

# Guía conceptual sobre el diseño de interfaces usables y centradas en el usuario

## Conceptual guide on the design of user-friendly and user-centered interfaces

Carlos Pinto, Ramith Piñeros, Kevin Rodríguez y Eduardo Vizcaíno

*Universidad Simón Bolívar, Barranquilla-Colombia*

**Resumen** | En este documento se destaca la importancia del diseño de interfaces de usuario adaptativas a múltiples plataformas a través de métodos, técnicas o patrones integradas desde el inicio al proceso de desarrollo de software. Por medio de la revisión y evaluación del panorama actual en cuanto a desarrollo de interfaces se evidenció, que por lo general se diseña para un dispositivo en particular dificultando la construcción de interfaces usables desde el punto de vista de la experiencia de usuario y limitando la interacción entre humano-computador. Además se sentaron las bases para la construcción de una guía de recomendaciones que permite el diseño de interfaces adaptativas al aplicar técnicas, patrones y definir así un enfoque conceptual que ayude a crear un proceso de diseño más ágil.

**Palabras clave:** Software, Ingeniería de Software, Interfaz de Usuario, Patrón de Diseño, Usabilidad.

**Abstract** | This document highlights the importance of designing adaptive user interfaces to multiple platforms through methods, techniques or integrated patterns from the beginning to the software development process. Through the review and evaluation of the current scenario in terms of interface development, it was evident that it is usually designed for a particular device, making it difficult to build user-friendly interfaces from the point of view of the user experience and limiting the interaction between human-computer. In addition, the foundations were laid for the construction of a guide of recommendations that allows the design of adaptive interfaces by applying techniques, patterns and thus define a conceptual approach that helps to create a more agile design process.

**Key words:** Software, Software Engineering, User Interface, Design Pattern, Usability.

### Introducción

El desarrollo de software es un proceso formal que involucra la interacción entre los usuarios, desarrolladores y las herramientas utilizadas para la creación de aplicaciones y es ahí donde la ingeniería de software actúa como disciplina para establecer los mecanismos que a través de modelos, patrones o metodologías, deberán aplicarse de forma estructurada y con técnicas científicas. De la misma manera el desarrollo de interfaces de usuarios se convierte en una ingeniería al utilizar, para su realización, precisamente ese conjunto de técnicas para el diseño y desarrollo de interfaces que permitan la creación de software interactivo y usable [1].

El proceso de desarrollo de software y por ende el diseño de interfaces se encuentra en constante evolución gracias a los avances acelerados de la tecnología y las nuevas formas con que cuentan los usuarios para acceder e interactuar con un sistema de información. Los patrones de diseño de interfaz

de usuario son soluciones para facilitar la construcción de interfaces hombre-máquina a través de una interfaz gráfica y aunque estos patrones han adquirido importancia, a su vez se han configurado como uno de los aspectos críticos en la ingeniería del software y por ello, es necesario hacer un análisis de lo disponible hoy en día que permita identificar cuáles son los más apropiados para garantizar la satisfactoria experiencia de usuario e ir más allá de una simple interfaz amigable.

Por estas razones, en el presente artículo se presenta una síntesis sobre los aspectos conceptuales del diseño de interfaces y de cómo adaptar aquellos procesos de diseño que garantizan la usabilidad y experiencia de usuario en una aplicación. Para ello se hace un análisis de los diferentes enfoques o modelos del estudio de interfaces, identificando cuáles son los más aplicados de acuerdo al concepto de usabilidad; además se explica cómo es el proceso de diseño de una interfaz desde el punto de vista de la ingeniería del software y se realiza un análisis comparativo de los diferentes patrones de diseño de acuerdo a las metodologías actuales identificando las herramientas y técnicas

disponibles para su implementación y desarrollo. Por último se determinan los atributos del concepto de usabilidad de una interfaz y los enfoques y técnicas para la evaluación de la experiencia de usuario para luego concluir con una guía de recomendaciones como aportes a un proceso de diseño centrado en el usuario.

