

	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 1 de 9

21.1

FECHA	jueves, 9 de diciembre de 2021
--------------	--------------------------------

Señores
UNIVERSIDAD DE CUNDINAMARCA
 BIBLIOTECA
 Fusagasugá

UNIDAD REGIONAL	Sede Fusagasugá
TIPO DE DOCUMENTO	Trabajo De Grado
FACULTAD	Ingeniería
NIVEL ACADÉMICO DE FORMACIÓN O PROCESO	Pregrado
PROGRAMA ACADÉMICO	Ingeniería de Sistemas

El Autor(Es):

APELLIDOS COMPLETOS	NOMBRES COMPLETOS	No. DOCUMENTO DE IDENTIFICACIÓN
Malaver Gómez	Yessica Alexandra	1069769511

Director(Es) y/o Asesor(Es) del documento:

APELLIDOS COMPLETOS	NOMBRES COMPLETOS
Arenas Amado	Angela Patricia
Vásquez Gómez	Eva Patricia

Diagonal 18 No. 20-29 Fusagasugá – Cundinamarca
 Teléfono: (091) 8281483 Línea Gratuita: 018000180414
www.ucundinamarca.edu.co E-mail: info@ucundinamarca.edu.co
 NIT: 890.680.062-2

*Documento controlado por el Sistema de Gestión de la Calidad
 Asegúrese que corresponde a la última versión consultando el Portal Institucional*

	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 2 de 9

TÍTULO DEL DOCUMENTO

Desarrollo de un Prototipo que Permita la Adopción de un Proyecto Bajo el Marco de Trabajo Scrum en Equipos Ágiles

SUBTÍTULO

(Aplica solo para Tesis, Artículos Científicos, Disertaciones, Objetos Virtuales de Aprendizaje)

EXCLUSIVO PARA PUBLICACIÓN DESDE LA DIRECCIÓN INVESTIGACIÓN

INDICADORES	NÚMERO
ISBN	
ISSN	
ISMN	

AÑO DE EDICION DEL DOCUMENTO

09/12/2021

NÚMERO DE PÁGINAS

129

DESCRIPTORES O PALABRAS CLAVES EN ESPAÑOL E INGLÉS (Usar 6 descriptores o palabras claves)

ESPAÑOL	INGLÉS
1. Burn Down Chart	1. Burn Down Chart
2. Impact Mapping	2. Impact Mapping
3. User Story Mapping	3. User Story Mapping
4. Scrum	4. Scrum
5. tiny Scrum	5. tiny Scrum
6. Tablero Kanban	6. Kanban board

FUENTES (Todas las fuentes de su trabajo, en orden alfabético)

Adzic, G. (2012). Impact Mapping: Making a big impact with software products and projects. https://www.impactmapping.org/assets/impact_mapping_20121001_sample.pdf

Alyahya, S., Alqahtani, M., & Maddeh, M. (2016). Evaluation and improvements for agile planning tools. 2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA), 217–224. <https://doi.org/10.1109/SERA.2016.7516149>

Autentia (Ed.). (s/f). Fichas ágiles. Recuperado el 2 de mayo de 2021, de <https://www.autentia.com/wp-content/uploads/libros/Autentia-MazosAgile-v1.0.pdf>

	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 3 de 9

Avendaño, M. (2017, julio 26). Construyendo el Producto Mínimo Viable con User Story Mapping [Liderazgo y gestión]. <https://es.slideshare.net/Marcoviaweb/construyendo-el-producto-minimo-viable-con-user-story-mapping>

Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). Manifiesto por el Desarrollo Ágil de Software. agilemanifiesto. <https://agilemanifesto.org/iso/es/manifiesto.html>

Cohn, M. (2004). User Stories Applied for Agile Software Development. Pearson Education, Inc. <http://athena.ecs.csus.edu/~buckley/CSc191/User-Stories-Applied-Mike-Cohn.pdf>

digital.ai. (2021). 15th State of Agile Report (Núm. 15; State of Agile Report, p. 22). digital.ai. <https://explore.digital.ai/state-of-agile/15th-state-of-agile-report>

Frost, B. (2013, junio 10). Atomic Design. bradfrost.com. <https://bradfrost.com/blog/post/atomic-web-design/>

Frost, B. (2016). Atomic Design. <https://atomicdesign.bradfrost.com/table-of-contents/>

Gartner, Inc. (2020a). Enterprise Agile Planning Tools Reviews and Ratings. Gartner Peer Insights. <https://www.gartner.com/reviews/market/enterprise-agile-planning-tools>

Gartner, Inc. (2020b). Magic Quadrant for Enterprise Agile Planning Tools (Núm. G00394014). Gartner, Inc. <https://aptasolutions.com/wp-content/uploads/2020/06/Rally-Software-Gartner-2020.pdf>

Gartner, Inc. (2021). Magic Quadrant for Enterprise Agile Planning Tools (Núm. G00733887). Gartner, Inc. <https://www.gartner.com/doc/reprints?id=1-25TS5TMU&ct=210420&st=sb>

Garzas, J. (2013, enero 8). Retrospectivas: Qué son y técnicas para hacerlas. JavierGarzas.com. <https://www.javiergarzas.com/2013/01/retrospectiva-software-agil.html>

Garzas, J. (2015, octubre 16). Algunas dinámicas para hacer retrospectivas ágiles. JavierGarzas.com. <https://www.javiergarzas.com/2015/10/dinamicas-retrospectivas-agiles.html>

Google. (2020). Angular. One framework. Mobile & desktop. Angular. <https://angular.io/>

Google for Startups. (2016). Rapid Prototyping: Native | Google for Startups. <https://www.youtube.com/watch?v=lusOgox4xMI>

International Scrum Institute. (s/f). The Scrum Framework (3a ed.). International Scrum InstituteTM. https://www.scrum-institute.org/contents/The_Scrum_Framework_by_International_Scrum_Institute.pdf

Jaikrishnan, V. (2018, noviembre 26). Burndown Charts: What Is It And How To Use Them? Zepel. <https://zepel.io/agile/reports/burndown/>

	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 4 de 9

Kua, P. (2006, marzo 9). The Retrospective Starfish. [thekua.com@rest. http://www.thekua.com/rant/2006/03/the-retrospective-starfish/](http://www.thekua.com/rant/2006/03/the-retrospective-starfish/)

Meldre, M. (2016). Using Impact Mapping in Agile Software Product Management [University of Tartu]. <https://www.semanticscholar.org/paper/Magnus-Meldre-Using-Impact-Mapping-in-Agile-Product-Meldre/971bae898d58133db6cf8c78240d8d7d648b3b6c>

Menzinsky, A., López, G., & Palacio, J. (2019). Scrum Master: Temario Troncal I (2.6.1). Lubaris Info 4 Media SL. https://www.scrummanager.net/files/scrum_manager.pdf

Menzinsky, A., López, G., Palacio, J., Sobrino, M. Á., Álvarez, R., & Rivas, V. (2020). Historias de Usuario. Ingeniería de Requisitos Ágil: Vol. 3.0. (Scrum Manager, Ed.). Creative Commons.

Navarro Cadavid, A., Fernández Martínez, J. D., & Morales Vélez, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. PROSPECTIVA, 11(2), 30–39. <https://www.redalyc.org/articulo.oa?id=496250736004>

Palacio, M. (2020). Scrum Master—Temario troncal 1 (Scrum Manager, Ed.; 3.0.51). Creative Commons. https://scrummanager.net/files/scrum_master.pdf

Palminio, D. (2018, diciembre 7). Agilismo en Ingeniería de Software: Retrospectiva: Retro del Barco. Agilismo en Ingeniería de Software. <http://agilismoeningenieriadesoftware.blogspot.com/2018/12/retrospectiva-retro-del-barco.html>

Patton, J. (s/f). The New User Story Backlog is a Map. Jeff Patton & Associates. Recuperado el 28 de octubre de 2020, de <https://www.jpattontassociates.com/the-new-backlog/>

Patton, J. (2014). User Story Mapping: Discover the Whole Story, Build the Right Product (M. Treseler, A. Jollymore, & K. Ebrahim, Eds.; 1a ed.). O'Reilly Media Inc. <https://issuu.com/victoriashyshmakova/docs/user-story-mapping-jeff-patton-pete>

Piza Martínez, C. A. (2016). Scrum a la colombiana: Breve acercamiento al scrum hecho en Colombia. TIA, 4(2), 88–92. <https://revistas.udistrital.edu.co/index.php/tia/article/download/9692/pdf>

Schwaber, K., & Sutherland, J. (2020). La Guía de Scrum La Guía Definitiva de Scrum: Las Reglas del Juego. Creative Commons. <https://www.scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>

Scrum Alliance. (2017). HOW SCRUM IS USED. State of Scrum 2017-2018 scaling and agile transformation, 5–9. <https://www.scrumalliance.org/resources/ebooks>

solid IT. (2020). JIRA vs Pivotal Tracker vs Team Foundation Server vs Trello System Properties Comparison. Project Management Zone. <https://project-management.zone/system/jira,pivotal-tracker,team-foundation-server,trello>

	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 5 de 9

Targetprocess. (2020). Scrum. Targetprocess. https://www.targetprocess.com/solutions/scrum/#tldr
VersionOne Inc. (2019). The 13th annual State of Agile Report. The 13th annual State of Agile Report, 2. https://stateofagile.com/#
VersionOne Inc. (2020). The 14th Annual State of Agile Report. The 14th Annual State of Agile Report, 2. https://explore.digital.ai/state-of-agile/14th-annual-state-of-agile-report

RESUMEN DEL CONTENIDO EN ESPAÑOL E INGLÉS (Máximo 250 palabras – 1530 caracteres, aplica para resumen en español):
<p>El quinceavo informe de estado ágil realizado por digital.ai “destaco una vez más a Scrum como el enfoque ágil más popular con un 66%” de uso a nivel de equipo (digital.ai, 2021). Sin embargo, pese a su gran reconocimiento aún existe en varios equipos confusiones acerca de su uso. Por ello, el presente trabajo, explora las herramientas disponibles para gestionar proyectos ágiles con el uso de Scrum, a fin de diseñar, desarrollar, probar y validar a nivel de usuario y de expertos un prototipo que permite a los equipos en sus inicios familiarizarse con el marco de trabajo. Así es como surge tiny Scrum, un aplicativo Web que guía en la aplicación de las diferentes reglas de Scrum.</p> <p>The 15th State of Agile Report by digital.ai "once again highlighted Scrum as the most popular Agile approach with 66%" usage at the team level (digital.ai, 2021). However, despite its great recognition, there is still confusion in several teams about its use. Therefore, this work explores the tools available to manage agile projects with the use of Scrum, in order to design, develop, test, and validate at the user and expert level a prototype that allows teams in their early stages become familiar with the framework. This is how tiny Scrum arises, a Web application that guides the application of the different Scrum rules.</p>

AUTORIZACIÓN DE PUBLICACIÓN
<p>Por medio del presente escrito autorizo (Autorizamos) a la Universidad de Cundinamarca para que, en desarrollo de la presente licencia de uso parcial, pueda ejercer sobre mí (nuestra) obra las atribuciones que se indican a continuación, teniendo en cuenta que, en cualquier caso, la finalidad perseguida será facilitar, difundir y promover el aprendizaje, la enseñanza y la investigación.</p> <p>En consecuencia, las atribuciones de usos temporales y parciales que por virtud de la presente licencia se autoriza a la Universidad de Cundinamarca, a los usuarios de</p>

Diagonal 18 No. 20-29 Fusagasugá – Cundinamarca
Teléfono: (091) 8281483 Línea Gratuita: 018000180414
www.ucundinamarca.edu.co E-mail: info@ucundinamarca.edu.co
NIT: 890.680.062-2

*Documento controlado por el Sistema de Gestión de la Calidad
Asegúrese que corresponde a la última versión consultando el Portal Institucional*

	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 6 de 9

la Biblioteca de la Universidad; así como a los usuarios de las redes, bases de datos y demás sitios web con los que la Universidad tenga perfeccionado una alianza, son: Marque con una "X":

AUTORIZO (AUTORIZAMOS)	SI	NO
1. La reproducción por cualquier formato conocido o por conocer.	X	
2. La comunicación pública, masiva por cualquier procedimiento o medio físico, electrónico y digital.	X	
3. La inclusión en bases de datos y en sitios web sean éstos onerosos o gratuitos, existiendo con ellos previa alianza perfeccionada con la Universidad de Cundinamarca para efectos de satisfacer los fines previstos. En este evento, tales sitios y sus usuarios tendrán las mismas facultades que las aquí concedidas con las mismas limitaciones y condiciones.	X	
4. La inclusión en el Repositorio Institucional.	X	

De acuerdo con la naturaleza del uso concedido, la presente licencia parcial se otorga a título gratuito por el máximo tiempo legal colombiano, con el propósito de que en dicho lapso mi (nuestra) obra sea explotada en las condiciones aquí estipuladas y para los fines indicados, respetando siempre la titularidad de los derechos patrimoniales y morales correspondientes, de acuerdo con los usos honrados, de manera proporcional y justificada a la finalidad perseguida, sin ánimo de lucro ni de comercialización.

Para el caso de las Tesis, Trabajo de Grado o Pasantía, de manera complementaria, garantizo(garantizamos) en mi(nuestra) calidad de estudiante(s) y por ende autor(es) exclusivo(s), que la Tesis, Trabajo de Grado o Pasantía en cuestión, es producto de mi(nuestra) plena autoría, de mi(nuestro) esfuerzo personal intelectual, como consecuencia de mi(nuestra) creación original particular y, por tanto, soy(somos) el(los) único(s) titular(es) de la misma. Además, aseguro (aseguramos) que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el derecho a la imagen, intimidad, buen nombre y demás derechos constitucionales. Adicionalmente, manifiesto (manifestamos) que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la elaboración, presentación, investigación y, en general, contenidos de la Tesis o Trabajo de Grado es de mí (nuestra) competencia exclusiva, eximiendo de toda responsabilidad a la Universidad de Cundinamarca por tales aspectos.

	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 7 de 9

Sin perjuicio de los usos y atribuciones otorgadas en virtud de este documento, continuaré (continuaremos) conservando los correspondientes derechos patrimoniales sin modificación o restricción alguna, puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación de los derechos patrimoniales derivados del régimen del Derecho de Autor.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, “*Los derechos morales sobre el trabajo son propiedad de los autores*”, los cuales son irrenunciables, imprescriptibles, inembargables e inalienables. En consecuencia, la Universidad de Cundinamarca está en la obligación de RESPETARLOS Y HACERLOS RESPETAR, para lo cual tomará las medidas correspondientes para garantizar su observancia.

NOTA: (Para Tesis, Trabajo de Grado o Pasantía):

Información Confidencial:

Esta Tesis, Trabajo de Grado o Pasantía, contiene información privilegiada, estratégica, secreta, confidencial y demás similar, o hace parte de la investigación que se adelanta y cuyos resultados finales no se han publicado.

SI ___ **NO** **X**.

En caso afirmativo expresamente indicaré (indicaremos) en carta adjunta, expedida por la entidad respectiva, la cual informa sobre tal situación, lo anterior con el fin de que se mantenga la restricción de acceso.

LICENCIA DE PUBLICACIÓN

Como titular(es) del derecho de autor, confiero(erimos) a la Universidad de Cundinamarca una licencia no exclusiva, limitada y gratuita sobre la obra que se integrará en el Repositorio Institucional, que se ajusta a las siguientes características:

- a) Estará vigente a partir de la fecha de inclusión en el repositorio, por un plazo de 5 años, que serán prorrogables indefinidamente por el tiempo que dure el derecho patrimonial del autor. El autor podrá dar por terminada la licencia solicitándolo a la Universidad por escrito. (Para el caso de los Recursos Educativos Digitales, la Licencia de Publicación será permanente).
- b) Autoriza a la Universidad de Cundinamarca a publicar la obra en formato y/o soporte digital, conociendo que, dado que se publica en Internet, por este hecho circula con un alcance mundial.
- c) Los titulares aceptan que la autorización se hace a título gratuito, por lo tanto, renuncian a recibir beneficio alguno por la publicación, distribución, comunicación

Diagonal 18 No. 20-29 Fusagasugá – Cundinamarca
 Teléfono: (091) 8281483 Línea Gratuita: 018000180414
www.ucundinamarca.edu.co E-mail: info@ucundinamarca.edu.co
 NIT: 890.680.062-2

 UDECA UNIVERSIDAD DE CUNDINAMARCA	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 8 de 9

pública y cualquier otro uso que se haga en los términos de la presente licencia y de la licencia de uso con que se publica.

d) El(Los) Autor(es), garantizo(amos) que el documento en cuestión es producto de mi(nuestra) plena autoría, de mi(nuestro) esfuerzo personal intelectual, como consecuencia de mi (nuestra) creación original particular y, por tanto, soy(somos) el(los) único(s) titular(es) de la misma. Además, aseguro(aseguramos) que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el derecho a la imagen, intimidad, buen nombre y demás derechos constitucionales. Adicionalmente, manifiesto (manifestamos) que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la elaboración, presentación, investigación y, en general, contenidos es de mí (nuestro) competencia exclusiva, eximiendo de toda responsabilidad a la Universidad de Cundinamarca por tales aspectos.

e) En todo caso la Universidad de Cundinamarca se compromete a indicar siempre la autoría incluyendo el nombre del autor y la fecha de publicación.

f) Los titulares autorizan a la Universidad para incluir la obra en los índices y buscadores que estimen necesarios para promover su difusión.

g) Los titulares aceptan que la Universidad de Cundinamarca pueda convertir el documento a cualquier medio o formato para propósitos de preservación digital.

h) Los titulares autorizan que la obra sea puesta a disposición del público en los términos autorizados en los literales anteriores bajo los límites definidos por la universidad en el “Manual del Repositorio Institucional AAAM003”

i) Para el caso de los Recursos Educativos Digitales producidos por la Oficina de Educación Virtual, sus contenidos de publicación se rigen bajo la Licencia Creative Commons: Atribución- No comercial- Compartir Igual.



j) Para el caso de los Artículos Científicos y Revistas, sus contenidos se rigen bajo la Licencia Creative Commons Atribución- No comercial- Sin derivar.



Nota:

Si el documento se basa en un trabajo que ha sido patrocinado o apoyado por una entidad, con excepción de Universidad de Cundinamarca, los autores garantizan

Diagonal 18 No. 20-29 Fusagasugá – Cundinamarca

Teléfono: (091) 8281483 Línea Gratuita: 018000180414

www.ucundinamarca.edu.co E-mail: info@ucundinamarca.edu.co

NIT: 890.680.062-2

	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 9 de 9

que se ha cumplido con los derechos y obligaciones requeridos por el respectivo contrato o acuerdo.

La obra que se integrará en el Repositorio Institucional está en el(los) siguiente(s) archivo(s).

Nombre completo del Archivo Incluida su Extensión (Ej. Nombre completo del proyecto.pdf)	Tipo de documento (ej. Texto, imagen, video, etc.)
1. DesarrolloPrototipoAdopcionScrum.pdf	Texto
2. tiny-Scrum-app.zip	Carpeta comprimida
3.	
4.	

En constancia de lo anterior, Firmo (amos) el presente documento:

APELLIDOS Y NOMBRES COMPLETOS	FIRMA (autógrafa)
Malaver Gómez Yessica Alexandra	Yessica A. Malaver G.

21.1-51-20.

**Desarrollo de un Prototipo que Permita la Adopción de un Proyecto Bajo el Marco de
Trabajo Scrum en Equipos Ágiles**

Yessica Alexandra Malaver Gómez

Trabajo de grado para optar el título de Ingeniero de Sistemas

Director Disciplinar

Angela Patricia Arenas Amado

Ingeniero de Sistemas

Asesor Metodológico

Eva Patricia Vásquez Gómez

Ingeniero de Sistemas

Universidad de Cundinamarca

Facultad de Ingeniería

Programa de Ingeniería de Sistemas

Fusagasugá

2021

Contenido

1.	Introducción.....	14
2.	Objetivos.....	17
	Objetivo General	17
	Objetivo Específicos	17
3.	Alcance y Limitaciones	18
	Alcance.....	18
	Limitaciones	18
4.	Marco Referencial	19
4.1.	Estado del Arte	19
4.2.	Marco Conceptual.....	26
	4.2.1. Metodologías Tradicionales y Ágiles	26
	4.2.2. Manifiesto Ágil.....	27
	4.2.3. Scrum	29
	4.2.4. Origen de Scrum	29
	4.2.5. Teoría de Scrum.....	29
	4.2.6. Historia De Usuario	30
	4.2.7. Roles de Scrum o Scrum Team.....	31
	4.2.8. Eventos de Scrum	32
	4.2.9. Artefactos de Scrum.....	34
	4.2.10. Técnicas de Scrum	34
	4.2.11. Desarrollo de Páginas Web.....	42
	4.2.12. Angular	42

4.2.13. Diseño Atómico o Atomic Design.....	43
5. Desarrollo	44
5.1. Materiales y Métodos	44
5.2. Metodología y Proceso Metodológico del Proyecto.....	45
5.2.1. Fases del Proyecto.....	45
5.3. Planeación y Administración del Proyecto.....	46
5.3.1. Cronograma de Actividades.....	46
5.3.2. Recursos y Fuentes Financieras	48
5.4. Construcción del Prototipo	49
5.4.1. Fundamentos para la Creación del Prototipo	49
5.4.2. Prototipo Propuesto.....	50
5.5. Desarrollo del Prototipo.....	57
5.5.1. Roles	57
5.5.2. Definición del Producto	58
5.5.3. Sprint Backlog del Sprint 1.....	64
5.5.4. Sprint Backlog del Sprint 2.....	67
5.5.5. Sprint Backlog del Sprint 3.....	71
5.5.6. Sprint Backlog del Sprint 4.....	73
5.5.7. Sprint Backlog del Sprint 5.....	77
5.5.8. Diseño del Software.....	82
5.5.9. Desarrollo del Software	87
5.5.10. Pruebas del Software	91
5.5.11. Validación del Software.....	93
6. Resultados.....	98
6.1. Resultados del Desarrollo de Software del Prototipo	98
6.2. Resultados de la Validación del Prototipo.....	109

6.2.1. Validación del Prototipo con Estudiantes	110
6.2.2. Validación del Prototipo con Expertos	115
6.2.3. Análisis General de la Validación.....	120
7. Recomendaciones	122
8. Conclusiones.....	124
Referencias Bibliográficas	125

Lista de Tablas

	pág.
Tabla 1 Metodologías tradicionales vs metodologías ágiles.....	27
Tabla 2 Cronograma de actividades.....	47
Tabla 3 Recursos y fuentes de financiamiento.	48
Tabla 4 Técnicas conceptuales del prototipo.	49
Tabla 5 Roles del Scrum Team del proyecto.	57
Tabla 6 Subroles y funciones del Scrum Team del proyecto.....	58
Tabla 7 Cronograma de los Sprints.....	62
Tabla 8 Sprint Backlog del Sprint 1.....	64
Tabla 9 Sprint Backlog del Sprint 2.....	68
Tabla 10 Sprint Backlog del Sprint 3.....	71
Tabla 11 Sprint Backlog del Sprint 4.....	74
Tabla 12 Sprint Backlog del Sprint 5.....	78
Tabla 13 Cuestionario de la validación del prototipo de software.....	94
Tabla 14 Comentarios de los expertos.	120

Lista de Figuras

	pág.
Figura 1 Cuadro mágico de Gartner de herramientas de planificación ágil empresarial 2020....	20
Figura 2 Cuadro mágico de Gartner de herramientas de planificación ágil empresarial 2021....	21
Figura 3 Ejemplo de una tarjeta de historia de usuario.....	31
Figura 4 Ejemplo de Impact Mapping.	35
Figura 5 Ejemplo de los componentes del Impact Mapping.....	36
Figura 6 Product Backlog plano y Product Backlog construido con la técnica User Story Mapping.....	37
Figura 7 Esquema del User Story Mapping.....	37
Figura 8 Gestión del User Story Mapping.....	38
Figura 9 Ejemplo de Sprint Backlog con Tablero Kanban.....	39
Figura 10 Ejemplo de la estructura de un burn down chart.....	40
Figura 11 Estructura de componentes de la dinámica "Estrella de mar".....	41
Figura 12 Ejemplo de los componentes de la dinámica "Barco de vela".....	42
Figura 13 Estructura del Diseño Atómico o Atomic Design.....	43
Figura 14 Prototipo de integración conceptual de los roles, artefactos, técnicas y eventos de Scrum.....	51
Figura 15 Relación de Impact Mapping con el User Story Mapping y priorización MoSCoW..	53
Figura 16 Relación del User Story Mapping con el Sprint Backlog.....	54
Figura 17 Relación del Sprint Backlog con el burn down chart.....	55
Figura 18 Relación del Sprint con las dinámicas para retrospectivas.....	56
Figura 19 Impact Mapping del proyecto.....	59

Figura 20	Vista de una historia de usuario personalizada en Jira Software.....	60
Figura 21	Vista del Product Backlog en Jira Software.....	61
Figura 22	User Story Mapping del proyecto.....	63
Figura 23	Diagrama de Contexto del Prototipo.....	83
Figura 24	Diagrama del Contenedor del Prototipo.....	84
Figura 25	Diagrama de componentes del prototipo.....	86
Figura 26	Diseño atómico del Impact Mapping del software.....	88
Figura 27	Diagrama de código del prototipo.....	89
Figura 28	Orden de los modelos que componen el JSON de la base de datos no relacional.	90
Figura 29	Ejemplo del testeo de una historia de usuario.....	91
Figura 30	Evidencia de testeo en Windows 10.....	92
Figura 31	Evidencia de testeo en Deepin.....	92
Figura 32	Ciclo de creación de un prototipo.....	93
Figura 33	Home de Inicio de Sesión de tiny Scrum.....	98
Figura 34	Login de tiny Scrum con Auth0.....	99
Figura 35	Aplicación de tiny Scrum en Auth0.....	99
Figura 36	Home de proyectos de tiny Scrum.....	100
Figura 37	Dialog de creación o edición de un proyecto.....	100
Figura 38	Módulo de Impact Mapping.....	101
Figura 39	Módulo de User Story Mapping.....	102
Figura 40	Vista completa del User Story Mapping.....	102
Figura 41	Historia de usuario en tiny Scrum.....	103
Figura 42	Planeación de un Sprint en tiny Scrum.....	104

Figura 43	User Story Mapping con los Sprint existentes.	105
Figura 44	Sprint existentes y Sprint Backlog.	105
Figura 45	Modulo de Sprint Backlog.....	106
Figura 46	Módulo de Burn Down Chart.....	107
Figura 47	Módulo de retrospectiva: dinámica del barco de vela.	108
Figura 48	Campos de la dinámica del barco de vela.	108
Figura 49	Módulo de retrospectiva: dinámica de la estrella de mar.	109
Figura 50	Interfaz de usuario: Grado de intuición y agradabilidad.	110
Figura 51	Estructura: Grado de facilidad de uso.	111
Figura 52	Estructura: Grado de organización de los módulos.	111
Figura 53	Utilidad: Grado de utilidad, de satisfacción y de recomendabilidad.	112
Figura 54	Rendimiento: Grado de rendimiento.	113
Figura 55	Rendimiento: Gravedad de fallos.	113
Figura 56	Pilares de Scrum: Grado de cumplimiento.	114
Figura 57	Reglas de Scrum: Grado de relación.	114
Figura 58	Perfil de los expertos.	116
Figura 59	Estructura: Grado de organización de los módulos (Expertos).	117
Figura 60	Utilidad: Grado de utilidad, de satisfacción y de recomendabilidad (expertos).	117
Figura 61	Pilares de Scrum: Grado de cumplimiento (expertos).	118
Figura 62	Reglas de Scrum: Grado de relación (expertos).	119

Glosario

- API REST o RESR API: Es una interfaz de programación de aplicaciones (API o API web) que se ajusta al estilo de arquitectura REST con el uso de solicitudes HTTP como:
 - GET: Lectura de datos.
 - POST: Crear datos.
 - PUT: Actualizar datos.
 - DELETE: Borrar datos.
- Arquitectura cliente – servidor: Es un modelo de diseño de software donde el servidor brinda una serie de servicios o recursos que son consumidos por el cliente.
- Arquitectura en capas: Consiste en dividir la aplicación en capas a fin de que cada capa tenga un rol definido.
- Auth0: Plataforma de identidad como servicio (IDaaS) que permite la autenticación y autorización de usuarios para aplicaciones.
- Base de datos: Sitio de almacén de datos.
- Cliente: Puede ser un ordenador o una aplicación informática, que requiere de la red para funcionar.
- Directiva en Angular: Permiten extender las funcionalidades del HTML, usando lógica que puede ser ejecutada en el DOM.
- DOM: En inglés “Document Object Model” y en español “Modelo de Objetos del Documento”, es un modelo de documento que se carga en el navegador y que representa el documento como un árbol de nodos, en donde cada nodo representa una parte del documento, y a los cuales se pueden acceder.

- **Endpoint:** Son las URLs de un API o un Backend que responden a una petición.
- **Enum o Enumerados en TypeScript:** Sirven para definir un tipo de datos con un conjunto de valores identificables, los cuales se pueden usar para asignarlos a una variable, o realizar comprobaciones en una sentencia de decisión.
- **IDaaS:** Del inglés “Identity as a Service”, es un servicio de autenticación basado en la nube que gestiona y operado por un proveedor externo.
- **Firebase:** Es una plataforma para el desarrollo de aplicaciones web y móviles, dentro de sus servicios esta Firebase Realtime Database que es una base de datos NoSQL alojada en la nube, donde los datos se almacenan en formato JSON.
- **HTTPS:** El protocolo seguro de transferencia de hipertexto, es un protocolo que permite establecer una conexión segura entre el servidor y el cliente. Es decir, es la versión segura de HTTP.
- **JSON (JavaScript Object Notation).** Es un formato ligero para almacenar y transportar datos, se usan generalmente cuando los datos desde un servidor a una página web y es fácil de leer.
- **Puerto lógico:** Se encuentra ubicado dentro del equipo informático y permite establecer comunicación con diferentes programas, realizar distribuciones de servicios y flujo de datos.
- **REST:** La transferencia de estado representacional (en inglés representational state transfer) es una interfaz para conectar varios sistemas basados en el protocolo HTTP, a fin de obtener y generar datos y operaciones , donde esos datos se devuelven en formatos como XML o JSON.

- Scrum - Formación de Rugby: Scrum es una formación en el deporte de Rugby donde ambos equipos “agazapados y atenazados entre sí, empujan para obtener el balón, y sacarlo de la formación sin tocarlo con la mano” (Menzinsky et al., 2019).
- Servicio: Conjunto de información que busca responder a las necesidades de un cliente.
- Servidor: Proveedor de servicios o de recursos, que puede ser un ordenador o una aplicación informática que envía información a los demás agentes de la red.
- Token: Es una referencia que regresa datos sensibles de forma segura.

Resumen

El quinceavo informe de estado ágil realizado por digital.ai “destaco una vez más a Scrum como el enfoque ágil más popular con un 66%” de uso a nivel de equipo (digital.ai, 2021). Sin embargo, pese a su gran reconocimiento aún existe en varios equipos confusiones acerca de su uso. Por ello, el presente trabajo, explora las herramientas disponibles para gestionar proyectos ágiles con el uso de Scrum, a fin de diseñar, desarrollar, probar y validar a nivel de usuario y de expertos un prototipo que permite a los equipos en sus inicios familiarizarse con el marco de trabajo. Así es como surge tiny Scrum, un aplicativo Web que guía en la aplicación de las diferentes reglas de Scrum.

Palabras Clave: Burn Down Chart, Impact Mapping, Dinámicas para retrospectivas, Scrum, tiny Scrum, tablero Kanban, User Story Mapping.

