

UNIVERSIDAD PRIVADA ANTENOR ORREGO

ESCUELA DE POSTGRADO



**PROPUESTA DE MÉTRICAS DE CALIDAD EN EL PROCESO DE
DESARROLLO DE SOFTWARE EN LA OFICINA DE SISTEMAS E
INGENIERÍA DE LA INFORMACIÓN DE LA UNIVERSIDAD PRIVADA
ANTENOR ORREGO**

TESIS

**PARA OPTAR EL GRADO ACADÉMICO DE MAESTRO EN
INGENIERÍA DE SISTEMAS CON MENCIÓN EN INGENIERÍA DE
SOFTWARE**

AUTOR:

Br. ARMANDO JAVIER CABALLERO ALVARADO

ASESOR:

Dr. LUIS VLADIMIR URRELO HUIMAN

Trujillo, Junio 2017

**PROPUESTA DE MÉTRICAS DE CALIDAD EN EL PROCESO DE
DESARROLLO DE SOFTWARE EN LA OFICINA DE SISTEMAS E
INGENIERÍA DE LA INFORMACIÓN DE LA UNIVERSIDAD PRIVADA
ANTENOR ORREGO**

Por: Br. Armando Javier Caballero Alvarado

Aprobado:

Ms. José Antonio Calderón Sedano

Ms. Eduardo Elmer Cerna Sánchez

Ms. Filiberto Azabache Fernández

Asesor: Dr. Luis Vladimir Urrelo Huiman

ACREDITACIÓN

El **Dr. LUIS VLADIMIR URRELO HUIMAN**, que suscribe, asesor de la Tesis con Título “**Propuesta de métricas de calidad en el proceso de desarrollo de software en la Oficina de Sistemas e Ingeniería de la Información de la Universidad Privada Antenor Orrego**”, desarrollado por el Bach. en Ingeniería de Computación y Sistemas: **Caballero Alvarado, Armando Javier**, acredita haber realizado las observaciones y recomendaciones pertinentes, encontrándose expedito para su revisión por parte de los señores miembros del Jurado Evaluador.

Trujillo, Junio del 2017.

El Asesor:

Dr. LUIS VLADIMIR URRELO HUIMAN

El Autor:

Br. CABALLERO ALVARADO ARMANDO JAVIER

PRESENTACIÓN

Señores Miembros del Jurado

Cumpliendo con los requerimientos estipulados en el reglamento de Grados y Títulos de la “Universidad Privada Antenor Orrego” para optar el grado de Maestro en Ingeniería de Sistemas con mención en Ingeniería de Software, pongo a vuestra disposición la presente tesis titulada: **PROPUESTA DE MÉTRICAS DE CALIDAD EN EL PROCESO DE DESARROLLO DE SOFTWARE EN LA OFICINA DE SISTEMAS E INGENIERÍA DE LA INFORMACIÓN DE LA UNIVERSIDAD PRIVADA ANTENOR ORREGO.**

Gracias

Trujillo, Junio del 2017

Br. Armando Javier Caballero Alvarado

DEDICATORIA

A Dios

Por mostrarme día a día que con esfuerzo, humildad, paciencia y sabiduría todo es posible.

A mi madre

Quien está en el cielo, por la formación recibida y por su esfuerzo y ejemplo constante y porque en gran parte gracias a ustedes, hoy puedo ver alcanzada mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles.

A mi hijo

Quien es el motor que me impulsa a seguir avanzando profesionalmente.

A mi hermano José Antonio, quien de manera sostenida me motivó a terminar este trabajo.

AGRADECIMIENTO

Un agradecimiento muy especial merece la comprensión, paciencia y el ánimo recibidos de mi familia y amigos.

A mi asesor el Dr. Luis Vladimir Urrelo Huiman quien con su paciencia y conocimiento ayudó a culminar en forma satisfactoria el presente informe, gracias a la orientación, el seguimiento y la supervisión continúa de la misma, pero sobre todo por la motivación y el apoyo recibido.

Mi más profundo y sincero agradecimiento a todas aquellas personas que con su ayuda han colaborado en la realización del presente trabajo, por sus apreciados y relevantes aportes, críticas, comentarios y sugerencias durante el desarrollo de esta investigación.

RESUMEN

PROPUESTA DE MÉTRICAS DE CALIDAD EN EL PROCESO DE DESARROLLO DE SOFTWARE EN LA OFICINA DE SISTEMAS E INGENIERÍA DE LA INFORMACIÓN DE LA UNIVERSIDAD PRIVADA ANTENOR ORREGO

Por el Br. Armando Javier Caballero Alvarado

Las empresas de desarrollo de software en nuestro país, son alrededor del 1%, cuya mayoría son micro y pequeñas empresas que cubren el 90% y solo el 4% son las grandes empresas, por tanto hay un mercado con gran disponibilidad para su incorporación y cubrir necesidades de todas las empresas a nivel nacional.

Asimismo con respecto a la calidad de software que se produce es aún muy bajo en los micros y pequeñas empresas debido a su informalidad que es un factor común en ellas. De ahí la importancia que se formalice las empresas de desarrollo en cuanto a sus procesos y modelos de medición para obtener producto de mayor calidad y sin defectos cuando salgan al mercado.

En tal sentido el presente trabajo desarrolla una propuesta de un Modelo de Medición que asegure el proceso de medición y análisis en una empresa de desarrollo de software, de manera que pueda alcanzar una certificación CMMI Nivel 2, para ello se compararon y adaptaron dos metodologías, tales como PSM y GQM, los cuales cubren características relevantes en los procesos de medición, estableciéndose un Modelo de Medición propuesto que tiene las siguientes fases: Definir objetivos y métricas, Elaborar un plan de medición, Ejecutar el plan de medición e Interpretar y evaluar la medición. Para el despliegue se usó herramientas como Code Metric Values y Simbia. Así se implementó el Modelo de Medición en el Área de Desarrollo de la OSIE de la Universidad Privada Antenor Orrego. Luego se evaluaron los atributos del modelo a través de opiniones de expertos para luego clasificarlos, obteniéndose, un puntaje final de 410.49 para los atributos favorables y de 164.80 para los atributos desfavorables, lo cual representa que el modelo tiene una implementación significativa en la mayoría de fases del ciclo de vida de desarrollo de software. Se analizaron los datos obtenidos de las métricas definidas para su evaluación, cuyos resultados nos dan un diagnóstico preciso del estado actual de la construcción de los sistemas que se desarrollan en el Área de Desarrollo de la OSIE.

Palabras clave: Modelos de Medición, CMMI, GQM, PSM.

ABSTRACT

PROPOSAL OF QUALITY METRICS IN THE PROCESS OF SOFTWARE DEVELOPMENT IN THE OFFICE OF SYSTEMS AND INFORMATION ENGINEERING OF THE PRIVATE UNIVERSITY ANTENOR ORREGO

By: Br. Armando Javier Caballero Alvarado

The software development companies in our country, around 1%, most of them are micro and small companies that only cover 90% of the total; and only 4% are big companies, so there is an available market for their incorporation and meet the needs of all companies in our national area.

Also, respect to the quality of software that is produced is still very low in micro and small companies due to their informality that is a common factor in them. Hence the importance of formalizing the development companies in terms of their processes and measurement models to obtain higher quality product and without defects when they come to market.

In this sense the present work develops a proposal of a Measurement Model that assures the process of measurement and analysis in a software development company, so that it can reach a certification CMMI Level 2. For that it was compared and adapted two methodologies, such as PSM and GQM, which cover relevant characteristics in the measurement processes, establishing a proposed Measurement Model that has the following phases: Define objectives and metrics, develop a measurement plan, execute the measurement plan and Interpret and evaluate the measurement. Tools such as Code Metric Values and Simbia were used for the deployment. Thus the Model of Measurement was implemented in the Development Area of the OSIE of the Private University Antenor Orrego. Then the attributes of the model were evaluated through expert opinions and then classified, obtaining a final score of 410.49 for the favorable attributes and of 164.80 for the unfavorable attributes, which represents that the model has a significant implementation in the majority of Phases of the software development lifecycle. We analyzed the data obtained from the metrics defined for its evaluation, the results of which provide an accurate diagnosis of the current state of construction of the projects developed in the Development Area of the OSIE (Office of System and Information Engineering)

Keywords: Measurements Process, CMMI, GQM, PSM.

ÍNDICE GENERAL

ÍNDICE DE TABLAS	10
ÍNDICE DE ILUSTRACIONES	11
CAPITULO I: INTRODUCCION	13
1.1. FOCALIZACIÓN DEL PROYECTO	14
1.2. EL PROBLEMA	15
1.3. ANALISIS	16
1.3.1. ENUNCIADO DEL PROBLEMA	18
1.4. ALCANCE	18
1.5. JUSTIFICACIÓN	18
1.6. ANTECEDENTES	19
1.7. VIABILIDAD	21
1.8. APOORTE	22
1.9. HIPÓTESIS	22
1.10. OBJETIVOS	23
1.10.1. General	23
1.10.2. Específicos	24
CAPITULO II: MARCO TEORICO	25
2.1. CICLO DE VIDA DEL PROYECTO	25
2.2. SCRUM	26
2.3. MEDIDAS, MÉTRICAS E INDICADORES	27
2.4. GQM	34
2.5. PSM	35
2.6. CMMI	36
2.7. GESTIÓN DE PROYECTOS EN OSIE	37
CAPITULO III: MATERIAL Y MÉTODOS	40
3.1. MATERIAL Y PROCEDIMIENTO	40
3.1.1. PROCEDIMIENTOS	40
3.1.2. METODOLOGÍA	42
CAPITULO IV: PROPUESTA DEL MODELO DE MEDICIÓN	44
4.1. MEDICIÓN Y ANÁLISIS EN CMMI v1.3	44
4.2. COMPARACION DE LAS METODOLOGIAS DE MEDICIÓN PROPUESTAS 46	46
4.3. DEFINICION DE ACTIVIDADES PARA IMPLEMENTAR EL MODELO DE MEDICIÓN	48
4.4. ALCANCE DEL MODELO DE MEDICIÓN PROPUESTO	50
4.5. ROLES Y SU ALINEACIÓN CON EL ÁREA DE DESARROLLO	50
4.6. RESPONSABILIDADES DE LOS ROLES	52

4.7.	METODOLOGÍA PROPUESTA PARA EL DESARROLLO DEL MODELO...	53
CAPITULO V: APLICACIÓN DEL MODELO DE MEDICIÓN, CASO: OFICINA DE SISTEMAS DE INFORMACIÓN Y ESTADÍSTICA.....		57
5.1.	APLICACIÓN DE LAS MÉTRICAS DE CALIDAD.....	57
5.1.1	PROCESO: DEFINIR OBJETIVOS Y MÉTRICAS.....	57
5.1.2	PROCESO: ELABORAR PLAN DE MEDICIÓN.....	67
5.1.3	PROCESO: EJECUTAR PLAN DE MEDICIÓN.....	67
5.1.4	PROCESO: INTERPRETAR Y EVALUAR LA MEDICIÓN.....	87
CAPITULO VI: RESULTADOS.....		89
6.1.	Propuesta del Modelo de Medición.....	89
6.2.	Despliegue del Modelo de Medición propuesto.....	90
6.3.	Validación del Modelo de Métricas de Calidad.....	91
CAPITULO VII: DISCUSIÓN DE RESULTADOS.....		96
7.1.	ANÁLISIS DEL PROCESO DE MEDICIÓN.....	96
7.2.	ANÁLISIS DEL MODELO DE MEDICIÓN PROPUESTO.....	97
7.3.	ANÁLISIS DEL DESPLIEGUE DEL MODELO DE MEDICIÓN.....	98
7.4.	ANÁLISIS DE LA VALIDACIÓN DEL MODELO DE MEDICIÓN PROPUESTO 98	
CONCLUSIONES.....		103
RECOMENDACIONES.....		105

ÍNDICE DE TABLAS

TABLA N° 1: CUADRO DE FUENTE CRITERIAL SUCESS FACTOR	48
TABLA N° 2: FUENTE PROPIA	50
TABLA N° 3: DESCRIPCIÓN DE LOS ROLES PROPUESTOS	51
TABLA N° 4: RESPONSABILIDADES DE LOS ROLES.....	53
TABLA N° 5: LISTA DE OBJETIVOS.....	58
TABLA N° 6: LISTA DE PREGUNTAS POR CADA OBJETIVOS	58
TABLA N° 7: PLANTILLA OBJETIVO GQM O1	59
TABLA N° 8: TABLA MÉTRICA DE ÍNDICE DE MANTENIMIENTO	61
TABLA N° 9: PLANTILLA OBJETIVO GQM O2	62
TABLA N° 10: TABLA MÉTRICA DE PORCENTAJE DE CÓDIGO REPETIDO	63
TABLA N° 11: PLANTILLA OBJETIVO GQM O3	64
TABLA N° 12: TABLA MÉTRICA DE PRUEBAS FUNCIONALES.....	66
TABLA N° 13 : TABLA DE RIESGO DE LAS MÉTRICAS DE CALIDAD.....	79
TABLA N° 14 : TABLA RESUMEN DE VALORES DE MÉTRICAS PARA EL PROYECTO DE RECURSOS HUMANOS.....	80
TABLA N° 15: TABLA RESUMEN DE VALORES DE MÉTRICAS PARA EL PROYECTO DE TRÁMITE DOCUMENTARIO	84
TABLA N° 16 : TABLA DE RESULTADOS DEL CÁLCULO DE LAS MÉTRICAS PARA LOS PROYECTOS	87
TABLA N° 17: PUNTUACIÓN DE ATRIBUTOS POR EXPERTOS, EN EL MODELO DE MEDICIÓN PROPUESTO	92
TABLA N° 18 : RESULTADO DE LA EVALUACIÓN DEL INDICADOR-ÍNDICE DE MANTENIBILIDAD	94
TABLA N° 19: RESULTADO DE LA EVALUACIÓN DEL INDICADOR-PORCENTAJE DE CÓDIGO DUPLICADO	94
TABLA N° 20 : RESULTADO DE LA EVALUACIÓN DEL INDICADOR-PRUEBAS FUNCIONALES	95
TABLA N° 21: RELEVANCIA FINAL DE CADA ATRIBUTO DEL MODELO DE MEDICIÓN PROPUESTO	99
TABLA N° 22: VALOR FINAL DE CADA INDICADOR DE LAS MÉTRICAS CON EL MODELO DE MEDICIÓN PROPUESTO PROYECTO DE RECURSOS HUMANOS	101
TABLA N° 23: VALOR FINAL DE CADA INDICADOR DE LAS MÉTRICAS CON EL MODELO DE MEDICIÓN PROPUESTO PROYECTO DE TRÁMITE DOCUMENTARIO	101

