



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y  
URBANISMO**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

**TESIS**

**DESARROLLO DE UN MODELO DE PROCESOS AD  
HOC PARA EL DESARROLLO DE SOFTWARE POR  
LICENCIA PARA UNA MYPE DE SERVICIOS DE TI  
BASADO EN ISO/IEC 29110**

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO  
DE SISTEMAS**

**Autor:**

**Bach. Carrasco Borda Aparicio**

**ORCID: <https://orcid.org/0000-0002-8792-0154>**

**Asesor:**

**Mg. Sialer Rivera Maria Noelia**

**ORCID: <https://orcid.org/0000-0002-2266-2051>**

**Línea de Investigación:**

**Infraestructura, Tecnología y Medio Ambiente**

**Pimentel - Perú**

**2022**

## **APROBACIÓN DEL JURADO**

### **DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110**

---

**Bach. Carrasco Borda Aparicio**  
**Autor**

---

**Mg. Sialer Rivera María Noelia**  
**Asesor**

---

**Mg. Aguinaga Tello Juan**  
**Presidente de Jurado**

---

**Mg. Bances Saavedra David E.**  
**Secretario de Jurado**

---

**Mg. Mejía Cabrera Heber Iván**  
**Vocal de Jurado**

## **Dedicatoria**

Ofrezco el trabajo realizado a Dios, por ofrecerme la existencia y condescender el haber logrado hasta este instante, tan significativo de la vida y formación profesional. A mi madre, por ser el cimiento, manifestar perenemente su apego y soporte incondicional. A mi hermana Reina que perpetuamente estuvo con sus oraciones desde España por el bien de salud de la familia. A mi esposa por alentarme alcanzar la meta con su apoyo incesante.

*ACB*

## **Agradecimientos**

Agradecimiento y gratitud a mis docentes de la Universidad Señor de Sipán que durante el trayecto de mi carrera universitaria me brindaron su apoyo incondicional y académico, y gracias a sus sapiencias hoy puedo sentirme próspero y contento. De igual forma, agradezco a mi docente de Tesis, que gracias a sus consejos y correcciones hoy puedo culminar este trabajo, a pesar de las dificultades que se presentaron durante el desarrollo de la investigación.

*ACB*

## Resumen

Esta investigación se centra en el desarrollo de un modelo de procesos ad hoc para el desarrollo de software por licencia en una MYPE de servicios de TI con sede en la ciudad de Trujillo. Ello, ante la problemática de que, las MYPES de dicho sector no poseen un modelo que les sirva de hoja de ruta para llevar a cabo sus procesos constructivos de software ya que, los modelos existentes son accesibles solo a grandes empresas, siendo ajenas a las pequeñas organizaciones tales como el caso de estudio el cual es una MYPE, ocasionando dificultades ya que muchas de éstas empresas no suelen contar con procesos definidos, sistemas de control, etcétera; generando así dificultades para adaptarse al ambiente empresarial en el que se desenvuelven. Por esta razón, se desarrolló un modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI basado en la ISO/IEC 29110. Para ello, primeramente, se caracterizó el perfil básico de dicha ISO en mención; luego, se elaboró un diagnóstico base del estado actual del proceso constructivo de software en una MYPE trujillana de servicios de TI; después, se diseñó el modelo el cual consta de dos (02) procesos bien definidos: Proceso de Gestión de Proyecto (03 actividades) y Proceso de Implementación de Software (04 actividades), fundamentado en la ISO en mención; posteriormente, se validó dicho modelo mediante juicio de expertos y; finalmente, se llevó a cabo una prueba piloto del modelo propuesto en el caso de estudio. Los resultados obtenidos mostraron que, el modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110 contó con un nivel de aceptación de 92 en una escala del 1 al 100, por lo que se consideró que tiene una valoración considerada como “Muy Buena”. Se concluyó que, dicho modelo mejoró el proceso de desarrollo de software en el caso de estudio, logrando un aminoramiento en los defectos de los proyectos desarrollados y logrando una satisfacción de clientes del 100% posterior a la puesta en marcha de dicho modelo.

**Palabras Clave:** ISO/IEC 29110, mejora de los procesos de software, calidad de procesos de software, Gestión de Proyectos, Implementación de Software, entidades muy pequeñas.

## **Abstract**

This research focuses on the development of an ad hoc process model for software development by license in an SME of IT services based in the city of Trujillo. This is due to the problem that SME s in this sector do not have a model that serves as a roadmap to carry out their software construction processes, since existing models are only accessible to large companies, being alien to small organizations such as the case study which is an SME, causing difficulties since many of these companies do not usually have defined processes, control systems, etc., thus generating difficulties in adapting to the business environment in which they operate. For this reason, an ad hoc process model was developed for the development of licensed software for an IT services SME based on ISO/IEC 29110. For this purpose, first, the basic profile of the ISO was characterized; then, a basic diagnosis of the current state of the software development process in a SME of IT services in Trujillo was elaborated; then, the model was designed, which consisted of two (02) well-defined processes: Project Management Process (03 activities) and Software Implementation Process (04 activities), based on the aforementioned ISO; then, the model was validated by expert judgment and finally, a pilot test of the proposed model was carried out in the case study. The results obtained showed that the ad hoc process model for the development of licensed software for an IT services SME based on ISO/IEC 29110 had an acceptance level of 92 on a scale of 1 to 100, so it was considered to have a rating considered as "Very Good". It was concluded that this model improved the software development process in the case study, achieving a reduction in the defects of the projects developed and achieving 100% customer satisfaction after the implementation of this model.

**Keywords:** ISO/IEC 29110, Software Process Improvement, Software process quality, Project Management, Software Implementation, Very Small Entities.

## ÍNDICE

I.	INTRODUCCIÓN .....	14
1.1.	Realidad Problemática. ....	14
1.2.	Trabajos previos.....	17
1.3.	Teorías relacionadas al tema. ....	29
1.4.	Formulación del Problema. ....	45
1.5.	Justificación e importancia del estudio. ....	45
1.6.	Hipótesis. ....	45
1.7.	Objetivos. ....	45
1.7.1.	Objetivo general.....	45
1.7.2.	Objetivos específicos.....	45
II.	MATERIAL Y MÉTODO .....	47
2.1.	Tipo y Diseño de Investigación. ....	47
2.2.	Población y muestra.....	47
2.3.	Variables, Operacionalización.....	48
2.4.	Técnicas e instrumentos de recolección de datos, validez y confiabilidad. 51	
2.5.	Procedimiento de análisis de datos.....	52
2.6.	Criterios éticos. ....	54
2.7.	Criterios de Rigor Científico. ....	54
III.	RESULTADOS. ....	56
3.1.	Resultados en Tablas y Figuras.....	56
3.2.	Discusión de resultados. ....	69
3.3.	Aporte práctico.....	72
IV.	CONCLUSIONES Y RECOMENDACIONES.....	125
4.1.	Conclusiones.....	125

4.2. Recomendaciones.....	126
REFERENCIAS.....	127
ANEXOS.....	132



## ÍNDICE DE FIGURAS

<i>Figura 1, Resultados del Indicador - Nivel de Claridad del Modelo.....</i>	<i>56</i>
<i>Figura 2, Resultados del Indicador - Nivel de Objetividad del Modelo .....</i>	<i>57</i>
<i>Figura 3, Resultados del Indicador - Nivel de Actualidad del Modelo.....</i>	<i>58</i>
<i>Figura 4, Resultados del Indicador - Nivel de Organización del Modelo.....</i>	<i>59</i>
<i>Figura 5, Resultados del Indicador - Nivel de Suficiencia del Modelo.....</i>	<i>60</i>
<i>Figura 6, Resultados del Indicador - Nivel de Intencionalidad del Modelo.....</i>	<i>61</i>
<i>Figura 7, Resultados del Indicador - Nivel de Consistencia del Modelo.....</i>	<i>62</i>
<i>Figura 8, Resultados del Indicador - Nivel de Coherencia del Modelo.....</i>	<i>63</i>
<i>Figura 9, Resultados del Indicador - Nivel de Metodología del Modelo .....</i>	<i>64</i>
<i>Figura 10, Resultados del Indicador - Nivel de Pertinencia del Modelo.....</i>	<i>65</i>
<i>Figura 11, Resultados de la Variable Independiente.....</i>	<i>66</i>
<i>Figura 12, Resultados del Indicador - Cantidad de defectos.....</i>	<i>67</i>
<i>Figura 13, Resultados del Indicador - Clientes satisfechos.....</i>	<i>68</i>
<i>Figura 14, Caracterización “Planificación del proyecto” .....</i>	<i>73</i>
<i>Figura 15, Caracterización “Ejecución del plan del proyecto” .....</i>	<i>73</i>
<i>Figura 16, Caracterización “Evaluación y control del proyecto” .....</i>	<i>74</i>
<i>Figura 17, Caracterización “Cierre del proyecto”.....</i>	<i>74</i>
<i>Figura 18, Caracterización “Inicio de la implementación del software” .....</i>	<i>75</i>
<i>Figura 19, Caracterización “Análisis de requisitos software” .....</i>	<i>75</i>
<i>Figura 20, Caracterización “Arquitectura de software y diseño detallado” .....</i>	<i>76</i>
<i>Figura 21, Caracterización “Construcción de software” .....</i>	<i>76</i>
<i>Figura 22, Caracterización “Integración de software y pruebas” .....</i>	<i>77</i>
<i>Figura 23, Caracterización “Entrega del producto” .....</i>	<i>77</i>
<i>Figura 24, Procesos del modelo ad hoc propuesto.....</i>	<i>90</i>
<i>Figura 25, Actividades del Proceso de Gestión de Proyectos .....</i>	<i>91</i>
<i>Figura 26, Diagrama SIPOC de “Planeación del Proyecto”.....</i>	<i>92</i>
<i>Figura 27, PGP-ACT-01: Planeación del Proyecto.....</i>	<i>93</i>
<i>Figura 28, Diagrama SIPOC de “Despliegue de Plan del Proyecto”.....</i>	<i>95</i>
<i>Figura 29, PGP-ACT-02: Despliegue de Plan del Proyecto .....</i>	<i>96</i>
<i>Figura 30, Diagrama SIPOC de “Evaluación y monitoreo del Proyecto” .....</i>	<i>98</i>
<i>Figura 31, PGP-ACT-03: Evaluación y Monitoreo del Proyecto .....</i>	<i>99</i>

<i>Figura 32, Actividades del Proceso de Implementación de Software.....</i>	<i>100</i>
<i>Figura 33, Diagrama SIPOC de “Análisis y requerimientos de software”.....</i>	<i>101</i>
<i>Figura 34, PIS-ACT-01: Análisis de Requerimientos de Software.....</i>	<i>102</i>
<i>Figura 35, Diagrama SIPOC de “Diseño y arquitectura de software”.....</i>	<i>103</i>
<i>Figura 36, PIS-ACT-02: Diseño y Arquitectura del Software.....</i>	<i>104</i>
<i>Figura 37, Diagrama SIPOC de “Construcción del software”.....</i>	<i>106</i>
<i>Figura 38, PIS-ACT-03: Construcción del Software.....</i>	<i>107</i>
<i>Figura 39, Diagrama SIPOC de “Entrega del software”.....</i>	<i>108</i>
<i>Figura 40, PIS-ACT-04: Entrega del Software.....</i>	<i>109</i>
<i>Figura 41, Indicadores considerados en la Ficha de Juicio de Expertos.....</i>	<i>110</i>
<i>Figura 42, Logo de NEDLEY SUPPORT S.A.C.....</i>	<i>113</i>
<i>Figura 43, Página Web Empresarial NEDLEY SUPPORT S.A.C.....</i>	<i>114</i>
<i>Figura 44, Valores empresariales de NEDLEY SUPPORT S.A.C.....</i>	<i>114</i>
<i>Figura 45, Diagrama de Gantt del Proyecto 04: Sistema de Mesa de Ayuda - Soporte Técnico.....</i>	<i>116</i>
<i>Figura 46, Diagrama de Gantt del Proyecto 05: Sistema Web de Recursos Humanos (POST-TEST).....</i>	<i>117</i>
<i>Figura 47, Diagrama de Gantt del Proyecto 06: Sistema de Mesa de partes digital (POST-TEST).....</i>	<i>118</i>
<i>Figura 48, Gráfico de esfuerzo por ciclo de pruebas piloto ejecutado en la empresa.....</i>	<i>123</i>
<i>Figura 49, Gráfico de porcentaje de errores por ciclo de pruebas.....</i>	<i>124</i>

## ÍNDICE DE TABLAS

Tabla 1. <i>Operacionalización de la Variable Independiente</i> .....	49
Tabla 2. <i>Operacionalización de la Variable Dependiente</i> .....	50
Tabla 3. <i>Expertos para validación de modelo</i> .....	51
Tabla 4. <i>Resultados del Indicador - Nivel de Claridad</i> .....	56
Tabla 5. <i>Resultados del Indicador - Nivel de Objetividad</i> .....	57
Tabla 6. <i>Resultados del Indicador - Nivel de Actualidad</i> .....	58
Tabla 7. <i>Resultados del Indicador - Nivel de Organización</i> .....	59
Tabla 8. <i>Resultados del Indicador - Nivel de Suficiencia</i> .....	60
Tabla 9. <i>Resultados del Indicador - Nivel de Intencionalidad</i> .....	61
Tabla 10. <i>Resultados del Indicador - Nivel de Consistencia</i> .....	62
Tabla 11. <i>Resultados del Indicador - Nivel de Coherencia</i> .....	63
Tabla 12. <i>Resultados del Indicador - Nivel de Metodología</i> .....	64
Tabla 13. <i>Resultados del Indicador - Nivel de Pertinencia</i> .....	65
Tabla 14. <i>Resultados del Indicador - Cantidad de defectos</i> .....	67
Tabla 15. <i>Resultados del Indicador - Porcentaje de Clientes Satisfechos</i> .....	68
Tabla 16. <i>Ficha de rol “Analista”</i> .....	77
Tabla 17. <i>Ficha de rol “Cliente”</i> .....	78
Tabla 18. <i>Ficha de rol “Diseñador”</i> .....	78
Tabla 19. <i>Ficha de rol “Programador”</i> .....	78
Tabla 20. <i>Ficha de rol “Gestor de Proyecto”</i> .....	79
Tabla 21. <i>Ficha de rol “Líder Técnico”</i> .....	79
Tabla 22. <i>Ficha de rol “Equipo de Trabajo”</i> .....	79
Tabla 23. <i>Diagnóstico PGP-ACT-01: Planeación del Proyecto</i> .....	80
Tabla 24. <i>Diagnóstico PGP-ACT-02: Despliegue de Plan del Proyecto</i> .....	81
Tabla 25. <i>Diagnóstico PGP-ACT-03: Evaluación y Monitoreo del Proyecto</i> .....	82
Tabla 26. <i>Diagnóstico PIS-ACT-01: Análisis de Requerimientos de Software</i> .....	83
Tabla 27. <i>Diagnóstico PIS-ACT-02: Diseño y Arquitectura del Software</i> .....	84
Tabla 28. <i>Diagnóstico PIS-ACT-03: Construcción del Software</i> .....	85
Tabla 29. <i>Diagnóstico PIS-ACT-04: Entrega del Software</i> .....	87
Tabla 30. <i>Matriz comparativo procesos propuestos vs procesos ISO/IEC 29110 vs procesos actuales</i> .....	89

Tabla 31. <i>Formato de registro de tareas por realizar</i> .....	91
Tabla 32. <i>Formato de relación de registro de empleados</i> .....	95
Tabla 33. <i>Formato de estimación de costo, tiempo y esfuerzo por tarea</i> .....	98
Tabla 34. <i>Formato de registro de requerimientos por realizar</i> .....	100
Tabla 35. <i>Expertos para validación de propuesta</i> .....	111
Tabla 36. <i>Expertos para validación del modelo propuesto</i> .....	112
Tabla 37. <i>Cronograma de construcción de proyectos</i> .....	115
Tabla 38. <i>Cuadro comparativo de los proyectos de prueba piloto del modelo propuesto basado en la norma ISO/IEC 29110</i> .....	119
Tabla 39. <i>Esfuerzo involucrado en cada actividad de pruebas ejecutados de los procesos de la empresa</i> . .....	122
Tabla 40. <i>Porcentaje de defectos encontrados tras la ejecución de cada proceso de prueba</i> .....	124

## ÍNDICE DE ANEXOS

Anexo 1. Resolución de aprobación del proyecto de investigación .....	132
Anexo 2. Carta de aceptación de la institución para la recolección de datos.....	134
Anexo 3. Matriz de consistencia lógica .....	135
Anexo 4. Instrumento de recolección de datos - Ficha de Juicio de Expertos .....	136
Anexo 5. Instrumento de recolección de datos - Ficha de Observación .....	138
Anexo 6. Instrumento de recolección de datos - Cuestionario.....	140
Anexo 7. Herramientas empleadas para el desarrollo del modelo ad hoc.....	141
Anexo 8. Resultados individuales de Juicio de Expertos.....	144
Anexo 9. Resultados individuales de Ficha de Observación.....	153
Anexo 10. Proyectos evaluados Pre Test.....	165
Anexo 11. Proyectos evaluados Post Test.....	168
Anexo 12. Diagrama de Gantt - Proyecto 04: Sistema de Mesa de Ayuda - Soporte Técnico (POST-TEST).....	171
Anexo 13. Diagrama de Gantt - Proyecto 05: Sistema Web de Recursos Humanos (POST-TEST) .....	173
Anexo 14. Diagrama de Gantt - Proyecto 06: Sistema de Mesa de partes digital (POST-TEST).....	176
Anexo 15. Evidencias gráficas de prueba piloto de los procesos desarrollados del modelo.....	180
Anexo 16. Evidencias fotográficas .....	193

## **I. INTRODUCCIÓN**

### **1.1. Realidad Problemática.**

Durante décadas, la funcionalidad de software se requirió principalmente en los productos para respaldar las soluciones de hardware, para automatizar procesos mecánicos o eléctricos, sin ofrecer ningún otro valor perceptible. Las empresas desarrollaron software como un costo necesario sin explorar el valor del software como tal. Sin embargo, hoy en día, los sistemas que alguna vez se construyeron exclusivamente a partir de componentes de hardware como, por ejemplo, automóviles y electrodomésticos conectados, hoy en día contienen una funcionalidad de software que les permite conectarse a Internet, intercambiar información con otros sistemas, adaptarse a las necesidades de las organizaciones y mejorar con el tiempo.

Actualmente, el software de la mayoría de los productos permite a las instituciones de todo el mundo lograr más, de lo que antes había sido posible. Rápidamente, el software es la estrategia más efectiva en sistemas que anteriormente existían sin él. Esto lo podemos ver que durante el año 2017 en Perú se vendió USD 98.7 millones solo en software por licencia y en el 2019, el valor ascendió a USD 273.89 millones, siendo Microsoft el líder del mercado de software por licencia, con una participación de mercado del 15.2% (Ochoa, 2019).

Este crecimiento de la demanda de los sistemas informáticos ha creado oportunidades para las instituciones que desarrollan software. Especialmente para entidades muy pequeñas (MYPEs); porque en gran medida, muchas de estas son las productoras de software para otras medianas y grandes compañías, todo ello a manera de un eslabón de una cadena de fabricación que brinda servicios y productos para cumplir las necesidades del mercado. Esta situación pone de relieve la progresiva necesidad de optimizar sus procesos de desarrollo de sistemas, para su mantenibilidad en el negocio por medio del desarrollo tanto de los productos como servicios de sistemas de calidad con recursos limitados.

En este panorama, las empresas pequeñas tienen el desafío de manejar múltiples proyectos de pequeña escala y de rápido movimiento que permiten poco espacio para procesos de gestión difíciles de manejar, pero que aún requieren un proceso de seguimiento eficiente y sencillo (Wynn & Clarkson, 2018).

Además, las MYPEs tienen características únicas, que hacen que sus estilos comerciales sean diferentes a organizaciones más grandes, y debido al pequeño número de individuos responsables en el proyecto y la organización, la mayoría de los procesos de gestión se realizan de forma informal y menos documentada (O'Connor & Laporte, 2012). También se reconoce la falta de adopción de estándares en empresas pequeñas, ya que la percepción es que han sido desarrollados para compañías grandiosas y de mayor magnificencia y que son constructoras de software y sin tener en cuenta la pequeña organización (O'Connor & Coleman, 2009). Dado que las empresas de software más pequeñas tienen menos recursos en términos de personal y dinero, existen muchos desafíos (Basri, Abubakar, Almomani, & Thangiah, 2020).

Las MYPEs no están adoptando estándares existentes o modelos de mejores prácticas comprobados porque perciben los estándares como desarrollados por grandes organizaciones y orientados a grandes organizaciones, provocando así el debate de que, en términos de número de empleados, el tamaño sí importa (Coleman, 2016). Los estudios las percepciones negativas de los estándares del modelo de proceso son impulsadas principalmente por juicios sobre costes, control burocrático y documentaciones (Wing, Andrew, & Petkov, 2017). Además, se ha informado que las PYMEs invisten inconvenientes para integrar las reglas con sus requerimientos de negocio y para fundamentar la diligencia de las normas internacionales en sus labores operacionales (Sánchez, Colomo, de Amescua, & O'Connor, 2016). Gran parte de estas PYMEs no pueden permitir los recursos para, o ver un beneficio neto en establecer procesos de software como definidos por las normas actuales y modelos de madurez (Basri, Abubakar, Almomani, & Thangiah, 2020).

En consecuencia, una nueva medida de ISO/IEC 29110 “Perfiles de ciclo de vida para compañías pequeñas”, tiene como fin satisfacer las necesidades de las MYPEs. El fin fundamental de este nuevo estándar es para ayudar y alentar a las organizaciones de software muy pequeñas a evaluar y mejorar el proceso del software y se prevé que este nuevo estándar podría alentar y ayudar a las organizaciones poco longevas de software a evaluar su proceso de desarrollo de software.

Se inició el enfoque utilizado para desplegar la ISO/IEC 29110 con los estándares internacionales preexistentes, como el estándar del periodo de vida del software en la ISO/IEC/IEEE 12207 y el estándar de documentación ISO/IEC/IEEE 15289. El grupo de trabajo que está detrás del desarrollo de esta norma aboga por el uso de proyectos piloto como un medio para acelerar la adopción y utilización de ISO / IEC 29110 por MYPEs (Laporte & O'Connor, 2017).

Pero, a pesar del notable esfuerzo para hacer que ISO / IEC 29110 sea más aplicable a las MYPEs que sus predecesores, se requiere tiempo y recursos considerables para comprender y aplicar el estándar; y todavía es necesario y a menudo requiere la intervención de consultores de procesos de software. Es más, si la certificación es el objetivo, lo que es el caso de muchas MYPEs, los registros de la aplicación sistemática de la norma deben mantenerse y exhibirse a pedido de las autoridades de certificación. Por tanto, la barrera de la adopción sigue siendo alta (O'Connor & Laporte, 2017).

Para paliar las dificultades de aplicar la política de ISO / IEC 29110, se ha introducido el conocimiento de paquete de implementación. Los paquetes de implementación son documentos adicionales, tarjetas de tareas, descripciones detalladas del proceso y plantillas. Aunque ellos mejoran la comprensión de la norma, son de poca ayuda para la realización concreta de modelos de referencia, y no garantizan el cumplimiento en todo el ciclo vital del proyecto. Además, los modelos de referencia aún deben adaptarse a las necesidades



operativas determinadas de la empresa pequeña que los adopta (Laporte, Munoz, Miranda, & O'Connor, 2018).

Según INEI (2019), en el Perú, existen 54,725 MYPEs dedicadas al software y comunicaciones, que representan solo el 2.3% del total de MYPEs existentes; menos del 1% de empresas son dedicadas al desarrollo de software y son enfocadas al desarrollo de ERP a medida y de licencia; generalmente estas empresas constan de profesionales individuales que desarrollan software (Diario Gestión, 2016). Según estos datos, las MYPEs de software aún tienen potencial de crecimiento y es de trascendental calidad que estas describan con un método de calidad que asegure que sus producciones y servicios satisfagan las necesidades de sus clientes.

En este documento, se propuso desarrollar un modelo de procesos ad hoc para abordar los problemas anteriores aplicando configuraciones de flujo de trabajo, conceptos y tecnología. Para ello, se modeló dicho modelo basándose en el perfil básico acorde con el planteamiento ofrecido por la ISO/IEC 29110 como un único, integrado, flujo de trabajo. Asimismo, se evidencia un flujo de trabajo que muestra las similitudes y las comparaciones entre el modelo propuesto, las actividades constructivas de software en la MYPE caso de estudio y los procesos esbozados en la ISO en mención. Se realizó pidiendo al usuario que responda una serie de preguntas, cuyas respuestas se utilizó para derivar automáticamente un flujo de trabajo correcto y compatible con ISO / IEC 29110, que se pudo integrar en motores de flujo de trabajo generalizados.

## **1.2. Trabajos previos.**

Castro et al. (2021), realizaron la investigación, *Proposal for the implementation of a quality standard for software development projects of a small company*, en la Universidad Peruana de Ciencias Aplicadas, en Lima - Perú. La construcción de software se ha constituido en una actividad de gran relevancia en cuanto al mercado de la ingeniería del software, por sobre otras actividades, en las que destacan las pequeñas organizaciones, quienes ocupan más del 90% de la totalidad de compañías que se dedican a este rubro desarrollador de software,

empero, existe una brecha en cuanto a la usanza de modelos o estándares que les permitan llevar a cabo un correcto proceso constructivo de software. Por esta razón, presentaron la implementación de un modelo para el desarrollo de software ajustado a la realidad de las microempresas peruanas fundamentado en el estándar ISO/IEC 29110 en su perfil básico, el cual fue seleccionado previamente de entre cuatro (04) modelos, el cual consideró cuatro (04) actividades para el proceso de Gestión de Proyectos y seis (06) actividades para el proceso de Implementación de Software, todas ellas modeladas en Bizagi Modeler e introduciendo, además de ello, documentación, herramientas, cronograma, roles y evaluación financiera para el modelo definido. Los resultados obtenidos mostraron que, en el análisis previo realizado a los dos (02) procesos del modelo, se hallaron brechas de cumplimiento en consideración a la norma ISO/IEC 29110, con un 10% de cumplimiento en Gestión de Proyectos y un 6% de cumplimiento en Implementación de Software. Se concluyó que, el modelo para el desarrollo de software ajustado a la realidad de las microempresas peruanas fundamentado en el estándar ISO/IEC 29110 en su perfil básico permite un cumplimiento mayor al 55% en ambos procesos constructivos de software.

Gunawan & Budiardjo (2021), realizaron la investigación, *A Quest of Software Process Improvements in DevOps and Kanban: A Case Study in Small Software Company*, en la Universitas Indonesia, en Java, Indonesia. Un buen proceso de software mejora los productos de software, empero, en el caso de las pequeñas organizaciones, el desarrollo de software es una cuestión de supervivencia debido a sus limitados recursos para desarrollar los mismos, verbigracia, XYZ Company, la cual es una pequeña organización que adoptó Kanban y DevOps y se enfrentó a retrasos en la entrega de software, por lo que requiere de un modelo adecuado. Por esta razón, presentaron un modelo de mejora fundamentado en el estándar ISO/IEC 29110 en su perfil básico, considerando cuatro (04) actividades para el proceso de Gestión de Proyectos y seis (06) actividades para el proceso de Implementación de Software, el cual fue posteriormente validado mediante el Método Delphi. Es necesario recomendar las mejoras del proceso de software para resolver este problema. Las mejoras

del proceso de software son el resultado de la medición y el análisis de los niveles de madurez utilizando el marco ISO 29110 en un estudio cualitativo. A continuación, se analizan mediante las herramientas de Lean Six Sigma, a saber, el análisis de las deficiencias, el análisis de la causa raíz y el análisis de Pareto. El método Delphi los validó y dio lugar a 18 recomendaciones de mejora en cuatro ámbitos: (a) producto, (b) personas, (c) tecnología y (d) proceso. Las mejoras abarcan dos procesos principales dentro del desarrollo de software, a saber: (a) Gestión de proyectos (PM) y (b) Implementación de software (SI). La empresa XYZ o cualquier empresa de software ágil podría adoptar las 18 recomendaciones de mejora para mejorar el proceso y la calidad del software.

Huapaya et al. (2021), realizaron la investigación, *Proposal for the implementation of the ISO/IEC 29110 part 5 - 2011 standard (basic profile) to improve the project management and software implementation processes of a Peruvian software development company*, en la Universidad Peruana de Ciencias Aplicadas, en la ciudad de Lima, Perú. La empresa Strategic Decision Consulting SAC es una pequeña compañía que brinda asesoría y consultoría en cuanto al desarrollo de software a sus clientes en la ciudad capital, logrando haber crecido exponencialmente durante el último quinquenio, empero, aún hay una brecha en cuanto al formalismo de cada uno de sus procesos constructivos de software, lo que podría conllevar pérdida de clientes de darse una ineficiente gestión de dichos procesos. Por esta razón, presentó una implementación del estándar ISO/IEC TR 29110-5-1-2:2011 en su perfil básico, considerando cuatro (04) actividades para el proceso de Gestión de Proyectos y seis (06) actividades para el proceso de Implementación de Software, todas ellas modeladas en Bizagi Modeler e introduciendo, además de ello, indicadores para ambos procesos definidos. Los resultados obtenidos mostraron que, existía una brecha en el proceso de la Gestión de los Proyectos (45%) y en el proceso de implementación de Software (40%) que se venía desarrollando en la compañía caso de estudio. Se concluyó que, el modelo basado en el estándar ISO/IEC TR 29110-5-1-2:2011 en su perfil básico permitiría el aminoramiento en cuanto a costes y el mejoramiento de los procesos constructivos de software en dicha compañía caso de estudio.

Rengan & Rusli (2021), realizaron la investigación, *Development of Software Process Guideline to Improve Software Process Using ISO/IEC 29110-5- 1-2 Software Implementation Process*, en la Universiti Teknologi Malaysia, en Johor Bahru, Malasia. Muchas organizaciones de desarrollo de software de todo el mundo son pequeñas y medianas compañías, por lo que, generalmente, no es posible implantar con éxito las metodologías para mejorar los procesos del software en dichos tipos de compañías, ya que éstas no pueden invertir el coste de la implantación de dichos modelos a su interna. Por esta razón, presentaron el modelo de mejoramiento de la calidad de software fundamentado en la ISO/IEC 29110-5- 1-2, el mismo que sirva de soporte a dichas organizaciones, para ello, caracterizaron los diversos modelos para dicho proceso de mejora tales como, verbigracia, CMMI, PSP, TSP, ISO, entre otras. Los resultados obtenidos mostrados un modelo, el cual se constituyó de un modelo de un solo proceso de cuatro (04) fases: iniciación (comprender las prácticas actuales y realizar un análisis de las deficiencias), desarrollo (realizar un informe de análisis de deficiencias y desarrollar las directrices del proceso), aplicación (introducir la directriz del proceso, seleccionar el proyecto, preparar la descripción del proyecto piloto e implementar el proyecto piloto) y análisis (recoger opiniones - post-mortem y actualizar las directrices del proceso). Se concluyó que, el modelo propuesto sí contribuye significativamente a las actividades de desarrollo del equipo de software.

Faustino & Mejía (2020), efectuaron la investigación, *Proposal for a software development framework based on the ISO/IEC 29110 standard: Public organizations*, en el Centro de Investigación en Matemáticas A.C., en Zacatecas, México. Las actividades constructivas de software son catalogadas como un nicho de mercado que está en constante crecimiento a nivel mundial, dentro de las cuales, las pequeñas organizaciones son las que tienen la mayor representatividad, no siendo ajena el país mexicano, que es una nación que cuenta con mucho movimiento en el sector constructor de software, involucrando a sus instituciones públicas, las que presentan inconvenientes en el proceso constructivo del software, brecha que podría ser solucionada con un

modelo a medida que le muestre los pasos a seguir para lograr un proceso exitoso. Por esta razón, presentaron un tipo de trabajo para el soporte de las chicas organizaciones públicas mexicanas para el mejoramiento del proceso constructivo de software, el cual unificó los requisitos del manual mexicano MAAGTICSI, la norma mexicana MOPROSOFT, el perfil básico del estándar ISO/IEC 29110, y la metodología ágil SCRUM; asimismo, identificaron a Jest, Git, GitLab, la Suite de Google y Podio como las herramientas a servir en el proceso implementador de dicho modelo propuesto. Se concluyó que, el modelo propuesto se encuentra desarrollado pensando en las pequeñas unidades organizacionales que desarrollan software en el sector estatal mexicano, por lo que existirán mejoras en el proceso constructivo de software.

Galván et al. (2020), realizaron la investigación, *Building a Guideline to Reinforce Agile Software Development with the Basic Profile of ISO/IEC 29110 in Very Small Entities*, en la Universidad Autónoma de Aguascalientes, en Aguascalientes, México. Producir software de calidad, considerando tiempos y presupuestos es una tarea que se logra con la usanza de estándares internacionales de calidad, sin embargo, la mayoría de estos son adecuados para las empresas grandes, no siendo así para las pequeñas organizaciones mexicanas que ocupan más del 90% de la totalidad de producción de software en dicho país norteamericano, por lo que deben de adecuar sus procesos considerando estándares que se ajusten a sus contextos, verbigracia, la ISO/IEC 29110. Por esta razón, presentaron la implementación de un modelo para el desarrollo de software ajustado a la realidad de dichas microempresas mexicanas, considerando los métodos ágiles XP y SCRUM que emplean, y fundamentado además en el estándar ISO/IEC 29110 en su perfil básico, el cual consideró cuatro (04) actividades y siete (07) sprint para el proceso de Gestión de Proyectos y seis (06) actividades y siete (07) sprint para el proceso de Implementación de Software, todas ellas modeladas en Bizagi Modeler e introduciendo, además de ello, roles, tareas, productos de trabajo, eventos y documentación para el modelo definido. Los resultados obtenidos mostraron que, el 100% de los sprint propuestos cumplen a cabalidad con alinearse a los procesos y actividades propuestas por la ISO/IEC 29110. Se concluyó que, el

modelo propuesto considerando las metodologías ágiles XP y SCRUM, y fundamentado además en el estándar ISO/IEC 29110 podría replicarse en los otros tres (03) perfiles con los que cuenta la ISO/IEC 29110 (entrada, intermedio y avanzado) y considerar además otras metodologías ágiles como UPEDU, Crystal, FDD, etcétera.

Minero et al. (2020), realizaron la investigación, *Evaluation of the Implementation of the ISO/IEC 29110 Standard at the Software Development Center from the Institute Technological Superior of Nochistlán*, en el Instituto Tecnológico Superior de Nochistlán, en Zacatecas, México. El CEDESOFITSN (por su acrónimo de Centro de Desarrollo de Software del Instituto Tecnológico Superior de Nochistlán) es un instituto que brinda servicios académicos de primer nivel en el país mexicano para lo cual busca hacer empleo del estándar ISO/IEC TR 29110-1:2016 en el desarrollo de asignaturas de construcción de software, por lo que necesita evaluar la puesta en marcha de dicho estándar en los procesos constructivos de software de sus alumnos y docentes. Por esta razón, presentaron la implementación de un estándar ISO/IEC TR 29110-1:2016 en su perfil básico, considerando cuatro (04) actividades para procesar la Gestión de los Proyectos y seis (06) actividades para el proceso para implementar el Software, todas ellas modeladas en Bizagi Modeler, el cual fue implementado en el CEDESOFITSN con la puesta en marcha de veintidós (22) proyectos elaborados en conjunto por los estudiantes y docentes de dicha casa de estudios, para posteriormente evaluar la implementación considerando un cuestionario de catorce (14) reactivos. Los resultados obtenidos mostraron que, dicho modelo propuesto era considerado como útil (100.0%), ayudó al cierre y entrega de los proyectos (85.7%), fue soporte para los desarrolladores de software en sus actuales actividades constructivas en las que se desenvuelven (80.0%). Se concluyó que, el modelo propuesto, desde la perspectiva de estudiantes y docentes del CEDESOFITSN, ayuda en el proceso constructivo de software ya que se centra en un proceso bien definido por etapas, en la cual se estandarizan actividades, roles, formatos y entregables, logrando la culminación de los proyectos según la planificación inicialmente expuesta.

Suteeca (2020), realizó la investigación, *A Software Process Gap Analysis Methodology for Very Small Entity*, en la Chiang Mai University, en Chiang Mai, Tailandia. Actualmente, la demanda de las TICS y de otros programas informáticos vienen aumentando vertiginosamente por lo que se han ido creando muchas unidades que construyen software, las mismas que, en su gran mayoría son muy pequeñas, sin alcanzar más de 25 colaboradores y que advierten la existencia de problemáticas en sus labores constructivas, por lo que necesitan de la estandarización y el mejoramiento de dicho proceso. Por esta razón, presentaron la evaluación de la implementación del estándar ISO/IEC TR 29110-1:2016 en su perfil básico, considerando diez (10) empresas pequeñas constructoras de software del norte de Tailandia que necesitan de implementar dicha norma en sus procesos constructivos de software, a quienes les entregaron dos cuestionarios de 55 reactivos totales, 33 para el proceso de la gestión de proyectos y 22 para el proceso de implementación del Software y con una escala de 1 al 4. Los resultados obtenidos mostraron que, para el proceso de gestión se consiguió una media de 2.48 y un promedio de 62.12% y, en cuanto al proceso de Implementación de Software se obtuvo una media de 1.81 y un promedio de 45.45%. Se concluyó que, según el análisis realizado mediante los instrumentos de recolección, existe riesgo en los procesos constructivos de software de las pequeñas organizaciones del norte de Tailandia.

Villanueva & Muñoz (2020), realizó la investigación, *Proposal of a framework for software product and process quality assurance for very small entities*, en el Centro de Investigación en Matemáticas A.C., en Zacatecas, México. En México, más del 75% de compañías constructoras de software son muy pequeñas organizaciones las que carecen de modelos que les permitan seguir con sus hojas de ruta para conseguir softwares de gran calidad, aduciendo que seguir modelos internacionales es muy costoso o pesado al momento de implementarlo. Por esta razón, presentaron un modelo de trabajo para asegurar la calidad del software, considerando el perfil básico de la ISO/IEC 29110 para la calidad del proceso constructivo de software y la ISO/IEC 25010 para la calidad del producto, el cual se constituyó de diez (10) actividades totales, cuatro

(04) actividades para el proceso de Gestión de Proyectos y seis (06) actividades para el proceso de Implementación de Software, todas ellas integrando las ocho (08) sub características de la calidad de software según la ISO/IEC 25010 y considerando las métricas de la ISO/IEC 25023 para la evaluación de la calidad final. Los resultados obtenidos mostraron que, la calidad del proceso constructivo de software y la calidad de dicho producto son las principales inquietudes con las que cuentan las pequeñas organizaciones constructoras de software, empero, implantar y dar cumplimiento a una serie de requisitos evidenciados en los estándares sigue siendo una brecha por cerrar en este tipo de unidades haciéndolas a veces improbable de alcanzar. Se concluyó que, el modelo propuesto brinda una alternativa solucionadora que asegure la calidad del software considerando las perspectivas del proceso mismo y del producto en sí, gracias a la integración de dos normas internacionales.

Buchalcevova (2019), realizó la investigación, *Using ArchiMate to model ISO/IEC 29110 standard for very small entities*, en la Prague University of Economics and Business, en Praga, República Checa. El estándar ISO/IEC 29110 delimita una hoja de ruta que permite a las pequeñas organizaciones seguir procesos para conseguir software de calidad y, el cual ha sido implementado en diversos países del mundo tales como, verbigracia, México, Tailandia, Canadá y Perú, sin embargo, aún existen brechas en la adaptación de dicho estándar en el contexto checo, sobre todo en una adaptación al modelo empresarial haciendo uso de una notación de modelado estándar que sirva a estas pequeñas organizaciones checas. Por esta razón, presentaron un metamodelo constructivo de software para pequeñas organizaciones checas basado en el perfil básico de la norma ISO/IEC 29110 y el lenguaje de modelamiento ArchiMate; considerando cuatro (04) actividades para el proceso de Gestión de Proyectos y seis (06) actividades para el proceso de Implementación de Software; el cual fue posteriormente evaluado mediante entrevistas cualitativas semi estructuradas constituidas por quince (15) preguntas a cuatro (04) expertos en desarrollo de software. Los resultados obtenidos mostraron que, según la apreciación de los expertos, el modelo propuesto era ventajoso al mostrar una visualización más grande (100%),



ayudaba en la comprensión del contexto constructivo (100%), permitió personalizar de manera más sencilla el perfil básico de la ISO en mención (75%) y mejora la comunicación dentro del equipo desarrollador (90%). Se concluyó que, metamodelo constructivo de software para pequeñas organizaciones checas basado en el perfil básico de la norma ISO/IEC 29110 y el lenguaje de modelamiento ArchiMate garantiza la consistencia de los procesos de desarrollo de software, obteniendo una excelente calidad de proceso.

Mejía et al. (2019), realizaron la investigación, *ISO/IEC 29110 implementation tools proposal (basic profile)*, en el centro de investigación en matemáticas A.C., en Zacatecas, México. En la actualidad, en la industria desarrolladora de software existen modelos de calidad que sirven como hoja de ruta para dichas actividades constructivas, sin embargo, son ampliamente aceptadas en empresas de gran renombre, empero, en pequeñas organizaciones aún existe un déficit en cuanto a la aceptación de dichos modelos, por lo que la ISO/IEC 29110 emerge como uno creado justamente para esa necesidad específica del contexto menor mencionado, pero el vacío que aún no ha sido colmado se enfoca en las herramientas que pueden valerse dichas compañías para la implementación o automatización de estos modelos. Por esta razón, presentaron una propuesta de modelo constructivo de software basado en la ISO/IEC 29110-4-1:2011 y enfocado en pequeñas organizaciones, considerando cuatro (04) actividades para el proceso de Gestión de Proyectos y seis (06) actividades para el proceso de Implementación de Software, asimismo, plantearon herramientas soporte para la puesta en marcha de dichos procesos, verbigracia, Trello, Suite de Google (Documentos, Hoja de Cálculo, Drive, Draw.io), OpenProject, Subversion y GitLab, Los resultados obtenidos mostraron que, en cuanto a la evaluación del modelo y las herramientas propuestas, los cuatro (04) expertos del Instituto Tecnológico Superior de Loreto determinaron que el 100% de procesos y tareas propuestas cumplían a cabalidad con las normativas establecidas en la ISO/IEC 29110-4-1:2011. Se concluyó que, el modelo y las herramientas propuestas permiten familiarizarse con la ISO/IEC 29110-4-1:2011 ya que se pueden mapear los procesos constructivos con los que cuentan las pequeñas organizaciones.

