

DOI: 10.20983/culcyt.2020.2.4.1

Diseño de una solución SaaS para el seguimiento de mercancía usando una metodología ágil

Design of a SaaS solution for tracking merchandise using an agile methodology

Karla Selene Ramírez-Márquez¹

¹ Universidad Autónoma de Ciudad Juárez

NOTA TÉCNICA

RESUMEN

El objetivo de esta nota técnica es proveer una solución de *software como servicio* que permita la gestión de los envíos mediante una herramienta de monitoreo SaaS para un óptimo control de los vehículos en ruta de las empresas distribuidoras. Para lograrlo, se escogió una metodología ágil con el fin de acelerar el proceso de desarrollo del software y, con base en ello, se seleccionaron artefactos de modelado conceptual para realizarlos en la etapa de análisis y diseño de la solución. La modelación conceptual del dominio es un proceso elemental para la ingeniería de software que permite reconocer el impacto de los múltiples elementos que interfieren. Por tanto, el presente texto se centró en el orden y descripción de los artefactos que se crearon con el fin de recopilar información, realizar un modelo y obtener requerimientos de los usuarios.

PALABRAS CLAVE: Software como servicio; metodología ágil; modelado conceptual.

ABSTRACT

The objective of this technical note is to provide a *software as a service* solution that allows the management of shipments through a SaaS monitoring tool for optimal control of distribution companies' road vehicles. To achieve this, an agile methodology was chosen in order to accelerate the software development process and, based on this, conceptual modeling artifacts were selected to be carried out in the analysis and design stage of the solution. The conceptual modeling of the domain is an elementary process for software engineering that allows to recognize the impact of the multiple elements that interfere. Therefore, this text focused on the order and description of the artifacts that were created in order to collect information, make a model and obtain user requirements.

KEYWORDS: Software as a service; agile methodology; conceptual modeling.

Correspondencia:

DESTINATARIO: Karla Selene Ramírez Márquez
INSTITUCIÓN: Universidad Autónoma de Ciudad Juárez
DIRECCIÓN: Av. del Charro núm. 450 norte, col. Partido Romero, C. P. 32310, Ciudad Juárez, Chih., México
CORREO ELECTRÓNICO: al194548@alumnos.uacj.mx

Fecha de recepción: 15 de octubre de 2020. **Fecha de aceptación:** 27 de diciembre de 2020. **Fecha de publicación:** 29 de diciembre de 2020.



I. INTRODUCCIÓN

Para el presente trabajo, el *dominio* de la aplicación está contextualizado en las pequeñas y medianas empresas (PyMES) de distribución. Estas producen o compran productos al mayoreo para comercializar con empresas más pequeñas y requieren de múltiples bodegas donde resguardan su mercancía. El *especialista* de este dominio es el personal administrativo encargado exclusivamente a la logística.

En diversas empresas, existe la necesidad de atender la ineficiencia de la realización y la poca confiabilidad de los procesos relacionados con el seguimiento de mercancía en empresas distribuidoras. Por tanto, se ejecutó un proyecto con el objetivo general de proveer una solución *software como servicio* (SaaS, acrónimo del inglés *Software as a Service*) que permita la gestión de los envíos mediante una herramienta de monitoreo basada en el modelo SaaS para un óptimo control de los vehículos en ruta de las empresas distribuidoras.

II. METODOLOGÍA

Como se menciona en [1], las pequeñas empresas presentan dificultades al aplicar modelos de mejora destinados a empresas de mayor tamaño, ya que estos suponen una alta inversión de dinero y recursos. En consecuencia, el artículo expone a las metodologías ágiles como una opción práctica y viable para incrementar la calidad de los servicios de estas empresas.

Para la creación de la solución del presente trabajo, ha sido necesario definir cómo se realizará tal proyecto. Por ello, se seleccionó la metodología Kanban, que es ágil, dado que las compañías que se dedican a la creación de SaaS usualmente compiten en la fecha de lanzamiento de sus productos para asegurar la mayor cantidad posible de clientes [2]. Las metodologías ágiles se alinean a estas necesidades, ya que su objetivo principal es la creación de software útil en tiempos cortos de producción y, por tanto, utilizarlas en la creación de un SaaS impulsa la productividad [3].

