

Análisis y planificación del desarrollo de un software para la integración de proveedores a través de metodologías ágiles

Jack Rueda Gil

Trabajo de grado para optar por el título de:
Magister en evaluación y gerencia de proyectos

Director

Jaime Andrés Álvarez Ospina

Magister en administración de empresas

Universidad Industrial de Santander

Facultad de Ingenierías Físicomecánicas

Escuela de Estudios Industriales y Empresariales

Maestría en Evaluación y Gerencia de Proyectos

Bucaramanga

2021

Dedicatoria

A mis padres y hermanos, cada paso de mi vida ha sido influido por su orientación y apoyo.

A mi esposa, incondicional y determinante.

Agradecimientos

A mi director de proyecto, un excelente ser humano de quien obtuve los conocimientos adecuados y precisos que me permitieron orientar el desarrollo de este proyecto.

A mis compañeros de maestría, uno de los mejores grupos de estudio (a pesar de la virtualidad), siempre compartimos una sana competencia y nos brindamos el mejor de los ánimos durante este peldaño profesional.

A la Universidad Industrial de Santander y todo el equipo administrativo de la maestría, nos ofrecieron la mejor asesoría y acompañamiento en cada fase.

Contenido

Introducción.....	12
1. Generalidades del proyecto	14
1.1. Planteamiento del problema	14
1.2. Justificación	15
1.3 Objetivos.....	16
1.3.1 Objetivo general.....	16
1.3.2 Objetivos específicos	16
2. Metodología.....	16
3. Resultados.....	28
3.1. Lista priorizada de producto.....	28
3.1.1. Acta de constitución del proyecto con enfoque ágil	28
3.1.2. Identificación de métodos de exploración del alcance	30
3.1.3 Realización de métodos de exploración del alcance.....	33
3.1.4. Priorización de lista de producto	44
3.1.5. Definición del equipo de trabajo	49
3.2. Selección de metodología ágil.....	52
3.2.1. Metodologías ágiles representativas	52
3.2.2. Selección de metodología	60
3.3. Actividades.	62

3.3.1. Producto mínimo viable (MVP)	62
3.3.2. Definición de épicas.....	64
3.3.3. Hoja de ruta ágil del producto.....	71
3.3.4. Estimación de puntos	72
3.4. Capítulo 4. Recursos de seguimiento.	74
3.4.1. Seguimiento del equipo.....	75
3.4.2. Seguimiento del avance	78
3.4.3. Pruebas.....	82
4. Conclusiones.....	84
5. Recomendaciones	86
Referencias Bibliográficas	87
Apéndices	92

Lista de tablas

Tabla 1. <i>Identificación de interesados</i>	17
Tabla 2. <i>Caracterización de los interesados</i>	18
Tabla 3. <i>Matriz de marco lógico para el desarrollo del proyecto</i>	24
Tabla 4. <i>Acta de constitución del proyecto con enfoque ágil</i>	28
Tabla 5. <i>Historias de usuario recolectadas</i>	34
Tabla 6. <i>Resumen de situaciones presentadas en el juego de roles + persona</i>	38
Tabla 7. <i>Equipo de trabajo</i>	50
Tabla 8. <i>Metodologías ágiles representativas</i>	58
Tabla 9. <i>Épica 1.</i>	64
Tabla 10. <i>Épica 2.</i>	65
Tabla 11. <i>Épica 3.</i>	67
Tabla 12. <i>Épica 4.</i>	69
Tabla 13. <i>Épica 5.</i>	69
Tabla 14. <i>Épica 6.</i>	70
Tabla 15. <i>Desglose de tareas por fase y estimación de puntos.</i>	72
Tabla 16. <i>Plantilla para las reuniones diarias.</i>	75
Tabla 17. <i>Ejemplo 1 del uso de la plantilla de registro de pruebas.</i>	83
Tabla 18. <i>Ejemplo 2 del uso de la plantilla de registro de pruebas.</i>	83

Lista de figuras

Figura 1. <i>Matriz de interesados.</i>	20
Figura 2. <i>Árbol de problemas.</i>	21
Figura 3. <i>Árbol de objetivos.</i>	23
Figura 4. <i>Equipo administrativo de Laboratorio Diesel del Oriente.</i>	33
Figura 5. <i>Mapa de proceso de trabajos</i>	41
Figura 6. <i>Tabla periódica de las técnicas de priorización de listas de producto.</i>	45
Figura 7. <i>Lista priorizada del producto a partir del método MoSCoW.</i>	49
Figura 8. <i>Metodologías ágiles más usadas en 2018</i>	61
Figura 9. <i>Ciclo de vida exploratorio de DAD.</i>	63
Figura 10. <i>Hoja de ruta ágil del producto.</i>	71
Figura 11. <i>Plantilla para la actividad de retrospectiva.</i>	77
Figura 12. <i>Simulación del desarrollo del sprint 1 en Trello.</i>	79
Figura 13. <i>Definición de puntos de historia para las tareas.</i>	79
Figura 14. <i>Gráfica de quemado descendente para el sprint 1 en progreso.</i>	80
Figura 15. <i>Simulación del desarrollo del sprint 2 en Trello.</i>	80
Figura 16. <i>Gráfica de quemado descendente para el sprint 2 en progreso.</i>	81
Figura 17. <i>Simulación de medición del avance del proyecto con la gráfica de quemado ascendente.</i>	82
Figura 18. <i>Historias de usuario del gerente.</i>	92

Figura 19. <i>Historias de usuario de la administradora.</i>	92
Figura 20. <i>Historias de usuario del jefe de taller.</i>	93
Figura 21. <i>Historias de usuario de la responsable SGSST.</i>	93
Figura 22. <i>Historias de usuario de la tesorera.</i>	94
Figura 23. <i>Historias de usuario de la recepcionista.</i>	94
Figura 24. <i>Pantallazos realizados por el jefe de taller.</i>	100
Figura 25. <i>Pantallazos realizados por la tesorera.</i>	100
Figura 26. <i>Pantallazos realizados por la administradora.</i>	101
Figura 27. <i>Pantallazo realizado por la responsable SGSST.</i>	101
Figura 28. <i>Pantallazo realizado por la recepcionista.</i>	102

Lista de apéndices.

Apéndice A. Evidencia de historias de usuario recolectadas.....	92
Apéndice B. Transcripción completa del juego de roles + persona.....	95
Apéndice C. Evidencias de pantallazos recolectados.	100
Apéndice D. Entrevista a experto.	102

Resumen

Título: Análisis y planificación del desarrollo de un software para la integración de proveedores a través de metodologías ágiles. *

Autor: Jack Rueda Gil**

Palabras Clave: Entrega Ágil Disciplinada (DAD), Metodologías ágiles, plan de desarrollo de software.

Descripción: El desarrollo de este proyecto consistió, inicialmente, en el análisis de la problemática de una empresa regional que invierte demasiado tiempo en el proceso de cotizaciones. Al realizar un planteamiento de una posible solución, se realizó una exploración del alcance con el equipo de trabajo de dicha empresa a través de varias herramientas de enfoque ágil (historias de usuario, juego de roles, pantallazos, mapa de procesos, arquetipos de clientes) y una entrevista a un experto en desarrollo de software; lo que permitió obtener una visión temprana de la solución deseada. Entonces, se procedió a realizar un análisis de las metodologías ágiles existentes y cuál era el enfoque de cada una de ellas. Así, por medio de una comparación global de las herramientas que cada una ofrecía, se seleccionó la más adecuada; teniendo en cuenta las variables que influían en la planeación de este proyecto en específico. Luego, se establecieron las actividades necesarias que comprendían las primeras etapas; definiendo las épicas, funcionalidades, y tareas que se realizarían en cada una de estas. Finalmente, se realizó un análisis de los recursos de seguimiento apropiados que permitieran fortalecer al equipo de trabajo con cada entrega y obtener una retroalimentación adecuada para todos los interesados del proyecto.

* Trabajo de grado.

** Facultad de Ingenierías Físicomecánicas, Escuela de Estudios Industriales y Empresariales, Director: Jaime Andrés Álvarez Ospina Magister en administración de empresas

Abstract

Title: Analysis and planning of the software development for the supplier's integration through agile methodologies*

Author: Jack Rueda Gil**

Key Words: Agile methodologies, Disciplined Agile Delivery, Plan for a software development.

Description: The development of this project consisted, initially, in the analysis of the problems of a regional company that invests too much time in the quotation process. When making an approach to a possible solution, a scope exploration was carried out with the work team of said company through various agile approach tools (user stories, role play, screenshots, process map, customer archetypes) and an interview with a software development expert; this allows to get an early insight into the desired solution. Then, an analysis of the existing agile methodologies and the focus of each one was carried out. So, by a global comparison of the tools that each one offers, the most appropriate was selected; considering the variables that influence the planning of this specific project. Then, the necessary activities that comprised the first stages were established; defining the epics, functionalities, and tasks that would be carried out in each of these. Finally, an analysis of the adequate follow-up resources was carried out, which made it possible to strengthen the work team with each delivery and obtain adequate feedback for all project stakeholders.

* Degree Work

** Faculty of Physicomechanical Engineering, School of Industrial and Business Studies, Director: Jaime Andrés Álvarez Ospina Master in business administration

Introducción

A nivel mundial se pueden identificar varias marcas representantes de equipos con motor de combustión interna. Solo en términos de vehículos, se conoce de la existencia de al menos 155 marcas distintas, entre fabricantes en serie y artesanales (Murias, 2018). Adicional a esto, cada marca puede aportar distintas líneas, pudiéndose reconocer que, solo Chevrolet, alcanza a manejar cerca de 140 líneas distintas de vehículos en Colombia; y, entre las marcas más reconocidas, se reconocen al menos 1100 líneas representativas del sector automotor nacional e importado (CarroYa.com, 2020). ANDEMOS (2020) indica en su informe que, a nivel nacional, durante los dos primeros meses del año 2020 se presentó un aumento en venta de vehículos de todo tipo de 15,4% y 11,1% respectivamente.

Cada año es notorio el incremento de variedades que ofrece el mercado actual en términos de marcas. Los acuerdos realizados con otros países permiten la introducción de maquinaria y equipo de distintos orígenes y precios. Luego, al momento de las reparaciones, muchos de estos representantes de marca no cuentan con el respaldo de las partes de recambio y es necesario que intervengan importadores independientes para suplir esa demanda. En relación con lo anterior, en Colombia, al año 2018, existían cerca de 5000 empresas relacionadas con el comercio de autopartes para vehículos y maquinaria especializada en general (DANE, 2020).

Respecto a la industria nacional de autopartes, González (2019), miembro de la junta nacional de la Asociación del sector automotor y sus partes (ASOPARTES), indica que:

“...en la cadena de producción hay otros actores activos (y no tan recordados) como los fabricantes de partes, los distribuidores, los representantes, los importadores, los mayoristas y minoristas, las servitecas, los lavaderos y los montallantas, entre otros. El 90 por ciento de las empresas del ramo son pequeñas y medianas y generan cerca del 60 por ciento del empleo de esta industria.”

Además, el mismo autor menciona que hubo una variación positiva del 19,5% respecto a la venta de autopartes con respecto al año anterior, que se ubicó sobre los 13 billones de pesos.

Este sector en crecimiento promueve la venta de sus productos bajo distintos canales; sin embargo, normalmente se relacionan directamente con las empresas que realizan mantenimiento y puesta a punto de los equipos.

Desde la perspectiva empresarial, es importante reconocer que, al poderse reducir el tiempo que dedica el personal administrativo a las tareas de búsqueda de repuestos, podrían enfocarse otras actividades que generan valor al buen nombre de la compañía, como la atención al cliente. A través de la herramienta que permita extender la necesidad de un repuesto a múltiples usuarios, las actividades repetitivas que a diario se realizan serán disminuidas a una única acción; de forma que, en un tiempo prudente, se cuente con la retroalimentación de las variables que juegan un papel importante en la decisión de compra; como son: costo, marca y disponibilidad. Por otra parte, los proveedores obtendrán una ventaja al poder incluir su inventario en el sistema; de modo que éste reconozca la necesidad a partir del nombre. Así, puede automatizar un sistema de oferta-demanda que hoy en día requiere de mucho desgaste por parte del personal de compras y ventas de las distintas compañías. De igual manera, el tiempo requerido para ofertar los productos con los que cuenta puede finalizar en una mejora de la rotación de su inventario. Finalmente, un tercer actor que se verá beneficiado es el cliente. Al poder obtener una retroalimentación más rápida y global de las partes existentes en el mercado, éste tomará decisiones más acertadas que culminarán en una reducción del tiempo de parada de su equipo y, posiblemente, en una mayor durabilidad de las reparaciones que realice. En relación, la rotación de trabajo en la compañía aumentará.

1. Generalidades del proyecto

1.1. Planteamiento del problema

Actualmente, la empresa Laboratorio Diesel del Oriente, presta el servicio de reparación de motores de combustión interna y sus accesorios externos; de cualquier equipo y marca que exista, por ejemplo: vehículos, plantas eléctricas, compresores, maquinaria agrícola y de construcción, entre otros. En consecuencia, ofrece la venta de repuestos según la necesidad de cada servicio de reparación solicitado.

Todos los clientes presentan diferencias entre sus necesidades. Algunos requieren rapidez en la atención sin importar la marca; mientras que otros, consideran que los repuestos originales son los más adecuados para sus equipos sin importar el costo. Por lo anterior, cada vez que se requiere un repuesto es necesario analizar la forma en que se relacionan las siguientes 3 variables: marca del repuesto, costo y tiempo de adquisición.

En términos generales, al realizar la búsqueda de disponibilidad de repuestos para un servicio dado, el área administrativa encargada de la operación de la empresa debe emplear un estimado del 80% de su tiempo en la consecución de dichas partes. Esto, debido a que la diversidad de proveedores es amplia y no todos venden lo mismo; pues pueden presentar especialidad en un producto o preferencia por una marca.

Puesto que en dicha empresa se realiza reparación de cualquier parte, de cualquier equipo, que use un motor de combustión interna, los proveedores son difícilmente clasificables. Por lo anterior, se debe consultar a cada uno de acuerdo con la necesidad y solo con la información escasa que se tiene sobre los productos que ofrecen. Así mismo, como las reparaciones de equipos similares con fallas similares pueden ser distantes; normalmente no se conserva continuidad en la información consultada.

Dicho esto, los proveedores son recordados de acuerdo con la solución que ofrecieron en su momento y no al portafolio de productos que puedan llegar a suministrar; de forma que representen una solución a largo plazo y no puntual.

1.2. Justificación

Los acuerdos comerciales realizados con otros países han permitido un ingreso notorio de nuevas marcas y líneas de equipos al mercado colombiano; desde vehículos hasta maquinaria agrícola, de todas las regiones del mundo buscan posicionarse y logran nuevos clientes que muy posiblemente lleven sus equipos a reparar en un sitio distinto a la casa matriz. En este punto, entran nuevos proveedores o se actualizan los existentes y es donde se debe contar con un método de clasificación adecuado.

Una de las principales características que permite diferenciar a una empresa de este sector, es la oportunidad de ofrecerle al cliente la variedad de opciones disponibles. Esto, se puede lograr a través de una correcta clasificación basado en el perfil de cada proveedor de acuerdo con la relación marca-costo-tiempo de entrega.

Luego, al solicitarse un repuesto, el área administrativa conllevará la solicitud de cotización de una forma más sencilla y práctica; permitiéndole atender otras necesidades de los clientes y logrando mayor satisfacción en el servicio prestado. Adicionalmente, la rapidez con que se consiga la información permitirá un mayor flujo de trabajo; que, en consecuencia, representará a la empresa un mayor flujo de caja puesto que los equipos no permanecerán tanto tiempo en la empresa por causa de repuestos de difícil consecución.

Es importante resaltar la importancia del desarrollo de esta solución, ya que, al presentarse una amplia variedad de proveedores (a la que cada vez se suman más empresas), es necesario contar con todos ellos cuando se presente un requerimiento específico, tales como: repuestos de

difícil consecución, originalidad, disponibilidad, menor precio, entre otros. Además, su uso permitirá trascender el método actual, en el cual se le pregunta a cada proveedor de forma independiente; logrando así una evolución de dicha tarea.

1.3 Objetivos

1.3.1 Objetivo general

Analizar y planificar un proyecto de software para la integración de proveedores, a través de metodologías ágiles, que permita reducir el tiempo requerido para la cotización de repuestos en la empresa Laboratorio Diesel del Oriente Bucaramanga en el año 2021.

1.3.2 Objetivos específicos

- Elaborar la lista de producto priorizada que permita dar a conocer al equipo de trabajo el alcance inicial de acuerdo con los requerimientos del cliente.
- Identificar la metodología ágil adecuada para la dirección del proyecto, con base en los requerimientos del cliente.
- Definir las actividades a llevar a cabo que permitan la realización del software de acuerdo con la metodología ágil seleccionada.
- Establecer los recursos de seguimiento necesarios durante el desarrollo del software, orientados a la metodología ágil seleccionada.

2. Metodología

Inicialmente, para la etapa predictiva, que corresponde a la planificación del desarrollo del proyecto, se realizará un enfoque de acuerdo con el marco lógico; ya que nos permite visualizar claramente el concepto general a partir de unos objetivos y actividades macro; que finalmente nos permitirá establecer de forma estructurada la forma en que se llevarán a cabo cada una de las actividades. Una vez se establezca lo mencionado, será en el diseño del plan donde se evalúe la

necesidad de alterar el enfoque predictivo; siempre que se considere necesario de acuerdo con la información que se vaya obteniendo durante su ejecución. De acuerdo con lo anterior, se lleva a cabo el análisis de los participantes respecto al desarrollo de un plan para la realización de un software que permita la integración de los proveedores. En la tabla 1 se encuentra la identificación de los interesados. En esta, se refleja en las observaciones aquellas características del grupo su aporte a la solución del problema. Luego, en la tabla 2, se muestra la caracterización de los grupos.

Tabla 1.

Identificación de interesados

Tipo	Nombre del grupo de interés	Grupos de interés	Observaciones
Directo	Área administrativa	Accionistas de la empresa	Principales beneficiarios económicos de la solución del problema.
		Personal administrativo de la empresa	Aquellos para los cuales se desarrollará el software. Se tendrá en cuenta para la clasificación y consulta de los proveedores.
Indirecto	Área comercial	Proveedores	La solución permitirá agilizar sus cotizaciones y tener mayor alcance en la oferta de su inventario.
		Clientes	La rapidez con que se gestionen las cotizaciones de repuestos disminuirá el tiempo de reparación.

Oponente	Externos	Competencia directa	Algunos de sus proveedores pueden ser también los de la empresa y podrían hacer alianzas para obstruir el flujo de la información.
		Mensajeros	Se verán afectados en la medida en que no se requiera de sus servicios para la cotización de partes en el local de los proveedores.
Neutral	Colaboradores	Personal operativo de la empresa	Son ajenos al desarrollo del software. Pueden llegar a ser una fuente de información fiable en cuanto a la calidad de las marcas ofertadas.
		UIS	Aquella que brindará el apoyo administrativo para el desarrollo del proyecto.

Tabla 2.

Caracterización de los interesados

Nombre	Posición	Influencia	Interés	Expectativa	Contribución
Área administrativa	Beneficiario	5	5	Disminuir el tiempo de duración de un equipo en las instalaciones. Mejorar la percepción del cliente respecto	Económica.

				al flujo de trabajo de la empresa.	
Área comercial	Cooperantes	3	5	Disminuir el tiempo que dedica a la búsqueda de proveedores	Participación en el desarrollo.
Externos	Oponentes	3	0	No se desarrolle el software.	N/A.
Colaboradores	Cooperantes	1	3	N/A.	Proveer información relevante respecto al desarrollo de la solución.

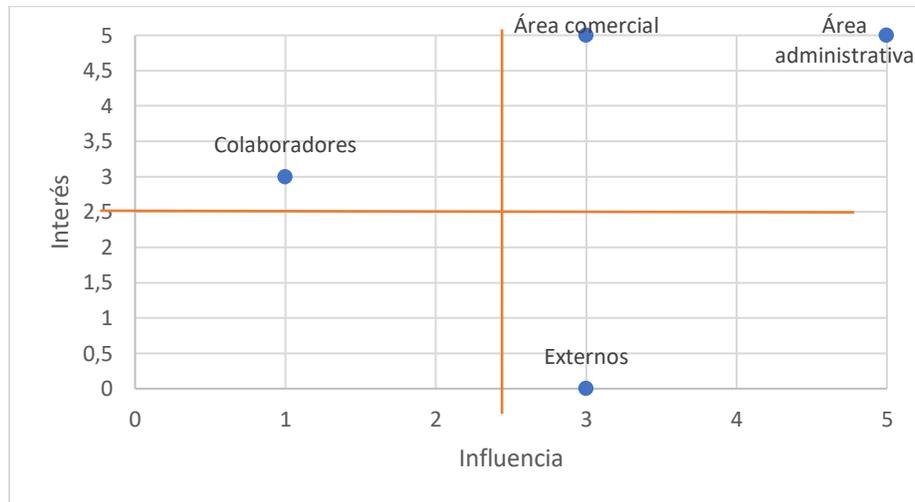
En la figura 1 se observa la matriz de interesados. Respecto a esta, se realizan las siguientes apreciaciones:

- Cuadrante 1 – Gestión cercana: En esta zona se encuentran el área comercial y el área administrativa. Serán los principales actores y de quienes se contará con todo el apoyo para la solución del problema.
- Cuadrante 2 – Satisfacer: En esta zona se encuentran los colaboradores. A través de sus aportes se podrá llegar a desarrollar el problema.

- Cuadrante 3 – Comunicar e informar: En esta zona se encuentra el grupo de externos. Se deberá establecer un plan para aumentar el interés de estos en la solución del problema.
- Cuadrante 4 – Supervisar: En esta zona no se encontró ningún grupo de interesados.

Figura 1.

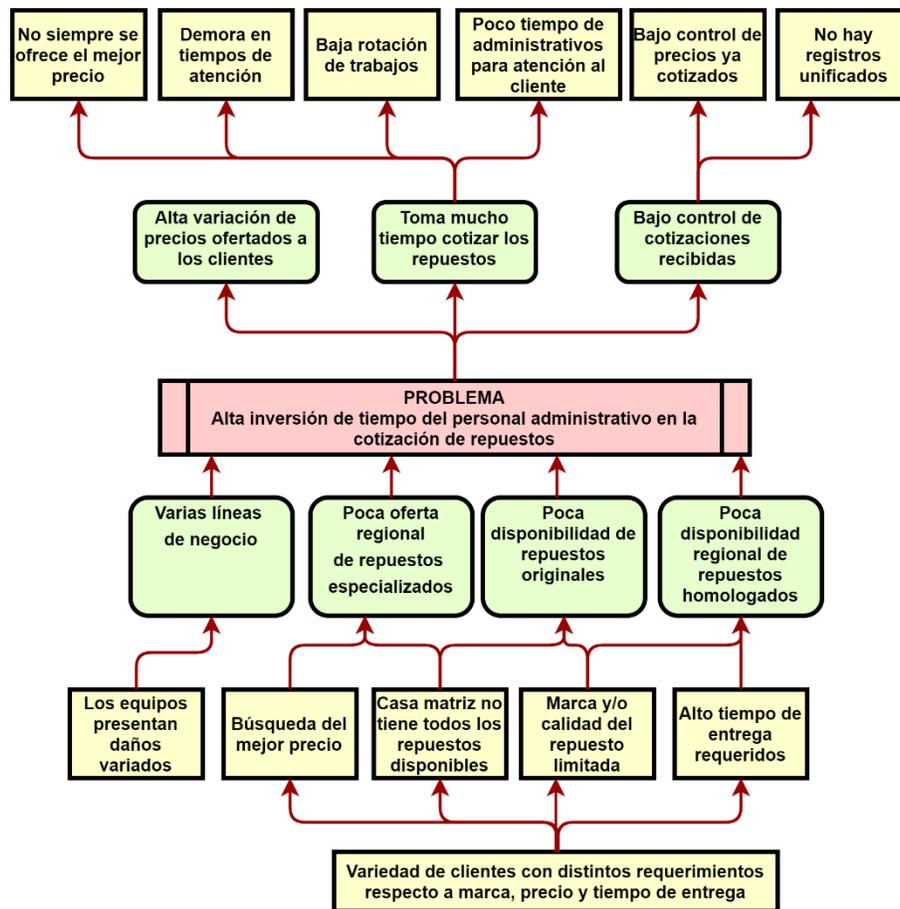
Matriz de interesados.



En la figura 2 se observa el árbol de problemas. Se identificó que una de las principales causas por las cuales se cuenta con varios proveedores, es la variedad de requerimientos que los clientes prefieren a la hora de decidir qué alternativa de reparación seleccionar. Inicialmente, se debe mencionar que, lo normal en una reparación de cualquier equipo, es que la casa matriz, aquella que representa la marca que vende, cuente con todos los repuestos para las reparaciones necesarias. Sin embargo, este principio es bastante alejado de la realidad. No solo por el hecho de que en muchas ocasiones no tienen un stock de repuestos, sino por la filosofía de dichas compañías en las que presentan al cliente como única opción el cambio de las partes y no sus reparaciones. Esta situación conlleva a los clientes a buscar alternativas más económicas y es donde aparecen las soluciones en los talleres multimarca.

