



Universidad
Inca Garcilaso de la Vega

Nuevos Tiempos. Nuevas Ideas

Facultad de Ingeniería de Sistemas, Cómputo y Telecomunicaciones

Aplicación web para la gestión de requerimientos de software en la empresa FabSF Services

Tesis para optar el Título de Ingeniero de Sistemas y Cómputo

Erick Franz Nolasco Ríos

Asesor

Msc. Héctor Henríquez Taboada

Lima – Perú
2018



DEDICATORIA

Este trabajo está dedicado a Dios que me dio la fuerza para seguir alcanzando mis objetivos, a mis padres que me apoyaron en todo el proceso de estudio. A mi novia por brindarme el apoyo en todo el ciclo de estudios y que me motiva a seguir adelante, y a mis amigos por las enseñanzas que aplico en la presente tesis.

ÍNDICE

ÍNDICE DE FIGURAS	5
ÍNDICE DE TABLAS	6
RESUMEN	7
ABSTRACT	8
INTRODUCCIÓN	9
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA	11
1.1 Situación Problemática	11
1.2 Problema de la investigación	13
- Problema General	13
- Problemas Específicos	13
1.3 Objetivos	13
- Objetivo General	13
- Objetivos Específicos	13
1.4 Justificación	14
1.5 Alcance	14
CAPÍTULO II: MARCO TEÓRICO	16
2.1. Antecedentes de la investigación	16
2.2. Bases teóricas	19
2.2.1. Aplicación web.....	19
2.2.2. Arquitecturas web.....	20
2.2.3. Arquitectura cliente/servidor.....	22
2.2.4. Modelos de desarrollo de software.....	23
2.2.5. Requerimientos de software.....	24
2.2.6. Gestión de requerimientos	24
2.2.7. Indicadores para la calidad de software	28
2.3. Glosario de términos.....	36
CAPÍTULO III: VARIABLES E HIPÓTESIS	37
3.1. Variables e Indicadores	37
3.2. Hipótesis	37
- Hipótesis General	37
- Hipótesis Específicas	37
CAPÍTULO IV: METODOLOGÍA PARA EL DESARROLLO DE LA INVESTIGACIÓN	38
4.1. Descripción de la metodología	38
4.1.1. Definición de Scrum.....	38
4.1.2. Fases de Scrum.....	39

4.1.3. Roles de Scrum.....	40
4.1.4. Artefactos de Scrum	44
4.1.5. Sprint (Iteración.....	46
4.1.6. Ventajas	49
4.2. Adaptación de la metodología.....	50
4.2.1. Fase I: Pre-juego.....	51
4.2.2. Fase II: Juego	51
4.2.3. Fase III: Post-juego.....	51
CAPÍTULO V: SOLUCIÓN TECNOLÓGICA	52
5.1. Fase I: Pre-juego.....	52
5.1.1. Planeamiento	52
5.1.2. Arquitectura	57
5.2. Fase II: Juego.....	59
5.2.1. Sprint 1.....	59
5.2.2. Sprint 2.....	67
5.2.3. Sprint 3.....	84
5.2.4. Sprint 4.....	89
5.2.5. Sprint 5.....	95
5.3. Fase III: Post-juego.....	99
CAPÍTULO VI: RESULTADOS	100
CONCLUSIONES	102
RECOMENDACIONES.....	103
REFERENCIAS BIBLIOGRÁFICAS.....	104
ANEXO	105



ÍNDICE DE FIGURAS

<i>Figura 1: Situación problemática.....</i>	<i>12</i>
<i>Figura 2: Aplicación web estática.....</i>	<i>19</i>
<i>Figura 3: Aplicación web dinámica</i>	<i>20</i>
<i>Figura 4: Arquitectura cliente/servidor.....</i>	<i>23</i>
<i>Figura 5: Modelo de desarrollo MVC</i>	<i>23</i>
<i>Figura 6: Evolución de los requerimientos.....</i>	<i>26</i>
<i>Figura 7: Gestión de cambios en los requerimientos.....</i>	<i>26</i>
<i>Figura 8: Características de la calidad interna y externa.....</i>	<i>28</i>
<i>Figura 9: Criterios de la funcionalidad.....</i>	<i>29</i>
<i>Figura 10: Criterios de la confiabilidad.....</i>	<i>30</i>
<i>Figura 11: Criterios de la usabilidad.....</i>	<i>31</i>
<i>Figura 12: Criterios de la eficiencia</i>	<i>32</i>
<i>Figura 13: Criterios de la mantenibilidad.....</i>	<i>33</i>
<i>Figura 14: Criterios de la portabilidad.....</i>	<i>34</i>
<i>Figura 15: Características de la calidad en uso.....</i>	<i>35</i>
<i>Figura 16: Características de la calidad ISO/IEC 9126.....</i>	<i>35</i>
<i>Figura 17: Ciclo de vida Scrum</i>	<i>38</i>
<i>Figura 18: Roles de Scrum.....</i>	<i>41</i>
<i>Figura 19: Product Owner.....</i>	<i>42</i>
<i>Figura 20: Equipo de desarrollo o Scrum team.....</i>	<i>43</i>
<i>Figura 21: Scrum Master.....</i>	<i>44</i>
<i>Figura 22: Product Backlog.....</i>	<i>45</i>
<i>Figura 23: Incremento funcional.....</i>	<i>46</i>
<i>Figura 24: Scrum diario o daily Scrum.....</i>	<i>48</i>
<i>Figura 25: Revisión del sprint.....</i>	<i>49</i>
<i>Figura 26: Fases de metodología Scrum.....</i>	<i>50</i>
<i>Figura 27: Arquitectura tecnológica de la solución.....</i>	<i>57</i>
<i>Figura 28: Arquitectura funcional de la solución tecnológica</i>	<i>58</i>
<i>Figura 29: Aplicación web publicada en servidor EsHost</i>	<i>99</i>

ÍNDICE DE TABLAS

<i>Tabla 1: Framework por lenguaje de programación</i>	24
<i>Tabla 2: Costos de personal</i>	52
<i>Tabla 3: Costos de suministros</i>	53
<i>Tabla 4: Costos de capacitación de usuarios</i>	53
<i>Tabla 5: Resumen de costos</i>	53
<i>Tabla 6: Roles para el desarrollo del proyecto</i>	54
<i>Tabla 7: Pila de producto</i>	55
<i>Tabla 8: Planificación de sprints</i>	56
<i>Tabla 9: Historia de usuario “Login”</i>	59
<i>Tabla 10: Historia de usuario “Registrar Usuarios”</i>	61
<i>Tabla 11: Historia de usuario “Modificar Usuarios”</i>	63
<i>Tabla 12: Historia de usuario “Eliminar Usuarios”</i>	64
<i>Tabla 13: Historia de usuario “Registrar Personal”</i>	66
<i>Tabla 14: Historia de usuario “Modificar Personal”</i>	68
<i>Tabla 15: Historia de usuario “Eliminar Personal”</i>	69
<i>Tabla 16: Historia de usuario “Registrar lenguajes de programación”</i>	71
<i>Tabla 17: Historia de usuario “Modificar lenguajes de programación”</i>	72
<i>Tabla 18: Historia de usuario “Eliminar lenguajes de programación”</i>	73
<i>Tabla 19: Historia de usuario “Registrar prioridad de requerimientos”</i>	74
<i>Tabla 20: Historia de usuario “Modificar prioridad de requerimientos”</i>	76
<i>Tabla 21: Historia de usuario “Eliminar prioridad de requerimientos”</i>	77
<i>Tabla 22: Historia de usuario “Registrar clientes”</i>	80
<i>Tabla 23: Historia de usuario “Modificar clientes”</i>	81
<i>Tabla 24: Historia de usuario “Eliminar clientes”</i>	82
<i>Tabla 25: Historia de usuario “Registrar requerimientos”</i>	84
<i>Tabla 26: Historia de usuario “Modificar requerimientos”</i>	86
<i>Tabla 27: Historia de usuario “Eliminar requerimientos”</i>	87
<i>Tabla 28: Historia de usuario “Evaluar requerimientos”</i>	90
<i>Tabla 29: Historia de usuario “Aprobar requerimientos”</i>	92
<i>Tabla 30: Historia de usuario “Rechazar requerimientos”</i>	94
<i>Tabla 31: Historia de usuario “Consulta de requerimientos”</i>	97
<i>Tabla 32: Historia de usuario “Generar reporte general de requerimientos”</i>	98
<i>Tabla 33: Niveles de respuesta – Escala de Likert</i>	100

RESUMEN

El servicio de fábrica de software brindado por FabSF Services llevaba a cabo sus actividades de aprobación o rechazo de requerimientos mediante un proceso de análisis para evaluar las propuestas de requerimientos y después de realizar la evaluación se envía una respuesta ya sea que las solicitudes son aprobadas o rechazadas de manera verbal, no teniendo una manera de visualizar estas respuestas en cualquier momento que se solicite y por lo tanto no es posible saber cuáles son los requerimientos que han sido aprobados, así como rechazados a lo largo del tiempo. Este proceso no es eficiente ya que muchas veces la información se pierde si no se plasma en algún tipo de documentación.

Así también la información de los requerimientos que son registrados en hojas de cálculo y al intentar completar los mismos se requiere consultar otras hojas de cálculo donde se consigue la información requerida porque pueden guardar relación y tener información detallada que aplique para su registro, ocasionando de esta manera la demora en tiempo en el proceso de registro.

El propósito de este trabajo de investigación fue resolver la problemática presentada anteriormente con el desarrollo de una aplicación web para la gestión de requerimientos para así optimizar la gestión, consulta y aprobación o rechazo de los requerimientos. Para desarrollar esta solución se utilizó la metodología SCRUM que está constituida por tres fases. Esta metodología fue la seleccionada debido a que tiene como objetivo brindar un producto exitoso en un ambiente donde las especificaciones son cambiantes.

Los resultados obtenidos gracias a las encuestas elaboradas en este trabajo de investigación, dan a conocer que la aplicación propuesta cumple los indicadores de calidad de funcionalidad, fiabilidad, usabilidad y eficiencia. Finalmente, se concluye que la aplicación web para la fábrica de software de FabSF Services influyó de manera satisfactoria en la optimización de las actividades de aprobación o rechazo, gestión y consulta de requerimientos, alcanzando los indicadores de calidad propuestos para dichas actividades del proceso de gestión de requerimientos.

Palabras clave: aprobación de requerimientos, rechazo de requerimientos, gestión de requerimientos, SCRUM, funcionalidad, usabilidad, eficiencia, fiabilidad.

ABSTRACT

The software factory service provided by FabSF Services carried out its activities of approval or rejection of requirements through a process of analysis to evaluate the proposals of requirements and after carrying out the evaluation a response is sent, whether the requests are approved or rejected verbally, not having a way to visualize these responses at any time that is requested and therefore it is not possible to know what are the requirements that have been approved, as well as rejected over time. These activities are not efficient since many times the information is lost if it is not reflected in some type of documentation.

Also the information of the requirements that are recorded in spreadsheets and when trying to complete them, it is necessary to consult other spreadsheets where the required information is obtained because they can be related and have detailed information that applies to their registration, causing this way the delay in time in the registration process.

The purpose of this research work was to solve the problems presented previously with the development of a web application for the management of requirements to optimize the management, consultation and approval or rejection of the requirements. To develop this solution, the SCRUM methodology was used, which consists of three phases. This methodology was selected because it aims to provide a successful product in an environment where the specifications are changing.

The results obtained thanks to the surveys prepared in this research work, reveal that the proposed application meets the indicators of quality of functionality, reliability, usability and efficiency.

Finally, it is concluded that the web application for the software factory of FabSF Services successfully influenced the optimization of the approval or rejection activities, management and consultation of requirements, reaching the proposed quality indicators for said activities of the management process of requirements.

Keywords: requirements approval, rejection of requirements, requirements management, SCRUM, functionality, usability, efficiency, reliability.

INTRODUCCIÓN

La fábrica de software, es un servicio brindado por FabSF Services que es una empresa cuyo giro principal es la prestación de servicios, tales como Outsourcing, HelpDesk y la mencionada fábrica de Software. Asimismo, basados en su experiencia cuenta con especializaciones en Aplicaciones para Móviles, Integración de Infraestructura, Business Intelligence, Facturación electrónica y Capacitación TI. En el servicio de fábrica de software surgen problemas que son: la demora en los procesos de aprobación o rechazos de los requerimientos, consulta de información de los requerimientos y la misma gestión de requerimientos. Así que para solucionar estos problemas a un bajo coste y de forma innovadora es por medio de una aplicación web para la gestión de requerimientos.

Una aplicación web para la gestión de requerimientos proporciona muchas ventajas, tales como: optimizar las consultas que se lleven a cabo por el personal encargado de verificar la viabilidad de los requerimientos solicitados a la fábrica de software, optimizar el uso de recursos como tiempo, dinero y personal para la atención de los requerimientos que son aprobados y que pueden iniciar su atención, así mismo tener información del personal que constituye la fábrica de software y los requerimientos que son atendidos en un repositorio de datos.

La aplicación web para la gestión de requerimientos es innovadora e importante para las fábricas de software que deseen llevar un control y seguimiento de los requerimientos que los clientes puedan solicitar. La implementación de la aplicación web puede producir ahorro de tiempo y por lo tanto de dinero ya que el personal de la fábrica de software puede realizar la atención teniendo mapeado los requerimientos que deben ser atendidos en un lapso de tiempo definido.

Por estas razones, este trabajo de investigación busca mejorar ese control que llevan sobre los requerimientos, así mismo el seguimiento que realizan sobre los mismos en la fábrica de software de FabSF Services mediante el desarrollo de una aplicación web para la gestión de requerimientos. La solución tecnológica planteada busca resolver la problemática, la cual se enfoca en el uso de hojas de cálculo de Excel. Los retrasos en tiempo que generan los procesos tienen un impacto sobre la verificación y diagnóstico que realizan los ingenieros para indicar si los requerimientos son viables o no.

El presente trabajo de investigación está conformado por los siguientes capítulos:

Capítulo I: Se explica la situación problemática, el problema general, los problemas específicos, el objetivo general y los objetivos específicos.

Capítulo II: Se explica el marco teórico, que incluye los antecedentes, las bases teóricas y el glosario.

Capítulo III: Se explica las variables, indicadores e hipótesis del presente trabajo de investigación.

Capítulo IV: Se explica de forma resumida, la metodología de desarrollo utilizada.

Capítulo V: Se explica la solución tecnológica, que abarca la elaboración de artefactos de constituyen la metodología de desarrollo utilizada.

Capítulo VI: Se explican los resultados obtenidos.

Finalizando de esta manera, con las conclusiones y recomendaciones.



CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.1. Situación Problemática

La demanda de desarrollo de software crece a un ritmo anual de 8.5% desde el 2010 debido a la fuerte demanda de servicios tecnológicos en el mercado abarcando requerimientos de las entidades financieras que busca un crecimiento a nivel nacional e internacional, y para lograrlo se necesita de creación o mejoras a los aplicativos que ya existen y se usan a diario. Una de las empresas es FabSF Services ubicada en el Centro de Lima que posee más de 11 años de experiencia brindando los servicios de fábrica de software, outsourcing TI, help desk y services office teniendo como especialidades inteligencia empresarial y desarrollo de aplicaciones móviles.

Un punto crítico en la atención de la fábrica de software es que los requerimientos se deben atender en una fecha ya planificada, por lo que se presenta demora en horas variantes (depende de la disponibilidad del líder de proyecto, ingeniero de fábrica y analista de calidad) en la evaluación para decidir si el requerimiento solicitado por el analista TI por parte de la entidad financiera es viable con recursos (ingenieros de software y analistas de calidad) que presenten disponibilidad para iniciar su atención en un fecha especificada.

En la fábrica de software, el líder de proyecto en el proceso de gestión de requerimientos registra los requerimientos en hojas de cálculo y al intentar completar los mismos se requiere consultar otras hojas de cálculo donde se consigue la información requerida porque pueden guardar relación y tener información detallada que aplique para su registro, ocasionando de esta manera la demora de doce minutos en el proceso de registro. Por la cantidad de requerimientos que son solicitados para evaluación tiende a causar confusión al tener que controlar cada requerimiento debido que la información permanece en hojas de cálculo.

Los ingenieros de fábrica no pueden conocer la información de los cambios requeridos en los requerimientos de los cuales serán responsables sin antes tener que abrir distintos documentos donde se verifique la información que solicita, lo cual requiere de tiempo para solo consultar datos necesarios, teniendo así un retraso al momento de consultar información de un requerimiento específico.

Cuando el líder de proyectos desea asignar recursos para la atención de un requerimiento lo debe realizar mediante un documento ingresando datos repetitivos del requerimiento los cuales ya se encuentran en otros documentos generando así pérdida de tiempo en gestión de aproximadamente tres a cinco minutos y en muchas ocasiones esos campos que hacen referencia a los datos del requerimiento quedan vacíos. Con datos vacíos no se puede realizar una comparación efectiva de la información teniendo como consecuencia el perder el seguimiento al requerimiento.

Después de realizar la evaluación se envía una respuesta ya sea que las solicitudes son aprobadas o rechazadas de manera verbal, no teniendo una manera de visualizar estas respuestas en cualquier momento que se solicite y por lo tanto no es posible saber cuáles son los requerimientos que han sido aprobados, así como rechazados a lo largo del tiempo.

También cuando los ingenieros de fábrica desean saber si el requerimiento ha sido aprobado o rechazado no tiene forma de saberlo sin que se le pregunte a los líderes o incluso mismo al supervisor de fábrica y que en muchas ocasiones no está presente en la fábrica, por lo tanto, no tienen fácil acceso a esa información cuando les es conveniente consultarlo para empezar a trabajar sobre el requerimiento. Finalmente se puede apreciar en la figura 1 los problemas que posee la fábrica de software de la empresa FabSF Services.

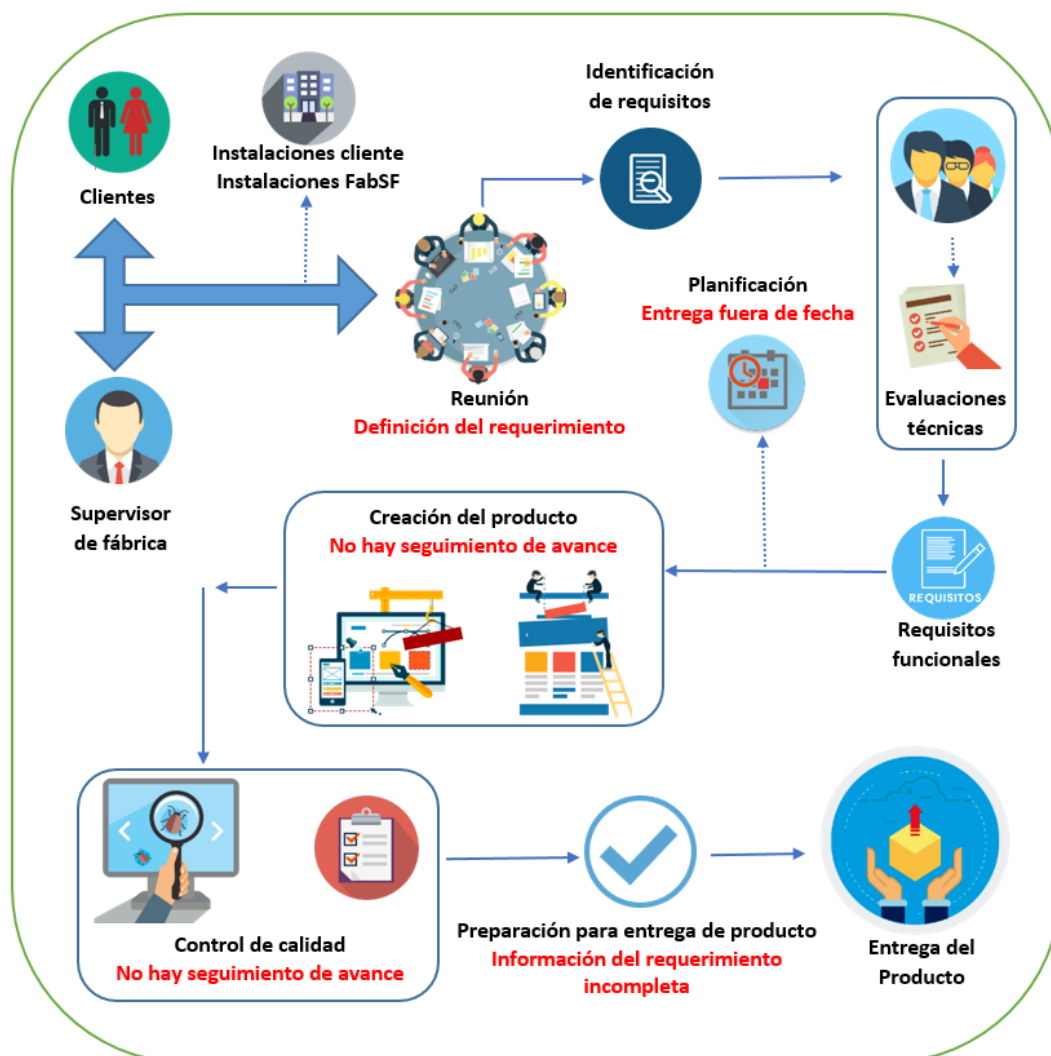


Figura 1: Situación Problemática (Elaboración propia).

1.2. Problema de la investigación

- Problema General:

¿En qué medida la aplicación web influye en la gestión de requerimientos de software en la empresa FabSF Services?

- Problemas Específicos:

- *¿En qué medida el nivel de funcionalidad de la aplicación web influye en la gestión de requerimientos de software en la empresa FabSF Services?*

- *¿En qué medida el nivel de usabilidad de la aplicación web influye en la gestión de requerimientos de software en la empresa FabSF Services?*

- *¿En qué medida el nivel de eficiencia de la aplicación web influye en la gestión de requerimientos de software en la empresa FabSF Services?*

- *¿En qué medida el nivel de fiabilidad de la aplicación web influye en la gestión de requerimientos de software en la empresa FabSF Services?*

1.3. Objetivos

- Objetivo General:

Determinar la influencia de una aplicación web en la gestión de requerimientos de software en la empresa FabSF Services.

- Objetivos Específicos:

- *Determinar la influencia del nivel de funcionalidad de la aplicación web en la gestión de requerimientos de software en la empresa FabSF Services.*

- *Determinar la influencia del nivel de usabilidad de la aplicación web en la gestión de requerimientos de software en la empresa FabSF Services.*

- *Determinar la influencia del nivel de eficiencia de la aplicación web en la gestión de requerimientos de software en la empresa FabSF Services.*

- *Determinar la influencia del nivel de fiabilidad de la aplicación web en la gestión de requerimientos de software en la empresa FabSF Services.*

1.4. Justificación

En el trabajo de investigación se propone el desarrollo de una aplicación web para la mejora del control de solicitudes de requerimientos, reducir el tiempo de realización y agilizar dicha operación. Realizar consultas relacionadas a la información de los requerimientos, ya sea por el líder de proyecto a cargo, recursos asignados para atención, fecha de planificación y horas de atención de una forma precisa mostrando la información solicitada en menor tiempo del que se realiza actualmente. Así mismo dichos recursos podrían ser asignados directamente desde la aplicación web y ser enlazados al requerimiento que se desea atender.

Llevar el control y seguimiento sobre los requerimientos ya que los líderes de proyectos podrían consultar el estado de los mismos sin demandar mucho esfuerzo y tiempo, así como la visualización de las fechas que son datos importantes que se deben tener muy en cuenta. Con esto se planifica la culminación de las pruebas realizadas por el área de calidad antes de la fecha de finalización establecida y así entregar los requerimientos sin sobrepasar la fecha de finalización definida, que suelen tener de dos a cuatro semanas de duración, y así el cliente pueda evaluar el producto final logrando de esa manera la satisfacción del mismo. Teniendo en cuenta que esto ocasiona que el cliente pueda confiar más en la empresa generando así mucha demanda de requerimientos para la fábrica de software y mejorar su productividad logrando competitividad en el mercado.

Se automatiza el procedimiento de atención de los requerimientos, manejando de una manera eficaz la información que se obtiene al registrar los mismos para la atención de los cambios solicitados a los aplicativos ya creados de los clientes que posee la empresa. Se mejora la consulta de la información realizándolo en un rango de 5 a 30 segundos.

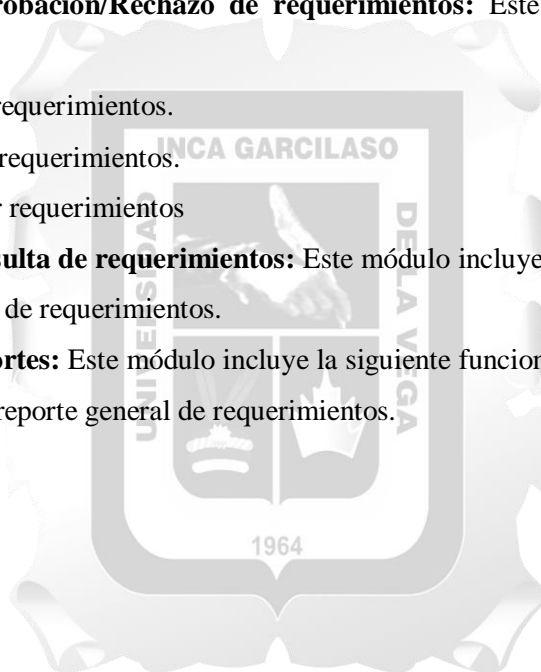
1.5. Alcance

El presente trabajo de investigación tiene como alcance optimizar el control y seguimiento de los requerimientos que son solicitados a la fábrica de software permitiendo la aprobación y rechazo de los mismos al finalizar su respectiva evaluación, considerando los siguientes aspectos:

- Estudio de la situación problemática de la fábrica de software de FabSF Services.
- Estudio de metodologías para el desarrollo de la aplicación web para la gestión de requerimientos.
- Adaptación de la metodología SCRUM para resolver la problemática de la fábrica de software de FabSF Services.
- Ejecución de la metodología SCRUM y elaboración de los artefactos de cada fase de la metodología SCRUM.

La solución comprende los siguientes módulos:

- **Módulo de Seguridad:** Este módulo incluye las siguientes funcionalidades:
 - Login.
 - Gestionar Usuarios.
- **Módulo de Mantenimiento:** Este módulo incluye las siguientes funcionalidades:
 - Gestionar personal
 - Gestionar lenguajes de programación.
 - Gestionar prioridad de requerimientos.
 - Gestionar clientes.
- **Módulo de Gestión de requerimientos:** Este módulo incluye la siguiente funcionalidad:
 - Gestionar requerimientos.
- **Módulo de Aprobación/Rechazo de requerimientos:** Este módulo incluye la siguiente funcionalidad:
 - Evaluar requerimientos.
 - Aprobar requerimientos.
 - Rechazar requerimientos
- **Módulo de Consulta de requerimientos:** Este módulo incluye la siguiente funcionalidad:
 - Consulta de requerimientos.
- **Módulo de Reportes:** Este módulo incluye la siguiente funcionalidad:
 - Generar reporte general de requerimientos.



CAPÍTULO II: MARCO TEÓRICO

2.1. Antecedentes de la investigación

Álvarez Abanto, Kevin David & Tantalean Gil, Gerson Atherley (2016). *“Sistema Web para agilizar la gestión operativa de los proyectos de software para la empresa Consorcio Fábrica de Software”*. Tesis de título profesional. Universidad Nacional de Trujillo. Trujillo - Perú.

El proyecto de tesis surge en la necesidad de solucionar el problema de continuas penalidades que recibe el Consorcio Fábrica de Software por el incumplimiento de entregar el producto en la fecha pactada con el cliente SUNAT (Cliente externo). Dicha tesis tiene la finalidad de demostrar que se puede agilizar la gestión operativa de los proyectos de software en la empresa Consorcio Fábrica de Software, lo que reduce notablemente los tiempos de las tareas en los procesos; específicamente hace referencia a reducir el tiempo en la búsqueda de actividades asignadas a colaboradores, reduce los tiempos en el intercambio de entregables (documentación y software) entre los equipos de trabajo (equipo de análisis, diseño, desarrollo y calidad) y además mejora el nivel de satisfacción de los colaboradores de la empresa Consorcio Fábrica de Software (cliente interno).