Abstract

The 15th State of Agile Report by digital.ai "once again highlighted Scrum as the most popular Agile approach with 66%" usage at the team level (digital.ai, 2021). However, despite its great recognition, there is still confusion in several teams about its use. Therefore, this work explores the tools available to manage agile projects with the use of Scrum, in order to design, develop, test, and validate at the user and expert level a prototype that allows teams in their early stages become familiar with the framework. This is how tiny Scrum arises, a Web application that guides the application of the different Scrum rules.

Key Word- Burn Down Chart, Impact Mapping, Dynamics for retrospectives, Scrum, tiny Scrum, Kanban board, User Story Mapping.

1. Introducción

Cuando los equipos de desarrollo de software eligen Scrum para sus proyectos, se enfrentan a un gran reto: saber y entender cómo funciona, cómo se usa y que técnicas o herramientas brinda. Pues, la malinterpretación de lo que ofrece Scrum como sus artefactos, técnicas o eventos hace que los equipos se olvidan de escudriñar la esencia de este marco de trabajo, tanto así que Piza Martínez (2016) de la Universidad Distrital de Colombia, asegura que Scrum “es el molde al que debemos ajustarnos” pero en el país lo ven antes “como un material maleable y que según quién lo use se convertirá en otra cosa, no Scrum” y efectivamente deja de ser un marco de trabajo para convertirlo en una metodología que a veces ni siquiera es ágil sino clásica, perdiendo el valor que puede llegar a generar, lo suficiente para que Piza afirme que en ese caso es mejor “hacer un alto en el camino y volver a RUP que es lo tradicional” (Piza Martínez, 2016).

Como consecuencia, la confusión persiste y hace que los equipos busquen encontrar en él fases o pasos a seguir forzándolo a comportarse como una metodología con métodos secuenciales para llenar el vacío que sienten, en vez de emplear su naturaleza de aprendizaje cíclica que busca con este maximizar el valor de negocio y si es necesario hacer cambios.

Lo anterior en parte se debe al no distinguirse la diferencia entre metodología y marco de trabajo, desconcierto que no solo es del país pues incluso se da en reportes mundiales sobre agilismo de VersionOne. Donde el decimotercer reporte anual dice: “Scrum es una vez más la "metodología" ágil más ampliamente practicada” (VersionOne Inc., 2019). Sin embargo para el año 2020 el decimocuarto reporte dice: “Scrum es el método/marco ágil más practicado”

(VersionOne Inc., 2020), aunque después lo llama “metodología”, como si se tratara de sinónimos.

Los errores mencionados anteriormente se deberían solucionar desde la academia, universidad o instituto, sin embargo aún hay instituciones que lo siguen permitiendo consciente o inconscientemente pues no siempre hay forma de aprender de los errores si no hay alguien que lo corrija y le haga ver su equivocación, generando que muchos estudiantes no conozcan la esencia de Scrum y si lo llegan a conocer se debe al fruto de su trabajo autónomo e investigaciones propias o tienen que esperar hasta cuando sean profesionales y lleguen a empresas que lo implementan y así adquirir el conocimiento por experiencias fuera de su proceso educativo.

Debido a las confusiones que se tienen con este marco de trabajo no es de extrañar que los encargados del proyecto no conozcan las funciones de su rol, pues “este tema de definir los roles cuando se está empezando es un poco complejo para el entendimiento de quienes se están desprendiendo de RUP o similares” (Piza Martínez, 2016), lo que puede generar que no haya separación de sus funciones creando desorden sobre el equipo de trabajo y en el proyecto.

Como vemos, es evidente que existe un problema de falta de claridad en los equipos ágiles respecto al uso de Scrum, sus artefactos, técnicas, eventos, y el hecho de que sea un marco de trabajo. Por lo que las confusiones y malas prácticas en el “uso” de Scrum en los proyectos de los equipos ágiles, hace necesaria la búsqueda de una estrategia conveniente para conocer el funcionamiento del marco de trabajo con más éxito a nivel mundial, tal como lo indica una encuesta realizada en el otoño del 2017 a 2000 miembros de Scrum Alliance, donde “94% de los encuestados usan Scrum en su práctica ágil” (Scrum Alliance, 2017). Y así poder aplicarlo de una forma que esté de acuerdo con el agilismo, las técnicas que ofrece e incluso difundir las

prácticas de Scrum pero de la forma correcta. Sin embargo, ¿Qué estrategia se puede emplear para una mejor adopción del marco de trabajo Scrum en un proyecto ágil?

De esta forma surge el trabajo en curso que busca por medio del diseño de un prototipo guiar a los equipos de desarrollo en la adopción de Scrum en sus proyectos, para así dar a conocer sus distintas técnicas y artefactos que permiten crear valor de negocio de forma incremental y obtener un ROI en etapas tempranas de un proyecto.

Así, teniendo un propósito claro, el presente documento muestra los resultados del proceso de búsqueda referencial de la investigación donde se analizó el marco de trabajo y las técnicas que han surgido para apoyarlo por medio de un análisis de antecedentes, en el cual se expresa las distintas herramientas de gran renombre a nivel de agilismo como Jira, Targetprocess, a fin de conocer sus puntos a favor y en contra y obtener una visión de la solución: generar un prototipo de una herramienta de uso básico, no de uso prolongado, pero que permita dar guía en la adopción de los diferentes roles, artefactos y eventos, funcionando como anticipo al innumerable mundo de herramientas especializadas en Scrum para desarrollo de software. Por lo que el proceso realizado para su desarrollo se amplía a continuación.

2. Objetivos

Objetivo General

Desarrollar un prototipo que permita la adopción de un proyecto bajo el marco de trabajo Scrum en equipos ágiles.

Objetivo Específicos

- Identificar artefactos y técnicas que faciliten la adopción de Scrum.
- Diseñar la relación de los artefactos y técnicas con su respectivo rol.
- Diseñar un aplicativo Web que guíe en el uso de los principales artefactos y técnicas de Scrum.

3. Alcance y Limitaciones

Alcance

Desarrollar un prototipo de software, en forma de aplicativo Web, con los siguientes 5 módulos: Impact Mapping, User Story Mapping (Product Backlog) que permitirá la planeación del Sprint, Sprint Backlog (Tablero Kanban), burn down chart, y Dinámicas para retrospectivas ya sea “Estrella de mar” o “Barco de vela”, con el acompañamiento de ayudas visuales que le permitirá al usuario conocer las acciones a realizar de cada rol Scrum.

Limitaciones

Durante el desarrollo del software, se presentó como reto y limitante el uso de nuevas tecnologías como Angular, Angular material, Firebase, ng2Chart y Auth0, puesto que se requirió de tiempo para aprender a usarlas y para encontrar la manera de emplearlas en el proyecto.

Por otro lado, el prototipo creado permite aplicar el Scrum establecido hasta ahora, dado que el marco de trabajo puede estar sujeto a cambios en las guías oficiales como ha ocurrido últimamente, o el Scrum Master puede establecer su manera de usar Scrum, generando que algunas de sus funcionalidades se amplíen, reduzcan o lleguen a sobrar a futuro. Así mismo, existen limitantes a nivel de usuario, ya que es necesario que cuente con conocimientos básicos sobre Scrum, pues de lo contrario podría apegarse a lo tradicional y forzar al marco de trabajo a comportarse como una metodología secuencial.

4. Marco Referencial

4.1. Estado del Arte

Menzinsky et al. (2019) afirma que Scrum fue definido teóricamente a principios de los 80 por Nonaka y Takeuchi quienes dieron protagonismo a los equipos de trabajo resaltando la importancia de que fueran autoorganizados y motivados, además de ofrecer un gran auge en el agilismo al solapar las fases del desarrollo de sistemas y ser precursores de la adaptación continúa debido a las incertidumbres que se pueden presentar y al “multiaprendizaje” que se busca generar con la aplicación del marco de trabajo (págs. 9, 12-15).

De tal manera, que al tener tantos años en curso Scrum, es muy común encontrar herramientas de apoyo virtuales que permitan cumplir con la parte técnica que maneja, es decir, los roles, eventos y artefactos. Por tal motivo, es importante analizar lo que brindan las herramientas más reconocidas y las que han sido empleadas en otros estudios para así conocer que ofrecen, que errores cometieron o que se puede mejorar de las mismas, por lo cual a continuación se realiza un breve estudio de los software que son distinguidos por sus funcionalidades en los artefactos y eventos de Scrum.

Como primera medida, es conveniente analizar la opinión del estudio e informe realizado por la empresa investigadora, consultora y asesora Gartner la cual definió un cuadrante mágico eligiendo 16 proveedores de herramientas de planificación ágil empresarial (EPA) en el 2020, generando una clasificación de líderes, retadores, visionarios, y “jugadores de nicho”, como se puede ver en la Figura 1. Destacando en el grupo de líderes Jira Align de Atlassian, Rally de Broadcom, VersionOne y Continuum de CollabNet VersionOne, Planview Lean y Agile Delivery de Planview, y Targetprocess. En el caso de Retadores se distinguen las herramientas

Azure DevOps and GitHub de Microsoft, y IT Business Management (ITBM) de ServiceNow. Para el grupo de visionarios se destacan Storyteller de Blueprint, y GitLab. Por último, en el caso de los “jugadores de nicho” se menciona a SwiftKanban, SwiftEASe, SwiftEnterprise and SwiftSync de Digité, Favro, IBM Engineering Workflow Management de IBM, SpiraPlan de Inflectra, ALM Octane y Project and Portfolio Management de Micro Focus, Polarion ALM de Siemens, y Jile de TCS según el informe de Gartner, Inc. (2020b).

Figura 1

Cuadro mágico de Gartner de herramientas de planificación ágil empresarial 2020.



Nota: La figura muestra las diferentes empresas proveedoras de herramientas de planificación ágil. Fuente: Gartner, Inc. (2020b).

También se tiene en cuenta la investigación realizada por Gartner para el 2021 como se ve en la Figura 2, donde Atlassian, Planview, Targetprocess, Broadcom, Digital.ai con su fusión

con CollabNet VersionOne, siguen siendo líderes de este cuadro mágico, pero ahora tras su evolución se suma ServiceNow y GitLab a su liderato; Microsoft conserva su postura de retador pero la comparte con IBM y Siemens que se transformaron en retadores, por ultimo no hay visionarios pues Blueprint se niega a participar en este estudio posiblemente a su cambio de enfoque, pero si hay jugadores de nicho que conservan su lugar como Inflectra, Digité, TCS, Favro, Micro Focus y se añade Planisware con sus herramientas: Planisware Enterprise y Planisware Orchestra.

Figura 2

Cuadro mágico de Gartner de herramientas de planificación ágil empresarial 2021.



Fuente: Gartner, Inc. (2021).

Otro estudio que se tiene en cuenta en este análisis es el realizado por Alyahya et al., (2016), en un artículo de IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA) titulado “Evaluation and improvements for agile planning tools, en el cual se estudiaron 30 herramientas las cuales coinciden con gran parte de las mencionadas por Gartner, tales como: Version One, Rally, TargetProcess, y Jira. Además menciona otras que aunque no son contempladas dentro de las 16 elegidas por Gartner, también son importantes como Pivotal Tracker.

Un último estudio contemplado es el de Project Management Zone realizado por la empresa austriaca de consultoría de TI (Tecnología de la Información) solid IT (2020), en el cual se realiza una comparativa entre 4 herramientas como lo son JIRA, Pivotal Tracker, Team Foundation Server y Trello.

Teniendo en cuenta las anteriores herramientas, a continuación se realiza una breve descripción de cada una de ellas de acuerdo con los informes mencionados.

- Jira. Los productos de Jira están liderando el mercado tales como Jira Align, Jira Software con su componente de Jira Agile para dar soporte en metodologías o marcos de trabajo ágiles como Scrum. Dentro de los componentes que brinda incluye gestión de versiones, creación del Product Backlog, de las historias de usuario, del Sprint Planning, Scrum board parecido a un tablero Kanban, e incluso la creación de Story Maps como lo menciona Jira en su página oficial, lo suficiente para que Gartner afirme que “Jira es la herramienta perfecta para Scrum” (Gartner, Inc., 2020a). Sin embargo, Gartner plantea una precaución importante “Aunque su interfaz de usuario es elegante, Jira Align de

Atlassian es muy extenso, por lo que puede llevar algo de tiempo y análisis encontrar la mejor manera de usarla en una organización” (Gartner, Inc., 2020b).

- Rally. “trabaja como un verdadero socio para grandes empresas con un enfoque en Lean y Agile” (Gartner, Inc., 2020b). Posee una interfaz de usuario actualizada la cual permite “la técnica de arrastrar y soltar utilizada para priorizar las historias de usuario en el Product Backlog” e incluso garantiza la planificación de los releases, las iteraciones o Sprint supervisando que “cuando la estimación total de tareas asignadas a un miembro excede su capacidad para la iteración, se mostrará un color rojo en la capacidad de estado del miembro que indica que está sobrecargado” (Alyahya et al., 2016).
- VersionOne. “CollabNet VersionOne es totalmente compatible con las metodologías Scrum, Kanban, XP y SAFe, así como con el "modelo Spotify” (...) la amplia funcionalidad y personalización del producto significa que requiere cierto esfuerzo para configurarlo correctamente” (Gartner, Inc., 2020b). Ahora hace parte de Digital.ai, se caracteriza porque permite la creación y gestión del product planning, release planning, Sprint Planning, e incluso el seguimiento de historias. Por otro lado, en la herramienta se puede observar que la creación de las historias de usuario solo se componen de un campo de descripción para indicar la información concerniente a la misma, además de otros datos relacionados al Sprint y el tiempo.
- Targetprocess. “Targetprocess ofrece una solución flexible a nivel de equipo con un excelente soporte para prácticas basadas en Kanban y Scrum” (Gartner, Inc., 2020b). Proporciona herramientas para Scrum en aspectos como “gestión del Backlog, Sprint

Planning, Sprint Progress y Daily Scrum, Sprint Review, y Sprint Retrospective” (Targetprocess, 2020).

- GitHub de Microsoft. Hace parte de los retadores en el cuadro mágico de Gartner, posibilita la gestión de proyectos, permitiendo la creación del Product Backlog y gestión del mismo dentro del Sprint.
- Pivotal Tracker. Ofrece la creación de historias de usuario para la realización del Product Backlog y planificación del Sprint.
- Trello. Ahora propiedad de Atlassian, “es una herramienta de colaboración y gestión de proyectos basada en la Web que se basa en listas tipo Kanban” (solid IT, 2020). Y tiene todo el potencial para ser empleado como visual story mapping.
- Team Foundation Server (TFS). Es propiedad de Microsoft y aunque no está incluido dentro de los productos estrella mencionados por Gartner, si tiene funcionalidades de creación del Product Backlog, y gestión del Sprint con la ayuda de un tablero Kanban, además del uso del burn down chart.

Además, hay otras herramientas que permiten la realización de técnicas como el User Story Map, las cuales son: Jira, Avion, Visual Paradigm, Miro, Cacao o “StoriesOnBoard, CardBoardIt, y FeatureMap” (Avendaño, 2017). FeatureMap es de resaltar ya que permite la realización del User o Visual Story Mapping como lo plantea el creador y autor de esta técnica, Jeff Patton en su libro “User Story Mapping: Discover the Whole Story, Build the Right Product” (Patton, 2014).

Además, otra técnica propuesta para este marco de trabajo es el uso del Impact Mapping creado por Gojko Adzic, algunos sitios web que lo ofrecen son: Miro, UXPressia y Creately.

De esta forma, con un panorama de herramientas que ofrecen acompañamiento en los proyectos que emplean Scrum, es imprescindible contrastar con lo que brinda el presente proyecto con el prototipo de software. La propuesta parte de la innovación al empezar desde la creación del Impact Mapping, y con los entregables del mismo pasar a la creación del Product Backlog no con una pila plana del producto sino con el User Story Mapping en una versión más sencilla a lo planteado por Jeff Patton y empleada por FeatureMap, con sus respectivas historias de usuario, que más allá de título y descripción como otras aplicaciones, se busca dar guía indicando “yo como, quiero, para...”.

Así, con el Backlog se pasa a la planeación del Sprint para la creación del Sprint Backlog en forma de tablero Kanban y del burn down chart, ofrecido por Jira y TFS. Además del módulo de Sprint Retrospective, con un tablero diferente al de Targetprocess y enfocado a las dinámicas: del barco de vela y la de la estrella de mar.

Como se puede ver, aunque hay bastantes softwares que ofrecen artefactos para el uso de Scrum, algunos muy avanzados como Jira, el proyecto no pretende ofrecer una herramienta de uso prolongado, más bien busca brindar una propuesta de solución de uso básico que permita dar guía en la adopción de los diferentes roles, artefactos y eventos, funcionando como anticipo al gran abanico de herramientas especializadas en Scrum para desarrollo de software.

4.2. Marco Conceptual

Las metodologías y marcos de desarrollo de software ágil surgieron alrededor de la década de los ochenta y noventa como alternativa de solución a la crisis del software, por lo que “buscaban reducir la probabilidad de fracaso por subestimación de costos, tiempos y funcionalidades en los proyectos de desarrollo de software” (Navarro Cadavid et al., 2013). En este contexto, es necesario analizar ciertos conceptos relacionados al agilismo y al foco central de esta de investigación: Scrum, los cuales se analizan a continuación.

4.2.1. Metodologías Tradicionales y Ágiles

Las metodologías ágiles surgen como reacción a las falencias de las metodologías tradicionales de desarrollo de software, estas últimas se caracterizan por un proceso rígido y secuencial que inicia en la fase de planeación en la cual se realiza el levantamiento de requisitos de una vez y para todo el proyecto. Por el contrario, el agilismo busca una mejora continua no solo del producto sino de los procesos por lo que es flexible y enfocado a la disciplina y organización del equipo; incluso a fin de tener un lenguaje en común se plantea el manifiesto ágil para el desarrollo de software, lo cual ayudara a metodologías como: Extreme Programming [XP], Dynamic System Development Method [DSDM], Crystal, Adaptative Software Development [ASD] y Feature-Driven Development [FDD] o marcos de trabajo como Scrum a tener un objetivo más claro centrado en la calidad según Navarro Cadavid et al. (2013). Otras diferencias entre estas metodologías se pueden ver en la Tabla 1.

Tabla 1

Metodologías tradicionales vs metodologías ágiles.

Metodologías Tradicionales	Metodologías Ágiles
Predictivos	Adaptativos
Orientados a procesos	Orientados a personas
Proceso rígido	Proceso flexible
Se concibe como un proyecto	Un proyecto es subdividido en varios proyectos más pequeños
Poca comunicación con el cliente	Comunicación constante con el cliente
Entrega de software al finalizar el desarrollo	Entregas constantes de software
Documentación extensa	Poca documentación

Fuente: Navarro Cadavid et al. (2013, pág. 31).

4.2.2. Manifiesto Ágil

Es un documento en el cual se acuerdan 4 valores y 12 principios para el desarrollo de software que se asienta en métodos ágiles. Fue creado en el año 2001 por 17 profesionales del software convocados por Kent Beck el creador de Extreme Programming (xP), durante la reunión “se acuñó el término “Métodos Ágiles” para definir a aquellos que estaban surgiendo como alternativa a las metodologías formales” (Menzinsky et al., 2019, pág. 9).

4.2.2.1. Valores del Manifiesto Ágil. Según Beck et al., (2001) se plantea, valoramos:

- A los individuos y su interacción, por encima de los procesos y las herramientas.
- El software que funciona, por encima de la documentación exhaustiva.
- La colaboración con el cliente, por encima de la negociación contractual.
- La respuesta al cambio, por encima del seguimiento de un plan.

4.2.2.2. Doce Principios del Manifiesto Ágil. Además de los 4 postulados anteriores,

Beck et al. (2001) también definen unos principios que se mencionan a continuación.

- 1) Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor.
- 2) Son bienvenidos los requisitos cambiantes, incluso si llegan tarde al desarrollo. Los procesos ágiles se doblan al cambio como ventaja competitiva para el cliente.
- 3) Entregar con frecuencia software que funcione, en periodos de un par de semanas hasta un par de meses, con preferencia en los periodos breves.
- 4) Las personas del negocio y los desarrolladores deben trabajar juntos de forma cotidiana a través del proyecto.
- 5) Construcción de proyectos en torno a individuos motivados, dándoles la oportunidad y el respaldo que necesitan y procurándoles confianza para que realicen la tarea.
- 6) La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
- 7) El software que funciona es la principal medida del progreso.
- 8) Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
- 9) La atención continua a la excelencia técnica enaltece la agilidad.
- 10) La simplicidad como arte de maximizar la cantidad de trabajo que no se hace, es esencial.
- 11) Las mejores arquitecturas, requisitos y diseños emergen de equipos que se autoorganizan.
- 12) En intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo y ajusta su conducta en consecuencia.

4.2.3. Scrum

Schwaber & Sutherland (2020) creadores de la Scrum Alliance y participantes de la creación del manifiesto ágil, definen que “Scrum es un marco de trabajo liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos”.

4.2.4. Origen de Scrum

A principios de la década de los ochenta Ikujiro Nonaka e Hirotaka Takeuchi identificaron y establecieron teóricamente a Scrum como una forma de trabajo centrada en los equipos por lo que debe su nombre a la formación de Scrum de los jugadores de Rugby.

Sin embargo, “Ken Schwaber y Jeff Sutherland copresentaron Scrum por primera vez en la Conferencia OOPSLA en 1995. Básicamente, documentó el aprendizaje que Ken y Jeff adquirieron en los años anteriores y publicó la primera definición formal de Scrum” (Schwaber & Sutherland, 2020).

4.2.5. Teoría de Scrum

Scrum se basa en el empirismo el cual “afirma que el conocimiento proviene de la experiencia y de la toma de decisiones con base en lo observado”, y en el pensamiento Lean el cual “reduce el desperdicio y se enfoca en lo esencial”. Además, Scrum plantea un enfoque “iterativo e incremental para optimizar la previsibilidad y controlar el riesgo” (Schwaber & Sutherland, 2020).

De esta forma, Scrum posee reglas que guían sus relaciones e interacciones como lo son: tres roles, tres artefactos y cuatro eventos formales los cuales “implementan los pilares empíricos

de Scrum de transparencia, inspección y adaptación” (Schwaber & Sutherland, 2020), y se busca promover cinco valores: Compromiso, Foco, Franqueza, Respeto y Coraje para promover la inteligencia colectiva. A continuación, se profundiza en las reglas y técnicas de Scrum.

4.2.6. Historia De Usuario

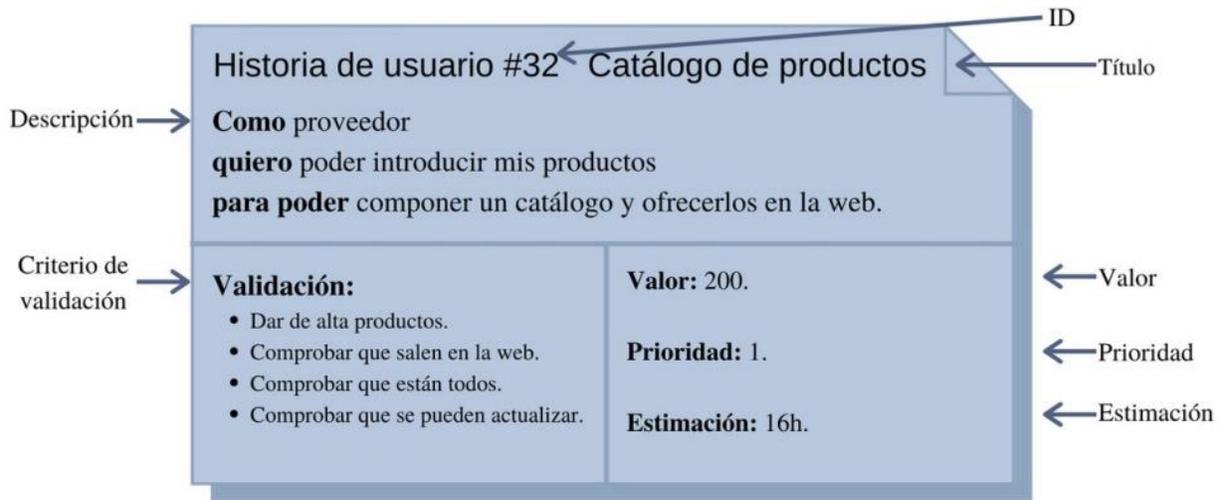
La historia de usuario (HU) es la conversación realizada con el cliente, que “describe la funcionalidad que será valiosa para un usuario o para el comprador de un sistema o software” (Cohn, 2004). Sin embargo, no existe un formato fijo para la historia de usuario, ya que es preferible no adoptar formatos rígidos, pero si es importante que cuente con estos 3 campos:

- Descripción: Síntesis de la historia de usuario. “Como [rol del usuario], quiero [objetivo], para poder [beneficio]” (Menzinsky et al., 2020).
- Estimación: Aproximación de esfuerzo necesario para implementar la historia de usuario, puede ser por tiempo o por puntos de historia. Donde “Los métodos de estimación habituales incluyen: Tamaño numérico (1 a 10), Tallas de camiseta (XS, S, M, L, XL, XXL, XXXL) o la secuencia de Fibonacci (0, 1, 2, 3, 5, 8, 13, 21, 34, etc.)” (International Scrum Institute, s/f).
- Prioridad: Orden de implementación de la historia de usuario.

Así mismo, puede contener otros campos opcionales como el id, titulo, valor de negocio y criterios de aceptación que son “pruebas de aceptación consensuadas con el cliente o usuario” (Menzinsky et al., 2020), como se ve en la Figura 3.

Figura 3

Ejemplo de una tarjeta de historia de usuario.



Nota: La Figura muestra un ejemplo del contenido que puede llevar una historia de usuario, mas no es que sea un formato fijo. Fuente: Menzinsky et al. (2020).

También, puede contener información relacionada a las tareas necesarias para cumplir con la HU, pues las tareas “resultan de la descomposición de las historias de usuario en unidades de trabajo adecuadas para gestionar y seguir el avance de su ejecución” y el desglose lo realiza el equipo de desarrollo (Menzinsky et al., 2020).

4.2.7. Roles de Scrum o Scrum Team

El Scrum Team se compone de un equipo pequeño multifuncional de personas que consta de un Product Owner, un Scrum Master y Developers.

- **Un Product Owner.** Es el responsable de maximizar el valor del producto y garantizar el retorno de inversión (ROI), de representar al cliente o interesados, y de la gestión del

Product Backlog, es una sola persona no un comité, aunque puede tener ayudantes las decisiones depende de él, según Schwaber & Sutherland (2020, pág. 6).

- **Un Scrum Máster.** Es el líder al “guiar a los miembros del equipo en ser autogestionados y multifuncionales” (Schwaber & Sutherland, 2020, pág. 6), por lo que se asegura de la mejora de las practicas al crear Incrementos, eliminar impedimentos, asegurarse de que todos los eventos de Scrum se lleven a cabo de la forma correcta. Por otro lado, le ayuda al Product Owner a gestionar el Product Backlog y les colabora a los interesados o stakeholders a comprender el marco de trabajo.
- **Development Team o Equipo de Desarrollo.** También conocidos como “Developers” son: “Las personas del Scrum Team que se comprometen a crear cualquier aspecto de un Increment utilizable en cada Sprint” (Schwaber & Sutherland, 2020, pág. 5). Tienen la responsabilidad de crear un plan del Sprint según el Product Backlog, y la solución de las tareas. “No se trata de un grupo de trabajo formado por un arquitecto, diseñador o analista, programadores y testers. Es un equipo multifuncional, en el que todos los miembros trabajan de forma solidaria con responsabilidad compartida” (Menzinsky et al., 2019). Aunque alguno de los miembros esté especializado en ciertas áreas se debe tener presente que la responsabilidad es de todos.

4.2.8. Eventos de Scrum

- **Sprint.** Es el nombre que recibe cada iteración del desarrollo, la cual inicia cuando termina la anterior. Durante el Sprint ocurren las actividades para resolver las historias de usuario. Comienza con un Sprint Planning y se finaliza con las reuniones de Sprint

Review y Sprint Retrospective, según International Scrum Institute (s/f). Incluso cada Sprint se puede considerar como si se tratara de un proyecto corto.

- **Reunión de Planificación o Sprint Planning.** Es la reunión en la cual se establece el trabajo a realizar en cada Sprint y marca el inicio del mismo, donde el Scrum Master se encarga de ser el moderador. Schwaber & Sutherland afirman que se abordan tres temas principalmente: ¿Por qué es valioso este Sprint?, ¿Qué se puede hacer en este Sprint?, y ¿Cómo se realizará el trabajo elegido? Como resultado se genera el Sprint Backlog, se fija la duración del Sprint, la fecha del Sprint Review y el objetivo del Sprint, como menciona Menzinsky et al. (2019).
- **Scrum Diario o Daily Scrum.** Es una reunión corta de aproximadamente 15 minutos para el equipo de desarrollo que tiene como propósito “inspeccionar el progreso hacia el Objetivo del Sprint y adaptar el Sprint Backlog según sea necesario, ajustando el trabajo planificado entrante” (Schwaber & Sutherland, 2020). Para ver el progreso se puede hacer uso del burn down el cual se actualiza en este mismo evento. El International Scrum Institute (s/f) menciona que cada miembro responde tres preguntas: ¿Qué actividades he realizado desde la última reunión diaria de Scrum? ¿Qué actividades planeo realizar hasta la próxima reunión diaria de Scrum? ¿Encontré o espero algún impedimento que pueda ralentizar o bloquear el progreso de mi trabajo?
- **Revisión del Sprint o Sprint Review.** En esta reunión se examina el resultado del Sprint y se determinan futuras adaptaciones de acuerdo con el progreso. En esta sesión de trabajo también se puede ajustar el Product Backlog con una duración máxima de cuatro horas si el Sprint duro un mes, de acuerdo con Schwaber & Sutherland (2020, pág. 10).

- **Retrospectiva del Sprint o Sprint Retrospective.** Se planifica las formas de aumentar la calidad y la efectividad, para ello se examina el último Sprint donde el Scrum Team “analiza qué salió bien durante el Sprint, qué problemas encontró y cómo se resolvieron (o no) esos problemas” (Schwaber & Sutherland, 2020, pág. 10). Y de esta forma se identifican los cambios a realizar, y así concluye el Sprint.

4.2.9. Artefactos de Scrum

- **Product Backlog.** Es una lista ordenada de las historias de usuario, el cual puede crecer y evolucionar a lo largo del desarrollo. Lo realiza el Product Owner mostrando el punto de vista del cliente como dice Menzinsky et al. (2019).
- **Sprint Backlog.** Según Schwaber & Sutherland (2020) se realiza seleccionando elementos del Product Backlog y se puede actualizar a lo largo del Sprint. Refleja la visión de los desarrolladores, que está “formada por la lista de tareas en las que se descomponen las historias de usuario que se van a llevar a cabo en el Sprint” (Menzinsky et al., 2019).
- **Incremento o Increment.** “Es la parte de producto realizada en un Sprint potencialmente entregable: terminada y probada” (Menzinsky et al., 2019). Aunque se pueden entregar varios Increments dentro de un Sprint.