Figura 2.

Árbol de problemas.



Así, los clientes llegan a presentar distintas necesidades. Algunos preferirán que su reparación se realice con repuestos originales, otros prefieren lo más económico que ofrece el mercado, otros cuentan con muy poco tiempo por las pérdidas que pueda producir la parada de un equipo y solo les interesa solucionar con cualquier opción, y, por último, aquellos clientes que desean que sus reparaciones se realicen con una marca específica distinta de la original. Todas estas alternativas, sumado a la variedad de fallas que pueda presentar un equipo respecto a todos sus componentes internos, son el conjunto de causas que conlleva a que la empresa requiera de varios proveedores para poder satisfacer las necesidades de sus clientes según sean presentadas.

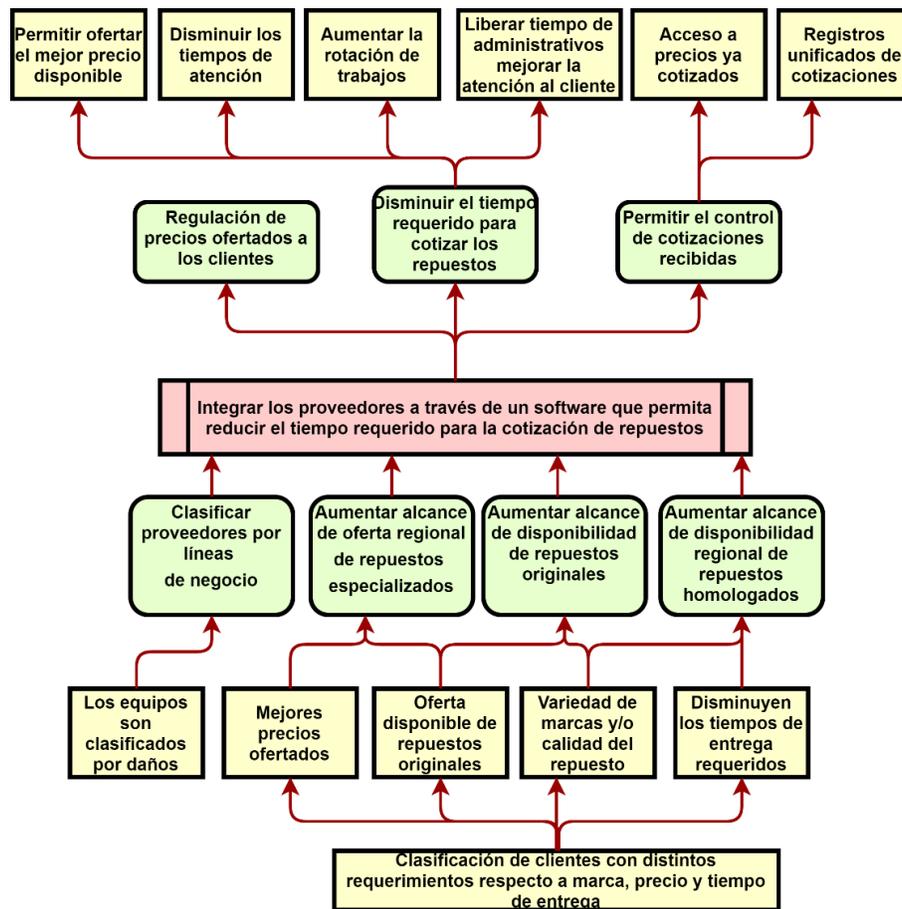
Luego, se tienen las consecuencias. Una de las principales se presenta en el gran tiempo que invierte el personal administrativo en la búsqueda de los proveedores. Esto conlleva a que los equipos se encuentren detenidos durante más tiempo del necesario e impida que en la misma empresa se reciban más trabajos por la baja rotación. Además, debido a la urgencia que algunos clientes presentan, no siempre se ofrece la mejor alternativa que ofrezca el mercado sino aquella que pueda solucionar el problema de forma rápida; por lo que el personal administrativo recurre a aquellos proveedores conocidos, que tenga cerca y que cuenten con la disponibilidad del repuesto. Sumado a esto, se encuentra el hecho de que no se realiza un registro de las consultas que se llevan a cabo. Por lo que, si se llega a presentar un requerimiento a futuro, nuevamente se inicia la búsqueda de un proveedor adecuado. En ocasiones se estudian las compras realizadas anteriormente, pero esto depende del equipo de contabilidad que no siempre está disponible para atender las necesidades.

Mas adelante, en la figura 3 se aprecia el árbol de objetivos; en el cual se proyectan las necesidades de la empresa con base en el problema principal planteado. Es así como se logra identificar claramente el alcance del proyecto a desarrollar.

A través del software se debe poder clasificar a los proveedores por líneas de negocio, aumentar el alcance de las ofertas regionales recibidas respecto a los repuestos especializados, aumentar el alcance de la disponibilidad de repuestos originales y aumentar el alcance de la disponibilidad regional de repuestos homologados. Todas estas características permitirán lograr los fines del objetivo principal, que, según se aprecia, lograrán: regular los precios ofertados a los clientes, disminuir el tiempo requerido para cotizar los repuestos y tener control de las cotizaciones recibidas para consultas futuras.

Figura 3.

Árbol de objetivos.



Dentro de los pasos a desarrollar en el marco lógico, lo siguiente corresponde al análisis de alternativas para seleccionar aquella que se llevará a cabo. No obstante, por tratarse de un proyecto de software que, al lograrse su desarrollo, están implícitas las soluciones a dichas alternativas; no se considera necesario realizar este análisis.

Por último, en la tabla 3 se aprecia la matriz de marco lógico que permite evidenciar la planificación del proyecto, distribuida por actividades y orientada al objetivo del proyecto.

Tabla 3.

Matriz de marco lógico para el desarrollo del proyecto

Objetivo/Actividad	Indicador	Medio de verificación	Supuestos
Mejorar la atención al cliente.	Satisfacción cliente	del Encuesta	-
Aumentar la rotación de equipos en reparación.	Tiempo de duración de un equipo de acuerdo con la reparación	Tabla con fechas de entrada y salida de los equipos reparados	
Incrementar las ventas de la empresa.	Ventas anuales	Sistema contable	
Analizar y planificar un proyecto de software para la integración de proveedores.	Planes de desarrollo de softwares planificados.	Resultados del proyecto.	Disponibilidad de empresas dispuestas a desarrollar el software.
S1: Elaborar la lista de producto priorizada.	Listas de productos priorizadas.	Lista de producto (Product Backlog)	Las necesidades del cliente se pueden separar
S2: Identificar la metodología ágil	Tabla de metodologías ágiles identificadas.	Análisis comparativo de metodologías ágiles.	en una lista. Existe una metodología ágil

<p>adecuada para el diseño del plan.</p>	<p>Metodologías ágiles seleccionadas.</p>	<p>Relación de actividades con las necesidades del cliente.</p>	<p>adecuada para el desarrollo del software.</p>
<p>S3: Definir las actividades a llevar a cabo.</p>	<p>Actividades definidas.</p>	<p>Recursos diseñados.</p>	<p>Existen herramientas de seguimiento adecuadas.</p>
<p>S4: Establecer los recursos de seguimiento necesarios para la dirección del proyecto.</p>	<p>Recursos de dirección identificados.</p>	<p></p>	<p></p>
<p>S1.1: Reconocimiento de la problemática en la empresa.</p>	<p>Problemáticas identificadas.</p>	<p>Árbol de problemas.</p>	<p>El cliente está dispuesto a suministrar la información</p>
<p>S1.2: Definición del equipo de trabajo.</p>	<p>Necesidades identificadas.</p>	<p>Árbol de objetivos.</p>	<p>respecto a sus necesidades.</p>
<p>S1.3: Identificación de las necesidades de la empresa.</p>	<p>Listas de necesidades creadas.</p>	<p>Lista de necesidades.</p>	<p></p>
<p>S1.4: Priorización de las necesidades.</p>	<p></p>	<p></p>	<p>La metodología ágil adecuada permite el</p>

S2.1: Identificación de las metodologías existentes.	Listas de metodologías ágiles creadas.	Lista de metodologías de desarrollo del software.	del
S2.2: Análisis comparativo de las metodologías identificadas.	Tablas de comparación de metodologías ágiles realizadas.	Evaluación de alternativas de metodologías ágiles.	de
S2.3: Selección de la metodología adecuada para el planteamiento del desarrollo.	Tabla de evaluación de metodologías ágiles realizadas.	Lista de actividades.	
S3.1: Identificar las actividades requeridas para el desarrollo del proyecto.	Listas de actividades creadas.	Diagrama de flujo.	Las actividades requeridas son suficientes para requerir un orden.
S3.2: Identificar el orden adecuado para el desarrollo de un software por actividades.	Diagramas de flujo del desarrollo del software creados.		
S3.3: Ordenar las actividades a realizar.	Listas de actividades		

	ordenadas	Lista de métodos.	Existen
Para S4:	realizadas.		herramientas
S4.1: Identificar los requerimientos de software necesarios para el desarrollo del entregable.	Listas de métodos de seguimiento creadas.	Lista de herramientas. Instrumentos diseñados.	para realizar el seguimiento del desarrollo del software.
S4.2: Identificar los métodos requeridos de seguimiento del desarrollo.	Listas de herramientas disponibles creadas.		Los instrumentos de seguimiento creados son adecuados para la población objetivo.
S4.3: Establecer las herramientas de seguimiento a utilizar.			
S4.4: Diseñar los instrumentos de seguimiento.	Cantidad de instrumentos diseñados.		

3. Resultados

3.1. Lista priorizada de producto.

3.1.1. Acta de constitución del proyecto con enfoque ágil

Inicialmente, se realiza el acta de constitución del proyecto desde la perspectiva ágil. Para esto, se tuvo en cuenta las recomendaciones de la guía práctica ágil (PMI®, 2017), que indica que se debe responder a las siguientes preguntas:

“¿Por qué estamos haciendo el proyecto? Esta es la visión del proyecto.

¿Quién se beneficia y cómo? Esto puede ser parte de la visión del proyecto y/o el propósito del proyecto.

¿Qué significa hecho para el proyecto? Estos son los criterios de lanzamiento del proyecto.

¿Cómo vamos a trabajar juntos? Esto explica el flujo de trabajo previsto”

Luego de realizar el análisis mencionado, se procede a realizar el documento según se observa en la tabla 4 descrita a continuación:

Tabla 4.

Acta de constitución del proyecto con enfoque ágil

Nombre del proyecto	Software de integración de proveedores
Cliente	Laboratorio Diesel del Oriente
Definición	
Objetivos (visión)	Beneficiarios
Reducir el tiempo requerido por parte del personal administrativo para solicitar	Directos <ul style="list-style-type: none"> • Área administrativa

cotización de repuestos a distintos proveedores al mismo tiempo.

- Accionistas de la empresa

Indirectos

Organizar la información recibida por parte de los proveedores, de modo que se clasifique de acuerdo con las características de cada producto cotizado.

- Proveedores
- Clientes

Conservar las ofertas recibidas para acceder al histórico de un producto, cliente, referencia o marca.

Incrementar el flujo de trabajo.

Producto terminado

Flujo de trabajo

Se permite el ingreso de proveedores a la base de datos, indicando el tipo de repuestos que puede suministrar y la marca.

Se permite realizar consultas diarias sobre el posible uso e interacción del software a los beneficiarios directos.

A través de correos electrónicos se enviarán

El personal administrativo puede realizar el envío de una solicitud de

los bocetos del diseño de la interfaz para que se valide con el equipo de trabajo.

repuestos a distintos proveedores que puedan satisfacer la necesidad.	El administrador y jefe de taller de la empresa aportarán la información relevante sobre los métodos adecuados para el ingreso y clasificación de proveedores.
Un posible proveedor responde a la solicitud de cotización y esta se evidencia en el software.	Se realizará una reunión semanal que permita evidenciar los avances del producto respecto a las entregas estipuladas.
El personal administrativo tiene permitido indicar a qué proveedor se le va a realizar la compra y esta información se conservará.	Al momento de la entrega del producto terminado se debe llevar a cabo una capacitación sobre el uso del software.
Las ofertas recibidas se pueden clasificar por marca, costo y/o tiempo de entrega.	El gerente validará los resultados con su equipo de trabajo en un periodo no mayor a 2 días posterior a la entrega.
El personal administrativo debe poder consultar a través de un histórico las ofertas, pudiéndose filtrar por producto, cliente, referencia o marca.	La retroalimentación de la interacción de cada entrega se realizará a través de video llamada o de forma presencial.

3.1.2. Identificación de métodos de exploración del alcance

Posteriormente, se realizó una identificación inicial de los métodos que las distintas metodologías ágiles llevan a cabo para realizar la exploración del alcance. Primero, se consideró

el pensamiento del diseño (design thinking) ya que es una metodología que permite obtener de manera anticipada una visión futura del escenario desde el punto de vista del usuario; centrándose en el usuario final (García, 2020). Algunas de las herramientas utilizadas en esta metodología son: juego de roles, plano de producto/servicio, storyboard, folletos y pantallazos, entre otros.

Adicional a lo anterior, se consideró el diseño de la lista priorizada de producto desde la metodología scrum. De acuerdo con Radigan (2020), las historias de usuario son una herramienta eficaz en la obtención de información deseada del usuario; ya que se expresan las características de acuerdo con el siguiente parámetro: como (rol) quiero (función) para (objetivo). De esta forma, los requisitos son plasmados y separados de aquellos que se consideran épicas, por ser funciones globales que requieren grandes cantidades de trabajo y pueden desglosarse; pero que conducen a un objetivo común. Ramos (2020) indica que el listado obtenido se puede dividir en características, fallas, trabajo técnico y adquisición de conocimiento; luego se prioriza y se socializa con el equipo y se crea el sprint backlog; de modo que permita el movimiento de ítems de un lado a otro a medida que se crea cada sprint basado en la lista

También, se consideró la forma en que se definen las tareas en la metodología Kanban. Berzal (2016) sugiere que se lleve a cabo de la forma en que se describe a continuación. Primero, se debe hacer una retroalimentación con el cliente para generar la lista de ideas. Luego se crean los niveles de prioridad, que pueden establecerse por alto, medio o bajo, costo de demora (COD: cost of delay) o costo de demora dividido en la duración (CD3: cost of delay divided by duration). Posteriormente, se realiza el upstream Kanban; que toma dichas ideas y las define mejor, recabando información para decidir si se realizan o no y cuyo propósito es refinar lo plasmado anteriormente. Ahora, para definir las tareas, se les debe identificar: nombre, descripción basada en criterios de verificación o lista de chequeo, fecha de entrega y persona asignada. Por último, se

establece un flujo visual que impida trabajar en más de 3 de estas al mismo tiempo; lo anterior, de acuerdo con la complejidad de cada proyecto y el tamaño del equipo de trabajo.

Otras de las metodologías consultadas fue Scrum-Ban, la cual, en términos de definición de tareas, utiliza las técnicas de scrum y las ordena en un tablero Kanban (KanbanTool, 2020). También, se indagó en la metodología XP que realiza una consulta con el cliente a partir de la fase de planificación de cada uno de sus ciclos de desarrollo. Lo anterior, con miras a proyectos cuyos requisitos no se encuentren claramente definidos (Junquera, 2019).

Finalmente, Ambler & Lines (2019) manifiestas una serie de herramientas de exploración del alcance desde la perspectiva ágil, las cuales se dividen en un diagrama de flujo que permite identificar, entre otras cosas, el propósito, uso, dominio y proceso. Para esto, se recomienda usar algunas herramientas como: mapas de impacto, mapas mentales, persona y escenario de uso.

A partir del análisis de cada perspectiva, y teniendo en cuenta el tipo de proyecto y las dificultades para llevar a cabo la recolección de información, se consideró realizar las siguientes actividades:

- Historias de usuario: esta herramienta permitirá conocer de primera mano las necesidades que tienen los usuarios respecto del producto que se pretende desarrollar. Los cargos que intervinieron son: gerente, administrador, jefe de taller, responsable SGSST, tesorera y recepcionista.
- Juego de roles + Persona: debido a que no se pudo contar con la presencia de los proveedores en las historias de usuario, se realizó el juego de roles que permite, a cada uno de los integrantes mencionados anteriormente, identificarse como un posible proveedor. A su vez, a través de la combinación con Persona, se pudieron establecer distintas situaciones en las que se proponía una solución a la problemática planteada. Esto permitió que, a partir del conocimiento con el que cuentan las

personas involucradas sobre las distintas situaciones que se han presentado, se pudieran construir escenarios con distintos problemas y posibles soluciones.

- Pantallazos: esta actividad se realizó con el ánimo de tener una visión temprana del producto terminado a través de los ojos del personal administrativo. Se consideró importante ya que puede ofrecer el primer boceto del diseño de la interfaz.
- Mapa de procesos: al poder determinar claramente a qué fase de todo el proceso se le está diseñando una solución, permitirá al grupo desarrollador dar a conocer las reglas de negocio e identificar a los actores principales de cada etapa.
- Arquetipos de clientes: se realizaron 3 tipos de clientes de modo que se le pueda dar a entender al equipo desarrollador del software cómo es el perfil de los clientes y qué esperan cada uno de ellos. De esta forma, podrán existir puntos claves en la presentación de la información que permita identificar entre los proveedores la mejor selección de acuerdo con el perfil del cliente de la empresa.
- Entrevista a experto: a través de esta información, se logrará recolectar información clave y contemplar, en la planeación, tanto el punto de vista del cliente (dueño del producto) como el del proveedor (desarrollador de software).

3.1.3 Realización de métodos de exploración del alcance

Historias de usuario

Se realizó una reunión un miércoles al medio día luego de almorzar. Todo el equipo administrativo de la empresa Laboratorio Diesel del Oriente, que se muestra en la figura 4, hizo parte del ejercicio de creación de las historias de usuario. Previo al inicio de la actividad, se les comunicó de qué se trataba el proyecto y cuán importante era su participación. Luego, se les entregó las tarjetas en blanco y se explicó el uso de estas a través del “como... quiero... para...”.

Figura 4.

Equipo administrativo de Laboratorio Diesel del Oriente.



También, se les manifestó que, durante la realización de esta actividad, podían identificar necesidades a partir de otros roles; ya que, al ser un equipo de trabajo, todos conocen a fondo la interacción que los demás tienen con los proveedores y las necesidades que pueden surgir a partir del desarrollo de la solución. Al principio fue necesario realizar uno o dos ejemplos; pero, tan pronto se entendió la dinámica, las ideas empezaron a fluir con mucha facilidad.

A continuación, en la tabla 5, se muestran los resultados. Las evidencias se pueden consultar en el apéndice A.

Tabla 5.

Historias de usuario recolectadas

Como...	Quiero...	Para...
Gerente	Un cuadro de búsqueda	Buscar cotizaciones por tipo de vehículo
	Que al decidir la compra se notifique a la recepción y tesorería	Que sepan qué repuestos llegan y para quién son

	Solicitar cotización de algo y que solo llegue a los que venden ese tipo de repuesto	Ahorrar tiempo en preguntarle a cada uno
	Poder agregar nuevos proveedores	Que se les consulte en solicitudes futuras
	Que si no me responden rápido mande una alarma al proveedor	Recordarle al proveedor que no me ha cotizado
Vendedora (administrador)	Una función que me muestre posibles proveedores por marca de equipo, vehículo o repuesto	Saber si existen en la base de datos antes de solicitar la cotización
	Una función que me muestre según sistema posibles repuestos a cambiar	Saber si hace falta algún posible repuesto en la cotización
	Escribir una referencia de repuesto	Obtener el precio
	Conocer mensajeros confiables disponibles	Agilizar el traslado de repuestos a nuestras instalaciones
	Poder ingresar varias opciones de búsqueda (fotos, referencias, medidas, etc.)	Tener varias opciones a la hora de buscar cotizar
	Una función que muestre los comparativos de precios cotizados	No consultar varias veces el mismo repuesto

Que el sistema permita guardar referencias o repuestos por medidas específicas

Facilitar la búsqueda de repuestos por específicos

Que el software me muestre cotizaciones básicas

Dar cotizaciones rápidas y aproximadas al cliente

Jefe de taller

Poder agregar fotos

Buscar repuestos por muestra

Que el inventario de los proveedores esté unido al software

Que se actualice el listado de precios de inmediato sin necesidad de estar buscando

Una función para obtener información del proveedor y del cliente

Tomar decisiones más rápidas en cuanto a la marca o el precio deseado

Que cada empleado tenga acceso a la herramienta

Enviar una solicitud de cotización al área encargada

Una base de datos de los equipos de nuestros clientes

Tener los datos más relevantes de los componentes del equipo, referencias, filtros, cantidad de líquidos, etc.

Responsable

Un filtro

Encontrar únicamente los proveedores que voy a necesitar en mi área

SGSST

Una función de alarma semanal

Tener conocimiento del estado de los elementos de protección personal

		antes de realizar la inspección general
	Un filtro de proveedores que priorice los que aún están vigentes (compras recientes)	Ahorrar tiempo en el proceso
Tesorera	Una función que me muestre con el número de factura el proveedor y detalle de la compra	Consultas futuras
	Una función que me muestre en orden cronológico vencimiento de proveedores	Controlar las compras vencidas
Recepcionista	Una alarma para que me avise cuando no se ha decidido una cotización en 3 días	Agilizar el trabajo
	Poner en el sistema la placa del vehículo y me muestre cuándo fue la última vez que se le pidió algo al vehículo	Agilizar el trabajo
	Que si llega un cliente solo sea digitar el nombre y la identificación y que me salgan todos los datos de él y que igual sea con placa del vehículo	No estar todas las veces agregando los mismos datos

Una función que me muestre cada tiempo los clientes que han salido Un servicio posventa

Juego de roles + Persona

Al finalizar la actividad anterior, se procedió a realizar de manera verbal un juego de roles. Se ubicó un celular en el centro del salón y se realizó la grabación de un video con el fin de que las ideas propuestas fueran ininterrumpidas. Ya que no era posible contar con todos los interesados en el momento (como los proveedores), se combinó con la perspectiva de Persona, la cual define que cada usuario puede escoger su rol y establecer una situación que pueda o no conllevar a un problema. Luego, entre el equipo de trabajo se define cómo se llevaría a cabo una posible solución. Una de las características principales de esta última técnica, es que lleva a los usuarios a lograr mayor empatía, ya que son ellos quienes deben idear una forma de enfrentarse al problema y que sea posible resolver a través del software que se propone desarrollar. Como en el anterior ejercicio, se les dio 2 ejemplos iniciales y a partir de ahí se desarrollaron los demás. A continuación, en la tabla 6, se muestra el resumen de las situaciones que se presentaron. La transcripción completa se puede encontrar en el apéndice B.

Tabla 6.

