

# Metodología para la construcción de Interfaces Gráficas Centradas en el Usuario

Sebastián Sastoque

Universidad Militar Nueva Granada  
Bogotá, Colombia  
sebsasto@gmail.com

Cristian Narváez

Universidad Militar Nueva Granada  
Bogotá Colombia,  
cristian.narvaezing@gmail.com

Germán Garnica

Universidad Militar Nueva Granada  
Bogotá, Colombia  
german.garnica@unimilitar.edu.co

## ABSTRACT

Today, most devices and computer systems use a Graphical User Interface to ease the interaction between users and computers. The creation of Graphical User Interfaces is clearly different from writing software code because different users' factors are involved in the process in order to generate usable interfaces and positive user experience of systems. In fact, although a system that meets all the required functionalities, poor design and defects in the Graphical User Interface generate a bad experience of the whole system. Accordingly, the development of graphical interfaces in large and complex systems is a difficult task, in which developers must face different problems, both at the software level and at the level of specification, interaction and usability of interfaces. In this context, there are solutions of user-centric design, aimed at the development of software elements based on users, that consider people needs, capabilities and issues and the context of use of computational tools. However, these methods are unintuitive and often create frustration in both users and developers for the construction and use of a usable interface. Therefore, this paper presents a methodology for the design of Graphical User Interfaces based on the fundamentals and principles of user-centered design, design thinking, agile development, usability and user experience. In addition, a case study based on an application for people with Complex Communication Needs is proposed, which is intended for future validation and evaluation of methodology.

## RESUMEN

Hoy en día, la mayoría de dispositivos y sistemas de computación utilizan una Interfaz Gráfica de Usuario para viabilizar la interacción entre el usuario y un artefacto, sistema o máquina. La creación de las Interfaces Gráficas de Usuario, es claramente diferente a escribir el código de un software, debido a que intervienen diversos factores de los usuarios para que las interfaces sean usables y generen una experiencia de usuario positiva de todo el sistema. De hecho, a pesar de que un sistema cumpla con todas las funcionalidades requeridas, un mal diseño y defectos en la Interfaz Gráfica generan una mala experiencia de todo el sistema. En consecuencia, el desarrollo de las interfaces gráficas en sistemas grandes y complejos es una tarea difícil de realizar, en ésta los desarrolladores deben enfrentar diferentes problemas tanto a nivel de software como a nivel de especificación, interacción y usabilidad de las interfaces. En este contexto, existen soluciones de diseño centradas en el usuario, que tienen como objetivo que el desarrollo de elementos de software se realice teniendo en cuenta las necesidades, capacidades y problemas de los usuarios y el contexto de uso de las herramientas computacionales. No obstante, estas metodologías son poco intuitivas y en muchas ocasiones crean frustración tanto en el usuario como en los desarrolladores para el

diseño e implementación de una interfaz usable. Por este motivo, este trabajo presenta una metodología, para la construcción de Interfaces Gráficas de Usuarios, basada en los fundamentos y principios del Diseño Centrado en el Usuario, el Pensamiento de Diseño, el Desarrollo Ágil, la Usabilidad y la Experiencia de Usuario. Adicionalmente, se plantea un caso de estudio basado en una aplicación para personas con Necesidades Complejas de Comunicación (NCC), que está pensado para la futura validación y evaluación de la metodología propuesta.

## Categories and Subject Descriptors

- Human-centered computing~Human computer interaction (HCI)~Graphical user interfaces
- Human-centered computing~Interaction design~User interface design
- Human-centered computing~Interaction design~User centered design

## General Terms

Design, Human Factors, Theory

## Keywords

Metodología, Interfaz Gráfica de Usuario, Diseño Centrado en el Usuario, Pensamiento de Diseño, Usabilidad, Experiencia de Usuario, Desarrollo Ágil, Diseño de Interfaces.

## 1. INTRODUCCIÓN

En la última década, debido al rápido desarrollo y madurez de las Tecnologías de la Información y las Comunicaciones, específicamente en el desarrollo de hardware y software, los dispositivos computacionales se han vuelto asequibles para la población en general. Adicionalmente, con la expansión de internet, el número de sitios web, sistemas y aplicaciones, que prestan servicios a las personas en diferentes áreas, ha crecido exponencialmente. En consecuencia, la experiencia de usuario se ha convertido en un elemento diferenciador entre las opciones de aplicaciones y sistemas que existen en el mercado [10]. Ya no se considera suficiente que una aplicación sea efectiva, flexible y de fácil aprendizaje, es necesario que esta se adapte a funciones útiles en la vida cotidiana de las personas, que generen experiencias satisfactorias [20].

En este contexto, una de las áreas altamente relacionadas con la mejora de la experiencia de usuario es la Interacción Humano Computador (HCI por sus siglas en inglés), cuyo objetivo es el diseño, evaluación e implementación de sistemas interactivos eficaces, eficientes y satisfactorios para los usuarios [29]. Uno de

los componentes esenciales del HCI son las interfaces de usuario, las cuales se definen como los medios con que el usuario puede comunicarse con una máquina, equipo, computadora o dispositivo, i.e., comprenden todos los puntos de contacto entre el usuario y el equipo [29]. Entre el conjunto de interfaces de usuario se encuentra las Interfaces Gráficas de Usuario (GUI por sus siglas en inglés), las cuales permiten la interacción del usuario por medio de elementos visuales y objetos gráficos que representan información y acciones disponibles en la interfaz [26]. Una ventaja relevante de las interfaces gráficas de usuario es que hacen el funcionamiento de un dispositivo más intuitivo, y por lo tanto más fácil de aprender y utilizar [16]. Por ejemplo, es más fácil la edición de un archivo de texto utilizando una interfaz gráfica que interactúa con el ratón y el teclado que usar una serie de comandos escritos.

En ese sentido, la construcción de las GUI es una parte esencial del proceso de desarrollo de software y en la mayoría de los casos, las interfaces gráficas, corresponden cerca de la mitad del código desarrollado en un sistema [17]. Así mismo, se ha demostrado que a pesar de que un sistema cumpla con todas las funcionalidades requeridas, un mal diseño y defectos en la GUI generan una mala impresión de todo el sistema [21]. De hecho, la experiencia demuestra que la aceptación por el usuario de un producto de software está determinada en gran medida por la calidad de su interfaz de usuario, en especial su interfaz gráfica, e incluso en mayor medida que por su potencia funcional [30].

Como consecuencia de ello, la calidad de las GUI tiene un gran impacto en el éxito comercial del software. Algunos estudios comprobaron, a partir del análisis de diferentes interfaces gráficas, que frecuentemente se presentan problemas relacionados con el diseño y por lo tanto se afecta negativamente la experiencia del usuario [26]. Esto se debe a que el diseño y construcción de las GUI a menudo se basa, en gran medida, en las propiedades técnicas de los dispositivos y en las suposiciones del desarrollador [30]. No es raro que en la especificación de los requisitos de la interfaz gráfica se definan las tareas y operaciones que el usuario debe poder realizar pero no se establezcan requerimientos para la facilidad de uso del sistema en absoluto [18].