Se tiene como resultado la reducción de tiempo en el acceso a la información de los requerimientos, en informar sobre las incidencias, en informar sobre las solicitudes de cambio y reducción en el tiempo de la transferencia de entregables. Como conclusión en los procesos de: acceso a la información de requerimientos, informar incidencias, informar solicitudes de cambio y transferencia de entregables, con el sistema propuesto se aprecia una reducción del tiempo en comparación con el sistema actual. Así mismo se aprecia la satisfacción del cliente interno con respecto a su función laboral.

Holgado San Martín, Antonio Agustín & Díaz Rodenas (2014). *“Diseño de una estrategia de negocios que genere ingresos recurrentes para una empresa de software”*. Tesis para optar grado de Magíster. Universidad de Chile. Santiago de Chile – Chile.

Se indica que la estrategia resultante para generar ingresos en una empresa existente, dedicada principalmente al desarrollo de software, consiste en crear y definir una gerencia con responsabilidad exclusiva para consolidar los servicios de continuidad operativa de aplicaciones, brindando los servicios de soporte a los sistemas incluidos en el alcance de los servicios.

A partir de los objetivos que se plantearon, utiliza un proceso de declaración de lo que se quiere lograr, seguido del análisis de los factores que influyen, para continuar con las definiciones del modelo de negocio y el modelo de gestión como resultado de los procesos anteriores, finalizando con el proceso de ejecución, como una forma de hacer seguimiento y control de la implementación de la estrategia definida. Se tiene como resultado una estrategia que genera ingresos recurrentes mediante

un contrato anual de servicios, considerando como principal costo la mano de obra de los profesionales de la industria de TI.

Y como conclusión el logro de los objetivos planteados en esta tesis, permitirá sustentar el crecimiento de la empresa por diferentes vías, comenzando por la mejora en la productividad de los equipos de trabajo y continuando con la prospección de clientes ampliando el mercado actual, así todo lo indicado servirá de base para iniciar el crecimiento internacional hacia otros países de la región, este es el desafío que motiva la implementación de éstas y otras iniciativas dentro de la empresa.

Coronado Gutiérrez, Jaime (2012). *“Gestión de la empresa Software Factory, mejoras en la operatividad, automatización del proceso de control de recursos y administración del conocimiento en el periodo 2012”*. Tesis de título profesional. Universidad Peruana de Ciencias aplicadas. Perú.

Una de las empresas virtuales formadas, es la llamada “Software Factory”, la cual busca replicar en todo aspecto, la forma de trabajar de una fábrica de software industrializada del mercado actual. De esta manera, se da paso a metodologías modernas de desarrollo que permiten mejorar en cuanto a recursos, tiempo y costos involucrados en el proceso de producción de software, dejando a un lado los métodos tradicionales de desarrollo. Así mismo, la fábrica de software, opera con un modelo ya trabajado el cual permite la gestión de reuso, de los activos trabajados por los colaboradores, de forma apropiada. Por consiguiente, una de sus principales metas es la consolidación como una empresa proveedora de servicios, dentro de la cual destaque el desarrollo de software; diferenciando las líneas de producción de software, según los clientes (Jefes de Proyectos) con los que se trabaje. De esta manera, se espera realizar un proceso de producción y gestión de activos que permita un manejo adecuado de los mismos dentro de la empresa, con la finalidad de mantener una base de conocimiento que persista en el tiempo y sea de utilidad para los colaboradores de la empresa “Software Factory”. Se tiene como resultado el incremento de la satisfacción de los clientes gracias a la web de la empresa, la cual brinda facilidades a sus clientes para el control en el desarrollo de sus proyectos. Como conclusión la herramienta implementada de automatización de procesos de la empresa Software Factory es de gran utilidad para las labores operativas como toma de asistencia, asignación de colaboradores, aceptación de contratos, entre otras.

Cáceres Soto, Gianfranco & Montoya García, Miguel Ángel (2017). *“Propuesta de una arquitectura empresarial para una fábrica de software”*. Tesis de título profesional. Universidad Peruana de Ciencias aplicadas. Perú.

Las empresas que prestan servicios buscan la manera de optimizar el uso de sus recursos, así como dar un servicio de calidad a fin de lograr la satisfacción de sus clientes. Por ello, el proyecto de tesis

surge en la necesidad de solucionar el problema de la deficiente asignación de recursos a los proyectos de software que son vendidos a los clientes en la fábrica de software de la empresa Hewlett Packard Enterprise. Por esta razón se propone una arquitectura empresarial que optimice el proceso de delivery, el cual realiza servicios de estimación y desarrollo de software. Con esta propuesta se espera como resulta que se maximice la capacidad de atención de los proyectos, reducir los tiempos muertos del personal y la reducción de costos generados en la ejecución de los proyectos solicitados. Como conclusión la arquitectura propuesta es más simple y eficiente logrando un fácil mantenimiento y soporte del proceso de delivery. La metodología de desarrollo empleada Scrum hizo posible el desarrollo rápido y flexible de la solución. Además, se indica que la arquitectura planteada como solución no abarcó el proceso de testing ni el de la entrega del producto.

Reyna Méndez, Milton José (2017). “Sistema web integrado para mejorar el seguimiento y control de la gerencia de cobranza en la Sunat - Lima “. Tesis de título profesional. Universidad Nacional de Trujillo. Trujillo - Perú.

Se indica como problema observado que en la gerencia de cobranza de la Sunat carece de disponibilidad de sistemas de información generando así documentación física y por ende se prioriza las hojas de cálculo como herramienta de trabajo, esto dificulta la clasificación de la información, así como el ordenamiento del mismo. También, los ordenadores del personal que labora en la gerencia de cobranza guardan información de los contribuyentes de manera independiente sin tener forma de buscarlos en cualquier ordenador; esta información no se controla de manera eficiente dificultando el seguimiento a los pagos de los contribuyentes. Para todo esto se plantea un modelo de sistema informático que permita tener alineados los procesos de cobranzas. El resultado esperado es optimizar la búsqueda de información en menor tiempo gracias a la implantación y uso del sistema, reducir costos del manejo de la información, reducir tiempos de registro y reducir número de errores a cero. Finalmente se concluye que un sistema de información de gestión comercial mejora significativamente el valor del negocio de la institución que se da porque dispone de información de forma oportuna, debido a que es óptimo y efectivo. Se realiza el debido seguimiento a los pagos de los contribuyentes sin tener carencia en la disponibilidad de sistemas de información.

2.2. Bases teóricas

2.2.1. Aplicación Web

Según Lerma, R. & Murcia, J. & Mifsud, E. (2013, p.16) las aplicaciones web pueden ser clasificadas de acuerdo a varios criterios, ya sea dependiendo de la configuración o modelo arquitectónico que emplee, según la tecnología o lenguaje utilizado o por la capacidad de comunicación que pueda tener con los usuarios en respuesta a sus interacciones con la interfaz que se les presenta. Son estos criterios según los autores mencionados que dan lugar a la clasificación de las aplicaciones web en: estáticas, dinámicas e interactiva:

Clasificación de las aplicaciones web

- **Aplicaciones web estáticas:** Hace referencia donde el usuario recibe una página web cuya interacción no tiene ningún tipo de acción, ni en la propia página, ni genera respuesta alguna por parte del servidor. Habitualmente este tipo de aplicaciones es construido usando el lenguaje de formato de documentos HTML solamente para la visualización de la información. La primera clasificación se evidencia en la figura 2:



Figura 2: Aplicación web estática (Lerma, R. & Murcia, J. & Mifsud, E., 2013).

- **Aplicaciones web dinámicas:** Hacen referencia a las aplicaciones en las que la interacción del cliente con la página web mostrada por el servidor produce algún tipo de cambio en la visualización del mismo, ya sea: cambios de formato, modificaciones en la página web, habilitar/deshabilitar contenido, entre otros. Estas páginas son construidas con los lenguajes: CSS (hojas de estilo) o las variaciones de JavaScript (VBScript, JScript, Flash, etc.) y HTML. La segunda clasificación se evidencia en la figura 3:



Figura 3: Aplicación web dinámica (Lerma, R. & Murcia, J. & Mifsud, E., 2013).

- **Aplicaciones web interactivas:** Hace referencia donde la interacción del usuario produce un cambio en el recurso recibido, las aplicaciones web interactivas se basan en que dicha interacción genera un diálogo entre el cliente y el servidor. En otras palabras, dicho diálogo puede ser ejecutado tanto en el cliente como en el servidor (e incluso en ambos).

El tipo de aplicaciones web interactivas es el que más se utiliza en Internet ya que hay interacción del usuario que produce un cambio en el recurso recibido, las aplicaciones web interactivas se basan en que dicha interacción genera un diálogo entre el cliente y el servidor. Las tecnologías implicadas en este tipo de aplicaciones varían mucho en función de si son ejecutadas en el lado del cliente o en el lado del servidor.

Sin embargo, San Miguel (2015, p.71) afirma que en la ingeniería de software se denomina aplicación web al software que los usuarios pueden utilizar accediendo a un servidor web, bien vía Internet, Extranet o Intranet, utilizando para ello un navegador web.

Otra forma de definirlo es como un conjunto de páginas web que interactúan unas con otras y con diversos recursos de un servidor o con otros elementos, incluidas las bases de datos. La interacción permite implementar aplicaciones como catálogos de productos virtuales o tienda virtual (e-commerce), administración de noticias y contenidos. Puede favorecer notablemente la agilización de tareas administrativas, entre otras.

2.2.2. Arquitecturas web

Según Carballeira (2016, p.162) un sistema de información web emplea la tecnología web para obtener información y realizar la entrega de productos y servicios. Deben:

- Integrar diversas fuentes por arquitecturas multi-capa.
- Ser extensibles para permitir la ampliación de su uso a nuevas tecnologías
- Gestionar información
- Dar información sobre usuarios y su entorno.

Así también se indica que los sitios web son una mezcla compleja de sistemas integrados entre sí y que el resultado final de esta mezcla será un sitio web que pueda resolver las necesidades del negocio: Ya sea servir las necesidades de los clientes y vender productos y servicios online.

Los modelos arquitectónicos han evolucionado desde las aplicaciones de interfaz de entrada común, que permite a un cliente solicitar datos de un programa ejecutado en un servidor web, el paso por las páginas JSP (Java Server Pages) y Servlets, el salto al Modelo Vista Controlador (MVC) hasta el modelo de capas.

Modelo de capas en la arquitectura de software

El término arquitectura cliente/servidor es empleado para hablar de la arquitectura del software. De igual manera, los términos arquitectura en 2 y 3 capas aplican al mismo concepto.

Según indica Carballeira (2016, p.164), las aplicaciones de software incluyen tres aspectos fundamentales:

- Hacer que los datos sean persistentes (D).
- Procesar los datos de forma acorde a la lógica de negocio (L).
- Presentar los datos de forma adecuada a los usuarios (P).

a) Arquitectura en dos capas

Esta arquitectura distribuye la aplicación en dos componentes lógicos, por lo tanto, las responsabilidades son divididas. Permite distribuir la carga que tiene la aplicación en dos ordenadores distintos.

Esta arquitectura presenta desventajas como:

- Incremento en la carga de la red.
- La lógica de la aplicación no puede ser reutilizada.
- Recursos limitados en las estaciones de trabajo.

b) Arquitectura en tres capas

En esta arquitectura la lógica de negocios y el uso de persistencia serán una unidad que presenta considerables desventajas como:

- Actualización de versión
- Gestor de base de datos pequeño solicita migración.

La lógica de la aplicación se encuentra en una capa intermedia para esta arquitectura, la cual está separada de los datos y de la interfaz de usuario. Los límites entre las capas son de carácter lógico, lo que ocasiona que se deben ejecutar las tres capas en el mismo ordenador.

Según Carballeira (2016, p.166), se presenta de la siguiente manera:

- **Presentación:** Presenta las siguientes responsabilidades.
 - Validación de datos de entrada
 - Amigable Navegación en el sistema.
 - Entre otros
- **Negocio:** Genera un conjunto de reglas de negocio que es obtenido como resultado de un análisis realizado funcionalmente.
- **Persistencia:** Comprende la responsabilidad de la lógica de persistencia.
 - Inserciones
 - Actualizaciones
 - Eliminaciones
 - Búsquedas

Finalmente se mencionan ventajas de usar este modelo:

- Separación de la interfaz de usuario de la lógica de la aplicación.
- Los objetos de negocio pueden incluir servicios por la red.
- Los cambios realizados en la información almacenada no influyen sobre la presentación.
- Ejecución de lógica de cada capa de forma independiente

2.2.3. Arquitectura cliente/servidor

El funcionamiento de la Web, indican Lerma, R. & Murcia, J. & Mifsud, E. (2013, p.14), se debe a la coexistencia de elementos software y hardware. Estos elementos abarcan desde los componentes físicos de Internet (hub, repetidores, puentes, entre otros) y los protocolos de comunicaciones (TCP, IP, HTTP, entre otros) hasta el uso del sistema de nombres de dominio (DNS) para la búsqueda y recuperación de recursos.

En este contexto, se entiende que el desarrollo en entornos web debe tener en cuenta la distribución de los elementos y la función que tiene cada uno de ellos. La configuración arquitectónica más habitual se basa en el modelo Cliente/ Servidor, basado en la idea de servicio, en el que el cliente es un elemento consumidor de servicios y el servidor es un proceso proveedor de servicios. En esta relación el intercambio de mensajes es el único elemento de acoplamiento entre los mismos; el elemento que solicita la información se denomina cliente, mientras que el elemento software que responde a esa solicitud se le conoce como servidor.

El proceso habitual se da cuando el cliente inicia el intercambio de información, solicitando datos al servidor, y este responde enviando uno o más flujos de datos al solicitante, que en este caso es el cliente; se puede apreciar lo entendido en la figura 4:

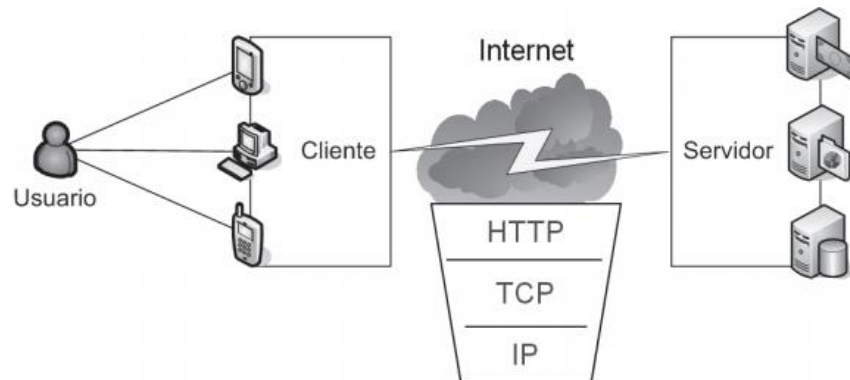


Figura 4: Arquitectura cliente/servidor (Lerma, R. & Murcia, J. & Mifsud, E., 2013).

2.2.4. Modelos de Desarrollo de Software

El Modelo Vista Controlador (MVC) según Carballeira (2016, p.221) es un patrón para el desarrollo del software, usado sobre todo en aplicaciones web. Está basado en la separación de tres componentes, que se visualizan en la figura 5 mostrada a continuación:

- Los datos - Modelo
- Lógica interna - Controlador
- Interface de usuario - Vista

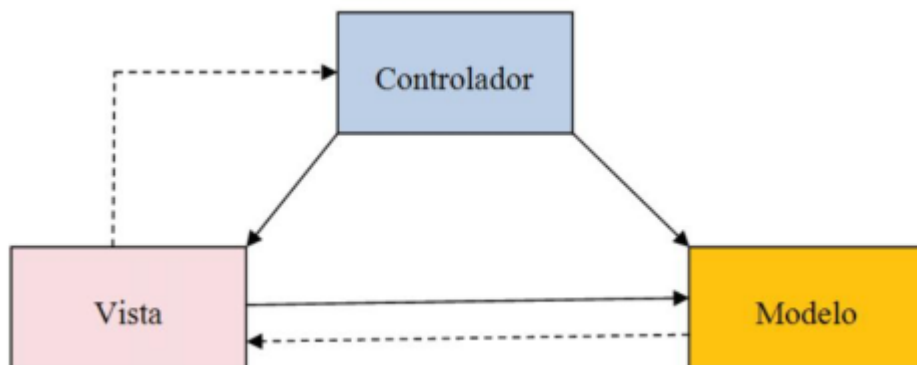


Figura 5: Modelo de desarrollo MVC (Carballeira, 2016).

El modelo trabaja cuando el usuario envía un formulario, lo cual gestiona el controlador, y genera la consulta hacia la información que se tiene almacenada en la base de datos mediante el modelo, después se envía una respuesta a la interfaz del usuario en donde el mismo puede visualizar lo solicitado en la aplicación.

Entre los framework que emplean este modelo según Carballeira (2016, p.221) se tienen a los siguientes, mostrados en la tabla 1:

FRAMEWORK	LENGUAJE DE PROGRAMACIÓN
CodeIgniter	PHP
Zend	PHP
Ruby on Rails	Ruby
Spring	Java
Catalyst	OPerl
Django	Python

Tabla 1. Framework por lenguaje de programación (Elaboración propia)

2.2.5. Requerimientos de software

Según Gómez (2011, p.3) los requerimientos especifican qué es lo que el sistema debe hacer (en otras palabras: sus funciones) y sus propiedades esenciales y deseables. Principalmente la recopilación de los requerimientos tiene como objetivo principal la comprensión de lo que los clientes y los usuarios esperan que haga el sistema. Un requerimiento expresa el propósito del sistema sin considerar como se va a implantar. Para ser más preciso en el tema, los requerimientos identifican el “qué” del sistema, mientras que el diseño establece el “cómo” del sistema. La recopilación y el análisis de los requerimientos del sistema es una de las fases más importantes para que el proyecto tenga éxito. Como regla se tiene que, el costo de reparar un error se incrementa en un factor de diez de una fase de desarrollo a la siguiente, por lo tanto, la preparación de una especificación adecuada de requerimientos reduce los costos y el riesgo general asociado con el desarrollo.

2.2.6. Gestión de Requerimientos

Para Gómez (2011, p.17) en la práctica, en casi todos los sistemas los requerimientos cambian. Las personas involucradas desarrollan una mejor comprensión de lo que quieren que haga el software, la organización que compra el sistema cambia, se hacen modificaciones a los sistemas de hardware, software y al entorno organizacional.

El proceso de organizar y llevar a cabo los cambios en los requerimientos se llama gestión de requerimientos.

El objetivo del analista es reconocer los elementos básicos de un sistema tal como lo percibe el usuario/cliente. El analista debe establecer contacto con el equipo técnico y de gestión del usuario/cliente y con la empresa que vaya a desarrollar el software. El gestor del programa puede servir como coordinador para facilitar el establecimiento de los caminos de comunicación.

Una vez que un sistema se ha instalado, inevitablemente surgen nuevos requerimientos.

Es difícil para los usuarios y clientes del sistema anticipar qué efectos tendrá el sistema nuevo en la organización. Cuando los usuarios finales tienen experiencia con un sistema, descubren nuevas necesidades y prioridades.

Las personas que pagan por el sistema y los usuarios de éste, rara vez son la misma persona. Los clientes del sistema imponen requerimientos debido a las restricciones organizacionales y de presupuesto. Éstos pueden estar en conflicto con los requerimientos de los usuarios finales y, después de la entrega, pueden tener que añadirse nuevas características de apoyo al usuario para que el sistema cumpla con sus objetivos.

Otro motivo por el que cambian los requerimientos es que en ocasiones el entorno de negocios y técnico del sistema cambian después de la instalación. Puede ser que se introduzca un nuevo hardware, o puede ser que surja la necesidad de que el sistema interactúe con otros sistemas. También cambian las prioridades del negocio, las legislaciones y las regulaciones, y esto debe estar reflejado en sistema.

“La gestión de requerimientos es el proceso de comprender y controlar los cambios en los requerimientos del sistema” (Sommerville, 2005).

Es necesario mantenerse al tanto de los requerimientos particulares y mantener vínculos entre los requerimientos dependientes de forma que se pueda evaluar el impacto de los cambios en los requerimientos. El proceso de gestión de requerimientos debe empezar cuando esté disponible una versión preliminar del documento de requerimientos.

Hay que establecer un proceso formal para implantar las propuestas de cambios y planear como se van a gestionar los requerimientos que cambian durante el proceso de obtención de requerimientos.

En la Figura 6 se muestra como se tiene una mejor comprensión de las necesidades de los usuarios conforme se va desarrollando la definición de los requerimientos.

Esta nueva comprensión retroalimenta al usuario, quien puede proponer entonces un cambio en los requerimientos.

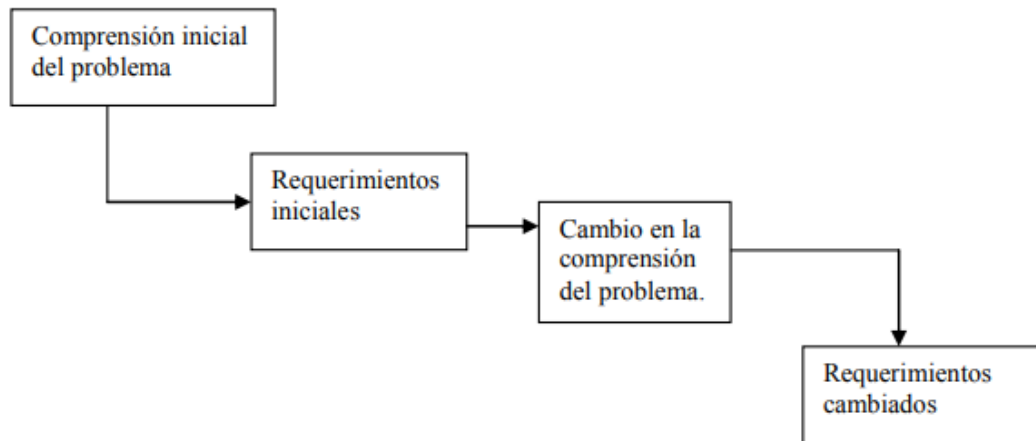


Figura 6: Evolución de los requerimientos (Sommerville 2005).

Desde una perspectiva evolutiva, los requerimientos se dividen en dos clases:

- ❖ **Requerimientos duraderos.** Son los relativamente estables, están relacionados directamente con la actividad principal de la organización.
- ❖ **Requerimientos volátiles.** Cambian durante el proceso de desarrollo o después de que éste se haya puesto en funcionamiento.

La gestión del cambio en los requerimientos según Sommerville (2005, p.149) se debe aplicar a todos los cambios propuestos en los requerimientos, teniendo como flujo el mostrado en la figura 7. La ventaja de utilizar un proceso formal para gestionar el cambio es que todos los cambios propuestos son tratados de forma consistente y que los cambios en el documento de requerimientos se hacen de forma controlada.

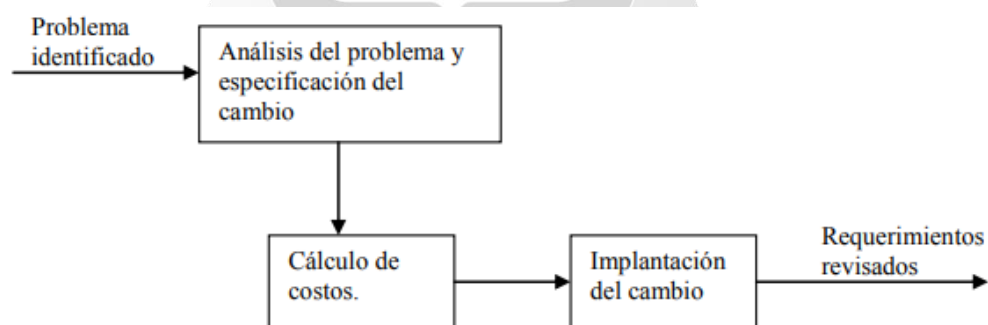


Figura 7: Gestión de cambios en los requerimientos (Sommerville 2005).

El proceso empieza con la identificación de un problema en los requerimientos o, algunas veces con una propuesta específica de cambio, que se analiza para verificar que sea válido. Después se valora el efecto del cambio propuesto para calcular su costo. Este costo se estima tomando en cuenta las modificaciones al documento de requerimientos, y si es el caso, al diseño e implementación del sistema.

Al finalizar esta etapa se toma una decisión sobre si se continúa con el cambio de requerimientos, de ser así, se lleva a cabo la implantación del cambio.

El documento de requerimientos debe organizarse haciendo sus secciones tan modulares como sea posible, de tal modo que se puedan hacer cambios en él sin tener que hacer grandes reorganizaciones o redactar nuevamente gran cantidad del mismo. Así se podrán cambiar o reemplazar secciones individuales sin afectar a otras partes del documento.

Algunas veces pasa que se requiere de forma urgente un cambio en los requerimientos. Entonces surge la tentación de hacer ese cambio directamente en el sistema y luego modificar de forma retrospectiva el documento de requerimientos. Esto conduce casi inevitablemente a que la especificación de requerimientos y la implementación del sistema se desfasen. Una vez que se han hecho los cambios en el sistema, los del documento de requerimientos se pueden olvidar o se hacen de forma que no concuerdan con los cambios del sistema.

Riesgos en la identificación de requerimientos

Según Gómez (2011, p.20) el desarrollo de proyectos complejos de software lleva implícita la existencia de ciertos riesgos que, si no se toman en cuenta y se analizan con cuidado, retrasarán considerablemente la entrega del producto o incluso podrían llegar a causar la cancelación del proyecto.

Un riesgo se puede definir, conforme a (Sommerville, 2005) y (Pressman, 2002), como:

- La probabilidad de que una circunstancia adversa ocurra.
- Un problema potencial que puede ocurrir o no.

Además (Soto & González, 2010) identifican los siguientes:

- No se comprende claramente el alcance del sistema.
- No se logran identificar con claridad los productos resultantes del proyecto.
- Ni los integrantes del equipo de desarrollo ni el cliente logran especificar de forma apropiada el área de aplicación.

Establecer buenas relaciones con los clientes permite identificar mejor los riesgos y controlarlos durante el desarrollo del proyecto.

2.2.7. Indicadores para la calidad de software

Según Largo & Marin (2005, p.11) la ISO/IEC 9126 permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoria de software. Los modelos de calidad lo describen así:

A. Calidad Interna y Externa

Se especifica seis características para la calidad interna y externa, las cuales, tienen criterios. Estos criterios se manifiestan externamente cuando el software es utilizado como parte de un sistema Informático, y son el resultado de atributos internos de software. Estas características se muestran en la figura 8.



Figura 8: Características de la calidad interna y externa. (Largo & Marin, 2005)

a) Funcionalidad

Funcionalidad según Largo & Marin (2005, p.15) es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es usado en condiciones específicas.

Un claro ejemplo para entenderlo mejor: es lo que hace el software para satisfacer las necesidades sin tener en cuenta de “cómo” lo hace y “cuando”. Estas características se muestran en la figura 9.

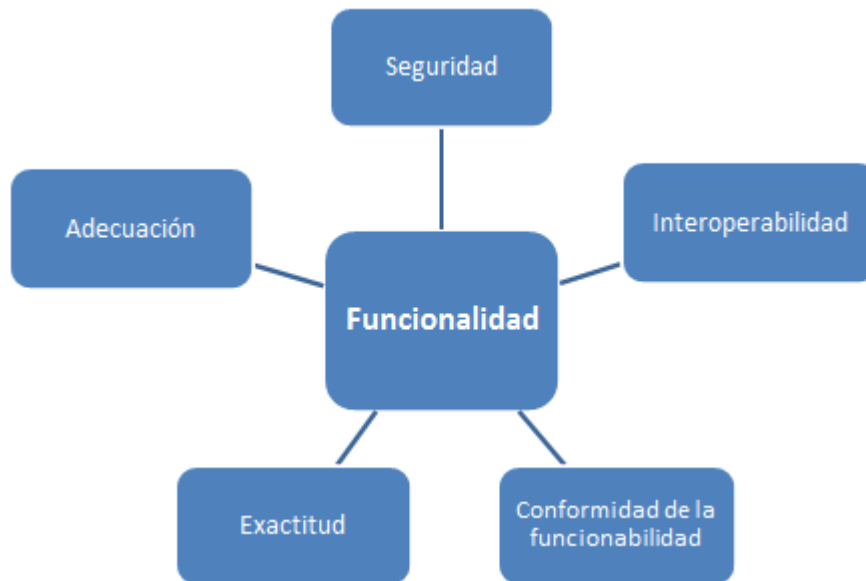


Figura 9: Criterios de la funcionalidad. (Largo & Marin, 2005)

Indican Largo & Marin (2005, p.15-16) que la funcionalidad se divide en 5 criterios:

- **Exactitud.** La capacidad del software para hacer procesos y entregar los resultados solicitados con precisión o de forma esperada.
- **Adecuación.** La capacidad del software para proveer un adecuado conjunto de funciones que cumplan las tareas y objetivos especificados por el usuario.
- **Seguridad.** La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados
- **Interoperabilidad.** La capacidad del software de interactuar con uno o más sistemas específicos.
- **Conformidad de la funcionabilidad.** La capacidad del software de cumplir los estándares referentes a la funcionalidad.

b) **Confiabilidad**

La confiabilidad expresada por Largo & Marin (2005, p.17) es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es usado en condiciones específicas.