Se optó por dicha metodología [4], dado lo siguiente:

- Tiene el principio del desarrollo incremental, ya que se dividirá el producto de software en pequeños prototipos funcionales y cada prototipo se realizará mediante etapas.

- Visualiza el flujo de trabajo y el avance de una manera simple.
- Limita el trabajo en curso con el fin de evitar cuellos de botella.
- Otorga flexibilidad en la asignación de prioridades de las tareas que conllevan los prototipos.
- Es personalizable, así que se utiliza en equipos de trabajo de distinto tamaño.

Es relevante mencionar que las metodologías ágiles consisten en iteraciones de revisiones y correcciones del prototipo, por lo que para generar una medida que indique la efectividad del trabajo realizado en las etapas se debe contabilizar las correcciones hechas en cada iteración [5]. Para garantizar la mejora continua del proyecto, una iteración debe tener un número menor de correcciones que la iteración anterior [6]. La etapa finaliza hasta que el usuario no realice ninguna corrección. De tal manera, uno de los indicadores de efectividad simples existentes consiste en la media de la disminución de las correcciones con respecto al tiempo.

En el proyecto, el tablero Kanban se estructuró con las siguientes etapas:

- Análisis/Diseño.
- Desarrollo (*backend*).
- Desarrollo (*frontend*).
- Pruebas.
- Despliegue.
- Revisión.
- Implementación.

El presente texto se centrará en los artefactos que se crearon en la parte de Análisis/Diseño. A grandes rasgos, esa etapa consiste en la recopilación rápida de información acerca de los requerimientos de los usuarios y la forma del despliegue dentro de la aplicación. Además, aborda la conceptualización del proyecto y su modelación.

En la construcción de un software, la parte más relevante es la definición y limitación de lo que se va a construir. Si esto se encuentra mal planteado desde el inicio, el software en ninguna medida podría ser la solución al problema planteado. Es por ello que se recurre al modelado conceptual en la parte de Análisis/Diseño, ya que es una herramienta de representación del contexto que se está abordando, incluyendo sus conceptos y participantes.

III. RESULTADOS Y DISCUSIÓN

Acorde a las actividades sugeridas en la metodología ágil, primero se realizó el Léxico Extendido del Dominio de Aplicación (LEL) para entender el significado de las palabras utilizadas en el dominio y después se desarrolló el modelo conceptual utilizando el Lenguaje Unificado de Modelado (UML). Estos dos artefactos sirven para determinar con claridad el dominio de las empresas distribuidoras.

Enseguida, se elaboraron épicas para definir los beneficios, a grandes rasgos, de lo que espera obtener el cliente y se realizaron historias de usuario, que son frases cortas de los servicios concretos que se desean.

Ambos artefactos son precedentes elementales para realizar los mapas de historia de usuario, los cuales son un desglose de las historias de usuario con el fin de crear requerimientos formales. Estos mapas están propiamente contextualizados por LEL y el modelo conceptual UML y debidamente redactados para ofrecer los beneficios esperados por las épicas planteadas. Por último, y como suma de lo anterior, se elaboraron las pantallas de usuario que permiten mostrar de manera específica el contenido que se visualizará.

A continuación, se presentan los artefactos creados en el proyecto.

A. Léxico Extendido del Dominio de Aplicación

Primeramente, se usó la ontología LEL como instrumento de representación del lenguaje utilizado en el dominio a analizar [7]. LEL se consideró el mejor punto de partida para la comprensión del contexto en el que está situado el usuario.

Es relevante mencionar que al inicio de la recopilación de requerimientos y especificaciones de un sistema de información se desconocen los procedimientos efectuados en el área y, por ende, se ignora el vocabulario usado por los especialistas del dominio. Por ello, se utilizó a LEL para describir las palabras y las nociones que se tienen de ellas, únicamente en el contexto logístico de las empresas distribuidoras. En la Tabla 1 se visualiza un extracto de este artefacto.

