

Integración de las buenas prácticas de experiencia de usuario y arquitectura de información en el desarrollo de software

Isaac David Alfaro Vargas
Escuela de Ingeniería
Ulacit
San José, Costa Rica
ialfarov068@ulacit.ed.cr

Julio Córdoba Retana
Escuela de Ingeniería
Ulacit
San José, Costa Rica
jcordobar022@ulacit.ed.cr

Resumen – *El presente artículo trata sobre las buenas prácticas en el desarrollo de software, específicamente en los ámbitos de experiencia de usuario, arquitectura de información y observaciones técnicas en el desarrollo front-end, como la creación de un sistema de diseño. Lo anterior al tomar como idea central su integración para llegar a aplicarlas y tener como resultado una buena usabilidad en el producto en desarrollo. También se repasan ciertos estándares que existen en la actualidad para el desarrollo de software, con énfasis en la usabilidad. Con el fin de integrar las buenas prácticas, se presenta el prototipo de un producto de software, el cual brinda un contexto financiero, con el objetivo de aplicar las prácticas y principios mencionados durante la investigación explicando el porqué de su uso y cuáles de los principios se cumplen. De las prácticas mencionadas, algunas pueden ser opcionales, según contexto del producto en desarrollo y otras son de importancia. El utilizar estas prácticas puede resultar en múltiples beneficios, uno de ellos es la mejor aceptación del usuario hacia el producto u organización. Por último, se debe destacar que en la implementación se debe buscar cómo mejorar los procesos de los usuarios y, por ende, sus actividades diarias.*

Palabras clave: *Experiencia de usuario, usabilidad, desarrollo de software, buenas prácticas, arquitectura de información.*

Abstract – *This article is about the best practices in software development, particularly in the field of user experience, information architecture, and technical observations within the front-end development, such as the creation of a design system. The main idea is the integration of them, to get to be applied and have as a result of good usability in the product under development. Also, certain standards that exist today for software development focusing on usability were reviewed. To integrate good practices, a prototype of a software product is shown, which presents a financial context. Created to apply the practices and principles mentioned during the research and explain why they are being used and which of the principles are being fulfilled. The practices mentioned may be optional depending on the context of the product under development and others are important to consider. Using these practices can result in multiple benefits, one of which is better user acceptance of the product or organization. It should be emphasized, that the*

implementation should seek to improve the users' processes and therefore their daily activities.

Keywords–*user experience, usability, software development, best practices, information architecture.*

I. INTRODUCCIÓN

En la actualidad, es común utilizar plataformas para llevar a cabo tareas cotidianas o simplemente para entretenimiento, tanto en formato *web* como aplicación móvil y es normal que los usuarios prefieran una que destaca de la competencia. Sin embargo, una empresa o un equipo de desarrollo debe asegurarse de que su producto sea ese que destaque, no solo en lo visual, lo cual tiene importancia, sino también en la usabilidad y en la experiencia que tiene el usuario.

Un estudio relacionado con el rendimiento de los sitios [1], reveló que el 88 % de los consumidores no están dispuestos a volver a un sitio *web* en el que tuvieron una mala experiencia. El mismo estudio reveló que al menos la mitad demostró tener una mala percepción de la empresa en general, después de esta mala experiencia. Estos datos pueden alertar a cualquier compañía o dueño de producto y llevar a que se haga una revisión de la experiencia que tienen los usuarios.

Para el desarrollo de un producto o servicio de *software* existen buenas prácticas, ya sea desde la estructura de programación, como los distintos *frameworks* y librerías actuales, hasta cómo debe ir la información para una mejor comprensión (arquitectura de información). Con el fin de desarrollar un producto que tenga una buena experiencia, se pueden seguir buenas prácticas de experiencia de usuario (del inglés UX por *user experience*) y de arquitectura de información.

La importancia de lo visual en un producto de *software* no debe menospreciarse, así lo revela un estudio [2], el cual determina que los usuarios juzgan la credibilidad de un sitio *web* en un 75 %, con base en la estética de este. Aunque desde el punto de vista técnico esto puede carecer de sentido, según este estudio, la confiabilidad de un sitio se ve afectada por su aspecto.

Además, las buenas prácticas del desarrollo, en este caso en la capa visual, lo que se conoce como *front-end*, influyen en la calidad de su experiencia. Lo anterior ya que al escribir mal una aplicación o sitio *web*, no tener orden u omitir buenas prácticas, se puede ver afectado desde el rendimiento del producto hasta la calidad visual y, por ende, la experiencia de los usuarios.

Al unir esas buenas prácticas de desarrollo, experiencia de usuario y arquitectura de información, se pueden observar beneficios. Entre estos beneficios se encuentran, a saber, mejor posicionamiento en el mercado de un producto o empresa, incluso apego o lealtad de los usuarios ante el producto, esto porque la satisfacción se incrementa, es un producto más fácil de utilizar y el cual sirve para lo que el usuario espera. Asimismo, ayuda a mejorar la eficiencia en algunas tareas y hay un aumento de la confianza por parte de los usuarios.

El presente artículo aborda estos conceptos, con el fin de determinar la viabilidad de unir sus mejores prácticas, además, evaluar si al hacerlo se obtienen mejores resultados. Esto desde la experiencia de usuario, en la que se mantenga la arquitectura de información para que esta sea fácil de encontrar y con el desarrollo *front-end* enfocado en mantener un desarrollo en concordancia con los otros dos conceptos, sin perder la calidad técnica.