### **Caso de estudio**

La búsqueda constante de nuevas interfaces de usuarios está asociada a los avances tecnológicos, al rol social de la computación y se debe a que cada día se incrementa el número de usuarios con diferente formación, nivel social y habilidades [2]. Cada vez hay más información cuya complejidad va en crecimiento, cada vez es necesario ofrecer más servicios de diferente índole y cada vez se dispone de nuevas y poderosas tecnologías, las cuales avanzan de una manera vertiginosa. Todo ello establece un gran desafío para la Interacción Humano-Computador, la cual debe mantenerse a la par de este avance y sacar el mejor provecho de la tecnología para ponerlo a disposición de los usuarios.

La adaptación de interfaces de usuarios UI (del Inglés User Interface) a diversos contextos o plataformas se ha convertido en un verdadero desafío para el desarrollo de aplicaciones desde la ingeniería de software, si tenemos en cuenta la creciente aparición de dispositivos y tecnologías diferentes y que el soporte ofrecido por los actuales ambientes de desarrollo de software adaptativo es limitado. [3]

Siendo indiscutible el acelerado aumento de diferentes dispositivos y tecnologías de usuario final el reto en el diseño de interfaces de usuario es plantear un diseño adaptado a todos estos aparatos que vaya más allá de la clásica pantalla de computador y con la capacidad de ser usados en cualquier lugar. [4]

Por estas razones consideramos pertinente realizar una exhaustiva comparación entre todos estos patrones de diseños evaluados a fin de realizar una guía de recomendaciones que sirva de base para la adaptación de todo el proceso de diseño de interfaces de usuarios desde la ingeniería de software donde se apliquen aquellas metodologías y patrones de diseño que logren mantener la experiencia de usuario igual de satisfactoria indistintamente del dispositivo desde el cual se accede a una aplicación.

### **Estado del arte**

#### **Patrones de Diseño de Interfaz de Usuario**

Los patrones de diseño de interfaz de usuario se utilizan para facilitar el desarrollo de aplicaciones, porque las aplicaciones usan una serie de patrones que las hacen similares. Por lo tanto se analizaron diversos patrones de

diseño, para diferentes dispositivos y plataformas. En la actualidad los sistemas operativos para dispositivos móviles más populares son Android, iOS, y Windows Phone. Cada sistema operativo tiene su propia identidad que se refleja en la apariencia y comportamiento de cada uno de los elementos gráficos. Sin embargo, todos comparten puntos de vista fundamentales que se manifiestan en el diseño de sus interfaces, como lo es la navegación, los cuadros de diálogo, notificaciones, entre otros. Se identificaron diferentes categorías de patrones de diseño de interfaz de usuario propuestos por diversos autores, para los dispositivos móviles se encuentran los planteados por Theresa Neil [5], Jennifer Tidwell [6], Mari Sheibley [7], Anders Toxboe [8] y UNITiD [9].

En diversas investigaciones y trabajos se menciona la importancia de utilizar patrones de diseño de interfaz de usuario. M. Nabuco et al [10] presentó un trabajo sobre la relevancia de utilizar los patrones de diseño de interfaz de usuario, debido a los problemas que enfrentan los desarrolladores al usarlos, sin embargo no se ocupan y esto provoca una pérdida de productividad y calidad. S. Kim [11] mencionó que existen muchas actividades de investigación para la ingeniería basada en modelos de interfaces de usuario para múltiples dispositivos. Se identificaron tres principales limitaciones para los enfoques convencionales y propuso un

marco de trabajo llamado PELUM (Pattern and Event based Logical UI Modeling, Patrones y Eventos Basados en Modelado de la Interfaz de Usuario) para modelar interfaces de usuario específicas para múltiples sistemas embebidos.

S. Kim [11] también afirma que el apoyo ofrecido por los entornos actuales de desarrollo para adaptar la interfaz de usuario es limitado, así como la creación de interfaz de usuario en aplicaciones eficientes basadas en servicios Web. K. Klemisch [12] presentaron un trabajo sobre la adaptación de interfaz de usuario a diversos contextos. También presentó un nuevo framework y una herramienta para la reutilización de componentes de interfaz de usuario en entornos de desarrollo.

