

UNIDAD DIDÁCTICA 4

USABILIDAD

**MÓDULO PROFESIONAL:
DESARROLLO DE INTERFACES**



CESUR
Tu Centro Oficial de FP

ÍNDICE

RESUMEN INTRODUCTORIO	3
INTRODUCCIÓN	3
CASO INTRODUCTORIO	3
1. USABILIDAD	5
1.1 Características y atributos	5
1.2 HCI (Human Computer Interaction).....	8
1.3 La experiencia de usuario UX (User Experience)	10
1.4 Normas ISO	11
1.5 Medida de usabilidad de aplicaciones y tipos de métricas	13
1.5.1 Evaluación heurística.....	13
1.5.2 Test de usabilidad	14
1.6 Pruebas de expertos y pruebas de usuarios	16
1.7 Herramientas para medir la usabilidad	17
2. ACCESIBILIDAD.....	22
2.1 Principios básicos de la accesibilidad.....	22
2.2 Pautas de accesibilidad	23
2.3 Estándares y normas internacionales	24
2.4 Medidas de accesibilidad	25
2.4.1 Evaluación heurística de accesibilidad	25
2.4.2 Test de accesibilidad	26
2.5 Herramientas para medir la accesibilidad	26
3. ESQUEMAS Y MAQUETAS.....	31
3.1 Esquemas (Wireframes).....	31
3.2 Maquetas (Mockups)	32
3.3 Herramientas para la realización de esquemas (wireframes) y maquetas (mockups)	33
4. PAUTAS DE DISEÑO	38
4.1 Principios básicos	39
4.2 Pautas de diseño de la estructura de la interface	41
4.2.1 Proporción en los diseños	42
4.2.2 Ejemplo de interfaz desktop	43
4.2.3 Diseño de aplicaciones móviles	45
4.3 Pautas de diseño del aspecto de la interface de usuario	49
4.3.1 Colores.....	49
4.3.2 Fuentes.....	52
4.3.3 Iconos	52
4.3.4 Distribución de los elementos.....	53
4.4 Pautas de diseño del aspecto en los elementos interactivos.....	54
4.5 Pautas de diseño en la presentación de datos	57

4.6 Pautas de diseño en el control de la aplicación.....	58
4.7 Pautas de diseño en los atajos de teclado.....	60
4.8 Pautas de diseño en el aseguramiento de la información	61
4.9 Pautas de diseño en aplicaciones multimedia.....	61
RESUMEN FINAL	63

RESUMEN INTRODUCTORIO

En esta unidad nos introducimos en uno de los puntos más importantes en la actualidad en el diseño de interfaces: la usabilidad y las pautas de diseño relacionadas con la usabilidad.

Comenzaremos definiendo todos los conceptos base, los principios generalistas y las corrientes y estudios que hoy en día se encuentran en un momento muy importante como es el HCI, la experiencia de usuario o la experiencia de cliente, entre otros.

Por otra parte, continuaremos con el estudio de los principios de diseño en diferentes apartados, desde la interfaz de una aplicación hasta su contenido, sus elementos, su interacción con el usuario o la navegabilidad. Tendremos en cuenta que hoy en día es muy importante distinguir entre sistemas desktop y sistemas móviles.

También aprenderemos sobre los diferentes test y procesos para la evaluación de nuestros diseños en diferentes momentos de nuestro proyecto, de forma que tengamos retroalimentación sobre los mismos.

INTRODUCCIÓN

Podríamos decir que uno de los campos que más ha crecido actualmente dentro del desarrollo software es el de la usabilidad y la experiencia de usuario.

El cambio más importante que se ha producido es poner al usuario como el centro de todo nuestro paradigma. Para ello, es necesario conocer los fundamentos de la usabilidad, tanto a nivel de diseño como a nivel de medición.

Las técnicas son muchas y los principios dependerán de muchos factores como el del tipo del soporte, el del tipo de usuario o el tipo de aplicación. Pero conocer diferentes partes en el diseño de una interfaz es imprescindible para producir una aplicación eficiente y útil para el usuario.

CASO INTRODUCTORIO

Acabas de ser contratado por una empresa de desarrollo de software que acaba de ganar un proyecto para el desarrollo de una aplicación para clubes de baloncesto.

Es importante, durante la definición de las especificaciones con el cliente, tener un buen diseño o mockup de la aplicación que vamos a desarrollar, ya que de esta forma

podremos acotar los tiempos y, además, el cliente podrá tener presente una vista previa de la aplicación.

Tienes en cuenta que es importante diseñar la aplicación siguiendo criterios de accesibilidad para que cualquier persona pueda interactuar con ella independientemente de sus condiciones psicomotoras.

Al final de esta unidad tendrás los conocimientos para poder desarrollar el mockup de acuerdo a unos principios, así como los test necesarios para probar a diferentes niveles.

1. USABILIDAD

El primer paso después de una primera visita al cliente es la planificación del proyecto y, para ello, se planifican los recursos, los hitos y los tiempos de desarrollo del proyecto.

Durante esta planificación se deben planificar los test que se quieren realizar, a qué nivel y con qué personas. Sabes que la aplicación tendrá que ser usada por personas sin conocimiento técnico, por lo que tiene que ser lo más intuitiva posible y, para ello, te basarás en las características de usabilidad que marcan las normas. Para comprobar que todo sale según lo esperado evaluarás la aplicación en base a test y evaluación heurística para garantizar su usabilidad.

Las interfaces gráficas de usuario son responsables de la interacción que hay entre los hombres y las máquinas. Una parte importante es la confección de estas interfaces y las herramientas que se usarán como, por ejemplo, sus elementos como los componentes o los controles. Hace falta también tener en cuenta algunos aspectos concretos de esta interacción cuando se hace el diseño de las interfaces.

En primer lugar, habrá que establecer una definición del término usabilidad. Esta palabra deriva del término inglés usability, es decir, indica si una cosa es usable o tiene un uso sencillo. La usabilidad define el grado de facilidad con que un usuario puede usar una herramienta concreta o cualquier otro objeto.

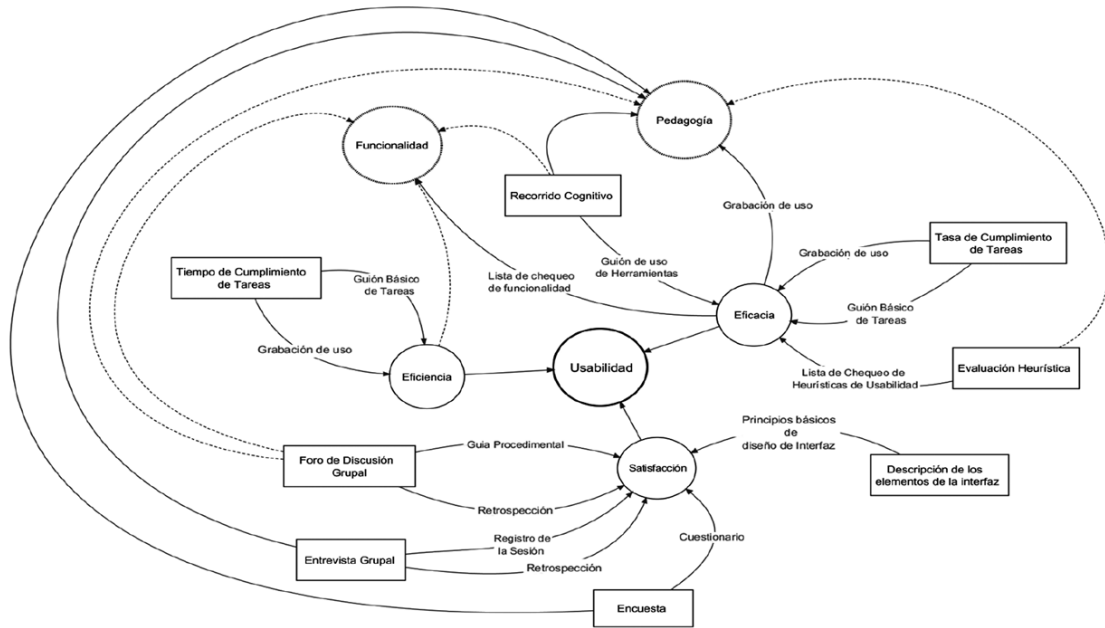
No obstante, hay muchas maneras de entender este concepto, puesto que se deben tener en cuenta muchos matices para poder definir o entender el significado de facilidad de uso. Además, también habrá que hablar de otros conceptos para definir la usabilidad, temas como efectividad, eficiencia y satisfacción del usuario. Y todos estos conceptos variarán la apreciación en función de si están afectados por los usuarios, sus objetivos o sus expectativas y la situación de uso de la herramienta o aplicación.

1.1 Características y atributos

La usabilidad es una característica que está relacionada con la medida de la calidad de los sistemas interactivos usados por usuarios específicos en un contexto de uso determinado para conseguir objetivos con las siguientes características:

- **Eficiencia:** se puede entender la eficiencia en el uso de las aplicaciones informáticas, es decir, una aplicación que obtiene unos buenos resultados en las medidas de usabilidad aportará más productividad al usuario y permitirá un uso más rápido, más eficiente de la aplicación. Se tratará de poder llevar a cabo más acciones e interacciones con el software en menos tiempo.

- **Efectividad:** precisión y completitud con la que el usuario alcanza los objetivos marcados. Relacionados con la efectividad están aspectos como la facilidad de aprendizaje, la tasa de errores cometidos y la facilidad de recordar sus funcionalidades y procedimientos.
- **Satisfacción:** ausencia de incomodidad y una actitud positiva hacia la utilización de la aplicación. La aplicación debe ser agradable de utilizar de forma que el usuario esté satisfecho al usarla.
- **Utilidad:** capacidad de la interfaz para ayudar en la realización de tareas. Facilidad de uso: si una aplicación es fácil de usar se realizarán las tareas más rápidamente.
- **Atractivo:** el significado del término atractivo está muy relacionado con otros ámbitos, pero ahora hay que relacionar esta característica con la usabilidad y las interacciones entre las personas y las máquinas. Una interfaz es atractiva para un usuario cuando este acepta de buen grado las características y el uso, mostrando una predisposición para utilizarla.
- **Facilidad de aprendizaje:** tiempo empleado para usar la aplicación con cierta eficiencia y llegar a recordar los procedimientos después de no usar la aplicación durante un tiempo determinado.
- **Apreciación:** actitud generada en el usuario por el uso de la aplicación.
- **Tolerancia al error:** esta característica, como su nombre indica, mide el grado en que la interfaz evita los errores o ayuda a superarlos.



Elementos de usabilidad, métodos y sus relaciones.

Fuente: https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-33052016000300008

El éxito de un programa depende directamente de la calidad de su interfaz gráfica. Al usuario final no le importa el tipo de tecnología que se utiliza detrás de la interfaz, no entiende de protocolos, paquetes o enrutamiento, así que solo juzga lo que ve y cómo se comporta frente a sus necesidades.

Sin embargo, y dadas las conclusiones de estas investigaciones, las empresas siguen sin invertir dinero y tiempo para la evaluación y mejoramiento de sus interfaces, aun sabiendo las altas tasas de retorno de dicha inversión.

A esta situación se le atribuyen diferentes causas, entre ellas: el simple desconocimiento de técnicas o herramientas para realizar esta labor, pero, sobre todo, como consecuencia de una cultura de desarrollo desfasada centrada en la tecnología y no en el usuario y sus tareas.



VÍDEO DE INTERÉS

Amplia tus conocimientos sobre usabilidad en el entorno de las interfaces de usuario:

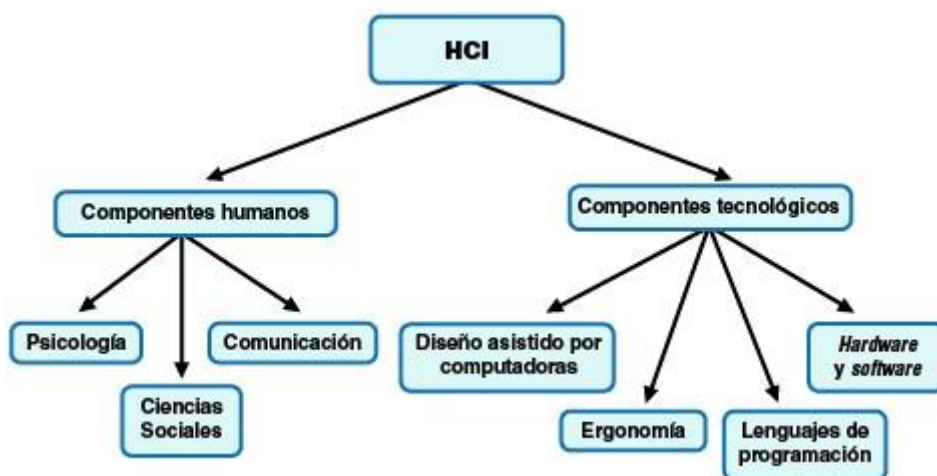


1.2 HCI (Human Computer Interaction)

La Interacción Persona-Ordenador es la disciplina que estudia el intercambio de información entre las personas y los ordenadores. Esta disciplina se centra en el estudio del diseño, evaluación e implementación de sistemas interactivos para uso humano.

La Interacción Persona-Computador es más conocida por su nombre en inglés, Human-Computer Interaction y sus siglas HCI. Y una de las asociaciones que acuña este término es la Association for Computing Machinery.

Mientras que, en otras disciplinas puramente técnicas, como el desarrollo del software o la ingeniería, aspectos como el hardware y el software son el centro de las investigaciones; en esta, donde el usuario aparece como nuevo elemento, las ciencias sociales y humanas son muy importantes. Elementos como la psicología cognitiva, el estudio del comportamiento, modelos mentales, procesos del aprendizaje, entre otros, están incluidos.



Disciplinas que intervienen en el HCI.

Fuente: https://libros.uvq.edu.ar/spm/521_hci_humancomputer_interaction.html

Licklider y Clark (1962) elaboraron una lista de 10 problemas que deberían ser resueltos para facilitar la interacción personas-ordenador. Según ellos, los cinco primeros problemas deberían ser resueltos de manera inmediata, el sexto en un tiempo intermedio y los cuatro últimos, a largo plazo:

1. Múltiples usuarios compartiendo los ordenadores en un mismo tiempo.
2. Comunicación mediante datos que son simbólicos y gráficos para el sistema de entrada y salida.
3. Un sistema interactivo de proceso de las operaciones en tiempo real.
4. Recuperación de la información a través de sistemas masivos de almacenamiento.
5. Sistemas que faciliten la cooperación entre personas en el diseño y programación de grandes sistemas.
6. Introducción de datos mediante el reconocimiento de la voz a partir de la escritura manual directa.
7. Comprensión del lenguaje natural, sintáctica y semánticamente.
8. Reconocimiento de la voz de varios usuarios por el ordenador.
9. Una teoría de algoritmos simplificado.
10. Programación heurística o a través de principios generales.

Hoy en día sabemos que la lista que Licklider y Clark proponían era muy acertada y, desde entonces, aún estamos en un proceso de aprendizaje y de evolución para alcanzar una solución, ya que la informática es una ciencia muy joven.

Hansen (1971) en su libro *User Engineering Principles for Interactive Systems* hace la primera enumeración de principios para el diseño de sistemas interactivos:

1. Conocer al usuario.
2. Minimizar la memorización, sustituyendo la entrada de datos por la selección de ítems, usando nombres en lugar de números, asegurándose un comportamiento predecible y proveyendo acceso rápido a información práctica del sistema.
3. Optimizar las operaciones mediante la rápida ejecución de operaciones comunes, la consistencia de la interfaz y organizando y reorganizando la estructura de la información basándose en la observación del uso del sistema.
4. Facilitar buenos mensajes de error, crear diseños que eviten los errores más comunes, haciendo posible deshacer acciones realizadas y garantizar la integridad del sistema en caso de un fallo de software o hardware.