Además, se indica cual es la definición de confiabilidad en la ISO/IEC 2382-14:1997, que es: "la habilidad de la unidad funcional de realizar una función requerida...".

En este caso se entiende que la confiabilidad debe abarcar un nivel específico de funcionamiento y no enfocarse a una función requerida. Estas características se muestran en la figura 10.



Figura 10: Criterios de la confiabilidad. (Largo & Marin, 2005)

Según Largo & Marin la confiabilidad se divide en 4 criterios:

- **Madurez.** La capacidad que tiene el software para evitar fallas cuando encuentra errores.
- **Tolerancia a errores.** La capacidad que tiene el software para mantener un nivel de funcionamiento en caso de errores.
- **Recuperabilidad.** La capacidad que tiene el software para restablecer su funcionamiento adecuado y recuperar los datos que pudiesen haber sido afectados en caso se presenten fallas.
- **Conformidad de la confiabilidad.** La capacidad del software de cumplir los estándares o normas relacionadas a la fiabilidad.

c) Usabilidad

La usabilidad según Largo & Marin (2005, p.18) es la capacidad del software de ser entendido, y usado en forma fácil y sin complicaciones.

Además, se explica que algunos criterios de funcionalidad, fiabilidad y eficiencia afectan la usabilidad, pero para los propósitos de la ISO/IEC 9126 ellos no clasifican como usabilidad.

La usabilidad está determinada por los usuarios que manejan el software, dirigidos para todos los ambientes, a la preparación del uso y el resultado obtenido. Estas características se muestran en la figura 11.



Figura 11: Criterios de la usabilidad. (Largo & Marin, 2005)

Estos criterios son:

- **Entendimiento.** La capacidad que tiene el software para permitir al usuario final entender si es adecuado, y de una manera sencilla como ser utilizado para las tareas y las condiciones particulares de la aplicación. También se debe tener en cuenta la documentación y de las ayudas que el software pueda contener.
- **Aprendizaje.** La forma como el software permite al usuario final aprender su uso. Además, se debe considerar la documentación ya que es importante.
- **Operabilidad.** La manera como el software permite al usuario operarlo y controlarlo.
- **Atracción.** La presentación del software debe ser atractivo al usuario que lo usará. Esto se refiere a las cualidades del software para hacerlo más agradable.
- **Conformidad de usabilidad.** La capacidad del software de cumplir los estándares o normas relacionadas a su usabilidad.

d) Eficiencia

Según Largo & Marin (2005, p.20) la eficiencia del software es la forma del desempeño adecuado, de acuerdo a un número recursos utilizados según las condiciones definidas. Además de ello se debe tener en cuenta otros aspectos como la configuración de hardware, el sistema operativo que se usa, entre otros. Estas características se muestran en la figura 12.



Figura 12: Criterios de la eficiencia. (Largo & Marin, 2005)

Definen 3 criterios que son:

- **Comportamiento de tiempos.** Los tiempos adecuados de respuesta y procesamiento, el rendimiento cuando realiza su función en condiciones específicas.
- **Utilización de recursos.** La capacidad del software para utilizar cantidades y tipos adecuados de recursos cuando este funciona bajo requerimientos o condiciones establecidas.
- **Conformidad de eficiencia.** La capacidad que tiene el software para cumplir con los estándares o convenciones relacionados a la eficiencia.

e) **Mantenibilidad**

La mantenibilidad según Largo & Marin (2005, p.21) es la cualidad que tiene el software para ser modificado, esto incluyendo correcciones o mejoras del software que se podría realizar, cambios en el entorno, y especificaciones de requerimientos funcionales.

Estas características se muestran en la figura 13.

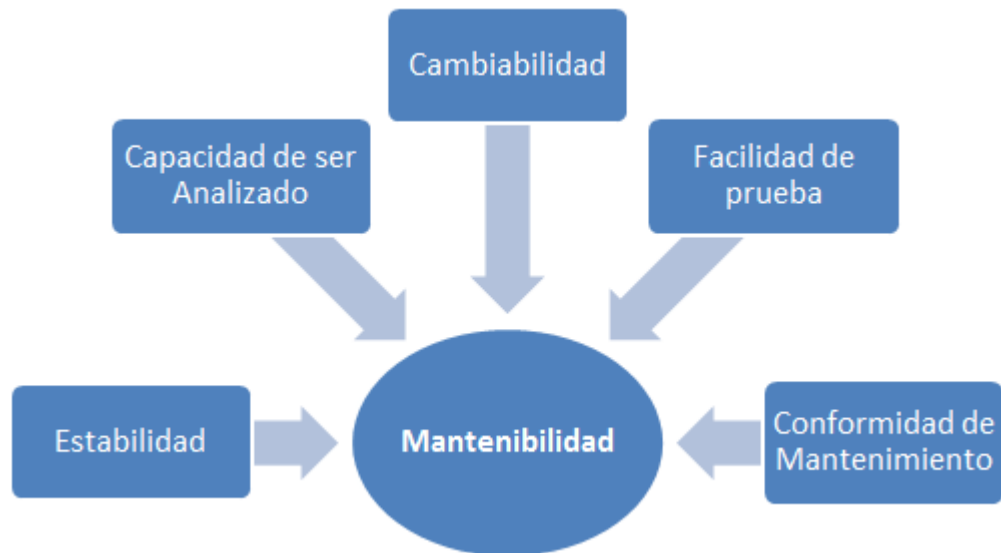


Figura 13: Criterios de la mantenibilidad. (Largo & Marin, 2005)

Largo & Marin (2005, p.22) describen sus 5 criterios:

- **Capacidad de ser analizado.** La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas.
- **Cambiabilidad.** La capacidad del software para que la implementación de una modificación se pueda realizar, incluye también codificación, diseño y documentación de cambios.
- **Estabilidad.** La forma como el software evita efectos inesperados para modificaciones del mismo.
- **Facilidad de prueba.** La forma como el software permite realizar pruebas a las modificaciones sin poner el riesgo los datos.
- **Conformidad de mantenimiento.** La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento.

f) Portabilidad

La portabilidad explicada según Largo & Marin (2005, p.23) es la capacidad que tiene el software para ser trasladado de un entorno a otro. Estas características se muestran en la figura 14.

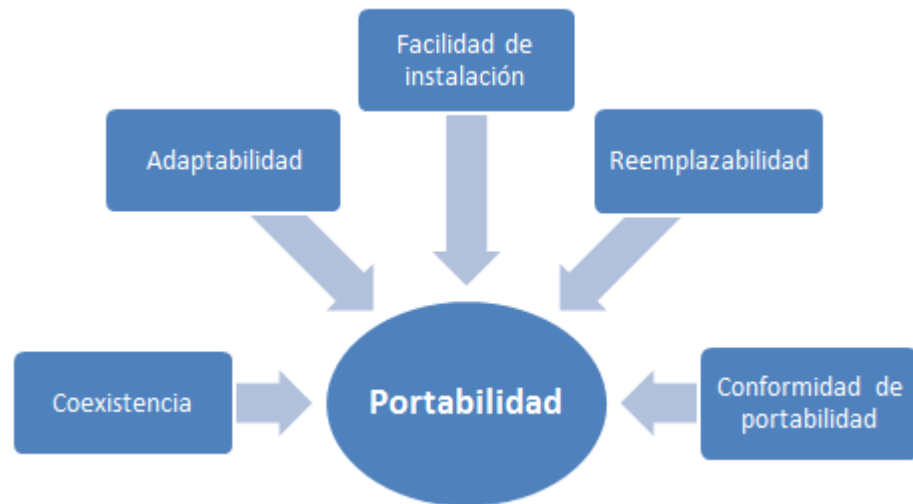


Figura 14: Criterios de la portabilidad. (Largo & Marin, 2005)

Se detallan 5 criterios:

- **Adaptabilidad.** Es como el software se adapta a distintos entornos específicos sin que reaccione de manera negativa ante el cambio.
- **Facilidad de instalación.** La facilidad del software para ser instalado en un entorno específico o donde desee el usuario final.
- **Coexistencia.** La capacidad que tiene el software para coexistir con otro o varios software, la forma de compartir los recursos comunes con otro software o dispositivo.
- **Reemplazabilidad.** La capacidad que tiene el software para ser reemplazado por otro software del mismo tipo, y para el mismo objetivo.
- **Conformidad de portabilidad.** La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad.

B. Calidad en Uso

Es el efecto combinado de las seis características de la calidad interna y externa del software. Largo & Marin (2005, p.12) menciona que hay cuatro características para la calidad en uso. Estas características se muestran en la figura 15.



Figura 15: Características de la calidad en uso. (Largo & Marin, 2005)

Al unir la calidad interna y externa con la calidad en uso se forma un modelo de evaluación más completo, se podría entender que la usabilidad del modelo de calidad externa e interna pueda ser igual al modelo de calidad en uso, sin embargo, no es así, la usabilidad es la forma de como los profesionales interpretan la funcionabilidad del software, mientras que la calidad en uso se puede entender como la forma que lo maneja el usuario final.

Largo & Marin (2005, p.12) indica que, si se unen los dos modelos, podríamos definir que las seis características del primer modelo tienen sus atributos y el modelo de calidad en uso sus cuatro características que sería un atributo, quedando como se aprecia en la figura 16.



Figura 16: Características de la Calidad ISO/IEC 9126. (Largo & Marin, 2005)

La calidad del software se evalúa teniendo en cuenta la etapa del desarrollo, se deben fijar las metas de la calidad en el desarrollo del producto final, así como en el mismo ya finalizado y tener en cuenta que las metas y criterios no son iguales si se trata de un software pequeño y un gran software empresarial.

2.3. Glosario de términos

- **Aplicación web.** Es una aplicación a la que se accede mediante un navegador y está alojada en algún servidor que cumple con los requisitos requeridos para el correcto funcionamiento de la aplicación dentro de una intranet o internet. (Niño, 2010).
- **Eficiencia.** Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas (ISO/IEC 9126-1:2001, 2001).
- **Fiabilidad.** Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido (ISO/IEC 9126-1:2001, 2001).
- **Funcionalidad.** Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen las necesidades implícitas o explícitas (ISO/IEC 9126-1:2001, 2001).
- **Usabilidad.** Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios (ISO/IEC 9126-1:2001, 2001).
- **Procesos de software.** Conjunto de actividades que tiene como objetivo desarrollar un software. Cada organización, conforme su conveniencia, crea y organiza estas actividades con el fin de lograr las especificaciones propuestas. (Arias, 2014).
- **Validación de requerimientos.** Actividad de validar el documento de especificación del requerimiento para verificar que no contenga errores que puedan generar cambios en el sistema en desarrollo o ya desarrollado y con consiguiente realizar un re trabajo tanto para realizar el desarrollo nuevo como la validación del mismo. (Sommerville, 2005).
- **Gestión de requerimientos.** La gestión de requerimientos es el proceso de comprender y controlar los cambios en los requerimientos del sistema (Sommerville, 2005).
- **Fábrica de Software.** Organización especializada que tiene por objetivo la producción de software obteniendo un producto final que contenga la calidad requerida por el cliente. Se logra esto a partir de la maximización de la efectividad de recursos, la estimación precisa y la minimización del tiempo y los costos, así como la mejora continua de la calidad de sus procesos y de sus productos. (Carranza, Z., 2008)

CAPÍTULO III: VARIABLES E HIPÓTESIS

3.1. Variables e Indicadores

a. Identificación de Variables

- Variable Independiente: *Aplicación web*
- Variable Dependiente: *Gestión de requerimientos de software en la empresa FabSF Services*

b. Operacionalización de Variables

- Indicadores Variable Independiente
 - Nivel de funcionalidad
 - Nivel de usabilidad
 - Nivel de eficiencia
 - Nivel de fiabilidad
- Indicadores Variable Dependiente:
 - Número de recursos asignados por requerimiento.
 - Número de requerimientos atendidos por fábrica de software.
 - Número de requerimientos en atención por fábrica de software.
 - Número de derivaciones realizadas por Gestor de Proyectos/Ingeniero de Software/Analista QA.

3.2. Hipótesis

- Hipótesis General:
El desarrollo de una aplicación web influye significativamente en la gestión de requerimientos de software en la empresa FabSF Services.
- Hipótesis Específicas:
 - *El nivel de funcionalidad de la aplicación web influye significativamente en la gestión de requerimientos de software en la empresa FabSF Services.*
 - *El nivel de usabilidad de la aplicación web influye significativamente en la gestión de requerimientos de software en la empresa FabSF Services.*
 - *El nivel de eficiencia de la aplicación web influye significativamente en la gestión de requerimientos de software en la empresa FabSF Services.*
 - *El nivel de fiabilidad de la aplicación web influye significativamente en la gestión de requerimientos de software en la empresa FabSF Services.*

CAPÍTULO IV: METODOLOGÍA PARA EL DESARROLLO DE LA INVESTIGACIÓN

Para el desarrollo del presente trabajo de investigación, se utilizó la metodología ágil SCRUM. Es una metodología de adaptación, iterativa, rápida, flexible y eficaz, diseñada para ofrecer un valor significativo de forma rápida en todo el proyecto. SCRUM garantiza transparencia en la comunicación y crea un ambiente de responsabilidad colectiva y de progreso continuo. El marco de Scrum, está estructurado de tal manera que es compatible con los productos y el desarrollo de servicios en todo tipo de industrias y en cualquier tipo de proyecto, independientemente de su complejidad (Satpathy, 2016).

4.1. Descripción de la Metodología

A continuación, se definirá la metodología Scrum, sus fases, los roles que participan, los artefactos que se elaboran y las actividades que se llevan a cabo en el desarrollo de la metodología:

4.1.1. Definición de Scrum

Según Alaimo (2013, p.21) Scrum es un marco de trabajo que nos permite encontrar prácticas emergentes en dominios complejos, como la gestión de proyectos de innovación. En lugar de proporcionar una descripción completa y detallada de cómo deben realizarse las tareas de un proyecto, genera un contexto relacional e iterativo, de inspección y adaptación constante para que los involucrados vayan creando su propio proceso. Esto ocurre debido a que no existen ni mejores ni buenas prácticas en un contexto complejo, ya que es el equipo de involucrados quien encontrará la mejor manera de resolver los problemas que poseen, tal como se muestra en la figura 17:

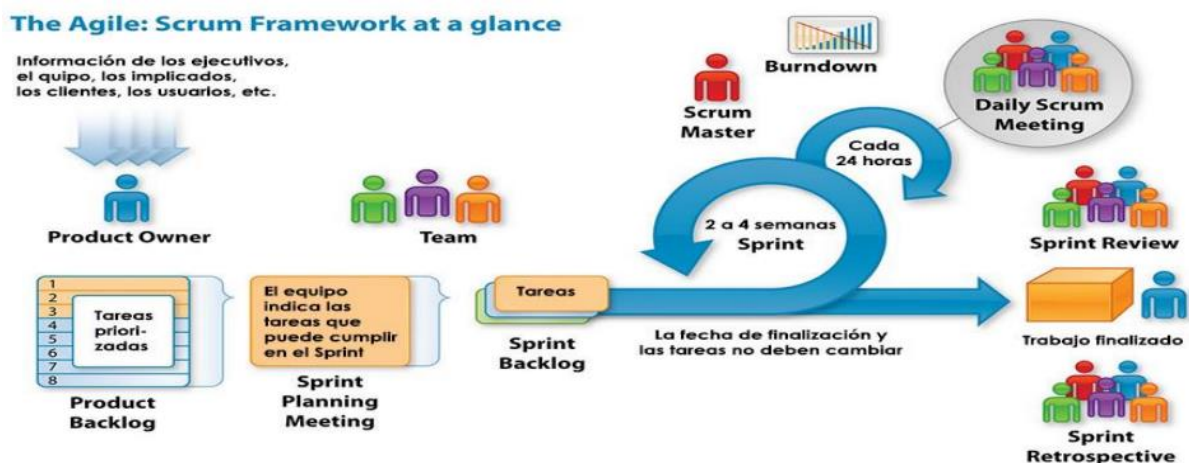


Figura 17: Ciclo de vida SCRUM. (Ponluiza, 2016)

El progreso de los proyectos que emplean Scrum se realiza y se verifica en una serie de iteraciones que tienen por nombre Sprints que tienen una duración pre-definida de no más de un mes.

Al comienzo de cada Sprint el equipo de desarrollo realiza un compromiso de entrega de una serie de funcionalidades o características del producto en cuestión.

Luego al finalizar el Sprint se espera que estas características comprometidas estén terminadas, lo que implica que pase por varias etapas como es el análisis, diseño, desarrollo, prueba e integración del producto.

En este momento es cuando se realiza una reunión de revisión del producto construido durante el Sprint, donde el equipo de desarrollo muestra lo construido al Product Owner y a cualquier stakeholder (interesados) que desee participar de la reunión. La retroalimentación obtenida en esta reunión puede ser incluida entre las funcionalidades a construir en futuros Sprints relacionados al proyecto.

4.1.2. Fases de Scrum

Las fases son 3: la primera fase de Pre-juego, donde se realiza la planificación del proyecto y se define la arquitectura del producto a desarrollar, la segunda fase Juego, donde se realiza el desarrollo del producto, y la tercera fase es Post-juego, donde se realiza el lanzamiento del producto con la debida aprobación del product owner.

A. Fase I: Pre-juego:

Según Caso (2014, p.2) se compone de dos etapas:

Planificación: Etapa donde se necesita información referente a la planificación del proyecto (hasta obtener el producto deseado por el cliente), y abarca actividades como: la visión, el presupuesto, el financiamiento, las herramientas de desarrollo, definir un equipo de trabajo, el product backlog o pila de producto y la planificación de los sprints a realizar:

- **Visión:** Se menciona lo que se pretende desarrollar con su finalidad.
- **Presupuesto:** Se calcula el costo del desarrollo mostrándose en un resumen de costos elaborado en base a: costo de personal, costo de suministros y costo de capacitación a usuarios finales.
- **Financiamiento:** Se indica el costo final para el desarrollo del producto y por quién será financiado.
- **Herramientas de desarrollo:** Se especifica las herramientas empleadas para la construcción del producto.
- **Equipo de Trabajo:** Se menciona a los encargados del proyecto: Scrum Master, Product Owner y Scrum Team o equipo de desarrollo.
- **Product Backlog o Pila de Producto:** Se detalla en una lista los requerimientos definidos mostrándose: código, nombre de la historia de usuario y la estimación en horas.
- **Planificación de sprints:** Se menciona la cantidad de sprint que tendrá y se detalla en una lista la planificación de los sprint indicando: código, nombre de historia de usuario,

número de sprint que contiene la historia de usuario, responsable del sprint, fecha de inicio y fin de desarrollo para cada historia de usuario.

Arquitectura: Etapa donde se presenta el diseño de la implementación de las funcionalidades en base a los ítems del product backlog. Se muestra lo siguiente: arquitectura tecnológica de la solución y arquitectura funcional de la solución tecnológica:

- **Arquitectura tecnológica de la solución:** Se muestra como está constituida el producto en número de capas.
- **Arquitectura funcional de la solución tecnológica:** Se muestran los módulos que tendrá el desarrollo, en otras palabras, se muestra la representación de las funcionalidades del producto.

B. Fase II: Juego:

En esta etapa se realiza el desarrollo de la aplicación. Esta etapa se divide en las iteraciones especificadas en la fase de pre-juego, se debe desarrollar mencionando inicialmente el número de sprint en el que se encuentra cada historia de usuario y luego detallar las historias de usuarios que contempla cada sprint.

C. Fase III: Post-juego:

Esta fase comienza cuando las características especificadas por el product owner se han completado. Dando lugar así a la debida preparación para el lanzamiento de la versión, incluyendo las pruebas antes del lanzamiento de la misma.

4.1.3. Roles de Scrum

Ponluiza, J. (2016, p. 50) indica que en un “equipo scrum” o “team scrum” se espera la intervención de tres roles en la construcción del producto solicitado: Product Owner (Propietario del producto), Equipo de Desarrollo y ScrumMaster. Estos roles se muestran en la figura 18.

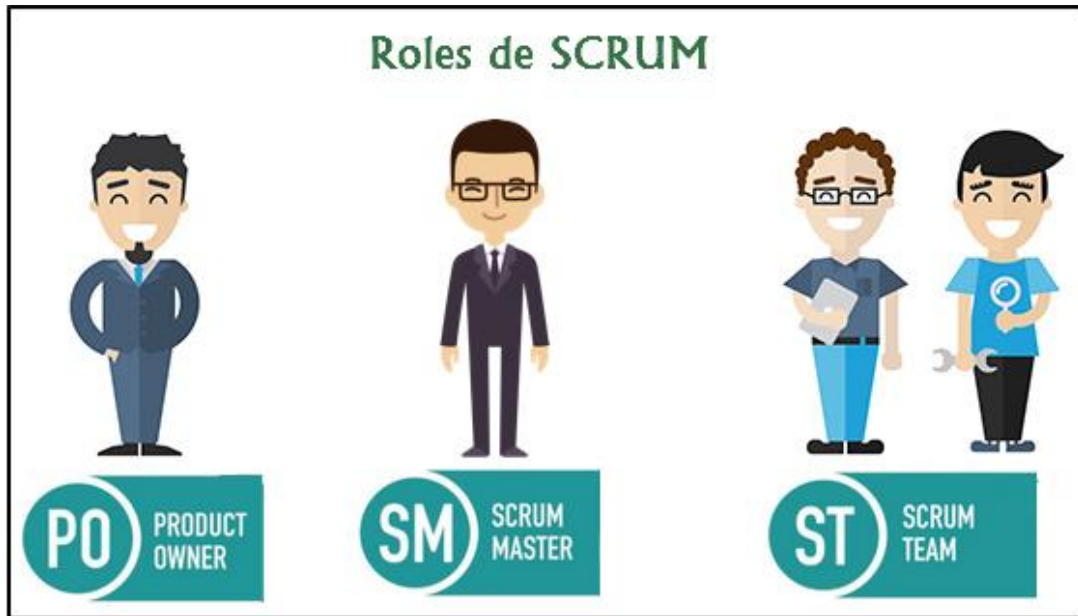


Figura 18: Roles de Scrum (Elaboración propia)

A. Product Owner

El Product Owner es la persona que tiene la responsabilidad de que el producto resulte exitoso desde el punto de vista de los stakeholders (interesados), se visualiza en la figura 19.

Sus principales responsabilidades según Alaimo (2013, p.26) son:

- Determinar la visión del producto.
- Gestionar las expectativas de los stakeholders.
- Recolectar los requerimientos.
- Determinar y conocer en detalle las características funcionales de bajo y de alto nivel.
- Generar y mantener el plan de entregas (release plan).
- Maximizar la rentabilidad del producto.
- Determinar las prioridades de cada una de las características por sobre el resto.
- Cambiar las prioridades de las características según avanza el proyecto.
- Participar de la revisión del Sprint junto a los miembros del Equipo de Desarrollo.
- Aceptar/rechazar el producto construido durante el Sprint y proveer feedback.

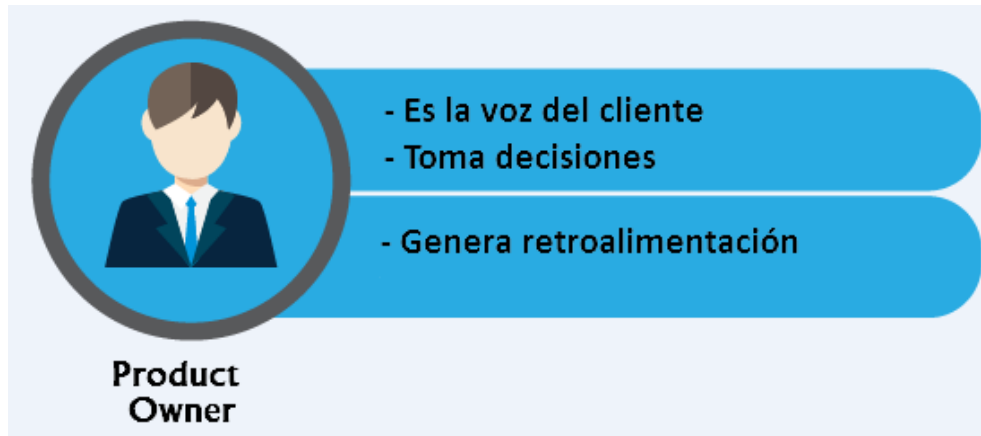


Figura 19: Product Owner (Elaboración propia)

El Product Owner se focaliza en maximizar la rentabilidad del producto y para ello hace uso de la priorización, de esta manera puede reordenar la cola de trabajo del equipo de desarrollo a cargo para que éste construya con anticipación las características o funcionalidades más solicitadas o requeridas.

Otra responsabilidad importante del Product Owner es la gestión de las expectativas de los stakeholders (interesados) mediante la comprensión completa de los problemas que posee el negocio y su descomposición hasta llegar al nivel de requerimientos funcionales.

B. Equipo de Desarrollo (Scrum Team)

El Scrum Team o el equipo de desarrollo está formado como indica Alaimo (2013, p.27) por todas las personas necesarias para la construcción del producto y es el único responsable por la construcción y calidad del producto terminado. El equipo de desarrollo es en todo momento auto-organizado, esto quiere decir que no existe un líder externo que asigne las tareas ni que determine la forma en la que serán resueltos los problemas que se presenten. El equipo de desarrollo se encarga de determinar la forma en que se realizará el trabajo y cómo resolverá cada problema que se pueda presentar.

La contención de esta auto-organización está dada por el objetivo a cumplir que es producir un incremento funcional potencialmente entregable que hace referencia a la transformación de las funcionalidades en software ejecutable y brindando la calidad respectiva. Se visualiza este rol en la figura 20.

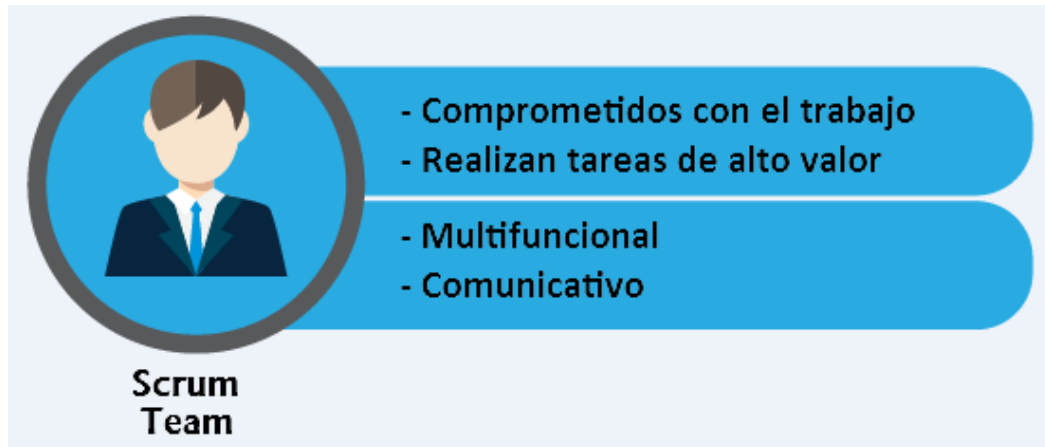


Figura 20: Equipo de desarrollo o Scrum Team (Elaboración propia)

Alaimo (2013, p.28) recomienda que un equipo de desarrollo se componga de 6 a 9 personas cuando se trata de proyectos que posean un nivel de complejidad alto. Cada una de las personas debe poseer todas las habilidades necesarias para realizar el trabajo requerido, esta característica se conoce como multi-funcionalidad y significa que dentro del equipo de desarrollo no existen especialistas exclusivos para una cierta función, sino más bien personas generalistas con capacidades especiales capaces de colaborar en distintos ambientes o funciones.

