

AgilUs: Construcción ágil de la Usabilidad

Alecia Eleonora Acosta

Centro de Ingeniería de Software y Sistemas (ISYS)
Escuela de Computación - Facultad de Ciencias - Universidad Central de Venezuela,
Venezuela
eleonora.acosta@ciens.ucv.ve

Resumen

En la actualidad, el desarrollo de aplicaciones útiles, usables y agradables a los usuarios constituye uno de los grandes retos de la Ingeniería de Software y la Interacción Humano Computador, y conlleva a grandes beneficios tales como el incremento de la productividad, la satisfacción y la reducción de costos de entrenamiento y soporte del software. En este trabajo se propone un método de desarrollo ágil que permite la construcción de la usabilidad desde las primeras etapas del ciclo de vida del software, y se presenta uno de los casos de estudio que ha permitido validar este método, a través de las evaluaciones de usabilidad realizadas durante el ciclo de vida del software. Las técnicas de evaluación de usabilidad propuestas en el método no son costosas ni requieren de infraestructura tecnológica compleja.

Palabras Clave: Método ágil de desarrollo de Software, usabilidad, evaluación de usabilidad.

Abstract. Actually, useful, comfortable and usable software development is one of the great challenges for Software Engineering and Human Computer Interaction. This software produces great benefits such as increase of productivity, satisfaction and reduction of training costs and software support. In this work, we propose an agile software development method for usability construction from first stages of software development. We present one of study cases which it allows us to validate this method, by applying usability evaluation during software life cycle. Usability evaluation techniques used by the method are not expensive nor do they require of complex technological infrastructure.

Keywords: agile software development methods, usability, usability evaluation.

1 Introducción

En los tiempos actuales, es cada vez más importante el desarrollo de las interfaces de usuarios usables, lo cual hace que surjan nuevos tópicos de investigación en el área, que involucran grupos multidisciplinarios de desarrollo, y donde se requiere producir lo más rápido posible debido a que hay mucha competencia en el mercado. En este sentido el diseño de una interfaz de usuario usable se ha vuelto un aspecto

fundamental para las aplicaciones interactivas, así como la entrega rápida de nuevos desarrollos de software. El desarrollo de aplicaciones útiles, usables y agradables a los usuarios conlleva a grandes beneficios tales como el incremento de la productividad, la satisfacción y la reducción de costos de entrenamiento y soporte del software.

Se plantean métodos de desarrollo de software que buscan agilizar la producción de software, tal como el enfoque ágil. Sin embargo, no se incluye en forma explícita la evaluación de la usabilidad de los productos de software a ser desarrollados. Algunos autores, como Nielsen [1] y compañías como ThoughtWorks [2], han descrito experiencias en cuanto a la incorporación de la usabilidad en el proceso de desarrollo de software, dando algunas recomendaciones para tal fin; pero sería conveniente precisar algunas técnicas a utilizar en cada etapa del desarrollo.

Son muchos los métodos propuestos para el desarrollo de software y son muchas las herramientas y técnicas con que cuentan los ingenieros de software para realizar su trabajo. Quizás no es el mismo caso de los especialistas en Interacción Humano-Computador, entendiendo que esta disciplina no tiene la misma madurez que la anterior; sin embargo, se tienen técnicas, guías y recomendaciones, entre otros mecanismos que permiten realizar las evaluaciones a fin de desarrollar software usable. El objetivo de esta investigación es proponer algún mecanismo de integración entre estas dos disciplinas que permita la producción de sistemas usable. Se propone un método que provea un conjunto de “buenas prácticas” para el desarrollo de software desde una perspectiva ágil, con la finalidad de obtener un producto usable. Se plantea el desarrollo de un sistema partiendo de un prototipo de la interfaz de usuario, y que incorpora la aplicación de diversas técnicas de evaluación de usabilidad desde el inicio del ciclo de vida de la aplicación.

Considerando que la esencia de los enfoques ágiles es el intercambio continuo y constante con los usuarios, y que ese también es el principio del diseño centrado en el usuario, el método propuesto hace énfasis la obtención de ese *feedback* desde etapas tempranas del desarrollo, con la aplicación de tormentas de ideas, encuestas y prototipos en papel, entre otras, a lo largo de la evolución del software hasta la obtener el producto final, incluida la entrega del producto. Así los resultados de estas evaluaciones permitirán asegurar la obtención de un producto usable.

La sección 2 expone los conceptos fundamentales de esta investigación: usabilidad, evaluación y métodos ágiles de desarrollo de software. La sección 3 presenta la propuesta concreta del método AgilUs y la sección 4 muestra los resultados obtenidos en un caso de estudio de aplicación de este método.