El objetivo principal de unificar conceptos es mejorar la usabilidad de un producto o servicio de *software*. Sin embargo, para determinar el nivel de usabilidad de este, se debe partir de una base, la cual pueden ser estándares de usabilidad. En el presente artículo se revisan estándares de usabilidad y la posible relación al aplicar la integración de las prácticas de los tres conceptos mencionados.

El objetivo del artículo es contestar la hipótesis, a saber, al tener un método/modelo en el cual se recopilen buenas prácticas de UX y AI, aplicándolas a un producto o servicio de *software*, ya sea aplicación móvil o página *web*, se mejorará su usabilidad.

II. METODOLOGÍA

La metodología que se utiliza para el artículo es cualitativa, esto porque según Hernández Sampieri, esta metodología resulta conveniente para entender y buscar patrones, experiencias y resultados. Además, los resultados no es necesario que se traduzcan a números ni que se analicen estadísticamente, tampoco generalizar de una manera probabilística [3].

Asimismo, la base de recolección de datos para el artículo inicia con la investigación acerca de la teoría y se crea el prototipo de una aplicación móvil y su contexto se explica más adelante. Esto con el objetivo de dar resultados según la hipótesis propuesta.

En el enfoque cualitativo el análisis de los datos se puede dar al mismo tiempo que la recolección [3], siempre según la investigación. Además, los datos por recolectar no tienen una estructura por la naturaleza del estudio.

III. MARCO TEÓRICO

A. Usabilidad

Este concepto ha sido de gran importancia durante los últimos años, debido a la constante evolución del *software*, sin embargo, se pueden encontrar diferentes definiciones. El término usabilidad llegó para sustituir la frase *amigable con el usuario* (*user friendly*), la cual da origen al concepto de usabilidad y que es común que se utilice en la actualidad, no obstante, su definición es subjetiva [4].

Por otro lado, el concepto propio de usabilidad se refiere a que un producto o servicio al utilizarse se disfrute, que sus funciones sean útiles y que mejore la vida de sus usuarios. Para eso se contemplan los sentimientos, pensamientos o sensaciones que pueda generar en los usuarios [5].

Además, se describe la usabilidad como la facilidad para un usuario de utilizar y navegar entre las partes de un sitio *web*. Hassan también lo relaciona con un término llamado *findability*, el cual describe como la posibilidad y facilidad de que el usuario encuentre la información que busca o necesita, ya sea con un buscador local (en el mismo sitio *web*) o con una buena implementación de arquitectura de información [6].

Es importante destacar que, entre las definiciones existentes, sin importar el concepto de usabilidad que se encuentre, en su mayoría se le da importancia principalmente al usuario. Lo anterior ya que es quien utiliza los sitios *web* o aplicaciones móviles y es quien decide si el producto que tiene a su disposición le sirve, tal y como esperaba para llevar a cabo las tareas deseadas.

B. Experiencia de usuario (UX)

Cualquier producto o servicio que se consume, se tiende a consumir con regularidad si es de agrado para el usuario y esto se presenta en muchos ámbitos, sobre todo en el del *software*, en el que se debe prestar más atención a lo que piensan los usuarios sobre los productos que utilizan con frecuencia. A partir de la suposición de que un usuario tiene una mala experiencia con una aplicación móvil, ya que es lenta y no logra encontrar lo que necesita, es decir, la aplicación no cumple con la mínima expectativa para la cual se instaló, tiene como primera consecuencia que el usuario no utilice más el aplicativo y lo desinstale. La experiencia en ese caso fue mala, además, el usuario pudo tener alguna sensación, desde enojo hasta frustración por no lograr lo que deseaba.

A esto se debe la importancia de generar una buena experiencia para el usuario, pero primero debe tenerse claro qué es la experiencia de usuario. Esta se define como la interacción del ser humano con un producto con una conexión emocional y toma en cuenta experiencias positivas y necesidades psicológicas. Además, al tocar las emociones del usuario se genera una lealtad, tanto al producto como a la marca relacionada [7].

Se puede encontrar un enorme beneficio si en el producto se implementa de manera satisfactoria, desde una empresa emergente hasta una gran empresa, ya que les permite colocar el producto de *software* sobre la competencia. Asimismo, pueden generar lealtad de los usuarios, tanto con nuevos como

con recurrentes y no solamente en un producto, sino con cualquier otro que desarrollen.

La empatía es un concepto importante en el diseño, al que se le debe prestar atención, esto ya que se basa en comprender emocionalmente las necesidades de los usuarios, con el fin de mejorar su trabajo [8]. Este beneficio se nota al instante, pues gracias a la empatía se logra conocer con mejor precisión la necesidad real del usuario.

C. Principios de UX

A continuación, se abordan algunos principios para tomar en cuenta en el momento de estudiar la experiencia de usuario [7]. Uno de ellos es sobre la primera impresión del usuario con un producto y la importancia de que esta sea la mejor. Además, en el ámbito del usuario, sobresale que se debe evitar una sensación de incapacidad, es decir, que en el momento de utilizarlo se sienta capaz de hacer las tareas y que esto sea de manera natural, que no sobresalga de lo cotidiano. Asimismo, una regla que destaca es ofrecer una buena atención al cliente.

En los principios que se enfocan en el producto destaca que el avance de la tecnología permite que en la actualidad se tengan muchos dispositivos. Por esto, permitir la compatibilidad del producto con muchos dispositivos es importante para brindar una buena experiencia. Además, aspectos como permitir que el usuario personalice ciertas características y que el diseño sea el adecuado, según el contexto en que se utilice el producto (entretenimiento, financiero, comunicación, entre otros) también son de importancia [7].

