

Unidad 1



Introducción al diseño de interfaces y aplicaciones para el diseño

Desarrollo de interfaces



Índice



1.1. Interacción persona-ordenador

1.2. Diseño de una interfaz. El diseño gráfico

1.3. Herramientas

- 13.1. Prototipos. Elementos clave de un prototipo
- 13.2. Aplicaciones para el desarrollo de interfaces

1.4. Planteamiento y diseño de una interfaz

- 14.1. Área de redacción
- 14.2. Área de producción
- 14.3. Área técnica
- 14.4. Área artística

1.5. El color

- 15.1. Sistema RGB
- 15.2. Matiz, saturación y brillo



Introducción

En este tema desarrollaremos la interfaz, el elemento que hace posible la comunicación entre personas y ordenadores. Veremos de donde surge esta idea de la comunicación y que teorías se siguieron para idearla.

Veremos el empleo de los prototipos y maquetas utilizadas para la creación de interfaces, con el fin de facilitar y reducir el tiempo empleado en su producción, así como los elementos que las interfaces modernas incluyen en su construcción.

Debemos tener en cuenta que este proceso suele conllevar un trabajo colaborativo con distintos agentes, por lo que veremos los diferentes agentes que participan en el proceso, tanto humanos como no.

Finalmente trataremos temas que pertenecen al ámbito gráfico como es el color y su clasificación en el sistema RGB.

Al finalizar esta unidad

- + Conoceremos los principios de la comunicación persona-ordenador.
- + Estudiaremos los elementos necesarios para la composición de interfaces.
- + Describiremos los distintos métodos para la elaboración de una interfaz.
- + Definiremos los elementos que necesitamos tener en cuenta para desarrollar una interfaz adaptada a nuestras necesidades.
- + Distinguiremos los diferentes elementos gráficos que nos podremos encontrar.



1.1.

Interacción persona-ordenador

La disciplina de la interacción persona-ordenador, IPO o HCI, estudia el intercambio de información producido entre ambos mediante sistemas interactivos, con el fin de mejorar este proceso.

Debemos entender que este tipo de relación no es natural, sino que ha sido desarrollada íntegramente por el ser humano, por lo que a pesar de que se ha intentado adaptar en la medida de lo posible a nuestras necesidades, no ha pasado por un extenso proceso de evolución que habría refinado el proceso correctamente, eliminando o subsanando los elementos discordantes.

Entre los muchos elementos problemáticos que se debían solucionar para poder crear una interacción persona-ordenador destacan los que fueron señalados en 1962 por Licklider y Clark como los 10 más problemáticos.

- > Uso simultáneo de un solo ordenador por diversos usuarios.
- > Un sistema de entrada y salida de comunicación con símbolos y gráficos.
- > Un sistema interactivo que funcione en tiempo real.
- > Empleo de memorias de alta capacidad y una rápida recuperación de la información.
- > Programación de grandes sistemas que además faciliten la cooperación.
- > Reconocimiento de diversos sistemas de comunicación como voz, texto y texto manual.
- > Comprensión del lenguaje natural, tanto en el ámbito sintáctico como en el semántico.
- > Reconocimiento de distintas voces por parte del ordenador.
- > Una teoría de algoritmos sencilla y válida.
- > Una programación general o heurística.

Al ser una disciplina nueva se tuvieron que establecer reglas que permitieran una homogeneización, como fueron los principios que definían el diseño de sistemas interactivos publicada por Hansen en 1971.

- > Descubrir las características y necesidades del usuario.
- > Introducir elementos claros e intuitivos que eviten la necesidad de estudio previo o memorización.
- > Optimizar la interfaz mediante una estructura robusta que permita el mejor uso, podemos extraer la información de la forma óptima a partir del uso del propio sistema.
- > Reducir al mínimo los fallos y crear contramedidas y anuncios para los fallos más comunes no solucionados, de modo que ante un problema se pueda descubrir una solución o, en su lugar, conocer el componente que produce el mal comportamiento del sistema.



1.2.

Diseño de una interfaz. El diseño gráfico

El diseño gráfico permite el empleo de distintos elementos de trabajo para la creación de comunicaciones visuales para aplicaciones. Dentro de las funciones de los diseñadores gráficos podemos encontrar algunas como:

- > La función estética.
- > La función publicitaria.
- > La función comunicativa.

