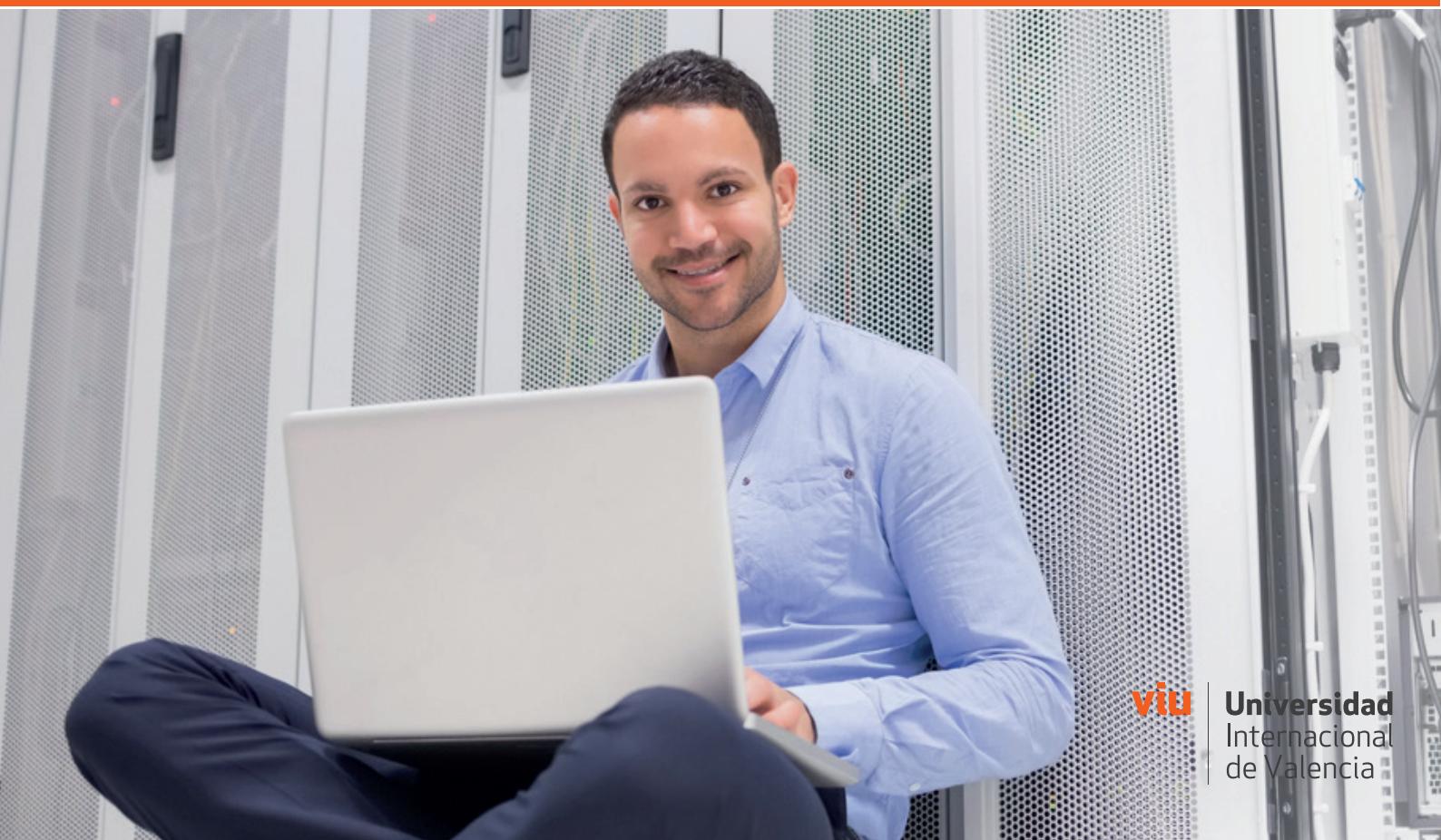


GRADO EN INGENIERÍA INFORMÁTICA

Mención Tecnologías de Información

DESARROLLO DE SISTEMAS INTERACTIVOS

Dra. Adelaide Bianchini





Este material es de uso exclusivo para los alumnos de la VIU. No está permitida la reproducción total o parcial de su contenido ni su tratamiento por cualquier método por aquellas personas que no acrediten su relación con la VIU, sin autorización expresa de la misma.

Edita

Universidad Internacional de Valencia

Grado en
Ingeniería Informática

Desarrollo de sistemas interactivos

Mención Tecnologías de Información

6 ECTS

Dra. Adelaide Bianchini

Índice

| | |
|--|----|
| TEMA1. INTRODUCCIÓN Y MOTIVACIÓN..... | 7 |
| TEMA 2. SISTEMAS INTERACTIVOS Y LA INTERACCIÓN HUMANO-MÁQUINA | 15 |
| 2.1. Conceptos preliminares | 18 |
| 2.2. Arquitectura de un sistema interactivo | 19 |
| 2.3. Las dimensiones de la interacción..... | 20 |
| 2.4. Los tipos de interacción..... | 23 |
| TEMA 3. EL DISEÑO CENTRADO EN EL USUARIO..... | 25 |
| 3.1. Actividades del proceso de diseño de interacción centrado en el usuario | 27 |
| 3.2. Modelo de proceso | 31 |
| TEMA 4. LOS REQUISITOS DE USUARIO | 33 |
| 4.1. Obteniendo y analizando los requisitos de usuario | 35 |
| 4.1.1. Obtención de requisitos de usuario | 36 |
| 4.1.2. Análisis y negociación de requisitos de usuario..... | 44 |
| 4.2. Especificando los requisitos de usuario | 45 |
| 4.3. Modelando usuarios y tareas. Enfoques | 45 |
| 4.4. Caso de estudio: <i>MyEnchantedMuseum</i> . Una vista a algunos documentos generados respecto a los requisitos de usuario | 47 |
| TEMA 5. PRODUCIENDO DISEÑOS. TÉCNICAS DE PROTOTIPADO..... | 53 |
| 5.1. Características y clasificación de prototipos..... | 54 |
| 5.2. Técnicas generales de prototipado. Mejores prácticas..... | 57 |
| 5.3. Prototipos generados para el sistema interactivo <i>MyEnchantedMuseum-Interactive Complex</i> | 59 |
| TEMA 6. DISEÑANDO LAS INTERFACES DE USUARIO | 65 |
| 6.1. Conceptos preliminares sobre interfaces de usuario..... | 65 |
| 6.2. Estilos y objetos de las interfaces..... | 68 |
| 6.2.1. Estilos de interfaces | 68 |
| 6.2.2. Objetos de las interfaces | 73 |

| | |
|--|-----|
| 6.2.3. Los mensajes en las interfaces..... | 77 |
| 6.3. Principios, directrices, guías y estándares de diseño de interfaces | 78 |
| TEMA 7. CALIDAD DE LOS SISTEMAS INTERACTIVOS: USABILIDAD, ACCESIBILIDAD Y AFFORDANCE..... | |
| 7.1. Usabilidad..... | 86 |
| 7.2. Métodos y técnicas para la evaluación de la usabilidad..... | 87 |
| 7.3. Proceso de evaluación de usabilidad y algunas métricas..... | 90 |
| 7.4. Accesibilidad..... | 93 |
| 7.5. Algunas consideraciones sobre los conceptos de usabilidad y accesibilidad..... | 95 |
| 7.6. Evaluación de las soluciones para el sistema interactivo <i>MyEnchantedMuseum - Interactive Complex</i> | 95 |
| TEMA 8. HERRAMIENTAS DE SOPORTE AL PROCESO DE DISEÑO Y DESARROLLO DE INTERACCIÓN E INTERFACES | |
| CONCLUSIONES..... | 99 |
| ANEXOS | 105 |
| Anexo A. Primitivas de interacción..... | 107 |
| Anexo B. Paradigmas de interacción..... | 107 |
| Anexo C. Características y clasificación de requisitos de software..... | 111 |
| Anexo D. Utilización de mensajes en las interfaces de usuario | 122 |
| Anexo E. Algunos principios de diseño de interfaces..... | 131 |
| Anexo F. Recursos en línea para la evaluación de usabilidad..... | 132 |
| GLOSARIO..... | 135 |
| ENLACES DE INTERÉS..... | 137 |
| BIBLIOGRAFÍA..... | 143 |

Leyenda



Glosario

Términos cuya definición correspondiente está en el apartado “Glosario”.



Enlace de interés

Dirección de página web.

Tema1. Introducción y motivación

"En el mundo de la tecnología digital, la forma, la función, el contenido y el comportamiento están tan inextricablemente vinculados, que muchos de los desafíos de diseñar un producto interactivo van directamente al corazón de lo que es un producto digital y lo que hace".

(Cooper, Reimann y Conin, 2007, p.33)

Las tecnologías actuales influencian y dan soporte a una amplia gama de tareas que los usuarios realizan cotidianamente. Las tecnologías están integradas (y embebidas) en distintos productos y contextos: desde el control de artefactos electrodomésticos, o de uso general, a distancia (domótica, IoT – Internet of Things), pasando por sitios web, comunidades en línea, aplicaciones para teléfonos móviles y tabletas, sistemas de bibliotecas en línea, museos interactivos, sistemas de navegación para distintos dispositivos, hasta aplicaciones ubicuas especializadas con las cuales un gerente puede revisar el rendimiento de su empresa, hacer la gestión de sus ingresos, egresos, pagos, etc. En conclusión, se utilizan tecnologías que son mucho más potentes que las utilizadas únicamente en los computadores años atrás, y hasta el observador más casual de los desarrollos tecnológicos ha notado que los **productos interactivos** pueden volverse rápidamente muy complejos.

Además de la tecnología subyacente, todos estos “aparatos” tienen otro aspecto en común: se debe **interactuar** con ellos a pesar de la diversidad de usuarios, **contextos de uso** y tareas que desempeñan con el producto.

Algunas preguntas que suelen plantearse los diseñadores de sistemas interactivos son:

- ¿Cómo diseñar un producto para usuarios distintos?
- ¿Cómo maximizar la efectividad y productividad del usuario en el uso de los dispositivos/sistemas?
- ¿Cómo mejorar la experiencia del usuario respecto al sistema?

El **desarrollo de sistemas interactivos** tiene que ver con el diseño, desarrollo e implantación de sistemas, entornos, productos y servicios interactivos de alta calidad que se adapten a los usuarios y a sus formas de vida.

Al igual que muchas disciplinas, su énfasis principal es en el **proceso de diseño** de la solución. En este ámbito, todo lo relacionado con la interacción tiene que ver con la forma, el contexto de uso y las tareas inherentes al producto con el cual se interactuará.

Según la **teoría de la Gestalt**, las **personas** perciben un producto no como un conjunto de características y atributos individuales, sino como “un todo” unificado en una relación con su entorno. Considerando lo anterior, no es posible diseñar de manera efectiva un producto interactivo si se descompone en una lista de componentes atómicos para luego encontrar una solución de diseño para cada uno de éstos. Incluso un producto relativamente simple debe considerarse en su totalidad y a la luz de su contexto en el entorno donde se utilizará. Por lo tanto, se requiere de un enfoque metódico que ayude a proporcionar la perspectiva holística necesaria para crear productos para que las personas los consideren útiles y “atractivos”.

En respuesta a la necesidad de técnicas, estrategias y enfoques, las investigaciones llevadas a cabo, desde al menos cuatro décadas, han dado origen a una disciplina llamada **Interacción Humano-Máquina** (HMI, por sus siglas en inglés *Human Machine Interaction*), y llamada **Interacción Humano-Computador** (HCI, por sus siglas en inglés *Human Computer Interaction*), en el ámbito de productos informáticos.

El Grupo SIGCHI (<https://sigchi.org>), de la ACM (Association for Computer Machinery, <https://www.acm.org>), es la asociación de profesionales más grande del mundo que trabaja en la investigación y práctica de la interacción humano-computador, y proponen el siguiente concepto para entender el objetivo de esta disciplina:

“Es la disciplina relacionada con el diseño, evaluación e implementación de sistemas informáticos interactivos para el uso de seres humanos, y con el estudio de los fenómenos más importantes con los que están relacionados”. (ACM SIGCHI, 1992, p.5).

El diseño de la interacción también es una tarea inherentemente humanista. La disciplina Interacción Humano-Máquina se preocupa significativamente por satisfacer las necesidades y deseos de las personas que utilizan un producto o servicio interactivo, y ante todo se centra en algo que las

disciplinas de diseño tradicionales no suelen explorar, es decir el diseño del comportamiento de los usuarios ante el producto (Benyon, 2014).

El diseño de la **interacción humano-computador** (IxD, que hace referencia al término en inglés, *Interaction Design*) es un campo amplio que se superpone con áreas como el **diseño centrado en el usuario** (UCD, por sus siglas en inglés *User Centered Design*), el diseño de la **interfaz de usuario** (UI, por sus siglas en inglés *User Interface*) y el diseño de la **experiencia del usuario** (UX, por sus siglas en inglés *User eXperience*). La disciplina Interacción Humano-Máquina, en muchos sentidos, fue precursora de la Experiencia Usuario (UX).

Es usual la confusión que se ha creado con los términos Interacción Humano-Computador (IxD), Interfaz de usuario (UI) y Experiencia de usuario (UX). Es oportuno aclarar cada uno de ellos:

- a. La **Interacción humano-computador (IxD)** es el área que se encarga del estudio de cómo deben diseñarse productos y sistemas interactivos: cuáles serán las acciones que llevará a cabo un usuario para alcanzar su objetivo al utilizar el producto y de qué modo deberá hacerlo.
- b. La **Interfaz de usuario (UI)** es el elemento mediador entre el producto y el usuario. En la interfaz se presentan todos aquellos elementos con los cuales interactuará directamente el usuario para accionar eventos y tareas en el producto. Para diseñar la interfaz hay que diseñar primero la interacción, que puede ser de distintos estilos y seguir distintos paradigmas.
- c. La **Experiencia de usuario (UX)** es un concepto más subjetivo, pues trata de las sensaciones, percepciones y satisfacción de un usuario al utilizar un *software* o un producto cualquiera. Para diseñar la experiencia de usuario se requiere de un buen diseño de interacción y un buen diseño de interfaz.

Sistemas interactivos

Existe una gama de productos en el mercado que entran en la categoría de sistemas interactivos, y cada uno de ellos ha cambiado la forma en que los usuarios interactúan, la facilidad de emitir "comandos" y su transformación en gestos, y el uso de canales multimedia.

Algunos casos emblemáticos son los *smartphones*, entre los cuales se puede nombrar del iPhone[©] de Apple que desde el 2007 cambió el día a día de los usuarios de telefonía celular. Otro ejemplo es el Nintendo WII[©], que con la explotación de tecnologías subyacentes (sensores), ha llevado la interacción a otro nivel: más allá del uso de dispositivos tradicionales como consolas, han diseñado un sistema basado en **interfaces hapticas** (del tipo cinestésico), cambiando la manera de interactuar. Con este juego la forma de "jugar" o hasta practicar deportes ha sido llevado a un nivel superior.

Dentro de los sistemas interactivos se cuenta también con plataformas y aplicaciones de redes sociales, con interacciones más tradicionales, pero con gran impacto debido a los diversos tipos de usuarios que las usan. Puede concluirse que los sistemas interactivos cubren una variedad de actividades del quehacer diario.

Para diseñar y desarrollar sistemas informáticos generales se requiere identificar y conocer sus **requisitos**, las capacidades de *software* y *hardware*, sus características, sus configuraciones y propiedades. En algunos casos se producen desde cero.

En el ámbito del diseño y desarrollo de sistemas interactivos, los asuntos fundamentales a conocer son los siguientes:

- a. ¿Qué es diseño y qué se debe considerar como proceso?
- b. ¿Qué tecnología, productos y dispositivos pueden crearse?
- c. ¿Para quién se desarrolla el sistema, y cómo lo apreciarán?
- d. ¿Qué actividades y tareas llevarán a cabo los usuarios con el sistema?
- e. ¿Cuál será el contexto y los usos del sistema, considerando las tareas que realiza el usuario?

Una reflexión de Mitch Kapor (1996, p.1) enuncia lo que es diseño, indicando que “es donde estás parado, con un pie en dos mundos: el mundo de la tecnología y el mundo de las personas y los propósitos humanos, y tratas de unir esos mundos”.

Los diseñadores de sistemas interactivos necesitan una variedad de habilidades y deben comprender una variedad de disciplinas para poder hacer bien su trabajo. En este sentido, Benyon (2014, p.18) recomienda a los diseñadores prepararse en una combinación de habilidades que les permita ser capaces de:

- Estudiar y comprender las **actividades, tareas, necesidades y posibles deseos** de las **personas**, los **contextos** dentro de los cuales algunas **tecnologías** pueden resultar útiles y generar requisitos para las tecnologías.
- Conocer las posibilidades que ofrecen las tecnologías.
- Investigar y **diseñar** soluciones tecnológicas que se adapten a las personas, las actividades que desean emprender y los contextos en los que ocurren esas actividades.
- Evaluar **diseños** alternativos e iterar hasta llegar a una solución.

Es evidente lo complejo que es encontrar en un único profesional todas estas capacidades. De ahí que se crean equipos de trabajo multidisciplinario para encarar un proceso de diseño adecuado y eficaz. En la Figura 1 se muestran aquellas disciplinas que contribuyen al diseño de sistemas interactivos. Las mismas están agrupadas mediante etiquetas que hacen referencia a personas, tecnologías, actividades y contextos, y diseño.

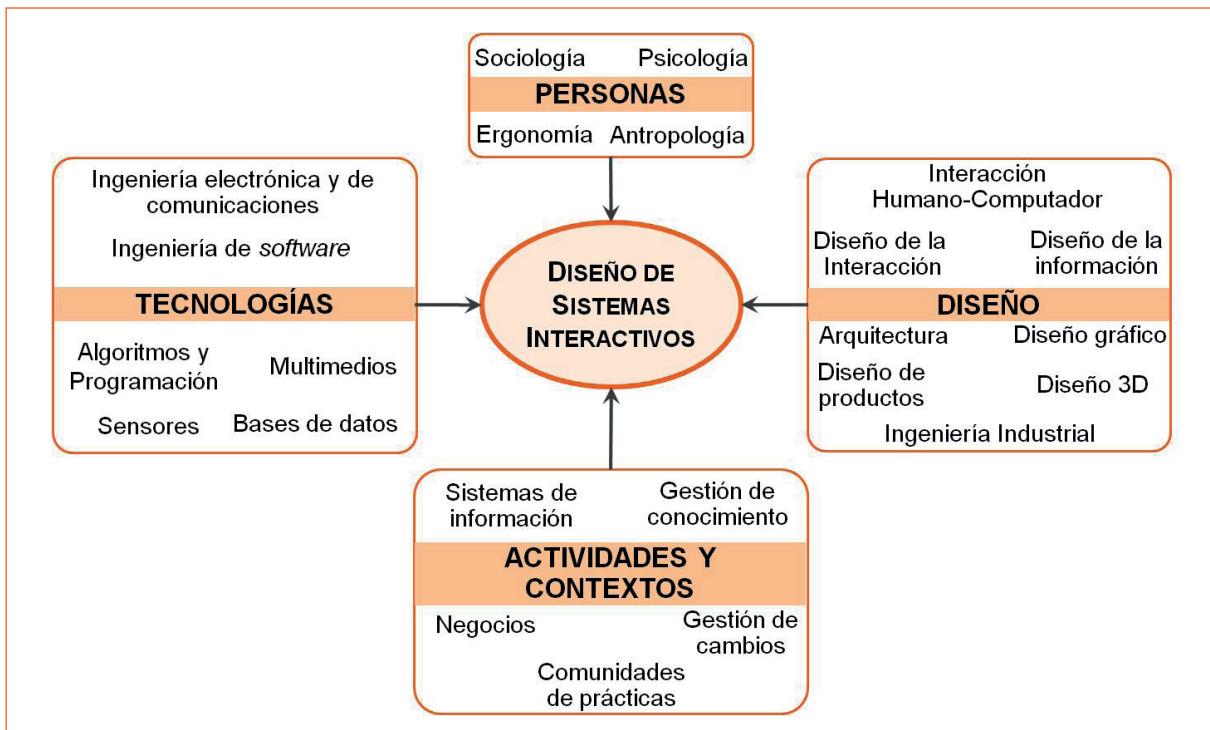


Figura 1. Algunas de las disciplinas que contribuyen al diseño de sistemas interactivos. Adaptado de Benyon (2014, p. 20)

Revisando los contenidos de la Figura 1, se deduce lo siguiente:

- Las personas son seres sociales, por lo que es importante que los enfoques y técnicas adoptados en las ciencias sociales se utilicen para comprender a las personas y la manera en que usan las tecnologías.
- Los diseñadores de sistemas interactivos necesitan conocer la tecnología, en la cual se incluyen *software* y *hardware*.
- La interacción se lleva a cabo en el contexto de alguna "comunidad de práctica". Este término se usa para denotar grupos de personas que tienen intereses compartidos, valoraciones respecto a lo que utilizan y participan en actividades similares.
- Se utilizan todo tipo de principios y prácticas en el diseño de sistemas interactivos.

Algunos aspectos que deben profundizarse en el desarrollo de sistemas interactivos son los siguientes:

- a. El proceso de desarrollo de un producto interactivo, que tiene como centro de gravedad a las personas, por lo tanto, se utiliza el enfoque el **diseño centrado en el usuario**. Este enfoque asegura, entre otras cosas, la obtención de productos seguros, sostenibles, éticos. Se refuerza así el hecho que "el diseño de la interacción es una tarea inherentemente humanista" (Benyon, 2014).
- b. La calidad de los sistemas interactivos. La forma de evaluar si las capacidades interactivas de un sistema son adecuadas, desde el punto de vista de la interacción humano-máquina (IxM) y el diseño de la interfaz (UI), está asociado al concepto de **usabilidad** y la **accesibilidad**. La

usabilidad es una característica de calidad que deben presentar productos y sistemas interactivos, y la accesibilidad se centra en la universalidad del uso de esos productos.

- c. Las herramientas disponibles para que los diseñadores puedan automatizar algunas tareas de desarrollo, **aunque lo fundamental es la manera de analizar el problema e idear una solución.**

Los **sistemas interactivos** (*software*) pueden clasificarse de la siguiente forma:

- Los **genéricos** o de uso general, considerados sistemas producidos por un grupo de desarrolladores con la finalidad de vender el producto a distintos usuarios. En este tipo se incluyen el *software* de uso de oficina, las herramientas para la gestión administrativa de empresas, etc. Como es de esperar, las características están definidas por el grupo de desarrolladores y éstos llegan a definir su especificación de una forma más ágil. En otras palabras, se conocen los requisitos con los cuales va a cumplir el producto.
- Los **personalizados**, hechos a la medida de las necesidades de una organización cliente, y desarrollados de forma específica para ella. Por ejemplo, un sistema de *software* para el control y seguimiento de proyectos de ingeniería para la construcción de barcos de pasajeros.

En este caso surgen problemas para los ingenieros informáticos cuando el dueño del producto (cliente que solicita el desarrollo de la solución) es el que cree que puede definir todo su comportamiento y características, y controla la especificación. Si además el contexto y dominio de la solución, está basado en unas reglas muy específicas, y es desconocido por los desarrolladores, aparecen serios problemas si no se alinean la terminología en el dominio del problema, el léxico y el contexto del negocio. **Hay que comunicar constantemente con los interesados del proyecto en cuestión, para refinar el alcance, los objetivos y las funcionalidades del producto final, entre otras cosas.**

Esta asignatura tiene como objetivo proporcionar los conceptos, el enfoque de diseño y desarrollo centrado en el usuario, el diseño de interacción e interfaces, los aspectos de calidad de un sistema interactivo y brindar información sobre herramientas que den soporte al proceso.

Este contenido se estructura de la siguiente forma:

- En el Tema 2 se presentan conceptos relativos a Sistemas interactivos y la interacción humano-máquina.
- El Tema 3 trata aspectos relacionados con proceso de diseño centrado en el usuario.
- En el Tema 4 se presenta el concepto de requisitos del usuario y las actividades para lograr el entendimiento del problema. Adicionalmente se ilustra un caso hipotético que se desarrolla a lo largo del contenido.
- El Tema 5 tiene como objetivo presentar algunas técnicas de creación de prototipos, como una herramienta para comunicar aspectos del diseño de un sistema interactivo a los interesados.

- El Tema 6 presenta los conceptos sobre interfaces de usuario, los distintos estilos y los objetos que sirven para construir una comunicación efectiva con el usuario durante el uso de un sistema interactivo.
- El Tema 7 tiene como objetivo presentar los aspectos de calidad de sistemas interactivos, como la usabilidad y accesibilidad, y se presentan algunas técnicas de evaluación.
- El Tema 8 presenta una panorámica de herramientas para el soporte al diseño de prototipos e interfaces de usuario.
- Finalmente se presentan conclusiones sobre los contenidos desarrollados en el ámbito de esta asignatura.
- Este material se nutre de una serie de anexos a los cuales se hace referencia en los distintos temas.

Tema 2. Sistemas interactivos y la interacción humano-máquina

“Cuando un usuario se encuentra con un dispositivo, se enfrenta a dos situaciones: la de ‘Ejecución’, donde intenta descubrir cómo usarlo, y la de ‘Evaluación’, donde intenta averiguar en qué estado se encuentra y si sus acciones lograron su objetivo.”

Donald A. Norman (2013, p.39)
Especialista en Ciencia Cognitiva en el dominio de la Ingeniería de la Usabilidad

Hasta fines de los años 70, casi todos los sistemas informáticos comerciales no eran interactivos. Los operadores de computadores configuraban (programaban) las máquinas para leer grandes volúmenes de datos (por ejemplo, datos bancarios y transacciones de los clientes), y un computador central procesaba cada entrada y generaba la salida adecuada, y la almacenaba de nuevo en el computador. Los únicos usuarios que interactuaban con los computadores eran profesionales del área de computación e ingenieros de otras disciplinas. Esto cambió de manera disruptiva con la aparición de la informática personal a principio de los años 80.

Debido a la evolución tecnológica, la reducción de costos y ampliación del ancho de banda en las telecomunicaciones, muchas de las actividades cotidianas que realizan usuarios de distintas edades, distintas competencias técnicas, y bagaje cultural, se realizan con sistemas interactivos.

Un sistema interactivo involucra a los usuarios de manera directa, es decir el usuario y el dispositivo informático intercambian información de manera frecuente y dinámica, a través de distintos canales y modos.

Un usuario, frente a un sistema interactivo, generalmente lleva a cabo el siguiente proceso (Norman, 2013):

- a. Determina el objetivo que desea alcanzar. Es decir, define una intención.
- b. Convierte la intención en una acción. Activa alguna tarea esperando un cambio de estado en el sistema.
- c. Percibe el cambio originado por su acción.
- d. Evalúa si ese cambio es el esperado, respecto al objetivo que quiere alcanzar.

Dado el proceso anterior, puede ocurrir que alguna “variable” no corresponda con lo requerido. Una forma de entender la naturaleza de la interacción entre un usuario y un sistema interactivo es mediante el Modelo de Ejecución/Evaluación de Don Norman (Norman, 2013), que se muestra en la Figura 2. Con cada interacción, los usuarios deben superar dos situaciones: comprender el estado actual del sistema y descubrir cómo cambiarlo según sus necesidades. Los diseñadores pueden apoyar los usuarios conociendo estas situaciones y uniéndolas con un modelo conceptual transparente (Whitenton, 2018).

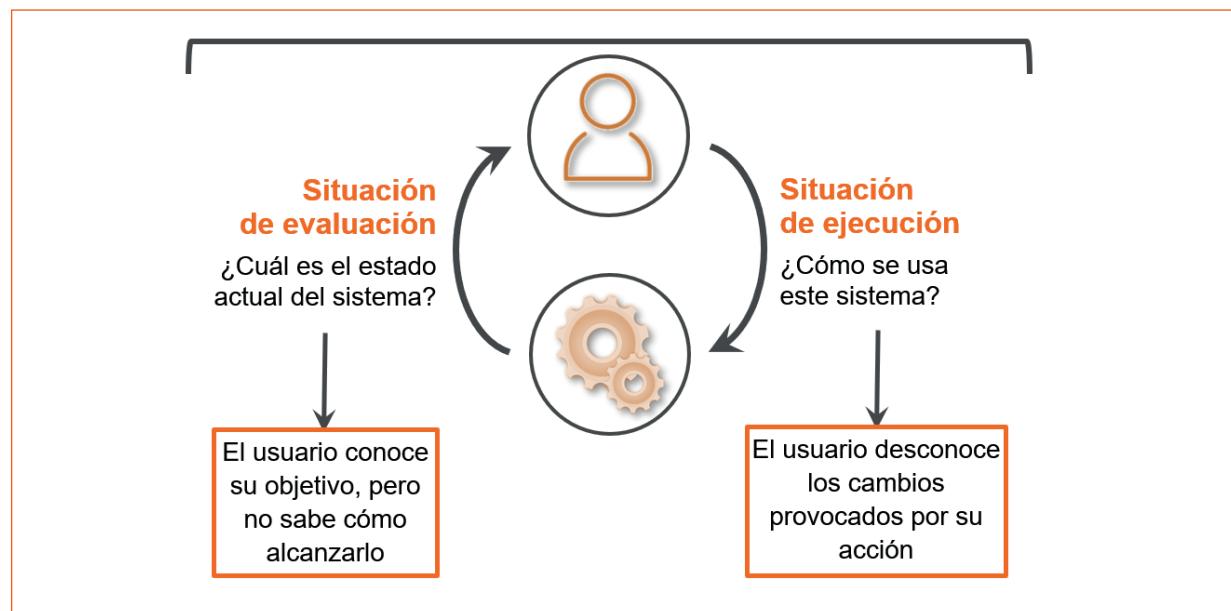


Figura 2. Modelo de Ejecución/Evaluación de Don Norman (Whitenton, 2018). Adaptado de <https://www.nngroup.com/articles/two-ux-gulfs-evaluation-execution/>

La naturaleza de los sistemas interactivos está condicionada por cómo se diseñan sus funcionalidades y tareas, el modelo del intercambio de información (estado actual vs estado final de la interacción), la situación o contexto de uso, los modelos mentales de los usuarios, la “visualización” de un cambio de estado, la sintaxis del “comando” del usuario para desencadenar una acción, entre otros aspectos.

La interpretación de un cambio de estado requiere esfuerzo, y la mayoría de las personas intentan minimizar este esfuerzo confiando en un **modelo mental** para comprender un sistema. **Un modelo mental es una teoría de cómo funciona un sistema, qué significan sus señales y cuáles serán los resultados de las diferentes acciones del usuario.** Para ahorrar tiempo, la mayoría de las personas confían en sus experiencias pasadas para construir rápidamente modelos mentales para nuevos sistemas.

Otro aspecto técnico al cual está ligado la interacción humano-máquina (IxD) es el concepto de **usabilidad**. Este concepto se articuló originalmente de forma algo ligera asociándolo a características tales como "es fácil de aprender, es fácil de usar".

La simplicidad detrás de esta caracterización le dio a la disciplina IxD una identidad innovadora y destacada tanto en informática como en el desarrollo de otros productos interactivos. Sirve para influir en el desarrollo de la ciencia y la tecnología informática de manera más amplia y efectiva. Dentro de la disciplina IxD, el concepto de usabilidad se ha reorganizado y reconstruido casi continuamente, y se ha vuelto cada vez más rico y a la vez también problemático. La usabilidad (en estos tiempos) incluye cualidades como diversión, bienestar, eficacia colectiva, estética adecuada, creatividad mejorada, flujo, apoyo para el desarrollo humano y otros.

El diseño de interacción es fundamental en muchas disciplinas, campos y enfoques relacionados con la investigación y el diseño de sistemas informáticos para las personas, así como distintos tipos de aparatos, máquinas y dispositivos. Se lleva a cabo idealmente por equipos multidisciplinarios.

Un ejemplo interesante donde se explotan muchas formas de interacción puede visualizarse en el video **I.D. VIZZION Concept** de la empresa Volkswagen (Volkswagen, 2018) que presenta un sistema innovador para sus "vehículos", en simbiosis con unos lentes que integran **realidad aumentada** (de ahí su nombre comercial VIZZION similar a *Vision*, en alemán) como se muestra en la Figura 3. Incluye un asistente virtual, con quien los pasajeros se comunican mediante comandos de voz y gestos, "conoce" las preferencias personales de sus huéspedes, y se adapta a cada uno de ellos.



Figura 3. Concepto del sistema I.D. VIZZION (Volkswagen). Elaboración propia a partir de imágenes recuperadas de <https://www.youtube.com/watch?v=AyihacfLto&feature=youtu.be>

► I.D. VIZZION Concept

<https://www.youtube.com/watch?v=AyihacfLto&feature=youtu.be>

El equipo de diseño y desarrollo estuvo formado por especialistas de interacción, de inteligencia artificial, ingenieros de distintas áreas, sociólogos, antropólogos, gerentes de productos, entre otros.

Este tema presenta una panorámica sobre la interacción humano-computador, un marco de referencia en el cual se basa la disciplina, las dimensiones y tipos de interacción.

2.1. Conceptos preliminares

El diseño de interacción (IxD) define la estructura y el comportamiento de los sistemas interactivos. Los diseñadores de interacción se esfuerzan por crear relaciones significativas entre las personas y los productos y servicios que utilizan, desde computadores hasta dispositivos móviles, electrodomésticos y más (IXDA, 2020).

La interacción humano-computador se fundamenta en un marco de referencia que se muestra en la Figura 4, donde se entrelazan distintos conceptos bajo el paraguas “Usos y contextos” de dispositivos (en este caso un computador o dispositivos similares).

Se observa que se basan en los entornos sociales, laborales y organizacionales (1), donde “vive” la solución informática, así como las áreas de aplicación y las tareas (2). Esto es un hecho fundamental pues dirige el proceso: el diseñador debe generar una solución asociada al contexto de uso, y por lo tanto se identifican los tipos de usuarios y sus respectivas competencias y capacidades para poder realizar las actividades y tareas; la aplicación debe ajustarse al aprendizaje humano, la adaptabilidad del sistema u otras estrategias (3).

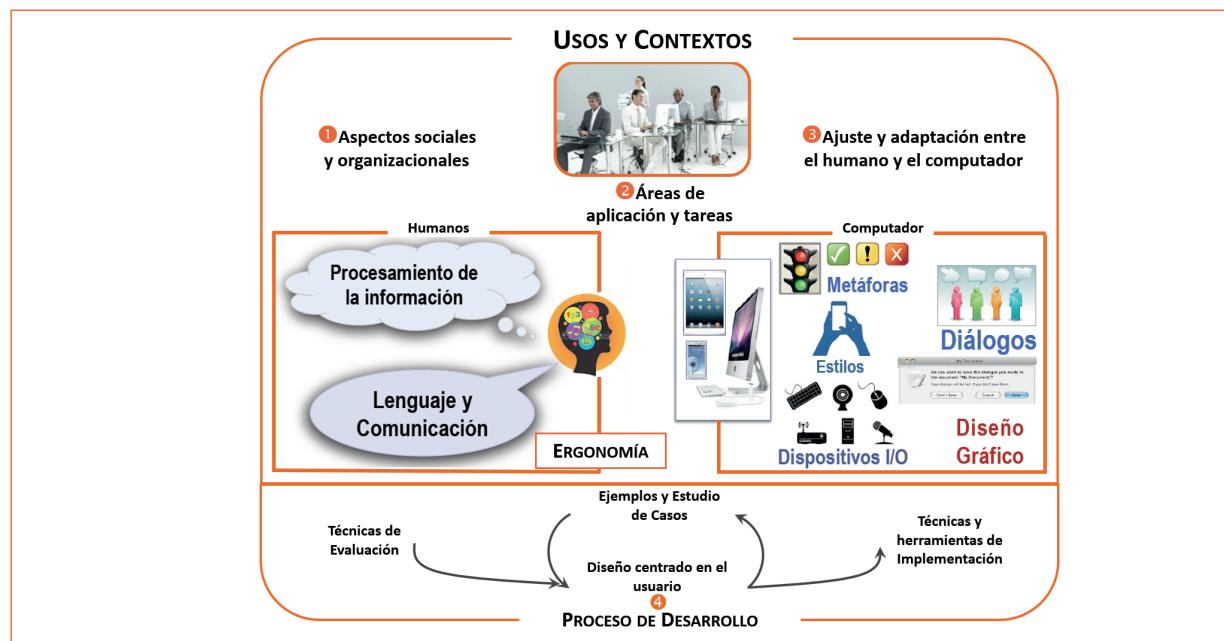


Figura 4. Marco de referencia para el estudio de la Interacción Humano-Computador. Adaptado de (ACM SIGCHI, 1992). Imágenes recuperadas de <https://pixabay.com> y <https://www.apple.com>

Además del uso y el entorno, **en el lado humano**, hay que tener en cuenta cómo es el procesamiento de la información, la comunicación y las características físicas de los usuarios (la ergonomía, sus capacidades cognitivas, motoras y sensoriales).

En el lado del computador/dispositivos, se ha desarrollado una variedad de tecnologías para apoyar la interacción con los humanos: los dispositivos de entrada y salida “conectan” el humano y los computadores. Los dispositivos y otros objetos se utilizan en una serie de estrategias para organizar un “diálogo” y mejorar la mediación: crear las interfaces de usuario, donde se implementan estilos y paradigmas distintos (como el uso de metáforas), se elabora la presentación de la información y su visualización (por ejemplo, elementos gráficos consistentes y coherentes), y se consideran distintos modos y canales de comunicación (sentidos).

Finalmente, está el proceso de desarrollo, el cual considera el diseño centrado en el usuario, (4) que incorpora el diseño de la interacción y de la interfaz de usuario, las técnicas y herramientas para implementarlos, las técnicas para evaluarlos (en especial la **usabilidad** y la **accesibilidad**) y una serie de casos de estudio. Cada uno de estos componentes del proceso de desarrollo está vinculado con los demás en relación de la influencia recíproca por la cual las elecciones realizadas en un área impactan sobre las elecciones y las opciones disponibles en las demás.

2.2. Arquitectura de un sistema interactivo

Un sistema de *software* puede describirse como un conjunto de componentes específicos. **Cuando se diseña la arquitectura de un sistema interactivo, el objetivo primordial es separar la interacción del usuario de la abstracción de datos y el procesamiento de esos datos.** En rasgos generales, la “arquitectura” de un sistema interactivo se muestra en la Figura 5.

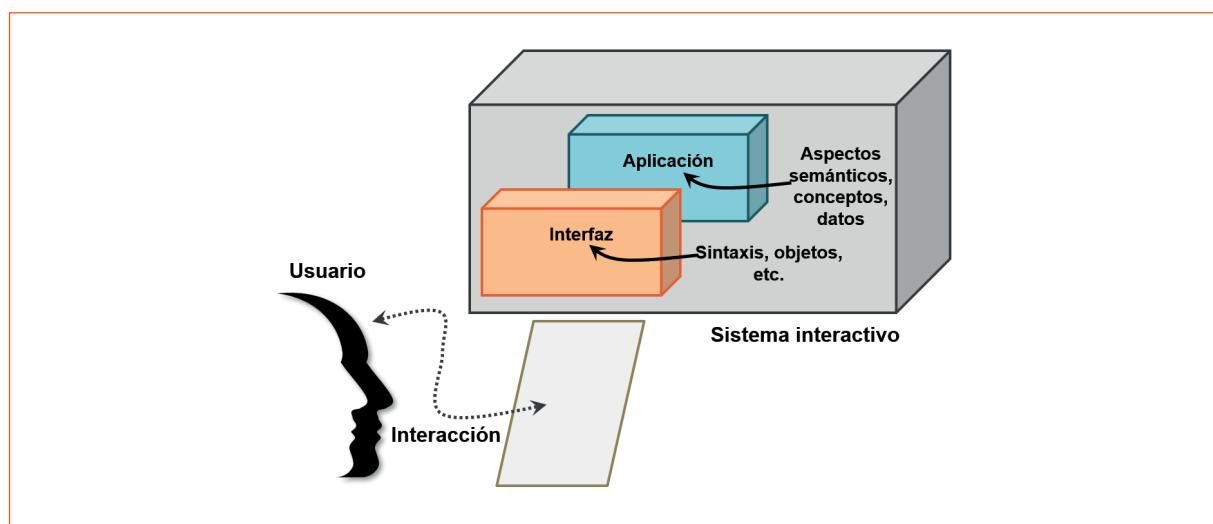


Figura 5. Componentes de una “arquitectura” genérica de un sistema interactivo. Adaptado de Orozco (2001)

A especificar la arquitectura de tales sistemas, el desafío es mantener el núcleo funcional independiente de la interfaz de usuario. El núcleo de los sistemas interactivos se basa en los requisitos funcionales del sistema y, por lo general, permanece estable. Sin embargo, las interfaces de usuario a menudo están sujetas a cambios y adaptaciones en el tiempo (ej. por el cambio de plataforma de despliegue, las

aplicaciones *responsive*). Por ejemplo, es posible que los sistemas deban admitir diferentes estándares de interfaz de usuario, metáforas, apariencia visual y estilos específicos del usuario, o interfaces que deben ajustarse para procesos comerciales de una empresa, a tipos de usuarios muy específicos. Esto requiere de **patrones arquitectónicos** que admitan la adaptación de partes de la interfaz de usuario sin causar efectos importantes a la funcionalidad específica de la aplicación y al modelo de datos subyacente al *software* (Bushman, Meunier, Rohnert y Sommerlad, 2001).

La arquitectura de un *software* interactivo descompone el sistema en tres particiones principales:

- a. El **modelo** contiene la funcionalidad central y los datos (conocimiento del dominio, aspectos semánticos y datos). **Es el componente central del patrón arquitectónico ya que administra directamente los datos, la lógica y las restricciones de una aplicación.**
- b. La **vista** muestra información al usuario. Se puede utilizar para representar cualquier salida de información en forma gráfica, como un diagrama o un gráfico. Las vistas solicitan información de su modelo y generan una representación visual para el usuario.
- c. El **controlador** gestiona la entrada del usuario. En esencia el controlador acepta una entrada y la convierte en comandos para modificar el modelo o actualizar la vista.

Uno de los patrones que proporcionan esta organización estructural fundamental para un *software* interactivo es el patrón Modelo-Vista-Controlador (MVC).

Las vistas y el controlador juntos “forman/diseñan” la interfaz de usuario. Un mecanismo de propagación de cambios garantiza la coherencia entre la interfaz de usuario y el modelo. El patrón MVC proporciona, probablemente, la organización arquitectónica más conocida para sistemas de *software* interactivos. En la Figura 6 se muestra el esquema de este patrón arquitectónico.

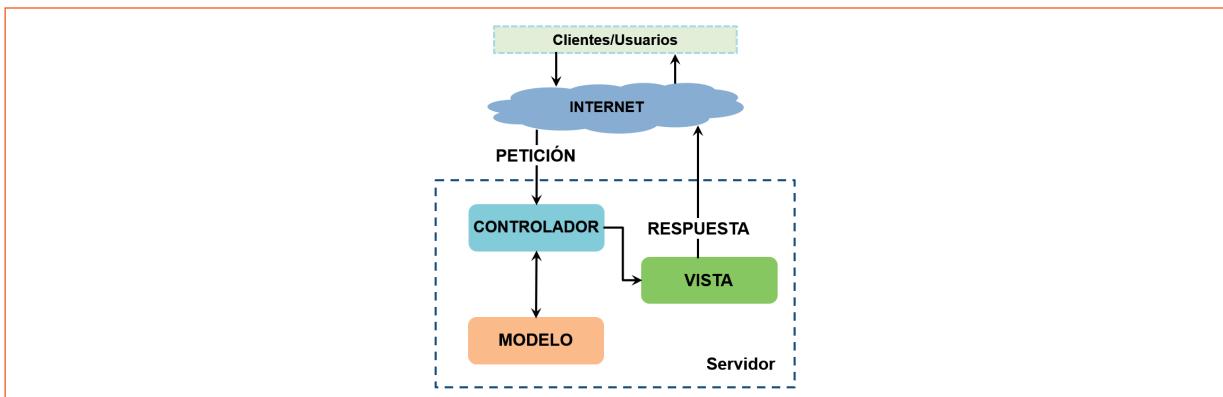


Figura 6. Esquema del patrón arquitectónico MVC. Adaptado de (Madeja, 2019). Recuperado de <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/122>

2.3. Las dimensiones de la interacción

En términos comunicacionales y operativos, **la interacción es el intercambio de información entre un usuario y un sistema**. El objetivo primordial al diseñar la interacción es facilitarle al usuario la activación de un evento. Este proceso tiene asociado un **lenguaje** con su respectiva sintaxis y semántica.

El intercambio se sustenta en palabras, movimientos (expresión corporal), sonido (expresión oral y acústica), visualización y percepción de espacio y tiempo, de forma que pueda mejorarse la comunicación y que el diálogo sea útil y flexible. A esto se le llama **lenguaje multimedial** (por su capacidad expresiva en distintos formatos y medios) y **multimodal** pues hace uso de distintos canales (los sentidos).

Al crear sistemas interactivos que explotan el uso de estos “lenguajes” se produce un impacto positivo en el usuario al considerar sus capacidades motoras y sensoriales, la percepción, y se mejora la experiencia del usuario. Con este lenguaje se comunica el contexto, las acciones y tareas disponibles en el sistema. **El problema reside en cómo presentar el diálogo entre el usuario y el sistema, describir en forma unívoca la comunicación en ambos sentidos, sin importar el modo (canal) y el medio utilizado.**

La especificación de este lenguaje generalmente va acompañada de gramáticas, alfabetos, diagramas de transición de estados (la transición entre tareas y entre acciones), entre otros.