Entre otros principios por tomar en cuenta está que la información suministrada sea suficiente, sin excederse en la cantidad del contenido. Por lo tanto, es importante considerar que el producto cumpla con los objetivos para los cuales se obtuvo y ofrecer funcionalidades necesarias, ya que se relacionan y al aplicarlas se brinda una buena experiencia [7].

D. Arquitectura de información

La arquitectura de información no se concentra en una sola definición, sino que tiene varias. Por ejemplo, es una disciplina que ayuda a que un producto, sitio *web* o aplicación móvil, tenga la información fácil de encontrar y, sobre todo, que sea entendible. La arquitectura de información no solo se define de esa manera, sino también como 1) el diseño de la estructura de ambientes de información compartida, 2) el diseño de elementos de una organización y toma en cuenta etiquetas, búsquedas y navegación, 3) presentar la información de manera que se mantenga la usabilidad, la comprensión y la capacidad de encontrar lo que se necesita [9].

Entre los principales puntos por considerar al implementar la arquitectura de información en el desarrollo de *software* se encuentra que el encargado de AI, quien debe tener la capacidad de diseñar la información, cree y etiquete el contenido que llegue al usuario final. Además, que contemple que su trabajo sea entendible para cualquier persona involucrada en el proyecto [10].

Entre las metodologías que más se utilizan en la actualidad se encuentran las historias de usuario. Estas historias son los requerimientos de un producto o también una expresión de lo que espera un usuario de este producto [11]. Por lo tanto,

primero se debe llevar a cabo el análisis sobre cómo se implementará la arquitectura de información y después se debe incluir dentro de estas historias de usuario. De esta manera, se tiene integrado en el desarrollo de *software*.

E. Principios de arquitectura de información

El entender la manera en que se puede mejorar un producto tecnológico con arquitectura de información puede ser un tanto complicado. Por este motivo, existen algunos principios que son de gran ayuda en el momento de comprender cómo y por qué aplicar AI al producto.

1) *Principio de objetos*: El primer principio se relaciona con los objetos, ya que intenta desmoronar cada componente de la vida real en partes, las cuales tienen sus atributos, propiedades y comportamiento [12]. En AI este principio trata sobre identificar y tratar cada elemento visual del *software* como un objeto, con el fin de que estos componentes estén dentro de una clasificación o categorización adecuada.

2) *Principio de elección*: El producto debe ofrecer múltiples opciones a los usuarios, que sean útiles y contemple un balance, es decir, que tenga las suficientes opciones para que se cumpla la necesidad para la cual se creó [13]. En este principio es importante rescatar que un usuario al no encontrar lo que busca pierde el interés, por lo que no solo es importante mantener las opciones principales, sino que las encuentre de manera fácil.

3) *Principio de aviso*: El ser humano puede procesar una cantidad limitada de información a la vez, por lo que mostrarle una gran cantidad puede ser abrumador. De esta manera, el principio afirma que en un inicio se debe mostrar la información necesaria para que el usuario entienda la que se encontrará más adelante [12].

4) *Principio de ejemplos*: El ser humano tiende a categorizar su entorno, aún más si encuentra semejanzas con ejemplos, por lo que la mejor manera de explicar categorías es mediante ejemplos de estas [14]. Lo anterior ayuda a que un usuario conozca en qué categoría se encuentra y lo que busca, para darle una idea de dónde encontrarlo.

5) *Principio de la clasificación múltiple*: La información debe estar separada según los elementos más relevantes, además, influencia en la eficacia de la navegación dentro de un producto [15]. Asimismo, es importante considerar esto porque es una necesidad de los usuarios que haya separación y se debe tomar en cuenta que los usuarios pueden interpretar estas clasificaciones de manera distinta.

6) *Principio de la navegación*: La navegación ayuda a que los usuarios se desplacen dentro de un producto, esto se suele cumplir con enlaces en la parte superior de una aplicación, filtros o navegación local (en la misma página). Se puede definir al crear jerarquías para decidir qué mostrar primero y qué tiene en usuario a su disposición [16]. Además, se debe tomar en cuenta mantener una estructura lógica, es decir, dividir los componentes de la navegación y no mezclar temas que no se relacionan.

7) *Principio de evolución*: El hacer un producto expandible al contemplar que crecerá, es decir, es escalable, es importante

[15], ya que ahorra tiempo y costos en el futuro. Por esto, este principio propone que al inicio es importante diseñar contemplando ese crecimiento.

F. Buenas prácticas de desarrollo de software (front end)

El desarrollo FE se refiere al campo del desarrollo de *software* en el que se programa lo relacionado con la interfaz de un producto, todo con lo que el usuario pueda interactuar y observar. Cada componente que se ve en un sitio o aplicación, desde botones, textos, animaciones, entre otros, es responsabilidad de la capa FE [17]. Además de la responsabilidad de los aspectos visuales, también se enfoca en la parte lógica del producto, por ejemplo, lo relacionado con la navegación dentro del producto, asuntos de seguridad, envío y recepción de datos y conexión con los servidores.