2 Usabilidad y Técnicas de Evaluación

Intuitivamente, la usabilidad puede definirse como la facilidad de uso de cualquier sistema computacional con el cual interactúe el usuario. Formalmente, existen diversas definiciones para esta cualidad del software, entre las cuales se pueden citar:

- Es una cualidad del software que se refiere a su capacidad de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso (ISO/IEC 9126)

- Usabilidad es una cualidad del software que determina la efectividad, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico (ISO/IEC 9241-11)
- Según Nielsen [3], la usabilidad es un atributo de calidad que indica qué tan fácil de usar es una interfaz de usuario. Un sistema es usable si es funcionalmente correcto (efectividad), eficiente de usar (eficiencia), fácil de aprender para usuarios novatos, fácil de recordar para usuarios ocasionales, tolerante a errores y subjetivamente agradable (satisfacción)
- “Effective (efectividad), Efficiency (eficiencia), Engaging (ser atractivo), Error-Tolerant (tolerante a errores) y Easy-to-Learn” (fácil de aprender) [4]

Como se observa, los autores citados anteriormente coinciden en atribuirle cualidades de eficiencia, comprensión, satisfacción, comodidad en el uso de los sistemas interactivos. Además, es importante destacar que un producto no es intrínsecamente usable. La facilidad de uso o usabilidad dependerá de un contexto y de usuarios particulares. La usabilidad no puede ser valorada en un producto aislado. Se trata de una cualidad centrada en el concepto de calidad en el uso. En definitiva, se refiere a cómo el usuario realiza tareas específicas en escenarios específicos con efectividad y satisfacción.

Según la ISO 9126, la usabilidad forma parte de los atributos que impactan en la calidad interna y externa de los productos de software, al igual que la funcionalidad, fiabilidad, eficiencia, mantenibilidad y portabilidad.

Los sistemas usables requieren menos entrenamiento, menos soporte para el usuario y menos mantenimiento; además se aprecia mejora en la productividad; se reduce el esfuerzo y permite a los usuarios manejar una variedad más amplia de tareas. Adicionalmente mejora la calidad del producto ya que el diseño centrado en el usuario resulta en productos de mayor calidad de uso, más competitivos en el mercado. Los sistemas difíciles de usar disminuyen la salud, bienestar, productividad y motivación de los usuarios, y pueden incrementar el absentismo.

Alcanzar la usabilidad en el desarrollo de software implica aplicar un Diseño Centrado en el Usuario, el cual es un conjunto de métodos y técnicas que tiene como paradigma central la inclusión del usuario como centro de desarrollo. A fin de realizar un diseño centrado en el usuario se debe tomar en cuenta las características del usuario, las actividades que realiza y el escenario donde se desempeña su actividad. Todos estos factores permitirán conocer cuáles son los requisitos y metas que se deben satisfacer en el sistema y cómo es la forma más apropiada de ofrecérselos a los usuarios para facilitar el uso del mismo.

En general, la evaluación es un aspecto fundamental a considerar en el diseño centrado en el usuario, debido a que permite saber si un sistema cumple las expectativas de los usuarios y se adapta a su contexto social, físico y organizacional. En particular, la evaluación de usabilidad determina el grado de usabilidad de un sistema interactivo. Su aplicación constante y continua en el desarrollo de software permitirá construir la usabilidad del producto final durante su ciclo de vida. Así, a fin de garantizar la usabilidad de un producto de software, es necesario aplicar técnicas de evaluación de usabilidad desde etapas iniciales del desarrollo.

Existen diversas técnicas de evaluación de usabilidad, que pueden ser aplicadas por distintos actores (usuarios, especialistas, entre otros) y en diferentes momentos del

desarrollo del producto, inclusive una vez que ya se ha puesto en producción. La Fundación Sidar-Acceso Universal (SIDAR) [5], en su recopilación de “métodos de usabilidad”, traducido por Alejandro Floría, los agrupa en cuatro (4) categorías, a saber: indagación, inspección, prueba y prototipaje. A continuación se describen estas categorías:

Indagación. Trata de llegar al conocimiento de algo a través de un conjunto de preguntas e investigaciones; especialmente si se refiere a un asunto oculto o secreto. Las técnicas que se agrupan en esta categoría deben proporcionar información acerca de la usabilidad de un producto que aún no se ha empezado a construir. En estas técnicas se debe hacer un gran trabajo de hablar con los usuarios y observarlos usando el sistema en tiempo real, o analizando respuestas que ellos dan a preguntas verbales y/o escritas. Las evaluaciones enmarcadas en esta categoría permiten identificar los requerimientos del usuario. Son indispensables en una etapa temprana de un proceso de desarrollo que culmina en la satisfacción de una necesidad del usuario, quien con eficiencia y efectividad podrá realizar las funciones le ofrece el producto. Técnicas de este tipo relevantes a este trabajo de investigación son: tormentas de ideas, entrevistas, encuestas, cuestionarios y evaluación de sistemas existentes.

Inspección. Consiste de una evaluación que puede implicar observación, medición y comparación con respecto a ciertas características. Aplicado a la usabilidad agrupa un conjunto de técnicas para evaluar la usabilidad en los que especialistas, conocidos como evaluadores, verifican el grado de usabilidad de un sistema basándose en la inspección o examen de la interfaz de usuario. Dentro de las técnicas de este tipo relevantes a este trabajo de investigación se encuentran: evaluación heurística, listas de comprobación y guías de estilo.