M. Nabuco et al. [10] también describe un enfoque dinámico de ingeniería inversa para extraer los patrones de interfaz de usuario de aplicaciones Web existentes y facilitar la construcción de modelos de prueba en el contexto de PBGT (Pattern Based GUI Testing, Patrones de pruebas basadas en GUI).

La interfaz de usuario de un teléfono inteligente es diferente al de una tableta. Por esta razón, S. Chin et al. [13] mencionaron dos consideraciones: la Personalización basada en Estados y la Personalización basada en Proyectos. Lograron personalizar la interfaz de usuario de un teléfono inteligente a una tableta utilizando las consideraciones antes mencionadas. J. Engel et al. [14] descubrieron un método basado en patrones para transformar las interfaces de usuario de los sistemas

interactivos a diversos contextos de uso. El objetivo principal del trabajo se encuentra en el modelado de las relaciones necesarias entre los diferentes patrones y la definición de un conjunto de patrones de transformación.

Tabla 1. Relación entre los elementos de patrones de diseño de interfaz de usuario. Fuente: elaboración propia.

CATEGORIA	PATRON UI	PLATAFORMA		
		WEB	MOVIL	TV
Formularios y controles	Formularios	*	*	*
	Barra de progreso	*	*	*
	Diálogos	*	*	
	Botones	*	*	*
	Divisor	*	*	*
	DashBoard	*	*	
Ayuda	Asistente	*	*	*
Herramientas	Barra de herramientas	*	*	
Navegación	Acordeón	*	*	*
	Barra de navegación	*	*	*
	Carrusel	*	*	
	Navegación por etiquetas	*	*	

CATEGORIA	PATRON UI	PLATAFORMA		
		WEB	MOVIL	TV
Búsqueda, Ordenamiento, Filtración	Paginadores	*		
	Buscador	*	*	*
Gráficos	Galería	*	*	
Señalización	Indicadores de carga	*	*	*
Lista de contenido	Listas	*	*	*
	Colecciones			*
	Tablas	*	*	*
Control de acceso	Login	*	*	*
Notificaciones	Notificación	*	*	*
Menú	Barra de etiquetas	*	*	*

Diseño visual	Menú contextual	*	*	
	Selector expandible	*		*
	Vista detalle	*		*
	Página de control	*		*
	Vista dividida	*		*

### Panorama del Diseño de Interfaces Multiplataforma en la actualidad

Diseñar interfaces de usuario multiplataforma está adquiriendo cada vez mayor importancia. Con una creciente diversidad de dispositivos, plataformas y tecnologías, los diseñadores se ven obligados muchas veces a crear versiones alternativas de las interfaces.

Actualmente, este proceso es muy costoso. Muy pocas herramientas IDE (Interface Development Environment) facilitan a la generación de interfaces multiplataforma. Una de las soluciones a este problema es mediante representaciones abstractas de las interfaces, que permiten especificar las interfaces independientemente de dispositivo, tecnología y plataforma. Además, también permiten a los diseñadores centrarse más en el modelo del diseño que en los detalles de visualización propios de cada plataforma.

Los lenguajes que se utilizan en diferentes frameworks para el desarrollo de interfaces de usuario, generalmente multiplataforma, que se describen a continuación son QT UI, Glade XML, XAML y XUL.

QT [15] permite el desarrollo de interfaces multiplataforma y tiene soporte para todos los componentes gráficos habituales, con la particularidad de que los programadores pueden extenderlos o modificarlos. Los componentes gráficos de QT emulan exactamente la apariencia de las plataformas de destino, según los estilos de cada una. QT Designer es una herramienta para el diseño de interfaces. Utiliza un documento XML para describir y almacenar la interfaz.

Glade XML Glade [16] es un generador de interfaces para GTK+ y GNOME, bajo licencia GNU GPL. Glade XML es el formato XML en el cual se almacenan las interfaces generadas por Glade. XAML (Extensible Application Markup Language) [17] es un lenguaje basado en XML para la descripción de interfaces de usuario para aplicaciones Windows. XAML consta de una serie de elementos XML para representar los principales componentes gráficos, así como el layout, paneles y manejadores de eventos.