Lo que se espera de un miembro de un equipo de desarrollo es que no solo realice las tareas en las cuales se especializa sino también todo lo que esté a su alcance para colaborar con el éxito del equipo.

El equipo de desarrollo tiene tres responsabilidades fundamentales:

- ✓ Proveer las estimaciones de cuánto esfuerzo será requerido para construir cada una de las características del producto.
- ✓ Comprometerse al comienzo de cada Sprint a construir un conjunto de características en el tiempo designado.
- ✓ Responsable por la entrega del producto terminado al finalizar cada Sprint.

C. ScrumMaster

El ScrumMaster es el Coach del equipo, y es quien lo ayuda a alcanzar su máximo nivel de productividad posible. Además Herranz, R. (2016, p.21) como se aprecia en la figura 21, indica que sus principales características del ScrumMaster son:

- Debe entender la mentalidad ágil y comportarse de acuerdo a tal, mostrando el enfoque y compromiso adecuado.
- Debe conocer y entender los roles y las responsabilidades que demanda Scrum.
- Debe ser capaz de identificar las habilidades y técnicas que posee el equipo y emplearlas de la mejor manera en las tareas definidas.

- Debe tener la capacidad de influir en el proyecto, pero sin cometer el error de dar órdenes.

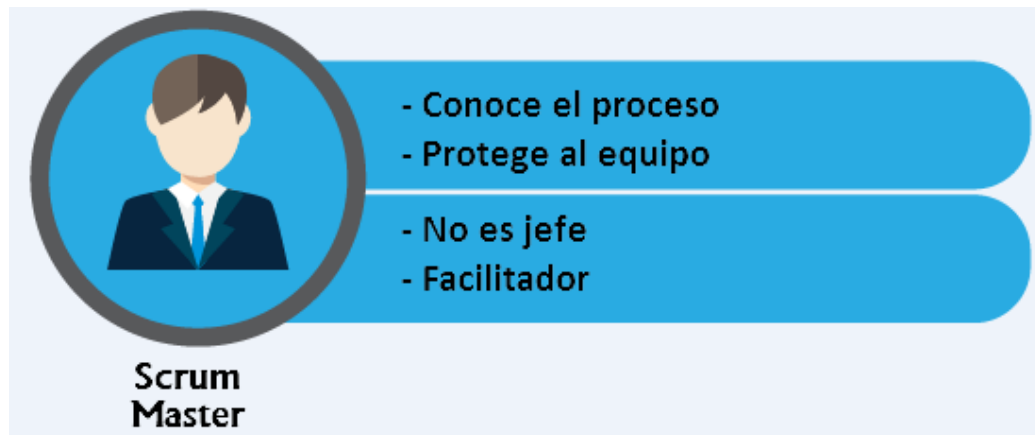


Figura 21: ScrumMaster (Elaboración propia).

Además de estas características, el ScrumMaster debe detectar problemas y/o conflictos que pudiese presentarse dentro del equipo de trabajo, esto es para respetar la filosofía auto-organizativa del equipo, lo ideal es que el equipo mismo sea quien resuelva estos problemas y en el caso de no poder resolverlos, deberá involucrarse al ScrumMaster.

4.1.4. Artefactos de Scrum

El proceso de Scrum posee una cantidad necesaria de elementos formales para llevar adelante un proyecto de desarrollo, Alaimo (2013, p.33) indica que son los siguientes: Product Backlog (Pila de producto), Sprint backlog (Pila de sprint) e incremento funcional potencialmente entregable.

A. Product Backlog (Pila de producto)

El primero de los elementos, y principal de Scrum, es el “Backlog del Producto” o también conocido como “Pila del Producto” o “Product Backlog”, se pueden apreciar en la figura 22.

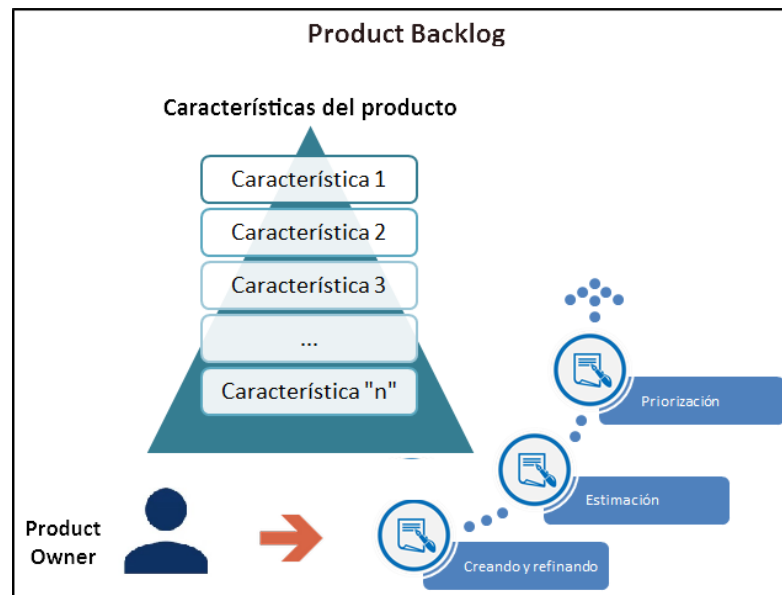


Figura 22: Product Backlog (Elaboración propia)

Según Kniberg (2007, p.17), el Product Backlog o pila de producto es un listado de ítems o características de un producto que se desea construir, que debe ser mantenido y priorizado por el Product Owner. Es importante que exista una clara priorización, ya que es esta priorización la que determinará el orden en el que el equipo de desarrollo transformará las características (ítems) definidas en un producto funcional terminado.

Esta prioridad es responsabilidad únicamente del Product Owner y, aunque el equipo de desarrollo es libre de realizar sugerencias o recomendaciones, es el Product Owner quien decide sobre la prioridad final de los ítems del Product Backlog, teniendo en cuenta el contexto de negocio, el producto a construir y el mercado en el que se encuentra.

B. Sprint Backlog (Pila de Sprint)

El Sprint Backlog es un subconjunto de ítems especificados en el Product Backlog y es el que proporciona el incremento del producto. Debido a que el Product backlog está organizado por prioridad, el Sprint backlog es construido con los requerimientos más prioritarios del Product backlog y con aquellos que quedaron pendientes por desarrollar en el Sprint anterior.

Navarro A., Fernández J. y Morales J. (2013, p.5) mencionan que una vez construido, el Sprint backlog debe ser aceptado por el equipo de desarrollo, pertenece a éste y solo puede ser modificado por el equipo.

Los requerimientos adicionales deben ser incluidos en el Product backlog y desarrollados en el siguiente Sprint, si su prioridad así lo especifica.

C. Incremento funcional potencialmente entregable

Navarro A., Fernández J. y Morales J. (2013, p.5) mencionan que el incremento es el resultado obtenido de la suma de todos los ítems culminados especificados en el Sprint backlog. Si hay ítems incompletos, estos deben ser devueltos al Product backlog con una prioridad alta para que sean incluidos en el siguiente Sprint. Se considera que un ítem está terminado siempre y cuando sea funcional. Finalmente, la suma de ítems culminados especificados en el Product Backlog es el producto a entregar. Se muestra el proceso de la transformación de los ítems hasta la obtención del producto terminado en la figura 23.

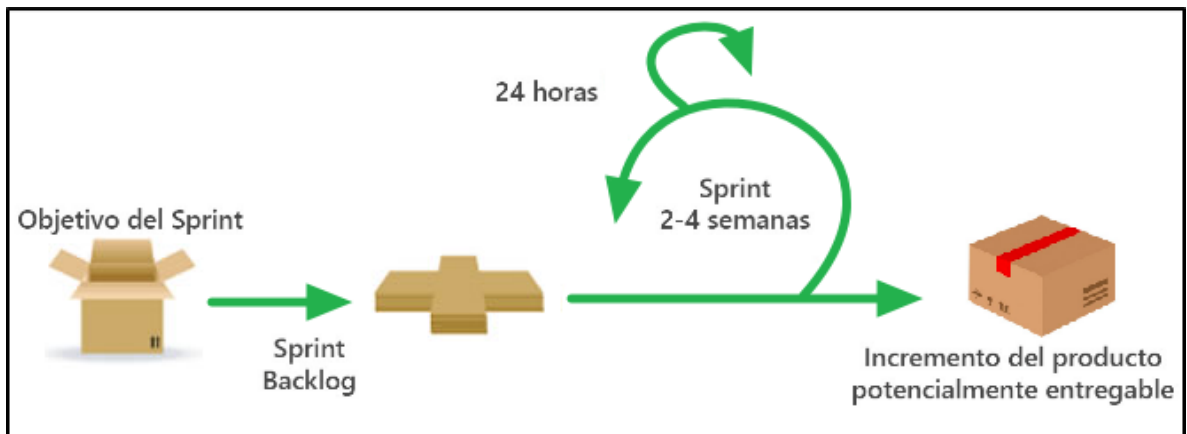


Figura 23: Incremento Funcional (Elaboración propia)

4.1.5. Actividades y eventos SCRUM

Las actividades y eventos que se presentan en la metodología Scrum son los siguientes: Sprint, Sprint Planning Meeting (Planificación de Sprint), Scrum diario, Revisión de Sprint y Retrospectiva.

A. Sprint (Iteración)

Las iteraciones llevan por nombre Sprints. Alaimo (2013, p.42) da a entender que Scrum, como todos los enfoques ágiles, es un proceso de desarrollo incremental e iterativo, esto significa que el producto se construye en incrementos funcionales entregados en periodos cortos para obtener retroalimentación con frecuencia.

Se recomienda en Scrum que la duración de un Sprint se realice entre 1 y 4 semanas, siendo 2 o 3 semanas lo más habitual que encontraremos en la industria afirma Alaimo (2013, p.42). Una de las decisiones que debemos tomar al comenzar un proyecto o al adoptar Scrum es justamente la duración de los Sprints.

Después de ello, el objetivo es mantener esta duración constante a lo largo del desarrollo del producto, lo que implicará que la duración de una iteración no cambie una vez que sea definida.

B. Sprint Planning Meeting (Planificación de Sprint)

Al comienzo de cada Sprint se realiza una reunión de planificación del Sprint donde se definirán los acuerdos y compromisos entre el equipo de desarrollo y el Product Owner sobre el alcance del Sprint.

Según Kniberg (2007, p.22), una planificación de Sprint produce, concretamente:

- Una meta de Sprint.
- Una lista de miembros (y su nivel de dedicación, si no es del 100%).
- Una Pila de Sprint (lista de historias incluidas en el Sprint).
- Una fecha para la presentación del producto obtenido en el Sprint.
- Un lugar y momento definidos para el Scrum Diario.

C. Scrum Diario

Uno de los beneficios de Scrum según Alaimo (2013, p.47) está dado por el incremento de la comunicación dentro del equipo scrum ya que esto facilita la coordinación de acciones entre los miembros del equipo de desarrollo y el conocimiento entre el equipo con las actividades que realizan. Por otro lado, se requiere que se conozcan los compromisos asumidos entre los miembros del equipo de desarrollo y dar visibilidad a los impedimentos que puedan surgir a lo largo del trabajo que se está realizando y que muchas veces es impedimento para lograr los objetivos.

Alaimo (2013, p.48) menciona tres objetivos:

- Incrementar la comunicación.
- Tener claro los compromisos.
- Reconocer los impedimentos, se logra mediante las reuniones diarias de Scrum (Daily Scrums).

Estas reuniones tienen una frecuencia diaria y no deberían durar más de 15 minutos. Estos 15 minutos no se pueden superar.

A la reunión diaria de scrum acude el ScrumMaster y el equipo de desarrollo, en el caso de que sea necesario, se podrá requerir la presencia del Product Owner y de los stakeholders (interesados). Pero de todas maneras es una reunión abierta donde cualquier interesado puede participar como observador, tal como se muestra en la figura 24.

Es importante destacar que en ningún momento se trata de solicitar algún reporte de avance o status al ScrumMaster ni a otras personas involucradas, por el contrario, es un espacio de estricta comunicación solo entre los miembros del equipo de desarrollo.

El objetivo de la primera pregunta (¿qué hice...?) es verificar el cumplimiento de los compromisos contraídos por los miembros del equipo en función del cumplimiento del objetivo del Sprint.

La finalidad de la segunda pregunta (¿qué voy a hacer...?) es generar nuevos compromisos hacia el futuro.

La última pregunta (¿qué problemas...?) apunta a detectar y verificar los impedimentos presentados.

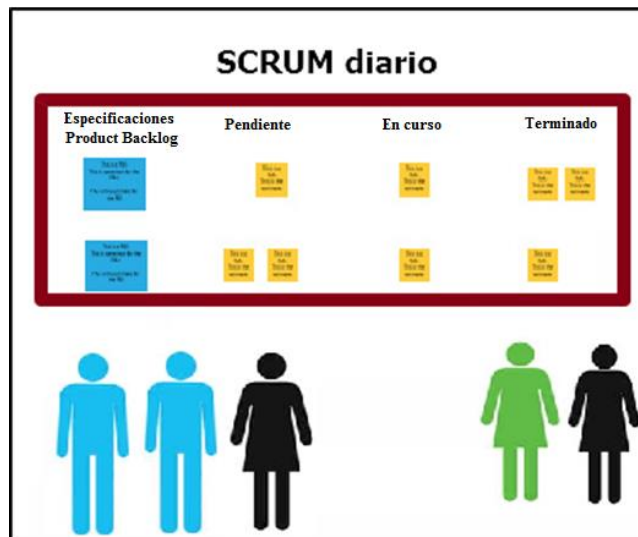


Figura 24: Scrum Diario. (Elaboración propia)

Es responsabilidad del ScrumMaster que se resuelvan lo antes posible, generando las reuniones que sean necesarias e involucrando a las personas requeridas.

D. Revisión de Sprint

Navarro A., Fernández J. y Morales J. (2013, p.4) indican que la revisión del sprint se realiza al final del Sprint y su duración es de un aproximado de cuatro horas para un proyecto de un mes (o una proporción de ese tiempo si la duración es menor). En esta actividad, el dueño del producto revisa lo que se desarrolló, identifica lo que no se desarrolló y discute acerca del Product Backlog, así mismo el equipo de desarrollo indica los problemas que se presentaron y la manera en que fueron resueltos, y muestra el funcionamiento del producto.

Esta reunión es de gran importancia para los Sprints que se desarrollarán después. La revisión del sprint se presenta en la figura 25.

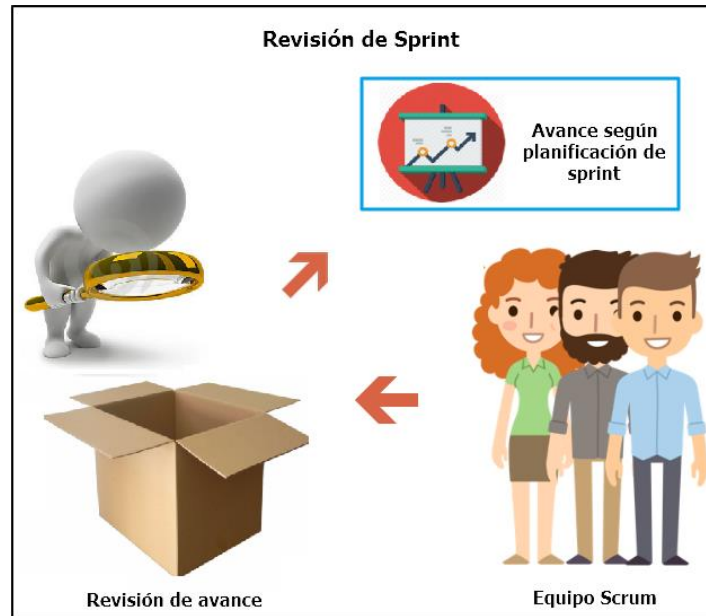


Figura 25: Revisión del Sprint. (Elaboración propia)

E. Retrospectiva

Según Navarro A., Fernández J. y Morales J. (2013, p.5) el tiempo aproximado de una reunión de retrospectiva es de tres horas, donde el equipo Scrum analiza cómo fue la comunicación, el proceso y las herramientas, claramente indican que estuvo bien, qué no, y se crea un plan de mejoras para el siguiente Sprint. El tiempo, se debe ajustar de manera proporcional en el caso de proyectos de duración menor a un mes.

Además, la retrospección del equipo de desarrollo da origen a la mejora continua. Mediante la retrospectiva el equipo puede reflexionar sobre la forma en que se trabajó y los acontecimientos que sucedieron en el Sprint que acaba de culminar.

4.1.6. Ventajas:

Según Florez & Grisales (2014, p.30), las principales ventajas que ofrece SCRUM son las siguientes:

- El cliente está satisfecho ya que recibe lo que necesita y esperaba.
- El coste en términos de proceso y gestión es mínimo, llevando a un resultado más rápido y barato.
- Ayuda a la empresa a ahorrar tiempo y dinero.
- Permite realizar proyectos en los que la documentación de los requerimientos de negocios no está muy claro como para ser desarrollada.
- Se desarrolla rápidamente y testea. Cualquier error puede ser fácilmente rectificado.
- Los problemas se identifican por adelantado en las reuniones diarias y por lo tanto se pueden resolver rápidamente.

- Hay visibilidad clara del desarrollo del proceso.
- Iterativo en su naturaleza, requiere continuo feedback del usuario.
- Fácil de manejar los cambios debido a los sprints tan cortos y el feedback constante.
- Poco control que insiste en la información frecuente del proceso en el trabajo mediante reuniones regulares.
- Las reuniones diarias hacen posible medir la productividad individual. Esto lleva a la mejor en la productividad de cada uno de los miembros del equipo.
- Para los desarrolladores las ventajas pueden ser motivación y satisfacción de realizar el trabajo de una manera eficiente.
- Es fácil entregar un producto de calidad en el tiempo estipulado.
- Puede trabajar con cualquier tecnología o lenguaje de programación.
- Hace el proceso del desarrollo de software más centrado y manejable.

4.2. Adaptación de la Metodología

El presente trabajo de investigación presenta la siguiente adaptación de la metodología la cual está constituida por 3 fases definidas por Scrum, que son las siguientes: Pre-juego, Juego y Post-juego. Tal como se muestra en la figura 26.

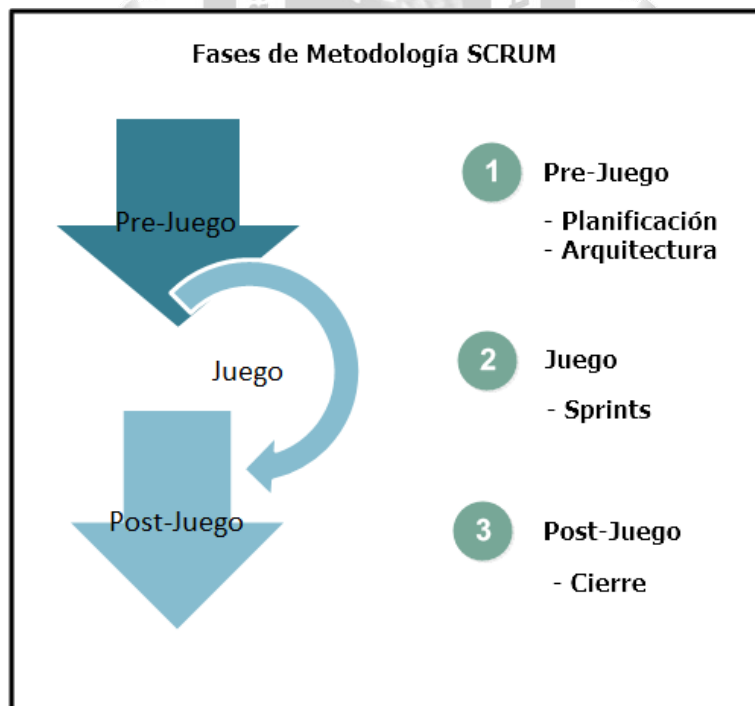


Figura 26: Fases de Metodología Scrum (Elaboración propia).

4.2.1. Fase I: Pre-juego

Según Caso (2014, p.2) se compone de dos etapas: Planificación y Arquitectura. Las cuales poseen la información referente a la planificación para el desarrollo de la solución tecnológica, el modo en cómo se desarrolla el mismo y a la arquitectura de la aplicación planteada como solución tecnológica.

Planificación. Dentro de la etapa de planificación se realizarán las siguientes actividades:

- Elaboración de la visión.
- Elaboración del presupuesto.
- Identificación del responsable del financiamiento.
- Definición de las herramientas de desarrollo.
- Definición del equipo de trabajo.
- Elaboración del backlog del producto (pila de producto).
- Planificación de sprints.

Arquitectura. Dentro de la etapa de arquitectura se realizarán las siguientes actividades:

- Elaboración del diseño de la arquitectura de la solución tecnológica.
- Elaboración del diseño de la arquitectura funcional de la solución tecnológica en base a los ítems del backlog.

4.2.2. Fase II: Juego

En esta fase se realiza el desarrollo de la aplicación. Esta etapa se divide en iteraciones que proveen como resultado funcionalidades incrementales al finalizar de cada una de ellas. El propósito es implementar un sistema listo para entrega en una serie de sprints. Cada sprint incluye las fases tradicionales del desarrollo de software: requerimientos, análisis, diseño, desarrollo y entrega, con esto el sistema va evolucionando a través de múltiples iteraciones.

Las actividades a realizar son:

- Sprint.
- Elaboración de historias de usuarios por sprint

4.2.3. Fase III: Post-juego

Esta fase comienza es cuando los requerimientos se han completado. Dando lugar así a la debida preparación para el lanzamiento de la versión, incluyendo la documentación final y pruebas antes del lanzamiento de la misma.

Las actividades a realizar son:

- Despliegue de la aplicación al servidor EsHost.
- Capacitación al personal.

CAPÍTULO V: SOLUCIÓN TECNOLÓGICA

En el presente capítulo teniendo en cuenta el modelo aplicativo del trabajo de investigación, se presenta la propuesta de solución sobre la aplicación web, para que mejore la gestión de la atención de los requerimientos al servicio de fábrica brindado por FabSF Services, esta propuesta de solución se desarrolló tomando en cuenta las fases de la metodología Scrum, mencionada en el capítulo IV.

5.1. Fase I: Pre-Juego

En la fase pre-juego se desarrolló lo siguiente:

Para la planificación. La visión, el presupuesto, el financiamiento, las herramientas de desarrollo, el equipo de trabajo, el backlog del producto y la fecha de desarrollo de cada sprint.

Para la Arquitectura. El diseño de la implementación de las funcionalidades en relación a las especificaciones del product backlog y el diseño de la arquitectura tecnológica que tiene la solución.

5.1.1. Planeamiento

A. Visión

Se pretende desarrollar una aplicación para la gestión de requerimientos que ayude en el control y seguimiento de los mismos para optimizar el proceso de atención en la fábrica de software FabSF Services en base a la aprobación o rechazo de los requerimientos solicitados que son analizados por el personal de fábrica indicando así si es viable el desarrollo y atención de los mismos.

B. Presupuesto

El costo de la aplicación es de S/. 18,940 y el detalle de los costos calculados se muestra a continuación.

En la tabla 2 se muestran los costos del personal:

CARGO	CANTIDAD	TIEMPO (MES)	COSTO (MES)	TOTAL
Ingeniero de Software	1	3	S/3,800	S/11,400

Tabla 2. Costos de Personal (Elaboración propia)

En la tabla 3 se muestran los costos de suministros:

CANTIDAD	DESCRIPCIÓN	COSTO UNITARIO	TOTAL
1	Millar de Papel A4 75 gr.	S/28	S/28
2	Cartuchos de Tinta para impresión	S/80	S/160
1	Impresora Multifuncional HP	S/370	S/370

	DeskJet GT 5820		
1	Dispositivo USB 16 GB	S/32	S/32
TOTAL			S/590

Tabla 3. Costos de Suministros (Elaboración propia)

En la tabla 4 se muestran los costos de capacitación de usuarios:

CANTIDAD	DESCRIPCIÓN	COSTO UNITARIO	TOTAL
5	Capacitación de Usuarios	S/250	S/1,250

Tabla 4. Costos de Capacitación de Usuarios (Elaboración propia)

En la tabla 5 se muestra el resumen de costos:

DESCRIPCIÓN	TOTAL
Costo de Programa de Titulación	S/5,700
Costos de Personal	S/11,400
Costos de Software	S/0
Costos de Suministros	S/590
Costos de Capacitación de Usuarios	S/1,250
TOTAL	S/18,940

Tabla 5. Resumen de Costos (Elaboración propia)

C. Financiamiento

El costo de la aplicación es de S/18,940 el mismo que no será financiado por la FabSF Services.

D. Herramientas de desarrollo

La aplicación web para la gestión de requerimientos es una aplicación web desarrollada en el lenguaje de programación PHP. Para su desarrollo se hizo uso de las siguientes herramientas:

- **Netbeans 8.2:** Es un entorno de desarrollo de código abierto que ayuda a crear diferentes tipos de aplicaciones de software. Principalmente se enfoca en proyectos Java, sin embargo, se pueden crear aplicaciones de software en distintos lenguajes como C/C++, PHP, javascript haciendo uso de plantillas y hojas de estilo CSS3. Wielenga (2015, p.3).
- **MySQL 5.7.17:** Es un sistema de administración de base de datos relacional (RDBMS). Es capaz de almacenar una enorme cantidad de datos de distintos tipos y distribuirlos para satisfacer las necesidades de las organizaciones que lo empleen. MySQL compete con sistemas propietarios como Oracle, SQL Server y DB2. Utiliza un lenguaje de consulta

estructurado (SQL) lo cual permite agregar, manipular y recuperar datos en función de criterios específicos. Gilfillan (2003, p.40).

- **PHP 7.2:** HyperText Preprocessor es un lenguaje interpretado libre, usado originalmente solo para el desarrollo de aplicaciones que actuarán del lado del servidor, capaces de generar contenido dinámico en la World Wide Web. PHP es utilizado en aplicaciones como Facebook, Drupal, Joomla, WordPress entre otros. Es un software gratuito y de código abierto. Arias (2015, p.13).
- **CodeIgniter 3.1.9:** Es un framework de PHP que está diseñado para el desarrollo de aplicaciones web empleando herramientas simples y organizadas. Posee un conjunto de librerías que se emplean comúnmente en desarrollo de aplicaciones minimizando la carga de código que pueda tener una tarea determinada del proyecto.

E. Equipo de Trabajo

Para el desarrollo de la aplicación web se formaron los roles mostrados en la tabla 6, la cual se presenta a continuación:

PERSONA	ROL	DESCRIPCIÓN
Alex Nahuel Guerrero Córdoba	Scrum Master	Gestor de proyecto
Rodrigo Juan Palacios Mendiola	Product Owner	Sub-Gerente FabSF Services
Erick Franz Nolasco Ríos	Scrum Team	Tesista

Tabla 6. Roles para el desarrollo del proyecto (Elaboración propia)

F. Product Backlog o Pila de Producto

En la siguiente tabla 7 se detalla la pila del producto (product backlog), que es una lista de requerimientos definidos desde la primera reunión que se realizó en la fábrica de software de FabSF Services con el sub-gerente, el cual indicó que se necesitarían algunos ítems para la construcción de una aplicación web de gestión de requerimientos.