Pruebas. Las pruebas de usabilidad son una forma de medir qué tan bien puede una persona usar un objeto hecho por el hombre, en particular un software. En las técnicas de evaluación de usabilidad que se agrupan en esta categoría, un conjunto de usuarios representativos trabajan con el sistema –o un prototipo– y los evaluadores toman nota de la interacción, particularmente de los errores y dificultades con las que se encuentren los usuarios. Dentro de las técnicas de este tipo relevantes en esta investigación se pueden citar: protocolo del pensamiento manifestado, protocolo de preguntas y las pruebas de aceptación.

Prototipaje. La aplicación de la técnica de prototipaje consiste en la construcción de un modelo del sistema que se desarrolla. Esta técnica es fundamental en el desarrollo e implementación de los métodos descritos para la inspección y pruebas de un producto, debido a que, habitualmente, no se aplican al producto final, sino un prototipo del mismo con unas determinadas características. Dentro de las técnicas de este tipo relevantes a esta investigación se encuentran: prototipaje en papel, prototipaje rápido y patrones de Interacción.

3 El Método AgilUs

En la Ingeniería de Software existen diversos tipos de metodologías de desarrollo de software, entre las que se destacan las tradicionales y las ágiles. Las metodologías

tradicionales son aquellas con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requerimientos y modelado. Las ágiles se caracterizan por ser iterativas e incrementales, y fácilmente adaptables a los cambios, involucrando al usuario en el desarrollo [6]. En la actualidad, estas metodologías ágiles tienen gran aceptación a nivel mundial y están basadas en un manifiesto publicado en 2001, cuyo objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales.

Las metodologías basadas en el "Manifiesto de desarrollo ágil de software" [7] les dan más importancia a los individuos, funcionalidad, colaboración con el cliente y adaptación al cambio, que a los procesos y herramientas, exceso de documentación, contratos y el seguimiento estricto de un plan. Las metodologías ágiles buscan ser más adaptables a los continuos cambios que se presentan durante el desarrollo de un sistema y para esto emplean un enfoque iterativo e incremental, con interacciones cortas, planificación adaptativa y entrega evolutiva. Se busca lograr que los cambios sean menos costosos, permitiendo que sean incorporados más fácilmente [8].

El Método AgilUs es un método de desarrollo ágil, resultado de una de las líneas de investigación desarrolladas en el Centro de Ingeniería de Software y Sistemas (ISYS) de la Escuela de Computación, Universidad Central de Venezuela. Se basa en el concepto de usabilidad, en la necesidad de desarrollar software usables. Se fundamenta en el análisis centrado en el usuario y en la participación de especialistas, con el objetivo de evolucionar el software, a fin de que éste alcance el mayor grado de usabilidad una vez culminado su desarrollo. AgilUs es un método de desarrollo iterativo e incremental que pone el mayor peso del desarrollo en la consecución de la usabilidad. Se centra en que la construcción y desarrollo de las interfaces de usuario no debe ser una adición estética que se da al final del desarrollo del sistema sino, muy por el contrario, el desarrollo de interfaces de usuario debe guiar las decisiones en Ingeniería de Software. En AgilUs son los usuarios, y no el cliente ni los programadores quienes guían el desarrollo del proyecto. Algunos trabajos relacionados con esta investigación pero más orientados a métodos de desarrollo tradicionales son la Tesis Doctoral de Granollers [9] y la propuesta de Ferré [10].

El Método AgilUs busca proporcionar un conjunto de actividades organizadas para construir la usabilidad en el diseño de interfaces de usuario durante el desarrollo de un producto de software. El proceso de desarrollo de software engloba las actividades de requisitos, análisis, prototipaje y entrega; así como las evaluaciones de usabilidad correspondientes a cada etapa del proceso. Se realizan en ciclos iterativos hasta alcanzar el producto final. En cada etapa del proceso de desarrollo de software, se incluyen actividades propias para la construcción de la usabilidad.

3.1 Principios

AgilUs centra el desarrollo de software en los siguientes principios:

- Integra la Interacción Humano Computador (IHC) y la Ingeniería de Software (IS). IS y IHC son complementarias, no son disciplinas excluyentes. Un diseño centrado en el usuario impacta positivamente en la calidad del software (ISO 9126-1).

- La usabilidad debe considerarse desde el principio del desarrollo. Si la IS y la IHC son complementarias y no excluyentes, y si la usabilidad aumenta la calidad del software, entonces es conveniente incluir la usabilidad desde el principio en el desarrollo como uno de los requisitos para impactar positivamente la calidad del producto final.
- La usabilidad determina la utilidad. Un software se considera útil en la medida que pueda ser usado a fin de producir resultados, en forma eficiente, intuitiva y satisfactoria para los usuarios.
- El usuario determina la usabilidad. La usabilidad no es una propiedad abstracta. Un software sólo será considerado usable en un contexto específico y por un tipo de usuario específico. El objetivo es lograr que todos los usuarios del software encuentren usables las tareas que pueden realizar.

