



Universidad Científica del Perú - UCP

*Registrado en el Asiento N° A00010 de la Partida N° 11000318, Personas Jurídicas de Iquitos,
Superintendencia de los Registros Públicos - SUNARP*

**FACULTA DE CIENCIAS E INGENIERÍA
PROGRAMA ACADÉMICO DE INGENIERÍA DE SISTEMAS DE
INFORMACIÓN**

TRABAJO DE SUFICIENCIA PROFESIONAL

**“PROPUESTA DE ACTUALIZACIÓN DE LA PLATAFORMA DEL
SISTEMA CLIENTE PARA LA INTEGRACIÓN DE PROCESOS DE
LA EMPRESA SOFTWARE Y SISTEMAS DEL PERÚ SAC –
IQUITOS 2021”**

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

AUTOR: Br. JARA PALACIOS, Eric José.

ASESOR: Ing. CASTILLO CHALCO, Isaac Duhamel.

SAN JUAN BAUTISTA – LORETO – MAYNAS – PERÚ

2021

DEDICATORIA

Este trabajo de investigación va dirigido a 3 personas de manera puntual, por el apoyo recibido durante mi etapa universitaria; que fue en vida mi querido amigo Joel Elgegren Villacrés, además de mi colega Héctor Rabines Arellano y quien fue en aquel momento mi pareja, la srta. Rita Baca Chumbe, por enseñarme a ver la vida como un desafío; además a mi familia, amigos y docentes que apoyaron con su granito de arena para que cumpla esta meta de tener un grado profesional.

Bach. Eric José Jara Palacios

AGRADECIMIENTO

Muestro todo mi agradecimiento a mi familia Jara Palacios, que fue el base fundamental para cumplir este objetivo, de llegar hacer un profesional totalmente hecho y derecho, además agradecer al sr. Ronald Gronerth y su familia por darme la oportunidad de confiar en mí y desarrollar vivencias muy benevolentes.

Bach. Eric José Jara Palacios

CONSTANCIA DE ORIGINALIDAD DEL TRABAJO DE INVESTIGACIÓN DE LA UNIVERSIDAD CIENTÍFICA DEL PERÚ - UCP

El presidente del Comité de Ética de la Universidad Científica del Perú - UCP

Hace constar que:

El Trabajo de Suficiencia Profesional titulado:

**“PROPUESTA DE ACTUALIZACIÓN DE LA PLATAFORMA DEL SISTEMA
CLIENTE PARA LA INTEGRACIÓN DE PROCESOS DE LA EMPRESA SOFTWARE
Y SISTEMAS DEL PERÚ SAC – IQUITOS 2021”**

De los alumnos: **JARA PALACIOS ERIC JOSÉ**, de la Facultad de Ciencias e Ingeniería, pasó satisfactoriamente la revisión por el Software Antiplagio, con un porcentaje de **1% de plagio**.

Se expide la presente, a solicitud de la parte interesada para los fines que estime conveniente.

San Juan, 9 de junio del 2021.



Dr. César J. Ramal Asayag
Presidente del Comité de Ética – UCP

Urkund Analysis Result

Analysed Document: UCP_IngenieriaDeSistemasDeInformacion_2021_TrabajoDeSuficienciaF
(D107915270)
Submitted: 6/4/2021 8:08:00 PM
Submitted By: revision.antiplagio@ucp.edu.pe
Significance: 1 %

Sources included in the report:

FT_SrCaicedo.pdf (D87076507)
<http://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/>
<https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion>
<https://www.ekon.es/diagrama-procesos-empresa/>
<https://lopezvictor01.jimdofree.com/app/download/9016081969/Qu%C3%A9+es+la+Ingenier%C3%ADa+de+Software.pdf?t=1485707627>

Instances where selected sources appear:

9

ACTA DE SUSTENTACIÓN DE TRABAJO DE SUFICIENCIA PROFESIONAL

FACULTAD DE CIENCIAS E INGENIERÍA

Con Resolución Decanal N° 405-2021-UCP-FCEI del 07 de junio del 2021, la FACULTAD DE CIENCIAS E INGENIERÍA DE LA UNIVERSIDAD CIENTÍFICA DEL PERÚ - UCP designa como Jurado Evaluador de la sustentación de tesis a los señores:

- | | |
|--|------------|
| • Ing. Jimmy Max Ramírez Villacorta, Mg. | Presidente |
| • Lic. Carlos Marthans Ruiz, Mg | Miembro |
| • Ing. Cesar Palacios Chávez. | Miembro |

Como Asesor: **Ing. Isaac Duhamel Castillo Chalco**

En la ciudad de Iquitos, siendo las 07:00 am del día martes 13 de julio del 2021, a través de la plataforma ZOOM supervisado en línea por la Secretaria Académica del Programa Académico de Ingeniería de Sistemas y de información de la Facultad de Ciencias e Ingeniería de la Universidad Científica del Perú., se constituyó el Jurado para escuchar la sustentación y defensa del Trabajo de Suficiencia Profesional: **“PROPUESTA DE ACTUALIZACIÓN DE LA PLATAFORMA DEL SISTEMA CLIENTE PARA LA INTEGRACIÓN DE PROCESOS DE LA EMPRESA SOFTWARE Y SISTEMAS DEL PERÚ SAC – IQUITOS 2021”**.

Presentado por el sustentante: **ERIC JOSE JARA PALACIOS**

Como requisito para optar el título profesional de: **INGENIERO DE SISTEMA DE INFORMACIÓN**

Luego de escuchar la sustentación y formuladas las preguntas las que fueron: **ABSUELTAS**


El Jurado después de la deliberación en privado llegó a la siguiente conclusión:

La sustentación es: **APROBADO POR MAYORIA**

En fe de lo cual los miembros del Jurado firman el acta.



Miembro



Presidente



Miembro

APROBACIÓN

Tesis sustentada en acto público el día 13 de julio a las 7:00 am de 2021



Ing. Jimmy Max Ramírez Villacorta, Mg.
Presidente del Jurado



Lic. Carlos Enrique Marthans Ruíz, Mg.
Miembro del Jurado



Ing. Cesar Augusto Palacios Chávez
Miembro del Jurado



Ing. Isaac Duhamel Castillo Chalco
Asesor

ÍNDICE

PORTADA.....	I
DEDICATORIA.....	II
AGRADECIMIENTO.....	III
CONSTANCIA DE ANTIPLAGIO	IV
ACTA DE SUSTENTACIÓN.....	V
APROBACIÓN	VII
ÍNDICE	VIII
ÍNDICE DE TABLAS	IX
ÍNDICE DE GRÁFICOS	X
RESUMEN	XIII
ABSTRACT	XIV
CAPÍTULO I. INTRODUCCIÓN.....	15
1.1. Planteamiento del Problema.....	17
1.2. Formulación del Problema	19
1.3. Justificación	20
CAPÍTULO II. MARCO TEÓRICO	21
2.1. Marco Teórico Referencial.....	21
2.2. Marco Teórico Conceptual.....	26
2.3. Análisis.....	94
2.4. Objetivos	105
2.5. Hipótesis	106
2.6. Variables.....	107
CAPÍTULO III. MATERIALES Y MÉTODOS.....	108
3.1. Lugar y Desarrollo de la Investigación	108
3.2. Tipo y Diseño de Investigación	108
3.3. Población y Muestra	109
3.4. Técnica, Instrumentos y Procedimientos de Recolección de Datos.....	109
CAPÍTULO IV. RESULTADOS.....	111
4.1. Análisis descriptivo de los datos de variable independiente:	111
CAPÍTULO V. DISCUSIÓN.....	131
5.1. Discusión	131
CAPÍTULO VI. CONCLUSIONES	133
6.1. Conclusiones	133
RECOMENDACIONES	135
REFERENCIAS BIBLIOGRÁFICAS	136
ANEXOS	139

ÍNDICE DE TABLAS

Tabla N° 01 – Cuadro comparativo de las metodologías, elaboración propia....	43
Tabla N° 02 – Cuadro que especifica los puntos que pertenecen en cada fase. Elaboración propia	45
Tabla N° 03 – Cuadro de tipo de relaciones que tiene los casos de uso. Elaboración propia	87
Tabla N° 04 – Cuadro de requerimientos funcionales. Elaboración propia	97
Tabla N° 05 – Cuadro de requerimientos no funcionales. Elaboración propia .	97
Tabla N° 06 – Operacionalización de variables.....	107
Tabla N° 07 – Pregunta 01. Elaboración propia	111
Tabla N° 08 – Pregunta 02. Elaboración propia	112
Tabla N° 09 – Pregunta 03. Elaboración propia	113
Tabla N° 10 – Pregunta 04. Elaboración propia	114
Tabla N° 11 – Pregunta 05. Elaboración propia	115
Tabla N° 12 – Pregunta 06. Elaboración propia	116
Tabla N° 13 – Pregunta 07. Elaboración propia	117
Tabla N° 14 – Pregunta 08. Elaboración propia	118
Tabla N° 15 – Pregunta 09. Elaboración propia	119
Tabla N° 16 – Pregunta 10. Elaboración propia	120
Tabla N° 17 – Pregunta 11. Elaboración propia	121
Tabla N° 18 – Pregunta 12. Elaboración propia	122
Tabla N° 19 – Pregunta 13. Elaboración propia	123
Tabla N° 20 – Pregunta 14. Elaboración propia	124
Tabla N° 21 – Pregunta 15. Elaboración propia	125
Tabla N° 22 – Pregunta 16. Elaboración propia	126
Tabla N° 23 – Promedio de mitigar riesgo de error. Elaboración propia	127
Tabla N° 24 – Promedio de administración de la información. Elaboración propia.	128
Tabla N° 25 – Promedio de dimensión calidad de servicio. Elaboración propia	129
Tabla N° 26 – Promedio de dimensión satisfacción de personal. Elaboración propia	130

ÍNDICE DE GRÁFICOS

Gráfico N° 01 – Ciclo de vida de un producto software se desarrolla fuera del ámbito productivo.....	27
Gráfico N° 02 – Ciclo de vida lineal.....	31
Gráfico N° 03 – Ciclo de vida en cascada puro.....	32
Gráfico N° 04 – Ciclo de vida en V.....	33
Gráfico N° 05 – Ciclo de vida tipo Sashimi.....	34
Gráfico N° 06 – Ciclo de vida en cascada con subproyectos.....	34
Gráfico N° 07 – Ciclo de vida iterativo.....	35
Gráfico N° 08 – Ciclo de vida por prototipos.....	36
Gráfico N° 09 – Ciclo de vida evolutivo.....	37
Gráfico N° 10 – Ciclo de vida incremental.....	38
Gráfico N° 11 – Ciclo de vida espiral.....	39
Gráfico N° 12 – Ciclo de vida orientado a objetos.....	40
Gráfico N° 13 – Visión del desarrollo de software.....	48
Gráfico N° 14 – Etapas del desarrollo de software.....	49
Gráfico N° 15 – Actividades del ciclo de desarrollo de la arquitectura y su mapeo dentro de las actividades técnicas del desarrollo software.....	51
Gráfico N° 16 – Visualización del patrón de diseño Factory Method.....	54
Gráfico N° 17 – Visualización del patrón de diseño Abstract Method.....	55
Gráfico N° 18 – Visualización del patrón de diseño Singleton.....	56
Gráfico N° 19 – Visualización del patrón de diseño Builder.....	56
Gráfico N° 20 – Visualización del patrón de diseño Prototype.....	57
Gráfico N° 22 – Visualización del patrón de diseño Object Pool.....	58
Gráfico N° 21 – Visualización del ciclo de vida un objeto del ObjectPool.....	58
Gráfico N° 23 – Visualización del patrón de diseño Adapter.....	59
Gráfico N° 24 – Visualización del patrón de diseño Bridge.....	60
Gráfico N° 25 – Visualización del patrón de diseño Composite.....	61
Gráfico N° 26 – Visualización del patrón de diseño Decorator.....	61
Gráfico N° 27 – Visualización del patrón de diseño Facade.....	62
Gráfico N° 28 – Visualización del patrón de diseño Flyweight.....	63
Gráfico N° 29 – Visualización del patrón de diseño Proxy.....	64
Gráfico N° 30 – Visualización del patrón de diseño Iterator.....	65

Gráfico N° 31 – Visualización del patrón de diseño Command.	66
Gráfico N° 32 – Visualización del patrón de diseño Observer.	67
Gráfico N° 33 – Visualización del patrón de diseño Template Method.	67
Gráfico N° 34 – Visualización del patrón de diseño Strategy.	68
Gráfico N° 35 – Visualización del patrón de diseño Chain Of Responsibility. .	69
Gráfico N° 36 – Visualización del patrón de diseño Interpreter.	69
Gráfico N° 37 – Visualización del patrón de diseño Mediator.	70
Gráfico N° 38 – Visualización del patrón de diseño Memento.	71
Gráfico N° 39 – Visualización del patrón de diseño Null Object.	71
Gráfico N° 40 – Visualización del patrón de diseño Visitor.	72
Gráfico N° 41 – Visualización del patrón de diseño State.	73
Gráfico N° 42 – Visualización de Tipos de Requerimientos No Funcionales. ..	75
Gráfico N° 43 – Visualización de la descomposición de un sistema.	76
Gráfico N° 44 – Visualización de un diagrama entidad relación.	77
Gráfico N° 45 – Visualización de una entidad.	78
Gráfico N° 47 – Visualización de una entidad débil.	78
Gráfico N° 46 – Visualización de una entidad fuerte.	78
Gráfico N° 49 – Visualización de una relación débil.	79
Gráfico N° 48 – Visualización de una relación.	79
Gráfico N° 52 – Visualización de un atributo clave parcial.	79
Gráfico N° 51 – Visualización de un atributo clave.	79
Gráfico N° 50 – Visualización de un atributo.	79
Gráfico N° 53 – Principales elementos de un diagrama de procesos.	81
Gráfico N° 54 – Ejemplo de diagrama de procesos.	81
Gráfico N° 55 – Ejemplo de diagrama de flujo.	82
Gráfico N° 56 – Clasificación de los diagramas UML.	85
Gráfico N° 57 – Composición de un diagrama de caso de uso.	86
Gráfico N° 58 – Relaciones de un diagrama de caso de uso.	87
Gráfico N° 59 – Visualización de una estructura secuencial.	90
Gráfico N° 60 – Visualización de una estructura selección.	90
Gráfico N° 61 – Visualización de una estructura repetición.	91
Gráfico N° 62 – Mapa conceptual de la programación orientada a objetos.	93
Gráfico N° 63 – Organigrama de la institución.	94
Gráfico N° 64 – Ciclo de vida evolutivo – análisis.	94

Gráfico N° 65 – Metodología de desarrollo de software SCRUM - análisis.....	95
Gráfico N° 66 – Actividades del ciclo de desarrollo de la arquitectura y su mapeo dentro de las actividades técnicas del desarrollo software espiral - análisis. ...	96
Gráfico N° 67 – Visualización del patrón de diseño Null Object - análisis.	96
Gráfico N° 68 – Visualización de diseño de arquitectura - análisis.....	98
Gráfico N° 69 – Visualización de diseño entidad – relación - análisis.	99
Gráfico N° 70 – Visualización de diagrama de procesos actual - análisis.	100
Gráfico N° 71 – Visualización de diagrama de procesos propuesto - análisis.	101
Gráfico N° 72 – Visualización de diagrama de flujo de proceso actual - análisis. ..	102
Gráfico N° 73 – Visualización de caso de uso del mantenimiento de la entidad – análisis.	103
Gráfico N° 74 – Visualización de caso de uso de gestión de actividades - análisis.	103
Gráfico N° 75 – Pregunta 01. Elaboración propia.....	111
Gráfico N° 76 – Pregunta 02. Elaboración propia.....	112
Gráfico N° 77 – Pregunta 03. Elaboración propia.....	113
Gráfico N° 78 – Pregunta 04. Elaboración propia.....	114
Gráfico N° 79 – Pregunta 05. Elaboración propia.....	115
Gráfico N° 80 – Pregunta 06. Elaboración propia.....	116
Gráfico N° 81 – Pregunta 07. Elaboración propia.....	117
Gráfico N° 82 – Pregunta 08. Elaboración propia.....	118
Gráfico N° 83 – Pregunta 09. Elaboración propia.....	119
Gráfico N° 84 – Pregunta 10. Elaboración propia.....	120
Gráfico N° 85 – Pregunta 11. Elaboración propia.....	121
Gráfico N° 86 – Pregunta 12. Elaboración propia.....	122
Gráfico N° 87 – Pregunta 13. Elaboración propia.....	123
Gráfico N° 88 – Pregunta 14. Elaboración propia.....	124
Gráfico N° 89 – Pregunta 15. Elaboración propia.....	125
Gráfico N° 90 – Pregunta 16. Elaboración propia.....	126
Gráfico N° 91 – Promedio de dimensión mitigar riesgo de error. Elaboración propia.	127
Gráfico N° 92 – Promedio de dimensión administración de la información. Elaboración propia.	128
Gráfico N° 93 – Promedio de dimensión calidad de servicio. Elaboración propia.....	129
Gráfico N° 94 – Promedio de dimensión satisfacción de personal. Elaboración propia.	130

RESUMEN

En este trabajo de investigación “Propuesta de actualización de la plataforma del sistema Cliente para la integración de procesos de la empresa software y sistemas del Perú SAC - Iquitos 2021”, se expone una propuesta de una solución de un determinado proceso interno de la empresa Software y Sistemas del Perú sac., que es la actualización de la plataforma del sistema “Cliente” en el cual se visualiza las licencias que tienen las entidades públicas con los programas que otorga la empresa, para que se pueda anexa de manera automática con el otro sistema para atenciones llamado “Sistemas de Atención Cliente - SAC” y así evitar el proceso humano y manual de actualizar las licencias de las entidades públicas, en resumen, dentro la empresa Software y Sistemas del Perú contiene en su haber 2 sistemas perjudiciales para el funcionamiento interno de alta calidad, uno es el sistema Cliente que está en lenguaje FoxPro v9.0 que gestiona las licencias de la entidades estatales, mientras el otro es el Sistema de Atención Cliente, que está hecho en el lenguaje PHP, el cual gestiona el soporte y apoyo en reporte de las entidades estatales mediante asistencias de soporte a los usuarios, pero se necesita saber en esas ocasiones de atención de soporte, la licencia y estado (si está permitido o denegado) de la entidad para poder tener una atención de soporte rápida y así tener una mejor gestión de atenciones, pero el problema surge ya que el proceso de actualización de licencias entre los 2 sistemas necesita intervención humana y además que es manual, entonces esto genera demora a la obtención de información precisa para los usuarios de soporte, entre otros inconvenientes. Este caso aplica para las áreas de soporte técnico y ventas, que entre ellas suman una cantidad de trabajadores que son 14, entonces se optó en primer momento analizar la situación de la empresa en la actualidad para establecer una propuesta de actualización precisa para solucionar los inconvenientes, la cual, es actualizar todo el funcionamiento y la plataforma del sistema Cliente para que pueda tener una integración de proceso completa con el Sistema de Atención Cliente.

Palabras claves: Propuesta de actualización, sistemas, soporte, ventas, integración de proceso.

ABSTRACT

In this research work "Proposal to update the Client system platform for the integration of processes of the software and systems company of Peru SAC - Iquitos 2021, A proposal for a solution of a certain internal process of the company Software y Sistemas del Perú is explained, which is the update of the platform of the "Client" system in which the licenses that public entities have with the programs are displayed granted by the company, so that it can be automatically attached to the other service system called "Customer Service Systems - SAC" and avoid the human and manual process of updating the licenses of public entities, in short, within the company Software y Sistemas del Perú contains to its credit 2 systems that are detrimental to the internal operation of high quality, one is the Client system that is in FoxPro v9.0 language that manages the licenses of state entities, while the other is the Client Service System, which is made in the PHP language, which manages the support and support in reporting of state entities through support assistance to users, but it is necessary to know in those occasions of support care , the license and status (if allowed or denied) of the entity in order to have a quick support service and thus have a better service management, but the problem arises since the license update process between the 2 systems needs intervention human and also that it is manual, then this generates delay in obtaining accurate information for support users, among other inconveniences This case applies to the technical support and sales areas, which include a number of workers that are 14, so it was first decided to analyze the current situation of the company to establish a precise update proposal to solve the problems , which is to update the entire operation and platform of the Client system so that it can have a complete process integration with the Client Service System.

Keywords: Proposal for update, systems, support, sales, process integration.

CAPÍTULO I. INTRODUCCIÓN

Software y Sistemas del Perú sac., empresa dedicada al rubro de venta de software presupuestal – administrativo y contable - financiero; mediante los programas Melissa y Clarissa que trabajan con la tecnología Octopus v1.0 patentado por el sr. Ronald Gronerth. Lo que brindamos a nuestro cliente es la capacidad de eficiencia, flexibilidad y rapidez para realizar sus reportes informativos. Se resalta que la empresa tiene muchos clientes, los cuales sobresalen los ministerios que tienen un servicio especial por la proporción que significa esas entidades, además de los procesos de solicitud de soporte o como se le brinda los manuales o tutoriales para que su trabajo sea de manera eficiente¹. Actualmente en la empresa tiene dos sistemas en el cual realiza procesos importantes de control y seguimiento para las áreas de soporte técnico y ventas, pero mientras va pasando el tiempo la tecnología de la información van quedando obsoletos, en este caso puntual sería el sistema Cliente que está hecho en Microsoft Visual FoxPro 9.0 que fue en su tiempo un lenguaje de programación muy vanguardista para la programación estructurada pero desde que Microsoft obtuvo el total uso de este lenguaje de programación se fue quedando obsoleto y atrasado a los nuevos paradigmas de programación por ello quedo sin soporte o atención al cliente de esta tecnología, entonces se recurría a parche de creación de una comunidad de programadores de FoxPro que ayudaban a subsanar algunos errores de este lenguaje, pero ahora en pleno año 2021 este lenguaje no se adapta a las nuevas tendencias de programación y por eso motivo su programación no están escalable. Se hará un estudio de posibilidades para verificar si es viable hacer una próxima actualización del sistema Cliente para que los procesos de la empresa sean mucho más eficientes y eficaces. Además, se plantea un proceso, de cómo sería el procedimiento de actualización de las licencias en los programas Melissa y Clarissa de acuerdo a cada entidad pública, entonces se tendrán que obtener la información necesaria y concisa que permita ver las carencias de la actual plataforma de lenguaje de programación y dar una solución precisa a la empresa, pero manteniendo comunicación con los interesados o stakeholders mediante chat, video

¹ OCAMPO, Edgard y RÍOS, Gerson, “*Software Melissa*”, 2019, <http://softwaremelissa.com/> (consultada al 13 de abril 2021).

conferencia, correo electrónico o llamada telefónica. La información que hay en la empresa se tiene que manejar de manera segura y confiable ya que es primordial para su buena administración de los clientes, por ellos se necesita emplear mucho criterio, para proporcionar un caso de estudio totalmente relacionado con lo mencionado en el primer párrafo, hay que tomar en cuenta la capacidad de qué tipo de información se está administrando e identificar los casos que son excepciones para analizar el proceso de negocio. Hay precisar que el desarrollo este proyecto de investigación determinara si es viable o no la propuesta que se está planteado a esta problemática que es la falta automatización aun proceso de circunstancial al interior de la empresa que es la actualización de licencias de las entidades, aunque es un proceso que puede ser simple de manera contextual, el riesgo que implica puede llegar a tener problemas en el marco legal ya que la normas estatales son muy claras y especificas a la falta de un acuerdo de venta, por eso es de preferencia que el factor humano que es la causa de más riesgo sea disminuido de manera significativa para obtener una calidad de servicio eficiente, eficaz y de muy alta cualidad.

1.1. Planteamiento del Problema

Para tener una gestión totalmente de alta calidad, trabajábamos con datos totalmente íntegros, sin embargo, el flujo de entidades que son cliente de la empresa que utilizan el software, son cada año, de más voluminosos cuantitativamente y cualitativamente por lo que las soluciones de tecnologías de la información que tiene la empresa llegó a cierto punto que hay mucho consumo de tiempo y riesgo de un error por el factor humano.

Dicho problema se presenta más comúnmente de lo que se espera, ya que, si analizamos un caso muy popular en estos últimos años, es en Estados Unidos de Norteamérica, más precisamente en la empresa Facebook por motivo que desea integrar procesos entre sus aplicativos para que sus proceso de negocio es la publicidad digital sea totalmente conformada por la interacciones que tiene sus aplicaciones con el usuario, pero el principal problema es la integración de sus procesos entre sus aplicaciones para que sus beneficios hacia los usuarios sean mejor, por ejemplo mientras utilizas el aplicativo de Whatsapp Business, que toda esa iteración que tienes con tus clientes sean integrados en todas las demás aplicaciones que la experiencia del usuario sea más completa.

Pero en otro ámbito más conocido a nivel nacional se puede conocer en la gestión pública, ya que, los sistemas SIAF aplicativo (software de escritorio) y SIAF WEB (aplicativo Web) son sistemas para el manejo de gestión presupuestal y gasto público, pero el principal problema de estos sistemas es la falta de procesos de integración para la sincronización de la información, porque en los cierres trimestrales de cada entidad existen diferencias que a la larga generan muchos subprocesos que a su genera mucha burocracia, porque se necesita que el documento o reporte pase por muchos cargos para finalizar el proceso.

Mientras ese problema también se presenta en la empresa Software y Sistemas del Perú sac., que es la falta de integración de los procesos para agrupar las licencias que tienen las entidades públicas como clientes, con la administración de atenciones de soporte técnico para de dicha empresa, ya que la principal causa es el proceso manual que se hace para transferir la licencias del sistema Cliente con las otras tecnologías de información que tiene en su haber la

empresa, es decir, la poca capacidad de interacción del sistema Cliente a nuevas tecnologías causa que sea complicado adaptar funciones más integradas para que no consuma muchos recursos. Pero si entendemos que las nuevas tendencias de programación son orientadas a objetos que permite un desarrollo de software más escalable durante el tiempo de vida o producción del software, esto significa que la plataforma en que está constituida el sistema Cliente, es de la antigua tendencia de programación estructurada que no permite fácilmente las modificaciones de los procesos principales, esto quiere decir, que el consumo de recursos para una buena práctica de programación llega ser muy difícil. Si se plantea que los procesos sean integrados para facilitar el flujo de trabajo se tiene que identificar de manera correcta el problema o cual es la causa del proceso de actualización de las licencias sea manual o no programado, en un entorno explicativo, se puede decir que es la falta de integración de procesos internos del sistema Cliente para que las funciones sean automáticas y no necesite intervención humana o manual, pero si lo decimos en manera técnica, entonces se comienza en el sistema Cliente exportando un archivo XLS (EXCEL) que después, tiene que ser cargado en otro sistema para tener las licencias actualizadas de las entidades públicas para que personal de soporte tenga una buena calidad de atención, pero haciendo este procedimiento hay el riesgo de cometer un error ya que el factor humano en ciertos puntos o situaciones tiende a fallas, además que el continuo procedimiento de actualizado no tiene una persona totalmente encargado, es decir, no hay un responsable para ese proceso.

1.2. Formulación del Problema

1.2.1. Problema general:

¿De qué manera la propuesta de actualización de la plataforma del sistema Cliente, incidiría la integración de procesos en la empresa Software y Sistemas del Perú sac – Iquitos 2021?

1.2.2. Problemas específicos:

- ¿Cómo la propuesta de actualización de la plataforma del sistema Cliente, mitigaría el riesgo de un error en los procesos internos en la empresa Software y Sistemas del Perú sac – Iquitos 2021?
- ¿En qué dimensión mejoraría la propuesta de actualización de la plataforma del sistema Cliente, con la administración de la información en la empresa Software y Sistemas del Perú sac – Iquitos 2021?
- ¿En qué medida mejoraría el índice de calidad de servicio con la propuesta de actualización de la plataforma del sistema Cliente en la empresa Software y Sistemas del Perú sac – Iquitos 2021?
- ¿En qué manera la propuesta de actualización de la plataforma del sistema Cliente mejorará la satisfacción del personal en la empresa Software y Sistemas del Perú sac – Iquitos 2021?

1.3. Justificación

En este trabajo de investigación justifica su proyección por un caso de proceso interno de la empresa Software y Sistemas del Perú sac., que tiene una propuesta actualizar y mejorar la calidad de su servicio centralizando o conectando de mejor manera a los sistemas que tiene en su haber para que así evitar errores en procesos manuales. Esta investigación dará las pautas de cómo se desarrollaría este proyecto, el cual sería una actualización a la plataforma del sistema Cliente parametrizando los índices como tiempo, económico y organizacional para que los interesados tengan un claro entendimiento del desarrollo de la actualización de la plataforma del sistema Cliente. Además, que se administrará de manera más automatizada los procesos para que el equipo de soporte, tenga una total confianza que la información que tiene en disponible sea actualizado y validada por el equipo del área de venta que gestionan las licencias hacia las entidades.

La metodología aplicada en este proyecto será muy importante para la administración de la información, para su posterior análisis y así obtener una integración más eficiente, pero esto dese ser gracias a los medios que multimedia y mantendrán la comunicación entre los interesados con el equipo de trabajo. Esta práctica ayuda a identificar los cambios pueden tener este proceso, pero llegar a desviar hacían el fin planteado, que, a su vez, el marco sustancial que debate este proyecto de investigación.

Poniendo en práctica, hay mucha necesidad de tener una buena solución al problema por motivo que la gestión empresarial siempre tiene como objetivo tener un buen uso de las tecnologías de la información, porque desea tener un nivel de operativo de alta calidad, para así generar una optimización de recursos de la empresa. Este trabajo de investigación tiene evaluar el impacto de su propuesta dada en la empresa que desea solucionar la problemática con la implementación de la actualización de la plataforma del sistema Cliente para favorecer el trabajo en las áreas de soporte y ventas en la empresa Software y Sistemas del Perú sac.

CAPÍTULO II. MARCO TEÓRICO

2.1. Marco Teórico Referencial

2.1.1. Antecedentes de estudios:

- Nivel internacional:

La Rosa, H. et al (2004), en su tesis “Propuesta de Diseño de un Sistema de Información para el Seguimiento de Ventas para una Industria de Manufactura”² de la Universidad Católica Andrés Bello en Caracas – Venezuela en el periodo 2001 al 2002; a base de este trabajo de investigación se toma de referencia porque busca dar una propuesta de un sistema de información que se determina mediante encuestas para dar un análisis conciso para recomendar una implementación a futuro, además que, da sugerencia generales de como el trabajo de las personas que participaron como muestra pueden mejorar sus procesos y la calidad. También se resalta este trabajo de tesis por motivo, que su aspecto fundamental, es referenciar los puntos de vista teóricos y conceptuales para cómo se desarrollaría los aspectos técnicos para una implementación del sistema, de acuerdo al análisis como guía principal para solucionar las necesidades de la empresa y optimizar los procesos existentes que tiene situación actual. Luego da pautas puntuales de cómo se debería automatizar procesos para aligerar la carga laboral de los empleados, mediante diagramas, que sirven para expresar de manera sencilla, factible y concisa la propuesta que se quiere recomendar a la empresa, que si bien se ajusta a las especificaciones no significa que se tenga un éxito asegurado ya que para un trabajo de investigación sea un elemento importante se necesita asegurar el contacto indispensable a los usuarios para entender la idiosincrasia de los proceso de la empresa, porque, no es lo mismo implementar un módulo de venta en una empresa que tiene su actividad económica a nivel básico que solicita la normal legal o tributaria, como otros que tienen que tener todos los aspectos legales y tributario cumplidos para obtener un margen de clientes más cualificados para tu negocio, es decir, no es lo mismo una empresa a nivel local que un trasnacional que tiene muchos más procesos.

² LA ROSA, Héctor. *Propuesta de Diseño de un Sistema de Información para el Seguimiento de Ventas para una Industria de Manufactura*, Venezuela, Caracas, 2004.

Salinas, O. et al (2015), en su tesis “Estrategia de Actualización de Tecnología de la Información (MATI) Énfasis en Gestión de Servicios y Productos TIC”³ de la Universidad Nacional de Costa Rica en Heredia – Costa Rica en el año 2015, este trabajo de investigación tiene como iniciativa dar una solución a la problemática de los procesos PADSÍ, según a las necesidades que solicita la institución que debe estar limitado o parametrizado por el PETIC (Plan Estratégico de Tecnologías de la Información), para los usuarios. Integrar una tecnología de la información se tiene que realizar estudios mediante encuestas para establecer variables y dar un diagnóstico preciso de acuerdo al análisis, pero para conocer los distintos factores implica tener fuentes para conformar los fundamentos de la propuesta que permitirá solventar los diagnósticos cumpliendo el objetivo general y específicos. Se conoce que su propuesta daría muchas oportunidades a mejoras internas a través de la planificación, integración, metodología y el ciclo de vida del software, para mejorar y aprovechar las nuevas tendencias de la tecnología, además que se sugiere recomendaciones para mejorar la efectividad de la situación actual de la institución para garantizar las mejoras de manera continua para que se pueda adaptar a los tiempos que corren. Se determina que la implementación de una propuesta de solución, justifica por medio de un plan piloto que apertura cambios para tener que incorporar las tecnologías de información, por consecuencia mejora las herramientas de trabajo para el desarrollo de un proyecto que se escalable en el futuro, de la misma manera se recalca que no se cumple en su totalidad todos los casos, pero haciendo una retroalimentación del asunto del proceso se mejora más el rendimiento del personal, además se cita los siguiente del autor de este mencionado trabajo, **“Se puede concluir, de forma general, que la adopción de nuevas tecnologías implica además de los temas descritos anteriormente, un cambio de cultura organizacional que va más allá de solo capacitar a las personas, el éxito de todo cambio requiere de un compromiso por parte de las personas, de una identificación con los objetivos del proyecto, ningún proceso de cambio se puede realizar sin las personas que lo apoyen”**⁴.

³ SALINAS, Obed. *Estrategia de Actualización de la Plataforma de Desarrollo de Sistemas de Información, en el Proceso de Administración y Desarrollo de Sistemas Institucionales (PADSI)*, INA, Costa Rica, Heredia, 2015.

⁴ *Ibíd.* pág. 126. La negrita es propio.

- **Nivel nacional:**

Gutiérrez, R. et al (2018), en su tesis “Propuesta de Implementación de Plataforma de Negocios de TI Online para Optimizar Tiempo y Costo en Cotizaciones”⁵ de la Universidad San Ignacio de Loyola de Lima – Perú en el año 2018, se realiza un estudio que tiene como finalidad dar la pautas del cómo sería una propuesta para la implementación de una plataforma que incorpore IT para la mejora de la Sales Specialists, aumento de la productividad y reducción de los costos, para que estos índices den un incremento de beneficio de utilidad en la empresa.

Además, este proyecto debe dar un diagnóstico claro de cómo se hacen los procesos manuales de cotización en la empresa, esto muestra o evidencia una baja productividad de dicho proceso, pero mediante un análisis de la información que se obtuvo por medio de la encuesta, para saber la realidad de cómo la empresa realiza sus procesos y administra sus actividades principales, teniendo esto en cuenta se da un estudio para proponer la mejora de optimización de tiempo y costo de cotizaciones. El proceso de análisis debe proponer varias mejoras al proceso de negocio de la empresa, esto equivale a decir, que debe mejorar los márgenes de tiempo, costo, productividad y el margen de utilidad mediante un enfoque cuantitativo, para dar a conocer como se realiza la medición de las variables con pre y post test para una estadística descriptiva que ayude observar los índices de mejora con la propuesta implementada de una plataforma de negocios de TI online.

Después del estudio y análisis de la información obtenido se tiene como resultado que la implementación de esta propuesta apoyaría mucho al proceso de negocio para reducir el tiempo para realizarlo, además optimizaría el costo de las cotizaciones. Demostrando que es un trabajo de investigación que menciona semejanzas de con la actual tesis constituida, ya que tiene una propuesta que luego es estudiada y analizada para mejorar o dar solución a una problemática que involucra la organización, además que tiene el mismo de enfoque de trabajo de investigación.

⁵ GUTIÉRREZ, Raphael. *Propuesta de Implementación de Plataforma de Negocios de TI Online para Optimizar Tiempo y Costo en Cotizaciones*, Perú, Lima, 2018.

Fermin, L. y Ingaruca, C. et al (2020), en su tesis, “Propuesta de Diseño Web y Móvil para la Gestión de Incidentes y Accidentes para una Empresa del Sector Minero”⁶ de la Universidad Peruana de Ciencias Aplicadas de Lima – Perú en el año 2020, en este trabajo de investigación se da en mención por motivo que da una propuesta que permita automatizar procesos de una gestión organizacional mediante una implementación de tecnologías web y móvil que ayudaría a mejorar el control y el seguimiento de la seguridad. Para esta propuesta planteada por los autores, toma como objetivo de estudio a la empresa Southern Peaks Minig, que tiene como objetivo importante cumplir con las políticas de seguridad que integra dicha empresa, pero para implementar esta propuesta se desarrolla la arquitectura de mejora de procesos de “Gestión de Incidentes y Antecedentes Ocupacionales” que se alinean con los puntos del PMBOK para tener un estudio con bases suficientes para justificar el desarrollo con las siguientes fases; inicio, planificación, ejecución y cierre del proyecto pero por cada fase hay entregables que hay que cumplir para hitos o la medición del alcance del proyecto, además que me menciona la identificación mediante caso de flujos de procesos para visualizar como es el proceso de seguimiento y control de seguridad dentro de la organización que en la actualidad en que se hizo el trabajo de investigación no presenta ningún incidente de dentro de sus instalaciones, por lo tanto esta propuesta tiene que mejorar a la identificación de incidentes a futuro o si hay un incidente como se debería proceder para atender y solucionar la problemática.

Los puntos de referencia de este trabajo de investigación para su análisis es tener un objetivo que se acople o se adapte a las políticas de la empresa que mediante las descripciones o pautas que los autores mencionan con los lineamientos del PMBOK, que esta descrito por su plan de alcance o las métricas que contiene este trabajo de investigación, además que muestra los procesos mediante por casos de uso y diagramas de flujo para que luego se aplicado para las tecnología web y móvil que apoyaría mucho al análisis previo a la implementación del proyecto y proporcionaría a los interesados como sería las interacciones de los usuarios con el sistema.

⁶ FERMIN, Luis y INGARUCA, Cesar. *Propuesta de Diseño Web y Móvil para la Gestión de Incidentes para una Empresa del Sector Minero*, Perú, Lima, 2020.

- **Nivel local:**

Ríos, P. y Vela, T. et al (2019), en su tesis, “Propuesta de la Aplicación Móvil Basada en Ubicaciones de Zonas ante Desastres Naturales en Iquitos”⁷ de la Universidad Nacional de la Amazonía Peruana (UNAP) de Iquitos – Perú en el año 2019, en este trabajo de investigación los autores tiene como objetivo principal desarrollar un aplicativo móvil que permite a la población tener información de zonas de ubicación de desastres naturales, pero como solo es una propuesta de investigación, el diseño de tesis es no experimental para que cuyos resultados planteados sean de acuerdo a los objetivos, además que usa la herramienta de la encuesta para la recolección de datos y posterior análisis. Resaltando que la población censada fueron 128 personas dando una proporción más clara a nivel de estadísticas descriptiva de los requerimientos a esta necesidad que en un futuro puede ser de conocimiento público ya que es el complemento de campañas de promoción realizas por el IIAP.

Manihuari, I. y Santillán, M. et al (2019), en su tesis, “Propuesta de Mejora para la Gestión de Información del Portal Web de la Municipalidad Distrital de Punchana – 2019”⁸ de la Universidad Científica del Perú (UCP) de Iquitos – Perú en el año 2019, en el mencionado trabajo de investigación los autores de quieren proponer una solución para mejorar la gestión de información en el portal del web de la Municipalidad Distrital de Punchana, ya que causa el problema de no encontrar la información oportuna para el ciudadano, además que le resulta muy difícil que brinde información contextualizada a los usuarios, pero en la situación actual en que se encuentra la página al momento de realizar este trabajo de investigación es el abandono del manteamiento y producción de actualización, por lo consiguiente se tiene que actualizar la información y adecuarlos a las normas públicas, en un resumen, el principal deber de esta investigación es promover la relación virtual entre la entidad pública con el ciudadano de a pie para ofrecer un mejor un servicio.

⁷ RÍOS, Paulo y VELA, Terry. *Propuesta de la Aplicación Móvil Basada en Ubicaciones de Zonas ante Desastres Naturales en Iquitos*. Perú, Iquitos, 2019.

⁸ MANIHUARI, Ingrid y SANTILLAN, Michael. *Propuesta de Mejora para la Gestión de Información del Portal Web de la Municipalidad Distrital de Punchana – 2019*. Perú, Iquitos, 2019.

2.2. Marco Teórico Conceptual

2.2.1. Ciclo de vida de desarrollo de software:

El desarrollo de un software o el ciclo de vida del desarrollo del software puede denominar como el procedimiento de cómo se realiza el desarrollo del producto de la tecnología de la información, mediante fases sucesivos que los mismos está compuesto por hitos que tienen que ser planeados para la conceptualización de la ejecución para aportar resultados a lo largo del proyecto y una posterior retroalimentación que permitirá optimizar los procesos básicos. En el ciclo vida del software tiene fases básicas para una futura implementación o actualización de un software, los cuales dependen de la metodología que consideran 3 etapas principales, que son los siguientes:

- **Planificación:**

Es el inicio del ciclo de vida del software, que da paso al análisis de los requerimientos que solicitan a base de las necesidades de la empresa, es decir, se tiene conocimiento de cómo es la idea del cliente para su solución, pero tomando en cuenta cómo serían esos procedimientos para obtener los resultados. En este punto se plantean los hitos o el alcance del proyecto para tener muy en cuenta para el siguiente punto.

- **Implementación:**

Una vez analizado los requerimientos o necesidades de la situación que tiene la empresa, se planea los hitos o alcances que se pasan a la siguiente fase, que es la implementación o ejecución, que se caracteriza por desarrollo del software mediante una metodología que posibilita las distintas formas de trabajar, sin embargo, en esta fase como se dijo se desarrolló lo planteado, pero sin tener mucha importancia las características técnicas para elaborarlo.

- **Despliegue:**

En esta fase de despliegue o producción, el sistema ya debe estar terminado y validado por las pruebas para evitar cualquier error de programación, es decir, es una de las últimas etapas del ciclo de desarrollo del software para distribución jerárquica del trabajo, pero lastimosamente en esta fase se puede evidenciar el éxito o el fracaso del proyecto.

Ya que depende de muchas circunstancias para la finalización, porque corre el riesgo de que el grupo de trabajo tenga muchos problemas para el desarrollo o que los interesados ya no tomen en cuenta el proyecto y lo abandone⁹.

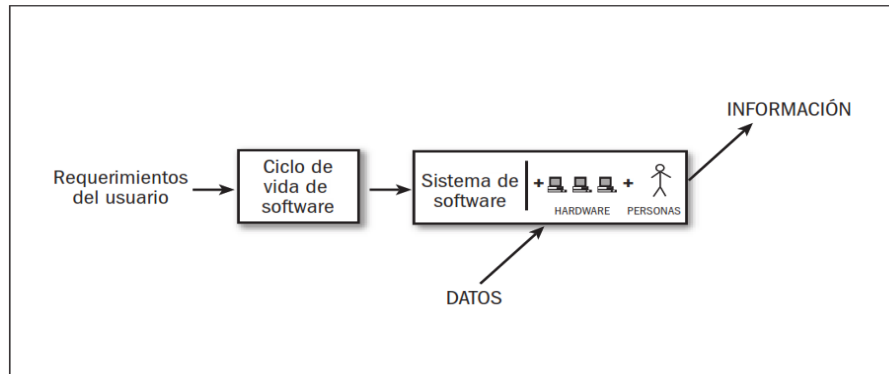


Gráfico N° 01 – Ciclo de vida de un producto software se desarrolla fuera del ámbito productivo.

Además de estas fases se añaden alguna que permiten que ciclo de vida del software sean más duradera, pero a veces las metodologías tienden a menospreciar u opcionales.

- **Inicio:**

En esta fase se da comienzo a una idea, que después precisa como se tiene que dar los objetivos y los recursos utilizados para la ejecución, además de mencionar las características implícitas o explícitas que tiene cada proyecto para asegurar el éxito o finalización, pero esto no significa que el proyecto tenga al final un buen resultado, pues, dependerá mucho de la comunicación de cada fase y de la constante comunicación entre los desarrolladores y los interesados.

- **Control de producción:**

En dicha fase, se analiza los procesos más no los requerimientos, porque se da inicio a las acciones correctivas para subsanar malos resultados o definir bien el producto; se menciona estos puntos porque hay una desviación de los requerimientos originales, planteados en la fase de planificación, esto causa rediseño del proceso que no afecte al resto de fases, pero también se puede dar la corrección del producto.

⁹ CANTONE, Dante. *La Biblia del Programador Implementación y Debugging*, Creative Andina Corp., 2006.

- **Pruebas:**

Ya teniendo las funciones terminadas del desarrollo, se busca validar los procesos para que tengas una buena ejecución y así no presentar errores en la etapa de producción, entonces para la fase de prueba se tiene que mucha importancia a la funcionalidad que está orientado, ya que durante la creación del proceso se busca solamente cumplir lo planteado, pero con las pruebas se busca tener un buen funcionamiento de lo desarrollado.

- **Documentación:**

En esta etapa depende mucho de que se orienta el trabajo, ya que, algunas metodologías desarrollan el software conjuntamente con la documentación, mientras que otros realizan toda la etapa de desarrollo para luego hacer los comentarios en el código y la documentación para descripción conceptual del funcionamiento del sistema, pero en ambos casos es indispensable hacer la documentación para basar alguna directivas o manual de uso de lo desarrollado.

- **Mantenimiento:**

En esta última fase se busca alargar el ciclo de vida del software mediante mejorar que se obtiene por la retroalimentación o las recomendaciones que va realizando los usuarios, para que así el software tenga más funciones integradas y pueda ser más elemental para el proceso de negocio de la empresa, sin embargo, esto va depender del tipo de desarrollo que haya planteado.

Estas etapas antes mencionadas como se dijo puede ser opcionales, pero pueden muy útil de acuerdo al modelo de ciclo de vida de quiera ejecutar¹⁰.

¹⁰ Ibíd. pág. 18.

Objetivo de cada etapa

En el ciclo de vida del software, tiene etapa en el cual se establece los hitos o actividades, ya que estos puntos son primordiales para el análisis del modelo determinado que está utilizando el proyecto¹¹, los cuales son:

- **Expresión de necesidad:**

El objetivo de esta etapa es la construcción de un documento que contenga los requerimientos y las finalidades que el sistema ofrece al usuario para su desarrollo, es decir, lo que se va implementar de acuerdo a las necesidades del usuario.

- **Especificaciones:**

Se puntualiza los requerimientos solicitados, de acuerdo al documento de la etapa anterior para tomarlo como punto de inicio.

- **Análisis:**

Se propone los elementos a desarrollar del sistema, los cuales son, las funcionalidades, las estructuras, las relaciones y su evolución temporal. En este punto, se tiene una noción evidente de cómo se va a desarrollar o construir el sistema.