XForms [18] es una especificación del W3C enfocada a formularios para una gran variedad de dispositivos, tanto

electrónicos como incluso en papel. Xforms propone un modelo de interfaz de usuario que sea rica en contenido, de manera que sea válida tanto para las aplicaciones comerciales como para las pequeñas aplicaciones de usuario personal e individual.

• **Herramientas para la creación de Interfaces de Usuarios.**

En la red hay multitud de herramientas para construir prototipos, desde programas profesionales hasta pequeñas aplicaciones web que nos permiten añadir enlaces a nuestros diseños para hacerlos navegables. A continuación, se muestra una selección de herramientas y recursos destacados por su utilidad, facilidad de uso y popularidad.

Tabla III. Herramientas y Winframes de creación de Interfaces de Usuario [1]

<b>Pencil Project</b>	Complemento de Firefox	SÍ	SÍ	NO	HTML, PDF, PNG, MS Word, OpenOffice.org
<b>SmartDraw</b>	Windows	SÍ	NO	SÍ	PSD, SWF, JPEG, PNG, GIF, TIFF, SVG, BMP, EMF, WMF, HTML
<b>Visio</b>	Windows	SÍ	SÍ	SÍ	SVG, EMF, GIF, JPEG, PNG, TIF, BMP, WMF, XML, PDF, XPS, HTML
<b>Wireframe Sketcher</b>	Windows / Mac / Linux	SÍ	NO	NO	PDF, PNG

Herramienta	Tipo de Software/Sistema Operativo	Biblioteca de Símbolos GUI	Wireframes interactivos	Colaboración	Formatos de exportación
<b>Axure RP</b>	Windows, Mac	SÍ	SÍ	SÍ	HTML, JPEG, PNG, BMP, GIF
<b>Balsamiq Mockups</b>	Servicio web / Windows / Mac / Linux	SÍ	SÍ	SÍ	XML, PDF, PNG
<b>Flair Builder</b>	Servicio web	SÍ	SÍ	NO	N/A
<b>Gliffy</b>	Servicio web	SÍ	SÍ	SÍ	SVG, Gliffy XML, JPG, PNG
<b>iPlotz</b>	Servicio web / Windows / Mac	SÍ	SÍ	SÍ	PDF, PNG, JPG, HTML
<b>Jumpchart</b>	Servicio web	NO	SÍ	SÍ	CSS / XHTML, WordPress
<b>Justinmind</b>	Windows, Mac	SÍ	SÍ	SÍ	HTML, PNG, JS

Herramienta	Tipo de Software/Sistema Operativo	Biblioteca de Símbolos GUI	Wireframes interactivos	Colaboración	Formatos de exportación
<b>MockFlow</b>	Servicio Web		SÍ	SÍ	HTML5, PDF, PowerPoint, PNG
<b>Mockingbird</b>	Servicio web	SÍ	SÍ	SÍ	PDF, PNG
<b>Napkin</b>	Windows / Mac / Linux	SÍ	NO	NO	N/A

Así vemos, que si bien existen varios lenguajes declarativos para la representación de interfaces de usuario, no se puede hablar de ninguna herramienta que basada en ellos dé realmente soporte para la creación de interfaces abstractas multiplataforma. Sí existen, no obstante, herramientas de desarrollo que permiten el desarrollo de aplicaciones para diferentes plataformas y en concreto de sus interfaces gráficas. NetBeans, CodeWarrior o Microsoft Visual Studio son ejemplos, que bien directamente o bien extendidos con ciertos paquetes, permiten desarrollar aplicaciones tanto para escritorio, como para web como para dispositivos móviles. No obstante, no resuelven el problema de portabilidad de la interfaz, en el sentido de que no están enfocadas a poder reutilizar la interfaz de una aplicación en una plataforma, por ejemplo de móvil en otra por ejemplo web. Evidentemente tampoco cumplen este objetivo las herramientas orientadas al desarrollo de páginas web, como Macromedia Dreamweaver o similares.

**Cualidades en el diseño de interfaces y su validación**

**Concepto de Usabilidad y sus atributos en el Diseño de Interfaces y la experiencia de usuario**

“Usabilidad” no puede explicarse simplemente como el proceso de generar estadísticas acerca del uso y la satisfacción de un producto. [19]. Además es necesario atender a los conceptos de utilidad y calidad. La utilidad definirá la medida de satisfacción de las necesidades del usuario y la calidad el grado de expectativa generada al usuario al compararla con productos similares en términos de eficiencia y rentabilidad.