Código	Historia de Usuario	Estimación (Horas)
HU1	Login	7
HU2	Registrar Usuarios	5
HU3	Modificar Usuarios	5
HU4	Eliminar Usuarios	5
HU5	Registrar personal	5
HU6	Modificar personal	5

HU7	Eliminar personal	5
HU8	Registrar lenguajes de programación	5
HU9	Modificar lenguajes de programación	5
HU10	Eliminar lenguajes de programación	5
HU11	Registrar prioridad de requerimientos	5
HU12	Modificar prioridad de requerimientos	5
HU13	Eliminar prioridad de requerimientos	5
HU14	Registrar clientes	5
HU15	Modificar clientes	5
HU16	Eliminar clientes	5
HU17	Registrar requerimientos	5
HU18	Modificar requerimientos	5
HU19	Eliminar requerimientos	5
HU20	Evaluar requerimientos	5
HU21	Aprobar requerimientos	18
HU22	Rechazar requerimientos	18
HU23	Consultar requerimientos	8
HU24	Generar Reporte general de requerimientos	8

Tabla 7. Pila del Producto (Elaboración propia)

G. Planificación de Sprints

El proyecto está dividido en 5 sprint o iteraciones los mismos que serán desarrollados en días laborables de lunes a viernes. En la siguiente tabla 8 se detalla la información:

CÓDIGO	HISTORIA DE USUARIO	SPRINT	RESPONSABLE	FECHA INICIO	FECHA FIN
HU1	Login	Sprint 1	Erick Franz Nolasco Ríos	10/06/2018	11/06/2018
HU2	Registrar Usuarios			11/06/2018	12/06/2018
HU3	Modificar Usuarios			13/06/2018	13/06/2018
HU4	Eliminar Usuarios			14/06/2018	15/06/2018
HU5	Registrar personal	Sprint 2		16/06/2018	17/06/2018
HU6	Modificar personal			18/06/2018	19/06/2018
HU7	Eliminar personal			20/06/2018	21/06/2018
HU8	Registrar lenguajes de programación			22/06/2018	23/06/2018
HU9	Modificar lenguajes de programación	Sprint 3		25/06/2018	26/06/2018
HU10	Eliminar lenguajes de programación			27/06/2018	28/06/2018
HU11	Registrar prioridad de requerimientos			29/06/2018	29/06/2018
HU12	Modificar prioridad de requerimientos			29/06/2018	29/06/2018
HU13	Eliminar prioridad de requerimientos	Sprint 4		29/06/2018	29/06/2018
HU14	Registrar clientes			03/07/2018	04/07/2018
HU15	Modificar clientes			05/07/2018	06/07/2018
HU16	Eliminar clientes	Sprint 5		07/07/2018	08/07/2018
HU17	Registrar requerimientos			09/07/2018	10/07/2018
HU18	Modificar requerimientos			11/07/2018	12/07/2018
HU19	Eliminar requerimientos	Sprint 4		13/07/2018	14/07/2018
HU20	Evaluar requerimientos			15/07/2018	16/07/2018
HU21	Aprobar requerimientos			17/07/2018	20/07/2018
HU22	Rechazar requerimientos	Sprint 5		21/07/2018	24/07/2018
HU23	Consultar requerimientos			25/07/2018	26/07/2018
HU24	Generar Reporte general de requerimientos	Sprint 5		26/07/2018	27/07/2018

Tabla 8. Programación de Sprints (Elaboración propia)

5.1.2. Arquitectura

A. Arquitectura Tecnológica de la Solución

Se muestra la arquitectura tecnológica de la solución planteada, se puede visualizar en la figura 27 como está constituida la arquitectura en capas y el número de capas que posee la aplicación web.

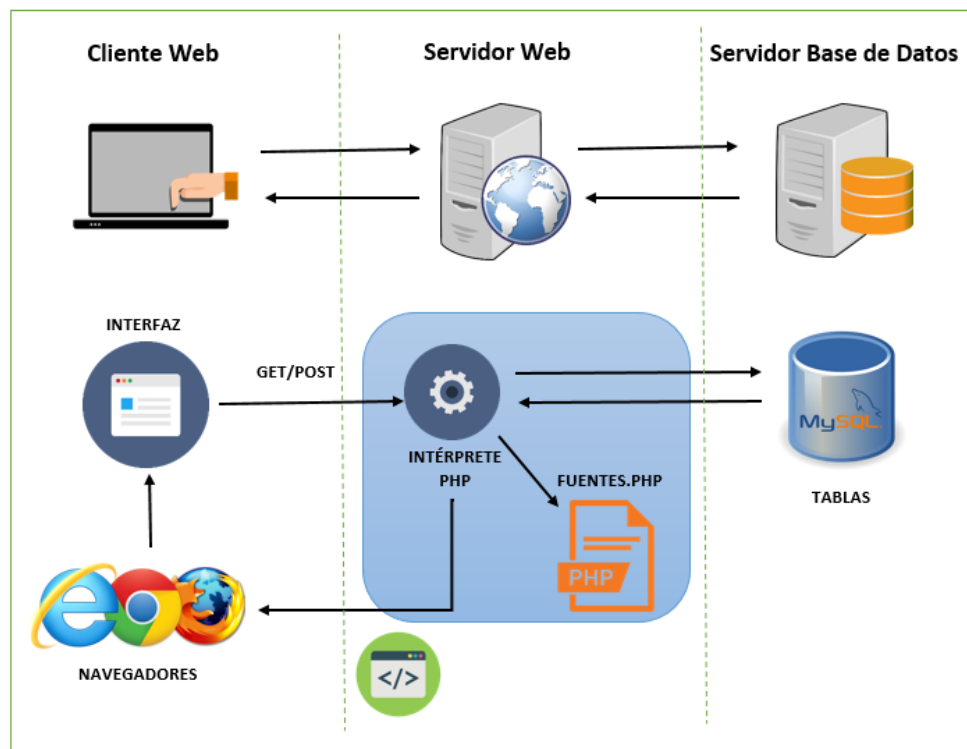


Figura 27. Arquitectura Tecnológica de la solución (Elaboración propia)

A continuación, se describirán las tres capas lógicas que se muestran en la figura:

- **Cliente web:** Esta capa está formada por la lógica de la aplicación a la que el usuario final tiene acceso mediante una interfaz de usuario. La lógica de la capa de cliente incluye a los navegadores como Internet Explorer, Google Chrome, Mozilla Firefox, etc.
- **Servidor web:** Esta capa está compuesta por la lógica que realiza las funciones principales de la aplicación, ya sea: procesamiento de datos, implementación de funciones de negocios y administración de recursos como la base de datos. En esta capa se procesan las solicitudes enviadas desde la capa de cliente. La lógica en esta capa está formada por el componente Hypertext Preprocessor (PHP).
- **Servidor Base de Datos:** La Capa de Datos está compuesta por los servicios que brinda el gestor de base de datos. Los datos son almacenados en un sistema de administración de bases de datos que en este caso es MySQL.

B. Arquitectura Funcional de la Solución Tecnológica

La aplicación web para la gestión de requerimientos se divide en paquetes basándose en los ítems especificados en el backlog los cuales conforman la arquitectura funcional de la solución tecnológica planteada.

En la siguiente figura se muestra la arquitectura funcional de la solución y se puede visualizar que posee 5 módulos. Cada uno de estos módulos está constituido por una serie de requerimientos funcionales que representan funcionalidades de la aplicación. Se presenta la arquitectura mencionada en la figura 28.

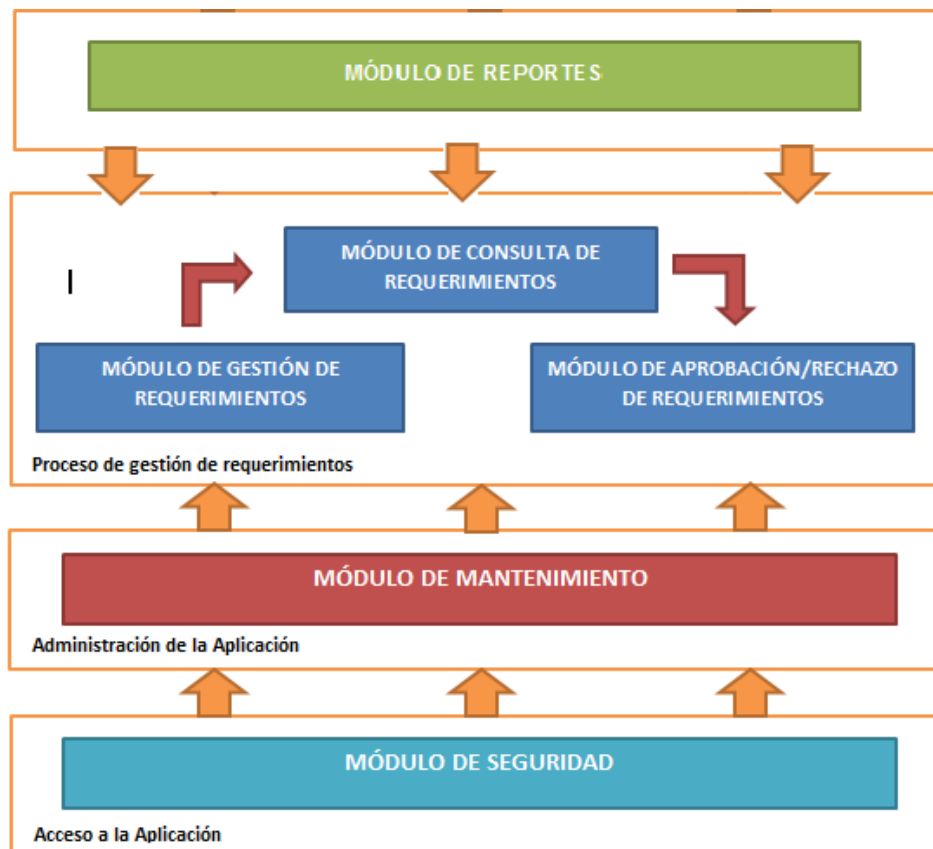


Figura 28. Arquitectura Funcional de la solución tecnológica (Elaboración propia)

- **Módulo de seguridad:** Controla el acceso de usuarios registrados a la aplicación web.
- **Módulo de mantenimiento:** Gestiona información de clientes, personal, lenguajes de programación y prioridades.
- **Módulo de gestión de requerimiento:** Gestiona la información de los requerimientos solicitados a la fábrica de software.
- **Módulo de consulta de requerimientos:** Realiza consultas de información de los requerimientos.
- **Módulo de aprobación/rechazo de requerimientos:** Permite la aprobación/rechazo de los requerimientos registrados en la fábrica de software.
- **Módulo de reportes:** Genera reporte por cada perfil (Líder de proyecto/Ing. De software/Analista).

5.2. Fase II: Juego

En fase contempla el desarrollo de los sprints que fueron planificados en la fase I. Cada sprint contiene historias de usuario y cada una de las mismas está relacionada con un conjunto de tareas las cuales deben realizarse para construir la funcionalidad especificada.

5.2.1. Sprint 1

A. Historia de Usuario “Login”

En la Tabla 9 se describe a la Historia de Usuario “Login”.

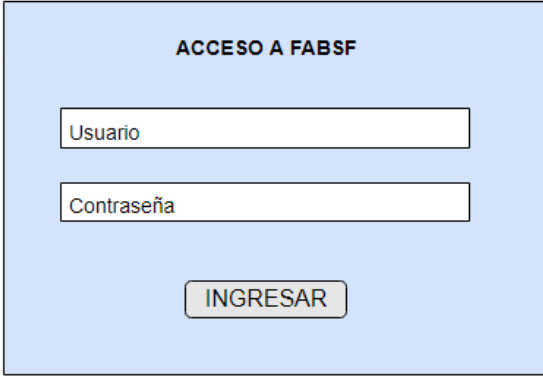
Historia de Usuario “Login”	
Código: HU1	Enunciado: “Como usuario necesito un formulario de acceso y gracias a esto acceder a la aplicación web ”.
Sprint: 1	
Descripción: El usuario debe completar el formulario de acceso al sistema ingresando su usuario y contraseña, presiona el botón “Ingresar” y finalmente ingresa a la aplicación.	
Criterios de aceptación: <ol style="list-style-type: none">1. El usuario debe estar registrado en la base de datos para ingresar a la aplicación.2. Si el usuario no existe en la base de datos mostrar mensaje “Usuario o Contraseña no válidos”.3. Si el usuario y/o la contraseña es incorrecta, se muestra mensaje “Usuario o Contraseña no válidos”.4. Cuando se finalice la sesión se mostrará el siguiente mensaje: “La sesión finalizó correctamente.”	
Prototipo: Login. 	
Observaciones: <ul style="list-style-type: none">- Queda desestimada la opción “Recuperar contraseña”.- Prototipo creado correctamente.	

Tabla 9. Historia de Usuario “Login” (Elaboración propia)

B. Historia de Usuario “Registrar Usuarios”

En la Tabla 10 se describe a la Historia de Usuario “Registrar Usuarios”.

Historia de Usuario “Registrar Usuarios”	
Código: HU2	Enunciado: “Como supervisor de fábrica necesito registrar usuarios y gracias a esto controlar la información de los usuarios que se registrarán en la aplicación”
Sprint: 1	
Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú principal “Gestión de Usuarios” y el sistema muestra la información de los usuarios registrados en un listado. Cuando se requiera crear un usuario, se presiona el botón “Crear nuevo Usuario” y el sistema muestra un formulario para ingresar la información relacionada al mismo. Luego, el supervisor de fábrica ingresa los datos del usuario a registrar: - Código de personal* - Usuario - Contraseña *Se listan los nombres del personal que aún no tienen un usuario registrado. Luego presiona el botón “Registrar” para confirmar el registro del nuevo usuario. Para limpiar los campos presionar el botón “Limpiar”.	
Criterios de aceptación: <ol style="list-style-type: none">1. Se debe validar que, si el usuario ya existe en la base de datos, no permita registrar el mismo y muestre mensaje “Actualmente el usuario ya está registrado, volver a intentar.”.2. Para registrar un usuario nuevo, el personal debe estar registrado en la base de datos.3. Para registrar un usuario nuevo debe haber personal sin usuario registrado.	
Prototipo:	

GESTIÓN DE USUARIOS

Filtro por nombre:

Cod. Personal	Usuario	Tipo Usuario	Fec. Creación	Acción	
Cell 1	Cell 1	Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 4	Cell 4	Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 7	Cell 7	Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 10	Cell 10	Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>

[Anterior 1,2,3](#) [Siguiete](#)

GESTIÓN DE USUARIOS NUEVO USUARIO

NOM. PERSONAL: ▼

USUARIO:

CONTRASEÑA:

Observaciones:

- Para regresar a la página anterior debe desarrollarse un botón “Regresar”.
- Se añadió botón “Regresar” en los prototipos.

Tabla 10. Historia de Usuario “Registrar Usuarios” (Elaboración propia)

C. Historia de Usuario “Modificar Usuarios”

En la Tabla 11 se describe a la Historia de Usuario “Modificar Usuarios”.

Historia de Usuario “Modificar Usuarios”	
Código: HU3	Enunciado: “Como supervisor de fábrica necesito modificar usuarios y gracias a esto controlar la información de los usuarios registrados en la aplicación ”
Sprint: 1	
Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú principal “Gestionar Usuarios” y el sistema muestra la información de los usuarios registrados en un listado. Cuando se requiera modificar los datos de un usuario, el supervisor de fábrica identifica al usuario en la lista, presiona el botón “Editar” y el sistema muestra los nombres y los campos para modificar sus datos: - Usuario - Contraseña Luego, el supervisor de fábrica modifica los datos, presiona el botón “Editar” para confirmar la modificación de los datos del usuario. Para limpiar los campos presionar el botón “Limpiar”. Para regresar a la página anterior presionar el botón “Regresar”.	
Criterios de aceptación: <ol style="list-style-type: none">1. El administrador debe modificar los usuarios que existan en la base de datos de la aplicación.2. Para modificar datos de un usuario, el personal relacionado al usuario debe estar registrado en la base de datos.3. Se modificará el usuario siempre y cuando no exista o este registrado en la base de datos.	
Prototipo:	

GESTIÓN DE USUARIOS

Filtro por nombre:

Cod. Personal	Usuario	Tipo Usuario	Fec. Creación	Acción	
Cell 1	Cell 1	Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 4	Cell 4	Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 7	Cell 7	Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 10	Cell 10	Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>

[Anterior 1,2,3](#) [Siguiete](#)

GESTIÓN DE USUARIOS

EDITAR USUARIO

NOM. PERSONAL:

USUARIO:

CONTRASEÑA:

Observaciones:

- Para regresar a la página anterior debe desarrollarse un botón “Regresar”.
- Se añadió botón “Regresar” en los prototipos.

Tabla 11. Historia de Usuario “Modificar Usuarios” (Elaboración propia)

D. Historia de Usuario “Eliminar Usuarios”

En la Tabla 12 se describe a la Historia de Usuario “Eliminar Usuarios”.


Historia de Usuario “Eliminar Usuarios”	
Código: HU4	Enunciado: “Como supervisor de fábrica necesito eliminar usuarios y gracias a esto borrar la información de los usuarios registrados en la aplicación”
Sprint: 1	
Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú principal “Gestionar Usuarios” y el sistema muestra la información de los usuarios registrados en un listado. Cuando se requiera eliminar un usuario, el supervisor de fábrica identifica al usuario en la lista y presiona el botón “Borrar”.	
Criterios de aceptación: <ol style="list-style-type: none">1. El administrador debe eliminar los usuarios que existan en la base de datos de la aplicación.2. Para eliminar un usuario, el personal debe estar registrado en la base de datos.3. Para eliminar un usuario debe haber personal con usuario registrado.	
Prototipo: <div style="text-align: center;"></div>	
Observaciones: <ul style="list-style-type: none">- No se requiere mensaje de confirmación para eliminar registro.	

Tabla 12. Historia de Usuario “Eliminar Usuarios” (Elaboración propia)

5.2.2. Sprint 2

A. Historia de Usuario “Registrar personal”

En la Tabla 13 se describe a la Historia de Usuario “Registrar personal”.

Historia de Usuario “Registrar personal”	
Código: HU5	Enunciado: “Como supervisor de fábrica necesito registrar al personal que será relacionado en los requerimientos y gracias a esto conocer los involucrados en el requerimiento ”.
Sprint: 2	
<p>Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú secundario “Módulo Mantenimiento”, luego seleccionar la opción “Gestionar personal” y el sistema muestra la información del personal registrado en un listado.</p> <p>Cuando se requiera crear un nuevo personal, se presiona el botón “Registrar nuevo personal” y el sistema muestra un formulario para ingresar la información relacionada al mismo. Luego, el supervisor de fábrica ingresa los datos del personal a registrar:</p> <ul style="list-style-type: none">- Primer nombre- Segundo nombre- Apellido paterno- Apellido materno- Género- DNI- Teléfono fijo- Teléfono celular- Correo electrónico- Domicilio- Fecha de Nacimiento- Cargo <p>Luego presiona el botón “Registrar” para confirmar el registro del nuevo estado.</p> <p>Para limpiar los campos presionar el botón “Limpiar”.</p> <p>Para regresar a la página anterior presionar el botón “Regresar”.</p>	
<p>Criterios de aceptación:</p> <ol style="list-style-type: none">1. Se deben ingresar los campos del nuevo personal correctamente.2. Se debe validar que no se ingrese un personal ya existente en la base de datos por medio del DNI, de lo contrario se mostrará el siguiente mensaje: “Actualmente el DNI está registrado, volver a intentar.”3. Se registrará personal si no existe en la base de datos el DNI ingresado.	
<p>Prototipo:</p>	

GESTIÓN DE PERSONAL

Filtro por nombre:

Cod. Personal	Nombre	Apellido	Correo	Cargo	Acción	
Cell 1	Cell 1	Cell 1	Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 4	Cell 4	Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 7	Cell 7	Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 10	Cell 10	Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>

[Anterior 1,2,3](#) [Siguiete](#)

GESTIÓN DE PERSONAL

NUEVO PERSONAL

PRIMER NOMBRE: SEGUNDO NOMBRE:

AP. PATERNO: AP. MATERNO:

GENERO: ▼ DNI:

TELF. FIJO: TELF. CELULAR:

CORREO ELECTRÓNICO: DOMICILIO:

FEC. NACIMIENTO ▼ CARGO: ▼

Observaciones:

- Ninguna.

Tabla 13. Historia de Usuario "Registrar personal" (Elaboración propia)

B. Historia de Usuario “Modificar personal”

En la Tabla 14 se describe a la Historia de Usuario “Modificar personal”.

Historia de Usuario “Modificar personal”																																				
Código: HU6	Enunciado: “Como supervisor de fábrica necesito modificar la información del personal que será relacionado en los requerimientos y gracias a esto tener la información actualizada ”.																																			
Sprint: 2																																				
Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú secundario “Módulo Mantenimiento”, luego seleccionar la opción “Gestionar personal” y el sistema muestra la información del personal registrado en un listado. Cuando se requiera modificar la información del personal de fábrica, el supervisor de fábrica identifica al personal en la lista, presiona el botón “Editar” y el sistema muestra un formulario para modificar sus datos, los cuales son los mismos que solicita al registrar el personal exceptuando el DNI, adicionalmente muestra el código del personal. Luego, el supervisor de fábrica modifica los datos, presiona el botón “Editar” para confirmar la modificación de los datos del personal. Para limpiar los campos presionar el botón “Limpiar”. Para regresar a la página anterior presionar el botón “Regresar”.																																				
Criterios de aceptación: <ol style="list-style-type: none">1. Para modificar un personal debe existir en la base de datos de la aplicación.2. Se deben ingresar la información a modificar correctamente.																																				
Prototipo: <div style="border: 1px solid black; padding: 10px; text-align: center;"><h3>GESTIÓN DE PERSONAL</h3><p>Filtro por nombre: <input type="text" value="Search..."/> <input type="button" value="Buscar"/> <input type="button" value="Actualizar"/> <input type="button" value="Registrar nuevo personal"/></p><table border="1"><thead><tr><th>Cod. Personal</th><th>Nombre</th><th>Apellido</th><th>Correo</th><th>Cargo</th><th colspan="2">Acción</th></tr></thead><tbody><tr><td>Cell 1</td><td>Cell 1</td><td>Cell 1</td><td>Cell 2</td><td>Cell</td><td><input type="button" value="EDITAR"/></td><td><input type="button" value="BORRAR"/></td></tr><tr><td>Cell 1</td><td>Cell 4</td><td>Cell 4</td><td>Cell 5</td><td>Cell6</td><td><input type="button" value="EDITAR"/></td><td><input type="button" value="BORRAR"/></td></tr><tr><td>Cell 1</td><td>Cell 7</td><td>Cell 7</td><td>Cell 8</td><td>Cell9</td><td><input type="button" value="EDITAR"/></td><td><input type="button" value="BORRAR"/></td></tr><tr><td>Cell 1</td><td>Cell 10</td><td>Cell 10</td><td>Cell 11</td><td>Cell 2</td><td><input type="button" value="EDITAR"/></td><td><input type="button" value="BORRAR"/></td></tr></tbody></table><p>Anterior 1,2,3 Siguiente</p><input type="button" value="Regresar"/></div>		Cod. Personal	Nombre	Apellido	Correo	Cargo	Acción		Cell 1	Cell 1	Cell 1	Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>	Cell 1	Cell 4	Cell 4	Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>	Cell 1	Cell 7	Cell 7	Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>	Cell 1	Cell 10	Cell 10	Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cod. Personal	Nombre	Apellido	Correo	Cargo	Acción																															
Cell 1	Cell 1	Cell 1	Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>																														
Cell 1	Cell 4	Cell 4	Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>																														
Cell 1	Cell 7	Cell 7	Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>																														
Cell 1	Cell 10	Cell 10	Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>																														

GESTIÓN DE PERSONAL

EDITAR PERSONAL

PRIMER NOMBRE:	<input type="text"/>	SEGUNDO NOMBRE:	<input type="text"/>
AP. PATERNO:	<input type="text"/>	AP. MATERNO:	<input type="text"/>
GENERO:	<input type="text" value="Options"/>	DNI:	<input type="text"/>
TELF. FIJO:	<input type="text"/>	TELF. CELULAR:	<input type="text"/>
CORREO ELECTRÓNICO:	<input type="text"/>	DOMICILIO:	<input type="text"/>
FEC. NACIMIENTO	<input type="text" value="4/22/2022"/> <input type="text" value="📅"/>	CARGO:	<input type="text" value="Options"/>

Editar
Limpiar

Regresar

Observaciones:

- Cambiar botón “Cancelar” por botón “Regresar”.
- No debe aparecer el campo DNI.

Tabla 14. Historia de Usuario “Modificar personal” (Elaboración propia)

C. Historia de Usuario “Eliminar personal”

En la Tabla 15 se describe a la Historia de Usuario “Eliminar personal”.

Historia de Usuario “Eliminar personal”	
Código: HU7	Enunciado: “Como supervisor de fábrica necesito eliminar al personal que no prestará sus servicios a la fábrica de software y gracias a esto tener la información actualizada del personal ”.
Sprint: 2	
Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú secundario “Módulo Mantenimiento”, luego seleccionar la opción “Gestionar personal” y el sistema muestra la información del personal registrado en un listado. Cuando se requiera eliminar un estado, el supervisor de fábrica identifica al personal en la lista y presiona el botón “Borrar”.	
Criterios de aceptación:	

1. Para eliminar un personal debe existir en la base de datos de la aplicación.

Prototipo:

GESTIÓN DE PERSONAL

Filtro por nombre:

Cod. Personal	Nombre	Apellido	Correo	Cargo	Acción	
Cell 1	Cell 1	Cell 1	Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 4	Cell 4	Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 7	Cell 7	Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 10	Cell 10	Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>

[Anterior 1,2,3](#) [Siguiente](#)

Observaciones:

- No se requiere mostrar mensaje de confirmación para eliminar registro.

Tabla 15. Historia de Usuario “Eliminar personal” (Elaboración propia)

D. Historia de Usuario “Registrar lenguajes de programación”

En la Tabla 16 se describe a la Historia de Usuario “Registrar lenguajes de programación”.

Historia de Usuario “Registrar lenguajes de programación”	
Código: HU8	Enunciado: “Como supervisor de fábrica necesito registrar los lenguajes de programación que puedan tener los requerimientos y gracias a esto conocer el lenguaje de programación con el cual se atenderá el requerimiento”.
Sprint: 2	
Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú secundario “Módulo Mantenimiento”, luego seleccionar la opción “Gestionar lenguajes de programación” y el sistema muestra la información de los lenguajes de programación registrados en un listado. Cuando se requiera crear un lenguaje de programación, se presiona el botón “Registrar nuevo Lenguaje” y el sistema muestra un formulario para ingresar la información relacionada al mismo. Luego, el supervisor de fábrica ingresa los datos del lenguaje de programación a registrar: <ul style="list-style-type: none"> - Lenguaje de Programación 	

Luego presiona el botón “Registrar” para confirmar el registro del nuevo lenguaje de programación.
 Para limpiar los campos presionar el botón “Limpiar”.
 Para regresar a la página anterior presionar el botón “Regresar”.

Criterios de aceptación:

1. Para registrar un lenguaje de programación debe ingresarse correctamente en el formulario.

Prototipo:

GESTIÓN DE LENGUAJE DE PROGRAMACIÓN

Filtro por lenguaje:

Cod. Lenguaje	Lenguaje de Programación	Acción	
Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>

[Anterior](#) [1,2,3](#) [Siguiete](#)

GESTIÓN DE LENGUAJE DE PROGRAMACIÓN

NUEVO LENGUAJE DE PROGRAMACIÓN

LENGUAJE DE PROGRAMACIÓN:

Observaciones:

- Ninguna.

Tabla 16. Historia de Usuario “Registrar lenguajes de programación” (Elaboración propia)

E. Historia de Usuario “Modificar lenguajes de programación”

En la Tabla 17 se describe a la Historia de Usuario “Modificar lenguajes de programación”.

Historia de Usuario “Modificar lenguajes de programación”																					
Código: HU9	Enunciado: “Como supervisor de fábrica necesito modificar los lenguajes de programación que puedan tener los requerimientos y gracias a esto mantener actualizada la información de los lenguajes en que se desarrollarán los requerimientos ”.																				
Sprint: 2																					
<p>Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú secundario “Módulo Mantenimiento”, luego seleccionar la opción “Gestionar lenguajes de programación” y el sistema muestra la información de los lenguajes de programación registrados en un listado.</p> <p>Cuando se requiera modificar un lenguaje de programación, el supervisor de fábrica identifica al lenguaje en la lista, presiona el botón “Editar” y el sistema muestra un formulario para modificar su dato. Luego, el supervisor de fábrica modifica el lenguaje de programación, presiona el botón “Editar” para confirmar la modificación.</p> <p>Para limpiar los campos presionar el botón “Limpiar”.</p> <p>Para regresar a la página anterior presionar el botón “Regresar”.</p>																					
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. Para modificar un lenguaje de programación debe existir en la base de datos de la aplicación. 																					
<p>Prototipo:</p> <div style="border: 1px solid black; padding: 10px; text-align: center;"> <h3>GESTIÓN DE LENGUAJE DE PROGRAMACIÓN</h3> <p>Filtro por lenguaje: <input type="text" value="Search..."/> <input type="button" value="Buscar"/> <input type="button" value="Actualizar"/> <input type="button" value="Registrar nuevo lenguaje"/></p> <table border="1"> <thead> <tr> <th>Cod. Lenguaje</th> <th>Lenguaje de Programación</th> <th colspan="2">Acción</th> </tr> </thead> <tbody> <tr> <td>Cell 2</td> <td>Cell</td> <td><input type="button" value="EDITAR"/></td> <td><input type="button" value="BORRAR"/></td> </tr> <tr> <td>Cell 5</td> <td>Cell6</td> <td><input type="button" value="EDITAR"/></td> <td><input type="button" value="BORRAR"/></td> </tr> <tr> <td>Cell 8</td> <td>Cell9</td> <td><input type="button" value="EDITAR"/></td> <td><input type="button" value="BORRAR"/></td> </tr> <tr> <td>Cell 11</td> <td>Cell 2</td> <td><input type="button" value="EDITAR"/></td> <td><input type="button" value="BORRAR"/></td> </tr> </tbody> </table> <p>Anterior 1,2,3 Siguiete</p> <input type="button" value="Regresar"/> </div>		Cod. Lenguaje	Lenguaje de Programación	Acción		Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>	Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>	Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>	Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cod. Lenguaje	Lenguaje de Programación	Acción																			
Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>																		
Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>																		
Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>																		
Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>																		

GESTIÓN DE LENGUAJE DE PROGRAMACIÓN

EDITAR LENGUAJE DE PROGRAMACIÓN

COD. LENGUAJE:

LENGUAJE DE PROGRAMACIÓN:

Editar
Limpiar

Regresar

Observaciones:

- Para regresar a la página anterior debe desarrollarse un botón “Regresar”.