Las prácticas que se pueden seguir en el desarrollo de *software* FE son amplias, tanto desde el ámbito visual hasta el ámbito meramente técnico. Ambas son de gran importancia y tienen un impacto significativo dentro del rendimiento y la experiencia de usuario. En cuanto al ámbito técnico, existe una gran cantidad de lenguajes, marcos de desarrollo (*frameworks*) y librerías, que ayudan a que los desarrolladores se enfoquen en lo importante, en donde los elementos principales ya se encuentran creados.

Lo anterior es una gran ayuda al desarrollar cualquier producto de *software*, sin embargo, el hecho de que existan múltiples tecnologías hace que cada una tenga sus mejores prácticas. De igual manera, también existen buenas prácticas un poco más generales, las cuales pueden aplicarse sin importar la tecnología en la cual se desarrolle.

Entre las buenas prácticas a nivel de desarrollo se puede comenzar con el análisis de cómo mejorar la carga de un sitio *web*. Esto es importante, ya que ayuda a mejorar el tiempo en que el usuario observa una página, de manera que cuanto más rápido puede tener una mejor experiencia. Lo recomendado es que la página se cargue y dibuje cuando se hace la solicitud del servidor, lo que agiliza el tiempo en que se muestra la página al usuario [18].

Otra buena práctica en el desarrollo es tener los archivos con el menor peso posible, ya sean estas imágenes, vídeos o *scripts*, cuidando siempre su calidad. Las hojas de estilo son archivos en los que se definen los aspectos visuales de una página *web*, por lo que también deben implementarse de una manera ligera. Primero, deben eliminarse las cosas innecesarias, segundo se debe buscar un nivel de compresión del mismo archivo, también se puede minimizar el tamaño al reutilizar componentes de estos archivos. Con respecto a los archivos multimedia, estos no tienen que cargarse en el momento que se carga el sitio *web*, pueden descargarse cuando se requieran o de manera no síncrona [18].

Por el modelo de desarrollo que existe en la actualidad, se suelen hacer peticiones al servidor más seguido para solicitar datos que son necesarios. Sin embargo, se recomienda que esto se haga poco, ya que cada petición puede tardar y dar la sensación de que el sitio es lento [18]. Eso se puede mejorar al hacer un primer llamado al servidor con los datos que son fundamentales para el uso de la página y los datos no tan

esenciales pueden solicitarse hasta en el momento en que el usuario lo requiera.

Otra buena práctica que existe es evitar las redirecciones dentro del mismo sitio o aplicación, ya que, en ocasiones, se suele cargar de nuevo el sitio desde cero [19]. Esto se puede solucionar en la actualidad, mediante SPA (en inglés sus siglas significan Single Page Application). Lo anterior ya que estas encargan precisamente de no requerir recargas del sitio y el uso de este es más fluido y con menos tiempo de espera [20].

G. Sistema de diseño (DS)

Un sistema de diseño es un conjunto de herramientas y guías que ayudan a tener un diseño consistente para una aplicación o sitio *web* [21]. Prácticamente todo lo que sea visual en una aplicación puede estar dentro de un DS. Esto beneficia en el tiempo de desarrollo, ya que no hay que volver a hacer elementos que ya están creados y si se quiere cambiar solo se cambia el componente.

Un sistema de diseño puede implementarse de dos maneras. La primera es que un equipo de desarrollo lo crea desde cero para después implementarlo en la aplicación. La segunda es que existen librerías creadas por grandes empresas que las ponen a disposición, por lo que el equipo de desarrollo se encarga de implementarlas. Esto se puede observar cuando existen aplicaciones o páginas *web* que no están relacionadas y coinciden los diseños de sus botones, textos, menús y otros componentes.

El desarrollo de un sistema de diseño no necesariamente lo lleva a cabo el equipo encargado de desarrollar el producto, sino que ellos pueden seguir los lineamientos propuestos por un equipo de diseñadores. Para crear un sistema de diseño se pueden tomar en cuenta los siguientes consejos.

Lo primero es realizar una evaluación, en caso de que se trate de un producto existente, estos son, textos, etiquetas, botones, entre otros. Posteriormente, se definen los colores, esto puede depender de una marca, tanto de empresa como del producto. La definición de colores incluye fondos, botones, títulos y cualquier componente que pueda existir en el producto [22]. En el esquema de colores es importante considerar el modo oscuro, ya que es una tendencia.

Otro elemento que forma parte de un DS es la tipografía, esto incluye definir una guía sobre cómo utilizarla, es decir, definir el tipo de fuente del producto y sus variantes según títulos, subtítulos, alertas o cualquier información que se presente. Los patrones son un punto fundamental, ya que ayudan a documentar cada componente del producto. Estos patrones contienen las propiedades que tienen elementos como botones, entradas de texto, barras de navegación, selectores, entre otros [22]. Su importancia radica en el ahorro de tiempo que supondrán en el desarrollo, ya que una vez creados, pueden reutilizarse a lo largo del desarrollo del producto.

Lo último es documentar lo que se creó, esto es lo que hace que sea un DS, lo cual incluye los componentes por utilizar y, sobre todo, cómo y cuándo utilizarlos. Esto es importante, ya que esta documentación está disponible para quien la necesite en un equipo de desarrollo [23].

H. Estándares de usabilidad

Entre los estándares existentes de ISO se encuentra asegurar calidad, rendimiento y disponibilidad de un producto de *software*, los cuales se pueden aplicar y relacionar en distintos puntos con la usabilidad. Sin embargo, no son exactos con respecto a la usabilidad, de manera que ISO creó el estándar 9241-210, que traducido su título original del inglés es Ergonomía de la interacción humano-sistema: Diseño centrado en el humano para sistemas interactivos [24].