- **Diseño:**

En esta etapa tenemos el conocimiento que debemos que hacer, pero ahora tenemos que plantearnos como se realizara el desarrollo del sistema, que factores se utilizaran y que funciones técnicas se tomaran en cuenta.

- **Implementación:**

Es la parte práctica de todas las etapas, ya que, se empieza a codificar los algoritmos y se construye las estructuras de los datos, además que se decide que lenguaje se va usar y con qué gestor de base de datos de suministrar el sistema, pero todo esto va acorde con las etapas de especificación, análisis y diseño para cumplir los objetivos.

¹¹ Ibíd. pág. 19.

- **Debugging:**
Esta etapa contiene las observaciones del programa realizado para evitar error o malos diseños de codificación, es decir, aquí no se busca cumplir el requerimiento (esto se realiza en la etapa de implementación), si no, encontrar la mayor cantidad de errores en tiempo de ejecución.
- **Validación:**
El objetivo de esta etapa busca verificar que el sistema desarrollado cumple con los requerimientos planteados inicialmente para que así, garantizar el buen funcionamiento del programa (mayormente la validación y el debugging están estrechamente relacionados).
- **Evolución:**
En resumidas palabras es la retroalimentación que se le da al sistema para subsanar errores, mejorar u optimizar algunas funcionalidades e integrar nuevas mejoras.

Modelos de ciclos de vida

Los distintos ciclos de vida se pueden diferenciar por 3 grandes percepciones¹².

- **El alcance del ciclo de vida:** depende de los límites planteados del proyecto, para tener conocimiento si es viable el desarrollo del producto.
- **La calidad y cantidad de etapas:** especifica en que dividimos el ciclo de vida, según el tipo y el proyecto adoptado.
- **La estructura y la sucesión de las etapas:** Es la retroalimentación entre las etapas.

Ciclo de vida lineal:

En este modelo de ciclo de vida, se descomponen las actividades de manera global para luego ser alineadas de manera lineal para ser realizadas una etapa a la vez, es decir se tiene que cumplir la etapa anterior para realizar el siguiente.

En este modelo cada actividad de una etapa es independiente entre sí, por lo que no se aplica retroalimentación entre ellas, aunque se puede aplicar correcciones en circunstancias puntuales, ya que se debe tener conocimiento de lo que se requiere antes de pasar a la siguiente etapa. Lo que resalta de este modelo es la poca información que requiere del cliente, ya que es muy sencilla su gestión y se acomoda a proyectos internos o muy pequeños. Pero no se recomienda para desarrollos que requieren retroalimentación por motivo que puede consumir muchos recursos¹³.

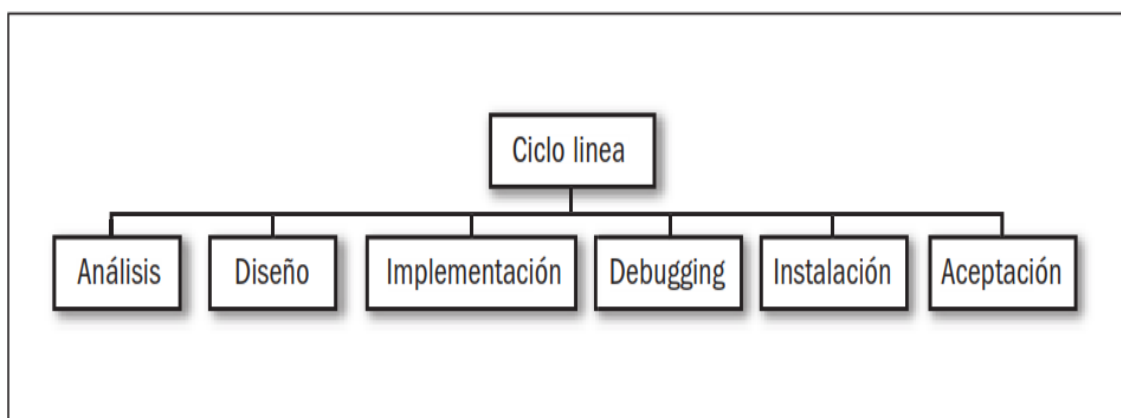


Gráfico N° 02 – Ciclo de vida lineal.

¹² *Ibíd.* pág. 21.

¹³ *Ibíd.*

Ciclo de vida en cascada puro:

En este modelo de ciclo de vida, planteado por Winston Royce en el año 1970, se caracteriza por admitir interacciones entre etapas o también se puede hacer revisiones después de cada etapa para verificar si se puede seguir con la siguiente etapa. Aunque contrae el modelo de ciclo de vida lineal, este modelo también es poco flexible y contiene muchas restricciones. La ventaja que tiene este modelo de ciclo de vida es tener un producto con alta calidad de desarrollo, sin tener personal altamente cualificado, además que su planificación es sencilla, pero si se comete un error no se detecta en la etapa siguiente, por que llegar consumir muchos recursos y es dificultoso subsanar errores¹⁴.

Lo adecuado para este modelo de ciclo de vida sería tener disponible desde el comienzo del proyecto todos los requerimientos para ser analizado y tener una buena finalidad de desarrollo del proyecto para evitar subsanar errores y que el proyecto tenga cambios no planificado.

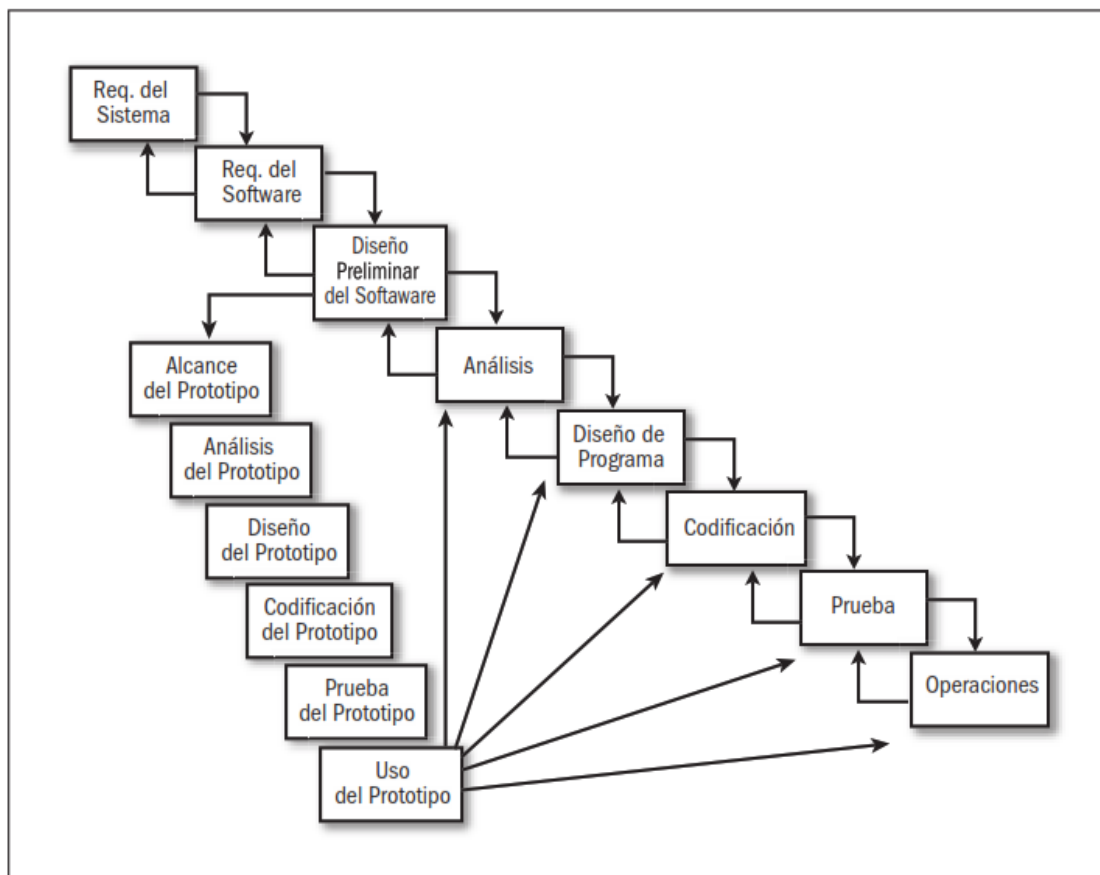


Gráfico N° 03 – Ciclo de vida en cascada puro.

¹⁴ Ibíd. pág. 23.

Ciclo de vida en V:

En este modelo de ciclo de vida, propuesto por Alan Davis se tiene las mismas etapas que el ciclo de vida de cascada puro pero la diferencia es que tiene subetapas de retroalimentación entre dichas etapas, para tener un análisis y seguimiento entre los objetivos de diseño y debugging. Las ventajas y desventajas son los mismo que el ciclo de vida de cascada puro, pero agrega controles entre etapas para lograr mejor corrección. Se puede usar este modelo cuando son aplicaciones simples que contenga pequeñas transacciones de base de datos¹⁵.

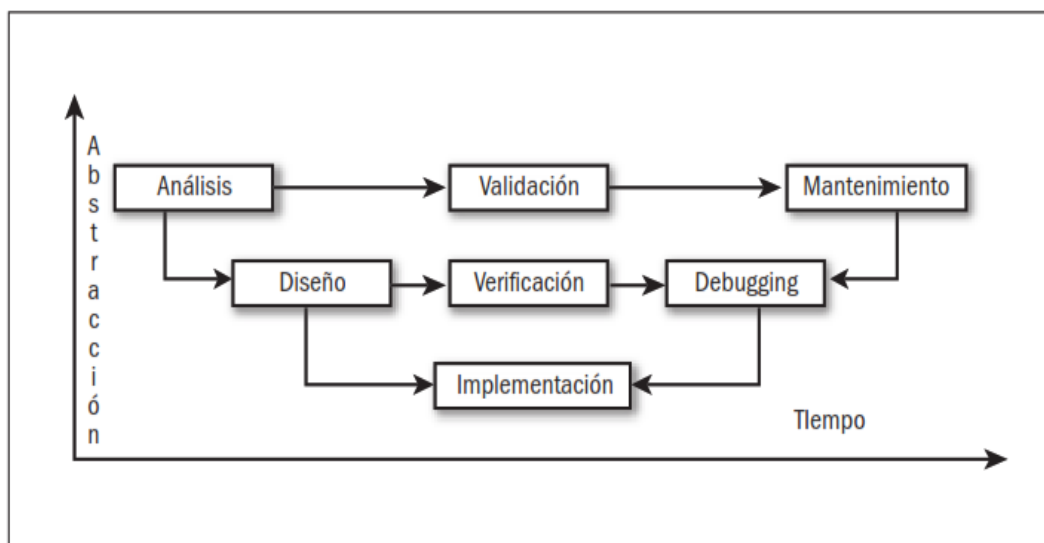


Gráfico N° 04– Ciclo de vida en V.

Ciclo de vida tipo Sashimi:

En este modelo de ciclo de vida, es similar al ciclo de vida de cascada puro, pero con la diferencia de sobreponer etapas, ya que este tipo de ciclo de vida aplica en muchos casos para aumentar eficiencia y la retroalimentación entre etapas, esto genera más calidad en el producto, pero su desventaja es también sobreponer las etapas por motivo que es difícil gestionar el inicio y el final de cada etapa, además de los problemas de comunicación¹⁶. Este modelo se puede aplicar a proyecto que comparten recursos para un ambiente productivo.

¹⁵ Ibíd. pág. 24.

¹⁶ Ibíd. pág. 25.

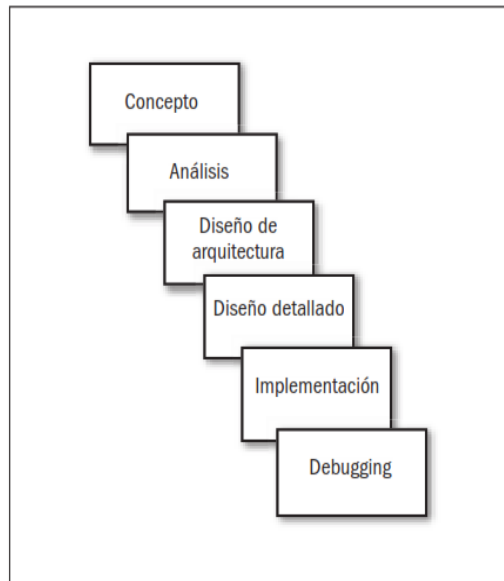


Gráfico N° 05 – Ciclo de vida tipo Sashimi.

Ciclo de vida en cascada con subproyectos:

En este modelo de ciclo de vida, cada cascada se divide por subetapas independiente de una con la otra, pero con desarrollos paralelos que permiten que más personas pueda trabajar al mismo tiempo, con la negativa que pueden surgir dependencia entre distintas subetapas que paralizen el trabajo grupal temporalmente¹⁷.

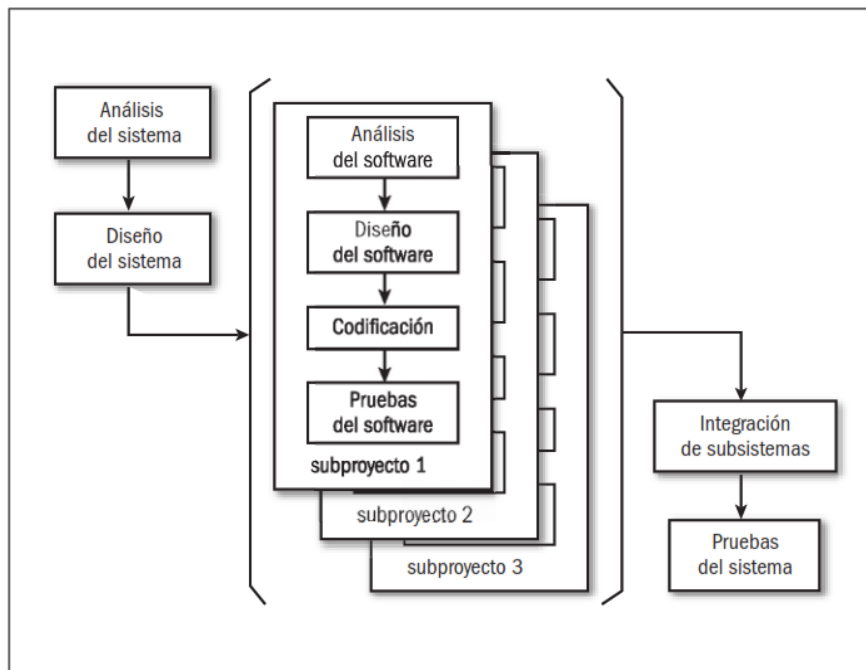


Gráfico N° 06 – Ciclo de vida en cascada con subproyectos.

¹⁷ Ibíd. pág. 26.

Ciclo de vida iterativo:

En este modelo de ciclo de vida, tiene relación con el ciclo de vida de cascada puro, aunque este modelo está enfocado en reducir el riesgo entre el interesado con el producto final, por una confusión en la solicitud de requerimientos.

Se gestiona a base de versiones, los cuales se mejoran de acuerdo a las interacciones de varios ciclos de vida en cascada, para que luego el cliente después de cada interacción evalúe el producto y pueda dar mejoras (las interacciones se repetirán hasta que el cliente quede satisfecho).

Este tipo de ciclo de vida se puede aplicar a proyectos que no tienen una buena obtención de requerimientos, para que luego se pueda dar distintas propuestas hasta conseguir la conformidad del cliente, es decir, es bueno en aplicaciones de medianas a grandes dimensiones, en el que los interesados recogen las propuestas hasta que cumplan todas las funcionalidades¹⁸.

Ejemplos:

- Migración de aplicaciones a otras arquitecturas de software.

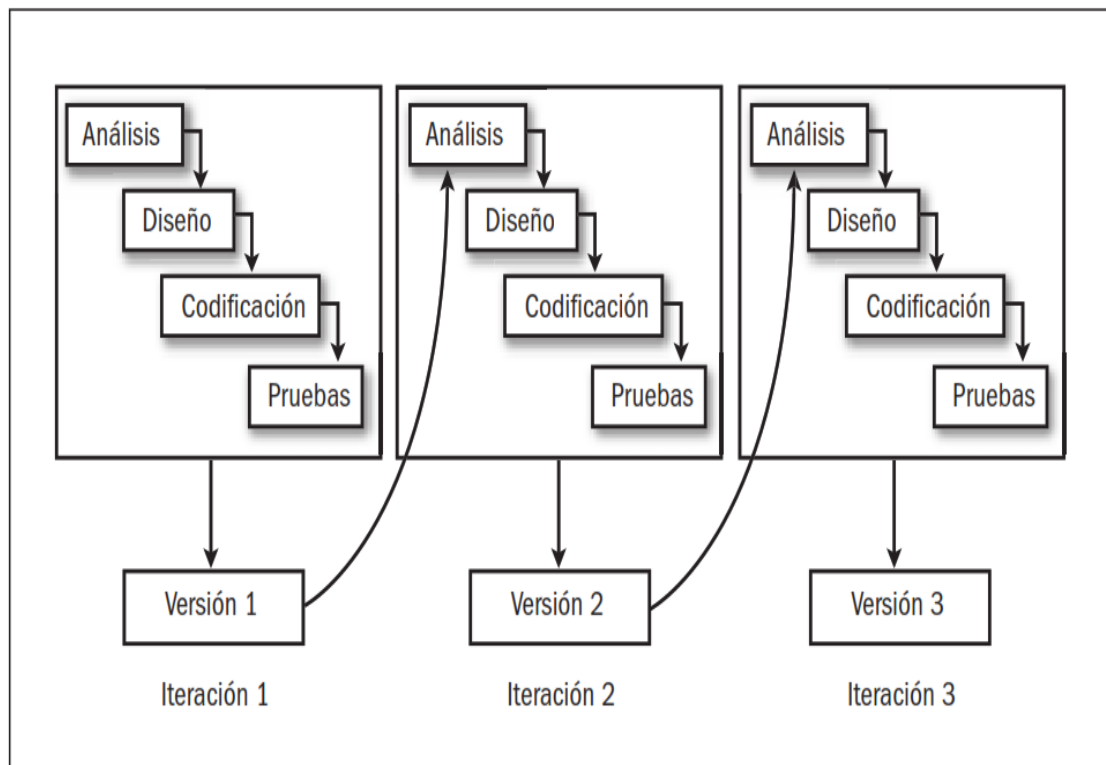


Gráfico N° 07 – Ciclo de vida iterativo.

¹⁸ Ibíd. pág. 28.

Ciclo de vida por prototipos:

En este modelo de ciclo de vida, utiliza el uso de programa prototipos para validar o cumplir los requerimientos solicitados por el usuario. Se desarrolla un determinado producto para cumplir las especificaciones básicas o precisas, para crear un prototipo para que tenga una funcionalidad provisiones, para luego realizar el producto final con las funciones que se dieron a conocer mediante el prototipo.

Pero este modelo de ciclo de vida llegar ser demasiado costoso y dificultoso de gestionar en un grupo de trabajo temporal, sin embargo, las aplicaciones con nuevas tecnologías permiten crear productos con una alta proporción de I+D o innovación, con especificaciones no muy claras que a veces puede ser de incertidumbre en el resultado, ya que se tiene el desconocimiento del éxito o fracaso del proyecto mediante este ciclo de vida¹⁹.

Mayormente se utiliza cuando se migra de una tecnología a otra para adoptar una aplicación.

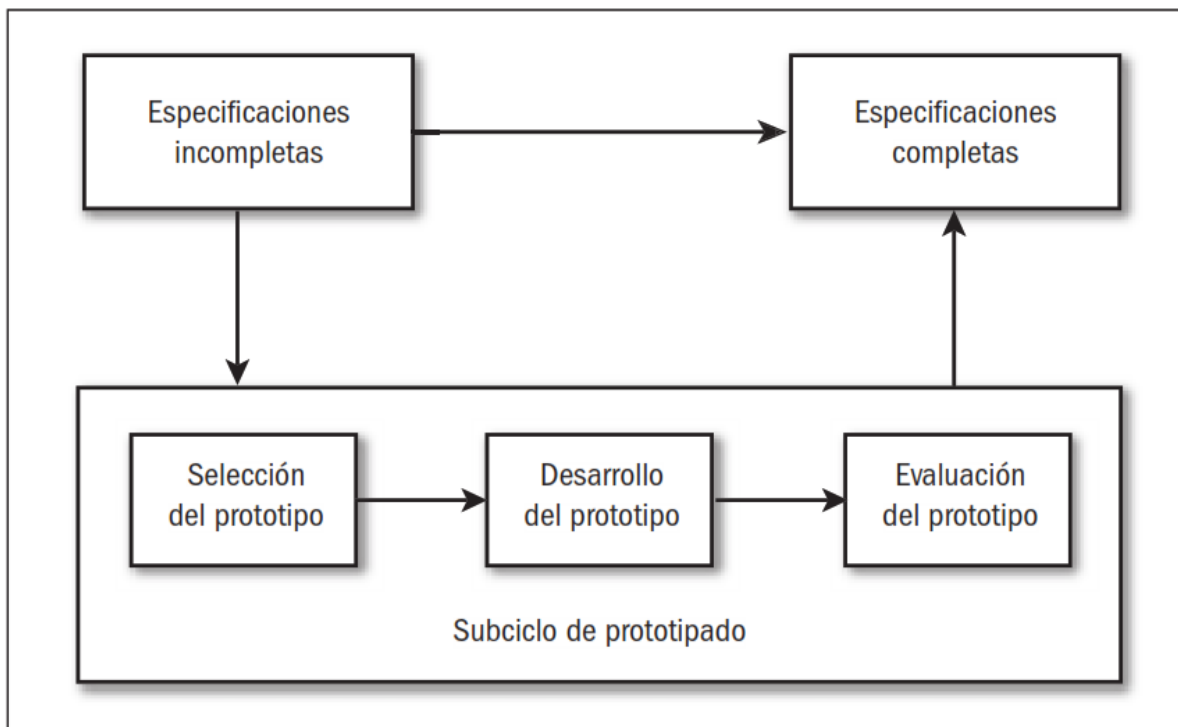


Gráfico N° 08 – Ciclo de vida por prototipos.

¹⁹ Ibíd.

Ciclo de vida evolutivo:

En este modelo de ciclo de vida, demuestra flexibilidad que puede tener el proyecto, por motivo que es muy difícil tener todos los requerimientos al inicio del proyecto, peor aún, tener una clara perspectiva desde un enfoque del usuario, porque, es dificultoso entender las ideas que desea transmitir, entonces, este modelo afronta esta problemática con interacciones de acuerdo al ciclo de; requerimiento–desarrollo–evaluación. Es muy eficaz este modelo cuando no tiene en claro todos o la mayor parte de los requerimientos iniciales, además que apoya mucho a la gestión de los cambios o nuevas solicitudes del usuario²⁰. Por ejemplos:

- Cuando se plantea centralizar la base datos de varios módulos de un negocio que tiene su sistema autónomo.

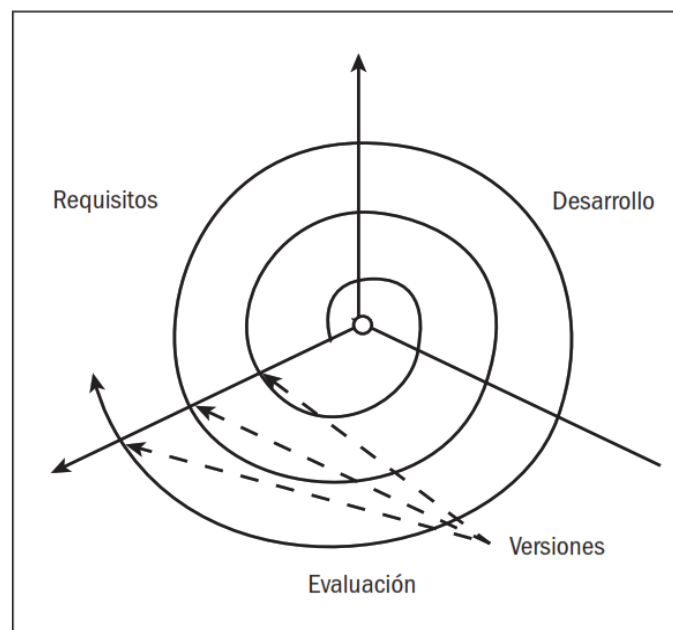


Gráfico N° 09 – Ciclo de vida evolutivo.

Ciclo de vida incremental:

En este modelo de ciclo de vida, tiene con fundamento de construir e incrementar las funcionalidades del software o proyecto por módulos que aumentan gradualmente capacidades del software, además que puede ser descentralizas en la tarea que puede dar a los miembros del grupo de trabajo.

²⁰ *Ibíd.* pág. 29.

Se conoce también como modelo cascada repetitivo, por motivo que se tiene que realizar ese ciclo mencionado para dar una versión entregable al usuario o cliente que contengan nuevas funcionalidades, pero este ciclo de vida tiene una particularidad, solo está pensado para ser útil para entregables rápidas, aunque también puede ser aplicable a cualquier proyecto, pero sin aprovechar al máximo esta modelo²¹.

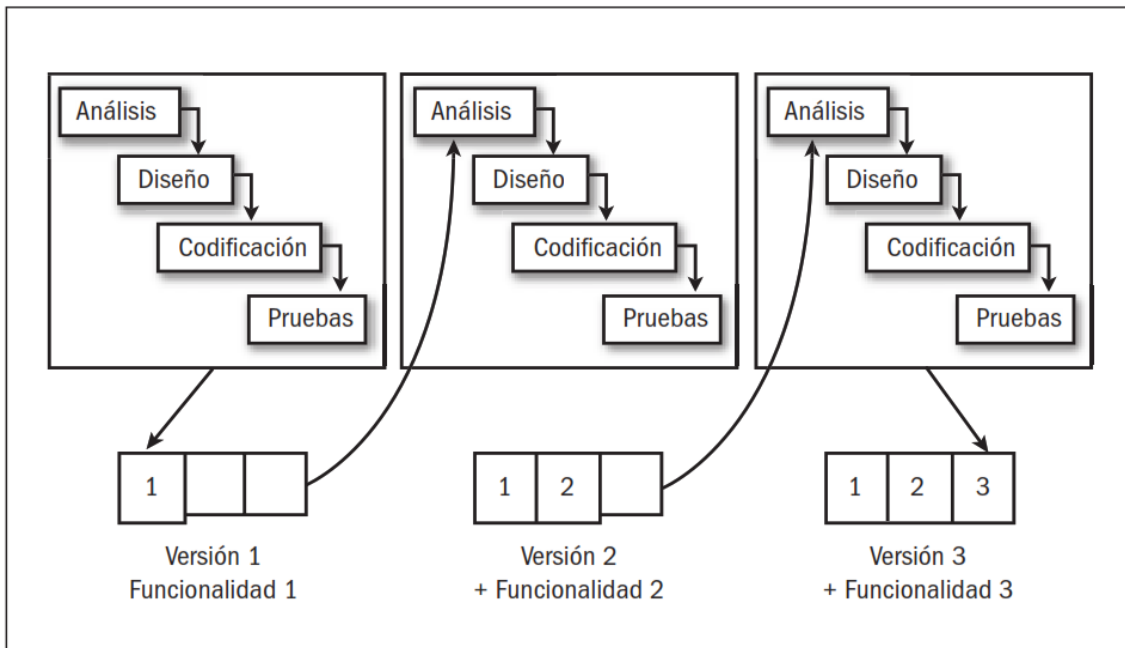


Gráfico N° 10 – Ciclo de vida incremental.

Ciclo de vida espiral:

En este modelo de ciclo de vida, fue diseñado por Boehm en el año 1988, cuyo propósito principal es la variación de acuerdo a una serie de ciclos repetitivos hasta llegar a la madurez del software que permite un producto final. Toma en cuenta los beneficios del ciclo de vida incremental y de prototipo, pero con el detalle que toma más prioridad en el aspecto de riesgo, debido al desconocimiento de los requerimientos en la fase inicial del proyecto. Mediante la trayectoria del ciclo (ejecución espiral), se va dando prototipos que van satisfaciendo las necesidades del usuario o cliente.

Desde un punto de partida en donde la fuente del desconocimiento del usuario se aprovecha para tener oportunidades que deben ser cumplidos con productos

²¹ *Ibíd.* pág. 30.

o entregables, que poco a poco va llegando a una etapa de madures transcurriendo por 4 actividades, los cuales son:

- Planificación: se hace la enumeración de los requerimientos iniciales o después de una interacción.
- Análisis de riesgo: toma mucha importancia los requerimientos para proseguir el desarrollo del producto.
- Implementación: se desarrolla el prototipo de acuerdo a los requerimientos.
- Evaluación: el cliente decide, si el entregable es conforme a su necesidad y finaliza el proyecto o, al contrario, rechaza el entregable y se continua con el proyecto.

Los beneficios en este modelo de se nota más en el desarrollo del software, ya que el proyecto puede iniciar con alto grado de incertidumbre de los requerimientos, además minimiza el riesgo de un retraso por casos de errores, ya que se soluciona en la próxima interacción de la espiral, pero no todo es beneficio en este modelo, ya que por cada espiral ejecutado hay un costo temporal que al final suma cada vuelta, además, que es dificultoso evaluar los riesgos y la necesidad que tienen el cliente o usuario. Es aplicable a grandes proyectos internos de una organización, ya que admite reclamos o solicitud de cambios de los requerimientos²².

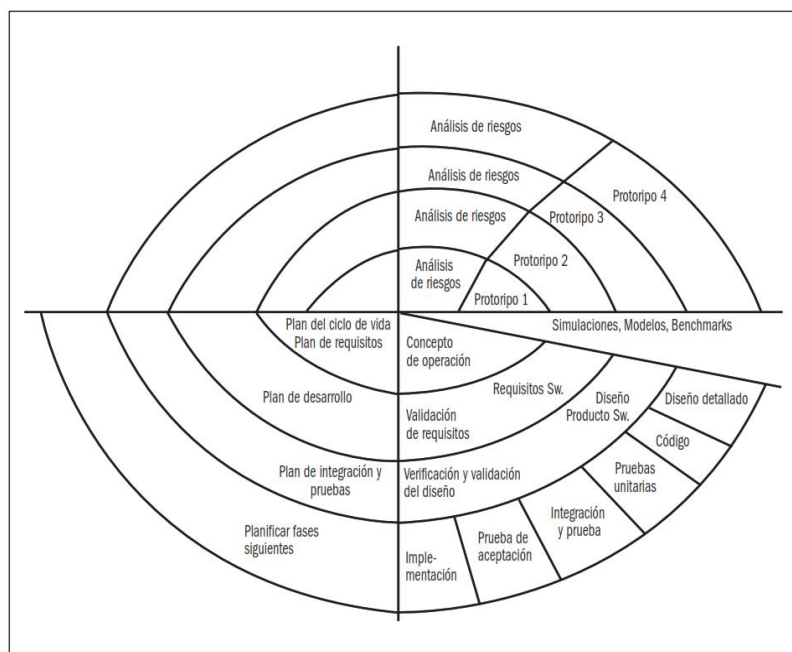


Gráfico N° 11 – Ciclo de vida espiral.

²² Ibíd. pág. 31.

Ciclo de vida orientado a objetos:

Fue presentada en los años 90 y es considerada una de las mejores para una implementación de un proyecto de software, por tener la filosofía del paradigma de programación orientada a objetos, ya que considera cada requerimiento del usuario como un objeto que tiene sus propiedades los cuales se caracteriza por tener sus atributos, mientras que el comportamiento es denominado como el método que tendrá el objeto. En un modo contextual, se define un objeto sobre caso de la vida real.

En este modelo de ciclo de vida orientado a objetos tiene como principal característica la abstracción de los requerimientos (objetos), es decir, es más flexible que lo demás modelos, que son difícil de adaptarse a los requerimientos del usuario, entonces tenemos como consecuencia que hay una reducción de la complejidad del problema o necesidad que se aborda. También aplica las fichas CRC (clase–responsabilidad–colaboración) como técnica de abstracción para analizar los requerimientos del usuario, cuyo mecanismo es, asignar un nombre al objeto, luego su comportamiento (métodos) y su dependencia (si necesita de otros objetos)²³.

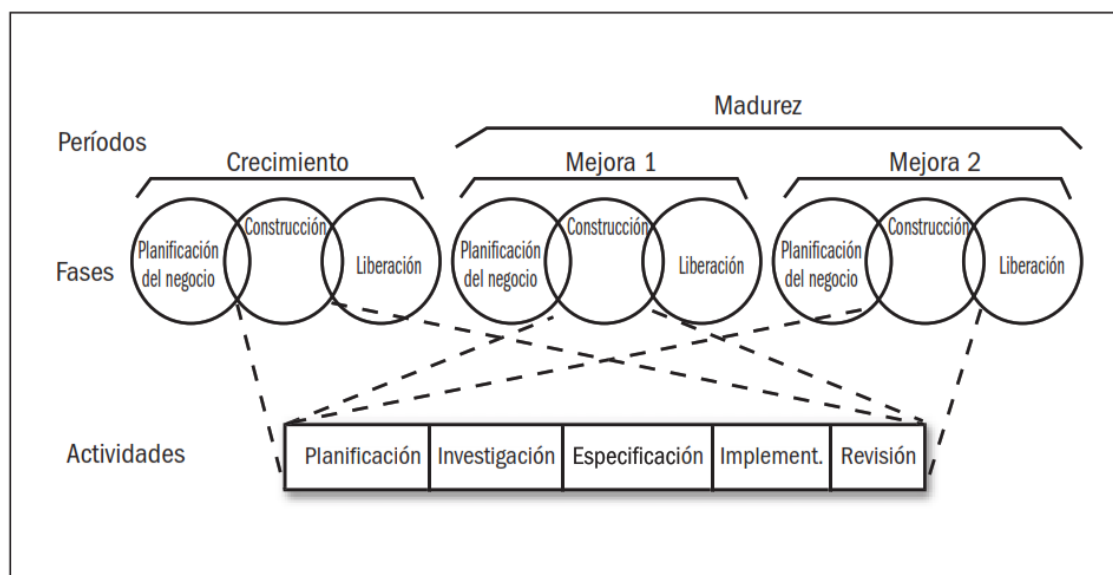


Gráfico N° 12 – Ciclo de vida orientado a objetos.

²³ Ibíd. pág. 33.

2.2.2. Metodología de desarrollo de software:

Se puede definir de muchas maneras, pero la principal característica es la organización sistemática para el ciclo de vida del software, pero además se puede entender como un conjunto de herramientas procedimientos y técnicos para lograr el desarrollo de un proyecto de software, es decir, es la abstracción de los procesos para esquematizar mediante etapas para la documentación, procedimientos, técnicos y herramientas para el proceso de negocio.

La metodología busca descomponer los requerimientos para que los desarrolladores identifique que tipo de técnicas se tiene que aplicar para su procedimiento y así dar información del sistema en construcción al cliente. Se hace mención de los objetivos de la metodología del desarrollo del software:

- Establecer puntualmente con los requisitos de un sistema software.
- Proporcionar procedimientos sistemáticos de desarrollo para la gestión y control de los procesos.
- Conformar un sistema de software dentro de las especificaciones previamente establecidas.
- Busca dar toda la información del sistema a los interesados, mediante la documentación y los comentarios de programación de los desarrolladores.
- Detalla las problemáticas, para la identificación los antes posible los requerimientos o posibles cambios que solicite el usuario.

La metodología de acuerdo a la definición de algunos autores:

Según Pressman (2010) menciona lo siguiente:

“Representa el proceso del software de manera esquemática, cada actividad estructural está formada por un conjunto de acciones de ingeniería de software y cada una de éstas se encuentra definida por un conjunto de tareas que identifica las tareas del trabajo que deben realizarse, los productos del trabajo que se producirán, los puntos de aseguramiento de la calidad que se requieren y los puntos de referencia que se utilizarán para evaluar el avance”²⁴.

²⁴ PRESSMAN, Roger. *Ingeniería del software un enfoque práctico*, Mc Graw Hill, 2010, séptima edición. La negrita es propia.

Según Sommerville (2002) menciona lo siguiente:

“Una representación simplificada de un proceso de software, representada desde una perspectiva específica. Por su naturaleza los modelos son simplificados, por lo tanto [Sic.] un modelo de procesos del software es una abstracción de un proceso real”²⁵.

Según Pérez (2011) menciona lo siguiente:

“Los marcos tradicionales imponen una planificación rígida y meticulosa del proyecto, soportada por herramientas y una carga de trabajo pesada en planificación, diseño y documentación, en un afán por hacer al desarrollo predecible dentro de un marco de temporalidad y costo; recordando que derivan de solventar los problemas de la crisis del software vivida décadas atrás. Las más difundidas y citadas en textos son: Rapid Application Development (RAD) [Sic.], Rational Unified Process (RUP) [Sic.] y Microsoft Solution Framework (MSF) [Sic.]”²⁶.

Rapid Application Development (RAD). Fue presentado por James Martin desde 1980, es la expresión del desarrollo rápido de las tareas o hitos, descomponiendo el trabajo orientada a la rapidez, aunque no es considerado parte de la metodología ágiles, buscar dar una solución a la necesidad del cliente de manera rápida. Involucra el desarrollo bajo el modelo iterativo, fabricando los entregables por medio de prototipos con la finalidad de aumentar la productividad.

Rational Unified Process (RUP). Establecidos por Ivan Jacobson, Grady Booch y James Rumbaugh. Se fundamenta a base del modelo cascada, ya que consiste en los casos de uso para describir el servicio que el usuario requiere del sistema, además que toma en cuenta las interacciones entre usuario – sistemas. Esta metodología se centra con la arquitectura que toma en cuenta en el proyecto, respetando las pautas y especificaciones para la composición del equipo del trabajo y los tiempos de cada entregable, es decir, es iterativo e incremental. Es considerada como parte de las metodologías clásicas más usadas para la

²⁵ SOMMERVILLE, Ivan. *Ingeniería de Software*, Pearson Educación, 2002. La negrita es propia.

²⁶ PÉREZ, Oiver. *Cuatro Enfoques Metodológicos para el Desarrollo de Software RUP – MSF – XP – SCRUM*, Revista Inventum No. 10, 2011, pág. 64-78. La negrita es propia.

aplicación del análisis, desarrollo y documentación de un sistema orientados a objetos, es muy beneficioso en proyecto de gran magnitud.

Microsoft Solution Framework (MSF). Es el conjunto de buenas prácticas recomendadas por Microsoft y Microsoft Consulting Services para desarrollo de aplicaciones de software, ya sea de manera individual o grupal porque sus componentes se adaptan, de igual manera se puede decir que es escalable durante el tiempo de vida o magnitud del proyecto²⁷. La metodología de desarrollo de software es una base muy importante para tener un proyecto productivo y eficaz, pero existen 2 grandes grupos de metodología de desarrollo de software, los cuales son; los ágiles y tradiciones.

Metodologías ágiles de desarrollo de software, es la que se impuesta en la última década, ya que mantiene las últimas tendencias de programación y así también como la organización de equipos de trabajo.

Metodologías tradicionales de desarrollo de software, siguen siendo válidas para el desarrollo de un proyecto, más un, en el ámbito de programación, porque es imprescindibles para generar productos de alta calidad.

Metodología Ágiles	Metodología Tradicionales
Consisten en heurísticas de buena producción de código.	Consiste en normas estandarizadas en el entorno de desarrollo.
Hace hincapié en el trabajo individual o grupal.	Hace hincapié en los procesos: roles, actividades y artefactos.
Flexible para la solicitud de cambios en el proyecto.	Llegar ser rígido a ciertos cambios del proyecto.
Los procesos tiende a tener pocos controles.	Los procesos tienden a tener muchos controles y seguimientos por las normas.
El usuario y/o cliente es parte del desarrollo.	El usuario y/o cliente interactúa con el grupo de trabajo mediante reuniones.
No tiene mucho protagonismo la arquitectura de software, se va realizando feedback y mejoramiento durante el tiempo (madures de software).	La arquitectura de software tiene un papel protagónico para el desarrollo del proyecto, se define en la fase inicial.
Es para grupos máximo de 10 personas, porque tiene pocos roles y es genérico.	Es para grupos grandes, aglomeran muchos roles y más especificaciones.
Orientados a proyectos de magnitud pequeñas y se ejecuta en el mismo lugar de desarrollo.	Es aplicable a proyectos de cualquier tamaño, pero suelen ser efectivas y aplicables a proyectos grandes.

Tabla N° 01 – Cuadro comparativo de las metodología, elaboración propia²⁸.

²⁷ *Ibíd.*

²⁸ CANÓS, José, LETELIER, Patricio y PENADÉS, Carmen. Metodologías Ágiles en el Desarrollo de Software, España, Valencia, 2003.

El desarrollo de software pertenece en 2 grupos metodológicos, que son:

Metodologías de desarrollo de software ágiles

Las metodologías que pertenecen a este grupo tiene a ser utilizadas en proyectos de desarrollo de software que llegan a tener muchos cambios y son flexibles, pero de magnitudes más pequeñas para que el proyecto tenga un método incremental y un trabajo autosuficiente e independiente²⁹. Las principales metodologías ágiles son:

- **Kanban:** esta metodología inventada y desarrolla por la empresa de automóviles Toyota, consiste en dividir el trabajo por tareas mínimas para otorgar pendientes individuales que a la larga de curso del proyecto se va finalizando con entregables para crear el producto, esto crearía un flujo de trabajo con tareas prioritarias e incrementatelas para sumar valor al producto (por hacer –haciendo –hecho).
- **Scrum:** desarrollado por Ken Schwaber, Jeff Sutherland y Mike Beedle; definen que la gestión de los procesos del proyecto puede ser flexible con los cambios, pero sus principales características son 2, el cual el primero es el desarrollo mediante interacciones definidas por “sprints” que tienen duración de 30 días y la segunda es la importancia de las reuniones a lo largo del proyecto, lo cual destaca reunión diarias de 15 a 30 minutos para definir el desarrollo con todos los cambios y modificaciones integrados.
- **Lean:** está diseñado para que pequeños grupos de desarrollo elaboren tareas en muy poco tiempo, priorizando las personas y el compromiso u dejando en un segundo plano el tiempo y el coste.
- **Programación extrema (XP):** Se basa en relaciones interpersonales (clave del éxito), porque su objetivo es crear un buen ambiente laboral para que haya retroalimentación constante de los requerimientos de parte del cliente. El beneficio de esta metodología, es la práctica que se consigue con su aplicación, que al mismo tiempo integra de forma efectiva y complementaria las perspectivas de negocio, valores humanos y trabajo en equipo.

²⁹ Blog Santander Universidades, “Metodología de desarrollo de software: ¿qué son?”, 2020, <https://blog.becas-santander.com/es/metodologias-desarrollo-software.html> (consultada el 15 de mayo 2021).

Esta metodología consiste su trabajo en 12 conceptos los cuales son³⁰:

- 1) Planificación: debe haber una comunicación continua entre el cliente con los programadores, para estigmatizar el esfuerzo que va requerir las implementaciones de los entregables y las interacciones.
- 2) Entregas pequeñas: producir entregables del sistema que sean operativos, aunque no cuenten con toda la funcionalidad previamente planificada, este tipo de entregables solo tienen como máximo 3 meses elaboración.
- 3) Metáforas: se define el sistema por una o por conjuntos de metáforas comunicadas por el cliente al equipo del desarrollo. La metáfora es la forma o manera de describir cómo debería funcionar el sistema.
- 4) Diseño simple: se la solución lo más simple posible que pueda funcionar y ser integrado en un momento determinado del proyecto.
- 5) Pruebas: son las verificaciones de ejecución del código, dichas verificaciones o pruebas son establecidas por el cliente antes desarrollar el código; cada modificación de código debe ser probadas.
- 6) Refactorización: es la constante reestructuración del código para evitar la duplicidad y optimizar los procesos, al mismo tiempo haciendo más al flexible al código.
- 7) Programación en parejas: se realiza en parejas de programadores para una menor cantidad de error, mejor diseño y tener una mejor retroalimentación.
- 8) Propiedad colectiva del código: cualquier desarrollador del grupo de trabajo puede hacer modificaciones del código.
- 9) Integración continua: cada pieza de código se integra al sistema mediante módulos (entregables) que permiten construir varias veces el mismo día.
- 10) 40 horas por semana: se debe respetar este tiempo de trabajo por semana, ya que, si hay horas extra, significa que hay problemas internos.
- 11) Cliente in-situ: el cliente debe estar presente y disponible durante la ejecución del proyecto, es la principal característica del éxito del proyecto XP, porque priorizan mucho la comunicación.
- 12) Estándares de programación: XP enfatiza que los programadores sigan ciertos estándares de programación para que él sea entendible.

Fase del Proyecto	Planificación	Diseño	Codificación	Pruebas
Puntos de Conceptos	1, 2, 3	4	5, 6, 7, 8, 9, 10, 11	12

Tabla N° 02 – Cuadro que especifica los puntos que pertenecen en cada fase. Elaboración propia

³⁰ CANÓS, José, LETELIER, Patricio y PENADÉS, Carmen. Óp. Cit. Pág. 5 y 6.

Metodología de desarrollo de software tradicionales

Se define por su rígida manera de tener todos los requerimientos al inicio del proyecto, es decir, tolera nada o pocos cambios, por ese motivo es como la contraparte de las metodologías ágiles, ya que su ejecución es lineal por etapas y las sucesiones entre ellas, debe ser el término de una etapa para que inicie la otra, por lo que no permite superponer dichas etapas ni volver hacia atrás una vez finalizado la etapa³¹. Este grupo tiene sus metodologías, las cuales son:

- **Waterfall (cascada):** se caracteriza por la forma de organizar las etapas de arriba hacia abajo, es decir, se desarrolla sus funciones en etapas diferentes, obedeciendo rigurosamente el orden de cada etapa que va bajando una vez se haya terminado la etapa actual con su respectiva revisión. Un detalle muy importante es que los requerimientos y especificaciones planteadas al inicio no están predispuesto a ser cambiados, teniendo por consecuencia que los resultados sean visibles en una fase del proyecto muy avanzado.
- **Prototipado:** es la fabricación de un prototipo de software para los usuarios de manera rápida, pero no con todas las funciones planteadas al inicio, sino, que se agregando durante el tiempo o la retroalimentación. Esta metodología permite arreglar algunos errores del proyecto o integrar nuevas funciones que mejoren la experiencia de uso del usuario, por ese concepto el modelo iterativo se identificado con esta metodología ya que es el ensayo de prueba y error para llegar a un producto que cumpla con todas las especificaciones.
- **Espiral:** es la combinación de los 2 modelos mencionados anteriormente, incluyendo de manera característica el análisis de riesgo que contiene 4 etapas los cuales son; planificación, análisis de riesgo, desarrollo de prototipo y evaluación del cliente. Fundamenta su nombre por el tipo de funcionamiento o ejecución que realiza sus procesos, que es de forma espiral, por lo que significa, que, si estamos más al centro de la espiral, más avanzado está el proyecto.

³¹ Blog Santander Universidades, "Metodología de desarrollo de software: ¿qué son?", 2020, <https://blog.becas-santander.com/es/metodologias-desarrollo-software.html> (consultada el 15 de mayo 2021).