Este lenguaje comprende una variedad de **objetos** que pertenecen a distintas dimensiones que se describen a continuación (Interaction Design Foundation, 2019), (Bianchini, 2020):

- Las **palabras**, en especial aquellas utilizadas cuando se presentan objetos para desencadenar acciones: el etiquetamiento de los campos en una forma o de un botón. Todas deben tener significado y no deben ser ambiguas. Su utilidad es la de **comunicar información suficiente y necesaria a los usuarios sin cargarlos cognitivamente**.
- Las **representaciones visuales**, se refieren a elementos gráficos como imágenes, tipografía e íconos que los usuarios disponen para comprender cómo interactuar. Al considerar elementos orientados a lo visual, las **imágenes son muy valoradas por los usuarios**, porque una imagen “contiene muchas palabras”, una historia, un hecho. Esto es un elemento útil, vista la tendencia de los usuarios de carecer de paciencia y utilizar el objeto o sistema sin mayores ayudas o “manuales”, mejorando de esta manera su productividad.
- Los **objetos físicos** y el **espacio** hacen referencia a preguntas como:
 - a. **Respecto a los objetos:** ¿A través de qué objetos físicos interactúan los usuarios con un producto? ¿Es un computador portátil, con un ratón o panel táctil, o es un teléfono inteligente que se controla con los dedos del usuario? ¿Tiene asociado algún dispositivo para luego utilizar interfaces hápticas? ¿Se le proveerá de sensores?
 - b. **Respecto al espacio**, se deben contestar preguntas como: ¿En qué tipo de espacio físico interactuará el usuario? ¿Está el usuario conduciendo un automóvil mientras usa la aplicación en un teléfono inteligente, o estará sentado en un escritorio mientras navega un sitio web? Todo esto afecta la interacción entre el usuario y el sistema. **El espacio tiene que ver con el contexto y es muy importante para decidir cómo debe verse un producto, de la misma manera que lo influye el tamaño promedio de una mano humana** (la ergonomía y la antropología dan muchas pistas para resolver estos asuntos).

- El **tiempo**. Si bien esta dimensión puede parecer un poco abstracta, **se refiere principalmente a los medios que cambian con el tiempo (animación, videos, sonidos)**. Muchos usuarios son imprevisibles respecto al tiempo de espera e incertidumbre sobre el resultado de una acción, como se evidencia en el Modelo de Ejecución/Evaluación de Don Norman (Figura 2).

El movimiento y los sonidos juegan un papel crucial en la retroalimentación visual y sonora de las interacciones de los usuarios. También es preocupante la cantidad de tiempo que un usuario pasa interactuando con el producto, y algunas preguntas a plantearse son: ¿Pueden los usuarios seguir el progreso de un proceso o reanudar su interacción algún tiempo después? ¿Cómo se le indica al usuario que su proceso lleva un “tiempo” para ejecutarse, o le falta una cantidad de tiempo para terminar? En una era saturada de información, donde los usuarios pueden sentirse tan necesitados de tiempo y con la posibilidad de cargarse cognitivamente con los datos, es absolutamente vital **comprender cuánto tiempo pasan en sus tareas a través de un dispositivo al utilizar un sistema o herramienta**.

- El **comportamiento**. Esto incluye aquellos mecanismos/instrumentos de un producto e involucra dos preguntas fundamentales: ¿Cómo realizan los usuarios las acciones en el sitio web o en un sistema informático de uso específico? Y, ¿cómo operan los usuarios el producto? En otras palabras, esta dimensión trata el “cómo” las dimensiones anteriores definen e influencian las interacciones que un usuario debería tener con un producto. También incluye aquellas reacciones, tales como respuestas emocionales o comentarios, de los usuarios acerca del producto. **Si bien las primeras cuatro dimensiones son vitales por derecho propio, ésta arroja luz sobre un aspecto más profundo del ser humano respecto al comportamiento ante el uso de un objeto.** De aquí se pueden identificar fortalezas, así como cualquier deficiencia. Uno de los objetivos de la disciplina IxD es el diseñar el comportamiento de los usuarios ante el producto (Benyon, 2014).

¿Por qué se deben considerar estas dimensiones?

En el proceso de creación de un producto interactivo, puede ser tentador comenzar en el nivel básico (o elemental) del diseño. Es decir, entrar al diseño de la interfaz (física) y qué estilos (paradigmas) de interacción usar (por ejemplo, basados en menú, llenar formularios, utilizar la voz para los comandos, seleccionar un ícono o introducir un comando textual, etc.). Un problema al tratar de utilizar este enfoque de diseño es que los objetivos críticos de usabilidad y las necesidades del usuario, no habrán sido considerados en forma adecuada.

El diseño de la interacción es el “primer” paso antes de diseñar las interfaces de usuario. En cuestiones concretas, cada una de las preguntas contestadas, y otras más, respecto a las dimensiones nombradas aportan información fundamental de los objetos de **la interfaz, que termina siendo el lenguaje entre el usuario y sistema**.

Durante el diseño de la interacción humano-computador se recurre a acciones primordiales, llamadas **primitivas o tareas de interacción**, asociadas a las dimensiones anteriores, que utilizadas en forma adecuada se alinean a los objetivos de usabilidad y dirigen el diseño de una buena interfaz. En el **Anexo A** se describen cada una de **las primitivas de interacción**.

2.4. Los tipos de interacción

Los tipos de interacción son las distintas formas en las que una persona puede interactuar con un dispositivo o sistema. Conocer e identificar los tipos de interacción permite (y ayuda) a los diseñadores a formular un modelo conceptual antes de iniciar el proceso de diseño de la interfaz.

Éstos no se usan en forma excluyente, ya que en un sistema interactivo se pueden explotar o combinar varios tipos de interacción posibles a la vez. Los tipos de interacción son los siguientes:

- **Orientado a comandos:** los usuarios “emiten” instrucciones al sistema. Puede llevarlo a cabo mediante la escritura formal de un comando (con sintaxis estricta y palabras claves determinadas). Puede seleccionar un comando disponible en un menú (si el ambiente es orientado a manipulación directa y uso de ventanas, iconos, *mouse*, pantallas táctiles, etc.). Puede utilizar la voz (canal multimodal) si la tecnología subyacente lo permite. Puede utilizar interacción orientadas a gestos (es decir interfaces hapticas), presionar botones, etc.
- **Orientado al diálogo:** la interacción, siempre, es un diálogo entre el usuario y el sistema. La novedad en este tipo de interacción es, si existe una tecnología (*hardware* y *software*) que lo permita, los usuarios pueden “escribir libremente” preguntas y el sistema “contesta” con contenido textual o “hablado”. Las restricciones son más asociadas al *software*, pues se requiere de un analizador de texto (léxico, sintaxis y semántica), contexto y demás. La pregunta debe seguir una sintaxis muy específica, su respectiva semántica, etc. El lenguaje natural es muy ambiguo.
- **Orientado a la manipulación:** los usuarios interactúan con objetos en un espacio virtual o físico manipulándolos (abriendo, sosteniendo, cerrando, colocando, ejerciendo fuerza o presión, etc.). No confundir este tipo de interacción con el paradigma de manipulación directa (ver Anexo B).
- **Orientado a la exploración:** los usuarios se mueven a través de un entorno virtual o un espacio físico. En este caso también se habla de tecnología adicional para este tipo de interacción como cascos, lentes para imágenes 3D, visores, sensores, tecnologías de realidad virtual y de realidad aumentada.
- **Orientado a la respuesta:** los usuarios son pasivos, y es el sistema el que desencadena la interacción y el usuario elige si responder o no. Está asociada al concepto de tecnología proactiva. Por ejemplo, la tecnología basada en la ubicación móvil puede alertar a las personas sobre puntos de interés.

Un ejemplo es Netflix (Netflix Inc., 2020) empresa fundada en 1997. Su servicio principal es la distribución de contenidos audiovisuales a través de una plataforma en línea (*streaming*) o servicio de VOD (*Video On Demand*) por retransmisión en directo. La aplicación realiza un seguimiento de la cantidad de tiempo que un usuario ha pasado viendo un contenido, y luego éste detiene su actividad de visualización en un momento determinado. La aplicación desencadena la interacción **para preguntar al cliente si desea continuar viendo dicho contenido.**

Otro ejemplo de producto con interacción orientado a la respuesta son los monitores del estado físico de un usuario como los de Fitbit (Healthy Metrics Research, Inc, 2020). El producto notifica al usuario el andamiento de varios hitos relacionados con el estado físico a lo largo del tiempo. Por defecto, los usuarios de este tipo de producto **reciben una notificación indicando que ha logrado realizar los 10.000 pasos en un día, sin ninguna iniciación por parte del usuario.** Este continuará con su día como de costumbre, y el dispositivo vibrará para decirle que ha cumplido cada uno de sus objetivos.

Los tipos de interacción están agrupados de forma que hacen referencia a los paradigmas más conocidos de interacción. **Un paradigma se refiere a un enfoque general que ha sido adoptado por una comunidad de investigadores y diseñadores para llevar a cabo su trabajo en términos de suposiciones, conceptos, valores y prácticas compartidas.** Se recomienda revisar el **Anexo B**, en cual se describen **los paradigmas de interacción** más conocidos y utilizados hasta estos días.

Tema 3. El diseño centrado en el usuario

"El modelo del usuario no es compatible con el modelo del ingeniero, por lo que tanto el usuario como el computador 'hacen las cosas', pero no necesariamente hacen las 'cosas correctas' desde el punto de vista del otro. Luego se produce una secuencia de malentendidos, que terminará en frustración para el usuario".

(Timblebay, 2018, p.13)

El término "diseño" se refiere tanto al proceso creativo de especificar algo nuevo como a las representaciones que se producen durante el proceso. Por ejemplo, para diseñar un sitio web, un diseñador producirá y evaluará varios diseños: el diseño de la interacción (asociado a cada una de las funcionalidades que brindará el sitio), el de la interfaz, la estructura de la información, la estructura de la navegación entre sus páginas (o el flujo entre las distintas funcionalidades), la apariencia, entre otras cosas (Benyon, 2014).

"El diseño rara vez es un proceso sencillo y generalmente implica mucha iteración, el descubrimiento y la exploración de requisitos [lo que el sistema debe hacer y las cualidades que debe tener], y las soluciones de diseño. Hay muchas definiciones de 'diseño'. La mayoría de las definiciones reconocen que tanto el problema como la solución deben evolucionar durante el proceso de diseño; rara vez se puede especificar completamente una idea, o producto, antes de que se haya realizado un trabajo de diseño".

(Benyon, 2014, p.11)

La gran variedad de interacciones en las que se “participa” en un sistema y las interfaces que se utilizan ofrecen un desafío aunado a la velocidad de los cambios tecnológicos y la aparición de nuevos paradigmas. Cada vez más diseñadores tienen que lidiar con el tema de las **partes interesadas** (en las cuales está el usuario final) que llevan a cabo múltiples interacciones con diferentes dispositivos en paralelo (Benyon, 2014).

Los investigadores en el área de IxD han insistido en la necesidad de utilizar el enfoque (también una filosofía) de **Diseño Centrado en el Usuario (DCU)**. El DCU se enfoca en los usuarios, sus necesidades y objetivos, y no solo en la tecnología que se requiere en el desarrollo de productos. Como consecuencia, un sistema bien diseñado aprovechará al máximo la habilidad y el juicio humano, será relevante para la actividad en cuestión y apoyará al usuario.

Norman (2013, p. 9), en su libro *The Design of Everyday Things Revised and Expansion Edition*, respecto al uso de productos interactivos, indica que:

“El diseño centrado en el usuario, es una filosofía basada en las necesidades e intereses del usuario, con énfasis en hacer que los productos sean utilizables y comprensibles. ... El diseño debería:

- *Facilitar [al usuario] la determinación de qué acciones son posibles en cualquier momento.*
- *Hacer visibles [al usuario] las cosas, incluido el modelo conceptual del sistema, las acciones alternativas y los resultados de las acciones.*
- *Facilitar [al usuario] la evaluación del estado actual del sistema.*
- *[El usuario debe] seguir los mapeos naturales entre las intenciones y las acciones requeridas, entre acciones y el efecto resultante, y entre la información visible y la interpretación del estado del sistema.”*

En el estándar internacional ISO 9241-210 (*International Organization for Standardization*, 2019) se define: “El diseño centrado en el ser humano [usuario] es un enfoque para el desarrollo de sistemas interactivos que tiene como objetivo hacer que los sistemas sean utilizables y útiles enfocándose en los usuarios, sus necesidades y requisitos, y aplicando factores/ergonomía humana y conocimientos y técnicas de usabilidad. Este enfoque mejora la efectividad y la eficiencia, ... y contrarresta los posibles efectos adversos del uso en la salud humana, la seguridad y el rendimiento”.



ISO 9241-210

<https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-2:v1:en>

Los principios del DCU para el proceso de diseño de sistemas interactivos (y otros productos) son:

- a. Un diseño se basa en una comprensión explícita de los usuarios, tareas y contextos de uso.
- b. Los usuarios y partes interesadas participan en todo el diseño y desarrollo.
- c. El diseño es impulsado y refinado por una evaluación centrada en el usuario.
- d. El proceso es iterativo.

- e. El diseño aborda toda la experiencia del usuario.
- f. El equipo de diseño incluye habilidades y perspectivas multidisciplinarias.

En los siguientes contenidos se tratan las actividades y el modelo de proceso del diseño de interacción centrado en el usuario.

3.1. Actividades del proceso de diseño de interacción centrado en el usuario

Considerando las anteriores definiciones sobre el objetivo y el área de interés del diseño centrado en el usuario y sus principios, las actividades que se llevan a cabo en el proceso son las siguientes:

a. Planificar el proceso

El inicio de un proyecto incluye una preparación inicial y estratégica para planificar y organizar todas las tareas de desarrollo. En general hay que conocer el dominio del negocio, sobre el cual se basará el sistema interactivo para definir el problema en miras de obtener una solución alineada a las reglas de ese dominio y objetivos del negocio (cliente).

Se requiere del entendimiento lo más completo del problema, interactuar con las personas que requieren de la solución, de manera que pueda definirse el alcance del sistema.

En esta actividad se debe aclarar la terminología asociada al dominio. Ésta sirve para **establecer el léxico común que ayuda al equipo de desarrollo para comunicar los avances y documentos con los interesados del proyecto**. Uno de los documentos que se crean es un **glosario** que se va enriqueciendo en las iteraciones.

Otro producto de esta actividad es el Documento de Visión y Alcance (DVA) del nuevo sistema interactivo. El DVA recoge los “deseos” del dueño del sistema a desarrollar, aunque debe estar basado en la situación actual en miras de mejorarla con el nuevo sistema. La primera versión de este documento define el alcance a un alto nivel. El alcance detallado (y quizás el documento final) se obtiene después de varias iteraciones con el cliente.

En la Tabla 1 se muestra una plantilla con los contenidos que debe aparecer en un Documento de Visión y Alcance del sistema.

Tabla 1
Plantilla de un documento de Visión y Alcance del sistema

| Documento de Visión y Alcance | |
|-------------------------------|--|
| Apartados del documento | Elementos de cada apartado |
| 1. Requisitos del negocio | 1.1. Antecedentes. Situación actual del negocio 1.2. Oportunidades del negocio 1.3. Objetivos del negocio 1.4. Métricas de éxito 1.5. Declaración de la visión 1.6. Riesgos del negocio 1.7. Suposiciones y dependencias del negocio |
| 2. Alcance y limitaciones | 2.1. Características principales 2.2. Alcance de la primera versión 2.3. Alcance de las siguientes versiones 2.4. Limitaciones y exclusiones |
| 3. Contexto del negocio | 3.1. Perfiles de los interesados 3.2. Prioridades del proyecto 3.3. Consideraciones para el despliegue final |

Nota: Elaboración propia.

b. Investigar y entender los contextos de uso del sistema interactivo

Siguiendo el marco de referencia que se visualiza en la Figura 4, deben conocerse los contextos de uso del sistema (dónde “vivirá” el sistema, así como las áreas de aplicación, afectación y las tareas), pues esto está alineado con las posibles interacciones que llevará a cabo el usuario con el sistema.

Contexto es un término que deriva del vocablo latino *contextus* y se refiere a todo aquello que rodea, ya sea física o simbólicamente, a un acontecimiento. A partir del contexto, por lo tanto, se puede interpretar o entender un hecho.

El **contexto de uso dentro del campo de la experiencia de usuario (UX)** corresponde a las condiciones particulares en las que un producto, bien sea tangible o intangible, será utilizado, atendiendo principalmente a los factores que influirán en su uso y en el grado de satisfacción de unos usuarios específicos (Interaction Design Foundation, 2019b)

El objetivo de esta actividad es recolectar información relevante para definir aspectos fundamentales del problema a resolver, con la intención de obtener una solución alineada a las reglas del dominio (y del cliente que solicita una solución). Conocidos los contextos de uso se pueden identificar los objetivos del sistema interactivo. Esta información se anexa al Documento de Visión y Alcance del sistema (DVA).

Para llevar a cabo esta actividad se requiere del uso de instrumentos para registrar la información inicial obtenida, ya sea a través de reuniones, entrevistas, cuestionarios, talleres, etc., donde intervienen las personas/organizaciones que conocen el dominio, y aquellos que plantearon la idea o necesidad. Otra estrategia utilizada es buscar productos similares a la

situación deseada, de manera que se tenga una línea de base en la cual arrancar la búsqueda de soluciones.

c. Obtener, analizar y especificar los requisitos de usuarios del sistema interactivo

Esta actividad cumple con varios objetivos: obtener los requisitos de los interesados (y usuarios), analizarlos y especificarlos de modo que sean descripciones no ambiguas, completas y no existan conflictos o inconsistencias entre ellos. Incluye comprender al tipo de usuario objetivo y el soporte que el sistema interactivo podría proporcionarle de manera útil y productiva. **Un producto fundamental es la lista de partes interesadas y la identificación de usuarios.**

El cliente sabe lo que quiere, pero es necesario que sus “deseos” sean expresados en forma no ambigua y completa. **La comprensión de las necesidades y requisitos del usuario se logra con la recopilación y el análisis de información, y su “transformación” en requisitos del sistema.**

En el **Tema 4** se detallan aspectos relacionados con el concepto de requisitos de usuario y esta actividad.

d. Producir diseños (modelos) alternativos que satisfagan los requisitos de usuario

La finalidad de esta actividad es proponer ideas para cumplir con los requisitos obtenidos y especificados en la actividad anterior. En el caso de diseñar un producto innovador, Preece et al (2019, p.57) afirman que “... identificar a los usuarios y las tareas representativas para ellos puede ser más difícil. Aquí es donde son valiosos los estudios o los *sprints* rápidos de diseño, que brindan comentarios auténticos de los usuarios sobre ideas iniciales. En lugar de imaginar quién podría querer usar un producto y qué querrían hacer con él, es más efectivo presentarlo y descubrirlo”.

Desde el punto de vista del diseño de interacción, esta actividad se puede dividir en dos partes:

1. Creación del diseño conceptual, o modelos conceptuales: los diseñadores los usan para ayudarse a comprender aspectos del concepto que desean representar. Los diseñadores deben considerar los sistemas, servicios y productos en relación con lo que deben hacer, cómo deben comportarse, cómo se ven y si los usuarios los entenderán de la manera prevista.

Un modelo conceptual representa una abstracción que describe lo que las personas pueden hacer con el sistema y qué conceptos son necesarios para comprender cómo interactuar con él.

2. Creación del diseño concreto, o modelos concretos: en este caso se consideran los detalles del sistema, donde se incluye el tipo de interacción, el diseño de los objetos de interfaz (menús, iconos, metáforas, ventanas de diálogo, otros paradigmas, etc.), y los aspectos sensoriales de la interfaz (gestos, sonidos, imágenes, videos, entre otros) que se utilizarán.

Existen tres tipos fundamentales de modelos concretos:

- El **modelo gráfico**, que es una visualización en 2D o 3D de una idea. Generalmente se realizan bocetos en papel, aunque puede utilizarse algunos programas genéricos para elaborarlos.
- El **modelo físico**, es una versión tangible (reducida o completa) de un objeto con el que se puede interactuar físicamente.
- El **modelo virtual**, es interactivo y foto realista basado en aplicaciones CAD (*Computer Aided Design*) que utilizan modelado de superficies y sólidos. Son conocidos también como maquetas digitales.

Como se ha indicado varias veces, el diseño de interacción implica diseñar el comportamiento del sistema interactivo, para luego diseñar su interfaz. Para que los usuarios puedan evaluar los diseños, se les debe proporcionar algún instrumento para que puedan “interactuar” con esos diseños. Esto se logra creando **prototipos**. Esto no significa que se deba implementar todo el *software* o sistema, de hecho, existen diferentes técnicas de creación de prototipos, y no todas proponen elaborar un *software* totalmente funcional.

En el Tema 5 se tratan aspectos sobre prototipos, las técnicas para su elaboración y distintas recomendaciones de uso.

e. Evaluar el producto desde el punto de vista de sus requisitos (fundamentalmente la usabilidad y la accesibilidad) durante todo el proceso.

Es la actividad para determinar la capacidad del sistema respecto a la usabilidad (y accesibilidad), o el diseño medido en términos de una variedad de criterios según el enfoque utilizado. Este tipo de evaluación no reemplaza las actividades relacionadas con el aseguramiento de la calidad y las pruebas, desde el punto de vista de la Ingeniería de *software*, para garantizar que el producto final responda a su objetivo. Son evaluaciones para medir y mejorar la interacción y el diseño de las interfaces.

En el **Tema 7** se presentan distintas métodos y técnicas de evaluación de usabilidad y accesibilidad.

f. Desplegar la solución

El producto se ha evaluado, refinado y aceptado para su despliegue y uso.

En la Tabla 2 se muestra un resumen de las actividades y los productos generados en durante el proceso de diseño centrado en el usuario.

Tabla 2
Actividades y productos del proceso de Diseño Centrado en el Usuario

| Actividad | Productos (entregables) |
|--|--|
| Planificar | <ol style="list-style-type: none"> 1. Dominio del sistema 2. Glosario de términos 3. Documento de visión y alcance del sistema interactivo (DVA) |
| Investigar y entender | <ol style="list-style-type: none"> 1. Contextos de uso 2. Productos similares identificados 3. Metas de usabilidad y accesibilidad 4. DVA actualizado |
| Obtener, analizar y especificar los requisitos del usuario | <ol style="list-style-type: none"> 1. Lista de usuarios y partes interesadas 2. Descripción de las necesidades de usuario 3. Especificación de los requisitos de usuario <ul style="list-style-type: none"> • Especificación de las interacciones de usuario • Especificación de las interfaces de usuario |
| Producir diseños | <ol style="list-style-type: none"> 1. Diseños conceptuales 2. Diseños concretos 3. Prototipos a distintos niveles de detalle |
| Evaluar los diseños | <ol style="list-style-type: none"> 1. Resultados de la evaluación de usabilidad 2. Resultados de la evaluación de accesibilidad |
| Desplegar la solución | <ol style="list-style-type: none"> 1. Producto evaluado y aprobado para su despliegue |

Nota: Elaboración propia

3.2. Modelo de proceso

Un modelo de proceso a menudo representa una secuencia armónica de actividades, generación de objetos y artefactos (entregables), transformaciones y eventos que incorporan estrategias para lograr la evolución de un sistema cualquiera (en este ámbito, un sistema interactivo) o producto.

Dichos modelos pueden usarse para desarrollar descripciones más precisas y formalizadas de las actividades del ciclo de vida del desarrollo. Su poder surge de la utilización de una notación, sintaxis o semántica lo suficientemente rica, que a menudo puede ser adecuada para el procesamiento computacional.

Estas actividades pueden realizarse utilizando **modelos de proceso** distintos, en base a los esquemas de trabajo de los desarrolladores, las situaciones de desarrollo, el tipo de producto final y la complejidad del producto, entre otras cosas.

En el caso del **DCU** se sigue un **modelo iterativo**, en el cual se hacen evaluaciones y revisiones sobre cada uno de los aspectos obtenidos en las distintas actividades. El modelo de proceso se muestra en la Figura 7, donde se evidencia el “respeto” a los principios enunciados, y se muestran los productos (entregables) que se generan y que se refinan en cada iteración.

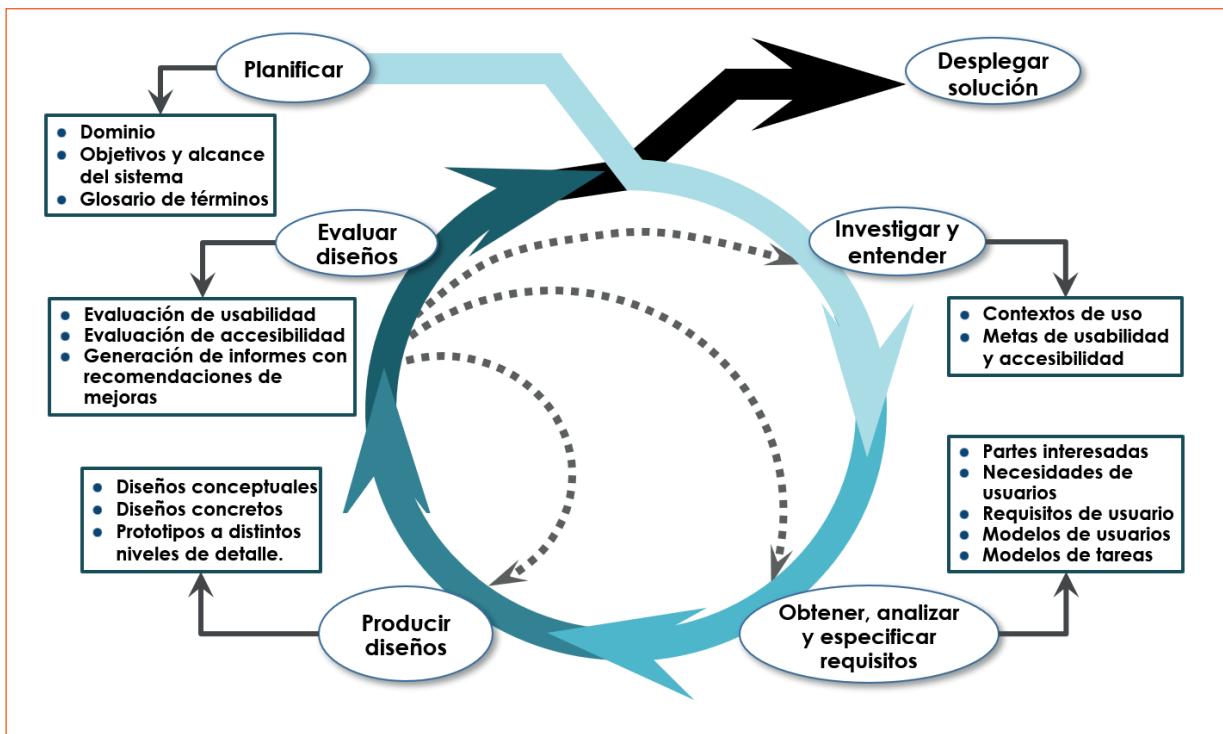


Figura 7. Modelo de proceso del Diseño centrado en el usuario. Elaboración propia

Tema 4. Los requisitos de usuario

"En software, rara vez se cuenta con requisitos que signifiquen algo. Incluso si lo hacemos, la única medida del éxito que importa es si nuestra solución resuelve la cambiante idea que el cliente tiene de lo que es su problema"

(Jeff Atwood - Desarrollador de software)

En este tema se tratarán las tareas y productos de las actividades de obtención, análisis y especificación de requisitos de usuarios para un sistema interactivo. Antes de entrar en detalles sobre las actividades del proceso de diseño, es necesario recordar el concepto de requisito.

Aclarando el concepto de requisito

La literatura está llena de definiciones de requisito, y si bien cada autor utiliza términos similares, en algunos casos el nivel de granularidad de la definición no aclara el panorama. En este contexto se utilizará la definición del 610.12-1900 de IEEE (IEEE, 1998), donde se indica que un **requisito** es:

- a. Una condición o capacidad requerida por un usuario para resolver un problema o alcanzar un objetivo.
- b. Una condición o capacidad que debe poseer un sistema o un componente de un sistema para satisfacer un contrato, estándar, especificación u otro documento formal asociado al sistema.
- c. Una representación documentada de las condiciones y capacidades según lo descrito en a) y en b).

Para (Sommerville y Sawyer, 2003, p. 4), "los requisitos se definen durante las primeras etapas del desarrollo de un sistema como una especificación de lo que se debe implementar. Son descripciones de cómo debe comportarse el sistema (lo que espera el cliente, la funcionalidad disponible para un usuario final, etc.), o de una propiedad o atributo del sistema. Pueden ser restricciones en el proceso de desarrollo del sistema."

Las siguientes afirmaciones son ejemplos de requisitos:

- 1. Una funcionalidad general** de un software, (un deseo) visto a nivel de un usuario.

Ejemplo: *un sistema de reserva de boletos debe incluir la funcionalidad de cancelación de la reserva realizada por un usuario autenticado.*

- 2. Una funcionalidad específica** en un sistema de activación de apertura de bóvedas en un sistema bancario.

Ejemplo: *Las bóvedas de todas las sucursales del banco se deben activar después que el gerente de la sucursal ingrese al área de seguridad de la bóveda.*

- 3. Una propiedad de un sistema** de vigilancia en términos de seguridad.

Ejemplo: *el sistema de vigilancia debe activarse con dos claves de seguridad distintas: la primera debe contener 8 caracteres y la segunda debe contener 5 caracteres.*

- 4. Una restricción** en el desarrollo de un sistema.

Ejemplo: *el sistema de software debe ser desarrollado en PHP versión x.y.*

- 5. El cálculo** del valor de una variable x en una aplicación de visualización de la proyección de figuras geométricas.

Ejemplo: *el valor de x se calculará utilizando la siguiente fórmula:*

$$x = \sin(45 - \alpha) * \cos(\delta * \omega - 90) + \text{Sqrt}(\alpha + \delta), \text{ con } \alpha \in [0,30], \delta \in [0,1], \omega \in [10,30].$$

Los requisitos también son utilizados para establecer tiempos, recursos necesarios y costos de desarrollo.

Los requisitos deben cumplir con distintas características que ayudan a minimizar los posibles errores en la descripción y su posterior especificación, para evitar estimaciones erróneas de presupuesto y tiempo de desarrollo, y utilizarlos en forma adecuada durante el diseño de la solución y en especial cuando se deben validar con los interesados.

La ingeniería de requisitos es el término que se utiliza para cubrir todas las actividades involucradas en el descubrimiento, la documentación, gestión y el mantenimiento del conjunto de requisitos de un sistema de software. El uso del término "ingeniería" implica que deben usarse técnicas sistemáticas y repetibles para garantizar que los requisitos del sistema sean completos, coherentes, relevantes, etc. (Sommerville y Sawyer, 2003).

Los requisitos se pueden clasificar en base a distintos criterios, y generalmente se asocian a distintos aspectos del sistema. En el **Anexo C** se presenta información detallada sobre **las características que debe tener un requisito de calidad, así como una clasificación completa de requisitos y sus respectivas descripciones.**

En la clasificación de requisitos, y considerando el contexto de esta asignatura, se encuentran los requisitos de interacción e interfaz. Esta clase de requisitos es esencial para entender la necesidad del usuario respecto a la interacción asociada a una funcionalidad, el canal de comunicación, las primitivas a utilizar, el paradigma, etc.

Algunos **ejemplos de requisitos de interacción e interfaz** son los siguientes:

1. Respeto a una funcionalidad

La aplicación debe presentar un formulario para el registro del usuario, con los siguientes datos: nombres y apellidos, correo electrónico, nombre usuario (al menos 8 caracteres alfanuméricos) y clave de acceso (al menos 8 caracteres y puede incluir únicamente los caracteres especiales '*', '\$' y '#')

2. Respeto al canal de comunicación

- a)** El dispositivo debe dar las indicaciones de uso mediante mensajes de voz.
- b)** El usuario podrá interactuar con una aplicación mediante gestos sobre la superficie de una pantalla táctil.

3. Respeto a las primitivas de interacción

La aplicación debe ser utilizada con puntero: ya sea a través de un *mouse*, *touchpad* o los cursores del teclado.

4. Respeto a las características de los usuarios

- a)** Para los usuarios de la clase A, la interacción debe ser orientada a la manipulación directa.
- b)** Para los usuarios menores de 5 años, los ítems de los menús deben incluir elementos gráficos.

Los requisitos enumerados anteriormente son aquellos que dirigen el proceso de diseño de la interacción, para luego realizar los diseños de las interfaces y el resto de la solución.

4.1. Obteniendo y analizando los requisitos de usuario

En el desarrollo de *software*, la obtención de requisitos para el diseño de la interacción puede llevarse a cabo para actualizar un sistema existente, o puede tener como objetivo desarrollar uno nuevo, innovador sin precedentes evidentes debido a cambios tecnológicos.

Puede haber un conjunto inicial de requisitos, o el proyecto puede tener que comenzar produciendo un conjunto de requisitos desde cero. Cualquiera sea la situación inicial y el objetivo del proyecto, las

necesidades, los requisitos, aspiraciones (“deseos”) y expectativas de los usuarios deben analizarse, discutirse, refinarse, aclararse y probablemente volverse a analizar. Esto requiere una comprensión de, entre otras cosas, los usuarios y sus capacidades, sus tareas y objetivos actuales, las condiciones bajo las cuales se utilizará el producto y las limitaciones en el rendimiento del producto (Preece et al, 2019).

En varios ejemplos de desarrollo de *software* y de otros tipos de productos, hay evidencias de fracasos por no haber llevado a cabo las tareas necesarias para obtener, analizar y refinar los requisitos de usuarios.

Durante esta actividad inicialmente se identifican los requisitos de los interesados respecto al nuevo sistema. **El producto final es la lista de interesados, el conjunto de Fichas de Requisitos de Usuarios y el Documento Inicial de Requisitos de Usuarios (DIRU).**

Esta actividad se lleva a cabo en el ámbito de desarrollo de sistemas y productos de toda índole. **En esta asignatura, se hace énfasis en todos aquellos requisitos funcionales que deben considerarse para aquellas funcionalidades y características que brindará el sistema (que influyen en el diseño de la interacción e interfaces de usuario), y respecto a requisitos no funcionales como la usabilidad y la accesibilidad.**

La actividad se divide en dos grandes tareas:

- Obtención de requisitos
- Análisis de requisitos

4.1.1. Obtención de requisitos de usuario

La obtención de requisitos es el conjunto de acciones involucradas en el descubrimiento de los requisitos que deberá cumplir el sistema. Es el núcleo del conocimiento de los requisitos. Durante esta tarea se identifican las necesidades, así como las limitaciones obtenidas de los interesados en el desarrollo. Se realiza en forma intercalada con la tarea de análisis de requisitos, para lograr una descripción inicial de esas necesidades.

Las acciones fundamentales son iterativas y se centran en la interacción y comunicación constante con los distintos interesados (clientes, usuarios, etc.), en la identificación de fuentes de información y la descripción (no necesariamente formal) de las necesidades de los interesados y de la organización. Para lograr mejorar las acciones, esta tarea se desarrolla en un ambiente colaborativo y analítico. El producto generado es el **Documento Inicial de Requisitos de Usuarios (DIRU)**.

Para esta tarea, el equipo de desarrollo requiere profundizar en las reglas del negocio, así como los aspectos específicos (**dominio y contextos de uso**) del producto que se desea. Tiene como insumo principal el **Documento de Visión y Alcance (DVA)** generado en la actividad anterior. Otra fuente de información es el conocimiento que tienen los interesados sobre los procesos del negocio y la especificación de sistemas similares que existen en el mercado.

La obtención de requisitos es un aspecto crítico y propenso a errores y la comunicación es de vital importancia entre todos los interlocutores. Cuando los desarrolladores comprenden y asimilan las necesidades recabadas, pueden pensar en soluciones alternativas.

Algunas **dificultades** que surgen durante esta actividad son las siguientes (Sommerville, 2005), (Gottesdiener, 2005), (Ryan, 2014) se muestran en la Tabla 3.

Tabla 3
Dificultades durante la obtención de requisitos

| Ámbito de la dificultad | Descripción |
|---|---|
| Conflicto entre las necesidades de los interesados. | Distintos interesados expresan requisitos/necesidades diferentes. A menudo conocen lo que desean del nuevo sistema en términos muy generales y les resulta difícil expresar sus necesidades o llegan a solicitar características poco viables. El equipo de desarrollo debe considerarlos a todos como fuentes potenciales y en actividades posteriores dedicarse a minimizar los conflictos que pudiesen aparecer. |
| Los interesados dejan de declarar algunos requisitos. | En general expresan (o narran) los requisitos en forma natural y usando un conocimiento implícito de su trabajo. Queda siempre alguna información no expresada adecuadamente. |
| Clases de interesados | La identificación errónea o incompleta del conjunto de interesados. |
| Tiempos | El tiempo dedicado a la captura de requisitos, a veces puede ser elevado por la poca disponibilidad de los interesados. |
| Actores económicos y del negocio pueden ser cambiantes. | Si este escenario ocurre recurrentemente durante la obtención, los interesados y los requisitos pueden cambiar, o aparecen otros. |

Nota: Adaptado de Sommerville (2005), Gottesdiener (2005) y Ryan (2014).

Una planificación adecuada de tareas y el uso de técnicas específicas puede mitigar las dificultades descritas.

En general las acciones que se llevan a cabo durante esta tarea y los productos generados se muestran en la Tabla 4 (Wiegers y Beatty, 2013), (Sommerville, 2005).

Tabla 4
Acciones y productos generados de la obtención de requisitos de usuarios

| Actividad de Obtención de Requisitos | |
|--|--|
| Tarea | Producto |
| Descubrir fuentes de información e identificar potenciales interesados | <ol style="list-style-type: none"> Lista de fuentes de información y su referencia a los documentos Lista de las clases de interesados |
| Planificar la actividad de indagación | Plan de indagación |
| Documentar los requisitos | Fichas de Requisitos de Usuarios para crear el documento de Requisitos de Usuario |

Nota. Adaptado de Wiegers K. y Beatty K. (2013) y Sommerville I. (2005).

A continuación, se describen cada una de las acciones y sus productos:

A. Descubrir fuentes de información e identificar potenciales involucrados

Esta acción tiene como objetivo la **generación de la lista de fuentes de información con sus respectivas referencias a distintos documentos y la lista de las clases de interesados**. Se analizan las funcionalidades del sistema actual (si existe) mediante la revisión sistemática de toda su documentación, y de los procesos del negocio. Durante esta acción **se detectan aquellos individuos que utilizan el sistema, se identifica su rol en los procesos del negocio y se obtienen las clases de interesados**. Adicionalmente es necesario informarse sobre proveedores de *software* y *hardware*, regulaciones, estándares y otras fuentes que lleven a la obtención de requisitos.

Respecto a las fuentes de información

Una fuente fundamental son **las características de cómo interactúan entre sí los otros sistemas de la organización** en miras de detectar aquellos requisitos de interfaces externas (*hardware* y *software*) ya que posiblemente el nuevo sistema tendrá que interactuar con otros productos de la organización. **Conocer estos últimos es de utilidad para descubrir interesados indirectos**. Otra fuente a no despreciar es la descripción y usos de productos similares existentes en el mercado.

Las regulaciones de organismos externos, como reguladores o autoridades de certificación, permisos, etc. son fuente importante de información. Ellos articulan reglamentos, guías y directrices que deben considerarse en el desarrollo de un sistema de *software*.

Algunas **buenas prácticas** para buscar fuentes de información con el fin de profundizar y conocer la problemática del negocio, así como las herramientas que dan soporte a los procesos, se describen en la Tabla 5:

Tabla 5
Buenas prácticas para buscar fuentes de información.

| Buenas Prácticas para buscar Fuentes de Información | |
|---|--|
| Aspectos | Descripción |
| 1. Identificar toda la documentación pertinente sobre la organización y sus servicios, más allá de lo recolectado durante la actividad inicial dedicada a investigar y entender el dominio y los contextos de uso. | Todos los procesos en un negocio tienen alguna descripción (documentación y especificación) en especial todo lo referente a la información gestionada y los procedimientos y flujos de trabajo que probablemente serán considerados en el nuevo sistema (<i>software</i>). Se debe utilizar siempre la última versión verificada. Se debe localizar la documentación que utilizan los usuarios (manuales, guías, descripción de procesos, estándares industriales del dominio del negocio, etc.) que podrían estar en formato digital o físico. |
| 2. Identificar aquellas fuentes externas de información. | Algunas soluciones de <i>software</i> pueden estar disponibles, en condiciones específicas, en el mercado. Una revisión de sus características y funcionalidades, así como su arquitectura y componentes externas, pueden ser de ayuda para la obtención de requisitos. Más específicamente hay que identificar toda la información de otros sistemas con los cuales interactuará el nuevo sistema con el fin de descubrir requisitos externos. Otras fuentes externas son las regulaciones y leyes, provenientes de entes gubernamentales, que influyen en la información que se gestionará en el sistema y cómo se alinearán a las necesidades del negocio sin crear conflictos. |
| 3. Documentar todo lo recuperado que se incluirá en el documento de especificación de requisitos del sistema. | Todo requisito debe referenciar a su origen, es decir debe registrarse el conjunto de fuentes que expresan dicho requisito (Sommerville y Sawyer, 2003). Se deben llevar a cabo todas las operaciones de respaldo de la documentación obtenida y analizada. |

Nota: Elaboración propia

Respecto a los interesados

El proceso de diseño de un sistema interactivo (*software*) se beneficia de los distintos puntos de vista y necesidades que tienen los participantes del proyecto. A los participantes de un proyecto de desarrollo se les conoce como **interesados, partes interesadas o involucrados** (*stakeholders* en inglés).

Según (Wieggers y Beatty, 2013, p. 60), **un interesado es una persona, grupo u organización que participa activamente en un proyecto, puede verse afectada o puede influir tanto en el proceso de desarrollo como en el producto final.** Los interesados pueden ser parte o no del equipo del proyecto, y/o de la organización cliente.

El conjunto de partes interesadas para un producto en particular puede ser más grande que el grupo de usuarios. Incluye a los clientes que solicitan el desarrollo de un producto, los usuarios que interactuarán con el sistema, el equipo de desarrolladores que lo diseñan y lo implementan, las entidades que imponen normas sobre su desarrollo y funcionamiento, etc. (Preece, Rogers y Sharp, 2002) (Preece et al, 2019).

En el conjunto de los interesados, se distingue **el cliente que desea el sistema para su empresa, conoce el contexto y las reglas del negocio**, y es el activador del proceso, pues el nuevo *software* resolverá problemas internos de su organización, para la venta de sus servicios, etc.

Un interesado fundamental es el “usuario” final, a veces “desconocido” o no “disponible”. Los desarrolladores de la solución tienen entre sus retos, adelantar todo lo concerniente al **usuario final del sistema**, ya que en ocasiones éste puede no participar en el proceso.

Parte de los productos de esta acción es la lista de involucrados con sus respectivos perfiles. La información que debe incluir cada perfil de las partes interesadas se muestra en la Tabla 6.

Tabla 6

Documentación de los perfiles de los interesados

| Documentación de los perfiles de interesados | |
|--|--|
| Aspectos | Descripción |
| Rol | Lista de las categorías a la cual pertenece el interesado. Se incluye además si es usuario directo, indirecto, externo, etc. |
| Grado de participación | Medida en la cual el interesado participará en el proceso de obtención de requisitos: <i>plenamente, limitada, indirecta</i> (alguien más puede sustituirlo) |
| Responsabilidades | Descripción breve del rol, respecto al proyecto. |
| Necesidades | Lista de las necesidades, “deseos” y expectativa que tiene con el producto. |
| Criterios de éxito | Descripción de las características o capacidades del producto que el interesado considerada de utilidad para su uso. |
| Preocupaciones | Listado de todas las limitaciones o factores del producto que impidan al interesados a aceptar el producto. |
| Capacidades técnicas | Descripción de las competencias y capacidades técnicas del interesado respecto a la tecnología. |
| Clase de interesado | Categoría a la cual pertenece (inicialmente) el interesado. |
| Entorno laboral | Descripción de las condiciones del entorno de trabajo que pueda afectar el uso del sistema. |

Nota: Elaboración propia.

La incorporación al perfil de los usuarios, de **información relevante sobre sus capacidades físicas, cognitivas y sensoriales, influye en las decisiones de diseño de la interacción y la interfaz ya que afectan directamente los aspectos de usabilidad y accesibilidad como características de calidad de los sistemas interactivos**.

En la Tabla 7 se describen aquellas capacidades que deben considerarse para el diseño de sistemas interactivos.

Tabla 7
Capacidades y aspectos de los usuarios a considerar en el proceso de diseño de sistemas interactivos

| Aspectos | Descripción |
|-----------------------------------|---|
| Habilidades físicas y sensoriales | Determinan en gran medida la adaptación del entorno de trabajo a las características del usuario (tamaño de objetos con el cual van a interactuar, tipo de dispositivos, etc.). Pueden aparecer casos en los que deben considerarse diseños ergonómicos particulares por las limitaciones en movilidad de los usuarios, capacidad visual, verbal, etc. |
| Habilidades cognitivas | Las diferencias en la capacidad cognitiva (razonamiento y conocimiento previo) se debe al grado de experiencia del usuario (tanto de su propio trabajo como del uso del dispositivos). De esta manera, existe una gran variedad de usuarios desde los expertos a los novedos, usuarios que diariamente utilizan los dispositivos o aquellos ocasionales, etc. |
| Diferencias de personalidad | Las diferencias en la personalidad puede provocar alteraciones en la propia comunicación. Con las nuevas tecnologías asociadas a interacción asistida, debe considerarse que el “asistente” virtual debe crear empatía con el usuario, y lograr maximizar la efectividad del diálogo. |
| Diferenciación cultural | Los entornos socio culturales donde “habitan” los usuarios pueden llegar a ser muy disímiles. El lenguaje utilizado, el idioma, las expresiones, la simbología, y hasta los colores pueden crear “ruido” durante el uso de los dispositivos. |

Nota: Elaboración propia.

A continuación, se presenta una lista de verificación (checklist) que puede facilitar la identificación de posibles interesados que no son evidentes al inicio del proyecto (Westfall, 2006, p. 7):

- ¿Qué tipo de personas utilizará el sistema interactivo?
- ¿Qué actividades comerciales (o no) son compatibles con el sistema interactivo y quién realiza, participa o administra esas actividades?
- ¿Qué trabajo se verá afectado por la introducción del nuevo sistema?
- ¿Quién recibirá los informes, resultados u otra información del sistema a desarrollar?
- Si el sistema falla, ¿quién podría verse afectado?
- ¿A quién se debe evitar el uso del sistema, o el uso de ciertas funciones / datos del sistema?
- ¿Para quién pudiese crear problemas este producto?