Este estándar presenta una guía para lograr la calidad en el uso de un producto, al incorporar actividades iterativas, es decir, es un proceso que se revisa constantemente. Este estándar menciona factores humanos, conocimientos y técnicas de ergonomía, con el fin de lograr eficiencia y eficacia al utilizar un producto de *software* [25].

Los principios que menciona son a alto nivel, es decir, sin mucho detalle. En primer lugar, está la garantía de que el diseño de un producto se base en el entendimiento de los usuarios, ambiente y el contexto. El segundo indica que se debe involucrar a los usuarios en todo el proceso de desarrollo, no solo al final, con el objetivo de observar si se cumple con las expectativas. El tercero busca que se mejore el producto mediante pruebas a prototipos y diseños preliminares y evalúa qué sirve y qué no [26].

El cuarto principio busca que el desarrollo se aplique en procesos iterativos, con el fin de presentar avances con frecuencia y afinar el producto. El quinto busca crear una buena experiencia mediante el estudio de percepciones y emociones. El sexto plantea que el desarrollo de la experiencia no se concentre en una sola disciplina, sino que incluya también otras para contar con puntos de vista variados. Entre las disciplinas pueden estar desarrolladores, mercadeo, diseñadores, entre otros [26].

Existen otros estándares ISO enfocados en usabilidad, como ISO 9241-151, el cual brinda una guía para crear interfaces centradas en el usuario en un entorno *web* [27]. Además, este estándar se relaciona con ISO 9241-210 para seguir los criterios de evaluación [25]. ISO 9241-171 también hace referencia al primer estándar mencionado y este se enfoca en guías orientadas a la accesibilidad en el *software*. Por último, el ISO/IEC 25062 brinda los métodos para crear reportes referentes a usabilidad [28]. Asimismo, está ISO/IEC 9126-4 que menciona métricas para la usabilidad [29], el cual se aborda en el siguiente segmento.

I. Métricas

Las métricas en la usabilidad ayudan a determinar si el producto de *software* cumple con el objetivo para el cual se desarrolló. Es importante considerar el contexto de este producto, ya que, según este, se definen sus métricas, por lo que no es una buena práctica generalizarlas ni medir todos los productos contemplando los mismos parámetros.

El medir el producto de *software* es importante porque ayuda a encontrar problemas que resultan difíciles de detectar. También ayuda a determinar objetivamente si el diseño cumple según lo que los usuarios esperan y a tomar decisiones de diseño de manera objetiva. Además, mediante la iteración, al

llevar a cabo cambios, se puede volver a medir para determinar si estos fueron de utilidad [30].

El estándar ISO/IEC 9126-4 recomienda que las métricas incluyan tres categorías. La primera es la eficacia, la cual busca que se completen las metas del usuario de manera exitosa. La segunda es la eficiencia, esta afirma que se deben utilizar los recursos necesarios para completar los objetivos del usuario con el producto. La tercera es la satisfacción, que busca el *comfort* y la aceptación del uso del producto [29].

Por otra parte, hay métricas específicas que pueden ayudar a medir un producto. Por ejemplo, medir el tiempo que tarda un usuario en llevar a cabo una tarea esencial, esto puede variar según el tipo de tarea, sin embargo, se debe mantener en un nivel aceptable para el usuario. Relacionado con las tareas, es importante la tasa de finalización, ya que esta métrica determina si los usuarios logran realizar las tareas para las cuales adquirieron el producto o incluso encontrar posibles puntos de atascos [31].

El capturar los errores para determinar cómo suceden y en qué momento ayuda a conocer posibles puntos de dolor de un producto, con el fin de mejorarlos. De igual manera, medir problemas de usabilidad ayuda a saber cuándo los usuarios ingresan a secciones incorrectas o no entienden el contenido. Con esos datos se puede mejorar la experiencia y eliminar la posible frustración de los usuarios. Además, como lo menciona el estándar ISO/IEC 9126-4, la medición de satisfacción es importante, ya que toma en cuenta la opinión del usuario, a quien se busca persuadir mediante el producto [31].

IV. ANÁLISIS DE RESULTADOS

A continuación, se revisa el prototipo de un producto de *software*. Este producto se trata de una aplicación móvil, mediante la cual se pretende brindar al usuario una herramienta financiera. El objetivo de la aplicación es que los usuarios puedan llevar el control de sus gastos e ingresos. El prototipo por revisar se creó para esta investigación, por lo que se intentó seguir las buenas prácticas y principios mencionados, con el fin de analizar los puntos más importantes y responder el porqué de las decisiones.

El prototipo se creó mediante una herramienta en línea llamada proto.io. Esta herramienta es una plataforma de prototipado [32], la cual sirve para crear interfaces de usuario, estudiar posibles interacciones de usuario y simular todo el posible flujo que hará el usuario al utilizar el producto.

Lo primero por destacar es el uso de colores de acuerdo con el contexto. A pesar de que la aplicación es para uso personal, al ser financiera, lo idóneo es utilizar colores poco extravagantes, que denoten un tono serio y que genere confianza, ya que se trata de las finanzas personales.

En la Figura 1 se muestra la paleta de colores escogida para la aplicación y se debe destacar que definir una paleta cumple con un requisito en el momento de crear un DS. El color azul es el color principal de la aplicación y tanto el blanco como el negro son variantes de estos y se utilizan para componentes. Con respecto a las variantes de colores verde y rojo, se usan para reflejar ingresos y gastos, respectivamente.