- **Incremental:** se la manera de cómo se va desarrollando el software hasta llegar ser un producto final, pero para el proceso incremental se va agregando funciones de acuerdo a las etapas para poder tener resultado lo antes posible en comparación de su contraparte que es el modelo cascada, es más, se puede utilizar el software, aunque no esté con el funcionamiento total, por lo que es el más flexible de todas las metodologías tradicionales.
- **Diseño rápido de aplicaciones (RAD):** se caracteriza en dar proyectos de software de alta calidad en un periodo corto de desarrollo, sin embargo, consume muchos recursos a comparación de las metodologías, pero se beneficia de tener flexibilidad al proyecto con una mayor interacción del usuario. Aunque, por otro lado, si nos metemos por el entorno técnico, los errores o bug de código puede ser frecuentes, entonces por consecuencia se limita las funciones del producto por disponer muy poco tiempo para subsanarlas.

Capability Maturity Model Integration (CMMI)

Más conocido como Integración de Sistemas Modelos de Madurez de Capacidades, permite la mejora y evaluación de procesos para el desarrollo, mantenimiento y producción de sistema de software. Aunque no se trate en una metodología en específico tiene un papel protagónico en la retroalimentación de las practicas del desarrollo de software con las organizaciones, ya que toma en cuenta los elementos esenciales de los procesos de negocio para desarrollar productos de software mucho más optimizados³², los beneficios son:

- Las relaciones explícitas que tiene los objetivos de negocio con la gestión de actividades, que a su vez están relacionadas con la ingeniería de los procesos que ayudan a facilitar el cumplimiento de estas.
- Permite adquirir mejores prácticas por la incorporación de la experiencia realizadas en otras áreas de estudio, tenemos como ejemplo; la gestión de riesgo y de proveedores.
- Cumplir con las normas ISO y las prácticas de alta madurez.

³² MegaPractical Soluciones de Negocio, "Metodología de desarrollo de software", 2021, <https://megapractical.com/cursos-de-capacitacion-en-linea/> (consultada el 16 de mayo 2021).

2.2.3. Arquitecturas de software:

Si tenemos que definir la arquitectura de software, antes, se tiene que entender el proceso de transformación de un desarrollo de software, desde la fase de inicio que indica los requerimientos, hasta la finalización que es el entregable, solución o producto.

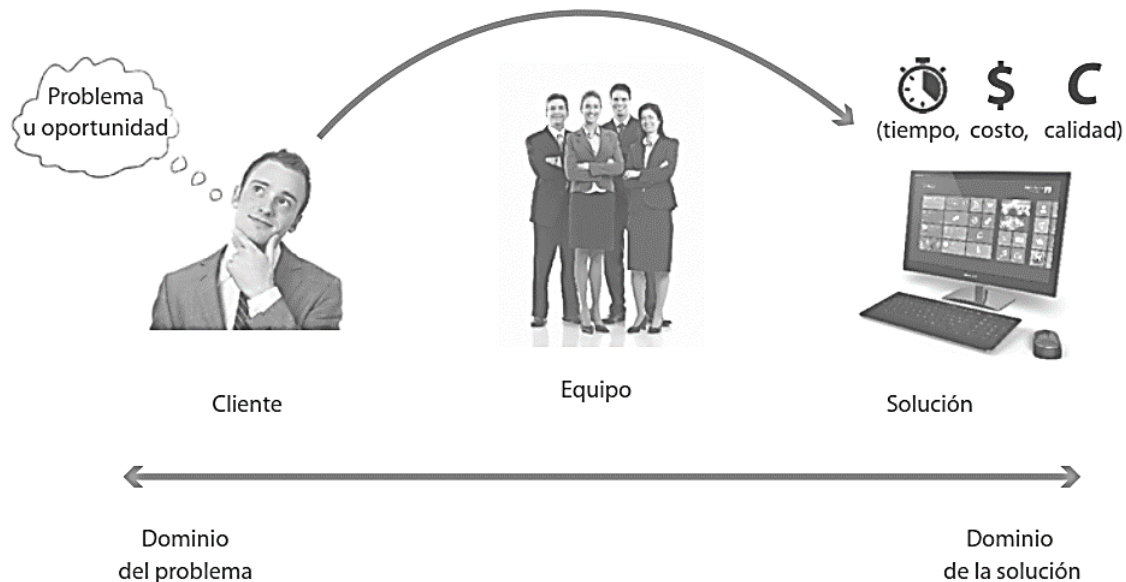


Gráfico N° 13 – Visión del desarrollo de software.

Pero para llevar a cabo estas fases hay actividades técnicas, las cuales son definidos y mostrado en el siguiente punto:

- **Requerimientos:** se identifica las necesidades de los interesados con la información suficiente para detallar las especificaciones de lo que debe ser el sistema.
- **Diseño:** se toma la decisión de cómo se resolverán los requerimientos establecidos anteriormente, es decir, identifica las partes del sistema para satisfacer las necesidades de los interesados, pero también se constituyen los grupos de desarrollo.
- **Construcción:** es el desarrollo del software, en otras palabras, es el ensamble de los entregables de cada grupo de desarrollo para que se integre al producto final, pero la estructura del diseño general del sistema completo no es igual al diseño de las partes, esto debe ser coordinado en el punto anterior.

- **Pruebas:** es la realización de determinadas pruebas o verificaciones de procesos para que requerimiento sea cumplido de manera totalmente satisfactoria, además que es validada para subsanar uno que otro error.
- **Implantación:** es la fase de transición del sistema a su entorno previamente programado, para su próxima entrada a producción.

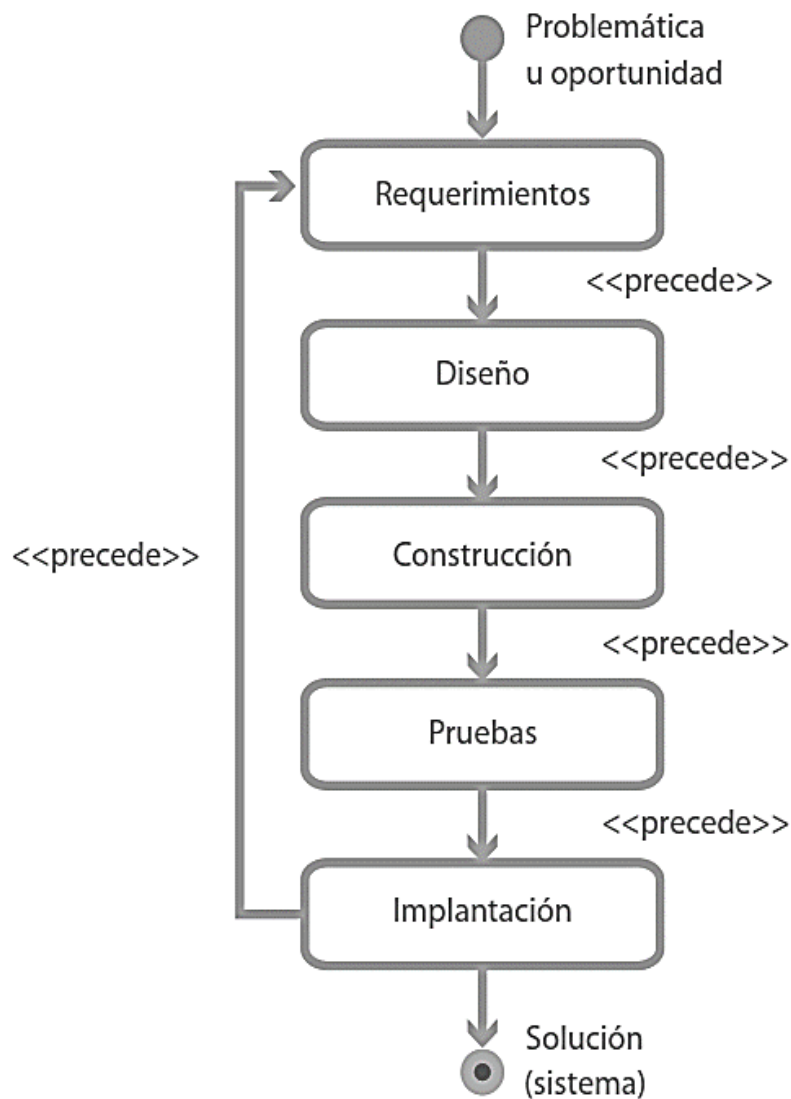


Gráfico N° 14 – Etapas del desarrollo de software.

No hay una definición exacta de arquitectura de software, por motivo que no hay una teoría completa o que descomponga todas las características de esta parte de la ingeniería de software, porque hay algunas que son como alternativas, mientras que otras son contrapuestas y no definen un bien común o universal.

Sin embargo, si lo vemos en perspectiva general, es la entremezcla de definiciones que dentro de los procesos de ingeniería de software se conoce como la conformación o topografía estática de sistema software para abstraer los procesos a nivel en el cual, contemplada las características como componentes³³.

Pero este tiene definiciones muy particulares las cuales son:

La definición según Clements (1996):

“La AS es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alta nivel de compresión y la supresión (...)”³⁴.

Por otra parte, la arquitectura de software define “elemento”, la forma individual de una parte del sistema que se debe desarrollar, que en aspectos técnicos se conocen como módulos, pero por lo demás, aquellos elementos tienen como propiedades las interfaces que establecen dependencias para que los módulos puedan conectarse entre ellos (los elementos pueden tener varias definiciones).

La arquitectura también señala que el diseño que tenga un sistema no solo tiene que tener aspectos de relación de desarrollo simultáneos para grupos de trabajos individuales, sino, también tiene que cumplir con los requerimientos, integraciones y la implantación para satisfacer las necesidades que tienen los usuarios o clientes.

Entonces el termino elemento significa lo siguiente:

- Se puede entender como objetos e hitos, dadas en el tiempo de ejecución.
- Se pueden entender como clases o módulos, en tiempo de desarrollo.
- Se pueden entender como nodos o carpetas, en el mundo real.

³³ CERVANTES, Humberto, VELASCO-ELIZONDO, Perla y CASTRO, Luis. *Arquitectura de Software, Conceptos y ciclo de desarrollo*. México, México D.F., 2016, pág. 4.

³⁴ *Ibíd.*

Ciclo de desarrollo de la arquitectura

En este punto se expone la relación de las actividades técnicas del desarrollo del sistema que fue mencionado anteriormente, con el ciclo de desarrollo de la arquitectura de software, que se va describir en la siguiente gráfica³⁵.

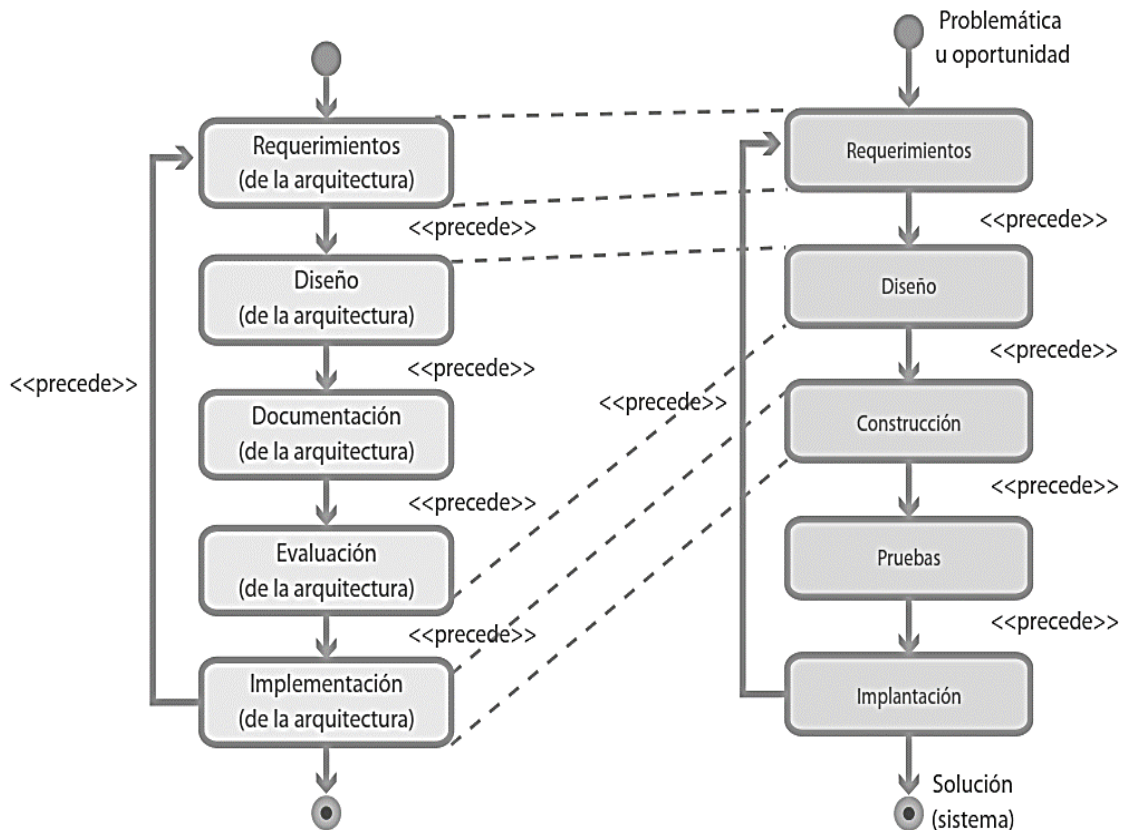


Gráfico N° 15– Actividades del ciclo de desarrollo de la arquitectura (lado izquierda) y su mapeo dentro de las actividades técnicas del desarrollo software (lado derecha).

- **Requerimientos de la arquitectura:** se enfoca en la captura, documentación y la obtención de los requerimientos que van tomar impacto en la arquitectura, se conocen como *drivers arquitectónicos*³⁶ y muestra los atributos de cada requerimiento, en algunos casos, su uso específico o restricción que puede ser relevantes en la arquitectura.
- **Diseño de la arquitectura:** se base mayormente por las creaciones estructurales de acuerdo a las clases de soluciones abstractas probadas por los patrones de diseño, además, con la elección de la tecnología.

³⁵ Ibíd. Pág. 5.

³⁶ Drivers arquitectónicos: dentro de los requerimientos que se consideran para el desarrollo de un sistema y que se derivan de los objetivos de negocio, existe un subconjunto que tiene una gran importancia relativa a la arquitectura.

- **Documentación de la arquitectura:** cuando ya está diseñada la arquitectura, es necesario darlo a conocer a los interesados mediante la documentación y el comentario del código para que los desarrolladores contextualicen el proyecto elaborado, esto incluye los bocetos previos y los bocetos de la estructura por medio de vistas.
- **Evaluación de la arquitectura:** la arquitectura de software toma un papel protagónico en el desarrollo del proyecto, para identificar los potenciales riesgos para que así se puede evaluar el diseño que ha sido creado en la documentación, esto se da antes de codificación del proyecto.
- **Implementación de la arquitectura:** cuando ya está establecido la arquitectura del proyecto, se puede desarrollar el proyecto de software a nivel de codificación, pero respetando diseño definido de la arquitectura.

La arquitectura puede tener los siguientes beneficios al proyecto de desarrollo de software, a pesar que a veces tiene restricciones por aspectos de costo, tiempo, elaboración y calidad³⁷, a continuación, se indicaran los puntos:

- Aumentar la calidad de los sistemas.
- Mejorar los tiempos de entrega de proyectos.
- Reducir el costo de elaboración del proyecto.

³⁷ Ibíd. Pág. 8.

2.2.4. Patrones de diseño:

Los patrones de diseño pueden demostrar el nivel de experiencia del programador, porque, da soluciones más concisas y optimizadas, ya que, utilizar patrones de diseño de programación ayuda a las buenas prácticas y perfecciona las técnicas. Pero es necesarios saber cuándo es tomado en cuenta cierto patrones y cuando no, por motivo que puede ser una ventaja o una desventaja, por tal motivo, se tiene que identificar los escenarios en donde se ejecutar el proyecto, para eso se necesita la documentación y que se acople a la arquitectura de software que se desea implementar³⁸.

- Establece diseños para la solución de problemas comunes o conocidos.
- Previene la insistencia de búsqueda de soluciones que fueron previamente creadas a problemas conocidos (que están solucionados).
- Estandarizar el lenguaje o sintaxis de desarrollo.
- Suministra ventajas a las nuevas generaciones de desarrolladores.

Los patrones de diseño según Laman (2004):

“(...) un patrón de diseño es una descripción de un problema y su solución, a la cual se le da un nombre, y se puede aplicar a nuevos conceptos. Es decir, un patrón provee una solución aceptada a un problema común y una terminología para distinguir esa solución (...)”³⁹.

Los patrones de diseño pueden ayudar en problemas concretos, pero también existen diferentes tipos de problema, por lo tanto, los patrones de diseño son clasificados por tres grupos principales:

- **Patrones Creacionales:** están relacionados a la creación de objetos, para controlar la forma que son construidas implementados mecanismos.
- **Patrones Estructurales:** es la forma como se relacionan entre clases, ayudan a crear los componentes flexibles y escalables.
- **Patrones de Comportamiento:** son las relaciones que tiene los procedimientos con la asignación de los objetos.

³⁸ BLANCARTE, Oscar. *Introducción a los Patrones de Diseño*. México, México D.F., 2016, pág. 18.

³⁹ MONTENEGRO, Isaac, RODRÍGUEZ, Luis y SALAZAR, Gabriela. *Uso de Patrones de Diseño de Software: un caso práctico*. Costa Rica, San José, 2012, pág. 46.

Patrones Creacionales

Funcionan para controlar la forma en que creamos los objetos, por motivo que el común desarrollo de codificación es crear los objetos al libre albedrío, pero en este grupo de patrones existe mecánicas de cómo crear las instancias de una forma más moderada⁴⁰.

Este grupo de patrones tiene el motivo o necesidad que solo exista una sola instancia de una clase. Puede que no sepa que objetos debemos instanciar, pero el principal motivo es saber identificar y utilizar los mecanismos que ofrece este patrón para que se adapten al problema. Los patrones que pertenecen a este grupo son:

- **Patrón Factory Method:** el uso de este patrón permite crear instancias dinámicas mediante la configuración que establece la manera que vamos a utilizar un archivo de texto. XML, propiedades, etc. Este patrón en particular tiene una función principal cuando está en tiempo de diseño o el subtipo de creación de objetos que tienen una clase Factory (los objetos se determinan de acuerdo a una clase Factory – creación individual).

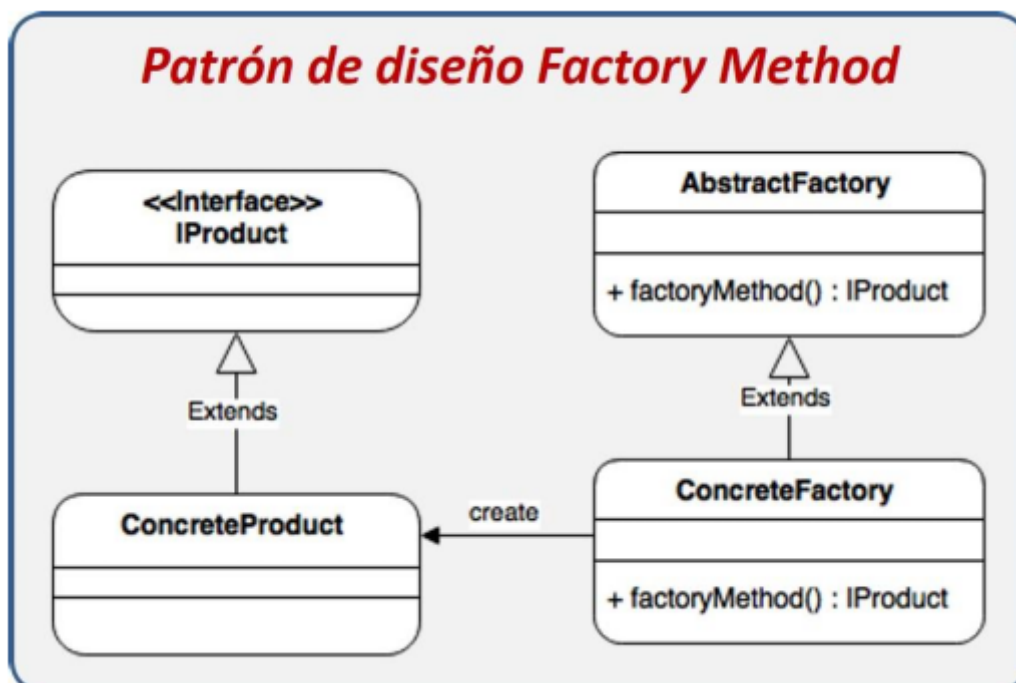


Gráfico N° 16 – Visualización del patrón de diseño Factory Method.

⁴⁰ BLANCARTE, Oscar. Op Cit.

- Patrón Abstract Factory:** agrupa las clases para que estas sean utilizadas en un bien común, a estos se le llama familias y son creadas mediante la clase Factory, que se utiliza cuando se necesita la familia de clases para resolver un problema, aunque requieran crear implementaciones paralelas para el mismo problema. Es decir, a comparación de patrón Factory Method que es creación de objetos individuales, mientras que el patrón Abstract Factory es un conjunto de objetos⁴¹.

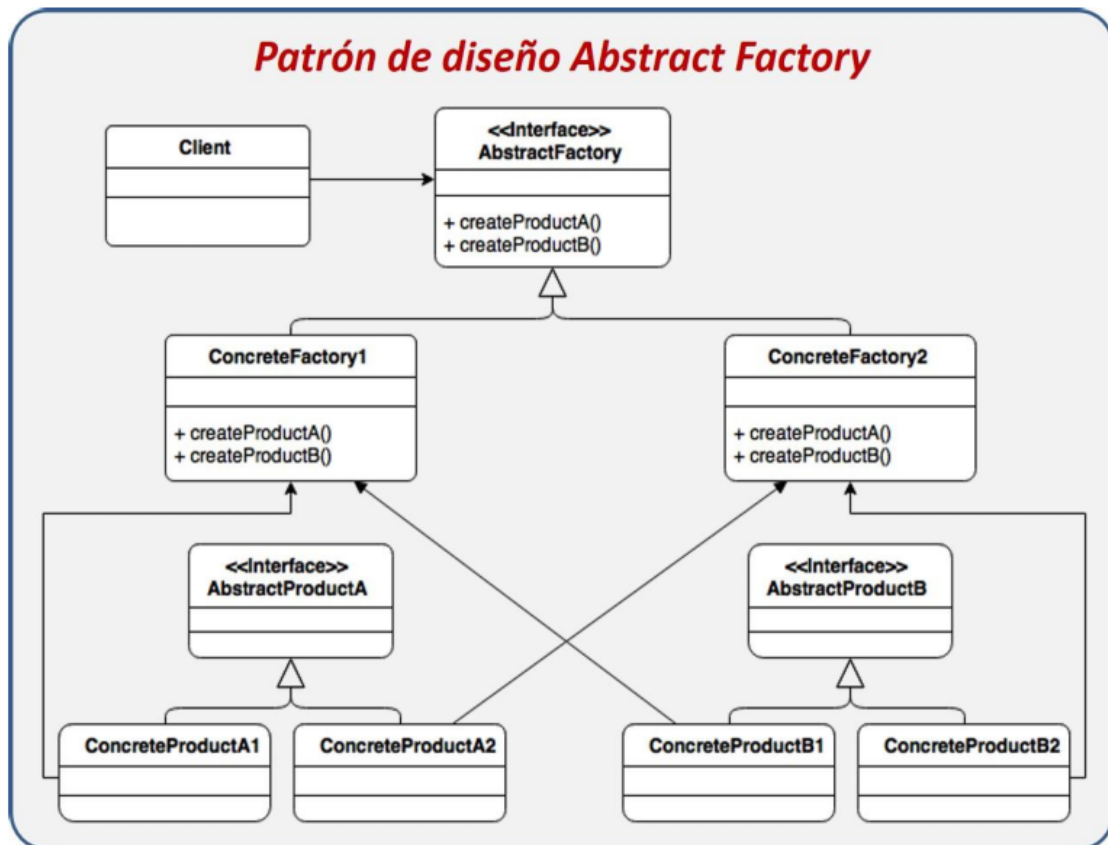


Gráfico N° 17 – Visualización del patrón de diseño Abstract Method.

- Patrón Singleton:** viene del termino soltero, que hace alusión a su nombre por tener una única instancia para toda la aplicación, restringiendo la creación de instancias de clases al libre albedria, mediante su operador “new” y aplicación de un constructor privado y un método estático. Este patrón es considera el más pequeño por la cantidad de clases requeridas para su implementación, además que solo requiere de la clase del mismo nombre del patrón, ya que utiliza un constructor privado para que impida la

⁴¹ Ibíd. Pág. 47.

creación instancias de manera directa, en un entorno contextual, se refiere a que el patrón solo garantiza la existencia de única petición, de una determinada clase en referencia al entorno global de la aplicación⁴².

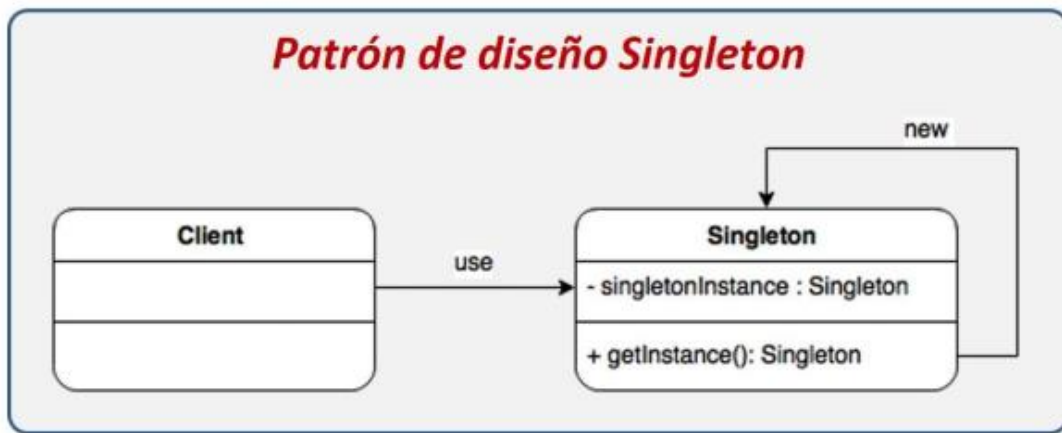


Gráfico N° 18 – Visualización del patrón de diseño Singleton.

- **Patrón Builder:** se considera uno de los más simples patrones porque permite crear objetos más complejos a través de los más simples. A este patrón se le puede ubicar en circunstancias en el cual hay que crear objetos compuesto de manera repetitivo y manual, lo que lleva a establecer las propiedades del objeto, esto puede ser tedioso y trabajoso si tenemos que crear los objetos de manera frecuente⁴³.

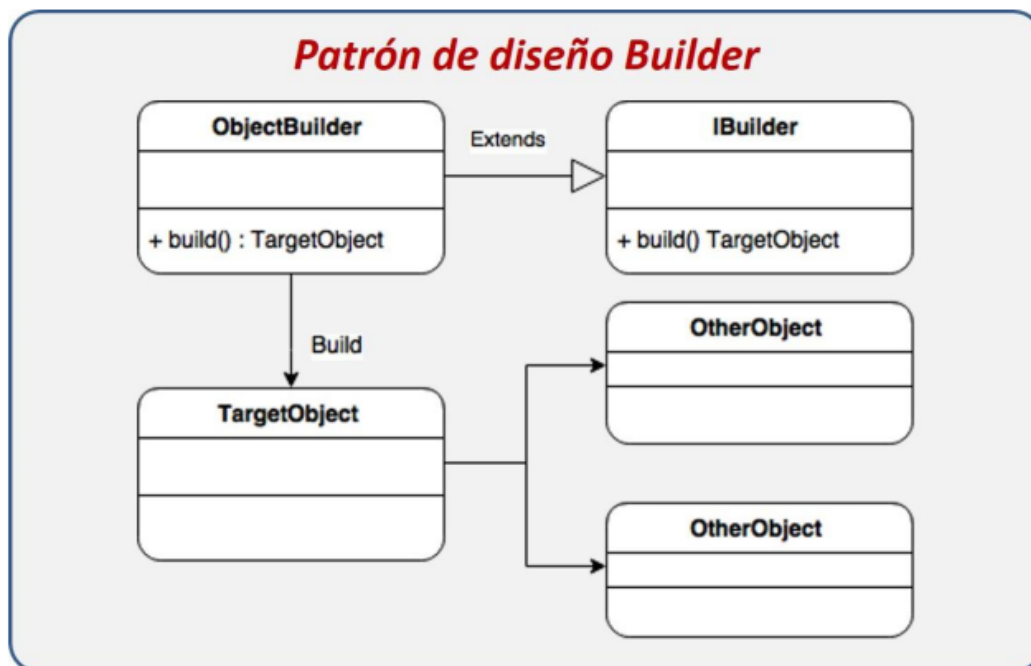


Gráfico N° 19 – Visualización del patrón de diseño Builder.

⁴² *Ibíd.* Pág. 68.

⁴³ *Ibíd.* Pág. 81.

- **Patrón Prototype:** este determinado patrón busca clonar los objetos, que después se busca un pool de prototipos para crear nuevos objetos. Para que así, sea más útil en la creación de objetos que fueron diseñados anteriormente o los objetos que tienen grandes estructuras⁴⁴.

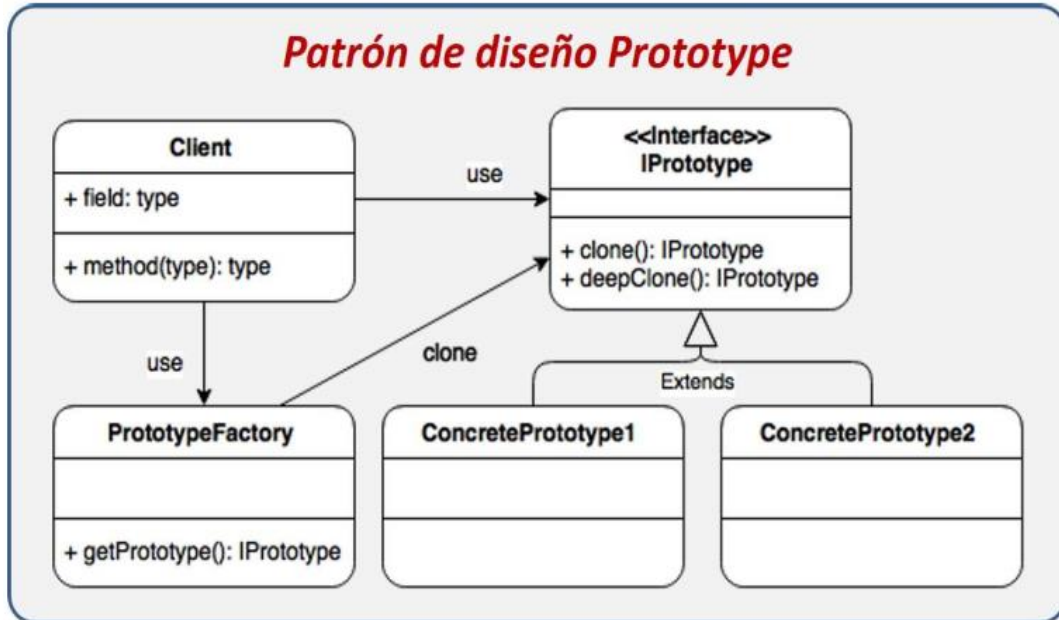


Gráfico N° 20 – Visualización del patrón de diseño Prototype.

- **Patrón Object Pool:** este patrón está especializado para trabajos con una gran cantidad de objetos, los cuales consumen muchos recursos computarizados, por eso el patrón utiliza los objetos en un determinado tiempo para luego ser desechados o ser reutilizados para evitar crear objetos cada vez que aplicativo requiera. Este patrón se caracteriza por solicitar al *ObjectPool* que le suministre objetos, para que su aplicación lo utilice y luego se un tiempo sea regresado al *ObjectPool* y así cuantas veces desea el patrón. El *ObjectPool* tendrá a su disponibilidad cierta cantidad de objetos para suministrar a un determinado proceso cuando solicitado, pero si el *ObjectPool* carecer de algún objeto, automáticamente creara uno para el proceso y lo suministrará, sin embargo, si el *ObjectPool* está a su máxima capacidad de objetos, ya no creara uno nuevo, sino, tendrá que poner en lista de espera al proceso hasta que el objeto este librado de sus funciones⁴⁵.

⁴⁴ Ibíd. Pág. 97.

⁴⁵ Ibíd. Pág. 111.

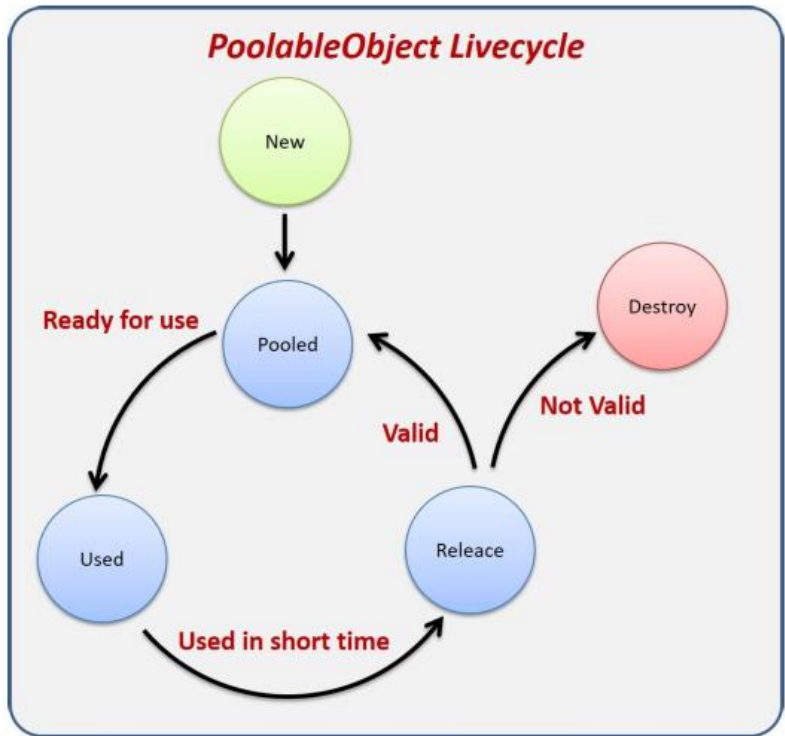


Gráfico N° 22 – Visualización del ciclo de vida un objeto del ObjectPool.

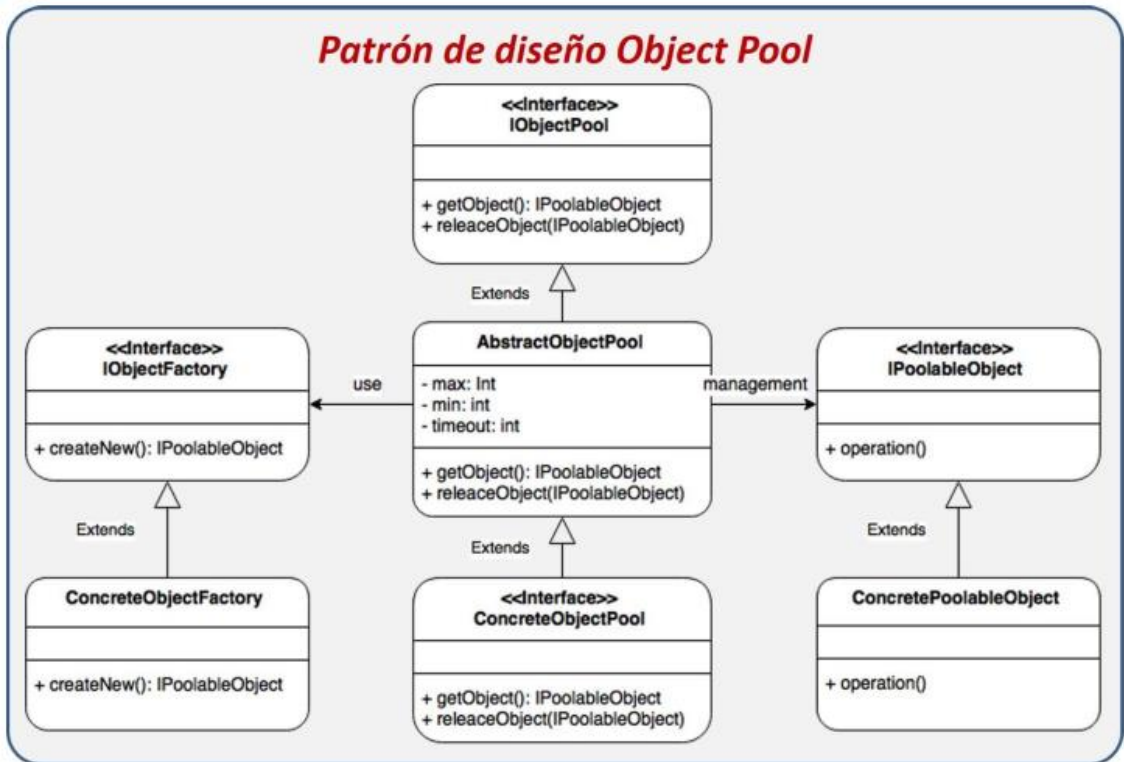


Gráfico N° 21 – Visualización del patrón de diseño Object Pool.

Patrones Estructurales

Explica la manera como los objetos y las clases pueden combinarse para formar una estructura de datos compleja y de más magnitud, ayudando a la estructura de los objetos, las relaciones y las herencias que hay entre clases, para así adaptar las interfaces, los módulos de agregar y la gestión de objetivos en el tiempo de ejecución. Ayuda a crear estructuras de clases versátiles para crear estructuras estáticas que a su vez se compongan objetos complejos, las cuales están conformadas de clases que ayudan con su forma homogénea en el tiempo de ejecución. Los patrones que pertenecen a este grupo son:

- **Patrón Adapter:** en este patrón se usa cuando hay interfaces de software inadaptables, pero que tienen funciones similares, entonces el patrón se caracteriza de homogeneizar la manera de trabajar los interfaces de software mediante métodos suministrados por el patrón de diseño⁴⁶.

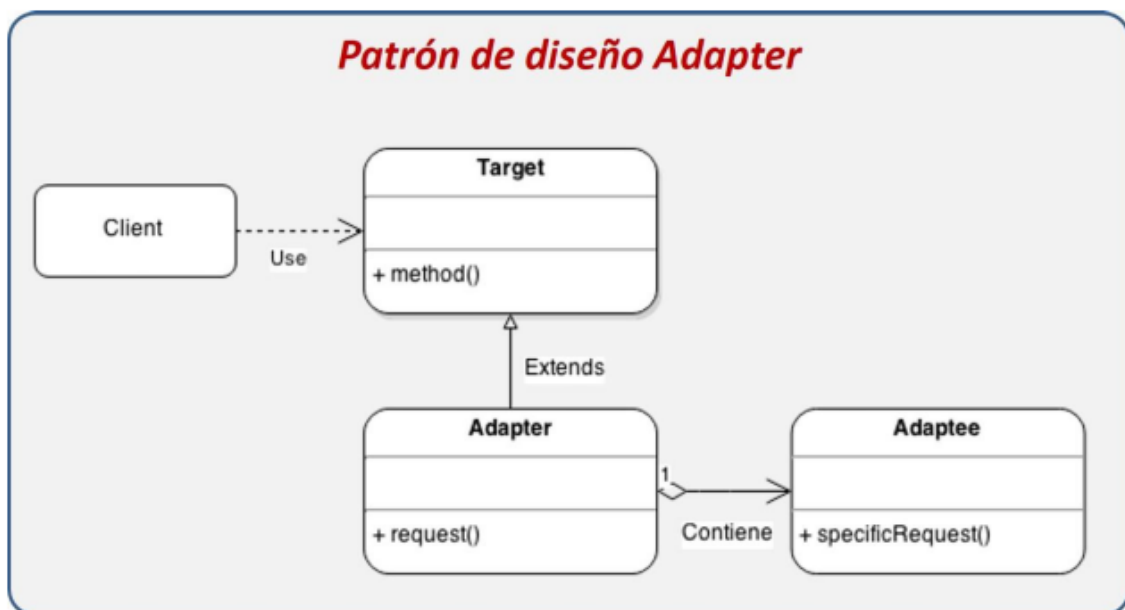


Gráfico N° 23 – Visualización del patrón de diseño Adapter.

- **Patrón Bridge:** este patrón desacopla el nivel de abstracción de su implementación, es decir, busca dar modificaciones por separado para que así no haya la necesidad de modificar el otro, sin embargo, este patrón también puede modificar los 2 aspectos, buscando dividir en 2 piezas el proceso de software para que así puedan modificarlo, pero el termino

⁴⁶ Ibíd. Pág. 140.

Bridge o puente viene por el añadido de crear una estructura de clases, el cual será la clase puente para que así pueda desacoplar otras clases que queremos utilizar, sin que este sepa el destino, dando como beneficio que un aspecto no afecta a la otra. Pero este patrón a diferencia del *Adapter* busca desacoplar las abstracciones de la implementación, es decir, que las interfaces de software son compatibles, mientras que el *Adapter* es para interfaces incompatibles⁴⁷.

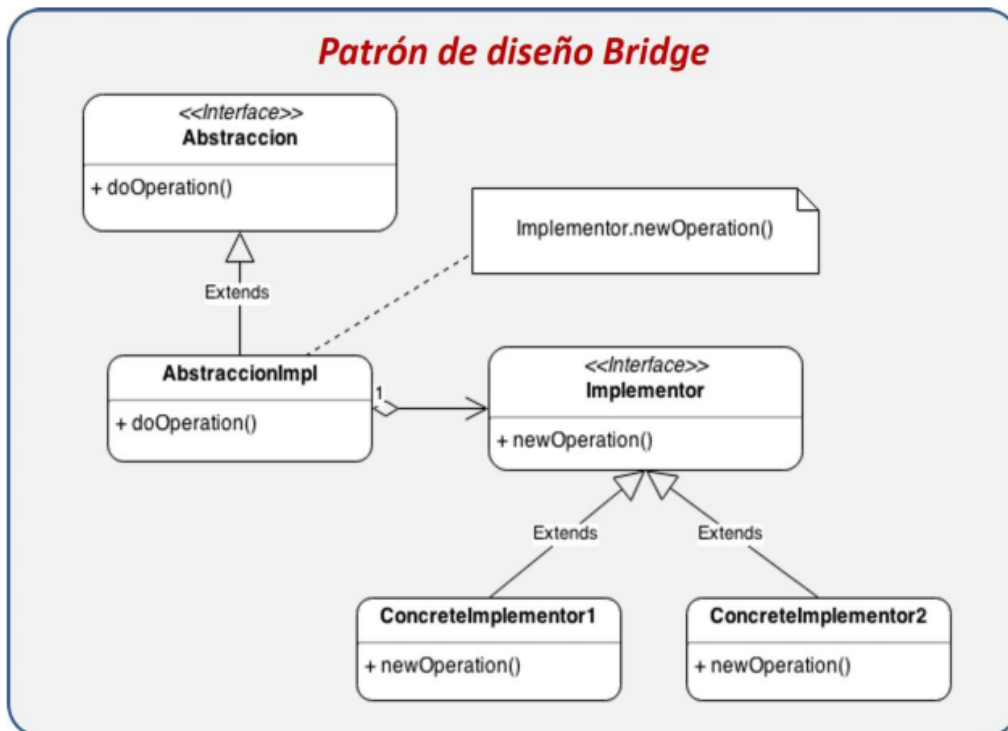


Gráfico N° 24 – Visualización del patrón de diseño Bridge.

- **Patrón Composite:** este patrón define construir una estructura compuestas de otras estructuras más simples para que así puedan manipular un grupo de elementos casi idénticos entre pero que tiene un orden específico, es decir, que estas estructuras de clases van a permanecer a una familia y en el cual va tener un ancestro en común. Esto tiene que componerse por 2 objetos llamados *Leaf* y *Composite*, en el cual, el ancestro común es el componente y *Leaf* son las estructuras simples que alimenta al ancestro, esto toma en cuenta entonces a la recursividad de los objetos para su máximo uso⁴⁸.

⁴⁷ *Ibíd.* Pág. 160.

⁴⁸ *Ibíd.* Pág. 176.

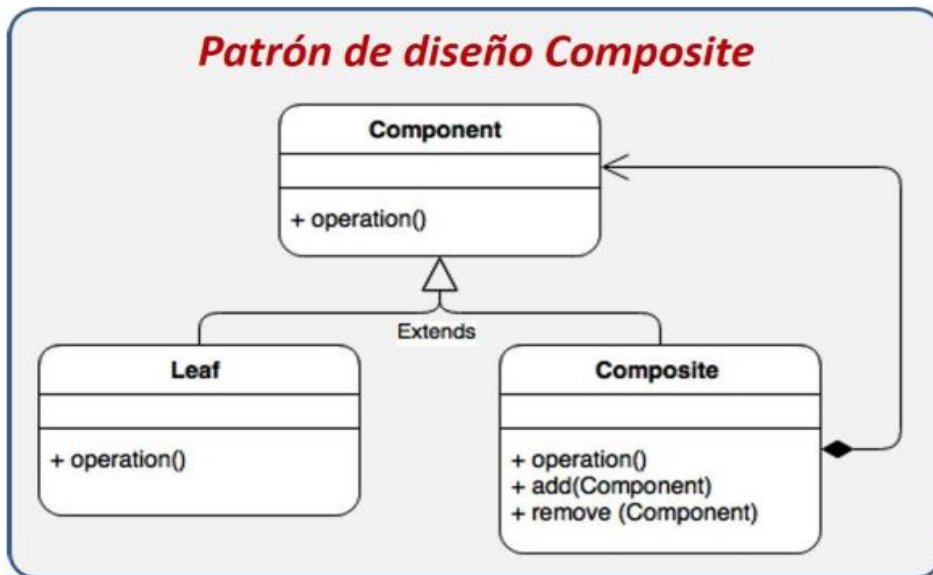


Gráfico N° 25 – Visualización del patrón de diseño Composite.

- Patrón Decorator:** este diseño soluciona los niveles de jerarquía que contenga subclasificación no se aplican, o necesita un gran impacto en todas clases que contenga el aplicativo. Este patrón de diseño da nuevas funcionalidades a un objeto de acuerdo a lo que va agregando el usuario, sin alterar la estructura de dicho objeto existente, entonces lo que hace es adicionar nuevas clases que contenga esas funcionalidades⁴⁹.