3.2 Buenas prácticas

Algunas de las “buenas prácticas” del desarrollo de software que se aplican en AgilUs, las cuales están enfocadas en satisfacer las demandas del usuario y el desarrollo iterativo e incremental, procurando la usabilidad en cada paso del proceso de desarrollo, son las siguientes:

- Diseño centrado en el usuario (DCU). El DCU es un enfoque de diseño y desarrollo que se centra en los deseos, limitaciones y necesidades de los usuarios finales de un software. En las técnicas de DCU es relevante que los desarrolladores realicen pruebas constantes para verificar el curso que lleva el desarrollo del sistema y su interfaz de usuario. De este modo, el usuario guía indirecta pero influyentemente el proceso de desarrollo del sistema. La diferencia fundamental entre éste y otros enfoques de diseño es que en el DCU se procura construir el sistema para adaptarse, a través de su interfaz, a cómo el usuario desea trabajar, en lugar de forzar al usuario a cambiar su modo de trabajar para adaptarse a lo que los desarrolladores consideraron apropiado [11].
- Diseño basado en prototipos. El desarrollo de software en AgilUs está guiado por la construcción de prototipos de alta fidelidad y la evaluación de los mismos por los usuarios y por especialistas en usabilidad. Se entiende entonces que, tras una inspección inicial, los desarrolladores producen un primer prototipo, los especialistas y el usuario lo evalúan, los analistas preguntan directamente al usuario sus opiniones sobre el desarrollo, y con esa retroalimentación, los desarrolladores se disponen a producir un siguiente prototipo. Este ciclo continúa hasta que se tiene un producto listo para la entrega, cuando las evaluaciones de usabilidad, requerimientos y calidad del software están completamente satisfechas.
- Desarrollo ágil, iterativo e incremental. Una de las máximas del desarrollo iterativo e incremental y del Manifiesto Ágil es la simplicidad. Se recomienda entonces desarrollar el sistema más simple que satisfaga las necesidades actuales de los usuarios, preparándose para cambios futuros. El desarrollo por incrementos permite proveer resultados sin necesidad de esclarecer todo los requisitos de una vez al inicio del desarrollo. La iteratividad permite regresar a etapas anteriores una vez recibida la retroalimentación producto de las evaluaciones realizadas.
- Usabilidad como atributo de la calidad. Como se indica en el estándar ISO/IEC 9126-1, la usabilidad es considerada un atributo de la calidad interna y externa del

software, y AgilUs hace énfasis en la producción de software usable, siguiendo la recomendación de este estándar internacional.

- Interacción continua con el usuario, propiciando un intercambio cara a cara. Naturalmente, para AgilUs la presencia constante y participativa del usuario es fundamental. El equipo de desarrollo sólo puede tomar decisiones tras realizar evaluaciones de usabilidad, y la usabilidad del sistema sólo puede ser determinada por el usuario.

3.3 Ciclo de vida

El ciclo de vida de AgilUs hace énfasis en la importancia del usuario y sus evaluaciones. Está basado en el desarrollo iterativo e incremental de prototipos de alta fidelidad hasta que se convierten en el producto final para entrega. Este producto *final* puede ser posteriormente modificado a través de un mantenimiento correctivo y/o evolutivo, que no está contemplado como parte del método.

En cada etapa del desarrollo se incluyen actividades para la construcción de la usabilidad. Se busca proporcionar una manera de proceder organizadamente para construir la usabilidad durante el desarrollo de un producto. El ciclo de vida engloba la definición de requisitos, análisis, prototipaje y entrega.



Fig. 1. El método AgilUs: etapas, actividades y artefactos

La Fig. 1 muestra un diagrama de la relación entre cada una de las etapas del ciclo de vida de AgilUs, con las actividades que se realizan y artefactos que se generan en cada etapa. A continuación se describen las etapas de este método:

- **Requisitos:** Se realiza el análisis global del problema a solucionar, se estudian productos similares existentes, se genera un perfil de usuario, y se define la lista de requerimientos a desarrollar. Esta etapa es importante en el desarrollo del software,

ya que un mal análisis de requisitos traería como consecuencia un software que no cumple con las necesidades del usuario.

- Análisis: Se lleva a cabo el análisis de la solución a desarrollar, se emplean diagramas de casos de uso y modelo de objetos del dominio, siguiendo la notación UML, para definir las funcionalidades que tendrá el producto a desarrollar.
- Prototipaje: Se implementa un prototipo rápido de la interfaz de usuario a partir de los patrones de interacción, el cual va evolucionando hasta convertirse en el producto final, se genera la guía de estilo, y se realizan evaluaciones de usabilidad apropiadas a esta etapa: las evaluaciones heurísticas y las listas de comprobación.
- Entrega: Se aplican las pruebas al sistema para certificar que la aplicación desarrollada sea un software usable y sin errores, finalmente se pone en producción la aplicación.

3.4 Algunos errores comunes

Varios de los errores que pueden cometerse al implementar AgilUs surgen a causa de una disminución deliberada o accidental de la participación del usuario en el proceso de desarrollo, o de una mala comprensión del rol que debe ocupar a la hora de tomar decisiones con respecto al diseño del sistema y sus interfaces, con base en las evaluaciones de usabilidad:

- El equipo de desarrollo, sin incluir al usuario, puede determinar la usabilidad del sistema. Los desarrolladores pueden llegar a sentir, gracias a la experiencia acumulada o a la poca complejidad del sistema, que están completamente familiarizados con los deseos del usuario y que pueden predecir o aproximar con exactitud sus reacciones, deseos, capacidades y carencias. Este es uno de los errores más frecuentes en el desarrollo de software y, si bien es muy importante que el equipo de desarrollo entienda lo mejor posible al usuario, esto no debe confundirse con la posibilidad de reemplazarlo. Sólo el usuario puede decidir si el sistema lo satisface; esto es, si el sistema es usable.
- El cliente y el usuario no es necesariamente el mismo. El cliente puede sentir que, ya que es el propietario del sistema, es su deber tomar las decisiones, por ejemplo en términos de usabilidad, dejando a los desarrolladores únicamente los aspectos técnicos. En AgilUs se considera que los usuarios deberían decidir, indirectamente a través de sus evaluaciones y comentarios, qué se debe hacer y por qué.