Figura. 1. Paleta de colores. Fuente: Elaboración propia, 2021.

La Figura 2 presenta dos de los componentes que más destacan en la aplicación. Estos son los botones de acción y un cuadro con borde circular que se utiliza para presentar información. Además, en la aplicación se utilizó la fuente San Francisco Pro.

El definir los componentes por separado contribuye en dos principios mencionados. El primero es el de definir y observar cada detalle de un producto como un objeto, en este caso, los botones, los textos y el cuadro de información son objetos de la aplicación. El segundo se cumple con definir cada componente, al crear un DS, por lo que cada uno es totalmente reutilizable en la aplicación. Al definir los componentes con estas características también se cumplen con tener patrones visuales en un DS.



Figura. 2. Componentes. Fuente: Elaboración propia, 2021.

La pantalla de bienvenida, mostrada en la Figura 3, al ser la primera experiencia del usuario con la aplicación debe proporcionar una buena impresión. Esta primera pantalla explica en pocas palabras para qué es la aplicación y tiene la opción para registrarse o iniciar sesión. Esta pantalla se presenta una única vez, después de que el usuario haya instalado la aplicación.

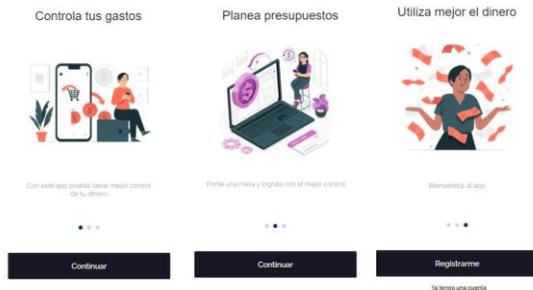


Figura. 3. Pantallas de bienvenida. Fuente: Elaboración propia, 2021.

La bienvenida es importante para garantizar el cumplimiento acerca de evitar repeticiones de procesos, ya que carece de sentido brindar la bienvenida al producto cada vez que se ingrese. Además, con la breve introducción se cumple con el principio de aviso en AI, ya que se le presenta al usuario para qué le sirve la aplicación. Adicionalmente, se puede observar en la Figura 3 que las pantallas de bienvenida presentan una cantidad de texto limitado, esto es para cumplir el principio de UX, el cual indica que la información suministrada debe ser suficiente para ser entendida.

En la Figura 4 se muestra el inicio de sesión, el cual sirve para que el usuario ingrese a la aplicación con su cuenta, en caso de no tener una, entonces que haga el registro. Este es con usuario y contraseña para garantizar el uso de elementos comunes para el usuario. Además, si el usuario lo desea puede iniciar la sesión mediante verificación biométrica, si el teléfono así lo permite. También es entendible lo molesto que es iniciar la sesión cada vez que se abre una aplicación, por lo que se tiene la opción para que la sesión se mantenga y evitar repetir este proceso.

Otra pantalla gemela a la de inicio de sesión es la de registro, en la Figura 4 esta pantalla tiene algunos campos más, que son necesarios para crear la cuenta, sin alargar el proceso. Esto con el fin de que, al ser la primera vez, pueda crear la cuenta lo más rápido posible para utilizar el producto. Es importante destacar que el registro y los datos que se soliciten dependen totalmente del contexto del producto.

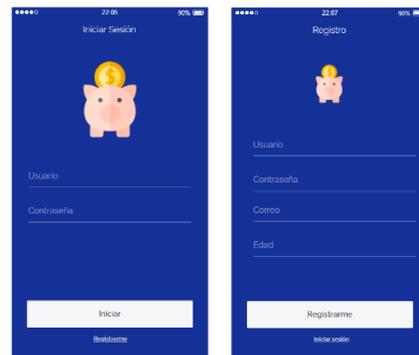


Figura. 4. Inicio de sesión y registro. Fuente: Elaboración propia, 2021.

Se solicita una cuenta de usuario, con el objetivo de cumplir el principio de compatibilidad para que el usuario pueda descargar la aplicación en cualquier otro dispositivo, lo que incluye una computadora, lo cual ayuda a eliminar la limitante entre plataformas. La pantalla de registro cuenta con pocos campos para que el usuario ingrese, esto al seguir el principio de UX que menciona que se debe cumplir con la función para la cual se creó, sin tener otros elementos o funcionalidades.

Al entrar a la aplicación se le muestra al usuario la pantalla principal. Como se puede observar en la Figura 5, esta cuenta con el balance del mes y un gráfico que refleja los movimientos. Seguido muestra dos cuadros de información, los cuales reflejan ingresos y gastos. En la parte inferior se presenta la navegación principal, con las opciones para ver las categorías, agregar un movimiento y ver reportes.

Esta pantalla, Figura 5, cumple con el principio de AI acerca de la navegación, ya que presenta tres opciones en la sección inferior, que al seleccionar el usuario visualice la información o haga la tarea que busca. Las cajas de información de ingresos y gastos pretenden ser una antesala a los movimientos, los cuales tienen la misma estructura visual, por lo que se cumple con el principio de AI sobre ejemplos, el cual pretende darle a entender al usuario lo que se le presentará. Además, las categorías ayudan a satisfacer dos principios, el primero es de clasificación en AI, ya que se pueden categorizar los movimientos y el segundo es el de UX relacionado con la personalización, pues el usuario puede crear las categorías según sea su necesidad.



Figura 5. Pantalla principal. Fuente: Elaboración propia, 2021.