ENLACE DE INTERÉS

Conoce más información acerca del HCI:

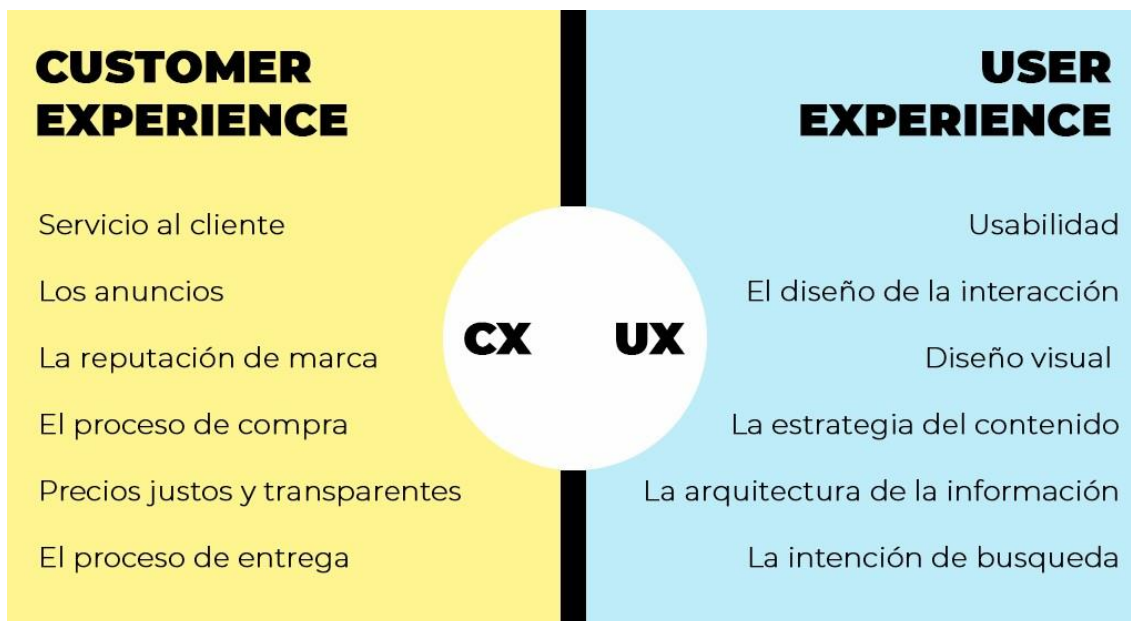


1.3 La experiencia de usuario UX (User Experience)

El concepto de experiencia de usuario está muy relacionado con las interacciones que un usuario realiza cuando usa una aplicación o un producto. Con esta definición podemos observar que el User Experience (UX) es la interpretación desde el punto de vista de un usuario frente a la definición de usabilidad que tiene un producto.

Hay varios términos que merece la pena distinguir dentro de la concepción y uso que realiza un usuario de un producto:

- **Experiencia de cliente:** conocida también por el término en inglés, Customer Experience (CX), se refiere a la experiencia de un consumidor con una compañía o empresa después de haber interactuado con esta. Comenzamos a hablar de experiencia de usuario en el primer momento que dicho usuario interactúa con la marca, no es necesario que haya tenido ninguna experiencia con un producto.
- **Experiencia de usuario:** en este caso User Experience (UX), nos referimos a la experiencia después del uso de un producto.



Diferencias entre CX y UX.

Fuente: <https://adigitalgo.com/marketing-digital/entendiendo-el-customer-experience-cx-y-el-user-experience-ux-a-traves-de-sus-diferencias/>

La imagen nos muestra una fantástica representación de las diferencias entre la experiencia de usuario y la de cliente. Hoy en día esta perspectiva de cliente se ha convertido en esencial y, por lo tanto, la especialización en aspectos más determinados como la experiencia de marca, la de compra, etc.



PARA SABER MÁS

Accede a esta web orientada al marketing, donde se desarrollan todos esos procesos de interacción del usuario con la empresa:



1.4 Normas ISO

Las siglas ISO corresponden a “Organización Internacional para la Normalización”. Se trata de una organización no gubernamental fundada en 1947, compuesta por representantes de organismos de normalización de más de ciento cincuenta países.

Las normas ISO se dividen por códigos que indican diferentes maneras de clasificar las definiciones de estándares, tanto de procedimientos y procesos como requerimientos y atributos de productos y servicios.

Hay varias normas ISO relacionadas con el término usabilidad. Si nos fijamos en las investigaciones referentes a los aspectos ergonómicos en la interacción hombre-máquina, tenemos algunos estándares internacionales, como, por ejemplo, los siguientes:

- **ISO 9241.** Aporta requerimientos y recomendaciones relacionados con las características del software y el hardware, como también del entorno que tiene que mejorar la usabilidad y los principios ergonómicos en el uso de las nuevas tecnologías con terminales visuales. Esta norma está dividida en muchas partes, añadidas a lo largo de los años noventa. Las partes 10 y de la 12 a la 17 son las que están relacionadas con el software e inciden especialmente en los diseñadores y los responsables de las pruebas de las interfaces de usuarios.
- **ISO 9126.** Esta norma desarrolla el modelo de calidad en el software proponiendo unos atributos de calidad como la funcionalidad, la fiabilidad, la usabilidad, la eficiencia, la facilidad de mantenimiento y la portabilidad.
- **ISO 13407.** Esta norma explica las actividades requeridas para el diseño de interfaces centradas en el usuario. Estas actividades o requerimientos hacen referencia a todo el ciclo de vida del desarrollo del software incidiendo en la fase de diseño de interfaces. Es una norma pensada para los jefes de proyectos informáticos o para los responsables del diseño de interfaces de este.



PARA SABER MÁS

Aquí encontrarás un resumen sobre las normas ISO y su contenido:



1.5 Medida de usabilidad de aplicaciones y tipos de métricas

Existen distintos métodos para conocer la usabilidad de una herramienta o sistema:

- Evaluación heurística.
- Test de usabilidad.



Medidas de usabilidad.

Fuente: <https://blog.ida.cl/estrategia-digital/test-usabilidad-fundamentales-asegurar-experiencia-usuario/>

1.5.1 Evaluación heurística

Según Jakob Nielsen, considerado el creador de la técnica, la evaluación heurística es “el nombre genérico de un grupo de métodos basados en evaluadores expertos que inspeccionan o examinan aspectos relacionados con la usabilidad de una UI (interfaz de usuario)”.

Estos métodos son complementarios entre sí. De hecho, estudios recientes han demostrado que las pruebas de usabilidad muestran dónde están los problemas, mientras que el análisis heurístico es más eficiente para proponer posibles soluciones.

La evaluación heurística de una aplicación está basada en la observación por parte de un experto en usabilidad o en interfaces humano–ordenador de ciertos parámetros o guías generales entre las que podemos citar:

- La coherencia en la presentación de la información.
- La visualización y coherencia de las acciones e interacción con el sistema.

- Los métodos de entrada/salida de información.
- El respeto por la ergonomía y los factores humanos.

Una posible tabla de heurísticas empleadas para la evaluación es la extraída del artículo científico de Reyes Vera, J. M., Berdugo Torres, M. I. y Machuca Villegas, L. (2016):

Heurística	Descripción
H0. Generalidades respecto a CMS	Las plataformas <i>e-learning</i> deben abordar unos elementos mínimos para ser consideradas como tal.
H1. Visibilidad del estado del sistema	El sistema debe mantener siempre a los usuarios informados del estado del sistema, con una apropiada retroalimentación y en un tiempo razonable.
H2. Lenguaje del usuario	El sistema debe hablar el lenguaje de los usuarios con las palabras, frases y conceptos familiares, en lugar de que los términos estén orientados al sistema.
H3. Control y libertad para el usuario	Los usuarios eligen a veces funciones del sistema por error y necesitan a menudo una salida de emergencia claramente marcada.
H4. Consistencia y estándares	Los usuarios no deben tener que preguntarse si las diversas palabras, situaciones o acciones significan la misma cosa.
H5. Ayuda para reconocer, diagnosticar y solucionar problemas	Los mensajes de error se deben expresar en un lenguaje claro (sin códigos extraños) indicando exactamente el problema y siendo constructivos.
H6. Prevención de errores	Prevenir la aparición de errores, resulta mejor que generar buenos mensajes de error.
H7. Reconocimiento antes que recordación	El usuario no deberá memorizar la información completa de la interacción.
H8. Flexibilidad y eficiencia de uso	Las instrucciones para el uso del sistema, deben ser visibles o fácilmente accesibles cada vez que se necesiten.
H9. Estética de diálogos y diseño minimalista	No debe existir información que sea irrelevante o que escasamente se necesite.
H10. Ayuda general y documentación	Es necesario disponer de una documentación que sea fácil de buscar, esté centrada en las tareas del usuario y que no sea muy extensa.
H11. Particularidades del usuario	Un sistema debe tener en cuenta las acciones que puede ejecutar un usuario para personalizar su apariencia.

Heurísticas de usabilidad utilizadas en la evaluación.

Fuente: https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-33052016000300008

1.5.2 Test de usabilidad

Un test de usabilidad es una medida concreta y objetiva de la usabilidad de una herramienta o sistema tomada a partir de usuarios reales con tareas reales.

Hay muchas escalas en las que se puede llevar a cabo un test de usabilidad: desde pequeños test de 5 usuarios y 2 o 3 días de duración hasta test de gran escala de varias semanas con muchos usuarios ejecutando sus tareas. Los test de gran escala pueden llegar a ser muy caros y se suelen utilizar cuando el proyecto tiene un impacto importante sobre muchas personas. Estos test se realizan en laboratorios especializados

con grandes equipos de observadores y multitud de recursos. Podemos distinguir cuatro tipos:

- **Test exploratorio.** Se llevará a cabo en las fases iniciales del ciclo de vida (requisitos, análisis, diseño). Tendrá como objetivo la evaluación de la eficiencia de los conceptos de diseño inicial y localizar errores iniciales en la definición de las necesidades y asunciones de los usuarios.
- **Test de evaluación de operaciones y aspectos del producto o servicio.** Durante las fases iniciales e intermedias del proyecto (desde la fase de toma de requisitos hasta la de desarrollo). Servirá para evaluar las conclusiones extraídas de los test exploratorios al principio del desarrollo del software para validar que no se han propagado los errores.
- **Test de validación.** Se lleva a cabo durante las fases finales (pruebas, finalización y transferencia). Servirá para evaluar si el producto final cumple los requisitos predeterminados de usabilidad establecidos al iniciar el proyecto permitiendo, de este modo, avanzarse a las posibles deficiencias del producto. Este test certificará la usabilidad del producto.
- **Test de comparación.** Este test implica a todas las fases del proyecto. Irá comprobando el producto con los que ofrece la competencia y también comprobará las diferentes alternativas de diseño con el objetivo de escoger la más sencilla de usar y de aprender. Este test se podrá aplicar en paralelo con otros tipos de test.



VÍDEO DE INTERÉS

Conoce uno de los test usados en métodos de investigación de usuario, un test de usabilidad que puede ser muy útil:



1.6 Pruebas de expertos y pruebas de usuarios

Hasta ahora hemos visto el concepto de usabilidad, sus dimensiones, sus características. Hemos visto métricas para medirlas, pero ahora habrá que analizar las interfaces mediante pruebas hechas por expertos en la materia y pruebas hechas por usuarios finales de las aplicaciones desarrolladas.

Utilizarán algunas de las métricas descritas, tanto las cualitativas en el caso de las pruebas de expertos como las cuantitativas en las pruebas con usuarios.

En primer lugar, habrá que tener muy claro que el que probamos será el sistema desarrollado (y, sobre todo, las interfaces), no los usuarios o verificadores. Cuanto antes realicemos las pruebas dentro del desarrollo del proyecto, mejor será para corregir errores, pero habrá que saber distinguir entre las pruebas que pueden hacer los expertos durante el desarrollo y las interfaces que se podrán mostrar a los usuarios.

En estos casos, habrá que tener en cuenta que ciertas decisiones tomadas a raíz de los resultados de las pruebas se tendrán que justificar muy bien ante los usuarios, acción que implica cierta burocracia. En otras situaciones, las pruebas ayudarán a elegir entre varias alternativas planteadas, puesto que el usuario nos acabará de dar un golpe de mano en algunas decisiones.

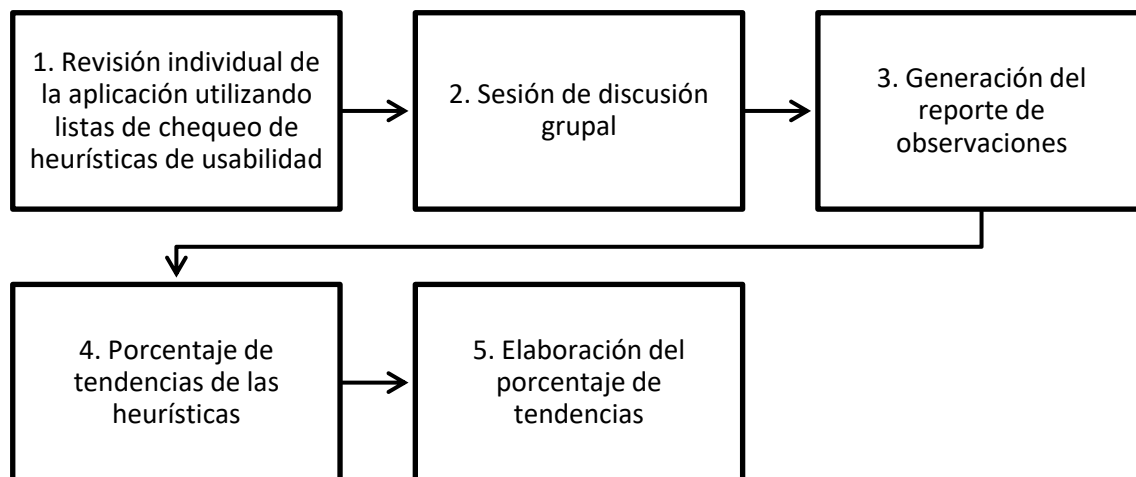
Para llevar a cabo las pruebas habrá que desarrollar previamente un plan de pruebas. Este plan tendrá que prever:

- El alcance de las pruebas (qué probaremos y qué no, hasta dónde llegaremos).
- Los propósitos de estas (cuáles son los objetivos, las razones o las justificaciones para probar en cada caso).
- Las fechas y los lugares donde se llevarán a cabo (si iremos a casa del usuario, si haremos que venga él, si necesitaremos una ubicación especial, cuánto rato tendrá que durar, en cuántas sesiones diferentes, etc.).
- Los participantes en las pruebas (cuántos usuarios, si los usuarios serán stakeholders o no tendrán ninguna vinculación con el desarrollo del proyecto, cuántos expertos, si estarán vinculados con el proyecto o serán externos).

Anteriormente, en las medidas heurísticas de usabilidad, podíamos ver un ejemplo de las medidas que se podrían plantear para analizar la usabilidad por parte de un experto. A partir de esta podemos plantear un formulario que el experto puede rellenar a modo de checklist y que puede tener la siguiente forma:

H1.- Viabilidad del estado del sistema	Ausente	Aceptable	Excelente
¿Cada parte del sistema comienza con un título o encabezamiento que describe el contenido de la pantalla?			
¿El esquema del diseño de los iconos y su apariencia visual es consistente en todo el sistema?			
...			

Estas listas de chequeo o formularios se crean a partir de los requisitos funcionales a evaluar. Están formadas por una serie de ítems expresados en forma de preguntas que permiten confirmar la aceptación o no del cumplimiento del requisito como vemos en la anterior tabla.



Proceso de evaluación.

Fuente: Elaboración propia de https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-33052016000300008

Como observamos en el diagrama, el proceso de evaluación tipo de un test de viabilidad puede tener los siguientes pasos:

- Revisión individual a través de los formularios creados.
- Revisión grupal.
- Generación de las estadísticas medidas.
- Generación del informe.

1.7 Herramientas para medir la usabilidad

Las herramientas para medir la usabilidad son soluciones que permiten evaluar cómo interactúan los usuarios con una aplicación o sitio web. Su objetivo es detectar problemas de uso, barreras de interacción y oportunidades de mejora en la experiencia

de usuario. Estas herramientas permiten obtener datos objetivos sobre la facilidad de uso de una interfaz, las más interesantes son:

- **Hotjar:** es una herramienta de análisis que proporciona mapas de calor (heatmaps) para entender dónde hacen clic los usuarios, hasta dónde se desplazan en la página y cómo interactúan con los diferentes elementos. También incluye la posibilidad de grabar sesiones de usuarios reales, permitiendo visualizar su comportamiento en tiempo real.



Mapa de Calor de una Web

Fuente: <https://fotografiaecommerce.com/blog/fotos-de-producto-mapas-de-calor-web/>

- **Google Analytics:** aunque se utiliza para análisis de tráfico, Google Analytics también proporciona datos útiles para la usabilidad, como la tasa de rebote, el tiempo promedio en la página y el flujo de usuarios. Estos indicadores permiten identificar posibles problemas en la navegación o en la estructura de contenidos.
- **Lyssna:** una plataforma especializada en pruebas de usabilidad que permite realizar test rápidos con usuarios reales, como pruebas de primeras impresiones o tests de preferencia de diseño. También permite realizar pruebas de cinco segundos para analizar lo que recuerdan los usuarios al ver una página durante unos pocos segundos, lo que ayuda a identificar si el diseño comunica el mensaje de manera efectiva.