Resumiendo a varios autores, se podría determinarse la usabilidad en términos de la calidad de la experiencia del usuario al interactuar con la interfaz atendiendo a todo lo que implica el concepto de calidad. Lo anterior se puede condensar en la unión de las siguientes características:

- La facilidad de aprendizaje, para que los usuarios inexpertos puedan “manejarse” rápidamente con el sistema.

- La eficiencia y escalabilidad, para que tanto los usuarios avanzados alcancen altos grados de productividad.
- La facilidad de respuesta de la interfaz al interactuar con ella.
- La intuitividad, para evitar que el usuario cometa errores y, en el caso de cometerlos, permitir al usuario dar un paso atrás y/o recuperar adecuadamente su trabajo.
- La satisfacción al usar el sistema; que impulse a los usuarios a volver.

### Enfoques para la Evaluación de una Interfaz

Evaluar una interfaz es el proceso por el que se determina el valor o la calidad de la misma en relación a unos objetivos marcados. Por lo tanto, no es sólo el medir dicha calidad, sino que requiere de cierto componente crítico [20] ya que con la evaluación se pretende alimentar el proceso de mejora continua de la interfaz de cara a conseguir que esta sea lo más “usable” posible: el objetivo básico de la evaluación es conseguir el grado máximo de usabilidad. Algunos ejemplos de metodologías para la evaluación de una interfaz se resumen en la tabla II.

Tabla II. Resumen de Metodologías para la evaluación de Interfaces.

METODOLOGÍA	CARACTERÍSTICAS
<b>Revisiones de Expertos</b>	Esta metodología consiste en la revisión del sistema por parte de un experto en usabilidad, sin que medie el usuario

<b>Evaluación Heurística</b>	Los expertos critican la interfaz confrontándola con una guía de elementos básicos de diseño elaborada previamente por otro grupo de expertos
<b>Revisión por Recomendaciones Previas</b>	Las recomendaciones son reglas e interpretaciones detalladas que deben seguirse a la hora de crear interfaces
<b>Inspección de la Consistencia</b>	Los expertos verifican periódicamente la consistencia de una familia de interfaces, en especial en lo referente a la terminología, color, composición y distribución de elementos de presentación

<b>Simulaciones de Usuario</b>	Los expertos simulan ser usuarios finales de la interfaz y recorren las funciones más usuales del sistema
<b>Juicio Formal</b>	Metodología de diagnóstico y sin objetivos establecidos basada en que los expertos se reúnen bajo la moderación de uno de ellos para discutir acerca de la interfaz tras haberla estudiado de forma individual
<b>Test de Usabilidad</b>	Este grupo de metodologías tiene el objetivo de buscar diferencias entre las diferentes interfaces comparándolas y, a la vez, buscando fallos
<b>Test de Validación</b>	Test de aceptación del producto el cual es acordado con el cliente antes de la implantación definitiva del sistema.
<b>Encuestas</b>	El objetivo de las encuestas es preguntar a los usuarios sus impresiones sobre cualquier aspecto de la interfaz, por ejemplo: presentación, disposición de iconos, menús, comandos etc.

### Recomendaciones conceptuales sobre el diseño de interfaces de usuario centradas en la experiencia del usuario

El diseño de interfaces se extiende más allá del simple acomodamiento y edición de elementos; el proceso representa la agregación de valor y significado a la plantilla que estructuramos para mostrar contenidos. Sin un buen diseño sería imposible darle un sentido a lo que mostramos en una pantalla, ya que gracias a él es posible simplificar lo complejo y dar lugar a la interacción.

Cualquier persona con acceso a un computador y a cualquier herramienta en internet, por ejemplo, puede ser capaz de diseñar interfaces, con algunos conocimientos técnicos previos y algunos principios básicos de programación puede estar lista para empezar a diseñar; sin embargo el punto clave no está precisamente en el “hacer” sino en el “saber hacerlo” ya que al aventurarse en este proceso son pocos los que respetan los principios básicos que se requieren para, además de diseñar una maqueta de interfaz, se logre el objetivo último de todo esto que es la satisfacción de las necesidades del usuario con el sistema de información a través de una interfaz “usable”.