4.2.10. Técnicas de Scrum

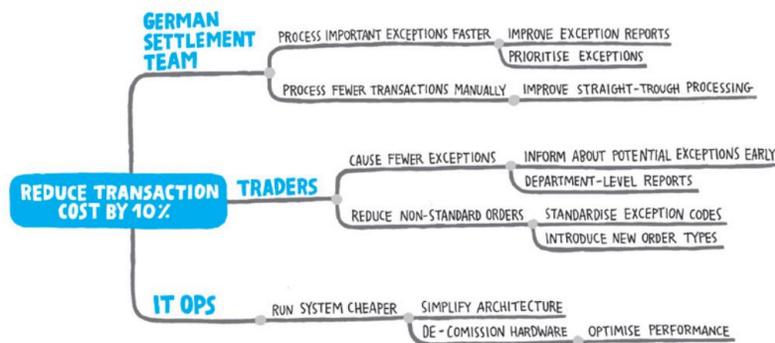
A lo largo del tiempo, diferentes autores han planteado nuevas técnicas como refuerzo de los artefactos que plantea el Scrum técnico, alguno de ellos es el Impact Mapping y el User Story Mapping. Por otro lado, como apoyo al proceso de definición de avances se hace uso de un tablero de estados.

4.2.10.1. Impact Mapping. Es una visualización del alcance del proyecto, que se crea en colaboración del personal técnico y de negocio, Fue creado por Gojko Adzic. “Es un mapa mental desarrollado durante una discusión facilitada” (Adzic, 2012), que busca responder a cuatro preguntas: ¿Por qué? ¿Quién? ¿Cómo? ¿Qué? Los cuales darán como respuesta: Meta, Actores, Impactos y Entregables. Un ejemplo de este mapa se puede visualizar en la Figura 4 y 5.

- Meta. Es el objetivo del proyecto.
- Actores. Se incluyen a los impactados directamente: actores primarios y secundarios como los proveedores de servicios; y como terceros a los impactados indirectamente.
- Impactos. Se definen de acuerdo con las actividades de negocio.
- Entregables. Se expresa que se va a hacer para generar el impacto, una serie de funcionalidades definidas de alto nivel.

Figura 4

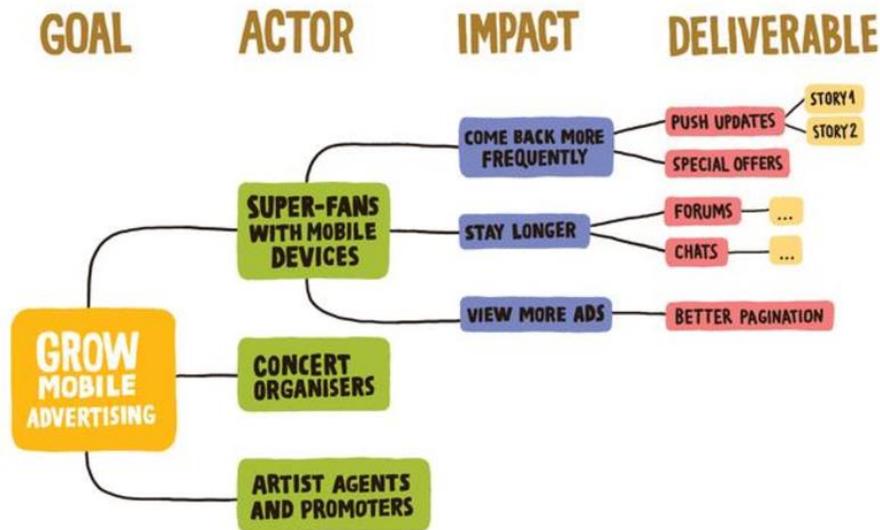
Ejemplo de Impact Mapping.



Nota: Ejemplo de Impact Mapping de procesamiento de transacciones. Fuente: Adzic (2012).

Figura 5

Ejemplo de los componentes del Impact Mapping.

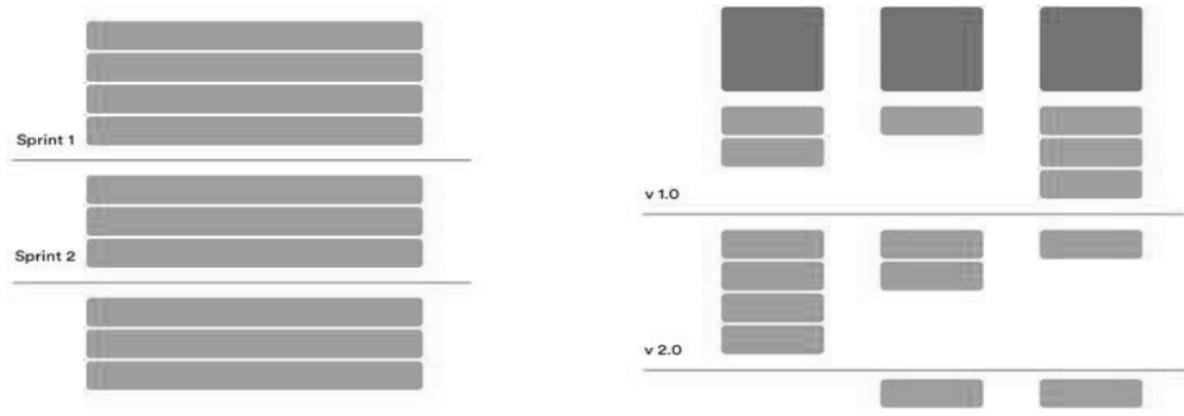


Fuente: Meldre (2016).

4.2.10.2. User Story Mapping o Visual Story Mapping. Es una técnica definida por Jeff Patton, como alternativa al Product Backlog común, pues consiste en “construir una pila de producto que vaya más allá de una lista unidimensional de historias de usuario” (Menzinsky et al., 2020), como se puede ver en la Figura 6, lo suficiente para que escribiera un artículo titulado “El nuevo backlog de historias de usuario es un mapa” (Patton, s/f). Donde se busca realizar un mapa que se compone de actividades, que juntas conforman el “Backbone”, y sus respectivas historias de usuario, que se organizan de la forma en que se ve en la Figura 7. Por otro lado, en vez de gestionarse de forma vertical se realiza horizontalmente, como lo indica la Figura 8.

Figura 6

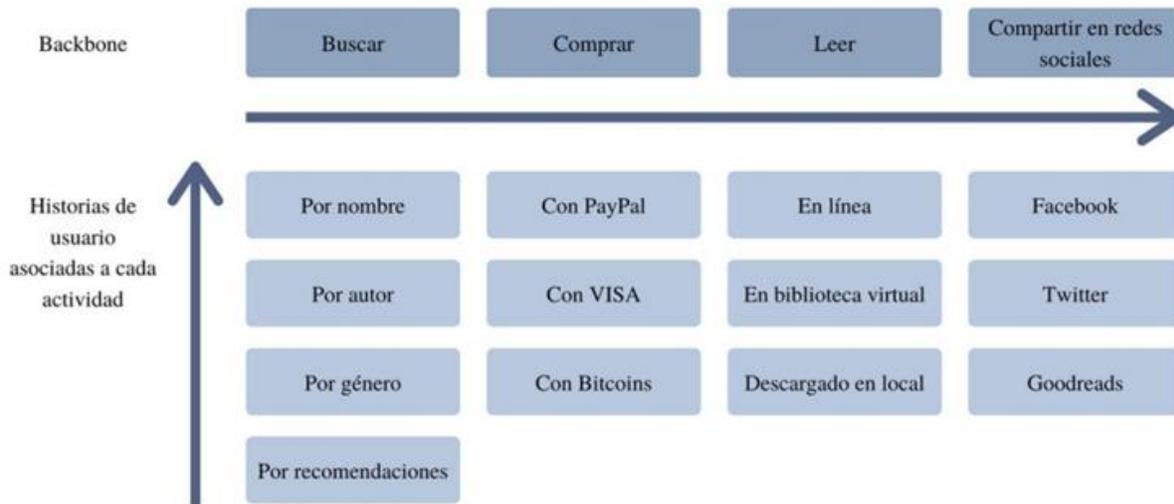
Product Backlog plano y Product Backlog construido con la técnica User Story Mapping.



Fuente: Menzinsky et al. (2020).

Figura 7

Esquema del User Story Mapping.

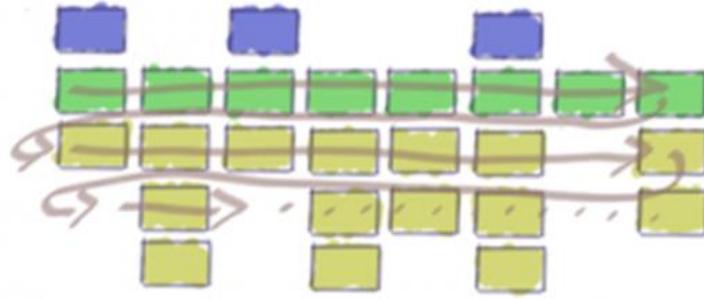


Nota: La figura es un ejemplo del esquema que debe tener User Story Mapping. Fuente:

Menzinsky et al. (2020).

Figura 8

Gestión del User Story Mapping.



Nota: La figura explica gráficamente la forma en que se debe leer la prioridad en el User Story Mapping, es decir horizontalmente. Fuente: *Patton (s/f)*.

4.2.10.3. Técnica MoSCoW. Es empleada para definir la priorización de las historias de usuario, fue definida en el año 2004 “por Dai Clegg de Oracle UK Consulting en el libro Case Method FastTrack: A RAD Approach. Su finalidad es obtener un entendimiento común entre cliente y el equipo del proyecto, (...) sobre la importancia de cada historia de usuario” (Menzinsky et al., 2020). Donde la clasificación es la siguiente.

- M – Must Have. Aquello que debe tener, es decir es la funcionalidad que es necesaria, y de prioridad alta.
- S – Should Have. Lo que debería tener y es recomendable, es decir se relaciona con una prioridad media.
- C – Could Have. Lo que podría tener y sería conveniente, pero depende de los tiempos y presupuesto del proyecto, por lo que se cataloga como una prioridad baja.

- W – Won't Have. Lo que no tendrá, es decir es una prioridad descartada por el momento y que puede ser relevante en un futuro.

4.2.10.4. Tablero de Estados o Tablero Kanban. El tablero de estados o también llamado Tablero de Scrum se basa en el famoso Tablero Kanban que busca especificar 3 estados: Pendiente, en curso y hecho. Como se ve en la Figura 9 es ideal para que el equipo identifique su progreso del Sprint Backlog; solo se encontrara la historia de usuario en terminado si las tareas ya están finalizadas a cabalidad.

Figura 9

Ejemplo de Sprint Backlog con Tablero Kanban.



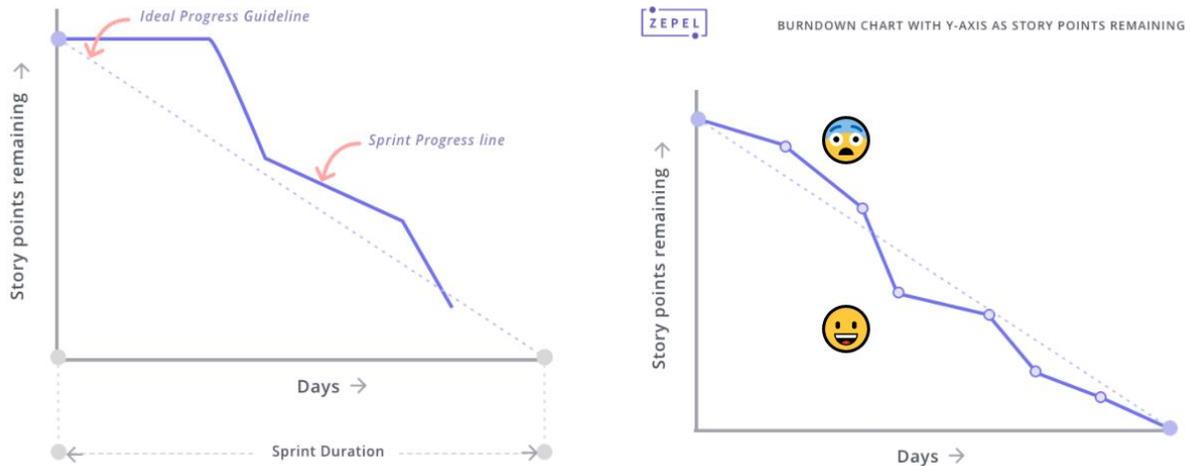
Nota: La Figura muestra un ejemplo del tablero Kanban, el cual se puede usar para complementar el Sprint Backlog. Fuente: Menzinsky et al. (2019).

4.2.10.5. Burn Down Chart. Es un gráfico de avance, que indica el trabajo pendiente y la velocidad en la cual se terminan las historias de usuario. “Los desarrolladores lo actualizan, durante el Sprint, a ser posible a diario, para monitorizar el ritmo de avance” (Palacio, 2020). Donde en el eje Y representa los puntos de estimación de las historias de usuarios que aún faltan

por realizar, y el eje X representa los días del Sprint. Como en la Figura 10 el avance ideal es la línea recta del gráfico, y el progreso real se construye a medida que pasan los días y se termina el trabajo. Si la línea de avance se mantiene arriba de la línea ideal durante varios días significa que se requerirá de más tiempo, de lo contrario, si se mantiene por debajo implica que se terminara el Sprint a tiempo o antes de lo previsto.

Figura 10

Ejemplo de la estructura de un burn down chart.



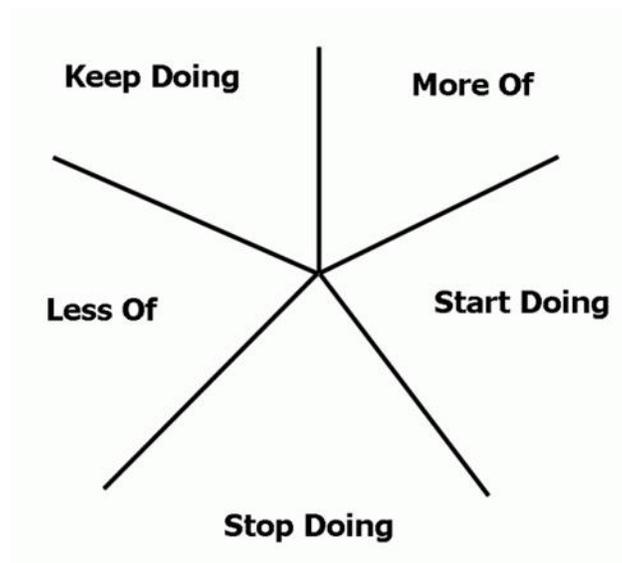
Fuente: Adaptación de Jaikrishnan (2018).

4.2.10.6. Dinámica Del Barco De Vela Y La Estrella. La retrospectiva del Sprint es un evento importante para poder ver el avance y las mejoras a realizar dentro del proyecto y se busca la mejora continua, pero en este proceso se puede hacer uso de dos dinámicas: la estrella de mar y el barco de vela.

- Estrella de mar. Fue creada por Patrick Kua y consiste en crear diagramas en forma de estrella de mar como en Figura 11, que busca tratar 5 temas: Continuar haciéndolo, menos de, más de, dejarlo de hacer y comenzar a hacerlo como dice Garzas (2013).

Figura 11

Estructura de componentes de la dinámica "Estrella de mar".



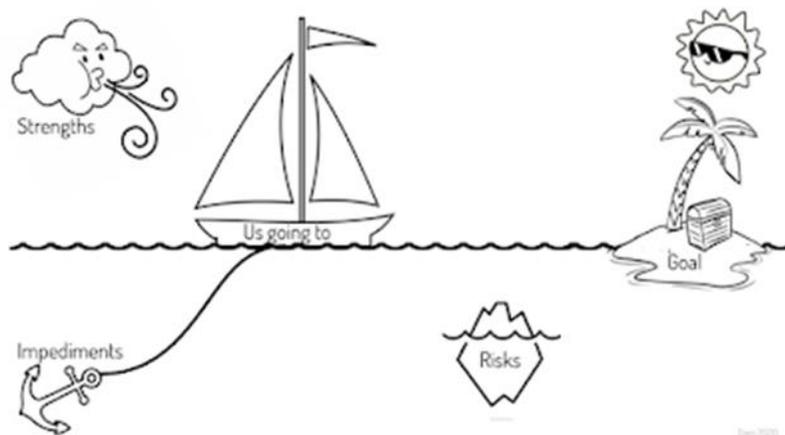
Nota: Keep doing (Sigue haciendo), More Of (Más de), Start Doing (Empezar a hacer), Stop Doing (Dejar de hacer), Less Of (Menos de). Fuente: Kua (2006).

- Barco de Vela. La historia consiste en que el equipo viaja en el barco hacia la tierra prometida. En la escena se dibuja un barco de vela y se identifica: un sol para indicar lo positivo que es la motivación para seguir adelante, un ancla saliendo del barco hacia el suelo con las cosas que frenan, rocas para indicar los riesgos, viento que marca las cosas buenas que empujan y acercan al equipo a la tierra prometida, un ejemplo de ello se ve en

la Figura 12. Y según Garzas (2015), tras la identificación se procede a generar el plan de cambio.

Figura 12

Ejemplo de los componentes de la dinámica "Barco de vela".



Nota: Ancla (Debilidades), Vela (Fortalezas), Rocas (Obstáculos), Sol (Agradecer), Isla (Meta).

Fuente: Palminio (2018).

4.2.11. Desarrollo de Páginas Web

El desarrollo Web es un término empleado para definir la creación de sitios web para internet o intranet, para ello se necesita de lenguajes de programación dirigidos a Web que permitan el desarrollo de este tipo de páginas o aplicativos. Por otro lado, además del lenguaje es necesario aplicar buenas prácticas y una de ellas es Atomic Design.

4.2.12. Angular

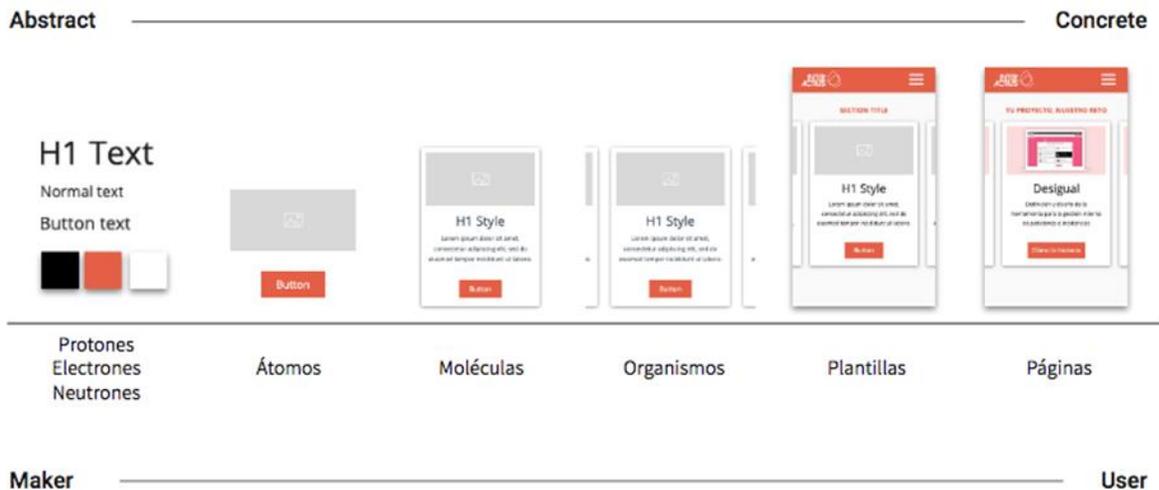
Es un framework del Lenguaje de programación: JavaScript, el cual permite la creación de “aplicaciones Web, móvil web, móvil nativo o de escritorio” (Google, 2020).

4.2.13. Diseño Atómico o Atomic Design

Brad Frost (2016), su creador, lo define como una metodología para crear sistemas de diseño de interfaces, que se basa en el desarrollo de componentes basándose en estructuras químicas como: átomos, moléculas, organismos; y lo complementa con plantillas y páginas. Un ejemplo se puede ver en la Figura 13.

Figura 13

Estructura del Diseño Atómico o Atomic Design.



Nota: La figura muestra la evolución desde los átomos hasta las páginas. Fuente:

<https://blog.interactius.com/atomic-design-para-dummies-5562930ce89c>

5. Desarrollo

5.1. Materiales y Métodos

Para realizar la aplicación Web se utilizó en un inicio un computador Asus X555Q de 8 Gb de memoria RAM, 1Tb de Disco duro, Procesador AMD Quad Core A10-9600P y sistema operativo (SO) Windows 10, sin embargo debido a que el procesador se ralentiza con la compilación de los archivos del proyecto, también se hizo uso de un computador Asus K43E de 6Gb de memoria RAM, 750 Gb de Disco duro, Procesador Intel Core i5, y sistema operativo Ubuntu con Deepin.

Dentro de los equipos se instaló un navegador Web que ejecuta JavaScript como Google Chrome, Visual Studio Code como editor de código fuente, Git como software de control de versiones, el entorno de ejecución Node.js v12, Angular CLI v11 que es un framework de código abierto desarrollado en TypeScript y que a su vez es un superconjunto de JavaScript, Angular Material como librería de componentes al igual que Chart.js y ng2-charts. Y en la Web se hace uso como base de datos no relacional: Firebase, y como plataforma de autenticación Auth0.

Así mismo, a nivel de método de desarrollo Front-End se hizo uso del Atomic Design o Diseño Atómico, pues su propuesta de estructura de componentes en el diseño de interfaces se ajustó a los elementos que brinda Angular.

Además, a nivel de comunicación interna se hizo uso de Microsoft Teams, WhatsApp, y el correo institucional. Para la gestión documental se creó una carpeta en One Drive, y para gestión de las actividades del proyecto se usa Jira Software y GitHub para control de versiones en la nube del desarrollo de software.

5.2. Metodología y Proceso Metodológico del Proyecto

El proyecto hace uso del marco de trabajo Scrum en lo relacionado con el desarrollo de software, desde el análisis, diseño, codificación y pruebas. Debido a que ayuda a generar valor a los equipos y organizaciones a través de soluciones adaptativas para problemas complejos, que se basa en el empirismo y el pensamiento Lean, promoviendo el aprendizaje continuo.

Así mismo, Scrum emplea un enfoque iterativo e incremental, pero no contiene fases ya que se compone de una serie de Sprints donde cada miembro del equipo es multifuncional y debe garantizar que se cumpla el ciclo de vida del software inherentemente, asegurándose de la planeación como proceso continuo, el diseño, el desarrollo y las pruebas, incluyendo la realización de la documentación respectiva sin ser extenuante. Se compone de 3 roles: Product Owner, Scrum Master y equipo de desarrollo; 3 artefactos: Product Backlog, Sprint Backlog, e Incremento; y 5 eventos, 4 eventos formales: Sprint Planning, Daily, Sprint Review y Sprint Retrospective, y 1 evento contenedor: El Sprint; estos eventos funcionan ya que se soportan bajo 3 pilares empíricos: la transparencia, la inspección, y la adaptación.

De esta manera, se define que el proyecto se compone de las siguientes fases a mencionar, sin embargo, todo lo que implica la ejecución del proyecto relacionado al desarrollo del aplicativo Web se realiza bajo el uso de Sprints de acuerdo con el marco de trabajo Scrum.

5.2.1. Fases del Proyecto

- Fase 1: Documentación.
 - Estudio de los roles, artefactos y eventos de Scrum.
 - Estudio de algunas técnicas de Scrum.
- Fase 2: Construcción del prototipo.

- Identificación de los módulos del proyecto.
 - Diseño de las características de los módulos.
- Fase 3: Desarrollo del prototipo.
 - Definición del producto: Creación del Impact Mapping y Product Backlog.
 - Desarrollo y Pruebas del prototipo (Sprints).
- Fase 4: Realizar pruebas de validación.
 - Selección de la muestra.
 - Aplicación de la prueba.
 - Tabular pruebas.
- Fase 5: Análisis de la información de la validación.
 - Crear diagramas.
 - Análisis.
- Fase 6: Divulgación de resultados.
 - Realización del documento.

Para la Fase 1 el resultado se evidencia en el Marco Referencial del proyecto, para las siguientes fases el proceso realizado se evidencia en secciones más adelante.

5.3. Planeación y Administración del Proyecto.

5.3.1. Cronograma de Actividades

El cronograma se puede visualizar en la Tabla 2. Y debido a que el proyecto se está realizando de forma individual, todas las actividades son responsabilidad de la autora.

Tabla 2*Cronograma de actividades.*

Fase	Actividad	Tiempo en Ejecución									
		Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	
Documentación	Estudio de los roles, artefactos y eventos de Scrum	■	■								
	Estudio de algunas técnicas de Scrum										
Construcción del prototipo	Identificación de los módulos del proyecto		■								
	Diseño de las características de los módulos		■								
Desarrollo del prototipo	Definición del producto: Creación del Impact Mapping y Product Backlog		■	■							
	Desarrollo y Pruebas del prototipo			■	■	■	■	■	■		
Realizar pruebas de validación	Selección de la muestra									■	
	Aplicación de la prueba									■	
	Tabular pruebas									■	
Análisis de la información de la validación	Crear diagramas										■
	Análisis										■
Divulgación de resultados	Realización del documento	■	■	■	■	■	■	■	■	■	■

Fuente: Elaboración propia.

5.3.2. Recursos y Fuentes Financieras

Los recursos y costos empleados durante el proyecto se pueden visualizar en la Tabla 3.

Donde las fuentes de financiamiento provienen de la autora.

Tabla 3

Recursos y fuentes de financiamiento.

Área	Recurso	Costo (\$)	
Materiales	Computador Asus X555Q	1 300 000	
	Préstamo computador Asus K43E	0	
	Visual Studio Code	0	
	Google Chrome	0	
	Cuenta en Firebase, Auth0, GitHub, Jira Software	0	
	Correo institucional y Teams	0	
	Servicio de Internet	630 000	
Recurso Humano	Product Owner	0	
	Scrum Master	0	
		Analista	0
	Developer Team	UED	0
		Programador	0
	Tester	0	
Bienes	Material Bibliográfico	0	
	Material de escritorio	100 000	
Servicios	Energía	500 000	
	Total	2 530 000	

Fuente: Elaboración propia.

5.4. Construcción del Prototipo

El propósito del prototipo es permitir y guiar a los equipos de trabajo en la adopción correcta de Scrum en sus proyectos, con el uso de técnicas que han tenido una gran acogida en el agilismo y que dan soporte a los distintos roles, artefactos y eventos del marco de trabajo.

5.4.1. Fundamentos para la Creación del Prototipo

En la construcción del prototipo se tuvieron en cuenta algunas técnicas ágiles que son empleadas por equipos que usan Scrum, sus componentes y creadores se pueden visualizar en la Tabla 4.

Tabla 4

Técnicas conceptuales del prototipo.

Técnica	Creador	Componentes
Impact Mapping.	Gojko Adzic (Alrededor del 2010).	<ul style="list-style-type: none"> • Meta. • Actores. • Impactos. • Entregables.
User Story Mapping.	Jeff Patton (Principios de la década del 2000).	<ul style="list-style-type: none"> • Backbone o Actividades. • Tareas. • Historias de Usuario.
Priorización MoSCoW.	Dai Clegg de Oracle UK Consulting (2004).	<ul style="list-style-type: none"> • Must Have (Alta). • Should Have (Media). • Could Have (Baja). • Won't Have (Nula o Descartada).
Tablero Kanban.	David J. Anderson (2004), con base al trabajo de Taiichi Ohn	<ul style="list-style-type: none"> • To Do (Por Hacer). • Doing (En curso). • Done (Hecho).

Técnica	Creador	Componentes
Burn Down Chart.	Ken Schwaber	<ul style="list-style-type: none"> • Eje X: Puntos por historias. • Eje Y: Días del Sprint. • Línea de tareas ideales realizadas. • Línea de tareas reales realizadas.
Dinámica para Retrospectiva: Estrella de Mar.	Patrick Kua (Alrededor del 2006).	<ul style="list-style-type: none"> • Más de. • Empezar a hacer. • Dejar de hacerlo. • Menos de. • Sigue haciendo.
Dinámica para Retrospectiva: Barco de Vela.	Luke Hohmann (2006).	<ul style="list-style-type: none"> • Debilidades - Ancla. • Fortalezas – Vela o viento. • Obstáculos o riesgos - Rocas. • Cosas por agradecer - Sol. • Meta – Isla.

Fuente: Elaboración propia.

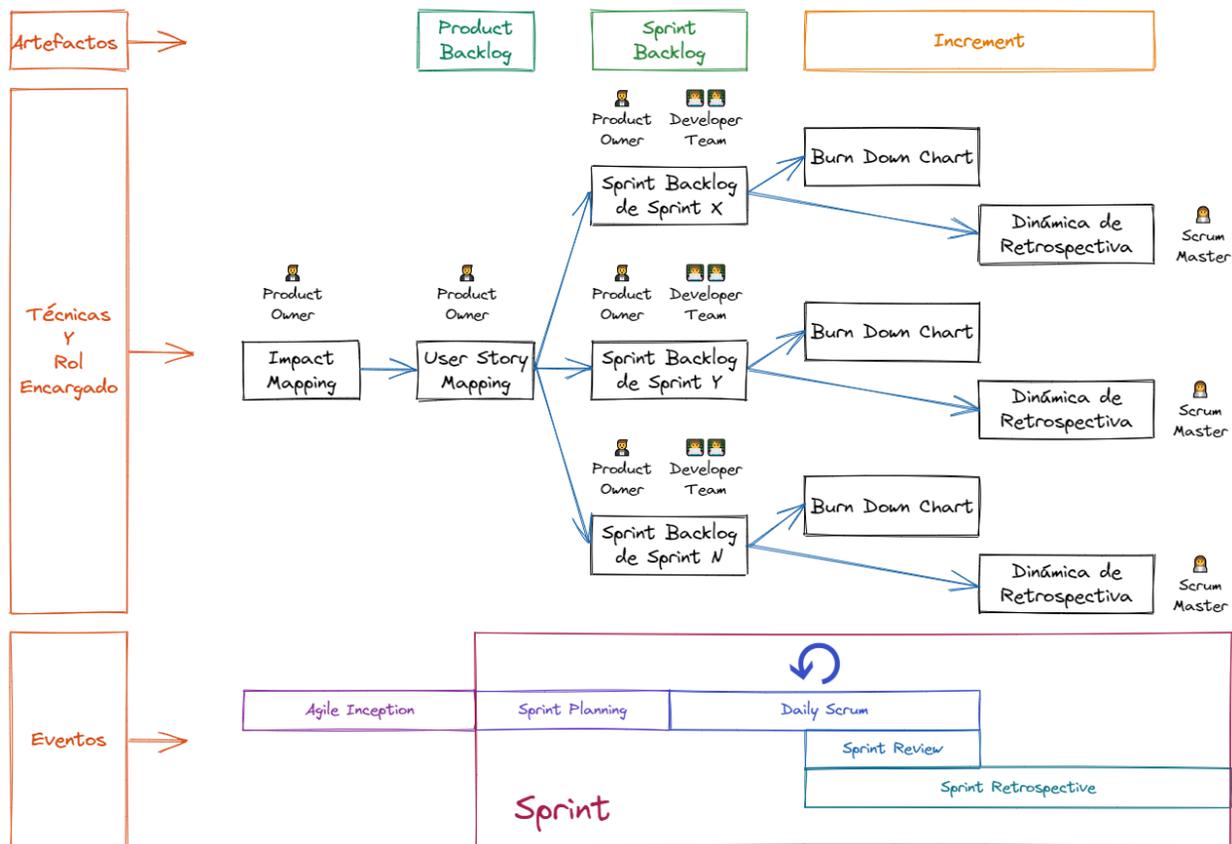
5.4.2. *Prototipo Propuesto*

En la Figura 14 se presenta la construcción del prototipo del cual se basa el aplicativo Web, pues relaciona los diferentes artefactos y eventos de Scrum con sus respectivas técnicas ágiles de forma vertical, e identifica el encargado de gestionar dicha técnica, donde cada miembro del equipo puede visualizar de forma transparente el trabajo de sus compañeros, pero así mismo tiene en algún momento el deber de administrar ciertos módulos. Donde se inicia con el Impact Mapping, se pasa al User Story Mapping, y de ahí en adelante existen N Sprint

Backlogs con sus respectivos burn down chart y dinámicas de retrospectiva, según la que haya sido seleccionada: Estrella de Mar o Barco de Vela.