Resumen de situaciones presentadas en el juego de roles + persona

Situación	Problema	Solución
1 y 2	Se consideró que se había solicitado una cotización, pero no dio clic en enviar	Generar confirmación de enviado, recibido y leído Generar alarmas programables en horario y periodicidad para recordar

		que no se ha enviado una solicitud de una cotización que se está creando
3	El proveedor cree que respondió a la cotización, pero no dio clic en enviar	Que las alarmas sean tanto para cliente como para proveedor
4	Solo se puede cotizar parte de lo solicitado	Permitir indicar si se cuenta con el repuesto original, homologado o remanufacturado, si está para importación o no y el plazo de entrega
5	El proveedor no conoce al cliente y resta importancia a la solicitud	Permitir que sea la empresa la que se identifique primero que la persona
6	El proveedor no responde y no se sabe por qué	Permitir que el proveedor envíe una retroalimentación indicando si no puede contestar inmediatamente, si está buscando el repuesto porque no lo tiene disponible o si se debe importar
7	Los proveedores dejan de responder ya que no se les compra	Permitir adicionar inventario a la plataforma para generar respuestas automáticas a partir de valores conocidos como: nombre del repuesto, aplicación, marca, línea, referencia
7.1	Los proveedores no desean ingresar el inventario por temor a que su competencia conozca todos sus precios	Permitir a los proveedores dar permisos a ciertos clientes para que accedan a sus precios

También, se puede cruzar la información de la solicitud con el inventario y sugerir al proveedor que presente uno de los repuestos con los que cuenta

8 No llegan los repuestos según se solicitaron Permitir al proveedor la opción de adjuntar fotografías del repuesto cotizado (o de su inventario) y que estas se notifiquen a recepción

Permitir al cliente indicar la persona encargada de este trabajo para que en la recepción sepan a quién dirigirse cuando llegue el repuesto

Pantallazos

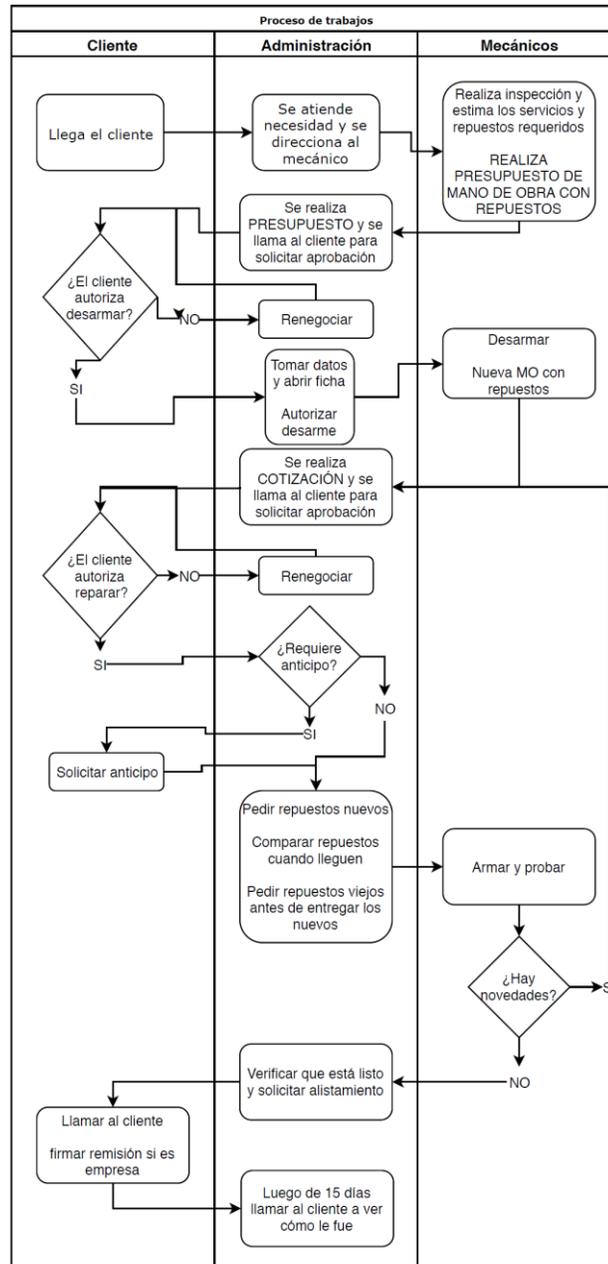
Finalmente, se concluyó la recolección de información de los usuarios con esta actividad. Se les solicitó a los participantes que dibujaran un posible escenario del uso del software. Dicho proceso es un nivel menor al guion gráfico (storyboard), ya que no pretende mostrar todo el panorama de una función sino un evento específico que consideren relevante. De esta manera, se podrá reflejar al equipo de trabajo, del desarrollo del software, la forma en que los usuarios imaginan el resultado final y la situación que ocurre en el momento que escogieron dibujar. Los resultados se pueden observar en el apéndice C.

Mapa de proceso

A continuación, en la figura 5, se muestra el mapa de proceso de trabajo que se lleva a cabo en la empresa cada vez que se recibe una nueva solicitud de servicio. Cabe resaltar que algunos clientes conocidos no pasan por el proceso de cotización previa, sino que se procede de una vez

Figura 5.

Mapa de proceso de trabajos



con la inspección y desarme de las partes. Adicionalmente, este diagrama es importante ya que puede aportar las reglas de negocio al equipo de trabajo, una vez se inicie con el desarrollo del software y en caso de que se requiera consultar la línea de flujo del proceso al cuál se le está proponiendo una solución.

Arquetipos de clientes

Jefe de flota

Fabián Pérez, de 31 años, es un ingeniero mecánico que labora en una empresa que cuenta con 18 vehículos para el traslado de la mercancía a nivel regional. Vive en un sector de estrato medio y cuenta con un vehículo propio para su transporte. De él dependen económicamente 2 personas, que son su esposa y su hija. Los fines de semana, procura en lo posible de desconectarse de la empresa para disfrutar con su familia.

Jerárquicamente, dentro de la empresa, Fabián tiene 2 jefes directos. Uno de ellos es la jefe de ventas que depende de la flota para cumplir con los objetivos propuestos. El otro es el gerente general de la empresa quién le pide encontrar el mejor balance entre costo/beneficio; pero siempre con el objetivo de tener la mayor cantidad de vehículos disponibles.

De esta manera, Fabián se convierte en un tipo de cliente cuya preocupación está orientada a habilitar los vehículos de la manera más rápida y con lo que se consiga en el mercado. Aquel proveedor que le ofrezca soluciones rápidas ganará su confianza; teniendo en cuenta que las soluciones, temporales o definitivas, permitan que la empresa se mantenga en operación con la menor interrupción posible. Claro que, todo lo anterior, desde un punto de vista económicamente viable.

Propietario de vehículo de alta gama

Juan Díaz, de 58 años, es un ingeniero metalúrgico jubilado de una de las mejores empresas del país y con los mejores salarios. Vive en un sector de estrato alto junto con su esposa quien es

una empresaria exitosa de la región. Sus hijos se encuentran estudiando y/o trabajando en el exterior. Al llegar a su hogar lo esperan 2 perros labrador que les hacen compañía. Los fines de semana comparte con sus amigos en el club o visitan lugares exóticos que les ofrezcan las mejores experiencias.

El vehículo de Juan es de alta gama y solo ha sido intervenido por el concesionario. En alguna ocasión, le recomendaron buscar una empresa externa para llevar a cabo un servicio de reparación pues los costos por reemplazo de partes superaban el 10% del valor del vehículo. Ante esto, decide buscar la empresa más reconocida en el medio y que cumpla con los mejores estándares de calidad. Luego de presentar la falla, Juan le solicita al taller que traten su vehículo con el mejor aprecio. Incluso, él mismo adquirió un plástico para envolver los cojines y el volante de modo que no se fueran a ensuciar.

Así, Juan es el tipo de cliente que pide que todos los repuestos que se le instalen a su vehículo sean originales; lo anterior sin importar el tiempo que se demoren o el costo que esto implique. También le interesa que las reparaciones se realicen con las herramientas adecuadas y que estas duren lo suficiente para que él pueda disfrutar de su vehículo.

A pesar de esto, cuando el presupuesto que se le presenta a Juan es elevado, en ese momento decide buscar alternativas de reparación con repuestos homologados, pero de alta confiabilidad; teniendo en cuenta el país de origen de los repuestos como primera medida.

Propietario de vehículo de gama media/baja

Saray Dávila, es una vendedora independiente de productos de calzado. Es cabeza de familia y de ella dependen económicamente sus 2 hijos. Su oficio se centra en la reventa de calzado y obtiene una comisión por cada contrato que se firma. Dentro de sus pasatiempos favoritos se encuentran los paseos familiares, los cuales realiza en una camioneta doble cabina con platón; la cual adquirió con el objetivo de realizar ella misma los traslados del producto desde la fábrica

hasta el comprador. De esta manera, también se ganaría el servicio de transporte de algo que ella misma ha vendido.

Dicha camioneta la adquirió con 10 años de uso y es de una marca poco conocida en el mercado, pues la introdujeron hace 1 año y aún no hay el respaldo suficiente en cuanto a repuestos. Adicionalmente, Saray busca que el vehículo se repare con la menor inversión posible, pues asegura que el servicio de traslado deja muy pocas ganancias y no desea tomar dinero de su salario para mantener el vehículo. Por otra parte, trata de alargar lo más posible las fallas para que la camioneta se repare solo cuando sea exclusivamente necesario.

Es por esto por lo que ella es un perfil de cliente que, o bien no puede acceder a los repuestos originales, o no cuenta con el presupuesto para realizar dicha inversión. Además, tiene en cuenta el costo por encima de los beneficios; siempre prefiere la reparación de las partes frente al cambio de estas, con la excepción de que el repuesto nuevo sea más económico y, por último, no desea realizar ningún otro tipo de inversión que no se requiera de forma inmediata.

3.1.4. Priorización de lista de producto

Luego de contar con toda la información relacionada anteriormente, se procedió a realizar la lista de producto. Para esto, se llevó a cabo un análisis de las técnicas de priorización. Dentro de los distintos métodos identificados, se debe resaltar que son independientes del tipo de metodología que se seleccione, ya que su estructura se relaciona más con el tipo de proyecto que se pretenda desarrollar y los recursos y participación de interesados con los que se cuente al momento de estructurarlos.

Zacarías (2019) indica que existen al menos 20 métodos de priorización relevantes. Estos, se han venido desarrollando a lo largo de los años en que las metodologías tradicionales y ágiles fueron identificando que los procesos existentes no eran suficientemente acertados para llevar a cabo algunos proyectos de desarrollo de software. Por lo tanto, estableció una “tabla periódica”,

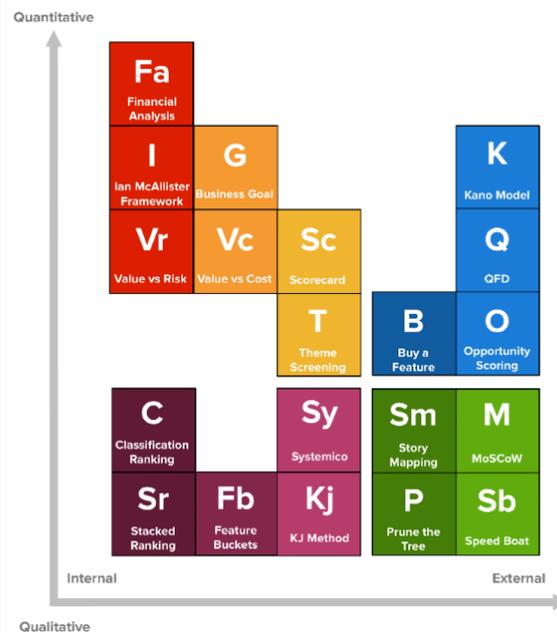
reflejada en la figura 6, de las técnicas de priorización en las cuales se ubican cada una de ellas con relación a la forma, cualitativa o cuantitativa, de considerar los aspectos relevantes. Adicionalmente, el autor indica que estas técnicas se podrían clasificar como aquellas que se consideran internas y externas; siendo las externas, aquellas en que las personas ajenas al equipo de trabajo son las que realizan la priorización. En forma gráfica, el modelo se podría plasmar de la siguiente manera:

Técnicas internas <- Usted – Equipo de trabajo – Interesados – Cliente -> Técnicas externas

También, el autor indica que las técnicas externas son mejores cuando se priorizan resultados abstractos, en contraparte, las internas convienen cuando se tienen soluciones concretas. A manera de resumen, se muestran a continuación las técnicas que se podrían desarrollar dependiendo del alcance del involucramiento y la forma de priorización.

Figura 6.

Tabla periódica de las técnicas de priorización de listas de producto.



Nota. (Zacarías, 2019)

Técnicas externas

- Cuantitativas:
 - El modelo KANO (The KANO model).
 - Despliegue de función de calidad (QFD: Quality Function Deployment).
 - Puntuación de oportunidad (Opportunity Scoring).
 - Comprar una característica (Buy a Feature).
- Cualitativas:
 - Mapeo de historias (Story Mapping).
 - MoSCoW (must, should, could, wont).
 - Podar el árbol del producto (Prune the Product Tree).
 - Lancha rápida (Speed Boat).

Técnicas internas

- Cuantitativas:
 - Análisis financiero (Financial analysis)
 - Valor presente neto (Net Present Value (NPV)).
 - Tasa interna de retorno TIR (Internal Rate of Return (IRR)).
 - Periodo de recuperación de descuentos (Discounted Payback Period).
 - Marco de trabajo de Ian McAllister (Ian McAllister's Framework).
 - Impacto en el objetivo comercial (Impact on Business Goal).
 - Valor VS Riesgo (Value vs. Risk).
 - Valor VS costo (Value vs. Cost).
 - Tarjeta de puntuación (Scorecard).
 - Proyección de temas (Theme Screening)
- Cualitativas

- Clasificación de lista (Classification Ranking).
- Modelo sistémico (Systemic Model).
- Listado apilado (Stacked Ranking).
- Balde de características (Feature Buckets).
- Método KJ (KJ Method).

Ahora, en lo que al proyecto respecta, se realizó la profundización en aquellas técnicas externas, dado que, en este caso, es el cliente el que está desarrollando la priorización y a partir de la información recolectada se define de forma abstracta lo que se desea obtener del producto. Adicionalmente, respecto al uso de técnicas cualitativas o cuantitativas, el autor recomienda que no hay que dejarse guiar por los métodos cuantitativos, a pesar de la confianza que puedan brindar los números; no obstante, cualquier escenario es posible siempre que se cuente con la información adecuada.

Inicialmente, se tiene el mapeo de historias. El autor indica que este proceso se lleva a cabo cuando una lista, meramente dicha, no es suficiente para organizar y priorizar el trabajo. Por lo que se sugiere realizar varias columnas de trabajo en las que se ubicarían las historias de usuario; luego, horizontalmente se ubican las historias de usuario en forma secuencial, pudiéndose agrupar aquellas similares en “actividades”. Uno de los inconvenientes al realizar este método es que se requiere contar con el equipo desarrollador para definir la secuencialidad y similitud, por lo que se descartó este método. Además, podría dejar por fuera información recolectada distinta a las historias de usuario.

El método “podar el árbol”, consiste en que las historias se ubican en forma de árbol. Cada rama corresponde a una funcionalidad del software y sus subdivisiones son requerimientos de cada funcionalidad. Luego, intervienen el cliente y los interesados para decidir qué quitar. Al final, se

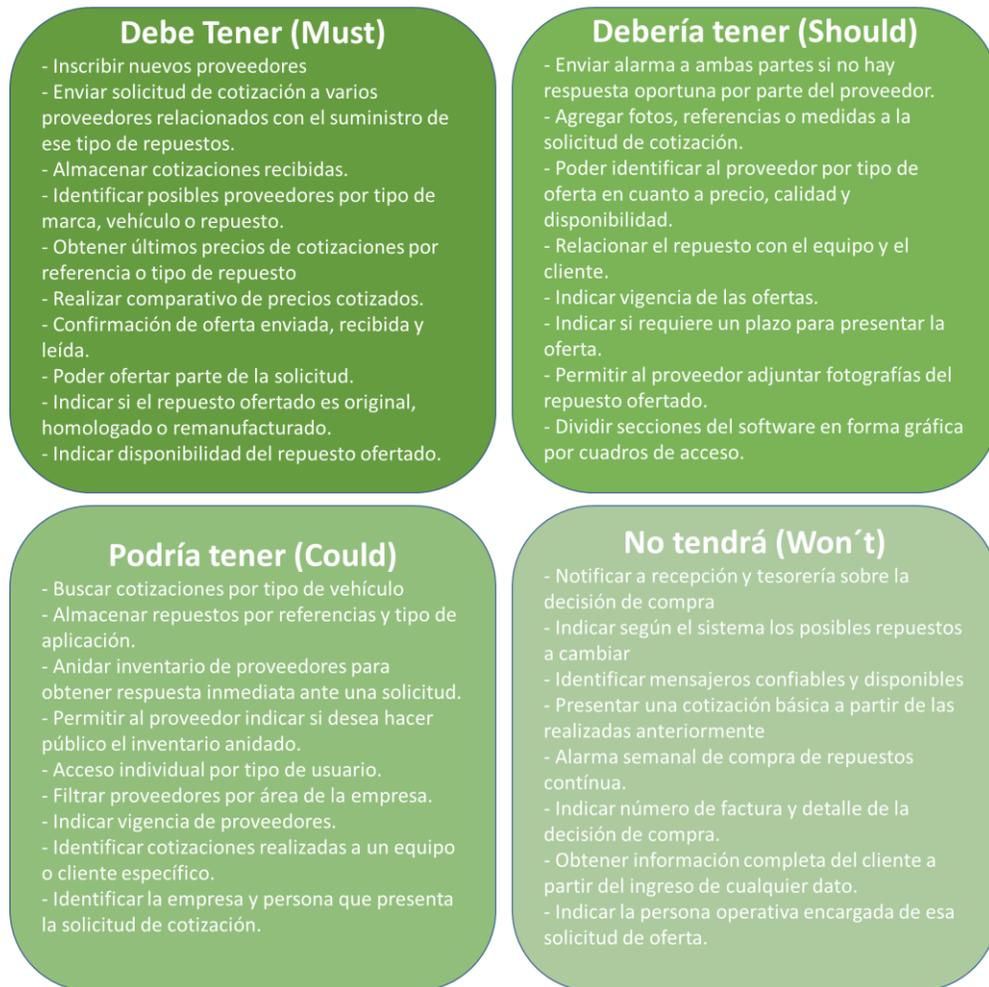
debe contemplar y si el árbol está balanceado (gráficamente hablando). El objetivo de este método consiste en pulir el resultado final de acuerdo a las necesidades del mercado. No obstante, se cuenta con la misma problemática del mapeo de historias para desarrollarlo.

Ahora, el método de “lancha rápida” supone que un bote, que es el producto, debe ir rápido. Sin embargo, cada funcionalidad que no es del agrado del cliente es un ancla que reduce rapidez; por lo que se deben empezar a quitar aquellas que no aportan significativamente al producto final o con las que se sienten inconformes. Así, se obtiene un producto más acorde a las necesidades del usuario. Dentro de lo que compete a este proyecto, se consideró que este método podría ser menos comunicativo que el analizado en el siguiente párrafo.

El método MoSCoW consiste en subdividir todas aquellas características del producto identificadas en la recolección de información; esto con el fin de ofrecer al equipo desarrollador una idea más clara de qué es importante y que no. De este modo, el listado se subdivide en 4 secciones: aquellas que debe tener (must), aquellas que debería tener (should), aquellas que podría tener (could) y las que no es necesario que tenga (won't). Si bien algunos otros pueden decantar mejor la información, se consideró este como el método más apropiado de acuerdo con la información y equipo de trabajo con el que se cuenta al momento de la realización de este análisis. A continuación, en la figura 7, se muestra la matriz desarrollada:

Figura 7.

Lista priorizada del producto a partir del método MoSCoW.



3.1.5. Definición del equipo de trabajo

Como final de capítulo, se realizó una consulta a un experto con el fin de obtener información real respecto a las personas que podrían hacer parte del equipo desarrollador. Para esto, se consultó al ingeniero de sistemas Kevin Arias. A continuación, se resaltan los puntos clave y útiles para esta sección. La entrevista completa se puede consultar en el apéndice D. De acuerdo con el entrevistado, se debe plantear inicialmente que, tanto si la compañía contrata a una empresa externa para la realización del software o incluye directamente a su personal en la nómina, existen ciertos perfiles que se deben tener en cuenta como mínimo. Adicionalmente, hay otros roles que entrarían

a formar parte del equipo siempre que se requieran. Un componente relevante por tener en cuenta respecto al perfil de los integrantes es que se considera que deba enfocarse en la proactividad y disposición para trabajo en equipo; situación que se analizará en profundidad cuando se identifiquen las herramientas de seguimiento adecuadas. Por otra parte, la dedicación del personal es un factor importante por tener en cuenta ya que no es necesario que todos hagan parte constante de la realización del proyecto. Lo mencionado, se expresa de manera más clara en la tabla 7.

Tabla 7.

Equipo de trabajo

Rol	Profesión	Dedicación	Requisitos	Tareas
Desarrollador Backend	Técnico programador	Tiempo completo	Saber programar	Manejo de datos Conexión con base de datos Procesos de negocio Reglas de negocio
Desarrollador Frontend	Técnico programador	Tiempo completo	Saber programar	Diseño de interacción (botones, chat, accesos, etc.)
Quality Assurance	Técnico programador.	Tiempo parcial	Preferiblemente al con final de conocimientos en cada fase programación. lograda.	Pruebas de operabilidad del software.

Gerente de proyecto	Profesional especializado	Tiempo parcial	Conocimientos en programación para gestión del equipo	Contacto con el cliente
----------------------------	---------------------------	----------------	---	-------------------------

Para proyectos de mayor complejidad

Diseñador / Diseñador. Experiencia de usuario	Diseñador.	Tiempo completo al principio, luego tiempo parcial	Experiencia en diseño de interfaz específica para mejorar la experiencia del usuario. Preferiblemente con conocimientos en diseño, programación y manejo de software.	Diseño de interfaz
--	------------	--	---	--------------------

Líder técnico	Ingeniero de sistemas	Tiempo parcial	Experiencia en programación y en liderazgo de equipos técnicos	Liderar al Backend y Frontend. Definir la arquitectura del software a desarrollar.
----------------------	-----------------------	----------------	--	--

Analista de negocio	Ingeniero de sistemas con especialización o maestría.	Tiempo completo al principio, luego	Entender el negocio hacia el software.	Contacto con el cliente. Proyección de trabajo a realizar. Análisis de trabajo realizado.
----------------------------	---	-------------------------------------	--	---

tiempo Experiencia en
 parcial programación,
 creación y
 sistemas.

Experto en seguridad	Ingeniero de sistemas, especialista en seguridad informática	de tiempo parcial.	Experto en seguridad.	en Desarrollar seguridad del software.
-----------------------------	--	--------------------	-----------------------	--

3.2. Selección de metodología ágil.

3.2.1. Metodologías ágiles representativas

A continuación, se realiza un análisis de la forma en que se llevan a cabo los proyectos con las distintas metodologías ágiles existentes con el fin de identificar y seleccionar aquella que pueda brindar una mejor estrategia de desarrollo. Al final de la descripción de la operación de cada metodología se realizará un análisis para establecer la afinidad de estas con el proyecto en cuestión.

Inicialmente, se contempló estudiar la metodología Lean ya que, de acuerdo con la guía práctica ágil (PMI®, 2017), es la que envuelve y de la que nacen las demás metodologías. Touza (2020) indica que ésta, inició como un modelo enfocado en manufactura, que consideraba, entre otras cosas, que “la producción debe basarse en la demanda y no en la oferta. Su objetivo es retirar todo aquello que simplemente no agrega valor al producto final; teniendo en cuenta desde reuniones hasta actividades y metodologías.

Una característica por resaltar es que Lean se enfoca en la adaptación y optimización de los procesos que constituyen el desarrollo de un producto, mientras que las metodologías ágiles, toman este principio y lo enfocan en el cliente. No obstante, ambos conllevan a que el cliente logre lo que desea. Es así como en la gerencia de proyectos a través de Lean nace y se constituye en el sector industrial y, a partir de esto, las metodologías ágiles se enfocaron principalmente en el “desarrollo creativo y de software”. De esta manera, se procede a revisar las metodologías ágiles adecuadas para este proyecto.

Entonces, a continuación, se analiza la metodología Scrum. De acuerdo con Trigas (2020), “tiene como base la idea de creación de ciclos breves para el desarrollo, que comúnmente se llaman iteraciones y que en Scrum se llamarán Sprint”. El autor indica que consta de 5 fases que “definen el ciclo de desarrollo ágil” las cuales se identifican como: concepto, especulación, exploración, revisión y cierre. Adicionalmente, a través de las reuniones diarias es como se realiza la gestión de dichas iteraciones. Los elementos que componen esta metodología son:

- Product Backlog: es una lista en la que se identifican las necesidades del cliente.
- Sprint Backlog: comprende las tareas que se deben llevar a cabo durante los sprint.
- Incremento: define cuándo se considera que una parte del sprint se encuentra terminada y totalmente operativa.

Las historias de usuario hacen parte de la construcción de dicha metodología y que ya fueron llevadas a cabo en el capítulo anterior y que culminaron en la lista del producto. Ahora, el mismo autor indica que para la preparación del proyecto se lleva a cabo una etapa llamada Sprint 0 que es donde se trata de “comprender el caso de negocio con la finalidad de tomar decisiones que agreguen valor al producto”. De esta manera, se procede a realizar la planificación de los sprint de modo que se realiza una reunión en donde intervienen los distintos roles como: el dueño del

producto, el scrum máster y el equipo de trabajo. En ella, se establecen las tareas y se concluye en un gráfico que representan los avances en horas de trabajo y fechas de entrega.

A su vez, se realiza una planificación de las actividades en horas de trabajo a través de un ejercicio que se denomina póker de planificación; en el que, los roles mencionados, estiman la duración requerida de cada actividad. Finalmente, luego de cada sprint se realiza una reunión de retrospectiva que permite identificar aquellas falencias que se tuvieron y cómo se podrían generar cambios para mejorar en los próximos sprint.

Otra de las metodologías más conocidas es Kanban. De acuerdo con León (2020), ésta es usada cuando se tienen proyectos de difícil planificación y que además es impredecible la cantidad de trabajo que se debe desarrollar o la naturaleza de este. Arango Serna, Campuzano Zapata, & Zapata Cortes (2015) indican que su objetivo se centra en minimizar el trabajo en progreso (TEP) de modo que solo se requieran aquellos trabajos que sean requeridas en el siguiente proceso (León, 2020).

El mismo autor menciona que los pasos que se deben tener en cuenta para llevar a cabo la gestión, si se decide usar esta metodología, son: “Visualizar el flujo de trabajo... establecer el estado del flujo... dividir el trabajo en ítems pequeños... limitar el trabajo en curso... y medir el tiempo empleado en completar un ciclo completo”. Además, Germanov (2019) indica que “... utiliza señales visuales que calculan: qué producir, cuándo producir y cuánto producir”, de modo que “se basa en eventos y no en segmentos de tiempo”.