Pérez & Giraldo (2019), realizaron la investigación, *Proposal of a Software Development Process for a Poultry Company in Colombia*, en la University of Quindío, en Armenia, Colombia. Las microempresas desarrolladoras de software son unidades organizacionales de menos de 25 colaboradores que aún despliegan sus actividades constructivas de manera empírica, sin seguir una hoja de ruta que les permita llevar una correcta gestión de la planificación e implantación de software, no siendo ajena la unidad de desarrollo de una reconocida compañía avícola de la zona cafetera de Colombia. Por esta razón, presentaron una mixtura de estándar del ISO/IEC 29110-5-1-2:2018 con el modelo IDEAL, considerando cuatro (04) actividades para el proceso de gestión de proyectos y seis (06) actividades para el proceso de implementar Softwares, todas ellas las especificaron en una página web de modo que sirva de guía a los colaboradores participantes del proceso constructivo del software, además de ello, realizaron pruebas con ocho (08) proyectos. Los resultados obtenidos mostraron que, dichos proyectos en el PRE TEST obtuvieron 37 retrasos en entregas parciales mientras que en el nivel POST TEST obtuvieron 3 retrasos evidenciándose un aminoramiento del 81% en retrasos, asimismo, en cuanto a la cantidad de proyectos en el nivel PRE TEST se dio cumplimiento a 3 entregas parciales a tiempo mientras que en el nivel POST TEST se dio cumplimiento a 23 entregas parciales evidenciándose un incremento del 766%. Se concluyó que, la prueba piloto realizada en dicha compañía colombiana caso de estudio logró evidenciar el aumento de la credibilidad de los procesos constructivos de software por parte de los colaboradores, así como también permitió distinguir falencias a corregir.

Castillo et al. (2020), realizaron la investigación, *Evaluation of the implementation of a subset of ISO/IEC 29110 Software Implementation process in four teams of undergraduate students of Ecuador. An empirical software engineering experiment*, en la Universidad de las Fuerzas Armadas ESPE, en Santo Domingo, Ecuador. Que las empresas que desarrollan softwares sean competentes depende principalmente de la capacidad que tengan para ofrecer sus productos con alta calidad dentro del presupuesto y el calendario aprobados. Casi todas las entidades muy pequeñas (VSE) que se dedican a

desarrollar software no reconocen las ventajas de aplicar las normas de software. Consecuentemente, se limitan a que las reconozcan como entidades que desarrollan softwares de calidad. Para esta investigación, los diferentes autores muestran los resultados que consiguieron mediante la aplicación de la ingeniería de software práctica en el experimento donde se utilizó la norma ISO/IEC TR 29110-5-1-2 "Software engineering - Lifecycle profiles for Very Small Entities (VSEs) - Part 5-1-2: Management and engineering guide: Grupo de perfil genérico: Perfil básico". La guía incluye 2 procesos: El proceso de gestión de proyectos (PM) y el proceso de implementación de software (SI). El objetivo del proyecto fue el desarrollo del producto de software que programe las citas médicas en el centro de bienestar estudiantil de una universidad de Ecuador. Participaron cuatro equipos de estudiantes de pregrado. Dos de ellos (equipos controlados) implementaron un subconjunto del proceso de IS, mientras que los otros dos (equipos no controlados) tuvieron libertad para elegir las actividades de desarrollo que posteriormente fueron mapeadas con las actividades del estándar. Todos los equipos desarrollaron el producto de software utilizando el marco SCRUM en el mismo plazo. Aunque el experimento se centró en el proceso SI, los equipos también utilizaron una versión adaptada del proceso PM definido por los profesores. La ejecución del experimento tropezó con varias dificultades. Por ejemplo, el plazo de seis semanas establecido en el diseño del experimento era demasiado corto, ya que los estudiantes trabajaban a tiempo parcial en el proyecto. Todos los equipos experimentaron esta dificultad, especialmente cuando tuvieron que construir y probar los componentes de software. En general, los equipos que utilizaron la guía ISO/IEC TR 29110-5-1-2 obtuvieron mejores puntuaciones en la prueba de calidad de los procesos del software.

Aranibar et al. (2019), realizaron la investigación *Quality assurance model for software development processes in SMEs*, en la Universidad Peruana de Ciencias Aplicadas, Lima, Perú. Según los autores, el problema principal que enfrentan las PYMEs que desarrollan software es que los modelos de calidad están dirigidos solo a las grandes empresas, dejándolas sin los beneficios que un producto con calidad genera en la optimización de procesos. Por esta razón

se decidió evaluar 3 modelos de calidad: Process and Product Quality Assurance (PPQA) del CMMI-DEV, Quality Assurance (QA) del ISO 12207, y el ciclo de Deming. Estos modelos proponen buenas prácticas, mejora gradual de procesos y mejora continua. El ciclo de Deming ha tenido un caso de éxito en cuanto a calidad y productividad, haciendo uso de validaciones continuas. De estos modelos, se adaptaron las practicas más necesarias y las más compatibles con una PYME. Finalmente se desarrolló el “Modelo de Aseguramiento de calidad en el Proceso de Desarrollo de Software para PYMEs” (MACSPS), que se basa en los tres modelos citados anteriormente, siendo un modelo simple, ágil y muy adaptable a diferentes metodologías de desarrollo. El modelo MACSPS se implementó satisfactoriamente en el proyecto de una PYME reduciendo defectos, mejorando el concepto de los requerimientos y elevando los porcentajes de éxito de las pruebas.

Basri et al. (2020), desarrollaron la investigación, *The organisational factors of software process improvement in small software industry: Comparative study*, en la Universidad de Ilorin en Nigeria. Según esta investigación, los modelos de calidad existentes no son apropiados para organizaciones muy pequeñas, lo que ocasiona que las microempresas entreguen productos de calidad deficiente. En este estudio, se trabajó sobre la perspectiva organizacional en las PYMES y la forma en que impacta la mejora de procesos de software (SPI), señalándose una lista de once factores que deben tenerse en cuenta. Se observó como el factor más crítico a la orientación empresarial, ya que se reconoce que los integrantes de una organización deben comprender plenamente la finalidad de los proyectos. En segundo lugar, se identificó al compromiso de la dirección como factor relevante, pues se considera que el apoyo en recurso económicos y humanos, además de darle el carácter de estratégico, incide positivamente en la mejora de procesos. Como resultado, se proporcionó una hoja de ruta para permitir la SPI en la pequeña empresa. El artículo logró alentar a los investigadores a utilizar la técnica de compromiso para analizar futuros estudios empíricos, validando la idoneidad de los factores identificados en un país específico.

### **1.3. Teorías relacionadas al tema.**

#### **1.3.1. Software por licencia**

Todas las empresas utilizan software para administrar procesos comerciales, comunicarse con empleados, clientes y proveedores, y para muchos otros fines. En casi todas las cuestiones, los productos de software requieren la activación de licencias o la aceptación de “términos y condiciones” antes de poder descargar, instalar o acceder a los programas.

Hay muchos tipos de licencias de software, con diferentes términos, acuerdos de soporte, restricciones y costos. Los usuarios deben comprender los conceptos básicos de las licencias de software para garantizar una comprensión completa de las responsabilidades y el cumplimiento de los términos y limitaciones legales.

Según Snyk (2022) un software por licencia es un software contratado mediante un documento entre el ente que creó e implemento la aplicación o los productos relacionados y sus usuarios finales. La licencia es el documento con texto diseñado que protege los derechos de propiedad intelectual de quien desarrolle el software, además limita los reclamos que puedan surgir al usarlas.

Un software por licencia es un sistema desarrollado por una empresa de servicios TI, para ser comercializado como producto de software a cambio de un pago mensual o anual por parte del cliente; el cual puede utilizar el sistema y todas o algunas de sus características conforme a sus necesidades. El pago que realiza el cliente le da el derecho expreso por parte del desarrollador de usar el producto bajo las especificaciones para el cual fue diseñado el sistema; y el acceso que obtiene el cliente es de uso del sistema y sus funciones, más no le da acceso al código de desarrollo. Este tipo de sistemas son desarrollados bajo la identificación de oportunidades de mercado que el profesional de software identifica y busca solucionarlo mediante el producto de software (Šmite, Moe, Šāblis, & Wohlin, 2017).

La licencia del software proporciona significados legales para que se distribuya y use el software. Los derechos de los usuarios finales, como el de instalación, sus garantías y la responsabilidad, también se especifican a menudo en las licencias de software, del mismo modo incluye la protección de su propiedad intelectual de quien los haya desarrollado. (Yang, Hosseinian, Jraisat, & Rangaswamy, 2021).

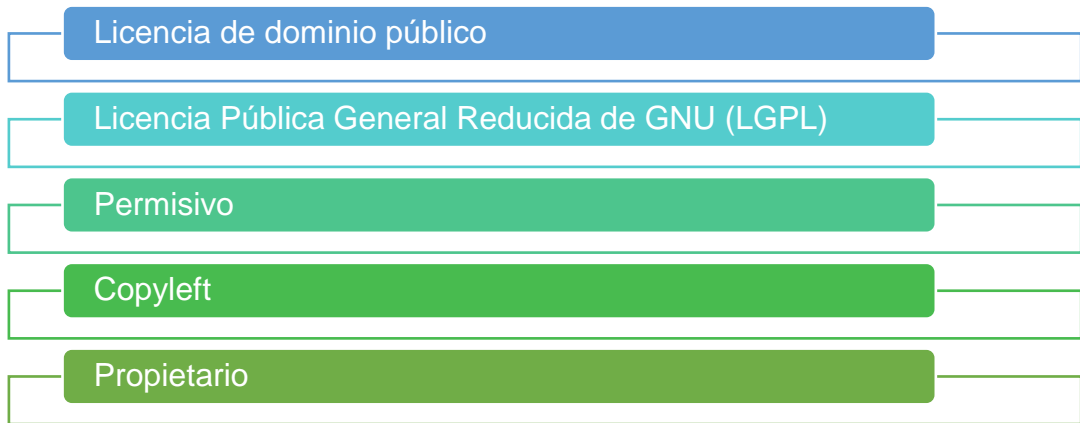
Según Snyk (2022) la mayoría del software cae dentro de una de dos (02) categorías que tienen claras diferencias en cómo se ven bajo la ley de derechos de autor:

- a. Propietario: también conocido como “código cerrado”. No otorgan tal autoridad para la modificación o reutilización del código y normalmente brindan software solo con código operativo y sin código fuente. Una licencia de software propietario a menudo incluye términos que prohíben la "ingeniería inversa" del código objeto con la intención de obtener el código fuente por parte del licenciataria (Snyk, 2022).
- b. Software gratuito y de código abierto (FOSS): denominado “código abierto”. Otorgan derechos al cliente que incluyen la modificación y reutilización del código de software, proporcionando el código fuente real con los productos de software. Este tipo de licencia de código abierto otorga al usuario la autoridad para modificar las funciones del software y la libertad de inspeccionar el código del software (Snyk, 2022).

### **1.3.2. Tipos de software por licencia**

Según Snyk (2022), existen cinco clases principales de software por licencias utilizadas para las diversas variedades de software y contratos comerciales. Estos cubren una amplia gama de licencias, que van de software libre (para el público) a software comercial por pago (para el propietario). Entre estos dos extremos, también hay tres categorías (GNU/LGPL, permisivo y copyleft) que se aplican a varias formas de

proyectos de código abierto. El incumplimiento de las cláusulas y condiciones de la licencia de código abierto puede dar lugar a la revelación de secretos comerciales o incluso a acciones legales por parte de los desarrolladores del proyecto.



#### **1.3.2.1. Licencia de dominio público**

Según Snyk (2022), es de uso libre, es decir, el software va dirigido al público, cualquiera puede usarla y modificarla sin restricción alguna. Esta es una licencia “permisiva” pues permite que se adopten los códigos en aplicaciones o proyectos y la reutilización del software como uno desee.

Por muchas razones, las empresas tienen que tener cuidado al adoptar software de este carácter en sus proyectos u otras aplicaciones importantes:

- a. Es posible que el software de dominio público no siempre se adhiera a las mejores prácticas de codificación o que no proporcione mucha seguridad que requiere la aplicación.
- b. El software sin términos en su licencia concretos no siempre es un código de dominio público. Asegúrese de que el software sea verdaderamente de dominio público antes de copiarlo, reutilizarlo o distribuirlo.

### **1.3.2.2. Licencia Pública General Reducida de GNU (LGPL)**

Según Snyk (2022), bajo una LGPL, quien los desarrolle tienen derechos para vincular bibliotecas con código abierto en su software. Los códigos resultantes se pueden licenciar bajo otra licencia, inclusive propietaria, cuando los proyectos se compilan o vinculan para incluir una biblioteca con licencia LGPL.

La advertencia es que, si cualquier parte de la biblioteca se copia en el código o se modifica, los términos de la licencia LGPL original se aplicarán al código desarrollado que utilizó la biblioteca.

### **1.3.2.3. Permisivo**

Según Snyk (2022), este tipo de licencia es bastante común y popular de las licencias del software con código abierto. Bajo una licencia permisiva, también conocida como “Apache” o “estilo BSD”, existen pocas restricciones o requisitos para la distribución o modificaciones del software. Otra variación de una licencia de software permisiva es la licencia “MIT”.

Las variantes de las licencias permisivas incluyen diferencias en los requisitos para conservar los avisos de licencia y los derechos de autor del software, así como la forma en que se puede usar el software (comercial o privado), los requisitos de marcas registradas y otras estipulaciones.

### **1.3.2.4. Copyleft**

Según Snyk (2022), los términos de esta licencia son restrictivos, conocidos como licencias recíprocas. Según los términos de una licencia copyleft, el código con licencia puede modificarse o distribuirse como parte de un proyecto de software si el nuevo código se distribuye bajo la misma licencia de software.



Esto significa que, si se especificó que el código incluido en el producto de software es para “uso personal únicamente”, el nuevo producto que se distribuye debe tener la misma designación/restricción.

Dado que el software original incluido con el nuevo proyecto permitía modificaciones y distribución, es posible que esta no sea la mejor licencia para los desarrolladores de software porque el código resultante también debe tener el tipo de licencia copyleft, incluida la disponibilidad del código fuente.

#### **1.3.2.5. Propietario**

Según Snyk (2022), estas licencias de software hacen que el software no sea apto para su copia, modificación o distribución. Esta es la licencia más restrictiva, que protege al desarrollador o propietario del uso no autorizado del software.

#### **1.3.3. Importancia del software por licencia**

Según Preston (2022), los softwares por licencia son importantes porque protegen la propiedad intelectual de quienes desarrollan el software. Las licencias también pueden proporcionar a los desarrolladores una vía clara de recurso legal cuando los usuarios no cumplen con las estipulaciones de los acuerdos de licencia. Las licencias de software tienen como objetivo garantizar que todas las partes que crean y usan el software puedan beneficiarse de él. Estas licencias también ayudan a definir hasta qué punto un usuario puede operar el software, acceder a él y modificar su código fuente con EULA (por su acrónimo de End User License Agreement) que establecen un esquema legal de los derechos de un usuario para instalar y usar el software después de comprarlo.

#### **1.3.4. Funcionamiento de un software por licencia**

Según Snyk (2022), los nuevos usuarios de un software normalmente firmarán un acuerdo de licencia de usuario final (EULA) que constituye una definición legal de la relación entre el licenciante (proveedor) y el licenciatario

(usuario o empresa). El EULA es un contrato que establece los derechos del comprador para instalar y utilizar el software.

Cada EULA contiene una cláusula que estipula cuándo un usuario final activa sus condiciones. Este puede ser el momento en que el usuario abre el empaque del producto o, por ejemplo, el momento en que se da clic al botón aceptando los términos del EULA para acceder a él.

Las aplicaciones basadas en la nube, como el software como servicio (SaaS), a menudo incluirán detalles de licencia en los EULA, que incluyen:

- a. Cargos mensuales o anuales por usuario
- b. Duración del acuerdo
- c. Términos de cancelación del acuerdo
- d. Recuperación de cualquier cargo si se cancela durante el acuerdo

Un uso adicional de la licencia de software es en los casos en que un desarrollador de software o una empresa otorgan autoridad para vender o distribuir el software bajo la marca de la otra parte. El desarrollador conserva la propiedad, pero la empresa de cambio de marca puede revender el producto de software. Este método de concesión de licencias se denomina “marca blanca” (Snik, 2022).

### **1.3.5. Procesos de desarrollo de software por licencia**

El desarrollo del software por licencia está basado en los procesos que desarrollan programas comunes con algunos cambios; dentro del proceso según Šmite et al. (2017) tenemos:

- a. Identificación de las oportunidades de mercado y los requerimientos del cliente.
- b. Transformación de esos requerimientos en exigencias de software.
- c. Modelamiento de procesos.
- d. Elaboración y ejecución de proyecto de desarrollo

- e. Testing del producto de software
- f. Desarrollo final del producto.

Los métodos se caracterizan por ciclos de desarrollo cortos e iterativos, realizados por equipos autoorganizadas. Estos equipos trabajan con técnicas como diseños más ligeros, refactorización de códigos, diseño de pruebas. Una característica de estos métodos es proporcionar productos de trabajo demostrables en un ciclo de desarrollo (Šmite, Moe, Šāblis, & Wohlin, 2017). Los métodos apuntan a cumplir con los siguientes objetivos:

- a. Desarrollar software más rápido,
- b. Desarrollar software de manera incremental y
- c. Desarrollar software enfocado en aumentar la complacencia del cliente.

Para conseguir estos objetivos, se proporciona un marco conceptual de prácticas y principios. Además, su objetivo es lograr la formalidad por licencia con el equipo de desarrollo auto organizado (Šmite, Moe, Šāblis, & Wohlin, 2017).

### **1.3.6. Modelos de procesos AD HOC**

Según Özcan & McCaffery (2019), han surgido muchos métodos ad hoc, los más populares son:

#### **a. Scrum**

El desarrollo de scrum también se centra en un equipo de individuos que actúan juntos, avanzando, trabajando fuerte e integrados; comprender su función es una tarea importante de un equipo scrum; el enfoque y las prioridades deben ser claros para el equipo.

- i. **Iterativo e incremental:** Scrum es el método de desarrollo incremental. Esto significa que el equipo de desarrollo vuelve a las tareas o actividades completadas y cambia, mejora o agrega cosas nuevas.

- ii. **Pregame, Game, Postgame:** Pregame, Game y Postgame son las tres fases principales del desarrollo scrum. En la fase previa al juego, la planificación se realiza junto con la arquitectura del sistema y el diseño de alto nivel. Durante la fase de juego, los sprints (iteraciones) están relacionados. Aquí es donde se realiza el análisis real, el diseño, desarrollo e ingeniería. El cierre es una actividad de la última fase, donde el proyecto se entrega al cliente.
- iii. **Scrum diario:** El scrum diario es una reunión breve que se lleva a cabo todos los días en la que el grupo discute lo hecho desde la reunión última, qué se debe hacer hasta la próxima reunión, qué problemas potenciales y otros problemas deben resolverse. El scrum master (líder del grupo) está a cargo de estas breves reuniones que tienen una duración aproximada de 15 minutos.
- iv. **Atrasos:** Existen dos tipos de atrasos en el desarrollo de scrum. La cartera de productos enumera todo lo necesario para entregar un producto final y el spring backlog enumera los elementos tomados del producto y atrasos que deben completarse durante un sprint (iteración).
- v. **Desarrollar, envolver, ajustar y revisar:** Los cuatro atributos: son actividades relacionadas con los sprints (iteraciones) en el desarrollo de scrum.
- vi. **Horario flexible:** Es posible que se requiera un entregable antes o después de lo planeado por el comenzando. Esto significa que los miembros del proyecto también deben pensar, trabajar y actuar con flexibilidad.
- vii. **Equipos pequeños:** Son pequeños y contienen de tres a seis desarrolladores. Desarrolladores, los documentadores y el personal de control de calidad forman estos equipos. Puede haber uno o más equipos trabajando en el mismo proyecto.

#### **b. Extreme programming (XP)**

- i. **Versiones pequeñas:** XP tiene dos visiones diferentes del concepto denominadas versiones pequeñas. En la primera vista,

el software se lanza, se ejecuta y se prueba. También ofrece valor comercial elegido por el cliente. Esto ocurre durante cada iteración. Lo que el cliente haga con el software proporcionado depende totalmente de ellos, pero generalmente lo evalúan o lo entregan a los usuarios finales. Proporcionar software visible al cliente lo mantiene abierto y tangible. En la segunda vista, los equipos de XP también lanzan su software a los usuarios finales.

- ii. **Programación en pareja:** Dos programadores se sientan uno al lado del otro en la misma máquina mientras escriben el código. Al hacer esto, todo el código es revisado por al menos otro programador.
  - i. **Diseño simple:** Un equipo de XP sigue un diseño adecuado para la funcionalidad actual del sistema. Ningún trabajo se desperdicia y el software está listo para seguir desarrollándose.
  - ii. **Desarrollo impulsado por pruebas:** Para comprender los comentarios, XP utiliza el desarrollo impulsado por pruebas. Por ambas pruebas de programador o pruebas unitarias, se garantiza que el código funciona y es de alta calidad. Los ciclos de agregar pruebas en el código existente son breves.
- iii. **Pruebas de clientes:** Para demostrar que una característica está funcionando, los programadores realizan una aceptación pruebas sobre sí mismos y sobre el cliente. Esto prueba que la funcionalidad se implementa de forma correcta.
- iv. **Estándar de codificación:** Al seguir un estándar de codificación, el código desarrollado se verá como lo escribió una sola persona
- v. **Propiedad colectiva:** Cualquier pareja de programadores puede mejorar o cambiar cualquier código en cualquier momento durante el proyecto. Al hacer esto, todo el código puede llamar la atención de muchas personas, lo que conduce a un mejor código y menos errores.

### c. Lean development

- i. **Elimine el desperdicio:** Eliminar todo lo que no le sirve al cliente, como valor.
- ii. **Ampliar el aprendizaje:** Al aprender de experiencias pasadas, los problemas resueltos deben documentarse para evitar el desperdicio al reconstruir la misma solución cuando se encuentre con el mismo problema en el futuro.
- iii. **Decidir lo más tarde posible:** Diferiendo el compromiso en el sentido de decisiones difíciles o imposibles, las decisiones que son difíciles de cambiar deben trasladarse al último minuto. También habrá un plan de respaldo si la decisión de elección no funciona
- iv. **Muda:** Cualquier actividad que requiera recursos pero que no contribuya a ningún valor de salida se considera como un desperdicio en el desarrollo de software lean, llamado muda.
- v. **Reutilización:** La reutilización de componentes permite mejorar la calidad y reducir los costos, ya que los componentes reutilizados ya han sido desarrollados, probados y aprobados; Valor: el valor es lo que el cliente cree que es útil o valioso para él
- vi. **QFD - QFD significa desarrollo de funciones de calidad.** Esta técnica ayuda a los equipos en la traducción de requisitos del cliente.

### d. DSDM

El método de desarrollo de sistemas dinámicos (DSDM) es un marco para desarrollar rápido las aplicaciones.

- i. **La activa participación de los usuarios es imperativa:** Al involucrar a una pequeña cantidad de usuarios a lo largo del proyecto, se reduce eficazmente los errores en términos de percepción del usuario, lo que reduce los costes de los errores.
- ii. **Los equipos deben estar facultados para tomar decisiones**  
De acuerdo con la autorización de DSDM, los participantes y

gerentes ralentizan los proyectos. Por lo tanto, los participantes deben recibir una autorización en relación con algunas actividades relacionadas con el proyecto. La autorización incluye: requisitos en práctica, qué funcionalidad debe estar en un incremento dado, priorización de requisitos y características, refinando detalles de la solución técnica. Al empoderar a los participantes con estas autorizaciones, se puede reducir el tiempo dedicado a la comunicación y las solicitudes

- iii. **Concéntrese en la entrega frecuente:** Al entregar errores frecuentes se pueden detectar rápida y contrarrestar fácilmente. Esto comprende tanto el código del programa como la documentación
- iv. **Adecuación para las empresas en el criterio de entregables aceptados:** El marco DSDM recomienda centrarse en ofrecer software que satisfaga las necesidades comerciales y que se deben realizar mejoras durante iteraciones posteriores
- v. **El desarrollo iterativo e incremental es obligatorio:** Para mantener la administración de proyectos complejos; dichos proyectos deben dividirse en pequeños paquetes de funciones. Cada lanzamiento presenta nuevas características hasta que todos cumplan los requisitos comerciales. DSDM significa que los incrementos más pequeños permiten cambiar requisitos
- vi. **Todos los cambios durante el desarrollo deben ser reversibles,** como aconseja DSDM, la iteración a través de pequeños incrementos significa que la pérdida de trabajo durante las fases de cambio es muy limitada
- vii. **Los requisitos se basan en el nivel alto:** Se necesitan algunos requisitos de alto nivel para limitar el grado de cómo los requisitos pueden cambiar en el ciclo de vida del plan.
- viii. **Las pruebas están integradas a lo largo del ciclo de vida:** DSDM requiere pruebas en los primeros períodos del proceso de desarrollo. Los grupos de control podrían probar elementos tales como documentos de entrevistas

- ix. **Enfoque colaborativo y cooperativo:** DSDM provoca la colaboración entre técnicos y personal empresarial en un proyecto para conseguir un entorno de confianza y honestidad. Esto permite recopilación de requisitos y comentarios honestos sobre el producto.

#### **1.3.4. ISO/IEC 29110**

El grupo de trabajo de la ISO / IEC JTC1 SC7 encargado de desarrollar el nuevo conjunto de estándares para MYPEs, utilizó el concepto de perfiles estandarizados ISO, de ISO / IEC / IEEE 12207 para desarrollar los nuevos estándares para el software de desarrollo de MYPEs. Desde un punto de vista práctico, un perfil es una especie de matriz, que identifica con precisión los elementos que se toman de los estándares existentes de los que no lo son. El enfoque general seguido para desarrollar este nuevo estándar para MYPEs consistió en los siguientes pasos:

- a. Desarrollar un conjunto de perfiles para MYPEs que no estén involucrados en el desarrollo de software crítico.
- b. Seleccionar los subconjuntos de proceso ISO / IEC / IEEE 12207 aplicables a las MYPEs que tienen hasta 25 personas
- c. Seleccionar la descripción de los productos, a ser producidos por un proyecto, usando estándar ISO / IEC / IEEE 15289
- d. Desarrollar pautas, listas de verificación, plantillas, ejemplos para respaldar los subconjuntos seleccionados.

Los requerimientos base de un proceso de desarrollo de programas son que debe alinearse a las necesidades del proyecto y éxito del proyecto de ayuda. Y esta necesidad debe ser informada por el argumento situacional donde en el proyecto debe operar y; por lo tanto, el proceso de desarrollo de programas más adecuado depende del contexto. La característica situacional central de las MYPEs a las que se dirige la norma ISO / IEC 29110 es el tamaño.



La serie de industria de software y sistemas ISO / IEC 29110 se creó como una solución para ayudar a las MYPEs a enfrentar desafíos como mejorar su producción y su calidad con costos mínimos. Esta norma fue desarrollada por ISO WG24. La serie ISO / IEC 29110 fue diseñada para ayudar a las MYPEs que no desarrollan sistemas críticos (compuestos por componentes de hardware y software) o software.

Esta serie de estándares de software y guías de administración e ingeniería persigue como fin, optimizar el proceso de desarrollo de programas de las MYPEs a través de la implementación de prácticas comprobadas que podrían implementarse fácilmente dentro de una MYPE y están enfocadas en obtener beneficios como la calidad de producto, los tiempos de entrega y los costos de producción (Laporte & O'Connor, 2017).

Castillo et al. (2020), define las características que caracterizan a la serie ISO / IEC 29110 para software son las siguientes:

- a. Posee un grupo de perfil genérico que tiene una hoja de ruta con cuatro etapas, llamados perfiles, que deben ser seleccionados por las MYPEs de acuerdo con sus objetivos: (a) el perfil de entrada debe seleccionarse si la MYPEs trabaja en proyectos pequeños como el esfuerzo de seis meses-persona y start-ups; (b) se debe seleccionar el perfil básico si el MYPE desarrolla solamente una aplicación por parte de un grupo de trabajo. Es el perfil único en el que se puede certificar una MYPE; (c) se debe seleccionar perfil intermedio si la MYPE desarrolla más proyectos paralelamente con más grupos de trabajo y (d) perfil avanzado debe seleccionarse si la VSE desea crecer y mantenerse como un procedimiento competitivo independiente para desarrollar software.
- b. Tiene dos procesos: El proceso de implementación de programas y el proceso de gestión de proyectos.
- c. Funciona en MYPEs usando algún enfoque, método o herramienta para su desarrollo.

- d. Proporcionan varios elementos del proceso como actividad, rol, tarea, objetivo y producto de trabajo.

Es importante resaltar que un “proceso condicional” es un proceso que puede ser obligatorio bajo algunas condiciones específicas, puede ser opcional bajo otras condiciones especificadas y puede estar fuera del alcance o no aplicable bajo otras condiciones especificadas; estos deben ser observados si se aplican las condiciones especificadas (Castillo, Sánchez, Villarroel, & Sánchez, 2020).

El análisis proporciona una evidencia de que los procesos de ISO / IEC 29110 están mejor cubiertos por los planes de estudio de las universidades porque los procesos proporcionan el conjunto mínimo de prácticas a realizar mientras se ejecuta un proyecto desde el inicio hasta la entrega de un software. Además, este mapeo presenta una clara diferenciación entre estos dos estándares que podría ayudar a los Centros de Desarrollo de Software a comprender por dónde empezar en la implementación de uno de ellos.

Estos dos estándares están mapeados para comprender los resultados del análisis en detalle y brindar recomendaciones sobre los programas académicos. El análisis proporciona una evidencia de que dichos procesos, están mejor cubiertos por los planes de estudio de las universidades, porque los procesos proporcionan el conjunto mínimo de prácticas a realizar mientras se ejecuta un proyecto desde el inicio hasta la entrega de un software (Muñoz, Mejía, Peña, Lara, & Laporte, 2019).

#### **1.3.5. ISO/IEC Perfil Básico**

Según ISO (2016), en el núcleo, el perfil básico de este estándar es una gestión e ingeniería, oficialmente conocida como ISO / IEC TR 29110-5-1-2, que se centra en el proyecto de ejecución de sistemas y gestión. El fin de esta norma es definir la implementación de programas y la gestión de proyectos de un subconjunto de ISO 12207 e ISO 15289 apropiado para

MYPEs. La razón real para incluir la gestión de proyectos es que el negocio principal de las MYPEs es el desarrollo de software y sus finanzas. El éxito depende de la finalización exitosa del proyecto dentro del tiempo y costo, así como en la obtención de beneficios.

Este estándar define dos procesos: La implementación de programas y Gestión de proyectos. La finalidad del primero es realizar metódicamente las actividades de diseño, análisis, integración, construcción y experimentos para nuevos productos cambiados de acuerdo con las exigencias del mercado. El propósito del segundo proceso es ejecutar de manera sinérgica las actividades del proyecto de implementación del sistema, lo que admite cumplir los fines de dicho proyecto en los factores esperados.

Los siete objetivos de la gestión de proyectos son:

- a. El trabajo se ejecuta acorde al acta de trabajo, aprobada por el cliente.
- b. El avance del trabajo se revisa y compara con lo planeado, y se va llevado un seguimiento del progreso.
- c. Cualquier cambio para por un análisis de tiempo, tecnología y costo.
- d. Todos los riesgos se administran durante la elaboración del trabajo.
- e. Se establece un marco para las versiones de programa.
- f. Mediante un sistema de calidad de desarrollo, se busca cumplir con lo solicitado.

Las cuatro actividades de la gestión de proyectos son:

- a. Planeación de trabajo: su objetivo principal es desarrollar y transmitir los planes de los proyectos efectivos y viables.
- b. Realización del plan de trabajo: Implementar las tareas reales del de acuerdo con el plan
- c. Prueba y control del proyecto: el fin es medir como se encuentra el proyecto y asegurarse de que el proyecto se desarrolle acorde con las

actividades y cronogramas, con los costos planeados y que cumpla los objetivos trazados.

- d. Cierre de proyecto: normalmente implica la publicación de los entregables finales y la comunicación final del proyecto a cada parte interesada.

El fin del proceso de implementación de sistemas es optimizar el trabajo de desarrollo de programas. Los requisitos son:

- a. Las tareas de sus actividades se ejecutan realizando el actual procedimiento.
- b. Se definen los puntos solicitados de software.
- c. El diseño arquitectónico y detallado del programa se desarrolla y se establece como base.
- d. Se originan los componentes del programa establecidos por su diseño.
- e. El programa se fabrica integrando los componentes del sistema, se verifica empleando situaciones de prueba e instrucciones de prueba.
- f. La configuración de programas que cumpla con la descripción de los requisitos del cliente.
- g. Se realizan verifica y valida lo implementado.

Las actividades del proceso de implementación del programa son:

- a. Inicio de la implementación del sistema: asegura que la metodología del proyecto especificada en las actividades de planificación del proyecto sea adoptada por el grupo de trabajo.
- b. Analizar los requisitos de los programas: analiza los requerimientos del cliente acordados y establece las necesidades del proyecto validado.
- c. Arquitectura de programa y diseño detallado: convierte los requisitos de software a la arquitectura del software del sistema y al diseño detallado del software.
- d. Construcción de software: crea el código y los datos del programa
- e. Pruebas e integración de programa

f. Entrega del producto: suministra el producto completado al cliente.

#### **1.4. Formulación del Problema.**

¿Cómo mejorar el desarrollo de software por licencia de una MYPE de servicios de TI?

#### **1.5. Justificación e importancia del estudio.**

Esta investigación aporta a nivel de conocimientos en ingeniería de software, al diseñar, validar y probar un modelo de procesos ad hoc para desarrollar software por licencia para una MYPE de servicios de TI, en base a la norma ISO/IEC 29110; la cual sirvió como base para otros estudios relacionados en el tema y que pueden ampliar más el mismo.

A nivel tecnológico, aporta en la mejora de desarrollo de software en las MYPEs, generando que éstas puedan brindar un producto de software de calidad a sus clientes; lo que nivel de justificación social, se basa en el concepto de que si la empresa gana todos ganan, generando así beneficios para los clientes, comunidad, empleados y familias en general.

#### **1.6. Hipótesis.**

Mediante el desarrollo de un modelo de procesos ad hoc basado en ISO/IEC 29110 se mejorará el desarrollo de software por licencia para una MYPE de servicios de TI.

#### **1.7. Objetivos.**

##### **1.7.1. Objetivo general.**

Desarrollar un modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110.

##### **1.7.2. Objetivos específicos.**

- a. Caracterizar ISO/IEC 29110 como norma base de procesos de construcción de software.

- b. Elaborar un diagnóstico base del estado actual del proceso de desarrollo de software de una MYPE de servicios de TI previamente seleccionada.
- c. Diseñar un modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI.
- d. Validar mediante juicio de expertos la propuesta del modelo de procesos ad hoc desarrollado.
- e. Realizar una prueba piloto del modelo de procesos ad hoc para el desarrollo de software por licencia en una MYPE de servicios de TI.

## II. MATERIAL Y MÉTODO

### 2.1. Tipo y Diseño de Investigación.

#### 2.1.1. Tipo de Investigación

Esta investigación fue tecnológica aplicada pues, se construyó un modelo de procesos ad hoc para desarrollar el software por licencia en base a la ISO/IEC 29110, que posteriormente fue implementada mediante una prueba piloto de tres (03) proyectos en una MYPE de servicios de TI, que para este contexto específico fue la compañía NEDLEY SUPPORT S.A.C., con lo que se pudo mejorar el desarrollo de software por licencia en dicha compañía trujillana caso de estudio, considerando los indicadores de la Tabla 2.

#### 2.1.2. Diseño de Investigación

Según su naturaleza, este estudio fue de diseño cuasi experimental pues, a pesar de que no se hicieron alteraciones al nivel de la Variable Independiente “Modelo de procesos ad hoc basado en ISO/IEC 29110”, sí se midieron los niveles de la Variable Dependiente “Desarrollo de software por licencia” en dicha MYPE de servicios de TI, pero en un contexto de ejecución de la prueba piloto del modelo propuesto en la compañía NEDLEY SUPPORT S.A.C. Por tanto, se tuvo el siguiente diseño cuasi experimental:

$$G_e \rightarrow X \rightarrow O_2$$

Donde:

$G_e$ : grupo experimental

$X$ : prueba piloto del modelo de procesos ad hoc en la compañía NEDLEY SUPPORT S.A.C.

$O_2$ : observación post test

### 2.2. Población y muestra.

En este estudio, la población y muestra a tomar en cuenta fue un caso de estudio, la compañía NEDLEY SUPPORT S.A.C., con sede en la Av. Costa Rica N° 231 Urb. Torres Araujo, Trujillo, Trujillo, La Libertad, ya que es una MYPE de servicios de TI, que es justamente el contexto de estudio que se requería y es una compañía a la que se tuvo acceso y permiso para desarrollar la investigación, tal y como consta en el Anexo 2.

## **2.3. Variables, Operacionalización.**

### **2.3.1. Variables**

**Variable independiente:** Modelo de procesos ad hoc basado en ISO/IEC 29110

**Variable dependiente:** Desarrollo de software por licencia

### **2.3.2. Operacionalización**



**Tabla 1.**

**Operacionalización de la Variable Independiente**

Variables	Dimensión	Indicador	Ítem	Técnica e instrumentos de recolección de datos
<b>Variable Independiente:</b> Modelo de procesos ad hoc basado en ISO/IEC 29110	Aceptación del Modelo	Nivel de Claridad	$N_{Claridad} = \frac{(CE_1 + CE_2 + \dots + CE_n)}{n}$	Técnica: Juicio de expertos  Instrumento: Ficha de Juicio de expertos  (Anexo 4)
		Nivel de Objetividad	$N_{Objetividad} = \frac{(CE_1 + CE_2 + \dots + CE_n)}{n}$	
		Nivel de Actualidad	$N_{Actualidad} = \frac{(CE_1 + CE_2 + \dots + CE_n)}{n}$	
		Nivel de Organización	$N_{Organización} = \frac{(CE_1 + CE_2 + \dots + CE_n)}{n}$	
		Nivel de Suficiencia	$N_{Suficiencia} = \frac{(CE_1 + CE_2 + \dots + CE_n)}{n}$	
		Nivel de Intencionalidad	$N_{Intencionalidad} = \frac{(CE_1 + CE_2 + \dots + CE_n)}{n}$	
		Nivel de Consistencia	$N_{Consistencia} = \frac{(CE_1 + CE_2 + \dots + CE_n)}{n}$	
		Nivel de Coherencia	$N_{Coherencia} = \frac{(CE_1 + CE_2 + \dots + CE_n)}{n}$	
		Nivel de Metodología	$N_{Metodología} = \frac{(CE_1 + CE_2 + \dots + CE_n)}{n}$	
		Nivel de Pertinencia	$N_{Pertinencia} = \frac{(CE_1 + CE_2 + \dots + CE_n)}{n}$	

Fuente, elaboración propia.

**Tabla 2.**

**Operacionalización de la Variable Dependiente**

Variables	Dimensión	Indicador	Ítem	Técnica e instrumentos de recolección de datos
<b>Variable Dependiente:</b> Desarrollo de software por licencia.	Defectos por proyecto	Cantidad de defectos	$CDP = \sum (DE)$	T: Observación I: Ficha de Observación (Anexo 5)
	Satisfacción del Cliente	Porcentaje de Clientes Satisfechos	$P_{CS} = \frac{(TC - CI)}{TC} * 100$	T: Encuesta I: Cuestionario (Anexo 6)

*Nota, Elaboración propia.*

## 2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.

### 2.4.1. Juicio de Expertos

Se hizo usanza del juicio de expertos como técnica para recopilar datos, la misma que se puesta en marcha solicitando la apreciación crítica de tres (03) expertos con experiencia proba en desarrollo de software, a quienes se les remitió mediante vía correo electrónico una ficha de juicio de expertos considerando los diez (10) indicadores mencionados en la Tabla 1. Asimismo, la ficha utilizada para recolectar la apreciación de estos expertos se encuentra en el Anexo 4. Los expertos para la aceptación del modelo de procesos ad hoc basado en ISO/IEC 29110 se encuentran a continuación:

**Tabla 3**

***Expertos para validación de modelo***

N°	Experto	Grado Académico	Experiencia
1	Ruiz Gómez Woolder	Doctor (c) en Ingeniería de Sistemas e Informática	Ingeniería de Sistemas, experiencia en Proyectos para desarrollar software e Infraestructura de TI en el sector público y privado. Anexo 8
2	Tapia Flores Jesús Benjamín	Maestro en Ingeniería de Sistemas	Ingeniería de Sistemas, experiencia en gestión de aplicaciones y gestión de procesos SCRUM. Anexo 8
3	Reyna Robles Jaime Luis	Maestro (c) en Ingeniería de Sistemas y Computación	Ingeniería de desarrollo de software y experto en modelo de procesos de la ISO/IEC 29110. Anexo 8

*Fuente, elaboración propia.*

Posterior a la recepción de las respuestas por parte de los tres (03) expertos mencionados previamente, los resultados fueron tabulados mediante hoja de cálculo de MS-Excel para su posterior análisis e interpretación, las mismas que fueron puestas a disposición en los resultados de esta investigación.

#### **2.4.2. Observación directa**

Para este informe, se hizo usanza de la observación como técnica de recapitulación de informaciones dado que, el investigador fue miembro del entorno donde se realizó una prueba piloto del modelo de procesos ad hoc para una MYPE de servicios de TI en base a ISO/IEC 29110. Para ello, se observaron seis (06) proyectos de desarrollo de software por licencia en dicha compañía trujillana caso de estudio y a los que se les tomaron datos mediante una ficha de observación, la misma que fue diseñada por el autor en su totalidad y que se encuentra especificada en el Anexo 5.

#### **2.4.3. Encuesta**

Para este informe, se hizo usanza de la encuesta como técnica de recapitulación de informaciones dado que, sirvió para determinar el porcentaje de clientes satisfechos con cada uno de los dos (02) grupos de proyectos, PRE TEST y POST TEST. Esta técnica fue empleada mediante el cuestionario que se encuentra especificado en el Anexo 6.

### **2.5. Procedimiento de análisis de datos.**

#### **2.5.1. Procedimiento de análisis de datos de la variable independiente**

Para conseguir la evaluación de la evaluación de la aceptación del modelo de procesos ad hoc basado en ISO/IEC 29110, se emplearon los siguientes indicadores:

- Nivel de Claridad
- Nivel de Objetividad

- Nivel de Actualidad
- Nivel de Organización
- Nivel de Suficiencia
- Nivel de Intencionalidad
- Nivel de Consistencia
- Nivel de Coherencia
- Nivel de Metodología
- Nivel de Pertinencia

Para ello se hizo usanza de la siguiente fórmula estadística:

$$N_{INDICADOR} = \frac{(CE_1 + CE_2 + \dots + CE_n)}{n}$$

Donde:

$N_{INDICADOR}$ : Nivel de Indicador

$CE$ : Calificación de Experto

$n$  = Número Total de Expertos

### **2.5.2. Procedimiento de análisis de datos de la variable dependiente**

Para conseguir la prueba de calidad del desarrollo de software por licencia, se emplearon los siguientes indicadores:

- Cantidad de Defectos:

$$CDP = \sum (DE)$$

Leyenda:

$CDP$ : Cantidad de defectos en el proyecto

*DE*: Defecto

- Porcentaje de Clientes Satisfechos:

$$P_{CS} = \frac{(TC - CI)}{TC} * 100$$

Leyenda:

*P<sub>CS</sub>*: Porcentaje de los Clientes Satisfechos

*TC*: Total de Clientes del Proyecto

*CI*: Clientes inconformes

## **2.6. Criterios éticos.**

Los criterios éticos considerados en esta investigación fueron:

- a. Respetar a los autores de los conceptos citándolos adecuadamente, ya sean los artículos, libros o estudios utilizados.
- b. Cumplir con los principios éticos del Colegio de Ingenieros del Perú, al que se pertenecía.
- c. Cumplir los reglamentos de la Universidad.
- d. Autorizaciones del comité de ética

## **2.7. Criterios de Rigor Científico.**

Los criterios científicos que sigue esta investigación serían las siguientes según Rada (2016)

- a. Credibilidad: El cual garantizó que la presente investigación fuese realizada por el autor, fue probada con instrumentos válidos y lo resultados fueran confiables.