La consideración de la existencia de distintos tipos de interesados, más el hecho de la constante comunicación durante la recolección de las funcionalidades y características del futuro producto, es reconocer la necesidad de tener distintas opiniones y vistas (a 360°) del producto y del desarrollo.

Respecto a las técnicas de obtención

Los interesados identificados no siempre estarán disponibles debido a sus ocupaciones y dedicaciones respecto al nuevo proyecto de desarrollo. Las empresas de desarrollo de *software* emplean diversas técnicas para la obtención de información a través de los interesados, pero hay que considerar también las preferencias por un enfoque sobre otros. Un interesado podría describir claramente cómo hace uso del sistema actual (si existe) o de lo que quisiera del nuevo, mientras que en otros casos es mejor observar a los interesados mientras realizan las distintas tareas dentro del proceso de negocio, de manera que se alcance un mismo nivel de comprensión de sus necesidades.

Las técnicas utilizadas para la obtención de requisitos pueden ser:

- 1. Las técnicas de facilitación, que se centran principalmente en descubrir los “deseos” del negocio y de los usuarios.** Es necesario trabajar directamente con los interesados ya que sus requisitos serán seguramente las funcionalidades que se implementarán en el sistema.
- 2. Las técnicas de obtención independiente complementan las descripciones de los requisitos que los interesados expresan y revelan,** ya que el nuevo sistema deberá incluir alguna funcionalidad necesaria que los usuarios finales quizás desconocen. Ambos tipos de técnicas son complementarias: en un momento determinado alguna de ellas servirá para determinar situaciones generales y otras servirán para descubrir detalles más concretos.

Las técnicas utilizadas para la obtención se aplican en todo el espectro de enfoques de desarrollo y la selección de técnicas generalmente se basa en las características del proyecto y los estándares y métodos seguidos por la empresa de desarrollo.

B. Planificar el proceso de obtención de requisitos

La planificación de la tarea de obtención de requisitos es una responsabilidad del analista líder del proyecto. El resultado es un plan de acción con el cual se pueden maximizar las posibilidades de éxito y se motiva a los interesados creándoles expectativas positivas sobre el nuevo sistema y sus aportes. Se establece de esta manera una relación de colaboración entre el equipo de analistas y los interesados del nuevo sistema.

El plan puede considerarse como un contrato donde se presentan los recursos, el calendario de tareas y se definen los entregables de una actividad o tarea específica.

Documentar los requisitos obtenidos

Los productos resultantes de la tarea de obtención de requisitos son documentos. Se deben recoger anotaciones rápidas, apuntes, esquemas, grabaciones, videos, etc., en cada momento y como resultado de las técnicas de indagación utilizadas. Toda la información debe organizarse para generar documentos manejables, que puedan utilizarse para validaciones con los interesados, y que respete ciertos estándares del proceso de requisitos.

Un ejemplo de documentación sobre aspectos relevantes de un requisito se describe en la Tabla 8

Tabla 8
Descripción de los elementos de una ficha de requisitos

| Elementos de una Ficha de Requisitos | |
|--------------------------------------|---|
| Información | Descripción |
| # de Requisito | Número (código) que identifica cada requisito |
| Incidencia | A qué parte del sistema afecta (escribir el cual apartado del documento de requisitos debe figurar) |
| Categoría | Negocio, Funcional (incluyendo sub categoría), No Funcional (incluyendo sub categoría) , Restricción, Interfaz externa, de evolución, etc. |
| Descripción | Frase que describe el requisito en el formato: "El sistema debe ..." "El sistema debería ..." |
| Razón | Aspectos que definen la importancia del requisito |
| Origen | Interesados que solicita el requisito |
| Prioridad | Valoración según interesados. Se puede utilizar el esquema de clasificación MoSCoW (en inglés, <i>M-Must</i> , <i>S-Should</i> , <i>C-Could</i> , <i>W-Won't</i>) → Debe, Debería, Podría, No debe. U otro estándar de la empresa de desarrollo. |
| Dependencia | Lista de requisitos que dependen de él |
| Conflictos | Lista de requisitos que contradicen el requisito |
| Referencias | Documentos y otros materiales necesarios para la comprensión del requisito |
| Historia | Lista de cambios que ha sufrido el requisito |

Nota: Elaboración propia.

Una buena práctica para minimizar errores durante la obtención y para entender la descripción de algunos requisitos (comportamientos y funcionalidades particulares) es utilizar casos de uso y **escenarios**.

Los escenarios y los casos de uso son técnicas (realmente modelos) eficaces para describir requisitos para los distintos puntos de vista de los interesados. Se pueden utilizar conjuntamente con algunos interesados con puntos de vista indirectos, como es el caso en el cual estos usuarios reciben algún resultado del sistema (por ejemplo, un reporte de datos). No se utilizan para la obtención de restricciones y requisitos no funcionales de alto nivel (Pressman, 2010). El conjunto de escenarios y/o casos de uso es una documentación muy técnica de cada requisito, sirve de instrumento para la validación de requisitos con los actores del sistema, y es reutilizable pues se incluye también en el documento de especificación de requisitos.

Al utilizar casos de uso para describir una funcionalidad, parte de la información que se incluye en su descripción son **las precondiciones y las post condiciones**. Las **precondiciones** definen los requisitos previos que deben cumplirse antes que el sistema pueda comenzar a ejecutar el caso de uso (es decir una funcionalidad). El sistema debe poder probar todas las condiciones

previas para ver si es posible continuar con el caso de uso. Las **post condiciones** describen el estado del sistema después de que el caso de uso se ejecutó con éxito. Esta información es necesaria para efectos de minimizar conflictos, incluir nuevos requisitos o desechar algunos existentes.

Desde el punto de vista del usuario, el sistema interactivo debe comunicar en qué estado se encuentra (precondición) y cuáles serán los próximos estados posibles (post condiciones). Estos aspectos están íntimamente relacionados con la retroalimentación que el sistema debe brindar al usuario, luego que este interactúe con el sistema, en cada una de las funcionalidades.

El producto final de la acción de documentación de requisitos es el **Documento Inicial de Requisitos de Usuarios (DIRU)**, que incluye las necesidades obtenidas en forma casi completa y organizada según distintos criterios. Cada empresa de desarrollo define su estándar en la elaboración de una plantilla del DIRU, o se apoya en plantillas disponibles en el mercado.

4.1.2. Análisis y negociación de requisitos de usuario

Esta tarea tiene como objetivo analizar el conjunto de requisitos, con la finalidad de asegurar que todos los interesados entiendan y examinen las descripciones, obtenidas durante la obtención, y minimizar la posibilidad de errores, omisiones, inconsistencias y otras deficiencias.

Esta tarea se alimenta de la información contenida en el documento inicial de requisitos de usuarios (DIRU), que va evolucionando a medida que se avance en el proceso de desarrollo. En esta tarea el DIRU se convierte en el **Documento de Requisitos de Usuario (DRU)**. En las siguientes tareas del proceso, la especificación de requisitos y luego la evaluación, este documento evoluciona, se completa con más detalles, se valida y se convierte en el **Documento de Especificación de Requisitos de Software (ERS)**.

Las acciones que se llevan a cabo, sus descripciones, y los productos que se obtienen en esta tarea se muestran en la Tabla 9.

Tabla 9
Descripción de acciones y productos de la tarea de análisis de requisitos

| Análisis y Negociación de Requisitos | |
|--------------------------------------|---|
| Acciones | Producto |
| Negociar los requisitos | Documento de requisitos acordados por los interesados |
| Priorizar los requisitos | Lista de requisitos ordenados según su prioridad |
| Clasificar los requisitos | <ul style="list-style-type: none"> • Lista de requisitos clasificados en forma multidimensional (facetas y categorías) • Documento de Requisitos de Usuario |

Nota: Elaboración propia.

4.2. Especificando los requisitos de usuario

Durante cada una de las tareas y acciones para lograr requisitos de usuarios se va especificando, en distintos niveles de detalle, muchos aspectos sobre el nuevo *software*. Se genera un conjunto de documentos (entregables) que contienen distintas formas de descripciones de requisitos. A este conjunto se le llama **Documentos de Definición del Sistema**.

El objetivo de esta tarea es lograr la especificación de los requisitos de usuario en forma completa, correcta, no ambigua, consistente, ordenada según algún criterio, y de forma que sean verificables y trazables. El producto de esta tarea es el documento **ERS – Especificaciones de Requisitos de software**.

Desde el punto de vista práctico, generalmente el documento ERS no termina de generarse (completamente) durante esta tarea y para todo el sistema, antes de comenzar el desarrollo, en especial cuando se está siguiendo el modelo de proceso iterativo incremental. Es suficiente con la especificación de requisitos para cada incremento antes de construir una solución para ese incremento. El desarrollo incremental es apropiado cuando se desea obtener cierta funcionalidad en las manos de los usuarios rápidamente. Esta estrategia es muy utilizada en las funcionalidades donde se exige una gran interacción del usuario durante el diseño de la solución.

En la Tabla 10 se muestran las acciones que se llevan a cabo, sus descripciones y los productos obtenidos.

Tabla 10
Descripción de las acciones y productos de la tarea de especificación de requisitos.

| Especificación de Requisitos | |
|---|---|
| Tarea | Producto |
| Definir la plantilla del documento de especificación de requisitos | Plantilla del documento de especificación de requisitos |
| Modelar los requisitos | Modelos de requisitos |
| Incorporar toda la información al documento de especificación de requisitos | Documento de Especificación de Requisitos de Software |

Nota: Elaboración propia.

Existen varias técnicas para especificar requisitos que son utilizadas en varios modelos de desarrollo de la ingeniería de *software*. Una consideración fundamental en la especificación, en el ámbito de sistemas interactivos, es el modelado de los usuarios para entender las tareas que llevará a cabo en el contexto de uso identificado. A continuación, se tratarán algunas técnicas para llevar a cabo esta acción.

4.3. Modelando usuarios y tareas. Enfoques

El modelado del usuario y sus tareas es una actividad fundamental pues es el centro de atención durante el desarrollo de sistemas interactivos. “Un modelo es una representación abstracta de entes

o fenómenos de una realidad en la que se consideran los aspectos relevantes de los mismos y se desechan los menos relevantes, sin que por ello deje de representar significativamente esta realidad" (Olsina, 1999, p. 202).

Preece et al (2019, p.403) afirman que "Dos técnicas que se utilizan comúnmente para aumentar la información sobre requisitos básicos y dar vida a los requisitos son las personas y los escenarios. A menudo utilizados juntos, se complementan entre sí para brindar detalles realistas que permitan al desarrollador explorar las actividades actuales del usuario, el uso futuro de nuevos productos y las visiones futuristas de las nuevas tecnologías. También pueden guiar el desarrollo a lo largo del ciclo de vida del producto".

La creación de modelos descriptivos de los usuarios es una herramienta poderosa para el diseño de interacción. "Debido a que se diseña para los usuarios, es importante que se pueda entender y visualizar los aspectos más destacados de las relaciones entre ellos, lo que quieren, con sus entornos sociales y físicos y, por supuesto, con los productos que esperamos diseñar (Cooper, Reimann, Cronin y Noessel, 2014, p.61). Estos autores proponen el modelo llamado **Personas**.

El conocimiento sobre el comportamiento de grupos de usuarios, es decir cómo "piensan", qué quieren obtener y por qué (sus objetivos y necesidades) es la base fundamental del DCU. Las **personas** (arquetipos compuestos), es decir modelos de usuarios, **no son usuarios reales. Se basan en patrones de comportamiento, y con las personas se puede desarrollar una comprensión de los objetivos de los usuarios verdaderos, en un contexto de uso.** A partir de ellos se puede idear y validar conceptos de diseño.

Si bien las **personas** pueden dar información sobre los posibles usuarios, estos modelos no son suficientes pues además debe conocerse la situación y contexto de uso. Por lo tanto, se utiliza, adicionalmente, el concepto de **escenarios** (Rosson y Carroll, 2002). Ambos modelos enriquecen tanto el proceso de diseño de interacción como el de las interfaces.

Un escenario es una "descripción narrativa informal" (Rosson y Carroll, 2002). **Describe las actividades o tareas humanas en una historia que permite la exploración y discusión de contextos, necesidades y requisitos. No describe necesariamente el uso de software u otro soporte tecnológico utilizado para lograr un objetivo.** El uso del glosario y la redacción realizada por los usuarios significa que los interesados pueden comprender los escenarios y así podrán participar plenamente en el desarrollo.

Al igual que otras estrategias centradas en el usuario, el diseño basado en escenarios cambia el enfoque del trabajo de diseño desde la definición de las operaciones del sistema (es decir, la especificación de requisitos funcionales) hasta la descripción de cómo las personas usarán un sistema para realizar tareas de trabajo y otras actividades (Rosson y Carroll, 2002).

Como bien lo afirma (Preece et al, 2019), "Escribir personas y escenarios puede ser difícil al principio, lo que lleva a un conjunto de narraciones que combinan detalles de la persona con detalles del escenario. Un escenario describe un uso de un producto o un ejemplo de lograr un objetivo, mientras que una persona caracteriza a un usuario típico del producto". En la figura 8 se evidencia esta diferencia gráficamente esta reflexión.

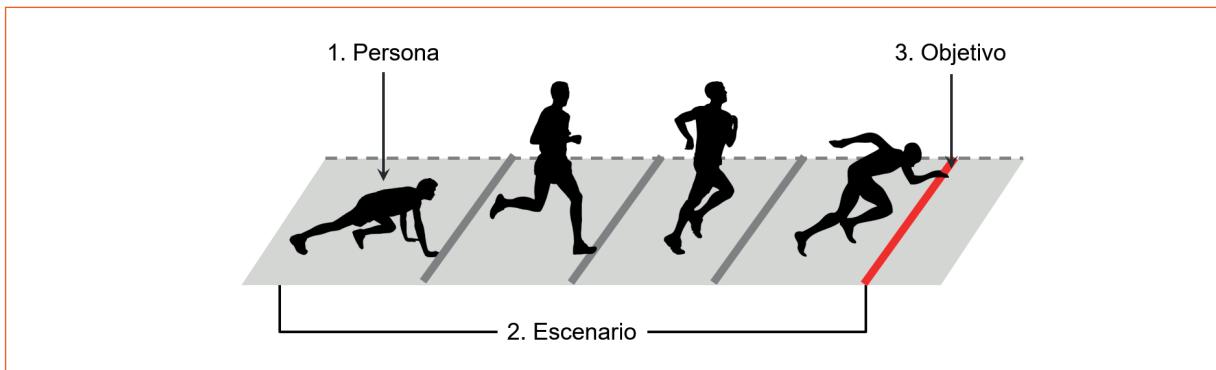


Figura 8. Diferencia entre persona y escenario. Elaboración propia. Concepto tomado de <http://www.smashingmagazine.com/2014/08/06/a-closer-look-at-personas-part-1/>

4.4. Caso de estudio: *MyEnchantedMuseum*. Una vista a algunos documentos generados respecto a los requisitos de usuario

A continuación, se presenta un escenario (hipotético) que se utilizará para recorrer las distintas actividades del proceso de diseño centrado en el usuario y los productos generados.

Un museo muy conocido a nivel mundial, y con sedes en distintas ciudades, utiliza un sistema de aplicaciones y dispositivos para que sus visitantes (reales y virtuales) puedan conocer sus colecciones, obras, autores, etc., y que los visitantes tengan una experiencia especial

La solución actual se llama MyMuseum y está conformado por tres componentes:

- a) *Un sitio web (www.mymuseum.org) que brinda información sobre cada una de las colecciones, los autores y sus obras, las técnicas utilizadas, aspectos históricos, noticias, eventos, contactos, etc. Además, dispone de recorridos virtuales (preestablecidos) para cada una de las colecciones (muestra un mapa de las instalaciones, y el visitante virtual selecciona las salas que desea "visitar"), con la posibilidad de "señalar" y observar "de cerca" las obras de interés en ese recorrido virtual.*
- b) *Un audio tour (DVD Player) con el cual el visitante, que se desplaza por los espacios del museo, escucha las indicaciones para alcanzar las distintas colecciones, es decir se le indica el camino a recorrer dentro del museo, y el usuario puede "conocer" información atinente a las obras. El dispositivo es llamado MyMuseumAudioGuide.*
- c) *Un kiosco multimedia (en un espacio especial dentro del museo). Este sistema brinda información multimedia sobre las distintas colecciones, y el visitante puede acceder a información específica sobre las obras y sus autores. El kiosco es llamado MyMediaMuseum.*

El nuevo curador ha diseñado una encuesta abierta, que los visitantes contestan al salir. El objetivo era obtener las opiniones sobre las instalaciones, los servicios y otros detalles atinentes.

Una sorpresa interesante, resultado de las encuestas, fue que los visitantes no se sentían muy a gusto con el dispositivo MyMuseumAudioGuide, pues un porcentaje alto de ellos (+30%) considera que no quieren seguir un recorrido preestablecido, sino que les gustaría moverse en el museo a su necesidad e interés, y que “alguien o algo” les explique lo que estén observando si así lo desean. Por lo tanto, el dispositivo MyMuseumAudioGuide, que usan en la actualidad, no resuelve totalmente esa necesidad de los usuarios.

Otra sorpresa, también resultado de la encuesta, fue que el MyMediaMuseum no llena las expectativas de algunos visitantes, pues en especial los niños y adolescentes esperan hacer “cosas” más creativas con la información visualizada (en especial las obras). Una de las sugerencias es que el MyMediaMuseum brindara la posibilidad de intervenir una obra: por ejemplo, cambiar colores y forma (si es una pintura), cambiar la forma (si es una escultura), entre otros, y finalmente ver cómo quedaría luego de esa intervención.

Respecto al sitio web, algunos visitantes (+25%) quieren que el recorrido hecho virtualmente pueda ser registrado de alguna manera, y que, al visitar físicamente el museo, se le “recordara” como realizar esos recorridos virtuales en la realidad de los espacios del museo.

El museo, y su curador, están interesados en una solución a sus problemas. Es decir, crear una nueva plataforma interactiva que brinde una nueva experiencia a todo tipo de visitantes: niños (K-12), adolescentes, adultos y tercera edad, con o sin discapacidades motoras y sensoriales.

Existen algunas limitaciones respecto al presupuesto y al tiempo de desarrollo. El Museo tiene disponible 500.000 € para el desarrollo del proyecto y quieren la solución activa en 6 meses, con motivo de la reinauguración de sus sedes a nivel internacional.

Tomando como referencia la situación descrita, luego de llevar a cabo la actividad **Investigar y entender los contextos y usos**, se obtuvo el Documento de Visión y Alcance del sistema, el glosario asociado a la situación planteada, y el dominio y contextos de uso de la nueva aplicación.

Tanto el equipo de desarrollo como la organización conciben un nuevo concepto para la solución al problema y deciden llamarlo **MyEnchantedMuseum – Interactive Complex**. Este concepto engloba cada uno de los componentes (disímiles, pero muy vinculados) para que los usuarios tengan una visión más completa de los servicios del museo. Para mejorar la lectura, se le llamará únicamente **MyEnchantedMuseum**.

Parte de lo generado respecto al glosario se muestra en la Tabla 11, y la información sobre contextos de uso y dominio se muestra en la Tabla 12.

Tabla 11

Parte del Glosario asociado al MyEnchantedMuseum – Interactive Complex

| Término | Descripción |
|--|---|
| Colección | Un conjunto de objetos materiales e immateriales (obras, artefactos, mente factos, especímenes, documentos, archivos, testimonios, etc.) que un individuo o un establecimiento, estatal o privado, se han ocupado de reunir, clasificar, seleccionar y conservar en un contexto de seguridad para comunicarlo, por lo general, a un público más o menos amplio. |
| Curador | Es la persona encargada de planificar, ejecutar y gestionar todas las actividades relacionadas con las salas, colecciones, obras y demás aspectos administrativos del museo. |
| Educación museal | Un conjunto de valores, conceptos, conocimientos y prácticas cuyo objetivo es el desarrollo del visitante; trabajo de aculturación, se apoya principalmente en la pedagogía y en el completo desarrollo, así como en el aprendizaje de nuevos saberes. |
| Kiosco multimedia | Dispositivo con información multimedia sobre las salas, servicios, obras, autores y demás informaciones atinentes al museo (historia, eventos especiales, etc.). |
| ... | ... |
| Proyecto | Es un conjunto de actividades emprendidas por una persona o una entidad para alcanzar un objetivo específico (concepto.de, 2019). Fuente: https://concepto.de/que-es-proyecto/#ixzz5jf8LU5TQ |
| Sistema actual para la guía a los visitantes | Un dispositivo portable (DVD Player) que guía a los visitantes en las distintos espacios del museo. Se entrega a los visitantes que así lo deseen. |
| Sistema nuevo | Sistema interactivo a ser desarrollado para cubrir las necesidades del cliente y de los visitantes del museo. |
| Visitante | Usuario de distintas nacionalidades, edades, capacidades y necesidades que decide visitar las distintas salas del museo. |
| ... | ... |

Nota: Elaboración propia.

Tabla 12

Dominio y contextos de uso del sistema interactivo MyEnchantedMuseum – Interactive Complex

| Aspectos | Descripción |
|------------------|--|
| Dominio | Museístico: institución de carácter permanente que adquiere, conserva, investiga, comunica, exhibe para fines de estudio, educación, contemplación y entretenimiento, conjuntos y colecciones de valor histórico y artístico (pintura y escultura) de diversos movimientos. Adaptado de (ICOM). |
| Contextos de uso | Educación: los visitantes estén motivados en conocer, entender y aprender. |
| | Entretenimiento: los visitantes puedan disfrutar de los servicios y "contenidos". |
| | Social: los visitantes puedan compartir las experiencias e información recolectada. |

Nota: Elaboración propia.

Como resultado de aplicar la actividad de obtención y análisis de requisitos al caso de estudio **MyEnchantedMuseum**, se obtuvo la lista de interesados que se muestra en la Tabla 13.

Tabla 13

Lista de algunos interesados para el desarrollo del sistema MyEnchantedMuseum – Interactive Complex

| Involucrados | Descripción |
|--|--|
| Administrador de tecnologías del museo | Responsable de las tecnologías instaladas y utilizadas en el museo |
| Analistas | Personal responsable de la obtención, análisis y obtención de requisitos |
| Curador Principal | Responsable general del museo |
| Curadores de cada colección | Lista de responsables de cada una de las colecciones |
| Responsables del sitio web del museo | Grupo de profesionales que desarrollan nuevas funcionalidades y actualizan los contenidos del sitio web del museo |
| Líder de proyecto | Responsable de la planificación del proyecto para el desarrollo del sistema <i>MyEnchantedMuseum – Interactive Complex</i> |
| Usuario | Modelo de posibles visitantes (modelados a discreción de los analistas) |

Nota: Elaboración propia.

A continuación, se muestra el material recopilado y la información sobre: los requisitos funcionales (Tabla 14), los no funcionales y restricciones (Tabla 15) obtenidos durante las actividades de obtención, análisis y especificación de requisitos. Las personas creadas se muestran en la Figura 9.

Tabla 14

Requisitos funcionales obtenidos, analizados y especificados para el sistema MyEnchantedMuseum – Interactive Complex

| Tipo | Descripción |
|-------------------------------|--|
| Requisitos de datos | <ol style="list-style-type: none"> El sistema debe almacenar la siguiente información acerca de las obras expuestas: autor, técnica, materiales, periodo, movimiento artístico. El sistema debe almacenar la información de los recorridos virtuales que hacen los usuarios en el sitio web. |
| Requisitos transaccionales | <ol style="list-style-type: none"> El sistema debe brindar una herramienta multimedia al usuario para intervenir obras según sus necesidades: cambio de color y forma. El sistema debe proveer una herramienta para guiar el usuario en un recorrido personal dentro del museo. El sistema debe permitir al usuario compartir el resultado de su intervención en cada obra. |
| Requisitos de interfaz | <ol style="list-style-type: none"> El sistema debe proporcionar herramientas para que el usuario pueda registrar un recorrido personalizado. El sistema debe incluir una herramienta para guiar el usuario con modalidad audio y video sobre la información de las obras. El sistema debe incluir una herramienta multimedia para que el usuario pueda intervenir obras de su interés (edición de color y forma). |
| Requisitos de navegación | <ol style="list-style-type: none"> El sistema debe incluir una herramienta web para que el usuario pueda navegar los contenidos sobre servicios, eventos, colecciones, recorridos y obras. |
| Requisitos de personalización | <ol style="list-style-type: none"> El sistema debe configurarse para los siguientes idiomas: inglés, español, francés, italiano, alemán, El sistema debe permitir la configuración de menús para los niños en edad preescolar. |

Nota: Elaboración propia.

Tabla 15

Requisitos no funcionales y restricciones del sistema *MyEnchantedMuseum – Interactive Complex*

| Tipo | Descripción |
|-------------------------|---|
| No Funcionales | 1. Funcionalidad 2. Capacidad de mantenimiento 3. Portabilidad 4. Eficiencia 5. Configurabilidad respecto a distintas plataformas 6. Usabilidad y Accesibilidad 7.... |
| Restricciones | 1. El presupuesto para el desarrollo del sistema es de 500.000 € 2. El tiempo de desarrollo y prueba no puede ser superior a 6 meses 3. El sistema debe incluir distintos modos de comunicación y distintos formatos de información (multimedia) 4.... |
| Requisitos de evolución | 1. El sistema debe poder crecer en contenidos, funcionalidades y plataformas 2.... |

Nota: Elaboración propia.



Lorena tiene 9 años. Le gusta pintar y hacer figuras con plastilina. Le gusta leer historias de piratas. Es muy creativa ...
 Quiere ser médico, pero le encanta el diseño. Ha creado una línea de pequeños vestidos para sus muñecas

Estudios: ...
 Grupo familiar: ...



Andrés tiene 42 años. Es padre de 2 niños, a los cuales acompaña, junto a su esposa, a los parques temáticos y a museos. Considera que la enseñanza que brindan esos lugares es fundamental para el crecimiento sano de sus hijos.

Estudios: contaduría....
 Grupo familiar:



Enrique tiene 38 años. Está casado y es padre de una niña. Por un accidente de auto, sufre de una parálisis en piernas y en el lado izquierdo de su cuerpo. Sin embargo, sigue trabajando y hace uso de tecnologías alternativas para cumplir con sus responsabilidades. Es muy creativo y le gusta el arte en todas sus dimensiones.

Estudios: Arquitectura
 Grupo familiar:

Figura 9. Ejemplo de algunas personas creadas para el sistema *MyEnchantedMuseum- Interactive Complex*. Elaboración propia.

Tema 5. Producido diseños. Técnicas de prototipado

"Es muy difícil diseñar productos para grupos focales [de usuarios]. Muchas veces, las personas no saben lo que quieren hasta que se lo muestras"

(Steve Jobs – Visionario y fundador de Apple Computer Inc.)

Cada vez que se genera una solución a un problema, se lleva a cabo un diseño. No alcanza con tener el diseño si no puede ser evaluado con los usuarios del futuro sistema. Ese diseño debe ser comunicado a los distintos interesados. Una estrategia para comunicarlo es mostrando modelos de lo que el usuario podrá realizar con el sistema. **Diseñar y comunicar el diseño es una de las actividades fundamentales del proceso de diseño centrado en el usuario.**

Para comunicar, expresar y describir a los usuarios lo que se está diseñando, considerando sus requisitos y necesidades, se hace uso de varias estrategias de modelado. Una estrategia muy utilizada es el **prototipado**, es decir la creación de prototipos. Esta técnica se utiliza para la elaboración de modelos concretos, durante **la actividad de producción de diseños**.

El prototipado proporciona un conjunto de herramientas y enfoques para probar, evaluar y explorar ideas adecuadamente antes de que se utilicen demasiados recursos. Es una técnica que los seres humanos conocen desde temprana edad: haciendo uso de materiales simples (papel, cartulinas, plastilinas), ya que han creado "maquetas" de objetos para jugar, aprender y comunicar.

Un prototipo es el resultado de esta técnica. Representa un modelo experimental simple de una propuesta de solución, utilizada para probar o validar las suposiciones de diseño y otros aspectos de su conceptualización en forma rápida y (relativamente) económica. Los diseñadores involucrados utilizan este instrumento para “comunicar” la solución a los interesados.

La creación de prototipos es esencial para evaluar y resolver problemas de usabilidad. También puede dar retroalimentación sobre áreas que necesitan mejoras. Una vez que un borrador de la idea de la solución diseñada esté en manos de usuarios reales, con el prototipo entenderán cómo será utilizado.

Un prototipo puede ser cualquier “cosa”: desde dibujos en papel, bocetos (representan el producto a baja fidelidad) hasta un artefacto que permita interactuar con algunos objetos y funcionalidades en una aplicación, o un sitio web totalmente funcional (representan el producto a alta fidelidad).

5.1. Características y clasificación de prototipos

Un prototipo puede ajustarse según las necesidades de un diseñador. En este sentido se habla de características y/o propiedades asociadas, entre las cuales están:

- a. **Respecto a la fidelidad/precisión.** La fidelidad del prototipo, es decir, su nivel de realismo. Responde a la pregunta ¿cuán fiel el prototipo representa el diseño de la solución?
- b. **Respecto a su objetivo.** Responde a la pregunta ¿el prototipo servirá para efectos exploratorios, experimentales u operacionales?
- c. **Respecto a la cantidad vs. calidad de funcionalidades representadas.** Responde a las preguntas ¿es estático o dinámico (se puede interactuar con él)? ¿cuál es el nivel de detalle de las funcionalidades modeladas? ¿Cuántas funcionalidades mostrará?
- d. **Respecto al alcance de las funcionalidades modeladas.** Responde a la pregunta ¿representará todo el producto o sólo algunos de sus componentes?
- e. **Respecto al nivel de ejecución.** Responde a la pregunta ¿cómo y cuánto se puede operar con el prototipo? Es decir, cuán ejecutable es el prototipo.

En base a las características anteriores, en la Tabla 16 se muestra la clasificación de prototipos respecto a fidelidad, objetivo y cantidad vs. calidad de las funcionalidades. En la Tabla 17 se muestra la clasificación respecto al alcance y el nivel de ejecución.

Tabla16

Clasificación de prototipos: respecto a su fidelidad, su objetivo y la cantidad vs calidad de funcionalidades

| Propiedad | Clases de prototipos | |
|---|--|---|
| | Baja | Alta |
| Fidelidad | Conjunto de dibujos (por ejemplo, una presentación de escenarios) que constituye una maqueta estática, no computarizada y no operativa de una interfaz de usuario para un sistema en planificación. | Conjunto de pantallas que proporcionan un modelo dinámico, computarizado y operativo de un sistema en planificación. |
| Objetivo | Exploratorio | Experimental |
| | Prototipo no reutilizable utilizado para clarificar las metas del proyecto, identificar requisitos, examinar alternativas de diseño o investigar un sistema extenso y complejo. | Prototipo utilizado para la validación de especificaciones de sistema. |
| Cantidad vs. Calidad de funcionalidades | Operacional | |
| | Prototipo iterativo que es progresivamente refinado hasta que se convierte en el sistema final. | |
| | Horizontal | Vertical |
| Cantidad vs. Calidad de funcionalidades | Modela muchas características de un sistema, pero con poco detalle. Dicho detalle alcanzará una profundidad determinada, va a resultar especialmente útil en las etapas tempranas de diseño y tiene como objetivo el test del modo de interacción global, al contemplar funciones comunes que el usuario va a utilizar frecuentemente. | Modela pocas características de un sistema, pero con mucho detalle. Va a resultar especialmente útil en etapas más avanzadas del diseño y tiene como objetivo el test de detalles del diseño. |
| | Diagonal | |
| | Prototipo horizontal hasta un cierto nivel, a partir del cual se puede considerar vertical. | |

Nota: Elaboración propia

Tabla 17

Clasificación de prototipos: respecto al alcance de las funcionalidades modeladas y al nivel de ejecución

| Propiedad | Propiedad | |
|--|---|---|
| | Global | Local |
| Alcance de las funcionalidades modeladas | Prototipo del sistema completo. Es un prototipo horizontal expandido que modela una gran cantidad de características y cubre un amplio rango de funcionalidades. Va a resultar muy útil a lo largo de todo el proceso de diseño. | Prototipo de un único componente o característica del sistema de usabilidad crítica. Va a resultar de utilidad en algunas etapas específicas del proceso de diseño. |
| Nivel de ejecución | <ul style="list-style-type: none"> • Prototipo guiado: es lo suficientemente operativo como para llevar a cabo una revisión o paseo cognitivo. • Prototipo de animación: es lo suficientemente operativo como para ser ejecutado por partes en modo de presentación en un ordenador. • Prototipo del Mago de Oz: permite ser ejecutado en modo de presentación y la participación de un tercer sujeto que, oculto a la vista del usuario que interactúa, gestiona la evolución de la presentación en función de las entradas que el usuario proporciona al sistema. • Prototipo Interactivo: es operativo en sentido estricto, se ejecuta y responde a las entradas que le proporciona el usuario participante, en tiempo real. • Prototipo Funcional: es operativo en sentido estricto, se ejecuta, responde a las entradas que le proporciona el usuario participante en tiempo real y efectúa alguna de las operaciones que se le solicitan. | |

Nota: Elaboración propia

Sobre el proceso de diseño de un prototipo

El proceso para llegar a una solución de diseño puede iniciar con un boceto, luego refinarlo para llegar a una plantilla (conocida como *wireframe* en este contexto), hasta lograr un prototipo funcional. Estos instrumentos se utilizan en distintos momentos del proceso de diseño (ideación, validación, refinamiento, evaluación de usabilidad) como se muestra en la Figura 10, en la cual puede inferirse cómo desde un boceto (que expresa una baja fidelidad respecto al producto), pasando por un *wireframe*, se puede alcanzar un diseño a mayor fidelidad (muestra características más fieles y parecidas a lo que será finalmente el producto).

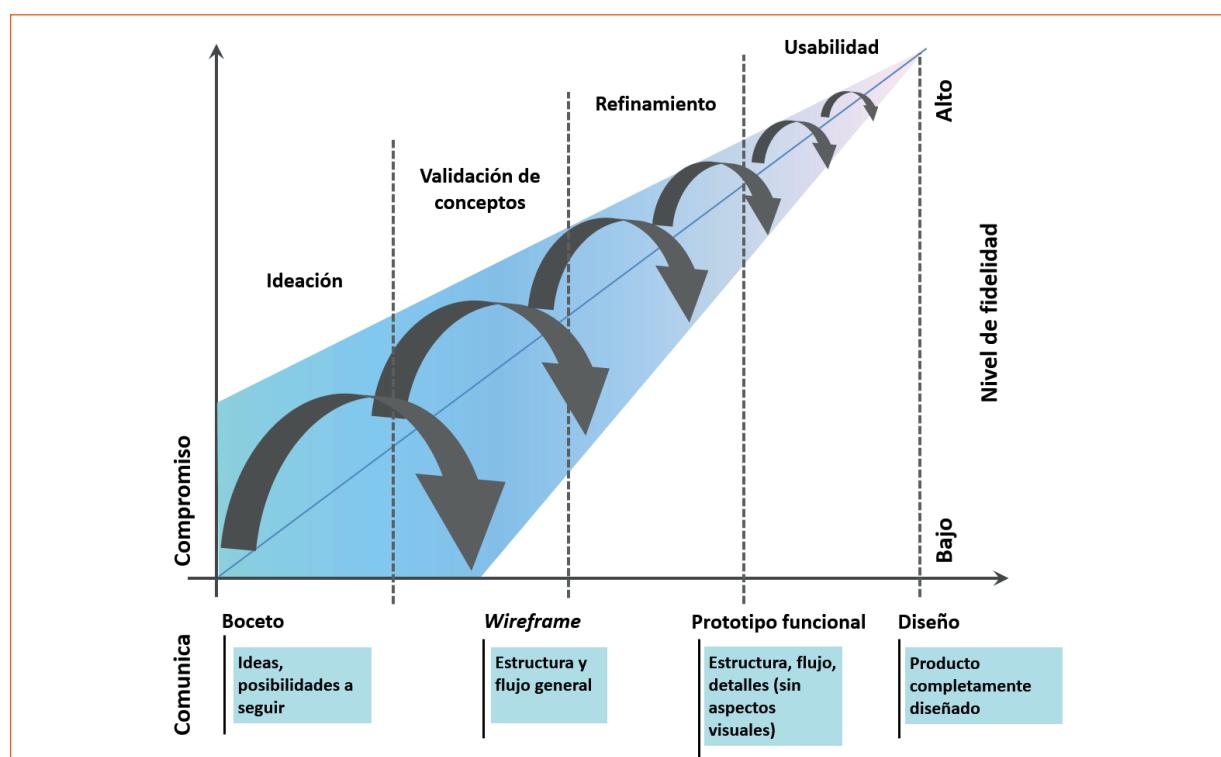


Figura 10. Desde un boceto, pasando por un wireframe, por un prototipo funcional, hasta llegar al diseño final de un producto interactivo. Adaptado de <https://www.uxmatters.com/mt/archives/2010/05/sketches-and-wireframes-and-prototypes-oh-my-creating-your-own-magical-wizard-experience.php>

En la Figura 10 también se aprecian las relaciones entre el nivel de compromiso y fidelidad de un prototipo y lo que está tratando de comunicar. Estos son los factores que importan al decidir dónde se encuentra el proceso en este continuo. Si bien las líneas que delimitan los bocetos, los *wireframes* y los prototipos de mayor fidelidad, no son muy claras, las fases de ideación, validación de conceptos, refinamiento y usabilidad sí lo son, y ahí es donde se debe centrar la atención.

El desarrollo de un tipo de prototipo específico está asociado a las complejidades del sistema, a las clases de interesados, la calidad de requisitos obtenidos, los costos y recursos, entre otros. En algunas ocasiones se desarrollan varias clases y tipos de prototipos para comunicar una funcionalidad o características del producto. En la Figura 11 se muestra un ejemplo de boceto y de un *wireframe*.

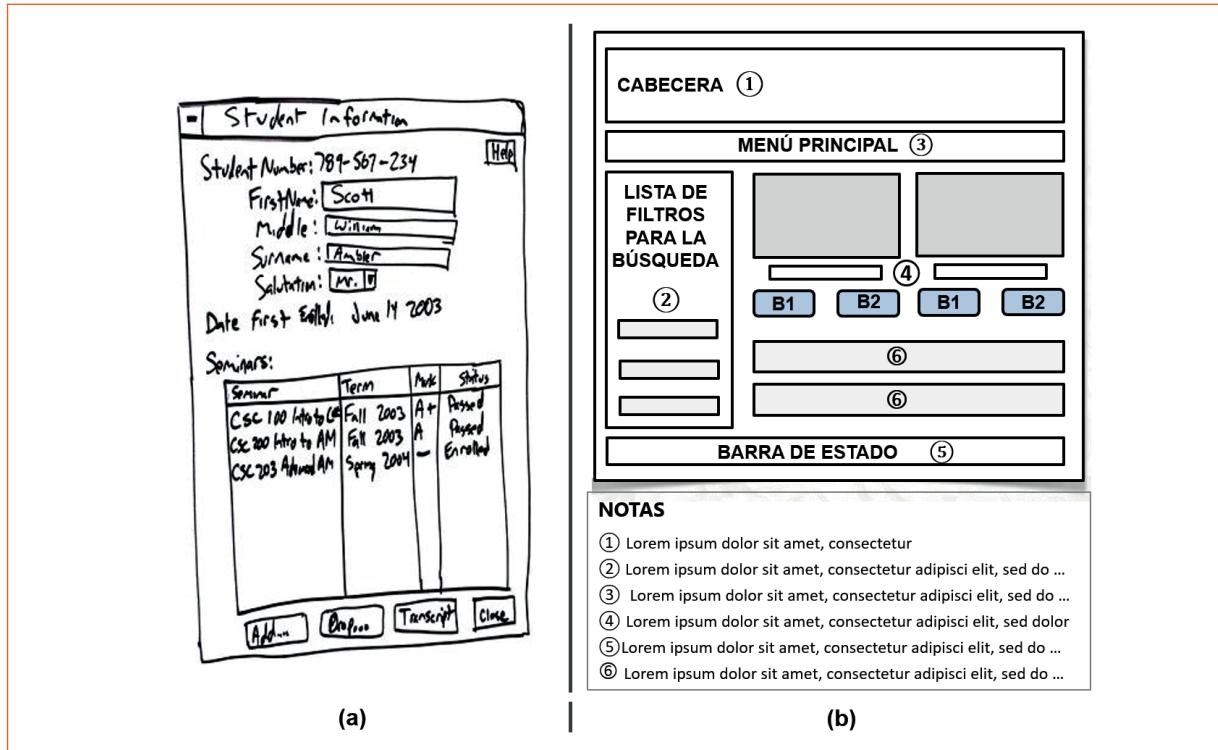


Figura 11. (a) Ejemplo de un boceto. Recuperado de <http://agilemodeling.com/artifacts/uiPrototype.htm>. (b) Ejemplo de un wireframe. Elaboración propia

5.2. Técnicas generales de prototipado. Mejores prácticas

Antes de comenzar a analizar las técnicas más conocidas para la creación de prototipos, es necesario recordar los elementos claves involucrados en el proceso de diseño y las pruebas, sin importar la técnica a utilizar. Cuando se construyen los prototipos, así como cuando se estén probando, deben considerarse los siguientes elementos:

- a. **Objetos:** estáticos e interactivos, incluidos el prototipo y otros objetos con los que las personas y/o el prototipo interactúan.
- b. **Ubicación:** lugares y ambientes especiales.
- c. **Interacciones:** digitales o físicas, entre personas, objetos y la ubicación.
- d. **Capacidades técnicas** para su creación.

Por ejemplo, si se está probando un prototipo en un laboratorio (ubicación), es necesario entender cómo simular el entorno natural en el cual el diseño atraerá a sus usuarios. Adicionalmente deben considerarse los otros objetos con los que se utilizará el prototipo, por ejemplo ¿Los usuarios usarán guantes o tendrán las manos ocupadas? ¿Qué implicaciones tendrá eso sobre cómo podrán usar un producto o servicio? ¿Cuáles son los tiempos y costos de creación?

Con estas preguntas en mente, las técnicas más utilizadas para la creación de prototipos para sistemas interactivos se listan y describen en la Tabla 18.

Tabla 18
Técnicas de prototipado

| Nombre | Uso | Coste | Desarrollo |
|--|--|---|---|
| Boceto | Primeras ideas | Muy bajo | Muy rápido |
| <i>Storyboard</i> | <ul style="list-style-type: none"> Reflejar contextos Describir el proceso de interacción Identificar y ubicar los usuarios y los objetos utilizados en la interacción | Muy bajo | Rápido |
| Prototipos de papel | <ul style="list-style-type: none"> Visualizar las funciones (más que la propia funcionalidad) Sintetizar ideas, de forma minimalista y simple | Muy bajo | Rápido |
| Maquetas | <ul style="list-style-type: none"> Reflejar la utilización Reflejar las características físicas | Relativamente bajo (depende de los materiales utilizados) | No tan rápido |
| Maquetas digitales | <ul style="list-style-type: none"> Visualizar las funciones (más que la propia funcionalidad) Mostrar el paradigma y tipo de interacción Mostrar la disposición de los elementos de interfaz de usuario | Bajo | No tan rápido |
| <i>Storyboard</i> navegacionales | <ul style="list-style-type: none"> Representar la interfaz de usuario Mostrar la navegación entre funcionalidades Representar los estados del sistema | Bajo | Rápido, pero depende de la forma de representación de los estados del sistema |
| Videos | <ul style="list-style-type: none"> Representar casos y situaciones interactivas Entender el contexto y el objetivo de la tarea Mostrar los usuarios y los objetos de la interfaz Escenificar las posibilidades futuras o de difícil acceso | Alto | Muy alto |
| Escenarios | <ul style="list-style-type: none"> Representar casos y situaciones interactivas Entender el contexto y el objetivo de la tarea Mostrar los usuarios y los objetos de la interfaz | Medio | Suele ser rápido |
| Prototipo Horizontal (<i>software</i>) | <ul style="list-style-type: none"> Mostrar la navegación Validar seguimiento de las tareas Mostrar el proceso interactivo en forma global | Medio | Medio / Alto |
| Prototipo Vertical (<i>software</i>) | <ul style="list-style-type: none"> Explorar funcionalidades concretas Medir el rendimiento Evaluuar con métricas específicas | Medio | Medio / Alto |

Nota: Elaboración propia.

Respecto a las mejores prácticas y recomendaciones de uso de prototipos, en la Tabla 19 se resumen algunos aspectos a considerar.

Tabla 19
Algunas recomendaciones y buenas prácticas en el uso de prototipos

| ¿Cuándo? | ¿Para qué? |
|---|--|
| Antes de comenzar... | Mostrar un concepto a la directiva de una organización. |
| Comenzando... | Reunir los requisitos iniciales de usuario. |
| Después de empezar... | Validar la evolución de los requisitos de usuario. |
| En las etapas intermedias... | Validar las especificaciones del sistema. |
| Entre las etapas intermedias y finales... | Preparar a los usuarios o crear una demostración de mercadeo del producto. |
| En las etapas finales... | Explorar soluciones a problemas de diseño o usabilidad específicos. |
| Beneficios de utilizar prototipos | |
| Para explorar y experimentar | Para explorar problemas, ideas y oportunidades dentro de un área específica de enfoque y probar el impacto de cambios incrementales o radicales. |
| Aprender y comprender | Los prototipos se pueden usar para comprender mejor la dinámica de un problema, producto o sistema al comprometerse físicamente con ellos y separar lo que los hace funcionar o fallar. |
| Involucrar, probar y experimentar | Crear prototipos es útil para interactuar con los usuarios finales o las partes interesadas, de manera que revelen una visión más profunda y experiencias más valiosas, para informar las decisiones de diseño en el futuro. |
| Inspirar y motivar | Los prototipos pueden usarse para vender nuevas ideas, motivar la aceptación de partes interesadas internas o externas, o inspirar a los mercados hacia nuevas formas radicales de pensar y hacer. |

Nota: Elaboración propia.