A continuación listamos algunas de las malas prácticas más comunes en el proceso de diseño de interfaces:

- **Sobre diseñar:** Uno de los mayores deseos de un diseñador es hacer que su trabajo sea extravagante y sobresalga del resto. Esto puede llevarnos al típico error conocido como “sobre diseñar”. Hacer un diseño cargado hace que más bien el usuario sienta rechazo a usarlo, e incluso que se sienta “intimidado” y piense que es una aplicación muy complicada. Otra desventaja de hacer un diseño “demasiado único” es que ciertas convenciones gráficas se pierden, por lo que el usuario de repente se siente totalmente perdido. En la medida de lo posible se debe diseñar interfaces que posean un lenguaje gráfico familiar, no necesariamente que sean copias de otras aplicaciones, pero sí que tengan una estructura y funcionamiento similar. Menos es más y el hecho de que el usuario no tenga que aprender a usar una nueva aplicación desde cero es fundamental.

- **Sub diseñar:** Este es todo lo opuesto al punto anterior. La falta de un estilo gráfico definido y atractivo puede hacer que el usuario considere la aplicación como de “mala calidad” y que el objetivo para la cual fue creada probablemente ni siquiera se alcance. El diseño de la aplicación dictará su personalidad, y debe ayudar al usuario a utilizarla fácilmente, con indicadores gráficos atractivos, pero no “sobre diseñados”. Esto es especialmente importante en aplicaciones con funcionalidades que requieren un cierto grado de credibilidad de parte de los creadores.

**Diseñar cosas muy pequeñas:** Otro error clásico es olvidar el medio por el cual nos comunicaremos con el usuario: el tacto, y por consiguiente, el dedo índice (en el mejor de los casos). Se debe evitar realizar botones, o cualquier otro gráfico que requiera interacción, que sean demasiado pequeños, o que estén muy unidos entre sí, porque esto lo que causará es que se seleccione cosas por error, o que no se seleccione nada del todo.

**Sobrecargar el diseño:** Suele olvidarse también que el espacio que tenemos para el diseño es bastante limitado. A veces se trata de meter mucha información en un espacio muy limitado, resultando en que la interfaz se sienta muy sobrecargada, con botones muy pequeños y con poco aire alrededor. En este caso es mejor seccionar y sintetizar la información, y desplegarla de manera limpia, poco a poco. El usuario no se sentirá aturdido y tendrá el poder de seleccionar específicamente qué información quiere que se le muestre en determinado momento.

Por lo anterior y atendiendo a todo el marco conceptual y técnico en el proceso de desarrollo de interfaces adaptables y usables, se listan en esta sección algunas recomendaciones básicas que servirán de punto de partida para lograr un correcto diseño de interfaces condensando los aspectos más importantes.

**Trabajo Claro y Preciso:** es importante que el usuario final sepa reconocer los elementos que está viendo y saber para qué se usan, además debe entender cómo la interfaz le ayudará a interactuar con la aplicación. Aunque la innovación es

fundamental para darle un valor agregado a lo que diseñamos es importante tener cuidado al salirse de los estándares llevando a la confusión ya que si el usuario se enfrenta a una interfaz desordenada muy difícilmente este la utilizará en el futuro. El objetivo de una interfaz es propiciar la interacción del usuario con el sistema detrás de la pantalla y es por ello que la claridad y la precisión de una interfaz es indispensable para marcar la ruta que el usuario debe seguir en el uso del sistema.

**Usuarios bajo Control:** la interfaz debe ser un escenario seguro y confiable, alejado de situaciones imprevistas e impredecibles. Cuando se planea una interfaz es bueno agregar ciertos elementos que guíen al usuario en su interacción con el sistema y que lo alerte de los pasos siguientes o los eventos que se desencadenan al realizar una acción. El usuario se sentirá cómodo al saber que puede controlar sus acciones en el sistema y que puede seguir con seguridad hasta alcanzar su objetivo.