- b. Transferibilidad: Mediante el cual el modelo de procesos ad hoc planteado se podría aplicar en otro proyecto de software de una MYPE y obtener resultados similares.
- c. Conformabilidad: Refiere a que el modelo de procesos ad hoc se puede aplicar varias veces al mismo proyecto y entregar los mismos resultados.

### III. RESULTADOS.

#### 3.1. Resultados en Tablas y Figuras.

**3.1.1. Resultados de la Variable Independiente:** Modelo de procesos ad hoc basado en ISO/IEC 29110

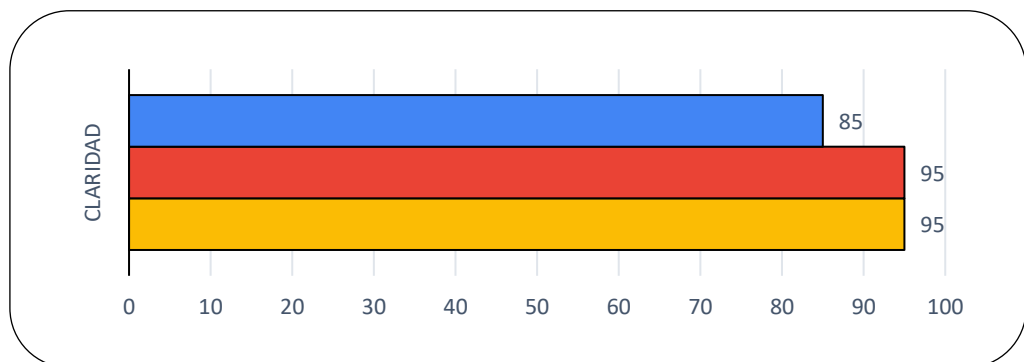
##### 3.1.1.1. Indicador “Nivel de Claridad del Modelo”

**Tabla 4**

***Resultados del Indicador - Nivel de Claridad***

Experto	Nivel
Experto 01	95.00
Experto 02	95.00
Experto 03	85.00
Promedio	91.67

*Fuente, elaboración propia.*



*Figura 1, Resultados del Indicador - Nivel de Claridad del Modelo*

*Fuente, Elaboración propia.*

Según los resultados, en cuanto al indicador “Nivel de Claridad”, se pudo evidenciar que, obtuvo un promedio de 91.67 considerando los puntajes establecidos por los tres expertos por lo que, dicho modelo ad hoc para desarrollar software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110 estaba formulado con lenguaje apropiado.



### 3.1.1.2. Indicador “Nivel de Objetividad del Modelo”

Tabla 5

#### Resultados del Indicador - Nivel de Objetividad

Experto	Nivel
Experto 01	95.00
Experto 02	95.00
Experto 03	80.00
Promedio	90.00

Fuente, elaboración propia.

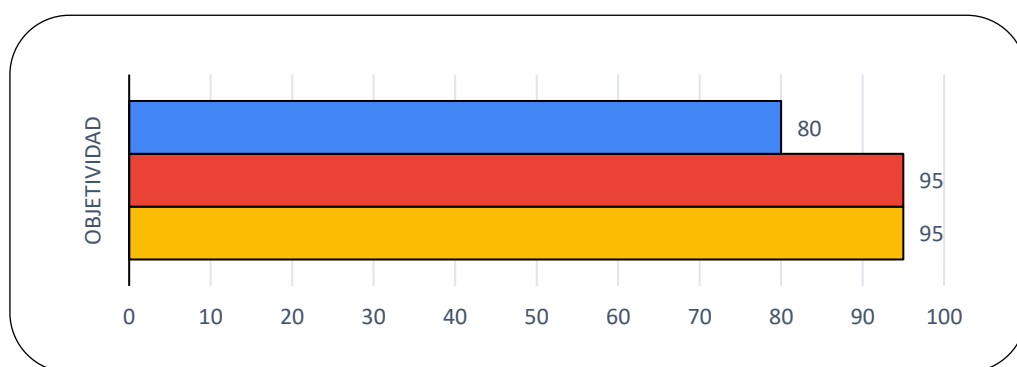


Figura 2, Resultados del Indicador - Nivel de Objetividad del Modelo

Fuente, Elaboración propia.

Según los resultados, en cuanto al indicador “Nivel de Objetividad”, se pudo evidenciar que, obtuvo un promedio de 90.00 considerando los puntajes establecidos por los tres expertos por lo que, dicho modelo de procesos ad hoc para que se desarrollen software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110 estaba expresado en conductas observables.

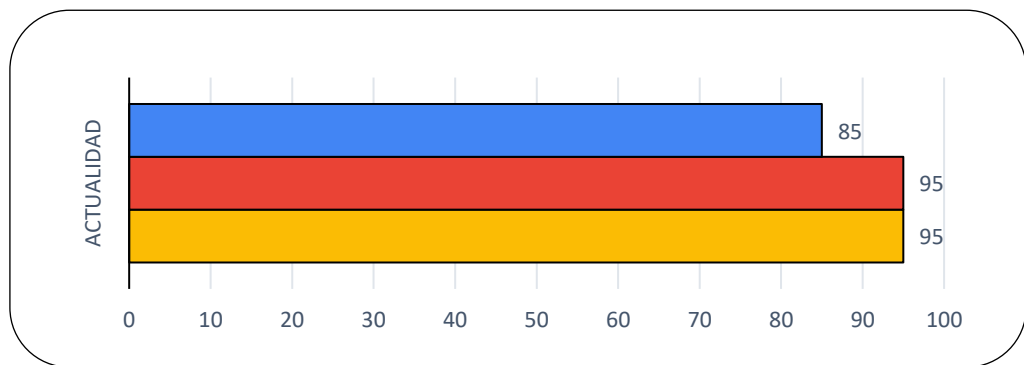
### 3.1.1.3. Indicador “Nivel de Actualidad del Modelo”

Tabla 6

#### **Resultados del Indicador - Nivel de Actualidad**

Experto	Nivel
Experto 01	95.00
Experto 02	95.00
Experto 03	85.00
Promedio	91.67

*Fuente, elaboración propia.*



*Figura 3, Resultados del Indicador - Nivel de Actualidad del Modelo*

*Fuente, Elaboración propia.*

Según los resultados, en cuanto al indicador “Nivel de Actualidad”, se pudo evidenciar que, obtuvo un promedio de 91.67 considerando los puntajes establecidos por los tres expertos por lo que, dicho modelo de procesos ad hoc para que se desarrolle el software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110 era adecuado avanzar la gestión de desarrollo del software.

### 3.1.1.4. Indicador “Nivel de Organización del Modelo”

Tabla 7

#### Resultados del Indicador - Nivel de Organización

Experto	Nivel
Experto 01	95.00
Experto 02	95.00
Experto 03	90.00
Promedio	93.33

Fuente, elaboración propia.

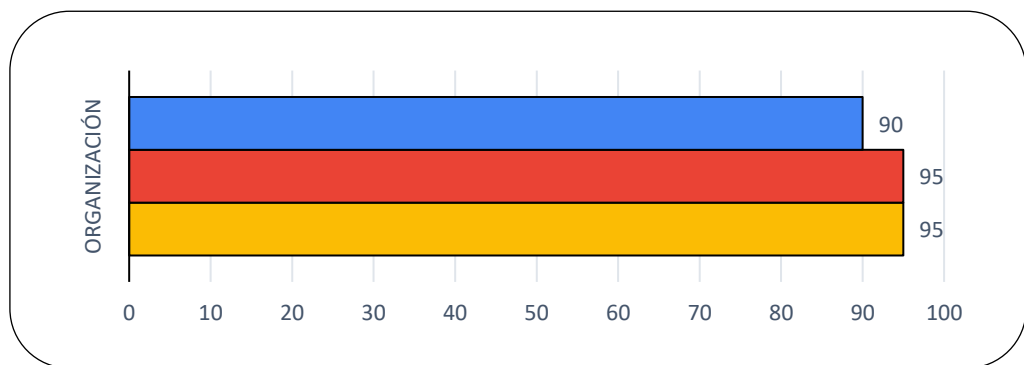


Figura 4, Resultados del Indicador - Nivel de Organización del Modelo

Fuente, Elaboración propia.

Según los resultados, en cuanto al indicador “Nivel de Organización”, se pudo evidenciar que, obtuvo un promedio de 93.33 considerando los puntajes establecidos por los tres expertos por lo que, dicho modelo de procesos ad hoc para que se desarrolle el software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110 estaba elaborado con una organización lógica.

### 3.1.1.5. Indicador “Nivel de Suficiencia del Modelo”

Tabla 8

#### Resultados del Indicador - Nivel de Suficiencia

Experto	Nivel
Experto 01	95.00
Experto 02	95.00
Experto 03	85.00
Promedio	91.67

Fuente, elaboración propia.

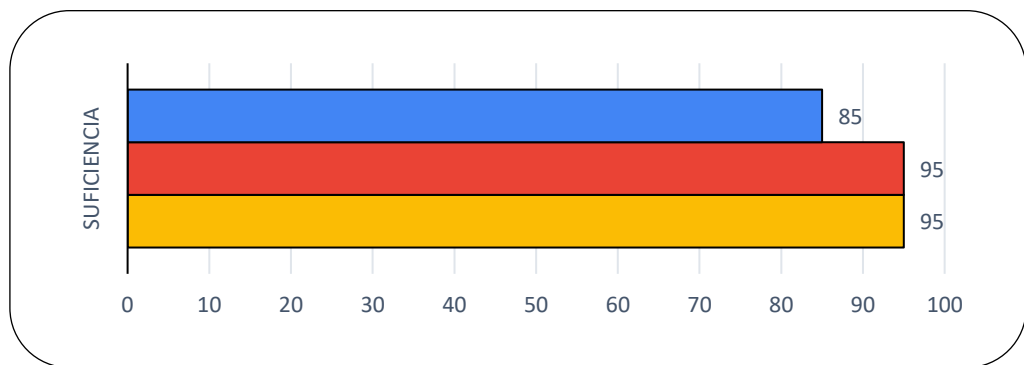


Figura 5, Resultados del Indicador - Nivel de Suficiencia del Modelo

Fuente, Elaboración propia.

Según los resultados, en cuanto al indicador “Nivel de Suficiencia”, se pudo evidenciar que, obtuvo un promedio de 91.67 considerando los puntajes establecidos por los tres expertos por lo que, dicho modelo de procesos ad hoc para que se desarrolle el software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110 comprendía aspectos en cantidad y calidad.

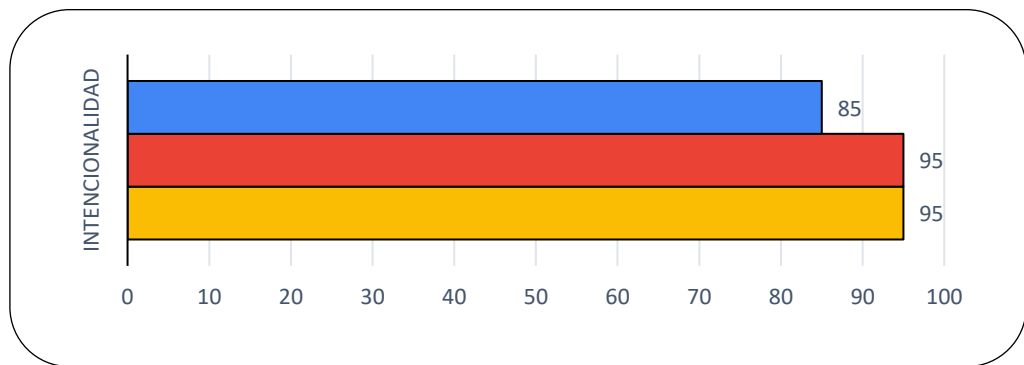
### 3.1.1.6. Indicador “Nivel de Intencionalidad del Modelo”

Tabla 9

#### **Resultados del Indicador - Nivel de Intencionalidad**

Experto	Nivel
Experto 01	95.00
Experto 02	95.00
Experto 03	85.00
Promedio	91.67

*Fuente, elaboración propia.*



*Figura 6, Resultados del Indicador - Nivel de Intencionalidad del Modelo*

*Fuente, Elaboración propia.*

Según los resultados, en cuanto al indicador “Nivel de Intencionalidad”, se pudo evidenciar que, obtuvo un promedio de 91.67 considerando los puntajes establecidos por los tres expertos por lo que, dicho modelo de procesos ad hoc para que se desarrolle el software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110 era adecuado para la valoración de la gestión de desarrollo del software.

### 3.1.1.7. Indicador “Nivel de Consistencia del Modelo”

Tabla 10

#### Resultados del Indicador - Nivel de Consistencia

Experto	Nivel
Experto 01	95.00
Experto 02	95.00
Experto 03	90.00
Promedio	93.33

Fuente, elaboración propia.

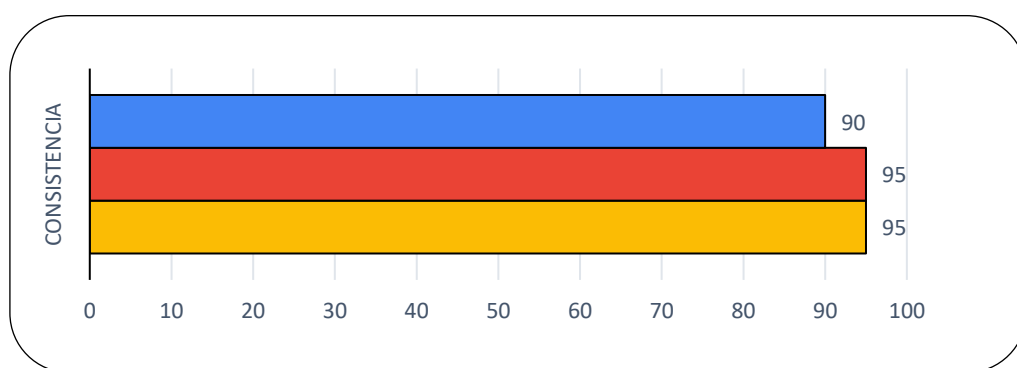


Figura 7, Resultados del Indicador - Nivel de Consistencia del Modelo

Fuente, Elaboración propia.

Según los resultados, en cuanto al indicador “Nivel de Consistencia”, se pudo evidenciar que, obtuvo un promedio de 93.33 considerando los puntajes establecidos por los tres expertos por lo que, dicho modelo de procesos ad hoc para que se desarrolle el software por licencia para una MYPE de servicios de TI en base a la ISO/IEC 29110 se encontraba basado en aspectos teóricos científicos.

### 3.1.1.8. Indicador “Nivel de Coherencia del Modelo”

Tabla 11

#### Resultados del Indicador - Nivel de Coherencia

Experto	Nivel
Experto 01	95.00
Experto 02	95.00
Experto 03	80.00
Promedio	90.00

Fuente, elaboración propia.

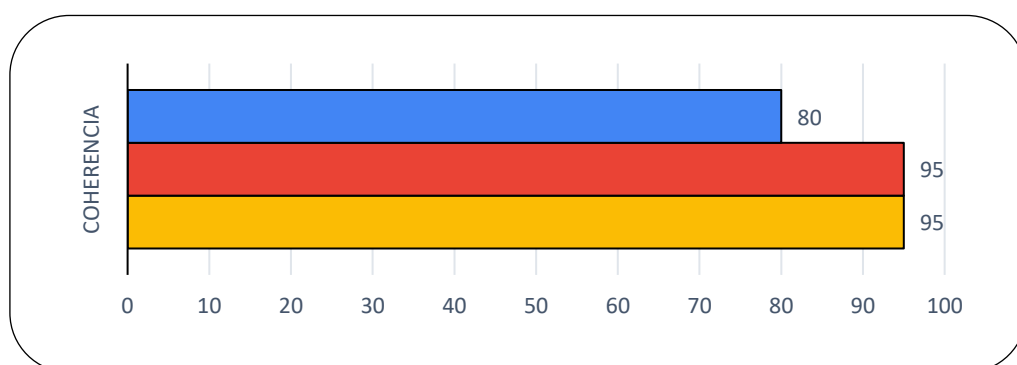


Figura 8, Resultados del Indicador - Nivel de Coherencia del Modelo

Fuente, Elaboración propia.

Según los resultados, en cuanto al indicador “Nivel de Coherencia”, se pudo evidenciar que, obtuvo un promedio de 90.00 considerando los puntajes establecidos por los tres expertos por lo que, dicho modelo de procesos ad hoc para que se desarrolle el software por licencia para una MYPE de servicios de TI en base a la ISO/IEC 29110 contaba con la relación necesaria de cada actividad de los procesos del modelo.

### 3.1.1.9. Indicador “Nivel de Metodología del Modelo”

Tabla 12

#### Resultados del Indicador - Nivel de Metodología

Experto	Nivel
Experto 01	95.00
Experto 02	95.00
Experto 03	90.00
Promedio	93.33

Fuente, elaboración propia.

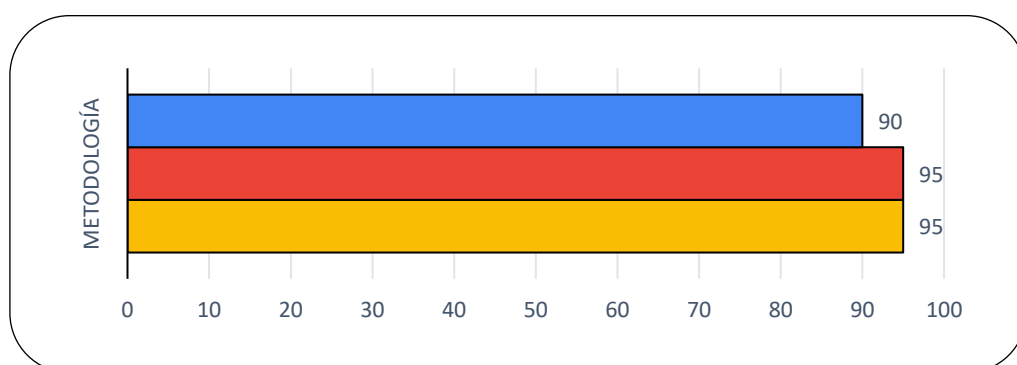


Figura 9, Resultados del Indicador - Nivel de Metodología del Modelo

Fuente, Elaboración propia.

Según los resultados, en cuanto al indicador “Nivel de Metodología”, se pudo evidenciar que, obtuvo un promedio de 93.33 considerando los puntajes establecidos por los tres expertos por lo que, dicho modelo de procesos ad hoc para que se desarrolle el software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110 respondía al propósito de la investigación.



### 3.1.1.10. Indicador “Nivel de Pertinencia del Modelo”

Tabla 13

#### Resultados del Indicador - Nivel de Pertinencia

Experto	Nivel
Experto 01	95.00
Experto 02	95.00
Experto 03	90.00
Promedio	93.33

Fuente, elaboración propia.

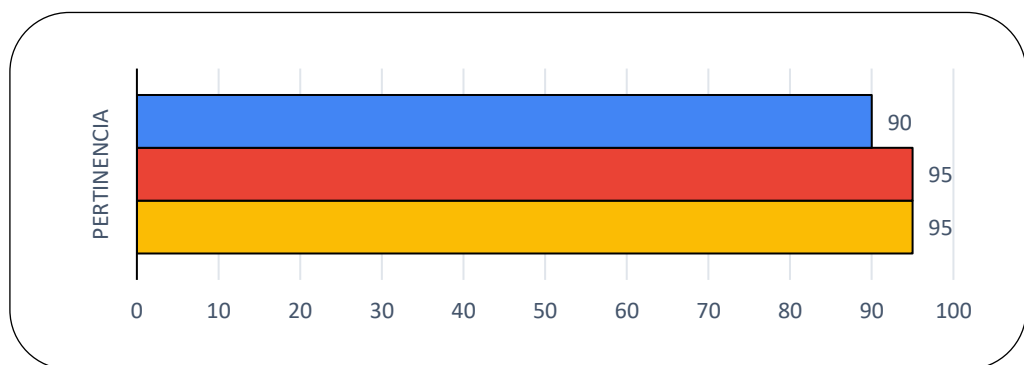


Figura 10, Resultados del Indicador - Nivel de Pertinencia del Modelo

Fuente, Elaboración propia.

Según los resultados, en cuanto al indicador “Nivel de Pertinencia”, se pudo evidenciar que, obtuvo un promedio de 93.33 considerando los puntajes establecidos por los tres expertos por lo que, dicho modelo de procesos ad hoc para que se desarrolle el software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110 era útil y apropiado para desarrollar el software.

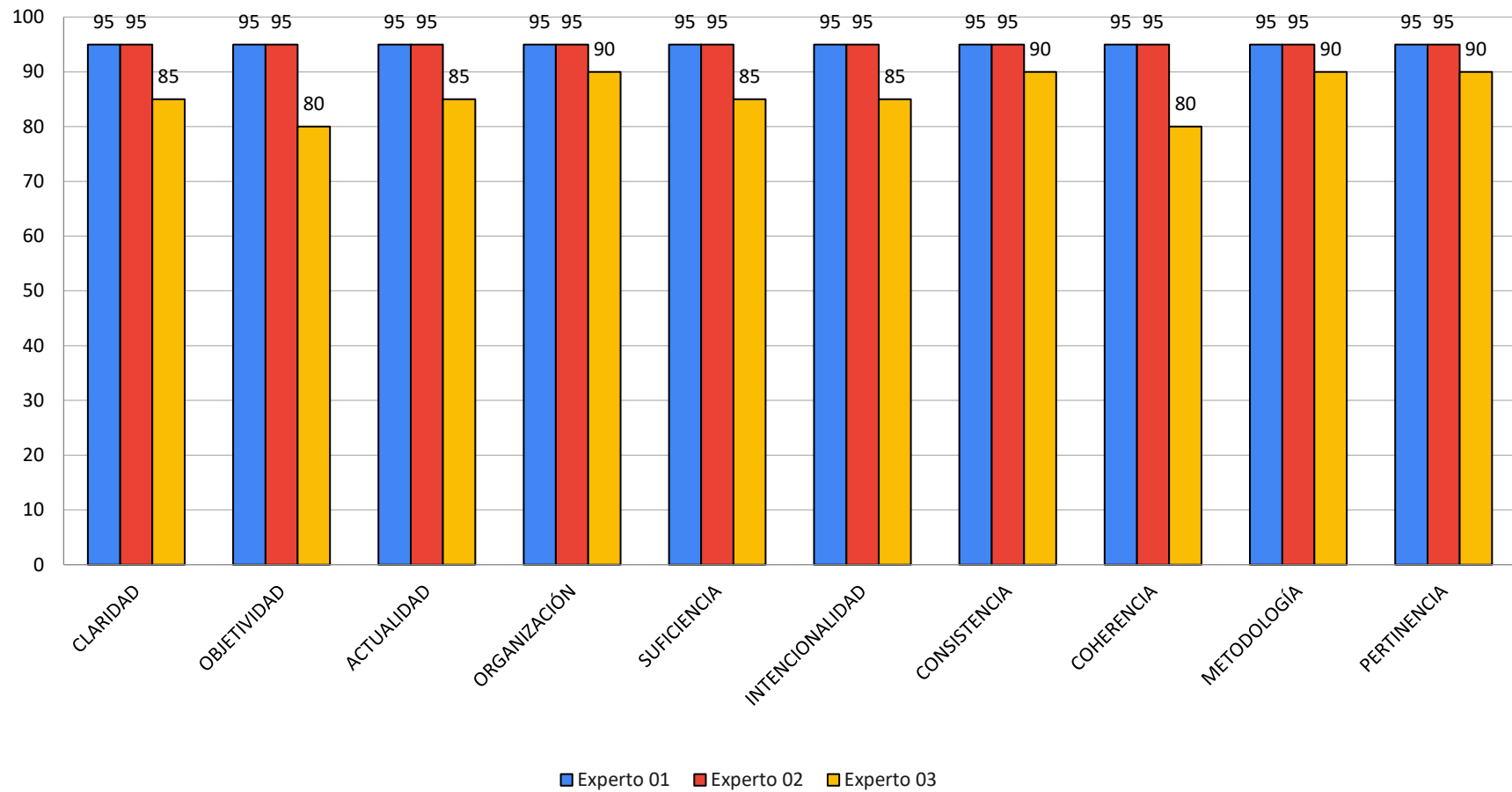


Figura 11, Resultados de la Variable Independiente

Fuente, Elaboración propia.

### 3.1.2. Resultados de la Variable Dependiente: Desarrollo de software por licencia

#### 3.1.2.1. Indicador “Cantidad de defectos”

Tabla 14

#### Resultados del Indicador - Cantidad de defectos

Nivel	Proyecto 01	Proyecto 02	Proyecto 03
Pre Test	7	8	8
Post Test	0	1	1

Nivel	Proyecto 04	Proyecto 05	Proyecto 06
-------	-------------	-------------	-------------

Fuente, elaboración propia.

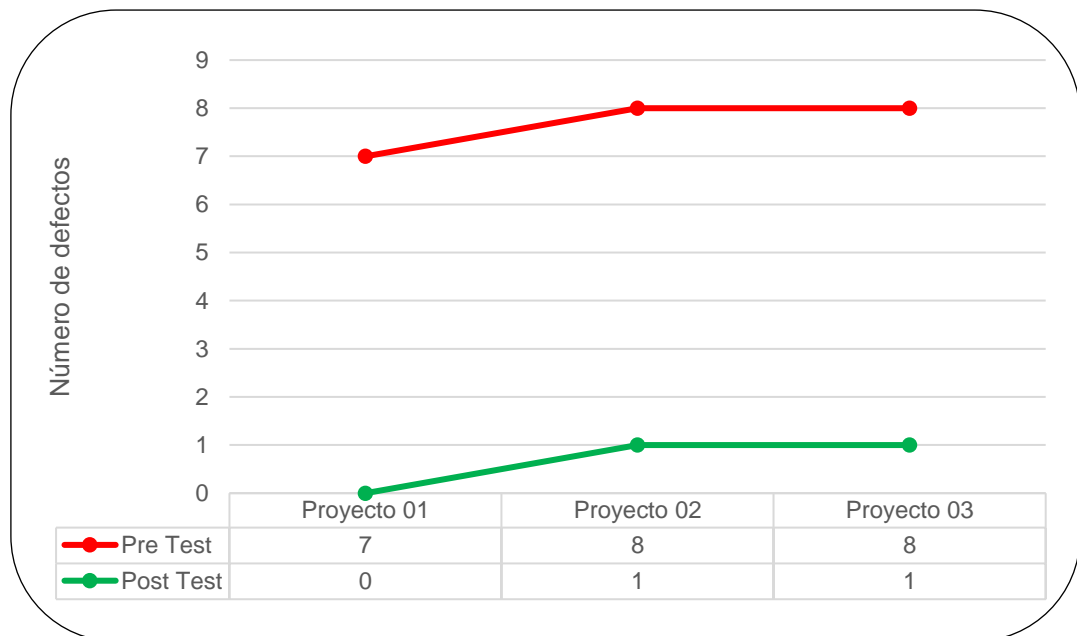


Figura 12, Resultados del Indicador - Cantidad de defectos

Fuente, Elaboración propia.

En la Figura 12 se grafican los promedios en cuanto a cantidades de defectos por grupo de proyectos correspondiente a ambos grupos, tanto pre-test, como post-test, donde quedó evidenciado un aminoramiento en veintiún (21) defectos entre ambos grupos de proyectos. Por tanto, se logró aminorar de veintitrés (23) defectos conseguidos en el conjunto pre-test a solamente dos (02) defectos obtenidos en el conjunto post-test.

### 3.1.2.2. Indicador “Porcentaje de Clientes Satisfechos”

Tabla 15

**Resultados del Indicador - Porcentaje de Clientes Satisfechos**

GRUPO	Fórmula	Resultados obtenidos de la encuesta	% Satisfacción
Pre Test	$P_{CS} = \frac{(TC - CI)}{TC} * 100$	$P_{CS} = \frac{(3 - 2)}{3} * 100$	33.33%
Post Test	$P_{CS} = \frac{(TC - CI)}{TC} * 100$	$P_{CS} = \frac{(3 - 0)}{3} * 100$	100.00%

Fuente, elaboración propia.

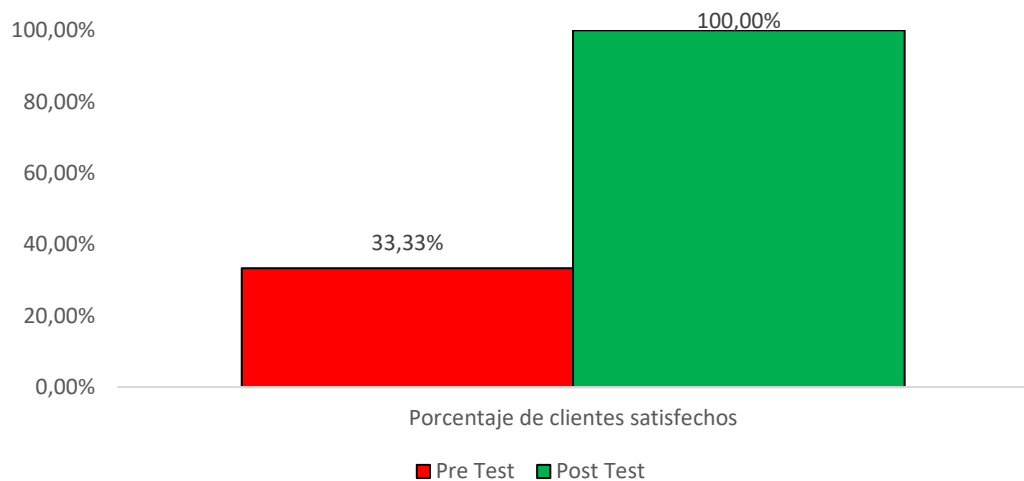


Figura 13, Resultados del Indicador - Clientes satisfechos

Fuente, Elaboración propia.

Se demostró que, los proyectos con más nivel de satisfacción de los clientes estaban en el grupo post-test con el 100.00% en menoscabo del grupo pre-test donde al inicio se tuvo 33.33%, sobre la presente entrega del servicio y/o producto software de la compañía NEDLEY SUPPORT S.A.C.

### **3.2. Discusión de resultados.**

Sobre el objetivo, caracterizar la ISO/IEC 29110 como norma base de procesos de construcción de software, se pudo evidenciar la existencia de dos (02) procesos bien definidos: Proceso de Gestión de Proyectos y Proceso de Implementación de Software, los cuales se subdividieron en cuatro y seis actividades respectivamente, así como también se logró identificar mediante diagrama SIPOC cada elemento clave en cuanto a proveedores, ingresos, procesos, salidas o clientes y, por último, se identificaron también los roles asignados a los miembros del equipo de trabajo con sus respectivas capacidades, evidenciándose con ello que el perfil básico de dicha norma ISO sirve de apoyo a las pequeñas organizaciones. Estos resultados concuerdan con los obtenidos por Mejía et al. (2019), quienes realizaron la investigación, realizaron la investigación, *ISO/IEC 29110 implementation tools proposal (basic profile)*, en México, en donde revelaron que, en la actualidad, en la industria desarrolladora de software existen modelos de calidad que sirven como hoja de ruta para dichas actividades constructivas, sin embargo, son ampliamente aceptadas en empresas de gran renombre, empero, en pequeñas organizaciones aún existe un déficit en cuanto a la aceptación de dichos modelos, por lo que la ISO/IEC 29110 emerge como uno creado justamente para esa necesidad específica del contexto menor mencionado, pero el vacío que aún no ha sido colmado se enfoca en las herramientas que pueden valerse dichas compañías para la implementación o automatización de estos modelos.

Respecto al objetivo, elaborar un diagnóstico base del estado presente del proceso de desarrollo de software de una MYPE de servicios de TI previamente seleccionada, para esta situación se consideró como caso de estudio a la compañía NEDLEY SUPPORT S.A.C., compañía trujillana de outsourcing tecnológico e Informático, la cual disponía de un proceso constructivo de software que consistía en un solo proceso de siete actividades totales, las cuales fueron presentadas mediante un análisis de procedimientos en matrices considerando las tareas de cada una de ellas, reflejándose que dicha PYME no podía permitirse los recursos para establecer procesos de software como definidos por las normas actuales y modelos de madurez existentes, verbigracia,

las ISO. Estos resultados concuerdan con los obtenidos por Castro et al. (2021), quienes realizaron la investigación, *Proposal for the implementation of a quality standard for software development projects of a small company*, en la Universidad Peruana de Ciencias Aplicadas, en Perú, en donde revelaron que, la construcción de software se ha constituido en una actividad de gran relevancia en cuanto al mercado de la ingeniería del software, por sobre otras actividades, en las que destacan las pequeñas organizaciones, quienes ocupan más del 90% de la totalidad de compañías que se dedican a este rubro desarrollador de software, empero, existe una brecha en cuanto a la usanza de modelos o estándares que les permitan llevar a cabo un correcto proceso constructivo de software.

Respecto al objetivo, diseñar un modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI, se llevó a cabo dicho modelamiento haciendo usanza de la herramienta de modelado de procesos de negocio BIZAGI, en su versión v.4.0.0.014, en la que se construyó el modelo considerando dos procesos (02) bien definidos: Proceso de Gestión de Proyecto (PGP) y Proceso de Implementación de Software (PIS), los cuales se subdividieron en tres y cuatro actividades respectivamente, así como también se lograron identificar formatos, objetivos, alcances, tareas, diagramas SIPOC y diagramas BIZAGI de cada una de las siete (07) actividades totales del modelo propuesto para la compañía NEDLEY SUPPORT S.A.C. Los resultados se cotejan con los revelados por Huapaya et al. (2021), quienes realizaron la investigación, *Proposal for the implementation of the ISO/IEC 29110 part 5 - 2011 standard (basic profile) to improve the project management and software implementation processes of a Peruvian software development company*, en Perú, en donde presentaron la implementación del estándar ISO/IEC TR 29110-5-1-2:2011 en su perfil básico, considerando cuatro (04) actividades para el PGP y seis (06) actividades para el PIS, todas ellas modeladas en Bizagi Modeler, donde además, mostraron que, existía una brecha en el proceso de Gestión de Proyectos (45%) y en el proceso de Implementación del Software (40%) que se venía desarrollando en la compañía caso de estudio, logrando

mediante su modelo el aminoramiento en cuanto a costes y el mejoramiento de los procesos constructivos de software.

Respecto al objetivo, validar mediante juicio de expertos la propuesta del modelo de procesos ad hoc desarrollado, el modelo diseñado previamente y que consta de siete (07) actividades totales fue validado por juicio de expertos quienes debieron poseer ciertas características para ser considerados como tal y, a los cuales, posteriormente, se les remitió una ficha mediante correo electrónico en la que, calificaron el modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110 considerando diez (10) indicadores, lo que conllevó a tener como resultado, un nivel de aceptación de 92%, por lo que se consideró que dicho modelo desarrollado poseía una valoración considerada como “Muy Buena”. Estos resultados concuerdan con los mencionados por Buchalcevova (2019), quien realizó la investigación, *Using ArchiMate to model ISO/IEC 29110 standard for very small entities*, en la Prague University of Economics and Business, en República Checa, en donde desarrolló un metamodelo constructivo de software para pequeñas organizaciones checas basado en el perfil básico de la norma ISO/IEC 29110 y el lenguaje de modelamiento ArchiMate; considerando cuatro (04) actividades para el proceso de Gestión de Proyectos y seis (06) actividades para el proceso de Implementación de Software; el cual fue posteriormente evaluado mediante entrevistas cualitativas semi estructuradas constituidas por quince (15) preguntas a cuatro (04) expertos en desarrollo de software, el cual, según la apreciación de los expertos, era ventajoso al mostrar una visualización más grande (100%), ayudaba en la comprensión del contexto constructivo (100%), permitió personalizar de manera más sencilla el perfil básico de la ISO en mención (75%) y mejora la comunicación dentro del equipo desarrollador (90%).

Respecto al objetivo, realizar una prueba piloto del modelo de procesos ad hoc para el desarrollo de software por licencia en una MYPE de servicios de TI, se realizó dicha prueba piloto partiendo de la necesidad de la compañía caso de estudio de poseer un modelo que le permita seguir una hoja de ruta en cuanto

a sus procesos constructivos de software, por lo que se siguieron, las siete (07) actividades de dicho modelo propuesto, evidenciándose un aminoramiento en cuanto al número de defectos en dos grupos de proyectos PRE TEST y POST TEST, así como también, un mejoramiento en el grado de satisfacción del cliente llegando a obtener un 100% en dicho indicador, reflejando la mejora del desarrollo de software por licencia en dicha compañía trujillana caso de estudio. Estos resultados concuerdan con los obtenidos por Pérez & Giraldo (2019) quienes realizaron la investigación, *Proposal of a Software Development Process for a Poultry Company in Colombia*, en Colombia, en donde, luego de implementar una mixtura del estándar ISO/IEC 29110-5-1-2:2018 con el modelo IDEAL, considerando cuatro (04) actividades para el PGP y seis (06) actividades para el PIS, logrando un aminoramiento del 81% en retrasos, asimismo, en cuanto a la cantidad de proyectos en el nivel PRE TEST se dio cumplimiento a 3 entregas parciales a tiempo mientras que en el nivel POST TEST se dio cumplimiento a 23 entregas parciales evidenciándose un incremento del 766% por lo que, la prueba piloto realizada en dicha compañía colombiana caso de estudio logró evidenciar el aumento de la credibilidad de los procesos constructivos de software.

### **3.3. Aporte práctico.**

#### **3.3.1. Caracterizar ISO/IEC 29110 como norma base de procesos de construcción de software**

Para describir las características más importantes de la norma ISO/IEC 29910, se utilizó la guía de gestión e Ingeniería de la propia norma.

##### **a.1. Caracterización Planificación del proyecto**

Durante este proceso se establecen las actividades, presupuesto, equipo y tiempos los cuales se utilizarán durante el desarrollo del proyecto, esto mediante estimaciones y evaluaciones de los diferentes aspectos para tener en cuenta, todo esto debido a que con una sólida y buena planeación se puede sentar las bases para un proyecto el cual se desarrollara con los mínimos inconvenientes



posibles.

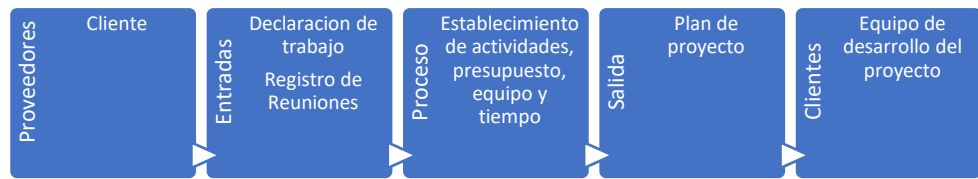


Figura 14, Caracterización “Planificación del proyecto”

Fuente, Elaboración propia.

### a.2. Caracterización Ejecución del plan del proyecto

En este proceso se empieza a desarrollar el plan de proyecto el cual fue establecido durante el proceso anterior, mediante el uso del plan de proyecto se definen las primeras tareas a desarrollar, con los responsables y todo esto con el fin de poder declarar oficialmente el inicio del desarrollo del proyecto.

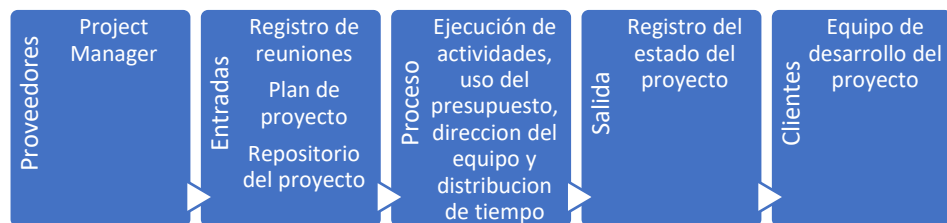


Figura 15, Caracterización “Ejecución del plan del proyecto”

Fuente, Elaboración propia.

### a.3. Caracterización Evaluación y control del proyecto

En este proceso se comprueba del avance del proyecto de acuerdo con el plan del proyecto, esto se realiza mediante las reuniones con el equipo de trabajadores para poder hacer la contrastación con el plan de proyecto, esto con el fin de poder evaluar cómo es que se han ido desarrollando las tareas asignadas y planificadas.

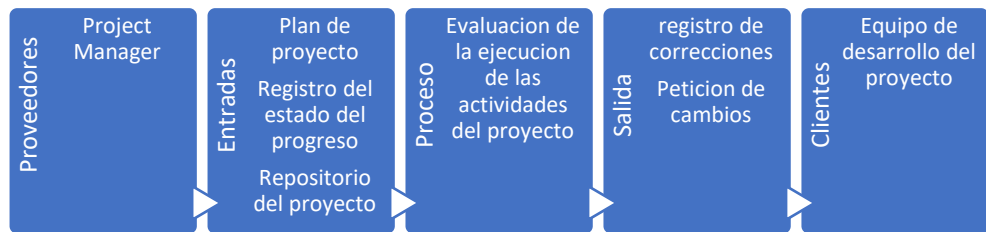


Figura 16, Caracterización “Evaluación y control del proyecto”

Fuente, Elaboración propia.

#### a.4. Caracterización Cierre del proyecto

La finalización del proyecto o también conocido como el cierre de este, en ese proceso se revisa que ya se hayan cumplido con cada tarea descrita en los planes del proyecto, se genera la documentación para el uso del producto y finalmente la versión final del producto a entregar, todo esto con la finalidad de poder cumplir con lo acordado con el cliente y así poder finalizar el proyecto.

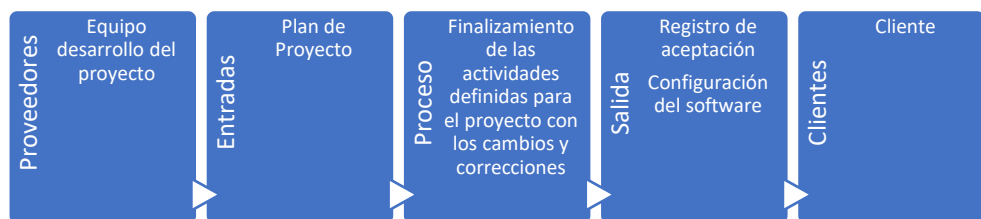


Figura 17, Caracterización “Cierre del proyecto”

Fuente, Elaboración propia.