5.3. Prototipos generados para el sistema interactivo *MyEnchantedMuseum-Interactive Complex*

Una vez terminadas las actividades de obtención, análisis y especificación de requisitos para el caso de estudio **MyEnchantedMuseum-Interactive Complex**, el equipo de desarrollo generó las siguientes soluciones:

Solución A

El sistema interactivo tendrá los siguientes componentes:

1. Un **dispositivo para guiar el usuario** [libremente] en las distintas salas del museo. Está basado en el **paradigma de interacción basado en Realidad Aumentada**. Dentro de este paradigma, los diseñadores decidieron **utilizar tecnología de reconocimiento de objetos**. El usuario podrá descargar en su dispositivo inteligente (*smartphone* o tableta) la aplicación necesaria que es capaz de “reconocer” cada una de las obras (pinturas y esculturas), e incluir información adicional sobre el objeto visualizado (seleccionado). Respecto a las **primitivas de interacción, el usuario deberá seleccionar/apuntar** el objeto de interés.

Desde el punto de vista de diseño de modelos, se decide utilizar un **prototipo de fidelidad alta, exploratorio, horizontal y la técnica seleccionada es una maqueta digital** que muestra las funcionalidades del sistema y los elementos de interacción de los cuales dispone el usuario.

Algunas características del prototipo creado se muestran en la Figura 12.



Figura 12. Algunas características del prototipo del dispositivo para guiar el usuario en el museo (Solución A). Elaboración propia.

2. **Un nuevo concepto de kiosco multimedia.** Se instalarán en la entrada de cada una de las salas (colecciones) de forma que puedan ser utilizados en el espacio físico asociado. Se incluyen contenidos en todos los formatos disponibles: texto, imagen, video y audio. En el kiosco [aplicación] se incluye la funcionalidad para que se pueda intervenir una obra seleccionada (edición de distintos aspectos, es decir forma y color). Luego de la edición, la imagen podrá ser compartida en las redes sociales.

El prototipo será de **fidelidad alta, exploratorio, horizontal y la técnica seleccionada es una maqueta digital que muestra la navegación** entre las obras y autores, además de las funcionalidades **de búsqueda de información y edición de [imágenes] de obras** (lo cual desencadena una instancia donde se presentan algunas herramientas de edición de imágenes). Incluirá elementos de interacción usando el paradigma manipulación directa.

Además, **se propone crear un prototipo en “papel” (realmente es en acrílico y cartón) con el cual se pueden observar las dimensiones del kiosco, así como las pantallas táctiles y su impacto en los espacios de las distintas salas**. En la Figura 13 se muestra el aspecto

que tendrá el kiosco. Como se observa es un tótem, que contiene 4 pantallas táctiles (de 32") dispuestas, en forma intercalada, de la siguiente forma:

- 2 de ellas colocadas a una altura de 1,30 metros del nivel de piso para 2 usuarios de pie frente a cada monitor;
- 2 a la altura de 0,80 metros del nivel de piso, para ser utilizado por 2 usuarios (k-12) sentados (en sendas sillas) y para usuarios discapacitados.

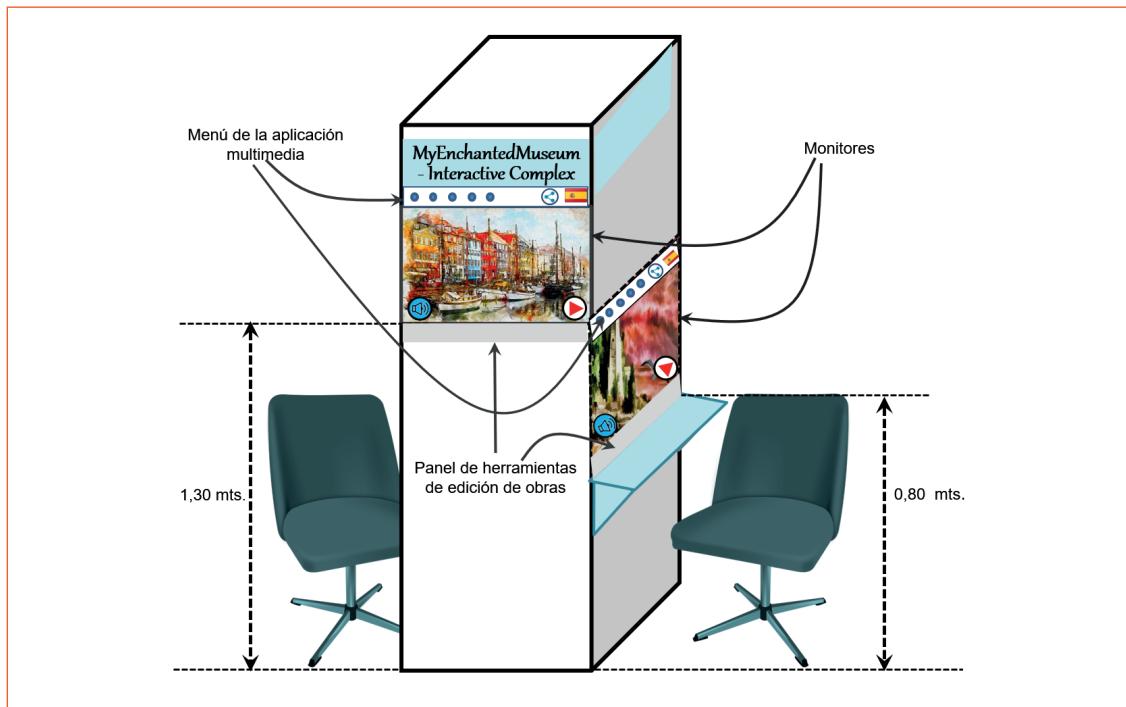


Figura 13. Prototipo del nuevo kiosco multimedia (Solución A). Elaboración propia.

3. En el **sitio web del museo actual**, se incluirá la funcionalidad para personalizar visitas virtuales. De esta manera un usuario puede "almacenar" su visita, convertirla en un recorrido que se podrá descargar en su *smartphone*.

El prototipo es de **baja fidelidad, exploratorio, vertical, y la técnica seleccionada es un storyboard para mostrar las tareas asociadas a la funcionalidad para almacenar la visita** seleccionada por el usuario.

4. Un **folleto interactivo del museo**, con información sobre los servicios del museo, cada una de las colecciones, el catálogo de la tienda del museo y la planimetría. El folleto es interactivo porque cada uno de los elementos de información están disponibles mediante la lectura de códigos QR presentes en el folleto. El usuario tendrá disponible la aplicación para la lectura de dichos códigos. El prototipo es de **alta fidelidad, exploratorio, vertical y la técnica seleccionada es un wireframe** con todos los elementos considerados. En la figura 14 se muestra el prototipo diseñado.

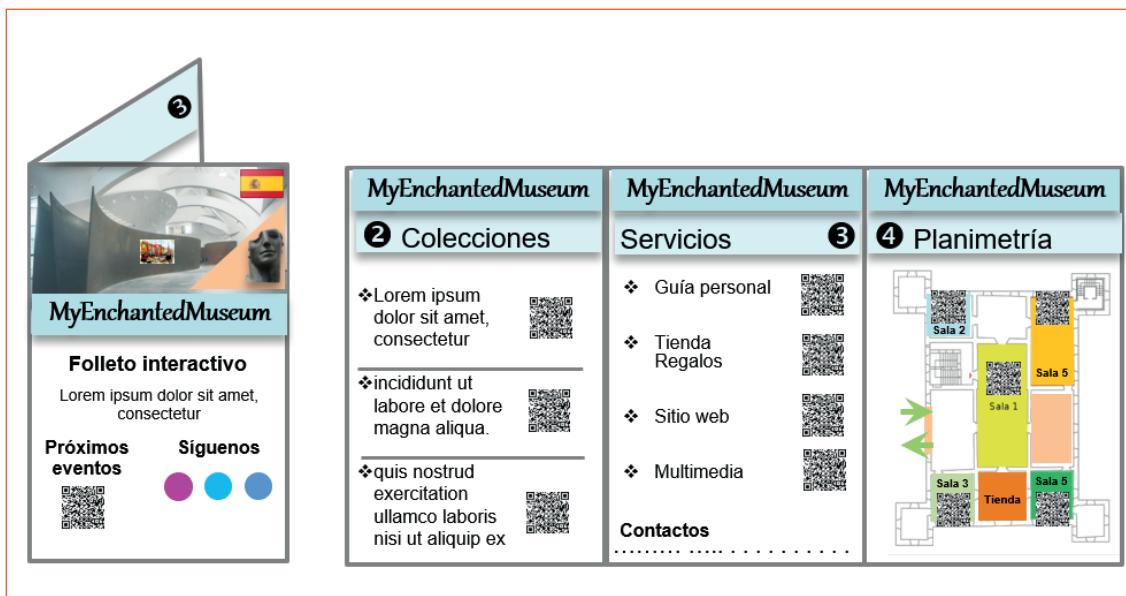


Figura 14. Prototipo del folleto interactivo (Solución A). Elaboración propia.

Solución B

En esta solución, el único cambio respecto a la Solución A es relativo al dispositivo para guiar al usuario en las distintas salas. Los demás componentes del sistema seguirán lo descrito en la solución A.

Algunos aspectos relacionados con el diseño del dispositivo son:

- Está basado en el **paradigma de interacción de realidad aumentada**, haciendo uso de la **tecnología QR** (en particular QR dinámico).
- Cada visitante podrá instalar la aplicación necesaria para hacer uso de esa tecnología disponible en el sitio web, a través del WIFI del museo, disponible para todos los visitantes.
- En cada una de las salas (colecciones) habrá un código QR con información general sobre la colección.
- En cada una de las obras habrá un código QR (en un lugar visible).
- Si bien se utilizará el paradigma realidad aumentada, en este caso la **primitiva de interacción a utilizar es (seleccionar/apuntar)**. En su *smartphone* o tableta, el visitante podrá obtener información textual, visual y auditiva sobre los aspectos de la obra, su autor, la técnica utilizada, materiales, etc.
- El prototipo es de **baja fidelidad, exploratorio, vertical y la técnica seleccionada es una maqueta digital**. Un aspecto del prototipo creado se muestra en la Figura 15.



Figura 15. Prototipo del dispositivo para guiar el usuario en el museo (Solución B). No incluye los objetos de interfaz.
Elaboración propia.

En el apartado 7.6 de este contenido se discuten la evaluación de los diseños, respecto a los requisitos de usuario y calidad de cada una de las soluciones.

Tema 6. Diseñando las interfaces de usuario

"La interfaz y la interacción entre el humano y el computador han cambiado con el aumento de la velocidad y el espectro de modos de comunicación. La interfaz está en todas partes (ubicua y móvil); la interfaz es visual (ver y mostrar); la interfaz es conversacional (hablar y escuchar). Y la interfaz es inteligente. En retrospectiva, nos ha llevado menos de medio siglo comenzar a crear la base que se puede utilizar para proporcionar una rica experiencia personal y social, además de ser útil".

(Norman y Kirakowsky, 2018, p.3)

6.1. Conceptos preliminares sobre interfaces de usuario

El diseño de la interfaz de usuario es una actividad que implica construir una parte esencial de la experiencia del usuario; los usuarios son muy rápidos para juzgar los diseños en cuanto a usabilidad y satisfacción. El primer requisito para **lograr** una **buen**a experiencia de usuario (ante un producto) es satisfacer sus necesidades exactas, sin problemas ni molestias, en cada una de las etapas de desarrollo del producto. Sin embargo, lograr una experiencia positiva en el usuario va mucho más allá de dar a los clientes lo que dicen que quieren o de proporcionar funciones en una lista de verificación. Deben considerarse en cada momento del desarrollo de un producto, aquellos aspectos cognitivos y subjetivos, habilidades y capacidades del usuario con el objetivo de lograr una satisfacción en distintos contextos de su utilización.

En definitiva, **la experiencia del usuario se refiere a las experiencias singulares y acumuladas que ocurren en los usuarios como consecuencia de su interacción con un objeto en un contexto dado**. El concepto de interacción ha sido desarrollado en el pasado contenido, y lo que sigue trata del diseño de **buenas** interfaces.

El **objetivo de una interfaz de usuario** es hacer que la funcionalidad de un producto sea **evidente** para el usuario. Una interfaz de usuario diseñada, en forma adecuada, ajusta la "imagen" que el usuario tiene sobre la realización de una tarea. **La interfaz de usuario es la superficie de contacto entre dos entes disímiles, por lo tanto, hay que crear un lenguaje para que pueda haber un diálogo entre esos entes.**

Su efectividad puede afectar la aceptación de todo el sistema; de hecho, en muchos casos puede afectar el éxito o el fracaso general de un producto. Como se ha indicado, **la interfaz de usuario es un producto del proceso de diseño de la interacción, y se juzga (evalúa) a través de su usabilidad**.

Las interfaces de usuario son los puntos de acceso donde los usuarios interactúan con los dispositivos. Las interfaces gráficas de usuario (GUI – *Graphic User Interfaces* por sus siglas en inglés) son aquellos elementos de control que se proporcionan a los usuarios para activar un evento en una secuencia de tareas. También existen las **interfaces multimodales**: aquellas controladas por voz implican interacción auditiva oral, mientras que las interfaces basadas en gestos son un ejemplo de cómo los usuarios interactúan con espacios de diseño en 3D a través de movimientos corporales.

Los diseñadores deben enfocarse en construir interfaces que los usuarios encontrarán altamente utilizables y eficientes (**y no relacionadas únicamente con su estética o aspecto gráfico**). Por lo tanto, el diseño incluye una comprensión profunda de los contextos en los que se encontrarán los usuarios al hacer esos juicios (eficiencia y productividad).

En una entrevista realizada a Norman en (Laurel y Mountford, 1990, p. 210), expresó "El problema real con la interfaz es que es una interfaz. Las interfaces se interponen en el camino. No quiero enfocar mis energías en una interfaz. Quiero centrarme en el trabajo ... no quiero pensar que estoy usando un computador, quiero pensar que estoy haciendo mi trabajo". Es decir, se debe crear la ilusión de que los usuarios no están interactuando con un dispositivo, sino que desean alcanzar objetivos directamente y sin esfuerzo. El diseño de la interfaz de usuario no es sólo colocar íconos, formularios y botones en una pantalla. Se debe lograr que **la interfaz sea efectivamente "invisible y natural"** ofreciendo a los usuarios elementos a través de los cuales puedan interactuar directamente con la realidad de sus tareas, sin carga cognitiva adicional.

Otra consideración es la de permitir que **los usuarios encuentren su camino sobre la interfaz de forma intuitiva: cuanto menos noten que deben usar los controles, más se sumergirán**. Esta dinámica se aplica a otra dimensión del diseño de la interfaz de usuario: su diseño debe tener tantas características agradables como sea apropiado.

Los **objetivos de una buena interfaz** son los siguientes:

- a. **Maximizar la velocidad de aprendizaje.**
- b. **Minimizar la tasa de errores.**

- c. Maximizar la velocidad de uso.
- d. Presentar una apariencia visual adecuada.

En el ámbito del diseño de interfaces existen **tres puntos de vista** distintos:

- El del usuario
- El del programador
- El del diseñador

Cada punto de vista tiene un modelo mental propio de la interfaz, que contiene los conceptos y expectativas acerca de la misma, desarrollados a través de cada experiencia.

El modelo del diseñador describe el conjunto de objetos que utilizará el usuario, su presentación para su uso y las tareas de interacción para su manipulación. Consta de las siguientes tres partes:

1. La **presentación** es lo primero que capta la atención del usuario, pero más tarde pasa a un segundo plano, y adquiere más importancia la interacción con el producto para poder satisfacer sus expectativas. **La presentación no es el aspecto más relevante, aunque algunos lo consideran para valorar una interfaz, de forma que confunden la estética y el diseño gráfico con la interfaz.**
2. La segunda parte del modelo define las **tareas de interacción** del usuario, a través de diversos dispositivos.
3. La tercera es la más importante, y es donde el diseñador **siguiendo un proceso desde lo más abstracto a lo más concreto, determina la ambientación adecuada, la organización de los objetos y de los contenidos, de forma que todo pueda encajar con el modelo mental del usuario, sus necesidades y los objetivos del producto.**

En la Figura 16 se muestran cómo se comportan “proporcionalmente” las partes del modelo del diseñador para entender las dificultades en el desarrollo.

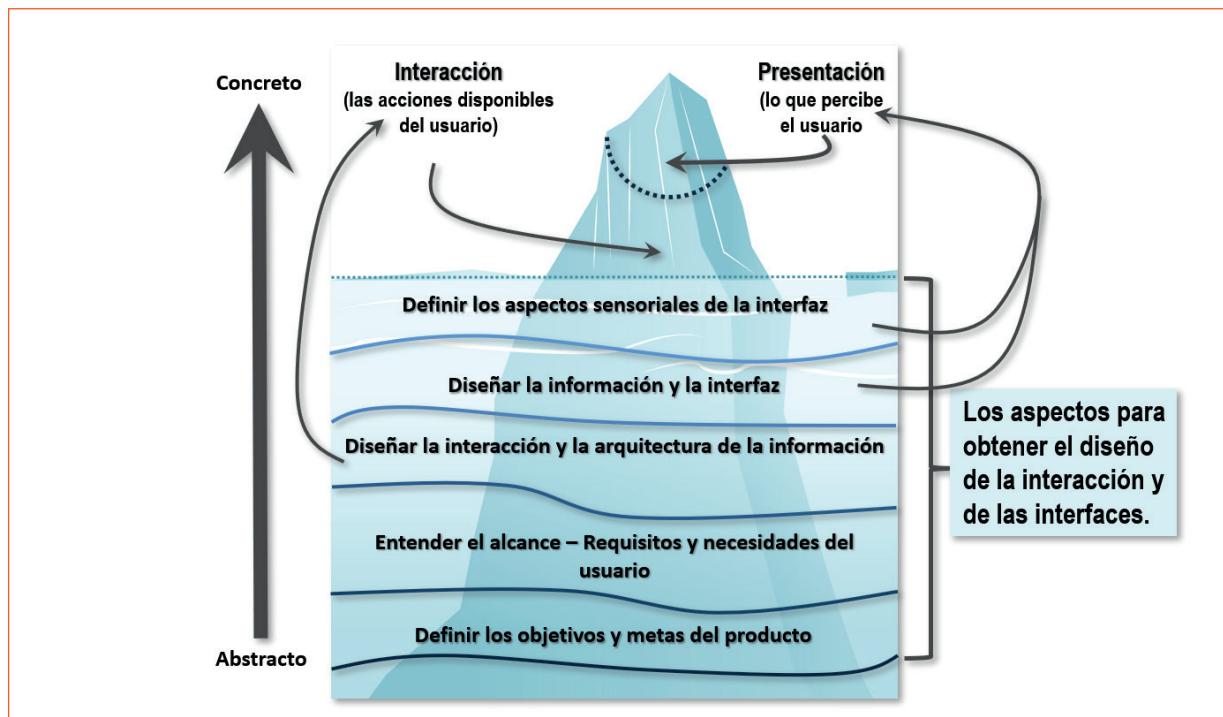


Figura 16. Modelo del diseñador. Adaptado del concepto “Look and feel Iceberg” IBM (2000), y de Diseño centrado en el usuario según Garrett, J. (2011).

Una de las estrategias de diseño es que el desarrollador conozca los estilos y objetos de las interfaces, cómo y cuándo se deben utilizar, así como los principios de diseño, estándares, guías y buenas prácticas. De esta manera puede lograr que la interfaz diseñada logre sus objetivos.

6.2. Estilos y objetos de las interfaces

6.2.1. Estilos de interfaces

El estilo de una interfaz viene definido por las primitivas de interacción que se utilizaron, los paradigmas y la tecnología subyacente. Cada primitiva implica la utilización de distintos canales, visualizaciones de objetos y su manipulación, así como las tecnologías de punta que permiten que los dispositivos sean cada vez más adaptables a contextos de uso y a los requisitos de los usuarios. **Sobre las primitivas y los paradigmas de interacción se recomienda revisar los contenidos en los Anexos A y B respectivamente.**

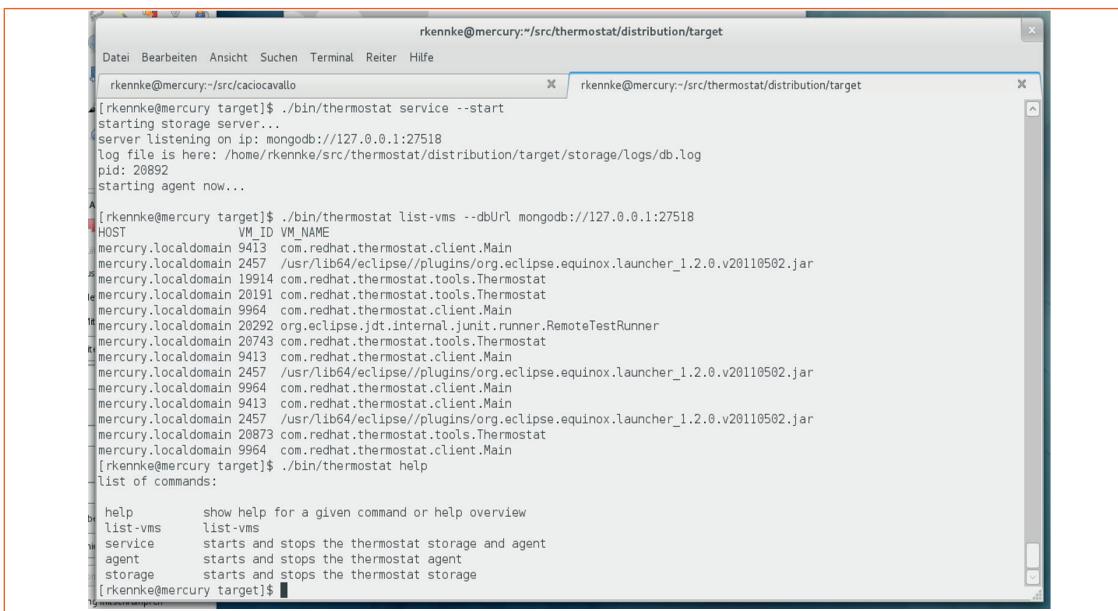
En la medida que las primitivas y paradigmas de interacción han ido apareciendo, de la misma manera aparecieron distintos estilos y objetos en las interfaces de usuario.

Los estilos de interfaces son los siguientes:

a) La interfaz por línea de comandos

Es considerado el primer estilo de interfaz de uso generalizado. Actualmente se sigue utilizando, en algunos contextos, donde los usuarios son expertos (ej. operadores de sistemas específicos, para dar comandos a nivel de sistema operativo como el caso de Unix, etc.).

El usuario “introduce” instrucciones directamente al computador mediante oraciones (palabras, verbos, caracteres especiales y teclas de función) con una sintaxis bien definida, por lo que no “perdona” los errores. Un ejemplo de este estilo se muestra en la Figura 17.



```

rkennke@mercury:~/src/thermostat/distribution/target
Datei Bearbeiten Ansicht Suchen Terminal Reiter Hilfe
[rkennke@mercury target]$ ./bin/thermostat service --start
starting storage server...
server listening on ip: mongodb://127.0.0.1:27518
log file is here: /home/rkennke/src/thermostat/distribution/target/storage/logs/db.log
pid: 20892
starting agent now...
[rkennke@mercury target]$ ./bin/thermostat list-vms --dbUrl mongodb://127.0.0.1:27518
HOST VM_ID VM_NAME
mercury.localdomain 9413 com.redhat.thermostat.client.Main
mercury.localdomain 2457 /usr/lib64/eclipse/plugins/org.eclipse.equinox.launcher_1.2.0.v20110502.jar
mercury.localdomain 19914 com.redhat.thermostat.tools.Thermostat
mercury.localdomain 20191 com.redhat.thermostat.tools.Thermostat
mercury.localdomain 9964 com.redhat.thermostat.client.Main
mercury.localdomain 20292 org.eclipse.jdt.internal.junit.runner.RemoteTestRunner
mercury.localdomain 20743 com.redhat.thermostat.tools.Thermostat
mercury.localdomain 9413 com.redhat.thermostat.client.Main
mercury.localdomain 2457 /usr/lib64/eclipse/plugins/org.eclipse.equinox.launcher_1.2.0.v20110502.jar
mercury.localdomain 9964 com.redhat.thermostat.client.Main
mercury.localdomain 9413 com.redhat.thermostat.client.Main
mercury.localdomain 2457 /usr/lib64/eclipse/plugins/org.eclipse.equinox.launcher_1.2.0.v20110502.jar
mercury.localdomain 20873 com.redhat.thermostat.tools.Thermostat
mercury.localdomain 9964 com.redhat.thermostat.client.Main
[rkennke@mercury target]$ ./bin/thermostat help
list of commands:

help      show help for a given command or help overview
list-vms  list-vms
service   starts and stops the thermostat storage and agent
agent     starts and stops the thermostat agent
storage   starts and stops the thermostat storage
[rkennke@mercury target]$ 

```

Figura 17. Ejemplo de una interfaz de línea de comandos. Elaboración propia.

b) Formularios

Es un estilo dedicado a la captura de datos específicos. Se proporciona en algunas aplicaciones donde la captura de datos es muy intensa, como el caso de transcripciones de algunas empresas que manejan un número importante de datos. Un ejemplo de este estilo de interfaz se muestra en la Figura 18.



The form is titled "Agregar Congreso" (Add Congress). It is divided into two main sections: "DATOS PRINCIPALES" (Main Data) and "CONFIGURACIÓN GENERAL" (General Configuration).

DATOS PRINCIPALES:

- Clave: Text input field.
- Nombre Completo: Text input field.
- Sede: Text input field.
- Tipo de Evento: A dropdown menu with the placeholder "Seleccione una opción" (Select an option).
- Fecha de Inicio: Text input field.
- Fecha de Término: Text input field.
- Fecha Larga: Text input field.
- Descripción del Evento (No mayor a 500 caracteres): Text area for event description.

CONFIGURACIÓN GENERAL:

- Tipo de Moneda: A dropdown menu with the placeholder "Seleccione una opción".
- Costo (incluyendo impuestos): Text input field with value "0.00".
- Tipo de cambio a MXN: Text input field with value "0.00".
- STATUS:**

 - Status del Evento: A checkbox labeled "Activo".
 - Activar idioma Inglés: A checkbox labeled "Activo".
 - Registro Normal: A checked checkbox labeled "Registro Normal".
 - Registro con claves: An unchecked checkbox labeled "Registro con claves".

- Link de Retorno (ej. http://www.dominio.com): Text input field.

A green "guardar" (Save) button is located at the bottom right of the form.

Figura 18. Ejemplo de una interfaz basada en uso de formularios. Elaboración propia.

c) Menús

Un menú es un conjunto de opciones visualizadas en algún dispositivo. El usuario, al seleccionar una opción, desencadena la ejecución de una acción asociada a ese ítem. Los menús se organizan en forma jerárquica, o en base a las tareas que forman parte de una funcionalidad. Algunos ejemplos de este estilo se visualizan en la Figura 19.

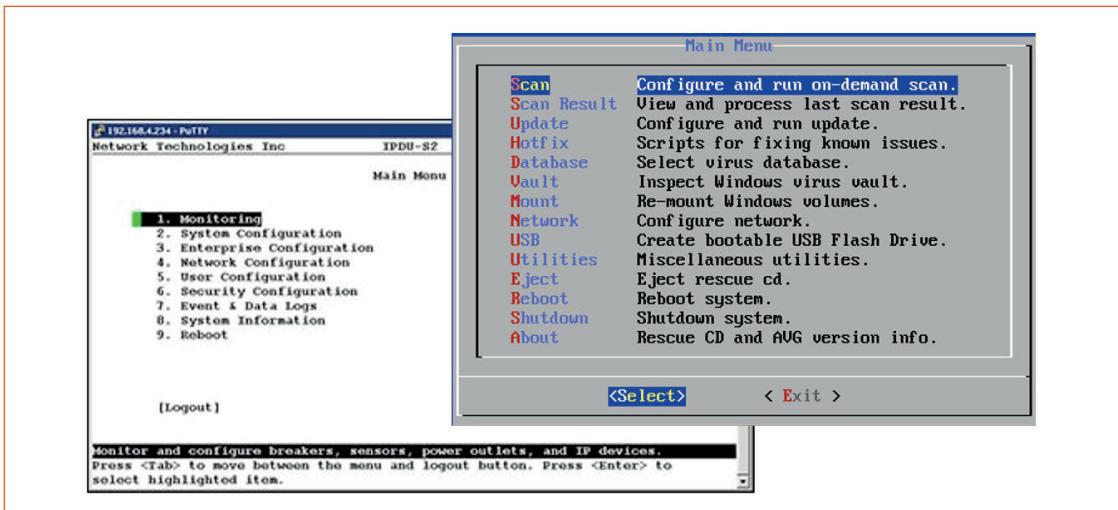


Figura 19. Ejemplo de una interfaz basada en menús. Elaboración propia.

d) Interfaces gráficas (GUI)

Este estilo se originó de la combinación del paradigma WIMP (ventanas, iconos, menús y apuntadores), el paradigma metáfora y el de manipulación directa. En este último se reconoce fielmente la utilización de todas las primitivas de interacción. Es un estilo muy elaborado, pues hace uso de muchos objetos que el diseñador debe conocer muy bien para incorporarlo en el momento del diseño de interfaces de este estilo. Un ejemplo se muestra en la Figura 20.



Figura 20. Ejemplo de una interfaz de manipulación directa. Elaboración propia. Imagen del iPad recuperada de Apple Computer Inc.

e) Interfaces multimodales - Lenguaje, gestos y más

Son interfaces cuyas implementaciones incluyen distintas capas para hacer posible el uso del habla, oído y gestos para poder interactuar con un dispositivo. Algunas de estas capas se dedican al reconocimiento de comandos de voz y sonido, reconocimiento facial, navegación por gestos y ojos, entre otras.

Se incorporan generalmente varios dispositivos o capas de *hardware* especial (para entrada y salida) tales como cámaras, sensores y micrófonos. Lo primordial, sin embargo, con este estilo es la cantidad de algoritmos en el *back-end* que hacen todo el trabajo de reconocimiento. El uso de este estilo de interfaz comenzó en las agencias gubernamentales de seguridad para la identificación de sus empleados. Hoy día aparecen en dispositivos de uso diario como *smartphones*, tabletas, computadores portátiles y otros electrodomésticos (domótica).

f) Interfaces con interacción asistida

La interacción asistida usa la metáfora del asistente personal o agente que colabora con el usuario. Desde el punto de vista de tareas de interacción, el usuario no lleva el control, sino que hace uso de un mediador, el asistente.

Un ejemplo que tiene varios años en uso es SIRI (de Apple Computer Inc), como agente/asistente en el ambiente iOS que está presente en dispositivos Apple. Una de las principales características de SIRI, es que utiliza procesamiento de lenguaje natural para poder interactuar con el usuario final del dispositivo móvil. Aplicaciones similares son Alexa para la plataforma Amazon Echo y Google Assistant para Android.

g) Realidad aumentada

El concepto de realidad aumentada aparece en el año 1992 y su creador es Tom Caudell. Fue utilizado para describir una pantalla que usarían los técnicos electricistas de la empresa Boeing que mezclaba gráficos virtuales con la realidad física. Este sistema les permitiría aumentar la eficiencia de su trabajo al facilitarles de alguna forma las operaciones sobre las tareas a realizar. La realidad aumentada agrega información sintética a la realidad que se visualiza a través de un dispositivo (“monitor” de alguna dimensión). La diferencia principal entre realidad virtual y realidad aumentada es la siguiente: por una parte, la realidad virtual implica inmersión del usuario en un mundo totalmente virtual, en cambio la realidad aumentada implica mantenerse en el mundo real con información “agregada” en forma virtual.

En este estilo de interfaces, según la tecnología, puede ser necesario la utilización de dispositivos especiales, tales como gafas para visualización de video y de óptica *see-through*, proyectores, monitores y dispositivos móviles o *HandHeld*. En algunos casos suelen utilizarse aplicaciones *ad hoc* capaces de “reconocer” el objeto (por ejemplo, con geolocalización, códigos QR asignado al objeto) sobre el cual se agrega la información “virtual”.

Este estilo de interfaces se dio a conocer en el mundo del entretenimiento (ej. el famoso juego *Pokemon*). Se utiliza como herramienta educativa, como medio para mejorar las visitas guiadas

en museos y ciudades, entre otros usos. En la Figura 21 se pueden observar dos ejemplos de uso de la realidad aumentada:

- a. Para asistir al diseño arquitectónico,
- b. Uso educativo.

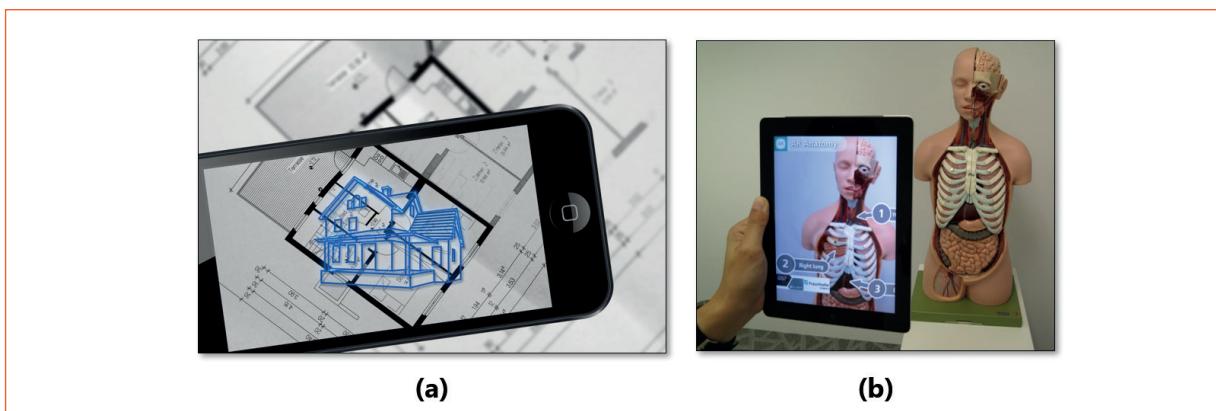


Figura 21. Ejemplo de una interfaz de realidad aumentada para distintos usos. Elaboración propia.

Los estilos que se han descrito explican el camino desde las interfaces codificadas y estrictas orientadas a líneas de comando, (CLI por sus siglas en inglés *Comand Line Interfaces*), pasando por las interfaces gráficas (GUI por sus siglas en inglés, *Graphic User Interfaces*) para llegar a la interfaces naturales (NUI por sus siglas en inglés *Natural User Interfaces*). Esta última es aquella donde el usuario interactúa con un dispositivo o aplicación sin utilizar “muchos” dispositivos externos (ratón, teclado, lápiz óptico, etc.) y en su lugar hace uso de movimientos/gestos de alguna parte del cuerpo como las manos, que sirven de elemento de control y activación de tareas. En la Figura 22 se resumen estos tres conceptos.

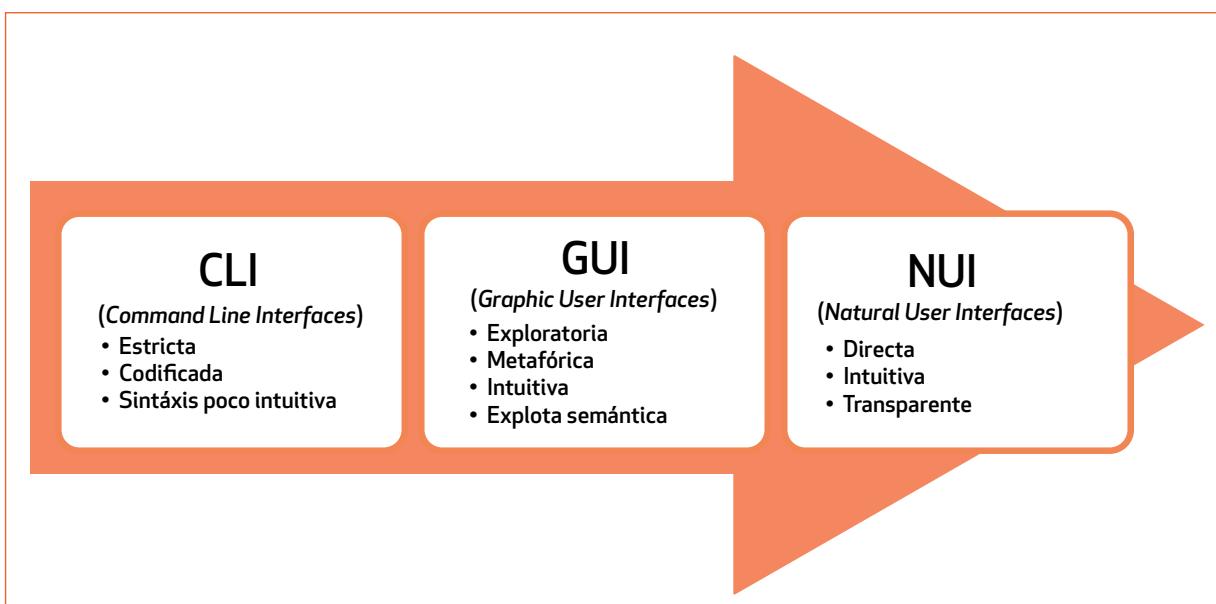


Figura 22. De las interfaces estrictas a las interfaces naturales. Adaptado de interaction-design.org (2019a).

6.2.2. Objetos de las interfaces

Los estilos de interfaces descritos anteriormente hacen uso de objetos distintos, cada uno de los cuales tienen un objetivo y una semántica (y praxis) bien definida. **Reconocer el uso adecuado de cada uno de los objetos se traduce en lograr interfaces usables y accesibles.**

Considerando que la mayoría de las aplicaciones utilizan interfaces gráficas se hará mayor énfasis en los objetos de estas interfaces, teniendo en cuenta el paradigma de manipulación directa y uso de metáforas. Los objetos se clasifican, por su uso, de la siguiente manera:

Comandos

Son el conjunto de objetos que se utilizan para activar funcionalidades y tareas. Entre los más utilizados están los Menús, Menús contextuales, *Pie Menus* (radiales), Botones, *Drop down*. Se utilizan en primitivas de interacciones de posicionamiento y selección. En la Figura 23 se muestran algunos de estos objetos.

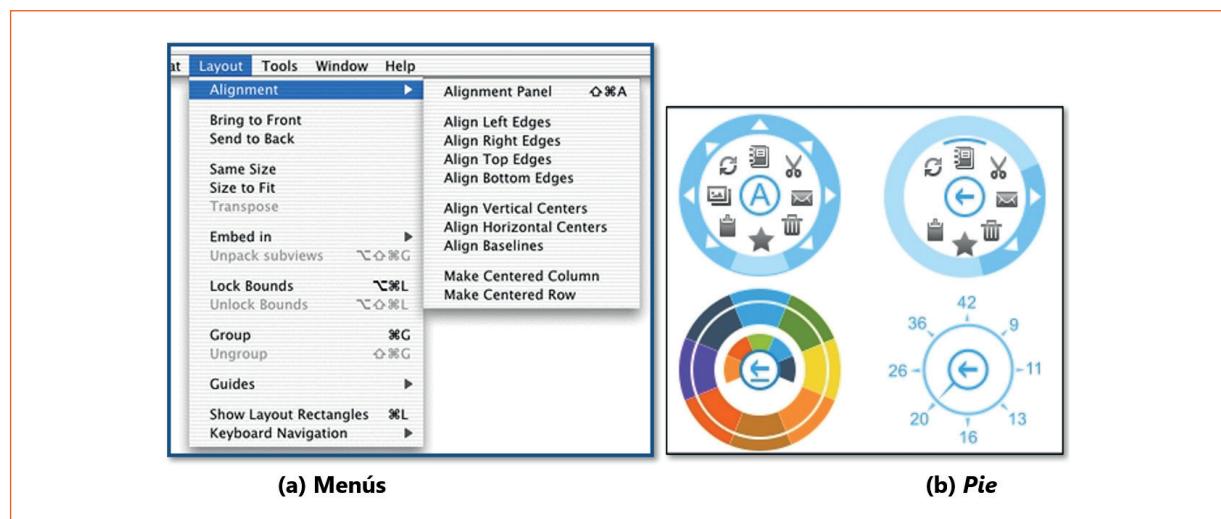


Figura 23. Ejemplos de objetos para la activación de comandos. Elaboración propia.

Entrada y salida de datos

Se utilizan para realizar las primitivas para seleccionar (un dato), insertar texto e insertar valor. Entre los objetos más utilizados están los *Check box*, *Radio button*, *Combo box*, *Drop-down list*, *Grid view*, *List box*, *Scrollbar*, *Spinner*, *Sliders*, *Search box* y *Textbox*, *Date Picker*, *Time Picker*, *Knob Control*. En la Figura 24 se destacan los *checkbox*, que permiten selección múltiple de opciones, y los *radio button*, *list box* y *combo box* donde se permite una única selección. En la Figura 25 se muestran otros objetos para introducir valores.

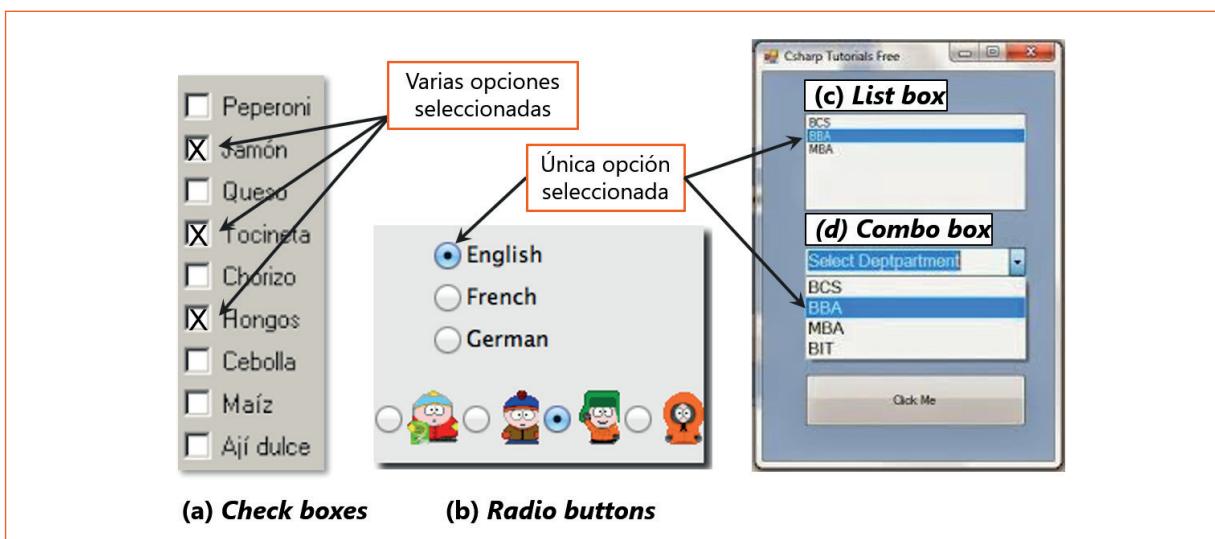


Figura 24. Ejemplos de objetos utilizados para la entrada y salida de datos. Se destacan aquellos objetos que permiten selección múltiple (a) y aquellas para seleccionar una sola opción (b, c y d). Elaboración propia.

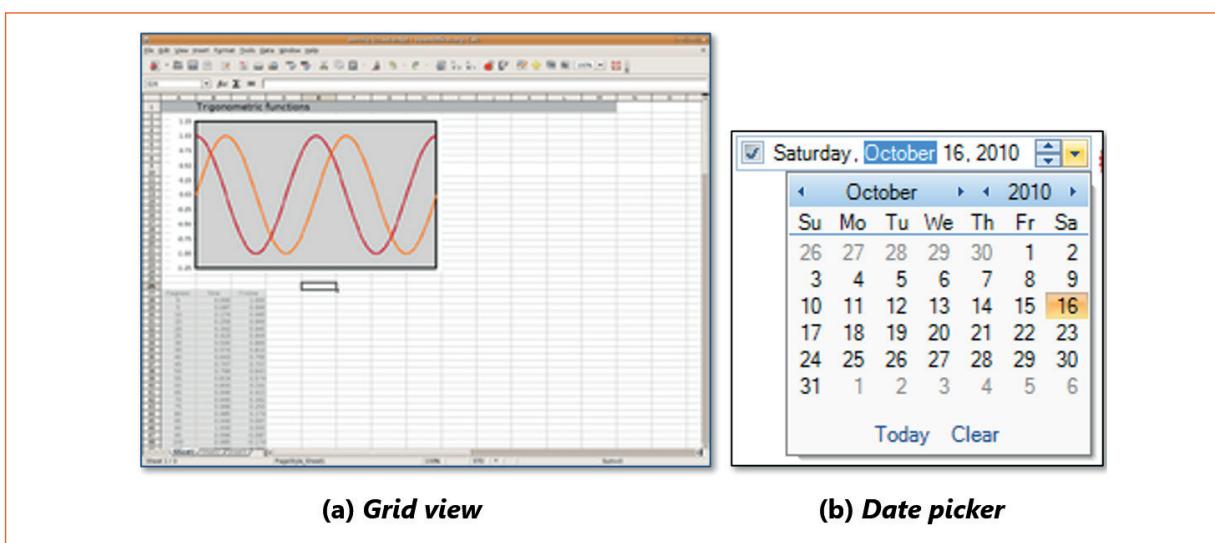


Figura 25. Ejemplos de objetos utilizados para introducir valores. Elaboración propia.

Informacionales

Estos objetos tienen el objetivo de brindar información (y retroalimentación) al usuario durante las tareas de interacción, mostrando la situación de la aplicación o dispositivo. Los más utilizados son los iconos, las barras de status, las etiquetas, las barras de progreso, los *tooltips*, los *balloon help*. En la Figura 26 se muestran algunos de estos objetos.



Figura 26. Ejemplos de objetos informacionales. Caso (a) para retroalimentación. Caso (b) proporcionan información. Elaboración propia.