Tabla 17. Historia de Usuario “Modificar lenguajes de programación” (Elaboración propia)

F. Historia de Usuario “Eliminar lenguajes de programación”

En la Tabla 18 se describe a la Historia de Usuario “Eliminar lenguajes de programación”.

Historia de Usuario “Eliminar lenguajes de programación”	
Código: HU10	Enunciado: “Como supervisor de fábrica necesito eliminar los lenguajes de programación que ya no se atiendan en la fábrica de software y gracias a esto mantener la información actualizada”.
Sprint: 2	
Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú secundario “Módulo Mantenimiento”, luego seleccionar la opción “Gestionar lenguajes de programación” y el sistema muestra la información de los lenguajes de programación registrados en un listado.	
Cuando se requiera eliminar un lenguaje de programación, el supervisor de fábrica identifica al mismo en la lista y presiona el botón “Borrar”.	
Criterios de aceptación:	
1. Para eliminar un lenguaje de programación debe existir en la base de datos de la aplicación.	
Prototipo:	

GESTIÓN DE LENGUAJE DE PROGRAMACIÓN

Filtro por lenguaje:

Cod. Lenguaje	Lenguaje de Programación	Acción	
Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>

[Anterior 1,2,3](#) [Siguiete](#)

Observaciones:

- No considerar mensaje de confirmación cuando se elimine un lenguaje de programación.

Tabla 18. Historia de Usuario “Eliminar lenguajes de programación” (Elaboración propia)

G. Historia de Usuario “Registrar prioridad de requerimientos”

En la Tabla 19 se describe a la Historia de Usuario “Registrar prioridad de requerimientos”.

Historia de Usuario “Registrar prioridad de requerimientos”	
Código: HU11	Enunciado: “Como supervisor de fábrica necesito registrar la prioridad establecida para los requerimientos que se atenderán y gracias a esto conocer la prioridad que posee requerimiento para realizar la debida atención”.
Sprint: 2	
<p>Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú principal “Módulo Mantenimiento”, luego seleccionar la opción “Gestionar prioridad” y el sistema muestra las prioridades en un listado.</p> <p>Cuando se requiera crear una prioridad, se presiona el botón “Registrar nueva prioridad” y el sistema muestra un formulario para ingresar la información relacionada al mismo. Luego, el supervisor de fábrica ingresa la descripción de la prioridad:</p> <p>- Prioridad</p> <p>Luego presiona el botón “Registrar” para confirmar el registro.</p> <p>Para limpiar los campos presionar el botón “Limpiar”.</p> <p>Para regresar a la página anterior presionar el botón “Regresar”.</p>	

Criterios de aceptación:

1. Ingresar correctamente el dato requerido.

Prototipo:

GESTIÓN DE PRIORIDAD

Filtro por prioridad:

Cod. Prioridad	Prioridad	Acción
Cell 2	Cell	<input type="button" value="EDITAR"/> <input type="button" value="BORRAR"/>
Cell 5	Cell6	<input type="button" value="EDITAR"/> <input type="button" value="BORRAR"/>
Cell 8	Cell9	<input type="button" value="EDITAR"/> <input type="button" value="BORRAR"/>
Cell 11	Cell 2	<input type="button" value="EDITAR"/> <input type="button" value="BORRAR"/>

[Anterior](#) [1,2,3](#) [Siguinte](#)

GESTIÓN DE PRIORIDAD

NUEVA PRIORIDAD

PRIORIDAD:

Observaciones:

- Se solicitó el desarrollo del mantenedor de prioridades para futuros cambios en el manejo del tipo de prioridad de cada requerimiento.

Tabla 19. Historia de Usuario “Registrar prioridad de requerimientos” (Elaboración propia)

H. Historia de Usuario “Modificar prioridad de requerimientos”

En la Tabla 20 se describe a la Historia de Usuario “Modificar prioridad de requerimientos”.

Historia de Usuario “Modificar prioridad de requerimientos”																					
Código: HU12	Enunciado: “Como supervisor de fábrica necesito modificar la prioridad establecida para los requerimientos que se atenderán y gracias a esto conocer la información actualizada ”.																				
Sprint: 2																					
Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú principal “Módulo Mantenimiento”, luego seleccionar la opción “Gestionar prioridad” y el sistema muestra las prioridades en un listado. Cuando se requiera modificar la prioridad, el supervisor identifica a la misma en la lista, presiona el botón “Editar” y el sistema muestra un formulario para modificarla. Luego, el supervisor de fábrica modifica la prioridad, presiona el botón “Editar” para confirmar la modificación. Para limpiar los campos presionar el botón “Limpiar”. Para regresar a la página anterior presionar el botón “Regresar”.																					
Criterios de aceptación: <ol style="list-style-type: none">1. Para modificar una prioridad debe existir en la base de datos de la aplicación.2. Para modificar la prioridad debe estar asociada a un código de prioridad.																					
Prototipo: <div style="border: 1px solid black; padding: 10px; text-align: center;"><h3>GESTIÓN DE PRIORIDAD</h3><p>Filtro por prioridad: <input type="text" value="Search..."/> <input type="button" value="Buscar"/> <input type="button" value="Actualizar"/> <input type="button" value="Registrar nueva prioridad"/></p><table border="1"><thead><tr><th>Cod. Prioridad</th><th>Prioridad</th><th colspan="2">Acción</th></tr></thead><tbody><tr><td>Cell 2</td><td>Cell</td><td><input type="button" value="EDITAR"/></td><td><input type="button" value="BORRAR"/></td></tr><tr><td>Cell 5</td><td>Cell6</td><td><input type="button" value="EDITAR"/></td><td><input type="button" value="BORRAR"/></td></tr><tr><td>Cell 8</td><td>Cell9</td><td><input type="button" value="EDITAR"/></td><td><input type="button" value="BORRAR"/></td></tr><tr><td>Cell 11</td><td>Cell 2</td><td><input type="button" value="EDITAR"/></td><td><input type="button" value="BORRAR"/></td></tr></tbody></table><p>Anterior 1,2,3 Siguiente</p><input type="button" value="Regresar"/></div>		Cod. Prioridad	Prioridad	Acción		Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>	Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>	Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>	Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cod. Prioridad	Prioridad	Acción																			
Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>																		
Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>																		
Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>																		
Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>																		

GESTIÓN DE PRIORIDAD

EDITAR PRIORIDAD

COD. PRIORIDAD:

PRIORIDAD:

Editar
Limpiar

Regresar

Observaciones:

- Para regresar a la página anterior debe desarrollarse un botón “Regresar”.
- Se añadió botón “Regresar” en los prototipos.

Tabla 20. Historia de Usuario “Modificar prioridad de requerimientos” (Elaboración propia)

I. Historia de Usuario “Eliminar prioridad de requerimientos”

En la Tabla 21 se describe a la Historia de Usuario “Eliminar prioridad de requerimientos”.

Historia de Usuario “Eliminar prioridad de requerimientos”	
Código: HU13	Enunciado: “Como supervisor de fábrica necesito eliminar la prioridad establecida para los requerimientos que se atenderán y gracias a esto conocer mantener la información actualizada”.
Sprint: 2	
Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú principal “Módulo Mantenimiento”, luego seleccionar la opción “Gestionar prioridad” y el sistema muestra las prioridades en un listado. Cuando se requiera eliminar una prioridad, el supervisor de fábrica identifica a la misma en la lista y presiona el botón “Borrar”.	
Criterios de aceptación: 1. Para eliminar una prioridad debe existir en la base de datos de la aplicación.	
Prototipo:	

GESTIÓN DE PRIORIDAD

Filtro por prioridad:

Cod. Prioridad	Prioridad	Acción	
Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>

Anterior 1,2,3 Siguiente

Observaciones:

- No se requiere mensaje de confirmación cuando se elimine un registro.

Tabla 21. Historia de Usuario “Eliminar prioridad de requerimientos” (Elaboración propia)

J. Historia de Usuario “Registrar clientes”

En la Tabla 22 se describe a la Historia de Usuario “Registrar clientes”.

Historia de Usuario “Registrar clientes”	
Código: HU14	Enunciado: “Como supervisor de fábrica necesito registrar los clientes asociados a los requerimientos y gracias a esto saber quién es el solicitante del requerimiento ”.
Sprint: 2	
<p>Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú principal “Módulo Mantenimiento”, luego seleccionar la opción “Gestionar clientes” y el sistema muestra los clientes en un listado.</p> <p>Cuando se requiera crear un cliente, se presiona el botón “Registrar nuevo Cliente” y el sistema muestra un formulario para ingresar la información relacionada al mismo. Luego, el supervisor de fábrica ingresa la información del cliente:</p> <ul style="list-style-type: none"> - Razón Social - RUC - Área Cliente - Primer Nombre - Segundo Nombre 	

- Ap. Paterno
- Ap. Materno
- DNI
- Género
- Correo Electrónico
- Teléfono
- Cargo

Luego presiona el botón “Registrar” para confirmar el registro.

Para limpiar los campos presionar el botón “Limpiar”.

Para regresar a la página anterior presionar el botón “Regresar”.

Criterios de aceptación:

1. Se debe validar que, si el cliente ya existe en la base de datos, no permita registrar el mismo nuevamente.
2. Si el DNI ingresado ya está registrado en la base de datos, se mostrará el siguiente mensaje: “Actualmente el DNI está registrado, volver a intentar.”
3. Si el RUC ingresado ya está registrado en la base de datos, se mostrará el siguiente mensaje: “Actualmente el RUC está registrado, volver a intentar.”
4. Para registrar nuevo cliente se debe ingresar los datos correctamente.

Prototipo:

GESTIÓN DE CLIENTES

Filtro por Raz. Soc.:

Cod. Cliente	Raz. Soc.	Líder TI	Área	Cargo	Acción	
Cell 1	Cell 1	Cell 1	Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 4	Cell 4	Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 7	Cell 7	Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 10	Cell 10	Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>

Anterior 1,2,3 Siguiente

GESTIÓN DE CLIENTES

NUEVO CLIENTE

DATOS EMPRESA

RAZÓN SOCIAL:

RUC: ÁREA CLIENTE:

DATOS PERSONALES

PRIMER NOMBRE: SEGUNDO NOMBRE:

AP. PATERNO: AP. MATERNO:

DNI: GENERO:

CORREO ELECTRÓNICO:

TELÉFONO: CARGO:

Observaciones:

- Ninguna.

Tabla 22. Historia de Usuario “Registrar clientes” (Elaboración propia)

K. Historia de Usuario “Modificar clientes”

En la Tabla 23 se describe a la Historia de Usuario “Modificar clientes”.

Historia de Usuario “Modificar clientes”	
Código: HU15	Enunciado: “Como supervisor de fábrica necesito modificar la información de los clientes solicitantes de requerimientos y gracias a esto mantener la información actualizada de los clientes ”.
Sprint: 2	
<p>Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú principal “Módulo Mantenimiento”, luego seleccionar la opción “Gestionar clientes” y el sistema muestra los clientes en un listado.</p> <p>Cuando se requiera modificar la información del cliente, el supervisor de fábrica identifica a la misma en la lista, presiona el botón “Editar” y el sistema muestra un formulario para modificar su información. Luego, el supervisor de fábrica modifica los datos, presiona el botón “Editar” para confirmar la modificación de los</p>	

siguientes campos:

- Razón Social
- RUC
- Área Cliente
- Primer Nombre
- Segundo Nombre
- Apellido Paterno
- Apellido Materno
- Género
- Teléfono
- Correo Electrónico
- Cargo

Para limpiar los campos presionar el botón “Limpiar”.

Para regresar a la página anterior presionar el botón “Regresar”.

Criterios de aceptación:

1. Para modificar un cliente debe existir en la base de datos de la aplicación.
2. Si el RUC ingresado ya está registrado en la base de datos, se mostrará el siguiente mensaje:
“Actualmente el RUC está registrado, volver a intentar.”

Prototipo:

GESTIÓN DE CLIENTES

Filtro por Raz. Soc.:

Cod. Cliente	Raz. Soc.	Líder TI	Área	Cargo	Acción	
Cell 1	Cell 1	Cell 1	Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 4	Cell 4	Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 7	Cell 7	Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 10	Cell 10	Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>

[Anterior](#) [1,2,3](#) [Siguiente](#)

GESTIÓN DE CLIENTES

EDITAR CLIENTE

CÓDIGO CLIENTE:

DATOS EMPRESA

RAZÓN SOCIAL:

RUC: ÁREA CLIENTE:

DATOS PERSONALES

PRIMER NOMBRE: SEGUNDO NOMBRE:

AP. PATERNO: AP. MATERNO:

DNI: GENERO:

CORREO ELECTRÓNICO:

TELÉFONO: CARGO:

Observaciones:

- Para regresar a la página anterior debe desarrollarse un botón “Regresar”.
- Se añadió botón “Regresar” en los prototipos.

Tabla 23. Historia de Usuario “Modificar clientes” (Elaboración propia)

L. Historia de Usuario “Eliminar clientes”

En la Tabla 24 se describe a la Historia de Usuario “Eliminar clientes”.

Historia de Usuario “Eliminar clientes”	
Código: HU16	Enunciado: “Como supervisor de fábrica necesito eliminar los clientes que ya no desean solicitar requerimientos y gracias a esto mantener la información actualizada”.
Sprint: 2	
Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú principal “Módulo Mantenimiento”, luego seleccionar la opción “Gestionar clientes” y el sistema muestra los clientes en un listado. Cuando se requiera eliminar un cliente, el supervisor de fábrica identifica al mismo en la lista y presiona el botón “Eliminar”.	

Criterios de aceptación:

1. Para eliminar un cliente, este debe existir en la base de datos de la aplicación.

Prototipo:

GESTIÓN DE CLIENTES

Filtro por Raz. Soc.:

Cod. Cliente	Raz. Soc.	Líder TI	Área	Cargo	Acción	
Cell 1	Cell 1	Cell 1	Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 4	Cell 4	Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 7	Cell 7	Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 10	Cell 10	Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>

[Anterior 1,2,3](#) [Siguiete](#)

Observaciones:

- Se requiere controlar la información del cliente, tener en cuenta campo “Razón Social”.

Tabla 24. Historia de Usuario “Eliminar clientes” (Elaboración propia)

5.2.3. Sprint 3

A. Historia de Usuario “Registrar requerimientos”

En la Tabla 25 se describe a la Historia de Usuario “Registrar requerimientos”.

Historia de Usuario “Registrar requerimientos”	
Código: HU17	Enunciado: “Como supervisor de fábrica necesito registrar los requerimientos y gracias a esto controlar la información de los mismos ”.
Sprint: 3	
Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú principal “Gestión de Requerimientos”, y el sistema muestra los requerimientos en un listado. Cuando se requiera crear un requerimiento, se presiona el botón “Registrar nuevo Requerimiento” y el sistema muestra un formulario para ingresar la información relacionada al mismo. Luego, el supervisor de fábrica ingresa la información del requerimiento:	

- Código de solicitud
- Líder TI (Solicitante del requerimiento)

- Código de requerimiento
- Nombre de requerimiento
- Descripción del requerimiento
- Responsable del requerimiento
- Fecha Inicio del requerimiento
- Fecha Fin del requerimiento
- Estado del requerimiento (No editable)
- Tipo de servicio
- Horas planificadas
- Prioridad
- Lenguaje de programación
- Clasificación

Luego presiona el botón “Registrar” para confirmar el registro.

Para limpiar los campos presionar el botón “Limpiar”.

Para regresar a la página anterior presionar el botón “Regresar”.

Criterios de aceptación:

1. Se debe validar que, si el código del requerimiento ya existe en la base de datos, no permita registrar el mismo nuevamente.
2. Si el código de solicitud ingresado ya existe en la base de datos, se mostrará el mensaje: “Actualmente el Código de Solicitud está registrado, volver a intentar.”.
3. Si el código de requerimiento ya existe en la base de datos, se mostrará el mensaje: ”Actualmente el Código de Requerimiento está registrado, volver a intentar.”.

Prototipo:

GESTIÓN DE REQUERIMIENTOS

Filtro por Nombre:

Cod. Req.	Nombre	Descripción	Estado	Fecha Inicio	Fecha Fin	Acción	
Cell 1	Cell 1	Cell 1	Cell 1	Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 1	Cell 4	Cell 4	Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 1	Cell 7	Cell 7	Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 1	Cell 10	Cell 10	Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>

[Anterior 1,2,3 Siguiente](#)

GESTIÓN DE REQUERIMIENTOS NUEVO REQUERIMIENTO

DATOS CLIENTE

CÓDIGO DE SOLICITUD: LÍDER TI:

DATOS REQUERIMIENTO

CÓDIGO REQ.: RESPONSABLE:

NOMBRE REQ.:

DESCRIPCIÓN:

FECHA INICIO: FECHA FIN:

ESTADO REQ.: TIPO SERVICIO:

HRS. PLANIFICADAS: PRIORIDAD:

LENGUAJE PROG.: CLASIFICACIÓN:

FEC. INICIO REAL: FEC. FIN REAL:

Observaciones:

- Para regresar a la página anterior debe desarrollarse un botón “Regresar”.
- Se añadió botón “Regresar” en los prototipos.

Tabla 25. Historia de Usuario “Registrar requerimientos” (Elaboración propia)

B. Historia de Usuario “Modificar requerimientos”

En la Tabla 26 se describe a la Historia de Usuario “Modificar requerimientos”.

Historia de Usuario “Modificar requerimientos”	
Código: HU18	Enunciado: “Como supervisor de fábrica necesito modificar los requerimientos y gracias a esto controlar la información de los mismos ”.
Sprint: 3	
<p>Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú principal “Gestión de Requerimientos”, y el sistema muestra los requerimientos en un listado.</p> <p>Cuando se requiera modificar la información del requerimiento, el supervisor identifica a la misma en la lista, presiona el botón “Editar” y el sistema muestra un formulario para modificar su información. Luego, el supervisor de fábrica modifica los datos, presiona el botón “Editar” para confirmar la modificación de los campos:</p> <ul style="list-style-type: none">- Nombre de requerimiento- Descripción del requerimiento- Responsable del requerimiento- Tipo de servicio- Horas planificadas- Prioridad- Lenguaje de programación- Clasificación- Fecha inicio real del requerimiento- Fecha fin real del requerimiento <p>Para limpiar los campos presionar el botón “Limpiar”.</p> <p>Para regresar a la página anterior presionar el botón “Regresar”.</p>	
Criterios de aceptación: <ol style="list-style-type: none">1. Para modificar un requerimiento debe existir en la base de datos de la aplicación.	
Prototipo:	

GESTIÓN DE REQUERIMIENTOS

Filtro por Nombre:

Cod. Req.	Nombre	Descripción	Estado	Fecha Inicio	Fecha Fin	Acción	
Cell 1	Cell 1	Cell 1	Cell 1	Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 1	Cell 4	Cell 4	Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 1	Cell 7	Cell 7	Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cell 1	Cell 1	Cell 10	Cell 10	Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>

[Anterior 1,2,3](#) [Siguiete](#)

GESTIÓN DE REQUERIMIENTOS

EDITAR REQUERIMIENTO

DATOS REQUERIMIENTO

CÓDIGO REQ.:

NOMBRE REQ.:

DESCRIPCIÓN:

TIPO SERVICIO: RESPONSABLE:

HRS. PLANIFICADAS: PRIORIDAD:

LENGUAJE PROG.: CLASIFICACIÓN:

FEC. INICIO REAL: FEC. FIN REAL:

Observaciones:

- Ninguna.

Tabla 26. Historia de Usuario "Modificar requerimientos" (Elaboración propia)

C. Historia de Usuario “Eliminar requerimientos”

En la Tabla 27 se describe a la Historia de Usuario “Eliminar requerimientos”.

Historia de Usuario “Eliminar requerimientos”																																									
Código: HU19	Enunciado: “Como supervisor de fábrica necesito eliminar los requerimientos y gracias a esto controlar la información de los mismos”.																																								
Sprint: 3																																									
<p>Descripción: El supervisor de fábrica ingresa a la aplicación y selecciona del menú principal “Gestión de Requerimientos”, y el sistema muestra los requerimientos en un listado.</p> <p>Cuando se requiera eliminar un requerimiento, el supervisor de fábrica identifica al mismo en la lista y presiona el botón “Borrar”.</p>																																									
<p>Criterios de aceptación:</p> <ol style="list-style-type: none"> 1. Para eliminar un requerimiento debe existir en la base de datos de la aplicación web. 																																									
<p>Prototipo:</p> <div style="border: 1px solid black; padding: 10px; text-align: center;"> <h4>GESTIÓN DE REQUERIMIENTOS</h4> <p>Filtro por Nombre: <input type="text" value="Search..."/> <input type="button" value="Buscar"/> <input type="button" value="Actualizar"/> <input type="button" value="Registrar nuevo requerimiento"/></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Cod. Req.</th> <th>Nombre</th> <th>Descripción</th> <th>Estado</th> <th>Fecha Inicio</th> <th>Fecha Fin</th> <th colspan="2">Acción</th> </tr> </thead> <tbody> <tr> <td>Cell 1</td> <td>Cell 1</td> <td>Cell 1</td> <td>Cell 1</td> <td>Cell 2</td> <td>Cell</td> <td><input type="button" value="EDITAR"/></td> <td><input type="button" value="BORRAR"/></td> </tr> <tr> <td>Cell 1</td> <td>Cell 1</td> <td>Cell 4</td> <td>Cell 4</td> <td>Cell 5</td> <td>Cell6</td> <td><input type="button" value="EDITAR"/></td> <td><input type="button" value="BORRAR"/></td> </tr> <tr> <td>Cell 1</td> <td>Cell 1</td> <td>Cell 7</td> <td>Cell 7</td> <td>Cell 8</td> <td>Cell9</td> <td><input type="button" value="EDITAR"/></td> <td><input type="button" value="BORRAR"/></td> </tr> <tr> <td>Cell 1</td> <td>Cell 1</td> <td>Cell 10</td> <td>Cell 10</td> <td>Cell 11</td> <td>Cell 2</td> <td><input type="button" value="EDITAR"/></td> <td><input type="button" value="BORRAR"/></td> </tr> </tbody> </table> <p>Anterior 1,2,3 Siguiente</p> <input type="button" value="Regresar"/> </div>		Cod. Req.	Nombre	Descripción	Estado	Fecha Inicio	Fecha Fin	Acción		Cell 1	Cell 1	Cell 1	Cell 1	Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>	Cell 1	Cell 1	Cell 4	Cell 4	Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>	Cell 1	Cell 1	Cell 7	Cell 7	Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>	Cell 1	Cell 1	Cell 10	Cell 10	Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>
Cod. Req.	Nombre	Descripción	Estado	Fecha Inicio	Fecha Fin	Acción																																			
Cell 1	Cell 1	Cell 1	Cell 1	Cell 2	Cell	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>																																		
Cell 1	Cell 1	Cell 4	Cell 4	Cell 5	Cell6	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>																																		
Cell 1	Cell 1	Cell 7	Cell 7	Cell 8	Cell9	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>																																		
Cell 1	Cell 1	Cell 10	Cell 10	Cell 11	Cell 2	<input type="button" value="EDITAR"/>	<input type="button" value="BORRAR"/>																																		
<p>Observaciones:</p> <ul style="list-style-type: none"> - Luego de eliminar un registro no es necesario que se muestre mensaje de confirmación. 																																									

Tabla 27. Historia de Usuario “Eliminar requerimientos” (Elaboración propia)

5.2.4. Sprint 4

A. Historia de Usuario “Evaluar requerimientos”

En la Tabla 28 se describe a la Historia de Usuario “Evaluar requerimientos”.

Historia de Usuario “Evaluar requerimientos”	
Código: HU20	Enunciado: “Como usuario necesito ver detalles de los requerimientos y gracias a esto visualizar la información de los requerimientos que se pueden aprobar o rechazar ”.
Sprint: 4	
<p>Descripción: El usuario ingresa a la aplicación y selecciona del menú principal “Requerimientos por Apr./Rch.” y el sistema muestra la información de los requerimientos que están listos para ser evaluados por los ingenieros según estén en sus bandejas respectivas para cada evaluación por estado.</p> <ul style="list-style-type: none"> - Código de requerimiento - Nombre de requerimiento - Descripción del requerimiento - Estado del requerimiento <p>Cuando se requiera visualizar el detalle de los requerimientos se debe seleccionar la opción “Ver Detalle”, la cual direcciona a la interfaz donde se muestra la información relacionada al requerimiento, mostrando los datos:</p> <ul style="list-style-type: none"> - Código de solicitud - Líder TI (Solicitante del requerimiento) - Código de requerimiento - Responsable del requerimiento - Nombre de requerimiento - Descripción del requerimiento - Fecha Inicio del requerimiento - Fecha Fin del requerimiento - Estado - Tipo Servicio - Horas planificadas - Prioridad - Lenguaje de programación - Clasificación - Fecha inicio real del requerimiento - Fecha fin real del requerimiento <p>Para proceder a la aprobación del requerimiento debe presionar el botón “Aprobar”.</p> <p>Para proceder el rechazo del requerimiento debe presionar el botón “Rechazar”.</p>	

Para regresar a la página anterior presionar el botón “Regresar”.

Criterios de aceptación:

1. Los requerimientos que se muestren en bandeja deben estar registrados en la base de datos.
2. La información del requerimiento que se muestra en “Detalles” debe estar directamente relacionada con el requerimiento solicitado.
3. Sólo el supervisor de fábrica tendrá las opciones de “Aprobar” y “Rechazar” dentro de la misma interfaz donde ese muestra la información del requerimiento.

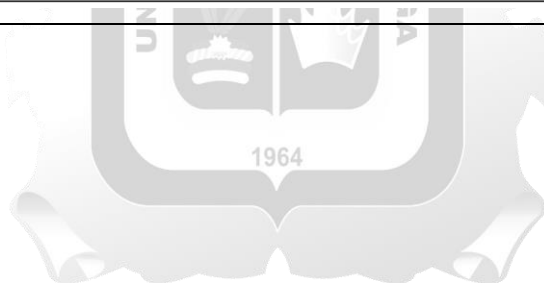
Prototipo:

APROBACIÓN/RECHAZO DE REQUERIMIENTOS

Filtro por Nombre:

Cod. Req.	Nombre	Descripción	Estado	Acción	Acción	Acción
Cell 1	Cell 1	Cell 1	Cell 2	<input type="button" value="VER DETALLE"/>	<input type="button" value="APROBAR"/>	<input type="button" value="RECHAZAR"/>
Cell 1	Cell 4	Cell 4	Cell 5	<input type="button" value="VER DETALLE"/>	<input type="button" value="APROBAR"/>	<input type="button" value="RECHAZAR"/>
Cell 1	Cell 7	Cell 7	Cell 8	<input type="button" value="VER DETALLE"/>	<input type="button" value="APROBAR"/>	<input type="button" value="RECHAZAR"/>
Cell 1	Cell 10	Cell 10	Cell 11	<input type="button" value="VER DETALLE"/>	<input type="button" value="APROBAR"/>	<input type="button" value="RECHAZAR"/>

[Anterior](#) [1,2,3](#) [Siguiente](#)



APROBACIÓN/RECHAZO DE REQUERIMIENTOS
INFORMACIÓN DEL REQUERIMIENTO

DATOS CLIENTE

CÓDIGO DE SOLICITUD: LÍDER TI:

DATOS REQUERIMIENTO

CÓDIGO REQ.: RESPONSABLE:

NOMBRE REQ.:

DESCRIPCIÓN:

FECHA INICIO: FECHA FIN:

ESTADO REQ.: TIPO SERVICIO:

HRS. PLANIFICADAS: PRIORIDAD:

LENGUAJE PROG.: CLASIFICACIÓN:

FEC. INICIO REAL: FEC. FIN REAL:

Observaciones:

- Se requiere se muestre un campo “Hrs. planificadas” las cuales indican las horas de atención para cumplir con el requerimiento.