#### b.1. Caracterización Inicio de la implementación del software

Durante este proceso se establecen las actividades, presupuesto, equipo y tiempos los cuales se utilizarán durante el desarrollo del proyecto, esto mediante estimaciones y evaluaciones de los diferentes aspectos para tener en cuenta, todo esto debido a que, con una sólida y buena planeación, se puede sentar las bases para un proyecto el cual se desarrollara con los mínimos inconvenientes posibles.

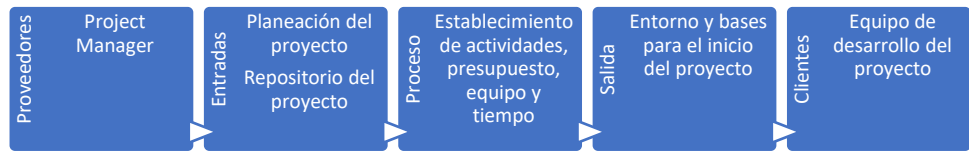


Figura 18, Caracterización “Inicio de la implementación del software”

Fuente, Elaboración propia.

### b.2. Caracterización Análisis de requisitos software

En este procedimiento, es que se establecen los diferentes requerimientos con los cuales el software debe cumplir, a fin de poder cubrir con todas las necesidades del cliente, normalmente después de haber sentado las bases del proyecto se tiene una reunión para poder acordar cuáles serán los requerimientos, todo esto como ya se reiteró antes para poder satisfacer al cliente.



Figura 19, Caracterización “Análisis de requisitos software”

Fuente, Elaboración propia.

### b.3. Caracterización Arquitectura de software y diseño detallado

En este proceso se define cuál será la arquitectura que se utilizará para poder desarrollar el producto, además de los mockups los cuales se utilizaran para la guía de la construcción de la UI, todo esto con la finalidad de poder tener una visión general del desarrollo del software, esto gracias al orden que brinda el uso de arquitecturas de software y los mockups.

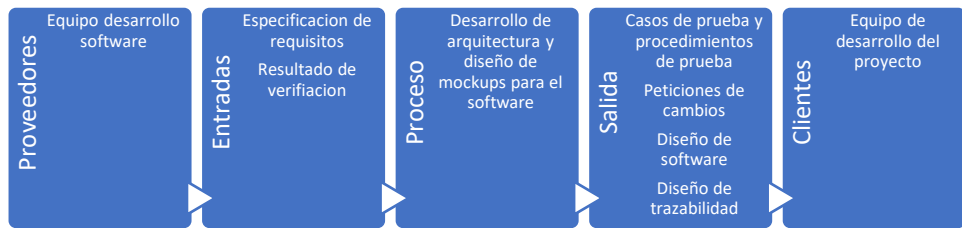


Figura 20, Caracterización “Arquitectura de software y diseño detallado”

Fuente, Elaboración propia.

#### b.4. Caracterización Construcción de software

En este procedimiento se construye la parte central del producto que tiene como fin el proyecto, esto mediante el uso de las diferentes guías que se han elaborado anteriormente como lo puede ser el plan de proyecto, la arquitectura de software y los mockups, todo con el fin de que el desarrollo se realice con el menor número de inconvenientes posibles.



Figura 21, Caracterización “Construcción de software”

Fuente, Elaboración propia.

#### b.5. Caracterización Integración del software y pruebas

En este proceso se realizan las pruebas para los distintos requerimientos estipulados anteriormente, la comprobación de la correcta implementación de los requerimientos se realiza a través de diferentes pruebas dependiendo del tipo de requerimientos (funcional o no funcional), además de los criterios de aceptación que se hayan planteado previamente, todo esto con la finalidad de que se asegure la satisfacción total de las necesidades del cliente.

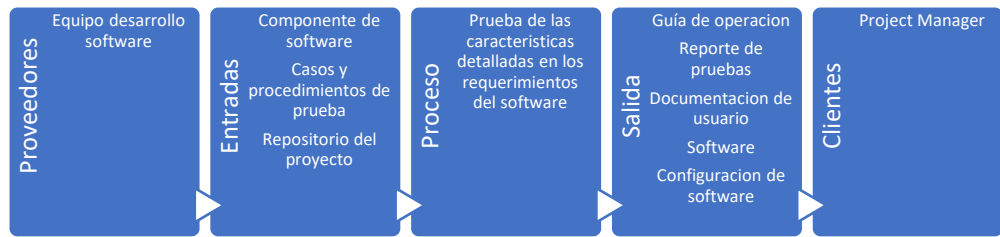


Figura 22, Caracterización “Integración de software y pruebas”

Fuente, Elaboración propia.

### b.6. Caracterización Entrega del producto

La finalización del proyecto o también conocido como el cierre de este, en ese proceso se revisa que ya se hayan cumplido con cada tarea descrita en los planes del proyecto, se genera la documentación para el uso del producto y finalmente la versión final del producto a entregar, todo esto con la finalidad de poder cumplir con lo acordado con el cliente y así poder finalizar el proyecto.



Figura 23, Caracterización “Entrega del producto”

Fuente, Elaboración propia.

Asimismo, en cuanto a los roles implicados en el ciclo vital del desarrollo de software según la ISO/IEC 29110, se cuentan con los siguientes:

#### Tabla 16

#### Ficha de rol “Analista”

<b>Identificador:</b>	AN
<b>Nombre:</b>	Analista (Analyst)
<b>Capacidades:</b>	<ul style="list-style-type: none"> <li>▪ Conocimiento y experiencia que permita obtener, especificar y analizar los requisitos.</li> <li>▪ Conocimiento en el diseño de interfaces de usuario y criterios ergonómicos.</li> <li>▪ Conocimiento de técnicas de revisión.</li> <li>▪ Conocimiento de Técnicas de edición.</li> <li>▪ Experiencia en desarrollo y mantenimiento de software.</li> </ul>

Fuente: (NTP-RT-ISO/IEC TR 29110-5-1-2, 2012, pág. 51)

## Tabla 17

### *Ficha de rol “Cliente”*

<b>Identificador:</b>	CL
<b>Nombre:</b>	Cliente (Customer)
<b>Capacidades:</b>	<ul style="list-style-type: none"> <li>▪ Conocimiento de los procesos del Cliente y habilidad para explicar los requerimientos del cliente.</li> <li>▪ El Cliente (representante del Cliente) debe tener la autoridad para aprobar los requisitos y sus cambios.</li> <li>▪ El cliente incluye usuarios representativos con la finalidad de asegurar que el entorno operacional sea dirigido de forma correcta.</li> <li>▪ Conocimiento y experiencia en el dominio de la aplicación.</li> </ul>

Fuente: (NTP-RT-ISO/IEC TR 29110-5-1-2, 2012, pág. 52)

## Tabla 18

### *Ficha de rol “Diseñador”*

<b>Identificador:</b>	DI
<b>Nombre:</b>	Diseñador (Designer)
<b>Capacidades:</b>	<ul style="list-style-type: none"> <li>▪ Conocimiento y Experiencia en Componente de Software y diseño de Arquitectura.</li> <li>▪ Conocimiento de técnicas de revisión.</li> <li>▪ Conocimiento y Experiencia en Planificación y ejecución de pruebas de integración.</li> <li>▪ Conocimiento de técnicas de edición.</li> <li>▪ Experiencia en desarrollo y mantenimiento de software.</li> </ul>

Fuente: (NTP-RT-ISO/IEC TR 29110-5-1-2, 2012, pág. 52)

## Tabla 19

### *Ficha de rol “Programador”*

<b>Identificador:</b>	PR
<b>Nombre:</b>	Programador (Programmer)
<b>Capacidades:</b>	<ul style="list-style-type: none"> <li>▪ Conocimiento y Experiencia en Programación, integración y pruebas unitarias.</li> <li>▪ Conocimiento de técnicas de revisión.</li> <li>▪ Conocimiento de técnicas de edición.</li> <li>▪ Experiencia en desarrollo y mantenimiento de software.</li> </ul>

Fuente: (NTP-RT-ISO/IEC TR 29110-5-1-2, 2012, pág. 52)

**Tabla 20**

***Ficha de rol “Gestor de Proyecto”***

<b>Identificador:</b>	GP
<b>Nombre:</b>	Gestor de Proyecto (Project Manager)
<b>Capacidades:</b>	<ul style="list-style-type: none"> <li>▪ Capacidad de liderazgo con experiencia para toma de decisiones, planificación, gestión de personal, delegación y supervisión, conocimiento de finanzas y desarrollo de software.</li> </ul>

Fuente: (NTP-RT-ISO/IEC TR 29110-5-1-2, 2012, pág. 52)

**Tabla 21**

***Ficha de rol “Líder Técnico”***

<b>Identificador:</b>	LT
<b>Nombre:</b>	Líder Técnico (Technical Leader)
<b>Capacidades:</b>	<ul style="list-style-type: none"> <li>▪ Conocimiento y Experiencia en el dominio del proceso de Software.</li> </ul>

Fuente: (NTP-RT-ISO/IEC TR 29110-5-1-2, 2012, pág. 53)

**Tabla 22**

***Ficha de rol “Equipo de Trabajo”***

<b>Identificador:</b>	ET
<b>Nombre:</b>	Equipo de Trabajo (Work Team)
<b>Capacidades:</b>	<ul style="list-style-type: none"> <li>▪ Conocimiento y Experiencia de acuerdo a sus roles dentro del proyecto: LT, AN, DIS y/o PR.</li> <li>▪ Conocimiento de los estándares usados por el Cliente y/o por la PO.</li> </ul>

Fuente: (NTP-RT-ISO/IEC TR 29110-5-1-2, 2012, pág. 53)

**3.3.2. Elaborar un diagnóstico base del estado actual del proceso de desarrollo de software de una MYPE de servicios de TI previamente seleccionada**

Acorde a los procedimientos planteados, se presenta el diagnóstico de la empresa. Para esto se realizaron diferentes listas con los puntos a cumplir presentados por la metodología y se revisa cuales cumple y cuales no cumple la empresa, se realizó análisis de procedimientos antes de la implementación del modelo de procesos propuesta:

## 1. Planeación del proyecto

**Tabla 23**

### ***Diagnóstico PGP-ACT-01: Planeación del Proyecto***

<b>Tareas</b>	<b>Cumple</b>	<b>No Cumple</b>
Primero se deben identificar todas las tareas que se deben realizar a lo largo de todo el proyecto.	X	
Se debe establecer los prerrequisitos para cada una de las tareas que deben ser realizadas.		X
Se debe determinar cuál será el flujo de desarrollo para cada una de las tareas identificadas.		X
Se debe establecer los posibles flujos alternos para cada una de las tareas		X
Hay que determinar el nivel de complejidad para cada una de las tareas.		X
Acordar el nivel de importancia de la tarea.	X	



Posteriormente hay que definir el tiempo aproximado de desarrollo teniendo en cuenta la dificultad. X

*Fuente, elaboración propia.*

## 2. Despliegue de Plan del proyecto

**Tabla 24**

### ***Diagnóstico PGP-ACT-02: Despliegue de Plan del Proyecto***

<b>Tareas</b>	<b>Cumple</b>	<b>No Cumple</b>
Se debe registrar quienes serán parte del equipo de trabajo para el desarrollo de las tareas establecidas.	<b>X</b>	
Establecer cuáles son los roles que deberán ser desempeñados durante el desarrollo.		<b>X</b>
Limitar las responsabilidades para cada uno de los roles los cuales han sido identificados.		<b>X</b>
Identificar las aptitudes que debe tener la persona encargada de cada uno de los roles.		<b>X</b>
Evaluar a los miembros del equipo para encontrar a la persona más adecuada para el rol.		<b>X</b>
Informar de manera general a todo el equipo quienes serán los encargados de cada uno de los roles para el conocimiento general de cada miembro del equipo.	<b>X</b>	
Aceptación de cada miembro del equipo de los roles asignados.		<b>X</b>

Retroalimentación con los miembros del equipo en busca de aclarar dudas con respecto a las responsabilidades de los roles establecidos. X

---

*Fuente, elaboración propia.*

### 3. Evaluación y Monitoreo del Proyecto

**Tabla 25**

***Diagnóstico PGP-ACT-03: Evaluación y Monitoreo del Proyecto***

Tareas	Cumple	No Cumple
Recobrar la relación de las tareas establecidas para el proyecto.		x
Recobrar la relación de los miembros del equipo junto con sus roles y sus responsabilidades.		X
Análisis del número de participantes por cada una de las tareas a realizar durante el proyecto		X
Análisis del costo operativo por cada una de las tareas, así como costos de materiales.	x	
Análisis de tiempo aproximado para la realización para cada una de las tareas teniendo en cuenta su importancia y su complejidad.	x	
Reunión con los miembros del equipo para contrastar la estimación de tiempo con la	X	

estimación de los encargados de la elaboración de las tareas.

Reunión entre los encargados de la planificación para deliberar estimaciones  
 Contraste entre la estimación de costo total contra el presupuesto del proyecto.

**X**

Contraste entre la estimación de tiempo contra el tiempo asignado para la culminación total del proyecto.

**x**

Registro de las estimaciones de costo, precio y esfuerzo para futura consulta en caso de inconvenientes

**x**

---

*Fuente, elaboración propia.*

#### **4. Análisis de Requerimientos de Software**

**Tabla 26**

***Diagnóstico PIS-ACT-01: Análisis de Requerimientos de Software***

<b>Tareas</b>	<b>Cumple</b>	<b>No Cumple</b>
Identificar cuáles son las tecnologías que se utilizarán para el desarrollo del producto.	<b>x</b>	
Realizar la descargar de todos los recursos relacionados a la instalación de las tecnologías que se vayan a utilizar.	<b>x</b>	
Revisar la documentación para la correcta instalación y ubicación dentro de los archivos del proyecto.		<b>x</b>

---

Prueba de concepto para verificar el correcto funcionamiento del recurso.	<b>x</b>
Antes de hacer cambios, siempre revisar la documentación de las tecnologías a utilizar.	<b>x</b>
Una vez que todo se haya instalado se debe hacer una copia de seguridad del ambiente de desarrollo operativo.	<b>x</b>
Subir la copia de seguridad al repositorio git que se asignó para el proyecto.	<b>x</b>

---

*Fuente, elaboración propia.*

## 5. Diseño y Arquitectura de Software

**Tabla 27**

### ***Diagnóstico PIS-ACT-02: Diseño y Arquitectura del Software***

<b>Tareas</b>	<b>Cumple</b>	<b>No Cumple</b>
Se debe realizar una reunión con todo el equipo de trabajo para hacer las consultas pertinentes con respecto a las tareas asignadas.	<b>x</b>	
Cada miembro del equipo que desarrolla el proyecto debe expresar cual es el estado actual de la tarea que se está desarrollando.	<b>x</b>	
Se debe consultar el documento de estimación de tiempos para ver si el progreso está dentro de los estimado.		<b>x</b>

En caso de que se hallara un atraso con respecto al progreso del proyecto debe anotarse.		x
Preguntar al encargado de la tarea que presenta retraso la causa por la cual se ha dado ese atraso.		x
En caso de ser un factor externo se debe ajustar la estimación de tiempo.	x	
En caso de ser un factor interno, brindar la asistencia necesaria al trabajador para la regulación del tiempo de finalización de la tarea.		x
Una vez identificado todo el progreso se debe actualizar el documento de tareas del proyecto, asimismo el documento de estimación de tiempos.		x
Finalmente informar a los miembros del equipo cuando será la próxima reunión para la posterior actualización del progreso del proyecto	x	

---

*Fuente, elaboración propia.*

## 6. Construcción del Software

**Tabla 28**

***Diagnóstico PIS-ACT-03: Construcción del Software***

Tareas	Cumple	No Cumple
--------	--------	-----------

---

Recuperar la relación de las tareas que se identificaron para la elaboración de proyecto.	<b>x</b>	
Identificar las tareas que cuentan con el mayor grado de complejidad e importancia por igual.	<b>x</b>	
Llamar al encargado de realizar dicha tarea.	<b>x</b>	
Reunirse para compartir inquietudes del encargado y poder tenerlas en cuenta en el caso de que se pueda cumplir tal escenario.		<b>x</b>
Identificar factores externos los cuales pueden afectar al desarrollo del proyecto.		<b>x</b>
Identificar factores internos los cuales puedan afectar al desarrollo normal del proyecto.	<b>x</b>	
Realizar planes de contingencia para todos los riesgos identificados.	<b>x</b>	
Registro de las medidas de contención para los riesgos identificados.		<b>x</b>
Comunicación con todo el equipo de trabajo acerca de las medidas a ser tomadas cuando se suscite alguno de los riesgos que fueron identificados a lo largo de este procedimiento.		<b>x</b>

---

*Fuente, elaboración propia.*

## 7. Entrega del producto

Tabla 29

### **Diagnóstico PIS-ACT-04: Entrega del Software**

<b>Tareas</b>	<b>Cumple</b>	<b>No Cumple</b>
Identificación de la versión estable del producto final que será entregada al cliente.	<b>x</b>	
Creación de la documentación para el correcto uso del producto final.		<b>x</b>
Revisión del manual para el uso del producto final.		<b>x</b>
Acordar la reunión con el cliente para poder hacer la entrega formal.	<b>x</b>	
capacitación en el uso del producto final.	<b>x</b>	
aclaración de las condiciones para el soporte técnica y de futuras actualización en caso de fallo del producto final.		<b>x</b>
Reconocimiento por parte del cliente para poder certificar su satisfacción con la entrega del producto final		<b>x</b>
Entrega final del producto junto con la documentación apropiada y también del acuerdo de soporte técnico		<b>x</b>

*Fuente, elaboración propia.*

### **3.3.3. Diseñar un modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI**

Para el diseño del modelo de procesos ad hoc, se requirió primeramente de un análisis comparativo entre los procesos y actividades de desarrollo de software por licencia de la compañía NEDLEY SUPPORT S.A.C., los procesos y actividades de desarrollo de software según el perfil básico de la ISO/IEC 29110 y los procesos y actividades del modelo de procesos ad hoc propuesto. Para ello, se diseñó una matriz comparativa procesos propuestos vs procesos ISO/IEC 29110 vs procesos actuales, la cual se muestra a continuación en la Tabla 30:



Matriz de coherencia entre las actividades/procesos/tareas y la ISO-29110 con los procesos actuales de la empresa

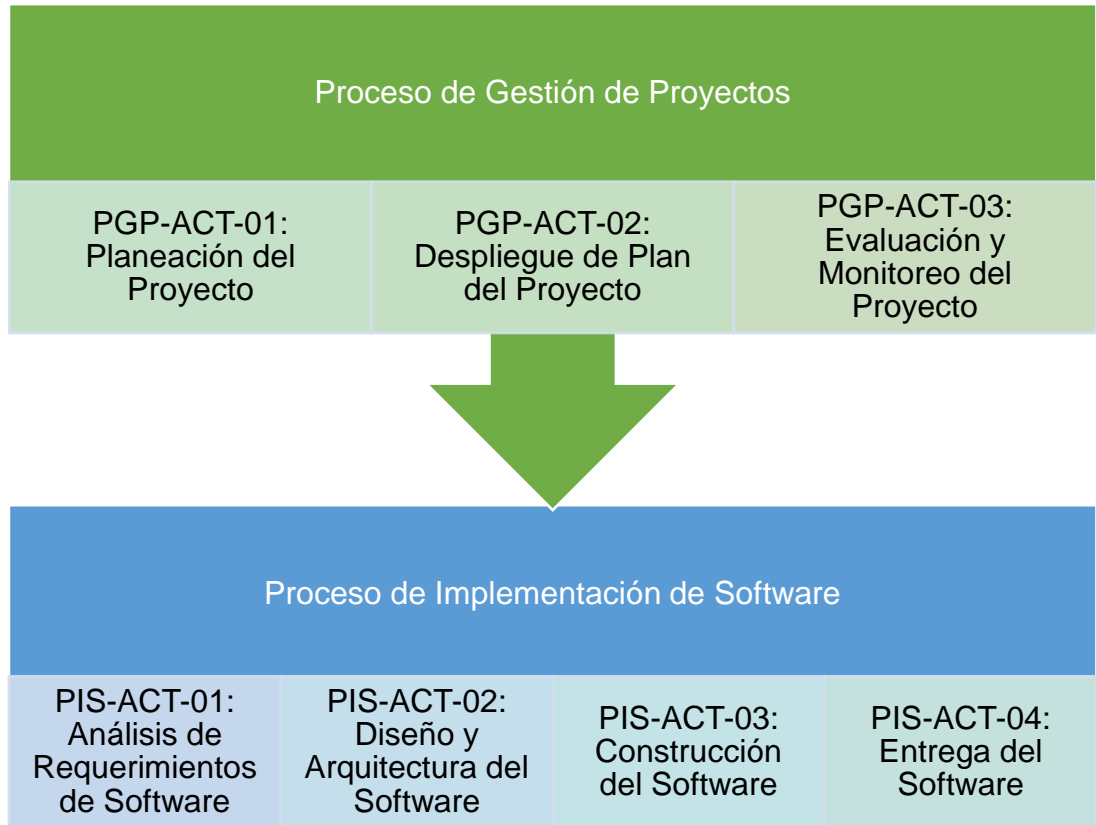
Tabla 30

Matriz comparativo procesos propuestos vs procesos ISO/IEC 29110 vs procesos actuales

ISO 29110	Planificación de proyecto	Ejecución del plan de proyecto	Evaluación y control del proyecto	Cierre del proyecto	Inicio de la implementación de software	Análisis de requisitos de software	Arquitectura de software y diseño detallado	Construcción de software	integración de software y pruebas	Entrega del producto
<b>Modelo de procesos ad hoc propuesto</b>	Planeación del proyecto	Despliegue de Plan del proyecto	Evaluación y Monitoreo del Proyecto			Análisis de Requerimientos de Software	Diseño y Arquitectura del Software	Construcción del Software		Entrega del Software
<b>Procesos de la empresa</b>										
Designación de tareas	x					x				
Definición de Requerimientos	x					x				
Asignación de tareas		x	x							
Control de avance							x	x		
Construcción de software			x					x		
Pruebas									x	
Entrega del producto				x						x

Fuente, Elaboración propia.

El modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110 consta de dos (02) procesos a saber, los cuales cuentan con sus respectivas actividades según se muestra a continuación:



*Figura 24, Procesos del modelo ad hoc propuesto*

*Fuente, Elaboración propia.*

En cuanto al Proceso de Gestión de Proyectos, se contó con las siguientes tres (03) actividades:

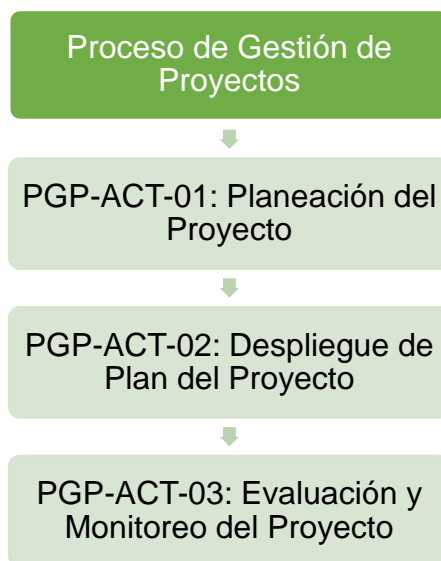


Figura 25, Actividades del Proceso de Gestión de Proyectos

Fuente, Elaboración propia.

### 1. PGP-ACT-01: Planeación del Proyecto

En primer lugar, durante este procedimiento es el cliente el que con ayuda del formato que se entregará a continuación ayudará al equipo a registrar todos los requerimientos que se tienen que cumplir durante el proyecto.

Tabla 31

*Formato de registro de tareas por realizar*

Formato de registro de tareas por realizar			
N° Tarea:		Nombre:	
Complejidad		Importancia	
Tiempo Aproximado			
<b>Pre-Requisitos</b>			
<b>Flujo regular</b>		<b>Flujo Alterno</b>	

A continuación, se dejará el detallado de todo el procedimiento para que se pueda comprender de mejor manera.

## OBJETIVO

El propósito de este procedimiento es establecer cuáles serán las tareas para realizar en el proyecto, asimismo cuáles serán los factores que van a intervenir al momento de realizarla junto con su duración.

## ALCANCE

Este procedimiento es aplicable para todos los equipos de desarrolladores que pertenecen a la organización.

## TAREAS

- a. Primero se deben identificar todas las tareas que se deben realizar a lo largo de todo el proyecto.
- b. Se debe establecer los prerequisites para cada una de las tareas que deben ser realizadas.
- c. Se debe determinar cuál será el flujo de desarrollo para cada una de las tareas identificadas.
- d. Se debe establecer los posibles flujos alternos para cada una de las tareas
- e. Hay que determinar el nivel de complejidad para cada una de las tareas
- f. Acordar el nivel de importancia de la tarea.
- g. Posteriormente hay que definir el tiempo aproximado de desarrollo teniendo en cuenta la dificultad.

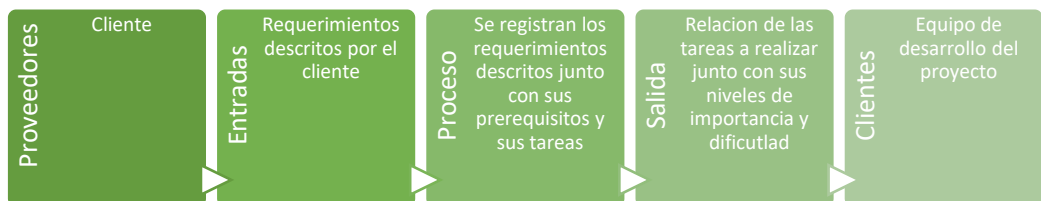


Figura 26, Diagrama SIPOC de "Planeación del Proyecto"

Fuente, Elaboración propia.

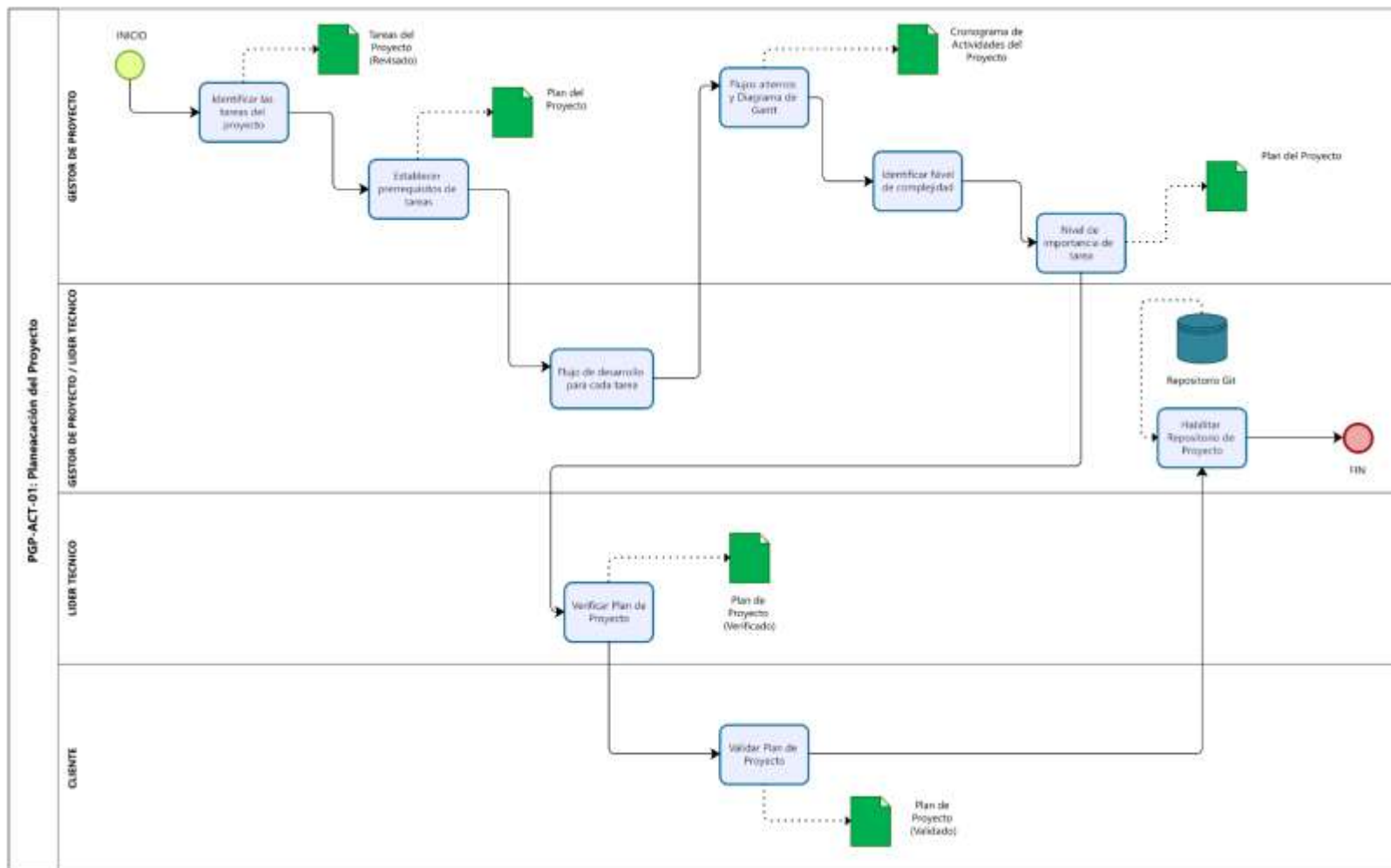


Figura 27, PGP-ACT-01: Planeación del Proyecto

Fuente, Elaboración propia.

## **2. PGP-ACT-02: Despliegue de Plan del Proyecto**

### **OBJETIVO**

El propósito de este procedimiento es asignar los roles que debe desempeñar cada uno de los miembros del equipo, así como las responsabilidades con las cuales debe cumplir.

### **ALCANCE**

Este procedimiento es aplicable para todos los equipos de desarrolladores que pertenecen a la organización.

### **TAREAS**

- a. Se debe registrar quienes serán parte del equipo de trabajo para el desarrollo de las tareas establecidas.
- b. Establecer cuáles son los roles que deberán ser desempeñados durante el desarrollo.
- c. Limitar las responsabilidades para cada uno de los roles los cuales han sido identificados.
- d. Identificar las aptitudes que debe tener la persona encargada de cada uno de los roles.
- e. Evaluar a los miembros del equipo para encontrar a la persona más adecuada para el rol.
- f. Informar de manera general a todo el equipo quienes serán los encargados de cada uno de los roles para el conocimiento general de todos los miembros del equipo.
- g. Aceptación por parte de los miembros del equipo de los roles asignados
- h. Retroalimentación con los miembros del equipo en busca de aclarar dudas con respecto a las responsabilidades de los roles establecidos.

**Tabla 32**

**Formato de relación de registro de empleados**

Relación de registro de empleados			
Nombre Completo	Rol	Responsabilidades	Aptitudes



*Figura 28, Diagrama SIPOC de “Despliegue de Plan del Proyecto”*

*Fuente, Elaboración propia.*

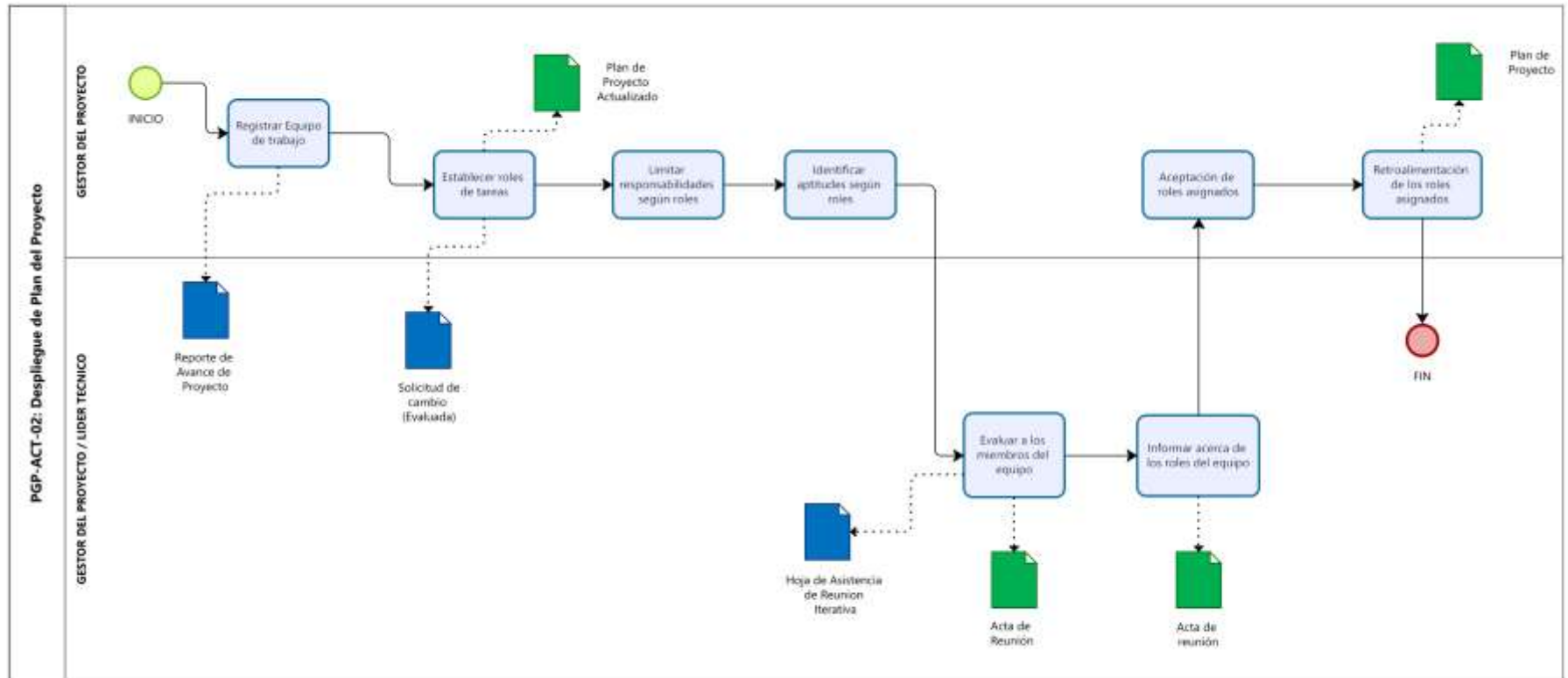


Figura 29, PGP-ACT-02: Despliegue de Plan del Proyecto

Fuente, Elaboración propia.



### **3. PGP-ACT-03: Evaluación y Monitoreo del Proyecto**

#### **OBJETIVO**

El propósito de este procedimiento es realizar una estimación a apropiada para todas las tareas del proyecto, así como una visión general de los costos y tiempos que deberán ser invertidos en la realización del proyecto.

#### **ALCANCE**

Este procedimiento es aplicable para todos los equipos de desarrolladores que pertenecen a la organización.

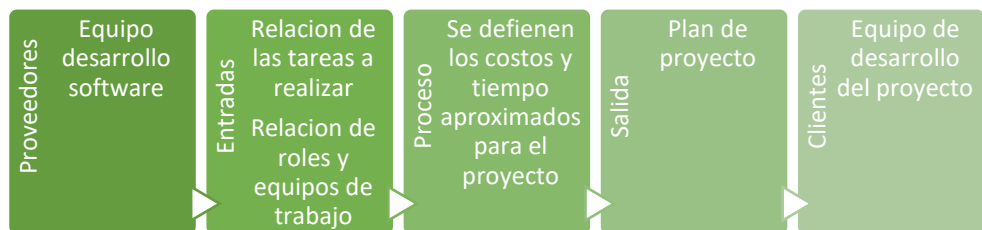
#### **TAREAS**

- a. Recobrar la relación de las tareas establecidas para el proyecto.
- b. Recobrar la relación de los miembros del equipo junto con sus roles y sus responsabilidades.
- c. Análisis del número de participantes por cada una de las tareas a realizar durante el proyecto.
- d. Análisis del costo operativo por cada una de las tareas, así como costos de materiales.
- e. Análisis de tiempo aproximado para la realización para cada una de las tareas teniendo en cuenta su importancia y su complejidad.
- f. Reunión con los miembros del equipo para contrastar la estimación de tiempo con la estimación de los encargados de la elaboración de las tareas.
- g. Reunión entre los encargados de la planificación para deliberar estimaciones.
- h. Contraste entre la estimación de costo total contra el presupuesto del proyecto.
- i. Contraste entre la estimación de tiempo contra el tiempo asignado para la culminación total del proyecto.
- j. Registro de las estimaciones de costo, tiempo y esfuerzo para futura consulta en caso de inconvenientes.

**Tabla 33**

**Formato de estimación de costo, tiempo y esfuerzo por tarea**

Estimación de costo, tiempo y esfuerzo por tarea				
N° Tarea	Encargado	Costo Aprox	Fecha Inicio	Fecha fin
<b>Total (S/)</b>			<b>Días totales</b>	



*Figura 30, Diagrama SIPOC de “Evaluación y monitoreo del Proyecto”*

*Fuente, Elaboración propia.*

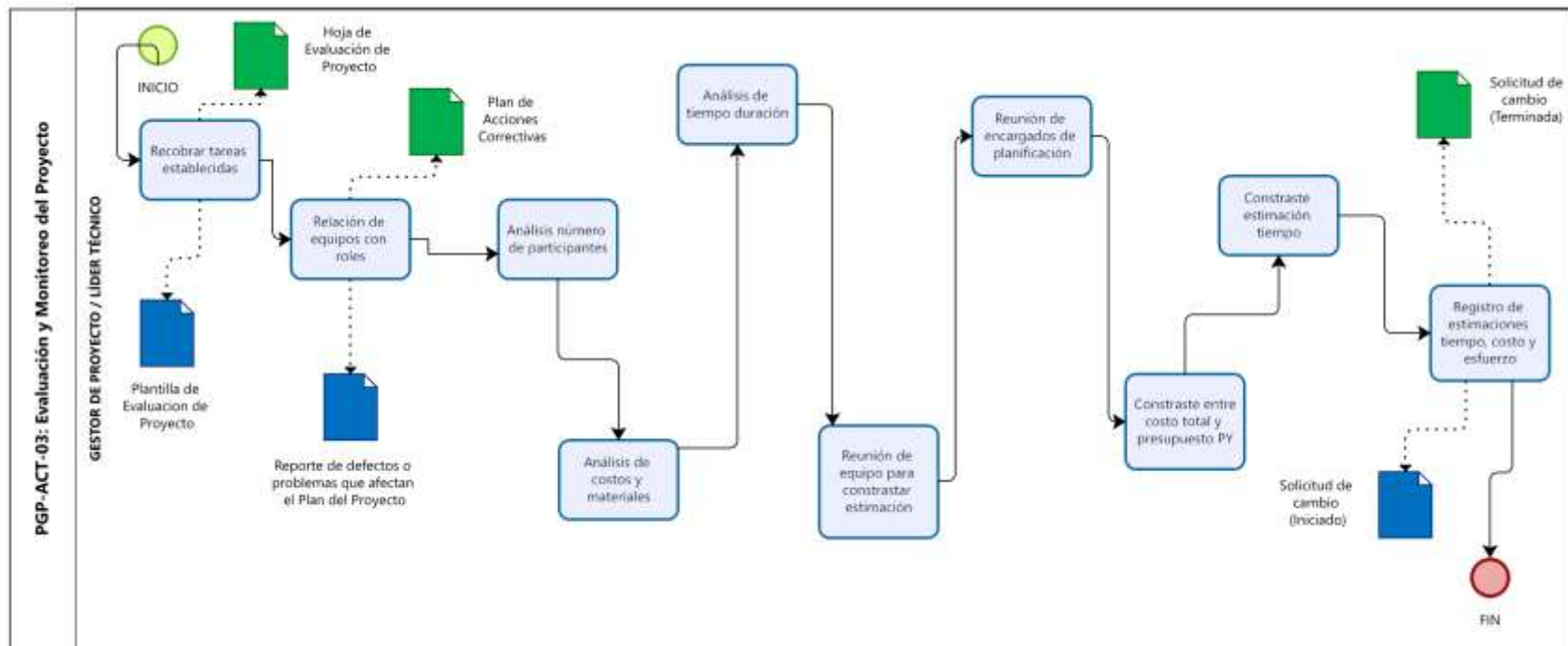


Figura 31, PGP-ACT-03: Evaluación y Monitoreo del Proyecto

Fuente, Elaboración propia.

En cuanto al Proceso de Implementación de Software, se contó con las siguientes cuatro (04) actividades:

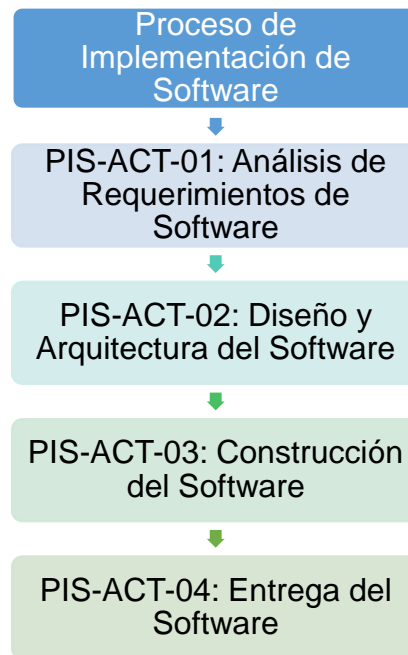


Figura 32, Actividades del Proceso de Implementación de Software

Fuente, Elaboración propia.

### 1. PIS-ACT-01: Análisis de Requerimientos de Software

Durante este procedimiento es el cliente el que con ayuda del formato que se entregará a continuación ayudará al equipo a registrar todos los requerimientos que se tienen que cumplir durante el proyecto.

Tabla 34

**Formato de registro de requerimientos por realizar**

Formato de registro de requerimientos por realizar			
<b>N° Tarea:</b>		<b>Nombre:</b>	
<b>Complejidad</b>		<b>Importancia</b>	
<b>Tiempo Aproximado</b>			
<b>Pre-Requisitos</b>			
<b>Flujo regular</b>		<b>Flujo Alternativo</b>	

## OBJETIVO

El propósito de este procedimiento es establecer cuáles serán las tareas para realizar en el desarrollo del software, asimismo cuáles serán los factores que van a intervenir al momento de realizarla junto con su duración.

## ALCANCE

Este procedimiento es aplicable para todos los equipos de desarrolladores que pertenecen a la organización.

## TAREAS

- a. Primero se deben identificar todas las tareas que se deben realizar a lo largo de todo el proyecto.
- b. Se debe establecer los prerequisites para cada una de las tareas que deben ser realizadas.
- c. Se debe determinar cuál será el flujo de desarrollo para cada una de las tareas identificadas.
- d. Se debe establecer los posibles flujos alternos para cada una de las tareas
- e. Hay que determinar el nivel de complejidad para cada una de las tareas
- f. Acordar el nivel de importancia de la tarea.
- g. Posteriormente hay que definir el tiempo aproximado de desarrollo teniendo en cuenta la dificultad.