- **Optimal Workshop:** herramienta ideal para realizar estudios de arquitectura de la información, como tests de card sorting (clasificación de tarjetas) y tree testing (pruebas de estructura jerárquica). Gracias a esta herramienta podemos evaluar como los usuarios perciben la organización y disposición de la información dentro de un sitio web, permitiendo mejorar la navegabilidad y estructura.
- **Crazy Egg:** similar a Hotjar, Crazy Egg ofrece mapas de calor, grabaciones de sesiones y análisis de clics. Además, proporciona una función de "Scrollmap" que muestra hasta qué punto de la página los usuarios se desplazan, lo que puede ayudar a identificar si hay información importante que no están viendo.
- **Maze:** una herramienta de pruebas de usabilidad remota que permite realizar tests con prototipos interactivos. Maze es especialmente útil para obtener resultados rápidos y precisos de cómo los usuarios navegan por una interfaz antes de que esté completamente desarrollada. Su integración con plataformas de diseño como Figma y Sketch facilita las pruebas en las fases iniciales del desarrollo.



ENLACE DE INTERÉS

Accede a la página oficial de la herramienta Maze, donde podrás encontrar más información sobre la misma:



EJEMPLO PRÁCTICO

En la aplicación que vamos a realizar para el club de baloncesto, una de las pantallas clave donde el cliente más ha puesto énfasis es en el registro del socio, ya que por palabras del cliente debe recoger perfectamente todos los datos que necesita el club y debe ser sencillo y no tedioso para el usuario.

¿Qué prueba de usabilidad deberíamos plantearnos? ¿Cuándo la deberíamos realizar?

Para responder a estas preguntas debemos cuestionarnos lo siguiente:

- Determinar qué medir: de todos los parámetros vistos nos vamos a centrar en los siguientes:
 - Facilidad de aprendizaje, es decir, sin explicación alguna o ayuda el usuario puede completar el formulario de inscripción.
 - Efectividad, con los parámetros introducidos en el formulario de inscripción se tienen todos los datos necesarios para abrir una ficha en el club.
 - Tolerancia al error o cómo el formulario ayuda a corregir posibles errores en la introducción de datos.
- Cuál será el método: la prueba que realizaremos es un test exploratorio con preguntas cerradas del tipo:

¿Consiguió finalizar todo el formulario?

Sí/No

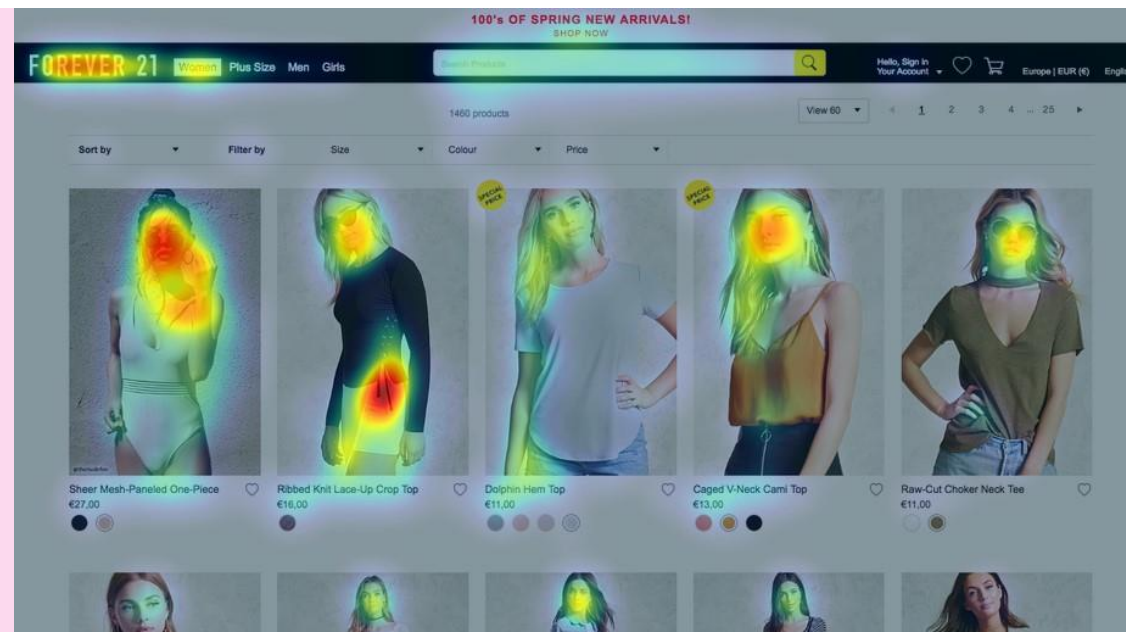
Hubo algún apartado que no consiguió responder

- Escenario y participantes: la directiva del club y algunos socios escogidos por el club pueden ser el foco del test, teniendo en cuenta que es importante tener una muestra variada de personas respecto a las nociones con las tecnologías.
- Realización de la prueba: la prueba se puede realizar con un formulario prueba a partir del mockup aceptado de la pantalla de inscripción.
- Analizar y corregir: a partir de los resultados se plantearían modificaciones sobre todo en las partes erróneas proporcionadas bien por los usuarios o bien a partir de los datos introducidos.



EJEMPLO PRÁCTICO

Trabajas en una empresa que dispone de una tienda online. Tu jefe te solicita realizar un análisis de la usabilidad y los productos de la tienda de ropa utilizando un mapa de calor generado a partir del comportamiento de los usuarios en la página.



Tienda Online

Fuente: <https://www.eyequant.com/resources/what-are-website-heatmaps-and-how-to-use-them/>

¿Qué podrías analizar sobre el comportamiento de los usuarios?

Vamos a realizar el análisis:

- Las zonas que aparecen en rojo y amarillo indican los lugares donde los usuarios interactuaron más (clics o puntos de atención visual), las zonas verdes y azules representan lugares con menos atención de los usuarios.
- El logo de la tienda en la parte superior izquierda recibe mucha atención. Esto es típico, ya que los usuarios suelen mirar la parte superior de la página para obtener información sobre la marca
- Los productos ubicados en el centro superior de la página, muestran altos niveles de interacción parece que estos productos tienen mayor atractivo visual o están mejor ubicados. Habría que analizar si poniéndolos en otra región recibirían la misma atención y si fuera así potenciar este tipo de productos.
- La barra de navegación también recibe una gran cantidad de atención, lo que indica que los usuarios buscan alternativas de filtrado y navegación rápidamente. En particular se interesan por la palabra clave mujer más que las demás.
- Los productos ubicados en las esquinas inferiores tienen menos interacción, esto puede deberse a que los usuarios no suelen desplazarse lo suficiente hacia abajo o que los productos no son visualmente atractivos.
- Los filtros (tamaño, color, precio) muestran poca interacción en comparación con otros elementos de la página, lo que podría sugerir que los usuarios prefieren navegar directamente por los productos en lugar de filtrar las opciones.



VÍDEO DE INTERÉS

Aquí podrás comprender cómo se enfocan las pruebas de usabilidad de Google:



2. ACCESIBILIDAD

El cliente solicita que la aplicación pueda ser usada por cualquier tipo de persona, ya que existe personal con discapacidades físicas y cognitivas en su plantilla. Para ello seguirás las pautas de accesibilidad marcadas por las normas ISO y por último lo evaluarás en base a los test de accesibilidad.

La accesibilidad en el diseño de interfaces son el conjunto de características y principios que aseguran que una aplicación o sitio web pueda ser utilizado por todas las personas, independientemente de sus capacidades o discapacidades. Un diseño accesible no solo beneficia a las personas con discapacidades, sino que también mejora la usabilidad para todos los usuarios. Al adoptar las pautas de accesibilidad, las aplicaciones serán más fáciles de usar en diferentes contextos, dispositivos y situaciones.

2.1 Principios básicos de la accesibilidad

Los principios básicos para garantizar la accesibilidad en el diseño de interfaces se centran en hacer que las aplicaciones sean perceptibles, operables, comprensibles y robustas. Estos son conocidos como los principios POUR:

- **Perceptible:** la información y los componentes de la interfaz deben ser presentados de manera que todos los usuarios puedan percibirlos. Esto incluye proporcionar alternativas textuales para el contenido no textual (por ejemplo, descripciones de imágenes), subtítulos para vídeos y un diseño de contraste adecuado para personas con discapacidades visuales.

- **Operable:** los elementos interactivos deben ser accesibles para que los usuarios puedan operar la interfaz con diferentes tipos de dispositivos de entrada, como el teclado, mouse, o dispositivos de asistencia. Los elementos como botones, enlaces y formularios deben ser navegables sin necesidad del uso del mouse.
- **Comprensible:** los usuarios deben ser capaces de entender el contenido y la interfaz. Esto implica un lenguaje claro, la disposición lógica de la información, y evitar interacciones que puedan ser confusas. Las interfaces deben ser predecibles y consistentes en su diseño.
- **Robusto:** el contenido debe ser lo suficientemente robusto como para que las tecnologías de asistencia, como los lectores de pantalla, puedan interpretar correctamente la interfaz, tanto en navegadores antiguos como en modernos.

2.2 Pautas de accesibilidad

Las pautas de accesibilidad son unos principios en forma de guía que si los seguimos conseguiremos que nuestra aplicación sea más accesible. Estas pautas se basan en estándares internacionales como las Web Content Accessibility Guidelines (WCAG), establecidas por el W3C. A continuación, se destacan algunas de las pautas más relevantes:

- **Texto alternativo para imágenes:** todas las imágenes deben tener un texto alternativo descriptivo para que los lectores de pantalla puedan comunicar el contenido visual a personas con discapacidad visual.
- **Contraste de color:** es fundamental que el texto y los elementos visuales tengan un contraste adecuado con el fondo para facilitar su legibilidad a personas con baja visión o daltonismo. Por ejemplo, los colores deben cumplir con una relación de contraste mínimo
- **Navegación mediante teclado:** todos los componentes interactivos, como botones, enlaces y formularios, deben ser accesibles a través del teclado. Esto asegura que personas con discapacidades motrices puedan navegar sin necesidad de un mouse.
- **Subtítulos y transcripciones:** los vídeos deben incluir subtítulos para personas con dificultades auditivas, y si es posible, una transcripción para contenidos multimedia, permitiendo la comprensión sin necesidad de sonido.

- **Identificación clara de los errores:** en formularios o procesos interactivos, los errores deben ser comunicados claramente, con instrucciones específicas sobre cómo corregirlos. Esto es clave para usuarios con discapacidades cognitivas.

En el diseño de aplicaciones móviles, la accesibilidad cobra aún más importancia debido a las limitaciones de espacio y la interacción directa. Algunas pautas clave para garantizar la accesibilidad en dispositivos móviles son:

- **Gestos alternativos:** proporcionar alternativas accesibles para gestos complejos (como deslizar o hacer zoom con dos dedos), permitiendo la misma funcionalidad a través de un teclado o un solo dedo.
- **Tamaño adecuado de los elementos táctiles:** los botones y otros elementos interactivos deben tener un tamaño suficiente para ser fácilmente accesibles, evitando que los usuarios con dificultades motrices tengan problemas para seleccionarlos.



PARA SABER MÁS

Accede para encontrar una serie de ejemplos de webs que incumplen requisitos de accesibilidad:



2.3 Estándares y normas internacionales

Las normas ISO relacionadas con la accesibilidad se centran en proporcionar directrices para garantizar que las aplicaciones, sitios web y otros sistemas tecnológicos sean utilizables por todos los usuarios, incluidas personas con discapacidades:

- **ISO 9241-171:** ofrece requisitos y recomendaciones específicos para garantizar la accesibilidad del software, abarcando aspectos relacionados con el diseño de interfaces para personas con discapacidades. Proporciona directrices sobre cómo hacer que el software sea accesible para todos los usuarios, especialmente aquellos con discapacidades físicas o cognitivas.

- **ISO/IEC 29138:** establece los requisitos de accesibilidad para productos de tecnologías de la información y comunicación (TIC). Cubre tanto hardware como software, orientando a los diseñadores y responsables de sistemas en cómo asegurar que sus productos sean accesibles y cumplan con las necesidades de todos los usuarios.
- **ISO 30071-1:** proporciona un marco para la gestión de la accesibilidad en productos y servicios digitales, asegurando que las necesidades de accesibilidad se integren a lo largo de todo el ciclo de vida del desarrollo, desde la planificación hasta la implementación y prueba del producto final.

2.4 Medidas de accesibilidad

Al igual que en la usabilidad, la accesibilidad la podemos medir en base a los siguientes criterios:

- Evaluación heurística.
- Test de accesibilidad.

2.4.1 Evaluación heurística de accesibilidad

La evaluación heurística de accesibilidad consiste en un análisis realizado por expertos que examinan cómo una aplicación cumple con las pautas de accesibilidad, como las WCAG (Pautas de Accesibilidad para el Contenido Web) y la ISO 9241-171. Esta evaluación se centra en identificar problemas que podrían afectar a personas con discapacidades al usar la aplicación.

Estas pruebas se basan en criterios como:

- **Uso del teclado:** asegura que todas las funcionalidades puedan ser operadas sin necesidad de un ratón, permitiendo el uso exclusivo del teclado.
- **Contraste de colores:** garantiza que el contraste entre el texto y el fondo sea suficiente para personas con baja visión o daltonismo.
- **Texto alternativo en imágenes:** evalúa si las imágenes tienen descripciones alternativas que puedan ser leídas por herramientas como lectores de pantalla.
- **Estructura de la página:** revisa si los encabezados, listas y otros elementos estructurales están correctamente etiquetados para una navegación adecuada.

2.4.2 Test de accesibilidad

Al igual que con la usabilidad, hay muchas formas de llevar a cabo un test de accesibilidad, desde pequeñas pruebas con grupos reducidos de usuarios y tecnologías asistivas, hasta evaluaciones a gran escala con participantes diversos que cubren todas las fases del proyecto. Las pruebas pueden llevarse a cabo tanto en laboratorios especializados como de manera remota. En general, podemos distinguir varios tipos:

- **Test de accesibilidad automatizado:** se utiliza software especializado para analizar el código del sitio o aplicación y detectar problemas de accesibilidad, como la falta de texto alternativo en imágenes, encabezados mal estructurados o bajo contraste de colores. Estos tests ofrecen una visión rápida, aunque no cubren todos los aspectos de accesibilidad que un usuario real puede enfrentar.
- **Test de accesibilidad manual:** consiste en la evaluación manual de ciertos aspectos de la accesibilidad que no pueden ser verificados automáticamente, como la navegación mediante teclado, la interacción con lectores de pantalla o la correcta interpretación de los elementos visuales y estructurales.
- **Test de accesibilidad con usuarios:** en este tipo de prueba se involucra a personas con discapacidades reales para que realicen tareas en la interfaz y se identifiquen las barreras de accesibilidad desde una perspectiva práctica. Estos tests proporcionan información directa sobre la experiencia del usuario final y revelan problemas que las evaluaciones automáticas o manuales podrían pasar por alto.
- **Test de accesibilidad remoto:** se realiza con usuarios que prueban la aplicación o sitio web desde sus propios entornos y dispositivos, proporcionando una visión más cercana a cómo interactúan en su vida diaria. Este enfoque permite obtener información relevante sobre el comportamiento de la interfaz en condiciones del mundo real, como diferentes dispositivos, conexiones a internet o configuraciones específicas.

2.5 Herramientas para medir la accesibilidad

Las herramientas para medir la accesibilidad son soluciones técnicas que permiten evaluar si una aplicación cumple con las normas y pautas de accesibilidad establecidas. Como hemos visto en el apartado de pruebas, existen tanto herramientas automáticas, que realizan análisis rápidos del código, como manuales, que requieren la intervención de expertos para una revisión más detallada:

- **WAVE (Web Accessibility Evaluation Tool):** es una herramienta automatizada que evalúa la accesibilidad de sitios web y genera un reporte visual destacando los problemas encontrados. WAVE señala errores comunes como la falta de texto alternativo en imágenes, encabezados mal estructurados y problemas de contraste de color.
- **Axe Accessibility Tool:** esta extensión para navegadores, como Chrome y Firefox, realiza análisis automáticos de accesibilidad directamente en el navegador, ayudando a identificar problemas dentro del código HTML y CSS.