La pantalla para ingresar movimientos, Figura 6, presenta un formulario simple, con lo necesario para poder registrarlos. Se inicia por lo principal que es el monto, después la opción para seleccionar si se trata de un ingreso o un gasto, la descripción, la fecha y después una categoría.

En el registro de movimientos se puede observar un patrón en los campos para ingresar datos, lo cual cumple con el propósito mencionado acerca de patrones en un DS. El formulario pretende evitar ser complejo para cumplir con el principio de UI, que menciona que el usuario debe sentirse capaz de utilizar un producto.



Figura 6. Ingreso de movimientos. Fuente: Elaboración propia, 2021.

La pantalla de reportes, Figura 7, muestra el listado de los movimientos realizados en el mes, así como el total resultante de estos. En sus colores, muestra los gastos e ingresos registrados por el usuario. Cada movimiento está compuesto por un cuadro de información, el cual contiene el monto del movimiento, la descripción y la categoría a la cual pertenece.

Con respecto a la pantalla mostrada en la Figura 7, el cuadro de información de cada movimiento contiene los datos necesarios para cumplir con el principio de UI, que destaca mostrar la información suficiente según el objetivo. Además, se cumple con la clasificación en dos puntos, se muestra el monto del movimiento del color, ya sea ingreso o gasto y la categorización a la cual se asignó. Asimismo, se puede observar el uso de los componentes creados, por lo que se cumple con los principios de objetos y patrones.



Figura 7. Listado de movimientos. Fuente: Elaboración propia, 2021.

V. CONCLUSIONES Y RECOMENDACIONES

Se concluye que entre las prácticas vistas las que más destacan de los conceptos de experiencia de usuario y arquitectura de información son experiencia, compatibilidad, personalización, clasificación, ejemplo y navegación. Además, se mencionaron consejos que se relacionan con el desarrollo del producto en el ámbito técnico, tanto como de programación como la implementación de un sistema de diseño. Por esto, se recomienda implementarlas y seguir la definición de cada uno de los principios para garantizar la calidad del producto conforme a los objetivos del usuario.

Entre los beneficios que se pueden obtener al implementar estas buenas prácticas en el desarrollo de un producto se encuentra mejor posicionamiento del producto o empresa, ganar más confianza y lealtad de los usuarios o clientes, mejorar la eficiencia en las tareas que se llevan a cabo y mejorar la experiencia de usuario. Las buenas prácticas en el desarrollo también colaboran con la agilidad en los procesos, como el uso de un sistema de diseño, el cual, después de implementarlo, se pueden reutilizar sus componentes. De esta manera, se recomienda llevar a cabo el esfuerzo para implementar las prácticas y principios mencionados en el artículo. Esto se debe hacer en beneficio y mejora de la vida del usuario para que estos beneficios se reflejen.

El cómo integrar estas prácticas puede depender tanto del contexto del producto como de los usuarios que lo utilizan, además del contexto del mismo principio. A pesar de que son buenas prácticas para garantizar el agrado del usuario, es posible que en alguna de ellas exista mayor dificultad para aplicarlas. Se recomienda analizar las prácticas y principios mencionados en el artículo, con el fin de determinar si estas son aplicables al producto por desarrollar. Asimismo, es importante destacar que en su mayoría se debe considerar estas prácticas entre los requisitos del producto y darles seguimiento.

Existen estándares para medir la usabilidad, entre ellos el más importante es ISO 9241-210, llamado ergonomía de la interacción humano-sistema. El estándar menciona puntos como factores humanos y técnicas de ergonomía. Además, busca garantizar que el usuario entienda el producto y que este cumpla con los objetivos para los cuales se creó. Por este motivo, se recomienda, como lo afirma el estándar, involucrar al usuario durante todo el proceso, tanto de desarrollo como de pruebas, con el fin de buscar una mejora continua, así como realizar evaluaciones con prototipos.

La usabilidad es la capacidad o facilidad de un usuario al utilizar un producto de *software* y las prácticas de experiencia de usuario, arquitectura de información y de desarrollo buscan precisamente eso, mejorar las actividades del usuario al adaptarse a sus necesidades. Por lo tanto, se concluye que al aplicar las prácticas y principios se puede mejorar la usabilidad siempre y cuando se tome al usuario como eje central. Además, se recomienda hacer una revisión antes de comenzar cualquier desarrollo, con el fin de considerar y priorizar los principios en los requerimientos del producto.