ÍNDICE DE ILUSTRACIONES

FIGURA N° 1 : ÁMBITO DE LA ESPECIALIDAD DE INGENIERÍA DE SOFTWARE	14
FIGURA N° 2: PROCESO DE RECOLECCIÓN, REGISTRO Y ANÁLISIS DE LAS MÉTRICAS.....	18
FIGURA N° 3: ESTRUCTURA GENÉRICA DEL CICLO DE VIDA DEL PROYECTO (PROJECT MANAGEMENT INSTITUTE, 2013, PÁGINA 39)	26
FIGURA N° 4: DIAGRAMA DE CICLO DE SCRUM	26
FIGURA N° 5: LAS CUATRO FASES DEL MÉTODO GQM	35
FIGURA N° 6: ACTIVIDADES DE PSM	36
FIGURA N° 7: COMPONENTES DEL MODELO CMMI	37
FIGURA N° 8: PROCESO DEL CICLO DE DESARROLLO DE SOFTWARE	39
FIGURA N° 9: PROCESO DE RECOLECCIÓN, REGISTRO Y ANÁLISIS DE LAS MÉTRICAS (FUENTE PROPIA)	54
FIGURA N° 10: PROCESO GENERAL DEL MODELO DE MEDICIÓN PROPUESTO	54
FIGURA N° 11: PROCESO DEFINIR OBJETIVOS Y MÉTRICAS	55
FIGURA N° 12: PROCESO ELABORAR PLAN DE MEDICIÓN	55
FIGURA N° 13: PROCESO EJECUTAR PLAN DE MEDICIÓN	56
FIGURA N° 14: PROCESO INTERPRETAR Y EVALUAR MÉTRICAS	56
FIGURA N° 15: ENTORNO VISUAL STUDIO 2015	68
FIGURA N° 16: OBTENER VALORES DE MÉTRICAS CON CODE METRICS VALUES	69
FIGURA N° 17: RESULTADO DE VALORES DE LAS MÉTRICAS PARA EL PROYECTO DE RECURSOS HUMANOS.....	69
FIGURA N° 18: EXPORTAR A EXCEL LOS VALORES DE LAS MÉTRICAS PARA EL PROYECTO DE RECURSOS HUMANOS	70
FIGURA N° 19: EXCEL CON VALORES EXPORTADOS DE CODE METRICS VALUES PARA EL PROYECTO DE RECURSOS HUMANOS	70
FIGURA N° 20: OBTENER VALORES DE MÉTRICAS CON CODE METRICS VALUES PARA EL PROYECTO DE TRÁMITE DOCUMENTARIO.....	71
FIGURA N° 21: RESULTADO DE VALORES DE LAS MÉTRICAS PARA EL PROYECTO DE TRÁMITE DOCUMENTARIO	71
FIGURA N° 22: EXPORTAR A EXCEL LOS VALORES DE LAS MÉTRICAS PARA EL PROYECTO DE TRÁMITE DOCUMENTARIO	72
FIGURA N° 23: EXCEL CON VALORES EXPORTADOS DE CODE METRICS VALUES PARA EL PROYECTO DE TRÁMITE DOCUMENTARIO.....	73
FIGURA N° 24: COMANDO PARA EJECUTAR SIMIAN EN EL PROYECTO DE RECURSOS HUMANOS.....	73
FIGURA N° 25: RESULTADO DE EJECUCIÓN DE SIMIAN PARA EL PROYECTO DE RECURSOS HUMANOS.....	74
FIGURA N° 26: ARCHIVO EXCEL EXPORTADO DESDE VISUAL STUDIO 2015 SIN MODIFICACIÓN – PROYECTO RECURSOS HUMANOS.....	75
FIGURA N° 27: LIMPIEZA Y TRANSFORMACIÓN DEL ARCHIVO EXCEL – PROYECTO RECURSOS HUMANOS.....	75
FIGURA N° 28: ARCHIVO EXCEL EXPORTADO DESDE VISUAL STUDIO 2015 SIN MODIFICACIÓN – PROYECTO TRÁMITE DOCUMENTARIO	76
FIGURA 29: LIMPIEZA Y TRANSFORMACIÓN DEL ARCHIVO EXCEL – PROYECTO TRÁMITE DOCUMENTARIO	76
FIGURA N° 30: ARCHIVO EXCEL MODIFICADO CON LOS CÁLCULOS NECESARIOS – PROYECTO RECURSOS HUMANOS	77

FIGURA N° 31: ARCHIVO EXCEL MODIFICADO CON LOS CÁLCULOS NECESARIOS – PROYECTO TRÁMITE DOCUMENTARIO	78
FIGURA N° 32: ARCHIVO EXCEL MODIFICADO CON LOS VALORES PARA MÉTRICA PORCENTAJE DE CÓDIGO DUPLICADO – PROYECTO TRÁMITE DOCUMENTARIO ...	78
FIGURA N° 33: PORCENTAJE DE CÓDIGO DUPLICADO PARA EL PROYECTO DE RECURSOS HUMANOS	81
FIGURA N° 34: GRÁFICO DE PORCENTAJE DE CÓDIGO DUPLICADO PARA EL PROYECTO DE RECURSOS HUMANOS.....	82
FIGURA N° 35: PORCENTAJE DE CÓDIGO DUPLICADO PARA EL PROYECTO DE TRÁMITE DOCUMENTARIO	85
FIGURA N° 36: PORCENTAJE DE PRUEBAS FUNCIONALES PARA EL PROYECTO DE TRÁMITE DOCUMENTARIO	86
FIGURA N° 37: PROCESO GENERAL DEL MODELO DE MEDICIÓN PROPUESTO	89
FIGURA N° 38: MODELO DE MEDICIÓN DETALLADO	90
FIGURA N° 39: PROCESO DEL MODELO DE MEDICIÓN PROPUESTO	91
FIGURA N° 40: PUNTUACIÓN PROMEDIO DE CUALIDADES EVALUADAS POR EXPERTOS EN EL MODELO DE MEDICIÓN PROPUESTO	92
FIGURA N° 41: RESULTADO DEL ALFA DE CROMBACH CON IBM SPSS.....	93
FIGURA N° 42: PROPORCIÓN DE LOS ATRIBUTOS FAVORABLES Y DESFAVORABLES DEL MODELO DE MEDICIÓN PROPUESTO	100

CAPITULO I: INTRODUCCION

Las organizaciones que desarrollan software ponen cada día más atención en la calidad de sus productos, procesos y recursos involucrados en su construcción. Algo que se debe tener en cuenta es la importancia de centrar la atención no solo en el producto final, sino que se considera la calidad como parte integral que debe tenerse en cuenta desde el principio y seguido en todo el ciclo de desarrollo del software.

Esta tendencia se ve en el auge de las certificaciones tales como CMMI (Capability Maturity Model Integration), ISO 15504, ISO 9003, etc. donde se destacan las actividades de medición y evaluación de la calidad de los productos de software dentro de las áreas clave de la madurez de una organización.

Las organizaciones que desean evaluar la calidad de sus productos de software deben contar con metodologías y programas de medición y evaluación (M&E) que le provean información válida para una toma de decisión más objetiva y robusta. Estas metodologías deben por un lado, establecer un conjunto de actividades y procedimientos para especificar, recolectar, almacenar y usar correctamente métricas e indicadores, y por el otro, asegurar que las medidas y los valores de los indicadores sean repetibles, consistentes y por ende comparables entre distintos productos de software.

1.1. FOCALIZACIÓN DEL PROYECTO

AMBITO DE LA ESPECIALIDAD DE INGENIERIA DE SOFTWARE

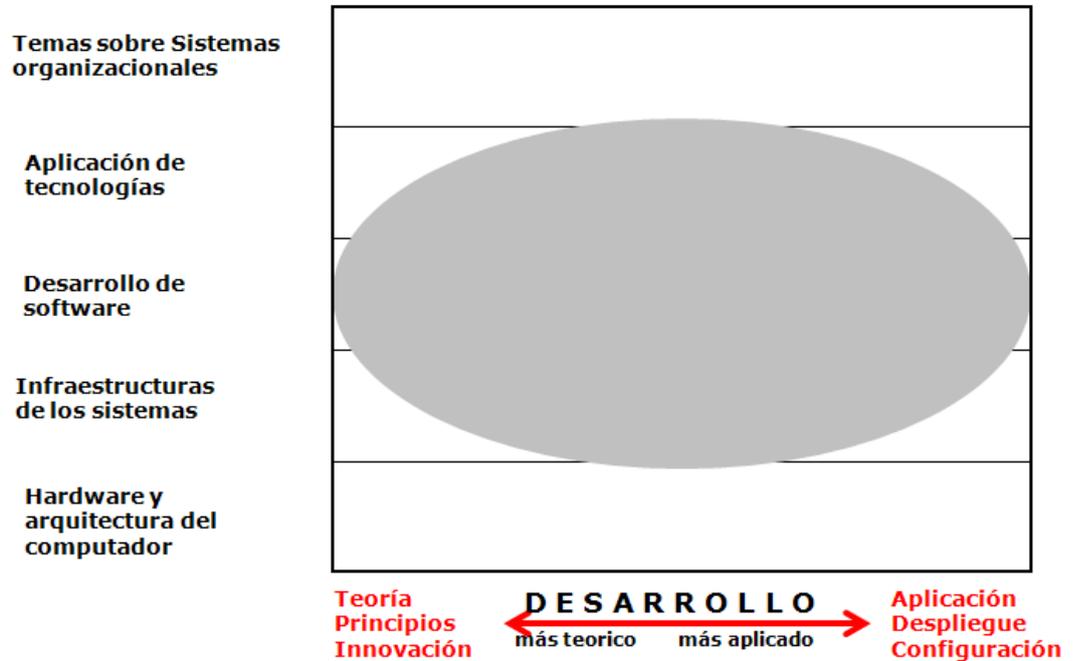


Figura N° 1 : Ámbito de la especialidad de Ingeniería de Software

Fuente: (Association for Computing Machinery, 2008)

El objetivo de SI está primariamente interesado con la información que los sistemas computarizados puedan proveer al apoyo de las organizaciones en la definición y logro de sus metas y los procesos que las organizaciones puedan implementar usando la tecnología de la información, que en nuestro caso es el desarrollo de una Propuesta de métricas para el proceso de gestión de proyectos y desarrollo de sistemas informáticos en la Oficina de Sistemas de la Universidad Privada Antenor Orrego.

1.2. EL PROBLEMA

La Oficina de Sistemas e Ingeniería de la Información y Estadística (OSIE) de la Universidad Privada Antenor Orrego, cuenta con un Área de Desarrollo de Sistemas, el cual trabaja en un entorno complejo y debe producir un producto de software, a medida, para satisfacer las necesidades cambiantes de los usuarios internos de la universidad.

A pesar de los avances significativos en la tecnología, el software es principalmente un producto hecho a mano. El mayor gasto individual en el desarrollo de software es el trabajo de los profesionales altamente calificados que diseñan, escriben y prueban el producto. Desafortunadamente los usuarios a menudo no entienden completamente sus propias necesidades hasta que hayan visto el producto. Y debido a que el proceso de desarrollo a menudo toma meses o años, el entorno tecnológico y comercial puede cambiar de manera significativa desde el principio de un proyecto hasta su fin.

Aunque la evaluación del progreso de un proyecto de desarrollo de software en cualquier punto es a menudo considerado como un arte más que una ciencia (D. Knuth. 1998), algunas organizaciones parecen producir versiones impresionantes de la tecnología de software que siguen para cambiar el mundo y mejorar la vida.

Actualmente la OSIE se encuentra en un proyecto de evaluación para el Capability Maturity Model Integration (CMMI), Nivel 2, y por lo tanto el personal del área de desarrollo se encuentra involucrado en este proceso. Asimismo se vio la necesidad de contar con información que permitirá a la Jefatura de la OSIE tener una visibilidad del comportamiento de los proyectos de desarrollo de

software con el objetivo de mejorar, así como también información que permitirá tomar decisiones y gestionar el desempeño de los desarrolladores que trabajan en dichos proyectos.

La OSIE actualmente tiene una serie de proyectos en ejecución y cuenta con ciertas métricas pero se necesita mejorarlas dentro del proceso de desarrollo de software y en consecuencia obtener productos de alta calidad. Falta mejorar las métricas que permitan realizar una gestión del desempeño de cada una de las actividades dentro del proceso de desarrollo y de cada uno de los roles que lo componen. Falta una métrica que permita conocer cuál es la volatilidad de los requerimientos en los proyectos. Falta mejorar una métrica que determine dónde están las mayores debilidades en el proceso de desarrollo a fin de poder corregirlas o apoyarlas para un mejor desempeño. Falta claridad en el uso del tiempo planificado en el proceso de desarrollo, cual es el gap con lo planificado vs ganado, y si efectivamente el tiempo es ocupado en las actividades planificadas por la coordinación de sistemas. Falta evidencia que indique que el proceso de prueba está contribuyendo a mejorar la calidad de los productos ni tampoco cual es el porcentaje de re trabajo producto de los errores encontrados. Falta mejorar la disciplina para la generación, recolección y análisis de métricas dentro del equipo de desarrollo. Es necesario un procedimiento para la recolección, análisis y almacenamiento de las métricas históricas del proceso. Falta formalizar y ejecutar las pruebas y por tanto que mejore la calidad de los productos y así lograr la satisfacción de los usuarios.

1.3. ANALISIS

En la actualidad dentro del mercado existen muchos modelos que permiten realizar mediciones del producto de software que permiten

garantizar la calidad de los proyectos informáticos de forma exitosa, en tal sentido, esto requiere que la OSIE tenga una madurez dentro del proceso de desarrollo de software.

Uno de los principales problemas de no poder aplicar éste modelo para la realización de métricas en el proceso de desarrollo de software es la falta de madurez y la falta de definición formal de los indicadores que se desean medir.

El área de desarrollo de la OSIE de la UPAO, está preparándose para la evaluación de la certificación de CMMI Nivel 2, el cual exige contar con el análisis y mediciones más formales y estrictas dentro de los proyectos informáticos.

Se cuenta con algunos indicadores que miden el avance en el desarrollo de software, pero no es tan real, ya que su medición es muy subjetiva y no está basado en un modelo formal.

Una propuesta de valor, es contar con una estrategia que ofrezca un área que de servicio de medición y análisis de software que proporcione a los usuarios datos cuantitativos y análisis de los procesos de manera que proporcione posibilidades de éxito.

Por tal motivo y considerando las necesidades de mediciones formales, nuestro problema consiste en:

Proponer métricas de calidad para el proceso de desarrollo de software en la Oficina de Sistemas e Ingeniería de la Información de la Universidad Privada Antenor Orrego.

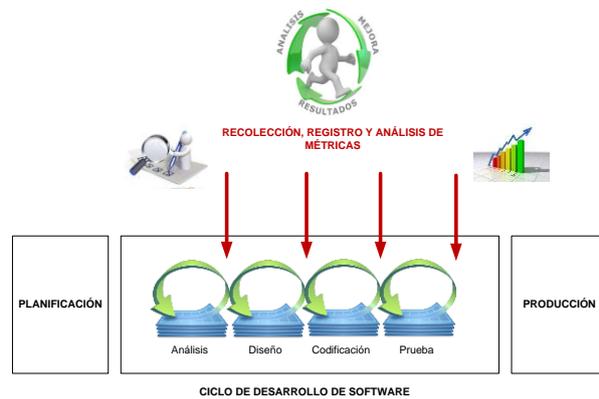


Figura N° 2: Proceso de recolección, registro y análisis de las métricas
(Fuente propia)

1.3.1. ENUNCIADO DEL PROBLEMA

¿Cómo implementar un modelo de medición para el producto de software en el Área de Desarrollo de la Oficina de Sistemas e Ingeniería de la Información de la Universidad Privada Antenor Orrego en el año 2013 alineado a métricas?

1.4. ALCANCE

La presente investigación tiene como alcance en la Fase de Construcción del proceso de desarrollo de software del Área de Desarrollo de la Oficina de Sistemas e Ingeniería de la Información de la Universidad Privada Antenor Orrego, en el periodo 2013.

1.5. JUSTIFICACIÓN

La investigación está orientada en revisar los Modelos de Medición para los productos de software y obtener un Modelo de Medición propuesto a partir de los principales que nos permita obtener un

Marco de Trabajo para la OSIE que permita garantizar la calidad de software.

1. Los proyectos de sistemas informáticos que se desarrollan dentro del área de desarrollo de la OSIE tienen un alcance a nivel de la organización interna, con una visión a una posterior comercialización dentro del mercado externo, para lo cual se necesita que los procesos de desarrollo estén definidos, así como establecer un Comité de Análisis y Medición del proceso de desarrollo dentro del ciclo de vida de desarrollo de software.
2. El no contar con métricas ni un proceso de medición al interior de la OSIE genera pérdida de competitividad al no tener una retroalimentación de la forma en la que se están ejecutando los diferentes proyectos ni del comportamiento de los procesos de desarrollo. Por lo tanto se hace imposible a su vez imponer metas que permitan mejorar la eficiencia y con ello la rentabilidad de la universidad.
3. Asimismo prepararse para la evaluación para la certificación de CMMI Nivel 2, que establece dentro de una de sus Áreas de Proceso el Análisis y Medición.

1.6. ANTECEDENTES

Las siguientes investigaciones forman parte de los antecedentes:

Autor: Natacha Lascano.

Título de la investigación: “Un Conjunto de Métricas para Proyectos de Transición de Software Offshore”, Universidad Blas Pascal, 2013.

Conclusiones: El aspecto más importante a destacar lo constituye el hecho de haber logrado alcanzar los objetivos propuestos dentro de los plazos previstos y con las restricciones existentes; dichos objetivos fueron:

- A nivel metodológico, el poder desarrollar una propuesta en base al uso de metodologías ampliamente aceptadas y que constituyan una guía para el tesista. Dicho objetivo se logró mediante el uso de Snowball al momento de realizar las entrevistas cualitativas, y Goal-Question-Metric (GQM) para definir los objetivos, por ejemplo.
- A nivel técnico, el poder utilizar una herramienta simple y de amplia utilización para desarrollar un utilitario que pudiera ser fácilmente aplicado en cualquier empresa de tecnología. Dicho objetivo se logró a través del uso de MS Excel 2013.
- A nivel profesional, el poder llevar adelante una investigación que produzca como resultado una propuesta que pudiera resultar de interés y aplicación directa en empresas reales del ámbito del trabajo del tesista. Dicho objetivo se logró a través de la definición, propuesta, y postulación del nuevo conjunto de métricas.
- A nivel académico, el poder desarrollar un trabajo de tesis que posibilite cumplimentar los requisitos previstos para la obtención del título de Magister. Dicho objetivo se logró mediante el desarrollo integral del presente trabajo.