**Prioridad a la Interacción Directa:** al diseñar una interfaz es primordial definir cuáles son las acciones primarias que el usuario debe ejecutar para continuar la interacción con la interfaz. Una vez definidas esas acciones primarias el diseño de la interfaz debe enfocar y priorizar esas acciones evitando generar interrupciones en su flujo creando en el usuario la sensación de que es el mismo quien manipula los objetos del sistema sin intermediaciones. Ese flujo de acciones primarias debe ser secuencial y una sola acción primaria por pantalla, dejando en un segundo plano las acciones secundarias que se deriven de una acción primaria garantizando que el usuario aprenderá fácilmente el proceso y así evitar confusiones.

**Generar Familiaridad en el Usuario:** la confianza del usuario es fundamental para una adecuada experiencia con la interfaz y por ende con el sistema o aplicación, para ello es indispensable que los elementos con los que se interactúen se asemejen lo más posible a los elementos cotidianos con que la persona está relacionando y que las acciones que esos elementos generen sean compatibles con los comportamientos de las cosas en el mundo real o entorno ya que cuando algo se comporta a la forma en como esperamos genera lazos de confianza y aumenta la expectativa de experiencia y usabilidad. Al predecir los comportamientos de los elementos de la interfaz garantiza la buena utilización de la aplicación y disminuye la curva de aprendizaje que el usuario debe alcanzar para la correcta utilización de la interfaz.

**Consistencia en los Elementos de la Interfaz:** el diseño de los elementos de la interfaz debe realizarse de manera consistente, atendiendo a la coherencia entre su diseño y utilidad. Es necesario cuidar que cada sección de la interfaz tenga relación con las otras y evitar así tener diferentes diseños que harán que el usuario se confunda y no identifique un estilo único en la interfaz que le genere fácil recordación; esto también debe aplicar para las funcionalidades y comportamientos de cada elemento, ellos deben ser homogéneos, con el objetivo de que los usuarios nuevos identifiquen y aprendan fácilmente el uso de la interfaz. La consistencia garantiza la uniformidad de la interfaz lo que

incrementa la productividad y la calidad del trabajo del usuario.

**Jerarquía y Orden Visual:** el orden y la jerarquía visual en una interfaz hace que se fortalezca su diseño, ya que esa jerarquía y niveles de importancia que se le da a cada elemento en la pantalla hace que la interfaz sea intuitiva y predictiva y le da un orden natural que permite que el usuario aprenda rápidamente a navegar en ella y logre identificar los puntos importantes y las relaciones entre contenido y diseño. Esta organización „Inteligente“ hace interfaces mucho más simples y fáciles de usar y ayuda a evitar la sobrecarga en la pantalla haciendo que con menos hagamos mucho más [21-24].

## Conclusiones

Dado al avance tecnológico y la existencia de nuevos dispositivos, el diseñar una interfaz gráfica de usuario se ha vuelto una tarea más complicada, debido al gran número de dispositivos disponibles, esto genera un debate entre sus capacidades y sus limitaciones. Además cada vez aparecen usuarios „expertos“ que no solo se limitan a esperar retroalimentación de una máquina sino que siempre buscan anticiparse y exigen más para obtener una adecuada experiencia. Se desarrolló una guía conceptual que permite la unión de los diferentes contextos de uso de una aplicación y así evaluar los diferentes patrones y técnicas de diseño de interfaces con el objetivo de que éstas se puedan adaptar a cada vez más dispositivos sin perder su enfoque de usabilidad.