Con respecto a lo planteado anteriormente, los desarrolladores comúnmente se enfrentan a diferentes retos tanto en los procesos de ingeniería de software como del diseño de la interfaz. Estos retos se derivan de la complejidad de los sistemas y los procesos de negocio que se desean soportar con la interfaz [10]. Entre estos retos, las principales dificultades para la creación de buenas interfaces de usuario están relacionadas con cuatro factores: (i) problemas de diseño, (ii) complejidad técnica, (iii) falta de herramientas adecuadas y (iv) reducida propuesta de métodos y metodologías [8, 10, 11]. A pesar de que algunos de estos problemas se han tratado en el estado del arte, la mayoría de las veces se enfocan a soluciones técnicas que no incluyen al usuario final y por lo tanto los problemas de usabilidad siguen presentándose. Por ejemplo, se han propuesto patrones de diseño de alto nivel, relacionados con la arquitectura de los sistemas, como el patrón Modelo/Vista/Controlador (con todas sus variantes) y el patrón Presentación/Abstracción/Control, que especifican la estructura de una aplicación GUI. Aunque este tipo de soluciones ayudan al desarrollador a nivel de organización del software, estos patrones no incluyen elementos funcionales y de facilidad de uso de las interfaces, y por lo tanto los usuarios no se ven involucrados en el desarrollo de las soluciones [10].

Adicionalmente, factores relacionados con la construcción de las interfaces gráficas, el control de las entradas del usuario, la gestión del diálogo del usuario con el sistema, entre otros, se han abordado de una manera insuficiente en el estado del arte, dejando retos de investigación relacionados con la construcción de las interfaces [15]. En este sentido, aún cuando se han propuesto diversas metodologías que incluyen al usuario en la creación de aplicaciones, como el Diseño Centrado en el Usuario (UCD por sus siglas en inglés) o el Pensamiento de Diseño (Design Thinking), estas están dirigidas al proceso general de creación de productos o servicios y no comprenden aspectos relevantes para la construcción de las GUI [27]. La investigación y los métodos para la construcción de interfaces se ha centrado en entender cómo realizar diseños visuales y optimizar las capacidades de los dispositivos de entrada y salida para la interacción con las interfaces gráficas [27]. Teniendo en cuenta lo anterior, se evidencia la necesidad de proponer métodos, para el diseño y construcción de GUIs, que tengan en cuenta al usuario, las capacidades de la tecnología, los conocimientos de los desarrolladores y otros expertos y además se integren a las metodologías y procesos de desarrollo de software comúnmente utilizados en el estado del arte [30, 15, 6, 27].

Con respecto a lo antes planteado, este trabajo presenta una metodología para el diseño y la construcción de interfaces gráficas de usuario, que están altamente relacionadas con el contexto y las capacidades de las personas. La metodología propuesta se basa en principios del Diseño Centrado en el Usuario, el Pensamiento de Diseño, el Proceso de Desarrollo Ágil y conceptos de Usabilidad y Experiencia de Usuario. Adicionalmente, se propone un caso de estudio enfocado a la creación de una GUI para una aplicación de Comunicación Aumentativa y Alternativa (CAA) dirigida a personas con Necesidades de Comunicación Complejas, con el fin de realizar una posterior validación de la metodología propuesta. La presentación de este trabajo está organizada de la siguiente manera: la sección 2 presenta trabajos relacionados con el desarrollo de interfaces centradas en el usuario; la sección 3 introduce la fundamentación teórica de la metodología; la sección 4 expone la metodología para el desarrollo de interfaces gráficas centradas en el usuario; la sección 5 propone un caso de estudio para la futura validación de la metodología; y finalmente la sección 6 presenta conclusiones y trabajo futuro.

## 2. TRABAJOS RELACIONADOS

Algunos autores han trabajado el área del diseño de interfaces gráficas de usuario desde diferentes perspectivas. Por ejemplo, Bentley [2] presenta catorce pasos que funcionan como una guía para crear interfaces en sistemas de tipo SAS. Esos pasos tienen en cuenta la selección de un equipo de diseño en el cual puede participar el usuario final, el análisis de las actividades y los usuarios, métodos de creación de ideas, selección de los componentes de la interfaz, prototipado y una evaluación que se realiza después de haber lanzado el sistema. Piasecki y Piezka [25] presentan una metodología para crear software con altos niveles de usabilidad teniendo en cuenta las metodologías para el diseño de interfaces, 12 arquetipos, análisis en estrella y Análisis y Diseño Orientado a Objetos Complejos (COOAD por sus siglas en inglés). Para identificar las necesidades y habilidades del usuario se hace una descripción y análisis de su contexto y para asegurar la efectividad de las interfaces propuestas se realizan evaluaciones de diferentes prototipos antes de la implementación tecnológica final. De otro lado, Morales [22] presenta un método que busca evitar errores asociados al descuido en algunos aspectos importantes de

las GUI y que se dan por los procesos de creatividad e iniciativa de los diseñadores. La propuesta tiene que ver con capas definidas al inicio del proyecto para que el diseñador tenga en cuenta la organización y relación entre los elementos de la interfaz, cómo el usuario va a interactuar con los componentes y cómo se va a informar al usuario de cambios en la interfaz. Los autores del trabajo proponen crear mínimo un prototipo para cada capa del proceso de desarrollo. Finalmente, Labib et.al. [17] presenta una interfaz de usuario temprana, que acompaña a la metodología de Desarrollo Temprano de Interfaces de Usuario (EUID por sus siglas en inglés), que tiene como objetivo permitir a los desarrolladores que se concentren en la lógica de la aplicación. La metodología tiene en cuenta una serie de requerimientos que son prototipados y evaluados por el usuario, que al ser aceptados, los prototipos son diseñados y construidos para el lanzamiento de la interfaz. Este proceso se da de forma iterativa.

Aunque en propuestas como [2, 25, 22] se tiene en cuenta al usuario y sus necesidades, no se presenta de forma explícita el rol que tiene el usuario en etapas diferentes a la validación del sistema o la obtención de la información para definir los requerimientos del sistema. Algunas metodologías formalizan la necesidad de crear un grupo multidisciplinario para el desarrollo de la GUI, lo anterior se convierte en una desventaja para algunos trabajos que solo ven el diseño de la interfaz como trabajo del diseñador y no de los desarrolladores e incluso del usuario final. También es necesario mencionar la tendencia existente de proponer el prototipado y evaluación constante de los diseños planteados por el equipo de desarrollo, normalmente estas evaluaciones son desarrolladas por los usuarios finales y el desarrollo se requiere hacer de forma iterativa.

### 3. FUNDAMENTACIÓN TEÓRICA DE LA METODOLOGÍA

La metodología para la creación de interfaces gráficas centradas en el usuario está fundamentada como se muestra en la Figura 1. La base de la metodología son los conceptos, principios y fundamentos del Diseño Centrado en el Usuario y los tres pilares que apoyan sus procesos son: (i) conceptos de Pensamiento de Diseño, (ii) fundamentos de Metodologías de Desarrollo Ágil y (iii) principios de Usabilidad y Experiencia de Usuario. En las siguientes subsecciones se realiza un análisis y resumen de cada uno de los fundamentos teóricos utilizados para la propuesta de la metodología.

#### 3.1 Diseño Centrado en el Usuario

El Diseño Centrado en el Usuario es un enfoque de diseño multidisciplinario, basado en la participación activa de las personas, que busca mejorar la comprensión de las necesidades del usuario y la tarea a través de la iteración de los procesos de diseño y evaluación [24]. El UCD es considerado ampliamente como la clave para el diseño y desarrollo de productos y servicios con altas condiciones de usabilidad y satisfacción del usuario [24]. El Diseño Centrado en el Usuario, enfocado al diseño de interfaces de usuario, se centra en objetivos de usabilidad, características de los usuarios, ambientes, tareas y el flujo de trabajo.



Figura 1. Bases Metodológicas de la propuesta de la metodología.