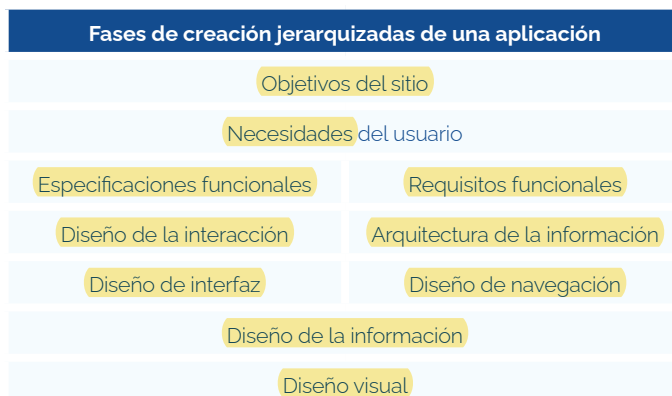
Existen cuatro tipos de elementos que los diseñadores gráficos pueden emplear en la construcción de una interfaz, estos son:

- > Conceptuales.
- > Visuales.
- > Relación.
- > Prácticos.

El empleo de los elementos necesarios de los nombrados anteriormente, junto con la búsqueda de la funcionalidad óptima, la inclusión de las funciones ya nombradas permitirá la creación de una interfaz de buena calidad que permita al usuario navegar por la aplicación con facilidad y de manera agradable.

Un problema con la funcionalidad de la página puede llevar a que esta no funcione correctamente, por otro lado, una mala estética puede llevar a que la página sea indeseable de hacer o, incluso, inoperable si, por ejemplo, la mala elección de colores dificulta su visualización o la mala jerarquización impide encontrar el elemento deseado.

Para una buena construcción de una aplicación es necesario que se deben pasar por diferentes fases hasta llegar a una construcción completa y correcta de la interfaz deseada, esas fases son:





Como podemos ver el primer paso será la búsqueda de los objetivos de la aplicación, para ellos deberemos buscar tanto los elementos que formarán parte de esos objetivos como las acciones que realizarán, hecho esto, y siguiendo un esquema similar a la orientación de objetos, separaremos estos objetos y acciones. Dentro de los objetos es probable que encontremos:

- > **Objetos origen.** Pueden ser movidos, siendo depositados en los objetos destino.
- > **Objetos destino.** reciben a los elementos origen.
- > **Objetos de la aplicación.** Elementos que conforman la aplicación y con los que los usuarios no pueden interactuar.

Una vez conformada la aplicación deberemos llevar a cabo una elección sobre el diseño de interfaces que se lleve a cabo.

El diseño será fundamental para el posterior uso de la aplicación, debemos seleccionar la interfaz idónea para cada aplicación, ya que cada aplicación, en función de sus características, requerirá unas necesidades específicas, desde un rápido acceso a todos los elementos si los objetos de la aplicación son muy pocos, hasta una ordenación y jerarquización concienzuda si los elementos que la componen son numerosos. Igualmente, en función de los destinatarios de la aplicación, es posible que debamos mantener una dificultad aún menor de uso marcando claramente cada selección si los destinatarios no poseen conocimientos informáticos.

Se deben realizar ciertas pruebas para comprobar que dicho diseño cumple con los requisitos funcionales adecuados, algunas de estas pruebas son:

- > **Tiempo de respuesta.** Algunos elementos de la aplicación requieren que el usuario reaccione y complete una acción en un tiempo especificado, deberemos calcular el tiempo óptimo para ello, de modo que sea suficiente para que se reaccione con comodidad, pero no lo suficiente para que resulte pesada la espera.
- > **Servicio de ayuda.** El uso de una interfaz puede ser complicada si no se tienen los conocimientos necesarios, con el fin de evitar problemas la inclusión de elementos de ayuda, como pueden ser, explicaciones emergentes, guías integradas, manuales de uso, etc., asisten al usuario para que lleve a cabo su labor de manera sencilla y correcta.
- > **Etiquetado.** Una buena nomenclatura en los botones y opciones, así como unos claros de los elementos visuales, conllevarán un diseño que permitirá una mayor facilidad de empleo y memorización de las funciones.



1.3.

Herramientas

La elaboración de una interfaz desde cero es una labor ardua e ineficiente, por lo que, con el fin de aliviar esta carga, el uso de elementos como maquetas o prototipos que sirvan de modelo sobre el que construir. Para llevar esto a cabo es posible el empleo de diversas herramientas, tanto libres como comerciales, que nos ayuden a la construcción de cuadros de diálogos, ventanas, menús.

Es necesario conocer las necesidades de nuestra interfaz para poder elaborar un diseño acorde a ellas, y que por tanto nos sea válido. Para localizar estas necesidades podemos emplear distintos elementos y seguir estos pasos:

- > **Empleo de formularios** a usuarios de carácter cuantitativo y cualitativo, con estos conoceremos la satisfacción y uso de nuestra interfaz.
- > **Recopilación de los datos relevantes** y procesarlos para descubrir las carencias de nuestra interfaz.
- > **Revisar y modificar** la interfaz en función de las necesidades. Repetir el proceso de forma cíclica periódicamente.