Figura 14

Prototipo de integración conceptual de los roles, artefactos, técnicas y eventos de Scrum.



Fuente: Elaboración propia con Excalidraw.

Así mismo, cada técnica tiene una forma de encadenarse a la siguiente, como se ve en la Figura 15. Después de identificar la meta o el objetivo del proyecto, sus respectivos actores e impactos con sus correspondientes entregables, estos últimos se convierten en las actividades o el Backbone del User Story Mapping, creándose una columna para cada una de ellas.

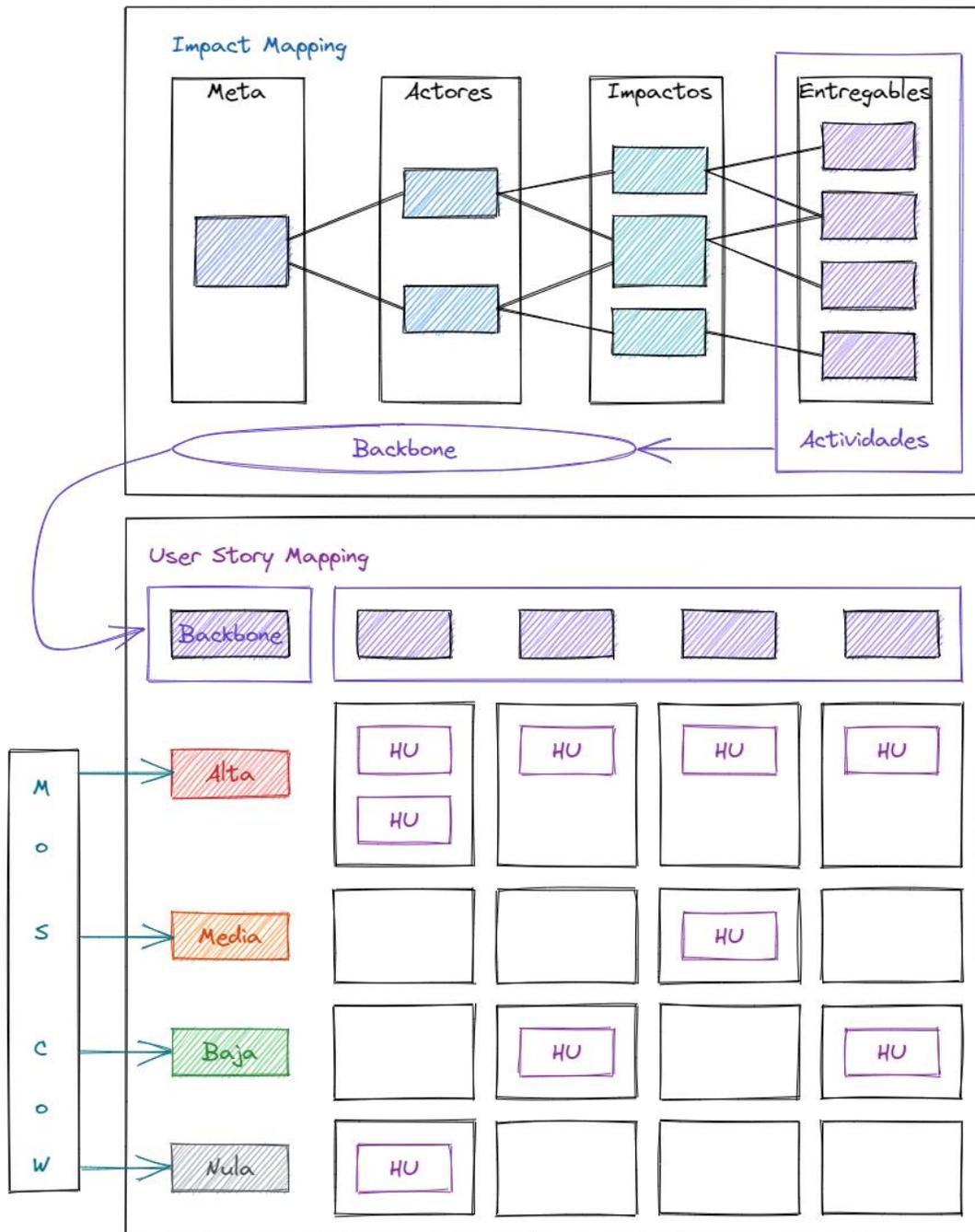
Aunque generalmente las divisiones en filas del User Story Mapping son para hacer referencias a Sprints o Releases o Versiones, en este caso se emplean por 4 prioridades según la técnica de Priorización de historias de usuario MoSCoW, ya que este software es ideal para usar por un corto tiempo hasta entender el marco de trabajo y ahí si migrar a otras plataformas, por lo que se cree que podría ser empleado por como máximo 3 Sprint aunque si el equipo lo desea seguir utilizando no hay ningún problema. Esto se da con el objetivo de que el equipo entienda que no todo es importante y que de ahí la importancia de priorizar y lograr distinguir lo urgente.

Después, como se ve en la Figura 16, al planificar un Sprint se deben elegir las historias de usuario a resolver, las cuales se ubican posteriormente en un Sprint Backlog en forma de Tablero Kanban inicialmente en la primera lista de Por Hacer, y son trasladadas a la lista de En Curso por el equipo de desarrollo o Developer Team, finalmente pasan a Hecho si es aprobada por el Product Owner.

El progreso en el tablero Kanban se evidencia en el burn down chart, como lo muestra la Figura 17. Donde se plantea un ejemplo de 4 historias de usuario con sus respectivos puntos de estimación que suman en total 13 puntos, los cuales se ubican en eje X del burn down chart, en contraste al número de días que tiene el Sprint en el eje Y, para finalmente tener una línea recta de progreso ideal y una línea de progreso real la cual disminuye a medida que llegan a Hecho las historias de usuario en el Sprint Backlog, según el día en que haya ocurrido. Siendo optimo estar por debajo del progreso ideal, pero preocupante si se está por encima.

Figura 15

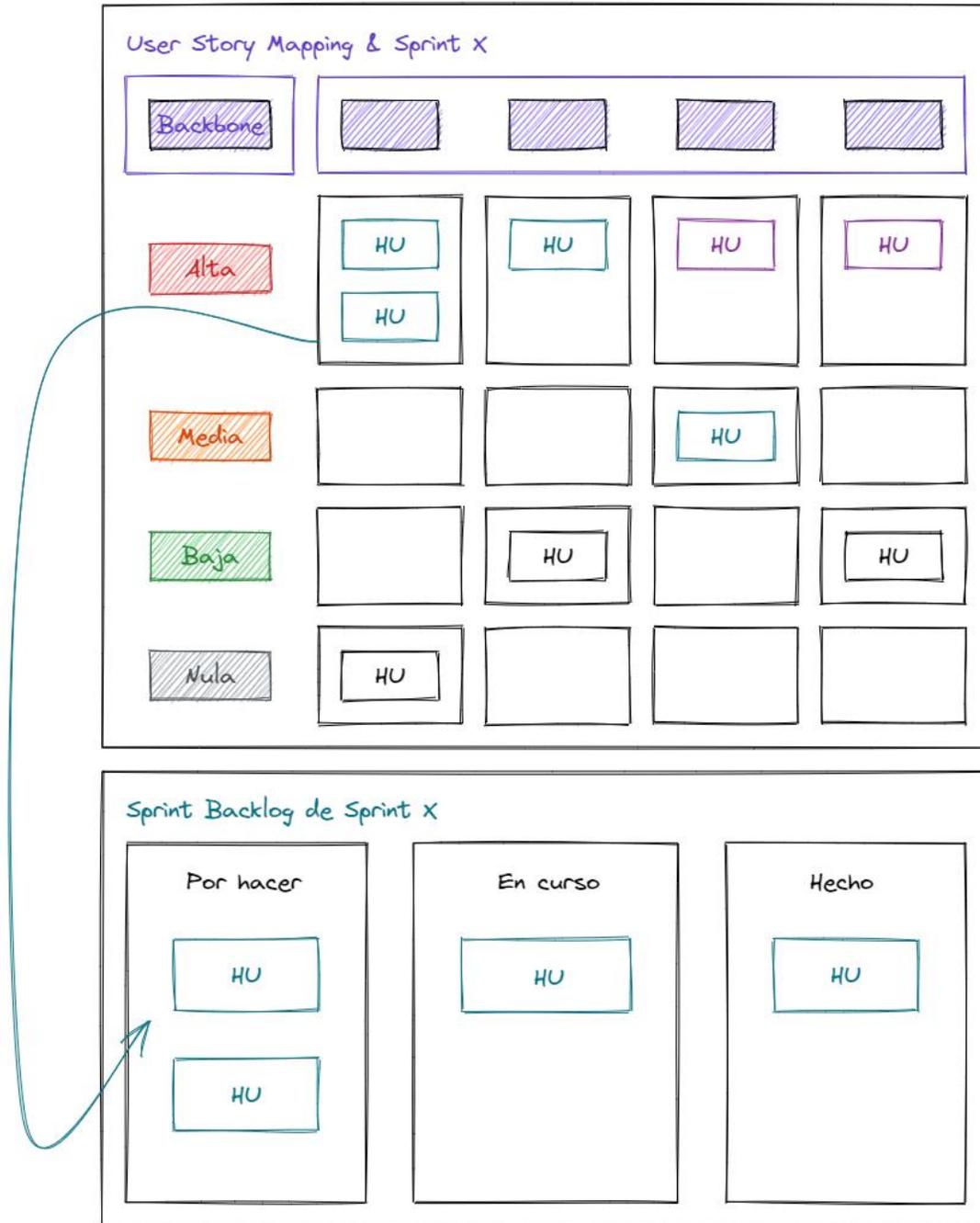
Relación de Impact Mapping con el User Story Mapping y priorización MoSCoW.



Fuente: Elaboración propia con Excalidraw.

Figura 16

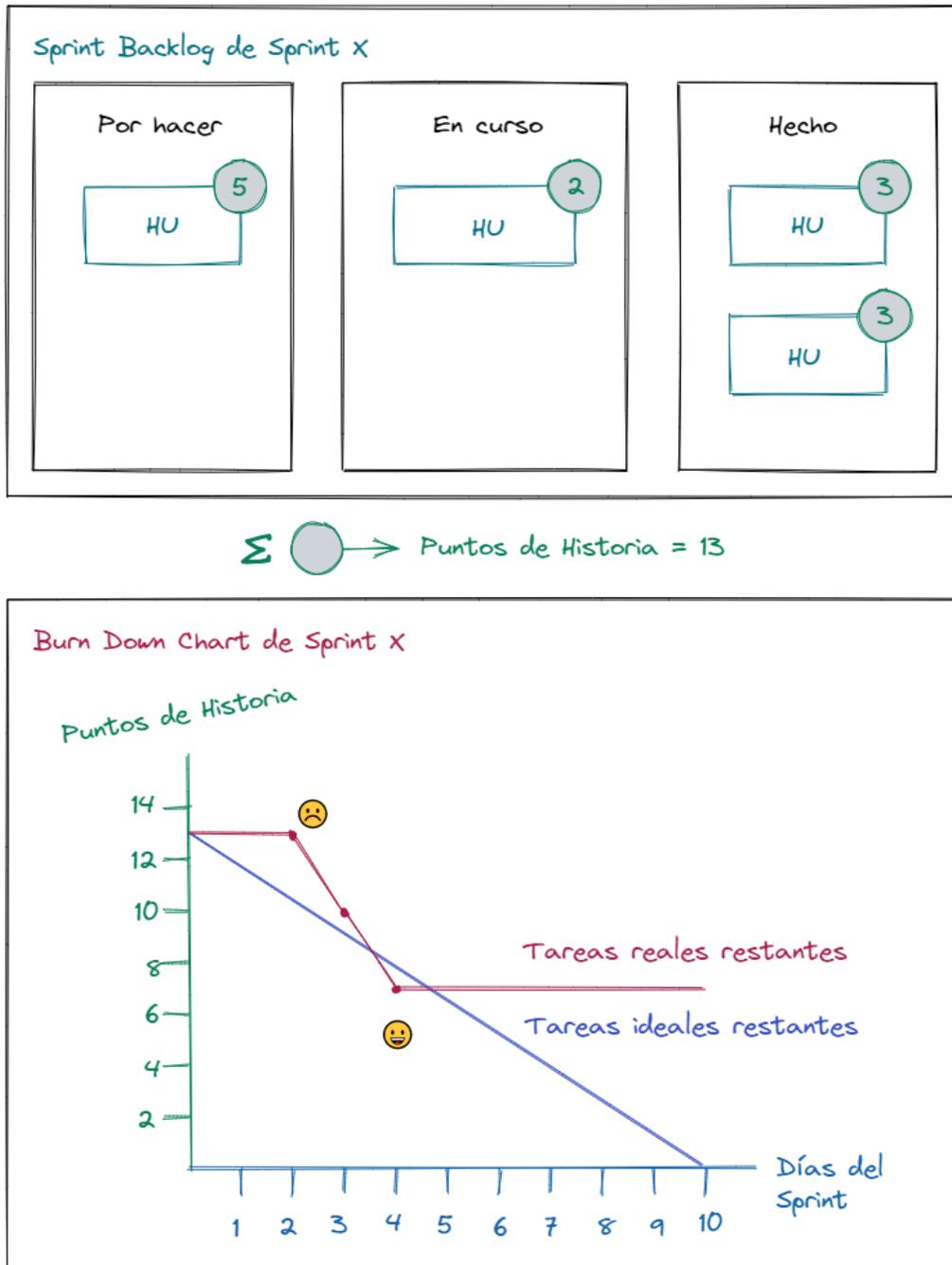
Relación del User Story Mapping con el Sprint Backlog.



Fuente: Elaboración propia con Excalidraw.

Figura 17

Relación del Sprint Backlog con el burn down chart.

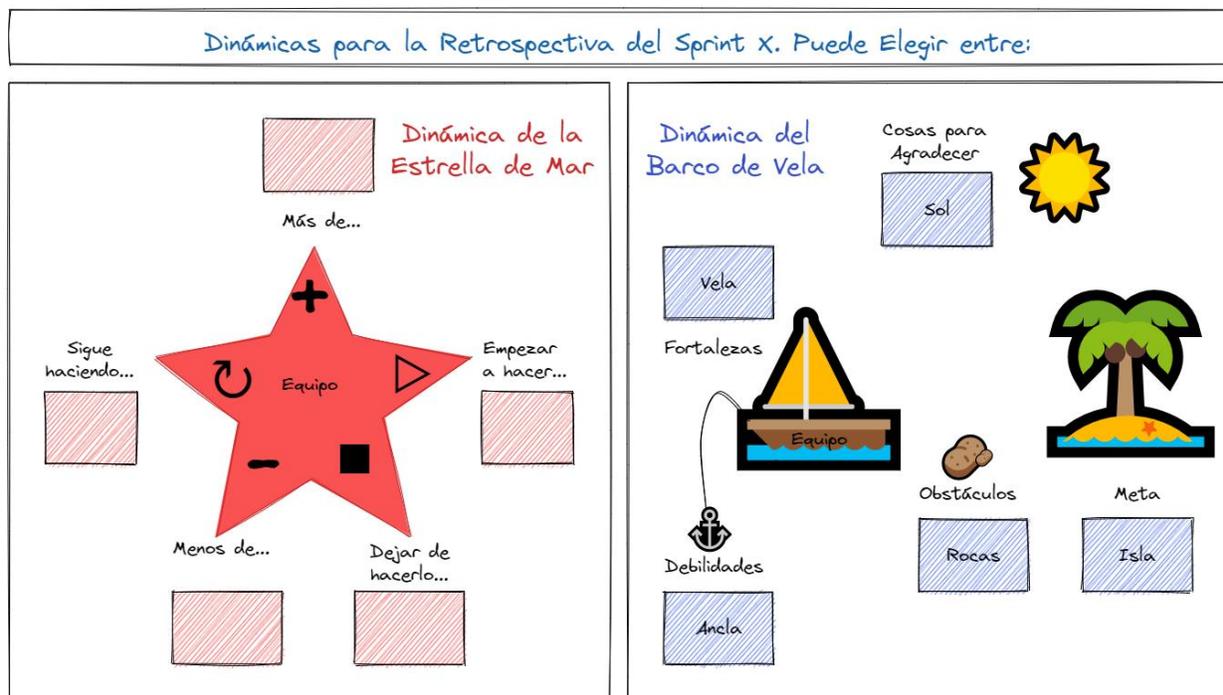


Fuente: Elaboración propia con Excalidraw.

Por último, se plantea que el Sprint finaliza con el evento de la retrospectiva, para el cual se disponen de dos dinámicas a seleccionar como se ve en la Figura 18, donde en el caso de la estrella de mar se busca que las personas no solo conozcan que salió bien o mal sino las acciones a realizar; por otro lado si el equipo elige la Dinámica del barco de vela podrá reflexionar sobre los problemas que ha habido en el Sprint pero así mismo sobre las oportunidades que se presentaron.

Figura 18

Relación del Sprint con las dinámicas para retrospectivas.



Nota: En la Figura se puede ver dos opciones para elaborar una retrospectiva, puede ser con la dinámica de la estrella de mar o con la dinámica del barco de vela. Fuente: Elaboración propia con Excalidraw.

5.5. Desarrollo del Prototipo

Para el desarrollo se siguió el marco Scrum, el cual incluye todas las actividades del desarrollo de software en un Sprint generándose un incremento en cada iteración, esto se debe a que el equipo de trabajo y en si cada miembro es multifuncional. Siendo estas las actividades:

- Recolección de necesidades en forma de historias de usuario.
- Análisis y priorización de las historias de usuario, a través del Product y Sprint Backlog.
- Diseño del software.
- Codificación y Pruebas del desarrollador.

A continuación se explican los roles y procesos que se llevaron a cabo.

5.5.1. Roles

El Scrum Team se compone de 3 roles: 1 Product Owner, 1 Scrum Master y un equipo de desarrollo o Developer Team, sus respectivas funciones se evidencian en la Tabla 5 y 6.

Tabla 5

Roles del Scrum Team del proyecto.

Roles	Función
Product Owner (PO)	Es una persona no un comité que se encarga de maximizar el valor del producto y de gestionar el Product Backlog. Representa a todas las partes interesadas.
Scrum Master (SM)	Persona responsable de establecer Scrum ayudando al equipo a comprender la teoría y la práctica, y a ser multifuncionales.
Developer Team (DT)	Son los desarrolladores de un incremento funcional del producto en cada Sprint.

Fuente: Elaboración propia.

Tabla 6

Subroles y funciones del Scrum Team del proyecto.

Scrum Team - Subroles	Función	Rol
Analista	Elaborar el Impact Mapping, las historias de usuario y el Product Backlog.	PO, SM
UED (Diseñador de experiencias de usuario)	Diseñar las interfaces de usuario asegurándose de una buena experiencia de usuario del incremento a realizar en el Sprint.	PO, DT
Programador	Implica el desarrollo front-end y back-end del aplicativo Web.	DT
Tester	Probar y verificar los criterios de aceptación	PO, DT

Fuente: Elaboración propia.

5.5.2. Definición del Producto

El análisis y la planeación inicial del proyecto, comprende la definición del alcance, de las historias de usuario y del Product Backlog del proyecto.

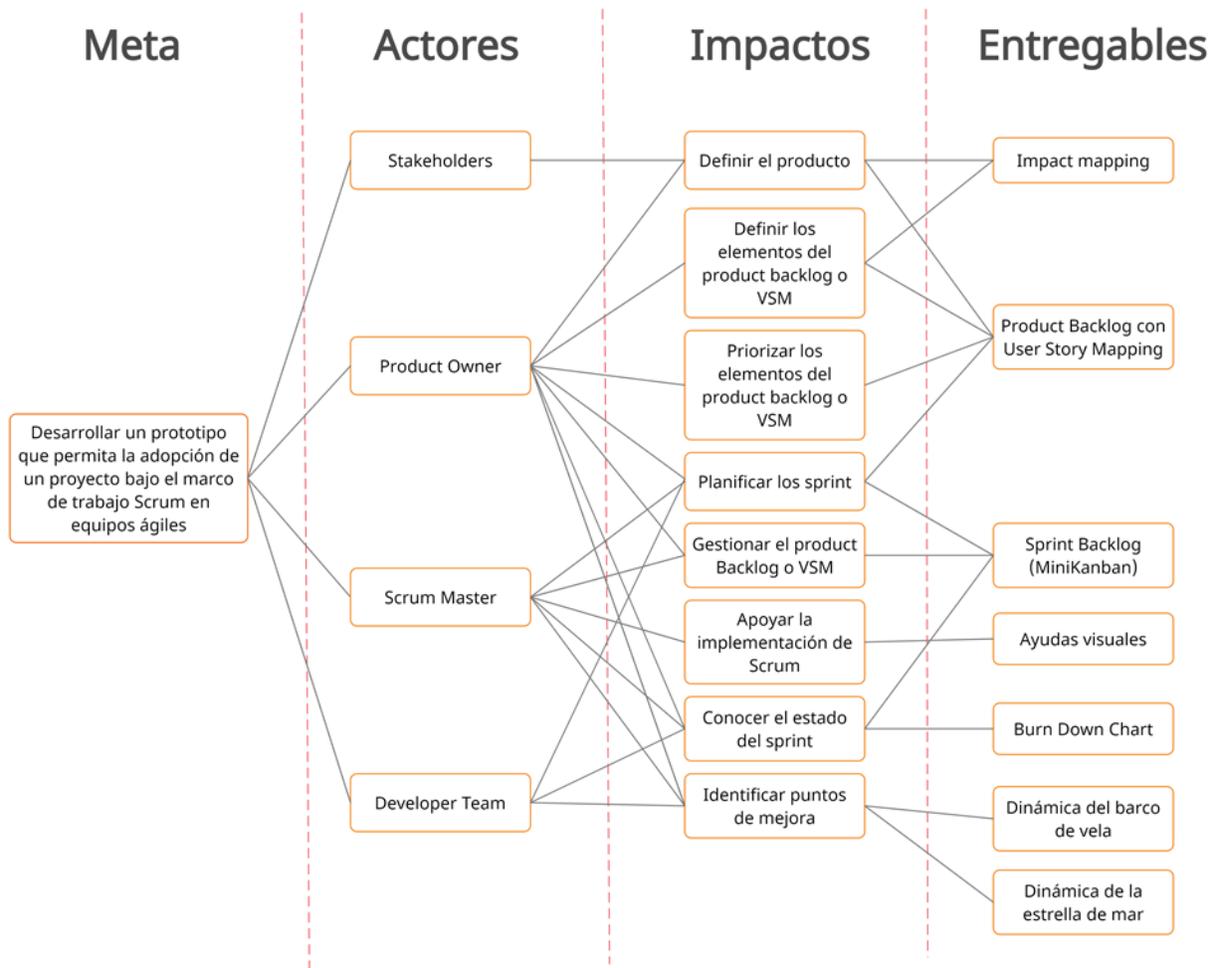
5.5.2.1. Incepción Ágil: Impact Mapping. En Scrum, aunque no se tienen fases, si debe empezar con un proceso de planificación y análisis a través de la incepción ágil a fin de definir el alcance, y es ahí donde el Impact Mapping se convierte en una herramienta de planificación y estrategia, es decir, un aliado que permite a través del uso de preguntas, “identificar qué es necesario conseguir y cómo hacerlo. Se utiliza en la fase de inception o lanzamiento de un proyecto para enfocar objetivos” (Autentia, s/f).

De esta forma, se empieza con la búsqueda de una herramienta que permita hacer uso de esta técnica, por lo que se analizan las mencionadas en el estado del arte, la primera fue la página Web UXPressia, sin embargo solo ofrecía la creación de un único actor en su versión gratuita;

debido a esta limitante se analizó el siguiente sitio web que es Miro, sin embargo ofrecía un Template que era la adaptación de un mapa mental por lo que también se descartó, y finalmente se hizo uso de Creately, la cual aunque al igual que Miro ofrecía un Template, su plantilla poseía las separaciones entre los componentes del Impact Mapping, y por ello se utilizó en el proyecto obteniendo los resultados de la Figura 19.

Figura 19

Impact Mapping del proyecto.



Fuente: Elaboración propia.

5.5.2.2. Product Backlog con User Story Mapping y Planeación de los Sprint. El

Product Backlog con sus historias de usuario se gestionó con Jira Software, en esta herramienta las historias de usuario se visualizan como se puede ver en la Figura 20. Donde los puntos de estimación por historia que se basan en la técnica: Estimación con puntos Fibonacci.

Figura 20

Vista de una historia de usuario personalizada en Jira Software.

The screenshot shows a Jira user story card for 'Impact Mapping - Crear Meta'. The card is in a 'Finalizada' (Completed) status, indicated by a green bar with a dropdown arrow and a 'Listo' (Ready) checkmark. The description is: 'Yo como: Product Owner Quiero: Definir alcance. Para: Conocer el valor del negocio.' The acceptance criteria are: 'Debe ser la primera tarjeta del mapa y la única en META.' and 'Al dar click sobre el texto de la tarjeta se debe poder editar.' The tasks are: 'Configurar los modelos y los servicios.', 'Crear el contenedor de META.', and 'Crear la Tarjeta.' The card also shows a 'Campos anclados' (Anchored fields) section with 'Responsable' (Responsible) set to 'Ym Jessica Malaver' and 'Story point estimate' set to '8'.

Campos anclados	
Responsable	Ym Jessica Malaver
Story point estimate	8

Fuente: Elaboración propia con Jira Software.

De igual forma, en Jira Software las historias de usuario se organizan en el Product Backlog en formato de lista plana y vertical como se puede ver en la Figura 21.

Figura 21

Vista del Product Backlog en Jira Software.

T	Clave ↑	Resumen	Responsable	Story point estimate
	SCRUM-1	Impact Mapping - Crear Meta	Yessica Malaver	8
	SCRUM-2	Impact Mapping - Crear Actores	Yessica Malaver	5
	SCRUM-3	Impact Mapping - Crear Impactos	Yessica Malaver	5
	SCRUM-4	Impact Mapping - Crear Entregables	Yessica Malaver	5
	SCRUM-5	Impact Mapping - Crear líneas	Yessica Malaver	8
	SCRUM-6	User Story Map - Generar Acciones	Yessica Malaver	8
	SCRUM-7	User Story Map - Importar Entregables	Yessica Malaver	8
	SCRUM-8	User Story Map - Crear prioridad en categorías	Yessica Malaver	3
	SCRUM-10	User Story Map - Crear, Editar y Eliminar HU	Yessica Malaver	8
	SCRUM-11	User Story Map - Añadir/Eliminar Criterios de Aceptación	Yessica Malaver	2
	SCRUM-12	User Story Map - Añadir / Eliminar Tareas	Yessica Malaver	2

Fuente: Jira Software.

Sin embargo, a fin de conocer a cuál actividad o entregable está ligada cada historia de usuario, se usó la herramienta online caco.com para construir el mapa de historias o User Story Mapping, que se puede visualizar en la Figura 22. Donde se establecieron las siguientes actividades partiendo de los entregables del Impact Mapping:

- Impact Mapping. Con 8 historias de usuario, abordadas 6 en el Sprint 1 y 2 en el Sprint 4.
- User Story Mapping. Con 11 historias de usuario, abordadas 2 en el Sprint 1, 6 en el Sprint 2, y 3 en el Sprint 5.
- Sprint Backlog. Con 5 historias de usuario, 3 abordadas en el Sprint 2 y 2 en el Sprint 5.

- Burn down chart. Con 3 historias de usuario, 2 solucionadas en Sprint 3 y 1 en el Sprint 5.
- Dinámica del barco de vela. Con 2 historias de usuario, 1 para el Sprint 3 y otra para el Sprint 5.
- Dinámica de la estrella de mar. Con 2 historias de usuario, 1 en el Sprint 3 y otra en el Sprint 5.
- Ayudas visuales. Con 4 historias de usuario, 2 ubicadas en el Sprint 3 y 2 en el Sprint 5.
- Home de Inicio de Sesión. Con 1 historia de usuario abarcada en el Sprint 4.
- Home de Proyectos. Con 4 historias de usuario, 3 en el Sprint 4, y 1 en el Sprint 5.

A su vez, los Sprints se entregaron en el tiempo de 1 mes, con puntos de estimación de 55 a 57, trabajándose de lunes a viernes, alrededor de 4 a 8 horas diarias, siendo los festivos, sábados y domingos días de margen en caso de contratiempos. En la Tabla 7 se puede ver el cronograma de la entrega de los incrementos.

Tabla 7

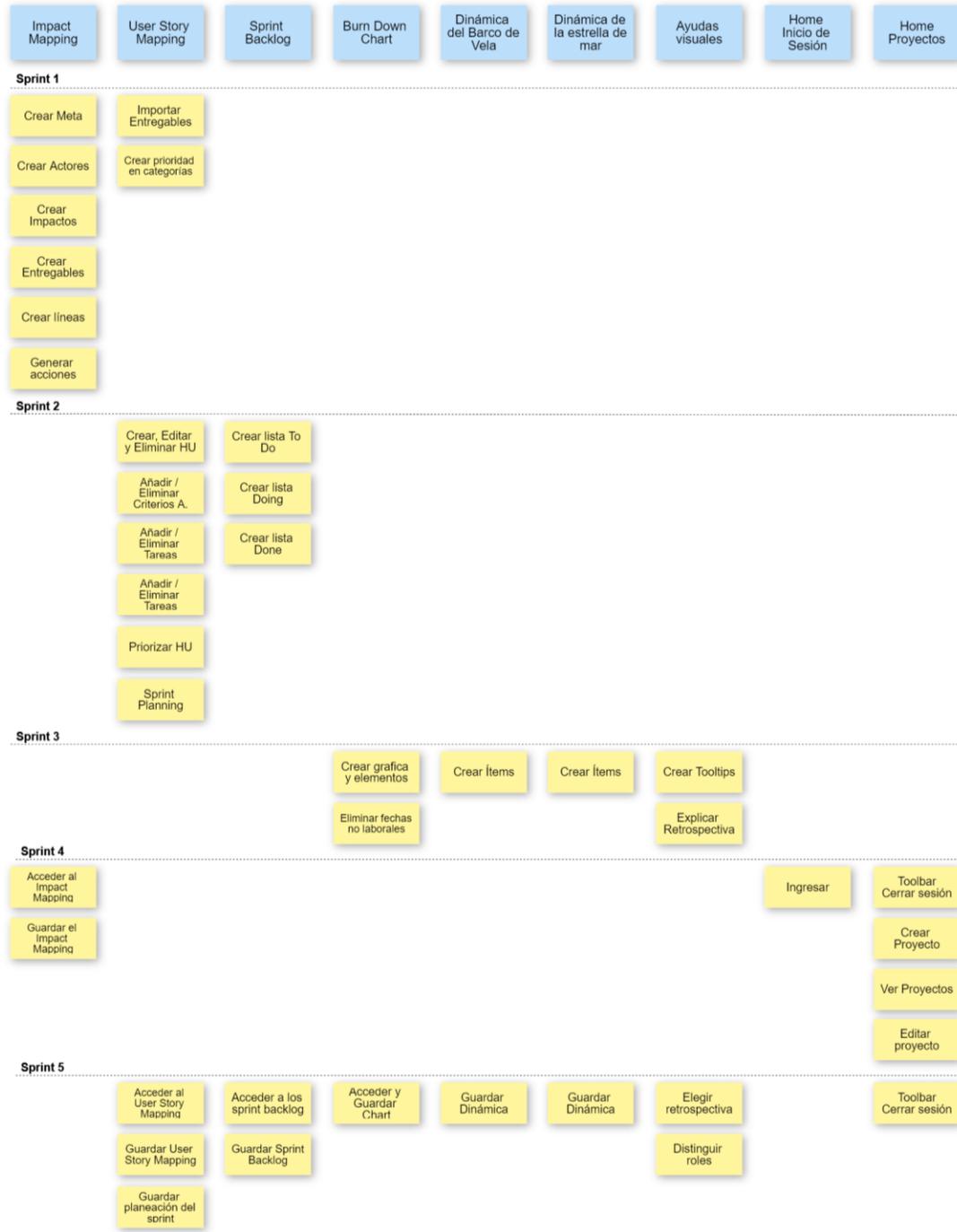
Cronograma de los Sprints.