Tabla 28. Historia de Usuario “Evaluar requerimientos” (Elaboración propia)

B. Historia de Usuario “Aprobar requerimientos”

En la Tabla 29 se describe a la Historia de Usuario “Aprobar requerimientos”.

Historia de Usuario “Aprobar requerimientos”	
Código: HU21	Enunciado: “Como usuario necesito aprobar requerimientos y gracias a esto continuar con el proceso de gestión de requerimientos hasta atender los mismos ”.
Sprint: 4	
<p>Descripción: El usuario ingresa a la aplicación y selecciona del menú principal “Requerimientos por Apr./Rch.” y el sistema muestra la información de los requerimientos que están listos para ser evaluados por los ingenieros según estén en sus bandejas respectivas para cada evaluación por estado.</p> <p>Cuando se requiera aprobar un requerimiento, se ubica el mismo en el listado que se muestra inicialmente y se presiona el botón “Aprobar”.</p> <p>Se abre una nueva interfaz mostrándose los datos:</p> <ul style="list-style-type: none"> - Código de requerimiento - Nombre del requerimiento 	

Así mismo se muestra un campo para ingresar el sustento de la aprobación a realizar.

Finalmente se muestran los botones “Aprobar” para continuar con la derivación y aprobar el requerimiento cambiando al siguiente estado y el botón “Regresar”, el cual retorna a la página inicial.

Criterios de aceptación:

1. Los requerimientos cuando sean aprobados no deben mostrarse en la bandeja del ingeniero evaluador.
2. Los requerimientos que se muestren en bandeja deben estar registrados en la base de datos.
3. El supervisor de fábrica sólo podrá evaluar los requerimientos cuando se encuentren en estado “Registrado”, cuando se aprueba el requerimiento cambia a estado “Estimación técnica”.
4. El ingeniero de software sólo podrá evaluar los requerimientos cuando se encuentren en estado “Estimación técnica”, cuando se aprueba el requerimiento cambia a estado “Estimación QA”.
5. El analista QA sólo podrá evaluar los requerimientos cuando se encuentren en estado “Estimación QA”, cuando se aprueba el requerimiento cambia a estado “Cotizado”.
6. El gestor de proyectos podrá evaluar los requerimientos cuando se encuentren en los siguientes estados:
 - “Cotizado”, cuando se aprueba el requerimiento cambia a estado “En desarrollo”.
 - “En desarrollo”, cuando se aprueba el requerimiento cambia a estado “Terminado”.

Prototipo:

APROBACIÓN/RECHAZO DE REQUERIMIENTOS

Filtro por Nombre:

Cod. Req.	Nombre	Descripción	Estado	Acción	Acción	Acción
Cell 1	Cell 1	Cell 1	Cell 2	<input type="button" value="VER DETALLE"/>	<input type="button" value="APROBAR"/>	<input type="button" value="RECHAZAR"/>
Cell 1	Cell 4	Cell 4	Cell 5	<input type="button" value="VER DETALLE"/>	<input type="button" value="APROBAR"/>	<input type="button" value="RECHAZAR"/>
Cell 1	Cell 7	Cell 7	Cell 8	<input type="button" value="VER DETALLE"/>	<input type="button" value="APROBAR"/>	<input type="button" value="RECHAZAR"/>
Cell 1	Cell 10	Cell 10	Cell 11	<input type="button" value="VER DETALLE"/>	<input type="button" value="APROBAR"/>	<input type="button" value="RECHAZAR"/>

[Anterior 1,2,3](#) [Siguiente](#)

APROBACIÓN/RECHAZO DE REQUERIMIENTOS

APROBAR REQUERIMIENTO

DATOS REQUERIMIENTO

CÓDIGO REQ.:

NOMBRE REQ.:

SUSTENTO DE APROBACIÓN

Observaciones:

- Se requiere un campo donde se pueda ingresar el sustento para aprobación del requerimiento.

Tabla 29. Historia de Usuario “Aprobar requerimientos” (Elaboración propia)

C. Historia de Usuario “Rechazar requerimientos”

En la Tabla 30 se describe a la Historia de Usuario “Rechazar requerimientos”.

Historia de Usuario “Rechazar requerimientos”	
Código: HU22	Enunciado: “Como usuario necesito rechazar requerimientos y gracias a esto continuar con el proceso de gestión de requerimientos hasta desestimarlos ”.
Sprint: 4	
<p>Descripción: El usuario ingresa a la aplicación y selecciona del menú principal “Apr./Rch. requerimientos” y el sistema muestra la información de los requerimientos que están listos para ser aprobados o rechazados según el estado que tengan.</p> <p>Cuando se requiera rechazar un requerimiento, se ubica el mismo en el listado que se muestra inicialmente y se presiona el botón “Rechazar”.</p> <p>Se abre una nueva interfaz mostrándose los campos deshabilitados:</p> <ul style="list-style-type: none"> - Código de requerimiento - Nombre del requerimiento 	

Así mismo se muestra un campo para ingresar el sustento del rechazo a realizar.

Finalmente se muestran los botones “Rechazar” para continuar con la derivación y rechazar el requerimiento cambiando al estado anterior y el botón “Regresar”, el cual retorna a la página inicial.

Criterios de aceptación:

1. Los requerimientos cuando sean rechazados no deben mostrarse en la bandeja del ingeniero evaluador, sino que retrocederá un estado hasta el ingeniero que lo tuvo con anterioridad.
2. Los requerimientos que se muestren en bandeja deben estar registrados en la base de datos.
3. El supervisor de fábrica sólo podrá evaluar los requerimientos cuando se encuentren en estado “Registrado”, cuando se rechaza el requerimiento cambia a estado “Desestimado”.
4. El ingeniero de software sólo podrá evaluar los requerimientos cuando se encuentren en estado “Estimación técnica”, cuando se rechaza el requerimiento cambia a estado “Derivado”.
5. El analista QA sólo podrá evaluar los requerimientos cuando se encuentren en estado “Estimación QA”, cuando se rechaza el requerimiento cambia a estado “Estimación técnica”.
6. El gestor de proyectos podrá evaluar los requerimientos cuando se encuentren en los siguientes estados:
 - “Cotizado”, cuando se rechaza el requerimiento cambia a estado “Estimación QA”.
 - “En desarrollo”, cuando se rechaza el requerimiento cambia a estado “Cotizado”.

Prototipo:

APROBACIÓN/RECHAZO DE REQUERIMIENTOS

Filtro por Nombre:

Cod. Req.	Nombre	Descripción	Estado	Acción	Acción	Acción
Cell 1	Cell 1	Cell 1	Cell 2	<input type="button" value="VER DETALLE"/>	<input type="button" value="APROBAR"/>	<input type="button" value="RECHAZAR"/>
Cell 1	Cell 4	Cell 4	Cell 5	<input type="button" value="VER DETALLE"/>	<input type="button" value="APROBAR"/>	<input type="button" value="RECHAZAR"/>
Cell 1	Cell 7	Cell 7	Cell 8	<input type="button" value="VER DETALLE"/>	<input type="button" value="APROBAR"/>	<input type="button" value="RECHAZAR"/>
Cell 1	Cell 10	Cell 10	Cell 11	<input type="button" value="VER DETALLE"/>	<input type="button" value="APROBAR"/>	<input type="button" value="RECHAZAR"/>

[Anterior 1,2,3](#) [Sigiente](#)

APROBACIÓN/RECHAZO DE REQUERIMIENTOS	
RECHAZAR REQUERIMIENTO	
<u>DATOS REQUERIMIENTO</u>	
CÓDIGO REQ.:	<input type="text"/>
NOMBRE REQ.:	<input type="text"/>
<u>SUSTENTO DE RECHAZO</u>	
<input type="text"/>	
<input type="button" value="Rechazar"/>	
<input type="button" value="Regresar"/>	

Observaciones:

- Se requiere un campo donde se pueda ingresar el sustento para rechazo del requerimiento.

Tabla 30. Historia de Usuario “Rechazar requerimientos” (Elaboración propia)

5.2.5. Sprint 5

A. Historia de Usuario “Consulta de requerimientos”

En la Tabla 31 se describe a la Historia de Usuario “Consulta de requerimientos”.

Historia de Usuario “Consulta de requerimientos”	
Código: HU23	Enunciado: “Como usuario necesito consultar los requerimientos y gracias a esto conocer la información de los mismos ”.
Sprint: 5	
Descripción: El usuario ingresa a la aplicación y selecciona del menú principal “Consulta de requerimientos” y el sistema muestra la información de los requerimientos. Cuando se requiera realizar una búsqueda, se ingresa como parámetro el nombre del requerimiento, mostrándose así la información relacionada al requerimiento. Para mayor detalle presionar el botón “Ver Información” para mostrar los siguientes campos: <ul style="list-style-type: none"> - Código de solicitud - Líder TI (Solicitante del requerimiento) 	

- Código de requerimiento
- Responsable del requerimiento
- Nombre de requerimiento
- Descripción del requerimiento
- Fecha Inicio del requerimiento
- Fecha Fin del requerimiento
- Nro. Recursos
- Estado
- Tipo Servicio
- Horas planificadas
- Prioridad
- Lenguaje de programación
- Clasificación
- Fecha inicio real del requerimiento
- Fecha fin real del requerimiento

Adicionalmente se muestra el botón “Historial”, el cual muestra el seguimiento de los requerimientos.

Como dato principal se muestra el código del requerimiento y después de ello lo siguiente:

- Usuario que aprueba/rechaza.
- Estado actual del requerimiento.
- Acción (Aprobado o Rechazado).
- Fecha en la que se aprobó o rechazó.
- Sustento

Así mismo se muestra el botón “Regresar”, el cual hace que se retorne a la página inicial.

Criterios de aceptación:

1. Los requerimientos que se muestren en bandeja deben estar registrados en la base de datos.
2. Se deben mostrar los requerimientos que cumplan con el parámetro ingresado.

Prototipo:

CONSULTA DE REQUERIMIENTOS

Filtro por Nombre:

Cod. Req.	Nombre	Descripción	Responsable	Estado	Tipo Serv.	Prioridad	Acción	
Cell 1	Cell 2	Cell 3	Cell 1	Cell 2	Cell 3	Cell 1	<input type="button" value="Ver Log"/>	<input type="button" value="Ver información"/>
Cell 4	Cell 5	Cell 6	Cell 2	Cell 3	Cell 1	Cell 2	<input type="button" value="Ver Log"/>	<input type="button" value="Ver información"/>
Cell 7	Cell 8	Cell 9	Cell 2	Cell 3	Cell 1	Cell 2	<input type="button" value="Ver Log"/>	<input type="button" value="Ver información"/>
Cell 10	Cell 11	Cell 12	Cell 2	Cell 3	Cell 1	Cell 2	<input type="button" value="Ver Log"/>	<input type="button" value="Ver información"/>

Anterior 1,2,3 Siguiente

CONSULTA DE REQUERIMIENTOS

INFORMACIÓN DEL REQUERIMIENTO

DATOS CLIENTE

CÓDIGO DE SOLICITUD: LÍDER TI:

DATOS REQUERIMIENTO

CÓDIGO REQ.: RESPONSABLE:

NOMBRE REQ.:

DESCRIPCIÓN:

FECHA INICIO: FECHA FIN:

ESTADO REQ.: TIPO SERVICIO:

HRS. PLANIFICADAS: PRIORIDAD:

LENGUAJE PROG.: CLASIFICACIÓN:

FEC. INICIO REAL: FEC. FIN REAL:

CONSULTA DE REQUERIMIENTOS
SEGUIMIENTO DE REQUERIMIENTO

Datos Requerimiento

CÓDIGO DE REQUERIMIENTO: **XXXX**

Seguimiento Requerimiento

USUARIO	ESTADO	ACCIÓN	FECHA	SUSTENTO
Cell 1	Cell 2	Cell 3	Cell 1	Cell 2
Cell 4	Cell 5	Cell 6	Cell 2	Cell 3
Cell 7	Cell 8	Cell 9	Cell 2	Cell 3
Cell 10	Cell 11	Cell 12	Cell 2	Cell 3

Regresar

Observaciones:

- Se requiere que se muestre un historial con todos los estados por los cuales pasó el requerimiento hasta ser “Terminado” o “Desestimado”.

Tabla 31. Historia de Usuario “Consulta de requerimientos” (Elaboración propia)

B. Historia de Usuario “Generar reporte general de requerimientos”

En la Tabla 32 se describe a la Historia de Usuario “Generar reporte general de requerimientos”.

Historia de Usuario “Generar reporte general de requerimientos”	
Código: HU24	Enunciado: “Como usuario necesito generar un reporte con la información de los requerimientos y gracias a esto llevar un control y mejorar la toma de decisiones ”.
Sprint: 5	
<p>Descripción: El usuario ingresa a la aplicación y selecciona del submenú principal “Reportes Generales” y el sistema despliega la opción “Reporte de requerimientos”.</p> <p>Para generar el reporte se selecciona dicha opción, y se mostrarán los datos de los requerimientos cuando el usuario sea el supervisor de fábrica:</p> <ul style="list-style-type: none"> - Nro. Correlativo - Código de requerimiento - Código de solicitud - Nombre de requerimiento - Estado 	

- Fecha Inicio del requerimiento
- Fecha Fin del requerimiento

Si el usuario es gestor de proyectos, ingeniero de software o analista QA, se mostrarán los siguientes datos en el reporte:

- Nro. Correlativo
- Código de requerimiento
- Nombre de requerimiento
- Estado
- Acción
- Fecha de acción

Así mismo se muestra la fecha, hora y el usuario que generó el reporte.

Criterios de aceptación:

1. Los requerimientos que se muestren en el reporte deben estar registrados en la base de datos.
2. Se debe mostrar el reporte en formato PDF.

Prototipo:



Reporte General de Requerimientos

N.º	CODREQ	CODSOL	NOMBREREQ	DESCRIPCIONREQ	FECHAINI	FECHAFIN
1	7979	77477	req7979B2	REQ 45	2018-01-09	2018-01-23
2	6959	77477	PRUEBA CRUDXD	DESCRIPCIN 1	0000-00-00	0000-00-00
3	9955	77477	PRUEBA223		0000-00-00	0000-00-00
4	7485		asdasqw	DESCRIPCIN 1222	0000-00-00	0000-00-00
5	1524		uyiyuiss		0000-00-00	0000-00-00
6	4444		wreqwe44		0000-00-00	0000-00-00
7	1231		asdasdas		0000-00-00	0000-00-00
8	8907		qweqweqwe		0000-00-00	0000-00-00
9	1124		asdassad1124		0000-00-00	0000-00-00
10	9090		wrwer66		0000-00-00	0000-00-00
11	6682		NUEVO2266		0000-00-00	0000-00-00
12	8700		NUEVO1906		0000-00-00	0000-00-00
13	6689		NUEVO2266		0000-00-00	0000-00-00
14	1090	31230	prueba	miray	2018-07-17	2018-07-31

Observaciones:

- Se requiere generar un reporte con todos los requerimientos registrados en fábrica.
- Se requiere un reporte por cada perfil (excepto supervisor de fábrica) mostrándose los requerimientos derivados.

Tabla 32. Historia de Usuario “Reporte general de requerimientos” (Elaboración propia)

5.3. Fase III - Post-Juego

En la presente fase se realizó el despliegue de la aplicación web en el Servidor EsHost el cual se encuentra funcionando operativamente en la nube, de la misma manera, también se realizó la capacitación respectiva al personal que tendrá el manejo de la aplicación web. Posteriormente se realizó la entrega de fuentes de la aplicación web al supervisor de la fábrica de software. Se aprecia en la figura 29 la aplicación web publicada en el servidor EsHost.

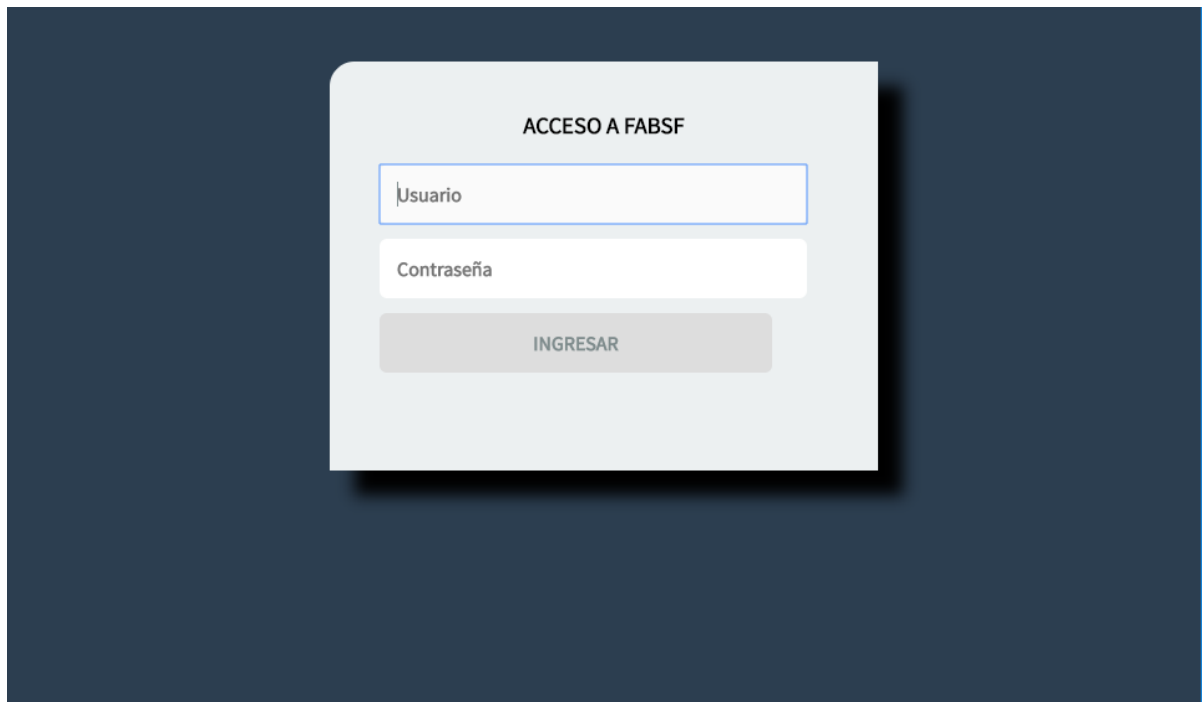
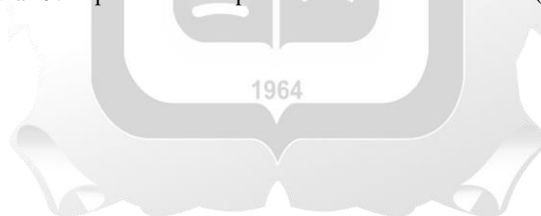


Figura 29. “Aplicación web publicada en servidor EsHost” (Elaboración propia)



CAPÍTULO VI: RESULTADOS

En el presente capítulo se sustenta el cumplimiento de los indicadores de calidad especificados en los objetivos y descritos en las bases teóricas del presente trabajo de investigación. Para lograr esto, se elaboró una encuesta (Ver Anexo III: Encuesta de evaluación de calidad sobre la aplicación web para la gestión de requerimientos de software en la empresa FabSF Services) empleando la escala de Likert, la cual consta de cinco niveles de respuesta.

NIVEL	RESPUESTA
1	Muy de acuerdo
2	De acuerdo
3	Ni de acuerdo ni en desacuerdo
4	Desacuerdo
5	Muy en desacuerdo

Tabla 33. Niveles de respuesta – Escala de Likert (Elaboración propia).

La población de estudio de la fábrica de software que usará la aplicación web para la gestión de requerimientos de software es de 52 personas. Según Icart, M. & Fuentelsaz, C. & Pulpón, A. (2006, p.55) la población de estudio es un conjunto de individuos que poseen características que se desean estudiar, sin embargo, cuando no se conoce la cantidad de individuos que constituyen este conjunto se emplea una muestra para realizar el estudio. La muestra es un subconjunto de la población que cumplen con las características a estudiar, y dicha muestra debe ser representativa para generalizar los resultados obtenidos.

En medida de que la población representa un número considerable de personas, se seleccionará como muestra a todas ellas. Por lo tanto, en base a la población de estudio de la fábrica de software de FabSF Services que es de 52 personas se determina que el número de encuestados será de la misma manera de 52 personas.

(Ver Anexo IV: Resultados de la encuesta de evaluación de calidad sobre la aplicación web para la gestión de requerimientos de software en la empresa FabSF Services).

- **Resultados respecto a la funcionalidad**

Para el indicador de funcionalidad, se evaluó la realización de la selección y programación de horarios de los tutores al ser las actividades de mayor uso.

Se concluye que la aplicación web para la gestión de requerimientos cumplió con el indicador de funcionalidad por las siguientes razones:

- La aplicación web para la gestión de requerimientos validó las credenciales (usuario y contraseña) ingresadas, teniendo como resultado un 82.7% indicando muy de acuerdo, 15.4% de acuerdo y un 1.9% ni de acuerdo ni en desacuerdo.
- La aplicación web para la gestión de requerimientos permitió realizar el seguimiento de los requerimientos que son atendidos por la fábrica de software satisfactoriamente, teniendo como resultado un 75% indicando muy de acuerdo, 23.1% de acuerdo y un 1.9% ni de acuerdo ni en desacuerdo.
- La aplicación web para la gestión de requerimientos redujo el tiempo de consulta de la información de los requerimientos en la fábrica de software, teniendo como resultado un 78.8% indicando muy de acuerdo, 19.2% de acuerdo y un 2 % ni de acuerdo ni en desacuerdo.

- **Resultados respecto a la usabilidad**

Se concluye que la aplicación web para la gestión de requerimientos cumplió con el indicador de usabilidad por las siguientes razones:

- Un 76.9% concluye que están muy de acuerdo en que fue fácil consultar la bitácora de las aprobaciones/rechazos de los requerimientos; mientras que un 21.2% concluye que están de acuerdo en que fue fácil, así mismo un 1.9% indica que no están ni de acuerdo ni en desacuerdo.
- Un 75% concluye que están muy de acuerdo en que la navegación para consultar la información de los requerimientos fue fácil; mientras que un 23.1% concluye que están de acuerdo en que fue fácil, así mismo un 1.9% indica que no está ni de acuerdo ni en desacuerdo.

- **Resultados respecto a la eficiencia**

Se concluye que la aplicación web para la gestión de requerimientos cumplió con el indicador de eficiencia por las siguientes razones:

- Un 76.9% concluye que está muy de acuerdo en que la consulta de información de los requerimientos se realizó con rapidez; mientras que un 21.2% concluye que están de acuerdo en que fue rápido, así mismo un 1.9% indica que no están de acuerdo ni en desacuerdo.
- Un 75% concluye que están muy de acuerdo en que el tiempo de respuesta de la generación del reporte con la información general de los requerimientos en la aplicación web para la gestión de requerimientos fue rápida; mientras que un 19.2% concluye que están de acuerdo en que fue rápida, así mismo un 5.8% indican que no están ni de acuerdo ni en desacuerdo.

- **Resultados respecto a la fiabilidad**

Se concluye que la aplicación web para la gestión de requerimientos cumplió con el indicador de fiabilidad por la siguiente razón:

- Un 75% concluye que están muy de acuerdo en que la aplicación web para la gestión de requerimientos controló de manera efectiva la información de los requerimientos al realizar las

derivaciones de los mismos; mientras que un 23.1% concluye que están de acuerdo, así mismo un 1.9% concluyen que no están ni de acuerdo ni en desacuerdo.

- **Resultados respecto a la funcionalidad - usabilidad**

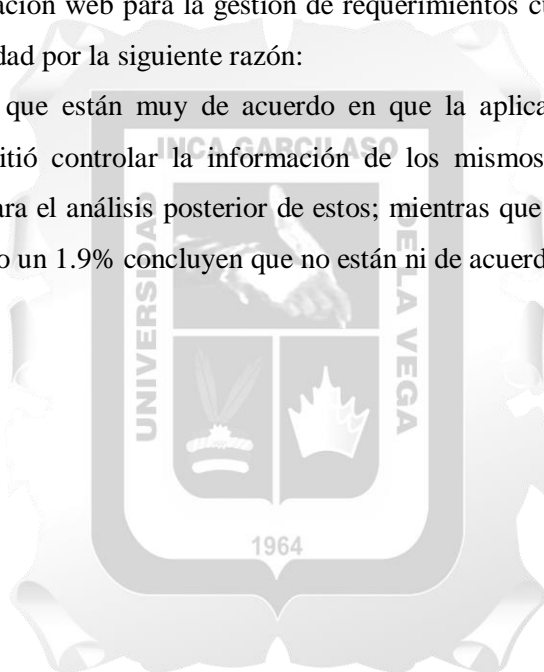
Se concluye que la aplicación web para la gestión de requerimientos cumplió con los indicadores de funcionabilidad y usabilidad por la siguiente razón:

- Un 75% concluye que están muy de acuerdo en que la aplicación web para la gestión de requerimientos mostró las opciones e interfaces asociadas a los perfiles correspondientes facilitando la aprobación/rechazo de los requerimientos; mientras que un 23.1% concluye que están de acuerdo, así mismo un 1.9% concluyen que no están ni de acuerdo ni en desacuerdo.

- **Resultados respecto a la funcionalidad - fiabilidad**

Se concluye que la aplicación web para la gestión de requerimientos cumplió con los indicadores de funcionabilidad y usabilidad por la siguiente razón:

- Un 76.9% concluye que están muy de acuerdo en que la aplicación web para la gestión de requerimientos permitió controlar la información de los mismos presentando en los reportes información fiable para el análisis posterior de estos; mientras que un 21.2% concluye que están de acuerdo, así mismo un 1.9% concluyen que no están ni de acuerdo ni en desacuerdo.



CONCLUSIONES

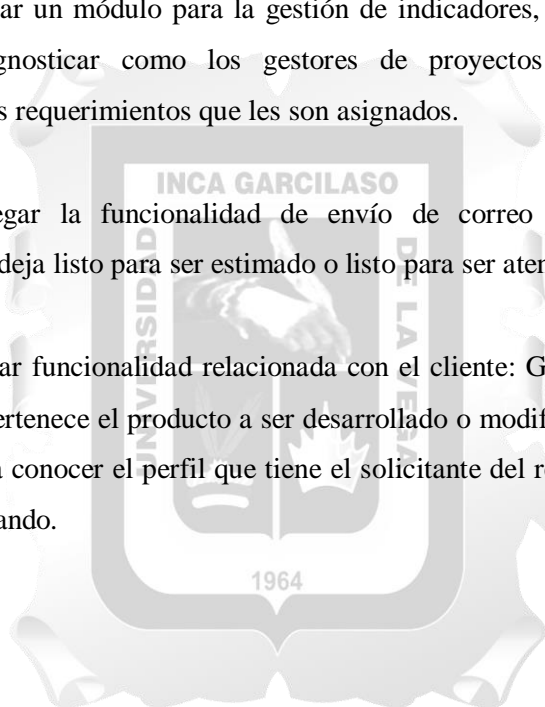
De los objetivos planteados al inicio de este trabajo de investigación, y los correspondientes resultados obtenidos en la aplicación web para la gestión de requerimientos en la empresa FabSF Services, se tienen las siguientes conclusiones.

- La funcionalidad de la aplicación web para la gestión de requerimientos influyó satisfactoriamente en el proceso de gestión de requerimientos, porque a través de los resultados de las encuestas se evidencia que se llegó a cubrir y satisfacer los requerimientos funcionales de los usuarios, características que definen a este indicador de calidad.
- La eficiencia de la aplicación web para la gestión de requerimientos, influye satisfactoriamente en el proceso de gestión de requerimientos, porque a través de los resultados de las encuestas se demostró eficiencia de la aplicación en cuanto a la generación de reportes y tiempos de respuesta en las consultas de información, características que define a este indicador de calidad.
- La usabilidad de la aplicación web para la gestión de requerimientos, influye satisfactoriamente en el proceso de gestión de requerimientos, porque a través del resultado de las encuestas, se demostró que la usabilidad de la aplicación web se enfoca en la navegación intuitiva y sencilla, además muestra claridad del contenido a la visión de sus usuarios, características que define a este indicador de calidad.
- La fiabilidad de la aplicación web para la gestión de requerimientos, influye satisfactoriamente en el proceso de gestión de requerimientos, porque a través del resultado de las encuestas de satisfacción, se demostró la fiabilidad en cuanto al control de acceso que posee la aplicación web, al no presentar fallas en las operaciones realizadas de manera constante, características que define a este indicador de calidad.