Asimismo, la constante y sucesiva consulta y verificación de los pasos completados, de las conclusiones logradas y de los productos generados ha demostrado constituir un mecanismo poderoso y eficaz para el logro del aseguramiento de la calidad.

Autor: René Luis Vidal Lara.

Título de la investigación: “Métricas para los procesos de Gestión e Ingeniería de Proyectos de Desarrollo de Software para una PYME”, Universidad de Chile, 2013.

Conclusiones: El trabajo realizado permitió identificar los principales problemas que tenían los procesos y proyectos de la empresa. En consecuencia dichos problemas hacían que los proyectos se volvieran lentos, costoso y con una calidad subsanada a costa de constantes re trabajos. Existía prácticamente un ciclo de vida (o filosofía) de desarrollo de software de “construir y arreglar”.

Por lo tanto uno de los grandes aportes que generó este trabajo fue ayudar efectivamente a que los proyectos se apegaran al proceso, a poder controlar que las actividades planificadas efectivamente se hiciera, a ser más pulcro en la planificación y uso de los tiempos, a efectivamente planificar las pruebas y realizarlas y no sacrificarlas cuando los tiempos empezaban a escasear, y a generar credibilidad entre nuestros clientes al mostrar nuestro compromiso con la calidad y con el uso óptimo de los recursos.

Por otro lado la empresa tiene hoy un mejor entendimiento de las actividades de aseguramiento de la calidad (procesos de Verificación y Validación) y cuál es el valor que aportan al negocio, así como también se mejoró la toma de decisiones en cuanto a la asignación de los recursos de acuerdo a su desempeño y productividad.

1.7. VIABILIDAD

El presente proyecto se desarrolló analizando los diferentes Modelos de Medición y realizando la propuesta de un nuevo

Modelo de Medición para el Área de Desarrollo de la OSIE, el cual se está utilizando actualmente para lograr certificaciones futuras en otros niveles de madurez.

1.8. APORTE

La presente investigación realizará los siguientes aportes:

a. Oficina de Sistemas e Ingeniería de la Información:

1. Permitirá mejorar los procesos y la calidad de software desarrollado por la OSIE.
2. Permitirá alcanzar niveles de madurez en el desarrollo de software mayores en las futuras evaluaciones.

b. Sociedad:

1. La OSIE con certificación CMMI nivel 2, permitirá ofrecer un mejor producto de software a nivel de la organización interna y posteriormente a clientes externos dentro del mercado global, el cual contribuirá con el desarrollo social del país.

c. Tesista:

1. Permitir plasmar los conocimientos adquiridos a los largo de la maestría a fin de lograr el grado de Maestro en Ingeniería de Software
2. Obtener conocimientos más profundos sobre el modelo GQM y PSM en los proyectos de sistemas informáticos
3. Sentar las bases para trabajos futuros en la presente área.

1.9. HIPÓTESIS

H1: Una propuesta de Modelo de Medición mejorará la calidad del producto de software usando métricas para el Área de Desarrollo de la Oficina de Sistemas e Ingeniería de la Información de la Universidad Privada Antenor Orrego, año 2013.

1.9.1. VARIABLES

- INDEPENDIENTE: Modelo de Medición para el proceso de desarrollo de software.
- DEPENDIENTE: Métricas de Calidad para el producto de software.

1.9.2. INDICADORES

Variable Independiente: Modelo de Medición para el proceso de desarrollo de software.

Indicadores:

- Cumplimiento para CMMI Nivel 2
- Facilidad de aplicación
- Adaptación del Modelo

Variable Dependiente: Métricas de Calidad para producto de Software

Indicadores:

- Índice de Mantenibilidad
- Porcentaje de Código Duplicado
- Porcentaje de Pruebas Funcionales

1.10. OBJETIVOS

1.10.1. General

Proponer un Modelo de Medición para el producto de software usando métricas de calidad para el Área de Desarrollo de la Oficina de Sistemas e Ingeniería de la

Información de la Universidad Privada Antenor Orrego, año 2013.

1.10.2. Específicos

- Analizar metodologías GQM y PSM mediante revisión bibliográfica.
- Definir las métricas apropiadas en el proceso de desarrollo de software de la OSIE.
- Diseñar un proceso para la definición, recolección y análisis de las métricas definidas.
- Plantear un procedimiento que permita el almacenamiento de los datos históricos de las métricas con el fin de que estos puedan ser recuperados y estén accesibles para su uso futuro.
- Implementar y usar las métricas al menos en dos (02) proyectos representativos y de corto tiempo de ejecución, que permita validar el modelo de las métricas que se han definido.

CAPITULO II: MARCO TEORICO

En el presente capítulo, se han recogido puntos básicos a revisar en el contexto teórico en el que se desarrolla el proyecto de métricas de calidad. Lo que se busca es revisar algunos conceptos de procesos relacionados al proyecto, también acerca de metodologías de desarrollo ágil, definición de GQM, PSM, CMMI y por ultimo revisar la metodología de desarrollo dentro de la OSIE.

2.1. CICLO DE VIDA DEL PROYECTO

El ciclo de vida de un proyecto es la serie de fases por las que atraviesa un proyecto desde su inicio hasta su cierre. Las fases son generalmente secuenciales y sus nombres y números se determinan en función de las necesidades de gestión y control de la organización u organizaciones que participan en el proyecto, la naturaleza propia del proyecto y su área de aplicación. Las fases se pueden dividir por objetivos funcionales o parciales, resultados o entregables intermedios, hitos específicos dentro del alcance global del trabajo o disponibilidad financiera. Las fases son generalmente acotadas en el tiempo, con un inicio y un final o punto de control. Un ciclo de vida se puede documentar dentro de una metodología. Se puede determinar o conformar el ciclo de vida del proyecto sobre la base de los aspectos únicos de la organización, de la industria o de la tecnología empleada. Mientras que cada proyecto tiene un inicio y un final definido, los entregables específicos y las actividades que se llevan a cabo variarán ampliamente dependiendo del proyecto. El ciclo de vida proporciona el marco de referencia básico para dirigir el proyecto, independientemente del trabajo específico involucrado. (Project Management Institute, 2013, página 38)

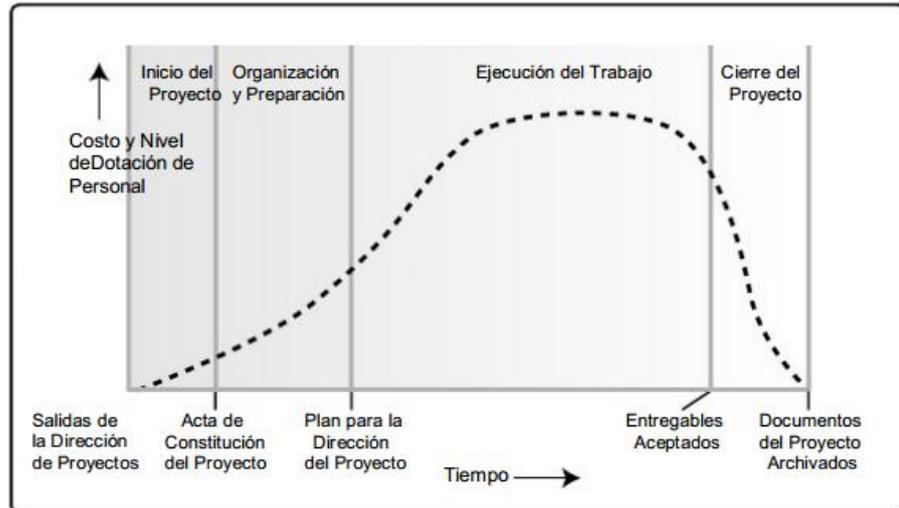


Figura N° 3: Estructura Genérica del Ciclo de Vida del Proyecto (Project Management Institute, 2013, página 39)

2.2. SCRUM

En 1995 Ken Schwaber presentó en OOPSLA 95 (Object-Oriented Programming Systems & Applications conference) (Schwaber, 1995), la implementación de Scrum Para software que él empleaba en el desarrollo de Delphi, y Jeff Sutherland en su empresa Easel Corporation (compañía que en los macrojuegos de compras y fusiones se integraría en VMARK, y luego en Informix y finalmente en Ascential Software Corporation).



Figura N ° 4: Diagrama de ciclo de SCRUM
(Fuente: Juan Palacio, Claudia Ruata. 2011)

Se basa en el principio ágil de desarrollo iterativo e incremental.

Al período de trabajo para desarrollar un incremento de producto se lo denomina "sprint", y se recomiendan duraciones entre una y cuatro semanas, si bien pueden contemplarse casos de hasta 60 días.

Establece una reunión al inicio de cada sprint para determinar el trabajo que se va a realizar, otra al final para evaluar el resultado, y revisiones diarias que realiza el equipo en su auto-gestión. (Juan Palacio, Claudia Ruata. 2011).

2.3. MEDIDAS, MÉTRICAS E INDICADORES

Aunque los términos de medida, medición y métricas se usan indistintamente, es importante tener en cuenta las diferencias sutiles entre ellos. Debido a medida puede ser utilizado como un sustantivo o un verbo, las definiciones del término puede llegar a ser confuso. Dentro del contexto de la ingeniería de software, una medida proporciona una indicación cuantitativa de la medida, cantidad, dimensión, capacidad, o el tamaño de algún atributo de un producto o proceso.

La medición es el acto de determinar una medida. El estándar IEEE Glosario de Terminología de Ingeniería de Software [IEE93b] métricas define "una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado." (Roger S. Pressman, 2010)

2.3.1. SEGUIMIENTO Y MEDICION DEL ASEGURAMIENTO DE LA CALIDAD (QA)

El propósito principal del QA es garantizar la correcta ejecución de las actividades previstos de QA mediante el uso de varias mediciones. Estas mediciones también

proporcionan la entrada de datos a las actividades posteriores de análisis y modelamiento.

1. Medidas de Calidad directa
2. Medidas de Calidad indirecta: Entorno, producto interno y actividades de medición

✓ Medidas del entorno, incluye:

i. Características de los procesos:

- Proceso usado: cascada, iterativo, espiral, etc.
- Actividades y sus relaciones
- Técnicas de desarrollos usados, etc.

ii. Características de las personas:

- Habilidades y experiencia
- Roles y responsabilidades
- Estructura del equipo y organización, etc.

iii. Características del producto:

- Expectativas generales de los usuarios finales
- Funcionalidad del producto a un alto nivel
- Configuración de hardware y software, etc.

✓ Medidas interna del producto

i. Artefactos de software, incluye especificaciones de requerimiento de software, diseño, programas, código, casos de prueba, documentos y otros.

ii. Atributos internos del producto, incluye control (ej. Control de flujo), datos (ej. Contador de operación), y presentación (ej. Diferentes reglas de sangría usada)

iii. Medición de estructuras, puede ser LOC, estructura sintáctica o entidades semánticas interconectadas.

- ✓ Medidas de actividades, mide el desarrollo de software específico y mantiene actividades y el esfuerzo asociado, tiempo y otros recursos.
 - i. Actividades de medición grano-grueso para todo el proyecto.
 - ii. Actividades de medición grano-medio para las fases de desarrollo individual, sub fases, o periodos de tiempo tales como semanas o meses.
 - iii. Actividades de medición grano-fino para actividades individuales. Por ejemplo, evaluaciones de pruebas de carga de trabajo para casos de prueba que son usados en varios modelos de confiabilidad. (Jeff Tian, 2005)

2.3.2. ESTADO DE LAS METRICAS

El área de medición del software ha sido muy activa desde hace varias décadas. Como resultado, hay muchas herramientas de métricas comerciales disponibles en el mercado. Tales indicadores económicos pueden ser el punto de partida para las organizaciones pequeñas, sin embargo, se necesita mucho más trabajo para estandarizar, validar e integrar las métricas en las prácticas de software. Este trabajo está motivado por el costo de defectos, características mensurables de software y sistemas de medición comunes para las organizaciones pequeñas.

2.3.2.1. Costos de Defectos

William T. Ward de HP (W.T. Ward, 1991), describe una iniciativa para mejorar la calidad de software. El autor utiliza una base de datos de métricas de software y un modelo de ganancias y pérdidas de la industria para desarrollar un método para calcular el costo real de los defectos de software. El esfuerzo estimado para descubrir y solucionar un defecto es de 20 horas. Este esfuerzo se calcula utilizando los puntos de datos múltiples de proyectos que fueron rastreados con la base de datos de calidad del software. La siguiente fórmula muestra el costo por defecto que se descubre y se soluciona durante la integración a través de las fases de liberación de un proyecto.

Costo de desarrollo de Software = SDRC + PL

Dónde:

SDRC: Costo de re trabajo por defecto de software, se determina por la cantidad de esfuerzo y los gastos necesarios para encontrar y corregir los defectos durante la integración a través de las fases de liberación.

PL: Pérdida-Ganancias, es la pérdida de ingresos causada por la baja venta de productos después de la liberación del ciclo de vida.

Por ejemplo, si un producto tiene 110 defectos de software encontrados y corregidos durante las

pruebas. Cada defecto requiere 20 horas de ingeniería para identificar y corregir. El esfuerzo total de trabajo es de 2,200 horas y si el costo por hora es de \$75, entonces el SDRC es de \$165,000, y el costo de re trabajo es de \$1,500. Estos gastos se podrían evitar si se hubiera utilizado las métricas para mitigar esos defectos.

2.3.2.2. Características medibles

Tom DeMarco (T. DeMarco, 1982) declaró “No se puede controlar lo que no se puede medir”, esta disciplina se aplican a las medidas para tener un mejor control de los proyectos y la calidad de los productos. Aunque la ingeniería de software es una nueva disciplina y en evolución, los expertos han propuesto métodos cuantitativos aplicables a todos los aspectos de los proyectos de software con el objetivo de lograr productos de alta calidad. Las actividades relacionadas con la medición son las siguientes:

- Estimación del costo y esfuerzo: modelos de estimación (Department of Computer Science, 2010) ayuda a mejorar la planificación y ejecución de proyectos de software. Se puede utilizar modelos matemáticos como COCOMO de Boehm (Center for Systems and Software Engineering, 2011), SLIM de Putnam (Quantitative Software Management, Inc., 2011) y puntos de función de Albrecht (R. Lytz, 2010).

- Medidas de productividad: se centran en el lado humano de un proyecto. Un factor clave para determinar con precisión la productividad está en reunir información suficiente sobre la productividad de cada persona (o equipo) en diferentes escenarios, tales como tipo de proyecto, estructura del equipo, habilidades, herramientas y medio ambiente (Roger S. Pressman, 2010).
- Recolección de datos: es una disciplina importante, que requiere diligencia y aplicación cuidadosa. El riesgo es que los datos sean inexactos que pueden dar lugar a mediciones que proporcionan evaluaciones falsas.
- Evaluación de la calidad: esta actividad abarca diversas medidas como la eficacia, fiabilidad, flexibilidad, portabilidad, usabilidad y corrección. Es necesario establecer normas que definen la calidad en términos de objetivos de proyectos específicos (Roger S. Pressman, 2010).
- Modelos de fiabilidad: a pesar que esta métrica es visto como un atributo de calidad, los modelos de evaluación de la fiabilidad están más relacionados con los fallos de software y se aplican en su mayoría durante las pruebas.

2.3.2.3. Métricas comunes

Muchos productos de métricas de software están disponibles en el mercado y se han desarrollado a lo largo de los años. A medida que varían en complejidad y sofisticación, muchos productos son

“accesibles” para organizaciones con recursos limitados.

- Métricas de procesos: estas métricas se centran en el desarrollo y mantenimiento de software. Se utilizan para evaluar la productividad de las personas, la productividad de toda la organización y la mejora de los procesos de software. Un buen ejemplo es la eficacia de un equipo en la eliminación de defectos a través del desarrollo, la detección de defectos a través de las pruebas y mejorar el tiempo de respuesta para las correcciones.

- Métricas de proyectos: son tácticas y en relación con las características y la ejecución del proyecto. Los indicadores derivados de las métricas del proyecto son utilizados por los gerentes de proyectos y desarrolladores de software para ajustar el flujo de trabajo de los proyectos y actividades técnicas. Las métricas de proyectos de software comunes son:
 - Líneas de código
 - Complejidad ciclomática
 - Puntos de función
 - Cobertura de código (Cenqua Pty Ltd., 2007)

- Métricas de producto: estos indicadores se centran en la medición de características clave del producto de software. Existen muchas métricas de

productos aplicables al análisis, codificación y pruebas. Las métricas comunes son:

- Métricas de especificación
 - Métricas de tamaño
 - Métricas arquitectural
 - Métricas de longitud
 - Métricas de complejidad
-
- Métricas de pruebas: estos indicadores miden la efectividad de las pruebas y casos de prueba.