TABLA 1
EXTRACTO DEL LÉXICO EXTENDIDO DEL DOMINIO DE APLICACIÓN

NOMBRE ELEMENTO	NOCIÓN	IMPACTO
Parada.	Consiste en el transporte del vehículo de un lugar a otro.	Con el objetivo de mantenimiento del vehículo. Con el objetivo de entregar/recibir mercancía. Con el objetivo de traspasar la mercancía a otro vehículo.

B. Modelo UML del dominio de aplicación

En el desarrollo de software, se debe asignar tiempo suficiente para representar el escenario del especialista del dominio. En la representación se deben exponer los componentes, relaciones, reglas y restricciones dadas para que la contextualización esté lo más acorde al mundo real. Este proceso es complejo, por lo que se utilizan herramientas especializadas llamados modelos conceptuales [8].

Se presenta al modelo UML como instrumento básico de la representación conceptual de la realidad en el dominio de aplicación. Se dio uso de la información previa, procurando diferenciar la que es necesaria y la que es omisible. En la Figura 1 se muestra un fragmento del modelo que se realizó mediante el software Dia [9].

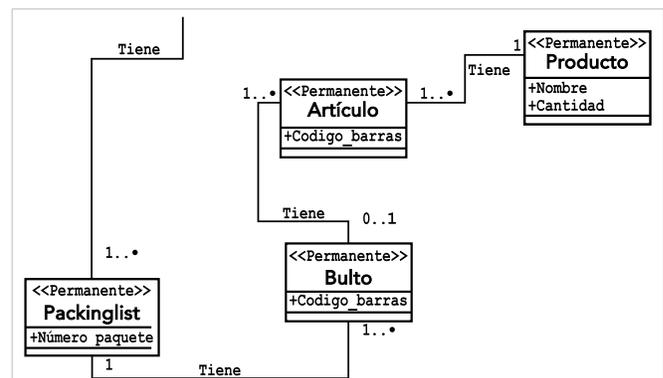


Figura 1. Parte del Modelo UML del dominio de aplicación realizado en el software Dia.

C. Épicas

Se puede hacer un modelado correcto del dominio de aplicación, sin embargo, si en la documentación del producto no se describe claramente el beneficio o mejora que los usuarios desean tener, es posible que la solución no satisfaga al cliente. Es por ello que en el proyecto se crearon épicas, las cuales son declaraciones de los beneficios que espera obtener el cliente [10].

En la Tabla 2 se expone una de las épicas creadas.

TABLA 2
PARTE DE LA ÉPICA TITULADA “MONITOREO Y GESTIÓN DE ENVÍOS”

NOMBRE	MONITOREO Y GESTIÓN DE ENVÍOS
PROPIETARIO	KARLA SELENE RAMÍREZ MÁRQUEZ
DESCRIPCIÓN	
	<ul style="list-style-type: none"> • Para PyMES de distribución. • Quienes necesitan gestionar los envíos que transportan mercancía mediante una herramienta de monitoreo para un óptimo control de los vehículos en ruta. • El SaaS para el seguimiento de mercancía. • Es un SaaS con arquitectura de microservicios que utiliza el personal de la empresa distribuidora. • Que ofrece una solución para el monitoreo y gestión de los envíos. • A diferencia de métodos manuales con documentación física y a otras aplicaciones que se encuentran personalizadas para una sola empresa. • Nuestra solución aplica una arquitectura de microservicios que acelera el rendimiento de los procesos. Ofrece una solución generalizada que es aplicable a cualquier empresa de distribución.

D. Historias de usuario

En las metodologías ágiles se redactan historias de usuario, las cuales son una representación escrita en una o dos frases de lo que el usuario espera del software [11]. Estas existen con el fin de verificar que los productos resultantes estén de acuerdo con las expectativas del cliente. Se optó por utilizarlas porque se recaban con facilidad y agilizan los procesos consecuentes. Las historias que fueron creadas para el proyecto contienen palabras especiales que son usadas por los especialistas del dominio, sin embargo, usando al LEL como apoyo, se puede lograr un entendimiento adecuado de las historias de usuario. A continuación, se presenta una historia del rol administrador en la Tabla 3.