## Referencias

- [1] Ferre, X.; Juristo, N.; Moreno, A.M., “Improving Software Engineering Practice with HCI Aspects”, Computer Science, Vol.3026, pp. 349-363, 2004.
- [2] Acosta A.E., Zambrano N. Importancia, problemas y soluciones en el diseño de la interfaz de Usuario. Centro de Ingeniería de Software y Sistemas (ISYS) Escuela de Computación, Facultad de Ciencias de la Computación. 2014.
- [3] Klemisch, K., Weber, I., Benatallah, B. Context-Aware UI Component Reuse. Advanced Information Systems Engineering, Spain, Springer Berlin Heidelberg, pp. 68–83. 2013.
- [4] Hassan Y, Francisco J, Fernández M Iazza G. Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información. 2014 [En Línea] Disponible en: <http://www.hipertext.net>.
- [5] Neil, T. Mobile Design Pattern Gallery. Segunda ed., O'Reilly Media, pp. 284. 2012.
- [6] Tidwell, J. Designing Interfaces. Segunda ed., O'Reilly Media, pp. 578. 2011.
- [7] Sheibley, M. Mobile Patterns. Disponible: <http://www.mobile-patterns.com>. 2013.
- [8] Toxboe. UI Patterns User Interface Design pattern Library. Disponible: <http://ui-patterns.com>. 2016.
- [9] UNITiD. Android Patterns. Disponible: <http://unitid.nl/androidpatterns>. 2016.
- [10] Nabuco, M., Paiva, A. C. R., Pascoal, J. Inferring User Interface Patterns from Execution Traces of Web Applications. In: Computational Science and Its Applications – ICCSA 2014, Portugal, Springer International Publishing, pp. 311- 326. 2014.
- [11] Kim, S. Pattern and Event Based Logical UI Modeling for Multi-Device Embedded Applications. Convergence and Hybrid Information Technology, Springer Berlin Heidelberg, pp. 560-567.
- [12] Klemisch, K., Weber, I., Benatallah, B. Context-Aware UI Component Reuse. Advanced Information Systems Engineering, Spain, Springer Berlin Heidelberg, pp. 68-83. 2013.
- [13] Chin, S., Iverson, D., Campesato, O., Trani, P. Beyond Mobile: Tablets and TV. Pro Android Flash, Apress, pp. 399-426. 2011.
- [14] Engel, J., Märtin, C., Forbrig, P. HCI Patterns as a Means to Transform Interactive User Interfaces to Diverse Contexts of Use. Human-Computer Interaction, Design and Development Approaches, USA, Springer Berlin Heidelberg, pp. 204-213. 2011.
- [15] <http://www.trolltech.com/products/qt/>
- [16] <http://glade.gnome.org>
- [17] C. Petzold. Create Real Apps Using New Code and Markup Model. Disponible En <http://msdn.microsoft.com/msdnmag/issues/04/01/Avalon/default.aspx>
- [18] <http://www.w3.org/MarkUp/Forms/>
- [19] GOULD, J. y Lewis C.; “Designing for usability: key principles and what designers think”. Communications of the ACM. 2015.
- [20] Nielsen, Jakob; “Ten Usability Heuristics”. [En Línea]. Disponible En: [http://www.useit.com/papers/heuristic/heuristic\\_list.html](http://www.useit.com/papers/heuristic/heuristic_list.html).
- [21] E. Martelo, M. Manotas and B. Vallejo, "Prototipo De Una Aplicación Móvil Con Realidad Aumentada Para Mostrar Puntos De Información De Ubicación De La Universidad Simón Bolívar En Barranquilla Colombia Mediante El Uso Del Navegador Móvil Junaio", Investigación e Innovación en Ingenierías, vol. 2, no. 2, 2014. DOI: <https://doi.org/10.17081/invinno.2.2.2048>

[22] A. Labrador, P. Wightman, A. Santander, D. Jabba and M. Jimeno, "Tis-Bad: A Time Series-Based Deobfuscation Algorithm", *Investigación e Innovación en Ingenierías*, vol. 3, no. 1, 2015. DOI: <https://doi.org/10.17081/invinno.3.1.2035>

[23] J. Carvajal Martínez, & A. Guzman, *Economía de mercado y democracia: elementos para una crítica al discurso del desarrollo promovido por las Instituciones Financieras Internacionales*. *Justicia*, vol. 22, n°. 31, 2017. <https://doi.org/10.17081/just.22.31.2602>

[24] F.A. Villa, J.D. Velasquez, y P. Sanchez, "Control del sobreajuste en redes neuronales tipo cascada correlación aplicado a la predicción de precios de contratos de electricidad", *Revista Ingenierías Universidad de Medellín*, 14

(26), 2015. <http://www.scielo.org.co/pdf/riium/v14n26/v14n26a11.pdf>