Contenedores

Son “grandes” elementos en los cuales se insertan objetos para realizar comandos (para la entrada y salida de datos, informacionales, etc.). Un ejemplo de estos objetos son las ventanas, los *accordions*, las barras de menú, los *ribbons* y los *toolbars*. En las Figuras 27 y 28 se muestran algunos de estos objetos.



Figura 27. Ejemplos de objetos contenedores: ventanas de distintos tipos y usos. Elaboración propia.

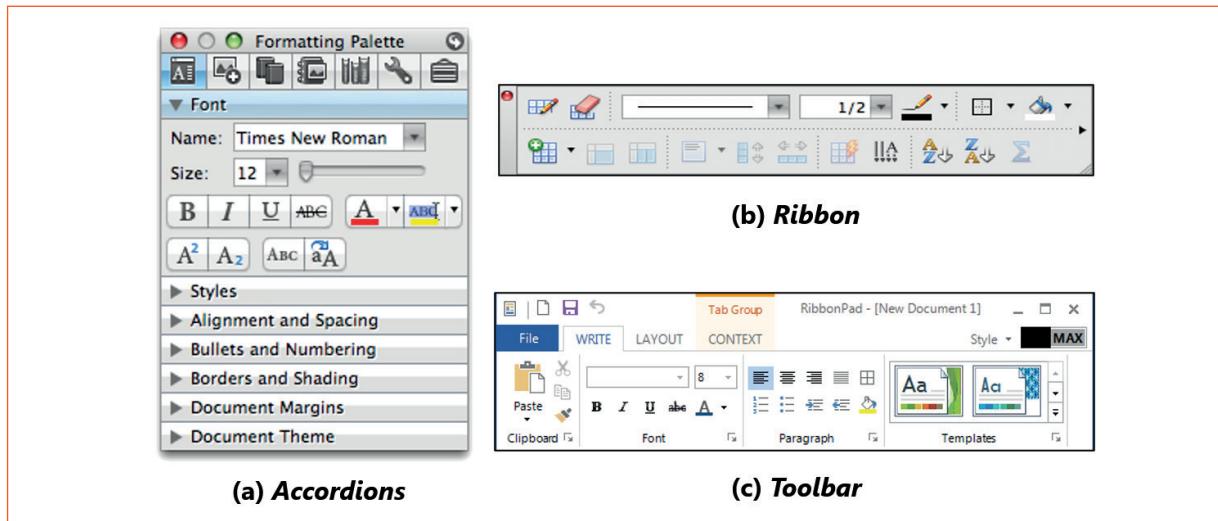


Figura 28. Ejemplos de objetos contenidos. Elaboración propia.

Navegación

Como lo indica su nombre, son objetos que orientan y ayudan al usuario en el caso que se requiera navegar por distintos contextos dentro de una funcionalidad o grupos de tareas en una aplicación. Los más representativos son *Address bar*, *Breadcrumbs*, enlaces hipertexto, *Treeview* (dentro de un sistema de ficheros).

Ventanas especiales

Son ventanas que se utilizan para mostrar cambios de contexto, alertas al usuario, y mostrar utilidades específicas de la aplicación. Entre estas ventanas se encuentran *About box*, *Alert dialog box*, *Dialog box*, *File dialog*, *Inspector window*, *Modal window*, *Palette window*. Algunas de ellas se muestran en la Figura 29.

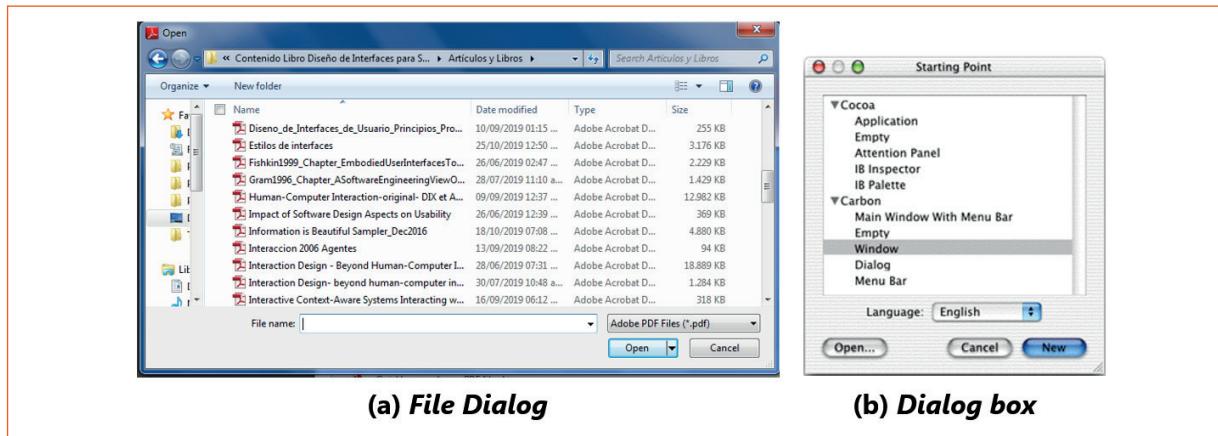


Figura 29. Ejemplos de ventanas especiales (a) *File dialog* muy utilizado para eventos sobre ficheros. (b) *Dialog box*, para alertar al usuario y con usos muy específicos. Elaboración propia.

Dentro de esta categoría de ventanas especiales, un caso de interés son las **ventanas de diálogo** (*Dialog box*), no sólo porque son muy utilizadas, sino que incorporan consigo algunas reglas que respetar.

Las ventanas de diálogo pueden “comportarse” (según el requisito de diseño) de dos maneras distintas dependiendo de lo que el diseñador espera del usuario cuando reaccione ante estos objetos. Ocurren las dos siguientes posibilidades:

a. Ventana de diálogo modal

Es un objeto de control gráfico subordinado a la ventana principal de una aplicación. Crea un modo que **deshabilita la ventana principal**, pero la mantiene visible, y la ventana de diálogo modal es una ventana secundaria frente a ella. **Los usuarios deben interactuar con la ventana modal antes de poder regresar a la aplicación principal.** Esto evita “interrumpir” el flujo de trabajo en la ventana principal.

De manera simple, en caso que en un diseño de interfaces se requiera utilizar un diálogo modal es porque la información que en ella se solicitará al usuario **es esencial para continuar**. En la Figura 30 se muestra un caso donde se utilizan diálogos modales.

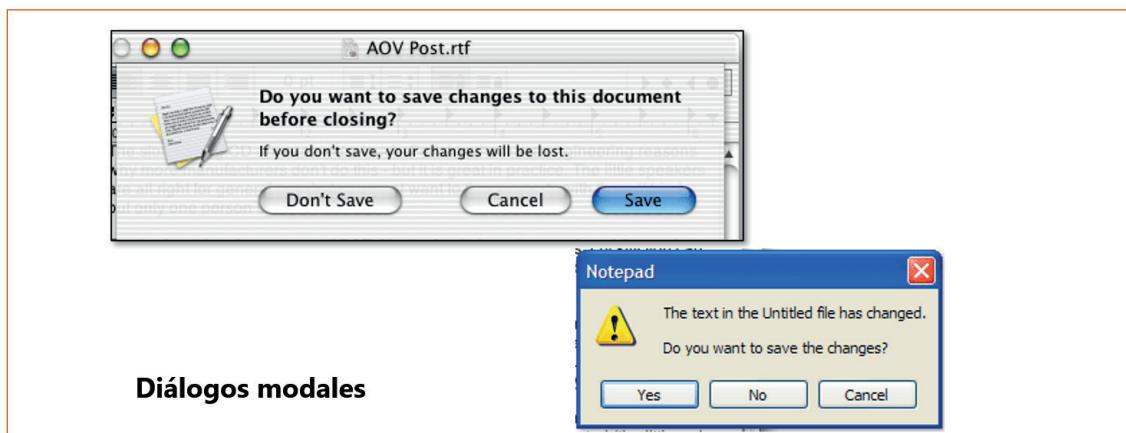


Figura 30. Ejemplos de uso de una ventana de diálogo modal. Elaboración propia.

b. Ventana de diálogo no modal

Una ventana de diálogo no modal **no deshabilita la ventana principal de la aplicación**, por lo que el usuario puede decidir cuál de ellas atender sin que se altere el trabajo en la aplicación. En otras palabras, **la información que se solicita o la acción que se espera del usuario a través de esta ventana no es esencial para continuar**.

6.2.3. Los mensajes en las interfaces

Además de los objetos descritos anteriormente, existen elementos que muchos no consideran como parte de la interfaces: son los mensajes. Estos objetos pueden hacer que una aplicación sea usable o no, que el usuario sea productivo y eficaz cada vez que necesita activar alguna funcionalidad, etc.

La manera en que se escribe el contenido de un mensaje va más allá de la sintaxis y ortografía utilizada, ya que se relaciona con el léxico utilizado hacia el usuario, y debe considerar la tarea que se está llevando a cabo y el contexto de la misma. El mensaje debe expresar correctamente

las opciones que puede seleccionar el usuario, sin generar ningún tipo de duda. En la Figura 31 se muestra un mensaje (a) que no informa totalmente la situación que debe encarar el usuario. La solución adecuada se muestra en la parte (b).

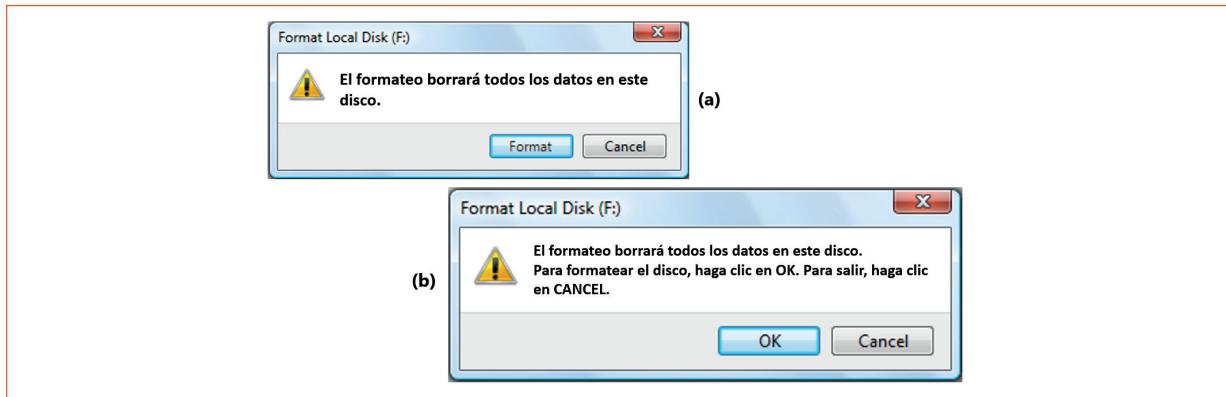


Figura 31. Uso de mensajes en ventanas de diálogos modales. Elaboración propia.

En un mensaje, que generalmente aparece en una ventana de diálogo, existen dos áreas muy diferenciadas: el texto del mensaje y las “palabras claves” utilizadas como etiquetas de los botones que puede accionar el usuario según sus necesidades. Los mensajes pueden agruparse en **4 tipos:**

- Mensajes de información
- Mensajes de advertencia
- Mensajes con preguntas
- Mensajes de error

Generalmente para los mensajes de información deben utilizarse ventanas de diálogo no modales, mientras que los otros deben ser modales, pues la decisión que tome el usuario puede ser inconveniente, o cambiar el estado de una aplicación durante la funcionalidad que se desea ejecutar.

En el **Anexo D** se describen algunas consideraciones en **la utilización de mensajes en las interfaces**.

6.3. Principios, directrices, guías y estándares de diseño de interfaces

La disciplina de interacción humano-computador y el diseño de interfaces de usuario se han alimentado de distintas disciplinas que han creado técnicas y estrategias, conocimiento y buenas prácticas de manera que puedan servir de recomendaciones para los diseñadores. A lo largo de este contenido se ha utilizado el vocablo usabilidad como característica de calidad de un *software* respecto a su utilización (lo cual implica interacción del usuario y diseño de la interfaz). Sin embargo, **otra forma de conceptualizar (y mejorar) la usabilidad es en términos de principios de diseño, uso de estándares y buenas prácticas, y directrices**. Dependiendo de la granularidad, el grado de rigurosidad y normativa, y el tipo de organización que presenta esas recomendaciones, estas se dividen en principios, guías y estándares.

1. Principios

Los principios son conceptos de alto nivel destinados a orientar a los diseñadores a pensar en diferentes aspectos de sus diseños. **Los principios de diseño se derivan de una combinación de conocimiento basado en la teoría, experiencia y sentido común. Tienden a escribirse de manera prescriptiva, sugiriendo a los diseñadores qué objetos proporcionar y qué evitar en la interfaz, qué hacer y qué no hacer en el diseño de interacción.** Más específicamente, están destinados a ayudar a los diseñadores a explicar y mejorar el diseño (Preece et al, 2002).

Los principios no existen con la intención de especificar cómo diseñar una interfaz real (por ejemplo, indicar al diseñador cómo implementar una metáfora particular, cómo estructurar una barra de menú, etc.), sino que actúan más como un conjunto de recordatorios para los diseñadores, asegurando que hayan proporcionado objetos adecuados en la interfaz.

Están sustentados en la investigación de como los usuarios actúan, aprenden y trabajan con un dispositivo. Un ejemplo de principio de diseño es el siguiente:

Minimizar la carga cognitiva del usuario

Para lograr una interfaz fácil de aprender y que el usuario no deba retener mucha información, hay que minimizar la carga cognitiva del usuario. Para atender este principio se deben considerar los siguientes aspectos:

- Confiar en el reconocimiento vs memoria.
- Proporcionar pistas visuales.
- Proporcionar opciones por defecto.
- Proporcionar atajos (en caso de usuarios más experimentados).
- Promover la sintaxis objeto – acción.
- Emplear metáforas del mundo real.
- Emplear la revelación progresiva para evitar abrumar al usuario.
- Promover la claridad visual.

En el **Anexo E** se presenta una lista (no exhaustiva) de principios y los aspectos a considerar en cada uno de ellos. **Para profundizar en este contexto, en (Scott y Neil, 2009) se presenta una recopilación completa con la descripción de principios de diseño de interacción e interfaces para ambiente Web.**

2. Directrices

Las directrices surgen de las buenas prácticas recogidas de muchas experiencias. **En ellas se recomiendan acciones basándose en un conjunto de principios de diseño.** Generalmente requieren de menos experiencia para entenderlas y utilizarlas pues son más específicas que los principios.

Cada directriz es un objetivo, aunque no se describe explícitamente como alcanzarlo. Permiten asegurar que se mantenga la consistencia de la interfaz en cada uno de sus componentes y usos, y pueden estar asociados a familias de productos. En este sentido muchas empresas de *software* han creado sus respectivas directivas de forma que puedan asesorar a sus desarrolladores en el diseño de las interfaces (así como en el caso de otros componentes del producto).

Ejemplo. Una directriz a seguir, respecto al principio "Minimizar la carga cognitiva del usuario", es el siguiente: "Proporcionar ayuda contextual para cada ítem de menú u objeto sobre el cual puede posicionarse un cursor".

3. Estándares

Los estándares son reglas o recomendaciones que también se basan en principios y buenas prácticas. **A diferencia de las directrices, surgen de estudios y acuerdos realizados por organizaciones autorizadas ya sea a nivel nacional o internacional. Generalmente los estándares están asociados a áreas de aplicación específica e industrias de manufactura.**

En el ámbito de la informática se han definido estándares tanto desde el punto de vista de los equipos y su utilización general, el proceso de desarrollo de *software*, así como con las interfaces de usuario. Existe una gran variedad de estándares que se aplican ya sea para dispositivos hasta aquellos aspectos ergonómicos como la inclinación del teclado, la distancia hacia el monitor, la disposición de las letras en el teclado (por ejemplo, el teclado QWERTY se considera un estándar).

Los estándares son utilizados por los desarrolladores de *software* para lograr cumplir con requisitos de calidad. En el caso de las interfaces el aspecto de calidad es la usabilidad del producto. Si se respetan los estándares, de alguna manera se crea una estrategia que puede ser repetida (y mejorada), que puede ser incorporada en los procesos de desarrollo de forma que las empresas vayan alcanzando mayores niveles de calidad.

Existen las siguientes familias de estándares (Martínez y Cueva, 2001):

- **Estándares con estatus legal:** son aquellos originados por organismos internacionales tales como ISO (*International Organization for Standardization*), ANSI (*American National Standards Institute*), IEEE (*Institute of Electrical and Electronics Engineers*), CEN (*Comité Européen de Normalisation*), el W3C (*World Wide Web Consortium*), IETF (*Internet Engineering Task Force*), entre otros.

En el ámbito de interacción usuario-computador e interfaces de usuario el estándar más conocido es la norma ISO 25010 (Modelo de calidad de *software*, en el apartado Usabilidad) (iso2500.com, 2019) y la norma ISO 9241-11:2018 (iso.org, 2019a), (iso.org, 2019b).

- **Estándares de facto:** no han sido legitimados por organizaciones de estandarización, y son creados a partir de las prácticas en distintas industrias de renombre en el mercado, o de centros de investigación de universidades. En el momento en el cual llegan a tener un uso importante y generalizado, se convierten en estándares legales. Algunos ejemplos representativos en el ámbito de las tecnologías de información y comunicación son: los protocolos TCP/IP, las interfaces MIDI de los instrumentos musicales digitales, la disposición de las teclas en un teclado de teléfonos, el formato de teclado QWERTY, etc.
- **Estándares de carácter propietario:** son aquellos que son propiedad de una entidad única (generalmente fabricantes de tecnologías). Si algún estándar propietario logra tener una gran aceptación del mercado puede convertirse en un estándar de facto y llegar a tener un estatus legal. Un ejemplo son algunos sistemas operativos que se utilizan en *smartphones*.

4. Guías de estilo

Son un conjunto de recomendaciones que las empresas fabricantes ponen a la disposición de desarrolladores de productos para sus dispositivos. Pueden ser de dos tipos:

- **Las guías de estilo comercial**, que pueden transformarse en estándares de facto.
- **Las guías de estilo corporativo** para uso exclusivo en productos de la plataforma de la empresa. **Se centran en presentaciones y comportamientos similares, y utilizan técnicas que se implementan en toda la gama de productos.** Dentro de este tipo de guías se presentan también las guías de diseño para productos relacionados (ejemplo, la suite de productos de Microsoft Office[©]) o para productos individuales. Es conveniente aclarar, (usando el ejemplo de Microsoft Office[©]) que dependiendo de la plataforma de despliegue final (Windows[©] o MacOS[©]) se siguen los estándares propietario respectivos.

Parte del objetivo de las guías de estilo es que aseguran la consistencia entre los objetos que se presentan en cualquier producto del mismo fabricante. Como ejemplo, siempre en el ámbito informático, se pueden nombrar los entornos de Apple (iOS, MacOS), Microsoft (en sistema operativo Windows), UNIX, Android, etc.

Las guías de estilo en ambiente Web han sido promocionadas por la WAI (*Web Accessibility Initiative*) y la W3C (*World Wide Web Consortium*).

En la Figura 32 se muestra la jerarquía entre cada uno de los conceptos desde estándares hasta guías de estilo de productos.

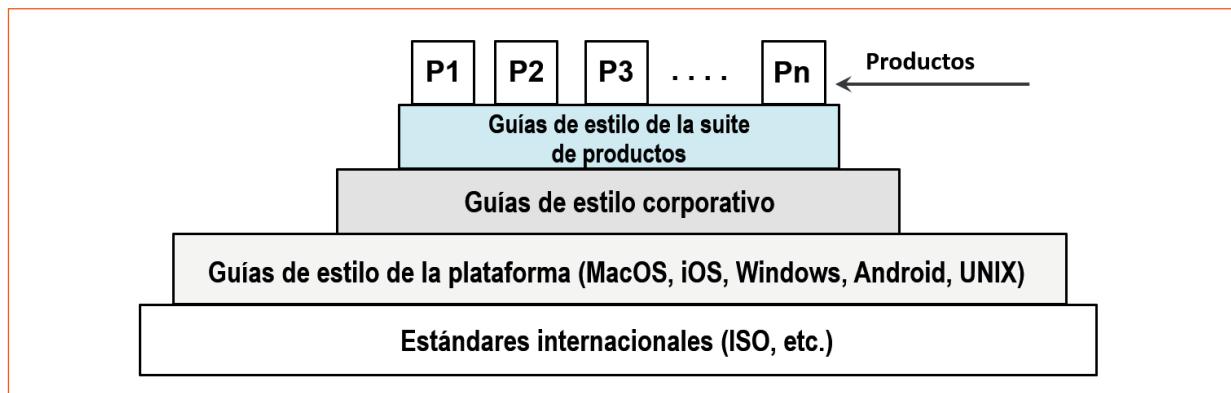


Figura 32. Jerarquía de uso de estándares y guías de estilo en el diseño de interfaces. Adaptado de Martínez y Cueva (2001, p. 25).

Actividades recomendadas

Considerando las soluciones indicadas para el caso de estudio, se propone:

- Diseñar la interfaz de usuario para el componente Web de **MyEnchantedMuseum – Interactive Complex**. Indique todas las herramientas de navegación y las funcionalidades que pueden llevarse a cabo. Diseñe en detalle la interfaz para que el usuario pueda “recuperar” el recorrido virtual realizado, si así lo desea.
- Diseñar la interfaz de usuario para el componente de guía de visitantes (Solución B) de **MyEnchantedMuseum – Interactive Complex**. El diseño debe considerar al menos tres funcionalidades/tareas que puede llevar a cabo el usuario.

Para cada una de esas actividades, justifique las decisiones de usos de elementos de interfaz, en base a principios de diseño, guías, estándares y otras normas que considere necesarias.

Tema 7. Calidad de los sistemas interactivos: usabilidad, accesibilidad y affordance

“...Todos los aspectos de cómo las personas usan un producto interactivo: cómo se siente en sus manos, qué tan bien entienden cómo funciona, cómo se sienten al usarlo, qué tan bien sirve a sus propósitos y qué tan bien encaja en todo el contexto en el que lo están utilizando”.

[De eso se trata...][nd]

(Alben, 1996, p.12)

Un **sistema interactivo de calidad** debe cumplir con **características** tales como:

- a. **Debe ser flexible.** El sistema debe ser lo suficientemente flexible de forma que permita una variedad de preferencias. Las medidas de flexibilidad se utilizan para evaluar el grado en que un producto o sistema se puede utilizar con niveles aceptables de efectividad, eficiencia, satisfacción y ausencia de riesgos en contextos más allá de los inicialmente especificados en los requisitos del sistema.
- b. **Debe ser robusto.** Un sistema es robusto si un usuario recibe los medios para lograr sus objetivos, evaluar su progreso y recuperarse de cualquier error cometido.

- c. **Debe ser útil.** Un sistema interactivo útil es aquel que permite a los usuarios alcanzar sus objetivos. Cuando se diseña un sistema interactivo, hay que conocer los objetivos y la funcionalidad asociada del sistema para que éste sea útil.

La evaluación de cualquier producto es parte integral del proceso de diseño. Se deben recopilar y analizar datos sobre las actuaciones (experiencias) de usuarios al interactuar con un artefacto, ya sea un boceto general, un prototipo operacional, un sistema de *software*, etc. En el ámbito de los sistemas interactivos, la evaluación se enfoca en el respeto de las **características de calidad como la usabilidad, la accesibilidad y el affordance** (considerado por algunos investigadores como un principio de diseño de interfaces).

Cabe recalcar que muchos países estipulan requisitos de salud y seguridad para el lugar de trabajo. Dichos requisitos se han desarrollado a partir del estudio de la "ergonomía" que analiza los aspectos físicos del diseño de productos. Si bien se comenzó a estudiar la ergonomía durante la segunda guerra mundial, sus conclusiones tardaron unos 30 años en filtrarse a los requisitos legales. Por la misma medida, los requisitos de la interacción comenzaron a aparecer en la ley. De hecho, la **Directiva CE 90/270/CEE** (EUR-Lex, 1990) exige que los empleadores al diseñar, seleccionar, poner en servicio o modificar *software* debe asegurarse que:

- a. Es adecuado para la tarea.
- b. Es fácil de usar y es adaptable.
- c. Proporciona retroalimentación.
- d. Muestra información en formato y a un ritmo adecuado para el usuario.
- e. Se ajusta a los "principios de ergonomía del *software*" y principios de diseño.

La investigación en el área ha crecido y desarrollado de manera tal que se ha creado la **Ingeniería de la Usabilidad, conocida como el conjunto sistematizado de métodos, técnicas y fundamentos teóricos que aseguren el cumplimiento de los niveles de usabilidad requeridos por un sistema**. Las investigaciones y avances sobre usabilidad aumentaron en forma exponencial con el crecimiento e implementaciones de sistemas interactivos en el web. Se debe principalmente a la complejidad del ambiente web y al cambio de paradigma que este ha generado, en especial el concepto de hipertexto donde un aspecto de interacción lo constituye la navegación.

Otro concepto fundamental cuando se desarrollan productos interactivos de uso masivo es la **accesibilidad**, entendida como **la capacidad de un producto que permite que sea utilizado por usuarios con determinadas características y discapacidades**.

El concepto de accesibilidad surge desde el momento de la aparición del web. Berners-Lee se refirió prácticamente a esa cualidad en su frase "El poder del web está en su **universalidad. El acceso** de todos, independientemente de la discapacidad, es un aspecto esencial" (W3C, 1997).

Según la W3C (*WWW Consortium*) "la accesibilidad web significa que los sitios web, las herramientas y las tecnologías están diseñadas y desarrolladas para que las personas con discapacidad puedan

usarlas. Más específicamente, las personas puedan: percibir, comprender, navegar e interactuar con la web y contribuir a la web.

La accesibilidad web abarca todas las discapacidades que afectan el acceso a la web, incluyendo: auditivo, cognitivo, neurológico, físico, habla y visual" (W3C, 2019).

La WAI (*Web Accessibility Initiative*) Iniciativa de Accesibilidad Web del W3C desarrolla especificaciones técnicas, pautas y recursos de apoyo que describen soluciones de accesibilidad. Las especificaciones técnicas se consideran estándares internacionales para la accesibilidad web, por ejemplo, la WCAG 2.0 que también es un estándar ISO: el ISO / IEC 40500.

Un aspecto importante que no siempre es considerado como una propiedad de producto de calidad es la **asequibilidad** (*affordance*).

En su libro "*The Design of Everyday Things Revised and Extended Edition*" Donald Norman (1988, p.11) describe la asequibilidad como "una propiedad de un objeto que 'contiene información' sobre lo que se puede hacer con él. Son elementos que dan una pista de cómo los usuarios pueden interactuar con algo, sin importar lo físico o lo digital. Por ejemplo, cuando [alguien] ve una manija de la puerta, 'ella trae consigo' el hecho que se puede usar para abrir la puerta. ... Los objetos físicos transmiten información importante sobre cómo las personas pueden interactuar con ellos, una propiedad que denomino "*affordance* [asequibilidad]".

El uso de la asequibilidad (*affordance*) no se limita al diseño de objetos físicos. De hecho, el concepto ha sido especialmente atractivo para los diseñadores de interfaces gráficas de usuario. En comparación con los diseñadores industriales tradicionales, las propiedades visuales de los objetos expresan la manera de utilizados. Por lo tanto, parecen estar particularmente bien posicionados para proporcionar lo que Norman (1988) llama "fuertes pistas visuales para el funcionamiento de las cosas". En la Figura 33 se muestra dos ejemplos de objetos de uso cotidiano (superficie horizontal de una estufa): (a) con baja asequibilidad, y (b) con alta asequibilidad.

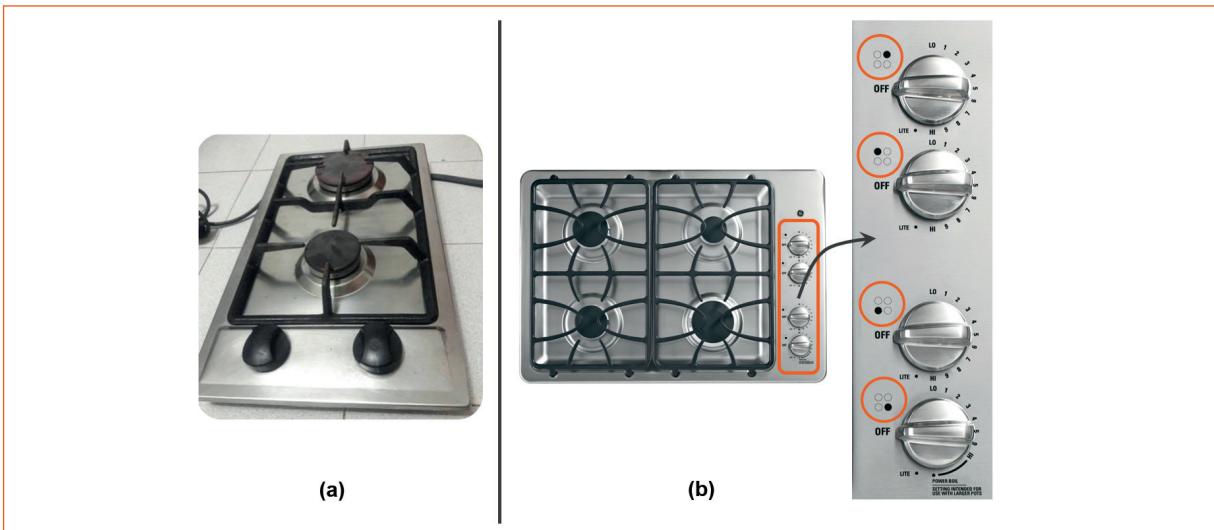


Figura 33. Ejemplo de objetos de la vida real: (a) Baja asequibilidad, imagen recuperada de <http://venta.brick7.co.ve>. (b) Alta asequibilidad, imagen recuperada de General Electric.

Algunos ejemplos de elementos de la interfaz de usuario, que proporcionan estas pistas necesarias, son botones y pestañas que pueden ser seleccionados/activados, los controles deslizantes (*sliders*) capaces de ser arrastrados, controles giratorios, así como otros elementos que lograr sugerir directamente las acciones adecuadas del usuario, sobre tales objetos.

7.1. Usabilidad

La usabilidad es una medida para determinar si los usuarios alcanzan sus objetivos durante la realización de una tarea de manera efectiva, eficaz y satisfactoria, en un contexto de uso específico. Durante el diseño de la interacción, pensando en la usabilidad, debe considerarse la optimización de las interacciones que los usuarios tendrán con los productos.

Los dispositivos como teléfonos inteligentes, tabletas y lectores electrónicos, junto con la omnipresencia de las aplicaciones móviles y la aparición de dispositivos *IoT* (Internet de las cosas), han aumentado la conciencia sobre la usabilidad y el diseño de interacción (Preece et al, 2019).

Es importante darse cuenta de que la usabilidad no es una propiedad única y unidimensional de un producto en el ámbito de la interacción e interfaz de usuario (Nielsen, 1993). La usabilidad tiene múltiples aspectos y se asocia tradicionalmente con los siguientes atributos:

- **Aprendizaje:** el sistema debe ser fácil de aprender para que el usuario pueda comenzar rápidamente a realizar alguna tarea con él. El sistema debe permitir que los nuevos usuarios sean efectivos, eficientes y satisfechos al aprender a usar un nuevo sistema.
- **Eficiencia:** el sistema debe ser eficiente de usar, de modo que una vez que el usuario haya aprendido a usarlo, **logre un alto nivel de productividad**.
- **Recordar vs Memorizar:** el sistema debe ser fácil de recordar. Si un usuario ocasional vuelve a utilizar el sistema después de un período de no haberlo utilizado, no debe aprender todo de nuevo.
- **Errores:** el sistema debe **promover una baja tasa de error**, de modo que los usuarios cometan pocos errores durante su uso. Si los cometen, los **usuarios deben poder recuperarse fácilmente de ellos**. Además, no deben ocurrir errores catastróficos.
- **Satisfacción:** en el estándar ISO/IEC 9241-11, se define la satisfacción como “las actitudes positivas, emociones y/o comodidad como resultado del uso de un sistema, producto o servicio” (Bevan, Carter, Earthy, Geis y Harker, 2016, p. 270). Estos tres elementos se relacionan con las respuestas cognitivas, afectivas y psicomotoras de un individuo. Este es un aspecto relativamente subjetivo.

El proceso de evaluación conlleva una serie de actividades, tales como la planificación del proceso de evaluación, la creación de tareas de prueba, el reclutamiento de participantes y evaluadores, y el análisis de los datos antes de preparar el informe de recomendación de mejoras. El gasto en que se incurre en obtener productos con altos niveles de usabilidad se destina principalmente a esa serie de actividades pues implican una gran cantidad de horas de trabajo y recursos.

7.2. Métodos y técnicas para la evaluación de la usabilidad

La evaluación de usabilidad es la actividad que comprende un conjunto de métodos que analizan la calidad de uso de un sistema interactivo, en diferentes etapas del ciclo de vida del desarrollo (Nielsen, 1993).

La evaluación de usabilidad que se **lleva a cabo durante el proceso de diseño es conocida como evaluación formativa**, y su objetivo es guiar el diseño. No es una evaluación lo suficientemente buena para probar todo el sistema.

Un aspecto importante de la evaluación formativa es que la audiencia para las observaciones y recomendaciones es el propio equipo del proyecto (los interesados), que se utiliza para mejorar de inmediato el diseño del producto o servicio y refinar las especificaciones de desarrollo. Los resultados pueden ser menos formales que en la evaluación sumativa, según las necesidades de los diseñadores, desarrolladores, gerentes y otros participantes del proyecto (Dauchot, 2019).

La evaluación sumativa, en cambio, **se lleva a cabo sobre un producto terminado o un prototipo funcional**. Estas pruebas se realizan para validar aspectos del diseño, y para probar la aceptación del sistema por parte de los usuarios finales.

En la Tabla 20 se presentan algunas circunstancias y el tipo de evaluación que debe considerarse.

Tabla 20
Consideraciones sobre tipos de evaluación de usabilidad

| Consideraciones sobre Tipos de Evaluación de Usabilidad | |
|--|---------------------|
| Preguntas | Tipo de evaluación |
| ¿Cómo está funcionando la interfaz en comparación con productos competidores? | Sumativa |
| ¿Qué problemas de usabilidad existen en la interfaz? | Formativa, sumativa |
| ¿Cómo se compara la interfaz con el punto de referencia de la industria? | Sumativa |
| ¿Cómo se compara la interfaz con el punto de referencia de la industria? | Formativa |
| ¿La interfaz del sistema cumple con los principios de usabilidad reconocidos? | Formativa |
| ¿Es este producto lo suficientemente bueno como para su lanzamiento o despliegue? (Decisión de ir / no ir) | Sumativa |

Nota: Elaboración propia. Contenido tomado de Alita (2019).

En la Figura 34 se muestra una clasificación de los métodos existentes según algunos criterios como la participación o no de usuarios, utilización de herramientas automatizadas, respecto a las técnicas utilizadas, etc.

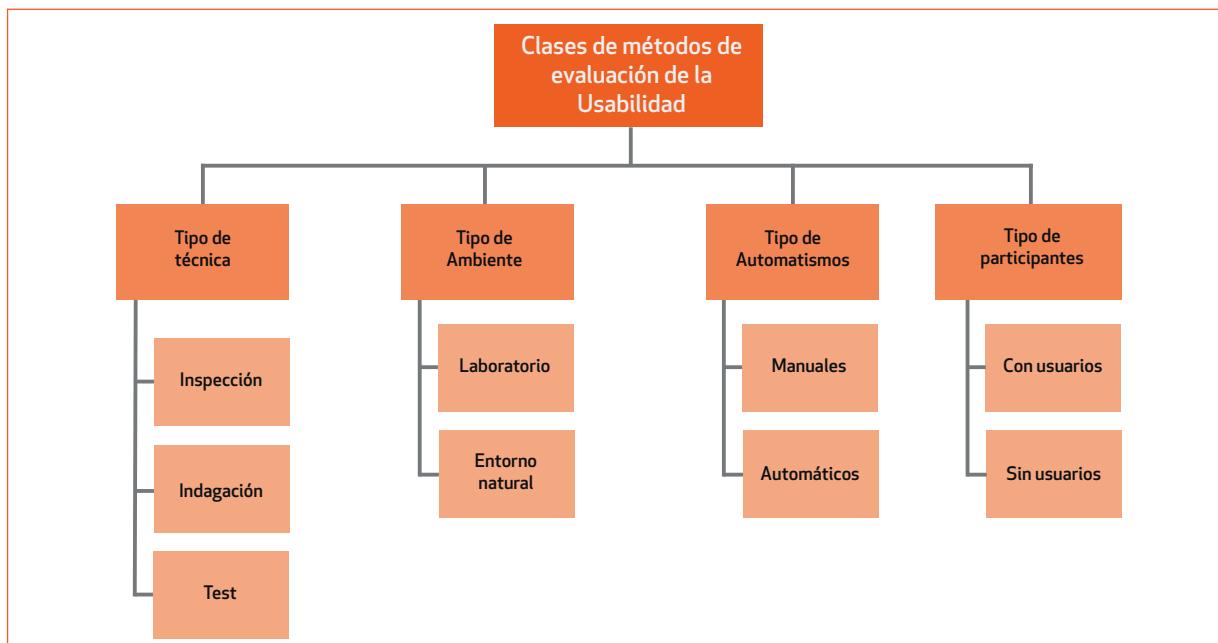


Figura 34. Clases de métodos de evaluación de la usabilidad. Adaptado de Granollers (2019).

Dependiendo del tipo de resultados que arrojen los métodos, también se les clasifica en cualitativos y cuantitativos.

Los **métodos más conocidos para la evaluación de la usabilidad** son los siguientes:

1. **Métodos de inspección:** los evaluadores (expertos en diseño de interacción e interfaces) examinan/inspeccionan aspectos del producto relacionados con la usabilidad que ofrece a sus usuarios. Incluye distintas técnicas, entre ellas:
 - **Evaluación Heurística:** El método fue desarrollado por Nielsen (1993) y consiste en **analizar la conformidad de la interfaz** con unos principios reconocidos de usabilidad (heurísticas). Generalmente los evaluadores se apoyan en las guías y los estándares según los cuales se sustenta el diseño del producto.
 - **Recorrido Cognitivo:** Los evaluadores construyen los escenarios de cada tarea y después asumen el rol del usuario trabajando con esa interfaz (Lewis y Rieman, 2008)
 - **Inspección de características:** se verifica si las características de un producto satisfacen las necesidades y exigencias del usuario.
 - **Inspección de estándares:** inspección exhaustiva de la interfaz para comprobar que cumple con los estándares establecidos.

Todas estas técnicas se pueden combinar o aplicar por separado para evaluar la usabilidad de una aplicación. En la práctica, las técnicas de inspección generalmente las llevan a cabo dos o tres evaluadores con experiencia en usabilidad. Por lo tanto, el proceso de inspección no involucra a usuarios reales, y los resultados vienen dados a juicio profesional de los evaluadores y se considera suficiente. El método de inspección es considerado cualitativo.

Cuando se lleva a cabo una **evaluación heurística**, el informe sobre el producto del proceso, contiene los problemas identificados y también las recomendaciones de cómo solucionarlos. Por este motivo es una de las técnicas más utilizadas dentro de los métodos de inspección.

2. Métodos de indagación: permiten descubrir y aprender para generar ideas de diseño, especialmente para obtener información de usabilidad sobre un producto que se desea producir. Incluye las siguientes técnicas (Massa, De Giusti y Pesado, 2012):

- **Indagación contextual:** su objetivo es comprender cómo los usuarios de los sistemas interactivos realizan sus tareas y las acciones que efectúan (Paganelli y Paternò, 2002).
- **Focus Group:** es una reunión de 6 a 9 participantes (involucrados) donde se discuten aspectos relacionados con el sistema. Un experto en usabilidad realiza la función de moderador.
- **Entrevistas:** pueden ser efectivas para una evaluación profunda, para descubrir motivaciones, valores y experiencias de los usuarios.
- **Cuestionarios:** lista de cuestiones planteadas con el objetivo de rescatar la información relativa a la interacción de los usuarios con el sistema y la percepción subjetiva de estos sobre la experiencia de su uso.
- **Grabación del uso:** se basa en "grabar" o "recoger" (haciendo uso de un programa monitor de sistema) todas las actividades realizadas por el usuario con el sistema para su posterior análisis.

3. Tests: está conformado por un conjunto de técnicas de evaluación sumativa (cuantitativa) y dependen de los usuarios reales para detectar problemas de usabilidad para una interfaz. Los usuarios representativos trabajan en tareas concretas utilizando el sistema (o el prototipo) y los "evaluadores" recogen los resultados para luego analizar la manera en que la interfaz da soporte a los usuarios con sus tareas. Algunas técnicas dentro de este método son:

- **Medida de las prestaciones:** está basado en la toma de medidas objetivas (rendimiento) u otro tipo de aspecto subjetivo que afecte la usabilidad del sistema (Abascal et al, 2001).
- **Pensando en voz alta:** se solicita al usuario que exprese verbalmente qué está pensando, qué no entiende, por qué lleva a cabo una acción o, duda mientras interacciona con el sistema (Paganelli y Paternò, 2002).
- **Eye-tracking:** permite documentar los puntos del sistema que ha estado mirando el usuario en cada momento (Pernice y Nielsen, 2009).

La base de las pruebas (*tests*) es que la retroalimentación de los usuarios reales es más contextualizada y específica, en comparación con el enfoque de inspección que se basa en los aportes generales de unos pocos seleccionados.

Sin embargo, este método requiere un gran número de usuarios reales para garantizar la fiabilidad de los resultados, ya que no todas las opiniones o críticas recopiladas de los usuarios son confiables. Debe asegurarse además de una locación adecuada para las pruebas (laboratorios).

7.3. Proceso de evaluación de usabilidad y algunas métricas

De acuerdo con el enfoque de la Ingeniería de la usabilidad, **una forma para mejorar la usabilidad es abordar el asunto desde las primeras fases del desarrollo de la aplicación.**

La evaluación de la usabilidad es una actividad “transversal”, es decir se lleva a cabo durante todo el proceso de diseño centrado en el usuario, en cada una de sus iteraciones. En otras palabras, el proceso de evaluación está inherentemente ligado al proceso de diseño centrado en el usuario. En la Figura 35 (proceso de diseño centrado en el usuario) se evidencia la actividad de evaluación y los caminos a seguir durante el proceso según los resultados obtenidos.

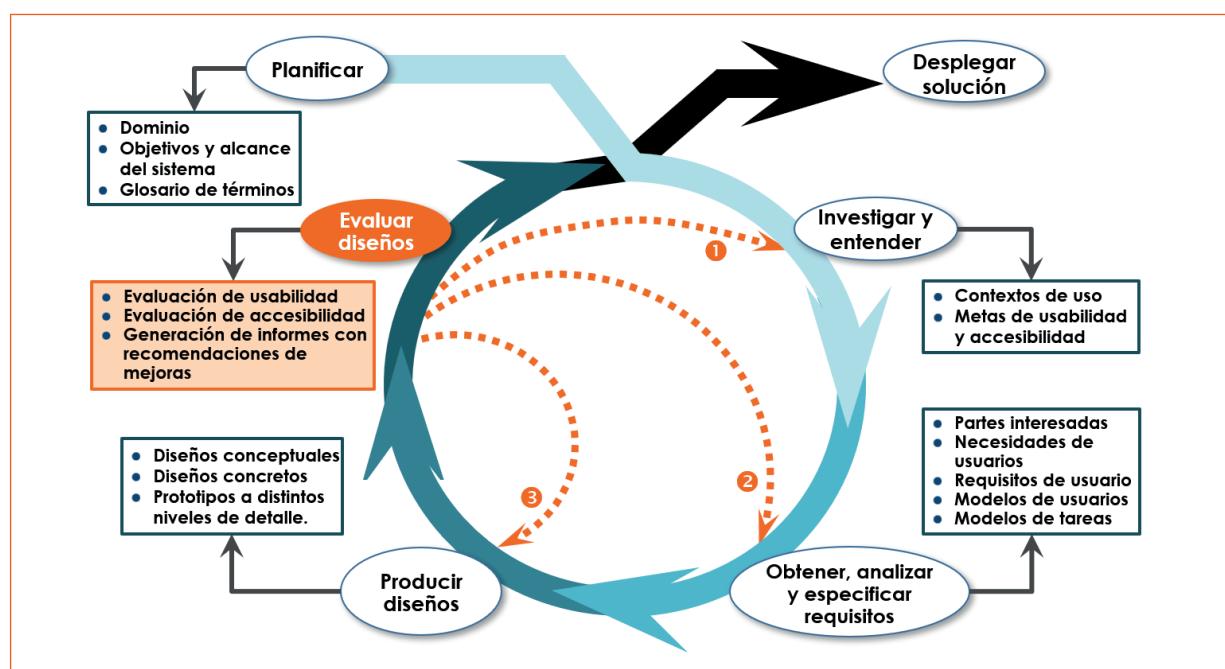


Figura 35. La actividad de evaluación de usabilidad durante el proceso de diseño centrado en el usuario. Elaboración propia.

Como el proceso es iterativo, dependiendo de los resultados obtenidos en cada una de las iteraciones, se vuelve a repetir o refinar los productos (obtenidos) de otras actividades y tareas del desarrollo del sistema. Así pues, es posible tener que abundar lo recolectado respecto al contexto de uso y el dominio de la aplicación (1), o bien revisar los requisitos obtenidos, volver a obtener y analizar nuevos requisitos de usuarios (2), o rehacer los modelos o prototipos (3) para comunicar la solución con los interesados.

Respecto a los aspectos a evaluar, algunos investigadores proponen utilizar listas de verificación basadas en criterios y buenas prácticas. En la Tabla 21, se resumen las “10 reglas de oro” propuestas por (Nielsen, 1993) para el diseño y evaluación de sistemas interactivos. **Esta estrategia se utiliza en los métodos de inspección, como el caso de la evaluación heurística.** Es de notar que cada una

de las reglas propuestas en la tabla, pueden ser descriptas/caracterizadas a un nivel de mayor detalle, y lograr métricas más elaboradas y atinentes a detalles específicos.