Sprint	Duración (días)	Inicio Sprint	Fin Sprint / Entrega	Mes				
				Abril	Mayo	Junio	Julio	Agosto
1	20	1/Abril/2021	30/Abril/2021	■				
2	20	1/Mayo/2021	31/Mayo/2021		■			
3	20	1/Junio/2021	30/Junio/2021			■		
4	20	1/Julio/2021	31/Julio/2021				■	
5	21	1/Ago./2021	31/Ago./2021					■

Fuente: Elaboración propia.

Figura 22

User Story Mapping del proyecto.



Fuente: Elaboración propia con caco.com.

5.5.3. *Sprint Backlog del Sprint 1*

Durante el Sprint 1 se tuvo como objetivo: “Crear los módulos de Impact Mapping y User Story Mapping que le permita al Product Owner generar la inepción ágil del proyecto”. Se manejaron 8 historias de usuario que se especifican en la Tabla 8, para 55 puntos de historia.

Tabla 8

Sprint Backlog del Sprint 1.

Actividad	HU	Descripción / Criterios de Aceptación / Tareas	Puntos
Impact Mapping	Crear Meta	<p>Descripción: Yo como: Product Owner. Quiero: Definir alcance. Para: Conocer el valor del negocio.</p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none"> • Debe ser la primera tarjeta del mapa y la única en META. • Al dar click sobre el texto de la tarjeta se debe poder editar. <p>Tareas:</p> <ul style="list-style-type: none"> • Configurar los modelos y los servicios. • Crear el contenedor de META. • Crear la Tarjeta. 	8
		<p>Descripción: Yo como: Product Owner. Quiero: Agregar actores. Para: conocer los afectados o relacionados con el proyecto.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Un contenedor guardara las tarjetas de los actores. • Al dar click sobre el texto de la tarjeta se debe permitir editar. • Al dar click en un botón se añadirán actores. • Se debe permitir mover tarjetas dentro del contenedor. <p>Tareas:</p> <ul style="list-style-type: none"> • Configurar los modelos y servicios. • Crear el contenedor de ACTORES. • Crear la Tarjeta. • Crear botón de añadir en la tarjeta del contenedor META. • Crear Drag and Drop en las tarjetas. 	5

Actividad	HU	Descripción / Criterios de Aceptación / Tareas	Puntos
Impact Mapping	Crear Impactos	<p>Descripción: Yo como: Product Owner. Quiero: Agregar impactos de los actores. Para: identificar las actividades de negocio.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> Las tarjetas de los impactos se almacenarán en un contenedor. Las tarjetas se deben poder editar. Al dar click en un botón se añadirán impactos. Se debe permitir mover tarjetas dentro del contenedor. <p>Tareas:</p> <ul style="list-style-type: none"> Configurar los modelos y los servicios. Crear el contenedor de IMPACTOS. Crear la Tarjeta. Crear botón de añadir impactos en las tarjetas del contenedor de ACTORES. Crear Drag and Drop en las tarjetas. 	5
Impact Mapping	Crear Entregables	<p>Descripción: Yo como: Product Owner. Quiero: Agregar entregables a partir de los impactos. Para: identificar los entregables a realizar.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> En un contenedor se acumularán las tarjetas de entregables. las tarjetas se deben poder editar. Al dar click en un botón se añadirán actores. Se debe permitir mover tarjetas dentro del contenedor. <p>Tareas:</p> <ul style="list-style-type: none"> Configurar los modelos y servicios. Crear el contenedor de ENTREGABLES. Crear la Tarjeta. Crear botón de añadir entregables en las tarjetas del contenedor de IMPACTOS. Crear Drag and Drop en las tarjetas. 	5

Actividad	HU	Puntos
Impact Mapping	<p>Crear líneas</p> <p>Descripción: Yo como: Product Owner. Quiero: que las tarjetas de los contenedores se relacionen con líneas. Para: identificar la secuencia del Impact Mapping.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Crear líneas de relación entre tarjetas de diferente contenedor. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear componente de líneas. • Implementar relaciones de las líneas con las tarjetas. 	8
Impact Mapping	<p>Generar Acciones</p> <p>Descripción: Yo como: Product Owner. Quiero: poder eliminar actores, impactos y entregables, y añadir o eliminar relaciones entre estos. Para: organizar el contenido del proyecto.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Se podrá añadir una relación si la tarjeta padre no contiene a la hija. • Se podrá eliminar una relación si hay más tarjetas padres. • Se podrá eliminar una tarjeta si no cuenta con más tarjetas padres ni más tarjetas hijas. • No se podrá realizar ninguna acción si la tarjeta no tiene más tarjetas padres y tiene tarjetas hijas. • Un botón activara las acciones. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear botón de activar acciones en todas las tarjetas de los contenedores META, ACTORES e IMPACTOS. • Crear botones de eliminar tarjeta, añadir y eliminar relaciones en las tarjetas de los contenedores ACTORES, IMPACTOS y ENTREGABLES. • Realizar validaciones de cuando se añade o elimina una relación o tarjeta. 	8
User Story Mapping	<p>Importar Entregables</p> <p>Descripción: Yo como: Product Owner. Quiero: Importar los entregables del Impact Mapping. Para: Definir las actividades que se van a hacer en el proyecto.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Importar los entregables del Impact Mapping. • Mostrar en el Backbone (actividades) del VSM y serán las columnas del mapa. <p>Tareas:</p> <ul style="list-style-type: none"> • Configurar modelo y servicio. • Crear contenedor del Backbone. 	8

Actividad	HU	Descripción / Criterios de Aceptación / Tareas	Puntos
User Story Mapping	Crear prioridad en categorías	<p>Descripción: Yo como: Product Owner. Quiero: Utilizar la métrica MoSCoW. Para: Priorizar las historias de usuario de un proyecto.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Realizar filas por cada ítem de la técnica de priorización MoSCoW con títulos como: Alta, Media, Baja y Nula. • Botón en cada celda para crear HU. <p>Tareas:</p> <ul style="list-style-type: none"> • Realizar contenedores para la prioridad. • Alinear los contenedores a fin de que intercepten entre actividad y prioridad. • Crear botón de añadir tarjeta (HU). 	8

Fuente: Elaboración propia.

5.5.4. *Sprint Backlog del Sprint 2*

Durante el Sprint 2 el objetivo fue: “Crear componentes del User Story Mapping, planeación de los Sprint y el módulo de Sprint Backlog que le permita al Scrum Team observar el avance del proyecto”. Incluyó 8 historias de usuario que se especifican en la Tabla 9, con una estimación de 56 puntos.

Tabla 9*Sprint Backlog del Sprint 2.*

Actividad	HU	Descripción / Criterios de Aceptación / Tareas	Puntos
User Story Mapping	Crear, Editar y Eliminar HU	<p>Descripción: Yo como: Product Owner. Quiero: Definir las Historias de Usuario (HU). Para: Especificar las actividades del proyecto.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Con el botón de Añadir HU se creará una tarjeta en el contenedor. • La HU creada tendrá un botón de editar, que abrirá una ventana modal que contendrá los campos de la HU. • La HU contendrá los datos de: Yo como, Quiero, Para, Estimación, Responsable, Criterios de aceptación y tareas. • Dentro de la modal abra un botón de eliminar HU. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear modelo. • Realizar conexión con el servicio. • Crear Dialog (Modal). • Conectar el botón de añadir con el Dialog. • Crear los campos de la HU. • Crear botón de eliminar HU. 	5
		User Story Mapping	Añadir/Eliminar Criterios de Aceptación

Actividad	HU	Descripción / Criterios de Aceptación / Tareas	Puntos
User Story Mapping	Añadir / Eliminar Tareas	<p>Descripción: Yo como: Developer Team. Quiero: Describir tareas por cada HU. Para: Saber que debo hacer para resolver la HU.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • En la ventana modal de la HU, se debe permitir ingresar las tareas. • Se debe contar con un botón para añadir tareas. • Se debe contar con un botón de eliminar en cada tarea. <p>Tareas:</p> <ul style="list-style-type: none"> • Conectar con el servicio que contiene las HU. • Crear botón de añadir. • Crear botón de eliminar. 	2
User Story Mapping	Priorizar HU	<p>Descripción: Yo como: Product Owner. Quiero: Priorizar las HU. Para: Saber que va y que no en el proyecto.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Se debe poder mover las HU entre los diferentes columnas de prioridad de Alta, Media, Baja y Descartada, en el tablero del User Story Mapping. • Se debe poder mover la HU entre las diferentes filas del User Story Mapping para poder cambiar las de actividad en caso de ser necesario. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear Drag and Drop Group. • Conectar con los servicios. 	8
User Story Mapping	Sprint Planning	<p>Descripción: Yo como: Scrum Team. Quiero: Seleccionar HU del User Story Mapping. Para: planificar un Sprint.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • El Product Owner puede elegir si se trata de la planeación de un Sprint nuevo o existente. • Puede seleccionar la fecha del Sprint y escribir el objetivo del Sprint. • Puede seleccionar las HU que pertenecen al Sprint. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear modelos y servicios. • Crear radio button para saber si es nuevo o existente el Sprint. • Ubicar campo de fecha y objetivo. • Crear botones en la HU para seleccionarlas. 	13

Actividad	HU	Puntos
Sprint Backlog Crear lista To Do	<p>Descripción: Yo como: Scrum Team. Quiero: Importar y Definir un listado ordenado de las Historias de Usuario seleccionadas del Visual Story Mapping. Para: Conocer cuáles son las HU a realizar durante el Sprint.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • En la acción del Sprint Planning, las HU seleccionadas se deben enviar al sprint de tal forma que sean visibles en el Sprint Backlog. • En el Sprint Backlog las HU deben aparecer en una lista de To Do. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear lista To Do con acción de drag and drop. • Sincronizar modelos y servicios del Visual Story Mapping y del Sprint Backlog. 	13
Sprint Backlog Crear lista Doing	<p>Descripción: Yo como: Developer Team. Quiero: indicar las HU que se están resolviendo. Para: que el resto del equipo conozca las HU que se están realizando.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Generar drag and drop entre la lista de To Do y Doing. • Las HU permanecen en esta lista si falta algún criterio o tarea por resolver. <p>Tareas:</p> <ul style="list-style-type: none"> • Generar drag and drop entre listas y en la lista Doing. • Crear estados en las HU para los criterios de aceptación y tareas. • Sincronizar con el servicio. 	5
Sprint Backlog Crear lista Done	<p>Descripción: Yo como: Scrum Master y Product Owner. Quiero: Calificar las historias de usuario. Para: Dar cierre a las tareas.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Si las HU tienen todos los criterios de aceptación y tareas chequeados, se debe pasar automáticamente a Done. • Desde Done no se puede hacer drag and drop a otras tablas a menos que se quite el check a algún criterio o tarea. • Las HU deben poderse organizar en la lista. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear funciones en los modelos y servicios. • Validar con checklist los estados en las HU. • Crear Drag and Drop en la lista Done. • Sincronizar con el servicio. 	8

Fuente: Elaboración propia.

5.5.5. *Sprint Backlog del Sprint 3*

Durante el Sprint 3 se planteó como objetivo “Generar los módulos para el burn down chart y las retrospectivas, e implementar ayudas visuales, a fin de conocer el progreso del Sprint y adecuar la herramienta visualmente” para ello se manejan 6 historias de usuario para un total de 55 puntos de historia, se pueden visualizar en la Tabla 10.

Tabla 10

Sprint Backlog del Sprint 3.

Actividad	HU	Descripción / Criterios de Aceptación / Tareas	Puntos
Burn Down Chart	Crear Gráfica y elementos	<p>Descripción: Yo como: Scrum Master. Quiero: Graficar las HU cerradas. Para: conocer la velocidad del equipo.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • El máximo valor en el eje Y es la sumatoria de los puntos de estimación de las HU seleccionadas el día de la planeación del Sprint. • Se debe graficar la línea de progreso ideal. • Se debe graficar la línea de avance real a medida que se cierran las HU. <p>Tareas:</p> <ul style="list-style-type: none"> • Añadir librería de gráficos. • Realizar componente del gráfico. <p>Hacer visible línea de progreso real e ideal.</p>	21
Burn Down Chart	Eliminar Fechas no laborales	<p>Descripción: Yo como: Developer Team. Quiero: Indicar las fechas no laborales. Para: Que no se tengan en cuenta esos días en el Sprint.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Por medio de un calendario indicar las fechas no laborales, teniendo la opción de cancelar día por día o todo un día de la semana. • Actualizar la gráfica del burn down chart según los días laborales. <p>Tareas:</p> <ul style="list-style-type: none"> • Añadir el componente de calendar. • Especificar condicionales para los días seleccionados. • Sincronizar la gráfica. 	13

Actividad	HU	Descripción / Criterios de Aceptación / Tareas	Puntos
Dinámica: Estrella de mar	Crear Ítems	<p>Descripción: Yo como: Scrum Team. Quiero: Identificar y crear acciones a implementar. Para: Construir el plan de mejora.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Crear 5 tarjetas que permitan escribir lo relacionado a los componentes de la dinámica: Más de, Menos de, Empezar a hacer, Dejar de hacerlo, Seguir haciendo. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear tarjetas • Configurar los modelos y servicios. 	8
Dinámica: Barco de Vela	Crear Ítems	<p>Descripción: Yo como: Scrum Team. Quiero: Identificar aspectos positivos, problemas, riesgos y oportunidades que ha habido en el Sprint. Para: Identificar los aspectos del plan de mejora.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Crear 5 tarjetas que permitan escribir lo relacionado a los componentes de la dinámica: Meta, Debilidades, Fortalezas, Obstáculos, Cosas por agradecer. • Relacionar cada ítem con su representación: Isla (Meta), Ancla (Debilidades), Vela (Fortalezas), Rocas (Obstáculos), Sol (Cosas por agradecer). • Otra tarjeta debe permitir ingresar la conclusión del plan de mejora. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear tarjetas. • Configurar los modelos y servicios. 	8
Ayudas visuales	Crear Tooltips	<p>Descripción: Yo como: Scrum Master. Quiero: que la herramienta sea lo más intuitiva posible. Para: que el Scrum Team comprenda las acciones que puede realizar.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Añadir etiquetas o textos alternativos sobre ciertas acciones o botones a fin de guiar al usuario sobre la finalidad del elemento. <p>Tareas:</p> <ul style="list-style-type: none"> • Incluir Tooltips en botones que no tienen texto. 	2

Actividad	HU	Descripción / Criterios de Aceptación / Tareas	Puntos
Ayudas visuales	Explicar Retrospectivas	<p>Descripción: Yo como: Scrum Master. Quiero: Que la herramienta en el módulo de retrospectivas tenga una ilustración y una explicación sobre qué consiste la dinámica. Para: Que el Scrum Team comprenda la dinámica.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • La ilustración debe contener los ítems o elementos de la dinámica. • El texto debe explicar lo que se debe realizar. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear ilustraciones. • Agregar la imagen en su respectiva retrospectiva. • Crear contenedor del texto. 	3

Fuente: Elaboración propia.

5.5.6. *Sprint Backlog del Sprint 4*

El objetivo del Sprint 4 es “Generar ingreso al aplicativo, a los proyectos y al Impact Mapping, con su respectivo almacenamiento a fin de acceder a la información y su secuencia lógica”, se compone de 7 historias de usuario con 55 puntos de estimación, como se ve en la Tabla 11.

Tabla 11*Sprint Backlog del Sprint 4.*

Actividad	HU	Descripción / Criterios de Aceptación / Tareas	Puntos
Home – Inicio de sesión	Ingresar	<p>Descripción: Yo como: Scrum Team. Quiero: Ingresar a la plataforma. Para: Poder tener acceso a los proyectos en los que soy colaborador.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • El usuario se debe registrar antes de iniciar sesión, si es la primera vez. • Si el usuario se encuentra registrado le debe permitir iniciar sesión. • Debe permitir iniciar sesión o registro con opciones por defecto como GitHub, Google, Microsoft, o con correo y contraseña alterno. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear Home de Inicio de sesión, con el botón de Ingresar. • Al dar click en el botón ingresar se debe generar la conexión con Auth0. • Al iniciarse sesión se debe redirigir a la ventana de proyectos. 	5
Home - Proyectos	Crear Tool Bar – Cerrar sesión	<p>Descripción: Yo como: Scrum Team. Quiero: Una barra de herramientas. Para: Saber si ya inicié sesión y tener acceso a cerrar sesión.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • La barra de herramientas debe contener el logo del aplicativo, el nombre de la persona que inicio sesión y su imagen, y un botón para cerrar sesión. • Al dar click en el botón de cerrar sesión se debe redirigir al home de inicio de sesión. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear toolbar. • Crear botón de cerrar sesión y conectarlo con auth0. • Recuperar de Auth0 el nombre y la imagen del usuario. 	3

Actividad	HU	Descripción / Criterios de Aceptación / Tareas	Puntos
Home Proyectos	Crear Proyecto	<p>Descripción: Yo como: Scrum Team. Quiero: Crear un proyecto. Para: Poder organizar la información del mismo.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Debe contener el campo de nombre del proyecto. • Debe contener los campos para introducir los correos de al menos el Product Owner, el Scrum Master, y 1 Developer. • Debe permitir añadir integrantes al Developer Team. • El botón para añadir proyecto debe estar en el home de proyectos. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear Dialog del proyecto. • Crear campo de nombre del proyecto. • Crear campos de correo para los integrantes del proyecto. • Crear botón de añadir integrante al Developer Team. • Crear botón para eliminar Developer Team. • Crear botones de Cancelar y Guardar. • Generar modelo y servicio. • Generar conexión al servicio que usa la API REST de Firebase. 	13
Home Proyectos	Ver Proyectos	<p>Descripción: Yo como: Scrum Team. Quiero: Ver los proyectos a los que pertenezco. Para: saber en cuales puedo colaborar y ver su información.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • El usuario debe poder ver a cuál proyecto está vinculado. • El usuario debe saber cuál rol ocupa en cada proyecto. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear tarjetas de proyectos. • Crear “chips” de angular material para indicar el rol al que pertenece el usuario. • Generar vínculo con el modelo y servicio. • Generar conexión al servicio que usa la API REST de Firebase. 	8

Actividad	HU	Descripción / Criterios de Aceptación / Tareas	Puntos
Home Proyectos	Editar Proyecto	<p>Descripción: Yo como: Scrum Team. Quiero: Modificar los datos básicos del proyecto. Para: corregir nombre del proyecto y correo de los integrantes, o añadir integrantes; en caso de ser necesario.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Debe haber un botón que permita editar proyecto. • Al dar click en el botón de editar proyecto se debe abrir el Dialog de edición. <p>Tareas:</p> <ul style="list-style-type: none"> • Reutilizar componente Dialog del crear proyecto. • Crear botón de editar. • Conectar con el servicio de proyectos con el servicio que usa la API REST de Firebase. 	5
Impact Mapping	Acceder al Impact Mapping	<p>Descripción: Yo como: Product Owner. Quiero: Crear el Impact Mapping de un proyecto. Para: conocer el alcance del mismo.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • En el home de proyectos, cada proyecto debe tener un botón que diga “Impact Mapping.” • Al dar click, debe redirigir al módulo de Impact Mapping. • Si es la primera vez, debe aparecer una tarjeta en el contenedor de Meta. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear botón de “Impact Mapping.” • Generar conexión al servicio que usa la API REST de Firebase. • Generar conexión al servicio de proyectos e Impact Mapping. • Validar si hay Impact Mapping o crear uno. 	13

Actividad	HU	Descripción / Criterios de Aceptación / Tareas	Puntos
Impact Mapping	Guardar el Impact Mapping	<p>Descripción: Yo como: Product Owner. Quiero: Guardar los cambios realizados en el Impact Mapping. Para: almacenar el progreso y poder consultarlo previamente, y que sea visto por el resto del equipo.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Debe haber un botón de guardar, que al dar click guarde la información e indique que ya lo hizo. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear botón de Guardar. • Almacenar información en Firebase con el uso de los servicios. • Generar un alert, una vez quede almacenada la actualización del Impact Mapping. 	8

Fuente: Elaboración propia.

5.5.7. *Sprint Backlog del Sprint 5*

El objetivo del Sprint 5 es “Unir los módulos del proyecto con su respectivo almacenamiento en base de datos, y su debida indicación del rol encargado, de tal manera que el equipo pueda construir un proyecto completo y reconociendo sus funciones según su rol”. Se compone de 10 historias de usuario y manejando un total de 57 puntos de estimación, como se puede ver en la Tabla 12.

Tabla 12*Sprint Backlog del Sprint 5.*

Actividad	HU	Descripción / Criterios de Aceptación / Tareas	Puntos
User Story Mapping	Acceder al User Story Mapping	<p>Descripción: Yo como: Scrum Team. Quiero: acceder al user story mapping del proyecto. Para: conocer su Product Backlog.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • En el home de proyectos debe haber un botón por cada proyecto que diga “User Story Mapping.” Al dar click debe redirigir al módulo de User Story Mapping si hay entregables en el Impact Mapping. • Si no hay entregables en el Impact Mapping, y se da click en el botón de “User Story Mapping”, se debe informar que debe crear Entregables en el Impact Mapping. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear botón de “User Story Mapping.” • Generar conexión al servicio que usa la API REST de Firebase. • Generar conexión al servicio de proyectos y de User Story Mapping. • Validar entregables en el Impact Mapping. 	13
User Story Mapping	Guardar User Story Mapping	<p>Descripción: Yo como: Product Owner. Quiero: Guardar los cambios realizados en el User Story Mapping. Para: almacenarlos y consultarlos posteriormente, además de ser vistos por el resto del equipo.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Debe haber un botón de guardar, que al dar click almacene la información e indique si lo hizo. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear botón de guardar. • Generar conexión con los servicios. • Generar un alert, que se active una vez se guarde la información en Firebase. 	8

Actividad	HU	Puntos
User Story Mapping	<p data-bbox="280 426 313 831">Guardar planeación del Sprint</p> <p data-bbox="337 426 1268 863"> Descripción: Yo como: Scrum Team. Quiero: Crear, Editar y Guardar un Sprint. Para: conocer que se va a resolver en un Sprint. Criterios de aceptación: <ul data-bbox="386 569 1256 636" style="list-style-type: none"> • Los Sprints nuevos o editados se deben guardar con el uso de un botón. Tareas: <ul data-bbox="386 680 1235 863" style="list-style-type: none"> • Conectar con la API REST de Firebase para conocer los Sprint existentes, guardar los nuevos y los cambios realizados en los existentes. • Crear botón de guardar. • Conectar con los servicios. </p>	8
Sprint Backlog	<p data-bbox="280 982 313 1377">Acceder a los Sprints Backlog</p> <p data-bbox="337 873 1338 1499"> Descripción: Yo como: Scrum Team. Quiero: Acceder a los Sprints Backlog. Para: conocer el progreso de los Sprint. Criterios de aceptación: <ul data-bbox="386 1020 1317 1272" style="list-style-type: none"> • En el home de proyectos, debe haber un botón por cada proyecto que diga “Sprint Backlog,” al dar click desplegara una lista de los Sprint existentes. • Al dar click en alguno de los Sprint, debe permitir redirigir a su respectivo Sprint Backlog. • Si es la primera vez que se accede a un Sprint Backlog, las HU se deben cargar a la lista de To Do. Tareas: <ul data-bbox="386 1316 1317 1499" style="list-style-type: none"> • Crear botón de “Sprint Backlog” en el home de proyectos. • Generar select de los Sprint, que se mostraran al dar click en el botón de “Sprint Backlog.” • Crear conexión con los servicios. • Cargar información en el módulo de Sprint Backlog. </p>	8

Actividad	HU	Descripción / Criterios de Aceptación / Tareas	Puntos
Sprint Backlog	Guardar Sprint Backlog	<p>Descripción: Yo como: Scrum Team. Quiero: Guardar los cambios realizados en el Sprint Backlog. Para: consultarlos previamente y que puedan ser vistos por los demás miembros del equipo.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Se deben guardar los cambios al dar click en un botón. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear botón de guardar. • Generar conexión con los servicios de Sprint Backlog y de almacenamiento de datos con Firebase. 	5
Burn Down Chart	Acceder y Guardar Chart	<p>Descripción: Yo como: Scrum Team. Quiero: Acceder al burn down chart del Sprint. Para: conocer el progreso del equipo e indicar los días no laborales.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Un botón debe redirigir desde el Sprint Backlog al burn down chart del Sprint. • Dentro del burn down chart debe haber un botón que permita guardar los días indicados como no laborales. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear botón de “Ir a Burn Down Chart” en el módulo de Sprint Backlog. • Generar conexión con los servicios. • Crear botón de guardar dentro del módulo de burn down chart. 	5
Ayudas visuales	Elegir retrospectiva	<p>Descripción: Yo como: Scrum Master. Quiero: Elegir una dinámica de retrospectivas. Para: emplear la más adecuada con el equipo.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Dentro del módulo de Sprint Backlog debe haber un botón que permita redirigir a la retrospectiva del Sprint. • Si no se ha realizado una retrospectiva en el Sprint, al dar click en el botón se debe preguntar cual retrospectiva se va a escoger, una vez seleccionada redirigir al módulo de la retrospectiva seleccionada. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear botón de “Ir a la retrospectiva” en el módulo de Sprint Backlog. • Generar Dialog para elección del tipo de dinámica para retrospectiva. • Generar conexión con los servicios de Sprint Backlog y de almacenamientos de datos en Firebase. 	2

Actividad	HU	Descripción / Criterios de Aceptación / Tareas	Puntos
Dinámica: Estrella de Mar	Guardar Dinámica	<p>Descripción: Yo como: Scrum Team. Quiero: Que los ítems contestados en la retrospectiva se guarden. Para: consultarlos en otra ocasión y recordar el plan de acción.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Al dar click en un botón se guardarán los cambios. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear botón de guardar. • Generar conexión con los servicios. 	3
Dinámica: Barco de vela	Guardar Dinámica	<p>Descripción: Yo como: Scrum Team. Quiero: Que los ítems contestados en la retrospectiva se guarden. Para: consultarlos en otra ocasión y recordar el plan de acción.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Al dar click en un botón se guardarán los cambios. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear botón de guardar. • Generar conexión con los servicios. 	3
Ayudas visuales	Distinguir roles	<p>Descripción: Yo como: Scrum Master. Quiero: que la herramienta mencione en los módulos cual es el rol encargado de gestionarlo. Para: que el Scrum Team comprenda las actividades que debe realizar cada rol.</p> <p>Criterios de aceptación:</p> <ul style="list-style-type: none"> • Cada módulo debe contener la información relacionada a cuál rol (Product Owner, Scrum Master, Developer Team) es el encargado de gestionarlo a través de una ficha o texto visible. • Se debe especificar las actividades a realizar del rol en el módulo. <p>Tareas:</p> <ul style="list-style-type: none"> • Crear formato del contenedor con la información relacionada al rol. • Aplicar formato en los módulos e introducir la información. 	2

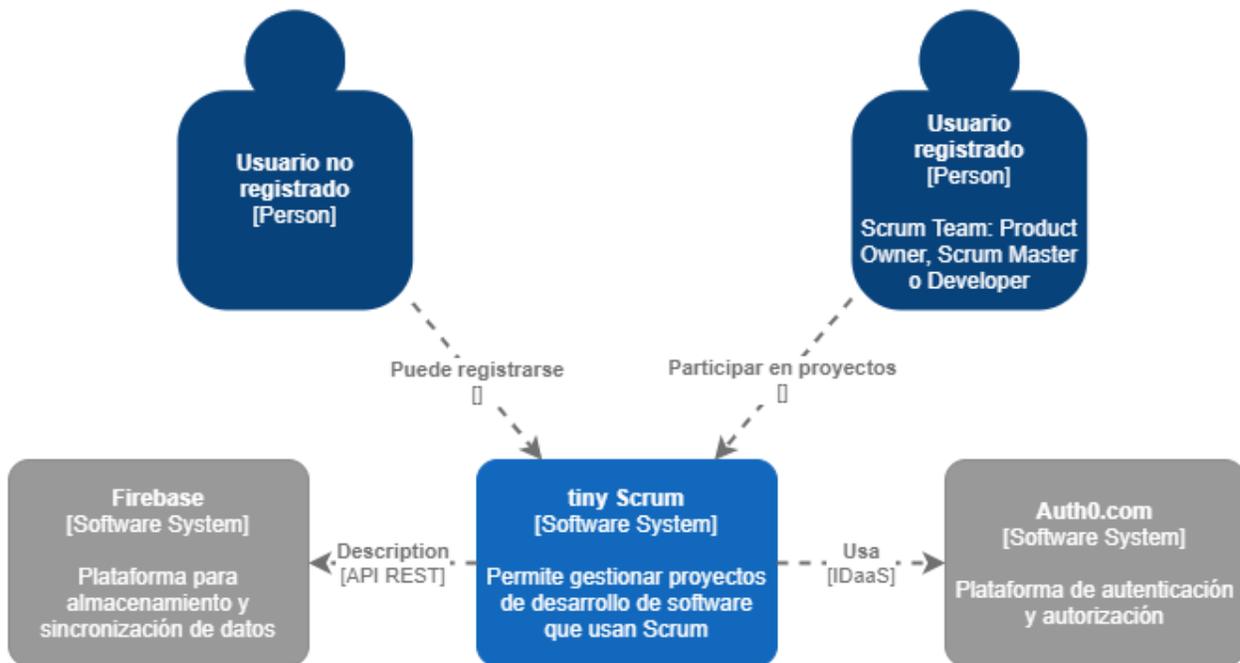
Fuente: Elaboración propia.

5.5.8. *Diseño del Software*

Después de la fase de planeación y análisis que se evidencia en las secciones anteriores de definición del producto y de los Sprints Backlog, durante cada Sprint se tuvo en cuenta el diseño del prototipo, como parte del ciclo de vida del software.

Esta fase se desarrolló con el uso del modelo C4 creado por Simon Brown, que permite visualizar la arquitectura del software a través de diagramas. Para ello, se hace uso de una jerarquía de abstracciones, creando una colección de diagramas de: Contexto, Contenedor, Componente y Código (Opcional). Los cuales comunican como se plantea construir el sistema bajo el concepto de arquitectura cliente-servidor en capas. A continuación se evidencian los primeros 3 niveles de abstracción de este proyecto, elaborados con diagrams.net (draw.io).

5.5.8.1. Nivel 1: Diagrama De Contexto Del Sistema. La Figura 23 permite ver que los usuarios que van a interactuar con el prototipo, llamado: tiny Scrum, una vez estén registrados podrán participar en la gestión de un proyecto que usa Scrum. Para ello, usa plataformas externas de apoyo para algunas de sus funcionalidades como el IDaaS (Identity as a Service – Identidad como un servicio) que ofrece Auth0 para autenticación de los usuarios, y Firebase para almacenamiento de datos en la nube, con el uso de su API REST.