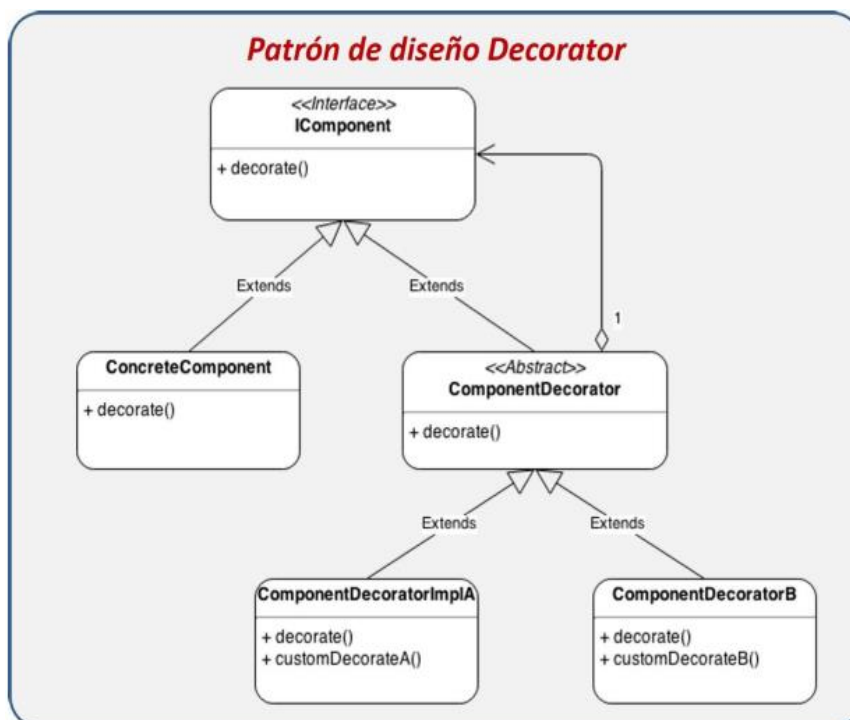


Gráfico N° 26 – Visualización del patrón de diseño Decorator.

⁴⁹ Ibíd. Pág. 199.

- Patrón Facade:** la particularidad de este patrón es la forma de esconder la complejidad de los grupos de subsistemas que proporciona una interfaz de software de alto nivel, entonces lo que realiza este patrón de diseño, que tiene se puede llamar fachada, es la comunicación entre todo ese grupo de subsistemas para realizar un proceso, pero para que este se realice se necesita conocimientos técnicos y funcionales, además de tener un orden específico⁵⁰.

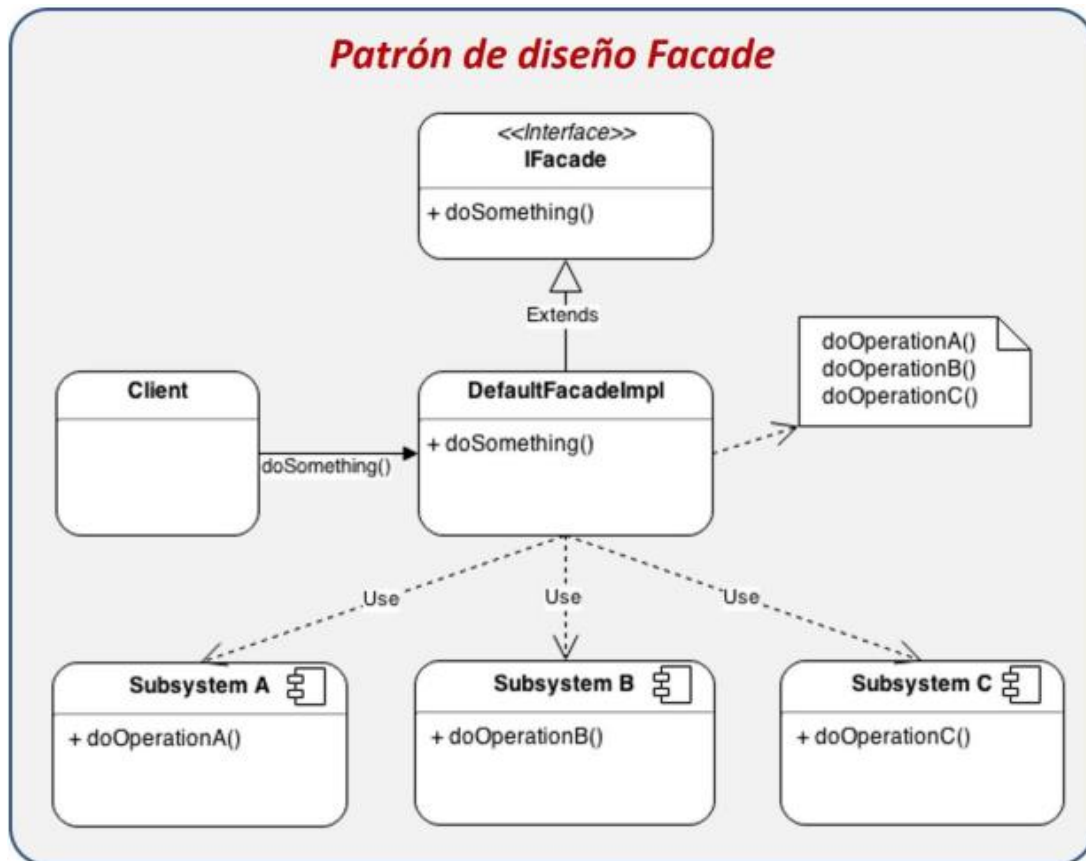


Gráfico N° 27 – Visualización del patrón de diseño Facade.

- Patrón Flyweight:** este patrón se concentra en la fabricación de los objetos ligeros, mediante la abstracción de partes recicladas de otros objetos que están compartidos, con la finalidad de tener objetos solicitado o creados por partes de otros objetos que están instanciados, para así tener la capacidad de reducir la memoria que utiliza la aplicación, optimizando la utilización de los recursos de hardware e eliminando la redundancia de los objetos que tienen las mismas propiedades⁵¹.

⁵⁰ Ibíd. Pág. 218.

⁵¹ Ibíd. Pág. 242.

Este patrón llamado en castellano peso mosca ayuda a que los aplicativos informáticos sea lo más eficiente posible, por ese motivo se divide los objetos en 2 partes:

Estado Intrínseco: es la parte idéntica o que se repite entre los objetos, dicho esto, es lo que tienen en común por lo que no varía o cambia, pero si cambia afecta a todos los objetos por general, es la parte compatible.

Estado Extrínseco: es la parte que única o diferencial que tienen los objetos para distinguirse de uno a los otros, por lo cual no compatible a ser reutilizada.

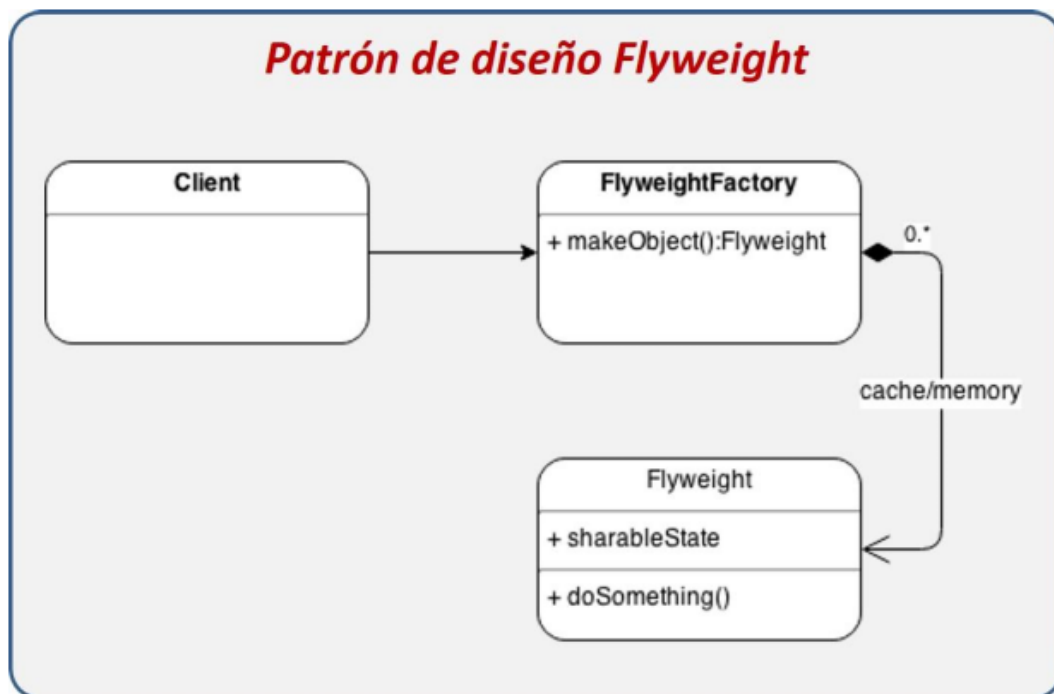


Gráfico N° 28 – Visualización del patrón de diseño Flyweight.

- **Patrón Proxy**: este diseño se centra en la mediación entre los objetos, ya que hace acciones antes y después de la acción principal o acción que realiza el usuario, por consiguiente, este patrón se caracteriza porque el usuario no es consciente de la mediación que se está realizando, por lo que el usuario solo identifica al objeto con su estructura esperada sin saber que esta interactuando con el Proxy⁵².

⁵² Ibid. Pág. 262.

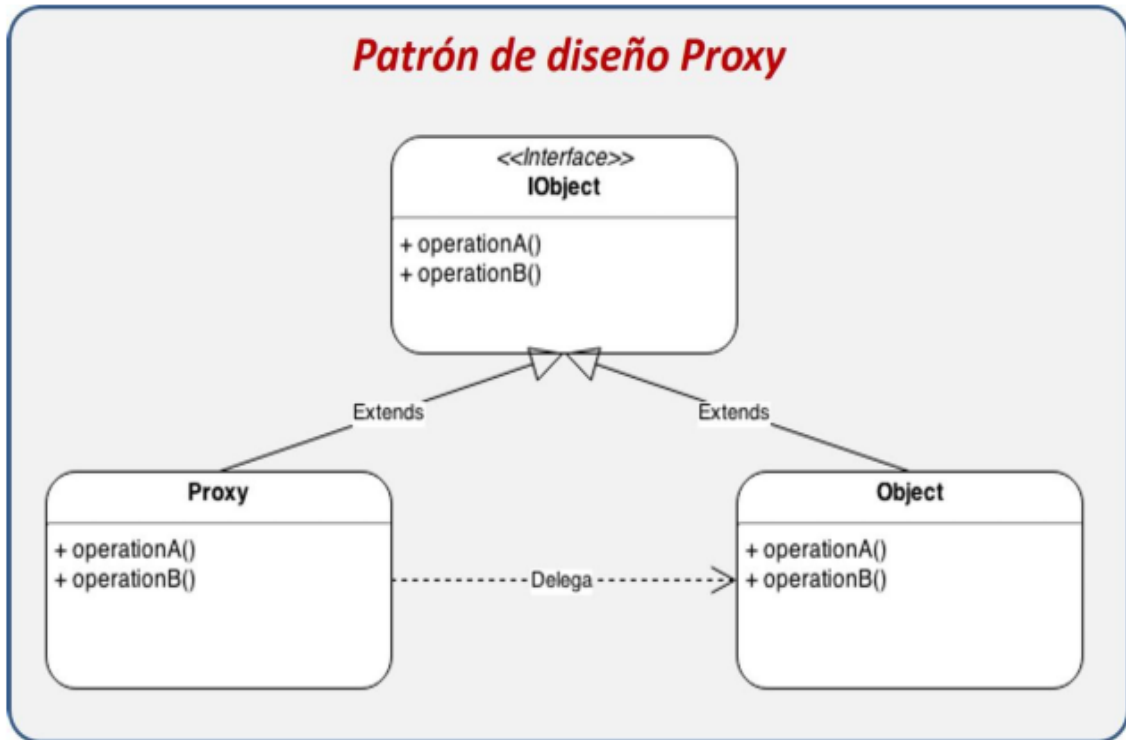


Gráfico N° 29 – Visualización del patrón de diseño Proxy.

Patrones de Comportamiento

Se caracteriza del resto de patrones, porque colabora, relacionan y delega de las responsabilidades entre otras clases, esto simplifica la forma en como los objetos se comunican e interactúan, es decir, se enfocan en asignar responsabilidades a los objetos y como dichos objetos se comunican entre ellos, mientras que los patrones creacionales y estructuradas se enfocan en la creación de objetos y la creación de estructuras respectivamente.

Este patrón de comportamiento es más común de tener casos de uso del día a día, así es más fácil identificarlo. Los patrones que pertenecen a este grupo son:

- **Patrón Iterator:** este patrón proporciona métodos para recorrer las estructuras de datos, sin la necesidad de tener un conocimiento de las mismas, además que este tipo de patrón permite recorrer las estructuras con un Iterator⁵³, que es una interface que contiene métodos que recorren los elementos de la estructura, los cuales son:

hasNext: es un método booleano, que regresa *true* si encuentra más estructuras existentes y *false* si ya no encuentra más estructuras y no hay más elementos.

Next: es cuando retorna la siguiente estructura de datos.

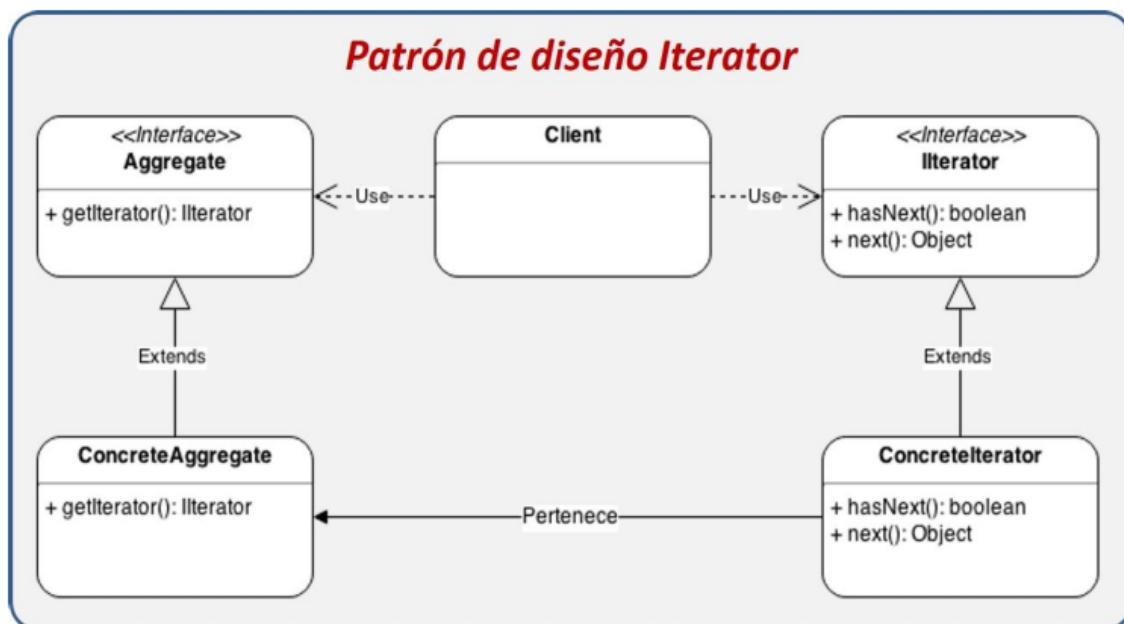


Gráfico N° 30 – Visualización del patrón de diseño Iterator.

⁵³ Ibíd. Pág. 280.

- Patrón Command:** nos permite ejecutar operaciones, los cuales no tiene conocimiento de los detalles de la estructura, pero las operaciones están compuestas por *comandos* (de ahí viene el nombre del patrón de diseño), a su vez cada operación es una implementación de una clase independiente, que están programadas para realizar una acción concreta, ya sea con parámetros o no. La ventaja es desarrollar los comandos necesarios para luego encapsular bajo una interfaz de ejecución⁵⁴.

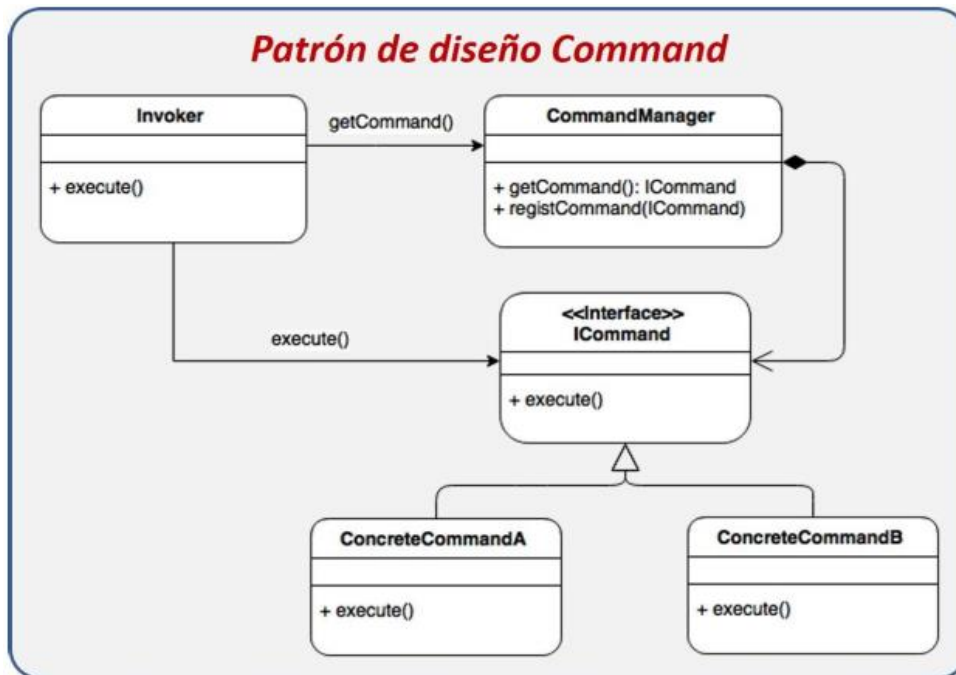


Gráfico N° 31 – Visualización del patrón de diseño Command.

- Patrón Observer:** este patrón de diseño observa los cambios que hay en los objetos, ya sea la forma o el efecto del cambio que hay en dicho objeto, los cuales serán informados por notificaciones los cuales son conocidos como Publicador– Suscriptor. Una de las principales características de este patrón es la forma de desacoplar los componentes de cada interfaz gráfica de usuario (GUI). Además, los objetos pueden ser observados por varios observadores, valga la redundancia, los cuales notificarán al mismo tiempo sobre los cambios realizados en un objeto. Se divide en 2 partes este patrón de diseño, los cuales son; el observable, que es el objeto que desea observar, y el observador, que es el objeto que está supervisando los cambios del observable⁵⁵.

⁵⁴ Ibíd. Pág. 294.

⁵⁵ Ibíd. Pág. 332.

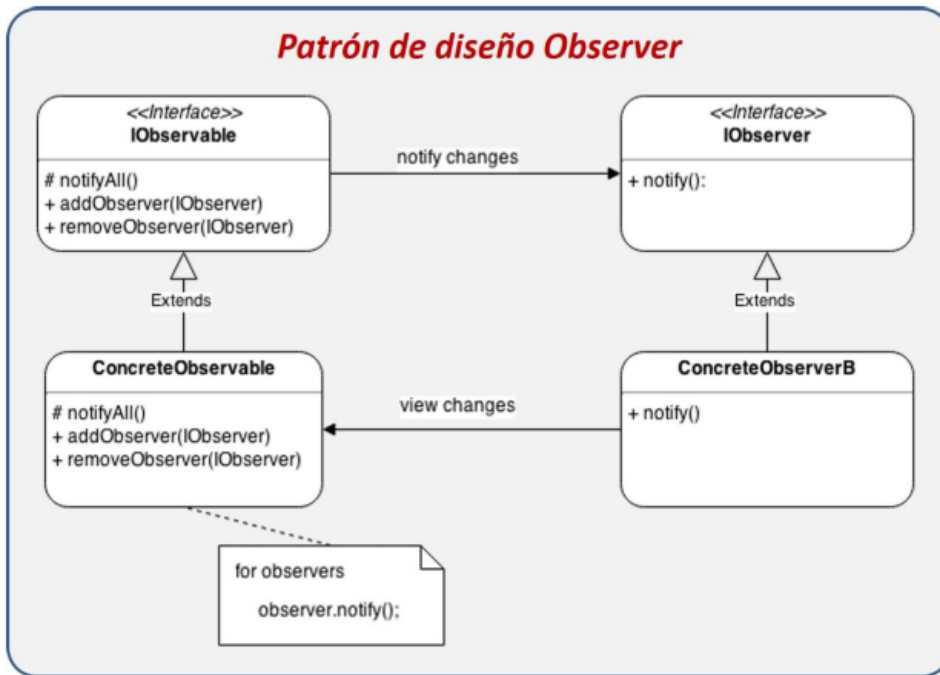


Gráfico N° 32 – Visualización del patrón de diseño Observer.

- Patrón Template Method:** este patrón de diseño busca centralizar la funcionalidad del código mediante la reutilización para implementar algoritmos que den fin o una solución, esto se logra con una clase base que esta con métodos predeterminados para cada algoritmo que después se puede implementar para su ejecución orden o si no de mantendrá de manera abstracta en la ejecución de las subclases. El patrón Template, da un método público que puede ser utilizada por los usuarios⁵⁶.

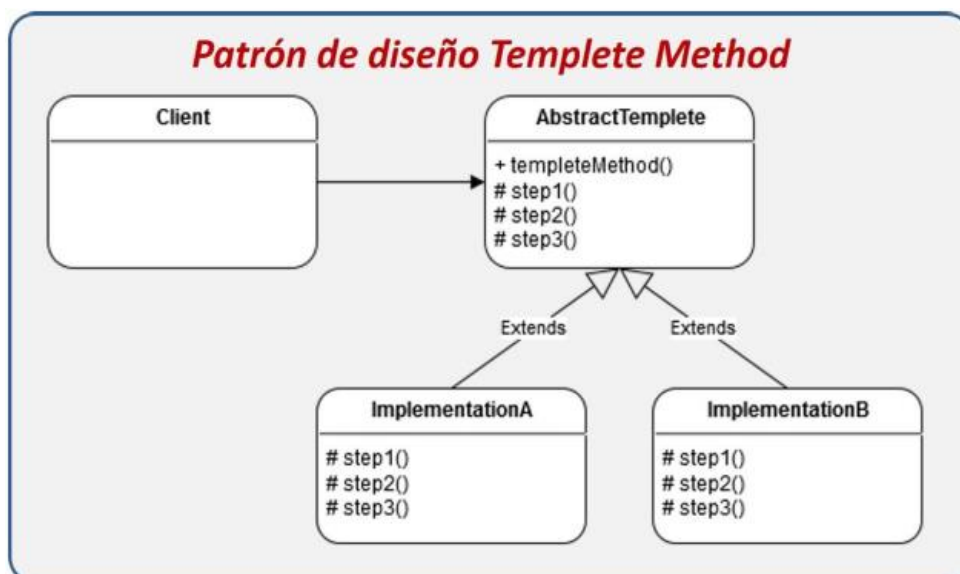


Gráfico N° 33 – Visualización del patrón de diseño Template Method.

⁵⁶ Ibíd. Pág. 349.

- Patrón Strategy:** este patrón se base en el polimorfismo, para que en tiempo de ejecución el comportamiento de la clase tenga una serie de intercambios, con el fin de que el objeto varié su comportamiento de distintas maneras, establecidas previamente por una estrategia, por eso, he ahí, el nombre del patrón que se conforma del contexto y las estrategias. El contexto encapsula las estrategias y su comportamiento que tiene relación a la estrategia antes mencionada, es decir, las estrategias son clases que definen el comportamiento preciso de estos, entonces, significa que entre más estrategias hay, más funcionalidades tiene el contexto⁵⁷.

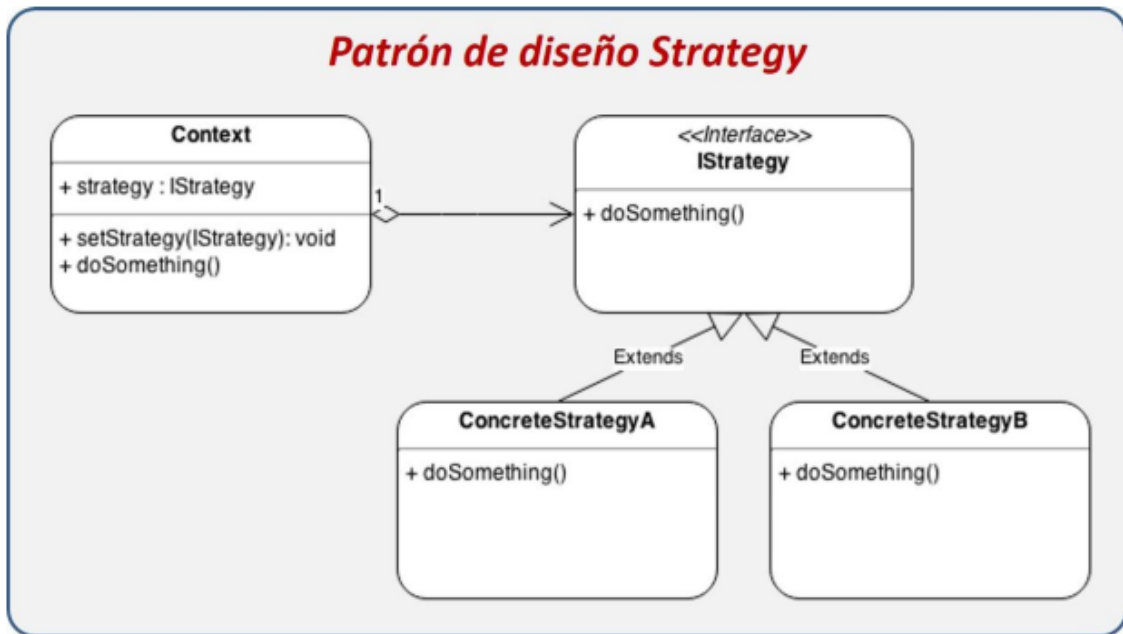


Gráfico N° 34 – Visualización del patrón de diseño Strategy.

- Patrón Chain of Responsibility:** es un patrón de diseño en el cual se distingue la versatilidad, porque resuelve el problema sin saber que objeto se deberá procesar, facilitando las soluciones del problema. El patrón de la Cadena de Responsabilidad se apoya en la estructura de forma de cadena, en el cual, hay una secuencia de objetos que tratan de resolver la petición, pero esto es gracias a que no acopla o mejor dicho, evita que la petición quede en un objeto que no lo pueda dar solución, por ese motivo la petición puede estar relacionado con la cadena de objetos pero solo puede ser solucionado con el objetos que tenga la capacidad de procesarlo⁵⁸.

⁵⁷ Ibíd. Pág. 374.

⁵⁸ Ibíd. Pág. 395.

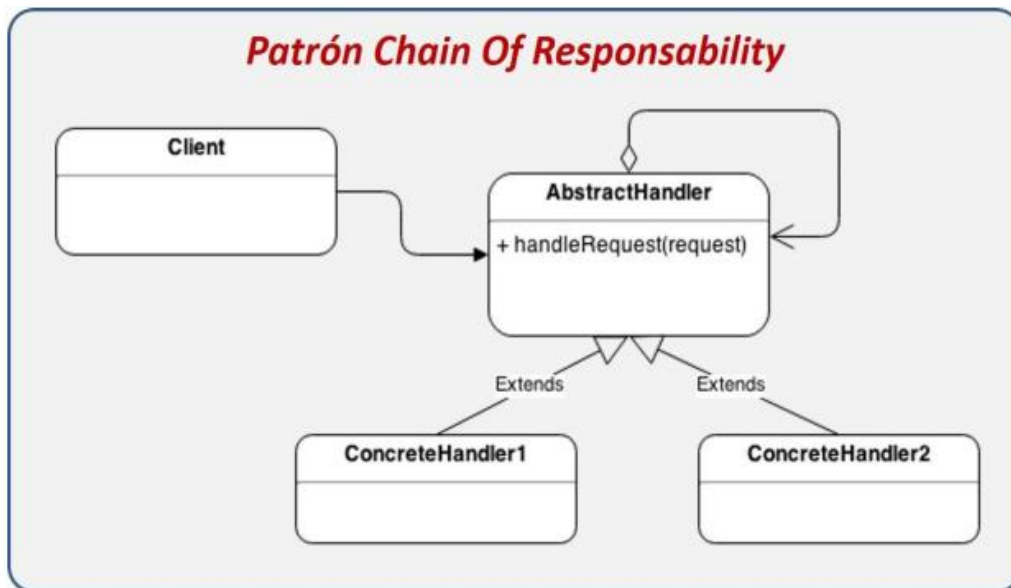


Gráfico N° 35 – Visualización del patrón de diseño Chain Of Responsibility.

- Patrón Interpreter:** es para interpretar y evaluar el lenguaje definido por las expresiones, es patrón accede a interpretar las expresiones de acuerdo a lenguaje de programación como a sintaxis estandarizados o creados por uno mismo pero que sean entendible. Este patrón puede a llegar a ser muy confuso debido que tenemos que saber los aspectos de polimorfismo, herencia y la recursividad, además que combina técnicas avanzadas de programación. Es crucial resaltar que el lenguaje definido tenga el formato **Backus-Naur form (BNF)** para que sea preciso y simple a la vez⁵⁹.

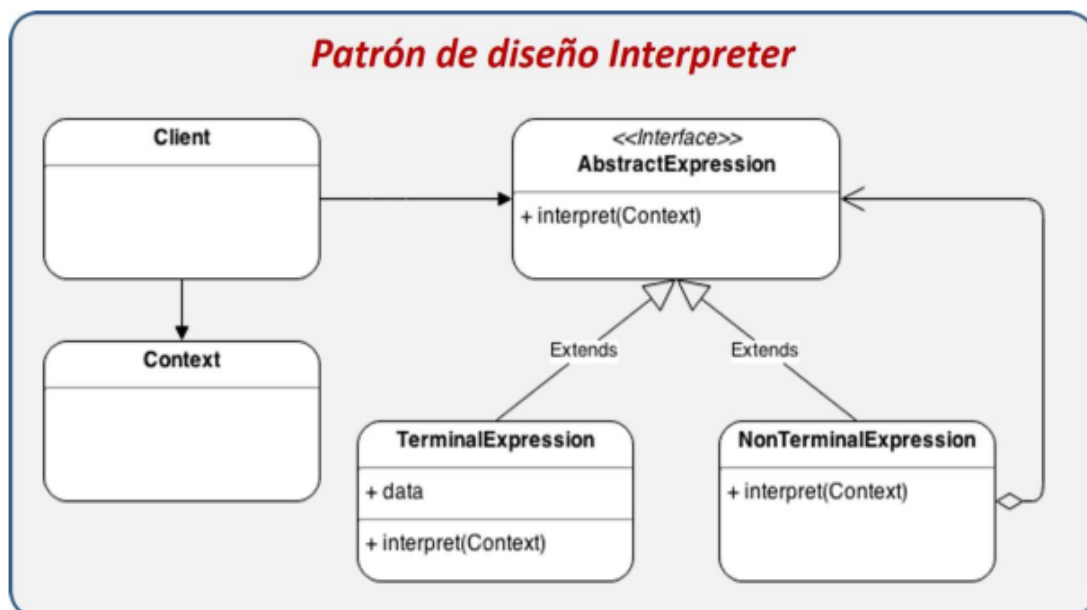


Gráfico N° 36 – Visualización del patrón de diseño Interpreter.

⁵⁹ Ibíd. Pág. 424.

- Patrón Mediator:** en este patrón se gestiona la forma como se comunican las clases entre sí, el patrón Mediator es muy útil para la gran cantidad de clases que se comunicarse de forma directa, además que es bidireccional por lo que las clases se podía con el resto de las clases mediante objetos. Este patrón de diseño puede evitar el siguiente problema que es la creación numerosa de las clases que a su vez tiene relación con otras clases en el mismo proyecto, entonces esto puede generar problemas en el acoplamiento de las clases del proyecto⁶⁰.

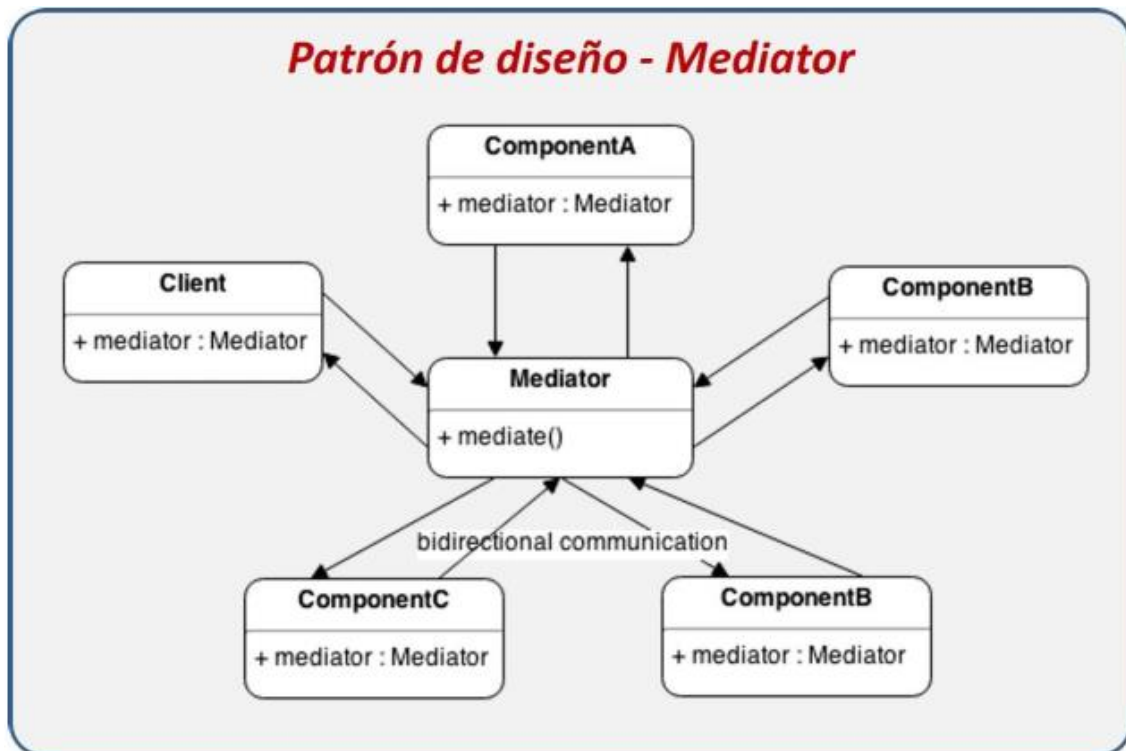


Gráfico N° 37 – Visualización del patrón de diseño Mediator.

- Patrón Memento:** este patrón de diseño tiene un función muy simple y básico que es capturar en un momento establecido el valor que tiene un objeto para que luego si hay una modificación con objeto, se puede volver a poner en el valor capturado a ese objeto. Este patrón se puede poner en caso cuando se hace edición de un documento y se desea guardar con todos los parámetros dicha edición⁶¹.

⁶⁰ Ibíd. Pág. 466.

⁶¹ Ibíd. Pág. 493.

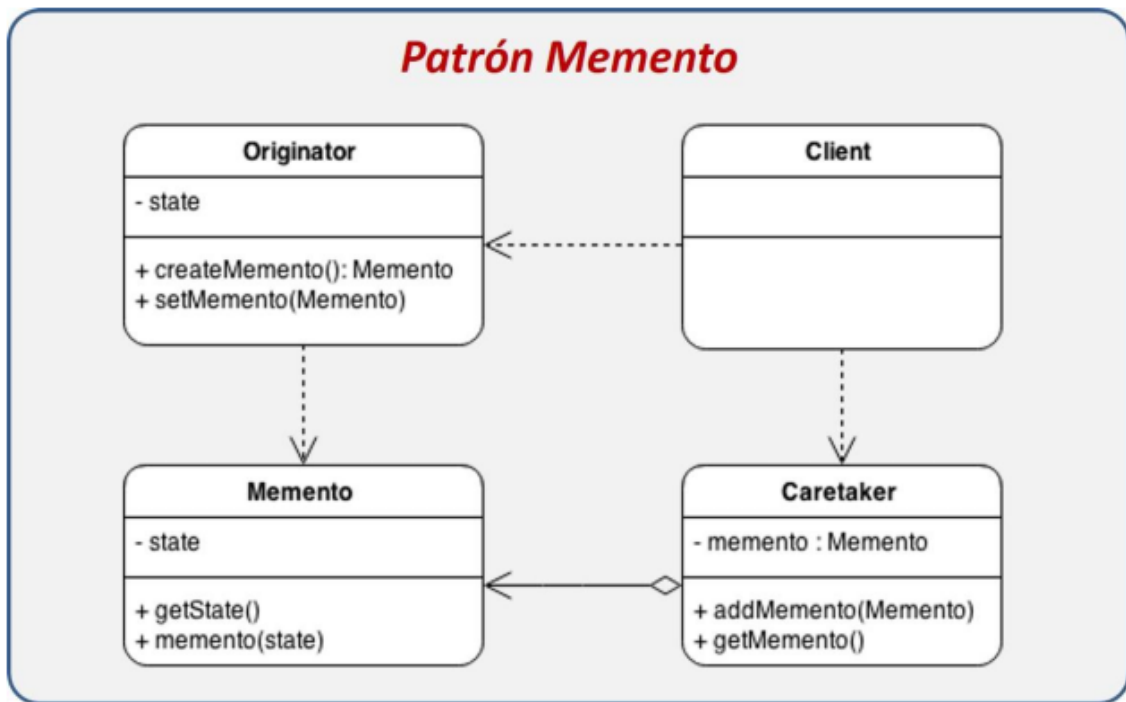


Gráfico N° 38 – Visualización del patrón de diseño Memento.

- Patrón Null Object:** este patrón tiene la principal función de evitar valores nulos o vacíos a los objetos para que no resulte en un error en tiempo de ejecución de código. Para que cumpla con su función el patrón hace uso de interfaces con estructura vacía en vez de un valor nulo⁶².

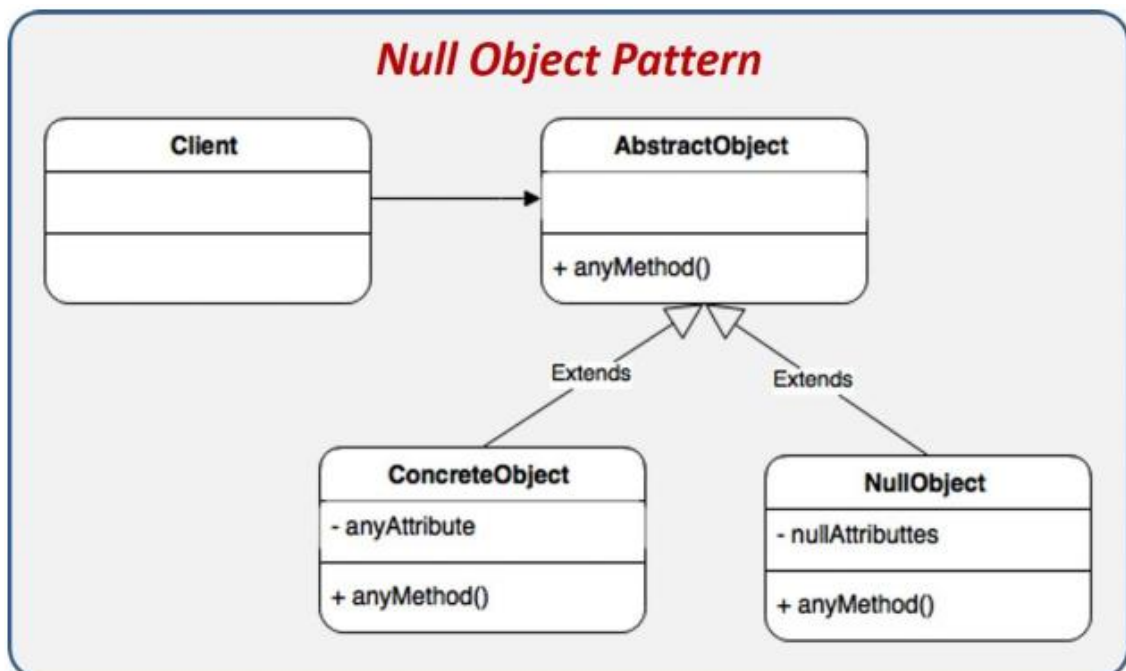


Gráfico N° 39 – Visualización del patrón de diseño Null Object.

⁶² Ibíd. Pág. 509.

- Patrón Visitor:** el diseño *Visitor* establece separar la lógica u operaciones que se puede ejecutar con la estructura de datos, es decir, la estructura de datos realiza operaciones internamente, pero dichas operaciones pueden variar de acuerdo a la cantidad o también se pueden desarrollar nuevas operaciones que se adapten a la estructura de datos. Teniendo en cuenta que si las operaciones aumentan, las estructura también crece, entonces por consecuencia la estructura de hace más completa⁶³.

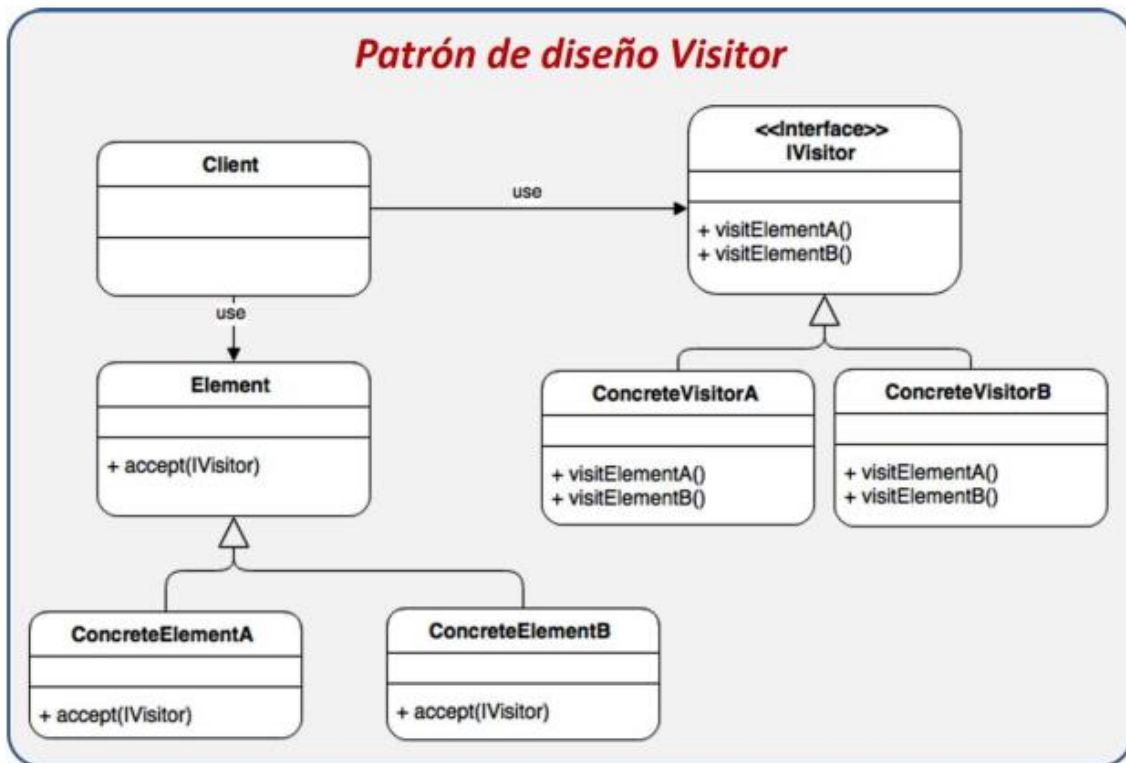


Gráfico N° 40 – Visualización del patrón de diseño Visitor.

- Patrón State:** este diseño tiene la peculiaridad de editar el comportamiento de las clases, pero dependiente en que circunstancia se encuentra el software, por lo que es imprescindible tener clases que representen distintos estados del software, en otras palabras, tiene que ver una clase que capture cada estado del software. El patrón State puede ser confundido con el patrón Memento por motivo que guardan o capturan estado del software, pero la diferencia puede ser que el Memento no altera el comportamiento del estado del objeto⁶⁴.

⁶³ Ibíd. Pág. 546.

⁶⁴ Ibíd. Pág. 521.

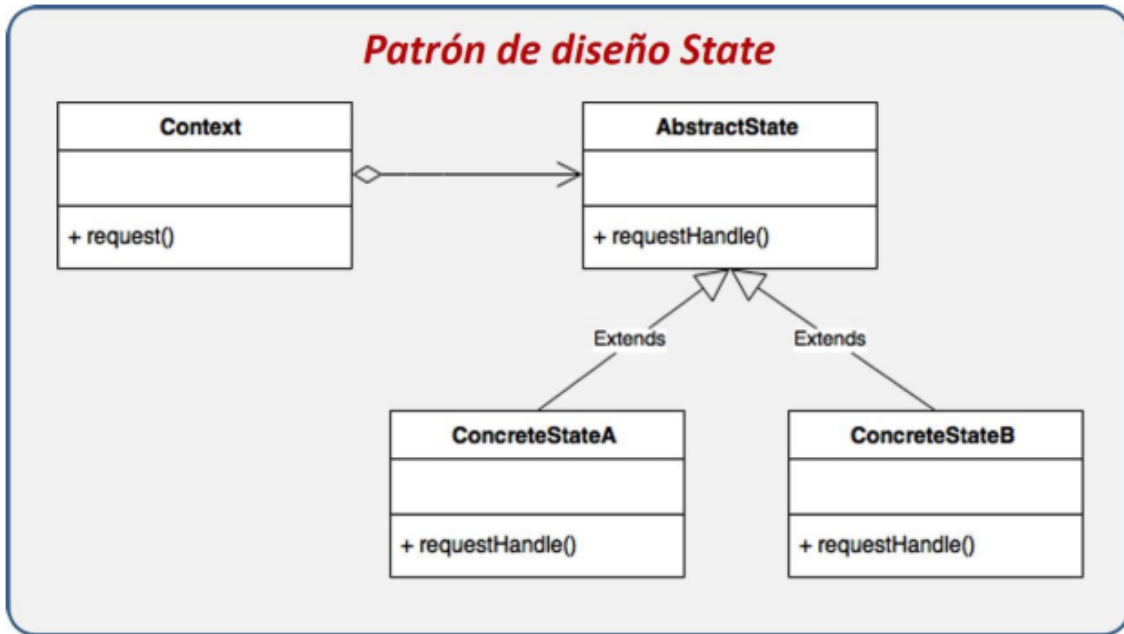


Gráfico N° 41 – Visualización del patrón de diseño State.

2.2.5. Modelamiento y Diagramas:

En este capítulo se va describir partes elementales de los proyectos mediante el diseño, para tener la finalidad de contextualizar el proyecto y así puedan los interesados comprender cada proceso realizado por los programadores para satisfacer las necesidades que fueron planteadas al inicio del proyecto. Pero para mostrar todos los procesos en un nivel alto de entendimiento, hay que saber analizar los procesos, para que se muestre los requerimientos cumplidos de manera correcta explicándolo en lenguaje estándar para que todos comprendan o tengan la misma idea de cuál es la finalidad del proyecto y como está yendo en tu ejecución. En la ingeniería de software se desarrolla los proyectos con bocetos o modelos existentes para que así se tenga un estándar y puedan ser descritos de acuerdo al análisis con las herramientas que ofrecen estos métodos. En este capítulo se va explicar cómo sería el desglose contextual de un proyecto con los siguientes puntos⁶⁵.