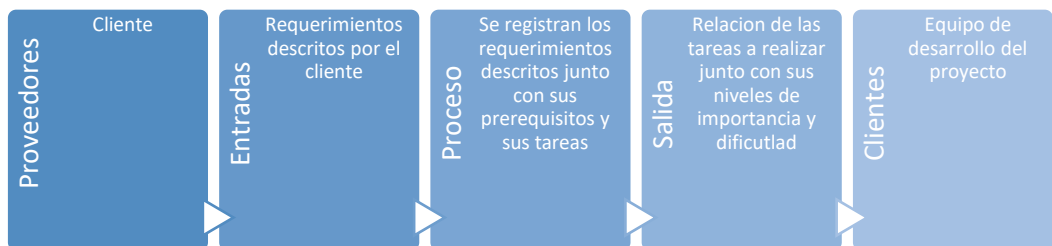


Figura 33, Diagrama SIPOC de “Análisis y requerimientos de software”

Fuente, Elaboración propia.

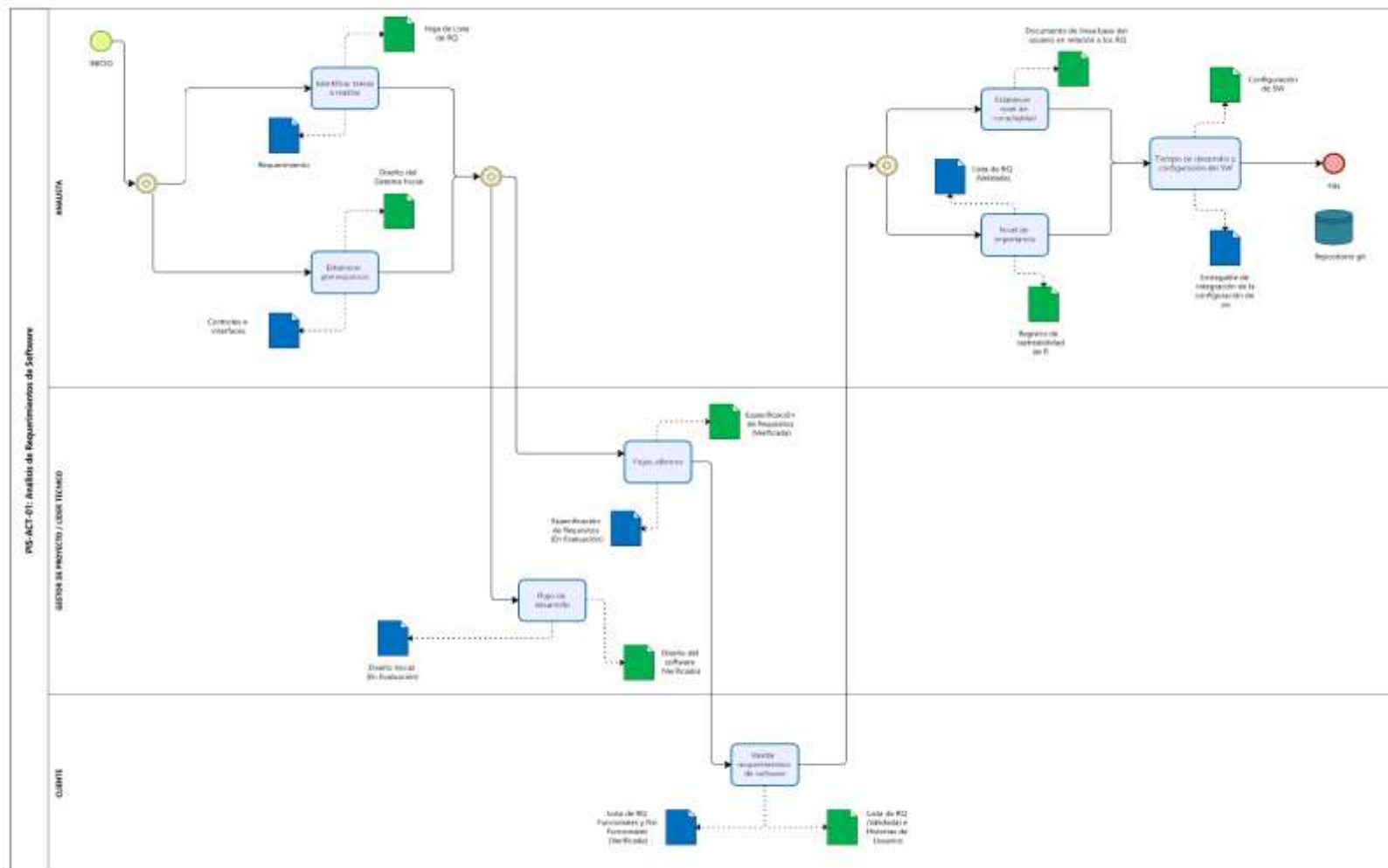


Figura 34, PIS-ACT-01: Análisis de Requerimientos de Software

Fuente, Elaboración propia.

## 2. PIS-ACT-02: Diseño y Arquitectura del Software

### OBJETIVO

El propósito de este procedimiento es poder establecer un entorno de desarrollo apropiado para el correcto desenvolvimiento del equipo de desarrollo del producto.

### ALCANCE

Este procedimiento es aplicable para todos los equipos de desarrollo dentro de la empresa.

### TAREAS

- a. Identificar cuáles son las tecnologías que se utilizarán para el desarrollo del producto.
- b. Realizar la descarga de todos los recursos relacionados a la instalación de las tecnologías que se vayan a utilizar.
- c. Revisar la documentación para la correcta instalación y ubicación dentro de los archivos del proyecto.
- d. Prueba de concepto para verificar el correcto funcionamiento del recurso.
- e. Antes de hacer cambios, siempre revisar la documentación de las tecnologías a utilizar.
- f. Una vez que todo se haya instalado se debe hacer una copia de seguridad del ambiente de desarrollo operativo.
- g. Subir la copia de seguridad al repositorio git que se asignó para el proyecto.

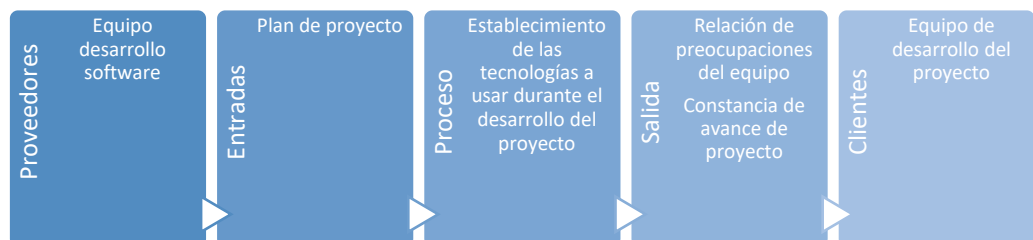


Figura 35, Diagrama SIPOC de “Diseño y arquitectura de software”

Fuente, Elaboración propia.

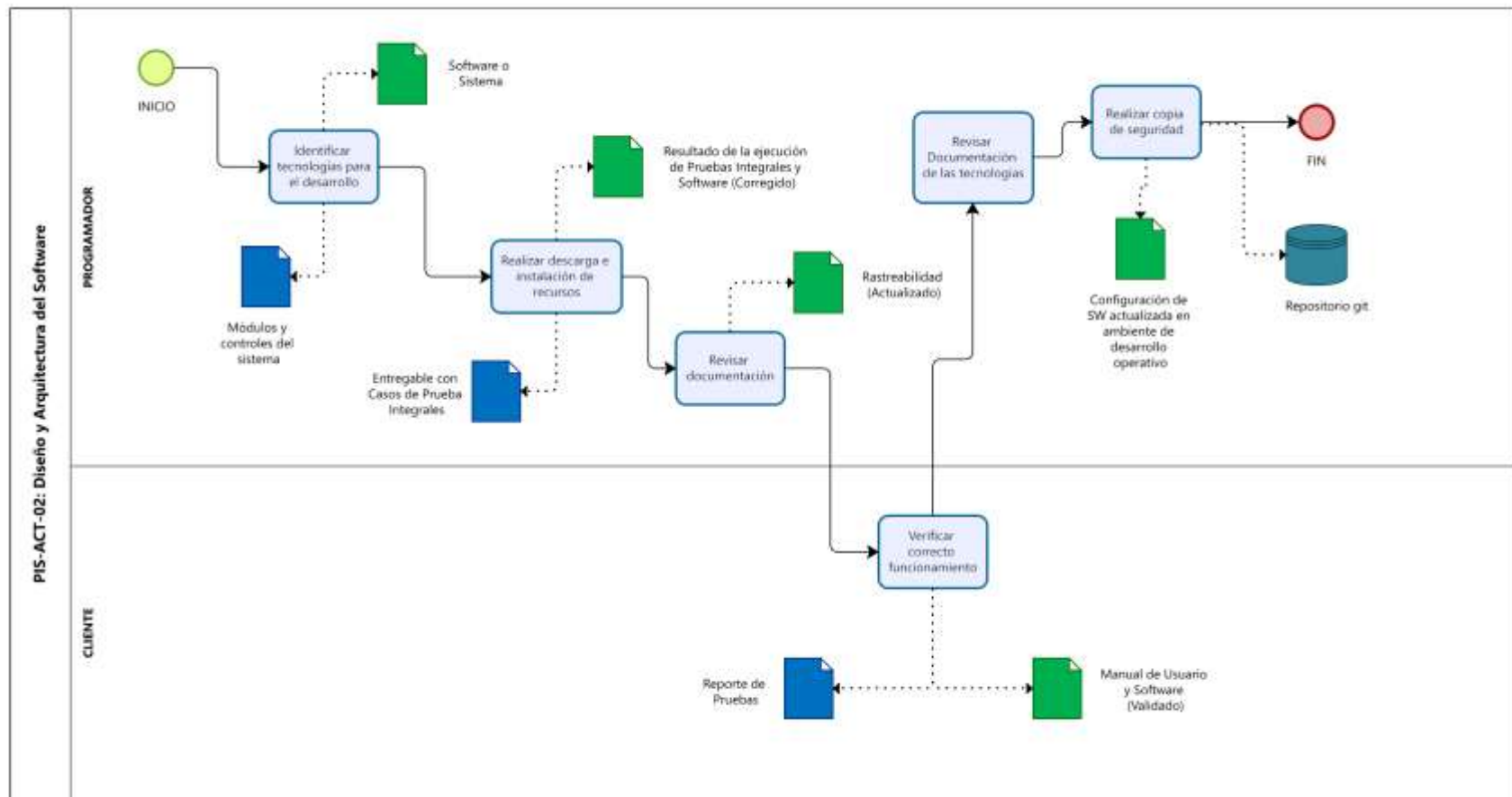


Figura 36, PIS-ACT-02: Diseño y Arquitectura del Software

Fuente, Elaboración propia.



### **3. PIS-ACT-03: Construcción del Software**

#### **OBJETIVO**

El propósito de este procedimiento es mantener una correcta supervisión y monitoreo de las tareas del proyecto, asimismo registrar inconvenientes por partes del equipo de desarrollo.

#### **ALCANCE**

Este procedimiento es aplicable para todos los equipos de desarrolladores que pertenecen a la organización.

#### **TAREAS**

- a. Se debe realizar una reunión con todo el equipo de trabajo para hacer las consultas pertinentes con respecto a las tareas asignadas.
- b. Los miembros del equipo de desarrollo del proyecto deben expresar cual es el estado actual de la tarea que se está desarrollando.
- c. Se debe consultar el documento de estimación de tiempos para ver si el progreso está dentro de los estimado.
- d. En caso de que se hallara un atraso con respecto al progreso del proyecto debe anotarse.
- e. Preguntar al encargado de la tarea que presenta retraso la causa por la cual se ha dado ese atraso.
- f. En caso de ser un factor externo se debe ajustar la estimación de tiempo.
- g. En caso de ser un factor interno, brindar la asistencia necesaria al trabajador para la regulación del tiempo de finalización de la tarea.

- h. Una vez identificado todo el progreso se debe actualizar el documento de tareas del proyecto, asimismo el documento de estimación de tiempos.
- i. Finalmente informar a los miembros del equipo cuando será la próxima reunión para la posterior actualización del progreso del proyecto.

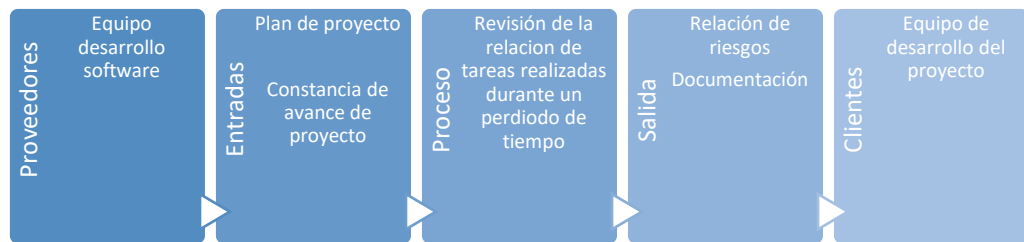


Figura 37, Diagrama SIPOC de "Construcción del software"

Fuente, Elaboración propia.

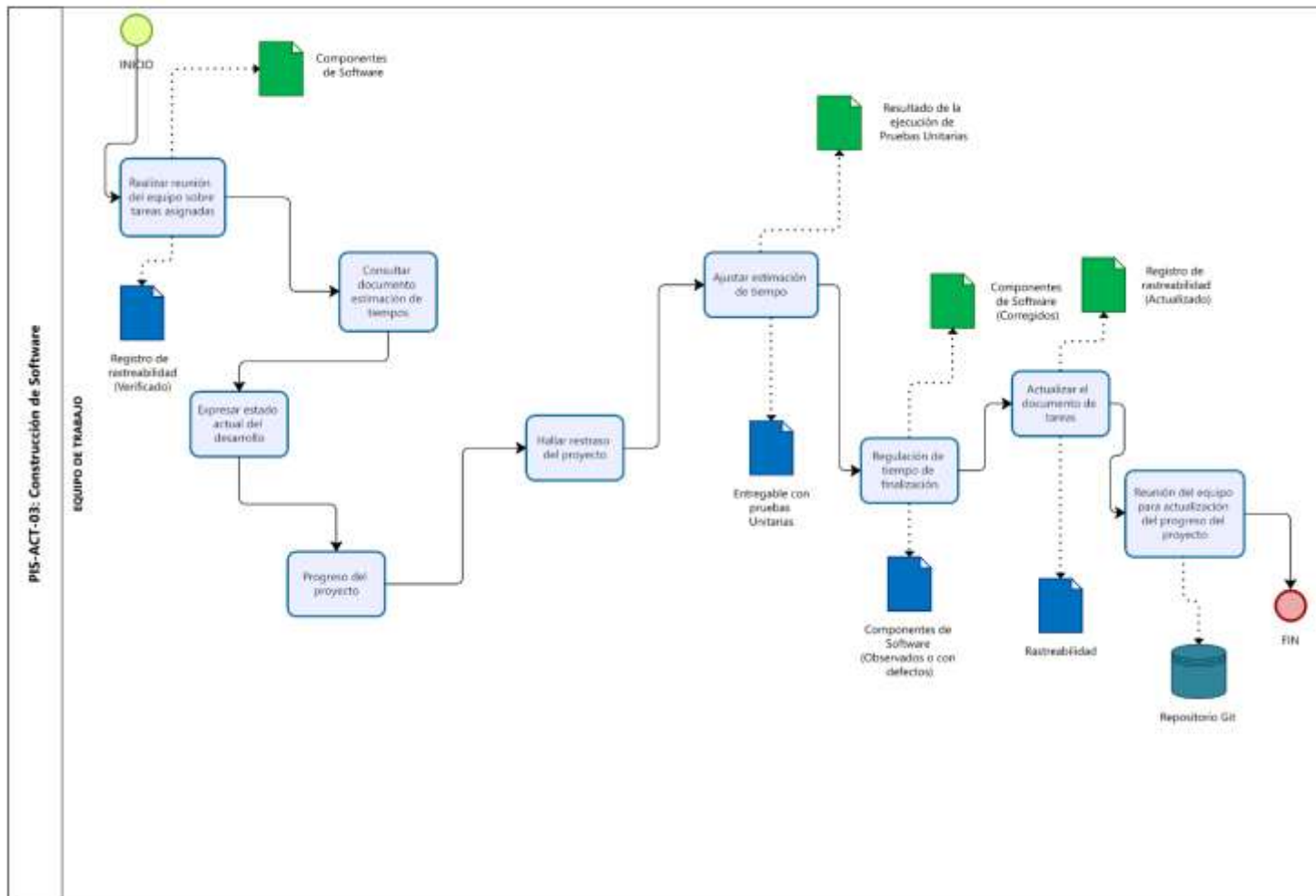


Figura 38, PIS-ACT-03: Construcción del Software

Fuente, Elaboración propia.

#### 4. PIS-ACT-04: Entrega del Software

##### OBJETIVO

El propósito de este procedimiento es realizar una correcta entrega del producto final, así como revisar y aclarar con el cliente cuáles son sus derechos de soporte y actualización.

##### ALCANCE

Este procedimiento es aplicable para los encargados del proyecto o en su defecto los encargados de la entrega del producto final del proyecto.

##### TAREAS

- a. Identificación de la versión estable del producto final que será entregada al cliente.
- b. Creación de la documentación para el correcto uso del producto final.
- c. Revisión del manual para el uso del producto final.
- d. Acordar la reunión con el cliente para poder hacer la entrega formal
- e. Capacitación en el uso del producto final.
- f. Aclaración de las condiciones para el soporte técnica y de futuras actualización en caso de fallo del producto final.
- g. Reconocimiento por parte del cliente para poder certificar su satisfacción con la entrega del producto final.
- h. Entrega final del producto junto con la documentación apropiada y también del acuerdo de soporte técnico.

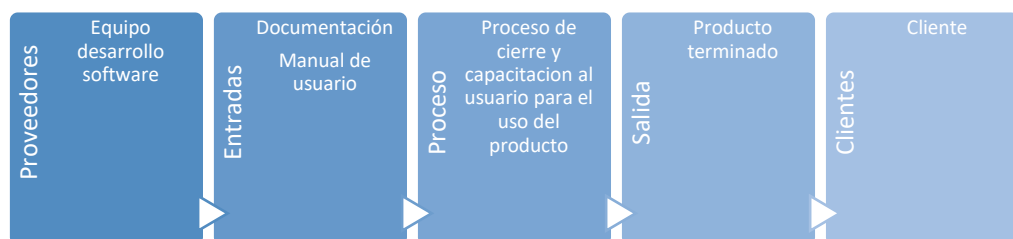


Figura 39, Diagrama SIPOC de “Entrega del software”

Fuente, Elaboración propia.

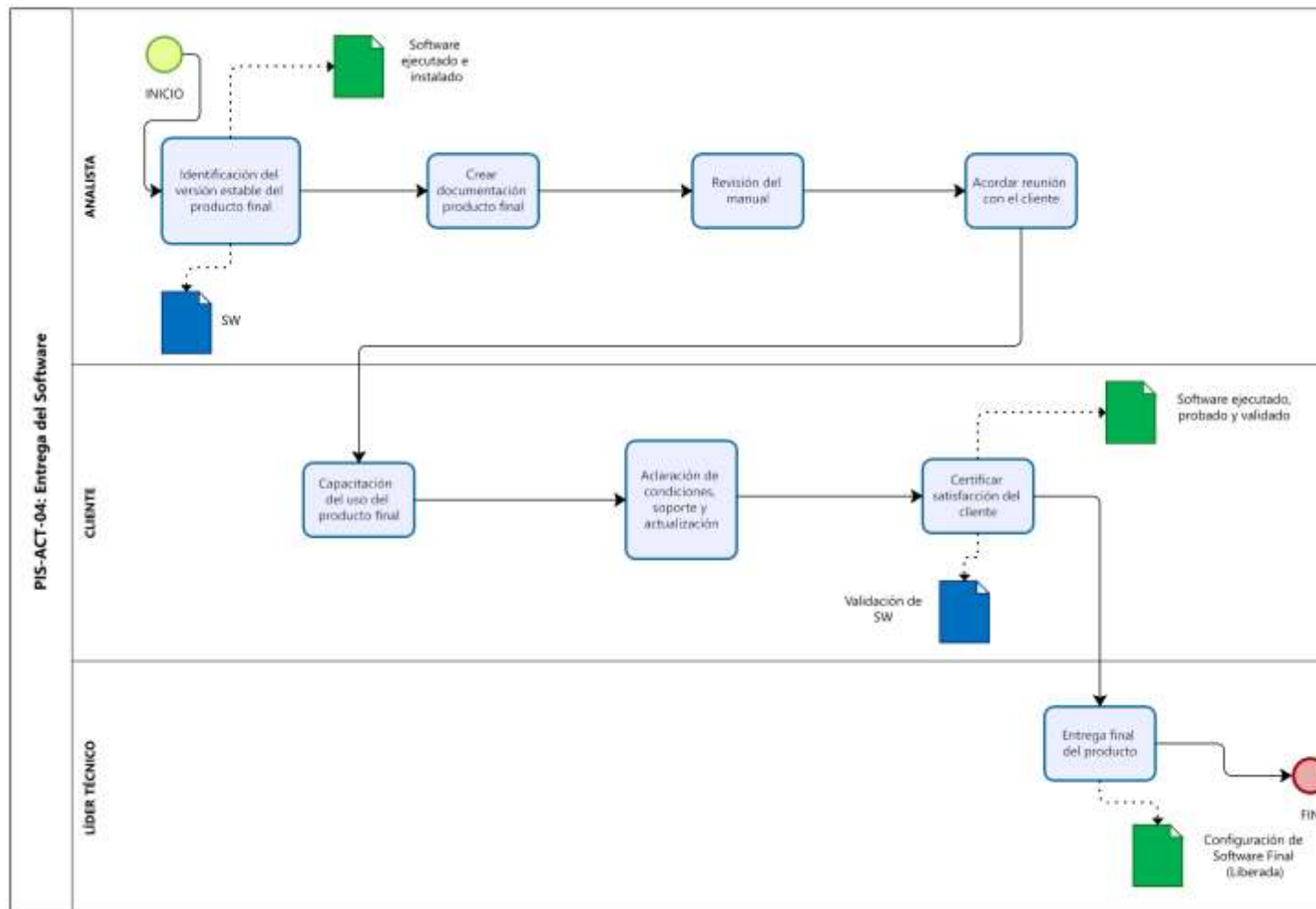


Figura 40, PIS-ACT-04: Entrega del Software

Fuente, Elaboración propia.

### 3.3.4. Validar mediante juicio de expertos la propuesta del modelo de procesos ad hoc desarrollado

Posterior al diseño del modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110, se necesitó validar dicho modelo por expertos. Para ello, primeramente, se debió de diseñar el instrumento de recolección de datos, más específicamente, la ficha de juicio de expertos, la cual se encuentra en el Anexo 4 y que consideró diez (10) indicadores tal y como puede visualizarse a continuación:

ASPECTOS DE VALIDACIÓN		Deficiente										Baja					Regular					Buena					Muy Buena				
Indicadores	Criterios	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100										
		CLARIDAD	Está formulado con lenguaje apropiado																												
OBJETIVIDAD	Está expresado en conductas observables																														
ACTUALIDAD	Adecuado al avance del desarrollo de software																														
ORGANIZACIÓN	Existe una organización lógica																														
SUFICIENCIA	Comprende los aspectos en cantidad y calidad																														
INTENCIONALIDAD	Adecuado para valorar el desarrollo de software																														
CONSISTENCIA	Basado en aspectos teóricos científicos del desarrollo de software																														
COHERENCIA	Entre cada uno de los pasos del modelo																														
METODOLOGÍA	La estrategia responde al propósito de la investigación																														
PERTINENCIA	Es útil y adecuado para la investigación																														

VALORACIÓN: \_\_\_\_\_

OPINIÓN DE APLICABILIDAD: \_\_\_\_\_

Lugar y fecha: Chiclayo, \_\_\_\_ Junio del 2022.

Figura 41, Indicadores considerados en la Ficha de Juicio de Expertos

Fuente, Elaboración propia.

Luego de tener diseñado dicho instrumento, se procedió a solicitar la apreciación crítica de tres (03) expertos para que validen el modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110, el mismo que fue desarrollado por los investigadores. Para verificar la validez y fiabilidad del modelo propuesto se validó que los expertos tengan como mínimo con experiencia en desarrollo de software, modelos de procesos de desarrollo de software, quienes deberán disponer con experiencia específica proba en ingeniería de desarrollo de software como se detalla a continuación:

- a. Poseer Grado Académico en Ingeniería de Sistemas o Ingeniería de Software o Ingeniería Informática.
- b. Poseer postgrado acorde con la Ingeniería de Sistemas.
- c. Poseer experiencia proba en procesos de desarrollo de software en la gestión pública o privada.
- d. Poseer experiencia profesional en labores asociadas a la Ingeniería de Sistemas.

Luego de determinar los criterios, se pudo establecer contacto con tres (03) expertos que cumplían con las características necesarias para evaluar dicho modelo. Los expertos en mención se muestran en la siguiente tabla:

**Tabla 35**

***Expertos para validación de propuesta***

<b>N°</b>	<b>Experto</b>	<b>Grado Académico</b>	<b>Experiencia</b>
1	Ruiz Gómez Woolder	Doctor (c) en Ingeniería de Sistemas e Informática	Ingeniería de Sistemas, experiencia en Proyectos de desarrollo de software e Infraestructura de TI en el sector público y privado. Anexo 8.
2	Tapia Flores Jesús Benjamín	Maestro en Ingeniería de Sistemas	Ingeniería de Sistemas, experiencia en gestión de aplicaciones y gestión de procesos SCRUM. Anexo 8.
3	Reyna Robles Jaime Luis	Maestro (c) en Ingeniería de Sistemas y Computación	Ingeniería de desarrollo de software y experto en modelo de procesos de la ISO/IEC 29110. Anexo 8.

*Fuente, elaboración propia.*

Posterior a la selección de los expertos, se procedió a la comunicación con cada uno de ellos mediante correo electrónico puesto que, se les envió sendos correos solicitando la validación, en los cuales se adjuntó el formato de Ficha de Juicio de Expertos que se muestra en el Anexo 4.

Luego de haber llevado a cabo la solicitud de validación, se procedió a la recepción de dichas “Ficha de Juicio de Expertos” con los resultados de cada uno de los expertos. Cada una de estas fichas se encuentran en el Anexo 8. A continuación, se muestran las valoraciones de cada uno de los expertos en cuanto al modelo propuesto:

**Tabla 36**

***Expertos para validación del modelo propuesto***

<b>N°</b>	<b>Experto</b>	<b>Promedio por experto</b>	<b>Nivel de Aceptación del Modelo</b>
1	Experto 01	95.00	
2	Experto 02	95.00	92%
3	Experto 03	86.00	

*Fuente, elaboración propia.*

Tal y como se pudo visualizar en la tabla anterior, el modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110 contó con un nivel de aceptación de 92 en una escala del 1 al 100, por lo que se consideró que tiene una valoración considerada como “Muy Buena”.



### **3.3.5. Realizar una prueba piloto del modelo de procesos ad hoc para el desarrollo de software por licencia en una MYPE de servicios de TI**

Para la realización de la prueba piloto del modelo propuesto, se requirió de una compañía caso de estudio, la misma que permitiera realizar las evaluaciones al desarrollo de software por licencia.

En este caso, se contó con el permiso de la empresa NEDLEY SUPPORT S.A.C., compañía trujillana que cuenta con número de RUC 20601609232 y que se encuentra establecida en la Calle Costa Rica N° 231 de la Urbanización Torres Araujo, en el distrito de Trujillo, provincia de Trujillo, departamento de La Libertad. El permiso respectivo se encuentra en el Anexo 2.



*Figura 42, Logo de NEDLEY SUPPORT S.A.C.*

*Fuente, Elaboración propia.*

La compañía en mención cuenta con una página web empresarial: <https://www.nedleysupport.com/> en la cual menciona “Ser una empresa de outsourcing tecnológico e Informático. Nos especializamos en servicios de Inteligencia de Negocios. Estamos capacitados en una amplia gama de software de gestor de Base de Datos y BI como Oracle, SQL Server, My SQL, Visual Studio, Power Builder, entre otros. Además, realizamos administración de sistemas ERP, Redes, servidores, firewall-proxy, gestión de proyectos, gestión de procesos, virtualización y soporte técnico e informático”.



Figura 43, Página Web Empresarial NEDLEY SUPPORT S.A.C.

Fuente, Elaboración propia.

La compañía centra su giro del negocio y sus servicios en 3 valores indispensables:

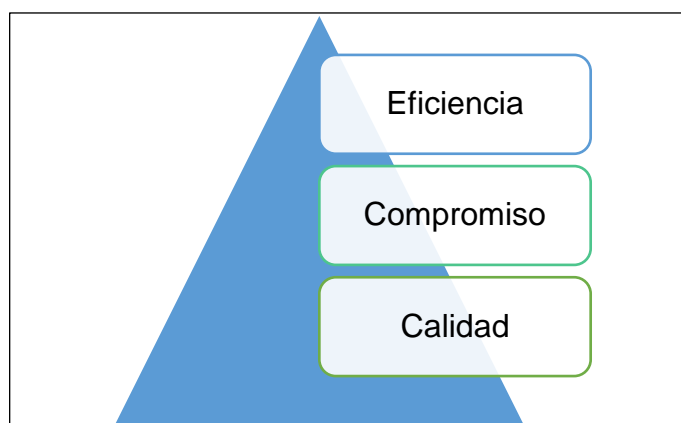


Figura 44, Valores empresariales de NEDLEY SUPPORT S.A.C.

Fuente, NEDLEY SUPPORT S.A.C.

La prueba piloto fue llevada a cabo evaluando seis (06) proyectos de desarrollo de software por licencia a los cuales se les hicieron las mediciones pertinentes en cuanto a los indicadores especificados en la Tabla 2. Dichos proyectos de desarrollo de software por licencia fueron desarrollados por NEDLEY SUPPORT S.A.C. y corresponden a proyectos construidos en meses distribuidos de la siguiente manera:

**Tabla 37**

***Cronograma de construcción de proyectos***

	06/21	07/21	08/21	09/21	10/21	11/21
PRE TEST	Pre-Test-01	Pre-Test-02	Pre-Test-03			
POST TEST				Post-Test-01	Post-Test-02	Post-Test-03

*Fuente, elaboración propia.*

Para el caso de los tres (03) proyectos PRE TEST, se desarrollaron los siguientes proyectos:

1. Proyecto 01: Sistema de Tempus Web - Control Tiempo (PRE-TEST)
2. Proyecto 02: Sistema de Ventas (PRE-TEST)
3. Proyecto 03: Sistema de Referencias (PRE-TEST)

En este caso específico, se realizaron las mediciones en cuanto a los indicadores especificados en la Tabla 2, pero sin considerar y sin seguir al pie de la letra, las tareas, las actividades y los procesos del modelo de procesos ad hoc basado en ISO/IEC 29110 propuesto para el desarrollo de software por licencia para NEDLEY SUPPORT S.A.C.

Dichos proyectos PRE TEST se encuentran especificados en el Anexo 10.

Por otro lado, para el caso de los tres (03) proyectos POST TEST, se desarrollaron los siguientes proyectos:

1. Proyecto 04: Sistema de Mesa de Ayuda - Soporte Técnico (POST-TEST)
2. Proyecto 05: Sistema Web de Recursos Humanos (POST-TEST)
3. Proyecto 06: Sistema de Mesa de partes digital (POST-TEST)

Dichos proyectos POST TEST se encuentran especificados en el Anexo 11.

En este caso específico, se realizaron las mediciones en cuanto a los indicadores especificados en la Tabla 2, pero ahora sí considerando y siguiendo al pie de la letra, las tareas, las actividades y los procesos del modelo de procesos ad hoc basado en ISO/IEC 29110 propuesto para el desarrollo de software por licencia para NEDLEY SUPPORT S.A.C.

Estos proyectos se gestionaron de manera independiente y fueron desarrollados en los meses de septiembre, octubre y noviembre del año 2021, tal y como fue especificado en la Tabla 37.

A continuación, se muestra el diagrama de Gantt del Proyecto 04: Sistema de Mesa de Ayuda - Soporte Técnico (POST-TEST):

Proyecto 04: Sistema de Mesa de Ayuda - Soporte Técnico																					
TÍTULO DEL PROYECTO		PROYECTO 04: SISTEMA DE MESA DE AYUDA - SOPORTE TÉCNICO																			
RESPONSABLE DEL PROYECTO		APARICIO CARRASCO BORDA																			
NÚMERO EDT	TÍTULO DE LA TAREA	RESPONSABLE DE LA TAREA	FECHA DE INICIO	FECHA DE ENTREGA	DURACIÓN	% COMPLETADO DE LA TAREA	Cronograma de desarrollo de las 7 actividades del modelo propuesto														
							1	2	3	4	5	6	7	8	9	10	11	12	13		
<b>1 PGP-ACT-01: Planeación del Proyecto</b>																					
1.1	Identificar las tareas del proyecto	Aparicio Carrasco Borda	01/09/21	08/09/21	7	100 %	x														
1.2	Establecer prerrequisitos y requisitos del proyecto	Aparicio Carrasco Borda	01/09/21	08/09/21	7	100 %	x														
1.3	Definir el tiempo aproximado de desarrollo del proyecto	Aparicio Carrasco Borda	01/09/21	08/09/21	7	100 %	x														
<b>2 PGP-ACT-02: Despliegue de Plan del Proyecto</b>																					
2.1	Registrar el equipo de trabajo y roles	Aparicio Carrasco Borda	09/09/21	16/09/21	7	100 %		x													
2.2	Evaluar los miembros del equipo	Aparicio Carrasco Borda	09/09/21	16/09/21	7	100 %		x													
2.3	Aceptación de los roles asignados	Aparicio Carrasco Borda	09/09/21	16/09/21	7	100 %		x													
<b>3 PGP-ACT-03: Evaluación y Monitoreo del Proyecto</b>																					
3.1	Recobrar la relación de las tareas establecidas para el proyecto	Aparicio Carrasco Borda	17/09/21	24/09/21	7	100 %			x												
3.2	Reunión con los miembros del equipo para contrastar la estimación de tiempo	Aparicio Carrasco Borda	17/09/21	24/09/21	7	100 %			x												
3.3	Registro de las estimaciones de costo, tiempo y esfuerzo para futura consulta en caso de inconvenientes	Aparicio Carrasco Borda	17/09/21	24/09/21	7	100 %			x												
<b>4 PIS-ACT-01: Análisis de Requerimientos de Software</b>																					
4.1	Determinar el flujo desarrollo del software	Aparicio Carrasco Borda	25/09/21	02/10/21	7	100 %				x											
4.2	Abrar el nivel de importancia de la tarea	Aparicio Carrasco Borda	25/09/21	02/10/21	7	100 %				x											
4.3	Definir el tiempo de desarrollo teniendo en cuenta la dificultad	Aparicio Carrasco Borda	09/10/21	10/10/21	7	100 %					x										
<b>5 PIS-ACT-02: Diseño y Arquitectura del Software</b>																					
5.1	Identificar cuáles son las tecnologías que se utilizarán para el desarrollo del producto	Aparicio Carrasco Borda	11/10/21	18/10/21	7	100 %						x									
5.2	Revisar la documentación para la correcta instalación y utilización	Aparicio Carrasco Borda	19/10/21	26/10/21	7	100 %							x								
5.3	Subir la copia de seguridad al repositorio git que se asignó para el proyecto	Aparicio Carrasco Borda	27/10/21	04/11/21	7	100 %									x						
<b>6 PIS-ACT-03: Construcción del Software</b>																					
6.1	Realizar reuniones con todo el equipo de trabajo	Aparicio Carrasco Borda	05/10/21	06/12/21	81	100 %							x	x	x	x	x	x	x	x	x
6.2	Consultar el documento de estimación de tiempos para ver si el progreso está dentro de los estimados	Aparicio Carrasco Borda	27/10/21	06/12/21	39	100 %													x	x	x
6.3	Actualizar el documento de tareas del proyecto	Aparicio Carrasco Borda	05/10/21	06/12/21	81	100 %													x	x	x
<b>7 PIS-ACT-04: Entrega del Software</b>																					
7.1	Identificación de la versión estable del producto final	Aparicio Carrasco Borda	07/12/21	10/12/21	3	100 %															
7.2	Revisión del manual para el uso del producto final	Aparicio Carrasco Borda	07/12/21	10/12/21	3	100 %															
7.3	Entrega del Proyecto	Aparicio Carrasco Borda	10/12/21	12/12/21	2	100 %															

Figura 45, Diagrama de Gantt del Proyecto 04: Sistema de Mesa de Ayuda - Soporte Técnico.

Fuente, Elaboración propia.

A continuación, se muestra el diagrama de Gantt del Proyecto 05: Sistema Web de Recursos Humanos (POST-TEST):

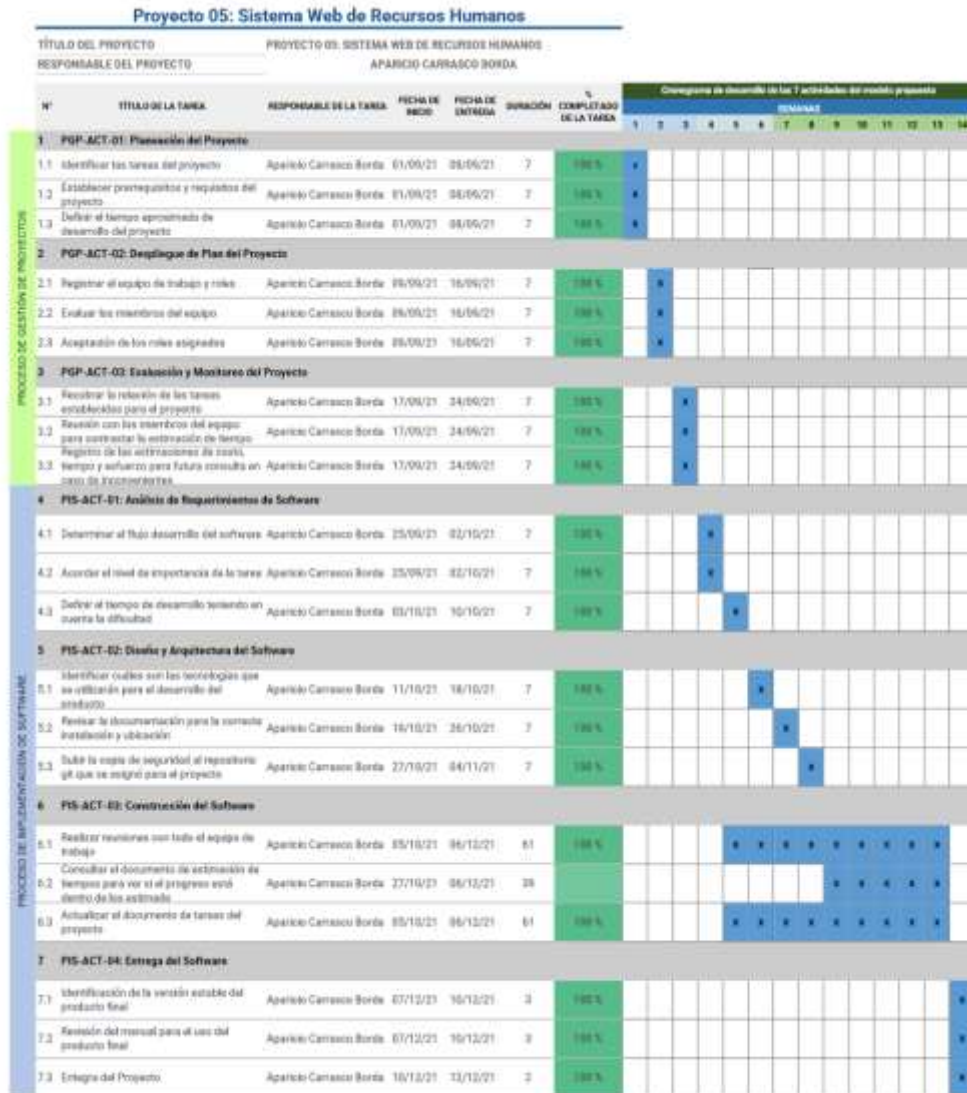


Figura 46, Diagrama de Gantt del Proyecto 05: Sistema Web de Recursos Humanos (POST-TEST).

Fuente, Elaboración propia.

A continuación, se muestra el diagrama de Gantt del Proyecto 06: Sistema de Mesa de partes digital (POST-TEST)



Figura 47, Diagrama de Gantt del Proyecto 06: Sistema de Mesa de partes digital (POST-TEST)

Fuente, Elaboración propia.

Se ha realizado la comparación de la prueba piloto de los proyectos desarrollados por NEDLEY SUPPORT S.A.C, tanto por ser un elemento diferenciador para sus clientes, como por la importancia interna para su cultura empresarial.