4 Aplicación de AgilUs a un caso de estudio: GEPECO

El Generador de Periódicos Comunes (GEPECO) es un producto de software cuyo objetivo principal es permitir al usuario, sin conocimientos especializados en diagramación, diseño de periódicos, ni informática, editar una publicación para así obtener un periódico comunal adaptado a las necesidades de su comunidad. GEPECO fue desarrollado como parte de un Trabajo Especial de Grado [12] y constituye uno de los casos de estudio que han permitido validar el método AgilUs. La aplicación de este método de desarrollo implicó la realización de cuatro etapas, a saber: requisitos, análisis, prototipaje y entrega, con sus respectivas evaluaciones de usabilidad; lo cual se resume a continuación.

4.1 Etapa 1: Requisitos

En el análisis de los requisitos se hizo un estudio de aplicaciones afines existentes en el mercado. Además se realizó una tormenta de ideas con el equipo de desarrollo y se aplicó una encuesta a los usuarios potenciales, en la cual se indagó sobre sus opiniones y preferencias en cuanto a características deseables en la aplicación. Los resultados obtenidos en esta etapa se resumen a continuación.

Tormenta de ideas. Esta técnica fue utilizada en el inicio del proceso de desarrollo, se realizó una serie de reuniones donde se discutieron diferentes ideas y opiniones referentes a la aplicación. Esto con el fin de llegar a un consenso entre todos los integrantes del equipo de desarrollo.

Análisis de aplicaciones afines existentes. Se realizó un análisis de sistemas de diagramación existentes con la finalidad de puntualizar sus ventajas y aplicarlas al programa a desarrollar; así como minimizar las desventajas o errores que estos pudieran tener. Las aplicaciones estudiadas fueron: Adobe PageMaker 7.0, Scribus, Quakxpress y Publisher 2007. De este análisis se tomaron ideas para la aplicación a desarrollar, tales como: minimizar la carga de memoria del usuario reutilizando metáforas conocidas, ofrecer plantillas diseñadas, ofrecer un asistente de guía al usuario para crear una publicación, entre otras.

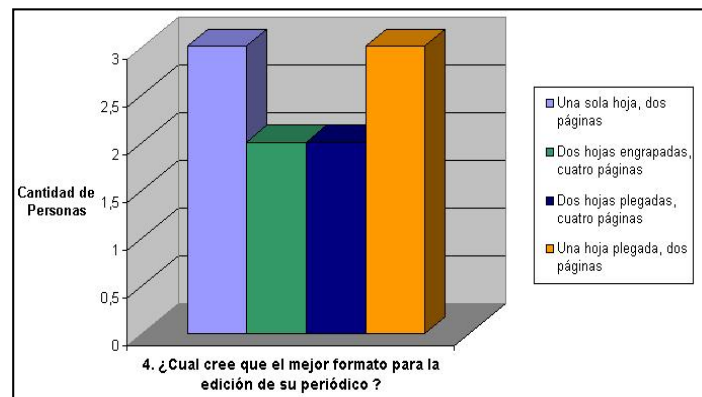


Fig. 2. Gráfico de barras correspondiente a la Pregunta 4

Análisis de encuestas. A fin de continuar determinando los requisitos funcionales y no funcionales, entre estos últimos se incluyen aspectos de la interfaz de usuario, se aplicó una encuesta, la cual permitió indagar: preferencias en el aspecto de la interfaz de usuario, algunas funcionalidades, experiencias previas, entre otras. De los resultados obtenidos se destacan aquellos relacionados con la forma de presentar la información en la interfaz de usuario, con características del periódico a ser generado y una pregunta que reflejó parte del perfil de los usuarios de la aplicación. A continuación se presenta algunos de los resultados obtenidos con la aplicación de la encuesta. El análisis completo de los resultados se puede consultar en [12]. En la *Pregunta 4*, los resultados se dividieron entre las cuatro (4) opciones, 30% para la primera y última opción y 20% para la segunda y tercera opción, por lo que se incluyeron los cuatro formatos en el sistema de diagramación, estos resultados se aprecian en la Fig. 2. En la *Pregunta 10*, la mayoría de los comentarios estuvieron

ligados a que el sistema debía ser de fácil manejo y que le brindara la oportunidad de editar el periódico a su gusto, adicionalmente se observó una gran cantidad de comentarios apoyando la idea del desarrollo de este sistema.