Tabla 21
Las 10 reglas de oro para el diseño y evaluación de sistemas interactivos

| Heurística | Descripción |
|--|--|
| 1. Visibilidad del estado del sistema | El sistema siempre debe mantener a los usuarios informados sobre lo que está sucediendo, a través de comentarios apropiados dentro de un tiempo razonable. |
| 2. Acercar el sistema al mundo real | El sistema debe hablar el idioma de los usuarios, con palabras, frases y conceptos familiares para el usuario, en lugar de términos orientados al sistema. Siga las convenciones del mundo real, haciendo que la información aparezca en un orden natural y lógico. |
| 3. Control del usuario y libertad de decisión | Los usuarios a menudo eligen las funciones del sistema por error y necesitarán una "salida de emergencia" claramente marcada para salir del estado no deseado sin tener que pasar por un diálogo extendido. Soporte para deshacer y rehacer. |
| 4. Consistencia y estándares | Los usuarios no deberían tener que preguntarse si diferentes palabras, situaciones o acciones significan lo mismo. Siga las convenciones de la plataforma. |
| 5. Prevención de errores | Es preferible un buen diseño, a tener buenos mensajes de error. Un buen diseño evita que ocurran problemas con el uso del sistema. |
| 6. Reconocimiento en lugar de recordar | Hacer objetos, acciones y opciones visibles. El usuario no debería tener que recordar información de una parte de la interacción a otra. Las instrucciones de uso del sistema deben ser visibles o fácilmente recuperables siempre que sea apropiado. |
| 7. Flexibilidad y eficiencia de uso | Los atajos (no conocidos por el usuario novato) a menudo pueden acelerar la interacción para el usuario experto, de modo que el sistema puede atender tanto a usuarios inexpertos como experimentados. Permitir a los usuarios personalizar las acciones frecuentes. |
| 8. Diseño estético y minimalista. | Los diálogos no deben contener información irrelevante o raramente necesaria. Cada unidad adicional de información en un diálogo compite con las unidades de información relevantes, disminuye su visibilidad relativa, y genera carga cognitiva en el usuario. |
| 9. Ayudar a reconocer, diagnosticar y recuperarse de errores | Los mensajes de error deben expresarse en lenguaje simple (sin códigos), indicar con precisión el problema y sugerir constructivamente una solución. |
| 10. Ayuda y documentación | Es mejor si el sistema se puede usar sin manual,. Sin embargo, puede ser necesario proporcionar ayuda y documentación. Cualquier información de este tipo debería ser fácil de buscar, centrada en la tarea del usuario, enumerar los pasos concretos que se deben realizar. El contenido no debe ser demasiado largo. |

Nota. Adaptado de Nielsen (1993).

Las métricas que se utilizan para evaluar la usabilidad en un producto pueden no servir para otro.

El contexto de uso también es fuente de consideración. Una aplicación en un móvil puede ser poco usable, pero su versión en un portátil sí lo sea. Con esto debe quedar claro que deben comprenderse las cualidades de las plataformas de despliegue como aspecto de diseño en la búsqueda de mejorar la usabilidad.

El estándar ISO/IEC 25022 proporciona **medidas relacionadas con el contexto de uso**, definiéndola como "el grado en que un producto o sistema se puede utilizar con eficacia, eficiencia, satisfacción y ausencia de riesgos tanto en contextos específicos de uso como en **contextos más allá de aquellos**

"**inicialmente explícitamente identificados**" (Bevan et al, 2016, p.273). En este caso se hace referencia a la flexibilidad del sistema interactivo respecto a su capacidad de anticipar otros modos de uso de la aplicación, que algunos llaman cobertura de contextos de uso.

Para medir la cobertura de contextos de uso, y basados en estándares ISO/EIC, se utilizan las siguientes métricas (Bevan et al, 2016):

- a. **Contextos de uso flexibles:** flexibilidad respecto a los contextos inicialmente no identificados durante el diseño.
- b. **Flexibilidad del producto:** la flexibilidad permite que los productos tengan en cuenta las circunstancias, oportunidades y preferencias individuales que podrían no haberse anticipado de antemano.
- c. **Independencia de las competencias del usuario:** evalúa la medida en que el producto puede ser utilizado por personas que no tienen conocimientos, habilidades o experiencia específicos. El producto podría estar destinado principalmente a un grupo de usuarios con conocimientos, habilidades o experiencia específicos, pero potencialmente utilizables por tipología más amplia de usuarios.

Algunas métricas utilizadas para evaluar la usabilidad se basan en el estándar ISO/EIC 25022, y se resumen en la Tabla 22.

Tabla 22

Algunas métricas utilizadas para evaluar aspectos de usabilidad de un sistema interactivo del estándar ISO/EIC 25022

| Aspectos y métricas | | |
|--|---|--|
| Eficacia | Eficiencia | Satisfacción |
| <ul style="list-style-type: none"> • Tareas completadas • Objetivos alcanzados • Errores en una tarea • Tareas con errores • Intensidad de error de tarea | <ul style="list-style-type: none"> • Tiempo de la tarea • Eficiencia de tiempo • Rentabilidad • Relación de tiempo productivo • Acciones innecesarias • Fatiga | <ul style="list-style-type: none"> • Satisfacción general • Satisfacción con características • Uso discrecional • Utilización de funciones • Proporción de usuarios no satisfechos • Proporción de quejas de usuarios sobre una característica particular • Confianza del usuario • Placer del usuario • Comodidad física |
| Aprendizaje | Operabilidad | Protección de error del usuario |
| <ul style="list-style-type: none"> • Completitud de la guía del usuario • Campos predeterminados de entrada • Mensaje de error • Comprensibilidad • Interfaz de usuario auto explicativa, intuitiva | <ul style="list-style-type: none"> • Consistencia operacional • Claridad en los mensajes • Personalización funcional • Interfaz de usuario personalizable/adaptable • Capacidad de monitoreo • Tolerancia (deshacer) • Categorización comprensible de información • Apariencia consistente • Soporte de dispositivo de entrada | <ul style="list-style-type: none"> • Evitar el error de operación del usuario • Corrección de error de entrada de usuario • Brindar al usuario la posibilidad de recuperarse de error |

Nota: Elaboración propia a partir de Bevan et al (2016, p.374).

Las empresas de desarrollo hacen uso de listas de verificación para llevar a cabo la evaluación de usabilidad mediante heurísticas. Una lista de verificación para la evaluación heurística de usabilidad está disponible en (Pierotti, 2014). El documento es muy completo y considera una serie importante de características y aspectos a tomar en cuenta. Respecto a una posible lista de verificación para aplicaciones en plataformas móviles se puede consultar (Yáñez, Cascado y Sevillano, 2014).

En el **Anexo F** se listan algunos recursos en línea para la evaluación de usabilidad y enlaces a listas de verificación.

7.4. Accesibilidad

La palabra accesibilidad, según la RAE, tiene una definición general que hace referencia a “que algo es accesible”. La definición en el contexto del uso de un producto es “la condición que deben cumplir los entornos, productos y servicios para que sean comprensibles, utilizables y practicables por todos los ciudadanos, incluidas las personas con discapacidad” (RAE, 2019).

A menudo se confunde el concepto de accesibilidad con la participación de personas con discapacidad. Sin embargo, muchos usuarios pueden ser “discapacitados” en distintos contextos y circunstancias. La accesibilidad se trata de personas.

Diseñar considerando la accesibilidad requiere de cierta previsión, desde el momento que se inicia la planificación para el diseño centrado en el usuario, se crean las “personas”, y con estas se consideran sus posibles problemas físicos o cognitivos.

Los aspectos del usuario que se deben considerar para un diseño accesible son:

- **Visual:** la hipermetropía, la ceguera, el daltonismo, son formas de discapacidad visual que se deben tomar en cuenta durante el diseño. Existe un abuso generalizado en el diseño gráfico especialmente con el uso de colores como único “elemento de información”. No siempre el usuario puede reconocer los colores. Una recomendación es seleccionar canales alternos (voz, sonido, audio) para la “presentación” de la información en el caso que se activen tareas, funcionalidades, seleccionan objetos, etc.
- **Motor / Movilidad:** esta categoría no solo se extiende a problemas con el uso de las manos y los brazos, sino también con otras afecciones musculares o esqueléticas. En este caso se recomienda revisar los estándares de factores humanos y ergonomía de las organizaciones internacionales.
- **Auditiva:** las discapacidades auditivas se presentan en diversos grados de severidad, hasta la sordera total. En el diseño de una interfaz no debe utilizarse un elemento auditivo (sonido, audio) para brindar una retroalimentación ante una tarea, debe conjugarse con otro canal (visual). Generalmente en las interfaces que son personalizables, los usuarios pueden decidir cuál canal utilizar para la interacción con la aplicación.
- **Estados de agitación (convulsiones):** algunas personas pueden verse afectadas por la luz, el movimiento, el parpadeo, etc., en la pantalla, lo que desencadena convulsiones. El problema

más común en esta categoría es la epilepsia fotosensible. Muchos diseños de interfaces se basan únicamente en los aspectos estéticos y abuso de animaciones, haciendo que el uso del color y el dinamismo de las imágenes originen problemas de accesibilidad.

- **Aspectos cognitivos, aprendizaje:** no todas las discapacidades son físicas. Existen discapacidades cognitivas y de aprendizaje que también pueden influir en la accesibilidad.

Respecto a las técnicas para evaluar la accesibilidad, la W3C en su iniciativa **WAI** (*Web Accessibility Initiative*) describe una serie de estándares y herramientas para evaluar la accesibilidad en ambiente Web. En el portal de **Nielsen Norman Group** están disponibles las guías de usabilidad para realizar un diseño de aplicaciones web accesibles.

 **WAI**
<https://www.w3.org/WAI/test-evaluate/>

 **Nielsen Norman Group**
<https://www.nngroup.com/reports/usability-guidelines-accessible-web-design/>

La importancia del concepto de accesibilidad ha hecho que el Parlamento Europeo haya tomado cartas en el asunto, de forma que en abril 2019 han promulgado la **Directiva (UE) 2019/882 del Parlamento Europeo y del Consejo de la Unión Europea** sobre los requisitos de accesibilidad de los productos y servicios.

Los detalles y contenido de la Directiva están disponibles en este enlace:

 **DIRECTIVA (UE) 2019/882**
<https://eur-lex.europa.eu/legal-content/ES/TXT/PDF/?uri=CELEX:32019L0882&from=ES>

Un resumen de la aplicación de esta directiva puede observarse en la Figura 36.



Figura 36. Resumen de la aplicación de la Directiva 2019/882 sobre los requisitos de accesibilidad de los productos y servicios. Recuperado de https://www.usableyaccesible.com/images/directiva_2019_882_accesibilidad.svg

7.5. Algunas consideraciones sobre los conceptos de usabilidad y accesibilidad

Existe **gran confusión entre los conceptos de usabilidad y accesibilidad**. Algunos autores consideran la accesibilidad como parte de la usabilidad. Una forma de comenzar a observar la distinción entre las dos características es mirando de cerca los problemas que atañen a los usuarios (Henry, 2008):

- Los **problemas de usabilidad** afectan a todos los usuarios por igual, independientemente de su capacidad; es decir, **una persona con discapacidad no está en desventaja en mayor medida por cuestiones de usabilidad que una persona sin discapacidad**.
- Los **problemas de accesibilidad** disminuyen el acceso de las personas con discapacidad a un producto. Es decir, **cuando una persona con discapacidad está en desventaja en relación con una persona sin discapacidad, es un problema de accesibilidad**.

“Cuando se consideran las discapacidades cognitivas y de lenguaje, la distinción entre usabilidad y accesibilidad es especialmente difícil de definir. Muchas de las pautas para mejorar la accesibilidad para personas con discapacidades cognitivas son las mismas que las pautas generales de usabilidad.” (Henry, 2008).

En el portal uiaccess.com se presenta un extenso material que se dedica a este tema, donde se hace referencia al libro (también en español) de (Henry, 2008a):



Uiaccess.com

<http://www.uiaccess.com/>



En español

<http://www.uiaccess.com/JustAsk/es/>

7.6. Evaluación de las soluciones para el sistema interactivo *MyEnchantedMuseum – Interactive Complex*

En este apartado se discuten las actividades de evaluación de las soluciones al sistema interactivo *MyEnchantedMuseum*. Una vez creados los modelos (prototipos de diversa fidelidad y posibilidades de ser operables) para cada solución, el equipo de desarrollo revisó si fueron considerados todos los requisitos del sistema, si cumplían con los principios de diseño y calidad como la usabilidad y la accesibilidad. De esta manera podrán **comunicar** y discutir los resultados con las partes interesadas.

Algunos preliminares respecto al proceso de evaluación de ambas soluciones fueron:

- a. La empresa de desarrollo trabaja con una **lista de verificación** alineada a principios de diseño y a estándares de calidad.
- b. El **método** utilizado para la **evaluación de usabilidad** fue **inspección**, y las **técnicas** utilizadas fueron **evaluación heurística** e **inspección de características**. El informe resultante de la

evaluación heurística no solo incluye las incidencias, sino que el especialista hace las recomendaciones para subsanar los problemas evidenciados. En lo que se refiere a la inspección de características, revisaron las necesidades y requisitos de los usuarios.

Respecto a la **accesibilidad** se utilizaron las recomendaciones de la **Directiva Europea 2019/882**.

Algunas de las conclusiones producidas durante la actividad de evaluación, de cada una de las soluciones, fueron las siguientes:

Caso Solución A

1. Respecto al desarrollo del dispositivo para guiar el usuario dentro del museo (utilizar la tecnología de realidad aumentada basada en reconocimiento de objetos y sistema de geolocalización), los diseñadores consideraron que esta solución no era recomendable y factible por las siguientes razones:
 - a. Los requisitos funcionales fueron respetados. Sin embargo, no se atendieron las restricciones respecto a los tiempos y recursos para implantar este dispositivo. Las estimaciones de los diseñadores, tanto en tiempos y disponibilidad económica, no hacían factible esta solución.
 - b. La tecnología para el reconocimiento de objetos debe actuar sobre obras relativamente parecidas, aunque distintas (otra restricción). Para refinar y mejorar el proceso de reconocimiento, se necesita tecnología de geolocalización muy precisa (esta tecnología no discrimina en espacios reducidos como el de cada sala en el museo). Además, los usuarios deben utilizar una tecnología de adquisición de alta calidad. Los *smartphones* y tabletas deben ser de la mejor gama del mercado, con cámaras de al menos 48 Mega píxeles, y memoria de 32 GB. Estas características se encuentran en dispositivos muy costosos, por lo que muchos usuarios estarían limitados en la visita al museo, si quieren utilizar este componente del sistema.
 - c. Respecto a los aspectos de usabilidad y accesibilidad (requisitos no funcionales) los diseñadores modelaron la interacción haciendo uso de las **personas** identificadas [modelos de usuarios]. Los dos casos particulares que llamó la atención fueron los usuarios (K-12) y los usuarios (de cualquier edad) con discapacidad motora (piernas, brazos o ambos). No todas estas clases de usuarios, si bien acompañados, podían "seleccionar" la obra de interés. De manera que esta tecnología de realidad aumentada no les brindará la posibilidad de mejorar la interacción y la experiencia del usuario.
2. Respecto al folleto interactivo, la evaluación arrojó como problema fundamental que el cambio (sutil) de tecnología, aunque con la misma primitiva de interacción, no era consistente con el uso del dispositivo para guiar el usuario. Si bien el paradigma es el mismo (realidad aumentada), el usuario podía crear un modelo cognitivo para la utilización del folleto, para luego cambiar su modelo cognitivo en la utilización del dispositivo guía. En otras palabras, los "objetos" de interés eran distintos: en un caso era una obra, en otro caso era un "cuadrado" con unos puntos.... el código QR.

3. Respecto a los otros componentes del sistema interactivo, las restricciones (recursos y tiempos) no aparecieron aspectos difíciles de resolver. En el caso de la usabilidad y accesibilidad, la evaluación arrojó problemas que pueden ser resueltos en forma expedita, aplicando los principios de diseño.

Caso Solución B

Es oportuno recordar que la única diferencia entre las Soluciones A y B es el dispositivo para guiar al usuario en el museo. Algunas conclusiones obtenidas de la actividad de evaluación fueron las siguientes:

- a. Los requisitos funcionales, no funcionales y las restricciones fueron respetados para producir el diseño y un desarrollo factible.
- b. En el caso específico de la “captura” de la obra de interés, el sistema de reconocimiento basado en QR dinámico (que permite actualizar la información incluida en el código cada vez que sea necesario), los usuarios requieren de un lector que pueden instalar en sus *smartphones* y tabletas, al ingresar al museo, o directamente del sitio web de la organización.
- c. Después de la evaluación de usabilidad y accesibilidad del dispositivo, los diseñadores no encontraron aspectos que influyen en la interacción del usuario con los servicios brindados, por lo que consideraron que la Solución B era factible en todos los sentidos. Además, el folleto interactivo se usará igual que el dispositivo para la “captura” de las obras de interés (a través del código QR), por lo tanto, se respeta el principio de consistencia.

Reflexión general sobre el proceso de evaluación y sus protagonistas

Cuando se lleva a cabo la evaluación de la interacción e interfaz, así como los aspectos de usabilidad y accesibilidad, se debe recordar lo siguiente:

- Es necesario mantener actualizada la lista de verificación respecto a las nuevas tecnologías.
- Es necesario que el evaluador (especialista en interacción e interfaces) esté actualizado respecto a la usabilidad y accesibilidad que brindan las nuevas tecnologías.
- Reconocer la necesidad de la formación para entender los estándares y basarse principalmente en las características de los usuarios.
- No considerar únicamente la tecnología de punta disponible, sino pensar en un compromiso entre lo que brinda la industria y lo que realmente necesita el usuario.

Actividad recomendada

Diseñe una lista de verificación para evaluar la interacción e interfaz, por Usted diseñada en las actividades anteriores, del nuevo sitio web del museo.

Tema 8. Herramientas de soporte al proceso de diseño y desarrollo de interacción e interfaces

"Las mejores herramientas para resolver un problema son lápiz, papel y neuronas"

(Adelaide Bianchini – Consultora de UX Design y Profesor VIU)

Las herramientas que existen en el mercado se pueden clasificar según las necesidades del diseñador. En este sentido hay una serie de productos dedicada a la creación de prototipos (ya sea de baja o alta fidelidad), otras se utilizan para el diseño exclusivo de wireframes, y otras soluciones son para "traspasar" el diseño de un prototipo funcional (ejecutable) de una plataforma a otra.

En la Tabla 23 se resumen algunas de las herramientas para el diseño de prototipos más utilizadas en el mercado.

Tabla 23

Resumen de características de algunas herramientas para la elaboración de prototipos.

| Herramienta | Características principales | Plataformas de despliegue |
|-------------------------|---|---|
| JustinMind (Prototyper) | <ul style="list-style-type: none"> Extremadamente rápido. Para el diseño y construcción de prototipos de alta fidelidad. Edición fácil arrastrando archivos nuevos sobre archivos existentes. Ofrece la posibilidad de publicar el diseño en el web para su revisión por otros usuarios. Ofrece una versión gratuita plenamente funcional con algunas limitaciones. | Web y aplicaciones para móviles |
| InVision | <ul style="list-style-type: none"> Fácil de aprender. Rápido: se configura la cuenta y crea el primer prototipo en pocos minutos. Rápido e intuitivo para agregar pantallas y crear zonas interactivas (arrastrar + soltar). Sistema de intercambio y comentarios para recopilar comentarios. Funciones de administración de archivos a través de una herramienta web o una carpeta similar a Dropbox para compartirlos y editarlos fácilmente. Visualización web simple del prototipo. | Web, escritorio y aplicaciones para móviles |
| Proto.io | <ul style="list-style-type: none"> Opción para agregar animación a elementos individuales y efectos de transición a enlaces. Buena capacitación y documentación de soporte. Control de grano fino para agregar interactividad a elementos individuales. Simulación efectiva de comportamientos de interacción de alta fidelidad. Soporte para la interacción basada en gestos. | Aplicaciones para móviles |
| Axure | <ul style="list-style-type: none"> Buena documentación de soporte. Controles de grano fino para agregar interactividad a elementos individuales. Bueno para crear prototipos de comportamientos de interacción complejos. Incluye biblioteca incorporada de <i>widgets</i> que se pueden personalizar con acciones y comportamientos específicos. Flexible: se puede utilizar para crear prototipos de productos para cualquier plataforma digital. | Web, escritorio y aplicaciones para móviles |
| Phoshare | <ul style="list-style-type: none"> Controles de grano fino para interactividad y apariencia de elementos. Opción para simular una sola página o todo el prototipo. Las características de colaboración permiten que varias personas editen / revisen el diseño. Buena documentación de soporte. Rápido e intuitivo para agregar archivos mediante acciones "<i>drag and drop</i>". Opción para agregar animación a elementos individuales y transiciones de pantalla. | Web, escritorio y aplicaciones para móviles |

Nota: Elaboración propia.

En la Tabla 24 se resumen algunas características de las herramientas para el diseño de wireframes.

Tabla 24

Resumen de características de algunas herramientas para el diseño de wireframes

| Herramienta | Características principales | Plataformas de despliegue |
|-------------|--|---|
| Balsamiq | <ul style="list-style-type: none"> Facilidad de uso. Permite realizar los wireframes directamente en el navegador. Tiene asociada una librería de objetos para Android y otros sistemas operativos (a través del portal Wireframes To Go) | Web y aplicaciones para móviles |
| Moqups | <ul style="list-style-type: none"> Herramienta en línea. Permite desarrollar prototipos en tiempos cortos. Tiene una interfaz intuitiva con más de 300 objetos. Tiene disponible una versión gratuita que permite tener dos proyectos activos, aunque con funcionalidades limitadas. | Web, aplicaciones móviles |
| Phoshare | <ul style="list-style-type: none"> Controles de grano fino para interactividad y apariencia de elementos. Opción para simular una sola página o todo el prototipo. Las características de colaboración permiten que varias personas editen / revisen el diseño. Buena documentación de soporte. Rápido e intuitivo para agregar archivos mediante acciones "drag and drop". Opción para agregar animación a elementos individuales y transiciones de pantalla. | Web, escritorio y aplicaciones para móviles |
| MockFlow | <ul style="list-style-type: none"> Está disponible como herramienta web. Permite construir el esqueleto de una página o de una aplicación. Incluye la funcionalidad de añadir enlaces para evaluar la navegación. La versión gratuita permite un único proyecto, sin límite de instancias de páginas/ventanas. | Web, aplicaciones móviles |

Nota: Elaboración propia.

Una manera de seleccionar una herramienta, vista la cantidad existente en el mercado, es utilizando los siguientes criterios (Cooper.com, 2013):

- **Tiempo para crear el prototipo / Facilidad de uso – capacidad de aprendizaje:** cuánto tiempo se necesita para crear el prototipo una vez que la herramienta estuvo en funcionamiento.
- **Fidelidad:** qué tan bien puede simular la interactividad prevista del prototipo con la herramienta (desplazamiento, transiciones, flujo entre ventanas/páginas y estados, visualización general y apariencia).
- **Colaboración / intercambio:** calidad de las características para compartir el prototipo con otros y/o trabajar en forma colaborativa en el prototipo.
- **Pruebas de usabilidad:** calidad de las funciones para realizar pruebas de usabilidad con el prototipo.
- **Soporte:** cantidad y calidad de tutoriales, documentación de ayuda, bibliotecas (elementos de la interfaz de usuario, *widgets*, iconos), wireframes listos para ser utilizados, etc.
- **Interacciones / gestos:** calidad de las características para agregar interactividad específica de interacción orientada a gestos al prototipo.

- **Animaciones:** calidad de las características para agregar comportamientos animados a las transiciones de pantalla y elementos individuales dentro de una pantalla.
- **Prueba de dispositivos:** calidad de las funciones para probar el prototipo en otros dispositivos.

En el portal **uxtools.co** (<https://uxtools.co/tools/prototyping>) se presenta una evaluación de herramientas de prototipado, la cual considera una lista muy completa de criterios para hacer una selección.

 **Flinto**

<https://www.flinto.com/>

 **InVision**

<https://www.invisionapp.com/>

 **Proto.io**

<https://proto.io/>

 **Axure**

<https://www.axure.com/>

 **Protoshare**

<http://www.protoshare.com/>

 **Solidify**

<https://solidifyapp.com/>

 **Easel**

<https://www.appvita.com/2014/02/07/easel-build-prototypes-designs/>

 **Justinmind**

<https://www.justinmind.com/>

 **Fluid**

<https://www.fluidui.com/>

En la Tabla 25 se muestra una comparación de algunas herramientas (medium.com,2016), entre las cuales están Azure, Briefs, Flinto, Fluid e InVision.

Tabla 25
Evaluación de herramientas para prototipado

| Tool | Time/ Speed | Fidelity | Sharing | User Testing | Support | Mobile & Touch | Dinamic Elements |
|--|----------------|----------|---------|-----------------|---------|-------------------|---------------------|
|  Axure Robust prototyping tool for websites and apps Last updated: Feb 24 | 40-80 mins | Average | Average | Average | High | Average | Good |
|  Briefs Live prototyping tool for iOS apps Last updated: Feb 24 | >80 mins | Good | Low | Average | Low | None | Average |
|  Flinto Simple prototyping tool for iOS apps Last updated: Feb 24 | 5-10 mins | Good | Good | Average | Low | Low | Average |
|  Fluid Prototyping tool for mobile apps Last updated: Feb 24 | 40-80 mins | Good | Good | Average | Average | Good | Good |
|  Form Native prototyping tool for iOS apps Last updated: Feb 24 | >80 mins | High | Low | Average | Good | High | High |
|  Framer Prototyping tool built on framer.js for prototyping mobile and desktop apps Last updated: Jul 14 | >80 mins | High | Average | Average | High | High | High |
|  HotGloo Prototyping tool for interactive and responsive wireframes Last updated: Feb 24 | 40-80 mins | Low | Average | Low | Average | None | Average |
|  Indigo Studio Robust prototyping tool for web, desktop and mobile apps Last updated: Feb 24 | 40-80 mins | Good | Average | Average | High | High | High |
|  InVision Collaborative click-through prototyping tool for web and mobile Last updated: Mar 2 | 5-10 mins | Good | High | Good | Good | High | Average |

Nota: Recuperado de <https://medium.com/@uxhow/ux-design-resource-guide-926676be3e0>

Es importante recalcar que, **si bien las herramientas pueden acelerar la creación y ayudan a verificar el comportamiento de los objetos en una interfaz, si el diseño no fue elaborado siguiendo los principios, guías y estándares, no se puede asegurar un buen resultado.**

Los diseñadores tienen la posibilidad de utilizar **patrones de diseño de interfaces**, que son objetos (de interfaz) probados y para cada uno de ellos existe una descripción detallada sobre aquellos casos donde pueden ser útiles, cuándo no utilizarlos y de esta manera se asegura un diseño adecuado.

Una de las referencias más importante sobre los patrones de diseño de interacción e interfaz, su descripción, cuándo y por qué pueden utilizarse, y otras propiedades está disponible en Tidwell (2010).

Conclusiones

Este libro tiene como finalidad brindar una herramienta de consulta para los estudiantes de la asignatura *Desarrollo de Sistemas Interactivos*.

El mundo en el que vivimos está lleno de dispositivos y sistemas interactivos, cada uno de ellos con características y objetivos distintos. Sin embargo, todos comparten una peculiaridad: existen distintos usuarios que los están utilizando. Conocer al usuario, entender sus necesidades, conocer el alcance y objetivo del sistema es la base fundamental del proceso de desarrollo.

El contenido se organizó partiendo del concepto de interacción, sus tipos y sus primitivas. Luego se dedica al ámbito ingenieril del diseño centrado en el usuario (DCU), y se trataron los aspectos metodológicos, modelos de proceso y los productos creados a partir de las actividades. Entender las necesidades y requisitos de usuarios, así como los contextos de uso y el dominio, del futuro sistema, es parte fundamental dentro del proceso del DCU.

Una de las actividades fundamentales del proceso de diseño es la creación de modelos, de forma que el equipo de desarrollo pueda comunicar con los interesados las soluciones a su problema. De ahí la importancia de tratar asuntos sobre prototipos y técnicas para su creación.

Una vez diseñado el prototipo de la solución, se refina para lograr un producto final. Éste debe mostrar aquellos componentes que den “pistas” al usuario de cómo usarlo de manera que sea productivo, eficaz, eficiente. El producto final debe incluir aquella capa con la cual se “comunica” el usuario con el sistema: la interfaz de usuario. Sobre este tema se trataron los estilos y objetos de interfaz, así como los principios y guías de diseño.

Otro tema de interés se refiere a la **calidad de un sistema interactivo**: un producto debe ser útil, usable y accesible. Una manera de verificar si la solución obtenida cumple con ciertos niveles de calidad es la evaluación de usabilidad y de accesibilidad.

Respecto a herramientas, todos los seres humanos las necesitan. En el mundo de la ingeniería de software se ha incrementado notablemente el uso de instrumentos de soporte a diseño y desarrollo de productos interactivos. Sobre este argumento se proponen una serie de productos existentes en el mercado, y cómo seleccionarlas respecto a las ayudas que proponen. Si bien las herramientas pueden automatizar partes del proceso de desarrollo, ellas no siempre son suficientes puesto que hay una serie de reflexiones subjetivas que debe tomar en cuenta el diseñador, y no siempre son automatizables.

En el contenido se incluye un problema hipotético (*MyEnchantedMuseum – Interactive Complex*) que se va analizando a medida que se avanza en los distintos temas.

A este contenido se le agrega una serie de Anexos, para facilitar la lectura, y se van recomendando a medida que se avanza en los argumentos atinentes.

Se utilizaron referencias de autores de prestigio en el área de diseño de interacción e interfaces, así como recursos en línea sobre el uso de estándares de desarrollo. Todo esto con el fin de presentar un contenido transversal sobre las buenas prácticas y recomendaciones para cada una de las actividades del proceso de diseño, aspectos de evaluación de usabilidad y accesibilidad, y acuerdos promovidos por organizaciones internacionales de estandarización.

El autor no pretende que este sea un libro que cubra de manera completa y formal todas las incidencias y asuntos del diseño de sistemas interactivos, sino que sea una referencia general y sirva de base para profundizar en el área. La experiencia ha demostrado que la tecnología, día a día, evoluciona muy rápido y siempre habrá algo nuevo que aprender. El libro se enriquece con los recursos bibliográficos atinentes, y se recomiendan lecturas que abordan en profundidad alguno aspectos específicos.

Anexos

Anexo A. Primitivas de interacción

Cuando el usuario realiza una interacción con el computador, generalmente “introduce” un elemento de información, y lo hace en distintas maneras y modos, utilizando distintos objetos como mediadores. Ese elemento de información tiene un objetivo y un significado en el contexto de la aplicación y en la tarea específica, así como los objetos tienen un objetivo y comportamiento específicos.

En este contexto, en el diseño de la interacción humano computador se consideran las siguientes **primitivas** o tareas de interacción, asociadas a las dimensiones tratadas en el tema 2, que utilizadas en forma adecuada se alinean a los objetivos de usabilidad y dirigen el diseño de una buena interfaz. Las primitivas son las siguientes:

a. Posicionamiento

Es la primitiva que se utiliza para **modelar la situación en la cual un usuario debe posicionarse sobre un espacio (2D o 3D) para indicar una posición u orientación**, obtener o introducir un valor (que representa una coordenada). Se puede usar para aplicaciones orientadas a dibujo y gráficas, posicionarse sobre una ventana para luego activarla, posicionarse sobre un mapa digital, etc.

La acción que realiza el usuario es la de mover un cursor en la pantalla para introducir o identificar un “lugar”, o introducir la coordenada directamente. Con la aparición del ratón (Figura 1A (a)), la tarea se lleva a cabo de forma que el usuario es capaz de hacer un *mapping* entre el espacio de movimiento del ratón en su escritorio real y su representación en el espacio virtual de acción (pantalla). Con las nuevas tecnologías, la tarea de posicionamiento puede hacerse directamente sobre las pantallas de los dispositivos utilizando los dedos como puntero (Figura 1A (b)).

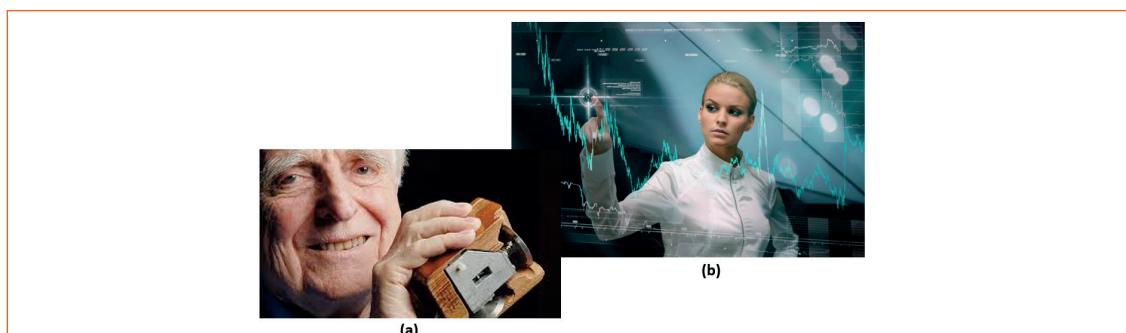


Figura 1A. (a) La primera versión del ratón y su inventor Douglas Engelbart. (b) Un usuario se posiciona sobre un dispositivo, haciendo uso de sus manos, para obtener una información. Elaboración propia. Imagen (a) recuperada del portal <https://www.iprofesional.com/tecnologia/164410-perifericos-A-los-88-anos-murió-Douglas-Engelbart-el-inventor-del-mouse-de-las-computadoras>. Imagen (b) recuperada de <https://www.pixabay.com>

Algunos de los aspectos a tener en cuenta cuando se decide utilizar esta primitiva de interacción son (Gea y Gutiérrez, 2002):

- **Definición del sistema de coordenadas:** el movimiento del objeto puede ser en función de un sistema de coordenadas del propio objeto, de la pantalla o del mundo. Este factor es muy importante en los sistemas de modelado de posicionamiento 3D (Realidad Virtual). Por lo tanto, el usuario debe tener un punto de referencia para situarse en esos ambientes.
- **Resolución / Velocidad:** en caso de movimiento con un dispositivo (cursor, ratón, dedo índice) las posiciones pueden ser discretas o continuas. Debe tenerse en cuenta la relación Control / Display (C / D) para mecanismos de posicionamiento indirecto, es decir, la distancia que hay que mover el puntero (ratón) en el escritorio para obtener una nueva posición diferente en pantalla. Deben proporcionarse elementos para controlar la resolución y la velocidad de desplazamiento del cursor utilizado.
- **Restricciones de interacción:** se pueden utilizar elementos que ayudan a orientar el usuario en esta tarea, como es el caso de una rejilla (*grid*) que facilita el posicionamiento en los puntos ajustados a valores. La interacción debe prever la posibilidad de acciones (alertas) si el desplazamiento se lleva a cabo “fuera” de la rejilla.
- **Retroalimentación:** debe considerarse el efecto de la tarea a través de un cambio de estado. Puede ser espacial (visualización gráfica del cursor en pantalla con cambios de color, forma o textura), relacionada con otros elementos (cambio de cursores), o textual (al representar un valor numérico en coordenadas cartesianas).

En la Figura 2A se muestran algunos de los aspectos descritos anteriormente.

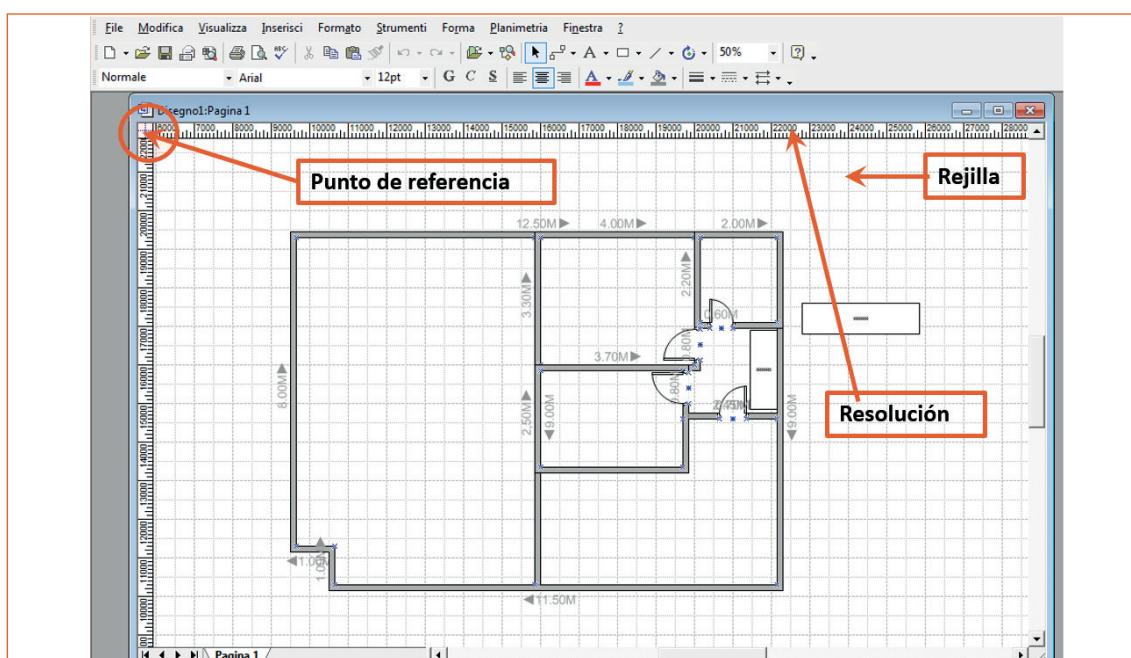


Figura 2A. Elementos de un programa de diseño arquitectónico (MSVisio). Rejilla, punto de referencia y visualización de la resolución. Elaboración propia.

b. Selección

Se refiere a la tarea para la obtención / selección de una alternativa (entre varias posibles). Se puede distinguir entre la selección sobre un conjunto finito o bien de un conjunto variable de elementos (dependiendo de la tarea pueden configurarse de manera distinta). En cualquier caso, debe considerarse que las alternativas pueden ser disjuntas o no.

Respecto a los conjuntos, sus elementos pueden ser **textos** (nombres o etiquetas de una acción) u **objetos** (elementos en el contexto de la aplicación).

La acción de selección se puede realizar de los siguientes modos:

- **Mediante un identificador**, en cuyo caso el usuario introduce el nombre (identificador del objeto que desea seleccionar).
- **Apuntando**, en cuyo caso se barre espacialmente una lista de posibles objetos a seleccionar, haciendo uso de un puntero (ratón, o el dedo índice si la tecnología lo permite).

Selección según el tipo de conjuntos

- **Objetos en conjuntos de tamaño variable**

La selección de objetos, en un **conjunto de tamaño variable**, debe facilitarse mediante la presentación en algún contenedor: listas dinámicas (actualizadas según un contexto) o ventanas en las cuales se disponen los objetos ya sea en manera textual o gráfica (por ejemplo: íconos). En principio la presentación/organización de los objetos depende de distintos criterios definidos en cada una de las tareas de la aplicación y que, en algunos casos, el usuario pudiese configurar (si así se prevé en el diseño de las funcionalidades).

- **Objetos en conjuntos de tamaño fijo**

En el caso de selección de objetos en un **conjunto de tamaño fijo** se utilizan otros tipos de mecanismos de presentación para mejorar la interacción. Algunos de los mecanismos más utilizados son:

- **Barra de menú**, la cual se visualiza en forma horizontal en la parte superior de una ventana. Contiene una serie fija de elementos (llamados menús).
- **Menú** es una lista de ítems, identificado con una etiqueta específica (nombre), dentro de la barra de menú. Un menú tiene como objetivo mostrar en forma “agrupada” o contextualizada las acciones habilitadas o no en un contexto dado y para su posible selección. Cada acción es un ítem dentro del menú. Si la acción prevé subacciones, estas deben estructurarse como jerarquías, casos especiales, etc.
- **Barra de herramientas**, en la cual se muestran conjuntos de íconos (representan acciones) o etiquetas textuales, los cuales pueden estar habilitados o no, al igual que sucede con los ítems en menús.

- **Lista desplegable**, es un mecanismo que permite al usuario seleccionar uno o más elementos de una lista que se encuentra dentro de un contenedor de varias líneas. Pueden ser de distintos tipos dependiendo del contexto de uso. Se describen en el tema sobre diseño de interfaces.
- **Casos particulares. Selección / Deselección de valores disjuntos o no**

Existe la posibilidad que el usuario deba interactuar sobre objetos que tienen solo dos posibles estados (seleccionado o no seleccionado). Se presentan generalmente dos casos:

1. La **lista presenta objetos** (opciones) **mutualmente excluyentes**, es decir el usuario debe seleccionar una única opción. En caso de que hubiese una anteriormente “seleccionada”, ésta debe cambiar al estado “no seleccionado” en forma “automática”. Esta posibilidad de interacción hace referencia a las llamadas casillas de verificación (esquema de selección de respuesta en examen, con distintas opciones de selección, pero con una única opción válida). Históricamente este tipo de mecanismo también se asocia a aquellos pulsantes presentes en una grabadora de sonido, donde solo uno podía estar activo.
2. La **lista de objetos** (opciones) **no son mutualmente excluyentes**, es decir el usuario puede seleccionar más de un objeto de esa lista. En este caso el mecanismo no deselecciona los objetos ya seleccionados, y es el usuario que debe llevar a cabo la tarea deseleccionar. Generalmente se utilizan elementos gráficos distintos a los de selección de opciones mutualmente excluyentes.

Cuando se diseña la interacción sobre estos objetos, y cualquiera sea el caso es necesario considerar la ubicación espacial, tamaño (visualización en distintos dispositivos), etiqueta (del objeto) y la semántica asociada para no confundir al usuario. De esta manera es natural reconocer si la selección es mutualmente excluyente o no.

c. Introducir texto

Es una primitiva de interacción en la que el usuario debe tener la facilidad para introducir un texto, ya sea utilizando un teclado (físico) o se le proporciona un mecanismo similar. Para que la interacción sea sencilla de llevar a cabo se deben considerar características tales como la resolución de pantalla para mejorar la legibilidad, establecer el tipo y tamaño de letra que no cree confusiones (muchas veces se muestra la letra “O” pero puede ser confundida con el número “0” por la tipografía utilizada), la presentación del área (evidenciar los campos) donde será insertado el texto, sus etiquetas, etc.

d. Introducir valor

Es la tarea en la cual el usuario debe introducir un valor (textual o numérico) de alguna manera cuantificable. Una forma evidente sería el uso de teclado (real o en pantalla), aunque en algunos casos es preferible suministrar al usuario mecanismos gráficos que pueden activarse mediante el ratón, o cursores del teclado. Las consideraciones para diseñar la interacción dependen

mucho de los valores que se desean introducir (valores discretos, continuos, la escala entre un valor y el próximo, etc.). Durante el diseño de la interacción y posteriormente los elementos de interfaz se deben considerar algunos principios para lograr soluciones usables y accesibles.

e. Arrastrar

Es la posibilidad de llevar a cabo un movimiento desde una posición inicial a otra final con la finalidad de “arrastrar” un objeto que se encuentra en algún contenedor (por ejemplo, mover un fichero desde un directorio a otro). El mecanismo más utilizado es el ratón y los dedos (si la tecnología subyacente lo permite). Si el usuario lleva a cabo esta tarea, la aplicación debe dar retroalimentación constante del objeto que se está moviendo o desplazando.

Para implementar cada una de las primitivas de interacción puede aplicarse un conjunto de técnicas que permitan facilitar y flexibilizar el modo de obtención, introducción, selección, etc., de objetos. La facilitación es tema de cómo se presenta esa capa existente entre la aplicación o dispositivo y el usuario, a la cual se le llama interfaz.

Anexo B. Paradigmas de interacción

Los diseñadores de interacción, para luego definir las interfaces, deben responder dos preguntas fundamentales:

- ¿Cómo se puede desarrollar un sistema interactivo para garantizar su usabilidad?
- ¿Cómo se puede demostrar o medir la usabilidad de un sistema interactivo?

Un enfoque para responder a estas preguntas es a través del uso de ejemplos probados (**paradigmas**), en los que los diseñadores creen. **Son utilizados por sistemas interactivos exitosos pues mejoran la usabilidad y, se utilizan para el desarrollo de productos futuros** (Dix, Finlay, Abowd y Beale, 2004).

Respecto a los paradigmas, Dix et al (2004, p.165) hace énfasis en que “Los grandes avances en tecnología informática han aumentado el poder de las máquinas y mejorado el ancho de banda de la comunicación entre humanos y computadores. Sin embargo, el impacto de la tecnología por sí sola no es suficiente para mejorar su usabilidad. A medida que nuestras máquinas se vuelven más potentes, la clave para una mayor usabilidad proviene de la aplicación creativa y considerada de la tecnología para acomodar y aumentar el poder del ser humano. Los paradigmas para la interacción han dependido en su mayor parte de los avances tecnológicos y su aplicación creativa para mejorar la interacción”.

Los paradigmas que se describen a continuación, de alguna manera muestran la evolución y avances tecnológicos en el ámbito del diseño de interacción. No se consideran como tipos o categorías mutualmente excluyentes, sino que cualquiera de ellos puede ser incorporado en los diseños o puede ayudar a crear paradigmas inéditos.

Los paradigmas de interacción que se han estado utilizando son los siguientes:

1. Tiempo compartido

Tiempo compartido significa que una sola computadora podría admitir múltiples usuarios. Aparece a principio de los años 60, gracias al investigador J.C.R. Licklider (scihi.org, 2019). La introducción del tiempo compartido significó el final del procesamiento por lotes, en el que los trabajos completos se comenzaron a procesar de manera individual.

2. Unidades de visualización de video

Las primeras aplicaciones de imágenes de pantalla se desarrollaron en aplicaciones militares. Sin embargo, fue Ivan Sutherland en el año 1962 al desarrollar *Sketchpad* (Sutherland, 1963), que incluía un lenguaje de simulación para modelos visuales y un dispositivo. Demostró que las computadoras podían usarse para crear modelos visuales de abstracciones.