Una de las facilidades que ofrece esta metodología es que “permite ver qué está pasando con todos tus proyectos, tareas y equipo de un golpe de vista” (Berzal, 2016). Aunque ésta nació como una respuesta ante las demoras en procesos de manufactura, los métodos de seguimiento le han permitido aplicarse en distintos tipos de proyecto, como los de tecnología. Para esto, y como herramienta de seguimiento, en la guía práctica ágil (PMI®, 2017) se establece un tablero Kanban

que muestra 5 etapas en las que se dividen los trabajos en desarrollo, las cuales son: listo para empezar, desarrollo y pruebas, desarrollo terminado, pruebas de sistema y finalizado. Esto, unido a la lista de producto, permitirá llevar a cabo los proyectos de desarrollo de software.

Ahora, la metodología Scrumban, “nace de la combinación de principios de los métodos ágiles de gestión de proyectos más importantes en la actualidad: Scrum y Kanban” (OBS, Bussiness school, 2020). Así, el mismo autor indica que, en un contexto empresarial, las tareas se gestionan con Scrum y se corrigen con Kanban; de modo que su aplicación se enfoca en proyectos de alta complejidad como: “Proyectos de mantenimiento... en los que los requisitos varíen con frecuencia... o que surjan errores de ejecución”. Al tablero de Kanban se le podrían agregar etapas como probadas o entregadas y una posible contextualización de lo que se pretende realizar en dichas tareas.

Por otra parte, Germanov (2019) menciona que las iteraciones se realizan lo más cortas posibles, máximo de 2 semanas; de modo que, en un proyecto en el que se puedan presentar varios cambios, los equipos se puedan adaptar fácilmente. También, el autor hace referencia a que la planificación se realiza con base en la demanda y solo ocurre cuando se tiene un bajo número de tareas pendientes; así, la priorización se realiza cada vez que surja un activador de planificación. Adicionalmente, dicho autor manifiesta que el tablero de Scrumban, se compone de 3 columnas básicas: “Tareas pendientes, en proceso y listo”. Pero, como se mencionó en el párrafo anterior, las tareas pendientes se pueden desplegar en otras columnas como “Diseño, desarrollo, revisión y pruebas”. A través de los límites de trabajo en progreso (WIP) y tareas pendientes es como se gestiona el desarrollo. Por último, existe una característica que posibilita la “congelación de funciones”, que establece que evitaría agregar más características cuando se aproxima la fecha de entrega, así, solo se trabajará en las que ya se encuentren desarrolladas.

A continuación, se realiza de manera simplificada una perspectiva de las demás metodologías identificadas:

- Programación extrema (XP): se enfoca en “entregar el software que los clientes necesitan en el momento en que lo necesitan” a través de las variables “costo, tiempo, calidad y alcance” y conlleva cuatro fases principales: exploración, planificación, iteración y producción (Vila, 2016).
- Crystal: ofrece 5 metodologías dentro de su estructura: Clear, Yellow, Orange, Red y Maroon. Cada una de estas se adapta al tipo de proyecto que se tenga de modo que entre más oscuro el color es más complejo de aplicar. Presenta “ciclos de desarrollo incremental” con un máximo de 4 meses y enfocándose en la comunicación y cooperación (León, 2020).
- Design thinking: se enfoca en la resolución de problemas a partir de la reducción del riesgo y aumento de probabilidades de éxito. Cuenta con las siguientes fases: “empatizar con el problema experimentado, definir el problema en detalle, idear diferentes formas de resolver el problema, crear prototipos de una baja resolución de la solución y probar la solución con los clientes” (León, 2020).
- Running Lean: se encuentra enfocada en la innovación de modo que sea eficaz y rápida que tiene en cuenta la velocidad a la que se aplica y el aprendizaje que se obtiene. Se define a partir del proceso que conlleva la interacción con el cliente, agilidad y el uso eficiente y responsable de los recursos (Blank, 2019).
- Lean launchpad: propone que a través del desarrollo de prototipos rápidos se puede llegar al producto final. De esta forma, también involucra al cliente para que los supuestos se vayan ajustando de acuerdo a lo que se considera crítico. Conlleva 4 pasos:

“proceso de descubrimiento de clientes... validación de clientes... creación de clientes... y creación de la empresa” (Alzate, 2015)

Adicional a lo mencionado hasta el momento, existe una metodología que vale la pena resaltar. SAFe (Scaled Agile Framework) tiene un “enfoque dirigido hacia la implementación de la agilidad en las empresas, y no solo a los equipos como es el caso de Scrum” (Viewnext, 2020). El autor indica que esta metodología cuenta con tres niveles para su implementación. El primero es Equipo, que define cómo se articula cada miembro con los demás en una especie de organigrama, y utiliza técnicas de Scrum para llevar a cabo este relacionamiento. El segundo es Programa, que busca definir qué es lo que desea la organización y cuáles son los objetivos que persigue con el desarrollo de la solución; así, presenta una forma ordenada de cómo se van a llevar a cabo las tareas a través de una “cadena de mando”. Y el tercer nivel es Portafolio, que permite asociar los resultados esperados con el valor que le aporta a la organización; entonces, se depura aquello que no funciona, lo que concluye en una mayor eficiencia del trabajo.

Finalmente, se analiza una metodología que reúne varias de las mejores prácticas, se trata de Disciplined Agile Delivery (DAD). De acuerdo con Garzas (2015):

“se basa en los principios ágiles y combina las prácticas extraídas de Scrum, XP, Kanban, Lean... El objetivo de DAD es ayudar a las empresas a implantar los valores y principios ágiles no solo en desarrollo software, sino desde la definición de requisitos, pasando por el desarrollo, hasta el despliegue del software y contacto con el cliente.”

El mismo autor indica que dicha metodología contempla que las empresas analizan cuáles son las prácticas que van a ser útiles y no todas coinciden en lo mismo. Entonces, se establecen una serie de estrategias que le permite a cada proyecto tomar aquellas que se relacionan mejor con el resultado proyectado. Por lo tanto, en DAD no existe un único ciclo de vida para desarrollar las

cosas y otorga la posibilidad de escalar el proyecto cuando se presente una necesidad de crecimiento.

DAD cuenta con 3 ciclos de entrega. Comienzo, que es una etapa de corta duración donde se definen los preliminares; lo que equivaldría al Sprint 0 en scrum. Luego, entra la construcción, que es donde se lleva a cabo el proyecto y entran las distintas estrategias que permiten adaptarse mejor de acuerdo con el tipo de estructura deseada. Así, se llega al tercer ciclo que es transición, donde se presenta la solución a los interesados. Cabe resaltar que dichos ciclos se pueden llevar a cabo de forma iterativa y/o incremental.

A continuación, en la tabla 8, se identifican los puntos clave para tener en cuenta de cada una de las metodologías analizadas:

Tabla 8.

Metodologías ágiles representativas

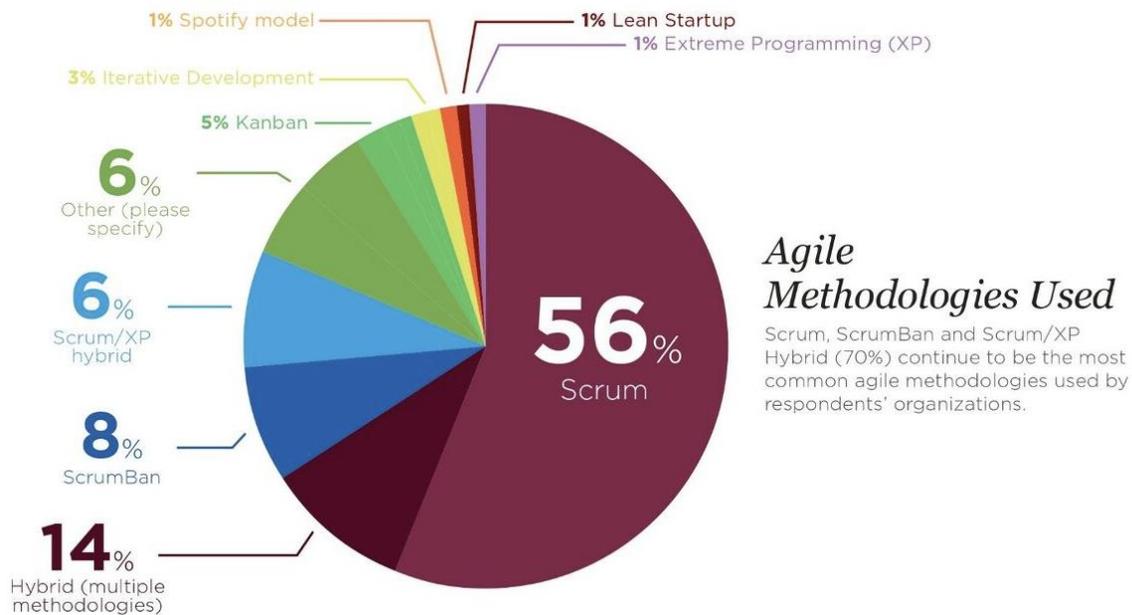
Nombre	Características
Lean	<ul style="list-style-type: none"> - Envuelve a todas las metodologías. - Se enfoca en adaptar y optimizar procesos. - Retira todo aquello que no genera valor al producto final. - Principalmente tiene aplicación industrial.
Scrum	<ul style="list-style-type: none"> - El trabajo se desarrolla en ciclos breves (iteraciones o Sprint). - Seguimiento: reuniones diarias, producto backlog, sprint backlog e incrementos. - Planificación de trabajo por roles.
Kanban	<ul style="list-style-type: none"> - Procura minimizar las tareas o trabajo en curso.

	<ul style="list-style-type: none">- Sin roles, todo el equipo debe tener la capacidad de realizar cualquier tarea.- Su planificación se basa en eventos: qué, cuando y cuanto se debe producir.
Scrumban	<ul style="list-style-type: none">- Para proyectos de alta variación, se gestiona con Scrum y se corrige con Kanban.- Iteraciones cortas que se adapten a los cambios.- A través del límite de trabajo en progreso permite congelar funciones si se aproxima la fecha de entrega.
XP	<ul style="list-style-type: none">- Enfoque en entregar al cliente cuando lo requiere.- 4 fases principales: exploración, planificación, iteración y producción.
Crystal	<ul style="list-style-type: none">- 5 metodologías, varía el color dependiendo de la complejidad del proyecto.- Trabajo por ciclos de desarrollo incremental enfocados en comunicación y cooperación.
Design Thinking	<ul style="list-style-type: none">- Resolución de problemas por reducción de riesgo y aumento de probabilidades de éxito.- 5 fases: empatizar, definir, idear soluciones, crear prototipos y pruebas.
Running Lean	<ul style="list-style-type: none">- Se basa en velocidad de aplicación y aprendizaje obtenido.- Se define por: interacción con el cliente, agilidad y uso eficiente y responsable de los recursos.

Lean Launchpad	<ul style="list-style-type: none">- Prototipos rápidos para llegar al producto final, superando cada vez un requerimiento crítico.- 4 fases: descubrimiento, validación, creación de clientes y de la empresa.
SAFe	<ul style="list-style-type: none">- Dirigido a implementación de agilidad en empresas.- 3 niveles: equipo, programa y portafolio.
DAD	<ul style="list-style-type: none">- Combina las mejores prácticas de las metodologías ágiles.- Parte de que cada empresa y proyecto tiene distinto enfoque.- Analiza cuáles prácticas son útiles.- 3 ciclos: comienzo, construcción y transición.

3.2.2. Selección de metodología

Inicialmente se contempló realizar un cuadro comparativo entre las metodologías analizadas. Sin embargo, se llegó a un punto en que las características que definen a cada una de estas no se relacionaban con las demás, lo que culminaba en un análisis innecesario. Por lo tanto, se procedió a estudiar aquellas que tuvieran un mayor impacto en el mundo. Para esto, Petrova (2019) acude al reporte de las metodologías más usadas en 2018, lo cual se aprecia en la figura 8:

Figura 8.*Metodologías ágiles más usadas en 2018*

Nota. (Petrova, 2019)

Como se puede observar, scrum se usa por más de la mitad de las organizaciones analizadas (cerca de 2000 organizaciones practicantes de ágil). Esto se relaciona directamente con el hecho de que ésta es la metodología más certificada hasta el momento y para la que se ofrecen más entrenamientos. Sin embargo, el autor menciona que un 78% de dichas empresas que usan Scrum, acuden a otros marcos de trabajo para complementar sus actividades.

En este punto, cabe resaltar que se pretende seleccionar aquella metodología que sea la más adecuada para el desarrollo del proyecto, teniendo en cuenta que se logre conseguir el talento humano requerido y en lo posible certificado para dicha labor. Desde una perspectiva general, las actividades realizadas en el capítulo anterior son la base que permitirán el inicio de varias de las metodologías analizadas, como Scrum, Kanban y DAD, entre otras. Cabe resaltar que en Scrumban no hay casi función del gerente; ya que, como no hay roles específicos, cada uno escoge qué tarea realizar. Respecto a esta metodología mencionada, hay una situación que puede ocurrir y es que

las personas escojan la misma tarea, no se actualice su estado o queden bloqueados porque nadie ha hecho una tarea.

Por lo tanto, se decide realizar la construcción del resto del proyecto a partir de la metodología ágil DAD (en adelante DAD), puesto que recoge las mejores prácticas de las demás y permite una mejor adaptación de acuerdo con lo que se desea realizar. Además, en concordancia con el análisis de las metodologías más usada, se considerará la posibilidad de recolectar distintas herramientas que complementen el objetivo propuesto.

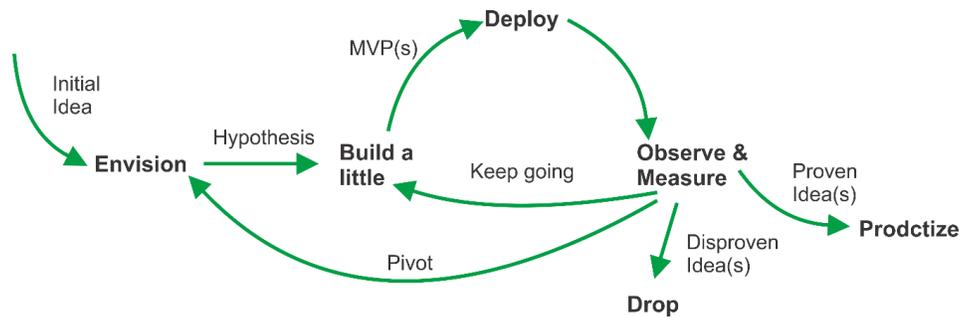
3.3. Actividades.

3.3.1. Producto mínimo viable (MVP).

Inicialmente, se debe considerar que DAD proporciona diferentes tipos de ciclos de vida del proyecto dependiendo de las características del problema a resolver y del conocimiento que se tenga sobre la verdadera solución del problema. Para esto, DAD (Ambler & Lines, 2019) relaciona 6 tipos de ciclo de vida: Ágil, Ágil de entrega continua, Lean, Lean de entrega continua, Exploratorio y Programa. De acuerdo con la descripción de cada uno, se seleccionó el ciclo de vida exploratorio que es el que se indica en la figura 9.

Figura 9.

Ciclo de vida exploratorio de DAD.



Nota. (Ambler & Lines, 2019)

Como se puede observar, luego de crear la hipótesis se debe realizar la construcción de una pequeña entrega, que será comprobable por el cliente en forma de producto mínimo viable (en adelante MVP). Según el autor, el MVP corresponde a un prototipo basado en la hipótesis, que no necesariamente corresponde al producto final y tiene como objetivo el verificar si el resultado es lo que el cliente desea o se debe reajustar. Por lo tanto, antes de crear las actividades se considera necesario definir aquellos requisitos que componen el MVP.

Ahora, teniendo en cuenta que en el capítulo anterior se llevó a cabo la fase exploratoria del alcance, se considera que el MVP se compone de los requerimientos de la sección “debe tener” (MUST) de la matriz MoSCoW. A manera de referencia, se mencionan nuevamente:

- Inscribir nuevos proveedores.
- Enviar solicitud de cotización a varios proveedores relacionados con el suministro de ese tipo de repuestos.
- Almacenar cotizaciones recibidas.
- Identificar posibles proveedores por tipo de marca, vehículo o repuesto.
- Obtener últimos precios de cotizaciones por referencia o tipo de repuesto.
- Realizar comparativo de precios cotizados.

- Confirmación de oferta enviada, recibida y leída.
- Poder ofertar parte de la solicitud.
- Indicar si el repuesto ofertado es original, homologado o remanufacturado.
- Indicar disponibilidad del repuesto ofertado.

Las actividades identificadas a continuación, corresponderán con la sección “Build a Little” que se indica en la gráfica del ciclo de vida exploratorio de DAD.

3.3.2. Definición de épicas

Debido a la naturaleza de la definición del proyecto, DAD (Ambler & Lines, 2019) ofrece una alternativa para indicar el nivel de detalle del plan; dentro de estas, se cuenta con la opción de definir los sprint desde una perspectiva de alto nivel. Esto permitirá al equipo organizarse de acuerdo con la forma en que ellos consideren oportuno llevar a cabo cada actividad. Sin embargo, es necesario proceder primero a mencionar las épicas, las cuales se dividirán en funcionalidades, y estas últimas en tareas. Así, las tareas serán las que definan los sprint. Pichler (2014) menciona una plantilla que puede ser usada para definir claramente los objetivos de alto nivel que se presentarán al equipo de trabajo y que serán una guía de lo que se debe obtener luego de terminar cada producto. Por lo tanto, se procede a la creación de las épicas desde esa perspectiva. Algunas características deseadas fueron agrupadas dentro de estas ya que se consideran parte del desarrollo que se llevará a cabo dentro de este. El resultado se aprecia en las tablas 9, a 14.

Tabla 9.

Épica 1.

Diseño técnico.

N. Versión: 0

Objetivo:

Establecer los parámetros acordes a las necesidades presentadas en el producto mínimo viable.

Método:

El líder técnico debe definir los siguientes elementos:

- Arquitectura del software
- Infraestructura
- Lenguaje de programación
- Comunicación entre elementos y bases de datos
- Comunicación con el usuario
- Seguridad de la infraestructura

Métricas:

Validación de los elementos definidos con el equipo de trabajo.

Finalidad:

Al culminar esta épica se obtendrá la información necesaria para que el equipo desarrolle el software.

Tabla 10.

Épica 2.

Inscribir usuarios.

N. Versión: 0.1

Objetivo:

Crear bases de datos que permita la creación de usuarios y de los proveedores, en la que se permita agregar, modificar o eliminar la información y esta debe poder ampliarse si se requiere.

Método:

Funcionalidad 2.1: Implementación de la infraestructura

Comprende la implementación de hardware y software necesarios para el funcionamiento del producto.

Funcionalidad 2.2: Montaje de plataforma

Se debe permitir la creación de nuevos usuarios y su rol, permitiendo o denegando la posibilidad de modificación ciertos campos; dichos roles se dividen en:

- Cliente: Diferenciado por nombre de la empresa y nombre de la persona, cargo, celular y correo electrónico.
- Proveedor: Diferenciado por nombre de la empresa y persona de contacto. Debe permitir el ingreso de 2 tipos de información:
 - o Información básica: nombre, apodo, NIT, teléfono, correo electrónico, celulares, App de mensajería y número, método de comunicación de preferencia indicado por el proveedor.
 - o Información de productos: sistema para el que vende los repuestos (motor, inyección, turbo alimentador, refrigeración, frenos, transmisión, suspensión, eléctrico, pintura, hidráulico, accesorios), marca de repuestos, marca de vehículo o equipos a los que aplican los repuestos.

La información se debe almacenar en una base de datos y debe permitirse agregar a través de un archivo XML. A los usuarios se les debe poder asignar un usuario y contraseña para el ingreso al sistema.

Métricas:

Los usuarios entienden y se les facilita el ingreso de la información, tanto de forma manual cómo agregando el archivo XML.

Finalidad:

El cliente del software podrá realizar la inscripción de sus proveedores, que serán la base para la construcción de las demás épicas. La retroalimentación de la forma en que el cliente interactúa con el software arrojará la necesidad de cambios si se presentan.

Tabla 11.

Épica 3.

Solicitudes de cotización.

N. Versión: 0.2

Objetivo:

Con base en la información de productos, se debe permitir a los usuarios crear una solicitud de cotización y enviarla a una selección de proveedores específica.

Método:

Funcionalidad 3.1: Cotización

En esta funcionalidad se desarrollarán la base de datos que contendrá las cotizaciones que se envían a los proveedores. Además, se definirán los elementos con los que el usuario interactuará con esta funcionalidad.

La cotización se creará en un formulario en la que se inscriba la necesidad y la información del producto al cual aplica.

Funcionalidad 3.2: Recomendación de proveedores

El software debe recomendar los proveedores con base en la información de productos seleccionada en el formulario.

Debe permitir seleccionar todos los proveedores recomendados.

Funcionalidad 3.3: Comunicación de solicitudes

La solicitud debe enviarse al (los) método(s) de comunicación de preferencia seleccionado(s) en la información básica del proveedor.

La solicitud debe guardarse en una base de datos.

Funcionalidad 3.4: Notificación de lectura

El software debe presentar una retroalimentación que indique qué proveedores recibieron la oferta y cuales la han leído dentro del sistema.

Métricas:

Los usuarios entienden y se les facilita el proceso de creación de una nueva solicitud de cotización.

En el momento de indicar una información de producto, el software reduce la cantidad de proveedores recomendados.

Finalidad:

Se espera que el cliente pueda interactuar con el formulario creado. Las funcionalidades describen el proceso que debe llevar al cliente a determinar qué proveedores son adecuados para la solicitud que se está creando y el estado en que se encuentra luego de ser enviada.

Tabla 12.*Épica 4.***Recepción de cotizaciones.****N. Versión: 0.21****Objetivo:**

Crear base de datos que recopile la información recibida por parte del proveedor.

Método:

El proveedor debe poder ingresar a la plataforma y agregar la información respectiva a la solicitud creada. Dicha información debe contener lo siguiente y debe permitir ampliarse: Marca, original u homologado, precio y disponibilidad. El usuario debe poder indicar a qué proveedor se le realizará la compra.

Métricas:

La información recibida se puede visualizar en una tabla de forma ordenada.

Tabla 13.*Épica 5.***Comparativo de ofertas.****N. Versión: 0.22****Objetivo:**

Permitir al usuario ordenar la información recibida.

Método:

Las ofertas se podrán ordenar y filtrar por: fecha de respuesta, marca (original u homologado), menor o mayor precio y disponibilidad.

Métricas:

Los usuarios presentan facilidad al momento de interactuar con la tabla.

Tabla 14.*Épica 6.***Historial****N. Versión: 0.23****Objetivo:**

Presentar información almacenada.

Método:

A través de una tabla, se debe presentar al usuario la información de las ofertas que se han realizado anteriormente y esta se debe poder ordenar según el proveedor, marca del vehículo o equipo, marca del repuesto, original u homologado, rango de fechas.

Se deben permitir ampliar los filtros que desee cada usuario para ordenar la información histórica.

Métricas:

El usuario interactúa con la información histórica de forma fácil.

Respecto del resultado anterior, cabe resaltar que la estimación de las funcionalidades de las épicas 4, 5 y 6 no se lleva a cabo ya que esto dependerá de cómo avance el proyecto a medida que los sprints se desarrollen. Es decir, desde la perspectiva ágil, el enfoque de planeación cumple una definición de las primeras etapas; de modo que las siguientes son establecidas de forma general y

al llegar a estas durante la ejecución es donde se realiza la revisión, teniendo en cuenta el trabajo desarrollado hasta el momento.

3.3.3. Hoja de ruta ágil del producto

En consecuencia, se realiza la hoja de ruta ágil, la cual “ayuda a los equipos a concentrarse en el panorama general” (Justinmind, 2018). Dicho autor, ofrece una plantilla que permite organizar los sprint definidos anteriormente y sus características principales de forma gráfica. A continuación, en la figura 10, se presenta el desarrollo que se llevó a cabo a partir de las definiciones establecidas en las fases indicadas.

Figura 10.

Hoja de ruta ágil del producto.



3.3.4. Estimación de puntos

Ahora, se realiza un desglose de las tareas y su respectiva estimación de la duración de estas. Dicha actividad se realizará solo sobre las fases 0, 1 y 2; ya que las demás dependerán de cómo avancen y se estructuren las anteriores. Para esto, DAD (Ambler & Lines, 2019) recomienda (entre otras alternativas) que se realice una estimación de puntos. Primero, se identifica aquella tarea que se considera más sencilla y a partir de esta se puntúan las demás con respecto a su dificultad relativa. Scrum Manager (2016) recomienda que se realice a partir de la secuencia de Fibonacci (1,2,3,5,8,13, etc.) ya que, al realizar la estimación de tareas más complejas, el riesgo de subestimarlas aumenta; por lo que dicha secuencia ofrece un rango aceptable de margen de error.

Por último, se realiza la estimación de las tareas que se puedan desarrollar dentro de un sprint. Para esto, se decidió que su duración sería de dos semanas y se proyectará el trabajo a realizar en un mes. Además, se tuvo en cuenta que la suscripción a la plataforma podría conllevar medio día de trabajo, teniendo en cuenta que se debe investigar si se acopla con lo definido en la etapa 1. Por lo tanto, se contempló que por cada sprint se considerará una velocidad de desarrollo de 20 puntos, inicialmente. Por otra parte, aunque el sprint 2 podría completar los 20 puntos con la siguiente tarea, se decide no incluir más ya que se culminaría la fase 2 y se requerirían las pruebas del avance y posterior presentación al cliente. La tabla 15 muestra el resultado.