REFERENCIAS

- [1] Gómez, «Why Web Performance Matters: Is Your Site Driving Customers Away?» 2010. [En línea]. Available: <https://ohmedia.ca/pub/images/blog/Why-Performance-Matters.pdf>
- [2] J. Toth, «Statistics on user experience» Invisionapp, noviembre 2015. [En línea]. Available: <https://www.invisionapp.com/inside-design/statistics-on-user-experience/>
- [3] R. Hernández Sampieri y C. Mendoza Torres, Metodología de la investigación rutas cuantitativa y mixta, México: McGraw-Hill Interamericana Editores, 2018.
- [4] N. Bevana, J. Kirakowskib y J. Maissela, «What is Usability?» Stuttgart, 1991.
- [5] D. Benyon, Designing User Experience, Pearson, 2019.
- [6] Hassan, «Introducción a la Usabilidad» 1 noviembre 2002. [En línea]. Available: http://www.nosolousabilidad.com/articulos/introduccion_usabilidad.htm
- [7] C. von Saucken, F. Lachner y U. Lindemann, «Principles for User Experience» de International conference on kansai engineering and emotion research, Linköping, 2014.
- [8] D. Siegel y S. Dray, «The map is not the territory: empathy in design» 2 marzo 2019. [En línea]. Available: <https://interactions.acm.org/archive/view/march-april-2019/the-map-is-not-the-territory>
- [9] L. Rosenfeld, P. Morville y J. Arango, Information Architecture, O'Reilly, 2015.
- [10] Modicum, «IA, UI and UX: the anatomy of awesome digital products» Modicum, 15 junio 2021. [En línea]. Available: <https://modicum.agency/blog/ia-ui-and-ux-the-anatomy-of-awesome-digital-products/>
- [11] G. Lucassen, F. Dalpiaz, J. M. van der Werf y S. Brinkkemper, «Improving agile requirements: the Quality User Story framework and tool» de Requirements Eng (2016), Utrecht, 2016.
- [12] D. Brown, «Eight Principles of Information Architecture» 25 mayo 2016. [En línea]. Available: <https://www.designprinciplesftw.com/collections/eight-principles-of-information-architecture>
- [13] Altexsoft, «How to Create Information Architecture for Web Design» Altexsoft, 29 Septiembre 2021. [En línea]. Available: <https://www.altexsoft.com/blog/uxdesign/information-architecture/>
- [14] M. Holladay, «The Eight Principles of Information Architecture» Medium, 16 noviembre 2018. [En línea]. Available: <https://medium.com/@hollabit/the-eight-principles-of-information-architecture-6eff11f907a>
- [15] Amelia, «Simple Guide to Information Architecture for Web Designers» Amelia, 7 marzo 2017. [En línea]. Available: <https://wpamelia.com/information-architecture/>
- [16] O. Gierszal y M. Lichaj, «Information Architecture for Web Design: the Big Picture» Brainhub, 15 abril 2021. [En línea]. Available: <https://brainhub.eu/library/information-architecture-for-web-design/>
- [17] M. Wales, «3 Web Dev Careers Decoded: front-end vs back-end vs Full Stack» Udacity, 8 diciembre 2020. [En línea]. Available: <https://www.udacity.com/blog/2020/12/front-end-vs-back-end-vs-full-stack-web-developers.html>
- [18] S. Xalabrí, «Buenas prácticas del desarrollo FrontEnd» Medium, 8 junio 2015. [En línea]. Available: <https://medium.com/@sergiodxa/buenas-pr%C3%A1cticas-del-desarrollo-frontend-40f44621841>
- [19] C. Fu, «Exploration of Web front-end development technology and optimization direction» de 2nd International Conference on Electronics, Network and Computer Engineering, Hainan, 2016.
- [20] Neoretic, «Single-page application vs. multiple-page application» 2 diciembre 2016. [En línea]. Available: <https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58>
- [21] R. Roşu, «Planning, designing, and developing Design Systems» Uxdesign, 13 August 2020. [En línea]. Available: <https://uxdesign.cc/design-systems-4baa50172c5d>.
- [22] M. Ellis, «7 Steps to Building Your Own Design System from Scratch» Medium, 20 noviembre 2018. [En línea]. Available: <https://uxpin.medium.com/7-steps-to-building-your-own-design-system-from-scratch-4ffadc9841c6>
- [23] W. Fanguy, «A comprehensive guide to design systems» Invision, 24 junio 2019. [En línea]. Available: <https://www.invisionapp.com/inside-design/guide-to-design-systems/>
- [24] ISO, «ISO 9241-210:2019» ISO, [En línea]. Available: <https://www.iso.org/standard/77520.html>
- [25] M. Romero, «Estándares ISO para el diseño centrado en el usuario» Infotecarios, 16 octubre 2020. [En línea]. Available: <https://www.infotecarios.com/estandares-iso-para-el-diseno-centrado-en-el-usuario/>
- [26] D. Travis, «ISO 13407 is dead. Long live ISO 9241-210!» Userfocus, 2011 junio 6. [En línea]. Available: <https://www.userfocus.co.uk/articles/iso-13407-is-dead.html>
- [27] K. Cagiltay, O. Alacam, N. Ocak y F. Erdal, «Developing ISO 9241-151 Product Certification Process: Challenges» de Proceedings of the Second international conference on Design, User Experience, and Usability: *web*, mobile, and product design, Turkey, 2013.
- [28] P. Brooks, «What on Earth is ISO 9241» UXBooth, 24 marzo 2015. [En línea]. Available: <https://www.uxbooth.com/articles/what-on-earth-is-iso-9241/>.
- [29] J. Mifsud, «Usability Metrics – A Guide To Quantify The Usability Of Any System» Usabilitygeek, 22 junio 2015. [En línea]. Available: <https://usabilitygeek.com/usability-metrics-a-guide-to-quantify-system-usability/>

- [30] E. Luchita, «Measuring user experience with usability metrics» Maze, 11 diciembre 2019. [En línea]. Available: <https://maze.co/blog/measure-usability-metrics/>
- [31] M. Batagoda, «Usability Metrics: Measuring UX Design Success» Adobe, 22 mayo 2020. [En línea]. Available: <https://xd.adobe.com/ideas/process/user-testing/usability-metrics-measuring-ux-design-success/>
- [32] M. Christou, «Introducing Proto.io 6» proto.io, 12 julio 2016. [En línea]. Available: <https://blog.proto.io/introducing-proto-io-6/>