El UCD se basa normalmente en 3 principios básicos [28]: (i) un enfoque temprano en los usuarios y las tareas, donde se recopile información estructurada y sistemática con el apoyo de expertos en el área (psicólogos, psiquiatras, diseñadores, etc...); (ii) medición empírica y evaluación del uso del producto con usuarios reales, a través de prototipos, cuyo objetivo debe ser medir la facilidad de aprendizaje y uso; y (iii) el diseño iterativo, donde el producto es modificado y probado repetidamente y a partir de las pruebas con prototipos de los modelos conceptuales e ideas de diseño se realice una completa revisión y replanteamiento del diseño del producto. Adicionalmente, la Organización Internacional para la Estandarización (ISO por sus siglas en inglés) propone, en la parte 210 de su norma ISO 9241, seis elementos para que un sistema se clasifique como centrado en el usuario [13]:

1. El diseño está basado en una comprensión explícita de usuarios, tareas y entornos.
2. Los usuarios están involucrados durante el diseño y el desarrollo.
3. El diseño está dirigido y refinado por evaluaciones centradas en usuarios.
4. El proceso es iterativo.
5. El diseño está dirigido a toda la experiencia del usuario.
6. El equipo de diseño incluye habilidades y perspectivas multidisciplinarias.

Normalmente, el UCD sigue una serie de métodos y técnicas que son bien definidas para el proceso de análisis, diseño y evaluación tanto del software como del hardware en el que intervienen las interfaces de usuario [11]. Aunque existen variaciones en las metodologías que se basan en el Diseño Centrado en el Usuario, normalmente se mantienen los principios y técnicas básicas en el proceso de diseño. Comúnmente una metodología basada en UCD, posee las siguientes etapas, que se muestran en la Figura 2:

1. **Análisis:** En esta etapa se identifican los usuarios, sus necesidades, y los ambientes en donde se desarrollarán las actividades. Adicionalmente, se definen los objetivos, retos,

actividades, contenidos y flujos de tareas que se espera que los usuarios logren por medio de la solución propuesta.

2. **Diseño:** En esta fase se realiza una especificación del producto y la importancia de cada uno de los componentes de la actividad. En este proceso se obtienen prototipos que permiten evaluar la propuesta de diseño. Se sugiere que el tiempo dedicado a esta fase sea dividido en 10% para presentación, 30% para interacción y 60% para el modelo conceptual.
3. **Evaluación:** En esta etapa se busca la evaluación de los prototipos propuestos por parte del usuario final. Este proceso es iterativo junto con la etapa de diseño, ya que se busca llegar a un producto que cumpla con los principios del diseño centrado en el usuario y efectivamente, sea fácil de usar. Se pueden realizar diferentes tipos de pruebas y lo ideal es que el usuario de todas las opiniones posibles.
4. **Implementación:** Luego de obtener el mejor prototipo posible con la iteración de las fases anteriores, en esta etapa se realiza el desarrollo e implementación del producto final.
5. **Despliegue:** Esta es la fase final del proceso de diseño donde el producto es llevado nuevamente al usuario final para su uso.



**Figura 2. Etapas de las metodologías de diseño centradas en el usuario.**

### 3.2 Pensamiento de Diseño (Design Thinking)

El Pensamiento de Diseño, es una metodología utilizada para crear ideas innovadoras que centra su eficacia en entender y plantear soluciones a las necesidades reales de los usuarios. En términos sencillos, es una disciplina que usa la sensibilidad y los métodos del diseñador para hacer coincidir las necesidades de las personas con lo que es tecnológicamente factible y con lo que una estrategia de negocios puede generar valor al cliente [13]. En términos generales, el pensamiento de diseño primero define un problema y luego implementa soluciones siempre con las necesidades de los usuarios como núcleo del desarrollo de conceptos. La metodología se centra en la búsqueda de necesidades, la comprensión, la creación, el pensamiento y el hacer. El fundamento del Pensamiento de Diseño es la acción y la creación, es decir, siempre busca crear y probar algo para seguir aprendiendo del usuario y mejorar el proceso de generación de ideas [3].

La metodología del Pensamiento de Diseño posee cinco factores diferenciales a otras metodologías [3]: (i) la generación de empatía, donde se busca entender los problemas, necesidades y deseos de los usuarios implicados en la solución; (ii) el trabajo en equipo, donde se busca que profesionales con distinta formación aporten al desarrollo de la solución de una manera singular desde su campo; (iii) la generación de prototipos, que ayudan a la validación y mejoras de las ideas; (iv) un ambiente lúdico, en el cual se busca trabajar bajo una atmósfera que permita disfrutar el proceso y permitir que cada profesional exponga al máximo su potencial; y (v) el uso de técnicas con gran contenido visual, que ayuden a proponer soluciones innovadoras y factibles desde el proceso de análisis y validación de las ideas.

La metodología de Pensamiento de Diseño, trabaja principalmente con tres escenarios [5]: el primero, el escenario de *Inspiración*, en el cual se busca la identificación de la necesidad y el problema y se analizan las opciones que tiene el equipo de trabajo para solucionarlas; el segundo, el escenario de *Ideación*, en el cual se busca el desarrollo de las ideas a través de modelos y prototipos que son validados y ajustados constantemente con el usuario final; y el tercero, el escenario de *Implementación*, se desarrolla la solución final de acuerdo a las visiones desarrolladas con los prototipos. Adicionalmente, la metodología de Pensamiento de Diseño define las siguientes cinco etapas, que se articulan con los tres escenarios explicados anteriormente [4], como se muestra en la Figura 3:

1. **Empatizar:** Este proceso busca comprender las necesidades de los usuarios implicados en la solución y su entorno.
2. **Definir:** En esta etapa se procesan y sintetizan todas revelaciones encontradas en la etapa anterior con el fin de formar una perspectiva clara para la creación de la solución.
3. **Idear:** En este proceso se busca explorar una amplia variedad de soluciones posibles a través de la generación de ideas para permitir ir más allá de lo obvio y explorar soluciones diferentes. En esta fase es importante que se eliminen los juicios de valor y que se favorezca el pensamiento expansivo.
4. **Prototipar:** En esta etapa se busca transformar las ideas en una forma física, en la que se pueda experimentar e interactuar con la solución propuesta, para validar las ideas y aprender del proceso para mejorar o refinar la solución y en caso de ser necesario desarrollar más empatía para generar más ideas.
5. **Probar:** En esta fase se realizan pruebas de los prototipos con los usuarios implicados en la solución con el fin de utilizar sus observaciones y comentarios para refinar los prototipos, identificar mejoras significativas, fallos a resolver y posibles carencias hasta llegar un prototipo que se acople a las necesidades y contexto del usuario.

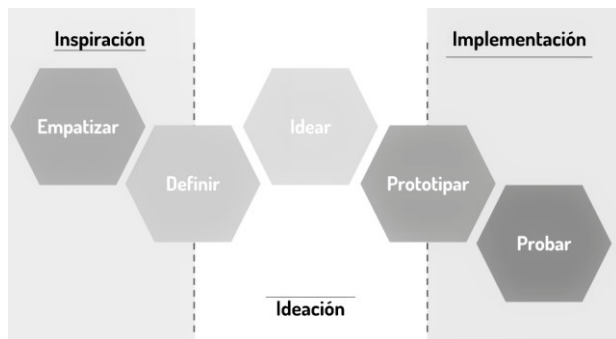


Figura 3. Etapas de la metodología del Pensamiento de diseño.