Tabla 15.

Desglose de tareas por fase y estimación de puntos.

Épica	Funcionalidad	Tarea	Estimación (puntos)	Desarrollo
1	Diseño técnico	Definir arquitectura del software	3	Sprint 1
		Definir infraestructura	3	

		Definir lenguaje de programación	3	19 puntos
		Definir comunicación entre elementos y bases de datos	3	
		Definir comunicación con el usuario	3	
		Definir la infraestructura	3	
2	Implementación de infraestructura	Suscripción a la plataforma	1	
		Desarrollo y configuración de la plataforma	8	Sprint 2
		Desarrollo de base de datos	5	20 puntos
	Inscribir usuarios	Desarrollo de plantillas de ingreso de información	5	
		Desarrollo de botones	2	
		Desarrollo de usuarios y roles	3	Sprint 3
3	Cotización	Desarrollo de base de datos de cotizaciones	5	20 puntos
		Desarrollo de formulario de cotización	8	
		Desarrollo de botones de interacción con el formulario	2	
	Recomendación de proveedores	Desarrollo de botón de filtro de proveedores por tipo de repuesto	1	
		Desarrollo de botón de filtro de proveedores por marca	1	

	Desarrollo de botón de filtro de proveedores por ubicación (disponibilidad)	1	Sprint 4
			15 puntos
Comunicación de solicitudes	Desarrollo de formato de correo electrónico y del mensaje para envío de notificación por este medio.	3	
	Establecimiento del enlace de conexión con teléfono y creación del mensaje para envío de WhatsApp y/o mensaje de texto.	5	
Notificación de lectura	Desarrollo de notificación de lectura de oferta solicitada por correo electrónico.	3	
	Desarrollo de notificación de lectura de oferta al ingresar el proveedor a la plataforma.	3	

3.4. Capítulo 4. Recursos de seguimiento.

Dentro de toda la gama de recursos de seguimiento con los que se cuenta en las metodologías ágiles, es importante considerar que, de la forma en que se realice, se creará un perfil del equipo de trabajo. De acuerdo con la entrevista que se realizó al experto, un equipo productivo puede llegar a terminar las tareas antes de tiempo; pero es la proactividad la que definirá que el equipo comunique la finalización de dichas tareas para abordar las siguientes. Es entonces donde toma fuerza la necesidad de que las actividades se enfoquen en esta última cualidad; así, se logrará un seguimiento controlado que no llegue a asfixiar las buenas intenciones del equipo, pero tampoco

que trabaje descontroladamente. No obstante, desde la perspectiva del modelo adaptativo, es necesario dejar claro que las actividades podrán variar siempre que se consideren innecesarias o incompletas durante la ejecución.

DAD (2019) realiza una división de los recursos en términos del desarrollo del producto (plan the release) y del trabajo de equipo (Evolve way of working, WOW). Adicional, se deben desarrollar aquellos documentos con los cuales se realizará seguimiento a las pruebas por parte de los usuarios. Entonces, desde esta perspectiva se realiza la división de los recursos de la siguiente manera: Seguimiento del avance, seguimiento del equipo y pruebas. Dichas secciones se describen a continuación.

3.4.1. Seguimiento del equipo

A través de estas herramientas, se logra obtener información clave que conlleva a que el equipo de trabajo se encuentre conforme con su trabajo y las tareas que desarrolla; siendo este el primer valor del manifiesto ágil. Su enfoque principal es ver cómo está trabajando el equipo y qué impedimentos tiene para continuar o posiblemente mejorar.

DAD (2019) cuenta con dos secciones orientadas al buen desarrollo del equipo de trabajo. En la primera de ellas, llamada “producir una solución potencialmente consumible” se refiere en primera instancia a las reuniones diarias, llamadas en inglés “daily standups” por el hecho de realizarse de pie y en un corto periodo de tiempo (15 minutos) con el fin de obtener solo la información clave necesaria. En esta, se ubican tarjetas (físicas o digitales) en tres columnas con el siguiente nombre: ¿Qué hice ayer? ¿Qué voy a hacer hoy? ¿Qué impedimentos tengo? Así, el líder del proyecto deberá organizar lo que considere conveniente para que el equipo de trabajo pueda desarrollar las tareas. En la tabla 16 se muestra el modelo de plantilla a utilizar.

Tabla 16.

Plantilla para las reuniones diarias.

¿Qué hice ayer?	¿Qué voy a hacer hoy?	¿Qué impedimentos tengo?
----------------------------	----------------------------------	-------------------------------------



La siguiente actividad recomendada dentro de la misma sección corresponde a la planeación de la iteración o del sprint. Ésta se lleva a cabo al inicio de cada sprint de modo que el equipo identifique y entienda cuál es el trabajo que se va a realizar. Sandoval (2016) indica que, inicialmente “el Product Owner presenta en orden ítems del Product Backlog al Equipo de Desarrollo, y el Equipo Scrum completo colabora para comprender el trabajo a realizar.” Luego, el equipo determina lo que está en capacidad de realizar teniendo en cuenta: el avance que presenta el proyecto en ese incremento y el comportamiento del equipo en los pasados sprint en términos de desarrollo y capacidad. Esto permite que nadie más que el equipo decida qué trabajo va a realizar. Luego, el mismo autor indica que se procede a la segunda parte, donde se estudia cómo va a ser desarrollado ese trabajo. Entonces, el equipo realiza una mínima planificación para acordar la forma en que van a trabajar inicialmente; y las actividades posteriores se van descomponiendo conforme avanza el desarrollo. Esto permite crear una solidez como equipo autogestionado al mismo tiempo que otorga confianza en la culminación de las tareas de cada sprint propuesto.

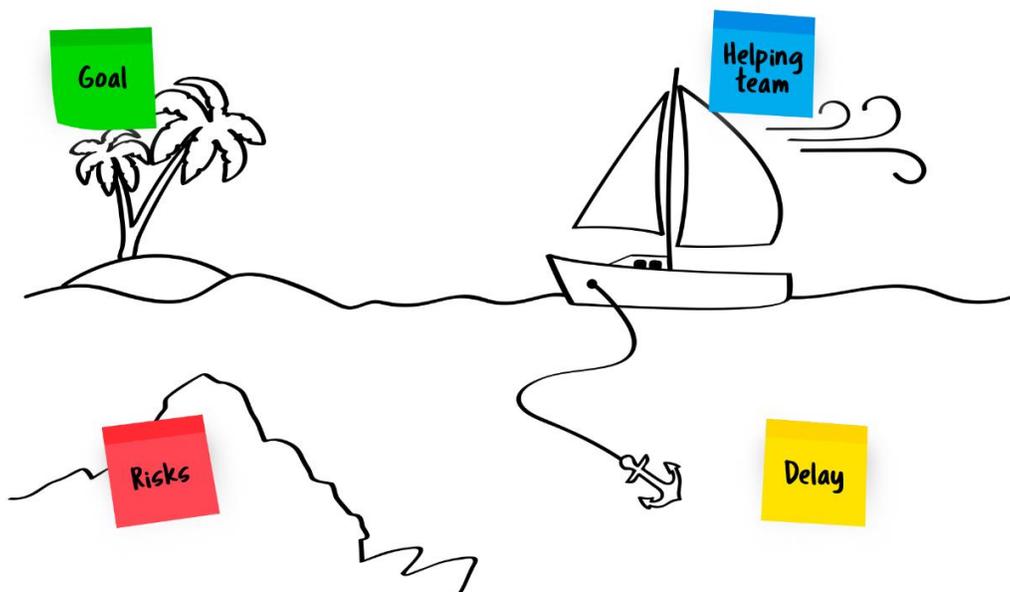
Ahora, se lleva a cabo la siguiente actividad que corresponde a la revisión de la iteración. IBM (2006) indica que ésta se puede realizar al terminar la planeación de la iteración o durante el desarrollo del sprint. Su objetivo consiste en asegurarse de que el plan de trabajo propuesto durante la planeación se está llevando a cabo; además, permite identificar defectos no detectados en el sprint que puedan estar afectando el normal desarrollo de las tareas.

Por último, en la segunda sección relacionada en DAD (2019) que mencionan como “identificar mejoras potenciales” se propone la realización de la retrospectiva. Una de las formas de llevarlo a cabo es a través de la plantilla del barco, mostrada en la figura 11, sugerida por Beattie & Takane (2017). Se desarrolla junto al equipo de trabajo y se ubican sobre un dibujo las actividades teniendo en cuenta la relación que se indica a continuación:

- El viento: lo que ayuda al equipo y permite el avance.
- El ancla: lo que lo demora o lo retrasa.
- El iceberg: los riesgos que puedan presentarse.
- El objetivo (isla): a donde se desea llegar en términos del sprint o del producto.

Figura 11.

Plantilla para la actividad de retrospectiva.



Nota. (Beattie & Takane, 2017)

3.4.2. Seguimiento del avance

Constituye la primera forma de seguimiento del proyecto. Una de las principales herramientas de las metodologías ágiles en general y que a su vez es sugerida por la metodología seleccionada es el tablero Kanban. DAD (2019) recomienda usarlo como parte de la visualización del proceso existente, ya que se pueden presentar las tareas de cada sprint en tableros independientes de tres o más columnas, de las cuales se parte inicialmente con: Lista de tareas, en proceso y hecho. A su vez, dicho tablero permite funcionar como radiador de información, el cuál es sugerido como parte de la comunicación de la visión. Éste último instrumento otorga, entre otras cosas, una conexión con el cliente de modo que sepa en qué proceso va sin que necesariamente consulte al equipo de trabajo; lo cual puede llegar a ser un factor distractor dentro de la dinámica del agilismo.

No obstante, es necesario que el equipo pueda identificar cómo el avance de las tareas va culminando en cada sprint en términos de puntos de historia. Dicho proceso se puede llevar a cabo a través de la gráfica de quemado descendente (burndown chart), que contrasta los puntos con la duración de cada sprint. La plataforma digital Trello permite la unión de los requerimientos mencionados anteriormente; pues los tableros creados se pueden editar por cualquier miembro del equipo de trabajo, al mismo tiempo que permite otorgar permisos a externos para que visualicen el trabajo que se está realizando. También, cuenta con programas que se adicionan a la plataforma y permiten agregar puntos de historia a las tareas programadas; a la vez que se crea la gráfica de quemado descendente respecto de la programación estimada de cada sprint. A continuación, en las figuras 12 a 16, se muestra la creación de 2 tableros correspondientes a los sprint 1 y 2; simulando un trabajo en progreso en cada uno de ellos (sin implicar que se desarrollen al mismo tiempo) y la gráfica resultante.

Figura 12.

Simulación del desarrollo del sprint 1 en Trello.

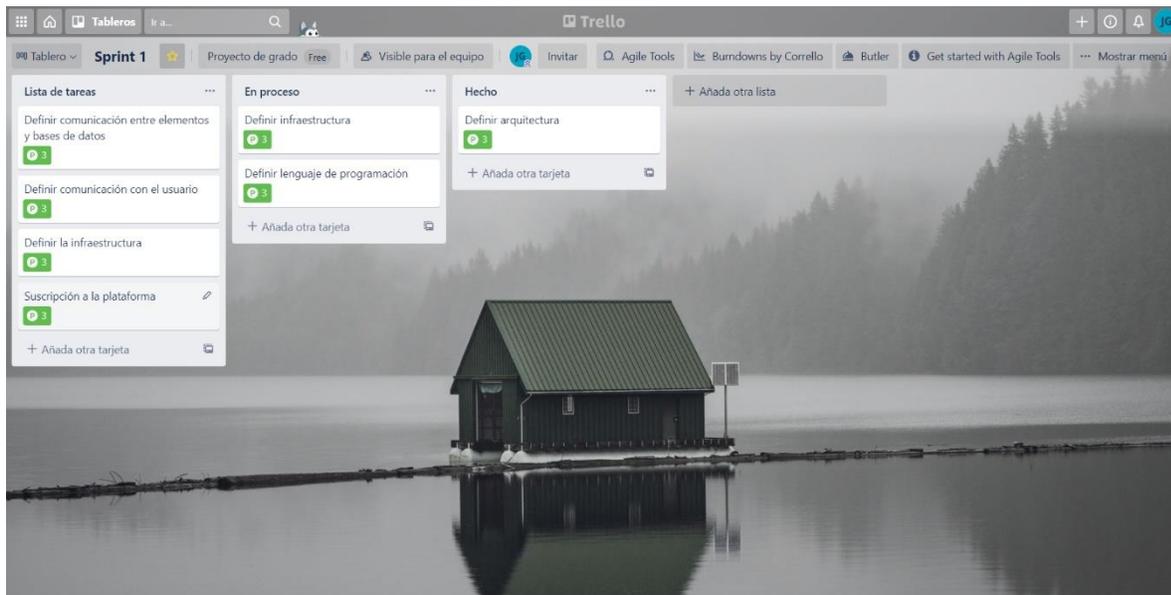


Figura 13.

Definición de puntos de historia para las tareas.

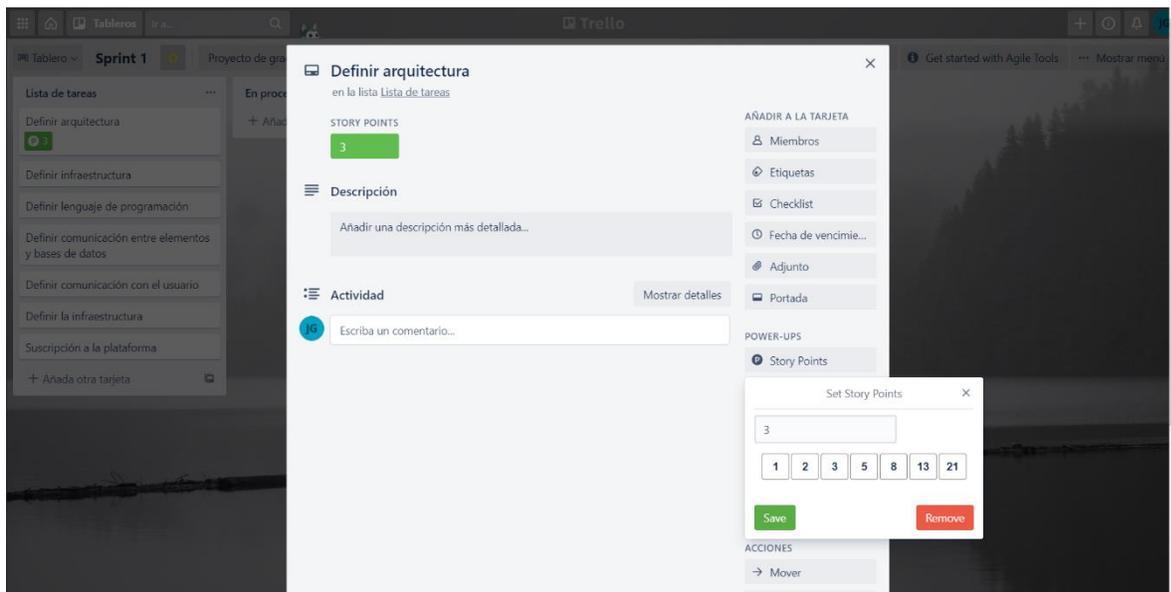


Figura 14.

Gráfica de quemado descendente para el sprint 1 en progreso.

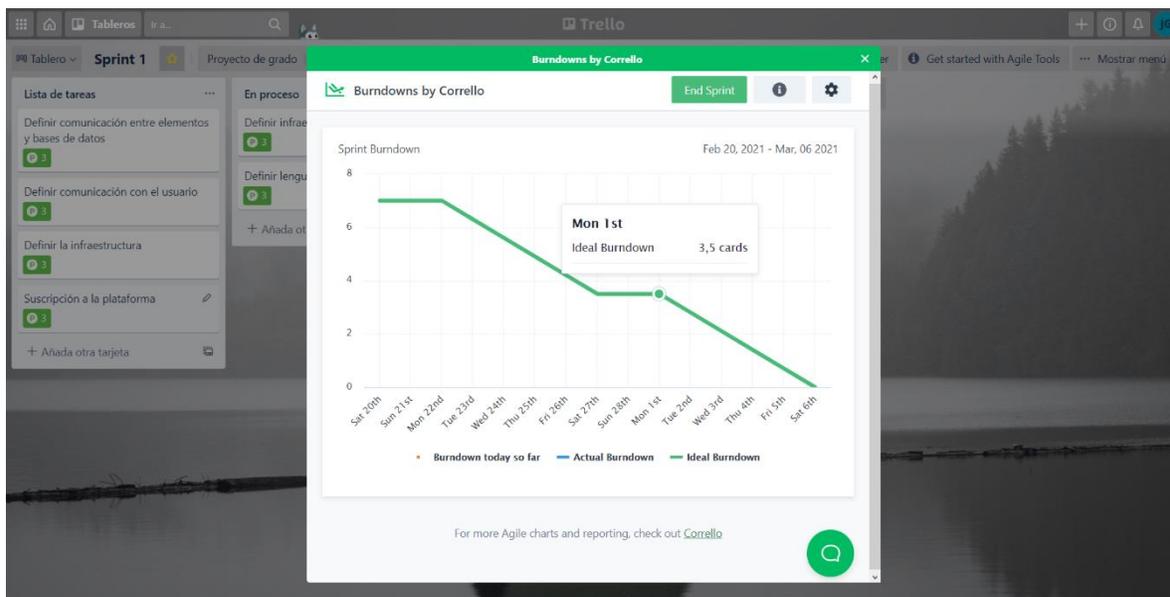


Figura 15.

Simulación del desarrollo del sprint 2 en Trello.

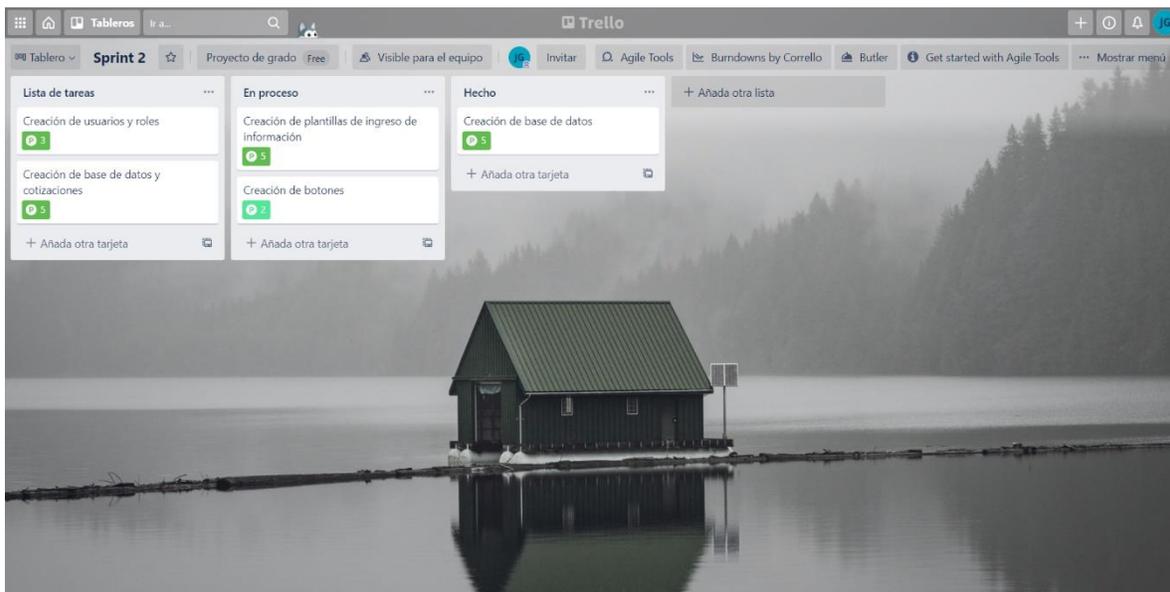
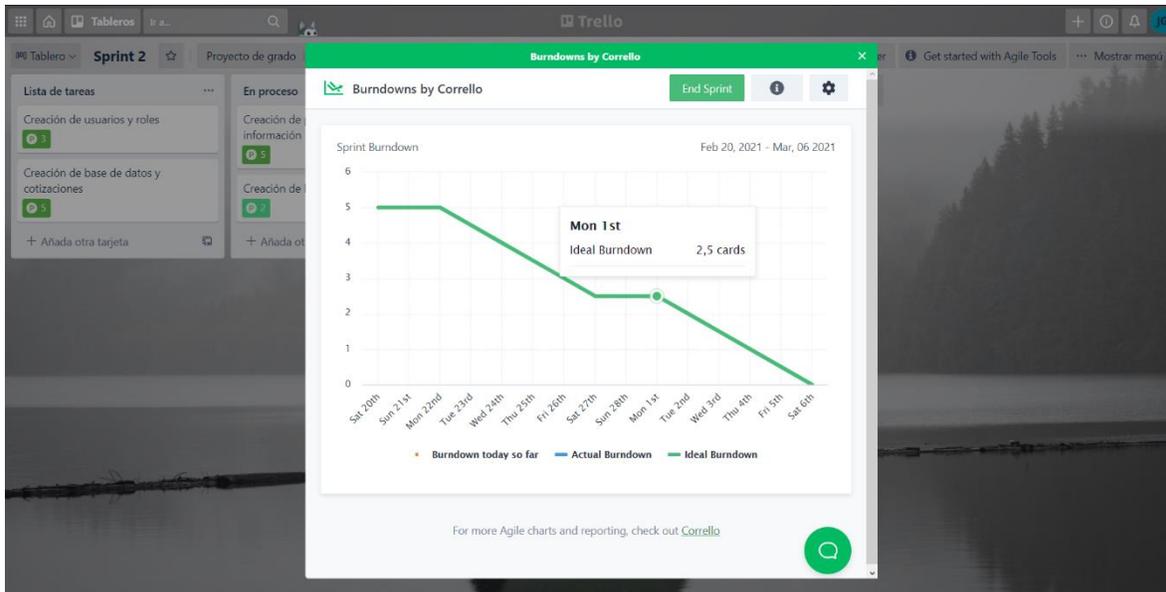


Figura 16.

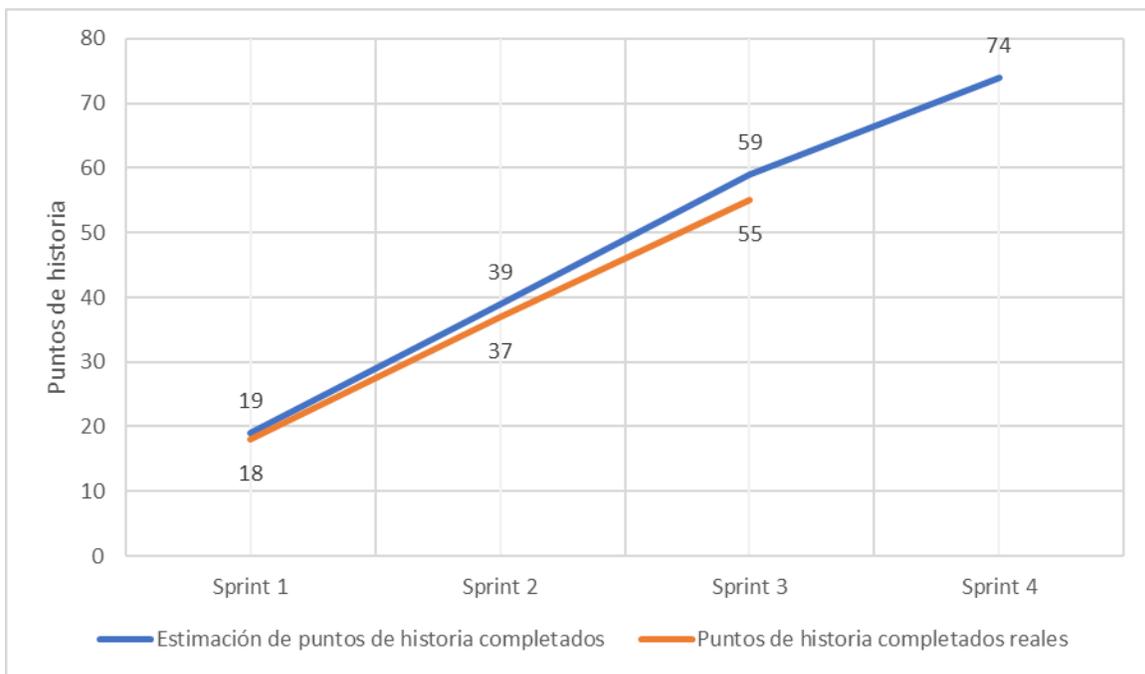
Gráfica de quemado descendente para el sprint 2 en progreso



También existe la Gráfica de quemado ascendente, la cual permite medir el avance del progreso general, comparando la cantidad de sprint con los puntos de historia completados. A su vez, refleja la velocidad del equipo, de modo que, si es mayor a la considerada, es posible que ellos trabajen de forma más eficiente o que las tareas hayan sido sobreestimadas en la puntuación (Canosa, 2019). A continuación, en la figura 17, se muestra la imagen que representa una simulación del avance del proyecto en el sprint 2 y con un retraso en el avance, medido en puntos de historia. El ejemplo presenta un caso en el que el equipo está presentando retraso al completar las tareas asignadas para cada sprint. Entonces, se debe indagar sobre las posibles razones que podrían conllevar a que esto ocurriera; tales como: falta de ciertas habilidades en el equipo de trabajo, cambios significativos en las historias de usuario, lista de producto mal refinada, entre otros.