ENLACE DE INTERÉS

Aquí encontrarás la extensión de Axe Accessibility Tool para Google Chrome, la cual se puede encontrar en la propia web store:

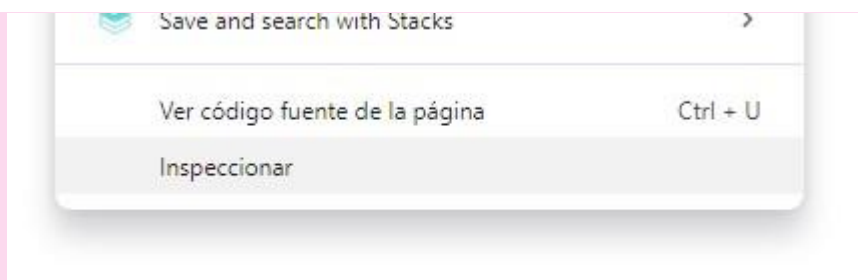


EJEMPLO PRÁCTICO

Nuestro jefe nos solicita analizar la accesibilidad de una aplicación de la empresa con una herramienta de análisis automático, la aplicación está alojada en la web: <https://www.goodreads.com/>

Para empezar con el análisis vamos a utilizar una herramienta llamada Axe Accessibility Tool para Google Chrome Para ellos instalaremos la extensión:

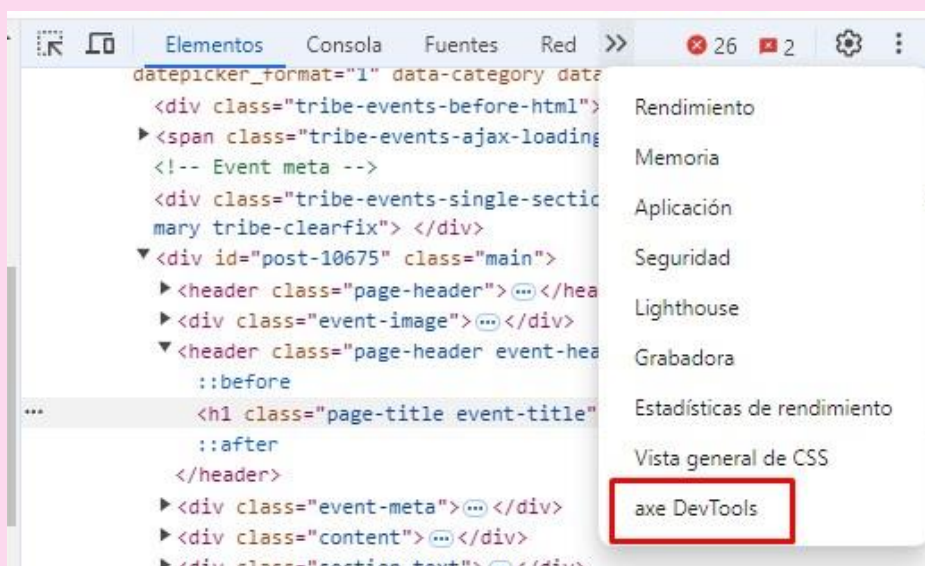
- Entramos en la web que nos solicita nuestro jefe: <https://www.goodreads.com/>.
- Pulsamos en botón derecho e Inspeccionar



Inspeccionar Elemento

Fuente: Elaboración propia

- Buscamos la pestaña de la herramienta:



Apertura de la Herramienta

Fuente: Elaboración propia

- Pulsamos en escaneo de la página completa



Escaneo Completo
Fuente: Elaboración propia

- Podemos ver que el análisis arroja 28 problemas de accesibilidad para analizar en nuestro equipo.



Total Issues: 28	
<u>Los elementos deben tener un contraste de colores suficiente</u>	7 ▶
<u>El elemento <html> debe tener un atributo lang</u>	1 ▶
<u>Los enlaces deben distinguirse del texto adyacente por un medio que no dependa del color</u>	3 ▶
<u>Los enlaces deben tener texto discernible</u>	1 ▶
<u>Los elementos deben estar contenidos en un o un </u>	16 ▶

Análisis de la web
Fuente: Elaboración propia

- **Lighthouse:** integrada en las herramientas para desarrolladores de Google Chrome, Lighthouse permite realizar auditorías de accesibilidad, rendimiento y SEO. En el caso de accesibilidad, genera un informe con puntuaciones sobre varios aspectos, desde la navegación con teclado hasta el uso adecuado de etiquetas HTML.
- **NVDA (NonVisual Desktop Access):** aunque no es una herramienta de evaluación, NVDA es un lector de pantalla gratuito que permite a los desarrolladores experimentar cómo interactúan las personas ciegas con las aplicaciones y sitios web.
- **Contrast Checker:** es una herramienta sencilla pero poderosa que verifica si el contraste entre el texto y el fondo cumple con los requisitos de accesibilidad. Este tipo de herramienta es útil para verificar que el contenido sea legible para personas con baja visión o daltonismo.

3. ESQUEMAS Y MAQUETAS

Una vez planificada la usabilidad y accesibilidad de la aplicación, vas a diseñar la experiencia de usuario con la aplicación para garantizar que se pueda empezar el desarrollo y el cliente quede más satisfecho. Para ello, tu jefe te solicita hacer un esquema o wireframe de la aplicación. Una vez validada, debes hacer su maqueta para garantizar que es lo que quiere el cliente.

En este apartado nos centraremos en dos herramientas del proceso de diseño de interfaces: los esquemas (wireframes) y las maquetas (mockups). Estas herramientas nos permiten visualizar y planificar la estructura y apariencia de nuestra aplicación antes de entrar en la fase de desarrollo.



VÍDEO DE INTERÉS

Atiende a esta explicación, a modo de introducción, de los conceptos de maqueta y esquema:



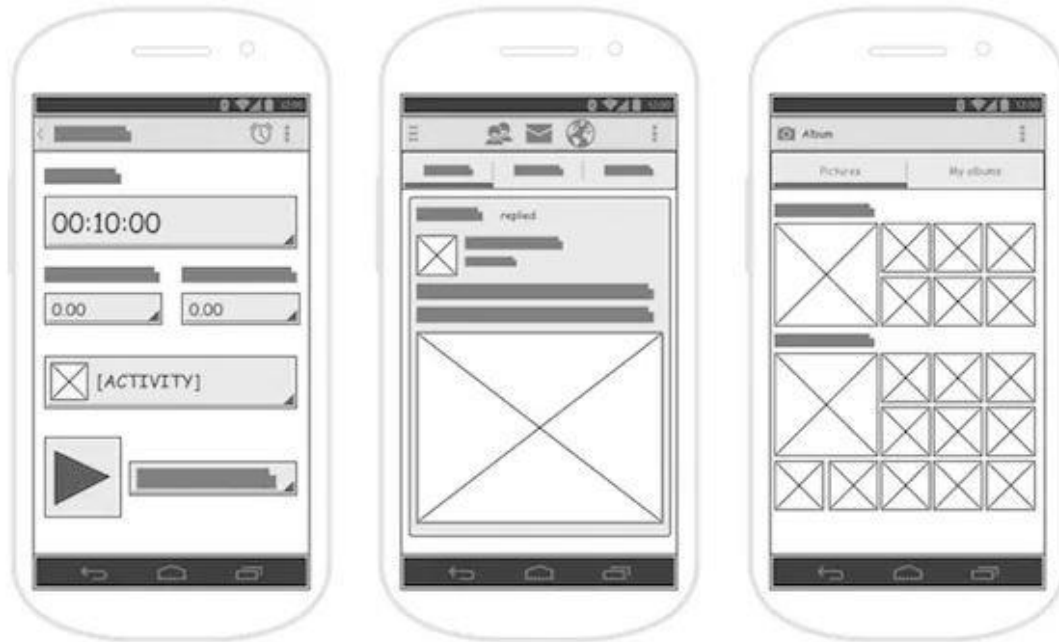
3.1 Esquemas (Wireframes)

Los esquemas o wireframes son representaciones visuales simplificadas de la estructura de una interfaz. Su objetivo principal es definir el layout básico y la organización de los elementos en pantalla sin entrar en detalles estéticos como colores o tipografías. Los wireframes sirven como guía para entender cómo se distribuirá la información y cómo el usuario interactuará con la aplicación.

Pautas para crear wireframes efectivos:

- **Enfocarse en la funcionalidad:** priorizar la disposición y organización de los elementos sobre el aspecto visual.
- **Simplicidad:** utilizar formas básicas y evitar detalles innecesarios que puedan distraer del propósito principal.

- **Claridad en la navegación:** representar claramente cómo el usuario se moverá entre las diferentes secciones de la aplicación.
- **Colaboración:** compartir los wireframes con el equipo y con los stakeholders para obtener feedback temprano y realizar ajustes necesarios.



Ejemplo Wireframe Aplicación

Fuente: <https://www.contratarprogramador.com/prototipo-wireframe-del-proyecto/>

3.2 Maquetas (Mockups)

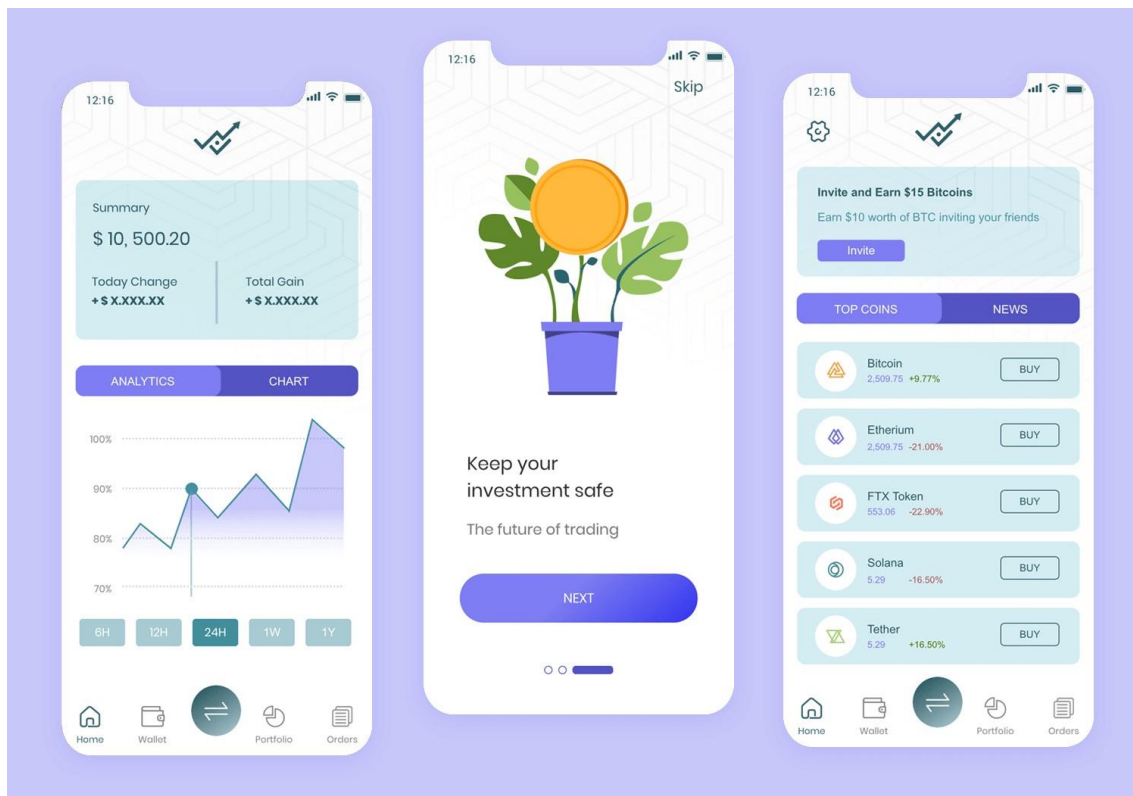
Las maquetas o mockups son representaciones más detalladas y realistas de la interfaz, incluyendo aspectos visuales como colores, tipografías, iconos y otros elementos gráficos. A diferencia de los wireframes, los mockups muestran cómo se verá la aplicación en su versión final, aunque sin funcionalidad operativa.

Importancia de los mockups:

- **Visualización del diseño final:** permiten ver cómo se combinarán todos los elementos visuales y evaluar la estética general.
- **Comunicación con el cliente:** ayudan a presentar una imagen más precisa de la aplicación, facilitando el feedback y la toma de decisiones.
- **Guía para desarrolladores:** sirven como referencia detallada durante la fase de implementación.

Pautas para crear mockups efectivos:

- **Consistencia visual:** mantener coherencia en el uso de colores, tipografías y estilos gráficos.
- **Atención al detalle:** incluir todos los elementos que estarán presentes en la aplicación final.
- **Iteración y feedback:** revisar y ajustar el diseño basado en las opiniones del equipo y del cliente.



Ejemplo Mockup Aplicación

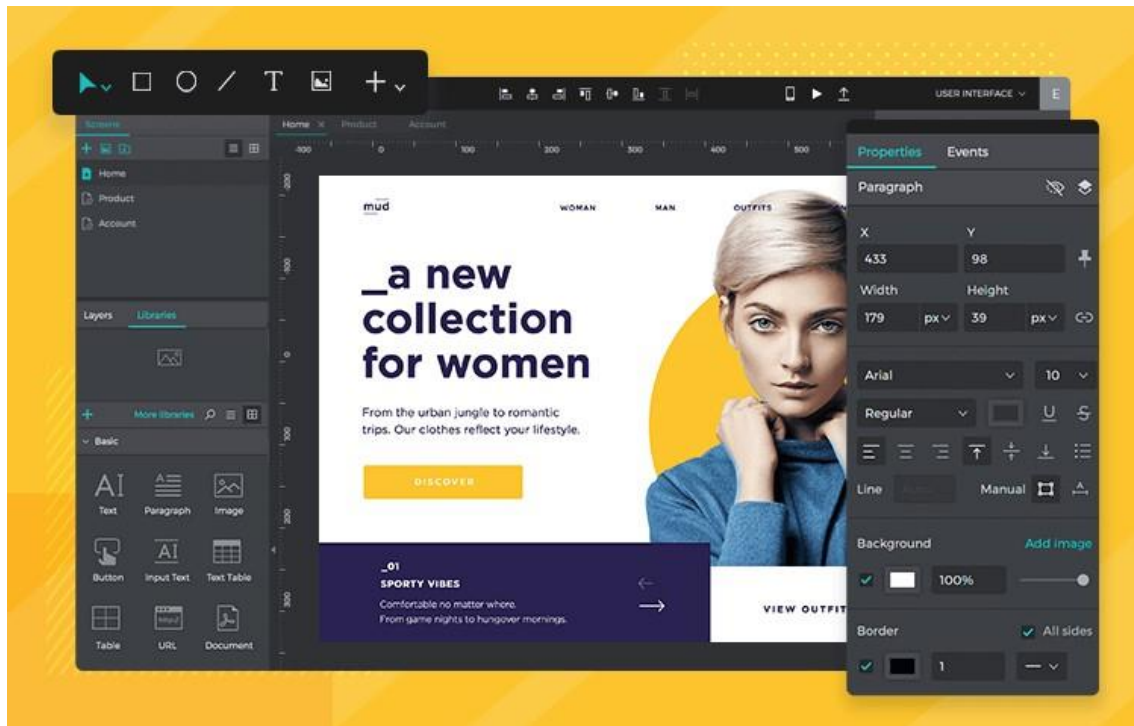
Fuente: <https://www.contratarprogramador.com/prototipo-wireframe-del-proyecto/>

3.3 Herramientas para la realización de esquemas (wireframes) y maquetas (mockups)

La importancia de la creación de borradores iniciales en el ciclo de vida de desarrollo de aplicaciones está fuera de duda hoy en día, hasta el punto de haber sustituido incluso a las tediosas especificaciones en papel por mockups mucho más gráficas que cliente, proveedor, desarrollador y diseñador gráfico entiende.