### 3.3 Desarrollo ágil

Las metodologías de desarrollo ágil describen un conjunto de principios para el desarrollo de software en las que las necesidades y soluciones evolucionan a través del tiempo [19]. En este tipo de metodologías se promueve la planificación adaptativa, el desarrollo evolutivo, el despliegue prematuro, la mejora continua y la respuesta rápida y flexible a los cambios [19]. Comúnmente, las metodologías ágiles distribuyen actividades de forma incremental para revisar funcionalidades y acondicionar las mismas a los cambios de requerimientos que se den durante el desarrollo del software. Esto permite entregar componentes de software constantemente al cliente para su evaluación.

Para formalizar el desarrollo ágil, en el año 2001 un grupo de desarrolladores crearon un manifiesto basado en 4 valores fundamentales [1]: (i) Individuos e interacciones sobre procesos y herramientas, (ii) Software funcionando sobre documentación extensiva, (iii) Colaboración con el cliente sobre negociación contractual y (iv) Respuesta ante el cambio sobre seguir un plan. Adicionalmente, el manifiesto se basa en doce principios que buscan cumplir los valores enunciados anteriormente [1].

En el estado del arte se han propuesto diferentes metodologías, que buscan cumplir el manifiesto de desarrollo ágil, entre las que se encuentran: Scrum, Programación Extrema (XP por sus siglas en inglés), Proceso Racional Unificado (RUP por sus siglas en inglés), Proceso Ágil Unificado (AUP por sus siglas en inglés), el Desarrollo Impulsado de Funciones (FDD por sus siglas en inglés) y Método de Desarrollo de Sistemas Dinámicos (DSDM por sus siglas en inglés), entre otros [7]. Aunque cada una de estas metodologías presentan características diferenciadoras, la mayoría se basan en los siguientes fundamentos [7]:

- **Procesos Iterativos, incrementales y evolutivos:** El trabajo de desarrollo se fragmenta en productos pequeños, que incrementalmente van cumpliendo con todas las funcionalidades del sistema. Estos productos incrementales se desarrollan en iteraciones, las cuales son cortos periodos de tiempo, en los que el fragmento del producto se desarrolla a través de procesos de planificación, análisis, diseño, implementación, pruebas unitarias y pruebas de aceptación. El objetivo de este fundamento es tener una versión disponible (con errores mínimos) al final de cada iteración. En la mayoría de los casos se requieren de varias iteraciones que van evolucionando hasta tener un producto completo.
- **Comunicación Eficiente y Personal:** Para el desarrollo del producto siempre se debe incluir un representante del cliente. Esta persona debe estar disponible cuando el desarrollador lo necesite para responder preguntas en cada iteración.

- **Empoderamiento:** Las personas involucradas en el desarrollo del software deben estar en la potestad de tomar decisiones rápidas en el proceso del desarrollo para completar la iteración.
- **Ciclos de retroalimentación y adaptación cortos:** Deben existir ciclos de retroalimentación cortos. Para ello se recomienda que periódicamente se realice una breve sesión en la que los miembros del equipo presenten entre sí sus avances hasta el momento, el plan de trabajo para cumplir la meta y cualquier obstáculo o impedimentos que puedan afectar el cumplimiento de la iteración.
- **Enfoque de Calidad:** En cada iteración se debe asegurar que el fragmento de producto resultante esté en condiciones óptimas de calidad y se pueda integrar fácilmente al desarrollo del producto final.
- **Adaptación:** Se debe contar con procesos que sean flexibles y fácilmente adaptables a través del tiempo.

### 3.4 Principios de Usabilidad y Experiencia de Usuario

La usabilidad se define como el grado en que un producto puede ser usado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso [12]. Es decir, la Usabilidad se refiere a la calidad de la experiencia de un usuario al interactuar con los productos o sistemas, incluyendo sitios web, software, dispositivos o aplicaciones. Consecuentemente, en la literatura se han definido los siguientes objetivos de la usabilidad [28]: (i) *Utilidad*, es decir, la cualidad del producto para permitir a los usuarios lograr sus objetivos; (ii), *Eficacia*, entendida como la medición cuantitativa de cómo un usuario logra la tarea; (iii) *Facilidad de Aprendizaje*, es decir, la capacidad del usuario para operar el sistema a un cierto nivel de competencia definido después de un período predeterminado de entrenamiento; y (iv) *Simpatía*, al integrar las percepciones, sentimientos y opiniones del usuario acerca del producto. Adicionalmente, en el proceso de diseño enfocado en usabilidad se basa en los siguientes principios [9]:

- **Enfoque temprano en Usuarios y Tareas:** Se debe entender quiénes son los usuarios y sus tareas. En este enfoque se debe comprender las características cognitivas, comportamentales, antropométricas y actitudinales de los usuarios, así como la naturaleza del trabajo y las tareas que se quieren cumplir.
- **Mediciones Empíricas:** Los usuarios potenciales deben ayudar en la medición de simulaciones y prototipos que estén relacionados con la realización de la tarea en un contexto real. Su rendimiento y reacciones deben ser observados, registrados y analizados.
- **Diseño Iterativo:** En el proceso de propuesta de un producto se debe implementar un ciclo de diseño, prueba, evaluación, y rediseño, que se repita tantas veces como sea necesario.

Teniendo en cuenta lo anterior, el objetivo final de la usabilidad es mejorar la Experiencia de Usuario (UX por sus siglas en inglés), la cual se define como las respuestas y percepciones de una persona que resultan del uso común o anticipado de un producto, sistema o servicio [13]. Para lograr una buena experiencia de usuario se propone que exista un balance entre tres componentes claves de la experiencia: el usuario, el contexto y el contenido. Para ello, en el estado del arte se ha adoptado el Panel de la Experiencia de

Usuario (UX Honeycomb) para el diseño de experiencias, que se muestra en la Figura 4, el cual básicamente muestra visualmente un panel de siete células, en la que cada una contiene los diferentes aspectos y atributos principales de una Experiencia de Usuario satisfactoria, que se listan a continuación [23]:

- **Útil:** El diseño debe ofrecer nuevas funcionalidades, mejores a las soluciones existentes, para empoderar al usuario a cumplir objetivos reales.
- **Deseable:** La solución debe ser presentada de tal manera que el usuario tenga una preferencia y gusto con respecto a otras soluciones.
- **Accesible:** El diseño debe poder ser utilizado por la mayor cantidad de usuarios posibles. La curva de aprendizaje debe ser pequeña con respecto a modelos y soluciones existentes.
- **Creíble:** Los elementos de diseño deben influir a que los usuarios confíen y crean en la solución propuesta.
- **Alcanzable:** Los elementos del diseño deben estar dispuestos de tal manera que las personas puedan encontrarlos y utilizarlos fácilmente. El usuario debe poder utilizar la solución de manera inmediata, independientemente si existe una limitación en el éxito y la productividad, hasta que logre apropiarse completamente de la solución.
- **Utilizable:** El diseño debe funcionar correctamente, sin problemas técnicos o fallas importantes.
- **Valioso:** El diseño debe ofrecer valor para la organización que lo distribuye a los usuarios. Si el valor es no lucrativo se debe traducir en el cumplimiento de la misión de la organización y si es lucrativo este debe contribuir con las líneas de negocio y mejorar la satisfacción del cliente.