1.3.1. Prototipos.

Elementos clave de un prototipo

Un prototipo es un modelo general prefabricado que nos permite una visión clara del resultado final de una interfaz construida de ese modo, es de gran utilidad a la hora de tomar la decisión sobre que diseño emplear ya que podemos ver diferentes muestras de los posibles productos finales.

El empleo de los prototipos produce ciertas ventajas como son:

- > **Aumento en la velocidad de desarrollo.** El empleo de un prototipo nos dará una gran parte del trabajo de programación ya hecho, facilitando en gran medida la tarea, ya que en muchos casos tan solo deberemos escribir sobre la plantilla, ya que esta se encuentra prefabricada con botones, menús, estilos, etc.
- > **Interacción con el cliente.** El prototipo nos permitirá conocer las opiniones del cliente, así como la inclusión de los cambios que este desee en una etapa temprana del proyecto, lo cual facilitará en gran medida su inclusión exitosa y sin esfuerzo.



Existen diferentes tipos de elementos modelo como el prototipo, entre ellos destacan los siguientes tres:

- > **Sketching.** Es el establecimiento en papel de un diseño básico de la aplicación con el fin de establecer la jerarquía de contenidos, en esta fase inicial aún no se contempla el diseño de la interfaz.
- > **Wireframing.** El dibujo, tanto en papel como ordenador, de los elementos del diseño, como pantallas, menús, botones, etc., es un primer vistazo a como debe de quedar la página, es la fase previa al diseño como tal, por lo que se suele incluir al cliente en busca de su opinión.
- > **Prototipo.** El prototipo es ya el modelo general sobre el que se trabajará, pasando este de ser una entidad general o especificarse como una plantilla y finalmente en la interfaz deseada.

Existen tres clases de prototipos:

- > **Horizontal.** Modelado de la funcionalidad, pero no de la información relevante.
- > **Vertical.** Modelado con poca funcionalidad, pero mucha información.
- > **Diagonal.** La suma de los dos anteriores, empleando las dos clases anteriores a partir de una primera elaboración con la metodología horizontal para después sobre esta continuar trabajando con una metodología vertical.

1.3.2. Aplicaciones para el desarrollo de interfaces

El prototipo siempre debe mantener los siguientes aspectos en él:

- > Identificar cada uno de los elementos de una pantalla.
- > Clasificar los elementos con una buena jerarquización.
- > Distribuir los elementos con el fin de que no sean agobiantes ni acaben demasiado dispersos.
- > Expansión adecuada para ocupar todo el espacio correctamente.
- > Empleo de patrones como estandarización.
- > Incluir todos los aspectos necesarios a nivel técnico de usabilidad y accesibilidad.

Una de las mayores ventajas del empleo de herramientas para la elaboración de los elementos visuales es la posesión de diferentes elementos empaquetados los cuales podemos reutilizar en diversas ocasiones, permitiéndonos el empleo de un código reutilizado, lo cual nos permite una mayor velocidad y la homogeneización en nuestra interfaz a partir de la estandarización.

Existen numerosas herramientas, entre ellas destacan algunas como las siguientes:

- > **Visual Studio.** Actualmente es el IDE más empleado, posee la capacidad de aceptar diversos lenguajes multiplataforma y cuenta con funciones de autocompletado de código.

- > **Monodevelop.** De carácter libre y gratuito, cuenta con elementos de procesadores de texto y gestión de proyectos.
- > **Glade.** Es empleado principalmente para trabajar con elementos XML.
- > **NetBeans.** Es de código abierto, posee una gran capacidad para extenderse a otros módulos.
- > **Eclipse.** De código abierto y multiplataforma, otorga una gran libertad de creación a nivel visual, permitiéndonos ver en tiempo real el trabajo que realizamos.

Herramientas y lenguajes soportados		
Nombre	Licencia	Lenguaje soportado
Visual Studio	Propietaria (versión Community libre)	C#, F#, Razor, HTML5, CSS, JavaScript, Typescript, XAML, XML
Monodevelop	Libre	C#, Java, .Net, Python
Glade	Libre	C++, C#, Java, Python
NetBeans	Libre	Java, HTML, PHP, Python
Eclipse	Libre	Java, C++, PHP



1.4.