El número de herramientas que tenemos hoy en día es muy elevado para la realización de mockups y esquemas para nuestras aplicaciones. Veamos herramientas que podemos usar gratuitamente para nuestros proyectos:

- **Moqups:** herramienta online que nos permite, además de realizar diseños de interfaces, otra serie de diseños. Es una herramienta con la que podemos colaborar con otros usuarios, de una forma limitada en su versión gratuita, pero suficiente para iniciarnos con estas herramientas.
- **Justinmind:** herramienta descargable multiplataforma y que nos permite realizar diseños a partir de modelos. Esta herramienta es muy potente y nos va a permitir realizar mockups muy atractivos



Interfaz de Justinmind

Fuente: <https://www.justinmind.com/blog/new-release-9-faster-easier-design-features/>



ENLACE DE INTERÉS

Aquí tienes la web oficial de JustinMind, donde podrás descargar la herramienta:





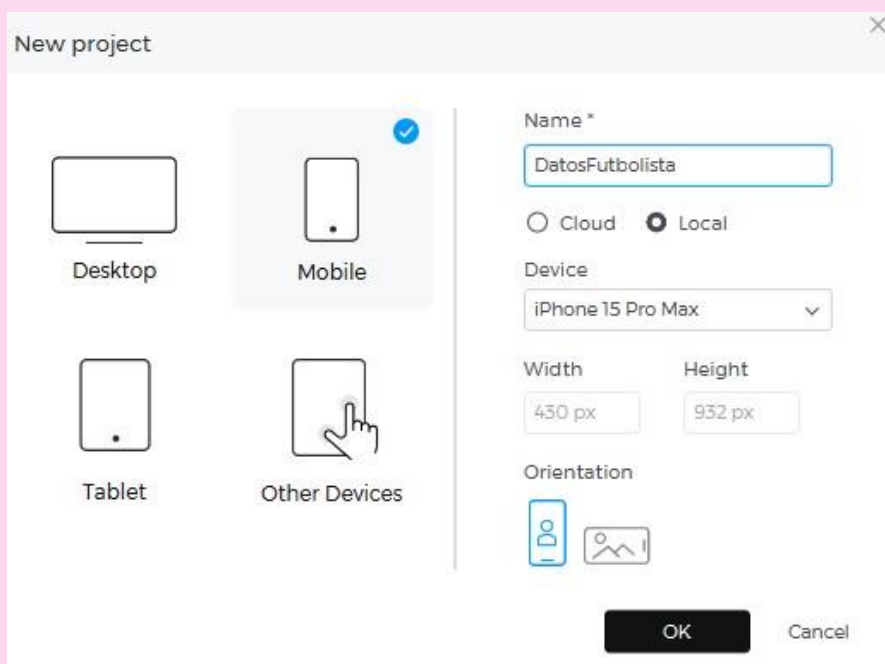
EJEMPLO PRÁCTICO

Nuestro jefe nos solicita hacer un mockup de una pantalla de aplicación móvil que recoja información sobre jugadores de fútbol, para ello tiene que recoger.

- Foto
- Nombre.
- Apellidos.
- Correo electrónico.

Como nuestro jefe nos ha solicitado nos centraremos en el diseño para móvil.

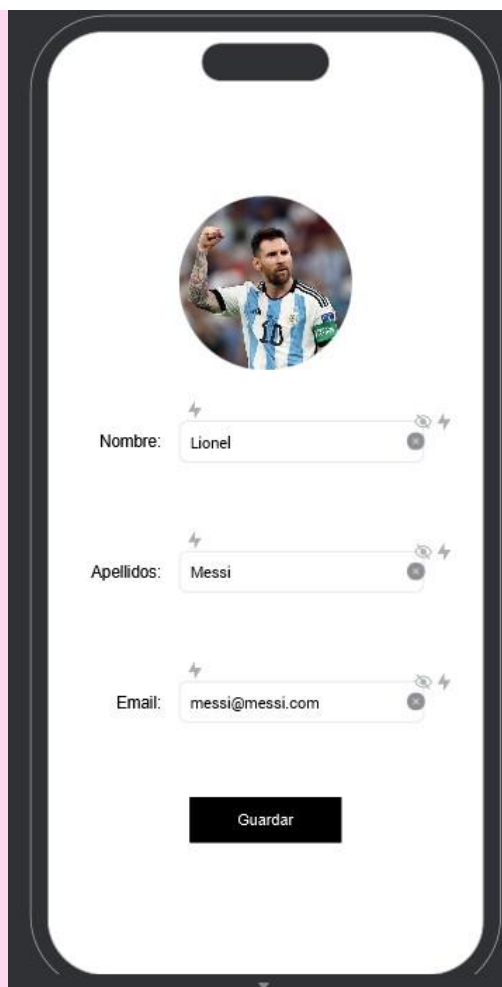
- Descargamos justinmind de su página oficial y creamos un nuevo proyecto, lo asignaremos para móviles y en local.



Creación del proyecto

Fuente: Elaboración propia

- Arrastraremos los campos que nos solicita nuestro jefe.

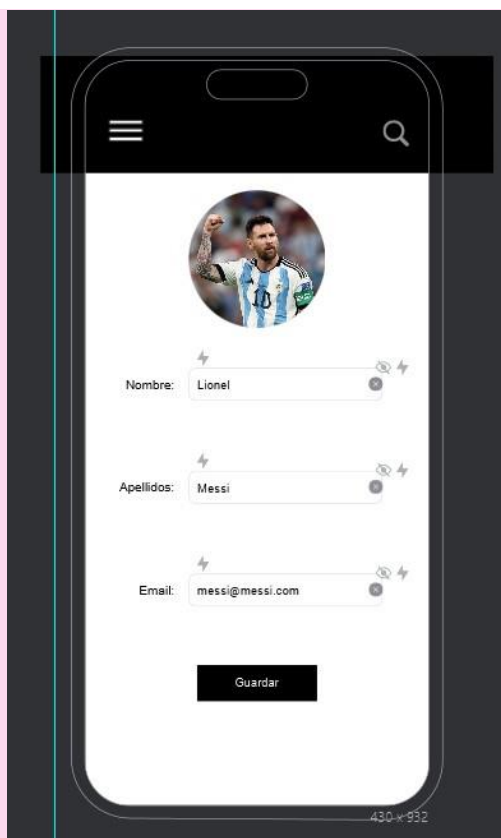


A mobile application interface for registering a football player. At the top is a circular profile picture of Lionel Messi. Below it are three input fields: 'Nombre:' with the value 'Lionel', 'Apellidos:' with the value 'Messi', and 'Email:' with the value 'messi@messi.com'. Each field has a lightning bolt icon on the left and a clear icon on the right. At the bottom is a black button with the text 'Guardar'.

Campos formularios futbolista

Fuente: Elaboración propia

- Agregamos el menú superior



A mobile application interface for registering a footballer. The screen features a dark header with a hamburger menu icon on the left and a search icon on the right. Below the header is a circular profile picture of a footballer in an Argentina jersey. Underneath the photo are three form fields: 'Nombre:' with the value 'Lionel', 'Apellidos:' with the value 'Messi', and 'Email:' with the value 'messi@messi.com'. Each field has a lightning bolt icon to its right, indicating a required field. At the bottom of the form is a black button with the white text 'Guardar'. The entire app mockup is set against a light pink background.

Menú formulario futbolista

Fuente: Elaboración propia

4. PAUTAS DE DISEÑO

Para la realización del diseño (maquetas y esquemas) necesitas garantizar que se siguen unos principios básicos de diseño, como la proporción, seguir unas pautas en iconos, colores, fuentes.... Una vez sigas estas pautas puedes garantizar que la aplicación va a ser más adecuada en parámetros de usabilidad y accesibilidad.

Cuando desarrollamos un proyecto de creación de un software, habrá que tener en cuenta, en todas las fases de su ciclo de vida, la creación de las interfaces que facilitarán la interacción con los usuarios. En cada una de las fases se tendrán que tomar decisiones referentes a las interfaces de usuario, en más o menos medida, que tendrán un efecto decisivo en el éxito del proyecto.

Para establecer las pautas de diseño, también habrá que conocer y entender el tipo de usuario para el cual se desarrolla el software y, en definitiva, las interfaces con las cuales interactuará. Estos podrán ser usuarios expertos, intermedios o bien noveles, sin mucha experiencia.

- **Un usuario experto.** Dentro de los usuarios expertos todavía podemos hacer una segunda clasificación. Hay que diferenciar entre el usuario experto en el uso de las tecnologías de la información y el usuario experto en las reglas de negocio en las que ubicamos el software. Sea como fuere, un usuario experto no necesita unos menús muy intuitivos, ni unos iconos con unas explicaciones muy claras, sino que con la experiencia y el sentido común un usuario experto puede usar una interfaz que se preocupa más de ofrecer mucha funcionalidad que mucha información. Por ejemplo, para este tipo de usuario experto, podremos ofrecer muchas funcionalidades con muchos menús y submenús o con más necesidad de pasos hasta llegar a la opción deseada. Este usuario tendrá la capacidad de encontrar la opción que necesita y podrá navegar por las opciones sin muchos problemas.
- **Los usuarios intermedios** son usuarios que se encuentran entre los expertos y los noveles. En este caso, no hablaremos de conocimiento de negocio. Pueden tener un nivel mediano de conocimiento o uso de las TIC. Esto hará que no necesitemos ofrecerles tantas facilidades ni información como a los usuarios noveles, pero sí que habrá que facilitarles el trabajo tanto como sea posible.
- **Un usuario novel.** Un usuario sin mucha experiencia (en el uso de las TIC o en conocimientos sobre las reglas del juego del negocio al que pertenece el software) necesita una interacción con el software mucho más guiada, con mucha más ayuda, tanto textual como gráfica. Un usuario novel necesita ver

unos iconos mucho más explícitos, que le faciliten la comprensión de las opciones. También necesita tener una ayuda muy accesible y fácil de entender como, por ejemplo, mostrar pequeñas frases o informaciones cuando el ratón pasa por encima de un icono, indicando la utilidad (denominadas tooltip o indicador de función).



ENLACE DE INTERÉS

Amplia tus conocimientos sobre los distintos roles dentro del User Experience (UX):



VÍDEO DE INTERÉS

Visualiza este vídeo sobre la usabilidad en el diseño de experiencias digitales, especialmente desde el minuto 7 al 15:



4.1 Principios básicos

En el diseño de una interfaz habrá que tener en cuenta algunos principios básicos. Por ejemplo, en 1971 Hansen proponía cuatro principios básicos:

- **Conocer al usuario:** sus capacidades, sus necesidades funcionales y su evolución.
- **Minimizar la memorización:** utilizando selección en lugar de entrada de datos, empleando nombres con sentido en lugar de códigos y nombres crípticos,

haciendo que los resultados de las operaciones sean predecibles y facilitando el acceso a ayudas y a la documentación.

- **Optimizar las operaciones:** posibilitando la ejecución rápida de operaciones usuales, preservando la consistencia visual, aprovechando las capacidades memorísticas del usuario y organizando y reorganizando las órdenes de acuerdo con la utilización que el usuario hace.
- **Permitir los errores:** dando al usuario mensajes de error entendedores, diseñando para evitar los errores comunes, permitiendo que las acciones sean reversibles y garantizando la integridad del sistema en caso de errores graves.

La mayor parte de la interacción con un ordenador se hace de manera visual. Esto hace que las decisiones sobre la tipografía, la distribución de la información en la pantalla o de la selección de los colores más apropiados no solo harán que la interfaz sea más o menos atractiva, sino que la harán más o menos usable y determinarán el éxito. Algunos principios de diseño gráfico que se deben tener en cuenta son:

- **El principio de agrupamiento.** Organizar el espacio visible en bloques separados de controles similares y con un título para cada bloque. Los sistemas de ventanas actuales aplican este principio de manera recursiva. Esto ayudará al usuario a encontrar la información que necesita y a formarse un modelo conceptual del funcionamiento del programa.
- **El principio de visibilidad y utilidad.** Los controles usados frecuentemente tienen que ser visibles y fácilmente accesibles. Análogamente se tendrán que diseñar sistemas para esconder o para comprimir los menos utilizados.
- **El principio de consistencia inteligente.** Utilizar una distribución de la información similar para funciones similares, para habituar los usuarios a encontrar la información en los mismos lugares en situaciones parecidas.
- **El principio de economía del diseño.** Omitir cualquier elemento que no aporte ninguna información a las interfaces gráficas de usuario.
- **El principio del color como suplemento.** Utilizar los colores con medida para enfatizar información, sin ser un elemento exclusivo para comunicar información. El color es más fácil de usar mal que de una manera correcta.
- **El principio de reducción del desorden.** Es un principio que resume el resto de principios vistos anteriormente. Si solo son visibles los controles utilizados más

frecuentemente, agrupados en grupos con sentido, con un uso minimalista del color y sin elementos superfluos, entonces se tendrá una interfaz suficiente atractiva y bastante funcional en la cual el desorden y la arbitrariedad se habrán reducido al mínimo.

4.2 Pautas de diseño de la estructura de la interface

La localización visual de los componentes es importante porque la relación entre los componentes es indicada por su posición. Esto se llama layout en diseño de interfaces. Un layout limpio es muy importante para crear un flujo visual de información sin problemas para el usuario.

La interfaz principal de una aplicación debe disponer de los siguientes elementos organizativos: cabecera, menú principal, menú de navegación, ruta, contenido y pie.

Cabecera:	Cabecera de la aplicación que contempla como estándar la inclusión del logotipo y denominación del organismo, así como el logotipo y denominación de la aplicación. Todo ello bien sobre un fondo corporativo o sobre el fondo representativo del diseño de la aplicación.
Menú principal:	Estandarización del menú principal, ubicado inicialmente en la barra superior, conteniendo los iconos para el cierre de la aplicación, acceso a la ayuda, vuelta al inicio de la aplicación, un método de contacto, etc.
Menú de navegación:	Menú que permita la navegación a otros elementos de la aplicación en el que se contempla hasta dos niveles de menú de navegación y se propone su ubicación como barra lateral izquierda para cumplir con las pautas de usabilidad.
Ruta:	Establece la ruta (migas de pan) desde el inicio de la aplicación a la situación actual donde se encuentre el usuario. Permite la navegación a los diferentes niveles, por lo que se ha navegado previamente a la llegada a la pantalla actual dentro de la aplicación.
Pie:	Establece la barra inferior de la aplicación que muestra el aviso legal, la propiedad de contenidos y la propiedad de la aplicación o cualquier información importante que deba ser mostrada en el uso del sistema de información.

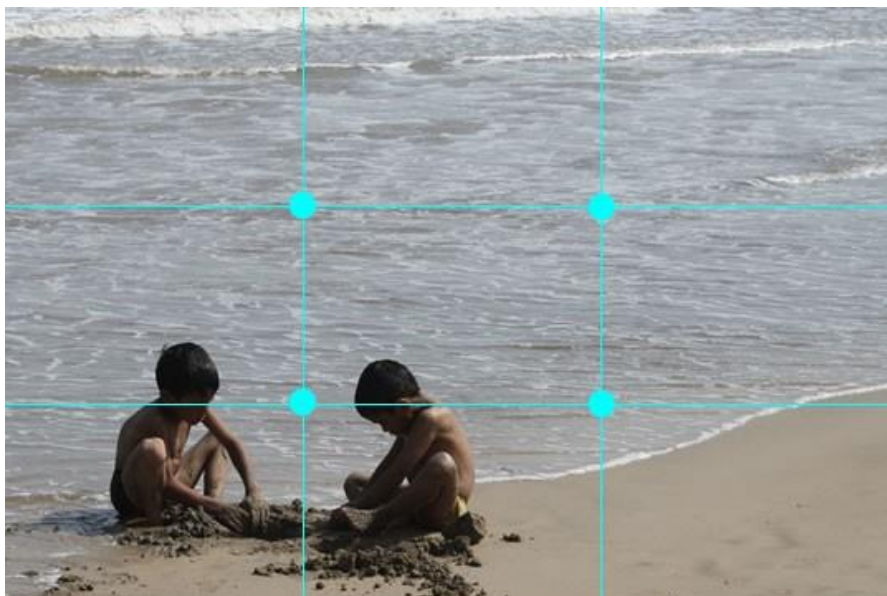
4.2.1 Proporción en los diseños

En el mundo del dibujo y del diseño las proporciones, tamaños y otros elementos están muy explorados y estos estudios se han trasladado al diseño web, de aplicaciones de escritorio (desktop) y aplicaciones móviles.

La proporción se utiliza para equilibrar y crear un punto de atención dentro de nuestro diseño, lo que se denomina punto focal, y que puede variar de un diseño a otro.