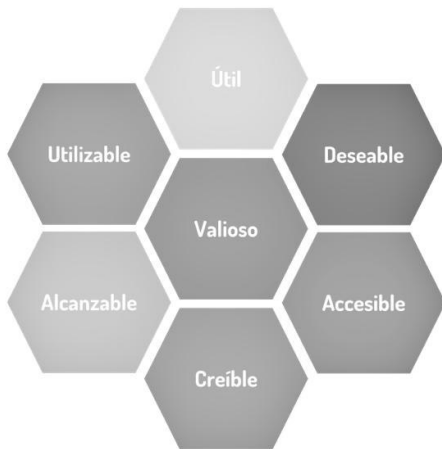


Figura 4. Panel de Experiencia de Usuario

Teniendo en cuenta los atributos y los principios enunciados anteriormente, el proceso de diseño de la Experiencia de Usuario debe estar enfocado en asegurar que ningún aspecto del UX, con una solución o producto, pase sin que este se haya planeado explícitamente en el diseño de la experiencia [8]. Esto significa, que se deben considerar todas las posibilidades de cada acción que el usuario puede hacer y comprender cuales son las expectativas del usuario en cada paso del camino que él elija seguir durante su experiencia. Para ello, se ha propuesto en la literatura, dividir los

componentes de la experiencia del usuario y así lograr entender mejor las tareas de la solución.

En el área del diseño de interfaces, una propuesta es dividir la experiencia en cinco planos: (i) Superficie, Esqueleto, Estructura, Alcance y Estrategia [23]. Debido a la dualidad de los productos de software se ha propuesto que estos planos sean divididos en dos tipos: (i) funcional, que ve el producto como un conjunto de herramientas que los usuarios utilizan para cumplir una serie de tareas y (ii) informativo, que ve el producto como un medio para que los usuarios encuentren, absorban y utilice información [23]. La Figura 5, muestra las tareas que se deben realizar en cada uno de los planos teniendo en cuenta la división explicada anteriormente.

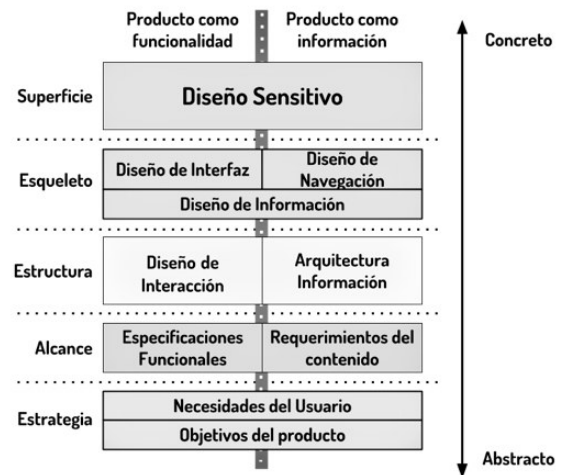
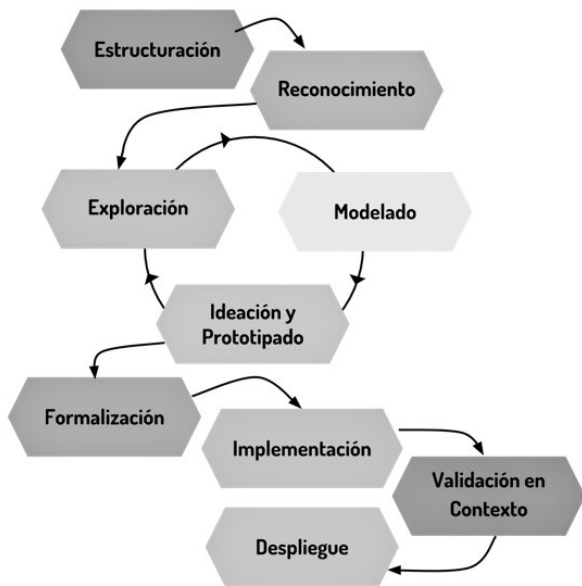


Figura 5. Planos de la experiencia de usuario para el diseño de interfaces Web

#### 4. METODOLOGÍA PARA LA CREACIÓN DE INTERFACES GRÁFICAS CENTRADAS EN EL USUARIO

Como se explicó anteriormente, la Metodología para la Creación de Interfaces Gráficas Centradas en el Usuario (ver Figura 6) se fundamenta en los principios del Diseño Centrado en el Usuario y tiene como pilares principios del Pensamiento del Diseño, Desarrollo Ágil, Usabilidad y Experiencia de Usuario. Esta propuesta parte del supuesto que existe un proyecto de desarrollo de aplicaciones o software, que se puede encontrar en una fase de ideación, planteamiento, diseño, implementación, evaluación o innovación, que requiere del uso de interfaces gráficas para la comunicación e interacción entre el usuario y el sistema.



**Figura 6. Fases de la Metodología para la construcción de Interface Gráficas Centradas en el Usuario**

A partir de la articulación de los diferentes conceptos tomados de la fundamentación teórica, la metodología propuesta tiene los siguientes atributos y principios:

1. El usuario interviene constantemente en el proceso asumiendo diferentes roles y diversos grados de incidencia.
2. Las actividades que realiza el usuario, en su contexto, es elemento funcional esencial que debe soportar la interfaz gráfica.
3. Buscar un enfoque diferente a las soluciones actuales para plantear ideas de solución.
4. Se deben realizar iteraciones con una duración de tiempo mínima que permita el desarrollo y comprobación de una idea solución.
5. Medición constante de la experiencia del usuario y la usabilidad de la solución teniendo en cuenta las características de eficiencia, eficacia, efectividad y satisfacción.
6. Todas las ideas de solución son válidas hasta que el usuario objetivo y el propósito de la actividad demuestren lo contrario.
7. La solución asume restricciones por las limitaciones de la tecnología en el proyecto, pero debe procurar impulsar innovaciones desde su diseño.
8. Las personas que intervienen en el desarrollo de la solución deben ser curiosas, observadoras, críticas y propositivas.
9. Existe una serie de requerimientos mínimos que debe seguir el diseño desde el inicio del desarrollo.
10. El desarrollo de las fases de la metodología es Iterativo e incremental.
11. El prototipo ideal y la solución debe cumplir con las características del Panel de la Experiencia de Usuario [12] y los objetivos de la usabilidad [28].

La metodología comprende 9 fases, como se muestra en la Figura 6: *Estructuración, Reconocimiento, Exploración, Modelado, Ideación y Prototipado, Formalización, Implementación, Validación en Contexto y Despliegue*. Cada fase posee un objetivo particular que está articulado con un conjunto de etapas y entregables o resultados esperados. Para el desarrollo de cada una de las fases y sus etapas, se propone adaptar las estrategias del método Planear-Hacer-Verificar-Actuar (PDCA por su siglas en inglés), utilizado en planes de mejoramiento continuo, donde en cada ciclo se realizan las siguientes tareas [14]:

- **Planear:** Se realiza una planeación de la fase y las actividades a realizar, recopilando la información necesaria y estableciendo los objetivos y metas a alcanzar.
- **Hacer:** Se realizan las actividades de acuerdo a lo propuesto en el plan.
- **Verificar:** Se verifica que el resultado de la fase de hacer cumpla con lo planteado en la fase de planear.
- **Actuar:** Se utilizan los resultados obtenidos para alimentar el macro proceso y en caso de ser necesario se vuelve a realizar el ciclo para obtener mejores resultados.

Adicionalmente, algunas herramientas de diseño, que ayudan a la creación de representaciones, obtención datos, relación y creación información y creación de ideas, deben ser entendibles por todas las personas que intervienen en el desarrollo de cada una de las fases y etapas propuestas. De esta manera, de acuerdo a las necesidades para la implementación de la metodología se enuncian a continuación una lista de herramientas de diseño sugeridas: Mapa de actores, Inmersión Cognitiva, Interacción Constructiva, Mapa Mental, Observación encubierta, ¿Qué, Cómo, Porqué?, Entrevistas (Individuales, Expertos, Grupales, Entre usuarios objetivos), Usuarios Extremos, Mapa de Empatía, SCAMPER, Focus Groups, Mapa del presente y Futuro, Saturar y Agrupar, Observaciones, Diagramas, Reuniones, Diagramas, Escenarios de uso, Prototipos (Baja Fidelidad, Alta Fidelidad, Exploratorio, Experimental, Operacional, Guiado, Mago de Oz, Interactivo, Funcional, Animado, etc...), Mock-ups, Líneas de experiencia, Storyboards, Collages, Planes de Proyectos, Dibujos, Inmersión, Lluvia de ideas, Mash-ups, Sesiones de Creación y Juego de Roles, entre otros.