Figura 23*Diagrama de Contexto del Prototipo.*

Nota: IDaaS (Identidad como servicio). Fuente: Elaboración propia con diagrams.net.

5.5.8.2. Nivel 2: Diagrama Del Contenedor. Este nivel se puede visualizar en la Figura 24. Diagrama del Contenedor del Prototipo, el cual explica de una forma técnica la forma de alto nivel de la arquitectura del software.

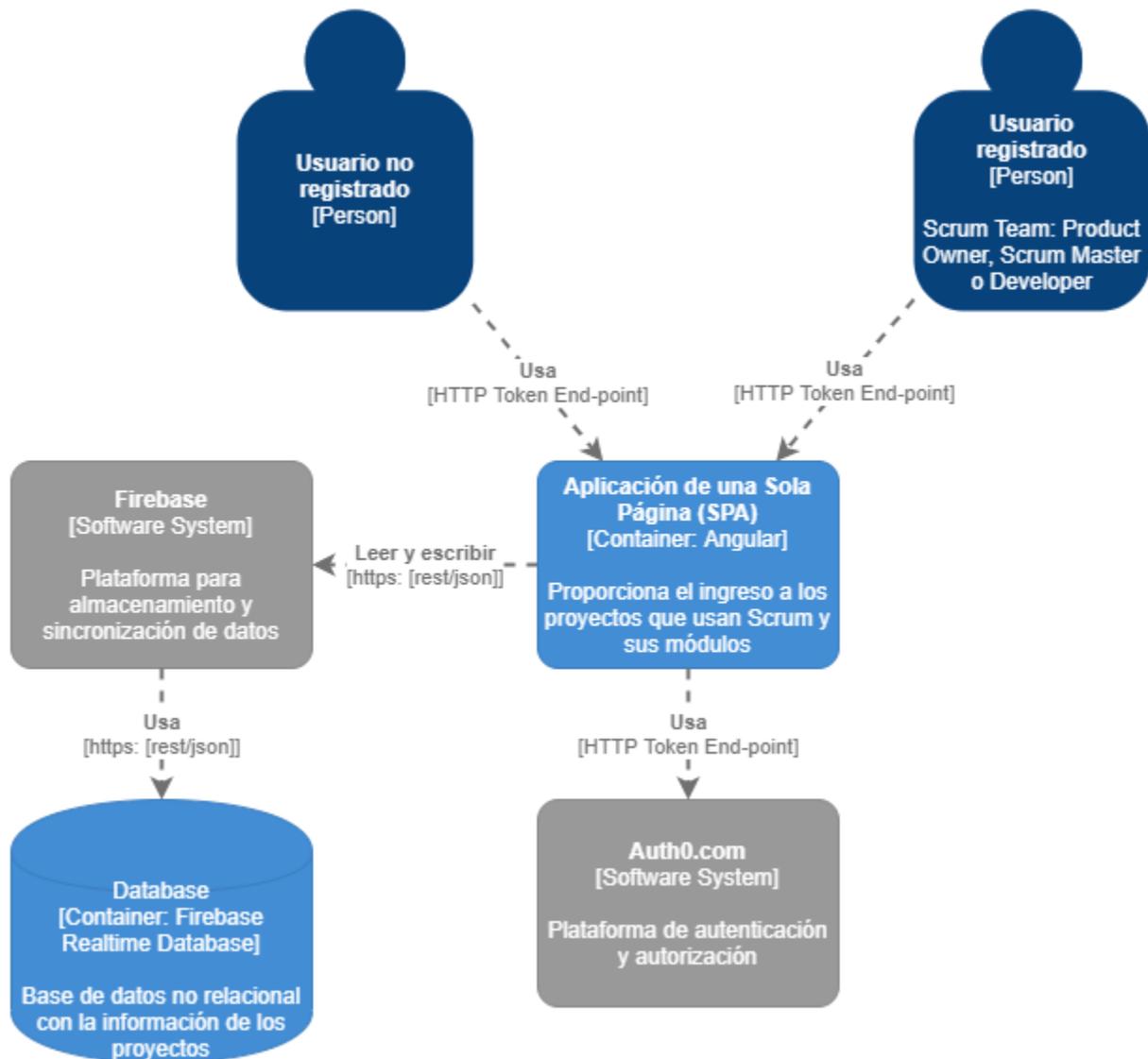
En este diagrama, se emplea un contenedor referente a la aplicación en Angular de una sola página (SPA), donde los usuarios acceden al sistema con una petición interna de tipo HTTP Token End-point para acceder a Auth0.

Además, el contenedor emplea la API REST de Firebase con el uso de un .json al final de la URL, y envía la petición desde un cliente HTTPS, pues solo así Firebase responde al tráfico encriptado para que sus datos permanezcan seguros. De esta manera, se genera la conexión con

un contenedor externo referente a la base de datos: Firebase Realtime Database, desde donde se escriben y leen datos.

Figura 24

Diagrama del Contenedor del Prototipo.

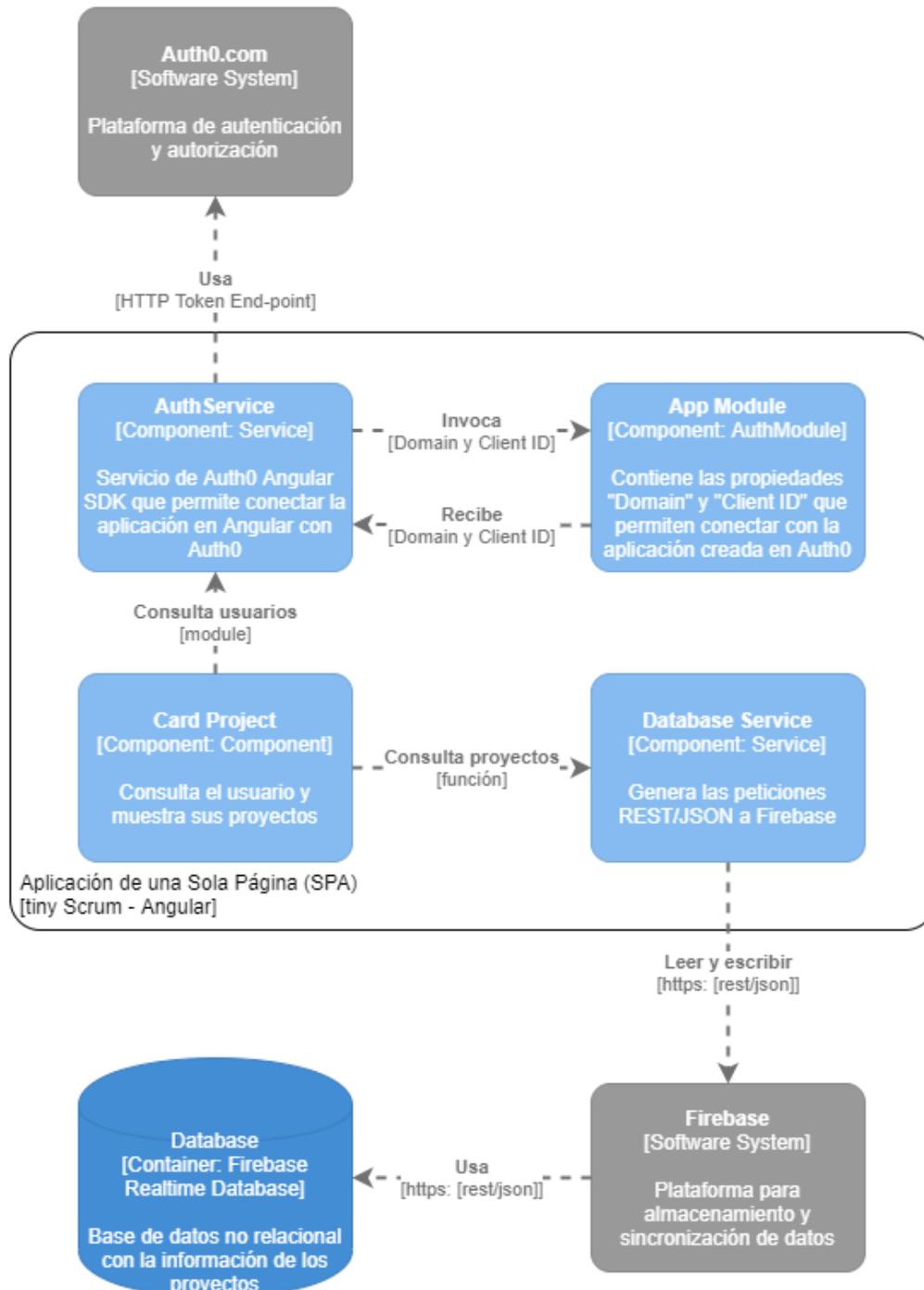


Fuente: Elaboración propia con diagrams.net.

5.5.8.3. Nivel 3: Diagrama De Componentes. Se puede visualizar en la Figura 25, el cual muestra cómo se descompone el único contenedor interno del sistema en una serie de componentes. Uno de estos, es el App Module (`app.module.ts`) el cual importa el `AuthModule.forRoot()` para configurar el módulo `AuthModule`, el cual contiene las propiedades “Domain” y “ClientID” presentes en la "Configuración" de la aplicación de una sola página que se registró con Auth0, y así establecer la conexión entre Auth0 y tiny Scrum. Así mismo, al hacer uso del Auth0 Angular SDK, se tiene por defecto el servicio `AuthService`, el cual consulta el “Domain” y “ClientID” para comprobar su acceso a la aplicación registrada en Auth0 y así brindar la verificación de la autenticación de usuario. Por otro lado, en Angular un servicio se encarga de conectarse con Firebase con el uso de peticiones API REST JSON/HTTPS, servicio que es consumido por varios componentes, entre ellos `Card-Project`, el cual también consume el servicio de auth0 a fin de generar la filtración de proyectos por usuario.

Figura 25

Diagrama de componentes del prototipo.



Fuente: Elaboración propia con diagrams.net.

5.5.9. Desarrollo del Software

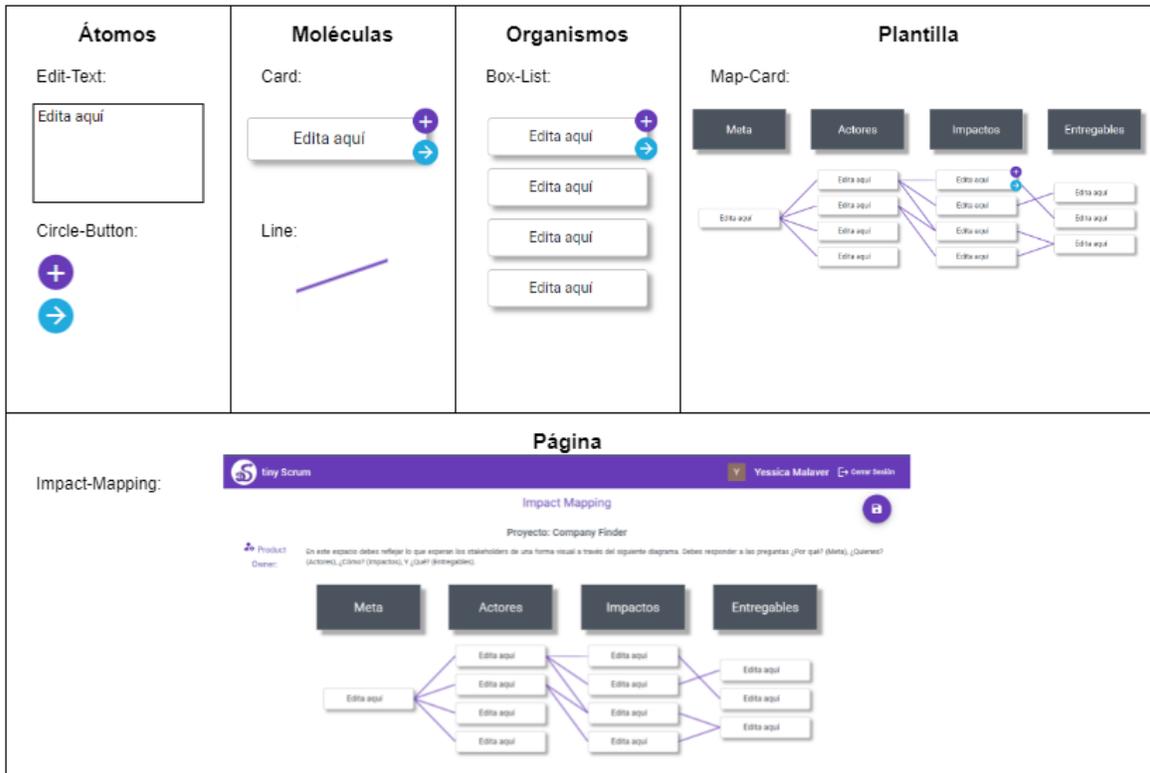
Durante cada Sprint se realizó esta fase del ciclo de vida del software, con base al diseño propuesto. Para ello, se usó como método de desarrollo front-end, el “Atomic Design” o diseño atómico, explicado en el marco conceptual, donde se plantea que “las interfaces están formadas por componentes más pequeños. Esto significa que podemos dividir interfaces enteras en bloques de construcción fundamentales y trabajar desde allí” (Frost, 2013). Esto permite que se pueda organizar y reutilizar el mayor código posible. Para ello su creador: Brad Frost, explica que el diseño atómico se compone por cinco etapas que crean sistemas de diseño de interfaces de una manera jerárquica, donde “no es un proceso lineal , sino un modelo mental que nos ayuda a pensar en nuestras interfaces de usuario como un todo cohesivo y una colección de partes al mismo tiempo” (Frost, 2016). Las cinco etapas del diseño atómico son:

- 1) Átomos: Elementos únicos de HTML nativo personalizados.
- 2) Moléculas: Combinación de elementos HTML.
- 3) Organismos: Secciones compuestas por “moléculas”.
- 4) Plantillas: Unión de secciones.
- 5) Páginas: Las plantillas usadas.

En la Figura 26 se puede ver un ejemplo de esta jerarquía en la cual se evidencia la descomposición del módulo del Impact Mapping desde la vista de la página, plantilla, organismos, moléculas hasta llegar a los átomos, la unidad más pequeña.

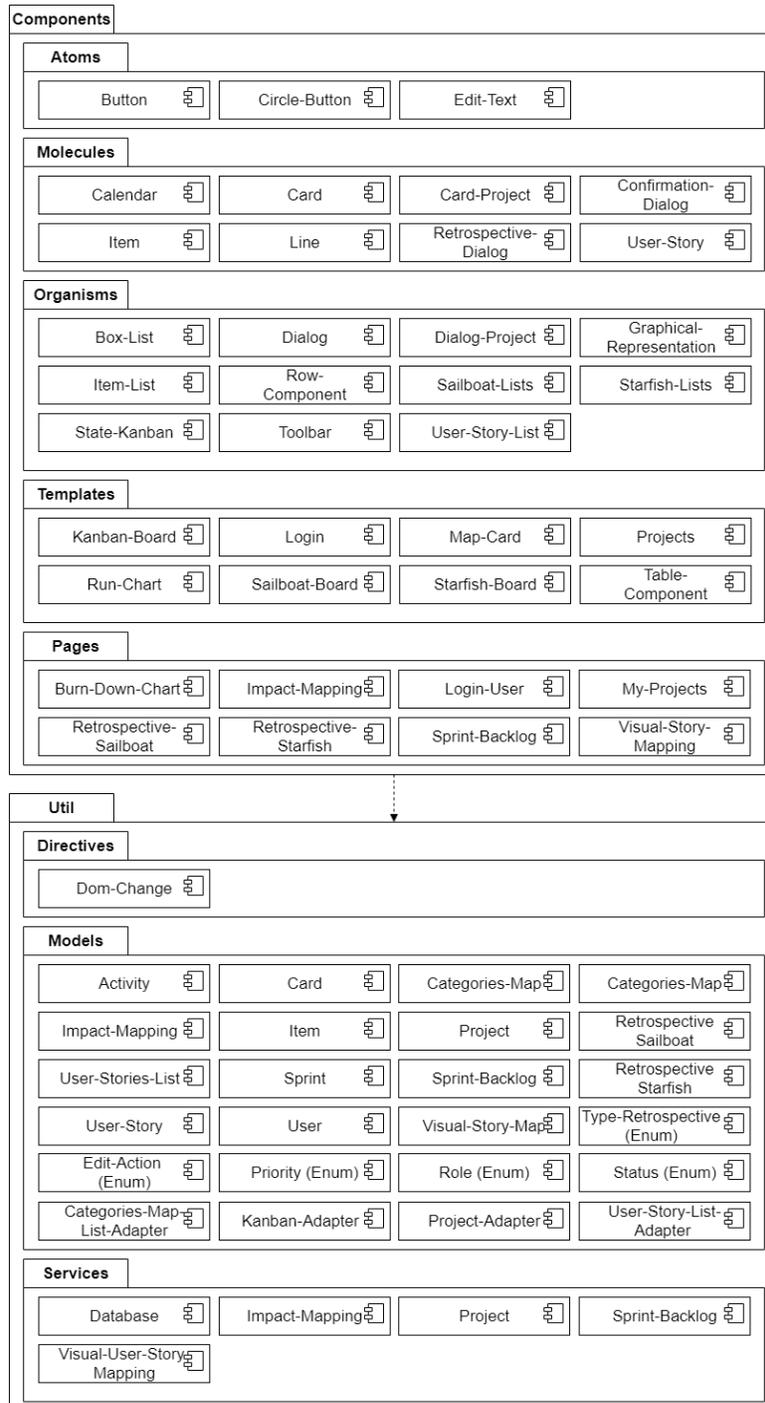
Figura 26

Diseño atómico del Impact Mapping del software.



Fuente: Elaboración propia con diagrams.net.

Por otro lado, esta misma categorización, se puede ver dentro del código en Angular, como lo muestra la Figura 27. El cual corresponde al Nivel 4 del modelo C4, y proporciona una vista de alto nivel de los componentes dentro del software. Los cuales están divididos en la jerarquía del diseño atómico, y otra carpeta que contiene las utilidades del sistema como las directivas, modelos y servicios. Donde la mayoría de los modelos son clases que contienen la abstracción de algún objeto que compone el JSON que se almacena en la base de datos, como ejemplifica la Figura 28, otros son enum, y otros contienen funciones.

Figura 27*Diagrama de código del prototipo.*

Fuente: Elaboración propia con diagrams.net.

Figura 28

Orden de los modelos que componen el JSON de la base de datos no relacional.

```

Project {
  id: string;
  name: string;
  users: User[{
    email: string;
    role: Role (ProductOwner, ScrumMaster, DeveloperTeam,);
  }];
  impactMapping: ImpactMapping {
    goals: CategoriesMap {
      id: string;
      title: string;
      nodes: Card[{
        id: string;
        content: string;
        x1: Number;
        y1: Number;
        x2: Number;
        y2: Number;
        isUpdating: EditAction (AddLine, Deleteline, DeleteCard, Disable, None, Edit);
        isDragging: boolean;
        nodes_id: string[];
      }];
    };
    actors: CategoriesMap;
    impacts: CategoriesMap;
    deliverables: CategoriesMap;
  };
  visualStoryMapping: VisualStoryMap {
    activities: Activity[{
      id: string;
      name: string;
    }];
    userStoryList: UserStoriesList[{
      id: string;
      activityId: string;
      priority: Priority (Must, Should, Could, wont);
      userStories: UserStory[{
        id: string;
        who: string;
        what: string;
        why: string;
        acceptanceCriterias?: Item[{
          id: string;
          content: string;
          status?: boolean;
        }];
        tasks?: Item[];
        estimation?: number;
        responsible?: string;
        activityId: string;
        status: Status (ToDo, Doing, Done);
        doAndDoingDate?: Date[];
        doneDate?: Date[];
      }];
    }];
  };
  sprintBacklog: SprintBacklog {
    sprints: Sprint[{
      id: string;
      objective: string;
      startDate: Date;
      finishDate: Date;
      todo: UserStory[];
      doing: UserStory[];
      done: UserStory[];
      daysExcluded: Date[];
      retrospectiveStarfish?: RetrospectiveStarfish{
        keepDoing: string;
        moreOf: string;
        startDoing: string;
        stopDoing: string;
        lessOf: string;
      };
      retrospectiveSailboat?: RetrospectiveSailboat{
        island: string;
        sail: string;
        anchor: string;
        rocks: string;
        sun: string;
        actionPlan: string;
      };
    }];
    typeRetrospective?: TypeRetrospective (RetrospectiveStarfish, RetrospectiveSailboat);
  };
}

```

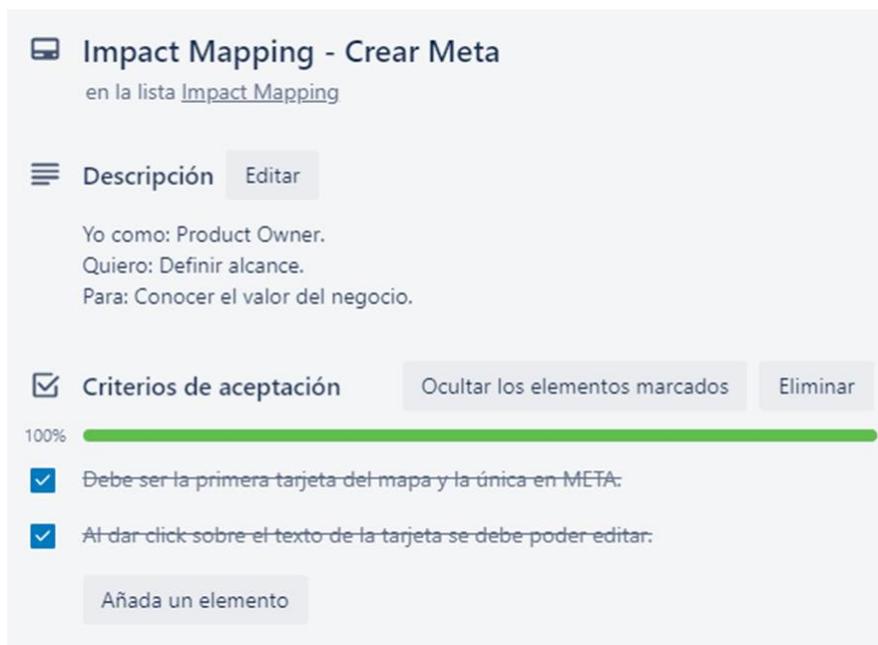
Fuente: Elaboración propia.

5.5.10. Pruebas del Software

En cada Sprint durante el desarrollo y solución de las historias de usuario, se verificaba que se cumpliera con los criterios de aceptación de cada una de estas, como se puede ver en la Figura 29. Para ello, se creaba un entorno local con el uso del Angular CLI, al ejecutar por consola en la ubicación de la carpeta del proyecto el comando: “ng serve” para activar el puerto lógico 4200 o “ng serve -open” que es igual a decir “ng serve -o” para abrir automáticamente el puerto 4200 en un navegador, y así poder interactuar y probar el código hecho, en el servidor web local que crea Angular CLI.

Figura 29

Ejemplo del testeo de una historia de usuario.

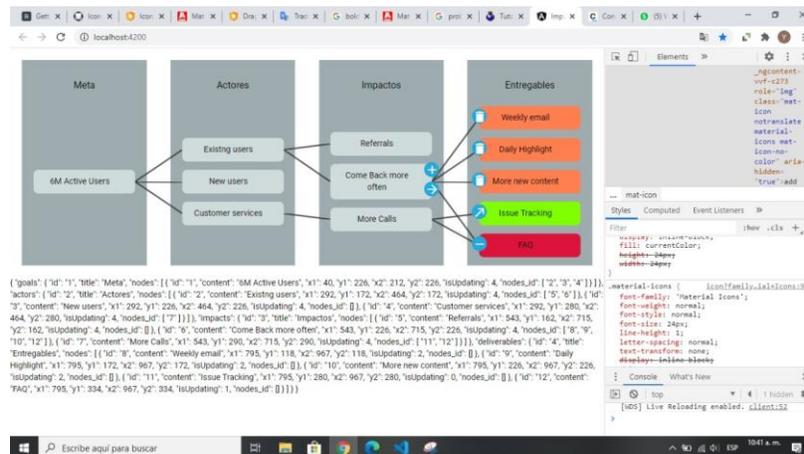


Fuente: Elaboración propia con Trello.

Así mismo, se probó en sistema operativo Windows 10 y Ubuntu con Deepin, que tuvieran el navegador Google Chrome, a fin de testear su funcionamiento, y el comportamiento al tener un cambio de resolución de pantalla, como se demuestra en la Figura 30 y 31.

Figura 30

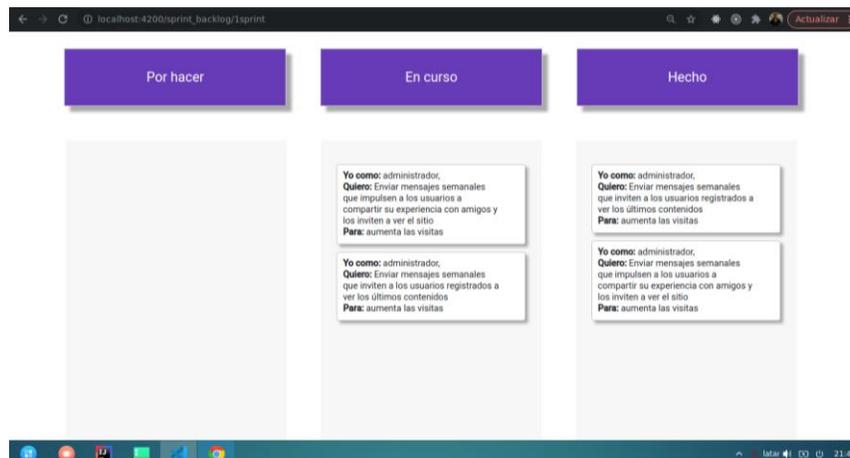
Evidencia de testeo en Windows 10.



Fuente: Elaboración propia.

Figura 31

Evidencia de testeo en Deepin.



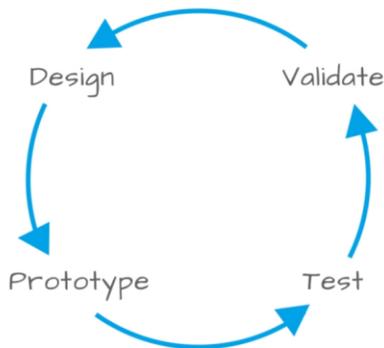
Fuente: Elaboración propia.

5.5.11. Validación del Software

Carl Sziebert y Kaushal Shah de “Google for Startups”, mencionan que para crear un prototipo es necesario, primero: diseñar el prototipo, segundo: construir el prototipo el prototipo de forma nativa, es decir escribir código (prototipar), tercero: probar prototipo en dispositivo, y cuarto: validar con usuarios reales para conocer como responderán a las funciones del aplicativo. Dentro de los beneficios de la creación de prototipos nativos es que se podrá obtener una experiencia realista al alcance de los usuarios durante la validación para recopilar comentarios valiosos, donde una buena experiencia de usuario va más allá de la apariencia de la aplicación puesto que significa “comprender como reaccionaran sus usuarios a la animación del diseño y las características de entrada de la experiencia, (...) integrando el proceso simple de prueba de prototipo de diseño y validación” (Google for Startups, 2016). El cual se ve en la Figura 32.

Figura 32

Ciclo de creación de un prototipo.



Nota: Ciclo propuesto por Carl Sziebert y Kaushal Shah de “Google for Startups”, que se compone de diseñar, prototipar, probar y validar, e iterar. Fuente: Google for Startups (2016).

Teniendo en cuenta, que la pregunta problema es: ¿Qué estrategia se puede emplear para una mejor adopción del marco de trabajo Scrum en un proyecto ágil?, la validación permite conocer si a través del uso de técnicas visuales que apoyan la teoría de Scrum es posible crear un prototipo que le permita a los equipos tener una experiencia funcional en el uso del marco de trabajo, a fin de comprobar si el aplicativo es lo suficientemente sólido.

Para ello se realiza el diseño del cuestionario evaluando la “Experiencia funcional en el uso de Scrum” respecto a ciertas dimensiones, que se subdividen en indicadores e ítems, como se puede visualizar en la Tabla 13.

Tabla 13

Cuestionario de la validación del prototipo de software.

Variable	Dimensiones	Indicadores	Ítems
Experiencia funcional en el uso de Scrum	Interfaz	Grado de intuición (Intuitiva)	La interfaz del prototipo es intuitiva: 5. Totalmente de acuerdo. 4. De acuerdo. 3. Ni de acuerdo ni en desacuerdo. 2. En desacuerdo. 1. Totalmente en desacuerdo.
		Grado de agradabilidad (Agradable)	La interfaz del prototipo es agradable de mirar: 5. Totalmente de acuerdo. 4. De acuerdo. 3. Ni de acuerdo ni en desacuerdo. 2. En desacuerdo. 1. Totalmente en desacuerdo.
	Estructura	Grado de facilidad de uso	El prototipo es de uso fácil: 5. Totalmente de acuerdo. 4. De acuerdo. 3. Ni de acuerdo ni en desacuerdo. 2. En desacuerdo. 1. Totalmente en desacuerdo.

Variable	Dimensiones	Indicadores	Ítems
		Grado de organización de los módulos	<p>La estructura de organización de los módulos es:</p> <p>Impact Mapping > User Story Mapping > Sprint Backlogs con sus respectivos Burn Down Chart y retrospectivas.</p> <p>Considera que la anterior estructura es:</p> <ol style="list-style-type: none"> 5. Muy clara. 4. Clara. 3. Normal. 2. Confusa. 1. Muy confusa.
Utilidad		Grado de utilidad	<p>El prototipo es muy útil:</p> <ol style="list-style-type: none"> 5. Totalmente de acuerdo. 4. De acuerdo. 3. Ni de acuerdo ni en desacuerdo. 2. En desacuerdo. 1. Totalmente en desacuerdo.
		Grado de satisfacción	<p>Como usuario, se encuentra satisfecho con las funcionalidades del prototipo:</p> <ol style="list-style-type: none"> 5. Totalmente de acuerdo. 4. De acuerdo. 3. Ni de acuerdo ni en desacuerdo. 2. En desacuerdo. 1. Totalmente en desacuerdo.
		Grado de recomendabilidad	<p>Como usuario, recomendaría este prototipo:</p> <ol style="list-style-type: none"> 5. Totalmente de acuerdo. 4. De acuerdo. 3. Ni de acuerdo ni en desacuerdo. 2. En desacuerdo. 1. Totalmente en desacuerdo.
Rendimiento		Grado de rendimiento	<p>El rendimiento del prototipo es:</p> <ol style="list-style-type: none"> 5. Optimo. 4. Bueno. 3. Regular. 2. Malo. 1. Muy malo.
		Gravedad de fallos	<p>El prototipo tiene una gravedad de fallos de:</p> <ol style="list-style-type: none"> 5. Imperceptibles.