3. Kits de herramientas de programación

El poder de los *kits* de herramientas de programación es que los componentes pequeños y bien entendidos se pueden componer de manera fija para crear herramientas más grandes. Una vez que estas herramientas más grandes se entienden, pueden continuar agregándose con otras herramientas y el proceso continúa.

4. Informática personal

A medida que avanza la tecnología, es cada vez más difícil distinguir entre lo que constituye una computadora personal o estación de trabajo y lo que constituye un *mainframe*. Algunos ejemplos de las primeras aplicaciones de computación personal son el lenguaje LOGO de Papert, Feurzeig y Bobrow, creado en 1966 con el objetivo de enseñar a los niños a utilizar un computador resolviendo problemas con enfoque lógico y "algorítmico". Otro ejemplo innovador fue NLS – *oN-Line Systems* de Douglas Engelbart.

A través de la plataforma NLS, Engelbart y sus colegas desarrollaron muchas de las técnicas de la informática moderna e interactiva. En lugar del procesamiento por lotes asíncrono, el NLS permitió a los usuarios manipular los datos en pantalla directamente y ver los resultados inmediatamente en tiempo real. En lugar de tarjetas perforadas, el NLS presentaba una pantalla similar a un radar con una interfaz gráfica de usuario (GUI, por sus siglas en inglés *Graphic User Interface*), en la que los usuarios manipulaban texto, símbolos y video en una serie de "ventanas" superpuestas. Por ejemplo, los usuarios podían insertar, eliminar y mover texto dentro de un documento. A través de enlaces de "hipertexto", los usuarios también pueden saltar rápidamente entre dos documentos relacionados. El NLS también apoyó la colaboración: cuando varios sistemas NLS se vinculaban entre sí, los usuarios podían trabajar simultáneamente en el mismo documento (Lemelson Center, 2018).

5. Sistemas orientados a ventanas e interfaces WIMP

Los humanos son capaces de realizar múltiples tareas al mismo tiempo y, por lo tanto, cambian con frecuencia su esquema tradicional de pensamiento. El computador personal se hace más flexible para lograr un diálogo efectivo con el usuario. Cuando se utiliza el paradigma de sistemas de ventanas se presentan mensajes al usuario en el contexto de su tarea, por lo que él puede distinguirlos de otros contextos habilitados en otro proceso.

Las interfaces WIMP que representan las siglas de W (*window – ventana*), I (*icon – ícono*), M (*menú*) y P (*pointer – apuntador*) se creó en el laboratorio del *Stanford Research Institute* en California. Su desarrollo continuó en Xerox PARC en el *Palo Alto Research Center*. La estación de trabajo Xerox Star de 1981 se considera la primera computadora en utilizar la metáfora del escritorio, las aplicaciones de productividad y un ratón de tres botones. Debido a problemas de ventas, no se logra comercializar como Xerox esperaba. En 1982 *Apple Computer* implementa la interfaz WIMP en su computadora LISA, y fue en 1984 cuando con la implementación del Macintosh, se logra incorporar todos los elementos de la interfaz, lleva el computador a otro nivel y comienza la carrera del computador personal dirigido a usuarios no experimentados en informática.

6. La metáfora

La metáfora es uno de los tropos fundamentales de la retórica, utilizada por oradores de distintos ámbitos para enriquecer sus “discursos”. Las metáforas se utilizan para enseñar nuevos conceptos en términos de aquellos que ya son conocidos. Esto también funciona con los computadores: muchas de las tareas en un computador se presentan como metáforas de tareas en un entorno de oficina. Uno de los usos más destacados respecto a metáforas en el ámbito de interacción e interfaces fue ideada por Alan Kay (Kay, 1989) y es la metáfora del escritorio.

Una metáfora puede ser inadecuada para promover (e incluso alinearse a) una comprensión completa del computador y sus prestaciones. Habría que agregar que las metáforas representan un sesgo cultural y, por lo tanto, no es muy sencillo crear una metáfora que se entienda en forma universal.

En la Figura 1B se muestran dos ejemplos de metáforas universales (a) Metáfora del semáforo, realmente son los colores del semáforo. En (b) se muestran los íconos que se utilizan en los procesadores de palabra para aquellas tareas de cortar, pegar y copiar, heredado del mundo de la imprenta.

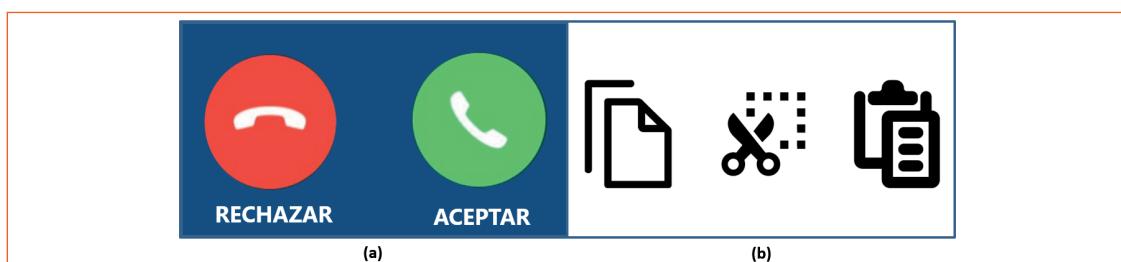


Figura 1B. Ejemplo de metáforas universales (a) Los “colores” del semáforo usado en otro contexto (b) Cortar, copiar y pegar (imprenta). Elaboración propia.

7. Manipulación directa

Es un concepto atribuido a Ben Shneiderman (Shneiderman, 1983) para describir el uso de los sistemas interactivos basados en gráficos como Sketchpad, Xerox Alto y Star. La retroalimentación rápida es solo una característica de la técnica de interacción conocida como manipulación directa. Destaca las siguientes características:

- Visibilidad de los objetos de interés.
- Acción incremental en la interfaz con retroalimentación rápida sobre todas las acciones.
- Reversibilidad de todas las acciones, por lo que se alienta a los usuarios a explorar sin penalizaciones severas.
- Corrección sintáctica de todas las acciones, por lo que cada acción del usuario es una operación legal.
- Reemplazo de lenguajes de comando complejos con acciones para manipular directamente los objetos visibles.
- Enfoque psicológico: metáfora modelo-mundo (compromiso directo).

Shneiderman (en 1982) utilizó por primera vez el término **manipulación directa** para referirse a una clase emergente de sistemas altamente utilizables y "atractivos" de la época, que incluyen editores de pantallas, sistemas de oficina, hojas de cálculo, sistemas CAD y videojuegos.

Estos sistemas presentaban interfaces gráficas que les permitían ser operados "directamente" usando acciones "manuales" en lugar de instrucciones escritas (lenguaje de comando). Las acciones eran mediadas por teclas de función especiales, menús en pantalla y dispositivos apuntadores como ratones, bolígrafos y joysticks. Tales ejemplos parecían cambiar todo el paradigma de interacción humano-computador yendo desde el diálogo (lenguaje de comando, esencialmente) a la manipulación por la utilización de objetos manipulables. Los programadores entendieron que era un lenguaje visual de manera que los usuarios creían que era un mundo interactivo completo.

Como lo evidencia Dix et al (2004, p. 171), la combinación del uso de metáforas y la manipulación directa se complementan, por lo cual indican que: "La metáfora del escritorio hace que el dominio informático de la gestión de archivos, generalmente descrito en términos de "files" y directorios, sea más fácil de entender al compararlo con la presentación en el entorno típico de la oficina, generalmente descrito en términos de documentos y carpetas. El paradigma de manipulación directa y el de la metáfora del escritorio requiere que los documentos y carpetas se hagan visibles para el usuario como íconos que representan los archivos y directorios subyacentes. Una operación como mover un archivo de un directorio a otro se refleja como una acción en el documento visible que se 'recoge y arrastra' a lo largo del escritorio de una carpeta a otra. En una interfaz de línea de comando para un sistema de archivo, es normal que los errores tipográficos en la construcción de la línea de comando para una operación de movimiento den como resultado un comando sintácticamente incorrecto (por ejemplo, escribir mal el nombre

del archivo da como resultado un error...). Es imposible formular una operación de movimiento sintácticamente incorrecta con el estilo de comando de levantar y arrastrar”.

En conclusión, se hace uso de todas las primitivas de interacción descritas anteriormente, acciones naturales que requieren del ambiente adecuado para llevarse a cabo.

8. Lenguaje (diálogo) vs acción

Algunos especialistas argumentan que las interacciones que se basan en lenguaje hablado nunca serán tan efectivas como el concepto de manipulación directa (asociada luego a interfaces gráficas), excepto en situaciones de casos especiales limitados. Esta visión subestima el potencial de las interacciones basadas en el diálogo. En primer lugar, continuará habiendo más y más aplicaciones para las cuales el paradigma de manipulación directa no es factible debido al tamaño del dispositivo con el que se está interactuando, porque la tarea que se está haciendo requiere usar los ojos o las manos, o las capacidades psicomotoras del usuario. En estos casos, el discurso proporciona una modalidad adicional natural y valiosa (Cohen y Oviatt, 1995). Incluso cuando sea disponible una interfaz gráfica de usuario, el diálogo hablado puede ser una modalidad adicional valiosa, porque agrega una flexibilidad considerable y reduce la cantidad de capacitación requerida. Por ejemplo, los diseñadores siempre se enfrentan a un dilema: proporcionar un conjunto relativamente básico de operaciones, obligando al usuario a realizar tareas complejas utilizando secuencias largas de comandos, o agregar comandos de nivel superior que hacen la tarea que el usuario desea.

Un problema al proporcionar comandos de nivel superior es que, en muchas situaciones, existe una amplia gama de tareas posibles; por lo tanto, la interacción se llena de opciones y el usuario requiere una capacitación considerable para aprender a usar el sistema.

Es importante darse cuenta de que una interacción orientada a uso de la voz (lenguaje-diálogo) por sí sola no resuelve este problema. Si se usa para reemplazar las operaciones de selección de menú con decir una frase predeterminada que realiza la operación equivalente, puede agravar el problema porque el usuario necesitaría recordar una lista potencialmente larga de comandos arbitrarios. Sin embargo, las interacciones orientadas a la conversación brindarían al usuario la oportunidad de expresar lo que quiere hacer en sus propios términos, tal como lo haría con otra persona, y el sistema se ocupará de la complejidad.

Si se presenta un requisito durante el diseño de la interacción de un dispositivo en el cual se requiere la inserción de valores o texto (primitivas ya descritas) y el conjunto de usuarios está comprendido en las edades k-12 hasta adultos, es muy probable que haya usuarios niños que no sepan leer/escribir y no podrán utilizar ese dispositivo o aplicación. La solución obvia es pensar en incorporar este paradigma.

9. Hipertexto

Como es natural, un artículo científico o de prensa, se lee de principio a fin, de forma lineal. Sin embargo, dentro de ese texto, a menudo hay ideas o notas al pie de página que instan al lector a desviarse hacia un tema más elaborado.

El formato lineal de información no proporciona mucho soporte para esta tarea de exploración aleatoria y asociada a enriquecer los contenidos (Dix et al, 2004). Esta forma de imaginar una estructura de texto no lineal fue enunciada por Vannevar Bush, en 1945, con la publicación de su artículo "As we may think" en *The Atlantic Monthly* (Bush, 1945).

Vannevar Bush, siendo el administrador científico de más alto rango en EE. UU. (durante la segunda guerra), reconoció que un gran inconveniente de muchos prolíficos esfuerzos de investigación era lo difícil en mantenerse en contacto con el creciente cuerpo de conocimiento científico en la literatura. Consideraba que las personas que pudieran mantenerse al tanto de un flujo cada vez mayor de información obtendrían las mayores ventajas de esa revolución científica. Con ese fin, describió un innovador y futurista aparato de almacenamiento y recuperación de información, el MEMEX – *MEMory EXtended* –, que fue construido con tecnología totalmente existente en 1945 y destinado a aumentar la capacidad humana para almacenar y recuperar piezas de conocimiento "conectadas" imitando nuestra capacidad de crear al azar enlaces asociativos en el cerebro.

Ted Nelson acuñó el término **hipertexto** a mediados de la década de 1960 para reflejar esta estructura de texto no lineal e implementó el sistema XANADU (scihi.org, 2018) para materializar las ideas de Bush e incluyó otras de sus investigaciones. Una implementación de este paradigma fue *HyperCard* y su creador fue Bill Atkinson en 1988, un *software* que se desarrolló en Apple Computer, y se entregaba en todos sus computadores.

El hipertexto ha sido definido como un enfoque para manejar y organizar información, en el cual los datos se almacenan en una red de nodos conectados por enlaces. Los nodos contienen textos y si contienen además gráficos, imágenes, audio, animaciones y video, así como código ejecutable u otra forma de datos se les da el nombre de hipermédio, es decir, una generalización de hipertexto.

Considerando cómo se representa el conocimiento humano, el hombre opera por asociación, saltando de un ítem al próximo, en forma casi instantánea. El paradigma hipertexto intenta modelar este proceso con enlaces entre pedazos de información contenidos en los nodos.

Los sistemas hipertexto están basados en un enfoque en el cual el usuario tiene la posibilidad de crear, agregar, enlazar y compartir información de fuentes diversas, proveyendo la posibilidad de acceder a documentos de manera no secuencial a diferencia de sistemas de información más tradicionales en los cuales el acceso es naturalmente secuencial.

La implementación más conocida de este concepto es el *World Wide Web* – WWW, creado por Berners-Lee en el CERN de Ginebra, a inicios de los años 90.

10. Multimodalidad

La mayoría de los sistemas interactivos todavía usan el teclado tradicional y un dispositivo apuntador para la activar tareas y están restringidos a una pantalla de visualización en color con capacidades de sonido para la entrada/salida. Cada uno de estos dispositivos de entrada y salida puede considerarse como canales de comunicación para el sistema y corresponden a ciertos canales de comunicación humana.

Un sistema interactivo multimodal es un sistema que se basa en el uso de múltiples canales de comunicación humana. Cada canal diferente para el usuario se conoce como una **modalidad de interacción**. En este sentido, todos los sistemas interactivos pueden considerarse multimodales, ya que los humanos siempre han usado sus canales visuales y táctiles para manipular un computador. Sin embargo, los sistemas multimodales genuinos dependen en mayor medida del uso simultáneo de múltiples canales de comunicación tanto para entrada como para salida de la misma manera natural que los seres humanos procesan la información y reaccionan a esas sensaciones.

Los diseñadores han querido imitar esta flexibilidad tanto en la articulación como en la observación al extender las expresiones de entrada y salida que admitirá un sistema interactivo. Por ejemplo, puede modificarse un gesto hecho con un dispositivo apuntador al hablar, indicando qué operación se realizará en el objeto seleccionado. Un ejemplo de esto estuvo disponible en el Mac a finales de los años 80: un programa que “aprendía” algunos vocablos (que eran nombres de ítems de menús) emitidos por el usuario, y “respondía” con la ejecución de la tarea.

Los sistemas multimodales, multimedia y de realidad virtual siguen siendo un área de investigación en diseño de sistemas interactivos.

11. CSCW - Computer-supported cooperative work – Trabajo colaborativo mediado por computadores

En el contexto de la informática personal se trataba de proporcionar a las personas suficiente potencia informática para que se liberaran de los “terminales” que operaban en un sistema de tiempo compartido. Lo interesante es que a medida que las redes de computadores se generalizaron, las personas retuvieron sus poderosas estaciones de trabajo, pues querían reconectarse con el resto de las estaciones de trabajo en su entorno de trabajo inmediato, e incluso en todo el mundo.

Un resultado de esta reconexión fue el surgimiento de la colaboración entre individuos a través del computador, llamada **trabajo colaborativo mediado por computador**, o CSCW (*Computer-support cooperative work*).

La principal distinción entre los sistemas CSCW y los sistemas interactivos diseñados para un solo usuario es que los diseñadores ya no pueden descuidar la sociedad en la que opera un único usuario (y en soledad). **Estos sistemas están diseñados para permitir la interacción entre humanos a través del computador (o dispositivos similares) y, por lo tanto, las necesidades de muchos deben estar representadas en un solo producto.**

Un buen ejemplo de un sistema de este tipo es el correo electrónico, otra metáfora por la cual las personas en ubicaciones separadas físicamente pueden “comunicarse” (de forma asíncrona) a través de mensajes electrónicos que funcionan de manera similar a los sistemas postales convencionales. Dentro de los ejemplos más específicos están los Sistemas de Gestión de Conocimiento presentes en universidades e institutos de investigación y desarrollo, o los casos de sistemas de CRM (*Customer Relationship Management*)

empresariales. Los CSCW siguen siendo tema importante en la investigación actual de la interacción humano-computador.

12. El World Wide Web – WWW - Web

El proyecto *World Wide Web – Web*, fue concebido en 1989 por Tim Berners-Lee, trabajando en el CERN, el Laboratorio Europeo de Física de Partículas en Ginebra. Su objetivo era el de crear un medio que permitiese la distribución generalizada de datos científicos generados en el CERN, y para compartir información entre físicos de todo el mundo. En 1991 se lanzó el primer navegador web basado en texto. Ya en 1993 había disponibilidad de otros navegadores gráficos, por ejemplo, *Mosaic*, quizás el más importante desarrollado por Marc Andreessen en el Centro Nacional para Aplicaciones de Supercomputadora (NCSA) en Champaign, Illinois. Este fue el momento decisivo en el que comenzó el crecimiento exponencial de la web, para dominar el tráfico de Internet y cambiar la visión pública de la informática (*World Wide Web Foundation*, 2019).

La web está construida sobre Internet y ofrece una interfaz de información predominantemente gráfica y fácil de usar, que oculta las complejidades subyacentes de los protocolos de transmisión, las direcciones y el acceso remoto a los datos.

Si bien Internet existe desde 1969, su concepto no se convirtió en un paradigma importante para la interacción hasta la llegada y la disponibilidad de interfaces gráficas (en navegadores) diseñadas para la web. **Los navegadores permiten a los usuarios acceder fácilmente a información multimedia (texto, imágenes, audio, animaciones, video), utilizando solo un ratón para señalar y seleccionar. Este cambio hacia la integración de la computación y la comunicación ha sido transparente para los usuarios.** El usuario entiende que pueden obtener noticias, artículos y otro tipo de información publicada prácticamente al instante.

Una reflexión sobre los cambios de uso de los computadores, con la aparición de la web se refleja en lo escrito por (Dix et al, 2004, p.179), a continuación:

"Actualmente, la web es una de las principales razones por las que los nuevos usuarios se están conectando a Internet (probablemente incluso comprando computadores en primer lugar), y se está convirtiendo rápidamente en una actividad importante para las personas tanto en el trabajo como por placer.

Es mucho más un fenómeno social que cualquier otra cosa, ya que los usuarios se sienten atraídos por la idea de que los computadores ahora son cajas que las conectan con personas interesantes y lugares emocionantes para ir, en lugar de casas sin alma que niegan el contacto social.

La informática solía ser vista como una actividad antisocial. La web ha desafiado esto al ofrecer una 'aldea global' con acceso gratuito a la información y un entorno social virtual. La cultura web ha enfatizado la liberalidad y (al menos en principio) la igualdad independientemente del género, la raza y la discapacidad..."

13. Interacción asistida – Agent based interaction

A menudo se utiliza la palabra "agente" [asesor, asistente] para describir a las personas que tienen una relación de ayuda o asistencia con algún usuario: agentes de viajes, agentes inmobiliarios, etc. El trabajo de dicho agente es actuar de forma autónoma para satisfacer los objetivos que pueda tener un cliente, darles una mayor sensación de productividad y reducir la carga de trabajo.

La imagen de un agente que actúa en un rol de ayuda es más evidente cuando ese rol se refleja directamente en la interacción con el usuario.

Una definición de interacción asistida es presentada por López, Montero, Molina, y González (2006, p.3) donde indican que "utiliza la metáfora del agente que colabora con el usuario en el mismo ambiente de trabajo. En este paradigma de interacción, el usuario en vez de dirigir la interacción trabaja en un entorno cooperativo en que el usuario y los agentes se comunican, controlan eventos y realizan tareas".

Un agente puede "observar" la interacción entre el usuario y una aplicación que presenta una interacción de manipulación directa, puede detectar los objetos que el usuario ve en la pantalla y puede tomar acciones invocando los comandos proporcionados por la interfaz. El agente puede agregar gráficos o animación a la interfaz, puede usar la entrada o salida de voz, o comunicarse a través de otras corrientes sensoriales (Lieberman y Selker, 2003).

Un ejemplo de interacción asistida fue presentado en el año 1987, llamado *Knowledge Navigator* (Apple, 1987). El producto, presentado mediante un video, mostraba la visión que tenía Apple de cómo sería la interacción hombre-computador y cómo sería la futura interfaz bajo esas suposiciones. En el video aparece una situación en la cual un profesor prepara material para una clase, además verifica su horario y calendario, recupera artículos, y cómo se comunica con un colega.

En el video se plantearon avances tecnológicos, como el habla continua y la comprensión del lenguaje natural, pero el elemento más significativo respecto a las interfaces convencionales de esos tiempos fue la presencia de Phil, un personaje que aparecía en la pantalla en un video animado, respondiendo a comandos, ofreciendo consejos, brindando recordatorios, haciendo un seguimiento de los objetivos del usuario (profesor), obteniendo e integrando datos de diversas fuentes cuando era necesario. Phil desempeñaba la función de un asistente a nivel humano.

Considerando este paradigma se reconoce que la disciplina HCI es realmente multidisciplinaria, ya que en este caso se hace uso de las técnicas y heurísticas de la Inteligencia Artificial, entre otras disciplinas.

14. Computación ubicua - Ubiquitous computing

La computación ubicua tiene sus orígenes en 1991 cuando Mark Weiser, científico informático se desempeñaba como CTO (*Chief Technology Officer*) de Xerox PARC. En la publicación

periódica *Scientific America*, Wieser escribe el artículo "The Computer for the 21st century" y en su primer párrafo indica que "Las tecnologías más profundas son las que desaparecen. Se entrelazan en el tejido de la vida cotidiana hasta que no se pueden distinguir de él" (Weiser, 1991, p.94).

La característica que define la computación ubicua es el intento de romper con el paradigma de interacción de escritorio tradicional y mover la potencia computacional al entorno que rodea al usuario. En lugar de obligar al usuario a buscar y encontrar los objetos de la interfaz que proporciona el computador, la informática ubicua sugiere que la interfaz en sí misma puede asumir la responsabilidad de localizar y servir al usuario (Weiser, 1994).

La nueva definición de trabajar o utilizar cualquier tecnología informática permite la interacción humana lejos de una única estación de trabajo/computador personal. Esto incluye tecnología basada en lápiz, dispositivos portátiles o de mano, pantallas interactivas a gran escala, infraestructura de red inalámbrica y tecnología de voz y visión. Los investigadores y desarrolladores se han dado cuenta que el concepto del diseño centrado en el usuario, en este contexto de computación ubicua, y considerando estas tecnologías presenta muchos **desafíos**. Algunos de estos son:

- Definir la experiencia de interacción física apropiada.
- Descubrir las características generales de la aplicación.
- Desarrollar teorías para diseñar y evaluar la experiencia humana dentro de la computación ubicua.

Según Weiser (1994) "el mundo no es un escritorio". Por lo tanto, se sugiere la creación de entornos más elaborados y orientados a crear **interfaces invisibles**. En este sentido Weiser (1994, p.7) afirma que:

"Una buena herramienta es una herramienta invisible. Por invisible, quiero decir que la herramienta no se entromete en tu conciencia; te enfocas en la tarea, no en la herramienta. Los anteojos son una buena herramienta: miras el mundo, no los anteojos. El ciego tocando el bastón siente la calle, no el bastón. Por supuesto, las herramientas no son invisibles en sí mismas, sino como parte de un contexto de uso. Con suficiente práctica podemos hacer desaparecer muchas cosas aparentemente difíciles: mis dedos saben de comandos de edición que mi mente consciente ha olvidado por mucho tiempo. Las buenas herramientas mejoran la invisibilidad".

15. Interacción basada en sensores y sensible al contexto (*Sensor-based and context-aware interaction*)

Permitir que los dispositivos actúen de manera inteligente e independiente es posiblemente el uso más obvio para considerarlos sensibles al contexto. Cuando un dispositivo o cualquier objeto inteligente "sabe" lo que está sucediendo en su entorno, puede facilitarle la vida a su usuario. El contexto a menudo se refiere al entorno del dispositivo, relevante desde el punto

de vista de ejecución de una aplicación. Los dispositivos capaces de adaptar su funcionalidad a entornos cambiantes se denominan **dispositivos sensibles al contexto**. En el ámbito de la interacción hombre-computador el objetivo tradicional es comprender al usuario y el contexto de uso y crear diseños que respalden los principales casos de uso y situaciones de uso **anticipadas**.

Al entrar en la computación sensible al contexto (*Context-Aware Computing*), hacer uso del contexto causa un cambio fundamental: se pueden admitir más de un contexto de uso. En tiempo de ejecución, cuando el usuario interactúa con la aplicación, el sistema puede decidir cuál es el contexto de uso actual y proporcionar una interfaz de usuario específicamente optimizada, para este contexto.

Al considerarse el concepto de interacción sensible al contexto, el proceso de diseño de interacción y de interfaz de usuario se vuelve más complejo a medida que, generalmente, aumenta el número de situaciones y contextos en los que se utilizará el sistema. A diferencia de los sistemas tradicionales, no se diseña para un solo conjunto de contextos de uso.

En el siguiente ejemplo, tomado del portal de la Interaction Design Foundation se explica este cambio de paradigma, no sólo para el usuario, sino también para el diseñador de la interacción e interfaz de usuario.



Interaction Design Foundation

<https://www.interaction-design.org/>

Ejemplo. Tomado de (*Interaction Design Foundation*, 2019). A un especialista en diseño de interacción e interfaces se le pide que diseñe una interfaz de usuario para un reloj de pulsera.

En su investigación, el diseñador descubre que:

- Las personas lo usarán en ambientes cerrados (interiores) y al aire libre (exteriores).
- Lo usarán tanto en la oscuridad como a la luz del sol.
- Lo usarán cuando corran para tomar el tren y cuando se sienten en una conferencia y estén aburridos.

Un buen diseñador termina con muchas ideas para lograr una interfaz de usuario para cada situación. Considerando alguno de los casos, se diseña pensando en lo siguiente:

- Cuando el usuario está corriendo para tomar el tren, la interfaz de usuario debe mostrar el tiempo resaltando los minutos y segundos en fuente tipográfica grande.
- Cuando el usuario asiste a una conferencia, la interfaz de usuario debe mostrar la hora en una fuente pequeña y, además, proporcionar otras informaciones.

En un proceso de diseño tradicional, el diseñador se daría cuenta, después de crear sus bocetos del diseño, que tiene que decidir cuál de sus ideas para la interfaz de usuario desea utilizar.

Seguramente se dará cuenta que respaldar todas las situaciones en un solo diseño no funcionará. El resultado típico es un compromiso, que a menudo pierde gran parte de las ideas que inicialmente se le ocurrieron.

Sin embargo, si adopta el enfoque de *Context-Aware Computing*, se podría crear un reloj sensible al contexto, donde se combinen todos los diseños optimizados para la situación en un solo diseño. Si el reloj se diseñó de modo que pudiera reconocer cada una de las situaciones evidenciadas en su investigación inicial (por ejemplo, correr para tomar el tren, asistir a una conferencia, etc.), el reloj podría reconfigurarse según el contexto reconocido.

Otro campo en crecimiento para incorporar este paradigma es la automatización del hogar (también conocida como la "casa inteligente") que aumenta la automatización de activación de dispositivos mediante el uso de sensores e inteligencia artificial.

En este anexo se describieron los distintos paradigmas de interacción, sus características y se muestra la evolución de las tecnologías que hicieron posible el diseño de la interacción humano-computador y las interfaces de usuario. Los paradigmas enumerados no son excluyentes, sino que pueden utilizarse en forma armónica y combinada, en base a los requisitos del usuario y de los dispositivos con los cuales interactúa.

Anexo C. Características y clasificación de requisitos de *software*

1. Características de un requisito

Los requisitos contienen una mezcla de contenidos/información sobre aspectos, declaraciones de comportamiento y propiedades de un sistema de *software*, así como restricciones de diseño, implantación y uso.

Los altos costes y las posibilidades de fracaso durante el desarrollo de un sistema de software se reducen significativamente definiendo correctamente los requisitos del nuevo sistema, a partir de la fase más temprana del desarrollo. **Todo lo recopilado en el momento de la obtención de requisitos es una combinación compleja de necesidades y deseos de diferentes interesados, que generalmente pertenecen a diferentes niveles de una empresa y al entorno en donde será utilizado el sistema.**

Para minimizar los posibles errores en la descripción y posterior especificación de los requisitos, evitar estimaciones erróneas de presupuesto y tiempo de desarrollo, utilizarlos adecuadamente en el diseño de la solución y en especial cuando se deben validar con los interesados, los requisitos deben cumplir con distintas características, tales como las que se indican a continuación (Arias, 2007) (Rodríguez, 2017):

- a) Deben estar escritos o “descritos” en forma **concisa, fácil de leer y entender.**
- b) Deben ser **capaces de ser especificados en forma escrita** o en alguna notación conocida por los interesados (desarrolladores, clientes y usuarios), como todo contrato o acuerdo entre varias partes.
- c) Deben ser **completos**, es decir deben estar especificados con información suficiente y sin necesidad de mayores detalles para su comprensión.
- d) Deben ser **expresados en forma precisa y no ambigua**, deben tener una única interpretación, y es importante la selección del lenguaje que se utilice para su definición.
- e) Deben ser **verificables**, cuando a través de un proceso una persona o herramienta puede constatar que el sistema satisface lo establecido en el requisito.
- f) Deben ser **consistentes**, pues un requisito no debe contradecir otro o generar algún conflicto entre ellos.
- g) Deben **poder priorizarse**, es decir, deben permitir determinar su importancia con relación a las otras necesidades de los usuarios.

Los requisitos se recogen en el **Documento de Especificación de Requisitos del Software**. Sin embargo, las declaraciones de cada uno de los “deseos” o funcionalidades no son suficientes para obtener tal documento con atributos de calidad. **Las especificaciones deben cumplir con las siguientes características:**

- a) **Deben ser completas.** Aunque es cierto que algunos requisitos pueden aparecer durante la verificación con los interesados, es fundamental tratar de registrar la especificación de todos los requisitos. A medida que aparezcan durante el proceso, deben registrarse y verificar dicha especificación.
- b) **Deben ser modificables.** Si al realizar un cambio sobre las definiciones (requisitos) ya establecidas de forma completa y consistente, se considera que algún requisito debe ser modificable por cualquier circunstancia, entonces cada cambio realizado al requisito debe ser registrado en el documento de especificación.
- c) **Deben ser trazables**, es decir se puede rastrear en forma adecuada el origen de cada requisito y de las funcionalidades o características asociadas a ellos. Se escriben en forma estructurada, de granularidad fina (por ejemplo, haciendo uso de modelos). Se mejora la trazabilidad cuando un requisito no puede ser descompuesto en otros, y es de utilidad al momento del diseño.
- d) **Deben ser útiles y utilizables**, en especial para las tareas de mantenimiento y evolución de los sistemas de software.

Los lectores del documento de especificación de requisitos de un nuevo sistema son a menudo ingenieros prácticos. Son profesionales que se sienten cómodos ante las descripciones de

implementación por su capacidad de entender enunciados de problemas muy abstractos. **Los requisitos deben escribirse de forma que sean comprensibles para los posibles lectores del documento.**

En un número importante de casos, el sistema de *software* que se trata de definir (mediante alguna especificación) es quizás solo uno de varios componentes que “vivirán” en un entorno informático y de soporte organizacional. **Para lograr una compatibilidad entre los componentes (y su entorno) y para cumplir con los estándares y con las preocupaciones de la organización, es posible que se deba especificar también las políticas de implementación que limiten las opciones de los desarrolladores del sistema.**

Por lo tanto, los requisitos invariablemente contienen una mezcla de información de problemas, declaraciones de comportamiento y de propiedades del sistema y restricciones de diseño y fabricación.

A continuación, se tratará lo relativo a la clasificación de requisitos para identificar el objetivo operativo del *software* a desarrollar, así como otras características que se deben tomar en cuenta respecto al producto final.

2. Clasificación de requisitos

Los requisitos son el lenguaje común y el medio más simple que utilizan los ingenieros para conectar las necesidades del cliente con los diferentes miembros del proyecto de desarrollo, tales como gerentes, desarrolladores, usuarios, etc.

En la literatura tradicional de la ingeniería de *software* existe una variedad de clasificaciones y/o tipologías de requisitos. En su mayoría están de acuerdo en clasificarlos en **requisitos funcionales (RF)**, **requisitos no funcionales (RNF)** y **restricciones**. Sin embargo, existen diferentes tipos de información asociada al concepto de requisitos que no encajan fácilmente en esta tipología. Tal es el caso de las definiciones que se listan en la Tabla 1C (Wieggers y Beatty, 2013, p. 7), donde se presenta un conjunto coherente de términos utilizados en la literatura para entender el concepto de requisito.

Tabla 1C
Algunos términos utilizados en la Ingeniería de Requisitos.

| Término usado en la literatura | Definición |
|--------------------------------|--|
| Requisitos de Negocio | Un objetivo comercial de alto nivel de la organización que construye un producto o de un cliente que lo obtiene. |
| Reglas del negocio | Una política, directriz, norma o regulación que define o restringe algún aspecto del negocio. No es un requisito de <i>software</i> en sí mismo, sino el origen de varios tipos de requisitos de <i>software</i> . |
| Restricción | Una restricción que se impone a las opciones disponibles para el desarrollador, para el diseño y la construcción de un producto. |
| Requisito de interfaz externa | Una descripción de una conexión entre un sistema de <i>software</i> y otro sistema, o con un dispositivo de <i>hardware</i> . |
| Característica | Una o más capacidades del sistema relacionadas lógicamente que proporcionan valor a un usuario y están descritas por un conjunto de requisitos funcionales. |

| Término usado en la literatura | Definición |
|--------------------------------|---|
| Requisito Funcional | Una descripción de un comportamiento que un sistema exhibirá bajo condiciones específicas. |
| Atributo de calidad | Un tipo de requisito no funcional que describe un servicio o característica de rendimiento de un producto. |
| Requisito del sistema | Un requisito de nivel superior para un producto que contiene varios subsistemas, que podría ser todo <i>software</i> o <i>software y hardware</i> . |
| Requisito de usuario | Un objetivo o tarea que las clases específicas de usuarios deben poder realizar con un sistema o un atributo de producto deseado. |

Nota. Adaptado de Wieggers y Beatty, 2013, p. 7.

Los conceptos anteriores y la relación que existe entre ellos se observan en la Figura 1C que representa un marco de referencia de la Ingeniería de requisitos según Wieggers y Beatty (2013). Los óvalos de la Figura 1C representan tipos de información de requisitos, y los rectángulos indican documentos en los que se almacena esa información. Las flechas sólidas indican que un cierto tipo de información normalmente se almacena en el documento indicado. Las reglas de negocio y los requisitos del sistema se almacenan por separado de los requisitos de *software*, como en un catálogo de reglas de negocios o en la especificación de requisitos del sistema, respectivamente. Las flechas punteadas indican que un tipo de información es el origen o influye en otro tipo de requisito (Wieggers y Beatty, 2013, p. 8).

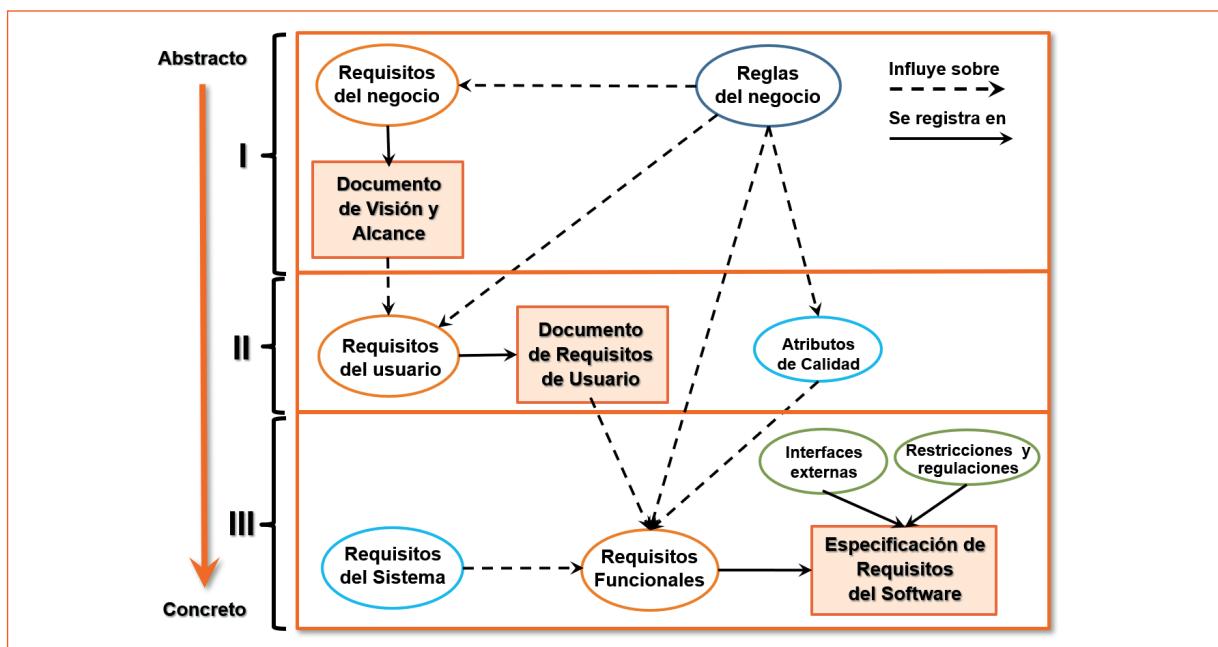


Figura 1C. Marco de referencia de la Ingeniería de requisitos. Relación entre los tipos de requisitos y documentos que se generan. Adaptado de Wieggers y Beatty (2013, p. 8).

En más detalles y en base a lo anterior se puede llegar a la siguiente **clasificación de requisitos** (Wieggers y Beatty, 2013):

- a. **Las reglas del negocio** definen el **dominio** del producto. Están conformadas por regulaciones, contratos, convenciones, etc. que rigen el negocio y su dominio.

- b. **Los requisitos del negocio** representan el “**por qué**” una organización decide emprender el desarrollo de un producto de software.
- c. **Los requisitos del usuario** describen “**qué**” tareas y acciones podrá realizar el usuario (todos) con el producto de *software*. Estas descripciones se recogen en el **Documento de Requisitos de Usuario**.
- d. **Los requisitos funcionales** especifican el comportamiento que muestra el producto bajo distintas circunstancias. Describen “**qué deben implementar**” los desarrolladores para que el usuario pueda realizar las tareas con el producto. En muchos casos se incluyen, además todas las descripciones de los datos que estarán asociadas al comportamiento, con la forma en que las funcionalidades serán utilizadas por el usuario final del sistema.
- e. **Los requisitos del sistema** describen los requisitos de cada uno de los componentes y subsistemas que conforman el producto final. Es habitual que un sistema esté conformado por el *software* (dividido en componentes), puede incluir un tipo particular o general de *hardware*, el equipo de usuarios son la parte operativa del sistema, y el grupo administrador del sistema, etc.
- f. **Los atributos de calidad**, las **interfaces externas**, las **restricciones y regulaciones** que actúan sobre el dominio del negocio y del producto, describen “**qué factores de calidad y de servicios, restricciones o regulaciones**” debe cumplir el producto. A este tipo de información se le asocia el nombre de **requisitos no funcionales**.

Al marco de referencia ilustrado de la Figura 1C se le ha agregado el nivel de detalle (desde lo más abstracto hacia lo más concreto) de los conceptos y contenidos que se recogen en los distintos documentos, a saber:

- **Nivel I – Nivel Negocio:** es el nivel más abstracto y es donde se recoge el conjunto de beneficios que logra dicha organización con el producto. Todo esto se “almacena” en el Documento de visión y alcance del negocio. **Los requisitos de negocio describen las razones que llevaron a una empresa a desear el sistema.** Las razones pueden expresarse como metas, objetivos o beneficios que la organización quiere alcanzar. En otros casos, las razones pueden definirse como las necesidades de la organización o los problemas que deben resolverse.
- **Nivel II – Nivel Usuario:** tiene como objetivo identificar las tareas que el usuario puede realizar en el sistema para alcanzar un objetivo. **Los requisitos del usuario** generalmente se expresan como casos de uso, historias de usuario (SCRUM MANAGER, 2019) o escenarios, que pueden ser entendidos por el cliente y otros interesados. Durante la gestión de requisitos orientada al usuario, los ingenieros también descubren características, llamadas atributos de calidad, con el propósito de proporcionar un valor agregado a la satisfacción del usuario. En este nivel se concretizan distintos aspectos que se almacenan en el **Documento de requisitos de usuario**.
- **Nivel III – Nivel Funcional:** se identifican los **requisitos funcionales y los no funcionales**, es decir una descripción más detallada del comportamiento del sistema y características (a veces

no tangibles) con las que debe cumplir el sistema. Para expresar de forma más concreta estos requisitos, se utilizan insumos adicionales como la información sobre las interfaces con otros sistemas o componentes externos (interfaces externas) que afectan el sistema, las restricciones del sistema, las reglas del negocio relacionadas con las limitaciones desde el punto de vista del desarrollo, así como el cumplimiento de estándares y restricciones más generales como el tiempo, presupuestos, etc. El producto obtenido en este nivel es el **Documento de especificación de requisitos del software**.

En la Tabla 2C se resume la información sobre los niveles y se describe la procedencia de los requisitos y los productos generados.

Tabla 2C

Descripción resumida de niveles de procedencia de requisitos y los productos que se obtienen.

| Nivel | Descripción | Productos generados |
|--|--|---|
| I Negocio (Razón de la creación del producto de software) | Son las razones (y deseos) de una organización o empresa para justificar el sistema solicitado; se generan todas las definiciones sobre la visión del producto asociado a los objetivos (beneficios) del negocio y el alcance. Son necesidades a alto nivel que el producto debe satisfacer. Con la visión del producto se determinan las características y usos del producto y las limitaciones del mismo. | Documento de visión y alcance del negocio |
| II Usuario (Actividades, funcionalidades y tareas que tendrá disponible el producto de software) | Son las definiciones del producto desde el punto de vista del usuario. Se incluyen las funcionalidades y tareas y todas las características asociadas a la calidad del producto. Estas definiciones son más detalladas y con ellas se logra enlazar los objetivos del negocio y la utilización del producto. Los analistas del problema deben tener capacidades y habilidades de comunicación para descubrir las necesidades y expectativas que tendrá el usuario con el producto de software. | Documento de Requisitos de Usuario |
| III Funcional (Definiciones concretas que los desarrolladores deben considerar para el diseño del software) | Este nivel produce descripciones detalladas de todos los requisitos funcionales y no funcionales que debe cumplir el sistema considerando las necesidades recogidas en el nivel negocio y el nivel usuario. En este nivel se obtiene el documento de especificación de requisitos de software, con el cual se establecen contratos (acuerdos) entre los técnicos y los clientes del sistema y son la base para la verificación de las características del producto. | Documento de Especificación de Requisitos del Sistema |

Nota: Elaboración propia.

La clasificación que propone Wiegers y Beatty (2013) no exhibe lo referente al comportamiento definitivo del sistema, la evolución que puede sufrir debido a cambios tecnológicos, nuevos atributos de calidad, etc.

De hecho, los cambios tecnológicos desde el punto de vista de plataformas de despliegue, y el caso de aplicaciones y sistemas de software en ambiente web, ha generado nuevos tipos de requisitos que no estaban presentes en los sistemas de software anteriores a Internet, el web, al uso de los *smartphone* y a las tabletas.

Con la intención de cubrir una gama importante de tipos de aplicaciones y productos de *software*, así como toda la información necesaria que enriquecen un documento de especificación de requisitos, **en este libro se utilizará la siguiente clasificación de requisitos** (Bianchini, Ortega y Suárez, 2008), (Bianchini, 2013), (Engels, Lohman y Wagner, 2006):

- 1. Requisitos del negocio**, donde se incluyen la misión, visión y reglas del negocio. Estos requisitos se refieren a un conjunto de información que describe una necesidad que lleva a uno o más proyectos a ofrecer una solución y los resultados comerciales finales deseados. Las oportunidades del negocio, sus objetivos, las métricas de éxito y una declaración de visión conforman este tipo de requisitos. Son la fuente para la creación del documento de visión y alcance del negocio.
- 2. Requisitos Funcionales (RF)**, son entendidos como las capacidades y funcionalidades en todas sus dimensiones, que debe exhibir un *software* con el fin de resolver un problema dado. Se consideran todos los requisitos atinentes a las necesidades del usuario desde los contenidos y la información a visualizar, gestionar, recuperar y navegar, las características de interacción e interfaz, la personalización del usuario, y los servicios intrínsecos del sistema. Estos requisitos llevan a la creación del documento de requisitos del usuario.

Esta categoría de requisitos se divide en:

- a) Requisitos de datos / información:** se denominan también requisitos de contenido, requisitos conceptuales o requisitos de almacenamiento de información.

Este tipo de requisitos responde a preguntas del estilo ¿qué información/datos debe almacenar y administrar el sistema para ser utilizados en las tareas de un usuario?

- b) Requisitos de interfaz:** son llamados también requisitos de interacción entre el usuario y el sistema.

Responden a las siguientes preguntas: ¿Qué paradigma(s) de interacción se le proporcionará al usuario del sistema? ¿Cómo va a interactuar el usuario con el sistema? ¿Cuáles son los elementos de la interfaz? ¿Cuáles serán las plataformas de despliegue?

- c) Requisitos de navegación:** recogen las necesidades de navegación del usuario en el cuerpo de contenidos que se gestione en la aplicación. Describe el flujo entre las distintas acciones en una misma funcionalidad y entre las distintas funcionalidades que proporciona el sistema.