En las siguientes subsecciones se introducen cada una de las fases de la metodología, se describe su objetivo, se plantean sus etapas y se introducen los resultados esperados de cada fase.

#### 4.1 Fase de Estructuración

Lograr una buena ejecución de la metodología requiere que exista una planeación de las actividades y una definición de elementos de trabajo. En consecuencia, esta fase consiste en estructurar una estrategia que permita ejecutar y completar todas las fases de la metodología. El objetivo de esta fase es establecer los procesos de gestión y los elementos de trabajo para la ejecución de las diferentes fases de la metodología. Considerando lo anterior, en esta fase se proponen las siguientes etapas:

1. Analizar los antecedentes del proyecto de desarrollo de software.
2. Definir el tiempo para la construcción de la interfaz.
3. Establecer el equipo de trabajo.

4. Fijar los recursos y materiales.
5. Identificar los espacios de trabajo.
6. Construir la estructura de trabajo.

El resultado de la fase de Estructuración es un conjunto de lineamientos estructurados que guiarán todas las fases del proceso de creación de la interfaz.

## 4.2 Fase de Reconocimiento

El reconocimiento y caracterización del usuario es el elemento más importante para la construcción de interfaces centradas en el usuario. Consecuentemente, esta fase consiste en realizar un estudio del usuario para entender sus necesidades y su entorno. El objetivo de la fase de Reconocimiento es establecer un conjunto de características del usuario (cognitivas, comportamentales, físicas, emocionales, etc), de las actividades y tareas que realiza en las que interviene la interfaz y del contexto en que realiza dichas actividades. A partir de esto, en esta fase se proponen las siguientes etapas:

1. Conocer al usuario.
2. Definir las necesidades y problemas del usuario.
3. Establecer el contexto de uso.
4. Determinar el conjunto de actividades o funciones que realiza el usuario en función de su objetivo.
5. Realizar la caracterización del usuario, contexto de uso y actividades.

El resultado de la fase de Reconocimiento es la caracterización del usuario objetivo, el contexto de uso y las actividades que necesita desarrollar por medio de la interfaz.

## 4.3 Fase de Exploración

Para crear soluciones significativas es necesario conocer a los usuarios y su desempeño en el desarrollo de la actividad. No solo basta tener una fundamentación teórica de quien es el usuario, se hace necesario el interactuar con él para entender a profundidad cómo realiza las tareas y los retos a los que se enfrenta. El objetivo de esta fase es analizar la ejecución de las actividades que realiza el usuario para definir elementos claves requeridos en la construcción de la interfaz. En ese sentido, se plantea el siguiente conjunto etapas:

1. Observar un conjunto de usuarios objetivos realizando la actividad o alguna de sus funciones para identificar intuiciones (insights) sobre elementos relevantes para la interfaz.
2. Recibir del usuario opiniones, ideas, sugerencias, retos o problemas que afronta desde su experiencia para realizar la actividad o alguna de sus funciones.
3. Identificar factores determinantes en el desarrollo de la actividad como patrones, elementos claves, conductas, comportamientos o protocolos, entre otros.
4. Realizar un análisis y conclusión de las observaciones realizadas.

El resultado de la fase de Exploración es un análisis de los factores determinantes en la actividad junto con las características y elementos a considerar en el desarrollo de la interfaz.

## 4.4 Fase de Modelado

Luego de comprender y conocer al usuario, es importante crear una coherencia sobre la información recolectada hasta el momento, es decir, se debe enmarcar el problema adecuadamente para crear una solución correcta. Esta fase consiste en procesar y sintetizar la información, obtenida de la fase de Exploración, para hacer relaciones y descubrir patrones racionales del usuario. El objetivo de esta fase es construir un modelo de interfaz, a partir de la creación de un enfoque o concepto de diseño, que servirá para orientar las ideas y prototipos de solución. Por lo tanto, el conjunto de etapas propuestas en esta fase es el siguiente:

1. Especificar los requerimientos del usuario y actividad para el desarrollo del modelo de interfaz.
2. Identificar los componentes y elementos utilizados durante la ejecución de las actividades analizadas.
3. Establecer la función y relación de los componentes identificados.
4. Determinar acciones con los componentes y elementos para especificar posibles formas de interacción.
5. Definir el modelo y alcance de la interfaz.

El resultado de la fase de Modelado es un modelo de interfaz el cual incluye componentes, requerimientos, funciones y relaciones de la interfaz.

## 4.5 Fase de Ideación y Prototipado

A partir de la comprensión y apropiación del modelo de interfaz, se debe buscar el mayor número de ideas de solución posibles para ponerlas a prueba con el usuario y alcanzar su mayor productividad. Esta fase consiste en concebir una gran cantidad de ideas propuestas, que dan muchas alternativas para elegir como posibles soluciones, en vez de trabajar en una sola mejor solución. El objetivo de esta fase es construir el prototipo ideal de interfaz, a partir de la generación de múltiples ideas de solución, que son validadas con el usuario objetivo. El conjunto de etapas propuestas a desarrollar para cumplir dicho objetivo son:

1. Realizar actividades encaminadas a la generación de ideas de solución.
2. Establecer un esquema (disposiciones, relaciones y acciones posibles) que involucre los componentes del modelo para cada una de las ideas generadas.
3. Establecer determinantes de lenguaje y estilo o aspecto para las ideas de solución.
4. Definir la tipología del prototipo de cada idea planteada.
5. Crear los prototipos y asegurar sus funcionalidades con respecto a la actividad realizada por el usuario.
6. Evaluar el prototipo en contexto con usuarios objetivo.
7. Concluir y refinar prototipos e ideas (repetir las etapas 1 a la 6) hasta obtener un conjunto reducido de prototipos ideales.
8. Describir detalladamente los prototipos ideales y asignarles una prioridad a partir de la productividad del usuario para realizar la actividad.

El resultado de la fase de Ideación y Prototipado es la descripción de un conjunto de prototipos ideales de la interfaz, junto con una



prioridad de formalización, que tienen en cuenta todos los lineamientos considerados en el modelo.

#### 4.6 Fase de Formalización

Luego de obtener un conjunto de prototipos ideales, teniendo en cuenta su prioridad, debe realizarse un proceso de formalización que considere la viabilidad técnica y tecnológica, utilizando indicadores que permitan evaluar y elegir la mejor solución posible. Esta fase consiste en la evaluación de la viabilidad de los prototipos ideales para precisar elementos de diseño que debe poseer la solución final. El objetivo de esta fase es formalizar un diseño final de interfaz que permita la interacción en pro de la realización de las actividades por parte del usuario. De esta manera, se propone el siguiente conjunto de etapas:

1. Identificar las tecnologías actuales para la implementación de los prototipos y determinar sus alcances.
2. Realizar un mapeo entre las tecnologías y los prototipos para elegir los prototipos que son viables y convenientes.
3. Definir de los elementos de aspecto, estilo y lenguaje para las modalidades sensoriales de la interfaz.
4. Validar en contexto con el usuario los prototipos modificados con los elementos de la etapa 3, realizar mejoras y elegir el diseño de interfaz final.
5. Descripción formal de la interfaz final.