Requerimientos Funcionales

Especifican los requerimientos principales o que tenga una interacción entre el sistema y su ambiente más determinada, en otras palabras, son los servicios o entrada particulares que intervendrán explícitamente al sistema. En los requerimientos funcionales se tiene que como característica que son los requerimientos identificados que luego son los que tiene un objetivo específico a resolver, es decir, tiene sus propios fines a cumplir ya sea una tabla base o que agrupe muchos procesos pero que tenga importancia en los procesos del sistema, además que todos los requerimientos funcionales deben ser cumplido. Por ejemplo:

- Validación automática de las licencias ingresados por el usuario.
- El sistema autenticara y controlara los movimientos del usuario
- Los campos monetarios o números deben estar con los formatos establecidos (el formato de moneda – nuevo sol y con cantidades exacta).
- Debe tener registros de control para una posterior autoría (fecha y hora).

⁶⁵ ERIKSSON, U., "Why is the difference between functional and Non-functional requirements important?", 2012, <http://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/> (consultada al 27 de abril 2021).

Requerimientos No Funcionales

En cierto punto se puede denominar contraparte de los requerimientos funcionales porque limitan los servicios o funcionalidades que tiene el sistema, restringiendo algunas dimensiones como tiempo, los procesos de desarrollo o las normas que se implementan en el proyecto, entonces teniendo entendido que los requerimientos no funcionales son la forma indirecta en que están vinculados las funciones del sistema, porque en realidad tiene una relación directa con las propiedades y restricciones, se entiende mejor en la siguiente imagen, en el cual, se ve la relación de los requerimientos no funcionales con los grupos generales⁶⁶.

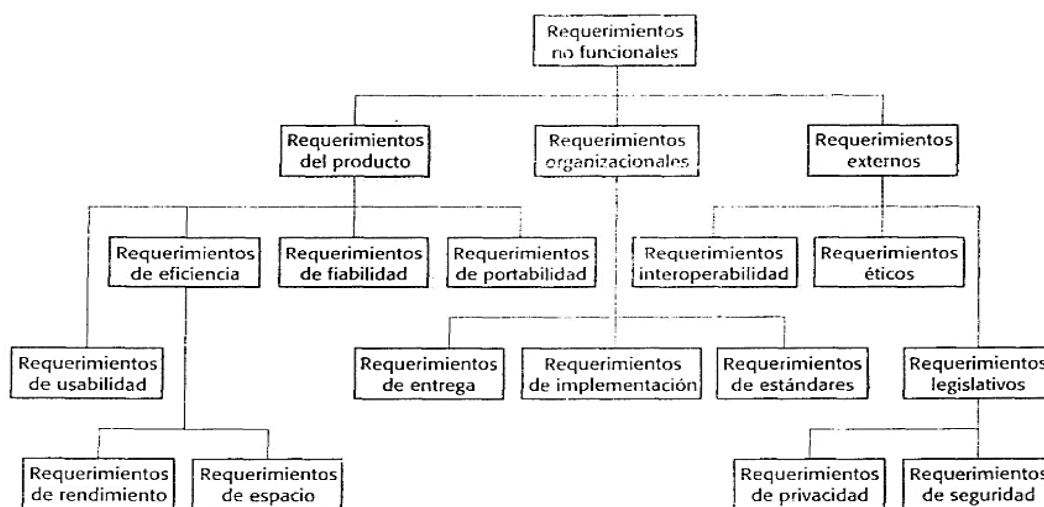


Gráfico N° 42 – Visualización de Tipos de Requerimientos No Funcionales (Sommerville, 2005).

- **Requerimientos del producto:** especifica el desempeño del producto, en cuestión de tiempo, rapidez, fiabilidad y cuantos recursos consumen para la ejecución, además, de la portabilidad y usabilidad.
- **Requerimientos organizacionales:** corresponde al alineamiento de las políticas y métodos que hay en la organización, es decir, se tiene que respetar las normas que exige el cliente para tener cierto grado de estándar de trabajo (documentación, diseño y lenguajes de programación).
- **Requerimientos externos:** es la interoperabilidad que le definen el sistema para interactuar con otros sistemas que tiene en haber la organización.

⁶⁶GOOGLE Sites, "Metodología Gestión de Requerimientos", 2021, <https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales> (consultada al 27 de abril 2021).

Diseño de Arquitectura

En este diseño, se busca alcanzar ciertos objetivos planteados para la composición de los componentes para satisfacer los requerimientos con todas las restricciones o políticas de la organización⁶⁷.

- Objeto se refiere a la estructura que componen la arquitectura de software (físicas, lógicas, ejecución y operación).
- Los componentes son los conceptos de diseño los cuales construye la arquitectura y que además agrega los patrones de diseño y las tecnologías.
- El conjunto de requerimientos representa los requerimientos funcionales y no funcionales para representar los atributos de calidad.

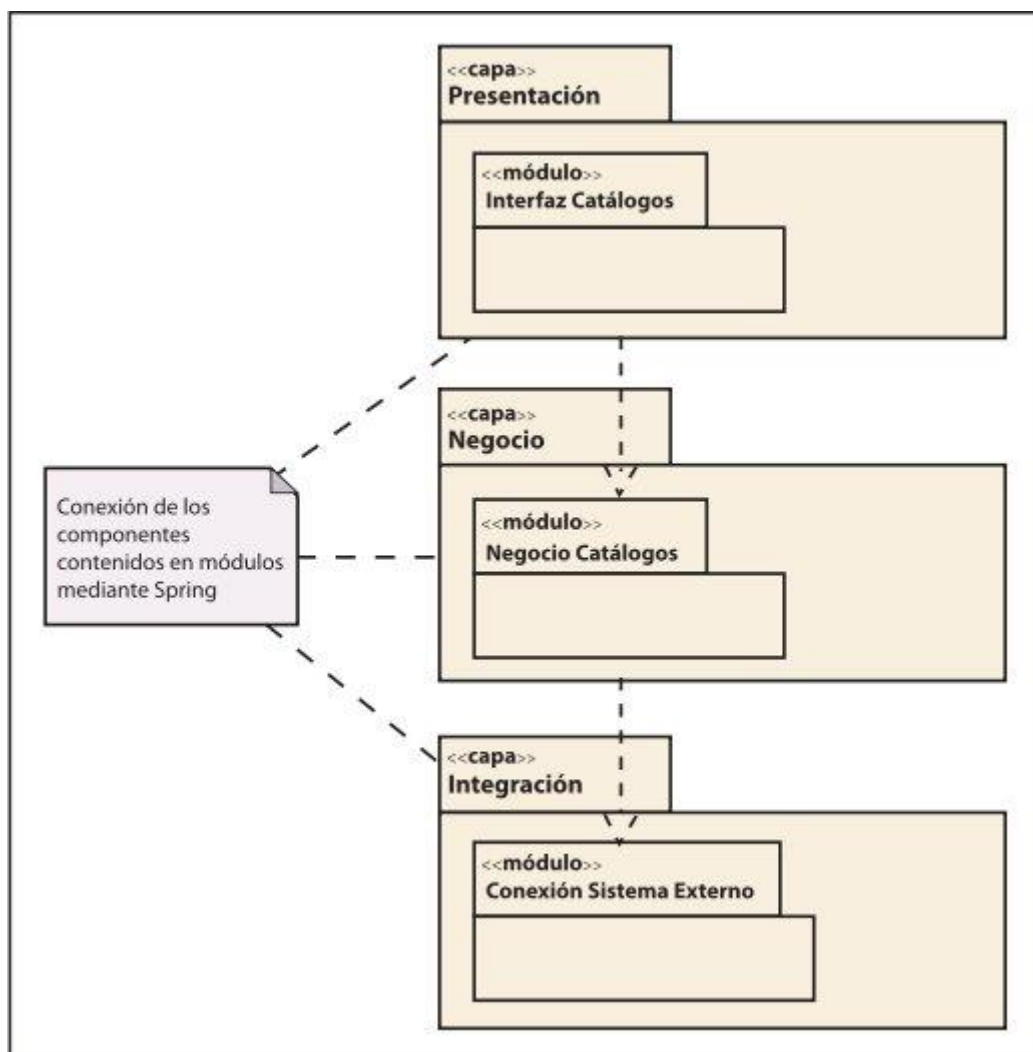


Gráfico N° 43 – Visualización de la descomposición de un sistema.

⁶⁷ RALPH, P. y WAND, Y., "A Proposal for a Formal Definition of the Design Concept", Lecture Notes in Business Information Processing, Vol. 14, pág. 103-136, 2009.

Diagrama Entidad – Relación (modelamiento de datos)

Es el diagrama que ilustra las entidades con sus respectivas relaciones de uno o muchos que lo componen en el sistema, además que usan el diseño de entidad relación para diseñar y depurar una base de datos relacional.

Ayuda mucho al diseño de un proyecto porque muestra cómo se ven las tablas en un entorno estructural gramatical, como emplea las entidades con sustantivos y las relaciones con verbos. Pero el diagrama entidad relación se componen por entidad, relación y atributo que cada uno tiene su símbolo definido que lo representa⁶⁸.

- **Los componentes del diagrama entidad relación:**

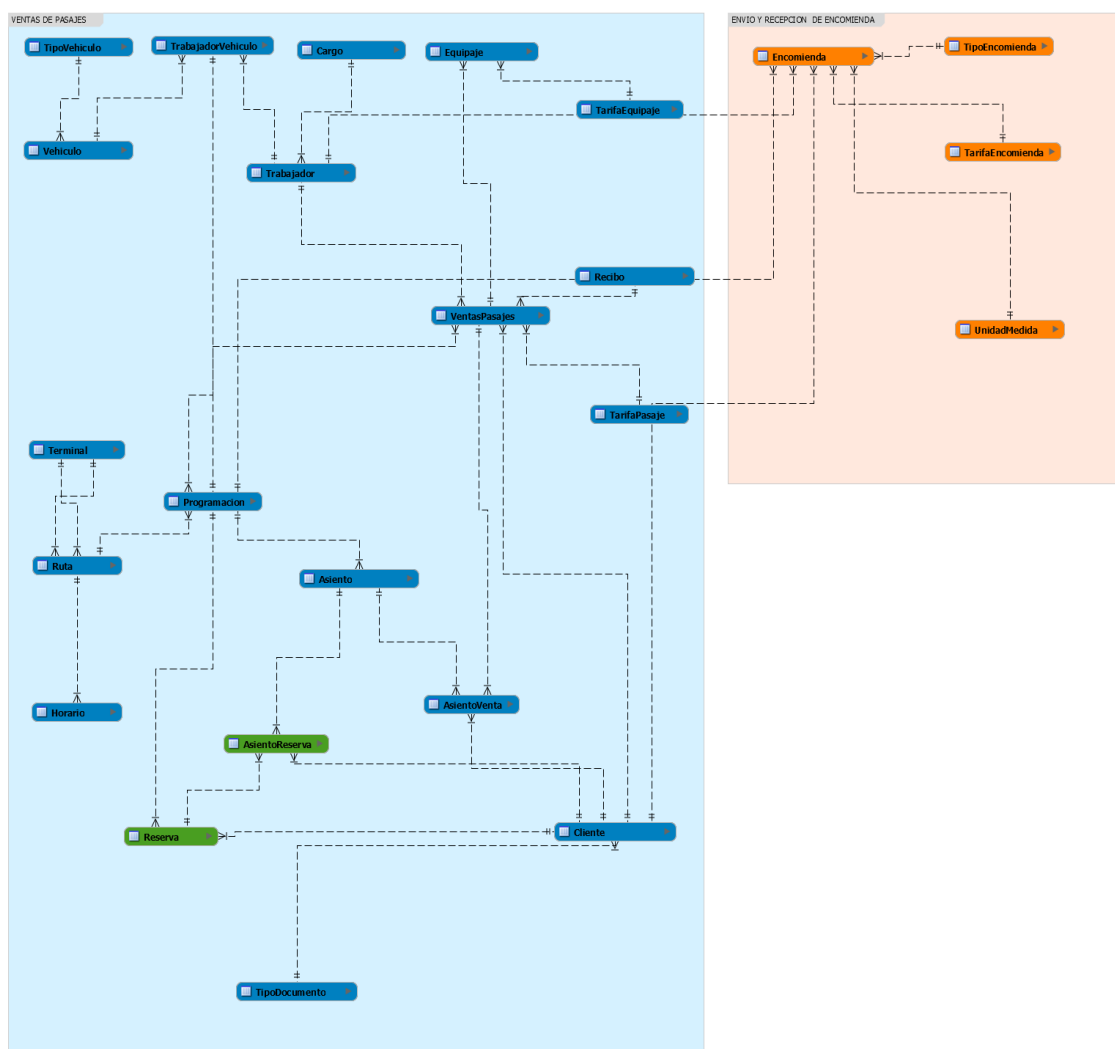


Gráfico N° 44– Visualización de un diagrama entidad relación.

⁶⁸ Lucid Software Inc. , “Qué es un diagrama entidad-relación”, 2021, <https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion> (consultada al 27 de abril 2021).

Entidad: es la persona u objeto que puede tener datos almacenados del mismo⁶⁹.



Gráfico N° 45 – Visualización de una entidad.

Tipo de entidad: son el conjunto de elementos que pueden definir en una categoría específica. Ejemplo: estudiantes, productos, etc.

Conjunto de entidades: se referencia con el tipo de entidad, pero lo define en cierto momento, es decir, se relaciona con la instancia del objeto. Ejemplo: persona de determinada sangre, etc.

Categoría de entidades: se clasifican en entidades fuertes, los cuales se definen por sus propias atribuciones, mientras que las entidades débiles son entidades asociativas o que están relacionadas (elementos) a entidades dentro de un conjunto de entidades.

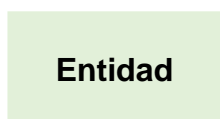


Gráfico N° 47 – Visualización de una entidad fuerte.



Gráfico N° 46 – Visualización de una entidad débil.

Claves de entidad: es el atributo que identifica a una entidad del conjunto, es decir, que hace diferenciar una entidad de las demás, pero estas claves se dividen en 4 partes, que son, *superclave*; es el conjunto de atributos que definen a una entidad de un conjunto de entidades, *clave candidata*; es una superclave, pero con la cantidad mínima (el conjunto de entidades puede tener más de una clave candidata), *clave primaria*; es una clave candidata que identifica al conjunto de base de datos, *clave extranjera*; identifica la relación que tiene la entidad con otras entidades.

⁶⁹ *Ibíd.*

Relación: es cómo interactúan las entidades entre sí, y si lo queremos contextualizar se agrega un verbo. Utiliza la recursividad, ya que la entidad se usa varias veces⁷⁰.



Gráfico N° 49 – Visualización de una relación.



Gráfico N° 48 – Visualización de una relación débil.

Atributos: es la característica o propiedad de la entidad (se lo presenta con círculos).

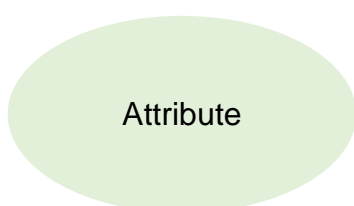


Gráfico N° 52 – Visualización de un atributo.

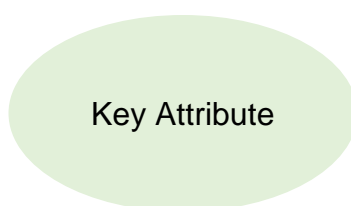


Gráfico N° 51 – Visualización de un atributo clave.

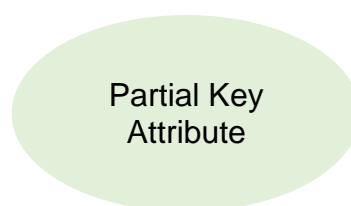


Gráfico N° 50 – Visualización de un atributo clave parcial.

Categoría de los atributos: los atributos en general tienen varios valores, pero los cuales se dividen en 3, *simples*; que son de valor mínimo y que no pueden separarse (ej.: número de celular), *compuesto*; contienen subatributos los atributos, *derivados*; son los derivados o cálculos de otro atributo (ej.: calcular la edad por la fecha de nacimiento).

Valores múltiples: es cuando el atributo contiene más de un valor, como varios nombres de una persona.

Valor único: solo tiene un valor único.

⁷⁰ *Ibíd.*

Diagramas de Procesos

Es la representación gráfica de los principales procesos que realiza la organización, en una manera contextual, es la secuencia de actividades de proceso. Dicho diagrama ayuda mucho a la funcionalidad del proceso y como entender a los interesados o terceros, que no tiene conocimiento previo del proyecto, sobre el flujo de trabajo involucrado. Además, que es una de las principales herramientas para el análisis de los procesos, es decir, ayuda a identificar los procesos que falte una optimización o necesiten una nuevamente implementación para integrar nuevas funcionalidades. Los elementos más importantes del diagrama de procesos se expresan por simbología que comúnmente está en un formato ANSI (American National Standards Institute), ya que deben expresar la simbología de los procesos y subprocesos, y las decisiones mediante conectores⁷¹. Se definirán los principales elementos:

- **Procesos o actividades:** su símbolo principal es el rectángulo, que significa acción de un proceso, que en mayor posibilidad se refieren a los procesos principales que tiene la organización, que los mismo tienen la característica de que deben estar siempre presente en la organización.
- **Subprocesos:** son aquellos que están con símbolos rectangulares con líneas dobles en cada lado del perímetro de la figura, entonces el subproceso expresa parte del proceso padre, es decir, son los complementos o partes de otros procesos más complejos (asignadas a tareas simples).
- **Nodos de decisión:** son la parte de en el cual, dependiendo la respuesta, se realizará uno u otra acción, generando una toma de decisión de “SI” o “NO”, además que está representado por el símbolo de un rombo (diamante) en una de sus puntas de derivaran las opciones previamente mencionadas.
- **Líneas de flecha:** su principal función es la conexión de los procesos, subprocesos y nodos de decisión, para mantener la coherencia del diagrama. Su representación de una línea con dirección que normalmente te dará camino del diagrama.

⁷¹ EKON. , “¿Qué es un diagrama de procesos y por qué es tan importante para tu empresa?”, 2020, <https://www.ekon.es/diagrama-procesos-empresa/> (consultada al 01 de mayo 2021).

- **Terminadores:** solo aparecen en el inicio y final del diagrama para tener conocimiento de la finalización, y están representadas por un rectángulo con esquinas redondeadas.

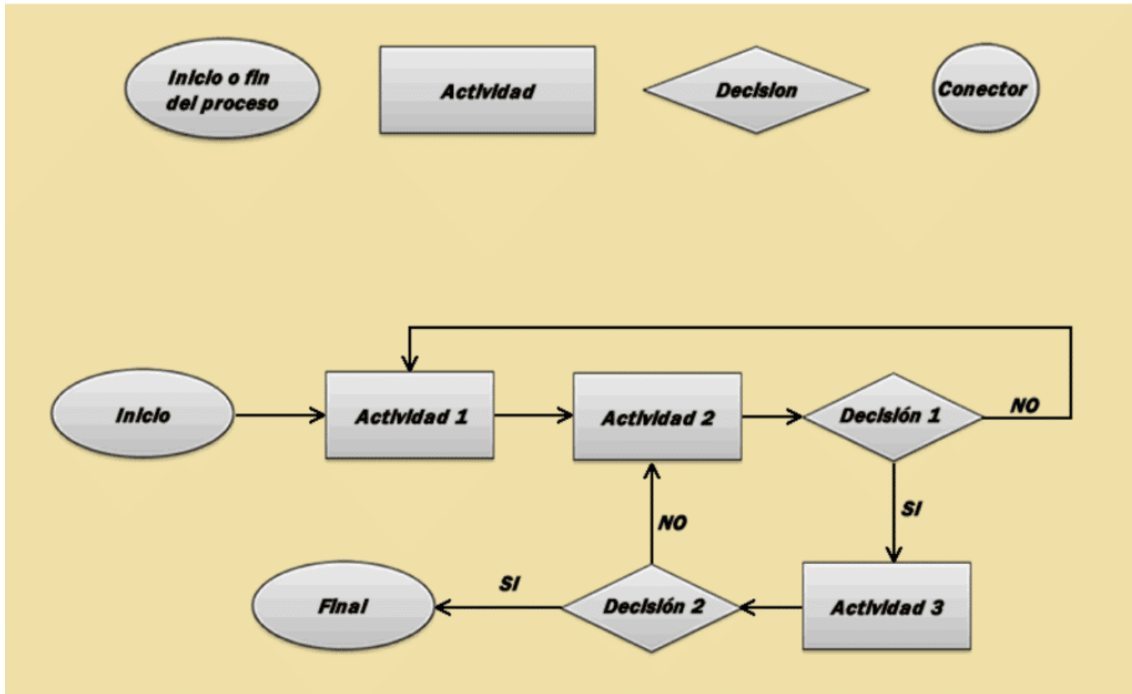


Gráfico N° 53– Principales elementos de un diagrama de procesos.

Ejemplo:

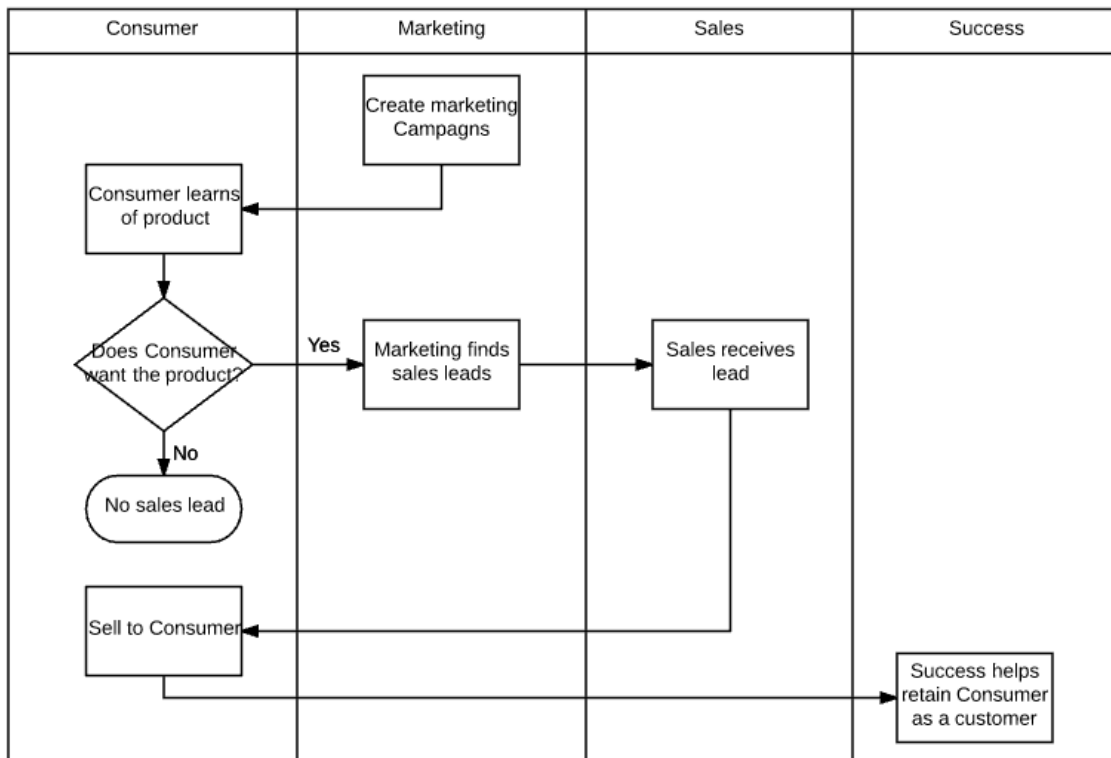


Gráfico N° 54– Ejemplo de diagrama de procesos.

Diagrama de flujo

En este diagrama se presenta al proceso desglosado para que así se pueda emplear en la documentación para futuros estudios, que propondrían la mejora de los procesos con las nuevas tecnologías. Este diagrama tiene muchas diversidades de símbolos que expresan una acción en el proceso, las cuales son; rectángulos, óvalos, rombos y otras figuras que tiene su particularidad o característica única. Este diagrama permite visualizar de manera entendible y contextual el proceso de negocio para que así las personas o interesados tenga entendido como se realiza cada actividad y a quien involucran⁷². Ejemplo:

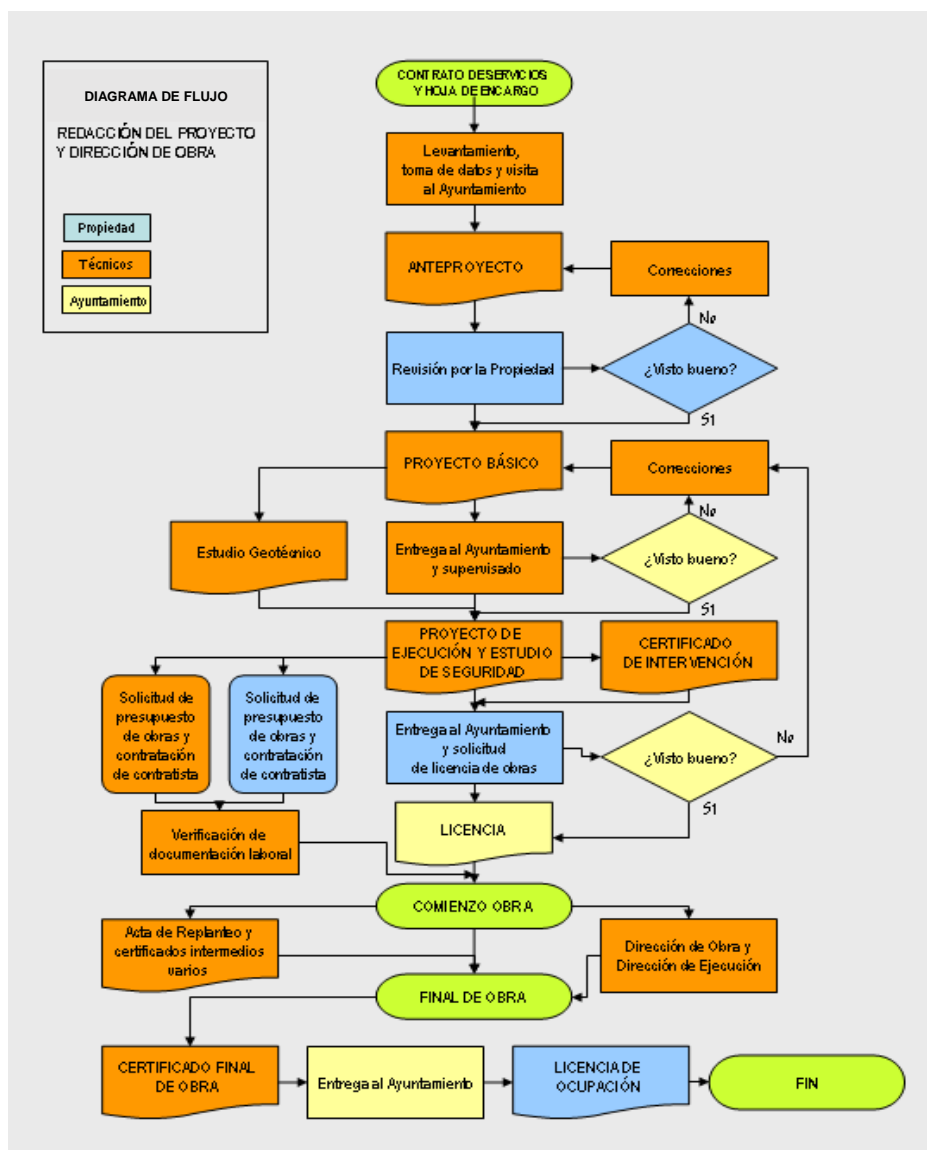
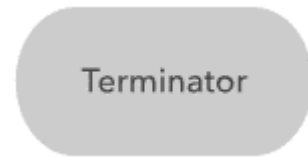


Gráfico N° 55 – Ejemplo de diagrama de flujo.

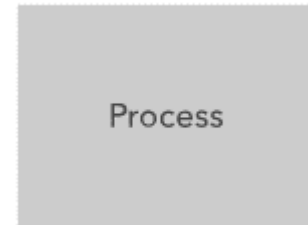
⁷²Lucid Software Inc., “Qué es un diagrama de flujo”, 2021, <https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-flujo/> (consultada al 01 de mayo 2021).

- **Símbolos de un diagrama de flujo:**

Terminales: Indica el inicio o fin de un proceso.



Proceso: Realización de una actividad por un relativo método o procedimiento.



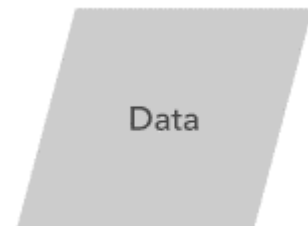
Documento: Es el tipo de documento que esta de entrada, utiliza o que resulta como entregable de una método.



Decisión: Son los puntos en el cual puede varios la acción siguiente por alternativas planteadas.



Datos (entrada / salida): Es el ingreso de datos a memoria desde un periférico para ser procesado y tener como salida información.



Datos almacenados: Es el proceso de grabado de datos en conjuntos de tablas de datos almacenados previamente creado.




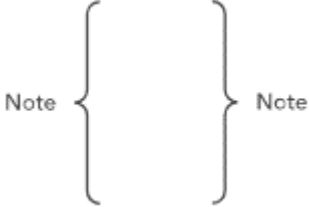



Flechas:	Indica la dirección de cómo se está realizando el proceso.	
Anotación:	Agrega una explicación o comentario del proceso.	
Proceso predefinido:	Se utiliza cuando ya existen procesos definidos, como llamadas de procedimientos.	
Referencia / conector dentro de una página:	Es la conexión de una parte del diagrama con otra, entendiendo que hay una entrada y una salida.	
Referencia / conector fuera de una página:	Es la conexión de una parte del diagrama con una hoja externa de otro diagrama, es decir, su conexión es externa del diagrama.	

Diagrama de casos de uso

Es la representación de las funciones del sistema desde un punto de vista del usuario para que pueda comprender el comportamiento del sistema o software en un caso concreto, en el diagrama de casos de uso a los usuarios se llaman actores, aunque no específicamente para usuarios humanos, sino, el actor puede considerado parte del rol en el caso de uso, además, el diagrama de caso de uso especifica la manera que hay que utilizar el sistema o también muestra los aciertos o errores para así, obtener nuevos requerimientos del sistema. Pero si tenemos que describir el diagrama de caso de uso, tenemos que mencionar el *Lenguaje de Modelado* o más conocido como *UML* (Unified Modeling Language) que permite modelar un sistema mediante gráficos, para que así el usuario puede visualizar las especificaciones, fase de construcción y la documentación del sistema⁷³. El UML tiene muchas variantes de diagramas, las cuales ayudan adaptar el modelado a diferentes aspectos que presente la entidad, los cuales se clasifican de la siguiente manera:

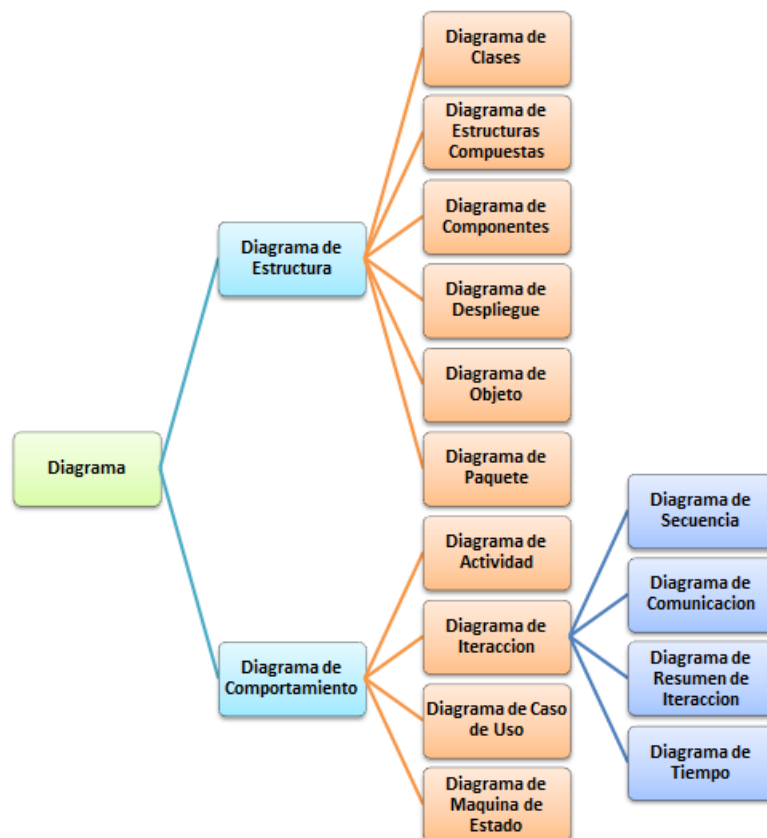


Gráfico N° 56 - Clasificación de los diagramas UML.

⁷³ CEVALLOS, Karla. "UML: Casos de Uso", 2015, <https://ingsoftwarekarlacevallos.wordpress.com/2015/06/04/uml-casos-de-uso/> (consultada al 09 de mayo 2021).

- **Elementos y estructura del diagrama de casos de uso**

Para que el diagrama de casos de uso sea comprensible a cualquier usuario, es decir, contextualizado para tener una información universal del funcionamiento del sistema hay que dividir en 4 elementos, los cuales son⁷⁴:

Actor: es la persona o sistema que interactúa en el caso, pero teniendo una figura humana esquematiza.

Sistema: es el entorno o caso de uso mencionado, que está esquematizado por un rectángulo.

Caso de uso: refiere a un ovalo que está dentro del sistema, que incluye un texto descriptivo del proceso.

Paquete: es un elemento opcional, pero muy útil en diagrama complejos, porque agrupa a los casos de uso⁷⁵.

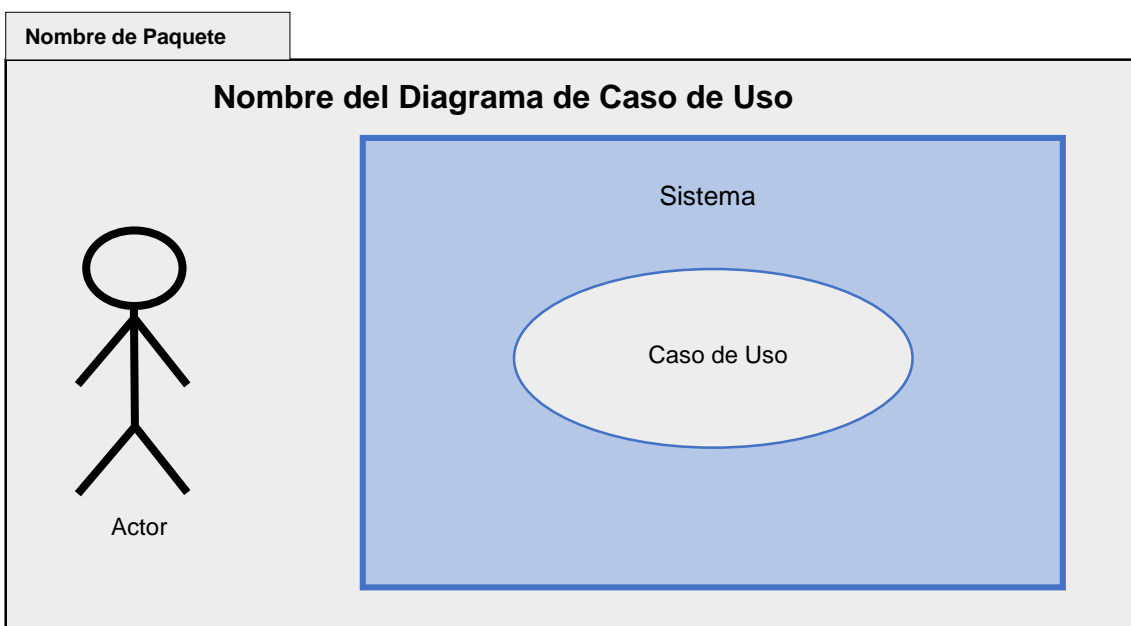


Gráfico N° 57 – Composición de un diagrama de caso de uso.

- **Relaciones**

Las relaciones son la comunicación que hay entre el actor y los casos de uso, puede describirse por una línea simple, pero utilizan etiquetas llamados “incluir” y “extender”.

⁷⁴ Ibíd.

⁷⁵ CREATELY, Karla. “Tutorial de diagramas de casos de uso”, 2021, <https://creately.com/blog/es/diagramas/tutorial-diagrama-caso-de-uso/> (consultada al 09 de mayo 2021).

Una relación de tipo “incluir” (usa) es el caso cuando necesita un caso de uso a otro para cumplir una tarea, mientras que tipo de relación “extender” da alternativas para determinados casos de uso⁷⁶.

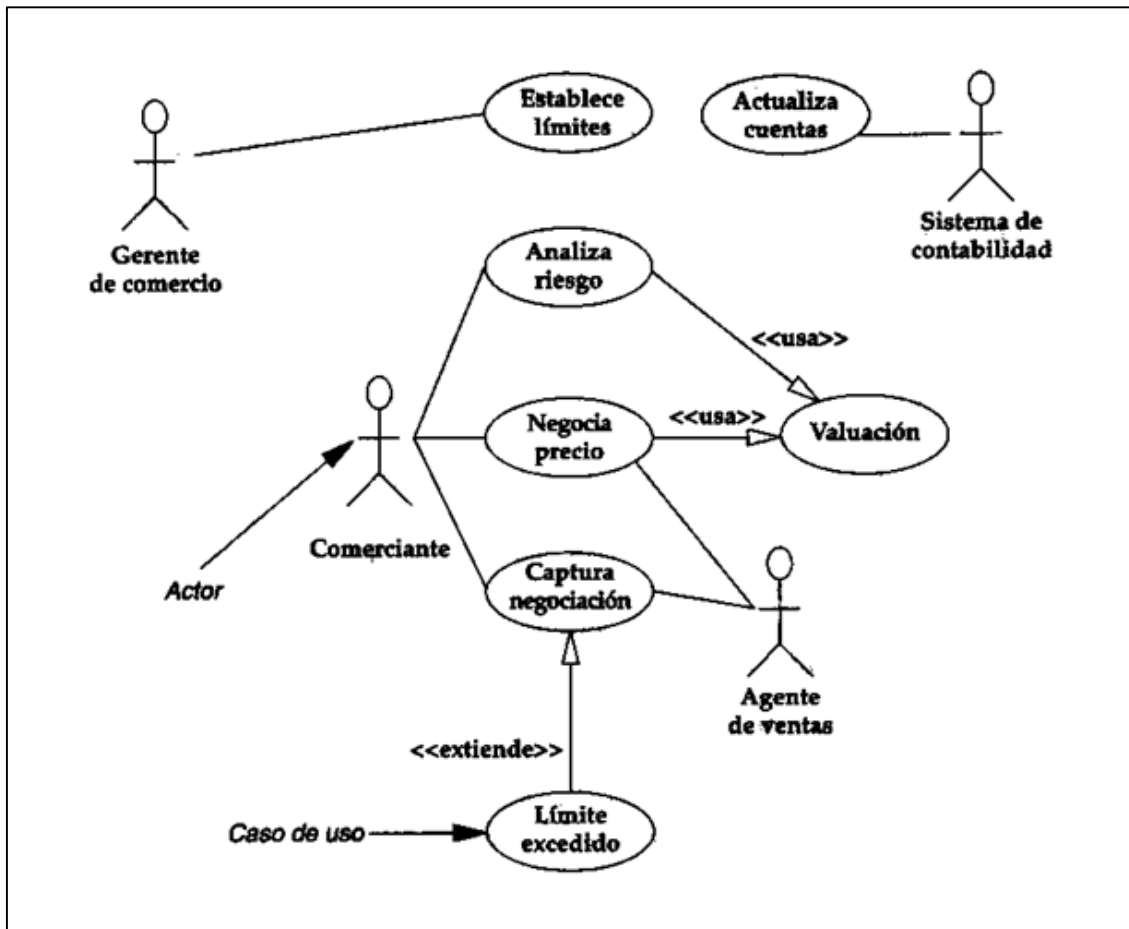


Gráfico N° 58 – Relaciones de un diagrama de caso de uso.

Relaciones de los casos de uso: se conocen ciertos tipos de relaciones, de acuerdo al comportamiento de los diagramas de caso de uso; los cuales son:

Relación	Símbolo	Concepto
Comunica	————	Es para conectar un actor con un caso de uso (no tiene punta de flecha).
Incluye	← - - - -	El caso de uso que tenga un comportamiento común con otro, puede usar la flecha con la etiqueta incluye.
Extiende	- - - - →	Si el caso de uso está conectado con un caso de uso básico, la flecha apunta del extendido al básico.
Generaliza	————→	Esta flecha se utiliza, en términos de UML “cosa”, cuando el caso de uso apunta a la cosa general.

Tabla N° 03 – Cuadro de tipo de relaciones que tiene los casos de uso. Elaboración propia

⁷⁶ Ibíd.

2.2.6. Lenguajes de programación:

Lenguajes de programación son las herramientas, en el cual, los individuos pueden interactuar con los ordenadores informáticos para crear productos que llegan ser aplicativos de software, es decir, un lenguaje de programación da la facilidad de comunicarnos con los equipos informáticos mediante instrucciones planteados para el formato del compilador, las instrucciones son conocidos como algoritmos, y así el equipo informático pueda interpretar la idea del usuario a lenguaje máquina. Los lenguajes de programación permiten tener una gran gana de beneficios, más ahora en pleno siglo XXI, toma un papel protagónico ya que ayudan a construir a las tecnologías de la información, para que así, se puedan automatizar los procesos a ciertas necesidades de los usuarios. En un contextual de la actualidad, que es la pandemia de COVID – 19, los lenguajes de programación pueden llegar hacer un instrumento para abrirnos a nuevas oportunidades por la creación de aplicaciones que ayuden a la gestión de trabajo remoto. Pero en otro aspecto, los lenguajes de programación tienen 2 principales paradigmas para el desarrollo de aplicaciones, que son; programación estructurada y orientada objetos, aunque existe muchos más, son de definiciones de procesos específicos, mientras lo que sean mencionado antes son los más usados en la historia de los lenguajes de programación, pero además los lenguajes de programación más utilizados comúnmente son; C#, Python, PHP, GO, Kotlin, VisualBasic, Java, Javascript, entre otros más.

El lenguaje de programación según (Wilson, 1993), indica lo siguiente:

“un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras (...).”⁷⁷.

El lenguaje de programación según (Ureña, 2010), indica lo siguiente:

“Conjunto de reglas o normas que permiten asociar a cada programa correcto un cálculo que será llevado a cabo por un ordenador (sin ambigüedades).”⁷⁸.

⁷⁷ WILSON, Leslie y ROBERT, Clark. *Comparative Programming Languages, Second Edition*, Bredevoort, Netherlands, 1993, pág. 75 (En inglés).

⁷⁸ UREÑA, Carlos. *El concepto de Lenguaje de Programación*. España, Granada, 2010, pág. 4.

Los lenguajes de programación se pueden distinguir en tipo de nivel de programación, es decir, para construir una aplicación hay maneras de cómo elaborar ya sea más eficientemente o más trabajoso, por eso se clasifican de la siguiente manera:

- **Lenguaje de programación de bajo nivel:** es el nivel en donde se orienta al lenguaje o sintaxis de máquina, es decir, trabajas con la interfaz básica del hardware para programar aplicaciones que están necesariamente al formato que sea entendible para el compilador. Por ese motivo este nivel se subdivide en 2 partes:

Lenguaje máquina: es el lenguaje más primitivo, para que el computador pueda interpretar los comandos, que está constituido por una colección de binarios de 1 y 0. Ej.: 10101110.

Lenguaje ensamblador: fue el intento de sustituir al lenguaje máquina, pero como principal diferencia fue que, aquel lenguaje está almacenado en texto de una determinada serie de instrucciones para los microprocesadores, sin embargo, las máquinas no comprenden este lenguaje por lo que tiene que pasar por un programa llamado Ensamblador. Ej.: MOV AX, 11^a.

- **Lenguaje de programación de alto nivel:** es aquel nivel en donde pertenecen los actuales lenguajes de programación, aunque están estandarizado en idioma de palabras del inglés, es posible la interacción entre el usuario y la máquina sean mucho más sencilla, porque existen un *traductor*, que interpreta las sentencias del lenguaje de programación al lenguaje de máquina, pero también hay un *compilador*, que interpreta todo el programa y lo almacena para que luego se haya la necesidad de volver a interpretar. Además, el lenguaje de alto tiene varias generaciones.

Primera generación: lenguaje máquina y el ensamblador.

Segunda generación: los primeros lenguajes de alto nivel, Fortran, Cobol.

Tercera generación: nacen los lenguajes imperativos, Pascal, Algol 8.

Cuarta generación: nacen los lenguajes de gestión de base de datos, sintaxis SQL.

Quinta generación: son lenguajes más avanzados, diseñados para inteligencia artificial.

Los lenguajes de programación como se mencionó antes se puede tonar en 2 paradigmas muy importantes para los desarrolladores, los cuales son:

Programación Estructurado

Fue la mejora de calidad, claridad y tiempo del desarrollo en su tiempo, ya que este paradigma se ubica después de la segunda generación de los lenguajes de programación, es decir, es la mejora de los primeros lenguajes de programación porque optimizaba los procesos para ahorrar trabajo al desarrollador. Este paradigma surgió en la década de 1960 por los señores Böhm y Jacopini, con el teorema del programa estructurado, que está dotado por estructuras de control consistentes y con una buena formación⁷⁹.

La programación estructurada es la forma de programar más sencilla y fácil, porque se conforma de 3 estructuras, las cuales son:

- **Secuencial:** es la realización de las instrucciones de acuerdo a un orden.

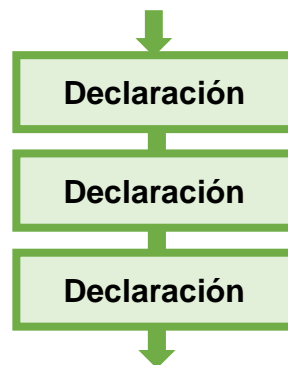


Gráfico N° 59 – Visualización de una estructura secuencial.

- **Selección:** es la base de una condicional o una sentencia de acuerdo al estado del programa, los cuales pueden ser if, then, else, endif, switch o case.