Perfil de usuario. Los usuarios potenciales del sistema son los miembros de las comunidades; es decir, personas sin conocimientos avanzados sobre periodismo ni diagramación de periódicos, por lo que no se considerarán usuarios expertos en el dominio del sistema, ni en las tecnologías necesarias para la generación automática de periódicos comunales impresos. Los únicos conocimientos que se requieren por parte de los usuarios es el uso de herramientas ofimáticas.

Lista de requerimientos funcionales y no funcionales. Luego de realizar el análisis de requerimientos a través de la tormenta de ideas, revisión de herramientas existentes y la aplicación de la encuesta, se obtuvo una lista de requerimientos funcionales y no funcionales para el desarrollo del sistema, de los cuales se destacan los siguientes. Entre los requerimientos funcionales: Crear un periódico a todo color, y da la posibilidad de permitir imprimir en blanco y negro para reducir costos; permitir la previsualización del periódico antes de imprimirlo; proveer plantillas de diseño; proveer ayuda en cada paso del asistente (wizard); mostrar mensajes de ayuda estilo *tooltips* en los botones y menús de la aplicación y proveer una funcionalidad que le permita al usuario guardar un periódico, para editarlo posteriormente. Como requerimientos no funcionales: que el sistema fuese sencillo e intuitivo, con un alto grado de usabilidad; fácilmente extensible ya que se trata de un producto de código abierto; mantenible y que pueda ejecutarse en un computador con una configuración básica.

4.2 Etapa 2: Análisis

En esta etapa se realizó el análisis de la solución: se definieron los casos de uso, el modelo de objeto del dominio (expresados en UML), se realizaron prototipos en papel y los patrones de interacción que describen la interfaz de usuario.

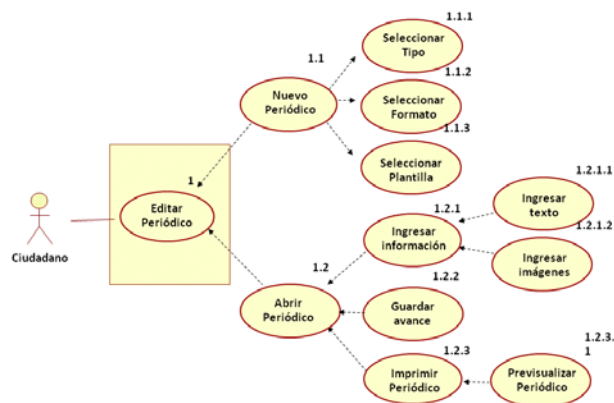


Fig. 3: Modelo de Casos de Uso de GEPECO

Modelo Casos de Uso. Un modelo de caso de uso describe la secuencia de las interacciones que se desarrollarán entre los actores y el sistema, en respuesta a un evento que inicia un actor. En esta aplicación se definió un actor principal denominado *Ciudadano*, el cual utilizará la aplicación para la creación de un

periódico comunitario. Este actor tendrá acceso a todas las funcionalidades de la aplicación (Ver Fig.3). El caso de uso *Editar Periódico* permite a un ciudadano iniciar la creación de un periódico desde una plantilla o abrir un periódico existente. El ciudadano tiene la opción de elegir realizar un periódico para imprimir o un periódico para colocarlo en la Web, escoger el formato (una o dos hojas), seleccionar la plantilla que va a utilizar e ingresar información al periódico (texto e imágenes).

Modelo Objetos del Dominio. Un modelo objeto del dominio describe los objetos que del dominio de la aplicación y las relaciones estáticas entre estos. En la Fig. 4 se muestra el modelo de objeto del dominio de GEPECO. Se especifican los siguientes objetos: Periódico Comunal (Publicación impresa generada), Plantillas (Diseño y diagramación a seleccionar para un periódico), Páginas (Conjunto de artículos), Artículos (Agrupación de imágenes y texto), Imágenes, Textos. Entre estos objetos se establecen las siguientes relaciones: un periódico comunal se basa en una plantilla, una plantilla posee de una (1) a cuatro (4) páginas, una página tiene asociado uno (1) o más artículo, un artículo tiene una (1) o más imágenes y tiene uno (1) texto.

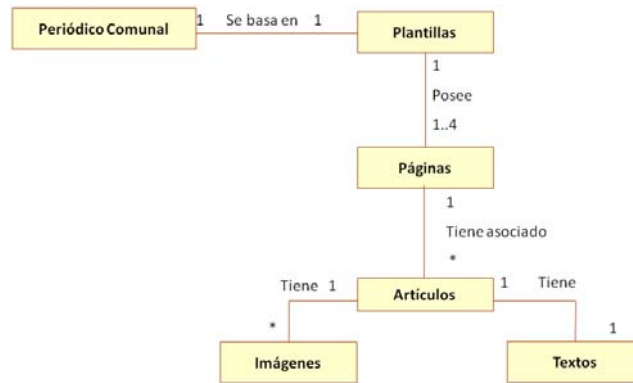


Fig. 4: Modelo Objetos del Dominio de GEPECO

Prototipo en papel. Durante el análisis de la aplicación se elaboró de un prototipo en papel de la primera pantalla que se le ofrece al usuario.