El resultado de la fase de Formalización es la descripción formal de una interfaz final, con todos los elementos de aspecto, estilo y lenguaje que está lista para implementación.

#### 4.7 Fase de Implementación

A partir de la descripción formal de la interfaz se debe realizar la implementación de la solución. Esta fase consiste en el desarrollo tecnológico de la interfaz para realizar los procesos de validación en contexto. Es importante aclarar, que no es necesario realizar una implementación de la interfaz que esté conectada con otros componentes del software, es decir, la interfaz debe cumplir con todas las funciones del software pero se pueden utilizar datos estáticos para su validación. El objetivo de esta fase es implementar la solución final para realizar la validación en contexto. Las etapas propuestas para cumplir este objetivo son:

1. Definir una metodología para la implementación tecnológica de la interfaz, que puede ser la misma utilizada en el desarrollo del software.
2. Implementar la interfaz utilizando la metodología definida en la etapa 1.
3. Realizar una validación funcional de la interfaz implementada.

El resultado de la fase de Implementación es la implementación funcional de la interfaz que tiene en cuenta todos los lineamientos de la descripción formal.

#### 4.8 Fase de Validación en Contexto

Con el fin de integrar la interfaz a la solución de software, es necesario evaluar que la GUI cumpla con todos los requerimientos y necesidades especificados en el modelo de la interfaz. Esta fase consiste en realizar una evaluación en contexto del usuario objetivo para realizar los ajustes finales de la integración de la interfaz en la solución del software. El objetivo de esta fase es validar la

usabilidad y la experiencia de usuario con la solución de interfaz propuesta en la actividad en contexto para la que se desarrolló. En esta fase se proponen las siguientes etapas:

1. Definir las pruebas y los protocolos en función de la verificación en contexto de la interfaz propuesta.
2. Realizar pruebas con diferentes grupos de usuarios, entre los que se incluyen usuarios objetivo y otro tipo de usuarios, para identificar y realizar mejoras posibles.
3. Evaluar por medio de pruebas en contexto con los usuarios objetivo la solución a ser integrada al software o aplicación.
4. Realizar las modificaciones finales, en caso de existir, a la solución de interfaz a partir de la retroalimentación de la evaluación en contexto con los usuarios objetivo.

El resultado de la fase de Validación en Contexto es la interfaz final, validada con usuarios objetivo, que puede ser integrada al proceso de desarrollo de software.

#### 4.9 Fase de Despliegue

Finalmente, se debe integrar la GUI al desarrollo o producto de software. Esta fase consiste en la integración de la solución de interfaz propuesta al desarrollo de software. El objetivo de esta fase es desplegar e integrar la interfaz a la solución final. Las etapas propuestas para esta fase son:

1. Apropiar la metodología de implementación propuesta en el proyecto de desarrollo de software.
2. Establecer comunicación con los demás miembros del equipo de desarrollo para exponer y explicar las funcionalidades de la interfaz.
3. Implementar conexiones e integración con los demás componentes de software.
4. Validar la funcionalidad de la interfaz, en relación a la actividad del usuario y los demás componentes y funciones del software.

El resultado de la fase de Despliegue es la interfaz final integrada al desarrollo de software.

### 5. CASO DE ESTUDIO

Esta sección propone un caso de estudio, para la validación de la metodología, que está basado en una aplicación de Comunicación Aumentativa y Alternativa para personas con Necesidades de Comunicación Complejas. A continuación se explica el contexto del caso de estudio y se plantean los retos que están relacionados con la creación de Interfaces Gráficas en los que la aplicación y uso de la metodología presentan ventajas considerables para el desarrollo y construcción de la interfaz.

La comunicación es una necesidad vital para el ser humano, ya que por su naturaleza, el hombre es un ser social que pertenece a una comunidad y está en constante relación con sus semejantes. Esto implica una reciprocidad interpersonal de transmitir, recibir e intercambiar intencionalmente ideas, sentimientos, pensamientos, opiniones e información en general, con otros. La manera más fácil en la que el hombre puede comunicarse está dada por un código específico conocido como el lenguaje, el cual está compuesto principalmente por la comunicación verbal y escrita, generalmente acentuada por el contexto socio-cultural, permitiendo a las

personas desarrollar su intelecto, valores, principios y conocimientos.

Puntualmente existen diferentes enfermedades derivadas de áreas afectadas en el cerebro (trastornos neurolingüísticos) como la afasia, apraxia, agrafia, entre otras, que generan limitaciones en la comprensión y producción del lenguaje oral y escrito. Para suplir estas Necesidades Complejas de Comunicación (NCC) durante varios años se han planteado los Sistemas de Comunicación Aumentativa y Alternativa (CAA), los cuales buscan aumentar la capacidad de comunicación por medio de técnicas y métodos alternativos a la forma oral ó escrita. Dichos sistemas varían en las tecnologías utilizadas, ya que se presentan desde tableros de comunicación análogos (Libros impresos, Fichas de Lotería) hasta dispositivos electrónicos y software especializados.. Los principales sistemas de CAA se basan en un conjunto de iconos o pictogramas combinados con palabras, que permiten su representación gráfica, utilizados por la persona con NCC a partir de la selección de un conjunto de pictogramas que le permitan expresar una idea, necesidad o pensamiento.

La tecnología ha sido de gran ayuda para las personas que utilizan CAA, ya que permite mejorar su competencia comunicativa, acceder a educación, empleo y hacer parte de la sociedad [9]. Con el uso de las aplicaciones móviles se ha dado una mayor adopción de tecnologías CAA, permitiendo el uso de dichos sistemas a un gran número de población, que es diversa tanto en su edad, cultura y limitaciones comunicativas. Los retos actuales que se presentan en los sistemas de CAA son primero mejorar el tiempo de generación de mensajes; segundo, tener en cuenta las limitaciones de las personas con trastornos neurolingüísticos, quienes por lo general no pueden seguir un orden sintáctico, establecer un conjunto deseado y poseer una lógica semántica (exigencia de los sistemas convencionales); y tercero, brindar el soporte para ser utilizados en contextos sociales como redes sociales, servicios de mensajería instantánea, correos electrónicos y conversaciones persona a persona.

En este contexto un grupo de investigación está desarrollando un proyecto cuyo objetivo es la propuesta método computacional válido de Comunicación Aumentativa y Alternativa, enfocado a personas con Necesidades Complejas de Comunicación, para ser aplicado en entornos sociales, usando modelos avanzados de recomendación, personalización, interacción humano computador y estrategias del lenguaje; con el fin de permitir la creación eficaz y eficiente de mensajes coherentes en su estructura semántica y sintáctica. Un elemento clave del proyecto es el desarrollo de la interfaz gráfica de Usuario, debido a que esta va a ser el medio para que los usuarios con NCC, pueda acceder a las funcionalidades propuestas en el método computacional y por lo tanto se presentan los siguientes retos a ser resueltos:

- El desarrollo de la interfaz debe iniciar desde el día uno del desarrollo del proyecto.
- La interfaz debe poder utilizarse en contextos sociales.
- El usuario objetivo cuenta con limitaciones motrices y cognitivas que le impiden usar el ratón o el teclado.
- El usuario objetivo tiene limitaciones para utilizar los canales de comunicación orales y escritos.
- El desarrollo de la interfaz debe ser independiente al desarrollo de las funcionalidades del método computacional de CAA,

debido a que se cuenta con equipos diferentes para ambos fines.