2.4. GQM

El método GQM (Victor R. Basili, D.M. Weiss. 1984). Las siguientes subsecciones introducen las fases básicas y conceptos del método GQM.

El método GQM contiene cuatro fases:

1. La fase de Planeación, durante el cual un proyecto es seleccionado, definido, caracterizado, y planeado para aplicar la medición, resultando un plan en el proyecto.
2. La fase de Definición, durante el cual el programa de medición es definido (se definen objetivo, preguntas, métricas e hipótesis) y documentado.
3. La fase de Colección, durante el cual se tiene en cuenta la colección de datos actuales, resultando en datos coleccionados.
4. La fase de Interpretación, durante el cual la colección de datos es procesado con respecto las métricas definidas dentro de los resultados de medición, que proporcionan respuestas a las preguntas definidas, después del cual se logran objetivos a ser evaluados. (Rini van Solingen, Egon Berghout. 1999)

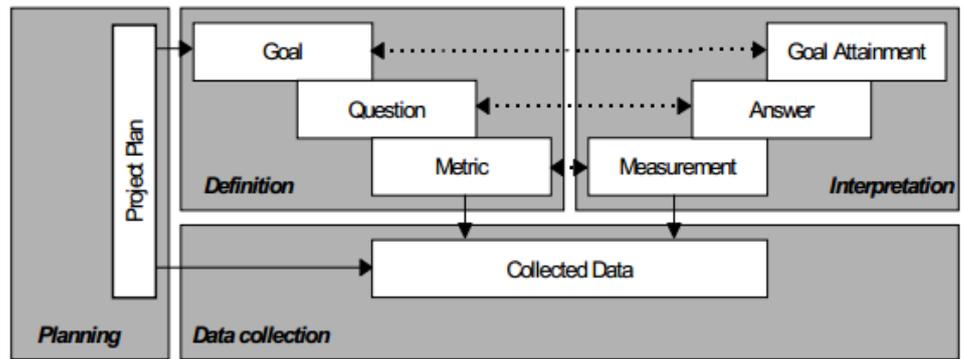


Figura N° 5: Las cuatro fases del método GQM

(Fuente: Rini van Solingen, Egon Berghout. 1999)

2.5. PSM

PSM fue seleccionado por que, como Card menciona, desde su creación PSM inicia como un esfuerzo del departamento de defensa de EE.UU (DoD, Department of Defense) para mejorar la gestión de grandes proyectos de adquisición de software (Card, D. 2003). Además, casi todos los conceptos de PSM han sido formalizados en la norma ISO 15939 (ISO/IEC 15939:2007. 2007), y el modelo del proceso de medición resultante y la terminología de esta norma también ha puede ser adoptada en el área proceso de medición y análisis del CMMI (Software Engineering Institute. 2010). Es también adecuado mencionar que acorde con al reporte de Card (Card, D. 2003), más de cinco mil personas han sido entrenadas en el proceso de PSM. Por lo tanto, estas evidencias soportan a PSM como uno de los métodos más usados en la medición del software.

Ahora bien, desde un punto de vista de los negocios, en PSM se pueden definir las necesidades de medición desde los objetivos de negocio de una forma sistemática, aunque no lo contempla formalmente como otras propuestas.

John McGarry et al. (2002) describen a PSM como sigue:

“La medición práctica del software es un proceso para diseñar e implementar un programa de medición del software basada en proyectos”.

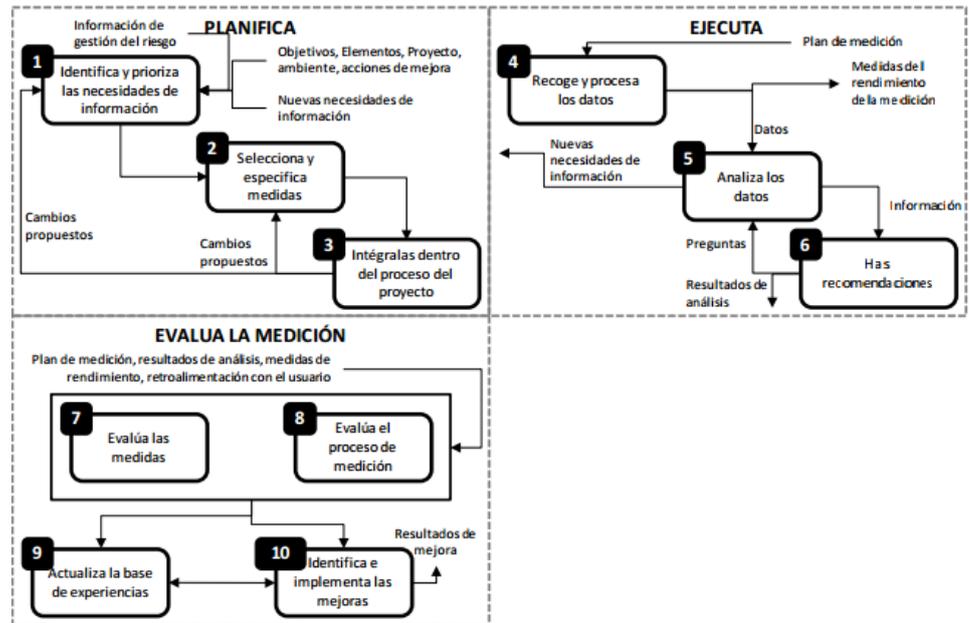


Figura N° 6: Actividades de PSM

(Fuente: Card, D. 2003)

El proceso de PSM se descompone en tres actividades principales: la planificación, ejecución, y evaluación de la medición (figura 2.4). PSM inicia con la actividad de planificar la medición, esta consiste en identificar las necesidades de información del proyecto e integrándolas dentro del proceso técnico y de gestión del proyecto.

2.6. CMMI

CMMI para Desarrollo es un modelo de referencia que cubre las actividades para desarrollar tanto productos como servicios. Las organizaciones de numerosos sectores, incluyendo aeroespacial, Banca, hardware, software, defensa, automoción y telecomunicaciones, utilizan el CMMI para Desarrollo.

CMMI para Desarrollo contiene prácticas que cubren la gestión de proyectos, la gestión de procesos, la ingeniería de sistemas, la

ingeniería de hardware, la ingeniería de software y otros procesos de soporte utilizados en el desarrollo y mantenimiento.

Use su juicio profesional y el sentido común para interpretar el modelo en su organización. Es decir, aunque las áreas de proceso descritas en este modelo representen comportamientos considerados buenas prácticas para la mayoría de los usuarios, las áreas de proceso y las prácticas se deberían interpretar usando un conocimiento profundo de CMMI-DEV, las restricciones de su organización y su entorno de negocio.

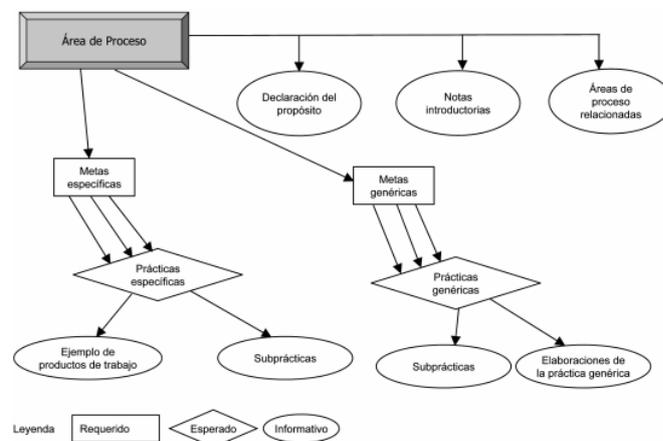


Figura N° 7: Componentes del Modelo CMMI

(Fuente: Software Engineering Institute. 2010)

2.7. GESTIÓN DE PROYECTOS EN OSIE

La Oficina de Sistemas e Ingeniería de la Información, 2012, establece que la gestión de los proyectos de software en la OSIE, se maneja a través del Área de Proyectos, el cual sigue la metodología de PMBOK.

Asimismo el Área de Desarrollo sigue de manera concordante con un modelo basado en los requisitos de CMMI, el cual considera dentro de todas las actividades que llegan como proyectos lo siguiente:

PROYECTOS

Definición: Es una secuencia de actividades únicas, complejas y relacionadas, que deben ser completadas en un tiempo específico dentro de un presupuesto. Todo proyecto debe seguir el mismo ciclo de vida comenzando con un plan de proyecto. Existen 2 tipos de proyectos

- Proyectos Grandes
- Proyectos pequeños

Características: Debe cumplir con algunos de los siguientes criterios:

Debe cumplir con algunos de los siguientes criterios:

El objetivo es desarrollar un nuevo aplicativo

- Esfuerzo estimado total mayor o igual a 400 horas
- Plazo estimado mayor o igual a 3 meses calendario
- Requiere visibilidad(Cuando es solicitado de áreas críticas rectorado)

Para definir la dimensión del proyecto debemos seguir las siguientes reglas:

- Proyectos Grandes
 - El esfuerzo total debe ser mayor a tres semana
 - Es una aplicación Critica debe existir una previa clasificación de las aplicaciones
 - Aplicación afecta a algún proceso critico
 - Asignación de recursos mayor a 50,000 soles
- Proyectos Pequeños
 - El esfuerzo total no debe ser mayor a tres semanas

- Que no afecte a áreas críticas

Ciclo de Vida de Desarrollo: Usara el Ciclo de Vida Iterativo e incremental.

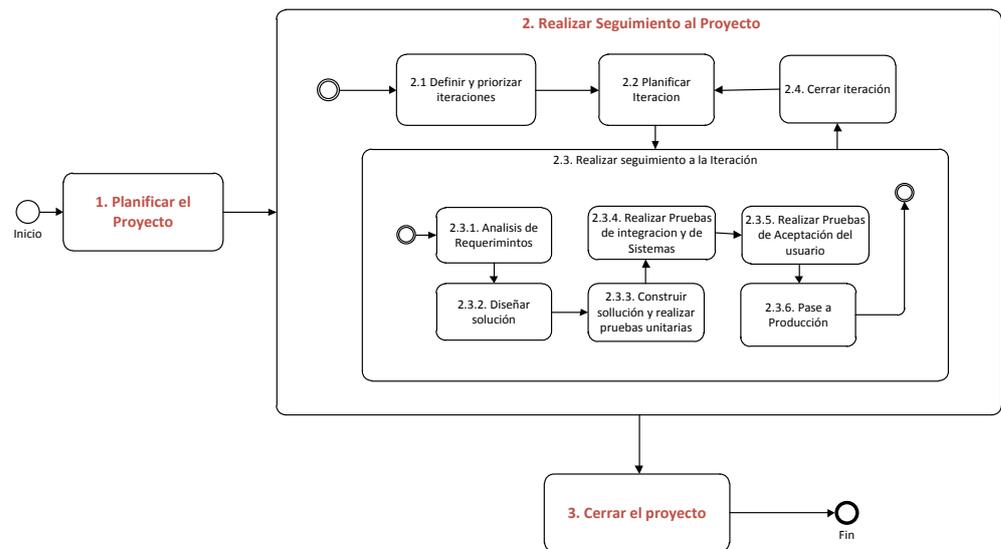


Figura N° 8: Proceso del Ciclo de Desarrollo de Software
 (Fuente: Oficina de Sistemas e Ingeniería de la Información. 2012)

CAPITULO III: MATERIAL Y MÉTODOS

En el presente capítulo, se plantean las técnicas, métodos e instrumentos utilizados para desarrollar la propuesta de métricas de calidad, definiendo una metodología a seguir.

3.1. MATERIAL Y PROCEDIMIENTO

3.1.1. PROCEDIMIENTOS

3.1.1.1. DISEÑO DE TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN

- TIPO DE INVESTIGACIÓN
 - Cuantitativa
- DISEÑO DE INVESTIGACIÓN
 - Descriptiva
- MODELO
 - Analítico
- TÉCNICA
 - Análisis de la literatura
 - Encuesta
 - Medición de los proyectos de sistemas según el marco propuesto

3.1.1.2. POBLACIÓN Y MUESTRA

- UNIVERSO

En el presente trabajo de investigación, son los proyectos desarrollados en la Oficina de

Sistemas e Ingeniería de la Información de
la Universidad Privada Antenor Orrego

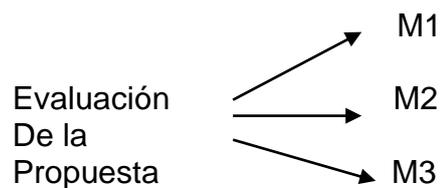
- MUESTRA

La muestra son los siguientes proyectos:

- Sistemas de RRHH
- Sistemas de Trámite Documentario

- DISEÑO DE PRUEBA

Evaluación de la Propuesta



M1, M2, M3: Métricas en el Modelo Propuesto

3.1.1.3. TÉCNICAS DE ANÁLISIS

3.1.1.3.1. LA RECOLECCIÓN DE DATOS

- Encuestas realizados a Jefe de la OSIE, funcionarios y desarrolladores
- Pruebas con herramienta de métricas

3.1.1.3.2. EL PROCESAMIENTO DE DATOS

El procesamiento de los datos se realizará a través de tablas con resultados de las encuestas en Excel

3.1.1.3.3. LOS RESULTADOS

Los resultados se presentarán en tablas y gráficos estadísticos.

3.1.1.3.4. ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

El análisis y la interpretación de los resultados se realizan a través de estadísticas descriptivas con los datos obtenidos de la aplicación del modelo propuesto en casos de estudios.

3.1.2. METODOLOGÍA

Para el desarrollo de la presente investigación se va a utilizar el procedimiento conformado por las siguientes fases:

- **ANÁLISIS DEL PROCESO DE DESARROLLO DEL ÁREA DE DESARROLLO DE LA OSIE:** Se revisará la documentación de la metodología actual que la OSIE usa para el desarrollo de software.
- **ESTUDIAR LAS METODOLOGÍAS ESTANDARES DE MEDICIÓN:** Con la ayuda del asesor se validarán las metodologías de estándares de métricas dentro del desarrollo de software para que posteriormente se realice una propuesta de modelo de medición.

- **ELABORACIÓN DE LA PROPUESTA DEL MODELO DE MEDICIÓN EN EL PROCESO DE DESARROLLO DE SISTEMAS DE LA OSIE:** Se realiza la propuesta en base a un marco metodológico que representa el nuevo modelo de medición.
- **APLICACIÓN DE LA PROPUESTA DEL MODELO DE MEDICIÓN EN EL PROCESO DE DESARROLLO DE SISTEMAS DE LA OSIE:** Posteriormente se desarrollará el modelo de medición en las fases dentro del proceso de desarrollo de software.
- **OBTENCIÓN DE RESULTADOS:** En base a la aplicación del modelo de métricas de calidad propuesta en la OSIE.
- **DISCUSIÓN DE RESULTADOS:** En base a estadística descriptiva sobre indicadores del modelo de medición propuesto. Se validará el modelo aplicándolo a los proyectos seleccionados.
- **CONCLUSIONES Y RECOMENDACIONES:** De acuerdo a los resultados obtenidos en la etapa previa se redactan los puntos concluyentes y se plantea las recomendaciones necesarias.

CAPITULO IV: PROPUESTA DEL MODELO DE MEDICIÓN

En el presente capítulo, se realiza una descripción de las metodologías PSM y GQM para la propuesta del modelo de medición, sus objetivos, alcance y su arquitectura de procesos, además se detallan las técnicas y mecanismos que hacen posible establecer las métricas.

4.1. MEDICIÓN Y ANÁLISIS EN CMMI v1.3

El área de proceso de Measurement and Analysis (MA) corresponde al nivel 2 en la representación por etapas y está ubicada dentro de la categoría de proceso de Soporte para la representación continua. Tiene como propósito desarrollar y apoyar la capacidad de medición utilizada para poder dar soporte a las necesidades de información de la gerencia.

Medición y análisis

Las prácticas en esta área de proceso se dividen en el establecimiento de un sistema de indicadores y de análisis en respuesta a las necesidades de información en la organización, y en la obtención de los valores que representan los diferentes indicadores para apoyar las decisiones que se requieren tomar. En el primer caso la técnica comúnmente utilizada es el Goal- Question-Metrics (GQM) o su variante Goal-Question-Metrics-Indicator (GQMI).