Nombre	Generador de Periódicos Comunales
Problema	El usuario desea generar un periódico impreso para su comunidad, de forma sencilla y rápida sin tener que tener conocimientos ni en diseño de publicaciones ni en el área informática.
Solución	Desarrollar una aplicación que permita que algún miembro de una comunidad pueda generar un periódico comunitario 
Contexto	Usuarios con acceso al Software Generador de Periódicos Comunales
Usabilidad	Fácil de aprender, Tolerante a errores y Satisfacción del usuario.
Fuerzas	El usuario no necesita tener conocimientos en diseño gráfico ni diagramación de publicaciones; ni necesita tener conocimientos informático avanzados pero sí debe tener conocimientos en el manejo de aplicaciones de ofimáticas.
Patrones Relacionados	Crear nuevo periódico, abrir periódico, guardar avance, ingresar información, vista preliminar, imprimir periódico, seleccionar tipo, seleccionar formato, seleccionar plantilla, editar texto, editar imagen.

Fig. 5: Patrón de sistema de GEPECO

Patrones de interacción. Un patrón de interacción describe una solución exitosa a un problema recurrente concerniente a la interfaz de usuario, en un contexto dado [13]. El lenguaje de patrones que describe la interfaz de usuario que ofrece GEPECO se puede consultar en [12]. A manera ilustrativa, la Fig. 5 muestra el patrón de sistema de la aplicación.

4.3 Etapa 3: Prototipaje

En esta etapa se ejecutó la implementación del sistema a partir de los patrones de interacción, se generó la guía de estilo, y se realizó la evaluación heurística de GEPECO.

Guía de estilo. Se define como guía de estilo al documento que recoge normativas y patrones básicos relacionados con el aspecto de un interfaz para su aplicación en el desarrollo de pantallas dentro de un entorno concreto. La guía de estilo de GEPECO describe los colores primarios, el aspecto de los botones, las operaciones por el teclado, entre otras.

Tabla 1: Resultados de la evaluación heurística de GEPECO

Problema	Heurística	Valoración	Solución
La metáfora de abrir tiene asociado un color muy parecido a una opción deshabilitada.	H3 y H4	1	Colocarle colores apropiados a la metáfora.
No previene errores al cerrar el programa, tal como guardar documento o guardar un periódico antes de cambiar a otro.	H9 y H10	4	Incluir mensajes de alerta para el usuario en casos donde sea necesario.
No muestra feedback al usuario cuando se está realizando alguna acción	H5	3	Implementar mensajes que le indiquen al usuario que su acción está en proceso.
No se puede cambiar la posición del texto, como centrar, justificar, etc.	H1 y H10	2	Agregar las opciones correspondientes.
En los tips de ayuda cuando se crea un periódico se pueden modificar	H4	0	Bloquear los campos de los tips de ayuda
La opción salir no funciona.	H6 y H10	2	Deben realizar la opción salir.

Evaluación Heurística. La evaluación heurística es un método de inspección cuyo objetivo es encontrar problemas de usabilidad en el diseño de una interfaz de usuario, tal que puedan ser atendidos como parte de un proceso de diseño iterativo. Se consideraron las diez (10) heurísticas de Nielsen propuestas en [14]. La escala usada para la valoración de los problemas va de 0 (menos importante) a 4 (catastrófico). La Evaluación Heurística fue realizada por los alumnos de la materia Interacción Humano Computador, dictada en el 5to semestre de la Licenciatura en Computación de la Universidad Central de Venezuela. A cada uno de los participantes se le entregó una planilla, y procedió a realizar el recorrido de la aplicación. En la Tabla 1 se exponen los problemas más resaltantes encontrados durante la evaluación y que fueron corregidos por el equipo de desarrollo de GEPECO.

4.4 Etapa 4: Entrega

Se aplicaron las pruebas de aceptación al sistema para certificar que la aplicación desarrollada cumplía con los requisitos de usabilidad exigidos por los usuarios. Finalmente se entrega a producción la aplicación. El ambiente de producción consta

de una máquina tipo PC o MAC, en la cual se puede estar ejecutando cualquier sistema operativo que admita correr la máquina virtual de Java.

Pruebas de aceptación. Las pruebas de aceptación permiten conocer la opinión general de los usuarios. Se usan para determinar la satisfacción subjetiva del usuario. Para los fines de la aplicación desarrollada, se aplicaron cuestionarios a fin de conocer el nivel de aceptación de los usuarios. Fue entregado a una muestra de 20 personas. A manera ilustrativa se presenta el resultado de la *Pregunta 6*. Se refleja en la Fig. 6 que para el 80% de los usuarios el periódico cumplió con sus expectativas, mientras que el 20% opina que sólo cumplió más o menos. Partiendo de los resultados obtenidos, se puede hacer un promedio de aceptación con base en las preguntas que califican de manera positiva y las que califican de manera negativa la aplicación. Al hacer este cálculo aproximado es posible evidenciar que un porcentaje mayor a la mitad de los usuarios tuvo opiniones positivas acerca de aspectos de usabilidad de GEPECO. Es importante hacer notar que las preguntas del cuestionario buscaban indagar opiniones acerca de aspectos de usabilidad. Es así como al obtener un mayor número de respuestas favorables, se puede decir que esta aplicación es usable.