Dos de estos patrones más conocidos son:

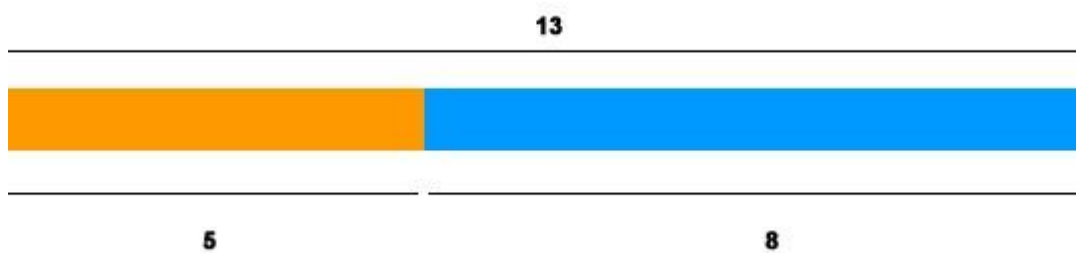
- **Regla de los tercios:** la escena se divide en una retícula básica de 3x3, donde hay unos puntos de intersección que se utilizan justamente para colocar los elementos importantes.



Ejemplo de la regla de los tercios.

Fuente: <http://www.ivoserrano.com/disenio-web/la-proporcion-en-el-diseno-web/>

- **Sección áurea:** esta es una proporción que está muy presente en la realidad y en la naturaleza y cuya aproximación es de 8:13. En la imagen vemos una simplificación de esta proporción, aunque hay muchos estudios al respecto y ejemplos de usos.



Proporción áurea.

Fuente: <http://www.ivoserrano.com/disenio-web/la-proporcion-en-el-diseno-web/>



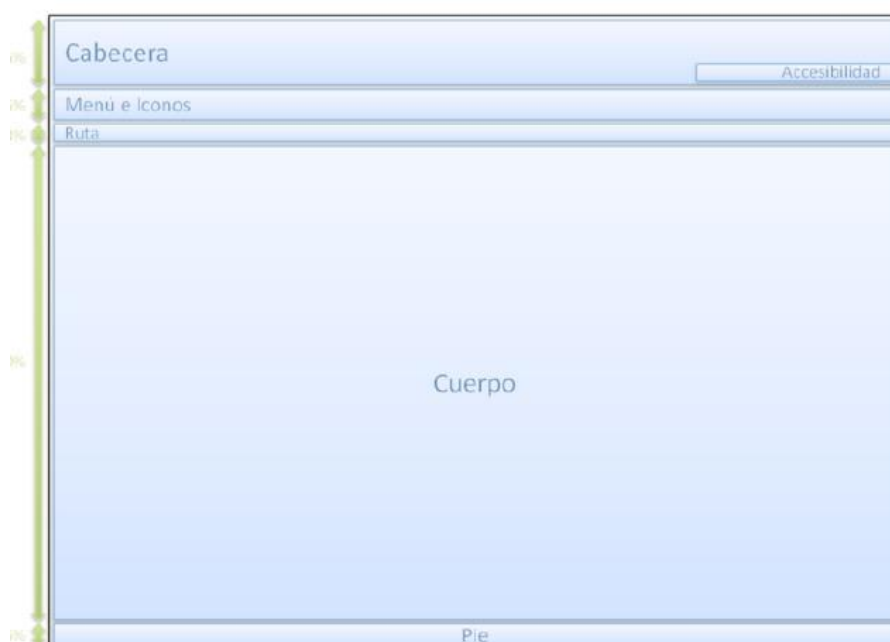
ENLACE DE INTERÉS

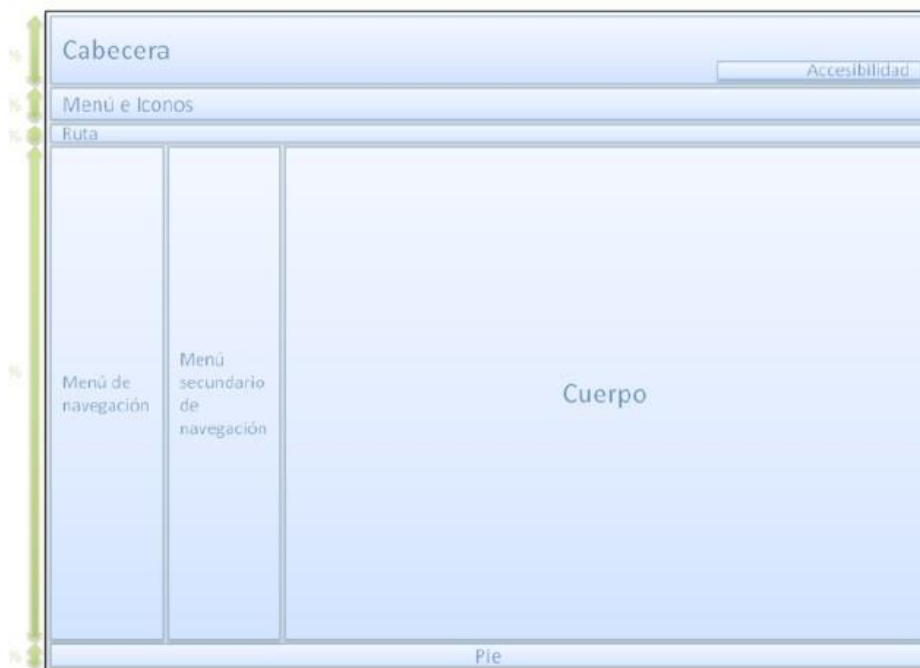
Profundiza sobre la proporción en el diseño web:



4.2.2 Ejemplo de interfaz desktop

A continuación, se muestran dos posibles estructuras de la interfaz principal con y sin menú de navegación en dos niveles a la izquierda:





Disposición del Layout en aplicación.

- **Cabecera de la aplicación:** para no desaprovechar espacio dedicado al contenido específico de la aplicación se deberá respetar un tamaño máximo de la cabecera. Por tanto, esta no debe superar el 15% del espacio total de la interfaz. La cabecera incluirá elementos corporativos, logotipos, imágenes, etc., respetando las normas establecidas en el manual de identidad corporativa de la organización.
- **Barra de menú e iconos:** esta barra no debe ocupar más del 3% del área total de la interfaz y debe estar visible en todas las pantallas, independientemente de la profundidad o nivel en el que se encuentre el usuario.
- **Cuerpo de la pantalla:** el cuerpo de la interfaz está dedicado a aportar la funcionalidad al usuario. Este ocupará al menos el 70% del área disponible en la misma.
- **Ruta de navegación:** la ruta de navegación indica al usuario donde se encuentra en cada momento y, por tanto, se debe colocar justo debajo de la barra de menú. El espacio dedicado a la misma debe ser inferior o igual al 3% del total.
- **Barra de accesibilidad:** la barra de accesibilidad debe permitir al usuario modificar, por ejemplo, el tamaño del texto, permitirle escuchar en vez de leer el contenido de la aplicación, etc. En caso de existir, estará incluida en la cabecera de la aplicación.

- **Menú de navegación:** el menú de navegación contendrá las opciones disponibles para el usuario, por lo que estas variarán en función del tipo de aplicación que se realice. Este se colocará a la izquierda del contenido y no superará el 25% del ancho de la pantalla:
 - Un menú principal que ocupará aproximadamente el 50% del total del área reservada para el menú de navegación y que contendrá acceso a las funcionalidades de la aplicación, estando siempre visible para facilitar la navegación.
 - Un menú secundario que variará en función del tipo de aplicación, ocupando el resto del espacio disponible. Este menú secundario puede ser en cascada o contextual.
- **Pie de la pantalla:** El pie de la interfaz está destinado a incluir información de contacto y no debe superar el 5% del área total.

4.2.3 Diseño de aplicaciones móviles

El móvil y todas sus variantes es quizá el dispositivo tecnológico donde más esfuerzos y estudios se han invertido dentro de la usabilidad y UX. Tenemos mucho material, ejemplos y frameworks que podemos usar justamente para diseñar nuestras aplicaciones móviles.

Algunos de los consejos muy extendidos sobre el diseño en aplicaciones móviles son:

- Tener en cuenta el tamaño de pantalla y proporciones. Mientras que el desktop supone un tamaño muy grande y adaptable, los dispositivos móviles (incluyendo tablets) tienen múltiples versiones de tamaños y proporciones que se deben tener en cuenta para el desarrollo e incluso con múltiples recursos gráficos.
- Orientación variable. En desktop la orientación es horizontal (excepto en soluciones muy específicas), mientras que en aplicaciones móviles es muy importante tener en cuenta tanto la orientación horizontal como vertical para el diseño.
- Recursos limitados.

Por lo tanto, en un proyecto móvil, la definición del layout es tremendamente importante para una correcta interfaz y experiencia de usuario.



ENLACE DE INTERÉS

Visualiza una recopilación de estudios y técnicas de diseño que Google ha desarrollado recientemente bajo Material You (Material 3), presentado en 2021 como la evolución más moderna de su sistema de diseño y que está incluida tanto en frameworks de diseño y desarrollo web, como desktop y móviles:

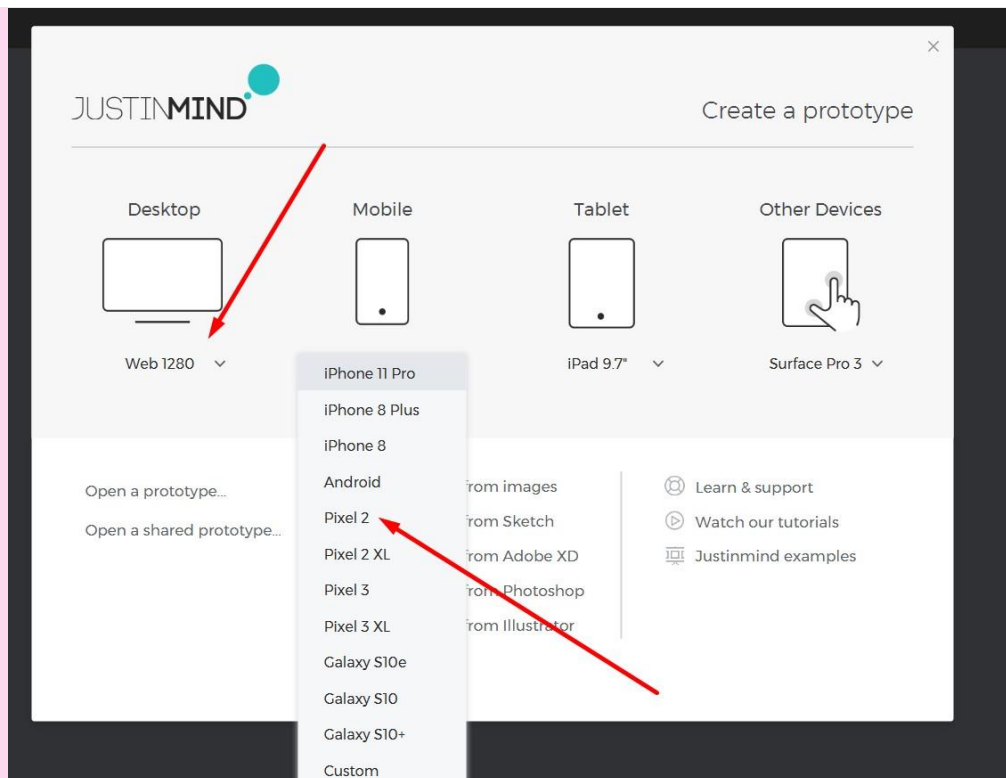


EJEMPLO PRÁCTICO

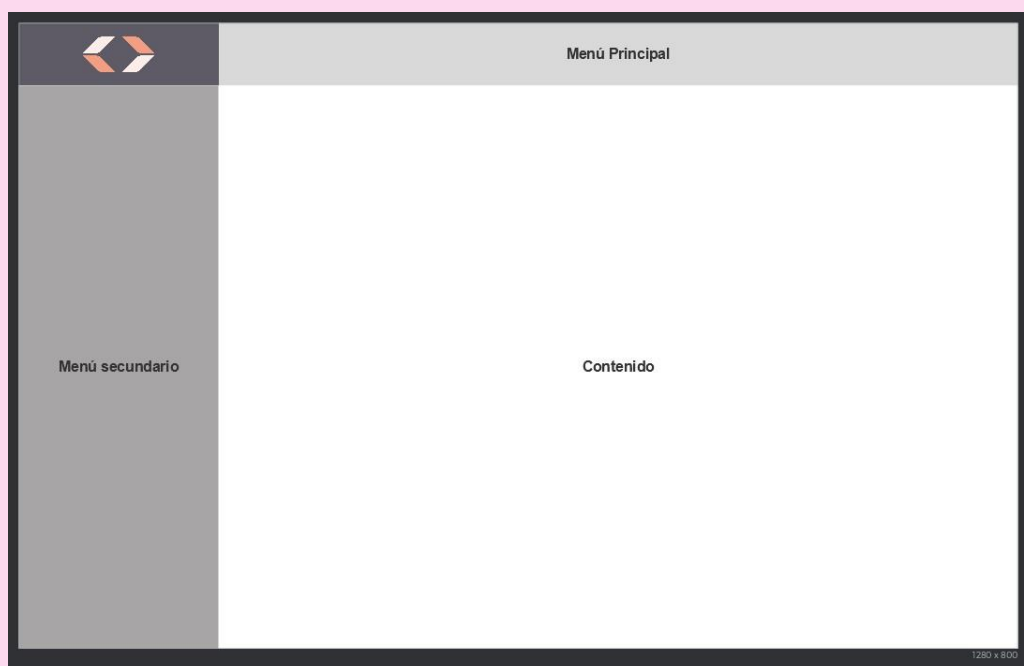
En la aplicación que vamos a realizar para el club de baloncesto, vamos a seguir trabajando con la pantalla de registro del socio, ya que va a ser una pantalla importante.

Esta pantalla nos servirá como modelo para el resto de pantallas. Teniendo en cuenta que solo necesitamos el layout de la pantalla, no los elementos, y que debemos tener tanto la pantalla para una visualización en desktop como en móvil, ¿qué herramienta debemos usar para realizar ese primer layout? ¿Cómo quedaría ese layout?

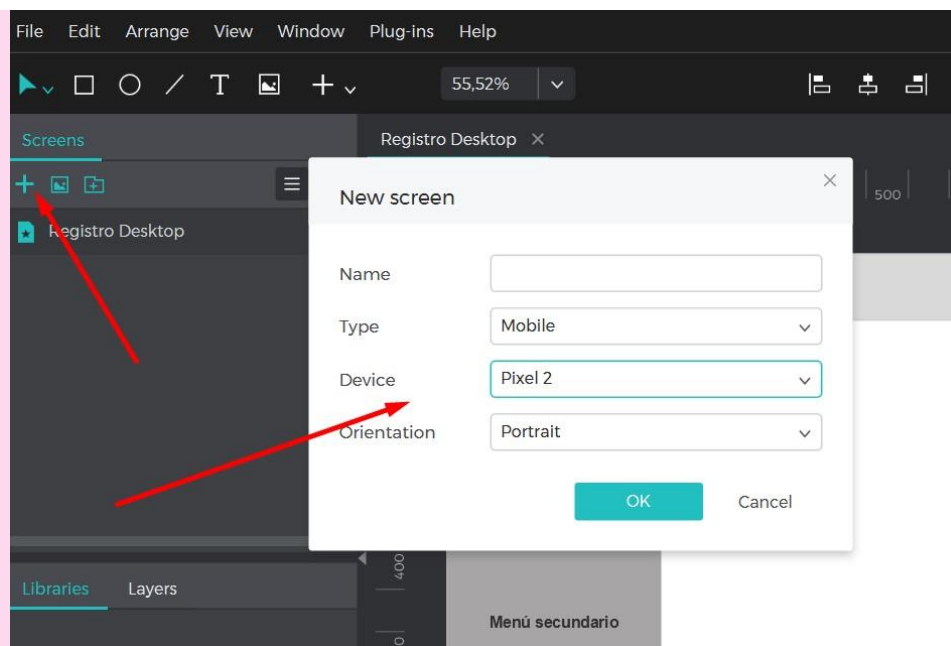
- Vamos a usar Justinmind para realizar el prototipado:
 - Primero realizaremos un prototipo para desktop/web 1280.
 - Segundo para un móvil tipo Android.