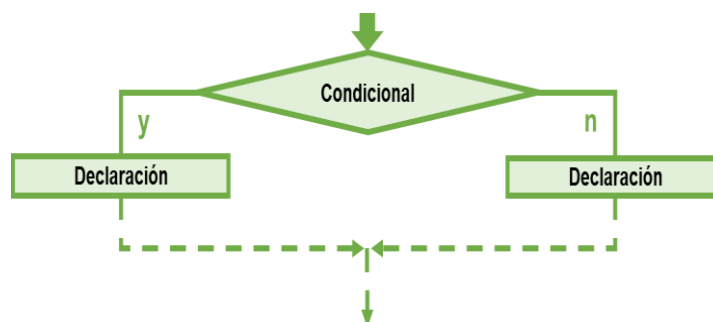


Gráfico N° 60 – Visualización de una estructura selección.

⁷⁹ GARCÍA-BERMEJO, José. *Programación estructurada en C – Primera Edición*, Pearson Prentice Hall.

- **Repetición:** es la realización de una sentencia de código en serie de redundancia hasta que el programa realice un estado determinado. Las expresiones usadas pueden ser While, Do While, For o Foreach, etc.

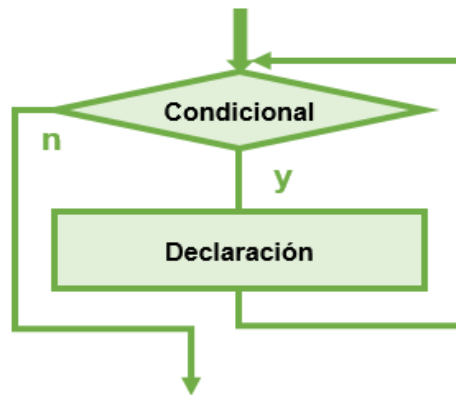


Gráfico N° 61 – Visualización de una estructura repetición.

Programación Orientado a objetos

Se caracteriza por llevar a elementos del mundo real al código, es decir, busca dar unas abstracciones de la vida real para plantear por secuencia de requerimientos para luego ser armado por una sintaxis de código. Este paradigma es reconocido por convertir las actividades del mundo real a entidades, esto gracias al modelo de la programación, ya que permite desarrollar los aplicativos de software orientado objetos.

Pero según (Joyanes, Rodríguez y Fernández, 2003), indica lo siguiente:

“En los lenguajes orientados a objetos el almacenamiento de información de diferentes tipos con un único nombre suele efectuarse en clases, pero las no almacenan sólo campos con el estado de un objeto, si no [SIC.] también su comportamiento”⁸⁰.

Mientras (Torres, 2016), indica lo siguiente:

“Se podría decir que la programación orientada a objetos usa mecanismos básicos para su implementación, los cuales interactúan entre sí, como las clases, los objetos, los mensajes y los métodos”⁸¹.

⁸⁰ JOYANES, Luis, RODRÍGUEZ, Luis y FERNÁNDEZ, Matilde. *Fundamentos de Programación – Segunda Edición*. Editorial Mc GRAW-HILL, España, Madrid, 2003, pág. 111.

⁸¹ TORRES, Manuel. *Programación Orientada a Objetos con Visual C# 2015 y ADO.NET 4.6*. Editorial Macro, Perú, Lima, pág. 21.

De acuerdo a los 2 autores citados, la programación orientados a objetos se caracteriza por tener varias subdivisiones, pero este argumento de característica será conformado por las fuentes de ambos autores⁸², las cuales son:

- **Objeto:** en un punto de vista contextual, es una entidad que tiene sus características y su comportamiento, además, el conjunto de estas conforma un sistema esto significa que los objetos deben estar instanciados por una clase determinada.
- **Clases:** en una definición simple, es la agrupación de objetos, pero considerando que las clases se definen por su nombre determinado, atributos, métodos que son las operaciones.

Nombre: se le da una definición única, comúnmente está asociada a la funcionalidad que tendrá.

Atributos: es la manera de dar las características a una clase, es decir, es la abstracción de los datos.

Métodos: permite que las clases tengan comportamiento, gracias a conjuntos de instrucciones en serie que son llamados en un momento determinado.

- **Modularidad:** es la manera de subdividir a la aplicación por lo que se crean módulos, que a su vez son partes más pequeñas del proceso que tiene una finalidad independiente al proceso.
- **Abstracción:** es la operacionalidad de dividir, las acciones comunes de la realidad a comunes objetos, es decir, busca dar un aspecto técnico a los requerimientos mediante métodos obtención de los datos para agruparlos en un conjunto de objetos y después almacenarlos en una clase.
- **Encapsulamiento:** como el propio nombre de la característica indica, es la manera de que los datos tengan capas de encubrimientos, el cual, se llama encapsular, pero en un entorno de programación se define la agrupación de datos y operaciones, de igual manera si hablamos de un objeto porque se convierte en algo independiente separándose de la clase que está relacionada, pero con diferencia que esta vez agrupa características y comportamiento.

⁸² JOYANES, Luis, RODRÍGUEZ, Luis y FERNÁNDEZ, Matilde. Óp. Cit. Pág. 357 y 370.
TORRES, Manuel. Óp. Cit. Pág. 18.

- **Herencia:** así como el polimorfismo, esta característica es fundamental para la programación orientada a objeto, ya que da las bases de implementar una clase padre que contiene sus características comunes entre las demás clases, que luego se hereda a una clase hijo con todas sus cualidades de la clase padre.
- **Polimorfismo:** es parte fundamental de la programación orientada objetos, por motivo que su aplicación permite proveer métodos con el mismo nombre, en una definición estrictamente literal, se refiere como la propiedad puede adoptar diferentes formas.

La programación orientada objetos ayuda a la organización de las relaciones de los objetos y de igual manera a la estructuración de las clases, pero un detalle que se debe mencionar, es que en los últimos 4 puntos de las características de la programación orientado objetos, están descrito los principales pilares del paradigma de programación orientada a objetos.



Gráfico N° 62 – Mapa conceptual de la programación orientada a objetos.

Metodología de desarrollo de software

Scrum: desarrollado por Ken Schwaber, Jeff Sutherland y Mike Beedle; definen que la gestión de los procesos del proyecto puede ser flexible con los cambios, pero sus principales características son 2, el cual el primero es el desarrollo mediante interacciones definidas por “sprints” que tienen duración de 30 días y la segunda es la importancia de las reuniones a lo largo del proyecto, lo cual destaca reunión diarias de 15 a 30 minutos para definir el desarrollo con todos los cambios y modificaciones integrados.

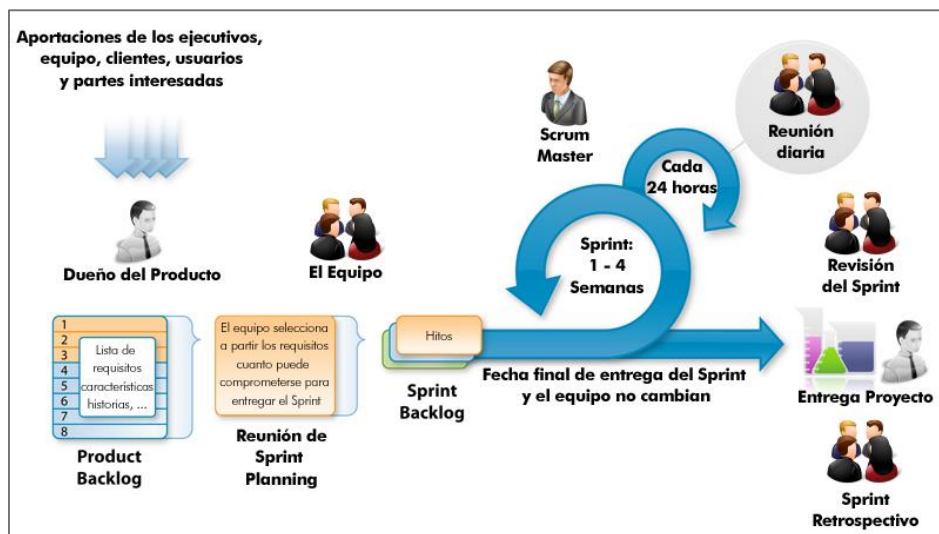


Gráfico N° 65 – Metodología de desarrollo de software SCRUM - análisis.

Arquitecturas de software

Si tenemos que definir la arquitectura de software, desde la fase de inicio que indica los requerimientos, hasta la finalización que es el entregable o solución.

- **Requerimientos:** se identifica las necesidades de los interesados con la información suficiente.
- **Diseño:** identifica las partes del sistema para satisfacer las necesidades de los interesados, pero también se constituyen los grupos de desarrollo.
- **Construcción:** es el desarrollo del software, en otras palabras, es el ensamble de los entregables de cada grupo de desarrollo para que se integre al producto final.
- **Pruebas:** es la realización de determinadas pruebas o verificaciones de procesos para que requerimiento sea cumplido de manera totalmente satisfactoria.
- **Implantación:** es la fase de transición del sistema a su entorno previamente programado, para su próxima entrada a producción.

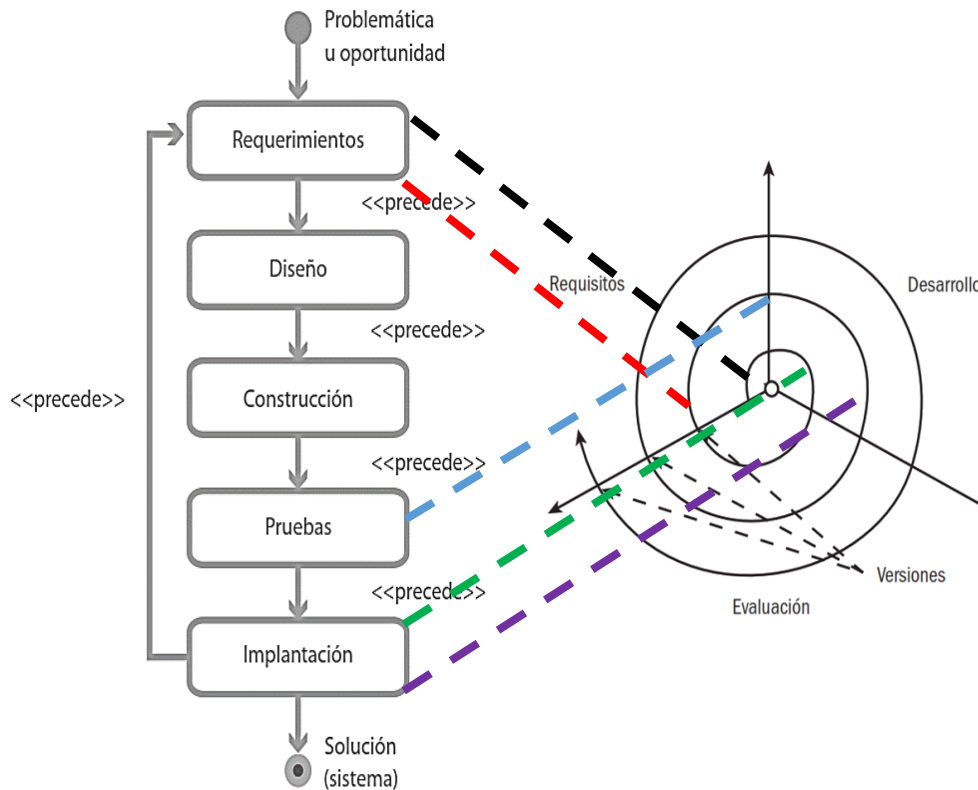


Gráfico N° 66 – Actividades del ciclo de desarrollo de la arquitectura (lado izquierda) y su mapeo dentro de las actividades técnicas del desarrollo software espiral (lado derecha) - análisis.

Patrón de diseño

Patrón Null Object: este patrón tiene la principal función de evitar valores nulos o vacíos a los objetos para que no resulte en un error en tiempo de ejecución de código. Para que cumpla con su función el patrón hace uso de interfaces con estructura vacía en vez de un valor nulo.

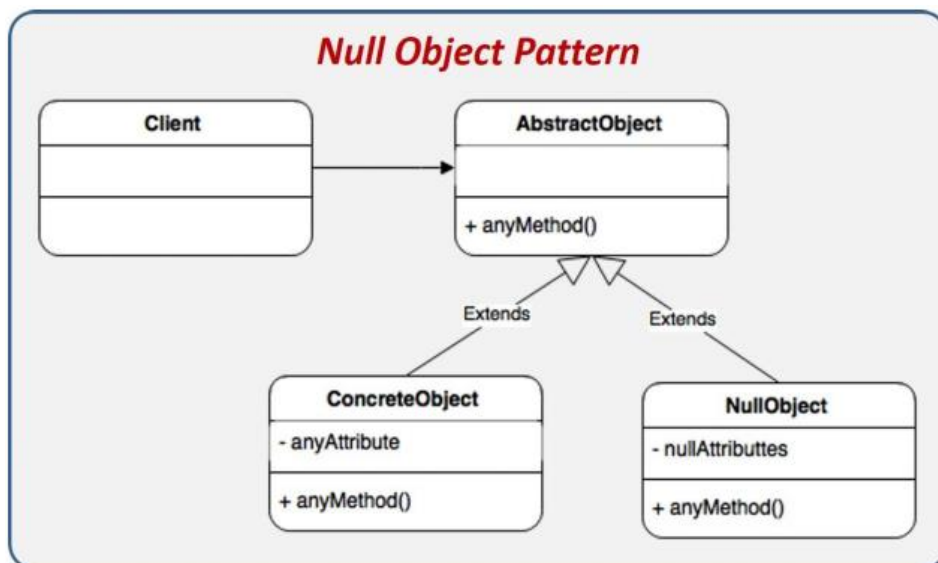


Gráfico N° 67 – Visualización del patrón de diseño Null Object - análisis.

Modelamiento y Diagramas:

- Requerimientos Funcionales:**

Nº	Requerimiento	Descripción	Prioridad
01	Gestión De Accesos	-Registrar, listar, actualizar y eliminar un usuario. -Actualizar contraseña de un usuario. -Asignar Roles de Usuario	5
02	Gestión De Administración	-Registrar, listar, actualizar y eliminar productos. -Registrar, listar, actualizar y eliminar entidades. -Registrar, listar, actualizar y eliminar contactos. -Registrar, listar, actualizar y eliminar trabajadores. -Registrar, listar, actualizar y eliminar actividades. -Registrar, listar, actualizar y eliminar tipo documentos. -Registrar, listar, actualizar y eliminar documentos. -Registrar, listar, actualizar y eliminar adquisición. -Registrar, listar, actualizar y eliminar nivel entidad. -Registrar, listar, actualizar y eliminar adquisición.	5
03	Gestión De Clientes	-Registrar, listar, actualizar y eliminar un cliente. -Listado rápido de los clientes con licencias.	5
04	Gestión de Ventas	-Registrar, listar, actualizar y eliminar una venta. -Registrar, listar, actualizar y eliminar detalle de venta. -Registrar, listar, actualizar y eliminar precios. -Registrar, listar, actualizar y eliminar seguimiento. -Registrar, listar, actualizar y eliminar campañas. -Registrar, listar, actualizar y eliminar asignaciones.	5
05	Gestión de Soporte	-Registrar, listar, actualizar y eliminar área. -Registrar, listar, actualizar y eliminar estado. -Registrar, listar, actualizar y eliminar persona. -Registrar, listar, actualizar y eliminar resultados. -Registrar, listar, actualizar y eliminar responsables. -Registrar, listar, actualizar y eliminar asignaciones. -Registrar, listar, actualizar y eliminar asistencia. -Registrar, listar, actualizar y eliminar tipo asistencia.	5
06	Generación de Informes o Reportes	-Generar reportes de control de pagos. -Generar reportes de ventas. -Generar reportes de clientes. -Generar reportes de trabajadores. -Generar reportes de acuerdo a las metas .	4

Tabla N° 04 – Cuadro de requerimientos funcionales. Elaboración propia

- Requerimientos No Funcionales:**

Nº	Requerimiento	Descripción	Prioridad
01	Seguridad	El ingreso al sistema estará restringido bajo contraseñas cifradas y usuarios definidos.	5
02	Portabilidad	El sistema debe brindar comodidad al usuario y a otras áreas que trabajan o necesitan del Área de personal.	5
03	Multiplataforma	El sistema deberá funcionar en distintos tipos de sistemas operativos y plataformas de hardware.	3
04	Rendimiento	El sistema debe soportar el manejo de gran cantidad de información durante su proceso.	3
05	Desempeño	El sistema no presentara problemas para su manejo e implementación.	1

Tabla N° 05 – Cuadro de requerimientos no funcionales. Elaboración propia

- **Diseño de Arquitectura:** este sería la arquitectura de desarrollo del software adaptándose al proceso de negocio.

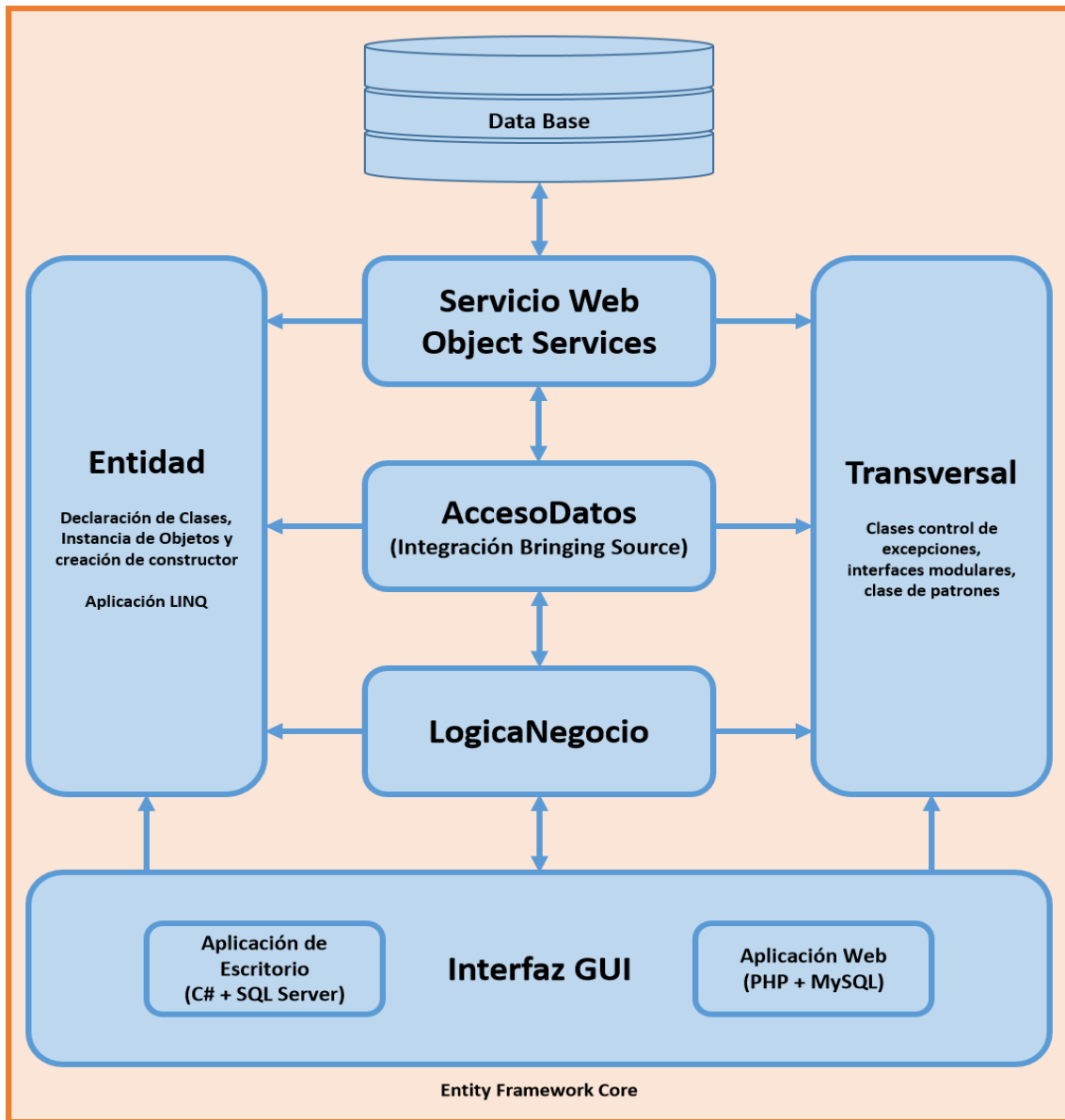


Gráfico N° 68 – Visualización de diseño de arquitectura - análisis.

- **Diseño Entidad – Relación (modelamientos de datos):**

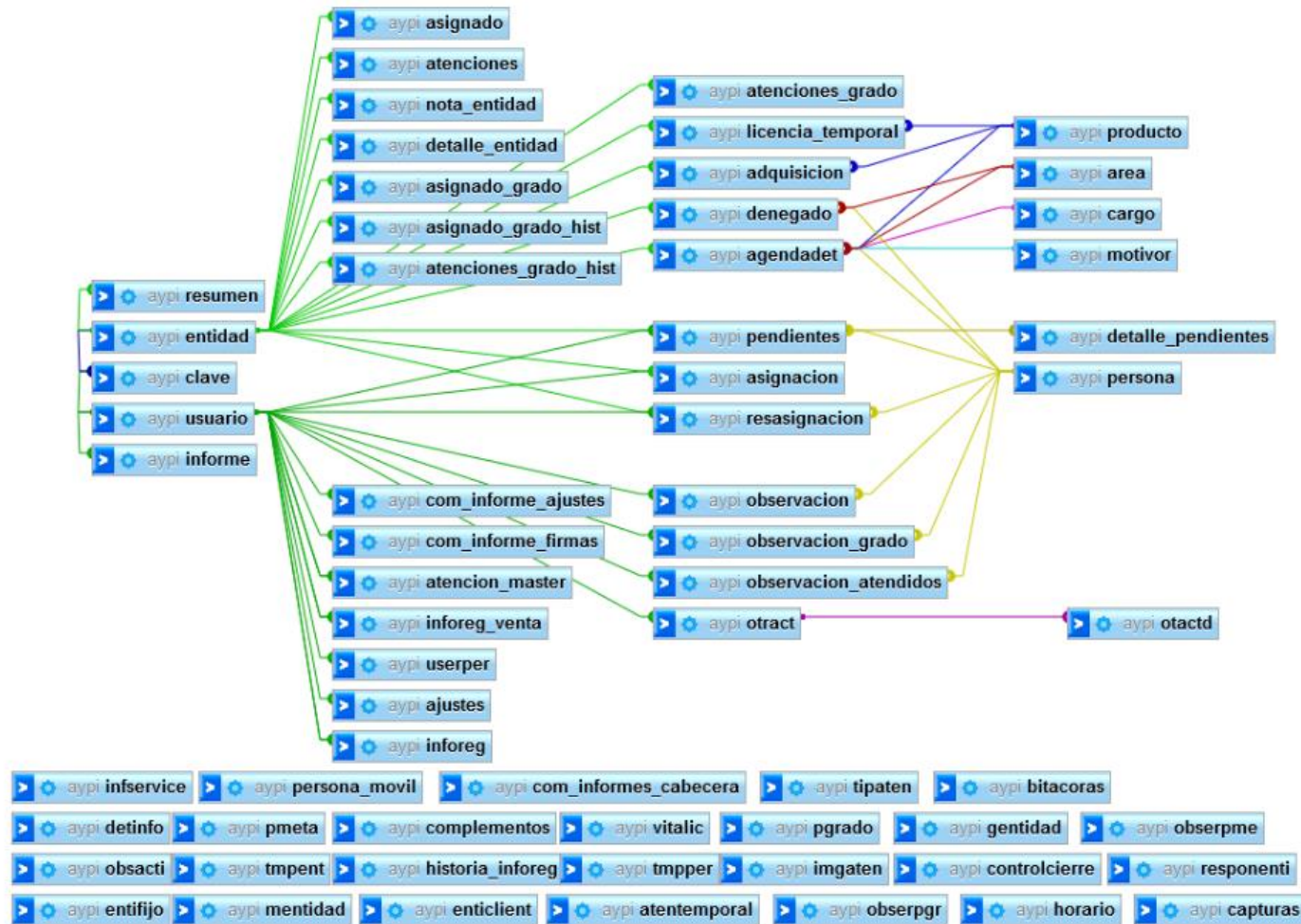


Gráfico N° 69 – Visualización de diseño entidad – relación - análisis.

- **Diagrama de Procesos:**

Diagrama de proceso, que explica el procedimiento actual de actualización de las licencias.

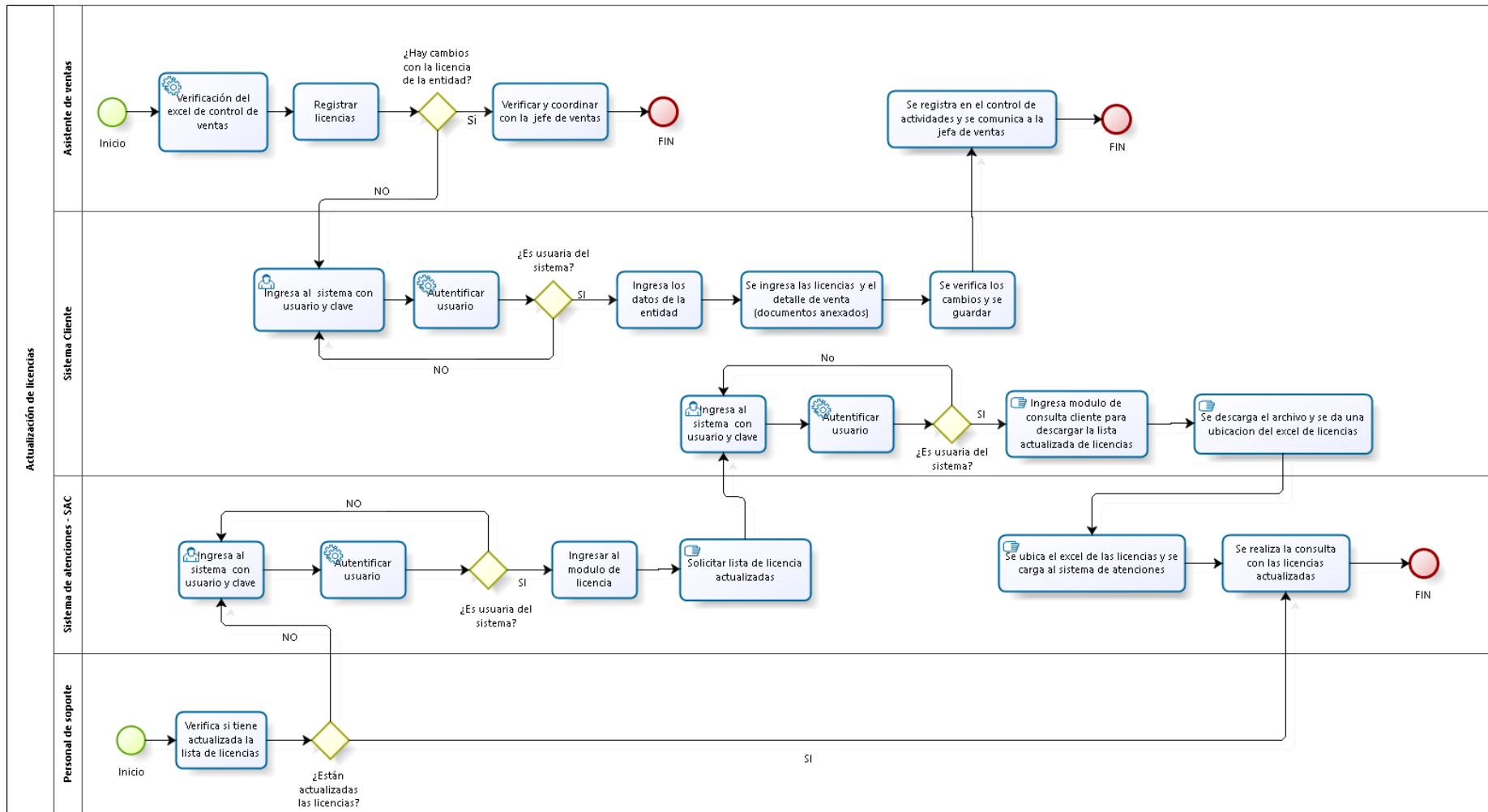


Gráfico N° 70 – Visualización de diagrama de procesos actual - análisis.

Diagrama de proceso, de cómo sería, la actualización de las licencias con la propuesta.

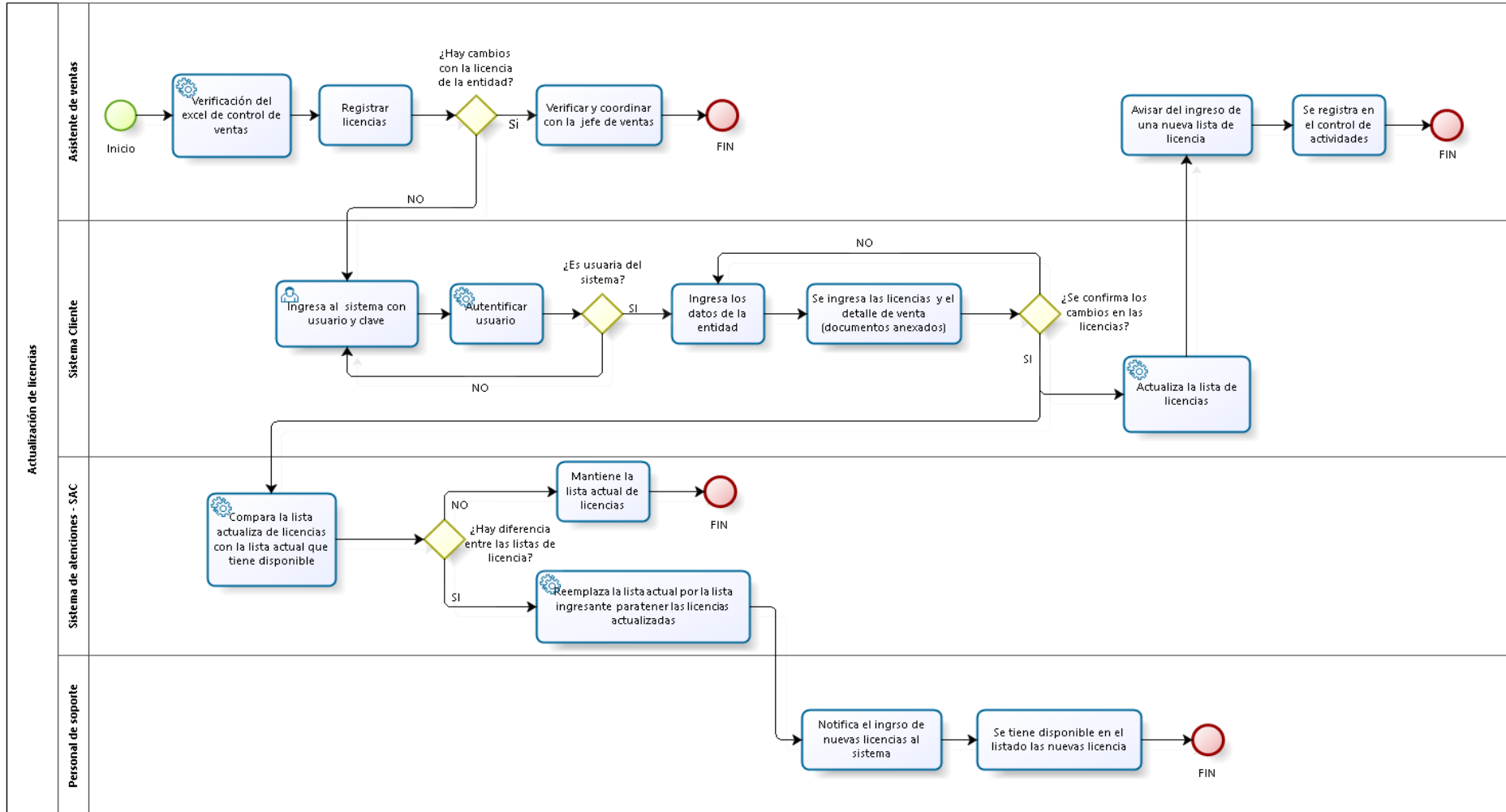


Gráfico N° 71 – Visualización de diagrama de procesos propuesto - análisis.

- **Diagrama de Flujo:** muestra el proceso en su entorno actual para la actualización de las licencias.

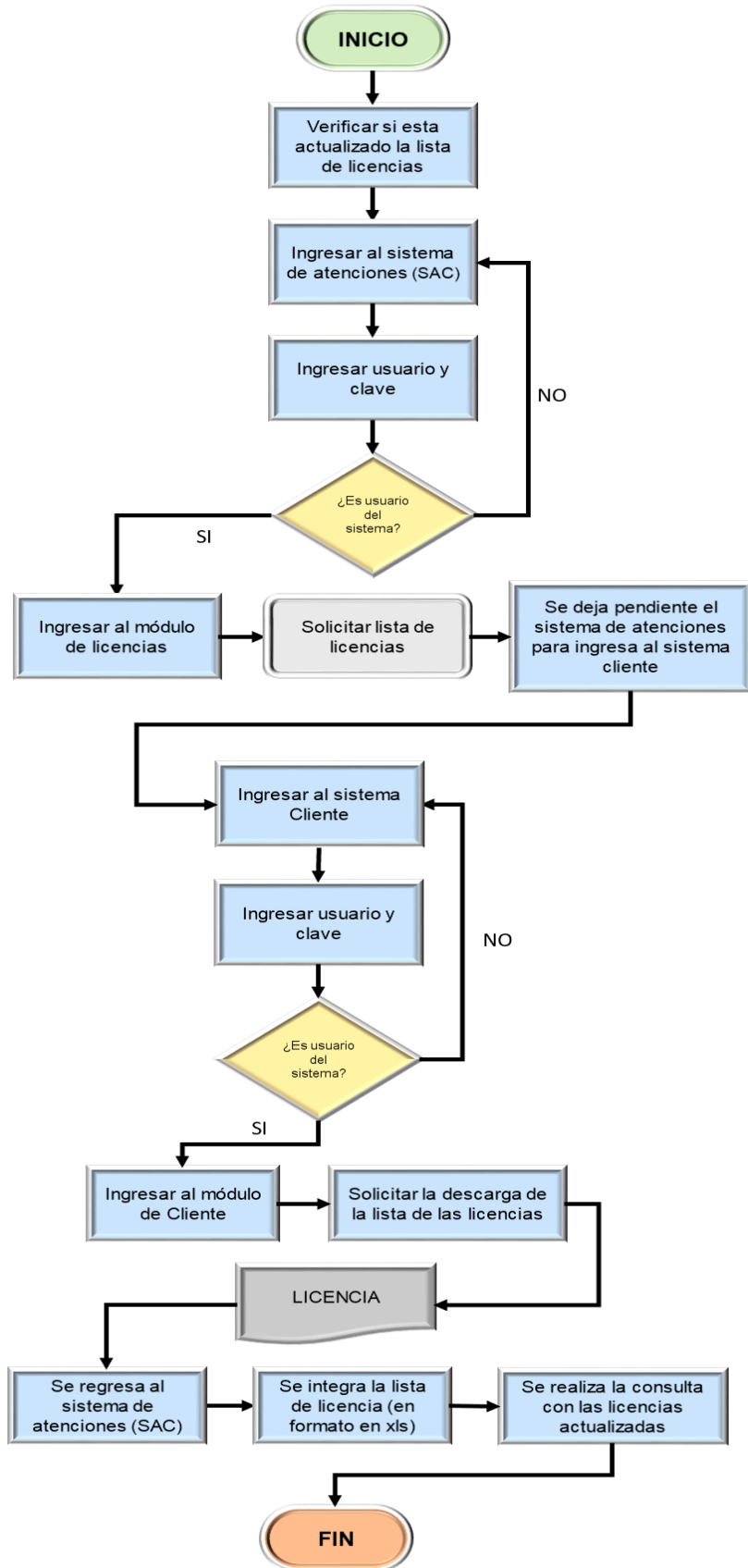


Gráfico N° 72 – Visualización de diagrama de flujo de proceso actual - análisis.

- **Diagrama de Casos de Uso:** se va a describir algunos casos de uso de para visualizar los procesos que puede integrar esta propuesta.

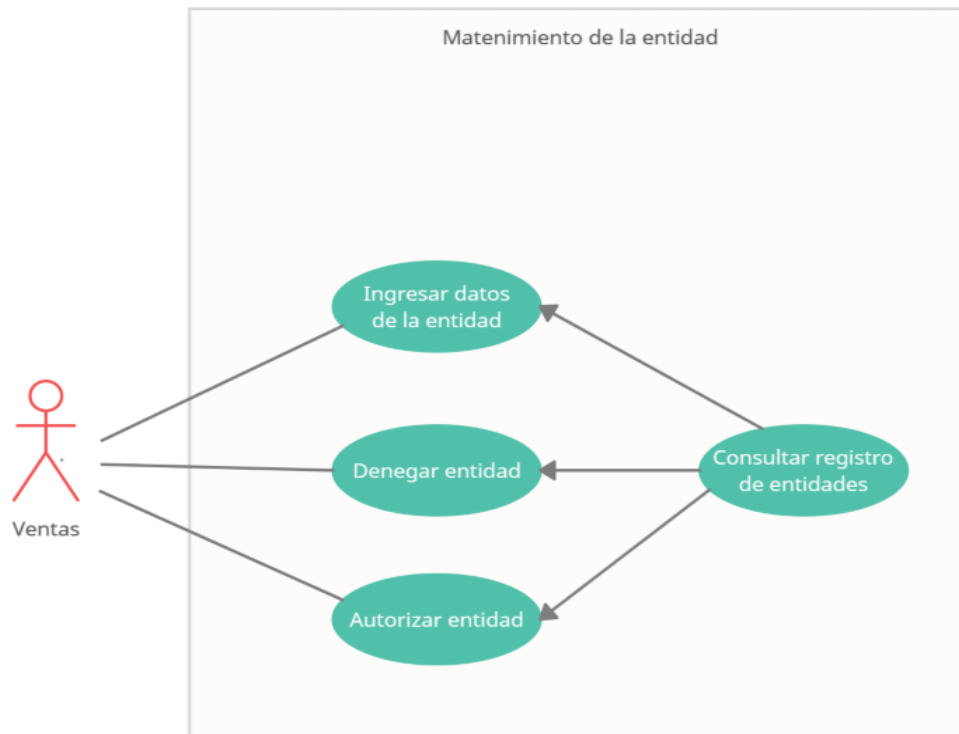


Gráfico N° 73 – Visualización de caso de uso del mantenimiento de la entidad - análisis.

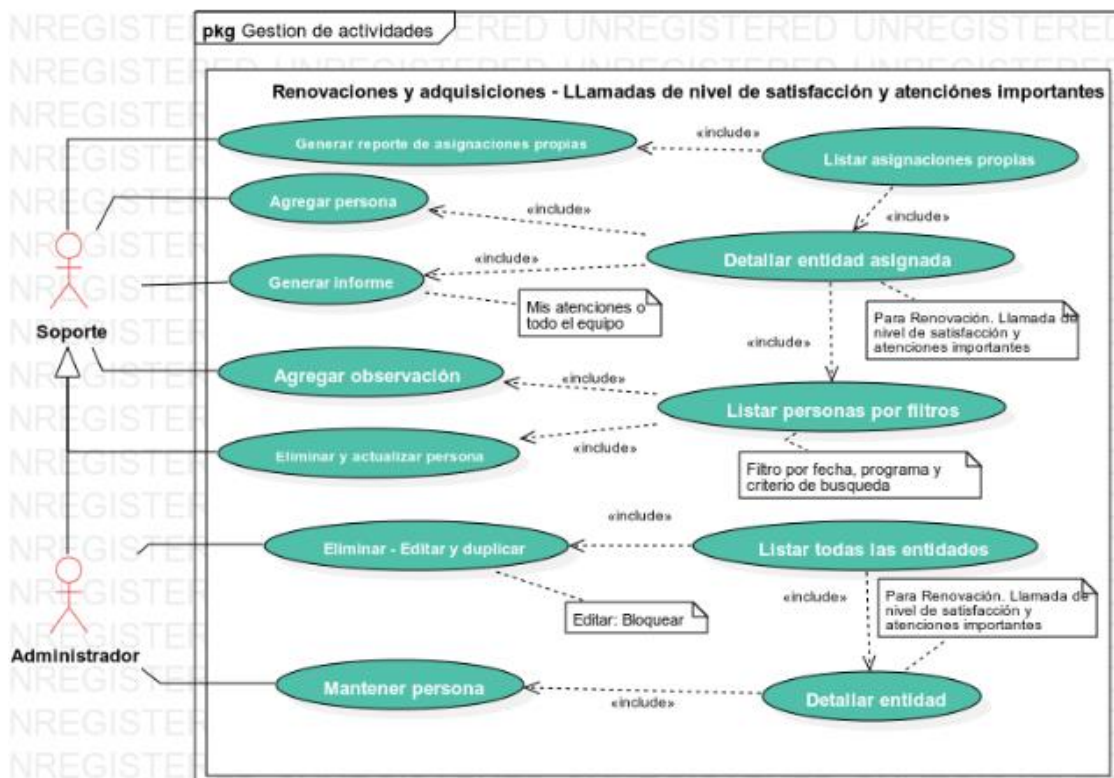


Gráfico N° 74 – Visualización de caso de uso de gestión de actividades - análisis.

Lenguajes de programación:

En esta parte del análisis del lenguaje de programación se tiene el conocimiento que actualmente el sistema Cliente está hecho en un lenguaje de programación estructural pero la propuesta intenta mejorar dicho sistema actualizando la plataforma, que es cambiar el lenguaje de programación de FoxPro v9.0 a uno más actual llamada C# con el gestor de base de datos SQL Server, que permitiría agregar más funciones y mejorar los existentes. Pero ahora se va explicar cómo está compuesto el actual sistema y como sería con la propuesta.

- **Sistema Cliente en la plataforma actual:**

Plataformas Tecnológicas: FoxPro v9.0.

Paradigma: Estructural.

Categoría de sistema: Sistema de información ejecutiva y gerencial (EIS y MIS).

Nivel de interacción: Medio.

Arquitectura: Interactivo repetitivo.

Patrón de diseño: Carece de uno.

Nivel Escalabilidad: Bajo.

- **Sistema Cliente con la propuesta de actualización de la plataforma:**

Plataformas Tecnológicas: C# y SQL Server.

Paradigma: Orientada objetos

Categoría de sistemas: Sistema de información ejecutiva y gerencial (EIS y MIS).

Nivel de interacción: Media – alta.

Arquitectura: Evolutivo.

Patrón de diseño: Null Object.

Escalabilidad: Alto.

2.4. Objetivos

2.4.1. Objetivo general:

Proponer la actualización de la plataforma del sistema Cliente, incidiría en la integración de procesos en la empresa Software y Sistemas del Perú sac – Iquitos 2021.

2.4.2. Objetivos específicos:

- Establecer la propuesta de actualización de la plataforma del sistema Cliente que mitigue el riesgo de error en los procesos internos en la empresa Software y Sistemas del Perú sac – Iquitos 2021.
- Establecer la propuesta de actualización de la plataforma del sistema Cliente para mejorar la administración de la información en la empresa Software y Sistemas del Perú sac – Iquitos 2021.
- Establecer la propuesta de actualización de la plataforma del sistema Cliente para mejorar la calidad de servicio en la empresa Software y Sistemas del Perú sac – Iquitos 2021.
- Establecer la propuesta de actualización de la plataforma del sistema Cliente para mejorar la satisfacción del personal en la empresa Software y Sistemas del Perú sac – Iquitos 2021.

2.5. Hipótesis

2.5.1. Hipótesis general:

Plantear una propuesta de actualización de la plataforma del sistema Cliente, incidiría en la integración de procesos en la empresa Software y Sistemas del Perú sac – Iquitos 2021.

2.5.2. Hipótesis específicas:

- La propuesta de actualización de la plataforma del sistema Cliente mitigará el riesgo de error en los procesos internos en la empresa Software y Sistemas del Perú sac – Iquitos 2021.
- La propuesta de actualización de la plataforma del sistema Cliente mejorará la administración de la información en la empresa Software y Sistemas del Perú sac – Iquitos 2021.
- La propuesta de actualización de la plataforma del sistema Cliente mejorará la calidad de servicio en la empresa Software y Sistemas del Perú – Iquitos 2021.
- La propuesta de actualización de la plataforma del sistema Cliente mejorará la satisfacción del personal en la empresa Software y Sistemas del Perú sac – Iquitos 2021.

2.6. Variables

Variable: Propuesta de actualización de la plataforma del sistema Cliente.

Variable	Definición Conceptual	Definición Operacional	Dimensiones	Indicaciones	Escala de medición
Variable Independiente X: Propuesta de actualización del sistema Cliente.	<p>Un sistema, en la definición informática es el cual se encarga de las actividades que apoyen con la administración, procesamiento, distribución y almacenamiento de la información a base de datos que son recopilados de acuerdo a los procesos principales o primordiales de una organización, además un sistema es la composición de interacciones que tiene un fin común ya sea su rubro.</p> <p>El proceso básico de un sistema para que cumpla su función es tener los siguiente componentes; los cuales son una entrada, proceso, salida y retroalimentación ya que conforman su característica principal de ingresar datos para una posterior salida con información mediante un proceso.</p>	<p>El grado de satisfacción con relación a la propuesta de la actualización de la plataforma del sistema Cliente, serán evaluados de acuerdo a los siguientes índices de evaluación () que son valorados por: Deficiente (1), Regular (2), Bueno (3).</p>	Mitigar el riesgo de error	<ul style="list-style-type: none"> - Las validaciones de los datos son integradas con el sistema. - Los registros de control permiten un seguimiento posterior de los datos. - Se cumple los requerimientos para terminar los procesos. - Se tiene una óptima interacción del usuario con el sistema. 	<p>Cualitativa Ordinal</p> <p>Deficiente (1)</p> <p>Regular (2)</p> <p>Bueno (3)</p>
			Administración de la información	<ul style="list-style-type: none"> - La estructura de las herramientas del sistema está organizada eficientemente. - La gestión de recursos del ordenador informático con la información es óptima. - La verificación de los datos ingresados al sistema es efectiva. - El uso del sistema permite que los proceso principales y secundarios sean fluidos. 	<p>Cualitativa Ordinal</p> <p>Deficiente (1)</p> <p>Regular (2)</p> <p>Bueno (3)</p>
			Calidad de servicio	<ul style="list-style-type: none"> - La transferencia de información es eficaz entre el personal. - El tiempo de espera para enviar y recibir información es reducido. - El personal nunca está saturado de procesos o pendientes. - Cuán importante es aumentar la disponibilidad de servicio para las consultas de los clientes. 	<p>Cualitativa Ordinal</p> <p>Deficiente (1)</p> <p>Regular (2)</p> <p>Bueno (3)</p>
			Satisfacción del personal	<ul style="list-style-type: none"> - Se distribuye de forma eficiente la comunicación entre el personal con el sistema. - El uso del sistema automatiza la carga laboral del personal. - Debe ser de suma importancia la información de la empresa. - La satisfacción del personal tiene que ser mejorado con la gestión de la empresa. 	<p>Cualitativa Ordinal</p> <p>Deficiente (1)</p> <p>Regular (2)</p> <p>Bueno (3)</p>

Tabla N° 06 – Operacionalización de variables.