Este tipo de requisitos es importante en el caso del diseño de aplicaciones web orientadas a contenido, específicamente para la definición de la **arquitectura de la información** y en los casos de creación de mapas de contenido.

- d) Requisitos de personalización:** describen cómo debe adaptarse el sistema en función de qué tipo de usuario interactúe con él y de la descripción actual de dicho usuario. Al identificar estos requisitos se obtiene el modelo de actores del sistema.

- e) **Requisitos transaccionales:** recogen las necesidades sobre qué debe hacer el sistema internamente, sin incluir aspectos de interfaz o interacción. También son conocidos como requisitos de servicios en algunos enfoques tales como los descritos en (Grünbacher, 2006) y (Escalona, 2004). Se utilizan para la definición en el *back-end* de las acciones asociadas a cada funcionalidad que proporciona el *software*.
3. **Requisitos No Funcionales (RNF)**, llamados también requisitos derivados de los atributos de calidad. Definen aquellos niveles deseables de calidad de las funcionalidades y demás servicios que provee el *software*. Como mencionan García, Conde y Bravo (2008), estos requisitos pueden no estar relacionados directamente con la funcionalidad del sistema y en muchas oportunidades suelen ser requisitos más críticos que los requisitos funcionales. Pueden ser verificables una vez diseñados aquellos aspectos relacionados con los requisitos funcionales.

Para definir el tipo de requisitos no funcionales, los autores y desarrolladores se basan en el estándar ISO/IEC 9126 (atributos de calidad). Éste define un modelo independiente de la tecnología para caracterizar la calidad de *software* y considera, entre otras, las siguientes características:

- a. **Funcionalidad:** describe la presencia de funciones para alcanzar propiedades definidas. Ejemplos: interoperabilidad, seguridad.
- b. **Confiabilidad:** describe la disponibilidad que tiene el producto para mantener sus niveles de rendimiento bajo condiciones específicas y en un tiempo dado. Ejemplos: madurez, tolerancia a fallas y las posibilidades de recuperación.
- c. **Usabilidad:** describe la satisfacción y el esfuerzo requerido por el usuario en la utilización del *software*; por ejemplo, la eficacia, la velocidad y la operabilidad.
- d. **Eficiencia:** describe la tasa entre el nivel de rendimiento del *software* y los recursos que esta utiliza bajo condiciones específicas.
- e. **Capacidad de mantenimiento:** describe el esfuerzo requerido para implementar aquellos cambios predeterminados en el *software*; por ejemplo: la estabilidad o las validaciones.
- f. **Portabilidad:** describe la factibilidad de que una aplicación o sistema de *software* pueda ser llevado de un ambiente a otro. Ejemplo: adaptabilidad, capacidad de instalación, capacidad de réplica.

En la actualidad, debido a la globalización, algunos productos pueden verse afectados por aspectos como la localización para aquellos sistemas que deben considerar las distintas culturas, idiomas, monedas, terminología, y otras características que se proporcionan a los usuarios. Estos aspectos son considerados también requisitos no funcionales, influyen en la implementación de distintas funcionalidades, en la instalación y despliegue del sistema, y por lo tanto inciden fundamentalmente en los requisitos funcionales.

4. **Restricciones y regulaciones**, son condiciones no negociables que afectan al proyecto y entre ellas se pueden considerar: presupuesto, competencia del equipo desarrollador, contratos sobre utilización de plataformas específicas, proceso de transición entre la solución actual y la propuesta, entre otros.
5. **Requisitos del Sistema**, son necesarios para definir servicios, componentes o algún tipo especial de *hardware/software* y las interfaces que existen para la comunicación entre esos elementos u otros sistemas.
6. **Requisitos de evolución**, se utilizan para planificar a mediano y largo plazo aquellos aspectos asociados al crecimiento, la evolución y cambios que pudiera sufrir el *software*. Ejemplos de esto pueden ser: "el número de usuarios concurrentes se incrementará cada año en X%", "el catálogo de productos se ampliará cada 6 meses con 3 categorías nuevas".

Haciendo una integración de los tipos de requisitos recién expuestos y utilizando el marco de referencia propuesto por Wieger y Beatty (2013), en la Figura 2C se presentan los tipos de requisitos, la relación entre ellos y los documentos que se van generando según la clasificación anteriormente descrita.

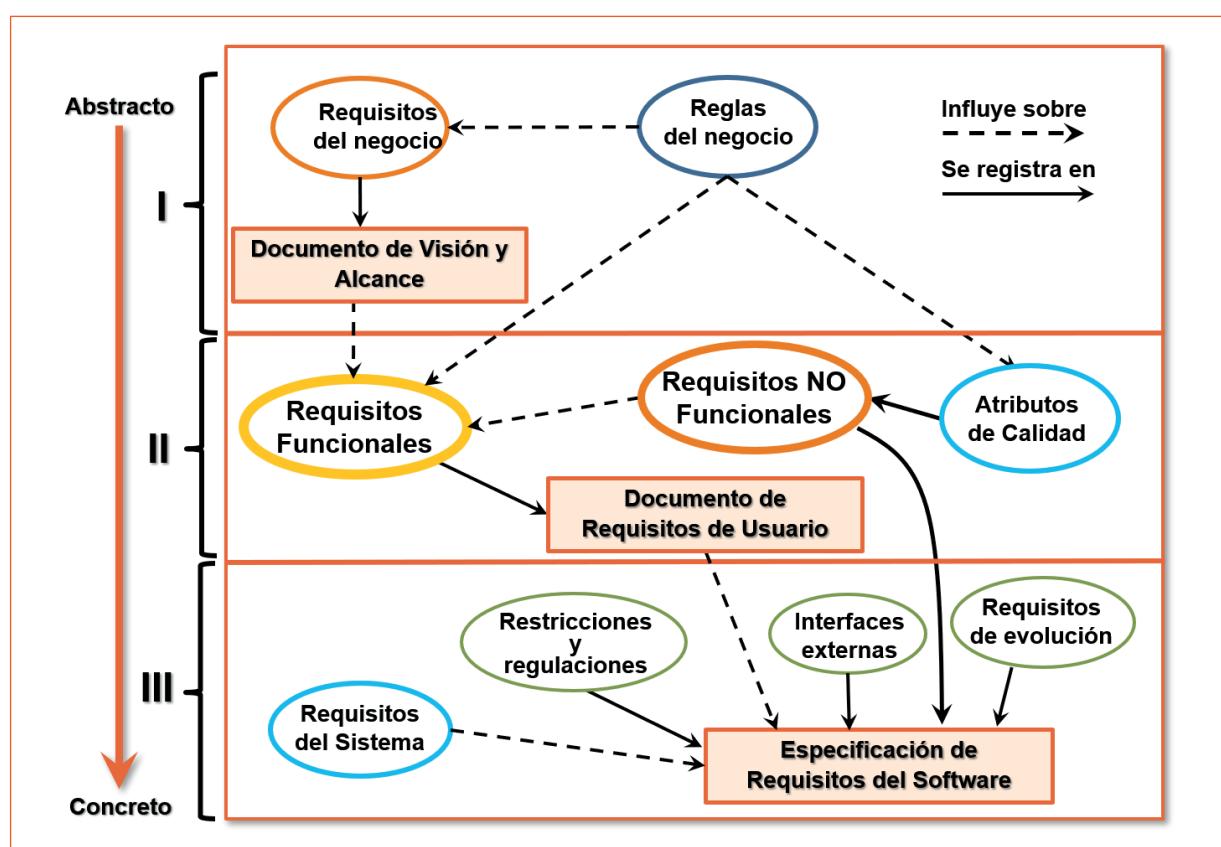


Figura 2C. Marco de referencia de la Ingeniería de requisito, integrando requisitos no funcionales, relación entre tipos de requisitos y documentos generados. Adaptado de Wiegers y Beatty (2013, p. 8).

Durante el proceso de la ingeniería de requisitos, se identifican los distintos requisitos, se evidencia la respectiva categoría, y se deben plasmar en el documento de especificación de requisitos del *software*.

Anexo D. Utilización de mensajes en las interfaces de usuario

Uso de palabras técnicas

Debe considerarse que el usuario no necesariamente posee conocimientos en ámbitos informáticos, especializados, por lo tanto, hay que evitar el uso de palabras que pertenecen a un léxico profesional o hacer referencia a una terminología muy específica. Se recomienda escribir mensajes con contenidos para todos los niveles de usuario. Algunos consejos prácticos son:

- Para cada mensaje, el diseñador debe pensar si el usuario realmente necesita saber todo lo escrito en el mensaje.
- El contenido debe presentarse en pocas líneas de texto. Se recomiendan menos de 30 palabras cuando sea posible.

Uso de palabras para números

El área de visualización, en muchas aplicaciones, puede resultar muy limitado. Se recomienda utilizar el número en lugar de la palabra, siempre que sea posible.

Ejemplo:

Poco recomendado: "Ha recibido cuatro nuevos correos"

Recomendado: "Ha recibido 4 nuevos correos"

Uso de textos en mayúsculas

Los mensajes se diseñan para que el usuario los lea. Está demostrado que la lectura de un texto en mayúsculas no es fácil de leer. Se recomienda utilizar las mayúsculas para encabezados cortos, etiquetas o en elementos de menús. En el mundo digital (*netiquette*), una palabra en mayúsculas es "traducido" como "gritos".

Género masculino y femenino para expresarse

Se recomienda utilizar palabras que no indiquen o hagan referencia directa al género del usuario. Si bien la informática está muy influenciada por el idioma inglés (donde no hay género, sino que hay palabras para cada género), es recomendable evitar un tratamiento incorrecto al usuario.

Mensajes de error

Los mensajes de error son necesarios para mantener informado el usuario sobre la situación de su desempeño. En lo posible considere que:

- En el contenido del mensaje debe estar explícito lo que ocurrió (el error) y el por qué.
- Indicar la acción que debe tomar el usuario para recuperarse de ese error.

Nunca tratar al usuario como el responsable de los errores

Si bien los usuarios cometen errores, también es cierto que debe pensarse en el principio de tolerancia en el diseño de interfaces. En este orden de ideas, lo importante es saber comunicar el problema/error, qué se cometió, en forma afirmativa y nunca negativa. Por ejemplo:

Mensaje incorrecto: "*Ha definido una clave incorrecta*"

Mensaje correcto: "*La clave debe contener sólo 8 dígitos*".

Anexo E. Algunos principios de diseño de interfaces

Anticipación

Las aplicaciones deberían intentar anticiparse a las necesidades del usuario y no esperar a que el usuario tenga que buscar la información, recopilarla o invocar las herramientas que va a utilizar.

Ejemplo: en un procesador de texto se anticipa a las necesidades del usuario, proporcionando las características del texto seleccionado -fuente, tamaño, alineación, etc.- permitiendo que el usuario pueda modificarlas ágilmente.

Buscar la eficiencia del usuario

Se debe considerar la productividad del usuario antes que la productividad de la máquina. Si el usuario debe esperar la respuesta del sistema por un período prolongado, estas pérdidas de tiempo se pueden convertir en pérdidas económicas para la organización.

Los mensajes de ayuda deben ser sencillos y proveer respuestas a los problemas. Los menús y etiquetas de botones deberían tener las palabras claves del proceso.

Considerar la Ley de Fitt

El tiempo para alcanzar un objetivo es una función de la distancia y tamaño del objetivo. Es por ello que es conveniente usar objetos grandes para las funciones importantes.

Uso de valores por defecto

No se debe utilizar la palabra "Por defecto" (*default*) en una aplicación o servicio para indicar que son valores iniciales. Puede ser reemplazada por "Estándar", "Definida por el Usuario", "Restaurar Valores Iniciales" o algún otro término específico que describa lo que está sucediendo.

Los valores por defecto deberían ser opciones inteligentes y sensatas. Además, los mismos tienen que ser fáciles de modificar.

Mantener informado al usuario

Aspectos a considerar:

- ¿Qué está haciendo el sistema?
- ¿Cómo se interpretan los comandos del usuario?
- El usuario debe saber en cada momento que está sucediendo

Algunos tipos de retroalimentación son:

- **Respuesta a un comando del usuario:** movimiento del cursor, selección de un menú, etc.
- **Estado actual:** herramienta seleccionada (de una paleta), color, posición, dirección de la carpeta (directorio), *scroll bars*.
- **Procesamiento por parte de la máquina:** instantáneos, cursores de espera, diálogos explicativos, barras de progreso.

El control es del usuario

Es el usuario el que está en control del sistema y no el computador (o aplicación). El usuario inicia y controla las actividades.

Tolerancia:

- Posibilidad de ofrecerle al usuario la alternativa de recuperarse de algún error.
- Nunca ofrecer un comando que lleve a un mensaje como "Comando ilegal".
- Utilizar controles que impidan introducir datos erróneos.

Interacción simple y natural, se traduce en considerar aspectos como:

- Minimizar los elementos de interfaz.
- Menos para aprender, para equivocarse, distraerse.
- Orden natural de la información:
 - Agrupación gráfica de la información relacionada.
 - Orden de acceso a la información debe ser como el usuario la espera.
 - Esconder o eliminar información irrelevante o usada ocasionalmente.

- Utilizar el lenguaje del usuario:
 - Usar terminología e iconografías familiares al usuario.
 - Traducir mensaje de error al lenguaje del usuario.
- Recordar la ley de Hick: el tiempo que lleva tomar una decisión aumenta con el número de alternativas y su complejidad.

Satisfacer múltiples niveles de habilidad de los usuarios

En caso de usuarios casuales, considerar:

- Incorporar tutoriales, *wizards*, *prompts*, ayudas.
- Proporcionar un modo de interacción simple: esconder los comandos complejos.
- Preferir manipulación directa.
- Usar valores iniciales.

En caso de usuarios expertos:

- Proporcionar atajos de teclado.
- Proporcionar el estilo de líneas de comando.
- Proporcionar la opción de cambiar a modo experto.
- Eliminar los *prompts* y diálogos de advertencia.
- Proporcionar mecanismos de interfaz extensible y personalizable.

Minimizar la memorización, y para eso lo importante es:

- Promover el reconocimiento.
- Basarse en la visibilidad de los objetos.
- Usar menús, íconos, diálogos, mensajes, palabras, metáforas.
- Recordar la ley de Miller: un usuario promedio solo puede mantener alrededor de 7 elementos en su memoria de trabajo.

Integridad y estética adecuada

- La información debe estar organizada en forma adecuada y consistente con los principios de diseño visual que se han tomado de referencia.
- El número de objetos de interfaz y su respectivo comportamiento debe ser limitado para aumentar la “usabilidad” de la interfaz.

- Se debe asegurar de mantener la semántica del lenguaje gráfico o del lenguaje asociado a la interfaz.
- No cambiar el significado de los objetos que son estándares conocidos.
- Recordar el efecto Von Restorff, también conocido como Efecto de Aislamiento: predice que cuando hay varios objetos similares presentes, es más probable que se recuerde el que difiere del resto.

Uso de metáforas

El uso de metáforas es un gran aliado comunicacional y práctico para los diseñadores de interfaces. Algunas de las razones son:

- **Utilidad práctica:** la metáfora es una descripción de una situación desconocida utilizando una situación conocida, y permite un entendimiento más rápido de la situación desconocida y se acelera el aprendizaje de un nuevo producto mediante el conocimiento que se tiene del mundo real.
- **Utilidad comunicacional:** la metáfora va creando comunicación mediante su posibilidad de dejar de ser metafórica y convertirse en expresión directa de la lengua;

Anexo F. Recursos en línea para la evaluación de usabilidad

1. Herramientas en línea para realizar evaluación heurística y otros aspectos de las interfaces.

Design Toolkit. La herramienta *Design Toolkit* de la UOC (Universitat Oberta de Catalunya, 2019) es un conjunto de herramientas de diseño que incluye métodos, modelos y principios de diseño que se proponen para llevar a cabo proyectos. El *toolkit* recoge las herramientas más representativas y más utilizadas. La herramienta está formada por fichas con la descripción de cada método, modelo o principio. Algunos de los métodos tienen una guía asociada que busca acompañar en su ejecución. Algunas de las fichas disponibles son:



Guía Arquitectura de la información

<http://design-toolkit-test.uoc.edu/es/guia/arquitectura-de-la-informacion/>



Guía Evaluación heurística

<http://design-toolkit-test.uoc.edu/es/guia/arquitectura-de-la-informacion/>



Guía Factores humanos

<http://design-toolkit-test.uoc.edu/es/guia/factores-humanos/>



Guía Observación contextual

<http://design-toolkit-test.uoc.edu/es/guia/observacion-contextual/>

 **Guía Persona**

<http://design-toolkit-test.uoc.edu/es/guia/persona/>

 **Guía Test con usuarios**

<http://design-toolkit-test.uoc.edu/es/guia/test-con-usuarios/>

2. **UXCheck.** Es una extensión del navegador Chrome que ayuda a identificar problemas de usabilidad a través de una evaluación heurística.

 **UXCheck**

<http://www.uxcheck.co/>

3. **IXCHECKLIST.** Herramienta en línea para verificar aspectos de usabilidad de sitios web.

 **IXCHECKLIST**

<https://idxchecklist.com/>

4. Lista de verificación de aspectos para llevar a cabo una evaluación heurística de usabilidad (Pierotti, 2014).

 **Heuristic Evaluation**

http://eitidaten.fh-pforzheim.de/daten/mitarbeiter/blankenbach/vorlesungen/GUI/Heuristic_Evaluation_Checklist_stcsig_org.pdf?source=post_page-----

5. Lista de verificación para aplicaciones móviles (Yáñez, Cascado y Sevillano, 2014).

 **Heuristic Evaluation on Mobile Interfaces: A New Checklist**

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4177852/>

6. Lista de verificación para la accesibilidad de contenidos web. Guía versión 2.0. (W3C, 2005).

 **Web Content Accessibility Guidelines 2.0 Checklist**

<https://www.w3.org/WAI/GL/2005/06/checklist-proto.html>

Glosario

Accesibilidad

Es la cualidad y capacidad de un producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

Arquitectura de la información

Es una disciplina que parte de la bibliotecología, pues su fin principal es catalogar la información en un espacio con contenidos disímiles o no. Se encarga de estudiar la mejor forma de mostrar jerarquías de la información, nombres de categorías, facetas y combinación de otras características de ese contenido. Suponga el caso de un sitio web orientado al comercio electrónico de productos informáticos: se hace evidente que los productos pueden ser recuperados en base al tipo físico, es decir *hardware* (computador, laptop, *smartphone*, servidores, dispositivos almacenamiento externo), o puede ser *software* (programas de ofimática, programas CAD, programas de juego, etc.). A cada una de ella se le asocian las facetas marca, precio, tecnología, sistema operativo, etc. Al obtener la arquitectura (organización y catalogación) de información se pueden diseñar aquellas herramientas para que el usuario pueda recuperar lo que necesite de una manera efectiva y productiva.

Características de calidad del software

Son propiedades de un producto o servicio que determina el punto de vista de calidad de los interesados principales y que son caracterizados por determinadas métricas. No están directamente asociados a funcionalidades del producto o servicio, aunque tienen una influencia significativa en la arquitectura. Las características de calidad más utilizados (a alto nivel) son: funcionalidad, rendimiento, disponibilidad, confiabilidad, usabilidad, tolerancia a fallas, etc. Los atributos de calidad en el entorno de desarrollo se refieren a los elementos a considerar para garantizar un adecuado desarrollo del producto y se relacionan con eficiencia, mantenimiento, reutilización y verificabilidad.

Contextos de uso

El contexto de uso dentro del campo de la experiencia de usuario (UX) corresponde a las condiciones particulares en las que un producto, bien sea tangible o intangible, será utilizado, atendiendo principalmente a los factores que influirán en su uso y en el grado de satisfacción de unos usuarios específicos.

Estos factores o variables que influyen en el uso pueden provenir del entorno (espacio, tiempo, temperatura, ruido, movimiento...), de consideraciones técnicas (conectividad, prestaciones, configuraciones...) o de la organización de los procesos de trabajo o del sistema interactivo.

Diseño centrado en el usuario

Es la estrategia más utilizada para el diseño de interacción. El proceso de diseño centrado en el usuario describe las fases a lo largo de un ciclo de vida de diseño y desarrollo, mientras se enfoca en obtener una comprensión profunda de quién usará el producto.

Diseño de la información

Es el diseño de cómo se presentará visualmente (gráficamente) la organización de la información que proporciona un producto. Está asociado al diseño de la navegación y a la arquitectura de la información, pues hace uso de las distintas categorías y facetas. En el diseño de la información deben quedar bien identificados aquellos objetos que desencadenan acciones (tareas), aquella que se utilizan para recorrer un espacio de información y aquellos espacios en los cuales se muestran resultados. En el diseño de la información se da una gran importancia a la utilización de etiquetas, así como a los íconos.

Diseño de la navegación

La navegación es una de las tareas fundamentales al utilizar un producto basado en hipertexto. La navegación, en un espacio de contenidos, es el resultado de la arquitectura de la información. El usuario debe "entender" la organización de los contenidos y la manera de encontrar/buscar lo que necesita. El diseño de la navegación incluye todos aquellos aspectos y elementos para que no se genere carga cognitiva y desorientación en el espacio que está navegando. Un ejemplo típico es el caso de aplicaciones de comercio electrónico de amplia gama de productos (ej. amazon.com), donde el usuario debe "encontrar" un producto y existe una lista importante de soluciones que responde a sus necesidades. Algunas de esas herramientas son los menús navegacionales, *breadcrumbs*, tabla de contenidos, mapa del sitio, categorías con sus respectivos filtros, etc.

Escenario

Un escenario es una "descripción narrativa informal" (Rosson y Carroll, 2002). Describe las actividades o tareas humanas en una historia que permite la exploración y discusión de contextos, necesidades y requisitos. No describe necesariamente el uso de *software* u otro soporte tecnológico utilizado para lograr un objetivo.

Experiencia de Usuario

Es un concepto subjetivo, pues trata de las sensaciones, percepciones y satisfacción de un usuario al utilizar un *software* o un producto cualquiera. Para diseñar la experiencia de usuario se requiere de un buen diseño de interacción y de interfaz.

Gestalt

Es una palabra alemana que no tiene traducción directa al castellano pero que viene a significar "Forma, Estructura y Totalidad" y hace referencia a una **corriente psicológica** surgida en Alemania a principios del siglo XX. La mente configura, a través de ciertas leyes, los elementos que llegan a ella a través de los canales sensoriales (la percepción) o de la memoria (pensamientos, inteligencia y resolución de problemas). La frase "El todo es mayor que la suma de las partes" se utiliza a menudo para explicar la teoría de la Gestalt.

Interfaz Háptica

La esencia de una interfaz háptica es facilitar los comandos de acción motriz del humano y facilitar la creación de imágenes táctiles. Tienen dos funciones principales: una basada en la posición y las fuerzas de la mano, muñeca, brazo, codo u hombro del usuario; y el segundo es importante para mostrar características y sus distribuciones espaciales y temporales en términos de variables de posición y fuerza (influencias que son las entradas a una computadora). Las interfaces hápticas pueden dividirse en dos grupos principales, desde el punto de vista de la sensación que producirán en el momento del contacto con la parte del cuerpo (las que producen estímulos) y las que producen estímulos táctiles. (Golledge, Rice y Jacobson, 2006).

Interacción humano-máquina / humano-computador

Es la disciplina que se encarga del estudio de los métodos, técnicas, principios y demás aspectos, de cómo deben diseñar productos y sistemas interactivos: cuáles serán las acciones que llevará a cabo un usuario para alcanzar su objetivo al utilizar el producto y de qué modo deberá hacerlo.

Interfaz humano-computador

Es el elemento mediador entre el producto y el usuario. En la interfaz se presentan todos aquellos elementos con los cuales interactuará directamente el usuario para accionar eventos y tareas en el producto.

Modelo mental

Un modelo mental es una teoría de cómo funciona un sistema, qué significan sus señales y cuáles serán los resultados de las diferentes acciones del usuario.

Patrón arquitectónico

Los patrones arquitectónicos se utilizan para representar soluciones a problemas para el diseño de arquitecturas de *software*. Incluyen una descripción de los elementos que constituyen la arquitectura, así como la relación y restricciones que existen entre ellos. Un patrón arquitectónico expresa un esquema de organización a un nivel de abstracción muy alto, pues no entra en los detalles de implementación, ni la forma en que se representan los subsistemas que pueden hacer parte del *software*.

Patrón de diseño

Un patrón de diseño es una solución comprobada a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características, entre las cuales está la efectividad al usarlo en problemas similares, debe ser reutilizable de forma que pueda ser aplicado a diferentes problemas de diseño en contextos distintos. Se utilizan en el desarrollo de *software*, al incluir soluciones probadas y generadas también a partir de estándares. Son de muy utilizados en el diseño de interacción e interfaces.

Personas

Las personas (arquetipos compuestos), son modelos de usuarios, pero no reales. Se basan en patrones de comportamiento, y con las personas se puede desarrollar una comprensión de los objetivos de los usuarios verdaderos, en un contexto de uso. A partir de ellos se puede idear y validar conceptos de diseño. (Cooper, Reimann, Cronin y Noessel, 2014)

Realidad aumentada

La realidad aumentada es el término con el que se conoce aquel conjunto de tecnologías que permiten que un usuario visualice parte del mundo real a través de un dispositivo tecnológico el cual "agrega" información gráfica y textual (información virtual) a la escena que se está visualizando (contexto real). De esta manera los elementos físicos tangibles se combinan con elementos virtuales, creando así una experiencia aumentada (respecto a la información que se obtiene) de la realidad y en tiempo real.

Tecnología Código QR

Un código QR (abreviatura de código de "respuesta rápida") es un tipo de código de barras que contiene una matriz de puntos. Se puede escanear usando un escáner QR o un *smartphone* con cámara incorporada. Una vez escaneado, el *software* del dispositivo convierte los puntos dentro del código en números o una cadena de caracteres. Como resultado de escanear un código QR con su teléfono puede abrir una URL en el navegador web de *smartphone*.

Todos los códigos QR tienen una forma cuadrada e incluyen tres contornos cuadrados en las esquinas inferior izquierda, superior izquierda y superior derecha. Estos contornos cuadrados definen la orientación del código. Los puntos dentro del código QR contienen información de formato y versión, así como el contenido en sí. Los códigos QR también incluyen un cierto nivel de corrección de errores, definido como L, M, Q o H. Una baja cantidad de corrección de errores (L) permite que el código QR contenga más contenido, mientras que una corrección de errores más alta (H) hace que el código sea más fácil de escanear.

Usabilidad

Es una característica de calidad de un producto o servicio. La usabilidad es una medida para determinar si los usuarios alcanzan sus objetivos durante la realización de una tarea de manera efectiva, eficaz, segura y satisfactoria, en un contexto de uso específico.

Wireframes

Son una representación gráfica de la distribución de elementos y objetos de interfaz y contenidos del que dispone el usuario al utilizar un producto o un servicio. Están atados al espacio de visualización (tomando en cuenta una tarea y un contexto específico). No presentan ninguna cualidad estética. Son de utilidad al momento de realizar el diseño de la información.

Enlaces de interés

Asociación de Interacción Persona-Computador (AIPO)

AIPO es una asociación profesional abierta cuyos objetivos son promover y difundir la Interacción Persona-Ordenador y servir de vínculo entre los científicos y profesionales que desarrollen actividades en este ámbito.

<http://aipo.es>

Interaction Design Foundation

Una organización dedicada al estudio del diseño de la interacción e interfaces, con una serie de recursos, libros y ejemplos.

<https://www.interaction-design.org>

Lista de verificación para evaluación heurística de usabilidad

Pierotti, D. (2014). Heuristic Evaluation. A System Checklist. Disponible en http://eitidaten.fh-pforzheim.de/daten/mitarbeiter/blankenbach/vorlesungen/GUI/Heuristic_Evaluation_Checklist_stcsig_org.pdf y en https://pajoooheshyar.sums.ac.ir/attachments/95-01-07-13594/4_5920067781323128838.pdf

No solo usabilidad.com

Una revista en línea con artículos sobre usabilidad, accesibilidad y distintos aspectos sobre el diseño de interacción, interfaces y arquitectura de información.

<http://www.nosolousabilidad.com/index.htm>

Patrones de diseño

Proporciona una serie de ejemplos de patrones de diseño de interfaces de todos los tipos (contenedores, informacionales, entrada de datos, navegación, retroalimentación, etc.) mostrando su uso en sitios web conocidos.

<http://ui-patterns.com/>

Simplemente pregunta: Integración de la accesibilidad en el diseño. De Henry Shawn (2007).

Libro en línea dedicado al tema de accesibilidad en el diseño centrado en el usuario. Incluye una serie de aspectos a considerar desde el análisis del problema hasta la evaluación.

<http://www.uiaccess.com/JustAsk/es/index.html>

WAI – Web Accessibility Initiative de la W3C

Portal dedicado a los fundamentos para alcanzar accesibilidad en sitios web.

<https://www.w3.org/WAI/fundamentals/accessibility-intro/>

Xerox Star User Interface

Videos donde se presenta la evolución del diseño de interfaces emprendido por los laboratorios de Xerox Park, hasta definir el concepto de manipulación directa en las interfaces (1982).

Parte (1) <https://www.youtube.com/watch?v=Cn4vC80Pv6Q>

Parte (2) <https://www.youtube.com/watch?v=ODZBL80JPqw>

Bibliografía

Abascal, J., Aedo, I., Cañas, J. Gea, M., Gil, A., Lorés, J., Martínez, A., Ortega, M., Valero, P., Vélez, M. (2001). *La interacción persona-ordenador*. Jesús Lorés Editor. AIPO, Asociación Interacción Persona Ordenador. España

ACM SIGCHI (1992). *Curricula for Human-Computer Interaction. ACM Special Interest Group on Computer -Human Interaction (SIGCHI)*. Curriculum Development Group.

ACM - Association for Computing Machinery, Inc. New York. Disponible en https://www.researchgate.net/publication/234823126_ACM_SIGCHI_curricula_for_human-computer_interaction

Alben, L (1996). Quality of experience: Defining the criteria for effective interaction design. *Interactions*, 3(3), pp. 11–15.

Alita, J. (2019). *Formative vs Summative Evaluations*. Nielsen/Norman Group. Disponible en <https://www.nngroup.com/articles/formative-vs-summative-evaluations/> Acceso Junio 2020.

Apple Computer Inc. (1987). *Knowledge Navigator*. Disponible en <https://youtu.be/umJslTGzXd0>

Arias C., M. (2007). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software". *Revista InterSedes* © Volumen VI, Número 10. Universidad de Costa Rica. Edición Digital: 26/07/2007.

Benyon, D. (2014). *Designing Interactive Systems: A comprehensive guide to HCI, UX and interaction design*. Third Edition. Pearson Education Limited.

Bevan, N., Carter, J., Earthy, J., Geis, T., Harker, S. (2016). New ISO Standards for Usability, Usability Reports and Usability Measures. M. Kurosu (Ed.): *Proceedings HCI 2016*, Part I, LNCS 9731, pp. 268–278. Disponible en https://www.researchgate.net/profile/Thomas_Geis/publication/304107978_New_ISO_Standards_for_Usability_Usability_Reports_and_Usability_Measures/links/5b3a1a0ba6fdcc8506e7bdbd/New-ISO-Standards-for-Usability-Usability-Reports-and-Usability-Measures.pdf

Bianchini, A. (2013). *WebFDM: Un método flexible para el desarrollo de aplicaciones Web*. PhD Thesis. Universidad Simón Bolívar, Caracas.

Bianchini, A. (2020). *Diseño de interfaces para software*. Manual del curso. Universidad Internacional de Valencia, España.

Bianchini, A., Ortega, M., Suárez, A. (2008). Captura y especificación de requerimientos en desarrollos web. *Proceedings InterTIC 2008*. Conferencia Iberoamericana IASK - International Association for the Scientific Knowledge". Madrid.

Bush, V. (1945) As we may think. *The Atlantic Monthly, July issue*. Disponible en <http://web.mit.edu/STS.035/www/PDFs/think.pdf>

Bushman, F., Meunier, R., Rohnert, H., Sommerlad, P., Satl, M. (2001). *Pattern-Oriented Software Architecture. A system of Patterns*. Volume 1. Wiley.

Cohen, P., and Oviatt, S. (1995). The Role of Voice Input for Human-Machine Communication. *Proceedings of the National Academy of Sciences* 92(22): pp. 9921–9927.

Cooper, A., Reimann, R., Cronin, D. (2007). *About Face 3: The Essentials of Interaction Design*. 3rd. Edition. Wiley Publishing, Inc.

Cooper, A., Reimann, R., Cronin, D., Noessel, C. (2014). *About Face 3: The Essentials of Interaction Design*. 4th. Edition. Wiley Publishing, Inc.

Cooper, A. (2018). *When Companies Question the Value of Design*. Disponible en <https://medium.com/s/story/whats-the-roi-of-ux-c47defb033d2>

Dauchot, N. (2019). *Formative vs. summative research. Quick and dirty versus slow and rigorous usability research*. Disponible en <https://uxdesign.cc/formative-vs-summative-usability-research-3dd2317b8b75> Acceso junio de 2020.

Dix, A., Finlay, J., Abowd, G., Beale, R. (2004). *Human-Computer Interaction*. Third Edition. Pearson Education Limited. Prentice Hall.

Engels, G., Lohmann, M., Wagner, A. (2006): The Web Application Development Process. *Web Engineering: The Discipline of Systematic Development of Web Applications*. Editado por Kappel, G., Proll, B., Reich, S., Retschitzegger, W. John Wiley & Sons Ltd.

Escalona, M.J., Koch, N. (2004): Requirements Engineering for Web Applications – A Comparative Study. *Journal of Web Engineering*, Vol. 2, No. 3 pp 193-212. Rinton Press.

EUR-Lex (1990). *Council Directive 90/270/EEC of 29 May 1990*. Disponible en español en <https://eur-lex.europa.eu/legal-content/ES/TXT/PDF/?uri=CELEX:31990L0270&from=EN>

García, F. J., Conde, M. Á., & Bravo, S. (2008). *Ingeniería del Software, Introducción a la Ingeniería de Requisitos*. Recuperado de <http://ocw.usal.es/ensenanzas-tecnicas/ingenieria-del-software/contenidos/Tema3-IntroduccionalIR-1pp.pdf>

Garrett, J. J. (2011). *The Elements of User Experience: User-Centered Design for the Web and Beyond*, Second Edition. New Riders.

Gea, M., Gutiérrez, F. (2002). El diseño. Capítulo 5 de *La interacción persona-ordenador*, Jesús Lorés Editor. AIPO, Asociación Interacción Persona Ordenador. Revisión año 2002.

Gottesdiener, E. (2005). *The Software Requirements Memory Jogger: A Pocket Guide to Help Software and Business Teams Develop and Manage Requirements*. First Edition GOAL/QPC. NH.

Grünbacher, P. (2006): Requirements Engineering for Web Applications. *Web Engineering: The discipline of Systematic Development of Web Applications*. Editado por Kappel, G., Proll, B., Reich, S., Retschitzegger, W. John Wiley & Sons Ltd.

Healthy Metrics Research, Inc (2020). Fitbit. Nuestra tecnología. Disponible en <https://www.fitbit.com/es/technology> Acceso abril de 2020.

Henry, S. (2008). *Just Ask: Integrating Accessibility Throughout Design*. Madison, WI: ET\Lawton. Libro en línea, disponible en <http://www.uiaccess.com/accessucd/>

Henry, S. (2008a). *Simplemente pregunta: Integración de la accesibilidad en el diseño*. Madison, WI: ET\Lawton. Libro en línea en español, disponible en <http://www.uiaccess.com/JustAsk/es/>

IBM (2000). The user experience. *The iceberg analogy of usability*. Developer Works. IBM Corporation. Disponible en <https://www.ibm.com/developerworks/library/w-berry/w-berry-pdf.pdf>

Interaction Design Foundation (2019). Context-Aware Computing. *The Encyclopedia of Human-Computer Interaction*, 2nd Ed. Disponible en <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/context-aware-computing-context-awareness-context-aware-user-interfaces-and-implicit-interaction>. Acceso abril de 2019.

Interaction Design Foundation (2019a). *Natural User Interfaces – What are they and how do you design user interfaces that feel natural?* Disponible en <https://www.interaction-design.org/literature/article/natural-user-interfaces-what-are-they-and-how-do-you-design-user-interfaces-that-feel-natural>. Acceso abril de 2019.

Interaction Design Foundation (2019b). *The Glossary of Human Computer Interaction*. Disponible en <https://www.interaction-design.org/literature/book/the-glossary-of-human-computer-interaction/context-of-use>. Acceso abril de 2019.

International Organization for Standardization (2019). *ISO 9241-210:2019(en) Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems*. Disponible en <https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-2:v1:en>

ISO - International Organization for Standardization (2019a). *ISO 9241-210:2010 - Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems*.

ISO – International Organization for Standardization (2019b). *ISO 9241-11:2018 - Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts*. Disponible en <https://www.iso.org/standard/63500.html>

Iso25000.com (2019). *Norma ISO-EIC 25010*. Disponible en <https://iso25000.com/index.php/normas-iso-25000/iso-25010>

IXDA Interaction Design Association (2020). *What is Interaction Design*. Disponible en <http://ixda.org/ixda-global/about-history/> Recuperado abril, 2020.

Kapor, M. (1996). A Software Design Manifesto. Chapter 1, in *Bringing Design to Software*. Winograd, T., Bennett, J., De Young, L., Hartfield, D. Editors. Stanford University and Interval Research Corporation. Disponible en <https://hci.stanford.edu/publications/bds/1-kapor.html>

Kay, A. (1989). *User Interface: A Personal View. VPRI Paper for Historical Context*. Viewpoints Research Institute. Disponible en http://www.vpri.org/pdf/hc_user_interface.pdf

Laurel, B., Mountford, S. J. (1990). *The Art of Human-Computer Interface Design*. Addison-Wesley Pub. Co. Reading, Mass.

Lemelson Center (2018). *The mother of all demos*. Lemelson Center for the Study of Invention and Innovation at Smithsonian's National Museum of American History. Disponible en <https://invention.si.edu/mother-all-demos>

Lewis, C. & Rieman, J. (2008). *Task-Centered User Interface Design*. Recuperado de <https://web.cs.dal.ca/~jamie/TCUID/readme.html>

Lieberman, H., Selker, T. (2003). *Agents for the User Interface*. In *Handbook of Agent Technology*, Jeffrey Bradshaw, ed., MIT Press. Disponible en http://web.media.mit.edu/~lieber/Publications/Agents_for_UI.pdf

The LOGO Foundation (2019). What is LOGO. MEDIA LAB – MIT. Disponible en https://el.media.mit.edu/logo-foundation/what_is_logo/index.html

López, V., Montero, F., Molina, J., González, P. (2006). *Interfaces de Usuario Inteligentes: Pasado, Presente y Futuro*. VII Congreso Internacional de Interacción Persona-Ordenador, Interacción 2006, Puertollano, España. Noviembre. Disponible en https://www.researchgate.net/publication/228801200_Interfaces_de_Usuario_Inteligentes_Pasado_Presente_y_Futuro

MADEJA (2019). *Patrón Modelo Vista Controlador. Marco de Desarrollo de Software de la Junta de Andalucía*. España. Recuperado de <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/122>

Martínez, A., Cuevas J. (2001). Estándares y guías de diseño. Capítulo 9 de *La interacción persona-ordenador*, Jesús Lorés Editor. AIPO, Asociación Interacción Persona Ordenador. España.

Massa, S., De Giusti, A. Pesado, P. (2012). Métodos de evaluación de usabilidad: una propuesta de aplicación en Objetos de Aprendizaje. *Proceedings WICC 2012 - XIV Workshop de Investigadores en Ciencias de la Computación*, Argentina, pp.922,926.

Netflix Inc. (2020). *Netflix. Acerca de la empresa*. Disponible en https://media.netflix.com/es_es/about-netflix Recuperado en mayo de 2020.

Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann.

Norman, D. (1988). *The Psychology of Everyday Things*. New York, Basic Books

Norman, D. (2013). *The Design of Everyday Things Revised and Expanded Edition*, Basic Books, Perseus Books Group. New York

Norman, K., Kirakowski, J. (2018). Introduction: Human-Computer Interaction Yesterday, Today, and Tomorrow. In *The Wiley Handbook of Human Computer Interaction*, Volume 1, First Edition. Wiley.

Olsina, L. (1999). *Metodología Cuantitativa para la Evaluación y Comparación de la Calidad de Sitios Web.* Tesis de Doctorado. Facultad de Ciencias Exactas Universidad Nacional de La Plata – Argentina.

Paganelli, L., Paternò, F. (2002). Intelligent analysis of user interactions with web applications. *Proceedings International Conference on Intelligent User Interfaces.* ACM Press 111-118.

Pernice, K. Nielsen, J. (2009). *How to conduct Eyetracking Studies.* Nielsen Norman Group Publications. CA. Disponible en https://media.nngroup.com/media/reports/free/How_to_Conduct_Eyetracking_Studies.pdf

Pierotti, D. (2014). *Heuristic Evaluation. A System Checklist.* Disponible en http://eitidaten.fh-pforzheim.de/daten/mitarbeiter/blankenbach/vorlesungen/GUI/Heuristic_Evaluation_Checklist_stcsig_org.pdf

Preece, J., Rogers, Y., Sharp, H. (2002). *Interaction design – beyond human-computer interaction.* First Edition. John Wiley & Sons, Inc.

Preece, J., Rogers, Y., Sharp, H. (2019). *Interaction design – beyond human-computer interaction.* Fifth Edition. John Wiley & Sons, Inc.

Pressman, R. (2010). *Ingeniería de Software. Un enfoque práctico.* 7^{ma}. Edición. McGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V. México.

RAE – Real Academia Española. *Diccionario del español jurídico* – Definición de accesibilidad. Disponible en <https://dej.rae.es/lema/accesibilidad>

Rodríguez, C. T. (2017). Impacto de los requerimientos en la calidad de software. *TIA Tecnología, Investigación y Academia*, 5(2), pp. 161-173. Recuperado de <https://revistas.udistrital.edu.co/ojs/index.php/tia/article/download/7607/pdf/>

Rosson, M., Carroll, J. (2002). Scenario-Based Design. Chapter 53 in J. Jacko & A. Sears (Eds.), *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications.* Lawrence Erlbaum Associates, pp. 1032-1050.

Ryan, M. (2014). The role of Stakeholders in Requirements Elicitation. *Proceedings of INCOSE International Symposium 2014.* INCOSE – International Council on Systems Engineering. Disponible en https://www.researchgate.net/publication/268821136_The_Role_of_Stakeholders_in_Requirements_Elicitation

Scihi.org (2018). *Ted Nelson and the Xanadu Hypertext System.* June 2018. Disponible en <http://scihi.org/ted-nelson-xanadu-hypertext-system/>

Scihi.org (2019). *J.C.R. Licklider and Interactive Computing.* March 2019. Disponible en <http://scihi.org/licklider-interactive-computing-arpanet/>

Scott, B. Neil, T. (2009). *Designing Web Interfaces: Principles and Patterns for Rich Interactions.* O'Reilly Media Inc: Newton.

Shneiderman, B. (1983). Direct Manipulation: A Step Beyond Programming. *IEEE Computer Journal.* Volume 16, Issue 8, Pages 57-69. August 1983. Disponible en: <http://iihm.imag.fr/blanch/ens/2016-2017/M1/TLI/readings/1983-Shneiderman-DirectManipulation.pdf>

Sommerville, I., Sawyer, P. (2003). *Requirements Engineering: A Good Practice Guide*. John Wiley & Sons, Reprint 2003.

Sommerville, I. (2005). *Ingeniería de Software*. Pearson- Addison Wesley – 7^{ma}. edición.

Sutherland, I. (1963). *Sketchpad: A man-machine graphical communication system*. PhD Thesis at MIT. Disponible en <https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-574.pdf>

Tidwell, J. (2010). *Designing Interfaces. Patterns for Effective Interaction Design*. 2th. Edition. O'Reilly.

Timblebay, H. (2018). Interactive Critical Systems and How to Build Them. *The Wiley Handbook of Human Computer Interaction*, Volume 1, First Edition. Edited by Kent L. Norman and Jurek Kirakowski. Wiley.

W3C – World Wide Web Consortium (1997). *World Wide Web Consortium Launches International Program Office for Web Accessibility Initiative*. Press release Government, Industry, Research and Disability Organizations Join Forces to Promote Accessibility of the Web. Disponible en <https://www.w3.org/Press/IPO-announce>

W3C – World Wide Web Consortium (2019). Introduction to Web Accessibility. Disponible en <https://www.w3.org/WAI/fundamentals/accessibility-intro/>

Weiser, M. (1991). The Computer in the 21st Century". *Scientific America*, September 1991. pp. 94-104. Disponible en <https://www.lri.fr/~mbl/Stanford/CS477/papers/Weiser-SciAm.pdf>

Weiser, M. (1994). The world is not a desktop. *ACM Interactions*. Volume 1, Issue 1, January 1994. pp. 7-8. Disponible en <https://www.dgsiegel.net/files/refs/Weiser%20-%20The%20World%20is%20not%20a%20Desktop.pdf>

Westfall, L. (2006). *Software Requirements Engineering: What, Why, Who, When, and How*. The Westfall Team – Partnering for Software Excellence. Recuperado de http://www.westfallteam.com/Papers/The_Why_What_Who_When_and_How_Of_Software_Requirements.pdf

Wiegers, K., Beatty, K. (2013). *Software Requirements*, Third Edition. Published by Microsoft Press. A Division of Microsoft Corporation. Redmond, Washington.

Whitenton, K. (2018). *The Two UX Gulfs: Evaluation and Execution*. NN/g Nielsem. Norman Group. Disponible en <https://www.nngroup.com/articles/two-ux-gulfs-evaluation-execution/>

World Wide Web Foundation (2019). *History of the Web*. Disponible en <https://webfoundation.org/about/vision/history-of-the-web/>

Yáñez, R., Cascado, D., Sevillano, J. (2014). Heuristic Evaluation on Mobile Interfaces: A New Checklist. *The Scientific World Journal*, Volume 2014, Article ID 434326. Hindawi Publishing Corporation

Agradecimientos

Autora

Dra. Adelaide Bianchini



**viu
.es**