Figura 17.

Simulación de medición del avance del proyecto con la gráfica de quemado ascendente.



Otras herramientas que son mencionadas por la metodología, como: revisión de hitos, diagrama de Gantt de alto nivel o detallado, diagrama PERT y salidas deseadas, entre otros, no se consideran convenientes de acuerdo con el tipo de proyecto que se está planeando. Esto con base en que, la perspectiva de agilismo que se ha venido manejando, no permite una creación de hitos o tareas pues se desconoce cómo llegue a cambiar el proyecto luego de que se presenten los avances al cliente.

3.4.3. Pruebas

Como último recurso de seguimiento, se considera necesario incluir aquellos que se enfoquen en los resultados. DAD (2019) ofrece una sección llamada “desarrollo de software” en la que menciona el desarrollo guiado por comportamiento (behavior driven development). Vergara (2019) indica que, más allá de ser una prueba del software realizado, se trata de desarrollarlo a

partir de la interacción esperada. Para esto, uno de los lenguajes usados es del tipo “Cómo-Deseo-Para (Role-Feature-Reason)”, que permite al equipo crear el producto a partir de la interacción esperada. Para esto, en el momento del desarrollo se tendrán en cuenta las historias de usuario que correspondan al mínimo producto viable identificado. Luego, se realizan las pruebas luego del desarrollo (test after development). Para esto, se diseñó una plantilla que puede ser usada tanto por el Quality Assurance como por los usuarios que el cliente determine. En las tablas 17 y 18, se aprecia un ejemplo que pudiera presentarse durante la realización de una prueba.

Tabla 17.

Ejemplo 1 del uso de la plantilla de registro de pruebas.

Versión: 0.1	Número de prueba: 4
Usuario	Quality Assurance
Acción realizada	Ingreso de cédula de un usuario
Problemas encontrados	Se permitió el ingreso de letras donde deben ir números
Sugerencias de modificación	Restringir el uso de cierto tipo de caracteres en los distintos campos e indicar al usuario el formato de este.

Tabla 18.

Ejemplo 2 del uso de la plantilla de registro de pruebas.

Versión: 0.2	Número de prueba: 2
<i>Usuario</i>	Jefe de taller
<i>Acción realizada</i>	Diligenciado de formulario de cotización

<i>Problemas encontrados</i>	Luego de presentar los proveedores recomendados, se observa que excluye a algunos que se recuerda que puedan ofertar.
<i>Sugerencias de modificación</i>	Presentar opciones al usuario sobre la posibilidad de incluir a un proveedor en una categoría desde esta sección.

De esta forma, se podrá otorgar una retroalimentación acertada al equipo de trabajo sobre la operabilidad del producto respecto de la fase de desarrollo en la que se encuentre. Es entonces de esta forma, como se llega a la planificación del desarrollo desde una perspectiva general y luego del proceso llevado a cabo. Partiendo inicialmente de una visión temprana y de alto nivel sobre el producto terminado, luego, explorando las necesidades de los usuarios y/o del cliente final para, a partir de esto, estimar una proyección del mínimo producto viable; entonces, proceder a seleccionar la metodología adecuada al problema y proyectar las actividades requeridas para los primeros meses del desarrollo. Y, por último, identificando cuáles herramientas son las adecuadas para el seguimiento del proyecto.

4. Conclusiones

Una de las herramientas principales a la hora de realizar la implementación de una metodología ágil, es el manifiesto ágil. En este, se pueden identificar los principios bajo los cuales se rigen estas metodologías y cómo esta perspectiva puede llegar a cambiar la forma de trabajo actual en los proyectos.

Ágil no es rápido, es entregar valor de forma temprana para obtener una retroalimentación casi constante por parte del cliente. Es importante resaltar al equipo de trabajo cómo este cambio de perspectiva va a mejorar el resultado final y, no necesariamente, a obtenerlo anticipadamente.

Se lograron identificar una alta diversidad de metodologías ágiles con enfoques distintos respecto al tipo de solución que se plantea, la forma de trabajo requerida o al nivel de incertidumbre o dificultad con el que se cuente en cada proyecto.

Existen distintos métodos y técnicas que pueden ser usados en cada metodología y que aportan de forma significativa a la construcción de un escenario ágil desde la perspectiva de sus principios. Dichos procedimientos, variarán dependiendo del grupo de personas con el que se cuente y de la fase en que se encuentre el avance del proyecto.

La tabla periódica de la priorización es una herramienta muy importante que tiene en cuenta los interesados que están haciendo la evaluación. Su uso debe explorarse más a fondo y las actividades propuestas en cada escenario deben estudiarse en su totalidad previo a cada encuentro.

Las metodologías híbridas, puede representar una solución a diversos problemas actuales que se viven durante la evolución de los proyectos. La posibilidad de mezclar conceptos predictivos y ágiles lograrán mejorar aquellos procesos que han llegado a su límite desde una sola perspectiva.

5. Recomendaciones

Siempre, para cada proyecto que se pretenda realizar, se debe revisar qué tipo de solución se requiere y cuál es la mejor forma de explorarlo a partir de las metodologías existentes. No es recomendable considerar que una sola metodología abordará todo tipo de problemáticas, por ejemplo, al interior de una organización.

Adicionalmente, es necesario estudiar las opciones metodológicas de forma objetiva. Puede que una de estas sea la más usada, incluso a nivel mundial, pero no necesariamente es la que se requiere para todo tipo de soluciones.

Las herramientas de trabajo estudiadas durante este proyecto tienen un mismo objetivo, pero varían significativamente. Si bien el enfoque se orienta a la reducción de la incertidumbre en distintas etapas del desarrollo del proyecto, se deben seleccionar dependiendo del número de personas con el que se cuente, el tipo de profesionales, el tiempo destinado a su uso y, entre otros, la información que se desea recibir como retroalimentación.

La implementación de ágil debe ser consensuada y no impuesta. Solo de esta manera se podrá asegurar que el equipo de trabajo entiende de qué se trata el cambio y manifieste su interés en llevarlo a cabo.

Como el primer principio del manifiesto ágil lo indica, es recomendable que todas las acciones permitan a la organización centrarse en el equipo; de modo que éste obtenga las mejores condiciones posibles que se le puedan otorgar para su el desarrollo de su trabajo.

Referencias Bibliográficas

- Alzate, A. (2015). *¿Qué es Lean LaunchPad?* Recuperado el 29 de 11 de 2020, de <https://www.icesi.edu.co/apps/2016/02/03/leanlaunchpad>
- Ambler, S., & Lines, M. (2019). *Choose your WOW!* Disciplined Agile Consortium.
- Ambler, S., & Lines, M. (2019). *Choose Your WoW!: A Disciplined Agile Delivery Handbook for Optimizing Your Way of Working (WoW)*. Disciplined Agile.
- ANDEMOS. (2020). *Informe vehículos mayo Colombia, 2020*. Recuperado el 07 de 2020, de <http://www.andemos.org/wp-content/uploads/2020/06/Informe-Vehiculos-2020-05.pdf>
- Beattie, T., & Takane, M. (21 de 11 de 2017). *Retrospectives*. Recuperado el 20 de 02 de 2021, de <https://openpracticelibrary.com/practice/retrospectives/>
- Berzal, I. (2016). *El método Kanban: gestión de proyectos y tareas para equipos*. Recuperado el 11 de 10 de 2020, de iberzal: <http://iberzal.com/metodo-kanban/>
- Blank, S. (23 de 02 de 2019). *El Método Running Lean ¿Cómo Nos Puede Ayudar A Innovar?* Recuperado el 29 de 11 de 2020, de <https://www.intrainnova.com/running-lean/#:~:text=El%20término%20Running%20Lean%20es,la%20implicación%20de%20los%20clientes.&text=Pretende%20agilizar%20el%20aprendizaje%20sobre,de%20recursos%20y%20tiempo%20posibles>.
- Canosa, A. (24 de 04 de 2019). *Seguimiento de un proyecto ágil: Burn-up y Burn-down*. Recuperado el 03 de 03 de 2021, de <https://sospnt.com/blog/52-seguimiento-de-un-proyecto-agil-burn-up-y-burn-down>

CarroYa.com. (2020). *Todas las marcas de carros en Colombia*. Recuperado el 07 de 2020, de <https://www.carroya.com/marcas-carros/>

DANE. (2020). *Encuesta anual de comercio (EAC)*. Recuperado el 07 de 2020, de https://www.dane.gov.co/files/investigaciones/boletines/eac/Anexos_estruc_eac_2018.xlsx

García, J. (27 de 07 de 2020). *Design thinking: qué es, características y fases*. Recuperado el 11 de 10 de 2020, de EAE Business School: <https://retos-directivos.eae.es/design-thinking-que-es-caracteristicas-y-fases/>

Garzas, J. (06 de 03 de 2015). *Otra estrategia para escalar Scrum: Disciplined Agile Delivery (DAD)*. Recuperado el 29 de 11 de 2020, de <https://www.javiergarzas.com/2015/03/disciplined-agile-delivery-dad.html>

Germanov, I. (12 de 08 de 2019). *Kanban vs Scrum vs Scrumban: ¿Cuáles son las diferencias?* Recuperado el 22 de 11 de 2020, de <https://ora.pm/es/blog/scrum-vs-kanban-vs-scrumban>

González, C. (05 de 12 de 2019). *El sector nacional de autopartes vendió 13 billones de pesos en 2018*. Recuperado el 07 de 2020, de <https://www.semana.com/contenidos-editoriales/sobre-ruedas/articulo/resultados-del-sector-de-autopartes-en-colombia/616206>

IBM Corp. (2006). *Tarea: Revisión del plan de la iteración*. Recuperado el 20 de 02 de 2021, de https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/tasks/iteration_plan_review_6D20D94.html

Junquera, A. (24 de 10 de 2019). *Metodologías ágiles: ¿qué diferencia hay entre Scrum, Kanban y XP?* Recuperado el 11 de 10 de 2020, de Grupo Digital: <https://www.grupodigital.eu/blog/metodologias-agiles/>

Justinmind. (10 de 05 de 2018). *What is an agile roadmap? Product development tips*.

Recuperado el 10 de 01 de 2021, de <https://blog.prototypr.io/what-is-an-agile-roadmap-product-development-tips-3d81fcb59c88>

KanbanTool. (2020). *Scrumban, lo mejor de Kanban y Scrum*. Recuperado el 11 de 10 de 2020, de <https://kanbantool.com/es/scrumban-scrum-y-kanban>

León, K. (2020). *Metodologías ágiles como herramientas fundamentales para el desarrollo de emprendimientos*. Universidad nacional abierta y a distancia., Bogotá. Recuperado el 21 de 11 de 2020, de <https://repository.unad.edu.co/handle/10596/33613>

Murias, D. (2018). *¿Cuántas marcas de coches existen actualmente en el mundo?* Recuperado el 07 de 2020, de <https://www.motorpasion.com/revision/cuantas-marcas-coches-existen-actualmente-mundo#:~:text=Todos%20conocemos%20los%20principales%20fabricantes,de%20auto m%20viles%20en%20el%20mundo.>

OBS, Bussiness school. (2020). *La metodología Scrumban. Cuándo y por qué utilizarla*. Recuperado el 22 de 11 de 2020, de <https://obsbusiness.school/es/blog-project-management/temas-actuales-de-project-management/la-metodologia-scrumban-cuando-y-por-que-utilizarla>

Petrova, S. (18 de 01 de 2019). *Adopting Agile: The Latest Reports About The Popular Mindset*. Recuperado el 29 de 11 de 2020, de <https://adevait.com/blog/remote-work/adopting-agile-the-latest-reports-about-the-popular-mindset>

Pichler, R. (12 de 03 de 2014). *A Template for Formulating Great Sprint Goals*. Recuperado el 2021 de 01 de 10, de <https://dzone.com/articles/template-formulating-great>

PMI®. (2017). *Guía práctica ágil*. Pennsylvania: Project Management Institute.

Radigan, D. (2020). *El backlog del producto: la lista de tareas pendientes definitiva*. Recuperado el 11 de 10 de 2020, de Atlassian Agile Coach:

<https://www.atlassian.com/es/agile/scrum/backlogs>

Ramos, J. (2020). *Scrum: ¿Qué es el Product Backlog?* Recuperado el 11 de 10 de 2020, de Programación y más: <https://programacionymas.com/blog/scrum-product-backlog>

Sandoval, J. (15 de 11 de 2016). *Sprint Plannings (Planificación de la Iteración)*. Recuperado el 20 de 02 de 2021, de <https://medium.com/@jlsandovaln/sprint-plannings-planificación-de-la-iteración-ddb78ac60536>

Scrum Manager. (17 de 10 de 2016). *Estimación de póquer*. Recuperado el 23 de 01 de 2021, de https://www.scrummanager.net/bok/index.php?title=Estimación_de_póquer

Touza, R. (2020). *Lean y Agile: Diferencias y similitudes*. Recuperado el 29 de 11 de 2020, de <https://robertotouza.com/agile/lean-y-agile-diferencias-y-similitudes/>

Trigas, M. (2020). *Metodología Scrum*. Recuperado el 21 de 11 de 2020, de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>

Vergara, S. (18 de 07 de 2019). *¿Qué es BDD (Behavior Driven Development)?* Recuperado el 20 de 02 de 2021, de <https://www.itdo.com/blog/que-es-bdd-behavior-driven-development/>

Viewnext. (2020). *SAFe (Scaled Agile Framework)*. Recuperado el 29 de 11 de 2020, de <https://www.viewnext.com/safe-scaled-agile-framework/>

Vila, J. (8 de 07 de 2016). *La Metodología XP: la metodología de desarrollo de software más existosa*. Recuperado el 29 de 11 de 2020, de <https://proagilist.es/blog/agilidad-y-gestion-agil/agile-scrum/la-metodologia-xp/>

Zacaría, D. (01 de 01 de 2019). *20 Product Prioritization Techniques: A Map and Guided Tour*. Recuperado el 14 de 11 de 2020, de <https://foldingburritos.com/product-prioritization-techniques/>

Apéndices

Apéndice A. Evidencia de historias de usuario recolectadas.

Figura 18.

Historias de usuario del gerente.

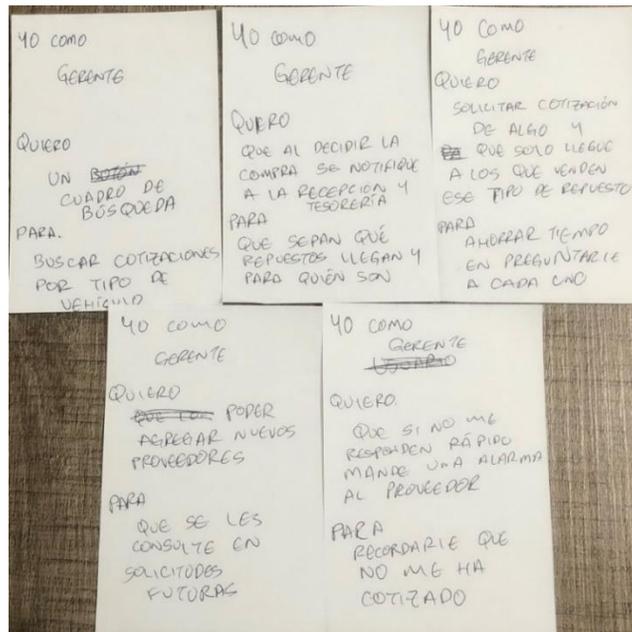


Figura 19.

Historias de usuario de la administradora.

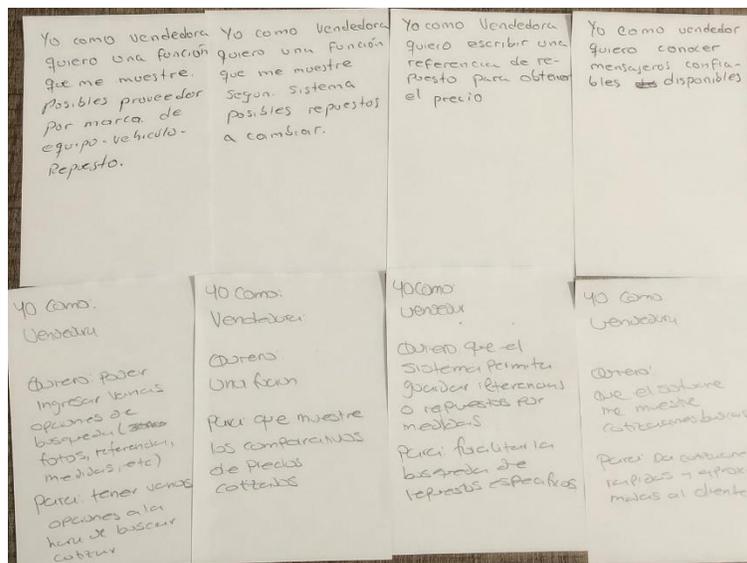


Figura 20.

Historias de usuario del jefe de taller.

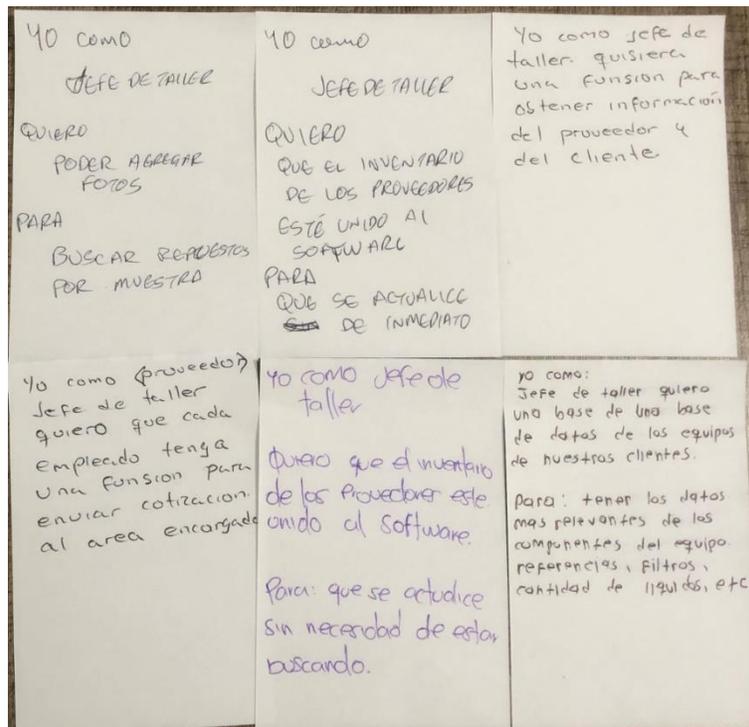


Figura 21.

Historias de usuario de la responsable SGSST.

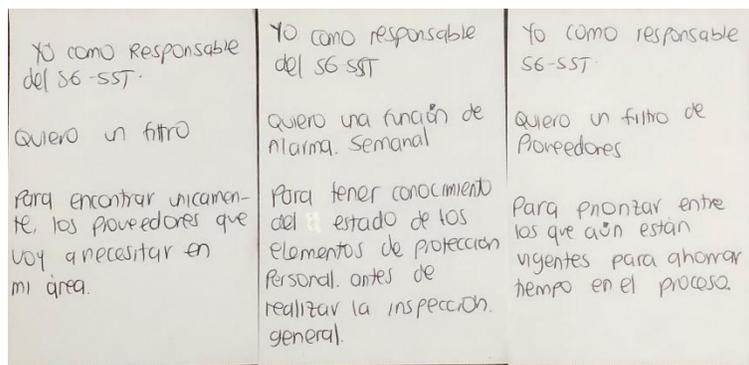


Figura 22.

Historias de usuario de la tesorera.

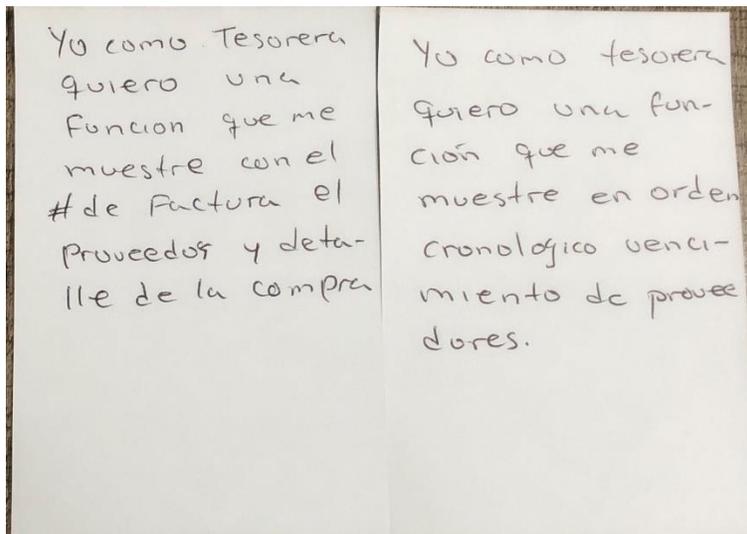
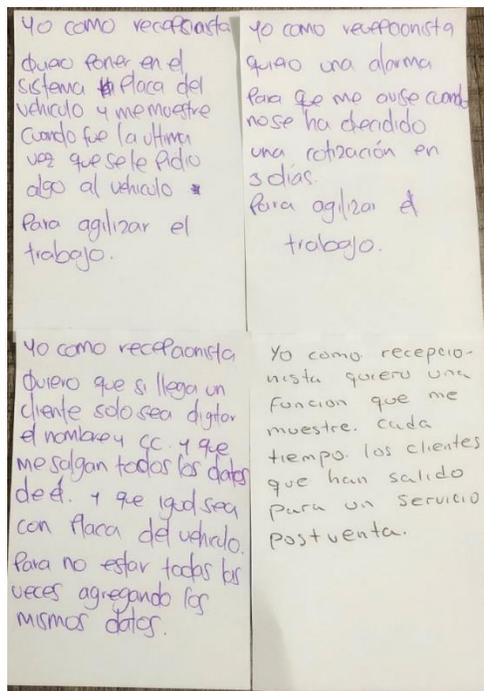


Figura 23.

Historias de usuario de la recepcionista.



Apéndice B. Transcripción completa del juego de roles + persona.

Durante la actividad que se llevó a cabo, se permitió que todas las personas intervinieran según las apreciaciones que tuvieran respecto a la situación planteada. A partir de este hecho, las situaciones, según se transcribieron, presentan puntos suspensivos para indicar la intervención de otro interlocutor.

Introducción: Vamos ahora a hacer algo por 15 minuticos que se llama juego de roles, entonces juego de roles qué es, teníamos el rol del sistema de gestión, teníamos el rol del jefe de taller, de recepcionista, del gerente, del tesorero, del administrador, además podemos hacer el rol de proveedor, es decir, yo soy un proveedor; y ante eso, vamos a generar una situación, luego generamos un problema y buscamos una solución. Entonces, por ejemplo, juego de roles:

Situación 1. Yo soy el administrador, solicité una cotización, usted es el proveedor y a mí no me ha llegado nada. ¿Qué solución habría a eso?, que se genere una confirmación de que el proveedor ya la recibió. ¿Cómo?, ah, fue enviado y fue recibido, como los dos chulitos de WhatsApp, por ejemplo, eso sería bueno, se envió, se recibió y se leyó, podría ser algo interesante para que uno sepa.

Situación 2. Se envió, resulta que no se ha recibido y en realidad no se envió, resulta que se trató de dar clic pero el clic se dio en otro lado y yo estaba en el afán y no me di cuenta que no le di enviar, entonces podría el sistema avisarme, ey, usted montó esta cotización, llevo 10 minutos esperando a que me diga si la va a enviar o no, podría ser, y puede ser personalizado, ósea, ey, no quiero que me avise a los 10 minutos porque sería muy cansón... como un borrador sería... si ósea, estoy montando el borrador y no le he dado enviar, entonces que me avise que no la he enviado, si, oiga, va a cotizar esto o no. También sería bueno que incluyera unas franjas horarias, bueno, después de las 6pm no me vaya a molestar hasta el otro día a las 8 am.