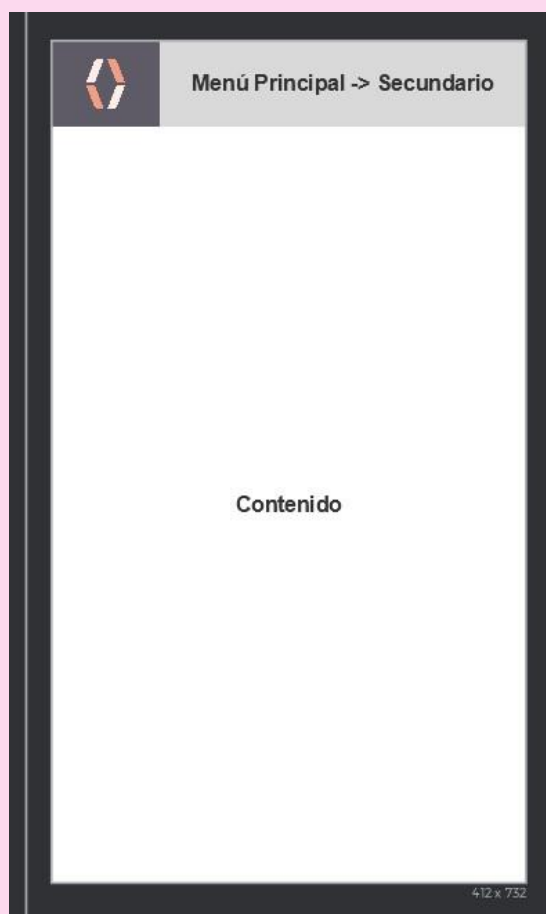
- En el caso del desktop tendremos el siguiente esqueleto:
 - Menú superior y principal que ocupará un 10% de la altura de la pantalla.
 - Menú secundario izquierdo que ocupará un 20% de la anchura.
 - Logo en la zona superior izquierda.
 - Contenido con el resto.



- Crearemos un segundo diseño para el móvil tipo:



- En el caso del móvil tendremos:
 - Menú superior que ocupará un 10% de la altura de la pantalla.
 - Logo en la zona izquierda superior.
 - Contenido con el resto.



4.3 Pautas de diseño del aspecto de la interface de usuario

El aspecto de las interfaces es tanto o más importante que la estructura y los otros elementos interactivos. Cuando se diseñan las interfaces de usuario es importante, como hemos visto, la estructura con componentes como los menús, las ventanas, los cuadros de diálogo y los atajos de teclado. Escoger los elementos y donde tienen que ir es una tarea importante.

En el diseño del aspecto de las interfaces de usuario nos tendremos que fijar en temas como los colores, las fuentes que se usarán para los diferentes apartados de la interfaz, los iconos y la distribución de los elementos.

Hay que tener en cuenta que para tratar todos estos aspectos no quedará clara la línea de separación entre el desarrollador de software o de aplicaciones informáticas y el diseñador. Muchas empresas delegan estas funciones a la misma persona, pero poco a poco esta tarea se empieza a diferenciar y se contrata una o más personas especializadas en diseño, específicamente para este apartado.

El diseño de la interfaz de usuario es un elemento muy importante (en algunos casos se podrá considerar crítico) en el producto final a librar. Debemos tener en cuenta que algunas decisiones serán muy personales, pero se tendrán que seguir unas indicaciones universales.

4.3.1 Colores

Los colores es lo primero que veremos cuando nuestra vista se fije en un objeto o, en nuestro caso, en una aplicación informática o una página web. Esta primera visión o interrelación de los usuarios con las interfaces es muy importante, prácticamente crítica. Un usuario no dejará de usar una interfaz por razones de discrepancias con los colores, pero, quizás, lo obligará a hacer cambios, si los puede pedir.

Las paletas de colores nos darán la opción de escoger entre una variedad muy amplia de colores para todos los elementos de la interfaz, tanto para el fondo como para los iconos y las letras. Incluso se podría escoger usar una fotografía de fondo.

Hay una serie de normas para la utilización efectiva del color en las interfaces gráficas de usuario. En caso de seguirlas, el diseño cromático de la interfaz será correcto desde un punto de vista cromático. Es muy difícil dar un conjunto de normas cerradas a causa de la gran cantidad de variables que influyen en la representación del color y en su percepción posterior.

Pautas de diseño

- Se debe utilizar el blanco como fondo de la interfaz para proporcionar una sensación de descanso para la vista y que sirva de relajación en los contrastes, haciendo que la navegación sea agradable y sin ruido visual.
- Se debe utilizar el gris oscuro para los textos en general, debido al alto contraste que ofrece sobre el fondo blanco.
- En cuanto a los colores principales se debe respetar el manual de identidad corporativa de la organización.



ENLACE DE INTERÉS

Comprueba esta guía de Color de Material Design, la cual te ayudará en la selección de los colores:



EJEMPLO PRÁCTICO

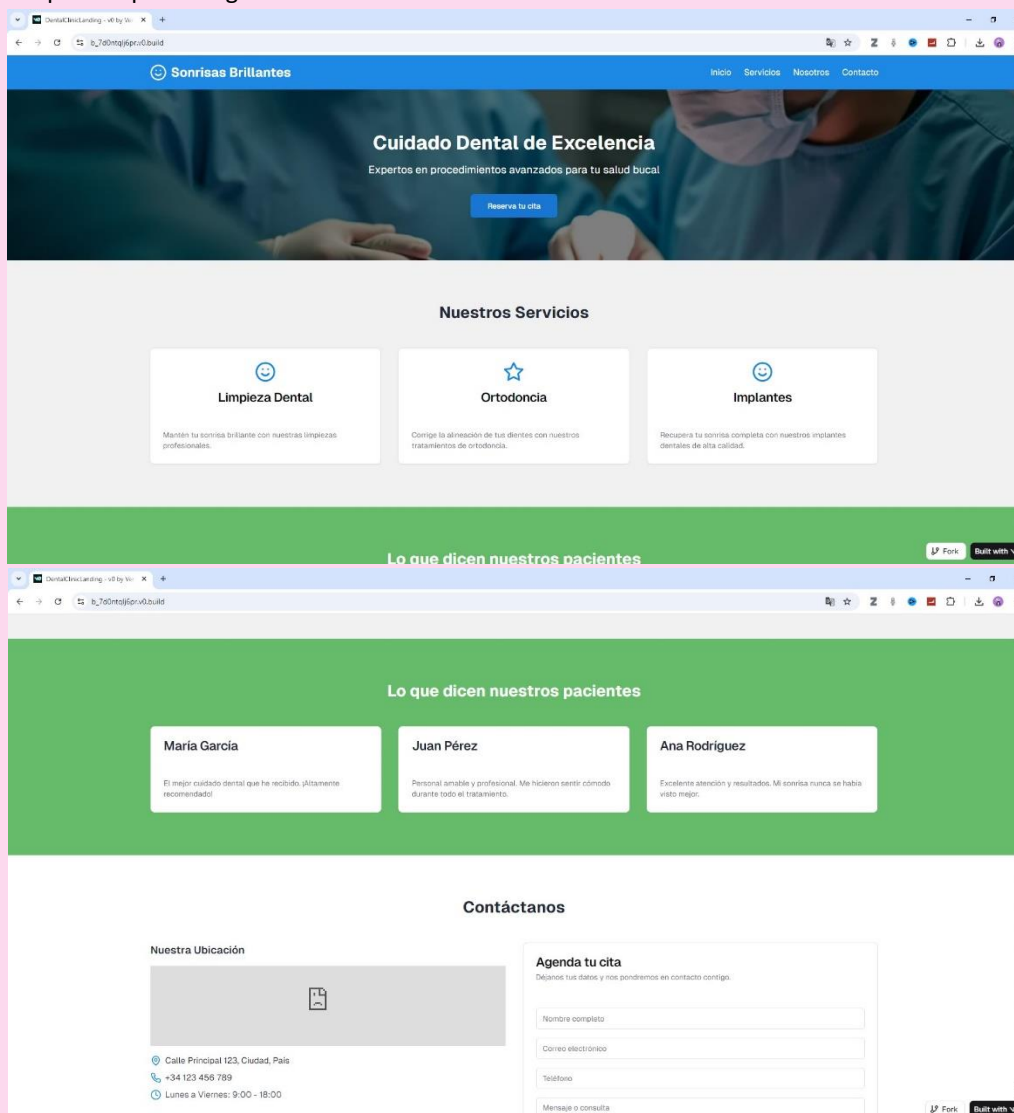
Tu jefe te ha solicitado que elijas la paleta de colores para una nueva página web que se va a desarrollar para una clínica dental siguiendo las pautas de diseño. La apariencia visual de la web debe transmitir confianza y profesionalidad, valores fundamentales en las clínicas de salud. Debes seleccionar una paleta de colores adecuada para la página web de la clínica dental, teniendo en cuenta el significado y el impacto psicológico de los colores en los usuarios.

Vamos a realizar el análisis:

- Colores Principales:
 - Azul: Es un color que se asocia con la calma, la confianza y la limpieza. Perfecto para reflejar profesionalidad y transmitir una sensación de seguridad y fiabilidad.
 - Blanco: El color blanco está relacionado con la pureza, la simplicidad y la limpieza. Se utilizará como color de fondo para las áreas de contenido, proporcionando una apariencia limpia y ordenada.
- Colores Secundarios:

- Verde: Representa salud, bienestar y equilibrio. Se usará en detalles como iconos o secciones que resalten información sobre los servicios de la clínica o testimonios de pacientes.
- Gris: Proporciona un contraste sutil y elegante, ideal para textos y secciones que requieran diferenciación sin saturar la vista. Se empleará para textos secundarios y descripciones.
- Colores de Resalte:
 - Amarillo Suave: Para destacar avisos importantes o promociones. Este color, en un tono suave, no altera la armonía general de la paleta, pero logra captar la atención de los usuarios.

Al final podría quedar algo así:



Clínica Dental
Fuente: Elaboración Propia

4.3.2 Fuentes

Hay muchos tipos de fuentes y muchas posibilidades de usarlas, pero hay que tener cuidado de que estas fuentes estén incorporadas al sistema operativo o que no sean compatibles, cosa que puede hacer que no se puedan usar de manera adecuada.

Además, como se ha hecho dicho anteriormente, habrá que utilizar fuentes llamadas básicas porque sean aceptadas por cualquier entorno de trabajo o sistema operativo. En caso de usar una fuente no básica, habrá que empaquetarla con la aplicación desarrollada e instalarla con el software de instalación.

La tipografía es la apariencia que tiene el texto y es el elemento de diseño gráfico más básico de una interfaz.

Pautas de diseño

- Seleccionar un tipo de letra sencillo que no provoque cansancio.
- Escoger un tamaño suficiente (el usuario siempre podrá cambiarlo).
- Usar un color con suficiente contraste.
- Escribir preferentemente en minúsculas (facilita la lectura).
- Emplear adecuadamente los recursos tipográficos (poner los títulos).



ENLACE DE INTERÉS

Aquí podrás profundizar sobre Material Design de Google, donde nos encontramos la herramienta de elección de tipografía:



4.3.3 Iconos

Los iconos que se usan en las interfaces también son importantes en la interacción con los usuarios. Ya hemos hablado de algunos iconos en el apartado de los menús, cuando hacíamos referencia al menú de barras de herramientas. Hablaremos del mismo tipo de

iconos con las mismas funcionalidades, pero en este apartado nos fijaremos más en los iconos de acción de las interfaces de usuario.

Los iconos ofrecen la representación de una unidad de significado. Si esta unidad de significado se quisiera expresar con texto, ocuparía mucho más espacio del que se ocupa usando un icono.

Pautas de diseño
<ul style="list-style-type: none">• Los iconos que se utilicen en diferentes elementos deben tener formato png y un tamaño adecuado de entre los siguientes:<ul style="list-style-type: none">○ 8x8 píxeles○ 12x12 píxeles○ 16x16 píxeles○ 24x24 píxeles

4.3.4 Distribución de los elementos

Llegados a este punto, se ha hablado de diferentes conceptos importantes en el aspecto de una interfaz de usuario como pueden ser los colores, las fuentes y los iconos. En el apartado anterior se han visto diferentes elementos de la estructura de las interfaces como los menús, las ventanas y los cuadros de diálogo. Ahora toca hablar de cómo se distribuyen estos elementos por la interfaz.

La distribución de los elementos tiene que permitir un uso de las interfaces y un aprendizaje óptimo de estas a los usuarios. Hay que decidir la manera de presentar los menús de texto, las barras de herramientas, los iconos, los cuadros de diálogo, etc., y su ubicación en la interfaz. Los elementos tienen que presentar una distribución uniforme teniendo en cuenta la agrupación de funcionalidades relacionadas y las zonas de la pantalla que los usuarios visualizarán más.

Los objetivos de la distribución de los elementos por la interfaz son los siguientes:

- Conseguir que la interfaz sea fácil de usar.
- Fácil de aprender.
- Segura.
- Fiable y efectiva a la hora de llevar a cabo las acciones necesarias para una aplicación.
- Tiene que ser consistente.

Es difícil dar una norma y pauta general, ya que depende mucho de la aplicación y del uso de la misma. Podemos, sin embargo, dar dos posibles buenas prácticas:

- Es interesante adaptar la interfaz desarrollada en el entorno de trabajo de esta, distribuyendo los elementos de manera análoga a otras aplicaciones o interfaces del sistema operativo, teniendo en cuenta también el dispositivo sobre el que se va a ejecutar la aplicación.
- Todos los componentes gráficos de una interface de una aplicación informática (menús, áreas de datos, iconos, diálogos, tableros o imágenes) están dispuestos encima de una retícula imaginaria que es respetada a lo largo de la aplicación (los menús no cambian de lugar ni tampoco lo hace el área de visualización de datos, por ejemplo).

4.4 Pautas de diseño del aspecto en los elementos interactivos

Hay otros muchos elementos que se usan en las interfaces de usuario, además de los que hemos comentado hasta ahora. Tanto los menús como las ventanas, los cuadros de diálogo o los iconos son elementos que facilitan la interacción entre el usuario y la interfaz.

Estos elementos permiten la navegación por la aplicación, muestran las alternativas que tiene el usuario y le ofrecen la posibilidad de ejecutar algunas acciones.

Algunos de estos elementos son:

- **Botones:** los botones nos sirven para construir las interfaces y actúan como interpretaciones visuales de las funcionalidades que representan. Los botones son parte, como los iconos, de la gramática visual de las interfaces de usuario.
- **Radio Buttons:** los botones en forma de radio o botones de opción ofrecen al usuario una lista de opciones que tienen a la izquierda o la derecha (menos habitual) una pequeña redonda para poder clicar adentro.
- **Checkbox:** estos botones son muy parecidos a los anteriores, pero con la particularidad de que, a la izquierda o la derecha del texto indicativo de las opciones, están representados con pequeños recuadros. En caso de seleccionar la opción correspondiente a un recuadro, este se llenará con un símbolo de check.
- **Listas desplegables:** otros elementos interactivos de las interfaces de usuario son las listas desplegables, conocidas en inglés como combo box. Una lista

desplegable es una combinación de un cuadro de texto (texto box) y un menú desplegable (list box).

- **Cuadros de diálogo:** los cuadros de diálogo son ventanas emergentes que interrumpen temporalmente el flujo del usuario para solicitar una acción, confirmación o información. Su diseño debe ser claro y evitar sobrecargar al usuario con opciones innecesarias. Los cuadros de diálogo deben tener mensajes breves y precisos, con botones claramente diferenciados para aceptar o cancelar. Además, se deben incluir opciones para cerrarlos fácilmente (por ejemplo, presionando "Esc").

Pautas de diseño	
Botones	<ul style="list-style-type: none"> • El nombre del botón debe ser un verbo en infinitivo en combinación con un adjetivo si se requiere. Por ejemplo, actualizar, guardar como, etc. • Debemos usar una o dos, como máximo, anchuras diferentes para botones en una misma ventana.
Radio Buttons	<ul style="list-style-type: none"> • Se utiliza cuando se dispone de espacio de pantalla; en caso contrario, usar listas desplegables • Usar grup box para identificar a qué categoría nos estamos refiriendo. • Se recomienda tener una de las opciones seleccionadas por defecto. • Las opciones deben estar claras.
Checkbox	<ul style="list-style-type: none"> • Las mismas pautas que con los Radio Buttons. • Usar un Radio Button solo para selecciones múltiples.
Listas desplegables	<ul style="list-style-type: none"> • Organizar la información a desplegar para no llevar a equívoco al usuario, bien organización por categorías, alfabética u otra. • Añadir un cuadro de texto buscador siempre que exista espacio. • Indicar claramente si la selección es múltiple o no.
Cuadro de diálogo	<ul style="list-style-type: none"> • Utilizar mensajes claros y concisos que indiquen la acción requerida. • Los botones de confirmación y cancelación deben ser fácilmente identificables y accesibles. • Incluir atajos para cerrar los cuadros de diálogo, como el uso de "Esc" para cerrar sin confirmar. • Evitar sobrecargar el cuadro con demasiadas opciones o información.