RECOMENDACIONES

A continuación se presentan las recomendaciones, con el propósito de optimizar aún más el proceso de gestión de requerimientos, mejorando la solución planteada en el presente trabajo de investigación.

- Se recomienda agregar la funcionalidad de controlar el tiempo de trabajo realizado por cada personal de fábrica, tanto para estimación como para atención de los requerimientos.
- Se recomienda agregar nuevas opciones de reportes en el Módulo de Reportes para poder facilitar a los usuarios la información necesaria para mejorar la toma de decisiones.
- Se recomienda agregar un módulo para la gestión de indicadores, la cual utilice la información registrada para diagnosticar como los gestores de proyectos están llevando a cabo la responsabilidad de los requerimientos que les son asignados.
- Se recomienda agregar la funcionalidad de envío de correo para indicar que posee un requerimiento en bandeja listo para ser estimado o listo para ser atendido.
- Se recomienda agregar funcionalidad relacionada con el cliente: Gestionar área del cliente, para conocer a que área pertenece el producto a ser desarrollado o modificado, así mismo el Gestionar cargo de cliente, para conocer el perfil que tiene el solicitante del requerimiento y con esto saber con quién se está tratando.



REFERENCIAS BIBLIOGRÁFICAS

- [1] Arias, M. (2015). Aprende Programación Web con PHP y MySQL 2a Edición. España: IT Campus Academy.
- [2] Arias, M. (2005). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. En InterSedes: Revista de las Sedes Regionales, VI (10), 1-13. Costa Rica: Universidad de Costa Rica.
- [3] Alaimo, M. (2013). Proyectos ágiles con Scrum: flexibilidad, aprendizaje, innovación y colaboración en contextos complejos. Cdad. Autónoma de Buenos Aires, 1049, Argentina: Ediciones Kleer.
- [4] Carballeira, J. (2016). Desarrollo de aplicaciones con tecnologías web. España: Unión Editorial para la Formación.
- [5] Carranza, Z., Ballester, J. M., Tinoco, A. (2008, noviembre). Fábrica académica de software. Interfases: Revista digital de la Facultad de Ingeniería de Sistemas, 3, 29.
- [6] Caso, N. G. (2004). SCRUM development process. 2-4.
- [7] Cáceres, G. & Montoya, M. (2017). “*Propuesta de una arquitectura empresarial para una fábrica de software* “. Tesis de título profesional. Universidad Peruana de Ciencias aplicadas. Perú.
- [8] De Luca, D. (2011). HTML5. Buenos Aires, Argentina: Fox Andina & Gradi S.A.
- [9] Fernández J., Morales J., Navarro A. (2013). Revisión de metodologías ágiles para el desarrollo de software. Prospectiva. Una nueva visión para la ingeniería, Vol. 11, núm. 2, 10.
- [10] Flórez, L., Grisales, F. (2014). *Formulación de criterios para la selección de metodologías de desarrollo de software*. Universidad Tecnológica de Pereira, Facultad de Ingenierías. Pereira. Colombia: Universidad Tecnológica de Pereira.
- [11] Geertjan W. (2015). Beginning NetBeans IDE. New York, United States: Springer Science Business Media New York.
- [12] Gilfillan, I. (2003). La biblia de MySQL. Madrid, España: Anaya Multimedia.

- [13] Gómez Fuentes, María del Carmen. (2011). Notas del curso: Análisis de requerimientos. Quinto Piso, Col. Ex Hacienda de San Juan de Dios, Del. Tlalpan, C.P. 14787, México D.F.: Publidisa Mexicana S. A. de C.V.
- [14] Herranz, R. (2016). Despegar con Scrum. España: Utópica Informática.
- [15] Icart Isern, M. & Fuentelsaz Gallego, C. & Pulpón Segura, Anna M. (2006). Sujetos de Estudio. En Elaboración y presentación de un proyecto de investigación y una tesina (p. 55). España: Publications I Edicions de la universitat de Barcelona.
- [16] ISO/IEC 9126-1:2001. (2001). *ISO/IEC 9126-1:2001. Software engineering - Product quality - Part 1: Quality model*. Suiza: International Organization for Standardization.
- [17] Kniberg, H. (2007). SCRUM Y XP DESDE LAS TRINCHERAS. Suecia: C4Media Inc.
- [18] Largo, C. & Marin, E. (2005). Guía Técnica para Evaluación de Software. Uruguay: PuntoExe.
- [19] Lerma-Blasco, R., Murcia J., Talón, E. (2013). Aplicaciones web. Madrid, España: McGraw-Hill/Interamericana de España, S.L.
- [20] Niño, J. (2010). Aplicaciones web: Ciclos Formativos. España: Editorial Editex.
- [21] Palacio, J. (2015). *Gestión de proyectos Scrum Manager*. Zaragoza. España: Scrum Manager.
- [22] Pecho, R. F. (2014). *Desarrollo de un sistema de gestión de biblioteca para mejorar la atención en el colegio Mariscal Castilla – El Tambo*. Tesis para optar el título profesional. Universidad Nacional del Centro del Perú, Facultad de Ingeniería de Sistemas. Huancayo - Perú.
- [23] Ponluiza, J. (2016). Diseño e implementación de un sistema web de gestión odontológica para la empresa eléctrica Riobamba S.A. Tesis para optar el título profesional. Escuela superior politécnica de Chimborazo, Facultad de informática y electrónica. Riobamba - Ecuador.
- [24] Pressman, R. (2002). Ingeniería del Software: Un enfoque práctico, 5ta edición. España: McGraw Hill.
- [25] Ramos, A., Ramos, J. (2014). Aplicaciones Web. Madrid, España: Paraninfo.

[26] Sommerville, I. (2005). Ingeniería del Software, 7ª Edición. Madrid, España: Pearson Addison Wesley.

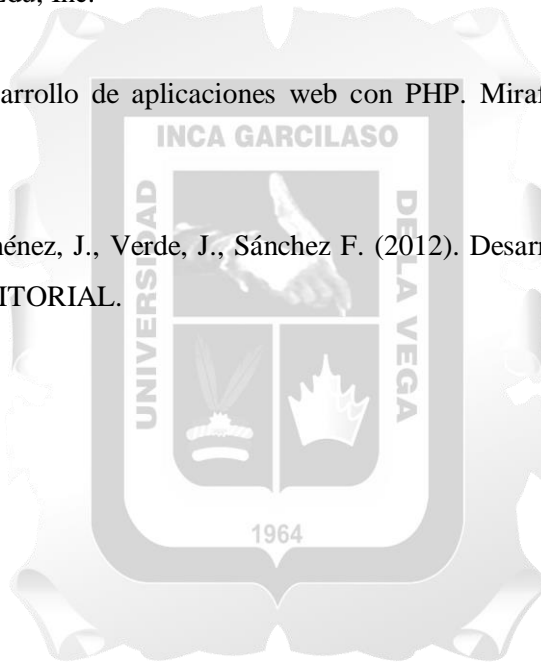
[27] Soto, I., González, P. (2010). Propuesta de un plan para la recuperación de proyectos de software. Teorías, Modelos y Aplicaciones de Matemáticas y Computación, Memorias de la 1ª y 2ª Semana de Computación y Matemáticas Aplicadas SCMA 08 y SCMA 09. Cuajimalpa, México D.F.: UAM Cuajimalpa.

[28] Talledo, José. (2015). Implantación de aplicaciones web en entornos internet, intranet y extranet. España: Ediciones Paraninfo, S.A.

[29] Tridibesh S. (2016). Una guía para el cuerpo de conocimiento de Scrum (Guía SBOK™) - 2016 Edición. Arizona, USA: VMEdU, Inc.

[30] Torres, M. (2014). Desarrollo de aplicaciones web con PHP. Miraflores, Perú: Empresa Editora Macro EIRL.

[31] Vara, J., López, M., Jiménez, J., Verde, J., Sánchez F. (2012). Desarrollo web en entorno servidor. Madrid, España: RA-MA EDITORIAL.



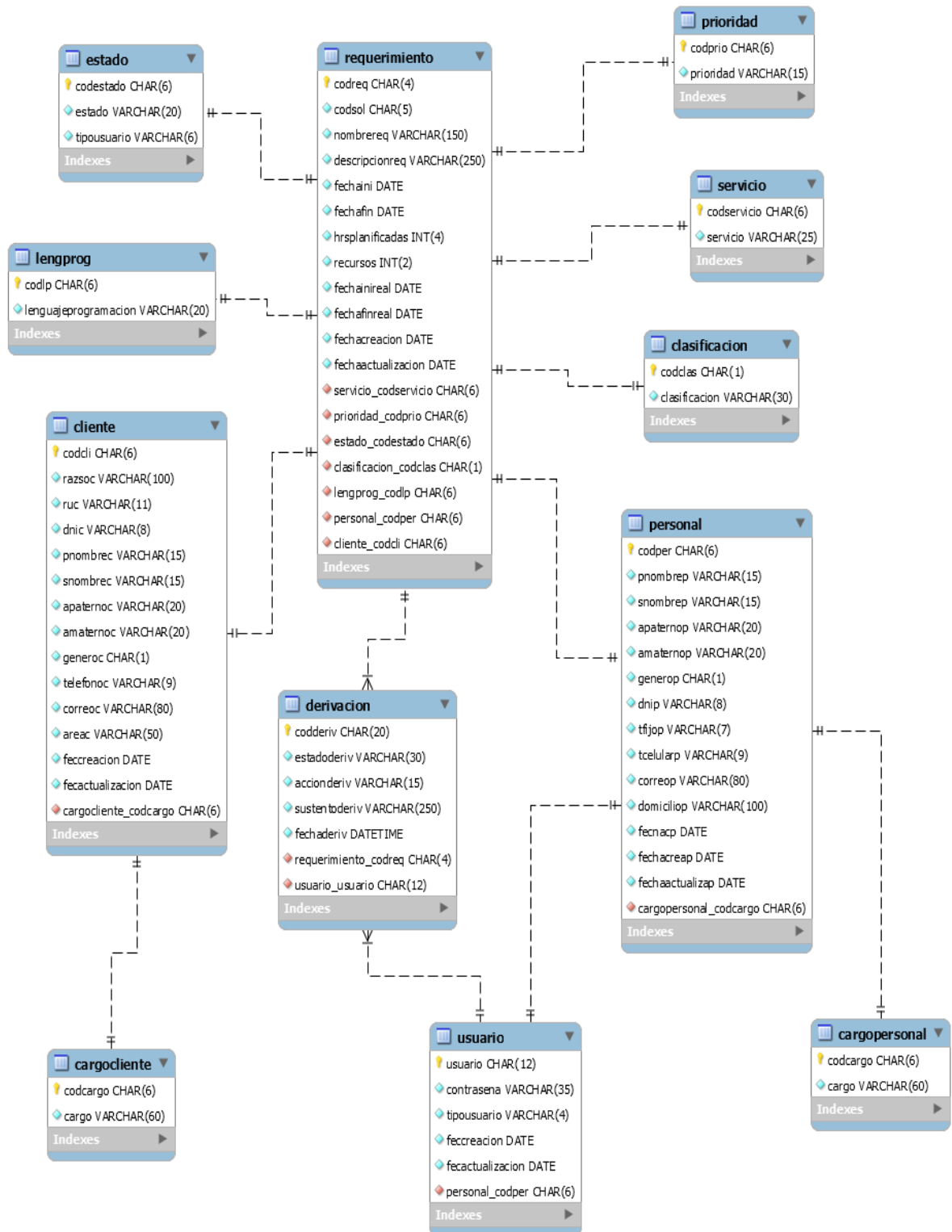
ANEXO I

MATRIZ DE COHERENCIA INTERNA

	PROBLEMAS	OBJETIVOS	HIPÓTESIS	VARIABLES	INDICADORES
GENERAL	¿En qué medida la aplicación web influye en la gestión de requerimientos de software en la empresa FabSF Services?	Determinar la influencia de una aplicación web en la gestión de requerimientos de software en la empresa FabSF Services.	El desarrollo de una aplicación web influye significativamente en la gestión de requerimientos de software en la empresa FabSF Services.	Independiente: Aplicación web	<ul style="list-style-type: none"> • Nivel de funcionalidad • Nivel de usabilidad • Nivel de eficiencia • Nivel de fiabilidad
ESPECÍFICO	¿En qué medida el nivel de funcionalidad de la aplicación web influye en la gestión de requerimientos de software en la empresa FabSF Services?	Determinar la influencia del nivel de funcionalidad de la aplicación web en la gestión de requerimientos de software en la empresa FabSF Services.	El nivel de funcionalidad de la aplicación web influye significativamente en la gestión de requerimientos de software en la empresa FabSF Services.	Dependiente: Gestión de requerimientos de software en la empresa FabSF Services	<ul style="list-style-type: none"> • Número de recursos asignados por requerimiento. • Número de requerimientos atendidos por fábrica de software. • Número de requerimientos en atención por fábrica de software. • Número de derivaciones realizadas por Gestor de Proyectos/Ingeniero de Software/Analista QA.
	¿En qué medida el nivel de usabilidad de la aplicación web influye en la gestión de requerimientos software en la empresa FabSF Services?	Determinar la influencia del nivel de usabilidad de la aplicación web en la gestión de requerimientos de software en la empresa FabSF Services.	El nivel de usabilidad de la aplicación web influye significativamente en la gestión de requerimientos de software en la empresa FabSF Services.		
	¿En qué medida el nivel de eficiencia de la aplicación web influye en la gestión de requerimientos software en la empresa FabSF Services?	Determinar la influencia del nivel de eficiencia de la aplicación web en la gestión de requerimientos de software en la empresa FabSF Services.	El nivel de eficiencia de la aplicación web influye significativamente en la gestión de requerimientos de software en la empresa FabSF Services.		
	¿En qué medida el nivel de fiabilidad de la aplicación web influye en la gestión de requerimientos software en la empresa FabSF Services?	Determinar la influencia del nivel de fiabilidad de la aplicación web en la gestión de requerimientos de software en la empresa FabSF Services.	El nivel de fiabilidad de la aplicación web influye significativamente en la gestión de requerimientos de software en la empresa FabSF Services.		

ANEXO II

MODELO FÍSICO DE BASE DE DATOS



ANEXO III

ENCUESTA DE EVALUACIÓN DE CALIDAD SOBRE LA APLICACIÓN WEB PARA LA GESTIÓN DE REQUERIMIENTOS DE SOFTWARE EN LA EMPRESA FABSF SERVICES

Según el indicador de funcionalidad

1. ¿La aplicación web para la gestión de requerimientos validó las credenciales (usuario y contraseña) ingresadas con éxito?
 - a) Muy de acuerdo
 - b) De acuerdo
 - c) Ni de acuerdo ni en desacuerdo
 - d) Desacuerdo
 - e) Muy en desacuerdo

2. Para usted ¿La aplicación web para la gestión de requerimientos permitió realizar el seguimiento de los requerimientos que son atendidos por la fábrica de software?
 - a) Muy de acuerdo
 - b) De acuerdo
 - c) Ni de acuerdo ni en desacuerdo
 - d) Desacuerdo
 - e) Muy en desacuerdo

3. ¿La aplicación web para la gestión de requerimientos redujo el tiempo de consulta de la información de los requerimientos en la fábrica de software?
 - a) Muy de acuerdo
 - b) De acuerdo
 - c) Ni de acuerdo ni en desacuerdo
 - d) Desacuerdo
 - e) Muy en desacuerdo

Según el indicador de usabilidad

4. En su opinión, ¿Fue fácil consultar la bitácora de las aprobaciones/rechazos de los requerimientos en la aplicación web para la gestión de requerimientos?

- a) Muy de acuerdo
- b) De acuerdo
- c) Ni de acuerdo ni en desacuerdo
- d) Desacuerdo
- e) Muy en desacuerdo

5. A su parecer, ¿La navegación para consultar la información de los requerimientos en la aplicación web para la gestión de requerimientos fue fácil?

- a) Muy de acuerdo
- b) De acuerdo
- c) Ni de acuerdo ni en desacuerdo
- d) Desacuerdo
- e) Muy en desacuerdo

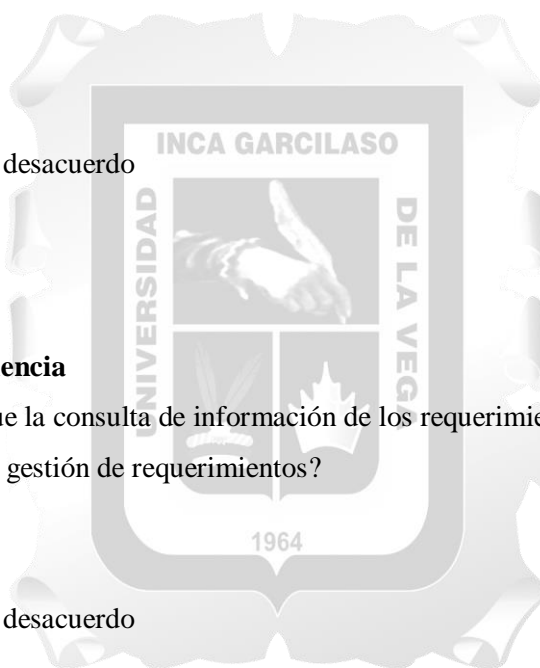
Según el indicador de eficiencia

6. ¿Considera usted que la consulta de información de los requerimientos se realizó con rapidez en la aplicación web para la gestión de requerimientos?

- a) Muy de acuerdo
- b) De acuerdo
- c) Ni de acuerdo ni en desacuerdo
- d) Desacuerdo
- e) Muy en desacuerdo

7. ¿El tiempo de respuesta de la generación del reporte con la información general de los requerimientos en la aplicación web para la gestión de requerimientos fue rápida?

- a) Muy de acuerdo
- b) De acuerdo
- c) Ni de acuerdo ni en desacuerdo
- d) Desacuerdo
- e) Muy en desacuerdo



Según el indicador de fiabilidad

8. Según su opinión, ¿Considera que la información de los requerimientos se controla de manera efectiva al realizar las derivaciones de los mismos?

- a) Muy de acuerdo
- b) De acuerdo
- c) Ni de acuerdo ni en desacuerdo
- d) Desacuerdo
- e) Muy en desacuerdo

Según el indicador de funcionalidad - usabilidad

9. Para usted, ¿La aplicación web para la gestión de requerimientos mostró las opciones e interfaces asociadas a los perfiles correspondientes facilitando la aprobación/rechazo de los requerimientos en la aplicación web para la gestión de requerimientos?

- a) Muy de acuerdo
- b) De acuerdo
- c) Ni de acuerdo ni en desacuerdo
- d) Desacuerdo
- e) Muy en desacuerdo

Según el indicador de funcionalidad - fiabilidad

10. Para usted: ¿La aplicación web para la gestión de requerimientos permitió controlar la información de los mismos presentando en los reportes información fiable para el análisis posterior de estos?

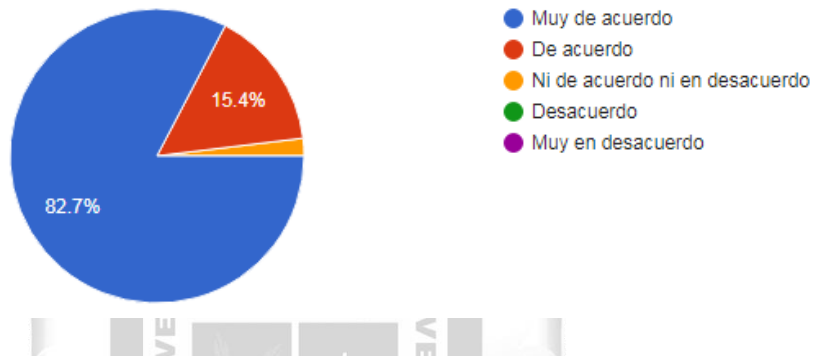
- a) Muy de acuerdo
- b) De acuerdo
- c) Ni de acuerdo ni en desacuerdo
- d) Desacuerdo
- e) Muy en desacuerdo

ANEXO IV

RESULTADOS DE LA ENCUESTA DE EVALUACIÓN DE CALIDAD SOBRE LA APLICACIÓN WEB PARA LA GESTIÓN DE REQUERIMIENTOS DE SOFTWARE EN LA EMPRESA FABSF SERVICES

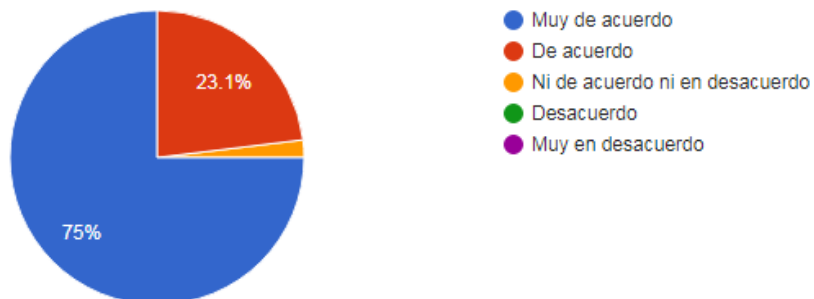
1. Según su opinión, ¿La aplicación web para la gestión de requerimientos validó las credenciales (usuario y contraseña) ingresadas con éxito?

52 respuestas



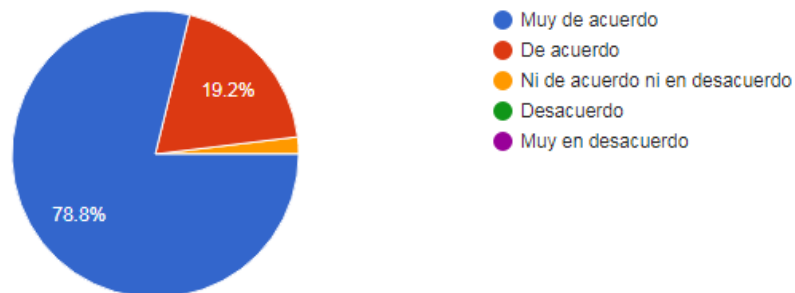
2. Para usted, ¿La aplicación web para la gestión de requerimientos permitió realizar el seguimiento de los requerimientos que son atendidos por la fábrica de software?

52 respuestas



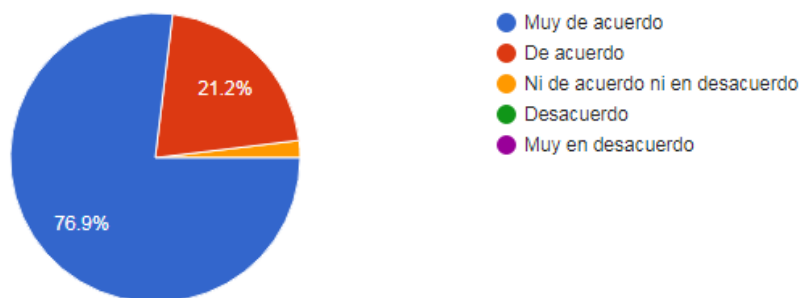
3. Para usted, ¿La aplicación web para la gestión de requerimientos redujo el tiempo de consulta de la información de los requerimientos en la fábrica de software?

52 respuestas



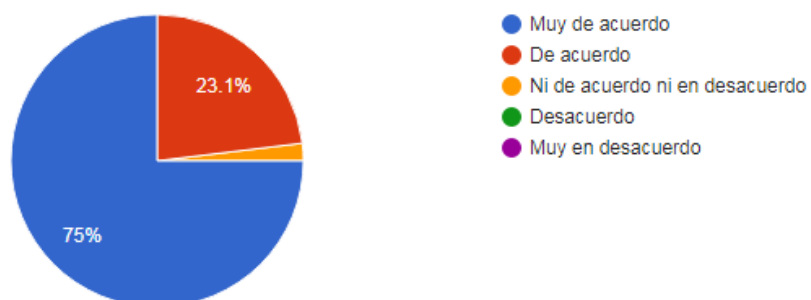
4. En su opinión, ¿Fue fácil consultar la bitácora de las aprobaciones/rechazos de los requerimientos en la aplicación web para la gestión de requerimientos?

52 respuestas



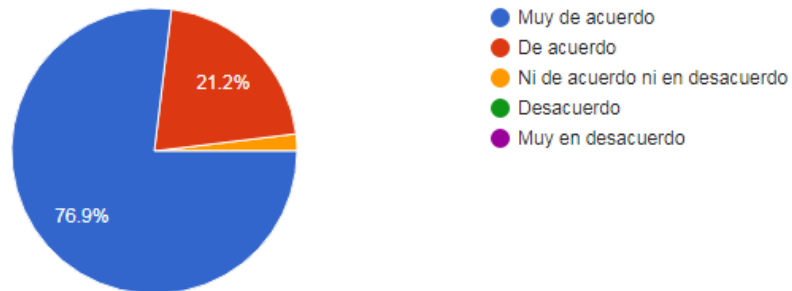
5. A su parecer, ¿La navegación para consultar la información de los requerimientos en la aplicación web para la gestión de requerimientos fue fácil?

52 respuestas



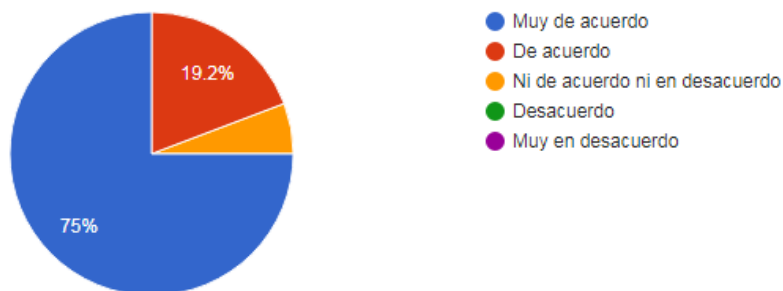
6. ¿Considera usted que la consulta de información de los requerimientos se realizó con rapidez en la aplicación web para la gestión de requerimientos?

52 respuestas



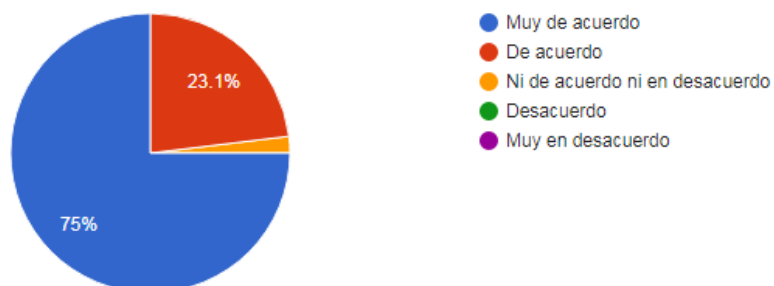
7. ¿El tiempo de respuesta de la generación del reporte con la información general de los requerimientos en la aplicación web para la gestión de requerimientos fue rápida?

52 respuestas



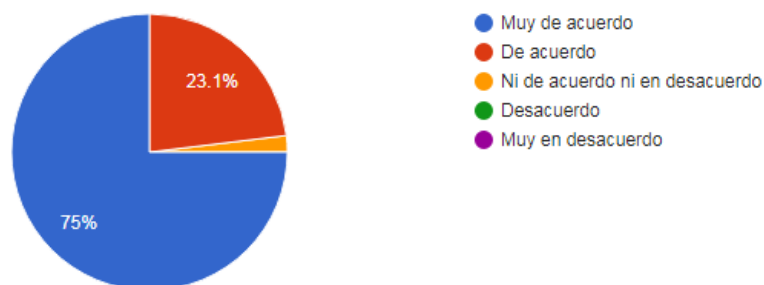
8. Según su opinión, ¿Considera que la información de los requerimientos se controla de manera efectiva al realizar las derivaciones de los mismos?

52 respuestas



9. Para usted, ¿La aplicación web para la gestión de requerimientos mostró las opciones e interfaces asociadas a los perfiles correspondientes facilitando la aprobación/rechazo de los requerimientos en la aplicación web para la gestión de requerimientos?

52 respuestas



10. Para usted: ¿La aplicación web para la gestión de requerimientos permitió controlar la información de los mismos presentando en los reportes información fiable para el análisis posterior de estos?

52 respuestas