Situación 3. *La inversa, que el proveedor envíe la cotización, él cree que la envió y no se envió, entonces una notificación que le avise al proveedor, oiga, usted no ha respondido tal mensaje, o tiene montado un borrador, pero no la ha enviado... ósea, que le avisen a uno que ya le enviaron la cotización... ósea, que le avisen a uno que ya le respondieron la cotización... ósea, que el borrador con aviso sea tanto para el que pide la cotización como para el que va a cotizar como proveedor.*

Situación 4. *No está el pedido completo, tengo cierta, de los ítems que me pidió cotizar tengo la mitad o la mitad más uno, los otros, o no los tengo, o los tengo homologados o los tengo dentro de tantos días, que dentro del mensaje en la plataforma tenga la opción de que, para cada ítem es particular, de la opción, global, o lo hay, o no lo hay, lo hay homologado, remanufacturado, etc.*

Situación 5. *Que mandan la cotización y si uno no menciona que es de la persona que cotiza a uno no le contestan, entonces, así como le pasa a usted, que no le cotizan, yo cuando me piden cotizar, que la administradora ha salido y eso, yo digo, hola que más habla de parte de esta persona, es que ella me pidió que le cotizara estos repuestos y a mí siempre me contestan, pero entonces yo siempre meto a la administradora en la colada, entonces como que haya un distintivo, porque uno muchas veces no se presenta y ellos no saben que es de Laboratorio Diesel y por eso no le contestan a uno, o no conocen la empresa como Laboratorio Diesel sino conocen es al nombre del administrador que es siempre quien ha cotizado, indiferente de cómo se llame la empresa. Lo importante sería que se reconozca la empresa.*

Situación 6. *Que no ha respondido el proveedor y yo no sé si es que no lo tiene, o no ha visto el mensaje, o no me quiere cotizar o lo está buscando, entonces sería bueno que el proveedor, si no me va a contestar de una, me pudiera retroalimentar, más tarde le mando el precio, lo estoy buscando, no lo tengo disponible lo tengo que cotizar, está para importar y ya se lo busco, para*

que en medio de la espera uno tuviera una retroalimentación de que sí va a participar en la cotización.

Situación 7. Hay proveedores a los que le cotizo, por ejemplo, inyección, y entre esos hay 3 proveedores a los que no les compro, ellos se molestan y a veces no me cotizan, ahí no sé qué hacer yo y es un problema para mí, porque yo necesito la cotización de ellos para poderme guiar para yo meterlos, pero muchas veces los precios de ellos son más altos, no les puedo comprar, y me dejan de cotizar por eso, entonces a veces mi respuesta de pronto es muy brusca y no me vuelven a cotizar. Sería bueno de pronto, retroalimentar al proveedor la razón por la que no fue seleccionado, por ejemplo, se encontraron propuestas más económicas, se encontraron repuestos de mejor calidad... para que ellos no queden ahí como esperando, que no se quede en espera sino que le digan a uno, oiga, usted no salió seleccionado por costoso, ah bueno listo, ya sé que fue por costoso... o porque sí tiene buenos precios pero no tiene disponibilidad... pero eso no nos asegura que siempre nos coticen porque puede darse el caso que ese más costoso, algún día es el único que tenga el repuesto entre todos, ese es el meollo... yo no veo respuesta de esa manera porque yo sé que van a dejar de cotizar... entonces sería bueno que los proveedores pudieran manejar su inventario a través de la misma plataforma, entonces, si el software le dice al proveedor, venga usted como lleva su inventario, no lo lleve así, aquí hay una manera más fácil de que lo lleve, y de paso, le está contando a su público objetivo, qué inventario tiene y a como lo está vendiendo, entonces en ese orden de ideas, el proveedor ya no se tiene que molestar en responderme, porque se retroalimenta de una vez, cuando usted escriba la referencia le va a salir de una vez quién la tiene, con precio y todo... y solamente se contactaría con el proveedor para comprarlo, y no tiene que estar molestándolo coticeme, coticeme, coticeme... entonces, el software debe permitir que el proveedor suba su inventario a la base de datos y que lo suba fácil, porque si es un complique no lo van a hacer... bueno, pero ahí viene otra variable que hay que tener en

cuenta a largo plazo, por lo que, cuando el software se masifique, hay algo que se llamaría ingeniería inversa, por así decirlo en este caso, es que un proveedor adquiriera una cuenta como comprador y quiera fisgonear los inventarios de otros proveedores competencia, si me entiende... entonces tiene que haber unos permisos... exacto, porque listo, ellos pueden retroalimentar un inventario para que el público lo mire, pero es que si el público son otros vendedores, claro se joden... entonces tiene que haber un certificado credencial... no sí claro, a nosotros nos interesaría ver en cuánto venden... bueno, aunque eso pasa hoy en día, alguien nos puede llamar a nosotros y ser la competencia... pero no es lo mismo llamar a averiguar cinco precios a ya ver todo su catálogo completo... debe tener la opción de que solo vean su inventario... eso sería una variable a tener en cuenta en el largo plazo de la aplicación, que ellos mismos den los permisos de qué proveedor puede verlos... ah eso, si el proveedor puede... dar los permisos, no a alguien en particular, sino a Laboratorio Diesel de poder consultar su inventario.

Situación 8. Yo solicité un repuesto en la cotización, entonces me cotizaron por ejemplo kit de embrague, y le di comprar, a recepción le llega una notificación que dice: se decidió comprar un kit de embrague de este proveedor y le llega, de acuerdo con lo que hablamos en las historias de usuario, a tesorería le llega la misma información. Pero resulta que cuando ella va a recibir, le llega algo que no es, se equivocaron, el mensajero entregó una cosa en un lado y otra en otro lado, y a recepción llegó, que se yo, o llegó incompleto... no pues, el mismo software debe tener la opción, como cuando usted hace un pedido en rappi, usted al final le da la opción de calificar, comentarios, etc. Alguna cosa, entonces usted le manda al proveedor la evidencia de lo que llegó mal, o llegó dañado, o no lo entregó o alguna cosa, antes de cerrar el pedido... bueno, pero aquí hablando nosotros, cómo haría la recepcionista para saber qué es lo que le tiene que llegar, para que cuando lo reciba, por ejemplo ella, que tenga una foto... no, es difícil, ahí toca que ella esté pendiente con el mecánico, como a ella le va a llegar con anticipación la notificación, ella no va

a esperar a que llegue el repuesto en la puerta para ir a buscar al mecánico que está en la tienda comiendo empanada, no, ella le dice al mecánico, viene el kit de embrague para este carro que usted está trabajando para que esté pendiente y ella lo vea, que muchas veces pasa eso, viene el repuesto y tiene uno que buscar al mecánico y resulta que le dio por irse a tomar una gaseosa y córrale a la tienda a buscarlo... no pero, si le llega la notificación a la recepcionista, le diga qué es, de dónde viene, consultar con, y ahí ponemos el mecánico que está atendiendo el carro para que ella sepa a quién remitirse para consultar si está bien... pero si eso lo hacen es general, porque no se puede con todos los repuestos, por ejemplo si yo pido una bomba de inyección, si ella es la que lo va a recibir, la bomba trae partes que no se necesitan instalar y que son herramientas... en este caso me gustaría que el proveedor incluyera una foto del repuesto que está vendiendo, y esa foto debería llegar a recepción, a tesorería no... es complicado porque es más trabajo para los proveedores y quién sabe si ellos estén dispuestos a mandar fotos... tendría que ser opcional... la foto tendría que ser del repuesto que esta acá (en la empresa)... ah pero nosotros posiblemente habremos cotizado con muestra... exacto, eso le iba a decir, que eso ya de pronto se pudo haber adelantado en la cotización, que el software le incluya unas casillas al proveedor para que incluya fotos de los repuestos que haya duda o algo, para que se sepa qué es lo que va a llegar... o, lo otro es, le entrego este lápiz y devuélvame este lápiz, le entrego esto otro que llegó adicional, ah pero usted no me lo va a devolver entonces quiere decir que no está como repuesto, si el mecánico me entrega 3 cosas yo le entrego 3, no 4, que es lo que generalmente ocurre para evitar que se pidan repuestos y no se instalen.

Apéndice C. Evidencias de pantallazos recolectados.

Figura 24.

Pantallazos realizados por el jefe de taller.

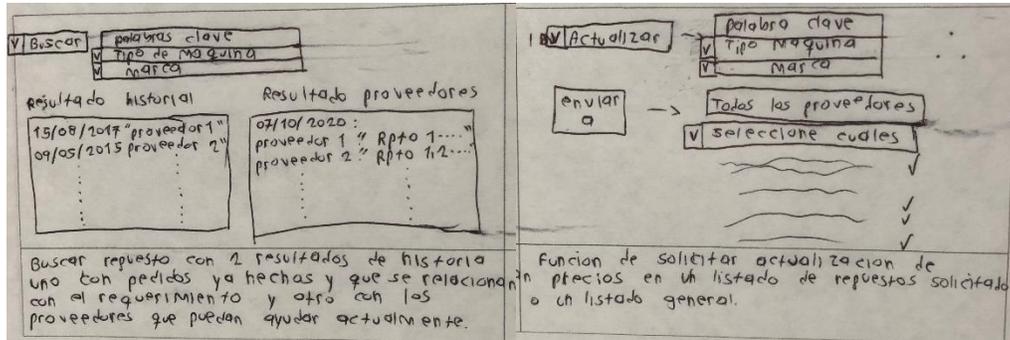


Figura 25.

Pantallazos realizados por la tesorera.

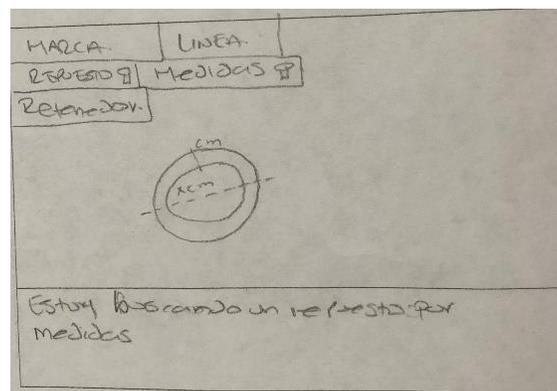
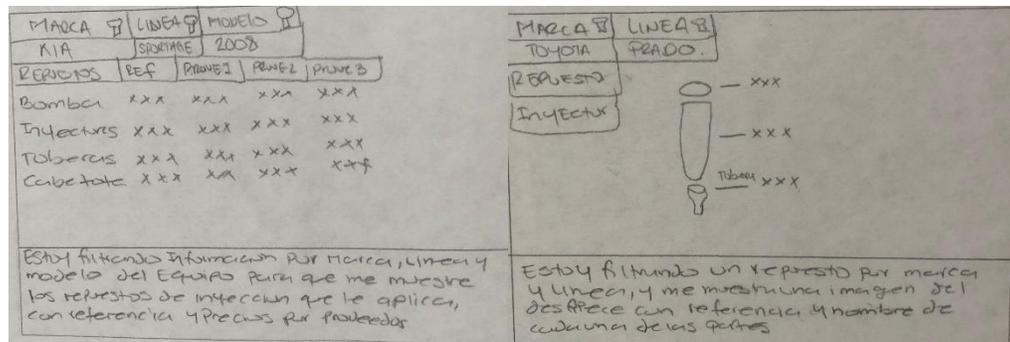


Figura 26.

Pantallazos realizados por la administradora.

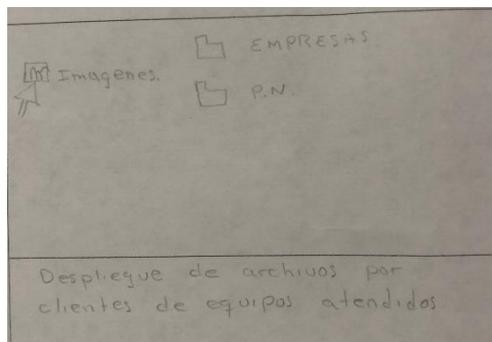
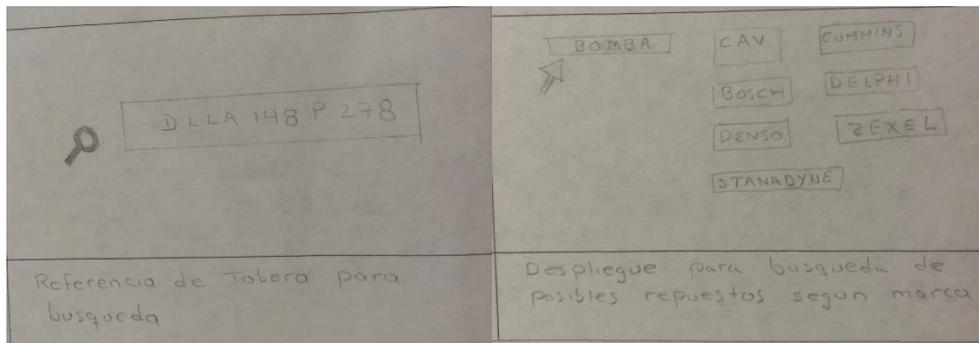


Figura 27.

Pantallazo realizado por la responsable SGSST.

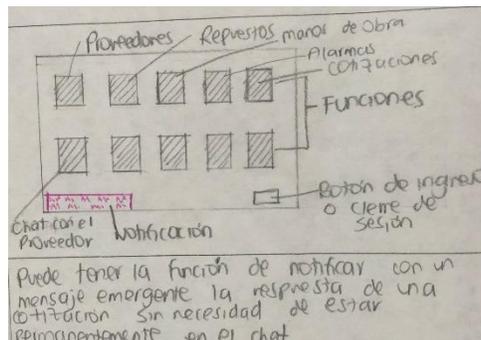
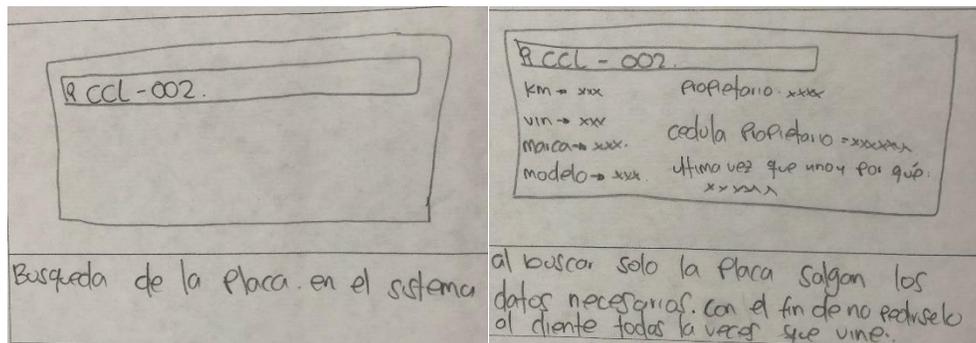


Figura 28.

Pantallazo realizado por la recepcionista.

**Apéndice D. Entrevista a experto.**

La siguiente entrevista se le realizó al ingeniero de sistemas Kevin Arias con el fin de recolectar información relevante que conllevara al desarrollo de un plan de desarrollo de software más acorde a la realidad. Las intervenciones del autor se indicarán con una A, las del entrevistado con una E.

Inicia el autor.

A: Bueno, venga le cuento de qué se trata. Yo tengo una empresa que se llama Laboratorio Diesel del oriente. La empresa repara motores Diesel. También estoy haciendo una maestría en evaluación y gerencia de proyectos de la UIS en la cual propuse hacer la planeación de la gerencia del software. Entonces de eso se trata mi proyecto. ¿Cuál es el software que estamos necesitando en nuestra empresa? Resulta que nosotros reparamos todo tipo de motores que se muevan con ACPM, vehículos, compresores, plantas eléctricas, retroexcavadoras. Y entonces tenemos un problema y es que tenemos demasiados proveedores; porque también nosotros reparamos cajas, transmisiones, frenos, suspensión, pintura, motor, inyección, turbos alimentadores; reparamos generadores, transferencias automáticas, tarjetas de control de motor y reguladoras de voltaje. Todo eso, entonces tenemos demasiados proveedores y normalmente la parte administrativa solo se acuerda de 1 o de un grupo pequeño, de 10 o 12 proveedores con los que más se mueven, pero

el resto se olvidan. Y cuando entra un proveedor nuevo, de pronto con buenos precios o de pronto con una propuesta interesante, se olvida porque se queda la información en una tarjeta y no hay un software para ingresar esa información. Entonces, la propuesta o lo que pretendo hacer es diseñar un software a través de metodologías ágiles; pero primero para el proyecto estoy estableciendo el plan de gerencia. No sé si le quede claro todo lo que le estoy diciendo o tenga alguna duda antes de continuar.

E: Si, a ver, digamos que hablando un poquito más del software y de lo que quieren lograr, es que, de alguna manera, ¿gestionar los proveedores de una mejor forma?

A: Si, la idea del software es integrar los proveedores, por ejemplo, le explico de pronto algunos alcances que estudiamos en las historias de usuario que ya se desarrollaron. Por ejemplo, que yo diga, necesito pastillas para un vehículo modelo tal, pastillas de freno, por ejemplo, y yo le digo envíelo a los proveedores de frenos y enviar la solicitud, y el software ya se encarga de que a todos los proveedores que estén inscritos que vendan pastillas de frenos, 8 o 10, a todos les llega la misma información. Oiga, la empresa X tiene esta necesidad de las pastillas, cóctele. Luego entrarían los proveedores a hacer una respuesta sobre esa cotización, el software recibe eso y nos lo entrega ordenado en una tabla.

E: ¿Ese proceso cómo lo hacen?

A: A mano, coge, necesito por ejemplo las pastillas, entonces se llama al proveedor 1, 2, 3, a algunos les gusta que los llamen, otro responde por WhatsApp, otros por correo, entonces depende, el proveedor tiene sus mañas y cada uno tiene sus cosas. Entonces toca acomodarse, pero toca hacer las llamadas uno por uno. Si no tiene tiempo toca hacer la llamada más adelante, algunos le dicen déjeme la solicitud y yo le averiguo y le cotizo. Entonces se está perdiendo mucho tiempo en la parte administrativa en esa búsqueda y esa consulta de los precios disponibles. Y, por ejemplo, muchas veces no se oferta lo mejor que ofrece el mercado porque el cliente tiene un

tiempo limitado también para hacer la reparación, entonces no puede esperar a que uno en 1 o 2 días le dé la cotización cuando recoja uno todos los precios, no, el necesita los precios es ya.

E: Digamos, ya un poquito, no sé si tenga esta información más específica. De entrada, ¿la notificación a los proveedores se haría por medio de qué?, de un mail, o se les daría un acceso a la plataforma que es la aplicación que desean construir.

A: Bueno si, esa parte todavía no está estudiada, pero sí en las historias de usuario se alcanzó a mencionar algo de eso, que se podría trabajar con un acceso a la plataforma y que les llegue a ellos un mensaje de texto, una notificación, un WhatsApp o un correo que diga: Laboratorio Diesel del oriente tiene una necesidad. Y ya el proveedor entra a la plataforma y la responde. Sin embargo, pues usted sabe que uno depende de cómo ellos interactúen, qué acceso a la tecnología tengan y hay muchas variables que se pueden generar. Entonces ingeniero, yo le voy a hacer unas preguntas, son 14 en realidad. Si de todas maneras durante el proceso salen más dudas con gusto yo se las voy resolviendo de acuerdo con lo que conozco de cómo funciona la empresa. No sé si le mencioné que la gerencia del desarrollo del software se va a plantear desde la perspectiva de las metodologías ágiles. Bueno, entonces, primero pues qué experiencia tiene usted trabajando con el desarrollo de software.

E: Bueno pues yo llevo 4 años trabajando en desarrollo de software. He participado en el desarrollo de algunos sistemas bastante grandes como la cámara de comercio de Barranquilla. Estuve involucrado, realmente el sistema como tal más grande en el que he participado hasta el momento y también en el que he estado que ha tenido como mayor cantidad de procesos que se realizan dentro del software, mayor cantidad de requerimientos. Luego, también he desarrollado otros tipos de software, no de esa envergadura. También algunos proyectos de emprendimiento, entonces sí, he participado en unos 3 o 4 sistemas, así como tal en diseño y metodología.

A: Bueno y en alguno de esos proyectos ha trabajado con metodologías ágiles, como scrum, Kanban, XP.

E: Digamos que oficialmente la gestión de los proyectos desafortunadamente no ha manejado metodologías ágiles pero mis compañeros, por ejemplo, mis amigos de emprendimiento y yo pues también conozco de cerca las metodologías ágiles. Realmente conozco y la única que he usado o he visto que han usado a mi alrededor es scrum.

A: Bueno, ya respecto como al desarrollo del software como tal, dos preguntas en una, qué tamaño debería tener el equipo de trabajo pues teniendo en cuenta el tiempo de desarrollo que pueda conllevar el software. Porque me imagino que va de la mano, si trabajan 2 se demora más y si trabajan 10 se demora menos. O hay un límite porque no, más de 4 personas no pueden trabajar porque si trabaja el uno el otro no. ¿Cómo sería ahí?

E: Claro, bueno, depende de demasiadas cosas. O sea, pueden ser, yo diría y realmente no es algo que pienso yo, sino que también he leído y me han dicho, realmente un equipo de trabajo, digamos que depende mucho. Porque si es contratado directamente de la empresa que va a desarrollar el software, entonces hay algunos roles que no van. Por ejemplo, si en Laboratorio Diesel del oriente, si ustedes contratan directamente, entonces, ustedes digamos que serían la parte gerencial del proyecto. En ese caso usted sería el cliente, pero es la misma persona que contrata los desarrolladores y el equipo de trabajo. Una cosa muy distinta, algo distinta, pero realmente al momento práctico es bastante la diferencia, es que se haga la contratación con una empresa de desarrollo de software. Hay varias diferencias, porque cuando lo hace directamente usted, pues usted asume, realmente, le saldría en teoría más económico, porque se ahorraría unos roles y algunas cosas; pero, pues, por ejemplo, la garantía del software, la obtención de los requerimientos y eso, al final del día puede que pasaran factura más adelante. Para hacer una contratación con una empresa entonces hay, digamos, garantía en muchas instancias.

Bueno, entonces, eso era para llegar un poquito a lo del tamaño del equipo. Mínimo debería haber una persona trabajando en Backend y Frontend. Backend, así como que muy por encima, es la parte del software que se encarga del manejo de los datos, las conexiones con las bases de datos, los procesos de negocio, las reglas de negocio. Qué pasa cuando ingresa esto, para dónde lo enviamos, qué pasa con esta data, la guardamos, la obtenemos; esa parte es el Backend, por eso se llama Back, atrás. Y el Frontend, es lo que está a la vista del usuario. Por ejemplo, en Facebook, los botones, los chats, entonces todo eso es parte del Frontend. Entonces, mínimo sería una persona para Backend y una persona para Frontend. También puede ser que una persona haga los dos, pero normalmente se tiene una especialidad. Una persona en Backend, uno en Frontend. Ah bueno, y una persona que haga la gestión del proyecto, que haga el contacto con el cliente que en este caso usted sería el cliente. Si se contrata una empresa externa, también ellos tendrían una persona allá o hasta dos personas en jerarquía, que harían contacto con el cliente. Entonces, en un equipo pequeño, serían esos. Esas tres personas serían lo ideal mínimo.

Ahora, si, digamos, el software tiene requerimientos específicos de que se necesita una parte de diseño muy específica, una parte de análisis de negocio muy específico, entonces sí, digamos la interfaz gráfica tiene que tener ciertas especificaciones porque pues, no sé, el proyecto los demanda, entonces ahí, si bien es un rol que más o menos está, solo casi que solo al principio del proyecto y luego da unas pinceladas a través del proyecto que es lo ideal, sería un rol necesario, que es un diseñador. Si se tiene demasiada lógica de negocio y es demasiado complejo, también se necesitaría un analista, pero eso ya es un analista de negocio que es una persona que entiende un poco el negocio y digamos que tiene una especialidad en obtener los requerimientos que realmente se necesitan para el software. Entonces, esos dos roles son opcionales en un proyecto pequeño. Pero si el proyecto tiene esos requerimientos ya iríamos en 5 personas mínimo.

A: Pues naturalmente el proyecto pretende que sea contratado, buscar un ingeniero para que haga todo, no, se sale de las manos, no está dentro del portafolio del negocio de la empresa entonces es mejor contratado. En mi perspectiva siempre ha sido mejor así. Entonces tendríamos el dueño del producto que parte de la misma empresa, tendríamos el supervisor o el scrum máster, bueno en este caso el supervisor, y dos más que sería mínimo. Ya si queremos, entonces un diseñador y el analista de negocio. ¿Esas profesiones de esas personas cómo deben ser? O sea, ¿deben ir orientados las dos personas fijas, todos son ingenieros de sistemas o algunos economistas o cómo sería?

E: Si, el Backend y el Frontend que al final del día van a ser los que van a construir como tal el software, ellos son de tiempo completo, desde principio a fin. Hum, se me olvidaba, también es probable que se necesite en un equipo, una persona que haga las pruebas del software. Se llama Quality Assurance.

A: Y perdón, ¿sería distinta?, o sea, si el proyecto se llegara a hacer pruebas con la parte administrativa, ¿es otro cuento aparte de la persona que hace las pruebas dentro de esa empresa que se contrate?, o sea, ¿la empresa está desarrollando el software, ellos necesitan una persona interna para hacer pruebas antes de entregarle el producto al cliente para que lo pruebe él también?

E: Si, claro, y de hecho las empresas tienen departamentos completos de Quality Assurance (aseguramiento de la calidad) porque digamos que es, no sé, lo que pasa es que cuando el desarrollador, los del Frontend y el Backend, ellos tienen que probar, obviamente tienen que cumplir los requisitos mínimos de pruebas, pero si ellos son los que lo prueban a fondo, al final van a estar sesgados porque ellos fueron los que construyeron el software. Entonces, tienen un sesgo que desvíe de pronto a veces probar a fondo ciertas funcionalidades, entonces eso da una mala impresión cuando llega el cliente porque hay cosas que no se probaron desde un punto de vista más externo, entonces, sí, digamos que, de manera latente, el desarrollador sabe que esto

debe fallar y no lo hace, no por mala intención, sino que tiene un sesgo. Entonces, en ese caso sí, digamos que es necesario una persona para las pruebas.