Tabla 38

**Cuadro comparativo de los proyectos de prueba piloto del modelo propuesto basado en la norma ISO/IEC 29110.**

PROYECTOS COMPARADOS	PRE-TEST (anexo10)			POST-TEST (anexo11)		
	Proyecto 01	Proyecto 02	Proyecto 03	Proyecto 04	Proyecto 05	Proyecto 06
<b>Número de líneas de código</b>	348.242 líneas de código	352.123 líneas de código	341.985 líneas de código	634.724 líneas de código	664.248 líneas de código	650.621 líneas de código
<b>Número de módulos</b>	(05 módulos) <ul style="list-style-type: none"> <li>📖 Usuario</li> <li>📖 Visor de marcación</li> <li>📖 Mant. Terminal</li> <li>📖 Monitor</li> <li>📖 Personal</li> </ul>	(05 módulos) <ul style="list-style-type: none"> <li>📖 Usuario</li> <li>📖 Ventas</li> <li>📖 Cobranzas</li> <li>📖 Almacén</li> <li>📖 Pedidos</li> </ul>	(05 módulos) <ul style="list-style-type: none"> <li>📖 Usuario</li> <li>📖 Referencias</li> <li>📖 Pasajes</li> <li>📖 Reportes</li> <li>📖 Archivo</li> </ul>	(06 módulos) <ul style="list-style-type: none"> <li>📖 Usuario</li> <li>📖 Asistencia</li> <li>📖 Herramientas</li> <li>📖 Plugins</li> <li>📖 Administración</li> <li>📖 Configuración</li> </ul>	((06 módulos) <ul style="list-style-type: none"> <li>📖 Usuario</li> <li>📖 Planilla</li> <li>📖 Contratos</li> <li>📖 Trabajadores</li> <li>📖 Solicitud</li> <li>📖 Seguro</li> </ul>	(06 módulos) <ul style="list-style-type: none"> <li>📖 Usuario</li> <li>📖 Reporte</li> <li>📖 Seguimiento</li> <li>📖 Bandeja</li> <li>📖 Nuevo</li> <li>📖 Courier</li> </ul>
<b>Tamaño del software</b>	Proyecto Mediano	Proyecto Mediano	Proyecto Mediano	Proyecto Mediano	Proyecto Mediano	Proyecto Mediano
<b>Rubro de cliente</b>	Brinda servicios de prevención, promoción, recuperación, rehabilitación, prestaciones	Presta servicios de fabricación y venta de muebles en general. venta de melanina al	Brinda servicios de prevención, promoción, recuperación, rehabilitación, prestaciones económicas y	Brinda una atención integral con calidad y eficiencia para mejorar el bienestar de los asegurados de la región,	Brinda servicios de prevención, promoción, recuperación, rehabilitación, prestaciones económicas y	Atención al Usuario en la recepción de los documentos, registro, foliación, clasificar y distribuir los documentos para

	<i>económicas y prestaciones sociales</i>	<i>por mayor y menor.</i>	<i>prestaciones sociales</i>	<i>personal asistencial y administrativo.</i>	<i>prestaciones sociales</i>	<i>respectiva atención.</i>
	(04)	(04)	(04)	(06)	(06)	(06)
	<i>participantes:</i>	<i>participantes:</i>	<i>participantes:</i>	<i>participantes:</i>	<i>participantes:</i>	<i>participantes:</i>
<b>Equipo de programación</b>	✓ Analista ✓ Diseñador ✓ Programador ✓ Líder Técnico	✓ Analista ✓ Diseñador ✓ Programador ✓ Líder Técnico	✓ Analista ✓ Diseñador ✓ Programador ✓ Líder Técnico	✓ Cliente ✓ Analista ✓ Diseñador ✓ Programador ✓ Gestor Proyecto ✓ Líder técnico	✓ Cliente ✓ Analista ✓ Diseñador ✓ Programador ✓ Gestor Proyecto ✓ Líder técnico	✓ Cliente ✓ Analista ✓ Diseñador ✓ Programador ✓ Gestor Proyecto ✓ Líder técnico
<b>Lenguaje de programación</b>	PHP	PHP	PHP	PHP	PHP	PHP
<b>Base de Datos</b>	MySQL	MySQL	MySQL	MySQL	MySQL	MySQL
<b>Calidad funcional</b>	<i>Si cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>
<b>Calidad estructural</b>	<i>No cumplía</i>	<i>No cumplía</i>	<i>No cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>
<b>Funcionalidad</b>	<i>Si cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>
<b>Confiabilidad</b>	<i>No cumplía</i>	<i>No cumplía</i>	<i>No cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>
<b>Usabilidad</b>	<i>No cumplía</i>	<i>No cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>
<b>Eficiencia</b>	<i>No cumplía</i>	<i>No cumplía</i>	<i>No cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>
<b>Capacidad de mantenimiento</b>	<i>No cumplía</i>	<i>No cumplía</i>	<i>No cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>
<b>Portabilidad</b>	<i>No cumplía</i>	<i>No cumplía</i>	<i>No cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>	<i>Si cumplía</i>

Fuente, elaboración propia.



Comparado los 06 proyectos de prueba piloto entre ellos 03 proyectos pre-test y 03 proyectos post-test, como se aprecia en la Tabla N° 38, se ha cotejado los proyectos en función del tamaño del proyecto. Para ello, en primer lugar, se tomaron en cuenta, según la clasificación de los proyectos de desarrollo software en tamaño y duración, atendiendo a las horas de ejecución de los mismos: (a) Proyectos pequeños: de 1 a 300 horas, (b) Proyectos medianos: de 301 a 5000 horas y (c) Proyectos grandes: más de 5000 horas. Se optó para el caso de la empresa NEDLEY SUPPORT S.A.C, que los proyectos de desarrollo de software de la prueba piloto se realizaron con proyectos medianos que oscilan de 301 a 5000 horas trabajadas.

Para medir número de líneas de código de los proyectos desarrollados de la prueba piloto, la empresa NEDLEY SUPPORT S.A.C, utilizó la herramienta CLOC, para contar las líneas en blanco, las líneas de comentarios y las líneas de código reales escritas. El número de módulo de los proyectos post-test aumentó en cuanto a los proyectos de prueba piloto pre-test, además de ello el número de líneas de código es superior en post-test. El número de participantes del equipo de desarrollo de software por licencia estuvo integrado por: Cliente, Analista, Diseñador, Programador, Gestor de Proyecto y Líder técnico en los proyectos post-test, sin embargo, en los proyectos de la prueba piloto pre-test, se encontró que no lo tomaban en cuenta al cliente.

Para que la empresa alcance dogmatizar la calidad de sus procesos fue imperioso llevar a cabo un proceso de evaluación y mejora de los mismos, que permitió alinear el modelo de desarrollo de software a las mejores prácticas de la Ingeniería del Software. Se evidenció que, con el modelo propuesto basado en la ISO/IEC 29110, se ha mejorado los procesos de desarrollo de software por licencia del caso de estudio. Para el caso de pre-test, los proyectos se han desarrollado sin aplicar ni seguir el modelo propuesto, pero en el post-test si se aplicaron a pie de la letra los procesos propuestos. Según la tabla 38 podemos apreciar que si se ha mejorado en cuanto a calidad.

La prueba piloto realizado con los procesos del modelo propuesto basado en la norma ISO/IEC 29110, en el contexto de la prueba piloto de software desarrollados por la empresa NEDLEY SUPPORT S.A.C, se aplicaron a productos de software en base al modelo propuesto para validar el proceso de mejora de los procesos constructivos de software en dicha empresa. Para ello se realizaron las pruebas de funcionamiento, tiempo de duración de ejecución, diseño, construcción y los defectos o errores que presentaron las aplicaciones diseñadas. En este caso la prueba piloto se aplica solo a mejoras del proceso del proceso constructivo del software, a continuación, se presenta las tablas y gráficas de la prueba piloto.

**Tabla 39**

***Esfuerzo involucrado en cada actividad de pruebas ejecutados de los procesos de la empresa.***

Esfuerzo (horas x 1 persona)														
Actividades del procedimiento de campo (Diagnóstico, reparación, Formulación del proceso, Asignación del proyecto y Contextualización)										<b>D</b>	<b>H</b>	<b>T</b>		
			<b>30</b>				<b>8</b>				<b>240</b>			
	<b>Actividades</b>	<b>D</b>	<b>H</b>	<b>T</b>		<b>Actividades</b>	<b>D</b>	<b>H</b>	<b>T</b>		<b>Actividades</b>	<b>D</b>	<b>H</b>	<b>T</b>
<b>Proyecto 04</b>	Planeación del proyecto	7	8	56	<b>Proyecto 05</b>	Planeación del proyecto	7	8	56	<b>Proyecto 06</b>	Planeación del proyecto	7	8	56
	Despliegue de Plan del proyecto	7	8	56		Despliegue de Plan del proyecto	7	8	56		Despliegue de Plan del proyecto	7	8	56
	Evaluación y Monitoreo del Proyecto	7	8	56		Evaluación y Monitoreo del Proyecto	7	8	56		Evaluación y Monitoreo del Proyecto	7	8	56
	Análisis de Requisitos de Software	14	8	112		Análisis de Requisitos de Software	14	8	112		Análisis de Requisitos de Software	14	8	112
	Diseño y Arquitectura del Software	21	8	168		Diseño y Arquitectura del Software	21	8	168		Diseño y Arquitectura del Software	21	8	168

Construcción del Software	35	8	280	Construcción del Software	35	8	280	Construcción del Software	35	8	280
Entrega del Software	7	8	56	Entrega del Software	7	8	56	Entrega del Software	7	8	56
<b>TOTAL</b>			<b>784</b>	<b>TOTAL</b>			<b>784</b>	<b>TOTAL</b>			<b>784</b>
Duración de pruebas (semanas)			<b>14</b>	Duración de pruebas (semanas)			<b>14</b>	Duración de pruebas (semanas)			<b>14</b>

D: Días H: Horas T: Total Horas

Fuente, elaboración propia.

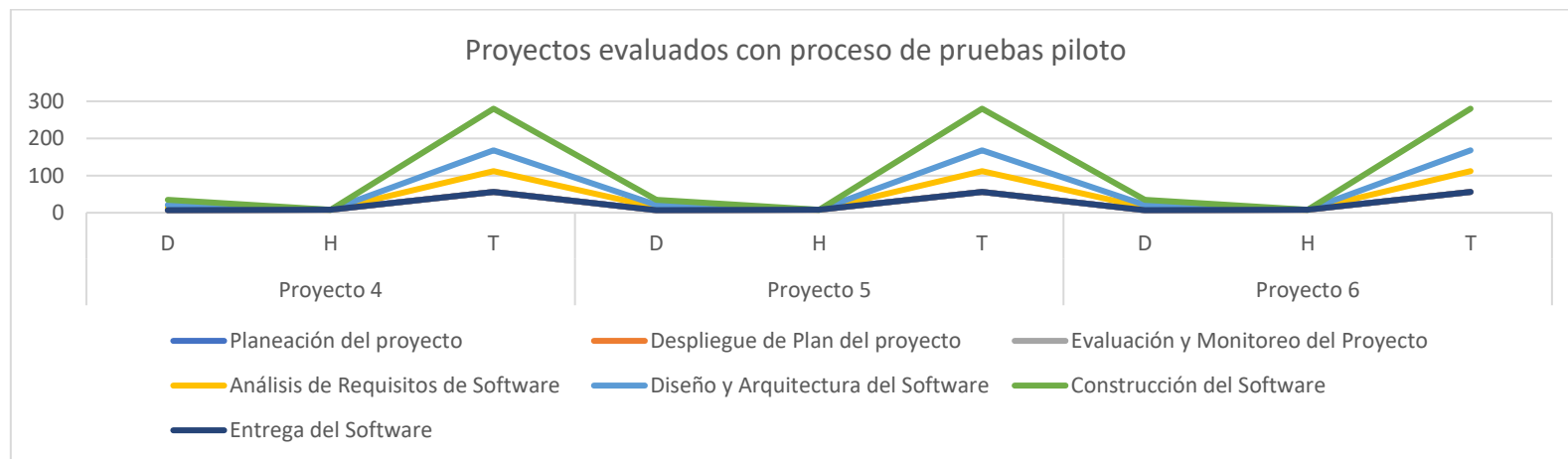


Figura 48, Gráfico de esfuerzo por ciclo de pruebas piloto ejecutado en la empresa

Fuente, elaboración propia.

**Tabla 40**

**Porcentaje de defectos encontrados tras la ejecución de cada proceso de pruebas**

<b>Proyectos</b>	<b>Ciclo / Pruebas</b>	<b>Defectos</b>	<b>Porcentaje</b>
<i>Sin aplicar el modelo propuesto</i>			
Proyecto 01	10	7	30.43%
Proyecto 02	10	8	34.78%
Proyecto 03	10	8	34.78%
<i>Aplicando el modelo propuesto</i>			
Proyecto 04	10	0	0.00%
Proyecto 05	10	1	4.34%
Proyecto 06	10	1	4.34%

Fuente, elaboración propia.

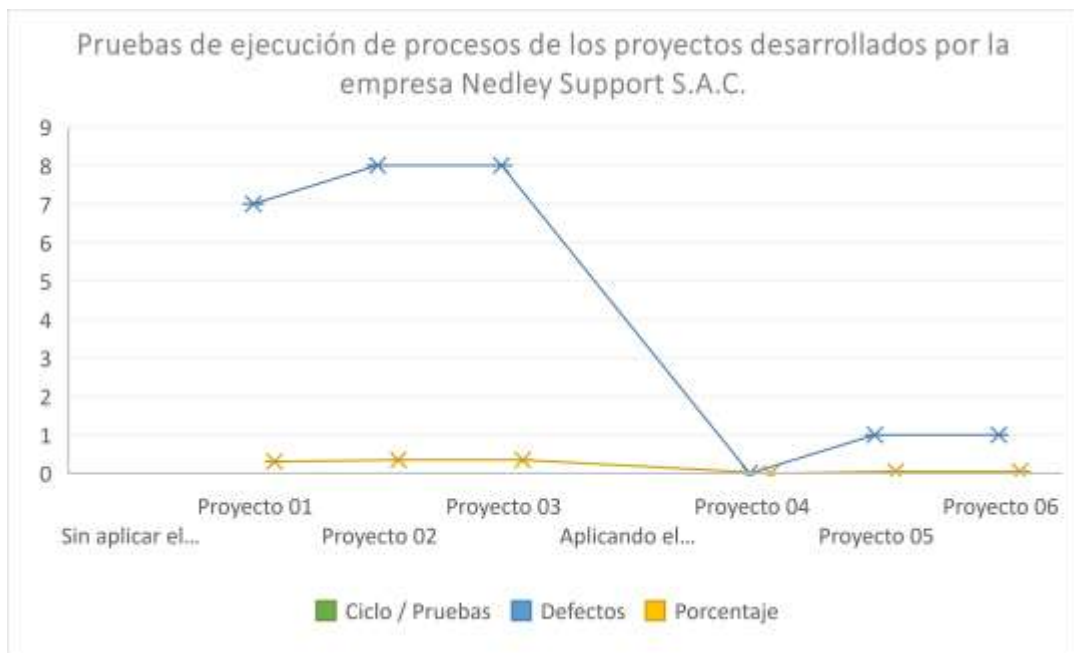


Figura 49, Gráfico de porcentaje de errores por ciclo de pruebas

Fuente, elaboración propia.

Como resultado de la realización de prueba piloto, se llegó a concluir que el modelo implantado es fundamental para el éxito de la apropiación de los procesos “nuevos” en la organización. Las evidencias de la documentación del proyecto 04 se detalla en el anexo 15, de prueba piloto realizado.

## **IV. CONCLUSIONES Y RECOMENDACIONES**

### **4.1. Conclusiones.**

- a. Se caracterizó la ISO/IEC 29110 como norma base de procesos de construcción de software, la cual se constituye de dos (02) procesos bien definidos: Proceso de Gestión de Proyectos y Proceso de Implementación de Software, los cuales se subdividen en cuatro y seis actividades respectivamente, sirviendo de apoyo a pequeñas organizaciones.
- b. Se diagnosticó el estado actual del proceso de desarrollo de software de una MYPE de servicios de TI, NEDLEY SUPPORT S.A.C., la cual disponía de un proceso constructivo de software en un solo proceso de siete actividades, la misma que no podía permitirse los recursos para establecer procesos de software definidos por las normas actuales existentes.
- c. Se diseñó un modelo de procesos ad hoc para el desarrollo de software por licencia para NEDLEY SUPPORT S.A.C., empleando BIZAGI, construyéndose por dos procesos (02): Proceso de Gestión de Proyecto (PGP) y Proceso de Implementación de Software (PIS), los cuales se subdividieron en tres y cuatro actividades respectivamente, con formatos, objetivos, alcances, tareas, diagramas SIPOC y diagramas BPMN.
- d. Se validó mediante juicio de expertos la propuesta del modelo de procesos ad hoc constituido por siete (07) actividades totales, considerando diez (10) indicadores, obteniendo un nivel de aceptación de 92%, por lo que se consideró que dicho modelo poseía una valoración considerada como "Muy Buena".
- e. Se realizó una prueba piloto del modelo de procesos ad hoc para el desarrollo de software por licencia en NEDLEY SUPPORT S.A.C., evidenciándose un aminoramiento en cuanto al número de defectos y un mejoramiento en la satisfacción del cliente, reflejando la mejora del desarrollo de software por licencia en dicha compañía trujillana.

#### **4.2. Recomendaciones.**

- a. Se recomienda a las MYPES de servicios de TI, adoptar el modelo de procesos ad hoc para el desarrollo de software por licencia desarrollado en esta investigación pues se encuentra basado en las prácticas descritas en la ISO/IEC 29110, por lo que está enfocado en una perspectiva de carácter internacional.
- b. Se recomienda a los investigadores, seguir los procesos de la ISO/IEC 29110 pues ayuda a la creación de los procesos propuestos, la generación de productos para la recolección de datos, debe estar acompañada de la validación por parte de las personas dentro de la empresa, para así comprobar su efectividad.
- c. Se recomienda al gerente general de la compañía caso de estudio, asegurar que los procesos, las actividades, las tareas y los formatos desarrollados en esta investigación sean seguidos al pie de la letra pues consideran los puntos esenciales que deben de practicarse en un proceso de construcción de software por licencia.

## REFERENCIAS.

- Aranibar, B., Méndez, J., & Mauricio, D. (2019). Quality assurance model for software development processes in smes.
- Basri, S., Abubakar, A., Almomani, M., & Thangiah, M. (2020). The organisational factors of software process improvement in small software industry: Comparative study.
- Buchalcevova, A. (2019). Using ArchiMate to model ISO/IEC 29110 standard for very small entities. *Computer Standards & Interfaces*, 65(1), 103-121. Obtenido de <https://www.sciencedirect.com/science/article/abs/pii/S0920548918303386>
- Castillo, L., Sánchez, S., Villarroel, J., & Sánchez, M. (2020). Evaluation of the implementation of a subset of ISO/IEC 29110 Software Implementation process in four teams of undergraduate students of Ecuador. An empirical software engineering experiment. *Computer Standards & Interfaces*, 70(1), 1-19. Obtenido de <https://www.sciencedirect.com/science/article/abs/pii/S0920548918304203>
- Castro, C., Quintana, L., Herrera, R., & Torres, J. (2021). *Propuesta de implementación de un estándar de calidad para proyectos de desarrollo de software de una pequeña empresa*. Lima: Universidad Peruana de Ciencias Aplicadas. Obtenido de <https://repositorioacademico.upc.edu.pe/handle/10757/656159>
- Coleman, G. (2016). Agile Software Development. *Software Quality Professional*, 19(1), 23-29. Obtenido de <https://www.semanticscholar.org/paper/6f696c50a49d8839c3c02fd0df0ce65f596e1794>
- Diario Gestión. (14 de Abril de 2016). *Perú tiene potencial para desarrollo de software dedicado al sector empresarial*. Obtenido de Gestión: <https://gestion.pe/tecnologia/peru-potencial-desarrollo-software-dedicado-sector-empresarial-117244-noticia/>
- Faustino, I., & Mejía, J. (2020). Proposal for a software development framework based on the ISO/IEC 29110 standard: Public organizations. *2020 9th International Conference On Software Process Improvement (CIMPS)* (págs.

- 132-140). Mazatlán: IEEE. Obtenido de <https://ieeexplore.ieee.org/document/9390135>
- Galván, S., Muñoz, M., Mejía, J., Laporte, C., & Negrete, M. (2020). Building a Guideline to Reinforce Agile Software Development with the Basic Profile of ISO/IEC 29110 in Very Small Entities. *International Conference on Software Process Improvement* (págs. 20-37). Cham: Springer. Obtenido de [https://link.springer.com/chapter/10.1007/978-3-030-63329-5\\_2](https://link.springer.com/chapter/10.1007/978-3-030-63329-5_2)
- Gunawan, F., & Budiardjo, E. (2021). A Quest of Software Process Improvements in DevOps and Kanban:: A Case Study in Small Software Company. *4th International Conference on Software Engineering and Information Management* (págs. 39-45). Yokohama: ACM Digital Library. Obtenido de <https://dl.acm.org/doi/abs/10.1145/3451471.3451478>
- Huapaya, J., Mori, F., Gamarra, N., & Philipps, P. (2021). *Propuesta de implementación del estándar ISO/IEC 29110 parte 5 - 2011 (perfil básico) para la mejora de los procesos de gestión de proyectos e implementación de software de una empresa peruana de desarrollo de software*. Lima: Universidad Peruana de Ciencias Aplicadas. Obtenido de <http://hdl.handle.net/10757/655410>
- Instituto Nacional de Estadística e Informática (INEI). (2019). *Perú: Estructura Empresarial, 2018*. Lima: INEI. Obtenido de [https://www.inei.gob.pe/media/MenuRecursivo/publicaciones\\_digitales/Est/Lib1703/libro.pdf](https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1703/libro.pdf)
- International Organization for Standardization (ISO). (2016). *ISO/IEC TR 29110-1:2016: Ingeniería de Software y Sistemas - Perfiles de ciclo de vida para Pequeñas Organizaciones (VSEs)*. Ginebra: ISO. Obtenido de <https://www.iso.org/obp/ui#iso:std:iso-iec:tr:29110:-1:ed-2:v1:es>
- Laporte, C., & O'Connor, R. (2017). Software process improvement standards and guides for very small organization: An overview of eight implementations. *CrossTalk: The Journal of Defense Software Engineering*, 30(3), 23-27. Obtenido de <https://doras.dcu.ie/21798/>
- Laporte, C., Munoz, M., Miranda, J., & O'Connor, R. (2018). Applying Software Engineering Standards in Very Small Entities: From Startups to Grownups.



- IEEE Software*, 35(1), 99-103. Obtenido de <https://ieeexplore.ieee.org/document/8239934>
- Mejía, J., Muñoz, M., Montoya, P., Girón, H., & Rodríguez, I. (2019). ISO/IEC 29110 implementation tools proposal (basic profile). *2019 8th International Conference On Software Process Improvement (CIMPS)* (págs. 1-11). León: IEEE. Obtenido de <https://ieeexplore.ieee.org/document/9082430>
- Minero, J., García, J., & Lara, E. (2020). Evaluation of the Implementation of the ISO/IEC 29110 Standard at the Software Development Center from the Institute Technological Superior of Nochistlán. *2020 9th International Conference On Software Process Improvement (CIMPS)* (págs. 12-18). Mazatlán: IEEE. Obtenido de <https://ieeexplore.ieee.org/document/9390105>
- Muñoz, M., Mejía, J., Peña, A., Lara, G., & Laporte, C. (2019). Transitioning international software engineering standards to academia: Analyzing the results of the adoption of ISO/IEC 29110 in four Mexican universities. *Computer Standards & Interfaces*, 66(1). Obtenido de <https://www.sciencedirect.com/science/article/abs/pii/S0920548918304173>
- O'Connor, R., & Laporte, C. (2012). Software Project Management in Very Small Entities with ISO/IEC 29110. *European Conference on Software Process Improvement* (págs. 330-341). Berlín: Springer. Obtenido de [https://link.springer.com/chapter/10.1007/978-3-642-31199-4\\_29](https://link.springer.com/chapter/10.1007/978-3-642-31199-4_29)
- Ochoa, V. (9 de Marzo de 2019). *Mercado de la informática en Perú crecerá 9.7% este año*. Obtenido de *Diario Gestión*: <https://gestion.pe/economia/empresas/mercado-informatica-peru-crecera-9-7-ano-260535-noticia/>
- O'Connor, R., & Coleman, G. (2009). Ignoring 'Best Practice': why Irish software SMEs are rejecting CMMI and ISO 9000. *Australasian Journal of Information Systems*, 16(1), 7-30. Obtenido de <https://ulir.ul.ie/handle/10344/2098>
- O'Connor, R., & Laporte, C. (2017). The evolution of the ISO/IEC 29110 set of standards and guides. *International Journal of Information Technologies and Systems Approach (IJITSA)*, 10(1), 1-21. Obtenido de <https://www.igi-global.com/article/the-evolution-of-the-isoiec-29110-set-of-standards-and-guides/169765>

- Özcan, Ö., & McCaffery, F. (2019). To what extent the medical device software regulations can be achieved with agile software development methods? XP—DSDM—Scrum. *The Journal of Supercomputing*, 75(1), 5227-5260. Obtenido de <https://link.springer.com/article/10.1007/s11227-019-02793-x>
- Pérez, F., & Giraldo, F. (2019). Proposal of a Software Development Process for a Poultry Company in Colombia. *2019 XLV Latin American Computing Conference (CLEI)* (págs. 1-10). Ciudad de Panamá: IEEE. Obtenido de <https://ieeexplore.ieee.org/document/9073869>
- Preston, R. (22 de Septiembre de 2022). *5 Types of Software Licenses: Definitions, Examples and Tips*. Obtenido de Indeed: <https://www.indeed.com/career-advice/career-development/types-of-software-license>
- Rada, D. (2016). El rigor en la investigación cualitativa: técnicas de análisis, credibilidad, transferibilidad y confirmabilidad. *Revista venezolana de investigación*, 7(1), 17-26. Obtenido de <https://www.semanticscholar.org/paper/c827fbfd78a3e0ad8808071c2fb9464cf2974d68>
- Rengan, S., & Rusli, H. (2021). Development of Software Process Guideline to Improve Software Process Using ISO/IEC 29110-5- 1-2 Software Implementation Process. *Open International Journal of Informatics*, 9(1), 52-59. Obtenido de <https://oiji.utm.my/index.php/oiji/article/view/168>
- Sánchez, M., Colomo, R., de Amescua, A., & O'Connor, R. (2016). The Route to Software Process Improvement in Small- and Medium-Sized Enterprises. *Managing software process evolution* (págs. 109-136). Cham: Springer. Obtenido de [https://link.springer.com/chapter/10.1007/978-3-319-31545-4\\_7](https://link.springer.com/chapter/10.1007/978-3-319-31545-4_7)
- Šmite, D., Moe, N., Šāblis, A., & Wohlin, C. (2017). Software teams and their knowledge networks in large-scale software development. *Information and Software Technology*, 86(1), 71-86. Obtenido de <https://www.sciencedirect.com/science/article/pii/S0950584917300435>
- Snik. (2022). *What Is a Software License? 5 Types of Software Licenses You Need to Know About*. Obtenido de Snyk Limited: <https://snyk.io/learn/what-is-a-software-license/>
- Suteeca, K. (2020). A Software Process Gap Analysis Methodology for Very Small Entity. *2020 Joint International Conference on Digital Arts, Media and*

- Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)* (págs. 190-193). Pattaya: IEEE. Obtenido de <https://ieeexplore.ieee.org/document/9090745>
- Villanueva, E., & Muñoz, M. (2020). Proposal of a framework for software product and process quality assurance for very small entities. *2020 9th International Conference On Software Process Improvement (CIMPS)* (págs. 141-147). Mazatlán: IEEE. Obtenido de <https://ieeexplore.ieee.org/document/9390148>
- Wing, J., Andrew, T., & Petkov, D. (2017). The changing nature of user involvement in information system development projects. *Conference on Information Communication Technology and Society (ICTAS)* (págs. 1-6). Durban: IEEE. Obtenido de <https://ieeexplore.ieee.org/document/7920655>
- Wynn, D., & Clarkson, P. (2018). Process models in design and development. *Research in Engineering Design*, 29(2), 161-202. Obtenido de <https://link.springer.com/article/10.1007/s00163-017-0262-7>
- Yang, W., Hosseinian, A., Jraisat, L., & Rangaswamy, E. (2021). Software License Audit and Security Implications. *Cybersecurity, Privacy and Freedom Protection in the Connected World* (págs. 171-181). Cham: Springer. Obtenido de [https://link.springer.com/chapter/10.1007/978-3-030-68534-8\\_12](https://link.springer.com/chapter/10.1007/978-3-030-68534-8_12)

## ANEXOS.

### Anexo 1. Resolución de aprobación del proyecto de investigación



UNIVERSIDAD  
SEÑOR DE SIPÁN

#### FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

#### RESOLUCIÓN N°0445-2021/FIAU-USS

Pimentel, 27 de mayo de 2021

#### VISTO:

El Acta de reunión N°1305-2021 del Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS remitida mediante oficio N°0227-2021/FIAU-IS-USS de fecha 19 de mayo de 2021, y;

#### CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21° señala: "Los temas de trabajo de investigación, trabajo académico y tesis son aprobados por el Comité de Investigación y derivados a la Facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El periodo de vigencia de los mismos será de dos años, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma.

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24° señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; es individual o en pares para obtener un título profesional. Asimismo, en su artículo 25° señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C."

Que, según documentos de Vistos el Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS acuerdan aprobar los temas de las Tesis a cargo de los estudiantes del curso de Investigación I que se detallan en el anexo de la presente Resolución.

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

#### SE RESUELVE:

**ARTÍCULO 1°: APROBAR**, el tema de la Tesis perteneciente a la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y MEDIO AMBIENTE, a cargo de los estudiantes del Programa de estudios de INGENIERÍA DE SISTEMAS según se detalla en el anexo de la presente Resolución.

**ARTÍCULO 2°: ESTABLECER**, que la inscripción del Tema de la Tesis se realice a partir de emitida la presente resolución y tendrá una vigencia de dos (02) años.

**ARTÍCULO 3°: DEJAR SIN EFECTO**, toda Resolución emitida por la Facultad que se oponga a la presente Resolución.

**REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE**


Dr. Mario Fernando Ramos Moscol  
Decano - Facultad de Ingeniería,  
Arquitectura y Urbanismo  
UNIVERSIDAD SEÑOR DE SIPÁN S.A.C.


MRA. María Huelia Sialer Rivera  
Secretaría Académica / Facultad de Ingeniería,  
Arquitectura y Urbanismo  
UNIVERSIDAD SEÑOR DE SIPÁN S.A.C.

Cc: Interesado, Archivo

**FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO  
RESOLUCIÓN N°0445-2021/FIAU-USS**

Pimentel, 27 de mayo de 2021

**ANEXO**

<b>N°</b>	<b>AUTOR (ES)</b>	<b>TEMA DE TESIS</b>
1	RIMARACHIN ESCRIBANO NERI RUT NIÑO MORENO NAJHELY YAMILETT	EVALUACIÓN DE TÉCNICAS DE CIFRADO PARA EL INTERCAMBIO DE DATOS DE INTERNET DE LAS COSAS EN EL ÁMBITO DE LA SALUD
2	GUEVARA CHAMBERGO JHON DENNIS BOBADILLA CAMPOS ROLANDO MARTIN	DESARROLLO DE UNA METODOLOGÍA DE GESTIÓN DE RIESGOS AD HOC BASADA EN MARCOS INTERNACIONALES Y BUENAS PRÁCTICAS PARA UNA EMPRESA MANUFACTURERA PERUANA
3	CIEZA CELIS JESUS ABELARDO OJEDA ROMERO ANTHONNY JHONATAN	EVALUACIÓN DEL DESEMPEÑO DE LOS ESQUEMAS DE SEGURIDAD DE RED PARA COMBATIR VULNERABILIDADES EN REDES INALÁMBRICAS BASADAS EN EL PROTOCOLO WPA2
4	MENDOZA FERRÉ ESPERANZA NATALY CABRERA SANCHEZ KEVIN ALONSO	COMPARACIÓN DEL RENDIMIENTO DE TECNOLOGÍAS DE VIRTUALIZACIÓN PARA EL DESPLIEGUE DE APLICACIONES CON ARQUITECTURA DE MICROSERVICIOS
5	TEMOCHE GOMEZ LENNIN BILLEY	DESARROLLO DE UN MÉTODO PARA DETECTAR CON EFICIENCIA LAS VULNERABILIDADES INFORMÁTICAS DE ATAQUE CROSS-SITE SCRIPTING UTILIZANDO TÉCNICAS DE APRENDIZAJE AUTOMÁTICO
6	CASTRO MEDINA MIGUEL ANGEL	IMPLEMENTACIÓN DE UNA METODOLOGÍA AD HOC DE GESTIÓN DE LA SEGURIDAD DE LA INFORMACIÓN PARA UNA EMPRESA EDITORA DE DIARIO REGIONAL PERUANO
7	MURO ESPINOZA JUAN JOSE	DESARROLLO DE UNA METODOLOGÍA AD HOC DE GESTIÓN DE LA SEGURIDAD DE LA INFORMACIÓN PARA UN INSTITUTO SUPERIOR PEDAGÓGICO PERUANO
8	DIAZ ZAVALA ROXANA KARINA FRIAS VASQUEZ LADY	DESARROLLO DE UNA METODOLOGÍA AD HOC DE GESTIÓN DE LA SEGURIDAD DE LA INFORMACIÓN PARA UNA UNIDAD DE GESTIÓN EDUCATIVA PERUANA
9	CARRASCO BORDA APARICIO	DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110
10	OTERO MORALES JAVIER LIZARDO AQUINO SOSA NOELIA STEPHANY	DESARROLLO DE UN MODELO DE PROCESOS BASADO EN NORMAS DE PEQUEÑAS ORGANIZACIONES PARA MEJORAR LA CONSTRUCCIÓN DE SOFTWARE EN UN ÁREA DE DESARROLLO DE GOBIERNO MUNICIPAL
11	CALDERON YNOÑAN PAMELA DEL CARMEN PRIETO NEIRA FRANCK ALBERSON	DESARROLLO DE UN MÉTODO BAJO EL ENFOQUE ÁGIL EN ENTORNOS DE EXPERIENCIA DE USUARIO UI/UX PARA ASEGURAR LA USABILIDAD WEB
12	FLORES TINEO HUGO GALVANI DOLORIER POMA RONY RAUL	EVALUACIÓN DE LA USABILIDAD EN ENTORNOS VIRTUALES DE APRENDIZAJE PARA USUARIOS DE LAS ZONAS RURALES DEL PERÚ UTILIZANDO LA NORMA ISO/IEC 25010
13	CHANCAFE CASTRO JULIO JOEL	DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE PARA UNA MUNICIPALIDAD BASADO EN ISO/IEC 29110
14	SALAZAR DAVILA GIANFRANCO STEVEN	COMPARACIÓN DE TÉCNICAS DE VALIDACIÓN DE REQUISITOS DE SOFTWARE PARA MEDIR LA INFLUENCIA EN EL ÉXITO DE LOS PROYECTOS DE DESARROLLO EN PEQUEÑAS EMPRESAS PERUANAS
15	RIOJA MESIA CHARLES SEGUNDO FERNANDEZ RIOJA JUAN NICANOR	IMPLEMENTACIÓN DE UN MODELO DE GESTIÓN DE INCIDENCIAS BASADO EN ITIL PARA MEJORAR EL SERVICIO DE TI EN UNA MUNICIPALIDAD DISTRITAL DE LA REGIÓN LAMBAYEQUE
16	ALFARO PAJARES JUAN PEDRO	EVALUACIÓN DEL DESEMPEÑO DE PROCESOS DE NEGOCIO GESTIONADOS POR BPM EN UNA EMPRESA CONSTRUCTORA PERUANA
17	MONSALVE FERNANDEZ LENIN ESTALIN	IMPLEMENTACIÓN DE UN MODELO DE GESTIÓN DE SERVICIOS DE TI BASADO EN ITIL PARA MEJORAR LA GESTIÓN DE LOS SERVICIOS DE LA DIRECCIÓN DE TECNOLOGÍA DE UN GOBIERNO REGIONAL PERUANO
18	PEREZ CAMPOS DE QUIROZ BETTY MAGALY	EVALUACIÓN DEL DESEMPEÑO DE PROCESOS DE NEGOCIO GESTIONADOS POR BPM EN UNA MICRO EMPRESA PERUANA DESARROLLADORA DE SOFTWARE
19	MONTJOY PITA BRUNO	DESARROLLO DE UN SISTEMA DE RECOMENDACIÓN AUTOMÁTICA PARA EL TRATAMIENTO DE LAS PLAGAS EN CULTIVOS DE ARROZ DE LAS VARIEDADES QUE SE PRODUCEN EN LA REGIÓN LAMBAYEQUE
20	CRUZ FLORES JOSE ANTONIO CHAVEZ ANGULO GERMAN NEPTALI	IMPLEMENTACIÓN DE ARQUITECTURA EMPRESARIAL BASADO EN METODOLOGÍA ÁGIL PARA ALINEAR LAS TECNOLOGÍAS DE INFORMACIÓN CON LOS OBJETIVOS DE NEGOCIO DE UN ESTABLECIMIENTO PERUANO DE SALUD BUCAL

## Anexo 2. Carta de aceptación de la institución para la recolección de datos

"Decenio de la Igualdad de oportunidades para mujeres y hombres"  
"Año del Bicentenario del Perú: 200 años de independencia"

### CARTA DE ACEPTACIÓN

16 de junio del 2021

**DR. MARIO FERNANDO RAMOS MOSCOL**

Decano de la Facultad de Ingeniería, Arquitectura y Urbanismo

**UNIVERSIDAD SEÑOR DE SIPÁN - USS**

Por el presente documento, yo Oscar Manuel Olaya Vásquez, identificado con DNI N° 70672139, en mi calidad de representante legal de la empresa "Tecnología de la Información y Servicios Informáticos Nedley Support S.A.C", autorizo al estudiante: Aparicio Carrasco Borda con DNI N°: 70060686, y con código universitario N°: 2161802200 de la carrera profesional de Ingeniería de Sistemas de la Universidad Señor de Sipán, a utilizar el nombre e información confidencial de la empresa que represento, para el desarrollo de su tesis denominado "**DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110**". La empresa, precisa que toda información proporcionada será para uso exclusivamente académico, caso contrario, el estudiante quedará sujeto a la responsabilidad civil por daños y perjuicios que cause, así como, a las sanciones de carácter penal o legal a que hubieres lugar.

Sin más por el momento, quedo de usted para cualquier duda o aclaración.

Atentamente,

  
Oscar Olaya Vásquez



### Anexo 3. Matriz de consistencia lógica

Alumno:		CARRASCO BORDA APARICIO		Alumno:			
Título		DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110					
Tipo de investigación	Problema	Variables	Indicadores	Población	Muestra	Método de recolección de Datos	Técnicas de procesamiento de datos
Tecnológica Aplicada	¿Cómo mejorar el desarrollo de software por licencia de una MYPE de servicios de TI?	<p>VI: Variable independiente: Modelo de procesos ad hoc basado en ISO/IEC 29110</p> <p>VD: Variable dependiente: Desarrollo de software por licencia</p>	<p>Indicadores variable independiente:</p> <ul style="list-style-type: none"> <li>Nivel de Claridad</li> <li>Nivel de Objetividad</li> <li>Nivel de Actualidad</li> <li>Nivel de Organización</li> <li>Nivel de Suficiencia</li> <li>Nivel de Intencionalidad</li> <li>Nivel de Consistencia</li> <li>Nivel de Coherencia</li> <li>Nivel de Metodología</li> <li>Nivel de Pertinencia</li> </ul> <p>Indicadores variable dependiente:</p> <ul style="list-style-type: none"> <li>Cantidad de defectos</li> <li>Porcentaje de Clientes Satisfechos</li> </ul>	En esta investigación, la población a tomar en cuenta fue un caso de estudio, la compañía NEDLEY SUPPORT S.A.C., con sede en la Av. Costa Rica N° 231 Urb. Torres Araujo, Trujillo, La Libertad, ya que es una MYPE de servicios de TI, que es justamente el contexto de estudio que se requería y es una compañía a la que se tuvo acceso y permiso para desarrollar la investigación, tal y como consta en el Anexo 2.	En esta investigación, la muestra a tomar en cuenta fue un caso de estudio, la compañía NEDLEY SUPPORT S.A.C., con sede en la Av. Costa Rica N° 231 Urb. Torres Araujo, Trujillo, La Libertad, ya que es una MYPE de servicios de TI, que es justamente el contexto de estudio que se requería y es una compañía a la que se tuvo acceso y permiso para desarrollar la investigación, tal y como consta en el Anexo 2.	<p>Variable Independiente</p> <p>Juicio de expertos</p> <p>Variable Dependiente</p> <p>Observación Directa</p>	<p>Variable Independiente</p> <p>Se envió una ficha de Juicio de Expertos a cada uno de los expertos quienes emitieron su veredicto en cuanto al modelo de procesos ad hoc considerando diez indicadores.</p> <p>Variable Dependiente</p> <p>Se logró procesar la información de cada uno de los indicadores de las dos dimensiones para esta variable.</p>
Diseño de investigación	Hipótesis	Objetivo General	Objetivos específicos	Método propuesto y desarrollado		Resultados preliminares	
Cuasi Experimental	Mediante el desarrollo de un modelo de procesos ad hoc basado en ISO/IEC 29110 se mejorará el desarrollo de software por licencia para una MYPE de servicios de TI.	Desarrollar un modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110	<p>a. Caracterizar ISO/IEC 29110 como norma base de procesos de construcción de software.</p> <p>b. Elaborar un diagnóstico base del estado actual del proceso de desarrollo de software de una MYPE de servicios de TI previamente seleccionada.</p> <p>c. Diseñar un modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI.</p> <p>d. Validar mediante juicio de expertos la propuesta del modelo de procesos ad hoc desarrollado.</p> <p>e. Realizar una prueba piloto del modelo de procesos ad hoc para el desarrollo de software por licencia en una MYPE de servicios de TI.</p>			<p>Modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110 considerando diez (10) indicadores, lo que conllevó a tener como resultado, un nivel de aceptación de 92%, por lo que se consideró que dicho modelo desarrollado poseía una valoración considerada como "Muy Buena".</p> <p>Se evidenció un aminoramiento en cuanto al número de defectos en dos grupos de proyectos PRE TEST y POST TEST, así como también, un mejoramiento en el grado de satisfacción del cliente llegando a obtener un 100% en dicho indicador, reflejando la mejora del desarrollo de software por licencia en dicha compañía trujillana caso de estudio.</p>	

## Anexo 4. Instrumento de recolección de datos - Ficha de Juicio de Expertos

### VALIDACIÓN POR JUICIO DE EXPERTOS INSTRUMENTO DE INVESTIGACIÓN

Título de la investigación:

**DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110**

Autor:

Bach. Carrasco Borda Aparicio

#### **Objetivo:**

El objetivo del presente informe es someter a evaluación el presente modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110.

#### **I. DATOS GENERALES DEL EXPERTO**

1.1. Apellidos y nombres del experto: \_\_\_\_\_

1.2. Grado Académico y Profesión: \_\_\_\_\_

1.3. Áreas de Experiencia Profesional: \_\_\_\_\_

1.4. Institución donde labora: \_\_\_\_\_

1.5. Nombre del instrumento motivo de evaluación: **VALIDACIÓN DE EXPERTOS DEL MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110.**

#### **II. VALIDACIÓN**

Se utilizarán los siguientes indicadores y criterios para la evaluación del instrumento: CLARIDAD, OBJETIVIDAD, ACTUALIDAD, ORGANIZACIÓN, SUFICIENCIA, INTENCIONALIDAD, CONSISTENCIA, COHERENCIA, METODOLOGÍA, PERTINENCIA.

Valoración: Deficiente - [50-200], Baja - [250-400], Regular - [450-600], Buena: [650-800], Muy Buena - [850-1000]



### ASPECTOS DE VALIDACIÓN

Indicadores	Criterios	Deficiente				Baja				Regular				Buena				Muy buena				
		5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	
<b>CLARIDAD</b>	Ésta formulado con lenguaje apropiado																					
<b>OBJETIVIDAD</b>	Está expresado en conductas observables																					
<b>ACTUALIDAD</b>	Adecuado al avance del desarrollo de software																					
<b>ORGANIZACIÓN</b>	Existe una organización lógica																					
<b>SUFICIENCIA</b>	Comprende los aspectos en cantidad y calidad																					
<b>INTENCIONALIDAD</b>	Adecuado para valorar el desarrollo de software																					
<b>CONSISTENCIA</b>	Basado en aspectos teóricos científicos del desarrollo de software																					
<b>COHERENCIA</b>	Entre cada uno de los pasos del modelo																					
<b>METODOLOGÍA</b>	La estrategia responde al propósito de la investigación																					
<b>PERTINENCIA</b>	Es útil y adecuado para la investigación																					

**VALORACIÓN:** \_\_\_\_\_

**OPINIÓN DE APLICABILIDAD:** \_\_\_\_\_

Lugar y fecha: Chiclayo, \_\_\_\_ Junio del 2022.