Las medidas que se definan deben ser fáciles de obtener, apoyar la toma de decisiones y no influir en el comportamiento de las personas en forma

negativa; en el sentido de que puedan provocar comportamientos no deseados o pérdidas de tiempo para obtener la información.

En la versión 1.3 del modelo no existen cambios significativos en cuanto a las metas y prácticas específicas para esta área de proceso. Como parte de la SP1.2 se anexa una tabla como ejemplo de las relaciones entre las métricas, los objetivos de medición, categorías y tipos de métricas.

Establecer el sistema de indicadores y análisis de información

SG1 Los objetivos y las actividades de medición está alineados con las necesidades de información y los objetivos identificados

- SP1.1 Establecer y mantener los objetivos de medición derivados de las necesidades de información y de los objetivos identificados.
- SP1.2 Especificar las medidas para tratar los objetivos de medición.
- SP1.3 Especificar cómo se obtienen y almacenan los datos de la medición.
- SP1.4 Especificar cómo se analizan y comunican los datos de medición.

Obtener la información para el análisis de resultados

SG2 Los resultados de la medición que tratan las necesidades de información y los objetivos identificados son proporcionados:

- SP2.1 Obtener los datos de la medición especificados.
- SP2.2 Analizar e interpretar los datos de la medición.
- SP2.3 Gestionar y almacenar datos de la medición, especificaciones de la medición y resultados del análisis.
- SP2.4 Comunicar los resultados de las actividades de medición y análisis a todas las partes interesadas.

4.2. COMPARACION DE LAS METODOLOGIAS DE MEDICIÓN PROPUESTAS

En la Tabla N° 1, se muestra una comparación de las metodologías analizadas en este trabajo, la cual está basado en el Criterio de Factores de Éxito para mejora de Procesos de Software.

De acuerdo al método de comparación se establecen solo los valores de afirmación o negación si la metodología cumple con las condiciones de factores de éxito.

La intención de esta comparación no es determinar cuál de las metodologías es la mejor, sino determinar las ventajas y desventajas de cada uno y tomar las mejores prácticas de ambas y aplicarlas en nuestra propuesta de métricas.

Preguntas sobre los factores de éxito	Metodologías comparadas	
	GQM	PSM
Administración de la mejora		
1. ¿Este método asegura una activa participación de todas las partes afectadas?	SI	SI
2. ¿Este método asegura la cooperación con la ingeniería del software?	SI	SI
3. ¿Este método asegura una capacitación planificada como parte de esta iniciativa?	NO	SI
Compromiso		

1. ¿Este método asegura el cumplimiento del compromiso de los niveles jerárquicos superiores?	NO	SI
2. ¿Este método asegura el cumplimiento del compromiso de los niveles medios?	SI	SI
3. ¿Este método asegura el cumplimiento del compromiso de los ingenieros de software?	SI	SI
Aspectos culturales		
1. ¿Este método asegura que las soluciones de mejoras serán desarrolladas teniendo en cuenta las características organizacionales y culturales de cada proyecto?	SI	SI
Plan		
1. ¿Este método asegura que el estado actual de los procesos esté claro?	SI	SI
2. ¿Este método asegura la relación entre los objetivos del negocio y los objetivos de mejoras establecidos?	SI	NO
3. ¿Este método asegura que los objetivos de medición están basados sobre necesidades?	SI	SI
4. ¿Este método asegura que		

será generado un plan de medición detallado?	SI	SI
Hacer		
1. ¿Este método asegura que el desarrollo de soluciones será probada antes de ser utilizada a gran escala?	SI	SI
2. ¿Este método asegura que las prácticas de soporte estarán siempre disponibles para los proyectos de desarrollo?	SI	SI
Revisar		
1. ¿Este método asegura que los resultados y las acciones serán revisados regularmente usando métricas?	SI	SI
Actuar		
1. ¿Este método asegura la sustentabilidad de la iniciativa de medición?	NO	SI

Tabla N° 1: Cuadro de Fuente Criterial Sucess Factor

4.3. DEFINICION DE ACTIVIDADES PARA IMPLEMENTAR EL MODELO DE MEDICIÓN

Proponemos 04 (cuatro) actividades para el modelo de medición, las cuales están fundamentadas en el análisis y la comparación de las metodologías de medición de proyectos de desarrollo de software y, en la selección de métricas e indicadores propuestos.

Actividades	Metodología usada
1. Identificar los objetivos del proyecto y alinearlos a los objetivos del negocio. Identificación y selección de métricas: este paso es iterativo en todo el ciclo de vida del proyecto.	GQM
2. Definir el Plan de Medición: elaborar un documento, que contenga como mínimo los siguientes puntos: <ul style="list-style-type: none"> • Establecer los roles de los participantes del proceso de medición. • Formas de recolección de la información • Presentación de la información • Evaluación de los resultados 	PSM
3. Ejecutar el Plan de Medición <ul style="list-style-type: none"> • Recoger y procesar datos • Analizar los datos 	PSM
4. Evaluación de los resultados:	

toma de decisiones en base a los resultados arrojados por los procesos de medición, retroalimentación y ajustes de las métricas.	PSM
--	-----

Tabla N° 2: Fuente propia

4.4. ALCANCE DEL MODELO DE MEDICIÓN PROPUESTO

Está relacionada con las actividades necesarias para asegurar el objetivo de la propuesta.

Cabe aclarar que las mediciones se harán exclusivamente en la Fase de Construcción del Ciclo de Desarrollo de Software.

La implementación de la propuesta ofrece las siguientes funcionalidades:

- Alinear de los objetivos estratégicos y los objetivos de medición. Esta actividad permitirá definir de una forma más precisa el conjunto de indicadores por cada objetivo de mejora.
- El diseño de un modelo operativo por cada objetivo de medición en el que se establecen las acciones para asegurar dicho objetivo.
- La definición de un plan de medición y sus indicadores correctamente alineados a cada objetivo de mejora. El plan y los indicadores permitirán formalizar la recogida, almacenamiento, reporte y soporte de la información medible en la OSIE.
- Realizar un plan de pruebas de los indicadores de medición en un ambiente de pruebas, con el propósito de validar el procedimiento y hacer los ajustes correspondientes al proceso propuesto.
- Realizar un plan de implementación formal dentro del Área de Desarrollo de la OSIE. Esto implica formalizar dentro de las prácticas y procedimientos de desarrollo de software.
- Realizar evaluación cada periodo de tiempo e interpretar los indicadores recogidos dentro del proceso de medición.

4.5. ROLES Y SU ALINEACIÓN CON EL ÁREA DE DESARROLLO

Equipo	Rol	Perfil	Alineación de
--------	-----	--------	---------------

			los roles con el personal interno de la OSIE
Equipo de implementación	Ejecutor	Es una persona identificada dentro de la OSIE que tiene la responsabilidad primaria para aplicar las métricas y está motivada para hacerlo.	Jefe de la OSIE
	Líder	El proceso completo es facilitado por el líder del proceso. Actúa como consultor en la ayuda de la OSIE al aplicar la propuesta correctamente.	Tesista
Equipo de Medición	Propietario del proceso de medición	Es el responsable del proceso de medición	Coordinador de Área de Desarrollo
	Analista de medición	Es el encargado de la planificación, rendimiento, evaluación y mejora de la medición. Es el responsable de la gestión de recogida y almacenamiento de los datos de medición	Tesista

Tabla N° 3: Descripción de los roles propuestos

4.6. RESPONSABILIDADES DE LOS ROLES

Rol	Responsabilidad
Ejecutor	Durante todo el proceso será un observador y algunos casos liderar el proceso.
Líder	<p>Proporcionar una charla motivacional acerca del proceso de medición con el fin de obtener la implementación del proceso.</p> <p>Evaluar las capacidades del personal para asignarla a su correspondiente rol del proceso.</p> <p>Guiar en la alineación de los objetivos estratégicos con los objetivos de medición.</p>
Propietario del proceso de medición	<ul style="list-style-type: none"> • Auxiliar con la definición de indicadores por objetivo de mejora. • Identificar las medidas de cada indicador apoyándose con el modelo GQM. • También, participar en la definición del indicador, su interpretación, el usuario de medición, y el criterio de decisión. <p>Definir las mejoras del proceso de implementación propuesto y la información cuantificable presentada a los usuarios de medición.</p>

Analista de medición	<p>Participar en la definición de los indicadores.</p> <p>Colaborar con la definición de medidas utilizando el modelo GQM. Además, determinar el nombre del indicador, la interpretación frente al objetivo de mejora, el usuario de medición del indicador.</p> <p>Cabe destacar su participación en los pasos más detallados como la definición del recurso a recoger datos medibles por cada medida base definida de cada indicador, y por cada recurso definir su procedimiento de recogida.</p> <p>Para terminar, definir las formulas en base a las medidas base o derivadas necesarias para obtener el valor deseado.</p>
	<p>Por cada indicador actualizar su modelo de análisis acorde al valor actual en el criterio de decisión.</p> <p>Crear o actualizar los reportes con los estados actuales de los objetivos y sus indicadores.</p>
	<p>Gestionar las peticiones de información de los usuarios de medición.</p> <p>Enviar los indicadores conforme al periodo de reporte de cada indicador. También, enviar el reporte al Ejecutor.</p>

Tabla N° 4: Responsabilidades de los roles

4.7. METODOLOGÍA PROPUESTA PARA EL DESARROLLO DEL MODELO

Esta propuesta se aplicara en los procesos del Ciclo de Desarrollo de Software, para efecto de esta tesis se aplicara en la Fase de Codificación. Escenario de donde se aplicara la propuesta de métricas de calidad:



Figura N° 9: Proceso de recolección, registro y análisis de las métricas (Fuente Propia)

A continuación se detalla los procesos que comprende las fases propuestas para obtener las métricas de calidad:

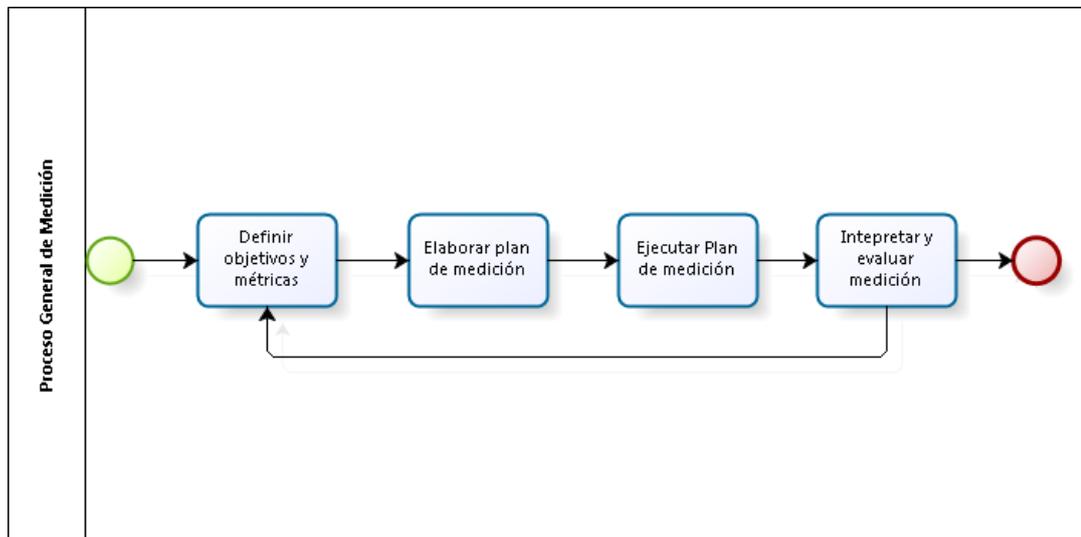


Figura N° 10: Proceso general del Modelo de Medición Propuesto

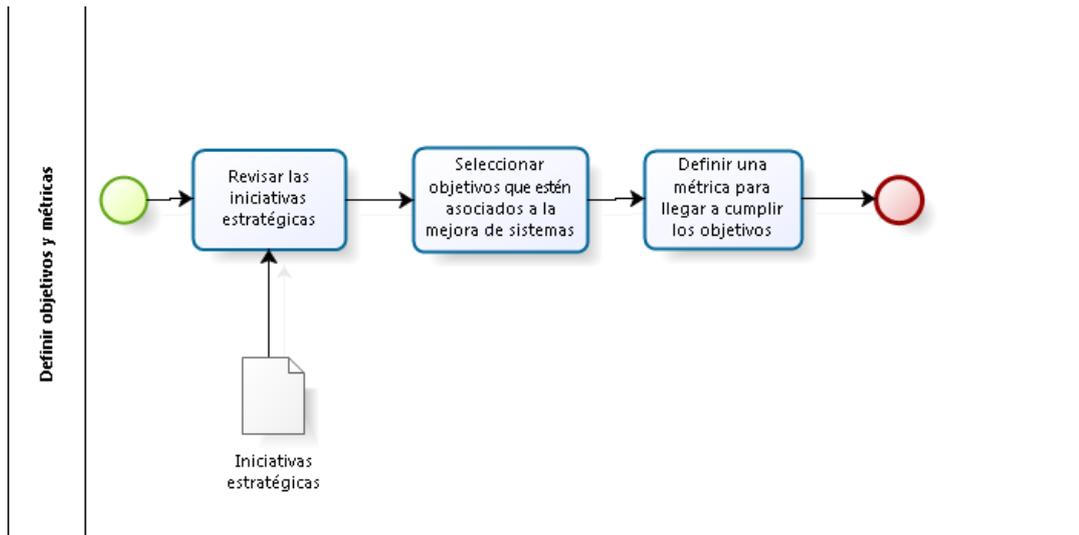


Figura N° 11: Proceso Definir Objetivos y Métricas

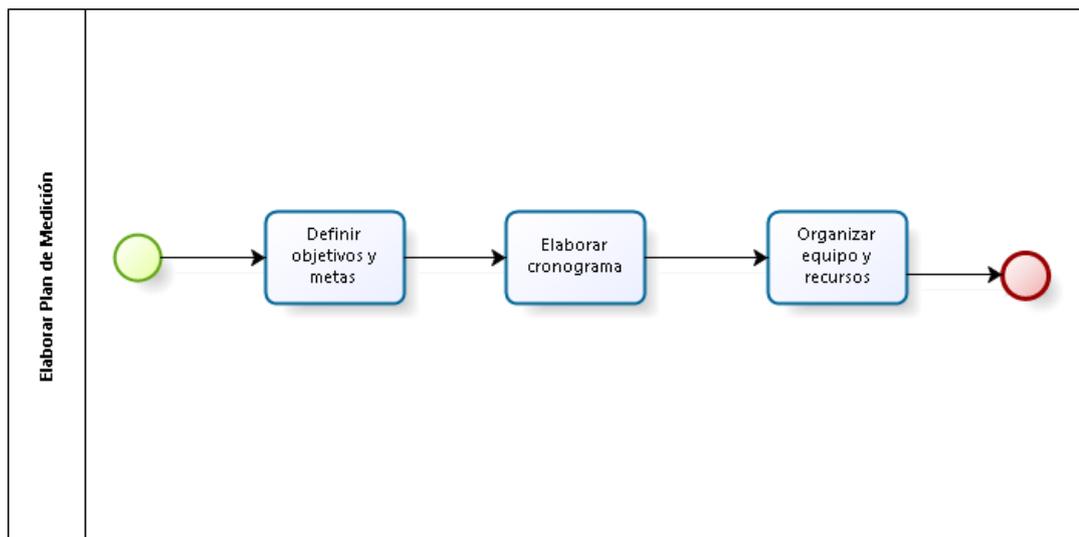


Figura N° 12: Proceso Elaborar Plan de Medición

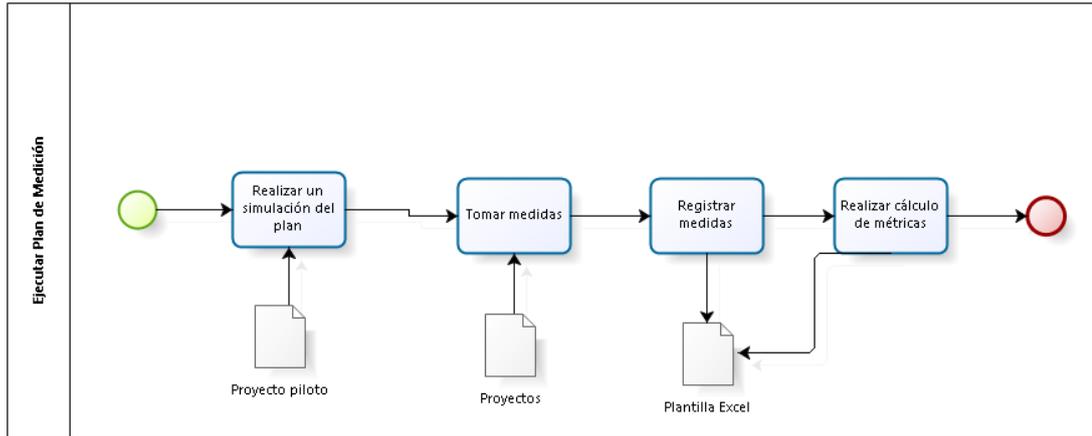


Figura N° 13: Proceso Ejecutar Plan de Medición

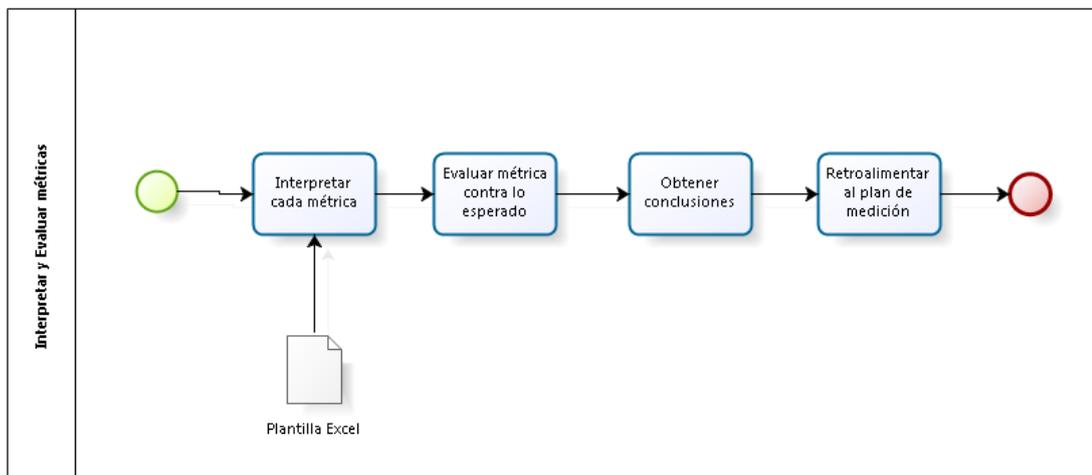


Figura N° 14: Proceso Interpretar y Evaluar Métricas

CAPITULO V: APLICACIÓN DEL MODELO DE MEDICIÓN, CASO:

OFICINA DE SISTEMAS DE INFORMACIÓN Y ESTADÍSTICA

En el presente capítulo, se presenta el desarrollo de la metodología así como la propuesta de métricas de calidad, cuya finalidad es definir medidas dentro del proceso del ciclo de vida de desarrollo de software, para esto se ha tomado como caso de estudio al Área de Desarrollo de la Oficina de Sistemas e Ingeniería de la Información de la UPAO.

Estas métricas permitirán conocer y mejorar la calidad de los productos de software a partir de determinados indicadores de medición para tomar decisiones oportunas.

La presente metodología se basa en las metodologías PSM y GQM. El ciclo de vida de este proyecto consiste, según lo descrito en el capítulo anterior, en seis fases: Establecer Objetivos, Generación de Preguntas, Especificación de Medidas, Preparar colección de datos, Recolectar, Validar y Analizar los datos y Analizar los datos para el logro de los objetivos y aprendizaje. El desarrollo de la solución se basa en la herramienta Code Metrics Values de Microsoft, el cual es una herramienta addons para el Microsoft Visual Studio 2015.

5.1. APLICACIÓN DE LAS MÉTRICAS DE CALIDAD

5.1.1 PROCESO: DEFINIR OBJETIVOS Y MÉTRICAS

5.1.3.1 FASE 1: ESTABLECER LOS OBJETIVOS

Objetivo institucional: Alinear los sistemas de información y comunicación para fortalecer los procesos académicos y administrativos.

Objetivo 1	Reducir el tiempo para resolver cualquier defecto en un 40%
Objetivo 2	Reducir el costo de mantenimiento en un 30%
Objetivo 3	Mejorar la calidad de los sistemas en un 80%

Tabla N° 5: Lista de Objetivos

5.1.3.2 FASE 2: GENERACIÓN DE PREGUNTAS

	Objetivo 1	Objetivo 2	Objetivo 3
¿Cuál es el monto que se gasta en mantenimiento por mes?		X	
¿Qué fracción de los costos de mantenimiento es invertida en casa sistema que se le da soporte?		X	
¿Cuál es el monto gastado en mantenimiento adaptativo, perfectivo y correctivo?		X	
¿Cuál es el tiempo total para resolver los defectos de todos los sistemas?	X		
¿Cuál es el tiempo total para resolver los defectos por cada sistema?	X		
¿Cuál es el costo total de resolver los defectos?	X		
¿Cuántos incidentes por mes se reciben por errores en los sistemas?			X
¿Cuál es el porcentaje de defectos que se encuentran en las pruebas?			X
¿Cuál es el tiempo de re trabajo producto de los defectos encontrados en las pruebas?			X

Tabla N° 6: Lista de preguntas por cada Objetivos

5.1.3.3 FASE 3: ESPECIFICACIÓN DE MEDIDAS

Objetivo GQM O1:

Actualmente el Área de Desarrollo de la OSIE, desarrolla proyectos de software con una metodología ágil llamada SCRUM y algunos estándares de programación definidos por el área de desarrollo. Sin embargo el costo de mantenimiento es alto y por tanto incorporar esta métrica apoyaría en mejorar la calidad de producto final. Para esto se plantea una métrica que mida el índice de mantenimiento del producto de software.

Objetivo GQM O1: Reducir el costo de mantenimiento	
Analizar	Efectividad del código fuente
Con el propósito de	Mejorar la calidad de los entregables finales
Con respecto a	La calidad del producto de software
Desde el punto de vista de	Los usuarios del proceso
En el contexto de	Desarrollo y mantenimiento de software
Preguntas asociadas a al mantenimiento	
P1	¿Cuál es el monto que se gasta en mantenimiento por mes?
P2	¿Qué fracción de los costos de mantenimiento es invertida en cada sistema que se le da soporte?
P3	¿Cuál es el monto gastado en mantenimiento adaptativo, perfectivo y correctivo?

Tabla N° 7: Plantilla Objetivo GQM O1

Logro del Objetivo GQM O1

En la mayoría de los casos cuando los usuarios finales solicitan mantenimiento a los desarrollos, se tiende a tener mucha dificultad en realizar esta actividad por la rotación continua de los programadores.

El propósito de esta métrica es seguir con estándares que permitan obtener código fuente limpia y sencilla de ser mantenido por los programadores.

Por otro lado se conoce que un adecuado uso de estándares de programación puede ayudar a la universidad ahorrar tiempo y costo en las atenciones solicitadas por los usuarios finales y darles satisfacción en sus requerimientos.

Métrica Propuesta: Índice de Mantenimiento

La Tabla N° 8 presenta la tabla con la definición de las métricas que permitirán conocer el índice de mantenimiento del producto de software antes de pasar a pruebas.

Indicador	Índice de Mantenimiento
Definición	Demostrar que el producto cumple con los requisitos de calidad del código fuente
Metas	Alcanzar el 30% de ahorro en gastos de mantenimiento de los sistemas.
Métricas	Complejidad ciclomática Líneas de código Acoplamiento de clases Profundidad de herencia

Periodicidad	Cada vez que se realice un paso al ambiente de pruebas
Procedimiento Análisis	<p>Calcular el índice de mantenimiento:</p> <p>IM= MAX(0,(171 - 5.2 * ln(V) - 0.23 * (CC) - 16.2 * ln(LOC))*100 / 171)</p> <p>Dónde:</p> <p>V : es el Volumen de Halstead CC : Complejidad Ciclomática LOC: Líneas de Código</p>
Forma de presentación	Gráfico de barra para cada proyecto
Herramientas	Visual Studio 2015, Code Metrics Values
Responsabilidades	Analista QA, Analista de Métricas.

Tabla N° 8: Tabla métrica de Índice de Mantenimiento

Objetivo GQM O2:

Mejorar la calidad del producto final implica tener un marco de trabajo y estándares de programación. Esto permitiría reducir los tiempos de atención en la resolución de cualquier defecto que haya ocasionado el incidente. Se plantea por tanto una métrica que reduzca el tiempo para resolver estos defectos de software.

Objetivo GQM O2: Reducir el tiempo para resolver cualquier defecto	
Analizar	Efectividad del código fuente
Con el propósito de	Mejorar la calidad de los entregables finales
Con respecto a	La calidad del producto de software
Desde el punto de vista de	Los usuarios del proceso
En el contexto de	Desarrollo y mantenimiento de software
Preguntas asociadas a al mantenimiento	
P1	¿Cuál es el tiempo total para resolver los defectos de todos los sistemas?
P2	¿Cuál es el tiempo total para resolver los defectos por cada sistema?
P3	¿Cuál es el costo total de resolver los defectos?

Tabla N° 9: Plantilla Objetivo GQM O2

Logro de Objetivo GQM O2

Es común ver que una mala práctica de los desarrolladores es el copiar y pegar parte de código fuente de un programa a otro, ya sea por falta de tiempo en los proyectos o por malos hábitos.

El código duplicado es uno de los peores enemigos de la mantenibilidad de un producto de software, ya que gasta recursos y tiempo innecesarios para poder hacer cambios o mejoras en el software.

En definitiva esto origina problemas posteriores en el mantenimiento de los sistemas y errores a largo plazo, ocasionando incidentes y re trabajo.

Métrica Propuesta: Porcentaje de Código Duplicado

La Tabla N° 10 presenta la tabla con la definición de las métricas que permitirán conocer el porcentaje de código duplicado del producto de software antes de pasar a pruebas.

Indicador	Porcentaje de Código Duplicado
Definición	Demostrar que el producto cumple con los requisitos de calidad del código fuente.
Metas	Alcanzar el 40% de ahorro en gastos para resolver cualquier defecto
Métricas	Porcentaje de código duplicado
Periodicidad	Cada vez que se realice un paso al ambiente de pruebas
Procedimiento Análisis	Calcular el porcentaje de código duplicado: $\%LCD = (NLR / LDC) * 100$ Donde: LCD: Líneas de Código Duplicado LDC: Líneas de Código NLR: Número de líneas duplicadas
Forma de presentación	Gráfico de torta para cada proyecto
Herramientas	Simbia Tools
Responsabilidades	Analista QA, Analista de Métricas.

Tabla N° 10: Tabla métrica de Porcentaje de Código Repetido

Objetivo GQM O3:

Mejorar la calidad del producto final implica tener un marco de trabajo y estándares de programación. Esto permitiría reducir los tiempos de atención en la resolución de cualquier defecto que haya ocasionado el incidente. Se plantea por tanto una métrica que reduzca el tiempo para resolver estos defectos de software.

Objetivo GQM O3: Reducir el tiempo para resolver cualquier defecto	
Analizar	Efectividad del código fuente
Con el propósito de	Mejorar la calidad de los entregables finales
Con respecto a	La calidad del producto de software
Desde el punto de vista de	Los usuarios del proceso
En el contexto de	Desarrollo y mantenimiento de software
Preguntas asociadas a al mantenimiento	
P1	¿Cuál es el tiempo total para resolver los defectos de todos los sistemas?
P2	¿Cuál es el tiempo total para resolver los defectos por cada sistema?
P3	¿Cuál es el costo total de resolver los defectos?

Tabla N° 11: Plantilla Objetivo GQM O3

Logro de Objetivo GQM O3

Cuando los proyectos sufren un retraso y el tiempo comienza a faltar es común ver que la tendencia es a sacrificar las pruebas y por tanto la calidad del

producto perjudicando en el largo plazo la imagen y credibilidad del área de desarrollo de la OSIE.

El propósito de esta métrica es hacer seguimiento a las pruebas y compararlo con el plan para poder tomar acciones de manera inmediata en que exista algún indicador de que las actividades de prueba no se están cumpliendo. Esto debido a que cuando existen riesgos de no cumplir los plazos del proyecto, se suelen sacrificar las pruebas a fin de recuperar estos atrasos. Luego al contar con una métrica formal del progreso de las pruebas, es más difícil que se ignore el problema.

Asimismo se sabe que un adecuado proceso de pruebas puede ayudar a la rentabilidad de la universidad ahorrando tiempo (y con ello dinero) en el desarrollo del producto y contribuyendo a mejorar la satisfacción del usuario.

Métrica Propuesta: Porcentaje de Pruebas Funcionales

La Tabla N° 12 presenta la tabla con la definición de las métricas que permitirán conocer cuál es el número de casos de prueba ejecutadas y cuál es la tasa de defectos que son encontrados por cada ejecución de las pruebas.

Indicador	Porcentaje de Pruebas Funcionales
Definición	Demostrar que el producto cumple con los requisitos de usuario al ejecutar los casos de prueba planificados para el producto.
Metas	Alcanzar el 80% de éxito de las pruebas funcionales en

	los requisitos del producto a la primera iteración luego de realizado el paso al ambiente de pruebas.
Métricas	<ul style="list-style-type: none"> • Número de casos de prueba ejecutados • Numero de defectos reportados
Periodicidad	Cada vez que se realice un paso al ambiente de pruebas
Procedimiento Análisis	<p>Calcular el porcentaje de los casos de prueba ejecutados y que no tengan un defecto reportado.</p> <p>%PF= 100-(ND/NCPE)*100</p> <p>Dónde:</p> <p>ND : Numero de defectos reportados</p> <p>NCPE : Número de casos de prueba ejecutados</p> <p>PF : Porcentaje de pruebas funcionales</p>
Forma de presentación	Gráfico de torta para cada proyecto
Herramientas	Casos de prueba ejecutadas, Registro de incidentes
Responsabilidades	Analista QA, Analista de Métricas.

Tabla N° 12: Tabla métrica de Pruebas Funcionales

Una vez que se ha identificado y diseñado las métricas de calidad, el siguiente paso fue adoptar el Modelo de Medición propuesto el que permitió establecer un marco de referencia para implementar la medición a nivel de los proyectos que se gestionan en el área de desarrollo de la OSIE.

A partir de este marco de referencia se elaboró un plan de medición, el cual queda establecido en el documento Plan de Medición (ver ANEXO 10) y que a

su vez pasa a ser el documento oficial que guía las actividades de recolección, medición y análisis de las fases establecidas en el Modelo de Medición propuesto y proyectos de la OSIE.

5.1.2 PROCESO: ELABORAR PLAN DE MEDICIÓN

Según ANEXO 10: se detalla el modelo del contenido del Plan de Medición.

5.1.3 PROCESO: EJECUTAR PLAN DE MEDICIÓN

5.1.3.1 FASE 1: PREPARAR RECOLECCIÓN DE DATOS

En esta fase tiene como objetivo preparar los datos para ser registrados de manera correcta y consistente:

- 1. Selección:** Este paso consiste en identificar cual es la fuente de información ya sea interna o externa para ser seleccionados solo los datos necesarios para nuestro estudio de las métricas.

Para nuestro caso se ha utilizado el IDE de Visual Studio 2015 que trae un plugin llamado Code Metrics Values de Visual Studio 2015, que obtiene valores para las métricas internas del producto de software.

Los resultados se pueden exportar a Excel haciendo más sencillo y flexible su manipulación para nuestro objetivo de medición.

- 2. Pre-procesamiento:** Este paso consiste en revisar la calidad de los datos obtenidos con la herramienta Code Metrics Values de Visual Studio

2015. Además de los valores de las métricas internas recogidas, se aplica las fórmulas definidas para poder hacer cálculos y obtener otras métricas.

- **Visual Studio 2015**

En la Figura 15, vemos el entorno del IDE de Microsoft Visual Studio 2015.