Entonces claro, una empresa tiene destinado un recurso para eso, entonces, tanto Quality Assurance como Backend como Frontend, esos tres roles, son técnicos, netamente técnicos, no tienen que ser ingenieros de sistemas, tienen que ser, normalmente, programadores. Bueno el Backend y el Frontend deben tener muy buenas nociones en programación como tal, y el Quality Assurance también en programación de pronto en otras ramas o tecnologías más específicas, pero también en programación.

Entonces iríamos, el Backend y el Frontend irían de tiempo completo. El Quality Assurance no está de tiempo completo, porque él hace las pruebas más bien un poco al final del desarrollo. Si, él no está todo el tiempo haciendo pruebas de cada una de las cosas pequeñas que se hacen, sino que él prueba por paquetes. Para eso también hay metodologías y ciertos procesos establecidos, pero él prueba ya las cosas que están listas para salir a funcionar en producción. Entonces él estaría más bien al final.

El analista y el diseñador están más bien al principio, y el scrum máster o el gerente del proyecto, a veces, si el proyecto es muy grande, hay scrum máster, hay líder técnico, a veces hay líder técnico encima de ese líder técnico, o sea, son 2 líderes técnicos, un montón de desarrolladores. En este caso, yo creo que en un proyecto como el que usted me describe, podría ser la misma persona el scrum máster y el gerente del proyecto, si se tiene alguien con el rol. Pero como usted dice que podría hacer la contratación con una empresa entonces ellos la harán dentro de su propia gestión, pero es para que se haga una idea de qué es lo que existe allá adentro. Y esa persona, digamos que sí está en todo el proyecto, pero no está de tiempo completo. Los scrum máster y gerentes del proyecto normalmente están en varios proyectos al mismo tiempo. Porque el proyecto no demanda todo su tiempo, su presencia todo el día.

A: Si ellos están en las reuniones, ah bueno, ¿pero ellos si ya tienen que ser ingenieros con alguna especialidad?

E: Ahora, el diseñador tiene que ser diseñador, debe tener experiencia en diseño de interfaz, depende de lo que se quiera. Tiene un rol un poquito más

A: O sea ¿tiene que ser como una mezcla de diseño gráfico con programación?

E: Un poco, si, realmente lo que tiene que saber es usar herramientas que luego les permita a los programadores decir, ah mire, esto es lo que, hay software en los que ayudan en el diseño y luego el programador toma ese diseño y lo hace realidad. Pero pues sí, digamos que el diseñador debe tener unos conocimientos más de diseño y de programación y eso simplemente saber usar los softwares que se necesitan para que pues el programador pueda entender lo del trabajo de él. Pero pues, ya si tiene conocimientos en programación, son útiles, pero no diría yo que son estrictamente necesarios.

El analista técnico, el analista de negocio, normalmente, son también programadores ingenieros de sistemas que tienen mucha experiencia ya en creación de sistemas y de pronto hacen alguna especialización, maestría o algunos cursos. Porque también eso pasa demasiado en esta industria, con ciertos cursos, no tienen que ser certificados por una universidad, uno ya adquiere unas habilidades que le permiten ahondar en otros roles, entonces el analista técnico puede ser un ingeniero con mucha experiencia, que ha realizado sistemas y entonces adquiere una visión más amplia de lo que realmente necesita un sistema que se está construyendo.

Y el scrum máster, esta persona si puede ser un ingeniero industrial. Lo que pasa es que, cual es la cosa, él tiene que gestionar el tiempo del equipo, entonces, si él tiene bases, conocimiento de gestión de equipo, de gerencia, está genial, pero si no tiene conocimientos de programación cómo va a saber si 3 días para realizar esa tarea están bien, o son 2 horas o son 3 semanas. Entonces, por

eso también para ser scrum máster, al final del día los buenos scrum máster tienen experiencia en programación.

A: ¿Algún miembro del equipo debe estar relacionado o conocer el negocio del cliente? ¿O ya para eso está el cliente para hacerle las preguntas?

E: Claro, el analista. El analista es quien, si el negocio es muy complejo, el analista es quien hace esa parte, a veces es un equipo de analistas realmente. Pero si, ese es el rol encargado de esos requerimientos.

A: Ah bueno ingeniero, venga le hago otra preguntica, ¿cuál sería el perfil mínimo del equipo de trabajo?, o sea, ¿debería ser impulsor, creativo, innovador, disciplinado, investigador, perfeccionista? ¿Cómo debe ser el perfil del equipo de trabajo en conjunto para que eso tenga éxito dentro de lo que usted sabe?

E: La verdad es que es, digamos que en general, se necesita una mezcla de proactividad y disposición a trabajar en el equipo, porque estamos hablando de equipos, pero a veces en los equipos hay gente que no encaja, entonces la comunicación es clave. Entonces sí, la comunicación y la proactividad diría yo que van de la mano, porque a veces, las buenas empresas gestionan a sus equipos por metas, entonces, el desarrollador puede quedarse sin tareas y puede quedarse callado, normal, ya cumplí con mis tareas, pero la productividad es importante en el sentido en que le da al equipo un impulso un poquito más grande del que podría tener, dado que pues eso denota un compromiso por parte de los integrantes. Entonces no sabría más allá cómo responderle porque pues al final del día sí, uno necesita profesionalismo, proactividad, buena comunicación para llevar a cabo el trabajo. Investigativo en este caso no, realmente no es necesario.

A: Bueno, otra preguntica ingeniero, la arquitectura del software, ¿qué arquitectura debería seguir el desarrollo de software?

E: Si, normalmente, por lo que he visto, hay arquitecturas como la de microservicios como por ejemplo Netflix. Básicamente qué permite los microservicios, poder escalar en tiempo real ciertas partes del negocio que sean necesarias en ese momento, entonces, si la gente necesita en Netflix, la mayoría de la gente está realizando procesos de streaming, la mayoría de gente está viendo, entonces esa tarea en el sistema es la que necesita más apoyo, más recursos, no tanto por ejemplo la parte de los pagos, que es importante la seguridad por ese lado, pero no se necesita un volumen gigante. Entonces los microservicios permiten escalar las aplicaciones. En este caso no sería necesaria por microservicios.

Yo creo que, lo que más se ha usado es la arquitectura de 3 niveles y sería también lo necesario aquí, porque, como le explicaba ahorita, le expliqué el Frontend y el Backend en dos capas, cierto, hay una tercera capa que es la que se conecta con el Backend que es la base de datos, sí, los datos como tal, entonces el Backend mantiene por dentro toda la lógica y las reglas del negocio, y la base de datos mantiene los datos, si, entonces el Frontend hace lo que se visualiza en la interfaz. Esa sería para mí la arquitectura que se debería seguir en un proyecto como el que estamos hablando.

A: Ah bueno listo, y ahora, bueno, respecto de las fases que se deben llevar a cabo, ¿cómo cree usted, o cuántas fases considera que conlleve el desarrollo del proyecto? ¿Qué fases?

E: Si, bueno, realmente, creo que lo que pasa con las metodologías ágiles es que varias de esas fases lo que se hace es que se iteran, entonces, pero sí me parece que hay 2 o 3 grandes bloques, desde mi experiencia, y esto realmente en esto no tengo tanta experiencia, le hablo un poco desde la teoría y lo que me acuerdo.

Como le comentaba, yo considero que hay 3 fases importantes, la primera es análisis y diseño. Una primera fase de análisis y diseño exclusivamente de eso es realmente importante porque es lo que hace que el proyecto que está destinado a valer N cantidad de dinero, no pues,

digamos un número, que algo que está destinado para valer 100 millones de pesos no salga costando 300 millones de pesos. Esa parte es lo que uno ve realmente vital que se haga y es, por lo que he visto realmente, esa es la parte delicada, porque hay ocasiones en las que se hace con toda la gana del mundo, pero se busca la asesoría incorrecta o se tiene a la gente incorrecta para hacer ese estudio o esa fase y sigue pasando lo mismo. Se gasta el dinero en la parte de análisis y diseño y sigue quedando mal y habiendo sobrecostos excesivos al final del proyecto, porque queda mal hecho y toca reponer en, digamos, mientras se va construyendo, eso da lugar a un caos allá adentro en el desarrollo luego cuando se está haciendo.

Entonces esa es la primera fase, la segunda es como tal el desarrollo del sistema, software, proyecto o lo que sea. En esta fase si estamos usando una metodología ágil, qué pasa, es una iteración constante sí, es una iteración, construcción de desarrollo del software, hacer unas pruebas a ese nivel, al nivel de lo que se construyó, hacer presentaciones, charlas constantes con el cliente, y si hay cosas que quedaron mal definidas, porque todo no se puede definir en la fase de análisis y diseño, es imposible, entonces si hay cosas que no van cuadrando pues se van digamos que retroalimentando durante el desarrollo del software, en esa fase grande que es el desarrollo como tal pero que es una constante iteración de desarrollo, pruebas, presentación corrección. Y eso es lo que permiten las metodologías ágiles, ir como que teniendo una visión de que se está realizando algo, pero al mismo tiempo, digamos que irlo haciendo bien. Se va haciendo algo, siempre se va viendo el resultado, pero al ritmo de que se van haciendo bien las cosas.

Y al final, pues una fase de pruebas y despliegue, que es en las que ya se prueba, digamos que, la totalidad del sistema muy desde el cliente. Sigue siendo el equipo de pruebas del desarrollo, pero se hacen pruebas más exhaustivas, más transversales en toda la aplicación. Si es una aplicación altamente escalable se hacen pruebas de carga, de seguridad, de integración de los componentes; cosas, digamos que, todo lo que se pueda probar se prueba en esa fase y, algunas

personas ponen una fase aparte, lo digo aquí en la misma, es que es cómo ya la puesta en producción, eso es otra fase del desarrollo. Una cosa es lo que está funcionando en los computadores de los desarrolladores o incluso en los servidores de la empresa desarrolladora de software y otra cosa es instalarlo para el cliente. Puede ser en la nube, en los servidores del cliente, puede ser donde sea, pero al final del día es una fase que se tiene que realizar y se tiene que tomar muy en serio porque no se puede dar por descontado. Porque no es como chasquear los dedos y decir listo, pónganse en producción. No realmente.

A: Y ahí mencionó algo que tiene que ver con la próxima pregunta ingeniero, la infraestructura adecuada para el proyecto cuál cree usted que sería, usted me mencionó la nube, los computadores, los terminales. ¿Cómo sería esa infraestructura?

E: Bueno, esto si no es como la arquitectura que, digamos que es una noción muy general. La infraestructura realmente depende demasiado de lo que se vaya a hacer. La infraestructura realmente depende demasiado de lo que se vaya a hacer del caso específico, depende de del cliente del software, depende, entonces digamos que las especificaciones del servidor como tal pues depende mucho de lo que se vaya del servidor o los servidores depender mucho de lo que se vaya a hacer, por ejemplo

Volvamos al ejemplo en Netflix, digamos que Netflix no fuera absurdamente gigante si no fuera una empresa más pequeña con algunas series de algunas películas, igual debería tener una base de datos gigante para guardar todo el volumen de información que tiene.

Por ejemplo, digamos que, en Wikipedia, debe tener unos servidores gigantes para poder guardar, pues claro es texto, pero es todo el texto del mundo, entonces los servidores deben tener un almacenamiento muy grande muy escalable

Entonces digamos que depende mucho de los requerimientos que se tengan, en el proyecto, en este caso por lo que puedo ver en el alcance y eso no es definitivo, porque cuando ya se entra

hacer las cosas más a fondo en un proyecto y pues uno se va dar cuenta dando cuenta cuáles son las necesidades reales del proyecto pero, yo no veo que sea demasiada exigencia en la cuestión de la infraestructura de los servidores, entonces yo creo que con una instancia en Amazon o en la nube o en cualquiera de las nubes, digo Amazon porque es la más conocida, con una distancia de tamaño mediano nada muy grande, yo creo que con eso bastaría para soportarla

La nube es un servidor físico, sólo que uno los puede manejar de manera remota, entonces por eso es por lo que ella es una nube, porque se puede manejar desde fuera, pero al final del día si es un servidor físico entonces... perdí la idea...

Entonces, es un servidor, y ahí se instala lo que se necesite del software, las partes del software, entonces si es una página web pues hay que entrar a comprar el dominio, comprar esa parte, y desplegar en una parte del servidor ciertas cosas que le permiten a usted desplegar o instalar la aplicación del Frontend, si la página web, por ejemplo.

Y el Backend, asimismo, que hace una aplicación aparte, También se instala en el servidor, la base de datos también se tiene que registrar en el servidor, y así, hay otros, digamos que otras arquitecturas que demandan diferentes maneras de instalación o diferentes fases entrar de infraestructura por ejemplo un cliente servidor, sí, esos programas que no tenía que instalar en el escritorio como los de contabilidad por ejemplo, ellos tienen un servidor por allá y ellos la parte del Backend y de los datos Y eso lo instalan por allá en otro lado por allá en la nube, y a las personas que lo va usar les toca instalar por así llamarlo el Frontend, les toca instalarlo en su propio computador, es un poco la idea de lo que se quiere pero yo diría que en este caso todo se podría instalar en un servidor

A: Bueno, eso también está conectado con la siguiente porque, bueno, una de las partes que salió también de las historias de usuario fue que de pronto el proveedor pudiera cargar su inventario a esa plataforma para que no tuviera que estar respondiendo a cada rato, sino que a medida que yo

fuera lanzando una pregunta o una necesidad, de una vez se retroalimentara del inventario que tiene el software con sus precios y sus cantidades y todo, y la disponibilidad. Entonces, en términos de seguridad, qué seguridad se requeriría para el software o si es la misma que ofrecería la plataforma que estamos hablando.

E: Ya que menciona lo de seguridad, y volviendo un poquito a la parte del equipo, y dependiendo de la seguridad que se necesite puede que también sea necesario un experto en seguridad. Y ese rol específico si tiene un perfil más orientado a eso, más orientado a la seguridad informática, eso ya casi que es otra rama, no iría totalmente distinta, pero si es una especialidad bastante separada del resto.

Y ahorita le hablé de parte de despliegue, de producción, de servidores, también hay un rol que hace eso, pero si no es algo demasiado complejo lo pueden hacer los mismos desarrolladores, y si no saben con una pequeña investigación de pronto no tan pequeña, pero con una investigación podrán realizarlo también. Pero para que se haga la idea un proyecto muy grande tiene todos esos roles y a veces se hacen equipos completos en cada uno de esos roles, porque es vital para la aplicación o para el software.

Entonces pues, de seguridad, si se tiene que cargar información sensible, si bueno, por ejemplo, si es en Amazon, la cuestión de las puertas de la base de datos y esas cosas sólo están expuestas dentro del servidor, no están expuestas a internet, dejarla expuestas a internet es un error mayúsculo. Eventualmente, algún robot de algún hacker va a llegar y sacar la información, puede que no pase nada, pero si se puede evitar para que se hace.

Yo diría que, con hacer ese tipo de cosas, por ejemplo, encriptar las contraseñas, son cosas básicas que se hacen, encriptar las solicitudes, utilizar token de autorización, esas cositas son básicas, pero en este caso yo diría que con eso cumple y no sería necesario como un rol así específico para eso.

Pero claro si se va a hacer una contratación con una empresa téngalo por seguro que se cuiden mucho y tienen su gente, entonces si necesitan algo de seguridad llaman a la persona específica y lo tienen una semanas en el proyecto para que se cuadre todo asesore el equipo, puede ser al principio o al final, o durante el proceso si se necesita un par de veces también le llaman, entonces por eso le digo que eso es lo bueno de hacerlo con una empresa, esa es la garantía, y al final del día, claro, uno podría contratar un ingeniero, pero puede que, claro está depende mucho del presupuesto, el alcance del proyecto, del tamaño, De los requerimientos de seguridad, de la infraestructura, el diseño, el negocio, pero siempre hacerlo con una empresa especializada en eso tiene sus ventajas a la larga

A: Bueno, y con respecto a las bases de datos, ¿también se manejarían en ese caso que hablamos de la infraestructura de la nube, también se manejarían como SQL, son bases de datos normales o tienen alguna complejidad?

E: Bueno, las bases de datos creo que hay más tipos ahorita, pero en general, son SQL y no SQL, entonces bases de datos relacionales y no relacionales

En este caso, las no relacionales se usan para guardar data no estructurada, digamos que cuando usted no sabe exactamente, por ejemplo, las empresas que usan inteligencia artificial, o que usan análisis de datos a gran escala y esas cosas, normalmente esa información no está bien estructurada. Bueno si estuviera bien estructurada, no se necesitaría hacer un análisis de datos, no se necesitaría hacer limpieza de datos, entonces cuando sean ese tipo de cosas, se tienden a usar, ayudar de bases de datos no relacionales, por qué al final del día casi toda empresa y casi todo el software tiene una parte de autenticación, de pagos, que son sensibles y esa información se guarda en las bases de datos relacionales, en este caso yo no veo demasiado información no estructurada porque todos son como datos de proveedores, inventarios, cosas así, entonces de pronto yo no lo

veo necesario así super por encima, pero todo depende de la fase de análisis y de lo que se establezca, para ver cuáles son las necesidades en el proyecto, pero en el momento yo diría que bases de datos relacionales. Ya una específica puede ser PostgreSQL o la que se defina.

A: El uso de servidores de la nube que venimos hablando, ¿cómo se llevaría a cabo?

E: Bueno realmente hay dos maneras, si el cliente tiene sus propios servidores literalmente es tener una torre en alguna localización del cliente, si una torre que es un servidor configurado no es muy diferente a tener un computador de escritorio sólo que para los servidores no es necesario una pantalla, no siempre es necesario realmente. Eso es una manera, pero qué pasa, que el servidor se puede quemar, se puede dañar, entonces por eso la nube están útil, porque simplemente paga una suscripción de 20 o 30 dólares y ya tiene toda la seguridad del mundo. Ah bueno y es que si usted lo instala, tiene que instalar toda la seguridad, entonces todas las configuraciones, bueno estos servidores que uno puede tener en Amazon por ejemplo o en Microsoft o en cualquiera de los servicios que tengan nube, tienen ya pre configurados, tienen esas ventajas y pues también que no se va a quemar, bueno, de pronto se puede quemar, pero ellos tienen lo que hacen es respaldo, si se cae una pues ellos tienen la otra, casi que eso es transparente para usted, entonces tiene muchas ventajas por una suscripción mensual que realmente vale la pena pagar. Cómo se puede llevar a cabo, pues comprando una suscripción en un servidor de algún servicio de la nube, sería mi recomendación.

A: Una vez entregado el software, ¿cómo se podría realizar esa administración, la administración del software ya cuando el cliente la obtenga?

E: Bueno, administración como tal, bueno, es que la administración del software depende de qué estamos hablando acá, porque si es un software de servicio como Netflix o Spotify o YouTube, Los servicios, si es algo así, la administración se hace desde la propia empresa que lo desarrolló. Por qué al final del día uno entrega un software, uno lo que entregues una licencia o una

suscripción, no entrega un código ni nada de eso, es un servicio lo que uno está dando, entonces recibí un pago por un servicio y ya pues ese tipo de cosas es tener servicio al cliente constante, por qué hay cosas que pueden fallar en un servicio, por eso esas empresas se preocupan tanto por el servicio al cliente. Si es un software que hace uno para una empresa y va y se lo entrega, entonces pues nada, la administración no tiene que ser el cliente, porque yo le estoy entregando un software que él necesita, entonces la administración le hace el cliente y como se lleva acabo esto, pues se hace una fase antes de la entrega, todo eso se presupuesta, entonces cuando se está haciendo la entrega se hace capacitación del personal si es necesario. Se entregan manuales de uso, se entrega toda la documentación necesaria para que el cliente pueda administrarlo.

Entonces qué pasa, aquí entra la fase de garantía, Entonces cada empresa define con el cliente puede ser un tiempo específico de garantía, o puede ser por funcionalidades, depende mucho del software y de las dos empresas tanto de la empresa como en la que lo desarrolla como del cliente, depende mucho de que sea lo que pacten. Pero normalmente se pacta alguna garantía de algún tipo, porque todo el software está susceptible a fallos y sobre todo en los primeros meses de uso.

A: ¿Hace parte de la administración que si se llega a llenar el espacio que se compró en la nube?

E: Claro, eso se contempla en el desarrollo si va a crecer demasiado, pues lo que hay que hacer es, de nuevo, estos servicios en la nube permiten tener digamos que un almacenamiento elástico, entonces a medida que va necesitando se va pagando un poquito más por ese espacio. Claro, eso se contempla en el desarrollo si va a crecer demasiado, pues lo que hay que hacer es, de nuevo, estos servicios en la nube permiten tener digamos que un almacenamiento elástico, entonces a medida que va necesitando se va pagando un poquito más por ese espacio. Se puede, por ejemplo, si es data, si después de un mes no se va a usar entonces se va borrando, y así nunca se va a llenar. Entonces depende de nuevo, todo depende, siempre depende, pero al final del día si

hay algo de cierto para todos esos dependes, y es que se tiene que encontrar la manera de que la administración se haga casi automática.

A: ¿Qué tecnología o lenguaje de programación se debe considerar en el desarrollo?

E: Bueno, tecnologías, si es una página web, hay toda esta parte del Frontend que le hablo, tiene un ecosistema alrededor de JavaScript, Qué es un lenguaje de programación, existen algo que llama Frameworks, que la traducción literal son marcos de trabajo, ellos le permiten a usted construir el Frontend que no tiene que hacerlo desde cero, básicamente le dan muchísimas herramientas y esto es algo vital en el desarrollo del software cuando la mayoría del tiempo uno no hace cosas desde cero sino que usa o adapta cosas que ya están hechas. Y eso es algo muy chévere y es lo que permite que se avance tan rápido. Entonces para el Frontend si estamos hablando específicamente, esos Frameworks hay uno que se llama Angular, hay otro que se llama React, qué son los dos más grandes y lo más populares, esas tecnologías se pueden usar, y el lenguaje de programación ahí sería JavaScript. También se hace uso de HTML que no es un lenguaje de programación sino un lenguaje etiquetado y CSS que tampoco es un lenguaje de programación sino es como unas hojas de estilo, qué es lo que ayuda a que se vean bonitas las páginas web y eso.

Nunca hablamos de aplicaciones móviles, pero si fueran aplicaciones móviles también existen sus propias tecnologías y sus propios lenguajes para desarrollar. En el Backend, qué también JavaScript tiene sus propios, depende de cómo se quiera hacer el Backend, se puede hacer con Java, se puede hacer con Python, Ésos son lenguajes de programación, y cada uno de ellos también tienen sus marcos de trabajo que permiten hacerlos de nuevo, que no sea el de cero, básicamente realmente que permiten hacerlo, si usted va a hacerlo desde cero no creo que haya mucha gente en el mundo que sean capaces de hacerlo desde cero, tal vez la parte de Google y

Microsoft. Y obviamente la base de datos, entonces se necesita SQL que no es un lenguaje de programación sino un lenguaje de Case entonces también sería eso.

Más allá de eso, cómo le digo, realmente depende de lo que quiera el cliente, De lo que quiera el equipo, de las especificaciones, porque esos lenguajes también tienen ciertas características que son especiales para esto o lo otro, entonces digamos que eso es un Overview de lo que se podría usar.

A: Y ya para terminar, el tráfico o el uso esperado del software ¿cómo se podría dimensionar?, pues, teniendo en cuenta que podría ser algo que lleguen a acceder muchos proveedores al mismo tiempo, ¿eso sería también a través de los servicios que ofrezca la nube?

E: Sí, realmente como le decía, uno puede, No sólo los micro servicios se pueden escalar, realmente todas las aplicaciones se pueden escalar si están concebidas para eso, entonces eso también se tiene que tener en cuenta En el desarrollo del software, básicamente, uno puede así mismo como le dije que el servicio en la nube lo permite, un servicio de almacenamiento elástico, también se pueden usar, o se puede tener un monitoreo del tráfico para que también este sea elástico, dependiendo si necesita más o menos tráfico, se puede tener un, digamos que una o dos instancias de la aplicación. Una instancia de la aplicación es básicamente, es duplicarla para poder tener El doble o un poquito menos, dependiendo de la aplicación, el doble de capacidad para responder porque, qué pasa, al final del día un cliente lo que hace es que cuando uno hace una acción en WhatsApp, cuando uno envía un mensaje, lo que hace el mensaje decir al servidor y decir oiga, necesito ir a ese otro lado, A esa otra persona, el servidor dice ah entonces espéreme que estoy enviando este otro, entonces hay demasiados y si sólo hay un servidor se va a saturar, entonces un mensaje en WhatsApp si hubiera un solo servidor se demoraría tres años en llegar a otro lado, porque se envían miles de millones de mensajes, entonces por eso es necesario escalarlo,

Ya no es uno solo sino que el trabajo se distribuye y esos son arquitecturas distribuidas y bueno ahí ya se empieza, se habla de otras cosas, pero si esa es una manera digamos que hacer eso en el software para que sea escalable.