CAPÍTULO III. MATERIALES Y MÉTODOS

3.1. Lugar y Desarrollo de la Investigación

El proyecto de investigación se realizará en las instalaciones de la empresa Software y Sistemas del Perú sac., con su sede principal ubicado en la Ricardo Palma N° 110. El desarrollo de la investigación es con enfoque cuantitativo.

3.2. Tipo y Diseño de Investigación

3.2.1. Tipo de Investigación

En esta categoría de trabajos de investigación existen cuatro tipos de estudios, los cuales son: exploratorio, descriptivo, correlacional y explicativo. Entonces como esta tesis se basa a un tipo de estudio investigativo “**descriptivo**”, que según (Hernández, 2006), comunica lo siguiente: “**busca especificar las propiedades, características y los perfiles de personas, grupos, comunidades, procesos, objetos, o cualquier otro fenómeno que se someta a un análisis**”⁸³, mediante este método buscar descomponer el problema para verificar sus diversos aspectos y dar un análisis objetivo.

3.2.2. Diseño de investigación

Este estudio “**no experimental**” según el autor Kerlinger y Lee (2002), indican que para un “**“En la investigación no experimental no es posible manipular las variables o asignar aleatoriamente a los participantes o los tratamientos”. De hecho, no hay condiciones o estímulos planeados que se administren a los participantes del estudio. (...)**”⁸⁴, es decir, no se pretende llegar a un análisis experimental o que involucre una implementación, sino que, busca ver el objetivo que se describen sus contexto a la situación actual. Pero este estudio busca dar pautas como se daría esta investigación mediante la valoración económico, cronológico y las métricas correspondientes para tener un objetivo claro de cómo sería la implementación del proyecto de la actualización de la plataforma del sistema Cliente. En resumen, es un estudio estrictamente de la variable, de cómo sería su comportamiento o condición a determinada circunstancias. Se analiza hechos ya existentes, pero no se construye ninguna situación.

⁸³ HERNÁNDEZ, Roberto. *Metodología de la investigación – Cuarta edición*. México, México D.F., 2006, pág. 102.

⁸⁴ *Ibíd.* Pág. 205. La negrita es propia.

3.3. Población y Muestra

3.3.1. Población

La población estuvo conformada por los trabajadores de la empresa Software y Sistemas del Perú SAC. – Iquitos, que en total serían de 14 integrantes.

3.3.2. Muestra

La muestra estuvo conformada por el personal de las áreas de soporte y ventas que son 14 personas que comprende el 100% de total de la población de la empresa Software y Sistemas del Perú sac.

3.4. Técnica, Instrumentos y Procedimientos de Recolección de Datos

3.4.1. Técnica de recolección de datos

Se aplicó la técnica de recolección mediante encuesta para obtener la información de suma importancia y necesaria para la variable: propuesta de actualización de la plataforma del sistema Cliente.

3.4.2. Instrumentos de recolección de datos

Se utilizó el instrumento de recolección de datos mediante cuestionario para exponer las preguntas para identificación de la propuesta de actualización de la plataforma del sistema Cliente.

3.4.3. Procedimientos de recolección de datos

El procedimiento de recolección de datos que se utilizó mediante la encuesta, sirve a su vez para obtener información sobre la situación actual de la empresa, pero además que tecnologías tiene en su haber para plantear la propuesta de actualización de la plataforma del sistema Cliente a base de un análisis objetivo de la investigación, que a su vez se definirá por la características y tiempo empleado en este trabajo de investigación. Sin embargo, también se hará interacción con los empleados de la empresa mediante para que la recolección se más directa y clara.

3.4.4. Procesamiento de datos y análisis estadístico

Para el procesamiento de datos, se tiene que concluir con las etapas de recolección para iniciar la fase de procesamiento que al final es un estudio de los datos, sea interpretada como información, ya sea, por análisis estadístico descriptivos, por motivo que solo se tiene con la encuesta de pre test para elaborar los resultados, por motivo que el mismo estudio del trabajo de investigación solo tiene alcance a dar conclusiones descriptivas, el cual, a su vez va ser apoyada por el programa SPSS versión 22 en español que es compatible para el sistema operativo Windows 10 Pro.

CAPÍTULO IV. RESULTADOS

4.1. Análisis descriptivo de los datos de variable independiente:

PREGUNTA 01:

¿Cuál es el nivel de verificación o control de los datos en el sistema Cliente?

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	3	21,4	21,4	21,4
Válido Regular	8	57,1	57,1	78,6
Bueno	3	21,4	21,4	100,0
Total	14	100,0	100,0	

Tabla N° 07 – Pregunta 01. Elaboración propia

¿Cuál es el nivel de verificación o control de los datos en el sistema Cliente?

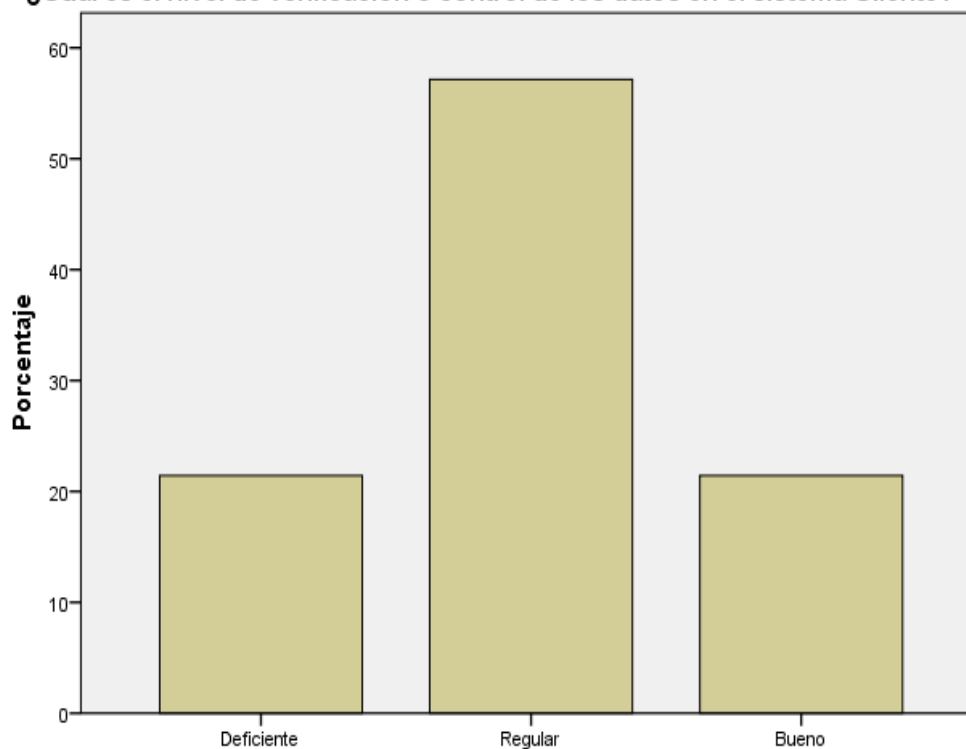


Gráfico N° 75 – Pregunta 01. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 07 y gráfico 75, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, el 57,1% indicaron que el nivel de verificación y control de los datos del sistema Cliente es “**regular**”, mientras que los siguientes valores tiene 21,4% de “**deficiente**” y 21,4% de “**bueno**”.

PREGUNTA 02:

¿Cuál es el nivel que permite el sistema Cliente de seguimiento de los datos ya sea registrados o listados?

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	0	0	0	
Válido Regular	9	64,3	64,3	64,3
Bueno	5	35,7	35,7	100,0
Total	14	100,0	100,0	

Tabla N° 08 – Pregunta 02. Elaboración propia

¿Cuál es el nivel que permite el sistema Cliente de seguimiento de los datos ya sea registrados o listados?

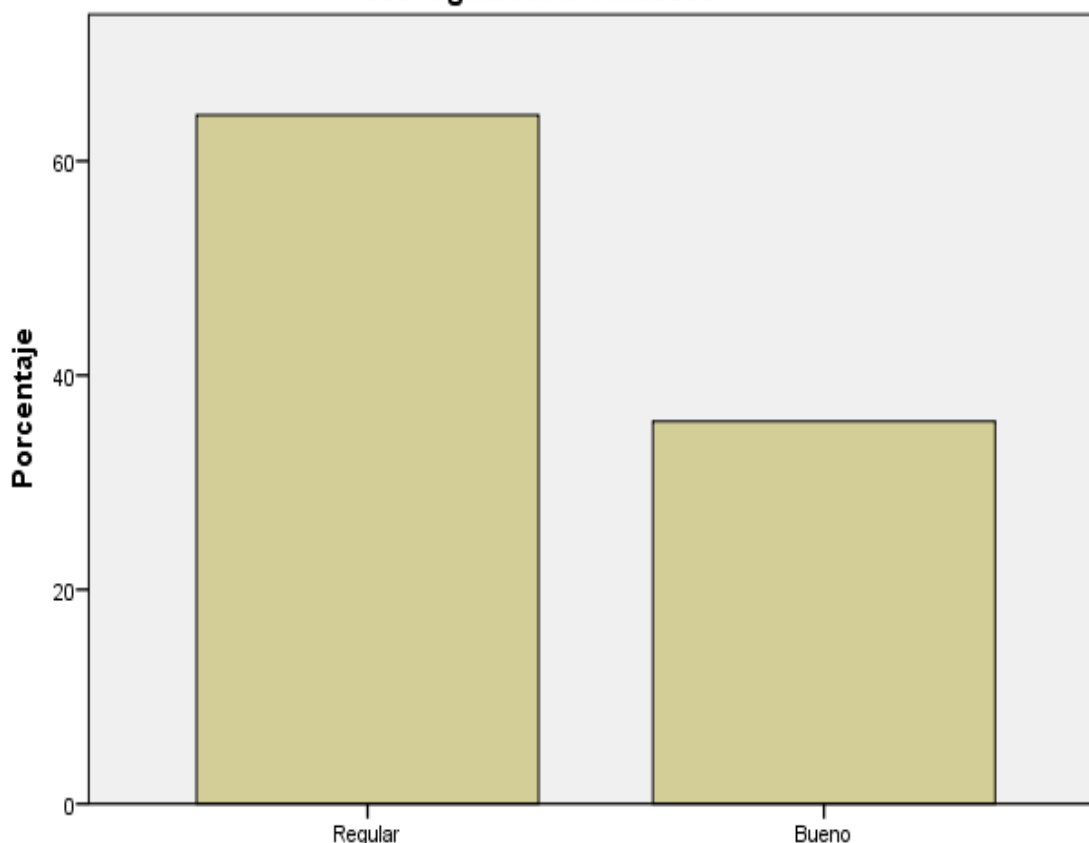


Gráfico N° 76 – Pregunta 02. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 08 y gráfico 76, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, el 64,3% indicaron que el nivel de seguimiento de los datos del sistema Cliente es “regular”, mientras que los siguientes valores tiene 0% de “deficiente” y 35,7% de “bueno”.

PREGUNTA 03:

¿Cuál es el nivel que cumple el sistema Cliente para cumplir sus procesos?

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	1	7,1	7,1	7,1
Válido Regular	9	64,3	64,3	71,4
Bueno	4	28,6	28,6	100,0
Total	14	100,0	100,0	

Tabla N° 09 – Pregunta 03. Elaboración propia

¿Cuál es el nivel que cumple el sistema Cliente para cumplir sus procesos?

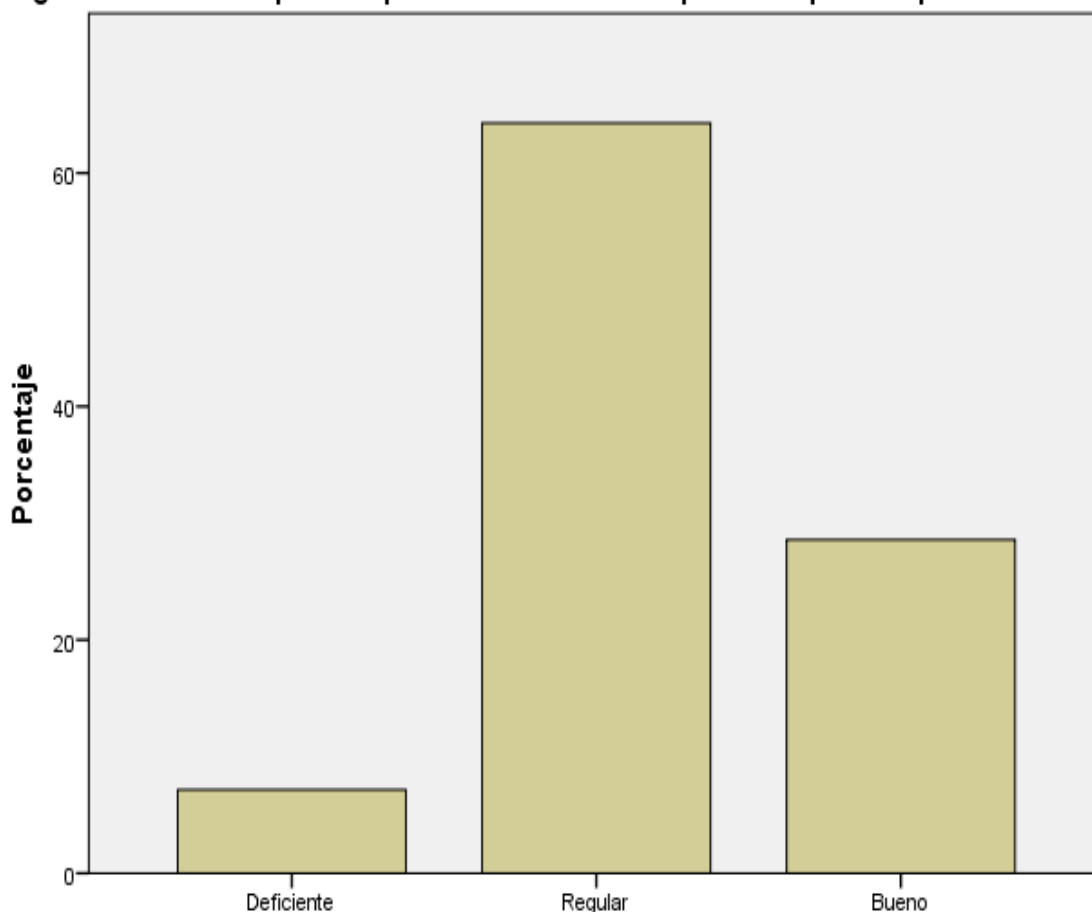


Gráfico N° 77 – Pregunta 03. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 09 y gráfico 77, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, el 64,3% indicaron que el nivel de cumplimiento de proceso del sistema Cliente es “**regular**”, mientras que los siguientes valores tiene 7,1% de “**deficiente**” y 28,6% de “**bueno**”.

PREGUNTA 04:

¿Cuál es el nivel de interacción del usuario con el sistema Cliente?

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	7	50,0	50,0	50,0
Válido Regular	6	42,9	42,9	92,9
Bueno	1	7,1	7,1	100,0
Total	14	100,0	100,0	

Tabla N° 10 – Pregunta 04. Elaboración propia

¿Cuál es el nivel de interacción del usuario con el sistema Cliente?

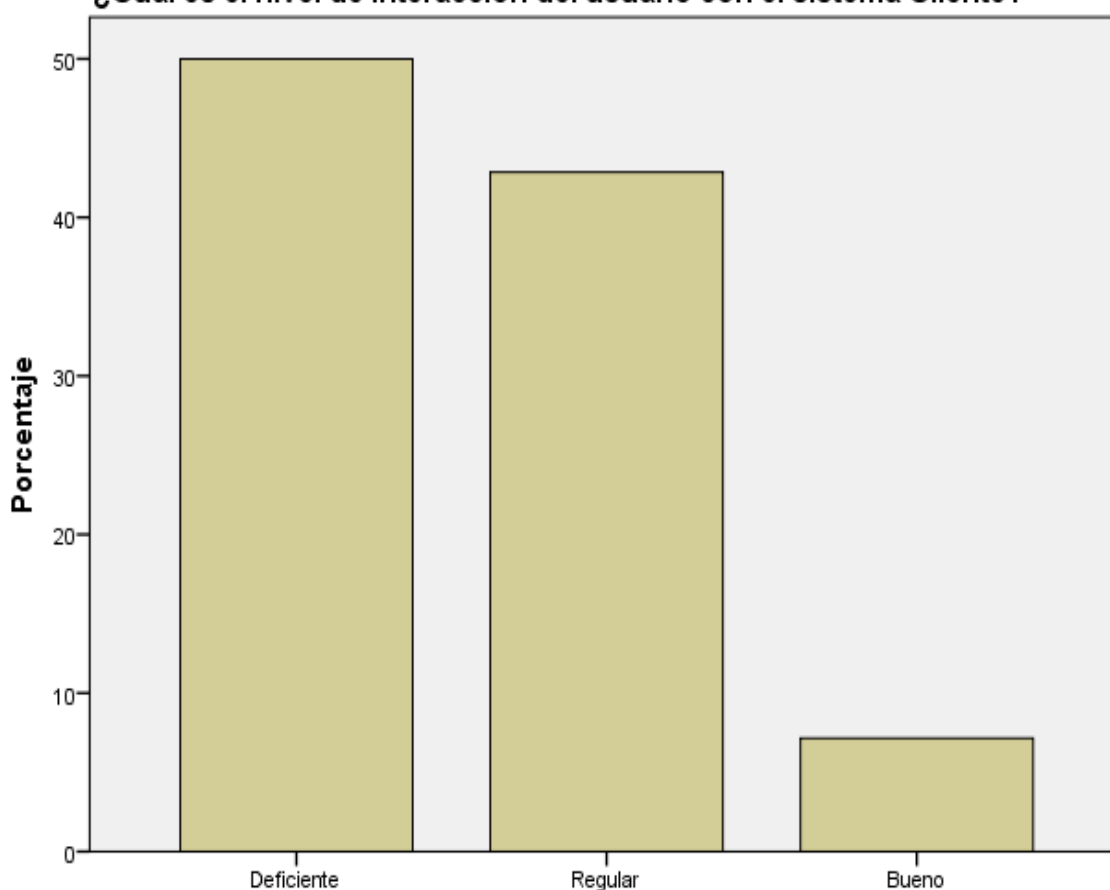


Gráfico N° 78 – Pregunta 04. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 10 y gráfico 78, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, el 50% indicaron que el nivel de interacción del sistema Cliente con el usuario es “deficiente”, mientras que los siguientes valores tiene 42,9% de “regular” y 7,1% de “bueno”.

PREGUNTA 05:

¿Cuál es el nivel de las estructuras de las herramientas del sistema Cliente?

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	2	14,3	14,3	14,3
Válido Regular	8	57,1	57,1	71,4
Bueno	4	28,6	28,6	100,0
Total	14	100,0	100,0	

Tabla N° 11 – Pregunta 05. Elaboración propia

¿Cuál es el nivel de las estructuras de las herramientas del sistema Cliente?

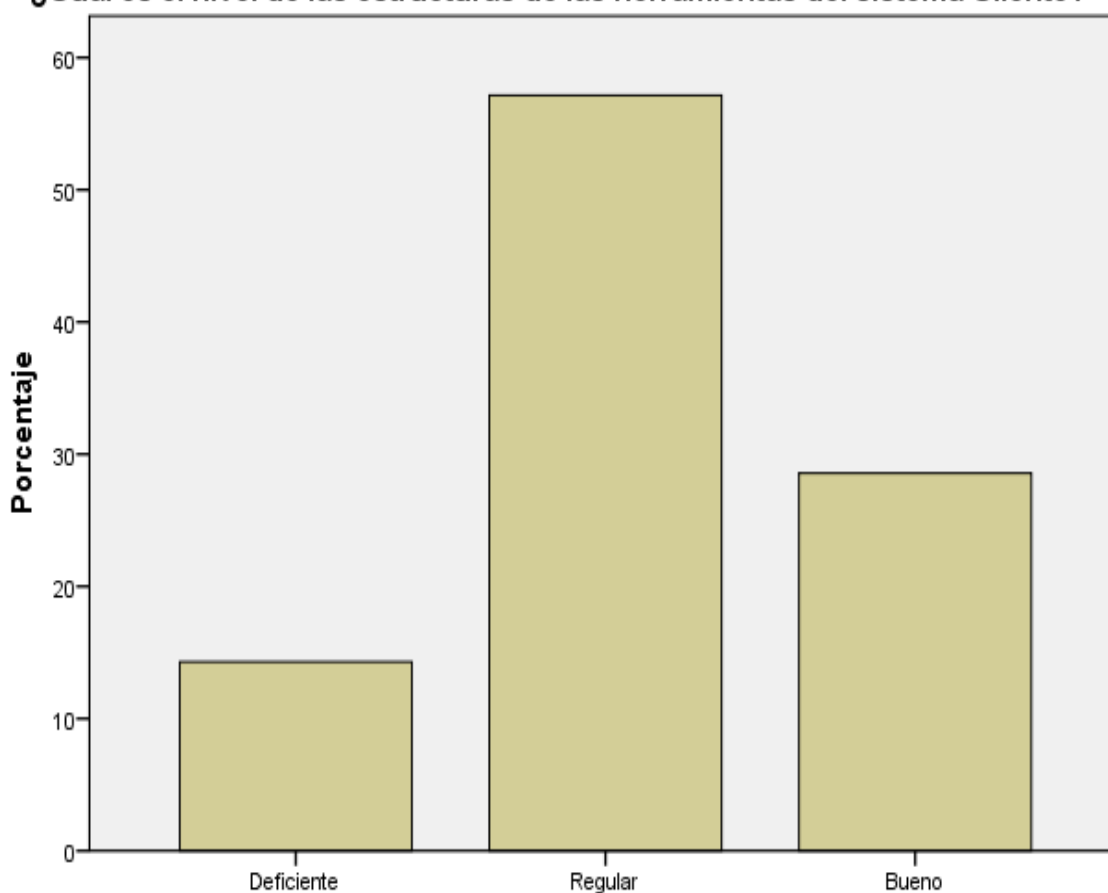


Gráfico N° 79 – Pregunta 05. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 11 y gráfico 79, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, el 57,1% indicaron que el nivel de las estructuras de las herramientas del sistema Cliente es “regular”, mientras que los siguientes valores tiene 14,3% de “deficiente” y 28,6% de “bueno”.

PREGUNTA 06:

¿Cuál es el nivel de la gestión de recursos del ordenador informático con la información?

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	3	21,4	21,4	21,4
Válido Regular	8	57,1	57,1	78,6
Bueno	3	21,4	21,4	100,0
Total	14	100,0	100,0	

Tabla N° 12 – Pregunta 06. Elaboración propia

¿Cuál es el nivel de la gestión de recursos del ordenador informático con la información?

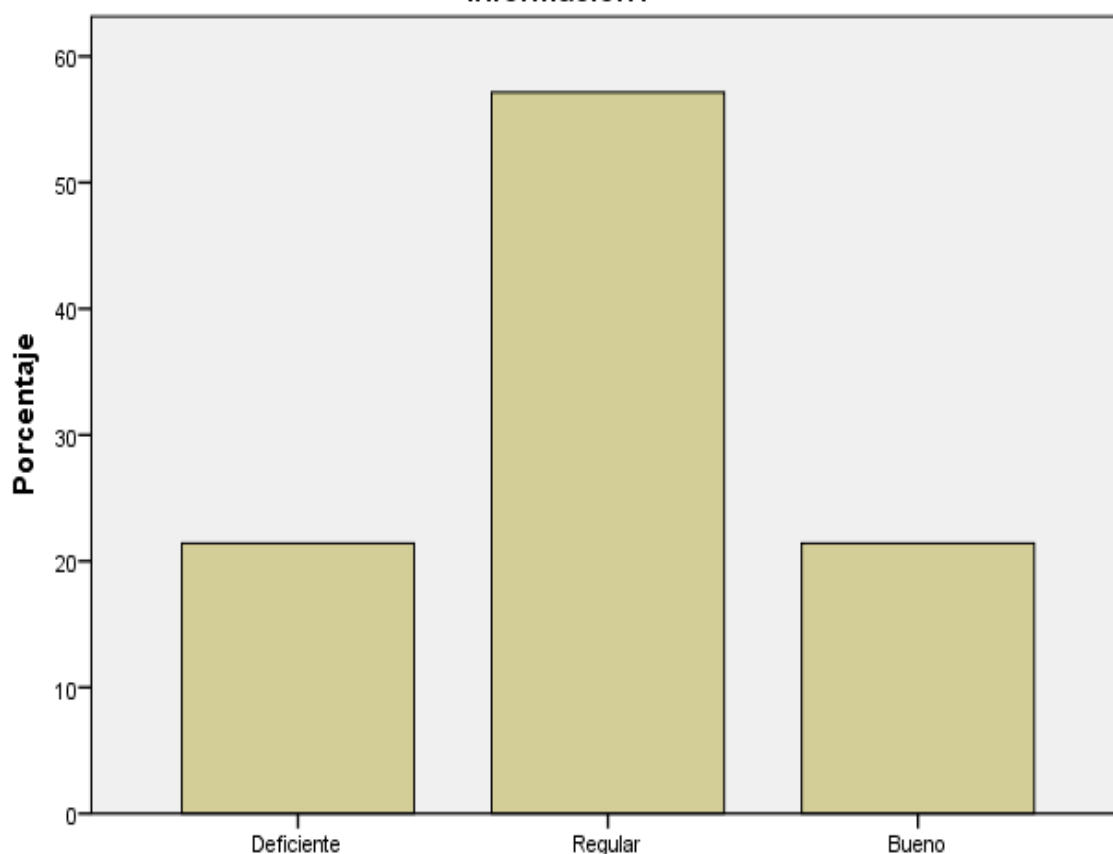


Gráfico N° 80 – Pregunta 06. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 12 y gráfico 80, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, el 57,1% indicaron que el nivel de gestión de recursos del ordenador es “regular”, mientras que los siguientes valores tiene 21,4% de “deficiente” y 21,4% de “bueno”.

PREGUNTA 07:

¿Cuál es el nivel de verificación de los datos ingresados al sistema Cliente?

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	3	21,4	21,4	21,4
Válido Regular	6	42,9	42,9	64,3
Bueno	5	35,7	35,7	100,0
Total	14	100,0	100,0	

Tabla N° 13 – Pregunta 07. Elaboración propia

¿Cuál es el nivel de verificación de los datos ingresados al sistema Cliente?

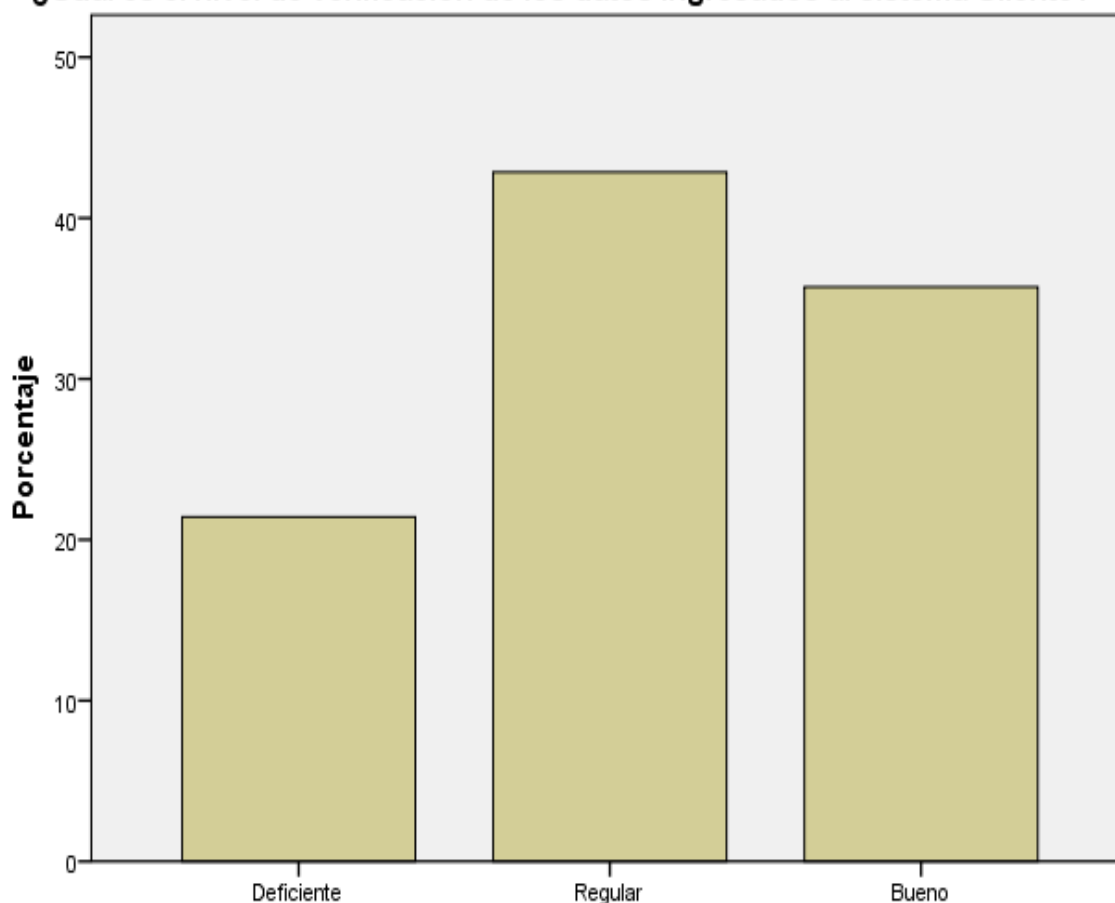


Gráfico N° 81 – Pregunta 07. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 13 y gráfico 81, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, el 42,9% indicaron que el nivel de verificación de los datos ingresados al sistema Cliente es “**regular**”, mientras que los siguientes valores tiene 21,4% de “**deficiente**” y 35,7% de “**bueno**”.

PREGUNTA 08:

¿Cuál es el nivel de uso fluido del sistema Cliente para realizar procesos principales y secundarios?

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	6	42,9	42,9	42,9
Válido Regular	5	35,7	35,7	78,6
Bueno	3	21,4	21,4	100,0
Total	14	100,0	100,0	

Tabla N° 14 – Pregunta 08. Elaboración propia

¿Cuál es el nivel de uso fluido del sistema Cliente para realizar procesos principales y secundarios?

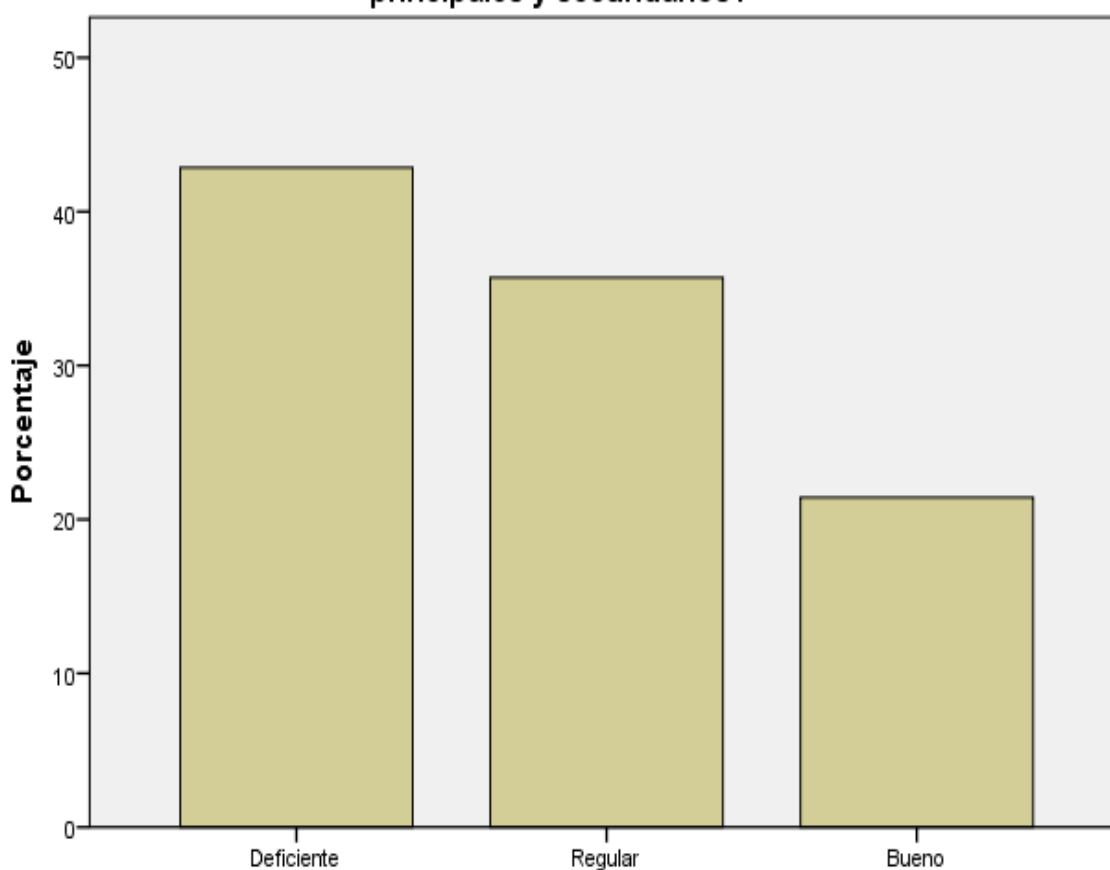


Gráfico N° 82 – Pregunta 08. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 14 y gráfico 82, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, el 42,9% indicaron que el nivel de fluidez para realizar procesos en el sistema Cliente es “deficiente”, mientras que los siguientes valores tiene 35,7% de “regular” y 21,4% de “bueno”.

PREGUNTA 09:

¿Cuál es el nivel de eficacia de la transferencia de información?

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	2	14,3	14,3	14,3
Válido Regular	11	78,6	78,6	92,9
Bueno	1	7,1	7,1	100,0
Total	14	100,0	100,0	

Tabla N° 15 – Pregunta 09. Elaboración propia



Gráfico N° 83 – Pregunta 09. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 15 y gráfico 83, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, el 78,6% indicaron que el nivel de eficacia de la transferencia de información en el sistema Cliente es “**regular**”, mientras que los siguientes valores tiene 14,3% de “**deficiente**” y 7,1% de “**bueno**”.

PREGUNTA 10:

¿Cuál es el nivel de tiempo de espera para enviar y recibir información?

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	2	14,3	14,3	14,3
Válido Regular	9	64,3	64,3	78,6
Bueno	3	21,4	21,4	100,0
Total	14	100,0	100,0	

Tabla N° 16 – Pregunta 10. Elaboración propia

¿Cuál es el nivel de tiempo de espera para enviar y recibir información?

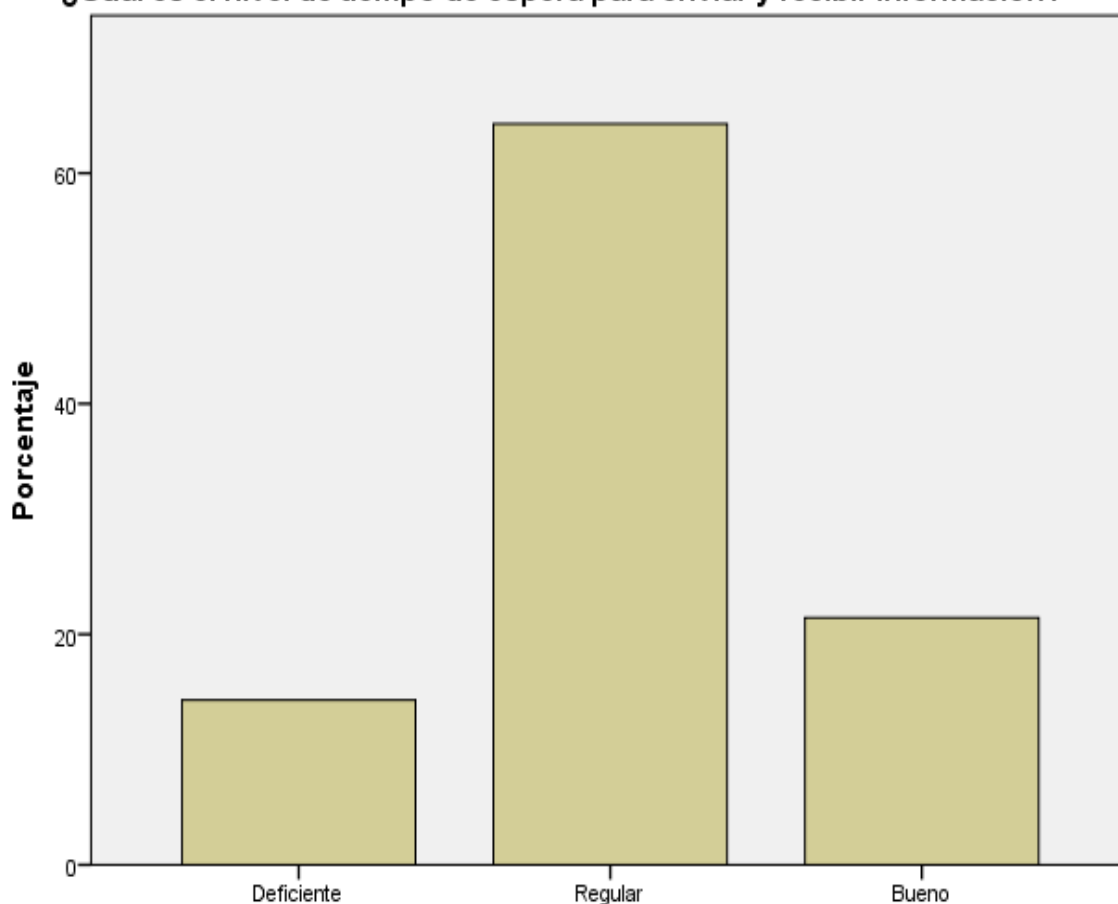


Gráfico N° 84 – Pregunta 10. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 16 y gráfico 84, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, el 64,3% indicaron que el nivel de tiempo de espera en el sistema Cliente es “regular”, mientras que los siguientes valores tiene 14,3% de “deficiente” y 21,4% de “bueno”.

PREGUNTA 11:

¿Cuál es el nivel de saturación de los procesos solicitados?

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	1	7,1	7,1	7,1
Válido Regular	6	42,9	42,9	50,0
Bueno	7	50,0	50,0	100,0
Total	14	100,0	100,0	

Tabla N° 17 – Pregunta 11. Elaboración propia

¿Cuál es el nivel de saturación de los procesos solicitados?

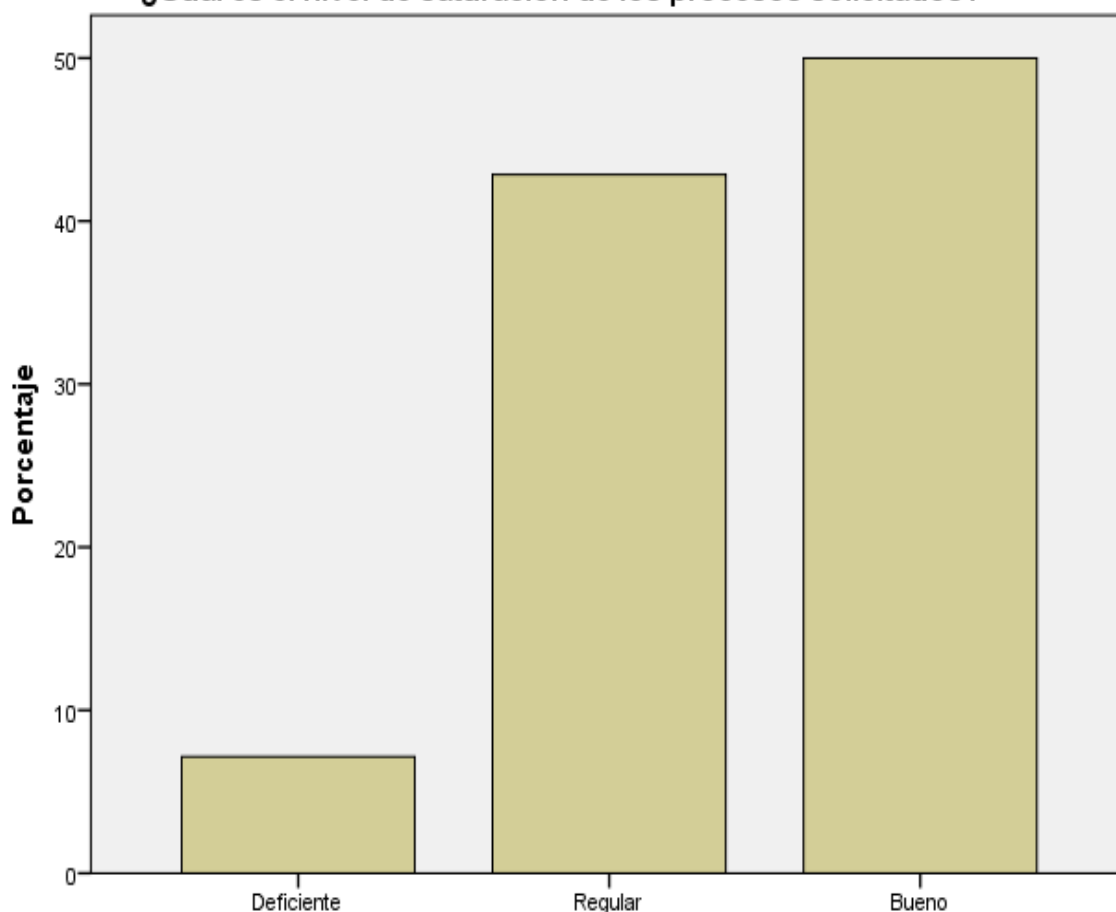


Gráfico N° 85 – Pregunta 11. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 17 y gráfico 85, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, el 50% indicaron que el nivel de saturación de los procesos del sistema Cliente es **“bueno”**, mientras que los siguientes valores tiene 42,9% de **“regular”** y 7,1% de **“deficiente”**.

PREGUNTA 12:

¿Cuál es el nivel de importancia de la disponibilidad de servicio del sistema Cliente?

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	1	7,1	7,1	7,1
Válido Regular	6	42,9	42,9	50,0
Bueno	7	50,0	50,0	100,0
Total	14	100,0	100,0	

Tabla N° 18 – Pregunta 12. Elaboración propia

¿Cuál es el nivel de importancia de la disponibilidad de servicio del sistema Cliente?

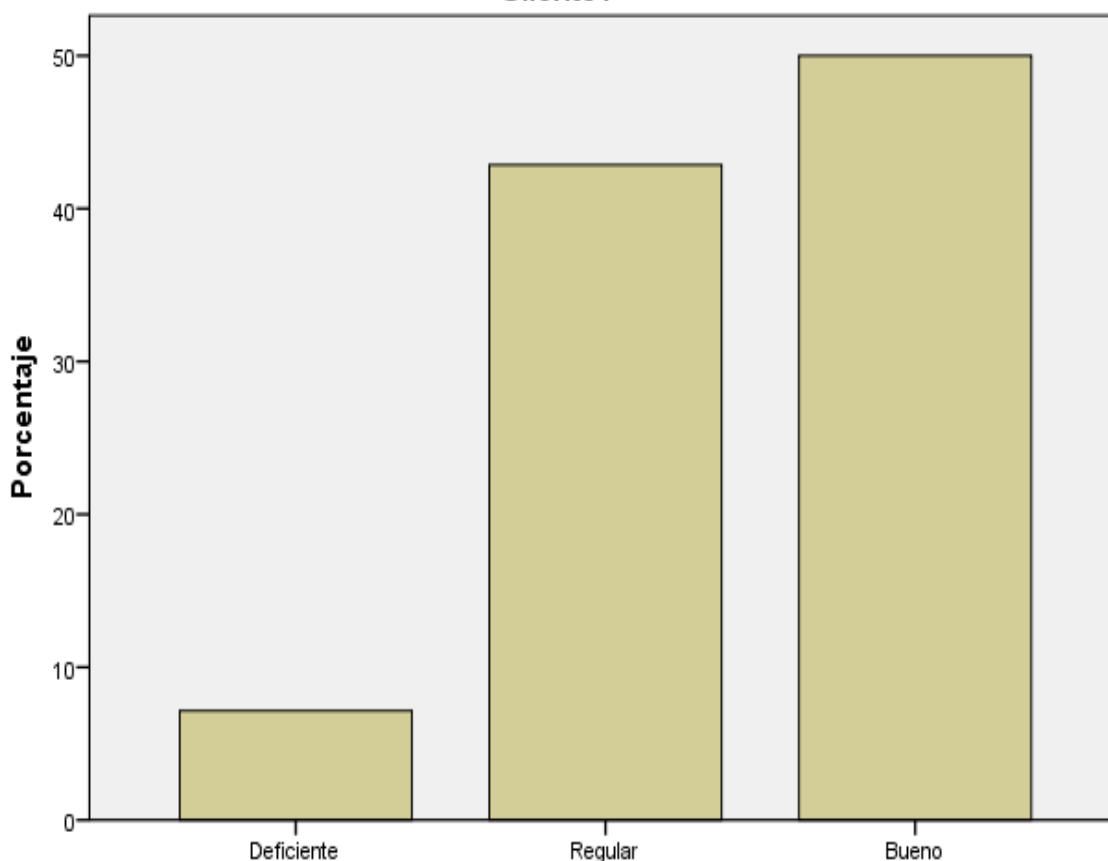


Gráfico N° 86 – Pregunta 12. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 18 y gráfico 86, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, el 50% indicaron que el nivel de importancia de disponibilidad de servicio en el sistema Cliente es “bueno”, mientras que los siguientes valores tiene 42,9% de “regular” y 7,1% de “deficiente”.

PREGUNTA 13:

¿Cuál es el nivel de eficiencia de la comunicación entre el personal con el sistema?

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	1	7,1	7,1	7,1
Válido Regular	12	85,7	85,7	92,9
Bueno	1	7,1	7,1	100,0
Total	14	100,0	100,0	

Tabla N° 19 – Pregunta 13. Elaboración propia

¿Cuál es el nivel de eficiencia de la comunicación entre el personal con el sistema?