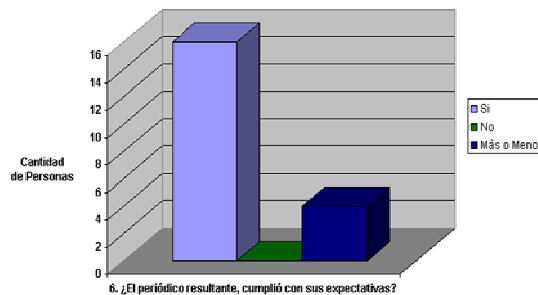


Fig. 6: Gráfico correspondiente a la pregunta 6

Conclusiones

AgilUs constituye una propuesta metodológica en concordancia con las exigencias en el desarrollo de software, incorporando la construcción de la usabilidad en el ciclo de vida. Se inscribe en la categoría de métodos ágiles debido a que reduce la cantidad de actividades y artefactos que se generan, propicia la participación de usuario y su carácter iterativo e incremental permite adaptar el desarrollo a los cambios. Incluye la usabilidad a fin de aplicar un enfoque de diseño centrado en el usuario y como un mecanismo para asegurar la calidad del software. Está orientado al desarrollo de sistemas con mucha interactividad con el usuario (*front-end*) y menos lógica de la aplicación (*back-end*). Como proceso ágil, se trata de determinar los requerimientos y perfiles de los usuarios para presentar un prototipo al usuario que irá evolucionando hasta convertirse en el producto final, sin realizar un diseño estructural.

Aun cuando se pudiera pensar que la incorporación de evaluaciones de usabilidad resta agilidad al proceso de desarrollo, se busca fomentar, a través de estas evaluaciones, la participación del usuario y la interacción cara a cara entre éste y el equipo de desarrollo, logrando un equilibrio entre agilidad y producción de software

usable. Siguiendo en la línea de la agilidad, las evaluaciones de usabilidad propuestas son rápidas, económicas y no requieren de plataformas tecnológicas complejas.

El método propuesto se ha validado a través de diferentes desarrollos; GEPECO constituye uno de estos casos de estudio, permitiendo detectar errores de usabilidad rápidamente y corregirlos en cada una de las etapas de desarrollo. La guía de estilo facilitó un diseño único de las pantallas. La evaluación heurística permitió mejorar el uso de los colores, redacción de mensajes, entre otros. Así, se consideraron hasta los pequeños detalles en el diseño de la interfaz de usuario y se pudo conocer las opiniones del usuario; logrando desarrollar un producto usable.

Actualmente se trabaja en la incorporación de aspectos de accesibilidad y la experiencia del usuario en AgilUs, a fin de producir software usable, accesible y que garantice una excelente experiencia al usuario.

Agradecimientos. Al Fonacit, adscrito al Ministerio del Poder Popular para la Ciencia y Tecnología, por financiamiento al proyecto “Acercamiento de la Tecnología Informática al Ciudadano” Nro 2005000166. A Lic. Víctor Sojo e Ing. Kiberley Santos por su contribución en las discusiones acerca de AgilUs.

Referencias

1. Nielsen, J.: Designing Web usability. New Riders. USA, 2000.
2. ThoughtWorks, www.thoughtworks.com
3. Nielsen, J., Loranger, H.: Usability Prioridad en el diseño Web. Anaya Multimedia. España, 2006.
4. Quesenbery, W.: What Does Usability Mean: Looking Beyond ‘Ease of Use’. Proceedings of the 48th Annual Conference, Society for Technical Communication. (2001)
5. La Fundación Sidar - Acceso Universal , www.sidar.org
6. Fowler, M.: The New Methodology. Última Actualización: 13 diciembre 2005
<http://martinfowler.com/articles/newMethodology.html>
7. Beck, K. et all.: Manifiesto por el Desarrollo Ágil de Software (2001)
<http://www.agilemanifesto.org/iso/es/>
8. Gabardini, J., Campos, L.: Balanceo de Metodologías Orientadas al Plan y Ágiles. Herramientas para la Selección y Adaptación. PMI Global Congress Proceedings. Argentina (2004)
9. Granollers, T.: MPIu+a. Una Metodología que integra la Ingeniería del Software, la Interacción Persona-Ordenador y la Accesibilidad en el contexto de Equipos de Desarrollo Multidisciplinares. Tesis Doctoral. Universidad de Lérida, España (2004).
<http://www.tesisenxarxa.net/TDX-0218107-133615/index.html#documents>
10. Ferre, X.: Integration of Usability Techniques into the Software Development Process. International Conference on Software Engineering. Oregon (2003)
11. Usability Professionals' Association, www.upassoc.org
12. Izarra, M., Guayaquil, A.: Un Generador de Periódicos Comunes: Llevando la Tecnología de la Información a la comunidad. Trabajo Especial de Grado. UCV. Venezuela 2006
13. Acosta, E., Zambrano, N.: Patterns and Objects for User Interface Construction. *Journal of Object Technology*. Vol. 3, No. 3, pp 75-90. ETH Zurich (2004)
http://www.jot.fm/issues/issue_2004_03/article1
14. Nielsen, J. Ten Usability Heuristics,
http://www.useit.com/papers/heuristic/heuristic_list.html