Planteamiento y diseño de una interfaz

Existen diversos elementos con los que debemos tener especial cuidado como son:

- > El número de elementos que formarán parte de nuestra aplicación, ya que no siempre más opciones se traducen en un mejor resultado.
- > La extensión utilizada por la aplicación.
- > La estandarización del diseño mediante el empleo de patrones.
- > Aspectos técnicos necesarios para su usabilidad y accesibilidad.

La creación de una interfaz debe pasar por el siguiente proceso para llegar al prototipo, sino se toma este directamente de un repositorio:

- > Definir el objetivo que buscamos.
- > Localizar las necesidades para una elaboración correcta y los requisitos que se deben cumplir.
- > Realizar un *Wireframing* con el primer esbozo de la idea.
- > Creación de un prototipo para su posterior implantación.

1.4.1. Área de redacción

En este espacio se deben recopilar los conceptos fundamentales para la creación de nuestra interfaz, los cuales se considerarán los aspectos principales de nuestra aplicación.

1.4.3. Área técnica

Área donde los técnicos comienzan a valorar las necesidades y los requisitos del proyecto. Ya trabajan sobre formato digital, creando prototipos y maquetas sobre las que posteriormente se trabajará.

1.4.2. Área de producción

Es necesario conocer las tendencias y necesidades actuales para poder elaborar una interfaz eficaz y atractiva.

Debemos tener en cuenta también que este tipo de proyecto no es individual, sino que incorpora diferentes agentes como son:

- > **Cliente.** Es el que inicia el proyecto con su encargo, también poseerá el último voto en su rumbo.
- > **Usuario.** El público objetivo de la aplicación.
- > **Presupuesto.** Importe económico aportado por el cliente.
- > **Plan de trabajo.** Programación del desarrollo del trabajo, desde el calendario a la distribución de las tareas a realizar.

1.4.4. Área artística

El estilo dado a la interfaz no se añadirá hasta su creación final, con todas las funcionalidades incluidas. Este aspecto dependerá tanto de los principios de la propia página, la imagen que desee proyectar; como la que los usuarios requieran, la cual haga cómodo y agradable trabajar con ella.

Por estas razones esta etapa del proyecto es importante, incluyendo en ocasiones otros agentes como los equipos de *marketing* para su creación.



1.5.

El color

La combinación de colores empleados en la interfaz será importante por múltiples razones, pero primero debemos tener en cuenta que el empleo de colores debe realizarse teniendo en cuenta las posibilidades del ojo humano.

Con el fin de solventar el problema de la capacidad del ojo humano, solo se emplean los llamados colores aditivos, los cuales sí puede percibir. Estos colores son el azul, B; el rojo, R; y el verde, G; los cuales conforman el llamado sistema de color RGB.

1.5.1. Sistema RGB

El sistema de colores aditivos RGB permite a un ordenador, mediante la combinación de estos tres, crear una paleta de color que incluye todos los tonos conocidos.

La escala de cada color se representa con 256 tonos (2^8), desde claro a intenso, del 0 al 255, la combinación de un valor de cada uno de los tres colores nos dará un color único, contando con un total de 16 777 216 combinaciones y por tanto el mismo número de tonos de color.

Los colores se almacenan en 8 bits binarios por números que se agrupan en bloques de 4 bits el cual nos devolverá un valor hexadecimal con el que se nombrará cada color.

Por **ejemplo**, el púrpura poseerá los siguientes valores:

Púrpura			
128	1000 0000	80	#800080
0	0000 0000	00	
128	1000 0000	80	
Sist. RGB	Binario de 8 bits	Hexadecimal	Código Hexadecimal completo



1.5.2. Matiz, saturación y brillo

Junto con la combinación de colores se incluyen tres propiedades a las escalas de cada color, estas son:

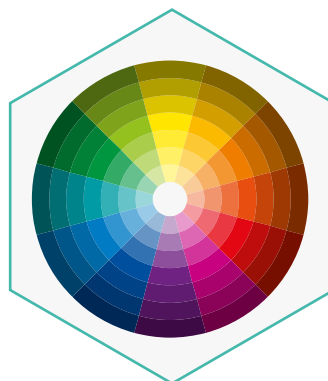
- > **Matiz.** Es el atributo que nos permite distinguir un color de otro, tomando los colores aditivos como los primarios, si un color se encuentra frente a otro en el círculo cromático se les considera complementarios.
- > **Saturación.** Define la intensidad de un color con relación al nivel de gris que contiene un color, por lo que a más gris menos saturación. Los colores saturados son los considerados intensos, y los no saturados los claros.
- > **Brillo.** Atributo que representa lo oscuro o claro de un color creando un degradado en función a su luminosidad.



Círculo
cromático



Ejemplo
de saturación



Ejemplo de
escala de brillo



 www.universae.com