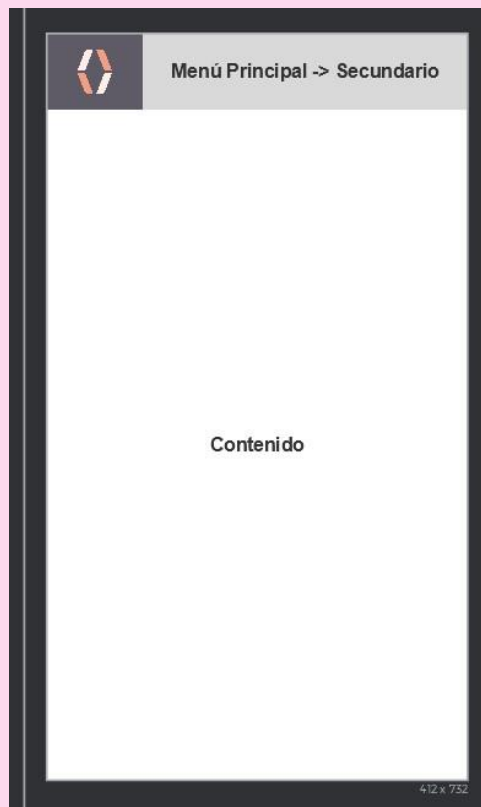
EJEMPLO PRÁCTICO

En la aplicación que vamos a realizar para el club de baloncesto, queremos completar el diseño de la página de registro de nuevo socio incluyendo los elementos necesarios dentro de nuestro mockup para reflejar:

- Nombre.
- Apellidos.
- Correo electrónico.
- Password.
- Rol (Tutor/Jugador/Entrenador).

Nos centraremos en el diseño para móvil. ¿Qué elementos incluiríamos en nuestro diseño?

- Teniendo el siguiente esqueleto con JustinMind para el registro móvil.



- Incluiremos:
 - Un label y cuadro de texto para los campos de Nombre, Apellidos y Correo electrónico.
 - Un label y password para el campo Contraseña.
 - Una lista desplegable para el rol.
 - Un botón de registro.

Menú Principal -> Secundario

Nombre

Apellidos

Correo

Contraseña

Repite contraseña

Jugador

REGISTRA

412 x 732

4.5 Pautas de diseño en la presentación de datos

Los datos que se mostrarán usando las interfaces de una aplicación tienen que seguir unas pautas concretas para no confundir a los usuarios. Según qué información se quiere presentar se debe tener acceso a un apoyo externo que contenga los datos o bien se tendrán los datos incorporados en nuestra interfaz.

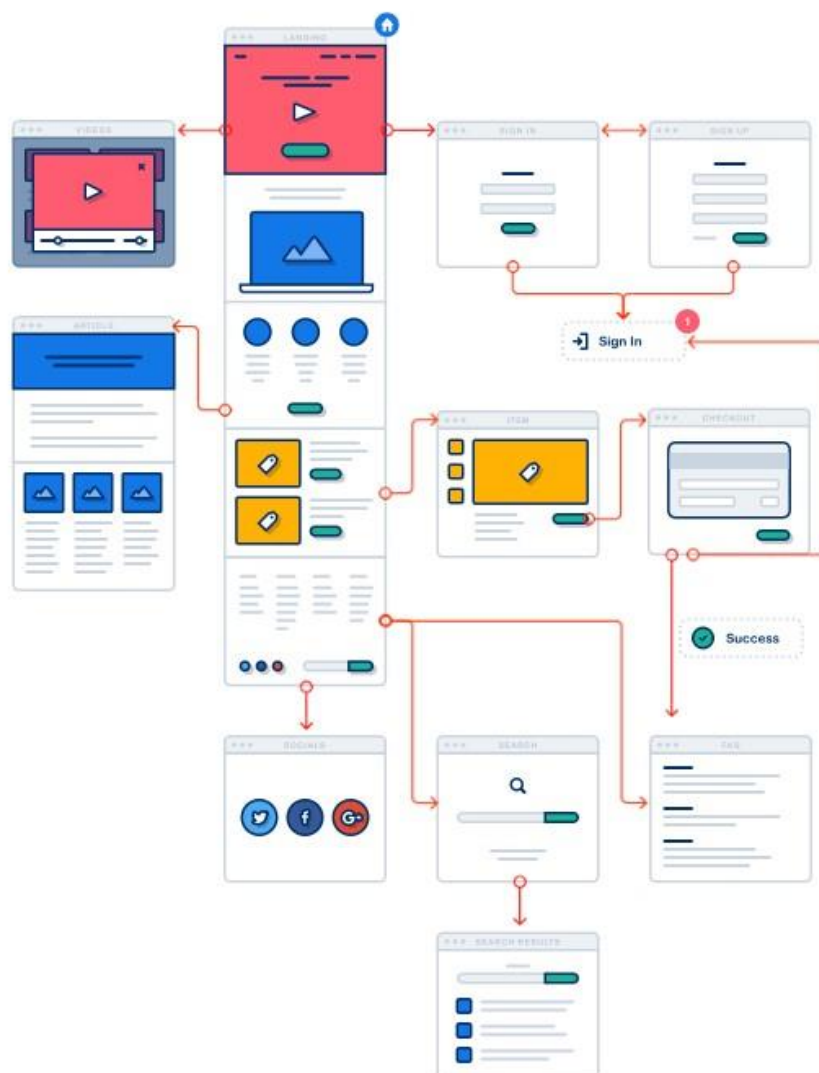
Cuando se diseña la presentación de los datos, debemos tener en cuenta algunas cuestiones:

- ¿Qué tipo de información quiere ver el usuario? ¿Quiere tener acceso a la información de gestión o quiere tener acceso a información estadística o de resumen para poder tomar decisiones?
- ¿Qué uso querrá hacer? ¿Querrá solo consultar o tendrá que poder modificar, borrar y crear nueva información?
- ¿Si los datos se modifican de manera continua, el usuario tiene que ver reflejados estos cambios en el mismo momento?

- ¿La información a mostrar será numérica o en modo texto?

4.6 Pautas de diseño en el control de la aplicación

La secuencia de control de la aplicación se refiere a las acciones que el usuario llevará a cabo con el software y la lógica de ejecución de este. Es decir, si el usuario pide ejecutar una operación que comporta varias acciones con la base de datos, tendremos que establecer un control de la secuencia de operaciones que se llevarán a cabo para garantizar que la información se ha mantenido consistente en la base de datos.



Flujo de la aplicación o mockflow.

Fuente: <https://ziuxlab.com/disenio-interfaces-web>

Toda aplicación con una interfaz de usuario ofrece muchos elementos y muchas funcionalidades a los usuarios. Estos elementos nos llevarán a poder llevar a cabo muchas operaciones, la mayoría de las cuales trabajarán con datos. Algunas de estas operaciones serán directas y se ejecutarán de manera inmediata. A otras operaciones

podremos llegar desde diferentes ubicaciones de la interfaz. Otras operaciones serán compuestas y comportarán más de un acceso o una modificación a la base de datos.

Un buen diseño de una interface (y de una aplicación en general) tiene que incorporar unos objetivos que incluyan acciones de control de consistencia, una necesidad mínima de acciones de control y flexibilidad en los controles de secuencia para adaptarse a las diferentes necesidades de los usuarios. En resumen, las acciones de control deben cumplir las siguientes pautas:

- **Consistencia en la navegación:** el flujo debe ser predecible, los usuarios deben poder anticipar los siguientes pasos basados en su experiencia previa en la misma aplicación utilizando patrones comunes de navegación (botones de siguiente, anterior, guardar y cancelar) de manera uniforme en todas las pantallas.
- **Retroalimentación clara y oportuna:** cada acción del usuario debe recibir una respuesta inmediata del sistema. Por ejemplo, tras hacer clic en un botón, mostrar un mensaje de carga o confirmar que la acción se realizó correctamente (ejemplo: "Guardado exitoso"). Asegúrate de que los estados de los botones reflejen la acción disponible (activo/inactivo).
- **Control de errores:** la aplicación debe prever posibles errores del usuario y ofrecer soluciones o mensajes claros que ayuden a resolver el problema sin interrumpir el flujo (por ejemplo, si falta completar un campo obligatorio, indicar con claridad cuál es y cómo corregirlo). Siempre ofrece al usuario la opción de deshacer o cancelar una acción en caso de errores.
- **Secuencias lógicas y jerárquicas:** diseña las pantallas en una secuencia lógica que guíe al usuario paso a paso en tareas complejas (por ejemplo, primero completar información básica, luego pasar a configuraciones más avanzadas). El usuario no debe poder saltarse pasos críticos sin completarlos.
- **Minimización de pasos innecesarios:** evita pantallas o interacciones que no aporten valor. Cada paso en la secuencia debe ser esencial y claramente relacionado con el objetivo final. Agrupa tareas similares o relacionadas en una misma pantalla si es posible, en lugar de obligar al usuario a cambiar constantemente entre pantallas.
- **Accesibilidad y atajos:** proporciona opciones de atajos (por ejemplo, atajos de teclado) para los usuarios avanzados que quieran realizar acciones rápidamente sin tener que navegar por todo el sistema. Incluye opciones para realizar saltos

entre diferentes partes del flujo de manera segura, como botones para volver al inicio o ir a una pantalla anterior sin perder los datos.

- **Validación y confirmación de datos:** en procesos que involucren múltiples pasos (por ejemplo, formularios largos), asegúrate de validar la información ingresada antes de permitir que el usuario continúe con el siguiente paso. Ofrece una vista de confirmación o un resumen antes de realizar una acción irreversible, como enviar un formulario o confirmar una compra.

4.7 Pautas de diseño en los atajos de teclado

Los atajos de teclado son combinaciones de teclas con las que realizamos acciones de manera rápida, sin necesidad de usar el ratón. Además de mejorar la velocidad de interacción, mejora la accesibilidad, ya que permiten a usuarios con limitaciones físicas o visuales ejecutar tareas de manera más cómoda y sin depender del uso del ratón.

En el diseño de nuestras aplicaciones deberíamos incluir al menos los siguientes atajos de teclado:

- **Ctrl + C (Copiar):** permite copiar el texto o elemento seleccionado al portapapeles, evitando la necesidad de usar el menú contextual.
- **Ctrl + V (Pegar):** pega el contenido copiado previamente en el portapapeles en la ubicación deseada.
- **Ctrl + X (Cortar):** similar a copiar, pero elimina el contenido original tras copiarlo al portapapeles.
- **Ctrl + Z (Deshacer):** revierte la última acción realizada, útil para corregir errores rápidamente.
- **Ctrl + Y (Rehacer):** reaplica la última acción deshecha, facilitando correcciones rápidas.
- **Ctrl + S (Guardar):** guarda el documento o el trabajo actual, indispensable para no perder el progreso.
- **Ctrl + P (Imprimir):** abre directamente la opción de imprimir el contenido actual, sin necesidad de navegar por menús.

Además, algunas aplicaciones permiten que los usuarios configuren sus propios atajos de teclado, lo que les da la libertad de adaptar las combinaciones a sus necesidades. Esto puede hacer que su experiencia sea más cómoda, sobre todo para quienes realizan ciertas tareas de forma repetitiva.

4.8 Pautas de diseño en el aseguramiento de la información

La información con que trabajaremos estará en una base de datos. La introducción, modificación o borrado de los datos se hará por medio de una interfaz. Una vez esta se ha usado y los datos están estables en la base de datos, esta tiene que estar disponible para cuando los usuarios la necesiten y tiene que estar asegurada.

Para conseguir esto hay que establecer un proceso de mantenimiento y una utilización que no comprometa la integridad. Además, se pueden usar otros sistemas para asegurar la disponibilidad de la información, lo cual minimizaría el riesgo de quiebra de los sistemas informáticos. Estos pueden ser sistemas de reserva o de emergencia que hagan copias de seguridad de manera automática y regular.

Un método para asegurar la integridad de la información consiste en establecer sistemas para evitar el acceso directo en las bases de datos por parte de usuarios no controlados, en caso de trabajar con una información crítica para las organizaciones. Estos sistemas pueden incluir un sistema de permisos para perfiles de usuarios o unas contraseñas. Con este sistema, solo un número determinado de usuarios tendrá permiso para modificar los datos.

4.9 Pautas de diseño en aplicaciones multimedia

En función del tipo de interfaces que se desarrollen o del tipo de aplicación o el entorno para el cual se plantee el uso de la aplicación, nos podemos encontrar con la necesidad de usar algunos elementos multimedia específicos como, por ejemplo, la animación, sonido, imágenes, etc.

De forma general podemos afirmar que el uso de elementos en las interfaces mejora la retención de la información. Ahora bien, debemos mantener un equilibrio y unos criterios siempre y cuando el objetivo no sea únicamente presentar dichos elementos.

Pautas de diseño	
Animaciones	<ul style="list-style-type: none">• Las imágenes en movimiento tienen el efecto de atraer al ojo humano, usado como efecto de atracción es correcto, como decoración hace que el usuario pierda el foco.• No se debe incluir una animación continua, ya que para el usuario es difícil concentrarse.• Debe usarse principalmente para atraer la atención.
Vídeo	<ul style="list-style-type: none">• El vídeo permite explicar y dar información de una forma más efectiva.

	<ul style="list-style-type: none"> • El consumo de ancho de banda es importante. Por lo tanto, se debe usar con el tamaño y calidad adecuados.
Audio	<ul style="list-style-type: none"> • El audio puede usarse de forma paralela al contenido visual. • Para no perder el foco del contenido del audio, también se puede usar ofuscando el texto. • Es recomendable usar correctamente el nivel de sonido del audio. • Es un elemento imprescindible para proporcionar accesibilidad.
Imágenes	<ul style="list-style-type: none"> • Se debe utilizar un número reducido de imágenes para que la aplicación sea lo más rápida y ágil posible. • Se recomiendan imágenes que contribuyan a hacer entender contenidos basados en texto y a hacer más atractivos textos extensos. • Es importante encontrar la calidad y la proporcionalidad adecuada para las imágenes.



VÍDEO DE INTERÉS

Visualizando el vídeo conocerás más sobre las leyes de UX y usabilidad, especialmente del minuto 2 al 12:



RESUMEN FINAL

En el desarrollo de software, las interfaces de desarrollo se han convertido en pieza clave para el éxito de una aplicación, ya que es el mecanismo de comunicación del usuario con los datos y la presentación de los mismos.

Por lo tanto, la usabilidad y los conceptos que la rodean se han convertido en esenciales para comprender cómo ese usuario se relaciona con nuestra aplicación y cómo la utiliza. Términos y ámbitos como la experiencia de usuario y experiencia de cliente proporcionan ese cambio de foco en el usuario.

Otro aspecto a analizar es la accesibilidad, que garantiza que una aplicación sea utilizable por todos, incluidas personas con discapacidades físicas o cognitivas. Siguiendo pautas como las establecidas por las normas ISO o las WCAG, se asegura que las aplicaciones sean perceptibles, operables, comprensibles y robustas. Esto no solo mejora la experiencia de las personas con discapacidades, sino que también optimiza la usabilidad para todos los usuarios. La accesibilidad ha pasado de ser un extra a un requisito en el diseño de interfaces modernas.

De esta manera, para diseñar una aplicación, necesitaremos conocer, por un lado, al usuario mediante test de usabilidad/accesibilidad y pruebas y, por otro lado, conocer diferentes buenas prácticas y pautas de diseño que hagan que nuestra aplicación alcance buenos resultados en efectividad, satisfacción, apariencia, errores, entre otros.

Hemos explorado la importancia de los esquemas (wireframes) y las maquetas (mockups) en el proceso de diseño de interfaces. Los wireframes nos permiten establecer la estructura y funcionalidad básica de nuestra aplicación, facilitando la comunicación y colaboración en etapas tempranas del proyecto. Por otro lado, los mockups nos ayudan a visualizar el diseño final, incorporando aspectos estéticos y brindando una representación más realista de cómo se verá la aplicación.

Todos estos diseños deben hacer en base a unas pautas, estas pautas son muchas y van evolucionando y particularizándose para cada una de las aplicaciones, además de aparecer nuevas herramientas y frameworks que nos ayudan en el proceso del diseño, como Material Design de Google.

Por último, la irrupción de los dispositivos móviles ha dado un vuelco a toda esta ciencia y conceptos pues, por un lado, el espacio es reducido y, por otro, la interacción con el usuario es máxima. De esta forma, cuando diseñamos una aplicación debemos pensar en múltiples usuarios y tipos de pantallas para alcanzar lo que se denomina una buena usabilidad de aplicación.