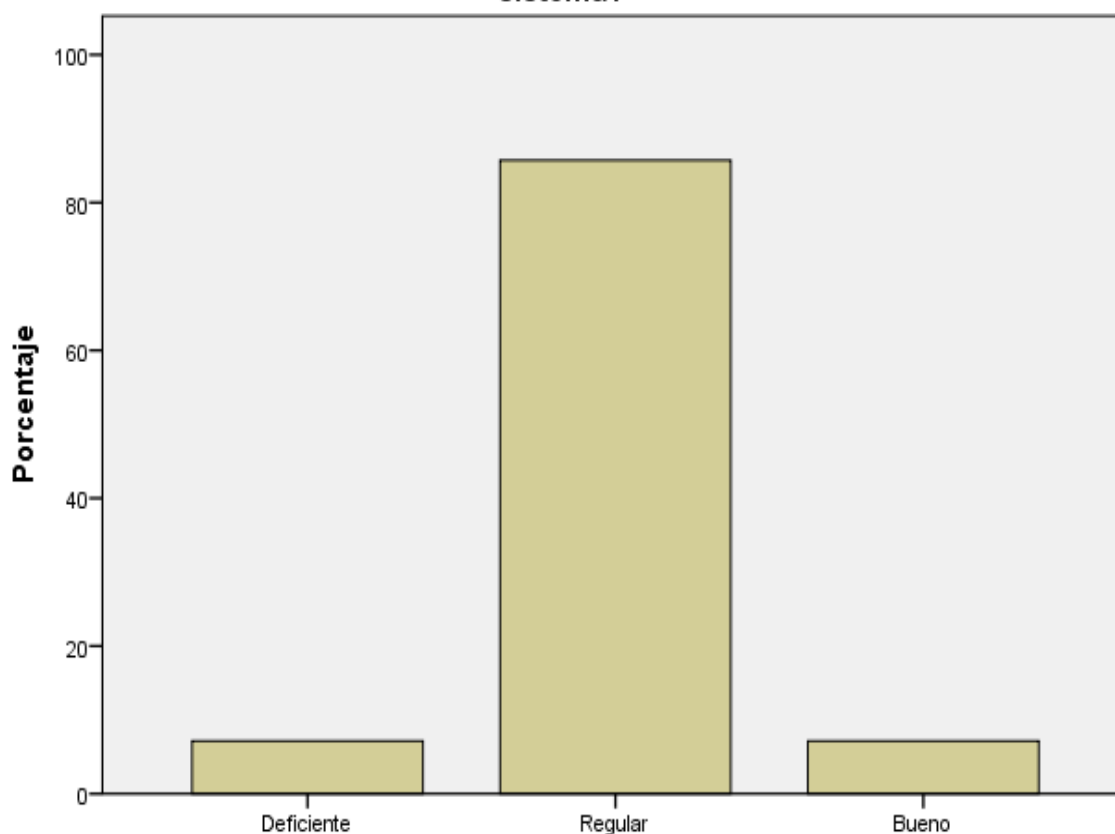


Gráfico N° 87 – Pregunta 13. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 19 y gráfico 87, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, el 85,7% indicaron que el nivel de eficacia de la comunicación entre personal con el sistema Cliente es “regular”, mientras que los siguientes valores tiene 7,1% de “deficiente” y 7,1% de “bueno”.

PREGUNTA 14:

¿Cuál es el nivel de automatización del sistema Cliente para la carga laboral del personal?

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	3	21,4	21,4	21,4
Válido Regular	5	35,7	35,7	57,1
Bueno	6	42,9	42,9	100,0
Total	14	100,0	100,0	

Tabla N° 20 – Pregunta 14. Elaboración propia

¿Cuál es el nivel de automatización del sistema Cliente para la carga laboral del personal?

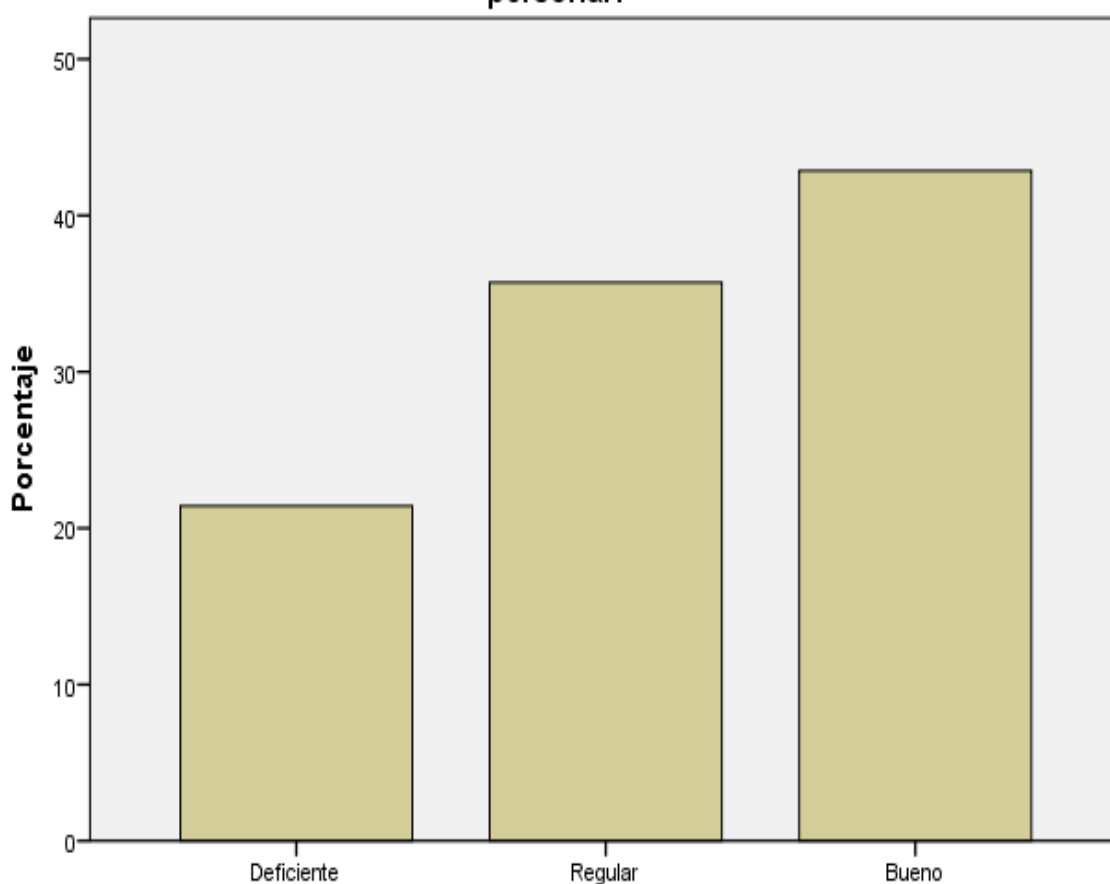


Gráfico N° 88 – Pregunta 14. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 20 y gráfico 88, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, el 42,9% indicaron que el nivel de automatización el sistema Cliente es “bueno”, mientras que los siguientes valores tiene 35,7% de “regular” y 21,4% de “deficiente”.

PREGUNTA 15:

¿Cuál es el nivel de importancia de la información de la empresa?

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	0	0	0	0
Válido Regular	0	0	0	0
Bueno	14	100,0	100,0	100,0
Total	14	100,0	100,0	

Tabla N° 21 – Pregunta 15. Elaboración propia



Gráfico N° 89 – Pregunta 15. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 21 y gráfico 89, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, el 100% indicaron que el nivel de importancia de la información sistema Cliente es **“bueno”**.

PREGUNTA 16:

¿Cuál es el nivel de satisfacción del personal con el sistema Cliente actualmente?

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	2	14,3	14,3	14,3
Válido Regular	8	57,1	57,1	71,4
Bueno	4	28,6	28,6	100,0
Total	14	100,0	100,0	

Tabla N° 22 – Pregunta 16. Elaboración propia

¿Cuál es el nivel de satisfacción del personal con el sistema Cliente actualmente?

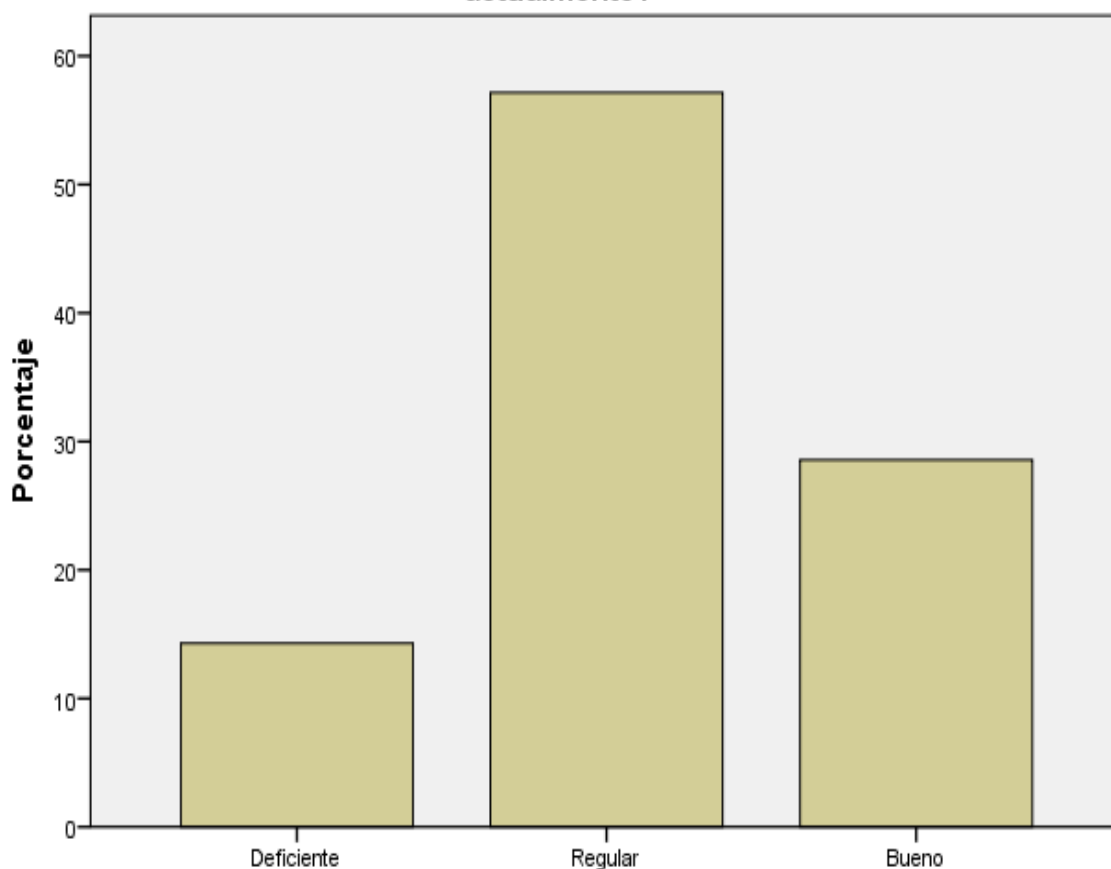


Gráfico N° 90 – Pregunta 16. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 22 y gráfico 90 del 100% de los empleados de la empresa Software y Sistemas del Perú sac, el 57,1% indicaron que el nivel de satisfacción con el sistema Cliente es “regular”, mientras que los siguientes valores tiene 14,3% de “deficiente” y 28,6% de “bueno”.

PROMEDIO Y RESUMENES DE DIMENSIONES

Análisis de Dimensión: Mitigar Riesgo de Error

Mitigar el riesgo de error

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	11	19,6	19,6	19,6
Válido Regular	32	57,1	57,1	76,8
Bueno	13	23,2	23,2	100,0
Total	56	100,0	100,0	

Tabla N° 23 – Promedio de dimensión mitigar riesgo de error. Elaboración propia

Mitigar el riesgo de error

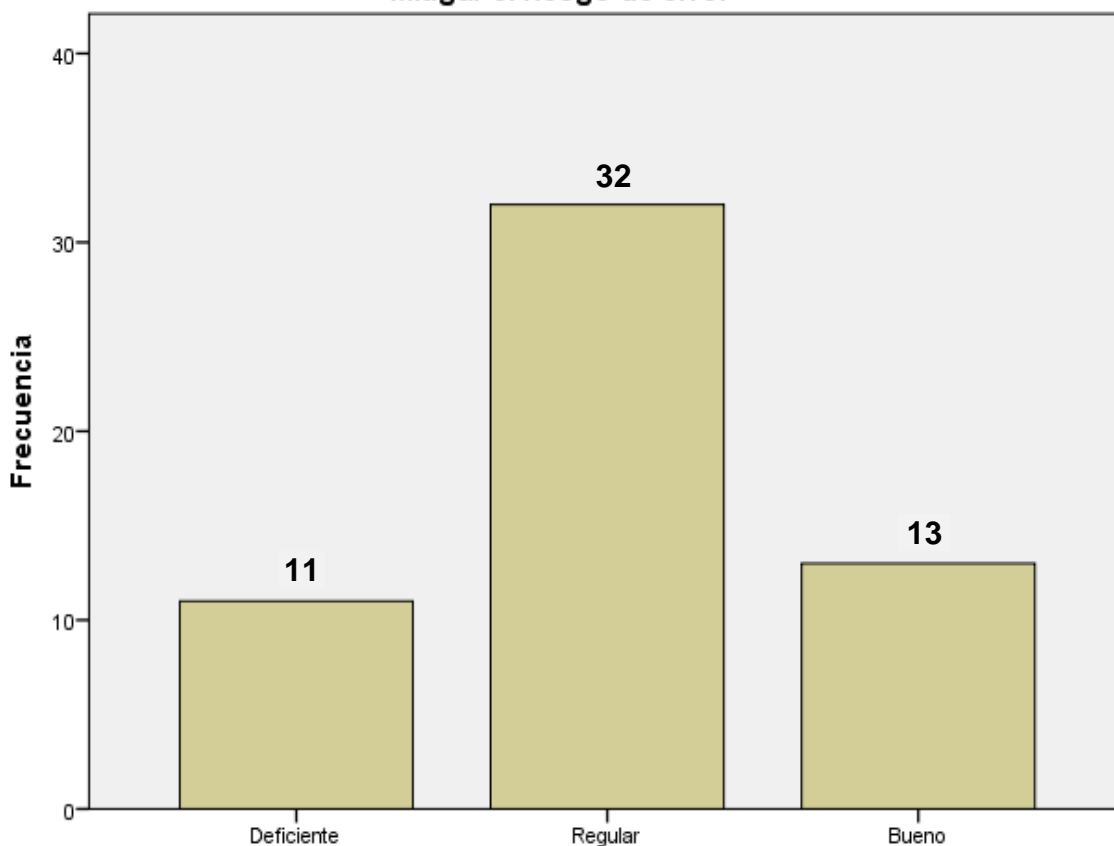


Gráfico N° 91 – Promedio de dimensión mitigar riesgo de error. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 23 y gráfico 91, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, que participaron en las preguntas 1, 2, 3, 4 (por cada pregunta hay 14 participantes, es decir, 56 respuestas en total) de la dimensión mitigar el riesgo de error, indicaron que el promedio de la escala ordinal es, 19.6% de “**deficiente**”, 57,1% de “**regular**” y 23,2% de “**bueno**”.

Análisis de Dimensión: Administración de la Información

Administración de la información

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	14	25,0	25,0	25,0
Válido Regular	27	48,2	48,2	73,2
Bueno	15	26,8	26,8	100,0
Total	56	100,0	100,0	

Tabla N° 24 – Promedio de dimensión administración de la información. Elaboración propia

Administración de la información

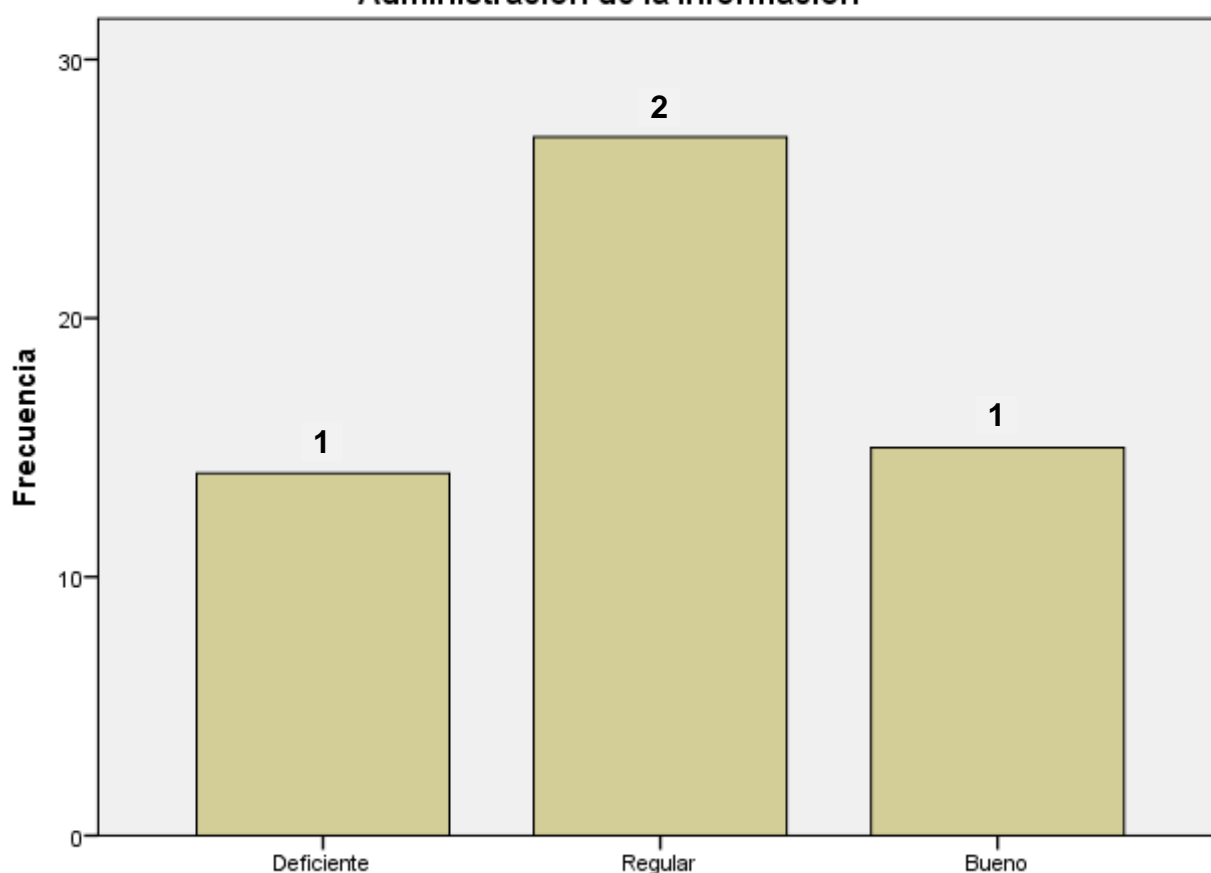


Gráfico N° 92 – Promedio de dimensión administración de la información. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 24 y gráfico 92, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, que participaron en las preguntas 5, 6, 7, 8 (por cada pregunta hay 14 participantes, es decir, 56 respuestas en total) de la dimensión administración de la información, indicaron que el promedio de la escala ordinal es, 25% de “deficiente”, 48,2% de “regular” y 26,8% de “bueno”.

Análisis de Dimensión: Calidad de Servicio

Calidad de servicio				
Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	6	10,7	10,7	10,7
Válido Regular	32	57,1	57,1	67,9
Bueno	18	32,1	32,1	100,0
Total	56	100,0	100,0	

Tabla N° 25 – Promedio de dimensión calidad de servicio. Elaboración propia

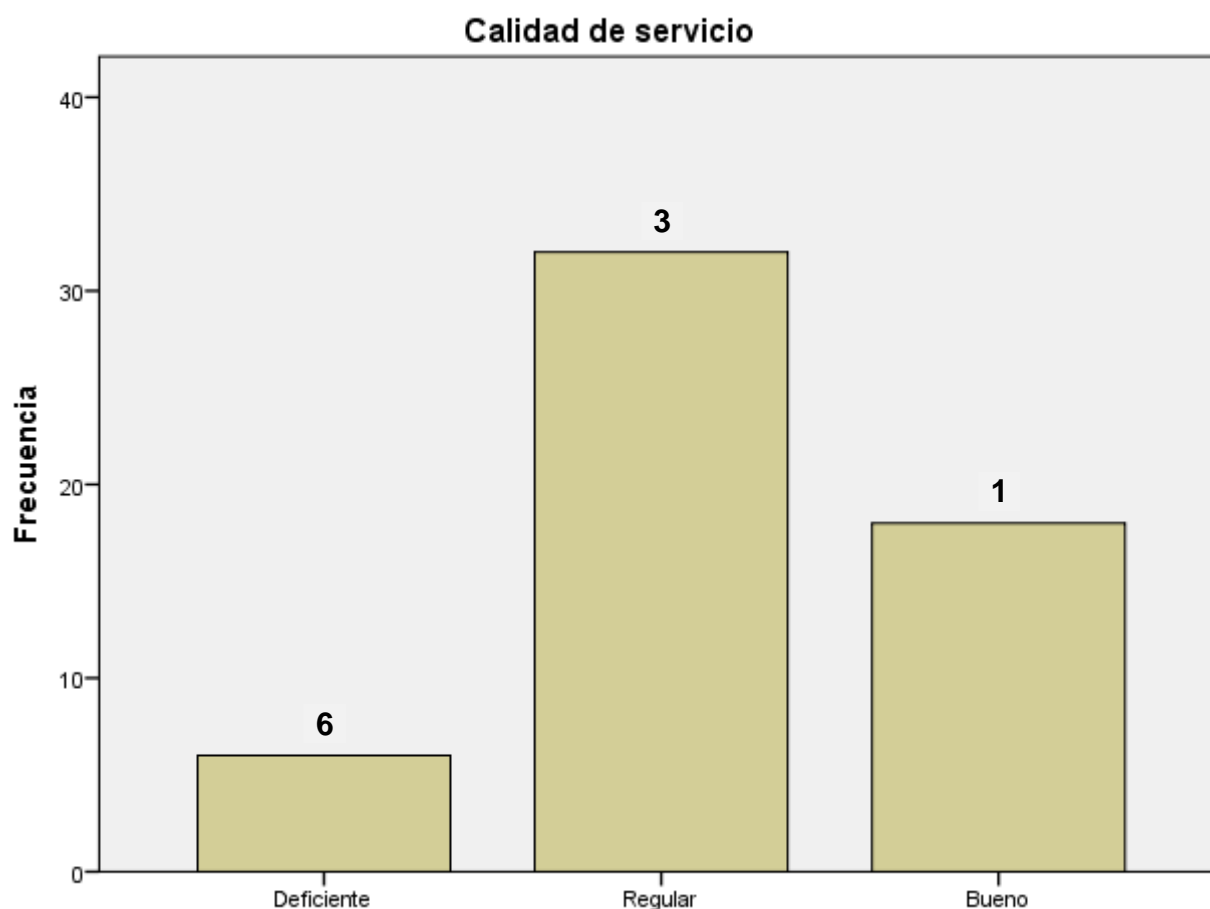


Gráfico N° 93 – Promedio de dimensión calidad de servicio. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 25 y gráfico 93, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, que participaron en las preguntas 9, 10, 11, 12 (por cada pregunta hay 14 participantes, es decir, 56 respuestas en total) de la dimensión calidad de servicio, indicaron que el promedio de la escala ordinal es, 10,7% de “**deficiente**”, 57,1% de “**regular**” y 32,1% de “**bueno**”.

Análisis de Dimensión: Satisfacción de Personal

Satisfacción de Personal

Escala de Medición	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Deficiente	6	10,7	10,7	10,7
Válido Regular	25	44,6	44,6	55,4
Bueno	25	44,6	44,6	100,0
Total	56	100,0	100,0	

Tabla N° 26 – Promedio de dimensión satisfacción de personal. Elaboración propia

Satisfacción de personal

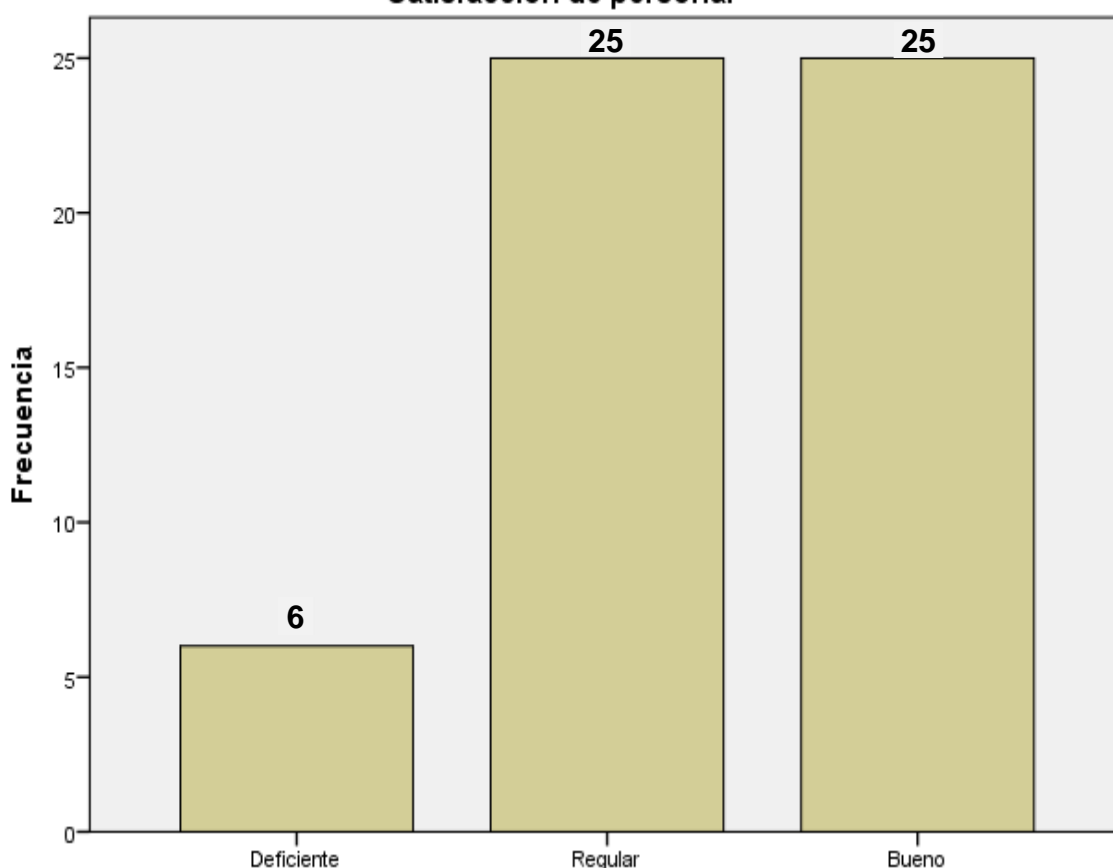


Gráfico N° 94 – Promedio de dimensión satisfacción de personal. Elaboración propia.

Contextualizando:

Como se puede visualizar en la tabla 26 y gráfico 94, del 100% de los empleados de la empresa Software y Sistemas del Perú sac, que participaron en las preguntas 13, 14, 15 y 16 (por cada pregunta hay 14 participantes, es decir, 56 respuestas en total) de la dimensión satisfacción del personal, indicaron que el promedio de la escala ordinal es, 10,7% de “**deficiente**”, 44,6% de “**regular**” y 44,6% de “**bueno**”.

CAPÍTULO V. DISCUSIÓN

5.1. Discusión

En este trabajo de investigación de una propuesta de actualización de la plataforma del sistema Cliente para integrar procesos en la empresa Software y Sistemas del Perú sac, busca fundamentar su propósito para mejorar los procesos internos verificando lo que se puede mejorar con la situación actual que tiene el sistema Cliente, pero para llegar a esa conclusión se realizó un análisis estadístico que constata las posibles mejoras que puede tener el sistema Cliente con la propuesta de la presente investigación.

La tesis de “Propuesta de Diseño de un Sistema de Información para el Seguimiento de Ventas para una Industria de Manufactura” que fue realizado por el sr. Héctor La Rosa (2004), puntualiza que desea dar una propuesta para optimizar los procesos existentes, integrando desarrollo de aspectos técnicos para la implementación del sistema, pero de una manera que tenga referencia contextual para los interesados.

Mientras en la otra tesis de aspecto internaciones, llamado “Estrategia de Actualización de Tecnología de la Información (MATI) Énfasis en Gestión de Servicios y Productos TIC” realizada por el sr. Obed Salinas (2015), indica que la adopción de nueva tecnología implica que hacer un cambio de cultura organizacional, además que se tiene que garantizar una buena identificación de los objetivos del proyecto.

Pero en el ámbito nacional, también hay investigaciones consolidadas, el cual es la “Propuesta de Implementación de Plataforma de Negocios de TI Online para Optimizar Tiempo y Costo de Cotizaciones” del señor investigador Raphael Gutiérrez (2018) , que señala que su propuesta, es para dar pautas de una implementación de una tecnología de información, pero teniendo como base un enfoque cuantitativo para el desarrollo de sus indicadores y estadística descriptiva para mejorar la situación actual de la empresa.

Asimismo, en el trabajo de investigación “Propuesta de Diseño Web y Móvil para la Gestión de Incidentes y Accidentes para una Empresa del Sector Minero” realizado por los investigadores Luis Fermin y Cesar Ingaruca (2020), muestra

que su análisis se tiene que alinear con las políticas de la empresa minera para que así respete o acople los objetivos de trabajo de investigación con los objetivos de políticas de seguridad, los cuales serán mostrados a base de diagrama de flujo y casos de uso, entre otros más, para fundamentar su proyección de implementación.

En un entorno local, fue un poco más complicado para encontrar un trabajo de investigación que tenga las referencias a este trabajo de investigación, pero el trabajo llamado “Propuesta de la Aplicación Móvil Basada en Ubicaciones de Zonas ante Desastre Naturales en Iquitos” elaborado por los señores Paulo Ríos y Terry Vela se acopla con los objetivos de este trabajo de investigación porque integra su operacionalización de su variable además que usa las mismas herramientas que son las encuesta y cuestionario para la obtención de datos.

De igual manera, en la tesis de entorno local con nombre “Propuesta de Mejora para la Gestión de Información del Portal Web de la Municipalidad Distrital de Punchana – 2019” que fue realizada por Ingrid Manihuari y Michael Santillán, tienen la misma razón de propuesta, que es la mejora de la gestión de información y al mismo tiempo poder contextualizarla para brindarlo a los usuarios.

CAPÍTULO VI. CONCLUSIONES

6.1. Conclusiones

La conclusión se fundamenta a base del análisis de evaluación pre-test, que usa la escala de medición cualitativa ordinal (deficiente, regular y bueno) para realizar este trabajo de investigación, que a su vez, permitió enfocar realmente como sería la resolución del planteamiento del problema, es decir, la Propuesta de Actualización de la Plataforma del Sistema Cliente para la Integración de Procesos de la Empresa Software y Sistemas del Perú sac – Iquitos 2021, permitiría mejorar los procesos existentes, además, implementaría la capacidad de desarrollar nuevas funciones que se integraría al procesos de acuerdo a los nuevos requerimientos que solicite la empresa.

Esta conclusión se alinea al objetivo general, ya que esta se fundamenta por el análisis de estadístico que se realizó en este trabajo de investigación, en donde las dimensiones, tuvieron un promedio de desempeño “regular”, es decir, la mayoría de los empleados de la empresa indicaron que la situación actual del sistema Cliente puede mejorar de manera significativa, para aumentar el desempeño y la eficiencia de sus procesos, para que así se pueda facilitar o automatizar la carga laboral. Pero si determinamos la conclusión por las dimensiones se puede obtener un resultado que apoyaría, lo fundamentado en el objetivo general, que a su vez los objetivos específicos serían resuelto, por motivo que los objetivos específicos están alineados a las dimensiones.

- El objetivo específico, que establece la propuesta de actualización de la plataforma del sistema Cliente que mitigue el riesgo de error en la empresa Software y Sistemas del Perú sac, se determina que el 57,1% de los empleados acepta que es una situación regular con el sistema Cliente, pero que debe ser mejorado para el control y verificación de los datos para evitar mayores errores en el futuro.
- El objetivo específico, que establece la propuesta de actualización de la plataforma del sistema Cliente para mejorar la administración de la información en la empresa Software y Sistemas del Perú sac, se determina que el 48.2% de los empleados acepta que es una situación regular con el sistema Cliente y que necesitan mejorar la administración de información.

- El objetivo específico, que establece la propuesta de actualización de la plataforma del sistema Cliente para mejorar la calidad de servicio en la empresa Software y Sistemas del Perú sac, se determina que el 57.1% de los empleados acepta que es una situación regular con el sistema Cliente, pero la expectativa de un mejor servicio sería lo expectante.
- El objetivo específico, que establece la propuesta de actualización de la plataforma del sistema Cliente para mejorar la satisfacción del personal en la empresa Software y Sistemas del Perú sac, se determina que hay una parte equitativa entre la situación regular con la buena, porque es aceptada por 44.6% de los empleados.

RECOMENDACIONES

- La gerencia general de la empresa Software y Sistemas del Perú sac, debería incorporar un Plan Estratégico de Tecnologías de la Información y Comunicación (PETIC) para incorporar nuevas tecnologías de manera eficiente y documentada para que el personal de la empresa, y así tengan una mejor estructura organizacional de las áreas internas.
- Diseñar e implementar a un futuro, las directivas de función de usuarios para los sistemas, ya que, este documento permitirá que los nuevos usuarios que ingresan a la empresa tengan adjuntado en su Manual de Organización y Funciones (MOF) las directivas del uso de los sistemas, que tiene en su haber la empresa Software y Sistemas del Perú sac.
- La gerencia general de la empresa Software y Sistemas del Perú sac, debe efectuar la ejecución de la propuesta planteada en este trabajo de investigación que contiene aspectos técnicos y contextuales, para que así los interesados estén con el conocimiento de cómo se realizaría el proyecto y la planeación de un expediente técnico.
- Se sugiere a la empresa Software y Sistemas del Perú sac., aumentar la relevancia del área de desarrollo, por motivo, que está carece de proyectos de implementación y análisis del estado situacional de los sistemas que tiene en su haber la empresa, de igual manera, el desconocimiento de los objetivos de corto y media plazo.
- Se recomienda a la empresa Software y Sistemas del Perú sac, que motive a sus empleados a la orientación de investigación y desarrollo (I+D) en su ámbito profesional y personal, para que esté, preparado para los cambios que puede efectuar la empresa o adaptarse a las nuevas herramientas que puede incorporar la empresa.

REFERENCIAS BIBLIOGRÁFICAS

- OCAMPO, Edgard y RÍOS, Gerson, “Software Melissa”, 2019, <http://softwaremelissa.com/>.
- LA ROSA, Héctor. Propuesta de Diseño de un Sistema de Información para el Seguimiento de Ventas para una Industria de Manufactura, Venezuela, Caracas, 2004.
- SALINAS, Obed. Estrategia de Actualización de la Plataforma de Desarrollo de Sistemas de Información, en el Proceso de Administración y Desarrollo de Sistemas Institucionales (PADSI), INA, Costa Rica, Heredia, 2015.
- GUTIÉRREZ, Raphael. Propuesta de Implementación de Plataforma de Negocios de TI Online para Optimizar Tiempo y Costo en Cotizaciones, Perú, Lima, 2018.
- FERMIN, Luis y INGARUCA, Cesar. Propuesta de Diseño Web y Móvil para la Gestión de Incidentes para una Empresa del Sector Minero, Perú, Lima, 2020.
- RÍOS, Paulo y VELA, Terry. Propuesta de la Aplicación Móvil Basada en Ubicaciones de Zonas ante Desastres Naturales en Iquitos. Perú, Iquitos, 2019.
- MANIHUARI, Ingrid y SANTILLAN, Michael. Propuesta de Mejora para la Gestión de Información del Portal Web de la Municipalidad Distrital de Punchana – 2019. Perú, Iquitos, 2019.
- CANTONE, Dante. La Biblia del Programador Implementación y Debugging, Creative Andina Corp., 2006.
- PRESSMAN, Roger. Ingeniería del software un enfoque práctico, Mc Graw Hill, 2010, séptima edición.
- SOMMERVILLE, Ivan. Ingeniería de Software, Pearson Educación, 2002.

- PÉREZ, Oiver. Cuatro Enfoques Metodológicos para el Desarrollo de Software RUP – MSF – XP – SCRUM, Revista Inventum No. 10, 2011, pág. 64-78.
- CANÓS, José, LETELIER, Patricio y PENADÉS, Carmen. Metodologías Ágiles en el Desarrollo de Software, España, Valencia, 2003.
- Blog Santander Universidades, “Metodología de desarrollo de software: ¿qué son?”, 2020, <https://blog.becas-santander.com/es/metodologias-desarrollo-software.html>
- Blog Santander Universidades, “Metodología de desarrollo de software: ¿qué son?”, 2020, <https://blog.becas-santander.com/es/metodologias-desarrollo-software.html>
- MegaPractical Soluciones de Negocio, “Metodología de desarrollo de software”, 2021, <https://megapractical.com/cursos-de-capacitacion-online/>
- CERVANTES, Humberto, VELASCO-ELIZONDO, Perla y CASTRO, Luis. Arquitectura de Software, Conceptos y ciclo de desarrollo. México, México D.F., 2016, pág. 4.
- BLANCARTE, Oscar. Introducción a los Patrones de Diseño. México, México D.F., 2016, pág. 18.
- MONTENEGRO, Isaac, RODRÍGUEZ, Luis y SALAZAR, Gabriela. Uso de Patrones de Diseño de Software: un caso práctico. Costa Rica, San José, 2012, pág. 46.
- ERIKSSON, U., “Why is the difference between functional and Non-functional requirements important?”, 2012, <http://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/>

- GOOGLE Sites, “Metodología Gestión de Requerimientos”, 2021, <https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales>
- RALPH, P. y WAND, Y., “A Proposal for a Formal Definition of the Design Concept”, Lecture Notes in Business Information Processing, Vol. 14, pág. 103-136, 2009.
- Lucid Software Inc. , “Qué es un diagrama entidad-relación”, 2021, <https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion>
- EKON. , “¿Qué es un diagrama de procesos y por qué es tan importante para tu empresa?”, 2020, <https://www.ekon.es/diagrama-procesos-empresa/>
- Lucid Software Inc., “Qué es un diagrama de flujo”, 2021, <https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-flujo/>
- CEVALLOS, Karla. “UML: Casos de Uso”, 2015, <https://ingsoftwarekarlacevallos.wordpress.com/2015/06/04/uml-casos-de-uso/>
- CREATELY, Karla. “Tutorial de diagramas de casos de uso”, 2021, <https://creately.com/blog/es/diagramas/tutorial-diagrama-caso-de-uso/>
- WILSON, Leslie y ROBERT, Clark. Comparative Programming Languages, Second Edition, Bredevoort, Netherlands, 1993, pág. 75 (En inglés).
- UREÑA, Carlos. El concepto de Lenguaje de Programación. España, Granada, 2010, pág. 4.
- JOYANES, Luis, RODRÍGUEZ, Luis y FERNÁNDEZ, Matilde. Fundamentos de Programación – Segunda Edición. Editorial Mc GRAW-HILL, España, Madrid, 2003, pág. 111.

- TORRES, Manuel. Programación Orientada a Objetos con Visual C# 2015 y ADO.NET 4.6. Editorial Macro, Perú, Lima, pág. 21.
- HERNÁNDEZ, Roberto. Metodología de la investigación – Cuarta edición. México, México D.F., 2006, pág. 102.

ANEXOS

ANEXO 01 – MATRIZ CONSISTENCIA

Título: PROPUESTA DE ACTUALIZACIÓN DE LA PLATAFORMA DEL SISTEMA CLIENTE PARA LA INTEGRACIÓN DE PROCESOS DE LA EMPRESA SOFTWARE Y SISTEMAS DEL PERÚ SAC – IQUITOS 2021.

PROBLEMA	OBJETIVO	HIPOTÉSIS	VARIABLES	INDICADOR	ÍNDICE	METODOLOGÍA
<p>General: ¿De qué manera la propuesta de actualización de la plataforma del sistema Cliente, incidiría la integración de procesos en la empresa Software y Sistemas del Perú sac – Iquitos 2021?</p> <p>Específicos: • ¿Cómo la propuesta de actualización de la plataforma del sistema Cliente, mitigaría el riesgo de un error en los procesos internos en la empresa Software y Sistemas del Perú sac – Iquitos 2021?</p>	<p>General: Proponer la actualización de la plataforma del sistema Cliente, incidiría en la integración de procesos en la empresa Software y Sistemas del Perú sac – Iquitos 2021.</p> <p>Específicos: • Establecer la propuesta de actualización de la plataforma del sistema Cliente que mitigue el riesgo de error en los procesos internos en la empresa Software y Sistemas del Perú sac – Iquitos 2021.</p>	<p>General: Plantear una propuesta de actualización de la plataforma del sistema Cliente, incidiría en la integración de procesos en la empresa Software y Sistemas del Perú sac – Iquitos 2021.</p> <p>Específicas: • La propuesta de actualización de la plataforma del sistema Cliente mitigará el riesgo de error en los procesos internos en la empresa Software y Sistemas del Perú sac – Iquitos 2021.</p>	<p>Independiente X: Propuesta de actualización del sistema Cliente.</p>	<p>Mitigar el riesgo de error</p> <ul style="list-style-type: none"> - Las validaciones de los datos son integradas con el sistema. - Los registros de control permiten un seguimiento posterior de los datos. - Se cumple los requerimientos para terminar los procesos. - Se tiene una óptima interacción del usuario con el sistema. <p>Administración de la información</p> <ul style="list-style-type: none"> - La estructura de las herramientas del sistema está organizada eficientemente. - La gestión de recursos del ordenador informático con la información es óptima. - La verificación de los datos ingresados al sistema es efectiva. - El uso del sistema permite que los proceso principales y secundarios sean fluidos. 	<p>Cualitativa Ordinal</p> <p>Deficiente (1)</p> <p>Regular (2)</p> <p>Bueno (3)</p>	<p>Tipo de investigación: Descriptivo.</p> <p>Diseño de la investigación: No experimental.</p> <p>Población: La población estuvo conformada por los trabajadores de la empresa Software y Sistemas del Perú SAC. – Iquitos, que en total serían de 14 integrantes.</p> <p>Muestra: La muestra estuvo conformada por el personal de las áreas de soporte y ventas que son 14 personas.</p>

<ul style="list-style-type: none"> • ¿En qué dimensión mejoraría la propuesta de actualización de la plataforma del sistema Cliente, con la administración de la información en la empresa Software y Sistemas del Perú sac – Iquitos 2021? • ¿En qué medida mejoraría el índice de calidad de servicio con la propuesta de actualización de la plataforma del sistema Cliente en la empresa Software y Sistemas del Perú sac – Iquitos 2021? • ¿En qué manera la propuesta de actualización de la plataforma del sistema Cliente mejorará la satisfacción del personal de la empresa Software y Sistemas del Perú sac – Iquitos 2021? 	<ul style="list-style-type: none"> • Establecer la propuesta de actualización de la plataforma del sistema Cliente para mejorar la administración de la información en la empresa Software y Sistemas del Perú sac – Iquitos 2021. • Establecer la propuesta de actualización de la plataforma del sistema Cliente para mejorar la calidad de servicio de la empresa Software y Sistemas del Perú sac – Iquitos 2021. • Establecer la propuesta de actualización de la plataforma del sistema Cliente para mejorar la satisfacción del personal de la empresa Software y Sistemas del Perú sac – Iquitos 2021. 	<ul style="list-style-type: none"> • La propuesta de actualización de la plataforma del sistema Cliente mejorará la administración de la información en la empresa Software y Sistemas del Perú sac – Iquitos 2021. • La propuesta de actualización de la plataforma del sistema Cliente mejorará la calidad de servicio de la empresa Software y Sistemas del Perú sac – Iquitos 2021. • La propuesta de actualización de la plataforma del sistema Cliente mejorará la satisfacción del personal de la empresa Software y Sistemas del Perú sac – Iquitos 2021. 		<p>Calidad de servicio</p> <ul style="list-style-type: none"> - La transferencia de información es eficaz entre el personal. - El tiempo de espera para enviar y recibir información es reducido. - El personal nunca está saturado de procesos o pendientes. - Cuán importante es aumentar la disponibilidad de servicio para las consultas de los clientes. <p>Satisfacción del personal</p> <ul style="list-style-type: none"> - Se distribuye de forma eficiente la comunicación entre el personal con el sistema. - El uso del sistema automatiza la carga laboral del personal. - Debe ser de suma importancia la información de la empresa. - La satisfacción del personal tiene que ser mejorado con la gestión de la empresa. 	<p>Técnicas e instrumentos:</p> <p>Se aplicó la técnica de recolección mediante encuesta para obtener la información de suma importancia y necesaria para la variable: propuesta de actualización de la plataforma del sistema Cliente.</p> <p>Procedimiento de datos y análisis estadísticos:</p> <p>Análisis estadístico descriptivo con cuadros y gráficos.</p>
---	---	--	--	--	--

ANEXO 02 – VARIABLE INDEPENDIENTE

CUESTIONARIO – Nº 01

I. PRESENTACIÓN.

En este presente cuestionario tiene como finalidad obtener información sobre el proceso de administración tecnológica de la información en la empresa Software y Sistemas del Perú sac – Iquitos 2021. Los individuos participantes solo tienen el valor para este trabajo de investigación, más la finalidad del trabajo de investigación es el título profesional de ingeniero de sistemas de información.

II. DATOS GENERALES

- a. Nombres y Apellidos :
- b. Fecha :
- c. Cargo :

III. INSTRUCCIONES

En el siguiente cuestionario se tiene que responder las afirmaciones de forma objetiva y más sincera posible, para tener una proporción de información más asomada a la realidad, recalando que esta información tiene solo uso exclusivo para este trabajo de investigación, para respetar estrictamente la confidencialidad de los participantes. Se recomienda leer cada pregunta, analizar y dar una respuesta marcando “X” en las casillas de las columnas disponibles (deficiente, regular y bueno – que son cualitativa ordinal), dependiendo a su criterio cómo ve la realidad actual de la empresa y del “**sistema Cliente**” que tiene disponible.

Caso:

1	2	3
Deficiente	Regular	Bueno

Ejemplo:

Nº	Ítems	Deficiente 1	Regular 2	Bueno 3
01	¿Cómo calificas tu profesionalidad?			X

IV. CONTENIDOS

Nº	Mitigar el riesgo de error	Deficiente 1	Regular 2	Bueno 3
01	Las validaciones de los datos son integradas con el sistema.			
02	Los registros de control permiten un seguimiento posterior de los datos.			
03	Se cumple los requerimientos para terminar los procesos en el sistema.			
04	Se tiene una óptima interacción del usuario con el sistema.			
Nº	Administración de la información	Deficiente 1	Regular 2	Bueno 3
01	La estructura de las herramientas del sistema está organizada eficientemente.			
02	La gestión de recursos del ordenador informático con la información es óptima.			
03	La verificación de los datos ingresados al sistema es efectiva.			
04	El uso del sistema permite que los proceso principales y secundarios sean fluidos.			
Nº	Calidad de servicio	Deficiente 1	Regular 2	Bueno 3
01	La transferencia de información es eficaz entre el personal.			
02	El tiempo de espera para enviar y recibir información es reducido.			
03	El personal nunca está saturado de procesos o pendientes.			
04	Cuán importante es aumentar la disponibilidad de servicio para las consulta de los clientes.			
Nº	Satisfacción del personal	Deficiente 1	Regular 2	Bueno 3
01	Se distribuye de forma eficiente la comunicación entre el personal con el sistema.			
02	El uso del sistema automatiza la carga laboral del personal.			
03	Debe ser de suma importancia la información de la empresa.			
04	La satisfacción del personal tiene que ser mejorado con la gestión de la empresa.			