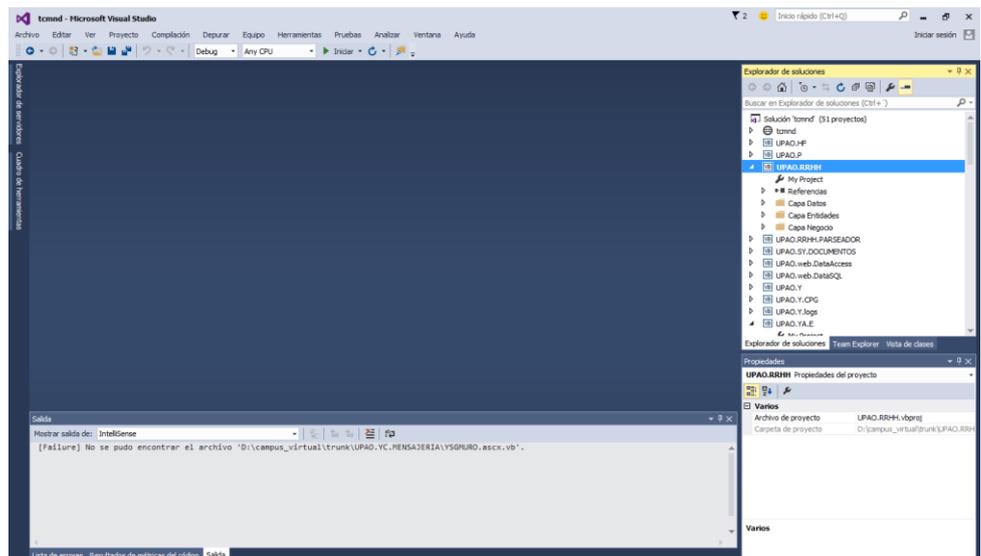


Figura N° 15: Entorno Visual Studio 2015

En la Figura 16, vemos el plugin de Code Metrics Values integrado al entorno del IDE de Microsoft Visual Studio 2015.

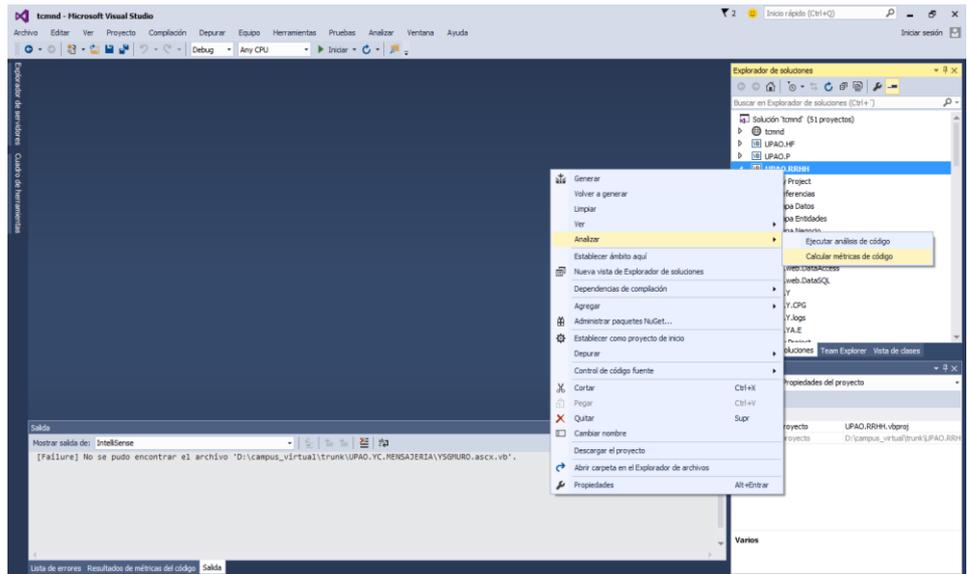


Figura N° 16: Obtener valores de métricas con Code Metrics Values para el proyecto de Recursos Humanos

En la Figura 17, vemos el resultado de obtener los valores de las métricas que calcula el Code Metrics Values, en este caso para el Proyecto de Recursos Humanos.

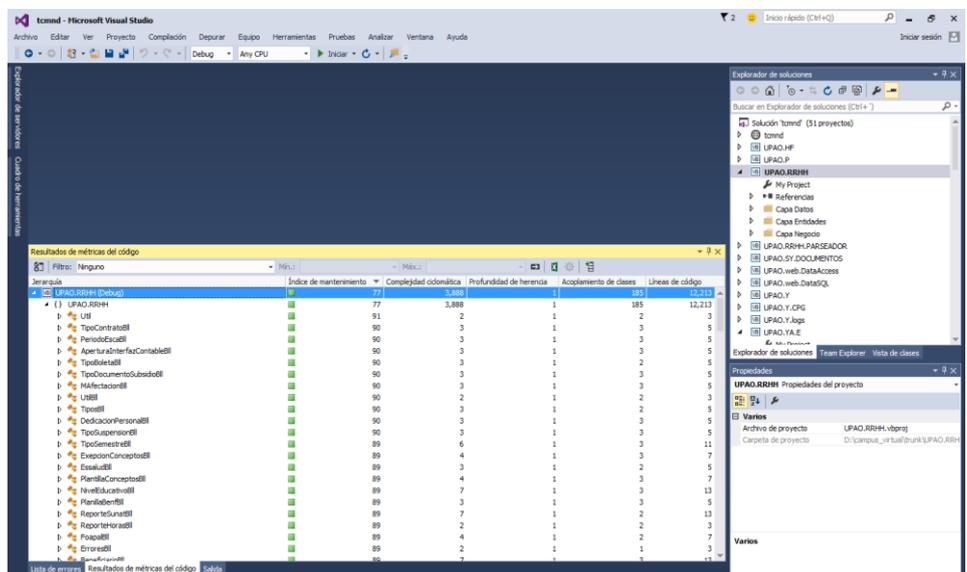


Figura N° 17: Resultado de valores de las métricas para el Proyecto de Recursos Humanos

En la Figura 18, vamos a exportar a Microsoft Excel los resultados de los valores de las

En Figura 20, ubicamos ahora en el IDE al Proyecto de Trámite Documentario para luego iniciar el proceso de cálculo de métricas.

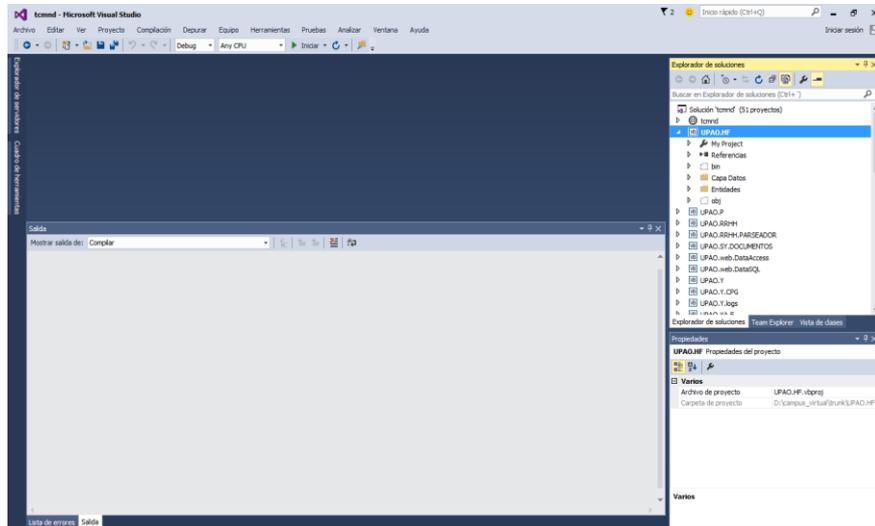


Figura N° 20: Obtener valores de métricas con Code Metrics Values para el proyecto de Trámite Documentario

En la Figura 21, vemos que Code Metrics Values ya hizo el cálculo para las métricas para el Proyecto de Trámite Documentario.

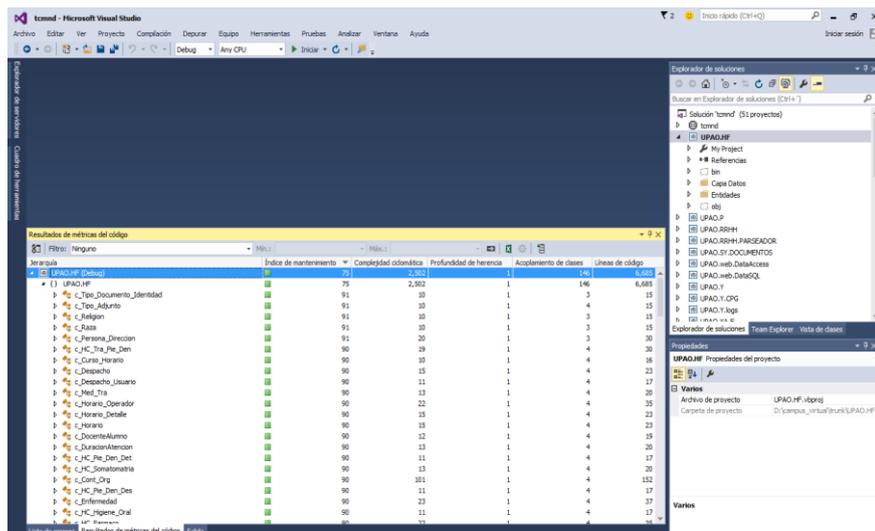


Figura N° 21: Resultado de valores de las métricas para el Proyecto de Trámite Documentario

En la Figura 22, vamos a exportar a Microsoft Excel los resultados de los valores de las métricas que calcula el Code Metrics Values, en este caso para el Proyecto de Trámite Documentario.

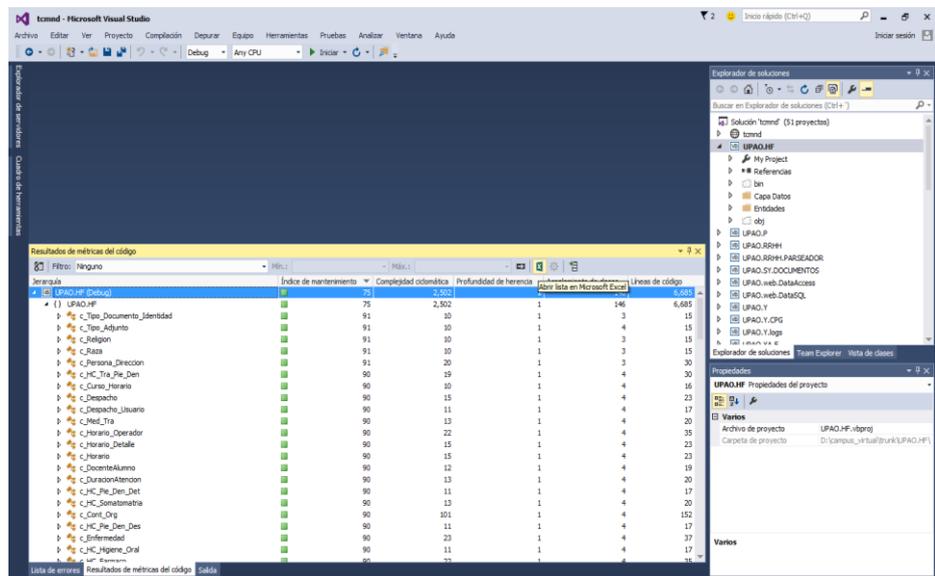


Figura N° 22: Exportar a Excel los valores de las métricas para el Proyecto de Trámite Documentario

En Figura 23, tenemos ya exportado a Microsoft Excel los valores de las métricas calculadas por Code Metrics Values, en un estado en bruto.

Ámbito	Espacio de nombres	Tipo	Miembro	Índice de mantenimiento	Complejidad ciclomática	Profundidad de herencia
Proyecto	UPAO.HF (Debug)	UPAO.HF		75	2502	1
Espacio de nombres	UPAO.HF (Debug)	UPAO.HF		91	10	1
Tipo	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identidad	98	1	
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identidad	p_codigo.Get() As Integer	98	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identidad	p_nombre.Get() As String	98	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identidad	p_usuario.Get() As String	98	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identidad	p_estado.Get() As String	95	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identidad	p_codigo.Set(Integer)	95	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identidad	p_nombre.Set(String)	95	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identidad	p_usuario.Set(String)	95	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identidad	p_estado.Set(String)	95	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identidad	f_listar_Act() As DataTable	84	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identidad	New()	73	1
Tipo	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjunto		91	10
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjunto	p_codigo.Get() As Integer	98	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjunto	p_nombre.Get() As String	98	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjunto	p_usuario.Get() As String	98	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjunto	p_estado.Get() As String	98	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjunto	p_codigo.Set(Integer)	95	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjunto	p_nombre.Set(String)	95	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjunto	p_estado.Set(String)	95	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjunto	f_list(Integer, String) As DataTable	82	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjunto	New()	73	1
Tipo	UPAO.HF (Debug)	UPAO.HF	c_Religion		91	10
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Religion	p_codigo.Get() As Integer	98	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Religion	p_nombre.Get() As String	98	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Religion	p_usuario.Get() As String	98	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Religion	p_estado.Get() As String	98	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Religion	p_codigo.Set(Integer)	95	1

Figura N° 23: Excel con valores exportados de Code Metrics Values para el Proyecto de Trámite Documentario

- **Simian**

La Figura 24, muestra el comando para ejecutar Simian y verifique las líneas de código duplicado en los archivos fuentes con extensión vb.

```

C:\Windows\system32\cmd.exe
C:\Users\nacaballeroa\Downloads\simian-2.4.0\bin>simian-2.4.0.exe **/*.vb >> RRHH.txt
    
```

Figura N° 24: Comando para ejecutar Simian en el Proyecto de Recursos Humanos

En la Figura 25, vemos el resultado que genera Simian una vez terminado el proceso de búsqueda de líneas de código repetidos en el código fuente.

```

MSRB Bloc de notas
Archivo Edición Formato Ver Ayuda

Found 22 duplicate lines in the following files:
Between lines 78 and 104 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\TipoSuspension.vb
Between lines 43 and 70 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\Procesos.vb
Between lines 43 and 70 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\TipoDocumentosSubsidio.vb
Found 22 duplicate lines in the following files:
Between lines 102 and 129 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\VacacionesEmpleado.vb
Between lines 115 and 139 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\SubsidioEmpleado.vb
Found 22 duplicate lines in the following files:
Between lines 5 and 46 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Negocio\TareaAdicionalB11.vb
Between lines 5 and 45 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\TareaAdicional.vb
Found 23 duplicate lines in the following files:
Between lines 34 and 62 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\TablaGeneral.vb
Between lines 22 and 50 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\GrupoOcupacional.vb
Between lines 22 and 50 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\TipoPersonal.vb
Found 27 duplicate lines in the following files:
Between lines 155 and 183 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Datos\ContratoDal.vb
Between lines 103 and 131 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Datos\ContratoDal.vb
Found 28 duplicate lines in the following files:
Between lines 186 and 180 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\WPlanilla.vb
Between lines 127 and 161 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\BoletaC.vb
Found 30 duplicate lines in the following files:
Between lines 7 and 44 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\PlantillaConceptos.vb
Between lines 7 and 45 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\ExpcionConceptos.vb
Found 30 duplicate lines in the following files:
Between lines 33 and 70 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\Procesos.vb
Between lines 33 and 70 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\TipoDocumentosSubsidio.vb
Found 33 duplicate lines in the following files:
Between lines 290 and 333 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Negocio\BoletaCB11.vb
Between lines 417 and 460 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Negocio\BoletaCB11.vb
Found 38 duplicate lines in the following files:
Between lines 718 and 761 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Datos\ContratoDal.vb
Between lines 658 and 699 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Datos\ContratoDal.vb
Found 40 duplicate lines in the following files:
Between lines 1035 and 1086 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Datos\WPlanillaDal.vb
Between lines 15 and 66 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Datos\BoletaCBal.vb
Found 46 duplicate lines in the following files:
Between lines 84 and 131 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Datos\ContratoDal.vb
Between lines 31 and 78 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Datos\ContratoDal.vb
Found 48 duplicate lines in the following files:
Between lines 10 and 71 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\TipoPersonal.vb
Between lines 10 and 71 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\GrupoOcupacional.vb
Found 50 duplicate lines in the following files:
Between lines 221 and 284 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\WPlanilla.vb
Between lines 216 and 279 in C:\Users\acaballeroa\Downloads\simian-2.4.0\bin\Capa Entidades\BoletaC.vb
Found 9216 duplicate lines in 859 blocks in 181 files
Processed a total of 18952 significant (26870 raw) lines in 181 files
Processing time: 1.189sec

```

Figura N° 25: Resultado de ejecución de Simian para el Proyecto de Recursos Humanos

5.1.3.2 FASE 2: RECOLECTAR, VALIDAR Y ANALIZAR LOS DATOS

1. Recolectar: Se usa la herramienta Excel que es lo que exporta el Code Metrics Values de Visual Studio 2015

En la Figura 26, se muestra el archivo Excel que fue exportado con Code Metrics Values listo para iniciar la preparación y limpieza.