Variable	Dimensiones	Indicadores	Ítems
			4. Insignificantes. 3. Notables. 2. Críticos. 1. Muy críticos.
Introducción de conceptos de Scrum	Grado de cumplimiento de los pilares de Scrum		El prototipo cumple con el pilar de Scrum de Transparencia (El proceso y trabajo es visible para todos los miembros del equipo): 5. Totalmente de acuerdo. 4. De acuerdo. 3. Ni de acuerdo ni en desacuerdo. 2. En desacuerdo. 1. Totalmente en desacuerdo.
			El prototipo cumple con el pilar de Scrum de Inspección (Evaluar el avance – Eventos de Scrum): 5. Totalmente de acuerdo. 4. De acuerdo. 3. Ni de acuerdo ni en desacuerdo. 2. En desacuerdo. 1. Totalmente en desacuerdo.
			El prototipo cumple con el pilar de Scrum de Adaptación (Cambiar la dirección del plan en caso de ser necesario – Eventos de Scrum): 5. Totalmente de acuerdo. 4. De acuerdo. 3. Ni de acuerdo ni en desacuerdo. 2. En desacuerdo. 1. Totalmente en desacuerdo.
		Grado de relación de los roles con los artefactos y técnicas	El prototipo de software permite evidenciar la relación que existe entre los roles del Scrum Team con los diferentes artefactos y técnicas: 5. Totalmente de acuerdo. 4. De acuerdo. 3. Ni de acuerdo ni en desacuerdo. 2. En desacuerdo. 1. Totalmente en desacuerdo.

Variable	Dimensiones	Indicadores	Ítems
		Grado de relación de los roles con los eventos	El prototipo de software permite evidenciar la relación que existe entre los roles del Scrum Team con los diferentes eventos: 5. Totalmente de acuerdo. 4. De acuerdo. 3. Ni de acuerdo ni en desacuerdo. 2. En desacuerdo. 1. Totalmente en desacuerdo.
		Grado de relación de los artefactos y técnicas con los eventos	El prototipo de software permite evidenciar la relación que existe entre los artefactos y técnicas con los eventos: 5. Totalmente de acuerdo. 4. De acuerdo. 3. Ni de acuerdo ni en desacuerdo. 2. En desacuerdo. 1. Totalmente en desacuerdo.

Fuente: Elaboración propia.

6. Resultados

6.1. Resultados del Desarrollo de Software del Prototipo

Las siguientes pantallas permiten una visión general del prototipo elaborado. La primera muestra el Home en la Figura 33. Donde el botón “Ingresar” genera la redirección a la aplicación creada en Auth0 para autenticación, como se evidencia en la Figura 34. Así mismo, la aplicación en Auth0 se puede ver en la Figura 35.

Figura 33

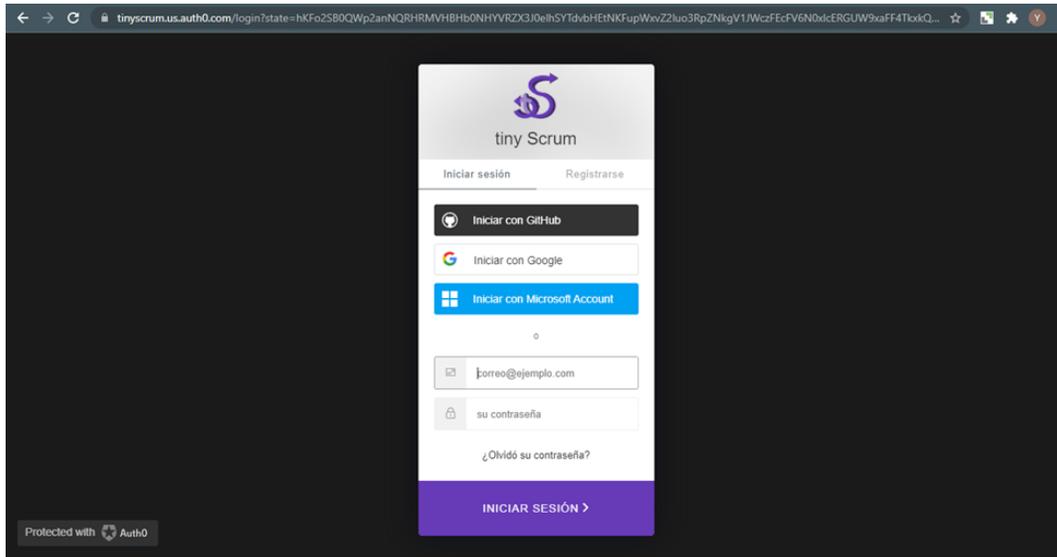
Home de Inicio de Sesión de tiny Scrum.



Fuente: Elaboración propia.

Figura 34

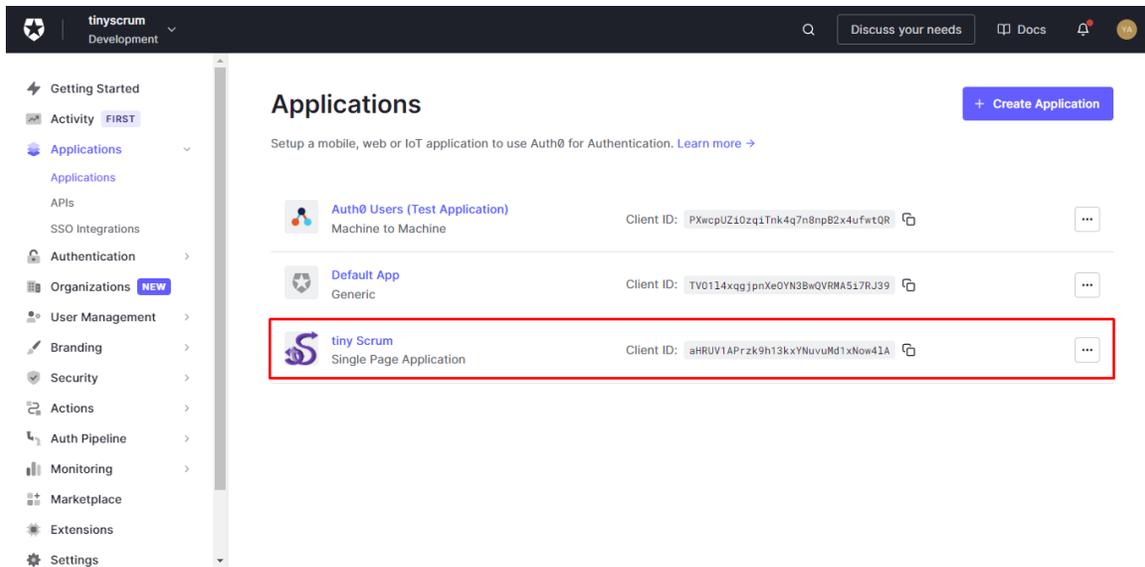
Login de tiny Scrum con Auth0.



Fuente: Elaboración propia.

Figura 35

Aplicación de tiny Scrum en Auth0.

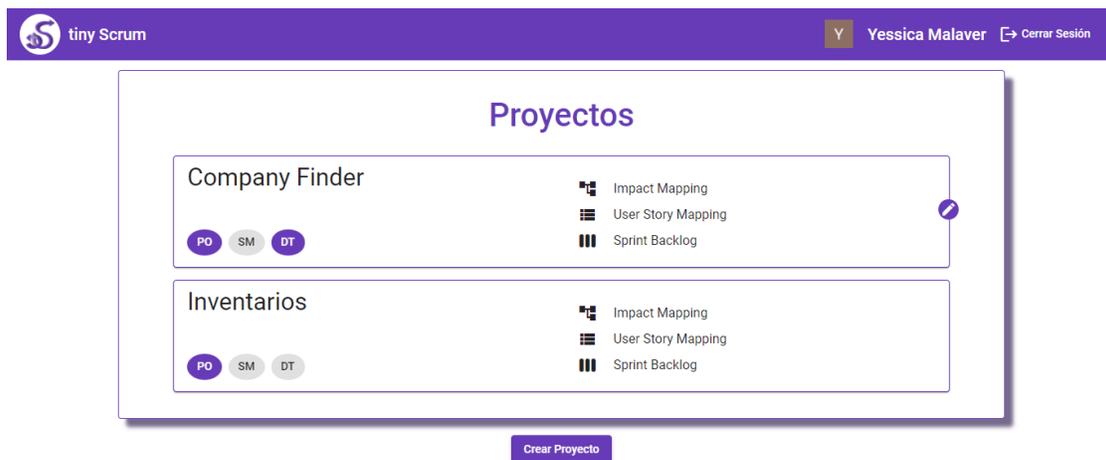


Fuente: Elaboración propia.

Una vez, el usuario se encuentre logueado, se redirigirá nuevamente al aplicativo desarrollado en Angular, como se ve en la Figura 36. Donde al dar click en “ Crear proyecto” se despliega un Dialog, un componente de Angular Material, como se ve en la Figura 37.

Figura 36

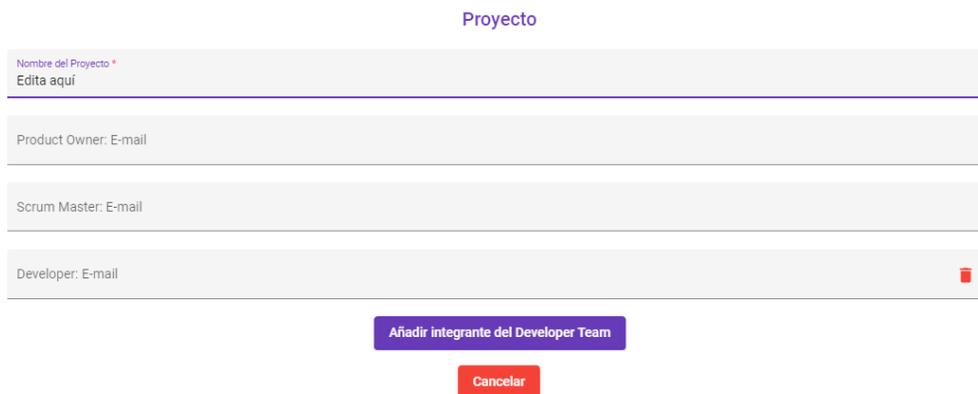
Home de proyectos de tiny Scrum.



Fuente: Elaboración propia.

Figura 37

Dialog de creación o edición de un proyecto.

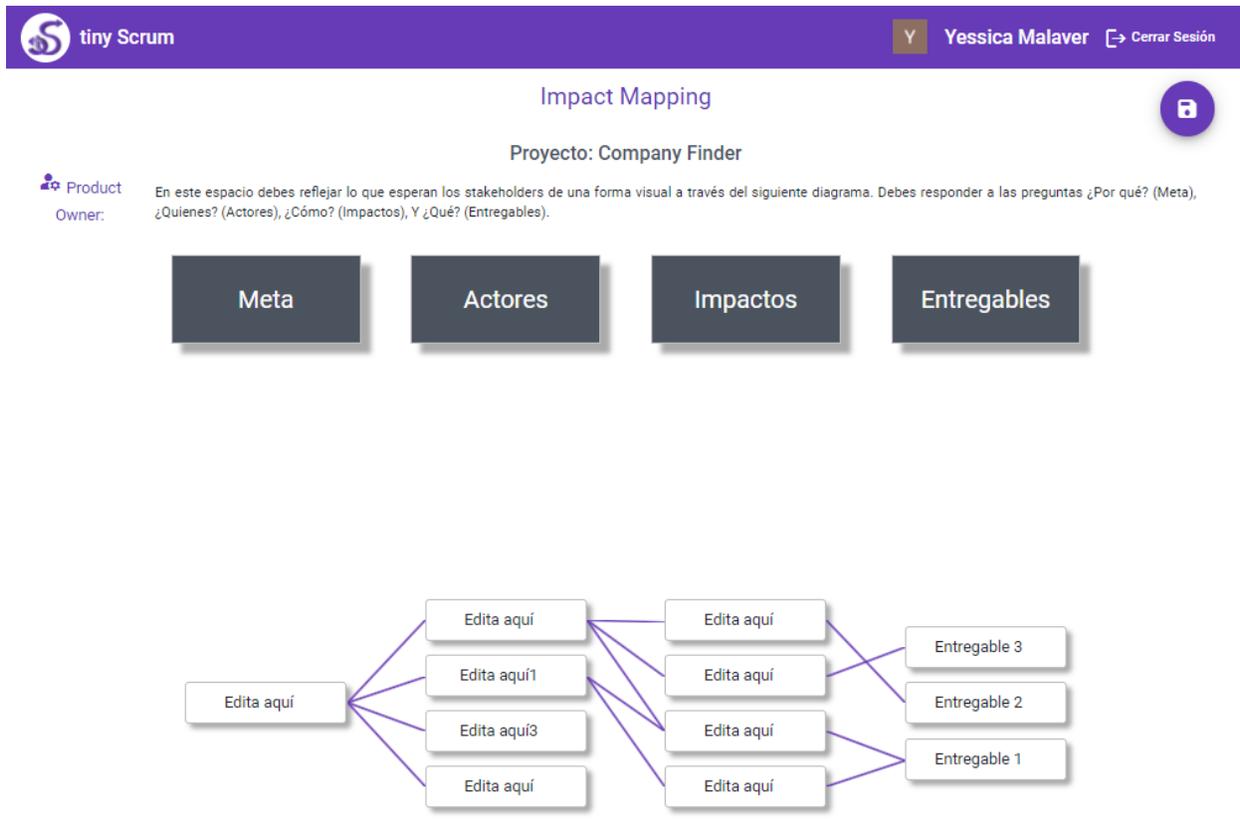


Fuente: Elaboración propia.

Así mismo, como se ve en el home de proyectos, cada proyecto tiene un módulo de Impact Mapping, donde al dar click redirige a la ventana que se ve en la Figura 38.

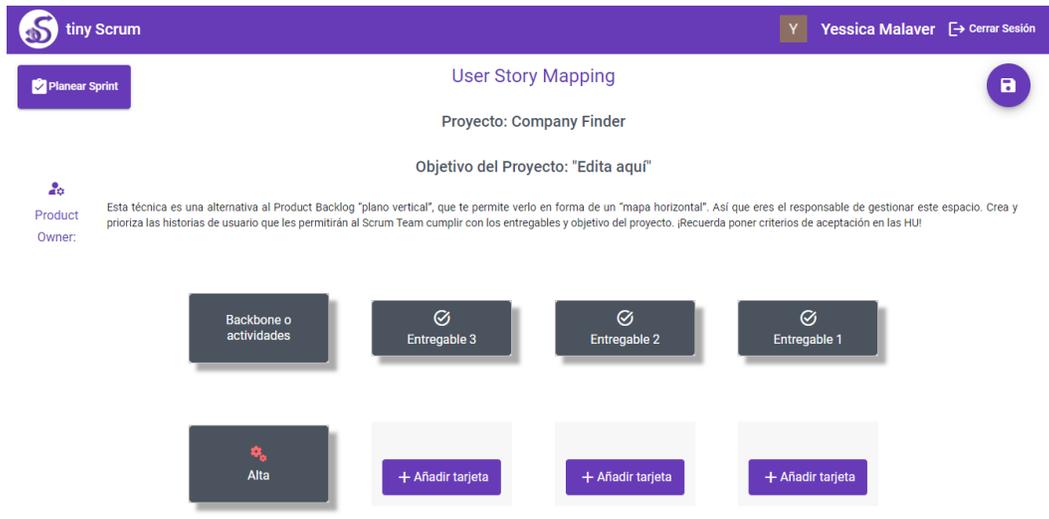
Figura 38

Módulo de Impact Mapping.

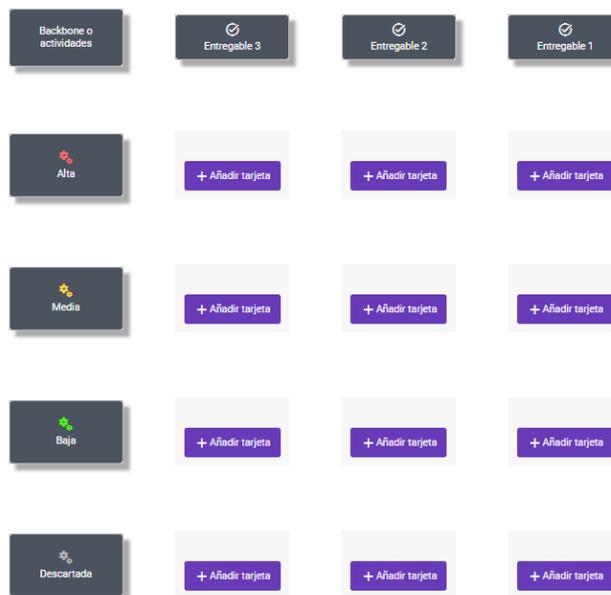


Fuente: Elaboración propia.

Por otro lado, si desde el home de proyectos se da click en un User Story Mapping, se podrá ver la pantalla de la Figura 39. En donde los entregables del Impact Mapping se convierten en las actividades o Backbone del User Story Mapping. El cual propone 4 niveles de prioridad según la priorización MoSCoW, de alta, media, baja y descartada, como se evidencia en la Figura 40.

Figura 39*Módulo de User Story Mapping.*

Fuente: Elaboración propia.

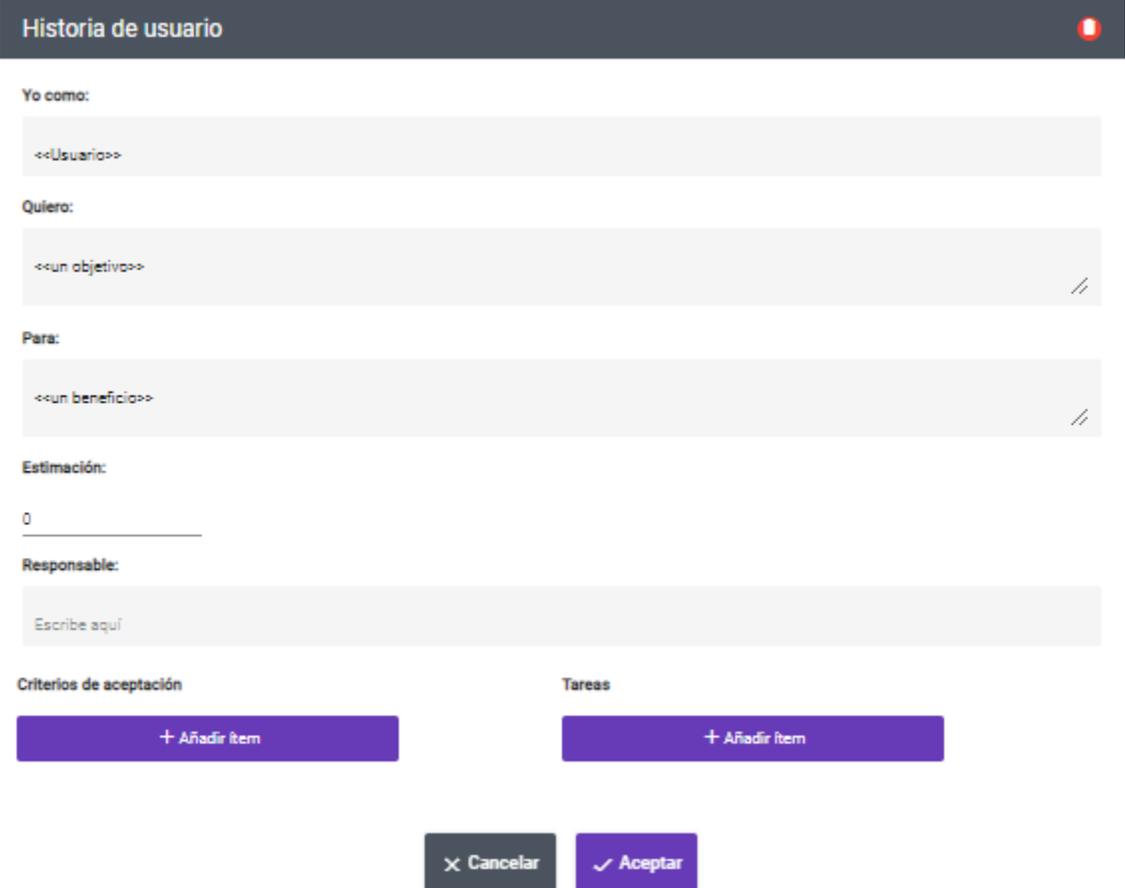
Figura 40*Vista completa del User Story Mapping.*

Fuente: Elaboración propia.

Si en el tablero del User Story Mapping se da click en añadir tarjeta, se añadirá una historia de usuario, donde al dar click en editar, permite ver la plantilla propuesta para la misma como se puede ver en la Figura 41, donde se divide el campo de descripción en Yo como, Quiero, Para, se permite añadir la estimación de forma numérica y un responsable si lo desea. Y por último, permite añadir criterios de aceptación por el Product Owner, y tareas que son empleadas por el Developer Team.

Figura 41

Historia de usuario en tiny Scrum.



Historia de usuario

Yo como:

<<Usuario>>

Quiero:

<<un objetivo>> //

Para:

<<un beneficio>> //

Estimación:

0

Responsable:

Escribe aquí

Criterios de aceptación

+ Añadir ítem

Tareas

+ Añadir ítem

× Cancelar

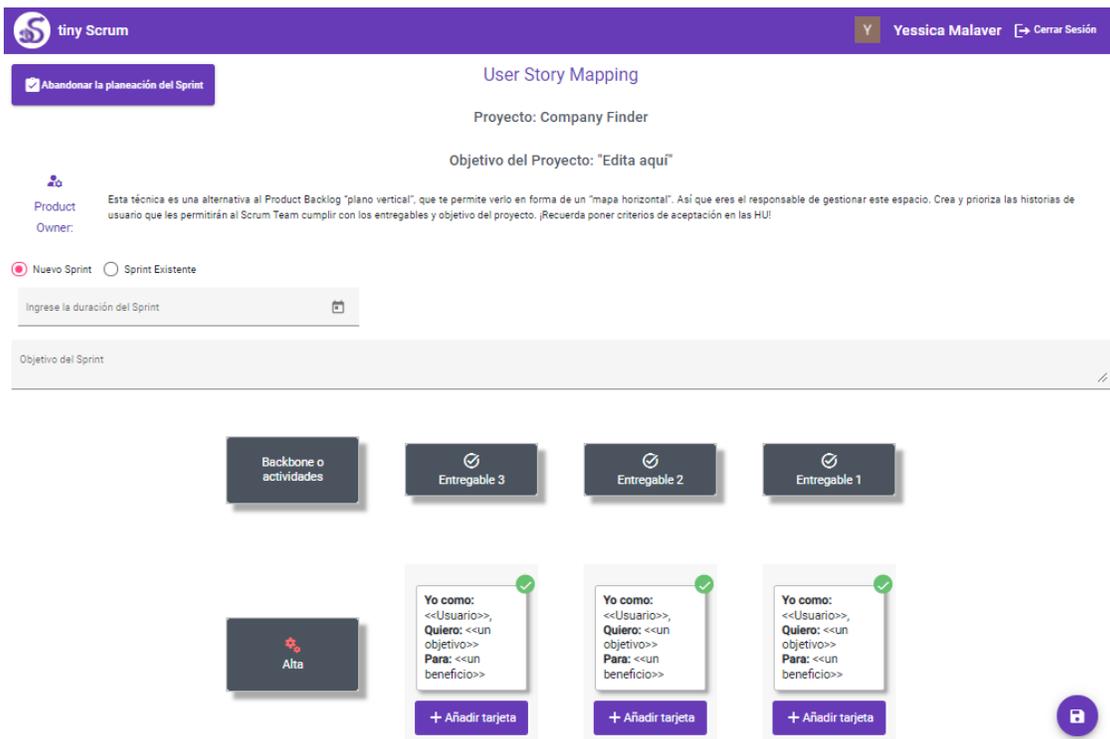
✓ Aceptar

Fuente: Elaboración propia.

Después de tener varias historias de usuario, se puede realizar la planeación del Sprint dentro del mismo tablero del User Story Mapping, al dar click en “Planear Sprint”, pudiéndose elegir si es uno nuevo o es uno existente, si es existente se elige uno de la lista, y posteriormente indicar o modificar la duración y objetivo del mismo, además de seleccionar o quitar historias de usuario al Sprint y dar click en guardar, como se ve en la Figura 42.

Figura 42

Planeación de un Sprint en tiny Scrum.



Fuente: Elaboración propia.

Los Sprint creados, se pueden visualizar debajo del tablero del User Story Mapping, como se puede ver en la Figura 43.

Figura 43

User Story Mapping con los Sprint existentes.

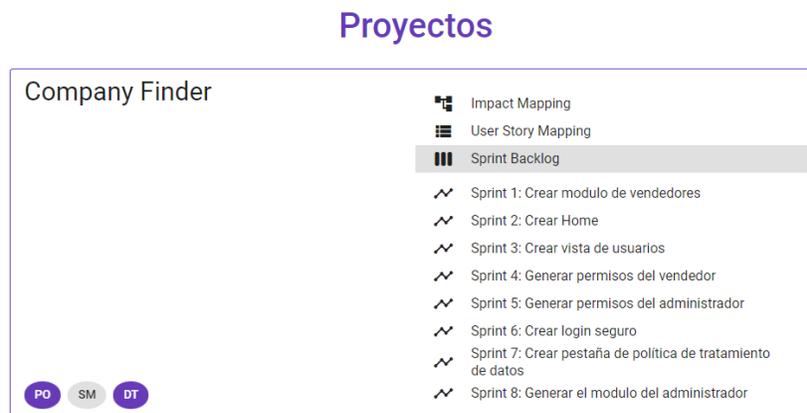


Fuente: Elaboración propia.

Por otro lado, para gestionar el Sprint, desde la ventana de proyectos se debe dar click en “Sprint Backlog”, donde se desplegará un select con los Sprints existentes como se ve en la Figura 44.

Figura 44

Sprint existentes y Sprint Backlog.

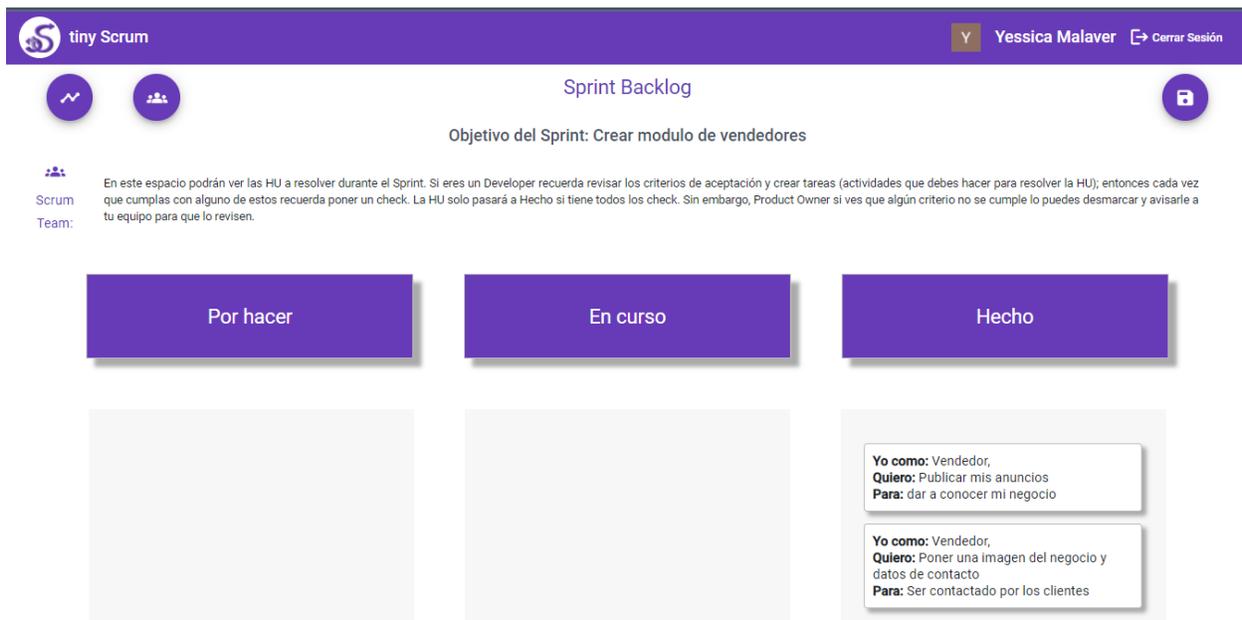


Fuente: Elaboración propia.

Si se elige uno de los Sprints en la pantalla anterior, se redirige al Módulo de Sprint Backlog como se ve en la Figura 45, el cual permite ver un tablero tipo Kanban con 3 listas: Por hacer, En curso y Hecho. En las cuales, se ubican las historias de usuario del Sprint según el estado de las mismas. Así mismo, dentro de este tablero se encuentran en la parte superior izquierda dos botones, el primero redirige al burn down chart y el segundo a la Retrospectiva.

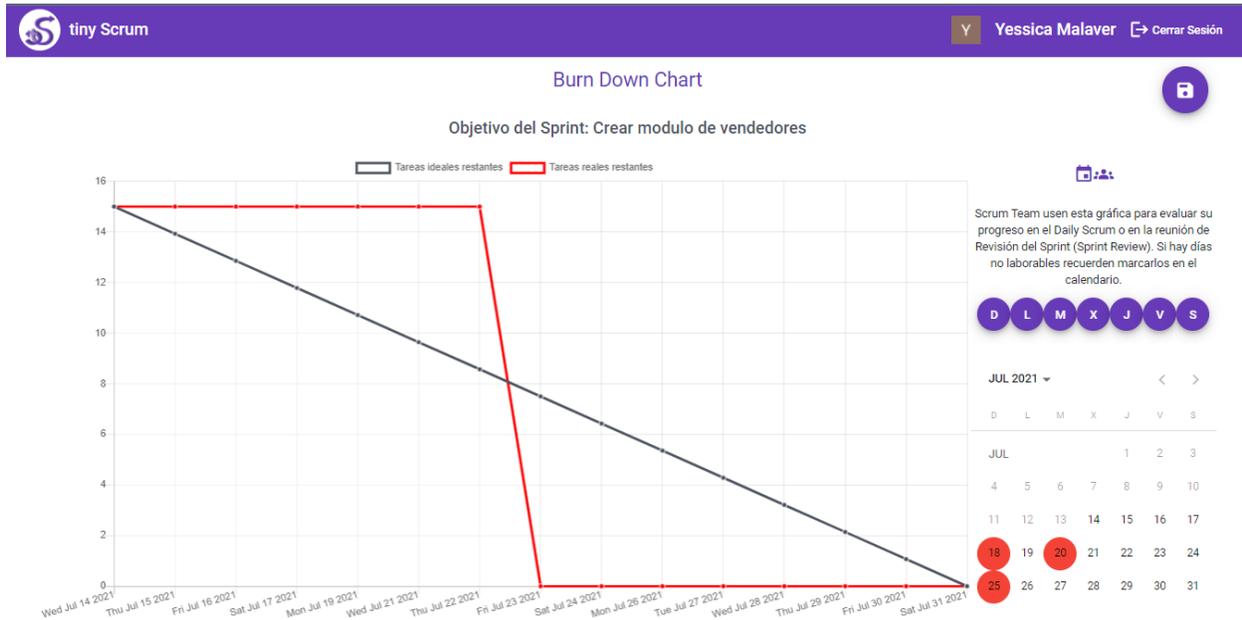
Figura 45

Modulo de Sprint Backlog.



Fuente: Elaboración propia.

Si se da click en el burn down chart, esta acción redirige a la ventana que se ven en la Figura 46, que permite ver la gráfica representando el progreso del Sprint, según la sumatoria de los puntos de estimación de las historias de usuario respecto al tiempo en días del Sprint. A través de 2 líneas, una representa el tiempo ideal y otra el tiempo real. Así mismo, como puede haber días no laborales, el calendario permite seleccionar aquellos días.

