en el diseño final.	Para crear prototipos de productos de software es necesario usar herramientas de desarrollo de software. Por ejemplo: Flash, Visual Basic o SmallTalk.	

Comparación entre los prototipos

Tipo	Ventajas	Desventajas
Prototipo de baja fidelidad	<ul style="list-style-type: none">- Menor costo de desarrollo.- Evalúa múltiples diseños de concepto.- Útil medio de comunicación.- Útil para identificar requisitos del mercado.	<ul style="list-style-type: none">- Comprobación de errores limitada.- Poca especificación detallada de código.- Manejado por el diseñador de software.- Utilidad limitada luego de establecer requisitos.- Poco útil para pruebas de usabilidad.- Presenta limitaciones en la navegación.
Prototipo de alta fidelidad	<ul style="list-style-type: none">- Funcionalidad completa.- Completamente interactivo.- Manejado por los usuarios.- Define claramente los esquemas de navegación.- Útil para explorar y realizar pruebas.- Presenta la apariencia del producto final.- Útil como herramienta de ventas y mercadeo.	<ul style="list-style-type: none">- Más costoso de desarrollar.- Consume más tiempo en su creación.- No es efectivo para recaudar requisitos.

Realización de prototipos I

- Para la realización se recomienda el uso de prototipos de baja fidelidad usando la simbología y los diagramas de UML.
- El uso de estos modelos permite aprovechar los diagramas para el diseño del sistema cuando se integra MODIHC con métodos o metodologías de desarrollo de software.
- La evaluación de los prototipos y diseños de la IU permiten localizar problemas que afecten su utilización,
- Algunas pautas para evaluar una IU, son las siguientes:
 - Visibilidad del estado del sistema
 - Semejanza del sistema al mundo real
 - Control y libertad por parte del usuario
 - Consistencia y estandarización
 - Prevención de errores
 - Reconocimiento de acciones y opciones
 - Flexibilidad y eficiencia en el uso

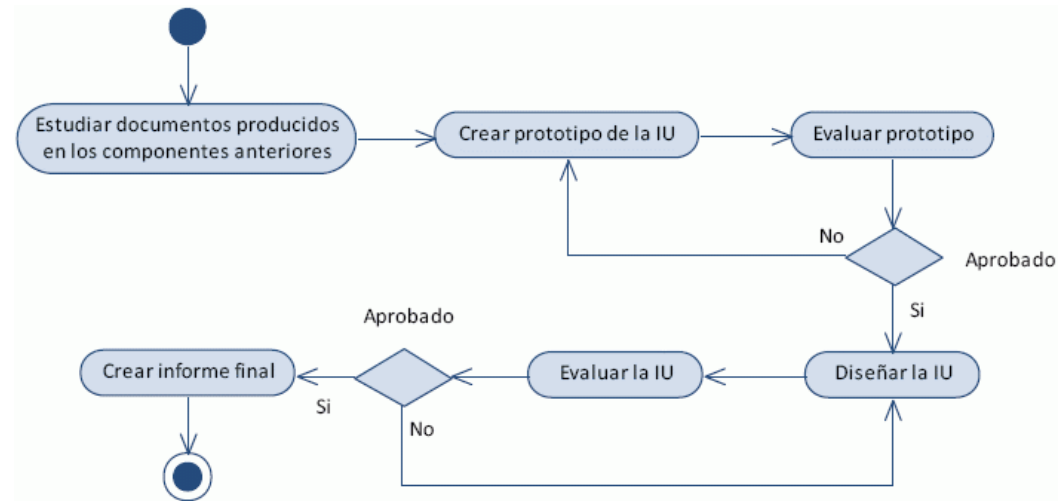
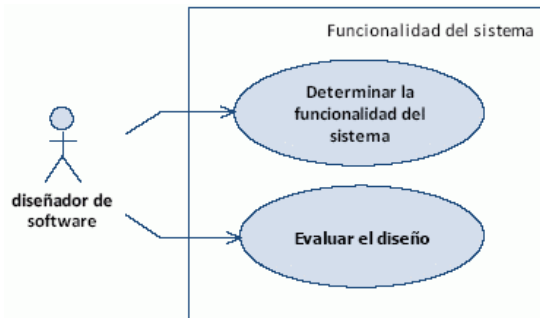
Realización de prototipos II

- ❑ Estética y diseño minimalista
- ❑ Reconocimiento de errores, diagnóstico y recuperación
- ❑ Ayuda y documentación
- Para establecer medidas que indiquen la severidad de los problemas en el uso de las interfaces de usuario, se deben reconocer los factores que determinan el grado de un problema.
- Entre estos factores se encuentran la frecuencia de ocurrencia, el impacto que causa la ocurrencia del problema, la persistencia del problema y el impacto en el mercado.

Nivel	Severidad del problema encontrado
0	No puede llegar a considerarse un problema.
1	Es un problema “cosmético” que no necesita ser corregido a menos que se disponga tiempo extra en el proyecto.
2	Es un problema menor y su corrección puede tener baja prioridad.
3	Es un problema mayor y su corrección debería tener alta prioridad.
4	Es una catástrofe para la utilización de la aplicación y es imperativo corregir el error.

Flujo de trabajo del componente

- Al igual que en los componentes anteriores, se propone un diagrama de casos de uso y un diagrama de actividades.



Integración con la IS

Ciclo de vida	Integración del flujo de trabajo del componente Funcionalidad del sistema
Cascada	<p>Tanto las actividades de realización de prototipos como las de diseño de la IU se deben realizar en la fase de Diseño.</p> <p>El diseño de la IU y de la IHC debe estar listo para el inicio de la fase de Codificación.</p>
Espiral	<p>Las actividades del componente Funcionalidad del sistema deben realizarse después de las del componente Tecnología.</p> <p>Para la fase de Planificación ya se encuentran definidos los objetivos y productos, las restricciones dependerán de la situación actual del proyecto.</p> <p>En la fase de ingeniería se deben realizar las actividades definidas para este componente.</p>
RUP y EUP	<p>En la actividad de requisitos, como parte de los requisitos de usabilidad, se diseña la IU. En la fase de Elaboración.</p>
WATCH extendido	<p>Este método tiene como primer paso dentro de la fase de Diseño arquitectural de la aplicación es el Diseño de la IU.</p>

Conclusiones

- Las cuatro componentes de MODIHC cuentan con una definición formal que permite al diseñador de software conocer la manera apropiada de aplicar el modelo y cómo integrarlo con algunos métodos de desarrollo de software.
- MODIHC fue concebido con el fin de proporcionar a la comunidad de la IS una herramienta útil para el diseño de la IHC de productos de software, generando interfaces de usuarios usables, útiles, eficaces y eficientes.
- MODIHC puede aplicarse en el desarrollo de productos de software de diversa naturaleza, ya que realmente es un modelo genérico.
- La integración de la evaluación durante el proceso de diseño de IU permite garantizar la aceptación de un producto de software por parte de los usuarios.