\_\_\_\_\_

Anexo 5. Instrumento de recolección de datos - Ficha de Observación

<b>TÍTULO</b>	<b>DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110</b>
<b>Proyecto</b>	Proyecto Sistema Web de Recursos Humanos (POST-TEST)
<b>Dimensión</b>	Defectos por proyecto
<b>Indicador</b>	Cantidad de defectos
<b>Responsable</b>	Bach. Carrasco Borda Aparicio
<b>Fecha</b>	08/11/2021
<b>PROCESO DE GESTIÓN DE PROYECTOS</b>	
Actividad 1: PGP-ACT-01: Planeación del Proyecto	
Defecto:	En caso de existir algún defecto, será reportado en esta celda.
Criterio actual	"SIN DEFECTO" / "CON DEFECTO"
Actividad 2: PGP-ACT-02: Despliegue de Plan del Proyecto	
Defecto:	
Criterio actual	
Actividad 3: PGP-ACT-03: Evaluación y Monitoreo del Proyecto	
Defecto:	
Criterio actual	

<b>PROCESO DE IMPLEMENTACIÓN DE SOFTWARE</b>	
Actividad 4: PIS-ACT-01: Análisis de Requerimientos de Software	
Defecto:	
Criterio actual	
Actividad 5: PIS-ACT-02: Diseño y Arquitectura del Software	
Defecto:	
Criterio actual	
Actividad 6: PIS-ACT-03: Construcción del Software	
Defecto:	Error en estimación de tiempos.
Criterio actual	
Actividad 7: PIS-ACT-04: Entrega del Software	
Defecto:	
Criterio actual	



Estimado cliente, el presente cuestionario es para conocer su satisfacción general respecto al producto de software entregado por la empresa. Muchas gracias de antemano. Sírvase marcar con una (X) la alternativa que se adecue a su criterio.





<b>TÍTULO</b>	<b>DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110</b>
<b>Proyecto</b>	
<b>Cliente</b>	
<b>Responsable</b>	Bach. Carrasco Borda Aparicio
<b>Fecha</b>	08/11/2021

<b>SATISFACCIÓN DEL CLIENTE</b>		
<b>Pregunta</b>	<b>SÍ</b>	<b>NO</b>
En general, con respecto al producto de software entregado por NEDLEY SUPPORT S.A.C., ¿usted se encuentra satisfecho?		
<b>Comentarios</b>		

Anexo 7. Herramientas empleadas para el desarrollo del modelo ad hoc

TIPO	HERRAMIENTA	JUSTIFICACIÓN
Herramienta de Gestión	<p>JIRA</p> 	<p>El proceso de Gestión de Proyectos tiende a realizarse de forma descentralizada en Sistema Inteligente ERP SAC, no logrando un adecuado control de las tareas. JIRA permite un control centralizado de la Gestión de Proyectos, así como también ofrece plantillas y colaboración para mejorar la productividad en los proyectos.</p>
Herramienta de Gestión	<p>29110-in-a-box</p> 	<p>Usando el plugin 29110-in-a-box para JIRA, se permite además agregar plantillas y flujos de trabajo basados en ISO 29110, ayudando a la organización a adecuarse fácilmente a este estándar.</p>
Herramienta de Gestión	<p>Trello</p> 	<p>Trello es un software de administración de proyectos con interfaz web y con cliente para iOS y Android para organizar proyectos. Empleando el sistema Kanban, para el registro de actividades con tarjetas virtuales organiza tareas, permite agregar listas, adjuntar archivos, etiquetar eventos, agregar comentarios y compartir tableros. Trello es un tablón virtual en el que se pueden colgar ideas, tareas, imágenes o enlaces. Es versátil y fácil de usar pudiendo usarse para</p>

		cualquier tipo de tarea que requiera organizar información.
Herramienta de Gestión	<p>MS-Project</p> 	<p>Microsoft Project (o MSP) es un software de administración de proyectos y programas de proyectos desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.</p> <p>El software Microsoft Office Project en todas sus versiones (la versión 2019 es la más reciente a enero de 2019) es útil para la gestión de proyectos, aplicando procedimientos descritos en el PMBoK del Project Management Institute</p>
Herramienta de Implementación	<p>Suite de Google</p> 	<p>Ofrece de manera gratuita diferentes servicios de oficina que podrían ser de utilidad en el proceso de implementación de la ISO/IEC 29110, tales como el correo electrónico, docs, slides, hojas de cálculo, formularios, servicio de almacenamiento, y más.</p>
Framework de Pruebas Unitarias	<p>Junit</p> 	<p>Actualmente no se utilizan herramientas automatizadas para la realización de pruebas unitarias. Esta herramienta ayuda a automatizar este proceso reduciendo el tiempo de pruebas.</p>

Herramienta de Cobertura	<p>CodeCover</p> 	<p>La herramienta de cobertura nos permite saber si todo el código fuente es utilizado. Esto evita defectos causados por condiciones que nunca se cumplen en el código.</p>
Herramienta de revisión de código	<p>Findbugs</p> 	<p>Herramienta de revisión de código estático que permite detectar bugs de diversos tipos en Java, así como también, detecta que las buenas prácticas de codificación se estén cumpliendo.</p>
Herramienta para repositorio de proyecto	<p>GitHub</p> 	<p>La herramienta permite alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.</p>
Herramienta para repositorio de proyecto	<p>Dropbox</p> 	<p>Dropbox es un servicio de alojamiento de archivos multiplataforma en la nube, operado por la compañía estadounidense Dropbox. El servicio permite a los usuarios almacenar y sincronizar archivos en línea y entre ordenadores y compartir archivos y carpetas con otros usuarios y con tabletas y móviles. Existen versiones gratuitas y de pago, cada una de las cuales tiene opciones variadas. La versión móvil está disponible para Android, Windows Phone, Blackberry e iOS (Apple). Actualmente cuenta con más de 500 millones de usuarios registrados en todo el mundo.</p>

Anexo 8. Resultados individuales de Juicio de Expertos  
EXPERTO 01 - Ruiz Gómez Woolder

**VALIDACIÓN POR JUICIO DE EXPERTOS  
INSTRUMENTO DE INVESTIGACIÓN**

Título de la investigación:

**DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110**

Autor:

Bach. Carrasco Borda Aparicio

**Objetivo:**

El objetivo del presente informe es someter a evaluación el presente modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110.

**I. DATOS GENERALES DEL EXPERTO**

- 1.1. Apellidos y nombres del experto: RUIZ GÓMEZ WOOLDER  
1.2. Grado Académico y Profesión: DRG / ING. DE SISTEMAS  
1.3. Áreas de Experiencia Profesional: SEB. INFORMATICA / DES. SOFTWARE  
1.4. Institución donde labora: DIGEMID - MINSA  
1.5. Nombre del instrumento motivo de evaluación: **VALIDACIÓN DE EXPERTOS DEL MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110.**

**II. VALIDACIÓN**

Se utilizarán los siguientes indicadores y criterios para la evaluación del instrumento: CLARIDAD, OBJETIVIDAD, ACTUALIDAD, ORGANIZACIÓN, SUFICIENCIA, INTENCIONALIDAD, CONSISTENCIA, COHERENCIA, METODOLOGÍA, PERTINENCIA.

Valoración: Deficiente - [50-200], Baja - [250-400], Regular - [450-600], Buena: [650-800], Muy Buena - [850-1000]





## Perfil profesional del evaluador 1.

### **Doctor © Woolder Ruiz Gómez**

Profesional en Ingeniería de Sistemas, con grado académico de Maestro en Ingeniería de Sistemas con Mención en Tecnologías de la Información, Candidato a Doctor, Consultor Senior con más de quince años de experiencia en Proyectos de desarrollo de software e Infraestructura de TI en el sector público y privado.

Ha desempeñado funciones principales dentro del Sector Estatal como en el Privado en las siguientes instituciones:

- ✚ Consultor en Tecnologías de Información, implementador de soluciones basados en GNU/Linux y Software Libre.
- ✚ Especialista en Gestión de Infraestructura de T.I Instituto Tecnológico de la Producción.
- ✚ Especialista en Administración de Servidores y Plataforma de Virtualización Ministerio del Interior Perú.
- ✚ Especialista en Infraestructura de Tecnología de la Información GNU/Linux y Software Libre Ministerio de Cultura (Perú).
- ✚ Administración de Infraestructura de Servidores Microsoft Windows y GNU/Linux, implementación y administración de soluciones basados en Software Libre, Virtualización de Servidores con VMware y Software Libre.
- ✚ Consultor externo de Tecnologías de Información, responsable de la virtualización y consolidación de los servidores Microsoft y GNU/Linux con herramientas open source, reinstalación y mejoramiento de los servicios de red del Ministerio, administración de redes y capacitación.

## EXPERTO 02 - Tapia Flores Jesús Benjamín

### VALIDACIÓN POR JUICIO DE EXPERTOS INSTRUMENTO DE INVESTIGACIÓN

Título de la investigación:

**DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110**

Autor:

Bach. Carrasco Borda Aparicio

**Objetivo:**

El objetivo del presente informe es someter a evaluación el presente modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110.

#### I. DATOS GENERALES DEL EXPERTO

- 1.1. Apellidos y nombres del experto: TAPIA FLORES JESUS BENJAMIN
- 1.2. Grado Académico y Profesión: INGENIERO SISTEMAS / DOCTOR (C)
- 1.3. Áreas de Experiencia Profesional: DESARROLLO DE SOFTWARE
- 1.4. Institución donde labora: ESF TECHNOLOGY S.A.C.
- 1.5. Nombre del instrumento motivo de evaluación: **VALIDACIÓN DE EXPERTOS DEL MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110.**

#### II. VALIDACIÓN

Se utilizarán los siguientes indicadores y criterios para la evaluación del instrumento: CLARIDAD, OBJETIVIDAD, ACTUALIDAD, ORGANIZACIÓN, SUFICIENCIA, INTENCIONALIDAD, CONSISTENCIA, COHERENCIA, METODOLOGÍA, PERTINENCIA.

Valoración: Deficiente - [50-200], Baja - [250-400], Regular - [450-600], Buena: [650-800], Muy Buena - [850-1000]

### ASPECTOS DE VALIDACIÓN

Indicadores	Criterios	Deficiente										Buena					Muy buena				
		5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
CLARIDAD	Esta formulado con lenguaje apropiado																			X	
OBJETIVIDAD	Está expresado en conductas observables																			X	
ACTUALIDAD	Adecuado al avance del desarrollo de software																			X	
ORGANIZACIÓN	Existe una organización lógica																			X	
SUFICIENCIA	Comprende los aspectos en cantidad y calidad																			X	
INTENCIONALIDAD	Adecuado para valorar el desarrollo de software																			X	
CONSISTENCIA	Basado en aspectos teóricos científicos del desarrollo de software																			X	
COHERENCIA	Entre cada uno de los pasos del modelo																			X	
METODOLOGÍA	La estrategia responde al propósito de la investigación																			X	
PERTINENCIA	Es útil y adecuado para la investigación																			X	

VALORACIÓN: 950

OPINIÓN DE APLICABILIDAD: BUENA PROPUESTA

Lugar y fecha: Chiclayo, 30 Junio del 2022.



## Perfil profesional del evaluador 2.

### **Tapia Flores Jesús Benjamín**

Ingeniero de Sistemas, Maestro en Ingeniería de Sistemas con Mención en Tecnologías de la Información, de la universidad Cesar Vallejo.

- ⇒ Desempeña funciones principales dentro del Sector Estatal como Desarrollador de Software, Especialista en Redes y Comunicaciones, Programador, Soporte TI, Proyectos TI, Consultoría en TICs y gestión de procesos SCRUM.
  
- ⇒ Amplia experiencia como consultor especializado en implementación de Proyectos de Tecnologías de la Información, con más de 3 años de experiencia comprobada en implementación de soluciones a nivel corporativo, habilidad para asimilar rápidamente cualquier plataforma tecnológica, con facilidad de comunicación a gran nivel, con conocimientos del idioma inglés a nivel intermedio.
  
- ⇒ Ha liderado proyectos de implementación de Infraestructura de Tecnologías de la Información en las Instituciones Públicas donde ha laborado.
  
- ⇒ Con amplia experiencia en procesos de adquisición de equipamiento de T.I (Bienes y Servicios).

**VALIDACIÓN POR JUICIO DE EXPERTOS  
INSTRUMENTO DE INVESTIGACIÓN**

Título de la investigación:

**DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110**

Autor:

Bach. Carrasco Borda Aparicio

**Objetivo:**

El objetivo del presente informe es someter a evaluación el presente modelo de procesos ad hoc para el desarrollo de software por licencia para una MYPE de servicios de TI basado en ISO/IEC 29110.

**I. DATOS GENERALES DEL EXPERTO**

1.1. Apellidos y nombres del experto: Reyna Robles Jaime Luis

1.2. Grado Académico y Profesión: Ingeniería de sistemas

1.3. Áreas de Experiencia Profesional: Desarrollo de software

1.4. Institución donde labora: Symbiosis

1.5. Nombre del instrumento motivo de evaluación: **VALIDACIÓN DE EXPERTOS DEL MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110.**

**II. VALIDACIÓN**

Se utilizarán los siguientes indicadores y criterios para la evaluación del instrumento: CLARIDAD, OBJETIVIDAD, ACTUALIDAD, ORGANIZACIÓN, SUFICIENCIA, INTENCIONALIDAD, CONSISTENCIA, COHERENCIA, METODOLOGÍA, PERTINENCIA.

Valoración: Deficiente - [50-200], Baja - [250-400], Regular - [450-600], Buena: [650-800], Muy Buena - [850-1000]



### Perfil profesional del evaluador 3.

#### **Reyna Robles Jaime Luis**

Ingeniero de Sistemas graduado de la Universidad César Vallejo, candidato a Maestría, con experiencia en desarrollo de Software, realiza labores en la empresa consultores Symbiosis

- ❖ Especialista en Diseño, implantación y mantenimiento de un Sistema de Gestión de la Calidad según la ISO 9001, outsourcing de la función de calidad, realización de auditorías internas, asistencia en auditorías externas, así como integración con otros sistemas de gestión (ambiental, seguridad de la información, innovación, etc.).
- ❖ Realiza labores para las organizaciones en la identificación y definición de sus procesos para fomentar la Gestión por Procesos, así como la reingeniería y la mejora de procesos como elementos para mejorar su competitividad.
- ❖ Además, ayuda a implantar a la organización, el orden y la limpieza a través de herramientas como la metodología 5S, para la mejora de una organización con análisis de las necesidades de los clientes y la evaluación de su grado de satisfacción con conocimiento en modelo de procesos de la norma ISO/IEC 29110. Lidera proyectos de desarrollo de software basadas en la implementación de normas técnicas de calidad.



Anexo 9. Resultados individuales de Ficha de Observación

Proyecto 01: Sistema de Tempus Web - Control Tiempo (PRE-TEST)

<b>TÍTULO</b>	DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110
<b>Proyecto</b>	Proyecto N°01 - Sistema de Tempus Web (Pre-test)
<b>Dimensión</b>	Defectos por proyecto
<b>Indicador</b>	Cantidad de defectos
<b>Responsable</b>	Bach. Carrasco Borda Aparicio
<b>Fecha</b>	27-08-2021
<b>PROCESO DE GESTIÓN DE PROYECTOS</b>	
Actividad 1: PGP-ACT-01: Planeación del Proyecto	
<b>Defecto:</b>	- No establece prerrequisitos para cada tarea - No establece flujos alternos
<b>Criterio actual</b>	Con defecto (2)
Actividad 2: PGP-ACT-02: Despliegue de Plan del Proyecto	
<b>Defecto:</b>	- No evalúa a los miembros del equipo con las aptitudes adecuadas.
<b>Criterio actual</b>	Con defecto (1)
Actividad 3: PGP-ACT-03: Evaluación y Monitoreo del Proyecto	
<b>Defecto:</b>	- No establece número de participantes
<b>Criterio actual</b>	Con defecto (1)

  
 Aparicio Carrasco

PROCESO DE IMPLEMENTACIÓN DE SOFTWARE	
Actividad 4: PIS-ACT-01: Análisis de Requerimientos de Software	
Defecto:	- No realiza la correcta documentación de tecnología
Criterio actual	Con defecto (1)
Actividad 5: PIS-ACT-02: Diseño y Arquitectura del Software	
Defecto:	- No existe control de documentación de estimación de tiempos.
Criterio actual	Con defecto (1)
Actividad 6: PIS-ACT-03: Construcción del Software	
Defecto:	- No se reúnen para compartir inquietudes
Criterio actual	Con defecto (1)
Actividad 7: PIS-ACT-04: Entrega del Software	
Defecto:	-
Criterio actual	sin defecto (0)

  
 Aparicio Comas

Proyecto 02: Sistema de Ventas (PRE-TEST)

<b>TÍTULO</b>	DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110
<b>Proyecto</b>	Proyecto 02: Sistema de Ventas (Pre-Test)
<b>Dimensión</b>	Defectos por proyecto
<b>Indicador</b>	Cantidad de defectos
<b>Responsable</b>	Bach. Carrasco Borda Aparicio
<b>Fecha</b>	30.08.2021
<b>PROCESO DE GESTIÓN DE PROYECTOS</b>	
Actividad 1: PGP-ACT-01: Planeación del Proyecto	
<b>Defecto:</b>	- No establece los prerrequisitos de las tareas - No determina cual será el flujo de desarrollo
<b>Criterio actual</b>	Con defecto (2)
Actividad 2: PGP-ACT-02: Despliegue de Plan del Proyecto	
<b>Defecto:</b>	- No establece retroalimentación con los miembros del equipo
<b>Criterio actual</b>	Con defecto (1)
Actividad 3: PGP-ACT-03: Evaluación y Monitoreo del Proyecto	
<b>Defecto:</b>	- No establece estimación de costo total contra el presupuesto del proyecto
<b>Criterio actual</b>	Con defecto (1)

  
 Aparicio Carrasco

PROCESO DE IMPLEMENTACIÓN DE SOFTWARE	
Actividad 4: PIS-ACT-01: Análisis de Requerimientos de Software	
Defecto:	- Documentación incorrecta de instalación y ubicación de los archivos del Py.
Criterio actual	Con defecto (1)
Actividad 5: PIS-ACT-02: Diseño y Arquitectura del Software	
Defecto:	- No existe consulta del documento de estimación de tiempos
Criterio actual	Con defecto (1)
Actividad 6: PIS-ACT-03: Construcción del Software	
Defecto:	- No establece identificación de factores externos del desarrollo
Criterio actual	Con defecto (1)
Actividad 7: PIS-ACT-04: Entrega del Software	
Defecto:	- Retraso en la entrega del producto final
Criterio actual	Con defecto (1)

  
 Apuricio Camasco

Proyecto 03: Sistema de Referencias (PRE-TEST)

<b>TÍTULO</b>	DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110
<b>Proyecto</b>	Proyecto N° 03: Sistema de Referencias (Pre-test)
<b>Dimensión</b>	Defectos por proyecto
<b>Indicador</b>	Cantidad de defectos
<b>Responsable</b>	Bach. Carrasco Borda Aparicio
<b>Fecha</b>	31.08.2021
<b>PROCESO DE GESTIÓN DE PROYECTOS</b>	
Actividad 1: PGP-ACT-01: Planeación del Proyecto	
<b>Defecto:</b>	- No determina cuál será el flujo de desarrollo para c/u de las tareas
<b>Criterio actual</b>	Con defecto (1)
Actividad 2: PGP-ACT-02: Despliegue de Plan del Proyecto	
<b>Defecto:</b>	- No existe documentación de capacitación de los miembros de los roles asignados
<b>Criterio actual</b>	Con defecto (1)
Actividad 3: PGP-ACT-03: Evaluación y Monitoreo del Proyecto	
<b>Defecto:</b>	- No detalla el registro de los miembros con sus roles y responsabilidades - No se reúne al equipo para deliberar estimaciones del costo total.
<b>Criterio actual</b>	Con defecto (2)

  
 Aparicio Carrasco

PROCESO DE IMPLEMENTACIÓN DE SOFTWARE	
Actividad 4: PIS-ACT-01: Análisis de Requerimientos de Software	
Defecto:	- No existe revisión de cambios de la documentación de tecnológica
Criterio actual	Con defecto (1)
Actividad 5: PIS-ACT-02: Diseño y Arquitectura del Software	
Defecto:	- No detalla causas del retraso con respecto al progreso del proyecto.
Criterio actual	Con defecto (1)
Actividad 6: PIS-ACT-03: Construcción del Software	
Defecto:	- No establece identificación de factores externos que puedan afectar.
Criterio actual	Con defecto (1)
Actividad 7: PIS-ACT-04: Entrega del Software	
Defecto:	- Retraso en la entrega del producto final
Criterio actual	Con defecto (1)

  
 Apuricio Camarisco

Proyecto 04: Sistema de Mesa de Ayuda - Soporte Técnico (POST-TEST)

<b>TÍTULO</b>	<b>DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110</b>	
<b>Proyecto</b>	Proyecto 04: Sistema de Mesa de Ayuda (Post-test)	
<b>Dimensión</b>	Defectos por proyecto	
<b>Indicador</b>	Cantidad de defectos	
<b>Responsable</b>	Bach. Carrasco Borda Aparicio	
<b>Fecha</b>	29-11-2021	
<b>PROCESO DE GESTIÓN DE PROYECTOS</b>		
Actividad 1: PGP-ACT-01: Planeación del Proyecto		
<b>Defecto:</b>	-	
<b>Criterio actual</b>	Sin defecto (0)	
Actividad 2: PGP-ACT-02: Despliegue de Plan del Proyecto		
<b>Defecto:</b>	-	
<b>Criterio actual</b>	Sin defecto (0)	
Actividad 3: PGP-ACT-03: Evaluación y Monitoreo del Proyecto		
<b>Defecto:</b>	-	
<b>Criterio actual</b>	Sin defecto (0)	

  
 Aparicio Carrasco

PROCESO DE IMPLEMENTACIÓN DE SOFTWARE	
Actividad 4: PIS-ACT-01: Análisis de Requerimientos de Software	
Defecto:	-
Criterio actual	Sin defecto (0)
Actividad 5: PIS-ACT-02: Diseño y Arquitectura del Software	
Defecto:	-
Criterio actual	Sin defecto (0)
Actividad 6: PIS-ACT-03: Construcción del Software	
Defecto:	-
Criterio actual	Sin defecto (0)
Actividad 7: PIS-ACT-04: Entrega del Software	
Defecto:	-
Criterio actual	Sin defecto (0)

  
 Apuricio Cuzasco



Proyecto 05: Sistema Web de Recursos Humanos (POST-TEST)

<b>TÍTULO</b>	DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110
<b>Proyecto</b>	Proyecto 05: Sistema Web de RR. HH (Post-Test)
<b>Dimensión</b>	Defectos por proyecto
<b>Indicador</b>	Cantidad de defectos
<b>Responsable</b>	Bach. Carrasco Borda Aparicio
<b>Fecha</b>	30-11-2021
<b>PROCESO DE GESTIÓN DE PROYECTOS</b>	
Actividad 1: PGP-ACT-01: Planeación del Proyecto	
Defecto:	-
Criterio actual	Sin defecto (0)
Actividad 2: PGP-ACT-02: Despliegue de Plan del Proyecto	
Defecto:	- S.
Criterio actual	- Sin defecto (0)
Actividad 3: PGP-ACT-03: Evaluación y Monitoreo del Proyecto	
Defecto:	- Retraso en la estimación de tiempo y costo
Criterio actual	Con defecto (1)

  
 Aparicio Carrasco

PROCESO DE IMPLEMENTACIÓN DE SOFTWARE	
Actividad 4: PIS-ACT-01: Análisis de Requerimientos de Software	
Defecto:	-
Criterio actual	Sin defecto (0)
Actividad 5: PIS-ACT-02: Diseño y Arquitectura del Software	
Defecto:	-
Criterio actual	Sin defecto (0)
Actividad 6: PIS-ACT-03: Construcción del Software	
Defecto:	-
Criterio actual	Sin defecto (0)
Actividad 7: PIS-ACT-04: Entrega del Software	
Defecto:	-
Criterio actual	Sin defecto (0)

  
 Apuricio Carrasco

Proyecto 06: Sistema de Mesa de partes digital (POST-TEST)

<b>TÍTULO</b>	DESARROLLO DE UN MODELO DE PROCESOS AD HOC PARA EL DESARROLLO DE SOFTWARE POR LICENCIA PARA UNA MYPE DE SERVICIOS DE TI BASADO EN ISO/IEC 29110	
<b>Proyecto</b>	Proyecto 06: Sistema de Mesa de partes digital (Post-Test)	
<b>Dimensión</b>	Defectos por proyecto	
<b>Indicador</b>	Cantidad de defectos	
<b>Responsable</b>	Bach. Carrasco Borda Aparicio	
<b>Fecha</b>	01-12-2021	
<b>PROCESO DE GESTIÓN DE PROYECTOS</b>		
Actividad 1: PGP-ACT-01: Planeación del Proyecto		
<b>Defecto:</b>	-	
<b>Criterio actual</b>	Sin defecto (0)	
Actividad 2: PGP-ACT-02: Despliegue de Plan del Proyecto		
<b>Defecto:</b>	-	
<b>Criterio actual</b>	Sin defecto (0)	
Actividad 3: PGP-ACT-03: Evaluación y Monitoreo del Proyecto		
<b>Defecto:</b>	-	
<b>Criterio actual</b>	Sin defecto (0)	

  
 Aparicio Carrasco

PROCESO DE IMPLEMENTACIÓN DE SOFTWARE	
Actividad 4: PIS-ACT-01: Análisis de Requerimientos de Software	
Defecto:	-
Criterio actual	Sin defecto (0)
Actividad 5: PIS-ACT-02: Diseño y Arquitectura del Software	
Defecto:	-
Criterio actual	Sin defecto (0)
Actividad 6: PIS-ACT-03: Construcción del Software	
Defecto:	- Retraso en tiempo por no haber considerado estimación correcta de tiempo (ferrados)
Criterio actual	con defecto (1)
Actividad 7: PIS-ACT-04: Entrega del Software	
Defecto:	-
Criterio actual	Sin defecto (0)

  
 Apellido Canales

Anexo 10. Proyectos evaluados Pre Test

Proyecto 01: Sistema de Tempus Web - Control Tiempo (PRE-TEST)



## Proyecto 02: Sistema de Ventas (PRE-TEST)

The screenshot displays a web browser window with the URL `beetech.pe/ergo/projects/1/task`. The browser's address bar and tabs are visible at the top, including links for Gmail, WHM, Carva, Facebook, TikTok, WhatsApp, YouTube, zoom, Punto Pe, Computrabajo, Envato, Webmail, Udemy, and Magnum. The application interface features a dark blue sidebar on the left with the 'Melamuebles' logo and a menu with items: Panel de Control, Gestión de Entidades, Compras, Ventas, Gestionar Almacén, Contabilidad, Talento Humano, **Proyecto** (highlighted), Proyectos, Tareas, Hoja de tiempo, Bug, Calendario de tareas, Tracker, and Configuración. The main content area has a header with navigation buttons for 'Melamuebles', 'Ventas', 'Clientes', and 'Productos', along with a '+ Acceso' button. Below the header, the title 'Sistema Integrado De Inventario' s Tareas' is displayed. The task management section is organized into four columns: 'To Do', 'In Progress', 'Review', and 'Done'. The 'To Do' column contains a task card for 'Estructuración de BD' with a 'Critico' priority, dated '17 Aug 2022', and assigned to a user. The 'In Progress' column is currently empty. The 'Review' column contains two task cards: 'Diagrama de casos de Uso' (priority 'Critico', dated '17 Aug 2022') and 'Resporte de Entregas' (priority 'Medio', dated '17 Aug 2022'), both assigned to a user.

Proyecto 03: Sistema de Referencias (PRE-TEST)

<span>Archivo</span> <span>Referencias</span> <span>Reprogramaciones</span> <span>Orden Atencion</span> <span>Pasajes</span> <span>Reportes</span> <span>Administrativo</span> <span>Salir</span>	
Centro Origen: <input type="text" value="512"/> <input type="text" value="CAP II COTABAMBAS"/>	Centro Destino: <input type="text" value=""/> <input type="text" value=""/>
Servicio Origen: <input type="text" value="[SELECCIONAR]"/>	Servicio Destino: <input type="text" value="[SELECCIONAR]"/>
Actividad Origen: <input type="text" value="[SELECCIONAR]"/>	Actividad Destino: <input type="text" value="[SELECCIONAR]"/>
Sub Activ Origen: <input type="text" value="[SELECCIONAR]"/>	Sub Activ Destino: <input type="text" value="[SELECCIONAR]"/>
Número de acto médico: <input type="text" value=""/> Nro.Historia: <input type="text" value=""/>	
Fecha Optima de Cita: <input type="text" value=""/> (Formato dd/mm/yyyy)	
<b>Identificación del Paciente</b>	
Autogenerado <input type="text" value=""/> <a href="#">Validar</a> <a href="#">(Historia)</a>	Condición del asegurado <input type="text" value=""/>
Autogenerado del titular <input type="text" value=""/>	Condición del titular <input type="text" value=""/>
Tipo documento <input type="text" value=""/>	Num. doc: <input type="text" value=""/> <b>Vigencia</b> <input type="text" value=""/> (dd/mm/yyyy)
Apellido Paterno <input type="text" value=""/>	Apellido Materno <input type="text" value=""/> Nombres <input type="text" value=""/>
Sexo: <input type="text" value=""/> Edad: <input type="text" value=""/>	Telefonos <input type="text" value=""/> Mail <input type="text" value=""/>
Fecha de Nac: <input type="text" value=""/> (Formato dd/mm/yyyy)	Ubigeo Nacimiento <input type="text" value=""/> <input type="text" value=""/>
Domicilio <input type="text" value=""/>	Ubigeo Domicilio <input type="text" value=""/> <input type="text" value=""/>
<b>Resumen de la Historia Clínica</b>	
Resumen - Historia <input type="text" value=""/>	
Motivo <input type="text" value="[SELECCIONAR]"/>	
Otros (Detalle): <input type="text" value=""/>	
Diagnóstico 1 <input type="text" value=""/> <input type="text" value=""/>	Presuntivo <input type="radio"/> Definitivo <input type="radio"/>
Diagnóstico 2 <input type="text" value=""/> <input type="text" value=""/>	Presuntivo <input type="radio"/> Definitivo <input type="radio"/>
Comentarios adicionales <input type="text" value=""/>	
<b>Médico responsable</b>	
C.M.P.: <input type="text" value=""/> <input type="text" value=""/> <input type="text" value=""/>	
<b>Adjunto</b>	
Adjuntar: <input type="text" value=""/>	
<input type="button" value="Ingresar"/> <input type="button" value="Cancelar"/>	

## Anexo 11. Proyectos evaluados Post Test

### Proyecto 04: Sistema de Mesa de Ayuda - Soporte Técnico (POST-TEST)

Título	Status	Type	Categoría	Solicitante - Solicitante	Ubicación	Fuente de solicitud	Fecha de apertura	Prioridad	Asignado a - Técnico	Fecha de resolución	Estadísticas - Tiempo de resolución	Descripción	Última modificación
Creación de email institucional	Cerrado	Solicitud	SOPORTE > SOFTWARE > DOMINIO Y CORREO > CREACION DE USUARIO	Tarrillo Herrera Mack Johan	RA JAEN	E-Mail	08-08-2022 14:37	Media	Rodriguez Segura Alonso Humberto	24-08-2022 08:21	4 días 5 horas 44 minutos	Estimados Buenas tardes, a través del presente email se remite el formato para creación de email institucional para el médico Leyder Villanueva Alejandria, se adjunta formato correspondiente, saludos cordiales. Ing. Mack Johan Tarrillo Herrera Coord (...)	25-08-2022 22:28
Memorando Circular N° 1313-GCTIC-ESSALUD-2022 // Necesidad de renovación tecnológica	Cerrado	Solicitud	SOPORTE > SOFTWARE > DOMINIO Y CORREO > CONFIGURACION	Administración Correo	GCTIC-P6-EL > GPROD-P6-EL > SGST-P6-EL	E-Mail	17-08-2022 12:22	Media	Rodriguez Segura Alonso Humberto	24-08-2022 15:22	2 días 0 horas 0 minutos	Buenos días, Se remite la siguiente comunicación para conocimiento, atención, cumplimiento y fines correspondientes. Memorando Circular N° 1313-GCTIC-ESSALUD-2022 Asunto: Necesidad de renovación tecnológica Atte., Administración del Servicio de (...)	25-08-2022 22:28
Reseteo de contraseña de email institucional	Cerrado	Solicitud	SOPORTE > SOFTWARE > DOMINIO Y CORREO > RESETEO DE CLAVE	Tarrillo Herrera Mack Johan	RA JAEN	E-Mail	12-08-2022 17:25	Media	Muñoz Rojas Eider Arnol	23-08-2022 16:35	2 días 14 horas 35 minutos	Estimados Buenas tardes, a través del presente email solicito reseteo de contraseña para el email institucional a href="mailto:farmacia.sanignacio@essalud.gob.pe,"farmacia.sanignacio@essalud.gob.pe, saludos cordiales. --	24-08-2022 22:29
Reiterar creación de email institucional	Cerrado	Solicitud	SOPORTE > SOFTWARE > DOMINIO Y CORREO > CREACION DE USUARIO	Tarrillo Herrera Mack Johan	RA JAEN	E-Mail	16-08-2022 09:12	Media	Muñoz Rojas Eider Arnol	23-08-2022 16:47	2 días 4 horas 35 minutos	Estimados Buenas tardes, a través del presente email se remite el formato para reiterar la creación de email institucional para el médico Leyder Villanueva Alejandria, se adjunta formato correspondiente, saludos cordiales. --	24-08-2022 22:29
Reseteo de contraseña de email institucional	Cerrado	Solicitud	SOPORTE > SOFTWARE > DOMINIO Y CORREO > RESETEO DE CLAVE	Tarrillo Herrera Mack Johan	RA JAEN	E-Mail	16-08-2022 11:59	Media	Muñoz Rojas Eider Arnol	23-08-2022 16:38	2 días 1 hora 39 minutos	Estimados Buenos días, a través del presente email solicito el reseteo de contraseña para el email institucional a href="mailto:farmacia.sanignacio@essalud.gob.pe,"farmacia.sanignacio@essalud.gob.pe, saludos cordiales. --	24-08-2022 22:29
Solicito reseteo de contraseña de correo	Cerrado	Solicitud	SOPORTE > SOFTWARE > DOMINIO Y CORREO > RESETEO DE CLAVE	Tarrillo Herrera Mack Johan	RA JAEN	E-Mail	23-08-2022 12:35	Media	Muñoz Rojas Eider Arnol	23-08-2022 16:37	4 horas 2 minutos	Estimados Buenos días a través del presente email solicito reseteo de contraseña del email a href="mailto:obstetricia.sanignacio@essalud.gob.pe,"obstetricia.sanignacio@essalud.gob.pe, saludos cordiales. De: "Alexander Leon" alexander.leon@essalud.gob.pe (...)	24-08-2022 22:29
Acceso a vpn	Cerrado	Solicitud	TELECOMUNICACIONES > SOFTWARE > VPN > CREACION DE USUARIO	Tarrillo Herrera Mack Johan	RA JAEN	E-Mail	05-08-2022 11:25	Media	Navarro Bonifacio Luis Eduardo	18-08-2022 11:12	3 horas 49 minutos	Estimados Buenos días, a través del presente email solicito la creación de usuario VPN para la Médico General Jessica Elizabeth Calderon Anaya para realizar trabajo remoto por motivos de salud, saludos cordiales.	19-08-2022 22:29
Reitero la creación de usuario VPN	Cerrado	Solicitud	TELECOMUNICACIONES > SOFTWARE > VPN > CREACION DE USUARIO	Tarrillo Herrera Mack Johan	RA JAEN	E-Mail	16-08-2022 09:12	Media	Malca Tafur Santos	16-08-2022 17:26	7 horas 47 minutos	Estimados Buenos días, a través del presente email reitero la creación de usuario VPN para la Médico General Jessica Elizabeth Calderon Anaya para realizar trabajo remoto por motivos de salud, saludos cordiales. --	18-08-2022 22:29



# Proyecto 05: Sistema Web de Recursos Humanos (POST-TEST)

**SISTEMA RRHH** ≡ USUARIO: admin





**ADMINISTRADOR** Online

- Usuarios <
- Planilla <
- Contrato <
- Procesar Solicitud <
- Derecho De Habiente <
- Trabajadores <
- Familiar <
- Cargo <
- Seguro <
- Tipo Contrato <
- Concepto <



SISTEMA DE RECURSOS HUMANOS - SEGURO SOCIAL DE SALUD 2022 -- Versión 1.0 fecha actual: 26/08/2022

## Proyecto 06: Sistema de Mesa de partes digital (POST-TEST)

-  Mis Trámites
-  Actualizar Datos
-  Cambiar Contraseña
-  Cerrar Sesión

GOB.PE > EsSalud > MPD

**Bandeja de Trámites**

Número de Solicitud / Expediente

Tipo de Expediente

Estado




Fecha Inicio

Fecha Fin

dd/mm/aaaa

dd/mm/aaaa

Seguimiento de NIT
Buscar
Limpiar
+ Nueva Solicitud

Acciones	SOLICITUD	FECHA DE SOLICITUD	TIPO EXPEDIENTE	N° EXPEDIENTE	ASUNTO	COMENTARIOS DEL CIUDADANO	OBSERVACIONES DE ESSALUD	ESTADO	ADJUNTOS
	5-78959-2021	19/10/2021 12:00	NO TUPA	0179-2021-NIT-0037792	Solicita Accion Administ...	Solicito se expida y se m...	Se aprueba documentac...	REGISTRADO	1
	5-28175-2021	26/04/2021 14:40	NO TUPA	0179-2021-NIT-0013022	Solicita Accion Administ...	SOLICITO INCORPORACI...	Se aprueba documentac...	REGISTRADO	1
	5-20835-2021	26/03/2021 08:00	NO TUPA	0179-2021-NIT-0009165	Solicita Accion Administ...	Solicito el cumplimiento...	Se aprueba documentac...	REGISTRADO	1

Mostrando registros del 1 al 3 de un total de 3 registros

←
1
→

Anexo 12. Diagrama de Gantt - Proyecto 04: Sistema de Mesa de Ayuda - Soporte Técnico (POST-TEST)

Proyecto 04: Sistema de Mesa de Ayuda - Soporte Técnico

TÍTULO DEL PROYECTO		PROYECTO 04: SISTEMA DE MESA DE AYUDA - SOPORTE TÉCNICO																		
RESPONSABLE DEL PROYECTO		APARICIO CARRASCO BORDA																		
Nº	TÍTULO DE LA TAREA	RESPONSABLE DE LA TAREA	FECHA DE INICIO	FECHA DE ENTREGA	DURACIÓN	% COMPLETADO DE LA TAREA	Cronograma de desarrollo de las 7 actividades del modelo propuesto													
							SEMANAS													
							1	2	3	4	5	6	7	8	9	10	11	12	13	14
<b>1 PGP-ACT-01: Planeación del Proyecto</b>																				
PROCESO DE GESTIÓN DE PROYECTOS	1.1	Identificar las tareas del proyecto	Aparicio Carrasco Borda	01/09/21	08/09/21	7	100 %	X												
	1.2	Establecer prerequisites y requisitos del proyecto	Aparicio Carrasco Borda	01/09/21	08/09/21	7	100 %	X												
	1.3	Definir el tiempo aproximado de desarrollo del proyecto	Aparicio Carrasco Borda	01/09/21	08/09/21	7	100 %	X												
	<b>2 PGP-ACT-02: Despliegue de Plan del Proyecto</b>																			
	2.1	Registrar el equipo de trabajo y roles	Aparicio Carrasco Borda	09/09/21	16/09/21	7	100 %		X											
	2.2	Evaluar los miembros del equipo	Aparicio Carrasco Borda	09/09/21	16/09/21	7	100 %		X											
	2.3	Aceptación de los roles asignados	Aparicio Carrasco Borda	09/09/21	16/09/21	7	100 %		X											

<b>3 PGP-ACT-03: Evaluación y Monitoreo del Proyecto</b>																	
3.1	Recobrar la relación de las tareas establecidas para el proyecto	Aparicio Carrasco Borda	17/09/21	24/09/21	7	100 %											X
3.2	Reunión con los miembros del equipo para contrastar la estimación de tiempo	Aparicio Carrasco Borda	17/09/21	24/09/21	7	100 %											X
3.3	Registro de las estimaciones de costo, tiempo y esfuerzo para futura consulta en caso de inconvenientes	Aparicio Carrasco Borda	17/09/21	24/09/21	7	100 %											X
<b>4 PIS-ACT-01: Análisis de Requerimientos de Software</b>																	
4.1	Determinar el flujo desarrollo del software	Aparicio Carrasco Borda	25/09/21	02/10/21	7	100 %											X
4.2	Acordar el nivel de importancia de la tarea	Aparicio Carrasco Borda	25/09/21	02/10/21	7	100 %											X
4.3	Definir el tiempo de desarrollo teniendo en cuenta la dificultad	Aparicio Carrasco Borda	03/10/21	10/10/21	7	100 %											X
<b>5 PIS-ACT-02: Diseño y Arquitectura del Software</b>																	
5.1	Identificar cuáles son las tecnologías que se utilizarán para el desarrollo del producto	Aparicio Carrasco Borda	11/10/21	18/10/21	7	100 %											X
5.2	Revisar la documentación para la correcta instalación y ubicación	Aparicio Carrasco Borda	19/10/21	26/10/21	7	100 %											X

PROCESO DE IMPLEMENTACIÓN DE SOFTWARE



Proyecto 05: Sistema Web de Recursos Humanos

TÍTULO DEL PROYECTO PROYECTO 05: SISTEMA WEB DE RECURSOS HUMANOS

RESPONSABLE DEL PROYECTO APARICIO CARRASCO BORDA

N°	TÍTULO DE LA TAREA	RESPONSABLE DE LA TAREA	FECHA DE INICIO	FECHA DE ENTREGA	DURACIÓN	% COMPLETADO DE LA TAREA	Cronograma de desarrollo de las 7 actividades del modelo propuesto														
							SEMANAS														
							1	2	3	4	5	6	7	8	9	10	11	12	13	14	
<b>1 PGP-ACT-01: Planeación del Proyecto</b>																					
1.1	Identificar las tareas del proyecto	Aparicio Carrasco Borda	01/09/21	08/09/21	7	100 %	X														
1.2	Establecer prerrequisitos y requisitos del proyecto	Aparicio Carrasco Borda	01/09/21	08/09/21	7	100 %	X														
1.3	Definir el tiempo aproximado de desarrollo del proyecto	Aparicio Carrasco Borda	01/09/21	08/09/21	7	100 %	X														
<b>2 PGP-ACT-02: Despliegue de Plan del Proyecto</b>																					
2.1	Registrar el equipo de trabajo y roles	Aparicio Carrasco Borda	09/09/21	16/09/21	7	100 %		X													
2.2	Evaluar los miembros del equipo	Aparicio Carrasco Borda	09/09/21	16/09/21	7	100 %		X													
2.3	Aceptación de los roles asignados	Aparicio Carrasco Borda	09/09/21	16/09/21	7	100 %		X													
<b>3 PGP-ACT-03: Evaluación y Monitoreo del Proyecto</b>																					

PROCESO DE GESTIÓN DE PROYECTOS



6 PIS-ACT-03: Construcción del Software																					
6.1	Realizar reuniones con todo el equipo de trabajo	Aparicio Carrasco Borda	05/10/21	06/12/21	61	100 %						X	X	X	X	X	X	X	X		
6.2	Consultar el documento de estimación de tiempos para ver si el progreso está dentro de los estimado	Aparicio Carrasco Borda	27/10/21	06/12/21	39										X	X	X	X	X		
6.3	Actualizar el documento de tareas del proyecto	Aparicio Carrasco Borda	05/10/21	06/12/21	61	100 %						X	X	X	X	X	X	X	X		
7 PIS-ACT-04: Entrega del Software																					
7.1	Identificación de la versión estable del producto final	Aparicio Carrasco Borda	07/12/21	10/12/21	3	100 %														X	
7.2	Revisión del manual para el uso del producto final	Aparicio Carrasco Borda	07/12/21	10/12/21	3	100 %															X
7.3	Entrega del Proyecto	Aparicio Carrasco Borda	10/12/21	12/12/21	2	100 %															X

Anexo 14. Diagrama de Gantt - Proyecto 06: Sistema de Mesa de partes digital (POST-TEST)

**Proyecto 06: Sistema de Mesa de partes digital**



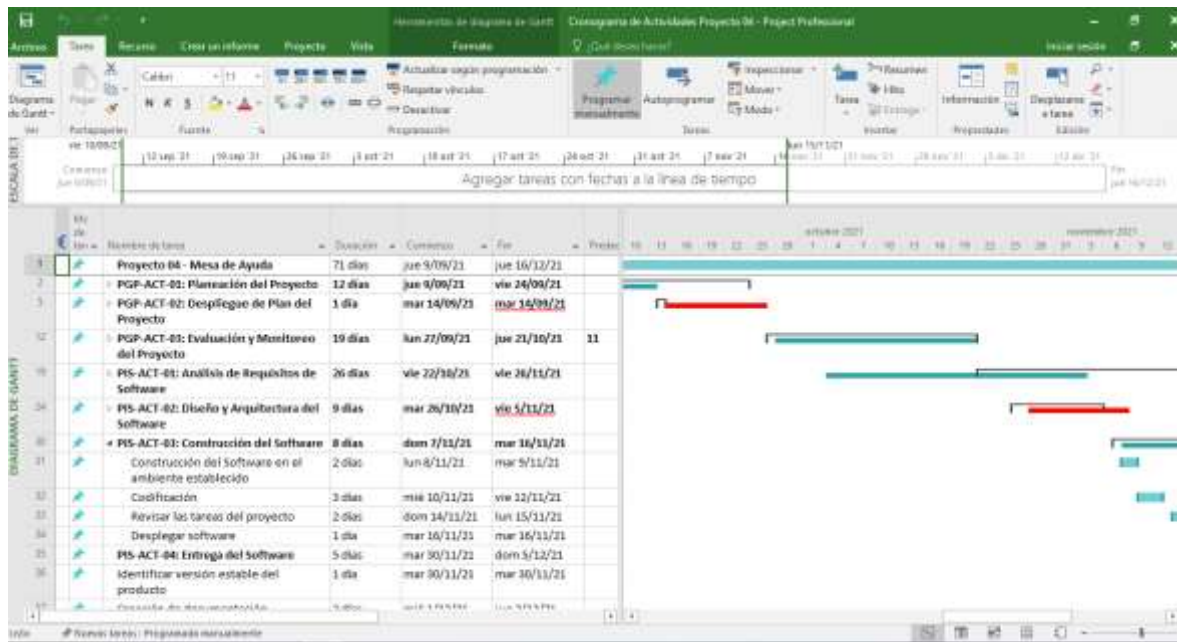




6 PIS-ACT-03: Construcción del Software																					
6.1	Realizar reuniones con todo el equipo de trabajo	Aparicio Carrasco Borda	05/10/21	06/12/21	61	100 %						X	X	X	X	X	X	X	X		
6.2	Consultar el documento de estimación de tiempos para ver si el progreso está dentro de los estimado	Aparicio Carrasco Borda	27/10/21	06/12/21	39										X	X	X	X	X		
6.3	Actualizar el documento de tareas del proyecto	Aparicio Carrasco Borda	05/10/21	06/12/21	61	100 %						X	X	X	X	X	X	X	X		
7 PIS-ACT-04: Entrega del Software																					
7.1	Identificación de la versión estable del producto final	Aparicio Carrasco Borda	07/12/21	10/12/21	3	100 %														X	
7.2	Revisión del manual para el uso del producto final	Aparicio Carrasco Borda	07/12/21	10/12/21	3	100 %															X
7.3	Entrega del Proyecto	Aparicio Carrasco Borda	10/12/21	12/12/21	2	100 %															X

Anexo 15. Evidencias gráficas de prueba piloto de los procesos desarrollados del modelo.