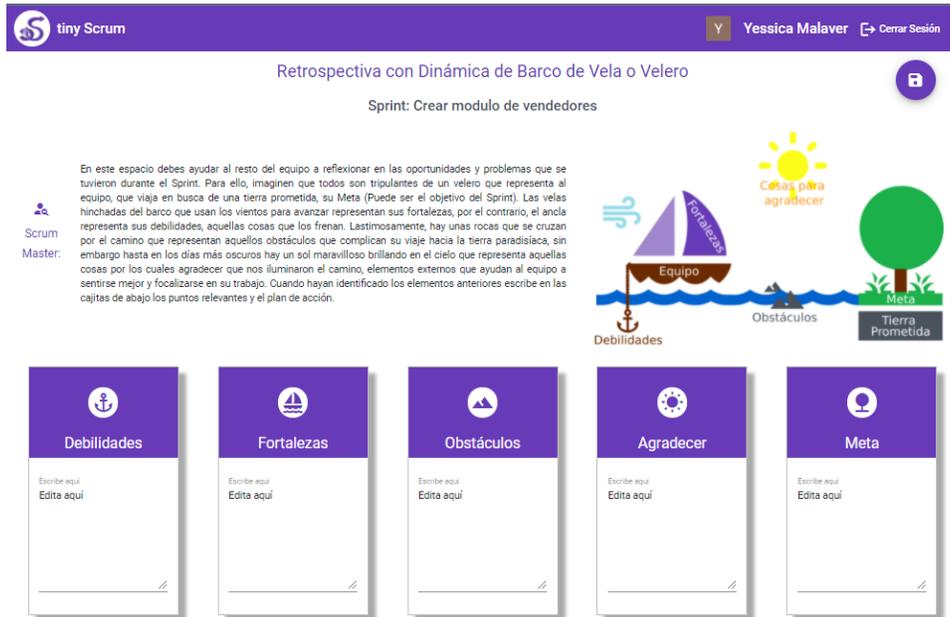
Figura 46*Módulo de Burn Down Chart.*

Fuente: Elaboración propia.

Por otro lado, si se da click en el botón de retrospectiva, el usuario debe elegir entre una de las dos técnicas disponibles, si elige la dinámica del barco de vela, se mostrará la ventana de la Figura 47. Donde el equipo debe identificar 5 ítems que se evidenciaron durante el Sprint: Debilidades, fortalezas, obstáculos, cosas por las cuales agradecer y la meta, a fin de tener un panorama general de esa información y construir un plan de acción como se ve en la Figura 48.

Figura 47

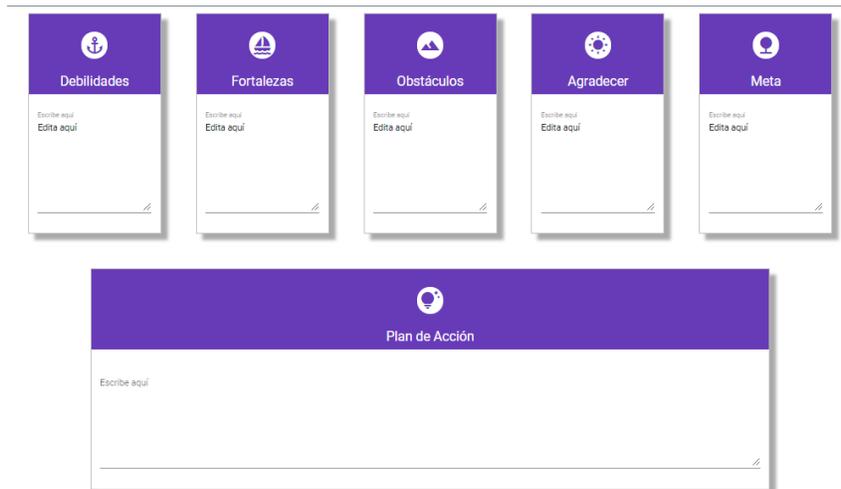
Módulo de retrospectiva: dinámica del barco de vela.



Fuente: Elaboración propia.

Figura 48

Campos de la dinámica del barco de vela.



Fuente: Elaboración propia.

Por otro lado, como en Figura 49 si se elige la dinámica de la estrella de mar, los ítems a tratar son: Sigue haciendo, Mas de, Empezar a hacer, Dejar de hacerlo y Menos de.

Figura 49

Módulo de retrospectiva: dinámica de la estrella de mar.

The screenshot displays the 'Retrospectiva con Dinámica de Estrella de Mar' interface. At the top, it shows the 'tiny Scrum' logo, the user 'Yessica Malaver', and a 'Cerrar Sesión' button. The main heading is 'Retrospectiva con Dinámica de Estrella de Mar' and the sprint is 'Sprint: Crear Home'. A central star diagram has five points with icons and labels: a plus sign for 'Más de ...', a play button for 'Empezar a hacer ...', a square for 'Dejar de hacerlo ...', a minus sign for 'Menos de ...', and a refresh icon for 'Sigue haciendo ...'. To the left of the star is a 'Scrum Master' icon and a paragraph of instructions. Below the star are five vertical input boxes, each with a header and a text area:

- Sigue haciendo**: Escribe aquí, Edita aquí
- Más de**: Escribe aquí, Edita aquí
- Empezar a hacer**: Escribe aquí, Edita aquí
- Dejar de hacerlo**: Escribe aquí, Edita aquí
- Menos de**: Escribe aquí, Edita aquí

Fuente: Elaboración propia.

6.2. Resultados de la Validación del Prototipo

Para realizar la validación del prototipo se tiene en cuenta el apartado de ítems de la Tabla 13. El cual se aplicó a completitud en la validación realizada con estudiantes, y de forma parcial en la validación con expertos, como se evidencia a continuación.

6.2.1. Validación del Prototipo con Estudiantes

El instrumento se aplicó a 23 estudiantes de Ingeniería de Sistemas de la Universidad de Cundinamarca sede Fusagasugá, donde 5 personas son de octavo semestre y 18 de noveno semestre. Donde la dinámica consistió en mostrarles el software y su funcionamiento para que finalmente contestaran el cuestionario haciendo uso de Microsoft Forms. A continuación se pueden ver las preguntas con sus respectivas respuestas y análisis. Así mismo, el formulario se puede visualizar a través del siguiente link:

https://forms.office.com/Pages/ShareFormPage.aspx?id=oGfaB0MfjE6Xf1-ItkcO5rvYU_XtjSVFuEMFQp7MgJZUNEs1RlpFM0xPVFAzV081ODRWNjRENIVRQy4u&sharetoken=9re5QulzJsBHGqr2X3hB

6.2.1.1. Interfaz de Usuario. Los participantes indicaron que estaban totalmente de acuerdo en un 60.9% que el prototipo contaba con una interfaz de usuario agradable, y el 39.1% restante también estaban de acuerdo. Así mismo, respecto a si la interfaz es agradable de mirar 56.5% estuvieron totalmente de acuerdo y 43.5% de acuerdo, en ambas preguntas no hubo opiniones negativas, como se evidencia en la Figura 50.

Figura 50

Interfaz de usuario: Grado de intuición y agradabilidad.

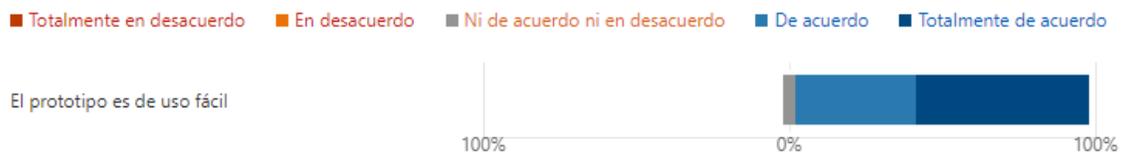


Fuente: Microsoft Forms.

6.2.1.2. Estructura. Ante la pregunta si el prototipo es de uso fácil solo una persona contestó que ni de acuerdo ni en desacuerdo equivalente al 4.3% de los encuestados, de resto 9 contestaron que estaban de acuerdo (39.1%), y 56.5% más de la mitad contestó que estaban totalmente de acuerdo, como se ve en la Figura 51.

Figura 51

Estructura: Grado de facilidad de uso.



Fuente: Microsoft Forms.

Por otro lado, respecto a la organización de los módulos, 8.7% (2 participantes) opinan que la estructura es normal, 52.2% indican que es clara (12 participantes), y el resto indica que es muy clara, equivalente al 39.1%, por lo que como muestra la Figura 52 no hay comentarios negativos para la pregunta “La estructura de organización de los módulos es: Impact Mapping > User Story Mapping > Sprints Backlogs con sus respectivos burn down chart y retrospectivas. Considera que la anterior estructura es:”.

Figura 52

Estructura: Grado de organización de los módulos.



Fuente: Microsoft Forms.

6.2.1.3. Utilidad. Para esta dimensión se analizan 3 ítems, como se puede ver en la Figura 53, el primer ítem preguntaba si el prototipo es muy útil, obteniéndose un porcentaje de 78.3% de los participantes totalmente de acuerdo, y 21.7% de acuerdo, lo cual indica el gran grado de utilidad captada por los encuestados. El segundo ítem averiguaba si el usuario se encontraba satisfecho con el prototipo, con 34.8% de los participantes de acuerdo, y el 65.2% restante totalmente de acuerdo. Para finalmente, conocer si recomendarían el prototipo, obteniéndose que todos los participantes lo harían, con un porcentaje de participantes del 73.9% totalmente de acuerdo, y 26.1% de acuerdo.

Figura 53

Utilidad: Grado de utilidad, de satisfacción y de recomendabilidad.



Fuente: Microsoft Forms.

6.2.1.4. Rendimiento. Esta dimensión se compone de 2 ítems, el primero indaga respecto al rendimiento del prototipo como se ve en la Figura 54. Donde fue calificado con 43.5% de los participantes como bueno, y 56.5% como óptimo, lo cual indica que es excelente.

Figura 54*Rendimiento: Grado de rendimiento.*

Fuente: Microsoft Forms.

Por otro lado, respecto a la gravedad de fallos, estos son considerados en 43.5% como insignificantes y por 56.5% de los participantes como imperceptibles. Lo cual indica que aunque la aplicación depende del navegador y de otros intermediarios por lo que puede experimentar fallos, aunque estos aparecen circunstancialmente no afectan la experiencia del usuario de forma crítica, como se ve en la Figura 55.

Figura 55*Rendimiento: Gravedad de fallos.*

Fuente: Microsoft Forms.

6.2.1.5. Pilares De Scrum. En la dimensión relacionada con los conceptos de Scrum según la Figura 56, se evidencia que el prototipo cumple con cada uno de los pilares del marco de trabajo, puesto que los participantes a nivel de transparencia consideran que cumple en un 69.6% totalmente de acuerdo, y 30.4% de acuerdo; para adaptación 60.9% totalmente de acuerdo y 39.1% de acuerdo; y para adaptación 65.2% totalmente de acuerdo y 34.8% de acuerdo.

Figura 56

Pilares de Scrum: Grado de cumplimiento.



Fuente: Microsoft Forms.

6.2.1.6. Reglas De Scrum. En la Figura 57 se evalúa si se evidencia la relación entre los diferentes roles, artefactos y eventos dentro del prototipo, donde más del 52% se encuentra totalmente de acuerdo en los diferentes ítems, 82.6% para la relación entre roles con los artefactos y técnicas, 69.6% para la relación entre los roles con los eventos y 52.2% para relación entre artefactos y técnicas con los eventos. Y el resto de los porcentajes estuvieron de acuerdo los participantes respectivamente. Lo cual indica el logro de los objetivos del prototipo.

Figura 57

Reglas de Scrum: Grado de relación.



Fuente: Microsoft Forms.

Finalmente, durante el proceso de validación durante sesiones en su mayoría por Teams, uno de los participantes indico en el chat: “Este prototipo me parece que será una herramienta que va a facilitar la comprensión y la implementación del marco de trabajo Scrum. Puede abarcar perfectamente desde un ámbito académico o de capacitación (sentí recordar muchas cosas con esta explicación) hasta un ambiente laboral en toda regla, es dinámica, agradable a nivel visual y muy intuitiva respecto a su uso y objetivo sin hablar de la enorme utilidad de la recolección de las diferentes técnicas mostradas. Es un trabajo excelente”. Lo cual indica y soporta los resultados positivos obtenidos durante la validación, puesto que todos los participantes estarían de acuerdo en recomendarla por lo que los esfuerzos por crear y construir el prototipo para la adopción de Scrum no fueron en vano.

Así mismo, otro de los participantes que se encuentra laborando en un entorno empresarial comento: “Yo he manejado algo similar que se llama Azure DevOps y el tablero Kanban, es como lo más cercano a lo que se está presentando, pero este prototipo esta super completo, de verdad eso si agiliza mucho y se necesita bastante, porque ocurre que en varios casos no usan un tablero de tareas o esa gestión de Scrum y es cuando se ponen retrasos, o muchos desordenes e impedimentos”, este punto de vista resalta la importancia de la implementación de las técnicas planteadas en el prototipo para una buena gestión de proyectos.

6.2.2. Validación del Prototipo con Expertos

Durante este proceso se hizo uso de la validación por juicio de experto, a fin de conocer si el prototipo posee una correcta estructura, si es útil y si lo recomendarían, además de si cumple con los pilares y reglas de Scrum, para ello se usa parte del cuestionario general que se muestra

en la Tabla 13, donde se hace uso de las dimensiones de: Estructura (Únicamente el indicador de grado de organización de los módulos), utilidad e introducción de conceptos de Scrum. A través de un formulario en Microsoft Forms, que se puede visualizar a través del siguiente link:

https://forms.office.com/Pages/ShareFormPage.aspx?id=oGfaB0MfjE6Xf1-ItkcO5rvYU_XtjSVFuEMFQp7MgJZUNVc1WU85N1RUTkdMRkxJNDZVOFY5UDBWTi4u&sharetoken=NWkwAWOJI7cP0omQRPdW

Se aplicó a 3 participantes parte del Developer Team en la actualidad sin embargo uno de ellos se ha desempeñado como Product Owner. Como se ve en la Figura 58 todos poseen entre 4 a 5 años de experiencia, cuentan con un título en ingeniería de sistemas o sus profesiones afines, certificados en Scrum, e incluso uno de ellos tiene especialización en Scrum.

Figura 58

Perfil de los expertos.



Fuente: Microsoft Forms.

Los participantes contestaron el cuestionario después de ver un video que describía al prototipo y su funcionamiento, se puede acceder a este través del siguiente link:

<https://www.youtube.com/watch?v=YtuGwRUfv2c>, los ítems tratados se analizan a continuación.

6.2.2.1. Estructura. Teniendo en cuenta que la estructura de organización de los módulos en el prototipo es: Impact Mapping > User Story Mapping > Sprint Backlogs con sus respectivos burn down chart y retrospectivas, los expertos consideraron que es 33.3% normal y 66.7% muy clara como se puede ver en la Figura 59, lo cual indica que el proceso de diseño del prototipo fue correcto a nivel estructural y modular.

Figura 59

Estructura: Grado de organización de los módulos (Expertos).



Fuente: Microsoft Forms.

6.2.2.2. Utilidad. En esta dimensión se obtuvo como resultado sorprendente que unánimemente los expertos consideran que el prototipo es muy útil, se encuentran satisfechos con las funcionalidades y lo recomendarían en un 100%, como se evidencia en la Figura 60. Lo cual significa, que el prototipo posee una gran utilidad en la gestión de proyectos de Scrum.

Figura 60

Utilidad: Grado de utilidad, de satisfacción y de recomendabilidad (expertos).



Fuente: Microsoft Forms.

6.2.2.3. Pilares De Scrum. Como expertos con certificados en Scrum o incluso especializaciones, los participantes conocen muy bien los pilares de Scrum de transparencia, inspección y adaptación, por lo que si el prototipo guía en el uso de este marco de trabajo con mucha más razón debe cumplir con cada uno de estos pilares, y de acuerdo con la opinión de los expertos el prototipo lo hace sin falta en un 100%, como se evidencia en la Figura 61.

Figura 61

Pilares de Scrum: Grado de cumplimiento (expertos).



Fuente: Microsoft Forms.

6.2.2.4. Reglas de Scrum. Entre los objetivos de este proyecto está “diseñar la relación de los artefactos y técnicas con su respectivo rol”, por lo que el primer ítem de este indicador indaga acerca de si se cumplió o no con este objetivo, por lo que se obtuvo como respuesta que el prototipo efectivamente cumple con esta meta según la opinión de los expertos que están 33.3% de acuerdo y 66.7% totalmente de acuerdo en que el prototipo de software permite evidenciar la relación que existe entre los roles de Scrum y los diferentes artefactos y técnicas, como se puede ver en la Figura 62.

El segundo ítem indaga acerca de la relación que existe entre los roles y eventos de Scrum, donde los expertos estuvieron 33.3% de acuerdo y 66.7% totalmente de acuerdo en que el prototipo permite evidenciar esta relación, y el tercer ítem pregunta si en el prototipo se evidencia la relación que existe entre los artefactos y técnicas con los eventos, donde los expertos están 33.3% de acuerdo y 66.7% totalmente de acuerdo en que el prototipo cumple con este criterio. Como se ve en la en la Figura 62.

Figura 62

Reglas de Scrum: Grado de relación (expertos).

El prototipo de software permite evidenciar la relación que existe entre:

[Más detalles](#)



Fuente: Microsoft Forms.

6.2.2.5. Comentarios Y Visión General De Los Expertos. Por último, en el formulario se dejó una pregunta abierta, que decía: “Por favor con sus propias palabras indique su opinión general acerca del prototipo, como aspectos positivos, recomendaciones u otro tipo de comentario que considere agregue valor en la actual validación (opcional)”, los comentarios recolectados se pueden visualizar en la Tabla 14.

Tabla 14*Comentarios de los expertos.*

Experto	Comentario
1	Para ser un prototipo está muy completo dado que cubre las ceremonias y eventos más importantes de un proceso de trabajo Scrum.
2	Excelente prototipo, se ve fácil e intuitivo. En mi concepto está listo para usar y recibir el feedback de los usuarios.
3	Muy buen prototipo, sería mejor si aparte del mensaje que indique según el rol quien debe llenar la información, también hallan accesos de solo lectura o edición según aplique al artefacto.

Fuente: Elaboración propia.

Los comentarios de los expertos son bastante gratificantes, pues implica que el prototipo aunque incluye en su mayoría las reglas de Scrum sigue siendo fácil e intuitivo, y aunque es una primera versión puede ser empleado por los usuarios en sus proyectos de Scrum. Además, uno de ellos propone una recomendación de conceder accesos de solo lectura o edición según el rol o artefacto que aplique, la cual puede ser tenida en cuenta para futuras versiones del prototipo.

6.2.3. Análisis General de la Validación

Los resultados de la validación del prototipo demostraron que el prototipo cumple con sus objetivos iniciales, pues tanto estudiantes como expertos consideran que la aplicación es de uso fácil y cuenta en su mayoría con una estructura clara, lo cual indica que los artefactos y técnicas identificados facilitan la adopción de Scrum de los equipos ágiles en sus proyectos.

Así mismo, el diseño e integración de los diferentes roles, artefactos, técnicas y eventos dentro del prototipo fue el ideal, pues estudiantes y expertos estuvieron de acuerdo en que estas relaciones se evidenciaban dentro del aplicativo.

Los comentarios realizados por parte de algunos estudiantes y por los expertos, además de la entera validación realizada confirman que el prototipo es la respuesta a la pregunta ¿Qué estrategia se puede emplear para una mejor adopción del marco de trabajo Scrum en un proyecto ágil? Pues los participantes independiente de su grado de experiencia en Scrum, están de acuerdo en que el prototipo es útil y se encuentran satisfechos con sus funcionalidades, razón por la cual lo recomendarían, además de ser catalogado como intuitivo y fácil de usar, muy completo incluso en comparación de otras plataformas.

7. Recomendaciones

Debido a que es un prototipo el objeto del actual estudio, el ciclo de creación de un prototipo propuesto por Carl Sziebert y Kaushal Shah de “Google for Startups”, propone que al terminar la validación se debe volver a la fase de diseño, a fin de encontrar una propuesta de valor sólida y el punto óptimo que sea de total aceptación por parte de los usuarios, teniendo como retroalimentación la validación realizada.

Entre una de las funcionalidades que se podría añadir en versiones futuras del prototipo propuesta por uno de los expertos que validaron el prototipo es que “sería mejor si aparte del mensaje que indique según el rol quien debe llenar la información, también hallan accesos de solo lectura o edición según aplique al artefacto”, por lo que en una futura versión del prototipo sería apropiado tener la opción de implementar permisos de acceso de lectura o edición según la función que ocupa un rol de Scrum dentro de un artefacto o técnica, por ejemplo que la única persona que puede editar el Impact Mapping y el User Story Mapping junto con la planeación de Sprint sea el Product Owner como se especifica en las ayudas visuales ya que es el encargado de gestionarlos, y que dentro del Sprint Backlog el Developer Team pueda editar únicamente las tareas de las historias de usuario. De tal manera que con el bloqueo de funcionalidades de edición el usuario tenga otra forma de conocer cuáles son las funciones que le compete a su rol. Pero conservándose la opción donde todos tienen la posibilidad de editar, pues esto permite que en una etapa inicial todos puedan conocer las funcionalidades que ofrece cada técnica en colaboración de las ayudas visuales, y a través de estas puedan conocer cual rol es el encargado de cada acción, pues en proyectos avanzados donde hay bloqueo de funcionalidades según su rol, muchos integrantes del equipo de un proyecto no conocen quien es el encargado de cierta

función, en cambio al tener acceso a una sesión abierta es mucho más fácil para el Scrum Master guiar a su equipo en la comprensión del marco de trabajo y las herramientas empleadas.

Así mismo, al tratarse de la primera versión de este prototipo, es posible que se encuentren errores que no fueron detectados durante el periodo de desarrollo o de pruebas, por tanto deben corregirse en futuras versiones, a fin de pasar la versión actual o futura del aplicativo a producción para que pueda ser consumido por varios usuarios interesados en cómo aplicar Scrum correctamente, pues no se abarcó esta etapa en el proyecto ya que la intención de este proyecto era demostrar la factibilidad de diseñar y desarrollar un prototipo que permitiera la adopción de Scrum.

Por otro lado, en caso de que llegue a haber futuras versiones, es necesario que estas se encuentren actualizadas con la guía de Scrum de Ken Schwaber y Jeff Sutherland, puesto que es un marco de trabajo que puede estar sujeto a cambios y actualizaciones.

Y por último, se recomienda realizar una investigación de tipo cuasiexperimental o experimental que permita conocer la efectividad que tiene el prototipo en la claridad de conocimientos acerca de Scrum, a través de un diseño pretest/posttest con dos grupos, uno de control y otro experimental, donde a ambos grupos se le aplique un mismo instrumento a fin de conocer los conocimientos que tienen de Scrum los participantes, y posteriormente el grupo experimental interactúa con el prototipo, mientras al grupo de control se le enseña Scrum con el método tradicional de clases magistrales, después de un periodo de tiempo, ambos grupos se exponen a la misma u otra prueba de conocimientos a fin de conocer si con el uso del prototipo se obtienen resultados más favorables que en comparación con una clase tradicional.

8. Conclusiones

El prototipo diseñado con el uso de técnicas visuales que apoyan la teoría de Scrum, desarrollado y testeado en Angular con el empleo de Auth0 y Firebase, y validado a través de estudiantes de Ingeniería de Sistemas y expertos certificados en Scrum, permitió conocer que los artefactos y técnicas seleccionados fueron los apropiados para la estructura del aplicativo, siendo 66.7% muy clara por los expertos y 91.3% clara o muy clara por los estudiantes, por lo que el 100% de los participantes lo recomendarían.

Del mismo modo, las ayudas visuales permitieron que se evidenciara la relación que existe entre los diferentes roles con los artefactos, técnicas y eventos de tal manera que los usuarios puedan tener una experiencia funcional y guiada en el uso del marco de trabajo, siendo el prototipo la estrategia que se puede emplear para una mejor adopción del marco de trabajo Scrum en un proyecto ágil, como respuesta a la pregunta problema.

Por tal motivo, pese a ser un primer prototipo se encuentra bastante completo para asistir a los equipos de trabajo en los proyectos de Scrum, a fin de aplicar y adoptar correctamente el marco de trabajo Scrum sea que se tenga poco o bastante conocimiento y claridad del mismo, cumpliendo con las reglas de Scrum planteadas por Ken Schwaber y Jeff Sutherland en la guía oficial.

Referencias Bibliográficas

Adzic, G. (2012). *Impact Mapping: Making a big impact with software products and projects*.

https://www.impactmapping.org/assets/impact_mapping_20121001_sample.pdf

Alyahya, S., Alqahtani, M., & Maddeh, M. (2016). Evaluation and improvements for agile planning tools. *2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA)*, 217–224.

<https://doi.org/10.1109/SERA.2016.7516149>

Autentia (Ed.). (s/f). *Fichas ágiles*. Recuperado el 2 de mayo de 2021, de

<https://www.autentia.com/wp-content/uploads/libros/Autentia-MazosAgile-v1.0.pdf>

Avenidaño, M. (2017, julio 26). *Construyendo el Producto Mínimo Viable con User Story*

Mapping [Liderazgo y gestión]. <https://es.slideshare.net/Marcoviaweb/construyendo-el-producto-minimo-viable-con-user-story-mapping>

Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M.,

Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C.,

Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifiesto por el*

Desarrollo Ágil de Software. agilemanifiesto.

<https://agilemanifesto.org/iso/es/manifesto.html>

Cohn, M. (2004). *User Stories Applied for Agile Software Development*. Pearson Education, Inc.

<http://athena.ecs.csus.edu/~buckley/CSc191/User-Stories-Applied-Mike-Cohn.pdf>

digital.ai. (2021). *15th State of Agile Report* (Núm. 15; State of Agile Report, p. 22). digital.ai.

<https://explore.digital.ai/state-of-agile/15th-state-of-agile-report>

- Frost, B. (2013, junio 10). *Atomic Design*. bradfrost.com. <https://bradfrost.com/blog/post/atomic-web-design/>
- Frost, B. (2016). *Atomic Design*. <https://atomicdesign.bradfrost.com/table-of-contents/>
- Gartner, Inc. (2020a). *Enterprise Agile Planning Tools Reviews and Ratings*. Gartner Peer Insights. <https://www.gartner.com/reviews/market/enterprise-agile-planning-tools>
- Gartner, Inc. (2020b). *Magic Quadrant for Enterprise Agile Planning Tools* (Núm. G00394014). Gartner, Inc. <https://aptasolutions.com/wp-content/uploads/2020/06/Rally-Software-Gartner-2020.pdf>
- Gartner, Inc. (2021). *Magic Quadrant for Enterprise Agile Planning Tools* (Núm. G00733887). Gartner, Inc. <https://www.gartner.com/doc/reprints?id=1-25TS5TMU&ct=210420&st=sb>
- Garzas, J. (2013, enero 8). *Retrospectivas: Qué son y técnicas para hacerlas*. JavierGarzas.com. <https://www.javiergarzas.com/2013/01/retrospectiva-software-agil.html>
- Garzas, J. (2015, octubre 16). *Algunas dinámicas para hacer retrospectivas ágiles*. JavierGarzas.com. <https://www.javiergarzas.com/2015/10/dinamicas-retrospectivas-agiles.html>
- Google. (2020). *Angular. One framework. Mobile & desktop*. Angular. <https://angular.io/>
- Google for Startups. (2016). *Rapid Prototyping: Native | Google for Startups*. <https://www.youtube.com/watch?v=lusOgox4xMI>
- International Scrum Institute. (s/f). *The Scrum Framework* (3a ed.). International Scrum Institute™. https://www.scrum-institute.org/contents/The_Scrum_Framework_by_International_Scrum_Institute.pdf

Jaikrishnan, V. (2018, noviembre 26). *Burndown Charts: What Is It And How To Use Them?*

Zepel. <https://zepel.io/agile/reports/burndown/>

Kua, P. (2006, marzo 9). *The Retrospective Starfish*. thekua.com@rest.

<http://www.thekua.com/rant/2006/03/the-retrospective-starfish/>

Meldre, M. (2016). *Using Impact Mapping in Agile Software Product Management* [University

of Tartu]. [https://www.semanticscholar.org/paper/Magnus-Meldre-Using-Impact-](https://www.semanticscholar.org/paper/Magnus-Meldre-Using-Impact-Mapping-in-Agile-Product-Meldre/971bae898d58133db6cf8c78240d8d7d648b3b6c)

[Mapping-in-Agile-Product-Meldre/971bae898d58133db6cf8c78240d8d7d648b3b6c](https://www.semanticscholar.org/paper/Magnus-Meldre-Using-Impact-Mapping-in-Agile-Product-Meldre/971bae898d58133db6cf8c78240d8d7d648b3b6c)

Menzinsky, A., López, G., & Palacio, J. (2019). *Scrum Master: Temario Troncal I* (2.6.1).

Lubaris Info 4 Media SL. https://www.scrummanager.net/files/scrum_manager.pdf

Menzinsky, A., López, G., Palacio, J., Sobrino, M. Á., Álvarez, R., & Rivas, V. (2020). *Historias*

de Usuario. Ingeniería de Requisitos Ágil: Vol. 3.0. (Scrum Manager, Ed.). Creative

Commons.

Navarro Cadavid, A., Fernández Martínez, J. D., & Morales Vélez, J. (2013). Revisión de

metodologías ágiles para el desarrollo de software. *PROSPECTIVA*, 11(2), 30–39.

<https://www.redalyc.org/articulo.oa?id=496250736004>

Palacio, M. (2020). *Scrum Master—Temario troncal I* (Scrum Manager, Ed.; 3.0.51). Creative

Commons. https://scrummanager.net/files/scrum_master.pdf

Palminio, D. (2018, diciembre 7). Agilismo en Ingeniería de Software: Retrospectiva: Retro del

Barco. *Agilismo en Ingeniería de Software*.

[http://agilismoeningenieriadesoftware.blogspot.com/2018/12/retrospectiva-retro-del-](http://agilismoeningenieriadesoftware.blogspot.com/2018/12/retrospectiva-retro-del-barco.html)

[barco.html](http://agilismoeningenieriadesoftware.blogspot.com/2018/12/retrospectiva-retro-del-barco.html)

- Patton, J. (s/f). *The New User Story Backlog is a Map*. Jeff Patton & Associates. Recuperado el 28 de octubre de 2020, de <https://www.jpattontassociates.com/the-new-backlog/>
- Patton, J. (2014). *User Story Mapping: Discover the Whole Story, Build the Right Product* (M. Treseler, A. Jollymore, & K. Ebrahim, Eds.; 1a ed.). O'Reilly Media Inc. <https://issuu.com/victoriashyshmakova/docs/user-story-mapping-jeff-patton-pete>
- Piza Martínez, C. A. (2016). Scrum a la colombiana: Breve acercamiento al scrum hecho en Colombia. *TIA*, 4(2), 88–92. <https://revistas.udistrital.edu.co/index.php/tia/article/download/9692/pdf/>
- Schwaber, K., & Sutherland, J. (2020). *La Guía de Scrum La Guía Definitiva de Scrum: Las Reglas del Juego*. Creative Commons. <https://www.scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>
- Scrum Alliance. (2017). HOW SCRUM IS USED. *State of Scrum 2017-2018 scaling and agile transformation*, 5–9. <https://www.scrumalliance.org/resources/ebooks>
- solid IT. (2020). *JIRA vs Pivotal Tracker vs Team Foundation Server vs Trello System Properties Comparison*. Project Management Zone. <https://project-management.zone/system/jira,pivotal-tracker,team-foundation-server,trello>
- Targetprocess. (2020). *Scrum*. Targetprocess. <https://www.targetprocess.com/solutions/scrum/#tldr>
- VersionOne Inc. (2019). The 13th annual State of Agile Report. *The 13th annual State of Agile Report*, 2. <https://stateofagile.com/#>

VersionOne Inc. (2020). The 14th Annual State of Agile Report. *The 14th Annual State of Agile Report*, 2. <https://explore.digital.ai/state-of-agile/14th-annual-state-of-agile-report>