TABLA 3
HISTORIA DE USUARIO DE LA FUNCIONALIDAD “CAPTURA DE DIRECCIÓN POR MEDIO DE MAPAS”

IDENTIFICADOR	SEG – 00004.
FUNCIONALIDAD	Captura de dirección por medio de mapas.
ROL	Administrador.
PRIORIDAD	Necesito.
DESCRIPCIÓN	Poder seleccionar un punto en un mapa, agregando esa dirección como dirección de bodega (sin necesidad de escribir la dirección).

E. Mapa de historias de usuario

Cabe destacar que las historias de usuario no son requerimientos formales. El especialista de la solución debe realizar un análisis de las historias para obtener requerimientos bien determinados y, con ello, realizar mapas de requerimientos ordenados.

Para el proyecto, estos mapas se realizaron en la plataforma web de Trello, que permitió desglosar los requerimientos de cada historia y además asignarle a un conjunto de historias una fase específica [12]. En la Figura 2 se presenta parte de lo realizado en Trello.

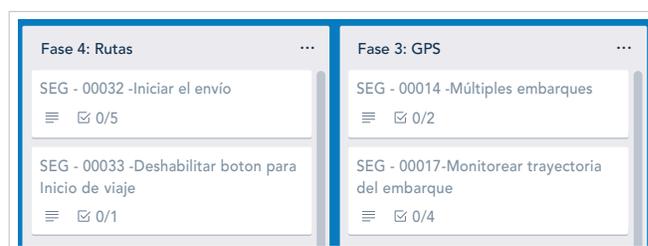


Figura 2. Extracto del mapa de historias de usuario realizada en el sistema web Trello.

Las fases fueron hechas para fungir como representación de los prototipos funcionales que serán probados por el usuario, de tal forma que no se podrá continuar con alguna fase posterior sin haber logrado una total satisfacción de la fase actual. En consecuencia, los mapas son propensos a cambiar en las iteraciones que se vayan efectuando, sin embargo, esto se hará con la finalidad de perfeccionar los requerimientos y, por ende, el software.

F. Diseño de pantallas

Con lo expuesto anteriormente, se prosiguió a realizar el diseño de las pantallas de usuario que muestran concretamente la interfaz gráfica de la aplicación SaaS. El propósito de diseñar las interfaces es definir las interacciones del usuario con el fin de que el sistema sea lo más simple y eficiente posible. Además, otro propósito es reducir los cambios imprevistos en la etapa de desarrollo de la aplicación para evitar contratiempos.

Las pantallas fueron creadas utilizando las herramientas de prototipado que ofrece el sitio web Mockflow [13]. Estas se pueden observar en la Figura 3.

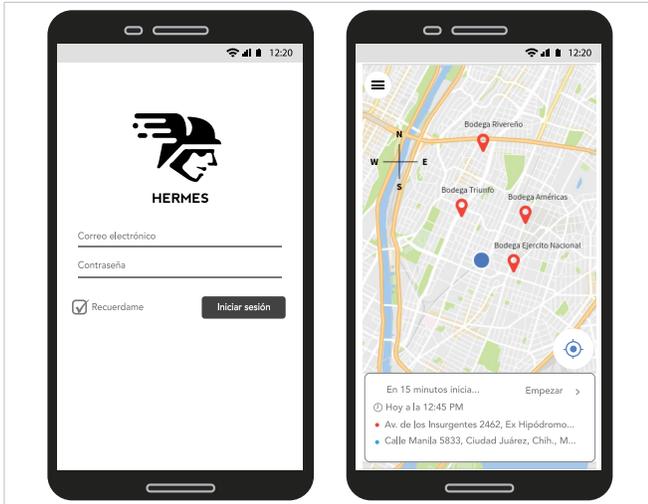


Figura 3. Pantalla de inicio de sesión y pantalla de inicio del usuario chofer diseñadas en el sistema web Mockflow.