## Planeación del proyecto



## Acta del proyecto

### Datos

Empresa / Organización	Nedley Support S.A.C
Proyecto	Proyecto 04: Mesa de Ayuda – Soporte Técnico
Fecha de preparación	12 de setiembre del 2021
Cliente	Seguro Social de Salud
Patrocinador principal	Aparicio Carrasco Borda
Gerente de proyecto	Aparicio Carrasco Borda

### Patrocinador / Patrocinadores

Nombre	Cargo	Departamento / División	Rama ejecutiva (Vicepresidencia)
Aparicio Carrasco	Jefe Proyecto	Soporte Técnico	Red Asistencial

### Propósito y justificación del proyecto

Desarrollar aplicación que gestiona recursos tecnológicos y humanos, para prestar servicios con la posibilidad de gestionar y solucionar todas las posibles incidencias de manera integral, junto con la atención de requerimientos relacionados con las Tecnologías de la Información y la Comunicación (TIC).

DOCUMENTO	PROPÓSITO/ESTRUCTURA
PLAN DEL PROYECTO	<p>Presenta cómo serán ejecutados los procesos y actividades del proyecto para asegurar su conclusión exitosa, así como la calidad de los productos entregables. Puede incluir los siguientes elementos y características:</p> <ul style="list-style-type: none"> <li>↓ Descripción de producto</li> <li>↓ Propósito</li> <li>↓ Requisitos generales del Cliente</li> <li>↓ Alcance descripción respecto de lo que está incluido y de lo que no está incluido</li> <li>↓ Objetivos del proyecto</li> <li>↓ Entregables y/o lista de productos a ser entregados al Cliente</li> <li>↓ Tareas, incluyendo verificación, validación y revisiones con el Cliente y Equipo de Trabajo que permitan asegurar la calidad de los productos de trabajo. Las Tareas pueden ser representadas como una Estructura de Descomposición del Trabajo (EDT).</li> <li>↓ Duración estimada de las Tareas</li> <li>↓ Recursos (humanos, materiales, estándares, equipos y herramientas), incluyendo la capacitación necesaria.</li> <li>↓ Incluye la identificación y programación de los Recursos.</li> <li>↓ Composición del Equipo de Trabajo.</li> <li>↓ Calendario de las Tareas del proyecto, indicando la fecha de inicio y fecha de finalización previstas para cada Tarea, y las relaciones y dependencias entre ellas.</li> <li>↓ Esfuerzo y el costo estimado.</li> <li>↓ Identificación de los riesgos del proyecto.</li> <li>↓ Estrategia para el control de versiones.</li> <li>↓ Herramientas de repositorio del producto o mecanismos identificados.</li> <li>↓ Localización y mecanismos de acceso para el repositorio especificado</li> <li>↓ Identificación y control de versiones definidos.</li> <li>↓ Respaldo y mecanismos de recuperación definidos</li> <li>↓ Mecanismos de almacenamiento, manipulación y entrega especificados (incluyendo archivo y recuperación).</li> <li>↓ Instrucciones de entrega</li> <li>↓ Elementos requeridos para la liberación del producto (por ejemplo, hardware, software, documentación, etc.)</li> <li>↓ Requisitos de entrega</li> <li>↓ Tareas a realizar en orden secuencial</li> <li>↓ Liberaciones aplicables identificadas</li> <li>↓ Identifica todos los Componente de software entregados con</li> <li>↓ información de la versión</li> <li>↓ Identifica cualquier procedimiento de copia de respaldo y recuperación necesarios.</li> </ul>

## Despliegue de Plan del proyecto

DOCUMENTO	PROPÓSITO/ESTRUCTURA
ACTA DE REUNIÓN	<p>Registro de los acuerdos en reuniones con el cliente y/o el Equipo de Trabajo. Se propone la siguiente estructura:</p> <ul style="list-style-type: none"> <li>↓ Propósito de la reunión</li> <li>↓ Asistentes</li> <li>↓ Fecha y lugar</li> <li>↓ Referencia a actas de reunión previas</li> <li>↓ Qué fue logrado</li> <li>↓ Identifica cuestiones planteadas</li> <li>↓ Cualquier asunto abierto</li> <li>↓ Acuerdos</li> <li>↓ Próxima reunión (en caso necesario)</li> </ul>
ACTA DE ACEPTACIÓN	<p>Documentación de aceptación por parte del cliente sobre los entregables del proyecto. La estructura propuesta del documento:</p> <ul style="list-style-type: none"> <li>↓ Registro de la recepción de la entrega</li> <li>↓ Identificación de la fecha de recepción</li> <li>↓ Identificación de los elementos entregados</li> <li>↓ Registro de la verificación de los criterios de aceptación definidos por parte del cliente</li> <li>↓ Identificación de cualquier asunto pendiente (en caso de ser aplicable)</li> </ul>
SOLICITUD DE CAMBIO	<p>Solicitud de una modificación para corregir un problema o incorporar una mejora en el software o en su documentación. Contiene la siguiente información:</p> <ul style="list-style-type: none"> <li>↓ Identifica el propósito del cambio</li> <li>↓ Estado de la solicitud</li> <li>↓ Información de contacto del solicitante</li> <li>↓ Sistema(s) impactado(s)</li> <li>↓ Impacto en la operación de sistemas existentes</li> <li>↓ Impacto en la documentación asociada</li> <li>↓ Críticidad de la solicitud y fecha en que se requiere</li> </ul>
RESULTADOS DE VERIFICACIÓN	<p>Documento de la ejecución de la verificación. Incluye el registro de:</p> <ul style="list-style-type: none"> <li>↓ Participantes</li> <li>↓ Fecha</li> <li>↓ Lugar</li> <li>↓ Duración</li> <li>↓ Lista de comprobación para la verificación</li> <li>↓ Elementos aprobados por la verificación</li> <li>↓ Elementos no aprobados por la verificación</li> <li>↓ Elementos pendientes de la verificación</li> <li>↓ Defectos identificados durante la verificación</li> </ul>
REGISTRO DE TRAZABILIDAD	<p>Documenta la relación entre los requisitos incluidos en la Especificación de Requisitos, los elementos del Diseño de software, los Componente de software, los Casos y los Procedimientos de Prueba. Incluye:</p> <ul style="list-style-type: none"> <li>↓ Especificación de los requisitos por rastrear</li> <li>↓ Proporciona el mapeo (hacia adelante y hacia atrás) de los requisitos a los elementos del Diseño de software, los Componente de software, los Casos de Prueba y los Procedimientos de Prueba.</li> </ul>

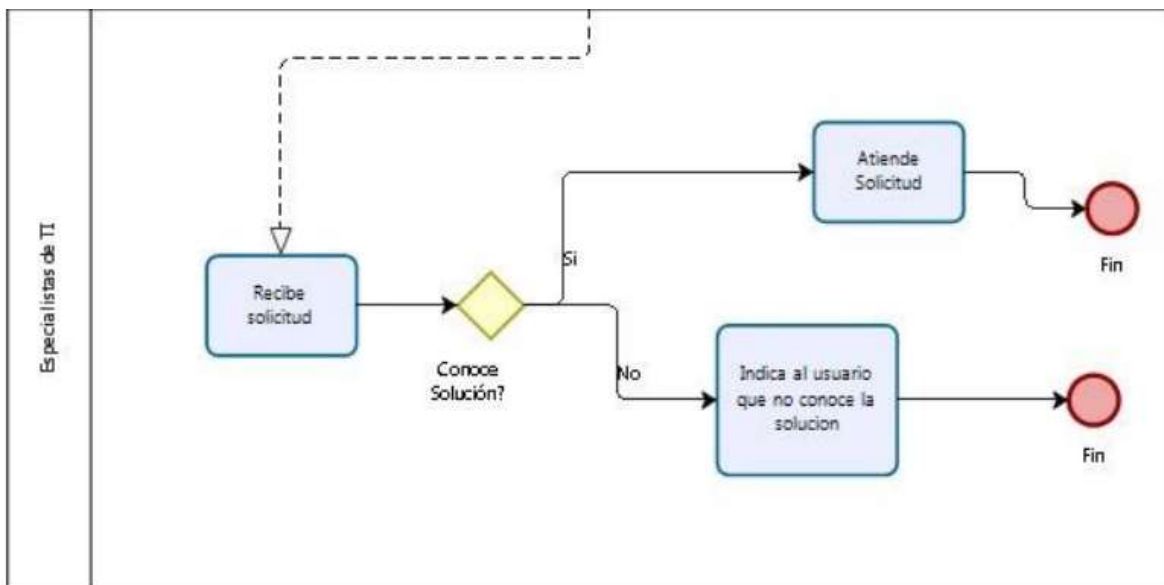
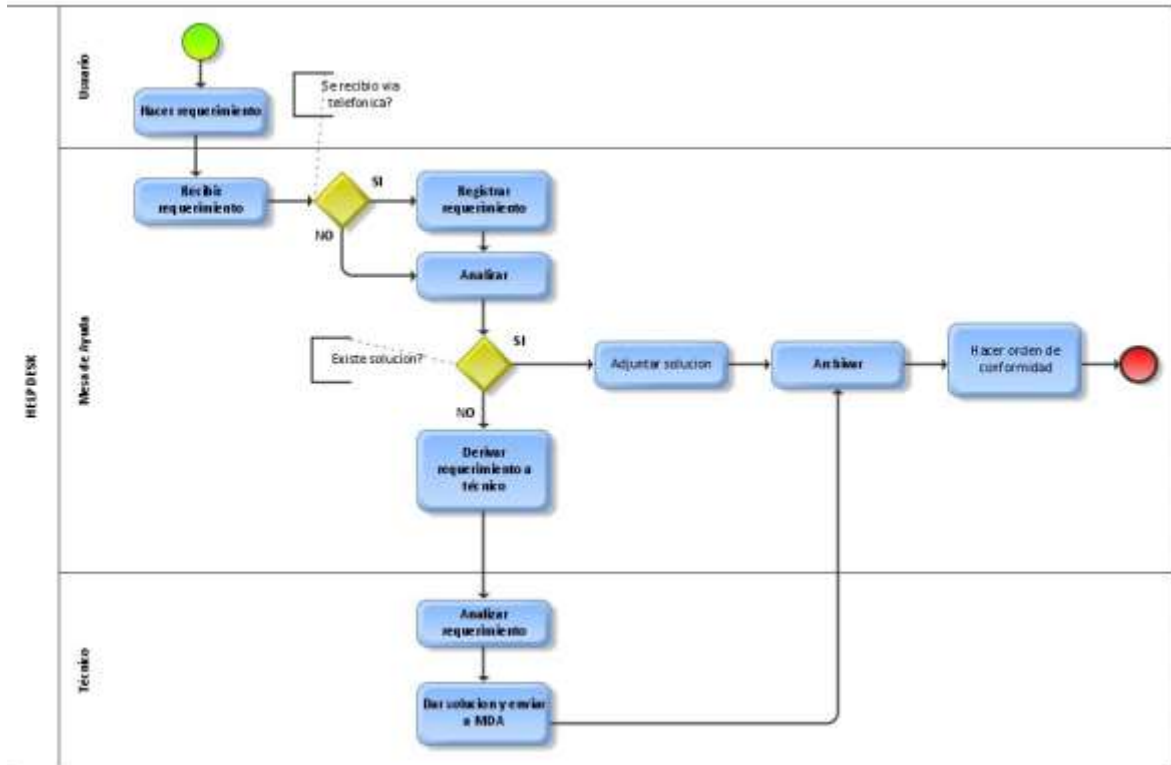
DOCUMENTO	PROPÓSITO/ESTRUCTURA
MANUAL DE USUARIO	<p>Describe la forma de uso del Software basado en la interfaz de usuario. Cuenta con las siguientes características:</p> <ul style="list-style-type: none"> <li>➤ Procedimientos del usuario para realizar tareas específicas utilizando el Software</li> <li>➤ Procedimientos de instalación y desinstalación</li> <li>➤ Breve descripción del uso previsto del Software (el concepto de operaciones)</li> <li>➤ Recursos provistos y requeridos</li> <li>➤ Entorno operacional requerido</li> <li>➤ Facilidad para reportar problemas y asistencia</li> <li>➤ Procedimientos para entrar y salir del Software</li> <li>➤ Relación y explicación de comandos del Software y de los mensajes del sistema hacia el usuario</li> <li>➤ Según corresponda a los riesgos identificados; este incluye advertencias, precauciones y notas con correcciones</li> <li>➤ Incluye los procedimientos para la solución de problemas y corrección de errores</li> <li>➤ Está escrito en términos comprensibles para los usuarios.</li> </ul> <p>Los estados aplicables son: preliminar, verificado e incorporado en la línea base.</p>
DOCUMENTACIÓN DE MANTENIMIENTO DE SOFTWARE	<p>Cuando se pone marcha un Sistema Informático es necesario que la documentación que explique cómo funciona y qué cometidos específicos tienen cada una de sus partes. Estructura planteada:</p> <ul style="list-style-type: none"> <li>➤ Descripción Funcional</li> <li>➤ Manual de Instalación</li> <li>➤ Manual de Introducción</li> <li>➤ Manual de Referencia</li> <li>➤ Guía de Operación o Guía del Operador</li> </ul>
DOCUMENTO DE PRUEBAS UNITARIAS	<p>Contiene la lista de pruebas unitarias ejecutadas por el programador. Incluye:</p> <ul style="list-style-type: none"> <li>➤ Id de la prueba</li> <li>➤ Nombre de la prueba</li> <li>➤ Descripción de la prueba</li> <li>➤ Condiciones de la prueba</li> <li>➤ Resultado esperado</li> <li>➤ Resultado obtenido</li> <li>➤ Fecha de ejecución</li> </ul>

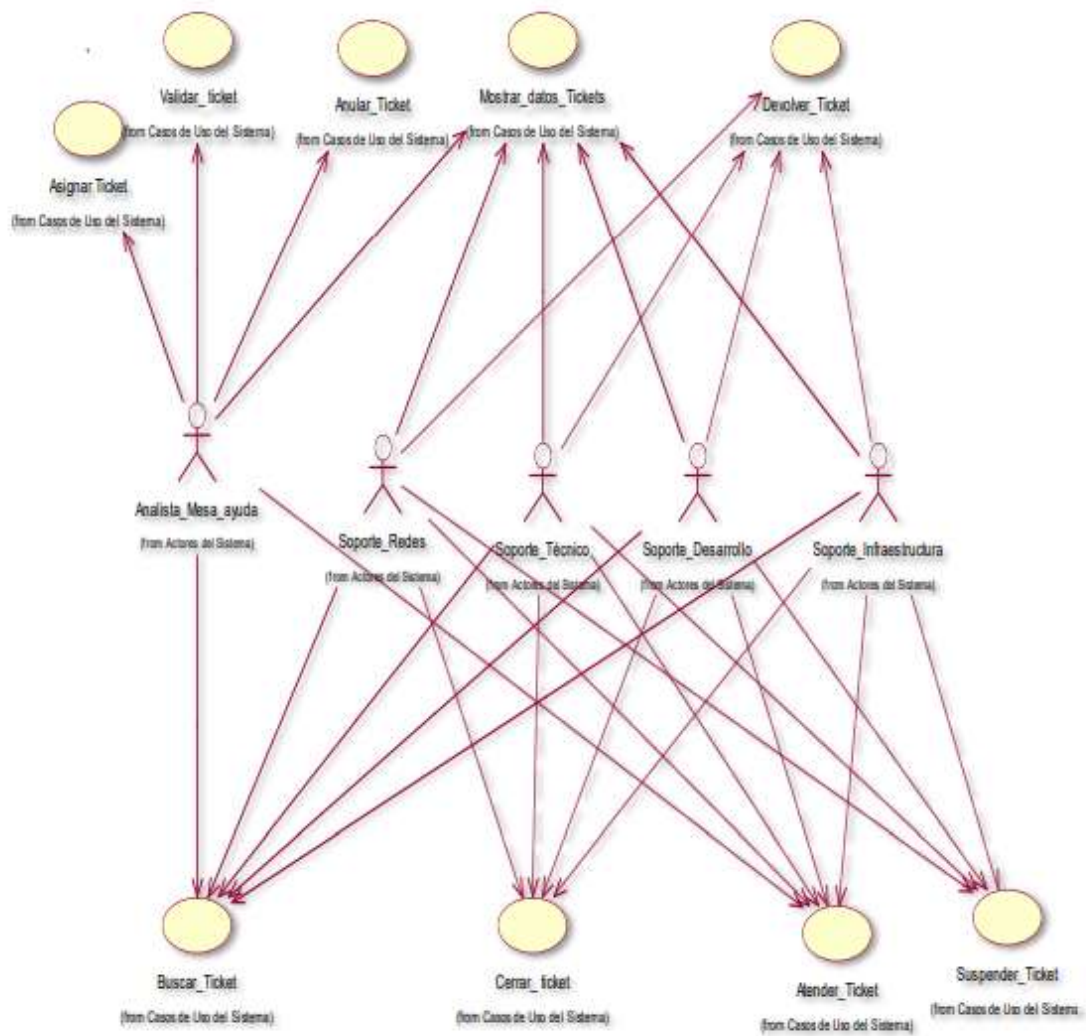
## Evaluación y Monitoreo del Proyecto

Estimación de costo, tiempo y esfuerzo por tarea					
Nº Tarea	Encargado	Costo Aprox	Fecha Inicio	Fecha fin	
Plan de Proyecto	Equipo de trabajo	1000.00	01/09/2021	08/09/2021	1
Acta de proyecto	Equipo de trabajo	150.00	09/09/2021	09/06/2021	1
Otras actividades de gestión de proyecto	Equipo de trabajo	500.00	10/09/2021	25/09/2021	1
Análisis de requisitos del sistema	Equipo de Trabajo	1200.00	26/09/2021	30/09/2021	1
Diseño y arquitectura	Equipo de trabajo	2200.00	01/10/2021	22/10/2021	1
Construcción de Software	Programador y su equipo	2700.00	23/10/2021	21/11/2021	1
Pruebas del sistema	Equipo de trabajo	500.00	22/11/2021	25/11/2021	1
Manual de usuario	Equipo de trabajo	200.00	26/11/2021	28/11/2021	1
Documentación y revisiones	Equipo de trabajo	1000.00	26/11/2021	05/12/2021	1
Capacitación al cliente	Equipo de trabajo	200.00	29/11/2021	04/12/2021	1
Entrega del producto	Equipo de trabajo	50.00	05/12/2021	05/12/2021	1
<b>Total (S/)</b>		9700.00		<b>Días totales</b>	71
		0			

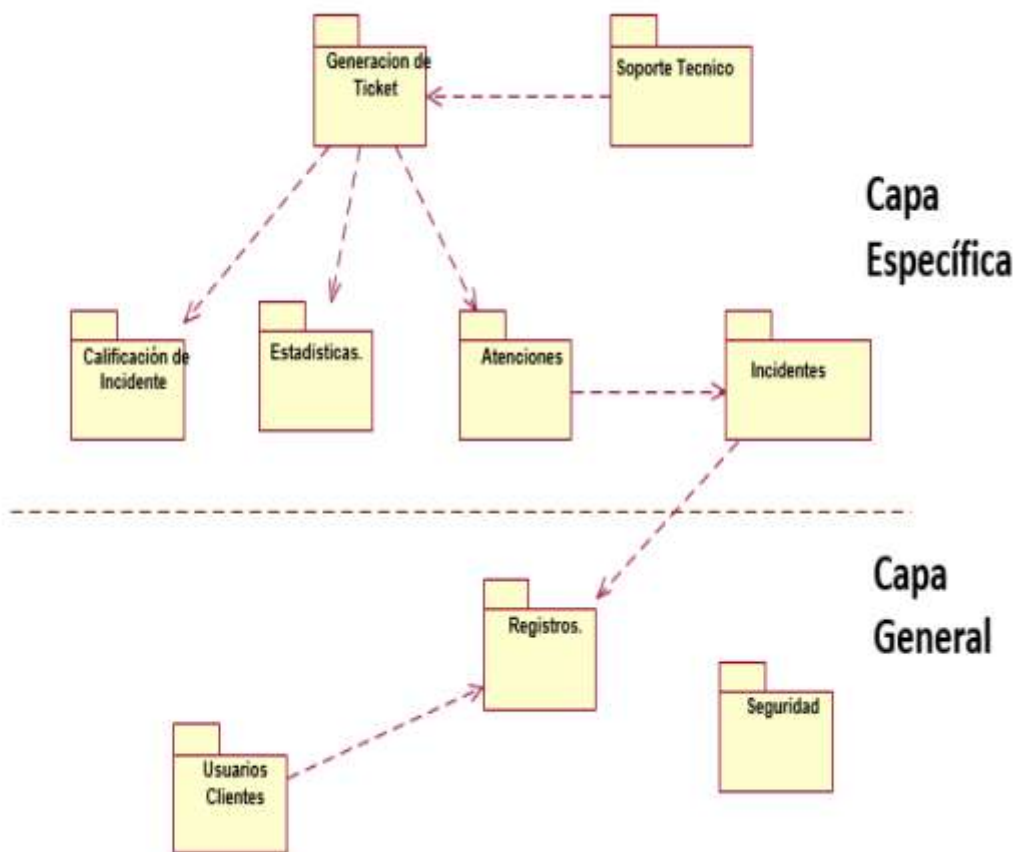


## Análisis de Requisitos de Software



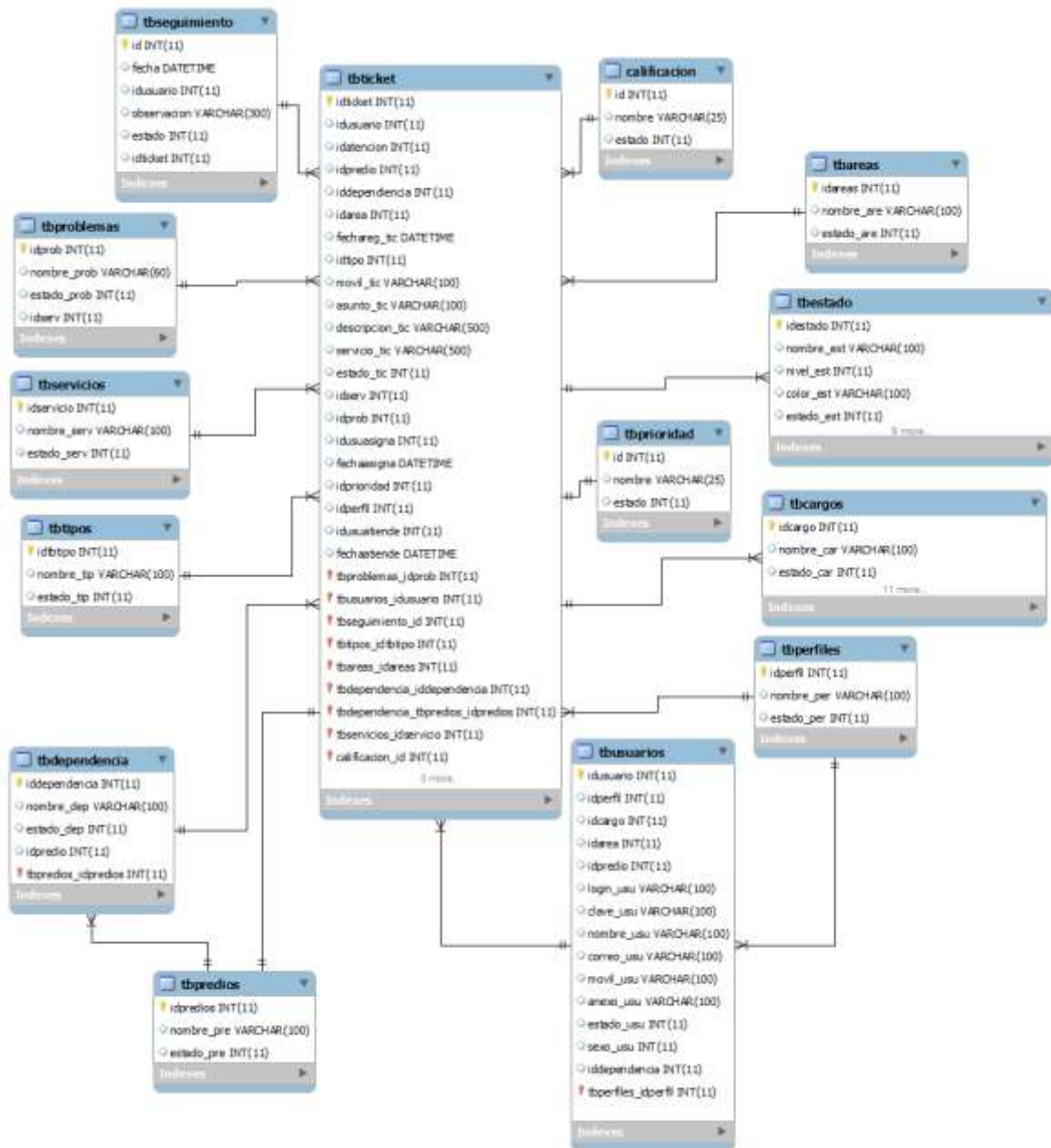


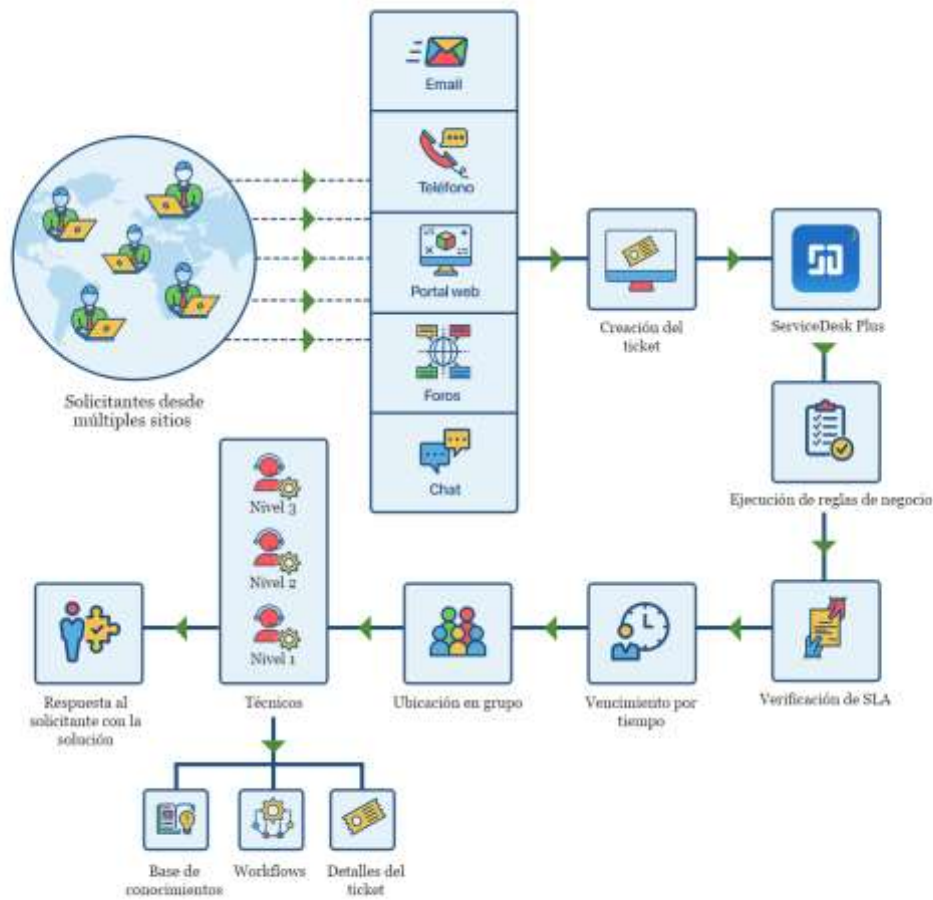
Diseño y Arquitectura del Software



```

1 CREATE TABLE `tbl_ticket`
2 (
3     rowid          integer AUTO_INCREMENT PRIMARY KEY,
4     entity         integer DEFAULT 1,
5     ref            varchar(128) NOT NULL,
6     track_id       varchar(128) NOT NULL,
7     fk_soc          integer DEFAULT 0,
8     fk_project     integer DEFAULT 0,
9     origin_email   varchar(128),
10    fk_user_create  integer,
11    fk_user_assign  integer,
12    subject         varchar(255),
13    message         text,
14    fk_statut       integer,
15    resolution      integer,
16    progress        varchar(100),
17    timing          varchar(20),
18    type_code       varchar(32),
19    category_code   varchar(32),
20    severity_code   varchar(32),
21    datec           datetime,
22    date_read       datetime,
23    date_close      datetime,
24    notify_tiers_at_create tinyint,
25    tms             timestamp,
26    import_key      varchar(14)
27 )ENGINE=InnoDB;
28
    
```





## Construcción del Software

```

messaging.php  agenda.php
<?php
require '../main.inc.php';
require_once DOL_DOCUMENT_ROOT.'/ticket/class/actions_ticket.class.php';
require_once DOL_DOCUMENT_ROOT.'/core/class/html.formticket.class.php';
require_once DOL_DOCUMENT_ROOT.'/core/lib/ticket.lib.php';
require_once DOL_DOCUMENT_ROOT.'/core/lib/company.lib.php';
require_once DOL_DOCUMENT_ROOT.'/core/lib/date.lib.php';
require_once DOL_DOCUMENT_ROOT.'/core/class/extrafields.class.php';
require_once DOL_DOCUMENT_ROOT.'/contact/class/contact.class.php';

// Load translation files required by the page
$langs->loadLangs(array('companies', 'other', 'ticket'));

// Get parameters
$id = GETPOST('id', 'int');
$ref = GETPOST('ref', 'alpha');
$track_id = GETPOST('track_id', 'alpha', 0);
$socid = GETPOST('socid', 'int');
$action = GETPOST('action', 'aZ09');

$limit = GETPOST('limit', 'int') ? GETPOST('limit', 'int') : $conf->liste_limit;
$sortfield = GETPOST('sortfield', 'alpha');
$sortorder = GETPOST('sortorder', 'alpha');
$page = GETPOSTISSET('pageplusone') ? (GETPOST('pageplusone') - 1) : GETPOST('page', 'int');
$page = is_numeric($page) ? $page : 0;
$page = $page - 1 ? 0 : $page;
if ($sortfield) $sortfield = 'a.datep,a.id';
if ($sortorder) $sortorder = 'DESC';
$offset = $limit * $page;
$pageprev = $page - 1;
$pagenext = $page + 1;

```

```

public function initAsSpecimen()
{
    $this->id = 0;

    $this->ref = 'TICKET-001';
    $this->track_id = 'XXXXXXXX';
    $this->origin_email = 'email@gmail.com';
    $this->fk_project = 1;
    $this->fk_user_create = 1;
    $this->fk_user_assign = 1;
    $this->subject = 'Subject of ticket';
    $this->message = 'Message of ticket';
    $this->fk_status = 0;
    $this->resolution = '1';
    $this->progress = '10';
    $this->timing = '30';
    $this->type_code = 'TYPECODE';
    $this->category_code = 'CATEGORYCODE';
    $this->severity_code = 'SEVERITYCODE';
    $this->datec = '';
    $this->date_read = '';
    $this->date_close = '';
    $this->stms = '';
}

/**
 * print selected status
 * @param string $selected selected status
 * @return void
 */
public function printSelectedStatus($selected = '')
{
    print_foreach_selectarray('search_fk_status', $this->status_short, $selected, $show_empty = 1, $key_in_label = 0, $value_as_key = 0, $option = '', $
        translate = 1, $maxlen = 0, $disabled = 0, $sort = '', $success = '');
}

```

```

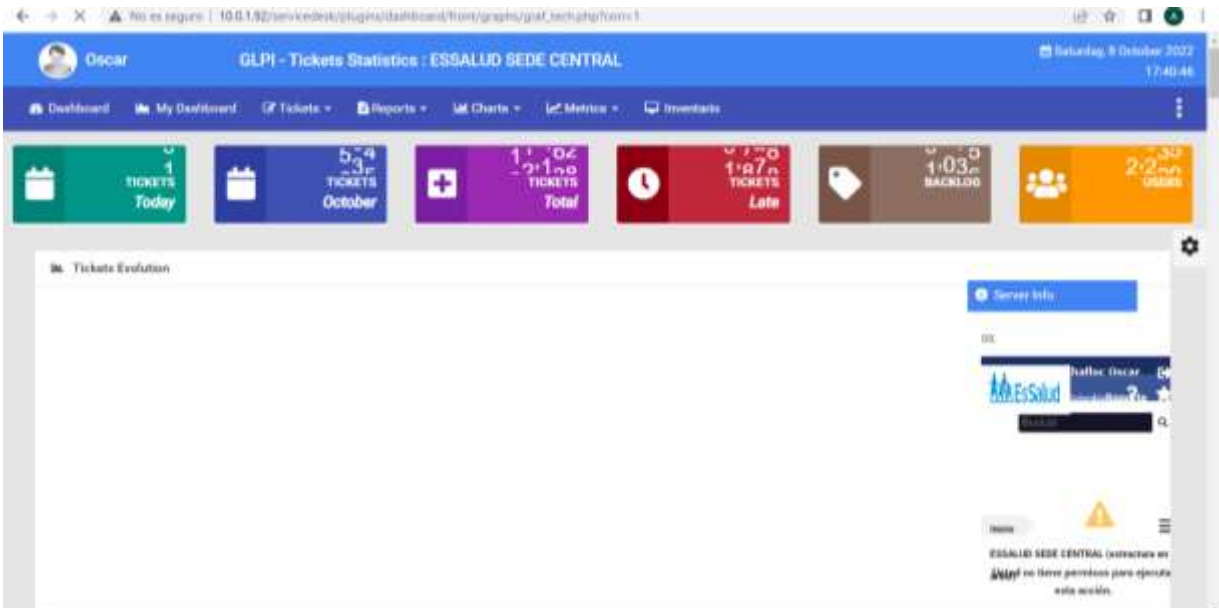
require '../main.inc.php';
require_once DOC_DOCUMENT_ROOT.'/ticket/class/actions_ticket.class.php';
require_once DOC_DOCUMENT_ROOT.'/core/class/html.formticket.class.php';
require_once DOC_DOCUMENT_ROOT.'/core/lib/ticket.lib.php';
require_once DOC_DOCUMENT_ROOT.'/core/lib/company.lib.php';
require_once DOC_DOCUMENT_ROOT.'/core/lib/date.lib.php';
require_once DOC_DOCUMENT_ROOT.'/core/class/extrafields.class.php';
require_once DOC_DOCUMENT_ROOT.'/contact/class/contact.class.php';

// load translation files required by the page
$langs->loadLangs(array('companies', 'other', 'ticket'));

// Get parameters
$id = GETPOST('id', 'int');
$ref = GETPOST('ref', 'alpha');
$track_id = GETPOST('track_id', 'alpha', 0);
$socid = GETPOST('socid', 'int');
$action = GETPOST('action', 'a200');

$limit = GETPOST('limit', 'int') ? GETPOST('limit', 'int') : $conf->liste_limit;
$sortfield = GETPOST('sortfield', 'alpha');
$sortorder = GETPOST('sortorder', 'alpha');
$page = GETPOSTISSET('pageplusone') ? (GETPOST('page', 'int')
    + 1) : (isset($page) ? $page : 0);
if (! $sortfield) $sortfield = 'a.datep,e.id';
if (! $sortorder) $sortorder = 'desc';
$offset = $limit * $page;
$pageprev = $page - 1;
$pagenext = $page + 1;

```



----- ...racterísticas - Urgencia es Todo

regla regla global grupo Buscar

Mostrar (número de elementos) 10000

Página actual en PDF paisaje Desde 1

Acciones

ID	Título	Status	Type	Categoría	Solicitante - Solicitante	Ubicación	Fuente de solicitud	Fecha de apertura	Prioridad	Asignado a - Técnico
145351	Memorando Circular N° 1431-GCTIC-ESSALUD-2022 // Necesidad de renovación tecnológica	Cerrado	Solicitud	SOPORTE > SOFTWARE > DOMINIO Y CORREO > CONFIGURACION	Administra Correo	GCTIC-P6-EL > GPROD-P6-EL > SGST-P6-EL	E-Mail	15-09-2022 09:03	Media	Rodriguez Segura Alonso Humberto
144328	creación de email institucional	Cerrado	Solicitud	SOPORTE > SOFTWARE > DOMINIO Y CORREO > CREACION DE USUARIO	Tarrillo Herrera Mack Johan	RA JAEN	E-Mail	07-09-2022 16:53	Media	Muñoz Rojas Eider Arnol
144908	Correo institucional muestra errores a enviar a hotmail y gmail	Nuevos	Solicitud		Tarrillo Herrera Mack Johan		E-Mail	12-09-2022 16:58	Media	
144234	Creación de email institucional	Cerrado	Solicitud	SOPORTE > SOFTWARE > DOMINIO Y CORREO > CREACION DE USUARIO	Tarrillo Herrera Mack Johan	RA JAEN	E-Mail	07-09-2022 11:18	Media	Muñoz Rojas Eider Arnol
144325	Creación de email institucional	Cerrado	Solicitud		Tarrillo Herrera Mack Johan		E-Mail	07-09-2022 16:46	Media	

## Entrega del Software

Bienvenidos a la Plataforma Gestión de Mesa de Ayuda

Usuario

Contraseña

Recuérdame

Enviar

[¿Olvidó su contraseña?](#)

Principal

Inicio de sesión: mack.tarrillo

Apellido	<input type="text" value="Aparicio Carrasco"/>
Nombre	<input type="text" value="Aparicio"/>
Idioma	<input type="text" value="Español (América Latina)"/>
Contraseña	<input type="password"/>
Confirmación de contraseña	<input type="password"/>
Política de seguridad de la clave	Longitud mínima de clave: 6 La clave debe contener: <b>Dígito</b>
Teléfono	<input type="text"/>
Teléfono móvil	<input type="text" value="915120166"/>
Teléfono 2	<input type="text"/>
Número Administrativo	<input type="text"/>
Ubicación	<input type="text" value="...:OP-P8-EL &gt; Despacho de Administración-E.L"/> ⓘ



Imagen/Foto

Archivo(s) (30 Mb máx) ⓘ  
Arrastre y coloque su archivo aquí o

Limpiar

Zona horaria

Correos electrónicos +

Claves de acceso remoto

Token personal

Regenerar



Anexo 16. Evidencias fotográficas