- La interfaz es el elemento más importante del desarrollo debido a que es el canal de comunicación entre el usuario y la aplicación.
- Se cuenta con un estudio teórico de los usuarios pero no se conoce como es su comunicación actualmente.
- Existe un conjunto de 8 categorías semánticas para permitir la comunicación de actividades diarias.
- Se debe incluir contenido multimedia (Audio, Texto, Videos e Imágenes) para la construcción de la frase, que emule los pictogramas utilizados en los sistemas convencionales de CAA.

## 6. CONCLUSIONES Y TRABAJO FUTURO

Este trabajo presenta una metodología de desarrollo de interfaces gráficas de usuario que está basada en los fundamentos y principios del Diseño Centrado en el Usuario, el Pensamiento de Diseño, el Desarrollo Ágil, la Usabilidad y la Experiencia de Usuario. La metodología, tiene como principios que el usuario interviene constantemente en el proceso de construcción de la interfaz, que la unidad de análisis son las actividades desarrolladas por el usuario y que cada fase de la metodología es iterativa e incremental. La metodología está diseñada para ser utilizada en proyectos de desarrollo de software (aplicaciones web, aplicaciones móviles, sitios web, aplicaciones de escritorio, etc...) que pueden estar en cualquier nivel de madurez y en los cuales pueden intervenir ingenieros, diseñadores y desarrolladores, entre otros. La metodología propone nueve fases: *Estructuración, Reconocimiento, Exploración, Modelado, Ideación y Prototipado, Formalización, Implementación, Validación en Contexto y Despliegue*. Cada fase posee un objetivo particular que está articulado con un conjunto de etapas y entregables o resultados esperados.

Adicionalmente, se propone un caso de estudio para la posterior validación de la aplicación de la metodología. El caso de estudio está basado en una aplicación de Comunicación Aumentativa y Alternativa para personas con Necesidades de Comunicación Complejas. Este caso se presenta como un reto para el desarrollo de interfaces debido a la alta dependencia que existe entre las características de los usuarios objetivos y el desarrollo de software para la mejorar la comunicación. En este contexto, la interfaz posee una gran importancia porque a pesar que desde el desarrollo del software se presenten diferentes funcionalidades como personalización predicción, uso de contenidos multimedia y relaciones semánticas, que soportan la actividad de comunicación del usuario y son complejas a nivel de desarrollo de software, si no existe una interfaz que esté diseñada para el contexto de comunicación y necesidades del usuario, las funcionalidades del software no tendrían como ser soportadas para que el usuario pueda acceder a ellas, a pesar de que estas brindan buenas oportunidades para aumentar la tasa de comunicación de la persona.

Como trabajo futuro se propone: (i) realizar la validación de la metodología utilizando el caso de estudio propuesto; (ii) realizar una descripción detallada de las etapas de cada una de las fases y mapear herramientas específicas para su ejecución; y (iii) proponer herramientas para la medición y evaluación de la experiencia de

usuario y usabilidad que estén adaptadas a todas las fases de la metodología.

## AGRADECIMIENTOS

Los autores de este trabajo quieren agradecer al Grupo de Investigación ACCEDER de la Universidad Militar Nueva Granada, en especial a las profesoras Marcela Iregui y Charlems Álvarez. Este trabajo fue financiado por la Vicerrectoría de Investigaciones de la Universidad Militar Nueva Granada, en el marco de proyectos de investigación científica con el proyecto INV ING 2103, vigencia 2016.

## REFERENCIAS

- [1] Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). *Manifiesto for agile software development*.
- [2] Bentley, J. (1999) 14 Steps to a Good GUI. In SAS® Users Group International Conference.
- [3] Brown, T. (2008). Design thinking. *Harvard business review*, 86(6), 84.
- [4] Brown, T. (2009). *Change by design*.
- [5] Brown, T., & Wyatt, J. (2015). Design thinking for social innovation. *Annual Review of Policy Design*, 3(1), 1-10.
- [6] Browne, D. (Ed.). (2016). *Adaptive user interfaces*. Elsevier.
- [7] Dingsøyr, T., Nerur, S., Baliyepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213-1221.
- [8] Garrett, J. J. (2010). *Elements of user experience, the: user-centered design for the web and beyond*. Pearson Education.
- [9] Gould, J. D., & Lewis, C. (1985). Designing for usability: key principles and what designers think. *Communications of the ACM*, 28(3), 300-311.
- [10] Hassenzahl, M., & Tractinsky, N. (2006). User experience-a research agenda. *Behaviour & information technology*, 25(2), 91-97.
- [11] Henry, S. L. (2007). *Just ask: integrating accessibility throughout design*
- [12] International Organization for Standardization (1998). *Norma ISO 9241 - Part 11: Guidance on usability. Ergonomic requirements for office work with visual display terminals*, 22 pages.
- [13] International Organization for Standardization (2010). *Norma ISO 9241 - Part 210: Human-centred design for interactive systems. Ergonomics of human-system interaction*, 32 pages.
- [14] Johnson, C. N. (2002). The benefits fo PDCA. *Quality Progress*, 35(5), 120.
- [15] Karagkasidis, A. (2008, July). *Developing GUI Applications: Architectural Patterns Revisited*. In EuroPLOP.
- [16] Kolhe, P. R., Khetri, G. P., & Deshmukh, N. K. (2012). Study of Standard Assumptions of Graphical User Interface (GUI) Based on Usability, Adaptability and Security Factors. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(12), 1-7.
- [17] Labib, C., Hasanein, E., & Hegazy, O. (2009). Early development of graphical user interface (GUI) in agile methodologies. *Journal of Computational Methods in Sciences and Engineering*, 9(1, 2S2), 239-249.
- [18] Luostarinen, R., Manner, J., Määttä, J., & Järvinen, R. (2010, October). User-centered design of graphical user interfaces. In *MILITARY COMMUNICATIONS CONFERENCE, 2010-MILCOM 2010* (pp. 50-55). IEEE.
- [19] Martin, R. C. (2003). *Agile software development: principles, patterns, and practices*. Prentice Hall PTR.
- [20] McCarthy, J., & Wright, P. (2004). Technology as experience. *interactions*, 11(5), 42-43.
- [21] McMaster, S., & Memon, A. (2008). Call-stack coverage for gui test suite reduction. *IEEE Transactions on Software Engineering*, 34(1), 99-115.
- [22] Morales, L. (2001, March). Structured user interface design methodology. In *CHI'01 extended abstracts on Human factors in computing systems* (pp. 51-52). ACM.
- [23] Morville, P. (2005, July). *Experience design unplugged*. In *ACM SIGGRAPH 2005 Web program* (p. 10). ACM.
- [24] Pea, R. D. (1987). User centered system design: new perspectives on human-computer interaction. *Journal educational computing research*, 3, 129-134.
- [25] Piasecki, M and Piezka, K. (2006) *Conceptual Methodology of Developing the User Interface*. Institute of Applied Informatics, Wroc law University of Technology.
- [26] Redmond-Pyle, D., & Moore, A. (1995). *Graphical user interface design and evaluation (guide): a practical process*. Prentice Hall,
- [27] Ritter, F. E., Baxter, G. D., & Churchill, E. F. (2014). *User-centered systems design: a brief history*. In *Foundations for designing user-centered systems* (pp. 33-54). Springer London.
- [28] Rubin, J., & Chisnell, D. (2008). *Handbook of usability testing: how to plan, design and conduct effective tests*. John Wiley & Sons.
- [29] Shneiderman, B. (2010). *Designing the user interface: strategies for effective human-computer interaction*. Pearson Education India.
- [30] Six, H. W., & Voss, J. (1991). *User interface development: Problems and experiences*. In *New Results and New Trends in Computer Science* (pp. 306-319). Springer Berlin Heidelberg.