IV. CONCLUSIONES

El modelo conceptual se define como una estrategia para el entendimiento de un dominio y/o la comprensión de las necesidades planteadas por los usuarios. Si el modelado es incorrecto en la realización de cualquier software, sin importar cuánto tiempo se le dedique a la construcción, no resolverá la necesidad planteada. Por esto, se debe utilizar la información representada en el modelo como base para la verificación con los usuarios y comprobar que este describe correctamente lo que se necesita. Los productos SaaS no son la excepción, por tanto, se deben realizar las investigaciones pertinentes para escoger la metodología más acorde a las necesidades y, por ende, que los artefactos de modelación de dicha metodología también sean los más adecuados.

REFERENCIAS

- [1] J. M. Navarro y J. Garzás, “Experiencia en la implantación de CMMI-DEV v1.2 en una micropyme con metodologías ágiles y software libre”, *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software*, vol. 6, no. 1, pp. 6-15, 2010. Disponible en: <https://www.redalyc.org/articulo.oa?id=92212873003>.
- [2] P. A. Quezada-Sarmiento y S. Mengual-Andrés, “Implementación de una solución web y móvil para la gestión vehicular basada en Arquitectura de Aspectos y metodologías ágiles: Un enfoque educativo de la teoría a la práctica”, *RISTI - Revista Ibérica de Sistemas e Tecnologías de Informação*, no. 25, pp. 98-111, 2017. DOI: 10.17013/risti.25.98-111.
- [3] S. Chávez, A. Martín, N. Rodríguez, M. Murazzo y A. Valenzuela, “Metodología ágil para el desarrollo de SaaS”, en *XIV Workshop de Investigadores en Ciencias de la Computación*, Rivadavia, 2012.
- [4] H. Lei, F. Ganjeizadeh, P. Kumar Jayachandran y P. Ozcan, “A statistical analysis of the effects of Scrum and Kanban on software development projects”, *Robotics and Computer-Integrated Manufacturing*, vol. 43, pp. 59-67, 2017. DOI: <https://doi.org/10.1016/j.rcim.2015.12.001>.
- [5] A. Rasnacis y S. Berzisa, “Method for Adaptation and Implementation of Agile Project Management Methodology”, *Procedia Computer Science*, vol. 104, pp. 43-50, 2017. DOI: 10.1016/j.procs.2017.01.055.
- [6] P. Serrador y J. K. Pintoc, “Does Agile work? — A quantitative analysis of agile project success”, *International Journal of Project Management*, vol. 33, no. 5, pp. 1040-1051, 2015. DOI: 10.1016/j.ijproman.2015.01.006.
- [7] G. Kaplan, G. Hadad y J. Doorn, Apuntes de la Cátedra, “Cátedra Ingeniería de Requerimientos”, Universidad Nacional de La Matanza, San Justo, 2019.
- [8] A. Olivé, *Conceptual Modeling of Information Systems*, Catalunya: Springer, 2007.
- [9] “Dia Diagram Editor”, The Dia Developers/Steffen Macke, 2004-2014. Consultado: Sept. 5, 2020. [En línea]. Disponible en: <http://dia-installer.de/>.
- [10] “Epics”, Scaled Agile, Inc., 2010-2020. Consultado: May. 8, 2020. [En línea]. Disponible en: <https://www.scaledagileframework.com/epic/>.
- [11] M. J. Rees, “A feasible user story tool for agile software development?,” *Ninth Asia-Pacific Software Engineering Conference, 2002*, Gold Coast, Queensland, Australia, 2002, pp. 22-30, DOI: 10.1109/APSEC.2002.1182972.
- [12] “Trello helps teams work more collaboratively and get more done”, Atlassian, 2020. [En línea]. Consultado: May. 9, 2020. Disponible en: <http://www.trello.com>.
- [13] “Super easy tool to brainstorm UI ideas”, MockFlow, 2020. Consultado: May. 9, 2020. [En línea]. Disponible en: <http://mockflow.com>.