Ambito	Proyecto	Espacio de nombres	Tipo	Índice de mantenimiento	Complejidad ciclomática	Profundidad de herencia	Acoplamiento de clases	Líneas de código
Proyecto	UPAO.RRHH (Debug)	UPAO.RRHH		77	3888	1	185	12213
Espacio de nombres	UPAO.RRHH (Debug)	UPAO.RRHH		77	3888	1	185	12213
Tipo	UPAO.RRHH (Debug)	UPAO.RRHH	Util	91	2	1	2	3
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	Util	92	1	1	1	1
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	Util	84	1	2	2	2
Tipo	UPAO.RRHH (Debug)	UPAO.RRHH	TipoContratoBil	90	3	1	3	5
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	TipoContratoBil	92	1	1	1	1
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	TipoContratoBil	84	1	2	2	2
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	TipoContratoBil	84	1	2	2	2
Tipo	UPAO.RRHH (Debug)	UPAO.RRHH	PeriodoEscalaBil	90	3	1	3	5
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	PeriodoEscalaBil	92	1	1	1	1
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	PeriodoEscalaBil	84	1	2	2	2
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	PeriodoEscalaBil	84	1	2	2	2
Tipo	UPAO.RRHH (Debug)	UPAO.RRHH	AperturaInterfazConta	90	3	1	3	5
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	AperturaInterfazConta	92	1	1	1	1
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	AperturaInterfazConta	84	1	2	2	2
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	AperturaInterfazConta	84	1	2	2	2
Tipo	UPAO.RRHH (Debug)	UPAO.RRHH	TipoBoletaBil	90	3	1	2	5
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	TipoBoletaBil	92	1	1	1	1
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	TipoBoletaBil	84	1	2	2	2
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	TipoBoletaBil	84	1	2	2	2
Tipo	UPAO.RRHH (Debug)	UPAO.RRHH	TipoDocumentoSubsic	90	3	1	3	5
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	TipoDocumentoSubsic	92	1	1	1	1
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	TipoDocumentoSubsic	84	1	2	2	2
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	TipoDocumentoSubsic	84	1	2	2	2
Tipo	UPAO.RRHH (Debug)	UPAO.RRHH	MAfectacionBil	90	3	1	3	5
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	MAfectacionBil	92	1	1	1	1
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	MAfectacionBil	84	1	2	2	2
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	MAfectacionBil	84	1	2	2	2
Tipo	UPAO.RRHH (Debug)	UPAO.RRHH	UtilBil	90	2	1	2	3
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	UtilBil	92	1	1	1	1
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	UtilBil	84	1	2	2	2
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	UtilBil	84	1	2	2	2
Tipo	UPAO.RRHH (Debug)	UPAO.RRHH	TiposBil	90	3	1	1	5
Miembro	UPAO.RRHH (Debug)	UPAO.RRHH	TiposBil	92	1	1	1	1
Tipo	UPAO.RRHH (Debug)	UPAO.RRHH	TiposBil	84	1	2	2	2

Figura N° 26: Archivo Excel exportado desde Visual Studio 2015 sin modificación – Proyecto Recursos Humanos

En la Figura 27, se muestra la limpieza que se va haciendo al archivo Excel para dejar solo los datos necesarios para las métricas.

Espacio de nombres	Tipo	Índice de mantenimiento	Volumen de Halstead	Complejidad ciclomática	Profundidad de herencia	Acoplamiento de clases	Líneas de código
UPAO.RRHH		77		3888	1	185	12213
	Util	91		2	1	2	3
	TipoContratoBil	90		3	1	3	5
	PeriodoEscalaBil	90		3	1	3	5
	PeriodoEscalaBil	92		1	1	1	1
	PeriodoEscalaBil	84		1	2	2	2
	AperturaInterfazContableBil	90		3	1	3	5
	AperturaInterfazContableBil	92		1	1	1	1
	AperturaInterfazContableBil	84		1	2	2	2
	AperturaInterfazContableBil	84		1	2	2	2
	TipoBoletaBil	90		3	1	2	5
	TipoBoletaBil	92		1	1	1	1
	TipoBoletaBil	84		1	2	2	2
	TipoBoletaBil	84		1	2	2	2
	TipoDocumentoSubsidioBil	90		3	1	3	5
	TipoDocumentoSubsidioBil	92		1	1	1	1
	TipoDocumentoSubsidioBil	84		1	2	2	2
	TipoDocumentoSubsidioBil	84		1	2	2	2
	MAfectacionBil	90		3	1	3	5
	MAfectacionBil	92		1	1	1	1
	MAfectacionBil	84		1	2	2	2
	MAfectacionBil	84		1	2	2	2
	UtilBil	90		2	1	2	3
	UtilBil	92		1	1	1	1
	UtilBil	84		1	2	2	2
	TiposBil	90		3	1	1	5
	TiposBil	92		1	1	1	1
	TiposBil	84		1	2	2	2

Figura N° 27: Limpieza y transformación del archivo Excel – Proyecto Recursos Humanos

En la Figura 28, se muestra el archivo Excel que fue exportado con Code Metrics Values listo para iniciar la preparación y limpieza.

Ambito	Proyecto	Espacio de nombres	Tipo	Indice de mantenimiento	Complejidad ciclomática	Profundidad de herencia	Acoplamiento de clases	Lineas de código
Proyecto	UPAO.HF (Debug)	UPAO.HF		75	2502	1	146	6685
Tipo	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identic	91	10	1	3	15
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identic	98	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identic	98	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identic	98	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identic	95	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identic	95	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identic	95	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identic	84	1	2	2	2
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Documento_Identic	73	1	1	5	5
Tipo	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjuntio	91	10	1	4	15
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjuntio	98	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjuntio	98	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjuntio	98	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjuntio	98	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjuntio	95	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjuntio	95	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjuntio	95	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjuntio	82	1	3	2	2
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Tipo_Adjuntio	73	1	1	5	5
Tipo	UPAO.HF (Debug)	UPAO.HF	c_Religion	91	10	1	3	15
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Religion	98	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Religion	98	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Religion	98	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Religion	98	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Religion	98	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Religion	98	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Religion	98	1	1	1	1
Miembro	UPAO.HF (Debug)	UPAO.HF	c_Religion	95	1	1	1	1

Figura N° 28: Archivo Excel exportado desde Visual Studio 2015 sin modificación – Proyecto Trámite Documentario

En la Figura 29, se muestra el proceso de limpieza de datos del archivo Excel dejando solo lo necesario para el cálculo de métricas.

Espacio de nombres	Tipo	Indice de mantenimiento	Volumen de Halstead	Complejidad ciclomática	Profundidad de herencia	Acoplamiento de clases	Lineas de código
UPAO.HF		75		2502	1	146	6685
	c_Tipo_Documento_Identidad	91		10	1	3	15
	c_Tipo_Adjuntio	91		10	1	4	15
	c_Tipo_Adjuntio	98		1	1	1	1
	c_Tipo_Adjuntio	98		1	1	1	1
	c_Tipo_Adjuntio	98		1	1	1	1
	c_Tipo_Adjuntio	95		1	1	1	1
	c_Tipo_Adjuntio	95		1	1	1	1
	c_Tipo_Adjuntio	95		1	1	1	1
	c_Tipo_Adjuntio	95		1	1	1	1
	c_Tipo_Adjuntio	82		1	3	2	2
	c_Tipo_Adjuntio	73		1	1	5	5
	c_Religion	91		10	1	3	15
	c_Religion	98		1	1	1	1
	c_Religion	98		1	1	1	1
	c_Religion	98		1	1	1	1
	c_Religion	98		1	1	1	1
	c_Religion	95		1	1	1	1
	c_Religion	95		1	1	1	1
	c_Religion	95		1	1	1	1
	c_Religion	95		1	1	1	1
	c_Religion	84		1	2	2	2
	c_Religion	73		1	1	5	5
	c_Raza	91		10	1	3	15
	c_Raza	98		1	1	1	1
	c_Raza	98		1	1	1	1
	c_Raza	98		1	1	1	1
	c_Raza	98		1	1	1	1
	c_Raza	95		1	1	1	1

Figura 29: Limpieza y transformación del archivo Excel – Proyecto Trámite Documentario

2. **Validar:** Una vez que se tenga el Excel se procede a ordenar y a validar los datos de acuerdo a las fórmulas establecidas para cada métrica de calidad.

En la Figura 30, se muestra el archivo Excel finalizado la modificación con la métrica de Índice de Mantenibilidad para el Proyecto de Recursos Humanos.

		E	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																
16																
17																
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																
29																
30																
31																
32																
33																
34																
35																
36																
37																
38																
39																
40																
41																
42																
43																
44																
45																
46																
47																
48																
49																
50																
51																
52																
53																
54																
55																
56																
57																
58																
59																
60																
61																
62																
63																
64																
65																
66																
67																
68																
69																
70																
71																
72																
73																
74																
75																
76																
77																
78																
79																
80																
81																
82																
83																
84																
85																
86																
87																
88																
89																
90																
91																
92																
93																
94																
95																
96																
97																
98																
99																
100																

Figura N° 30: Archivo Excel modificado con los cálculos necesarios – Proyecto Recursos Humanos

En la Figura 31, se muestra el archivo Excel finalizado la modificación con la métrica de Índice de Mantenibilidad para el Proyecto de Trámite Documentario.

	Indice de Mantenimiento	Complejidad Ciclomática	Profundidad Herencia	Acoplamiento Clase	LDC
Proyecto					
1 TRAMITE DOCUMENTARIO	75	2518	142	896	6728
Clases	74.80	17.73	1	6.31	
1 c_Adjunto	90	22	1	4	34.00
2 c_AdjuntoDal	60	8	1	9	36.00
3 c_Area	90	11	1	4	17.00
4 c_Area_Tratamiento	90	12	1	4	19.00
5 c_Area_TratamientoDal	61	9	1	9	37.00
6 c_AreaDal	61	6	1	9	27.00
7 c_Asig_Cajero	90	18	1	4	28.00
8 c_Asig_CajeroDal	59	9	1	9	43.00
9 c_Campania_Gratis	89	15	1	4	25.00
10 c_Campania_GratisDal	61	16	1	9	63.00
11 c_CategoriaEnfermedad	89	31	1	4	56.00
12 c_CategoriaEnfermedadDal	58	23	1	9	102.00
13 c_Cit_Vir	90	17	1	4	26.00
14 c_Cit_VirDal	60	6	1	9	29.00
15 c_Color	90	19	1	4	31.00
16 c_ColorDal	60	16	1	9	67.00
17 c_Consultorio	90	101	1	4	152.00
18 c_ConsultorioDal	47	6	1	9	71.00
19 c_Cont_Org	90	10	1	4	16.00
20 c_Cont_OrgDal	62	9	1	9	35.00
21 c_Curso_Horario	90	15	1	4	23.00
22 c_Curso_HorarioDal	90	11	1	4	17.00
23 c_Despacho	60	6	1	9	29.00
24 c_Despacho_Usuario	59	6	1	9	30.00
25 c_Despacho_UsuarioDal	90	12	1	4	19.00

Figura N° 31: Archivo Excel modificado con los cálculos necesarios – Proyecto Trámite Documentario

En la Figura 32, se muestra el resultado del archivo Excel final de líneas de código duplicado para el Proyecto de Trámite Documentario.

	Líneas duplicadas	Líneas no duplicadas	Total
Proyecto			
1 TRAMITE DOCUMENTARIO	4131	2597	6728
Clases			
1 c_Adjunto	31		
2 c_AdjuntoDal	44		
3 c_Area	39		
4 c_Area_Tratamiento	31		
5 c_Area_TratamientoDal	8		
6 c_AreaDal	17		
7 c_Asig_Cajero	17		
8 c_Asig_CajeroDal	49		
9 c_Campania_Gratis	3		
10 c_Campania_GratisDal	10		
11 c_CategoriaEnfermedad	47		
12 c_CategoriaEnfermedadDal	40		
13 c_Cit_Vir	27		
14 c_Cit_VirDal	42		
15 c_Color	4		
16 c_ColorDal	1		
17 c_Consultorio	10		
18 c_ConsultorioDal	24		
19 c_Cont_Org	19		
20 c_Cont_OrgDal	27		
21 c_Curso_Horario	32		
22 c_Curso_HorarioDal	44		

Figura N° 32: Archivo Excel Modificado con los valores para métrica Porcentaje de Código Duplicado – Proyecto Trámite Documentario

3. **Analizar:** Finalmente se procede a analizar los resultados de acuerdo a los rangos establecidos por cada métrica de calidad para verificar si está dentro de los rangos de calidad de software.

	Índice de Mantenibilidad	Complejidad Ciclomática	Acoplamiento de Clases	Profundidad de Herencia
Excelente	> 70	< 10	< 10	=< 3
Muy bueno	> 60	10 – 12	< 15	4
Bueno	40 – 60	12- 15	< 20	5
Malo	20 – 40	15 – 20	< 30	6 - 8
Muy malo	< 20	> 20	> 30	> 8

Tabla N° 13 : Tabla de Riesgo de las métricas de calidad

- PROYECTO DE RECURSOS HUMANOS:

a) Métrica: Índice de Mantenibilidad

$$IM = \text{MAX}(0, (171 - 5.2 * \ln(V) - 0.23 * (CC) - 16.2 * \ln(LOC)) * 100 / 171)$$

Dónde:

V : Volumen de Halstead

CC : Complejidad Ciclomática

LOC: Líneas de Código

El índice de mantenibilidad de todo el Proyecto de Recursos Humanos es de: **76.90**

Éste valor cae en el rango de 20-100, lo cual nos permite afirmar que es un producto de software que es de fácil mantenimiento.

Métrica: Complejidad Ciclomática

El valor que hemos calculado es de un promedio de 22.22 para todo el Proyecto de Recursos Humanos, el cual según la tabla de valores, vemos que es muy malo.

Métrica: Acoplamiento de Clases

El valor que hemos calculado es de un promedio de 1 para todo el Proyecto de Recursos Humanos, el cual según la tabla de valores, vemos que es Excelente.

Métrica: Profundidad de Herencia

El valor que hemos calculado es de un promedio de 4.88 para todo el Proyecto de Recursos Humanos, el cual según la tabla de valores vemos que es bueno.

Vamos a resumir en una tabla los valores obtenidos para el Proyecto de Recursos Humanos:

Complejidad Ciclomática	22.22
Acoplamiento de Clases	1
Profundidad de Herencia	4.88
Líneas de Código	12,213
Índice de Mantenibilidad	76.90

Tabla N° 14 : Tabla resumen de valores de métricas para el Proyecto de Recursos Humanos

b) Métrica: Porcentaje de Código Duplicado

El cálculo se hace a través de la siguiente fórmula:

$$\%LCD = (NLR / LDC) * 100$$

$$\%LCD = (9,216/12,213)*100$$

$$\%LCD = 75\%$$

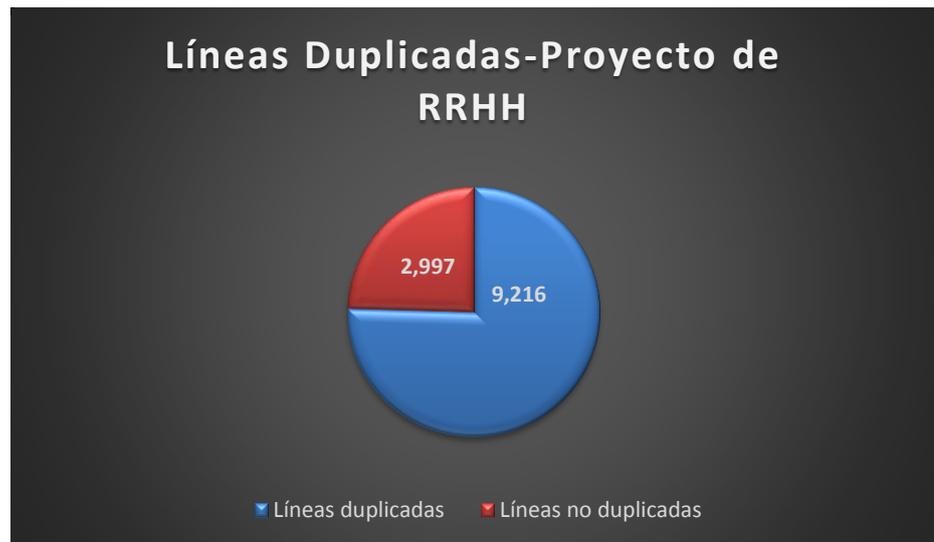


Figura N° 33: Porcentaje de Código Duplicado para el Proyecto de Recursos Humanos

Como se puede observar en el gráfico (Figura 33), el porcentaje de código repetido es de 75%, lo cual muestra que más de la mitad en todo el proyecto ha sido copiado y pegado de líneas de código fuente del total del software, lo que nos predice que posteriormente habrá muchos problemas para su mantenimiento y por tanto requiere de tiempo para poder hacer cambios en el sistema.

c) Métrica de Porcentaje de Pruebas Funcionales

$$\%PF = 100 - (ND/NCPE) * 100$$

$$\%PF = 100 - (55/59) * 100$$
$$\%PF = 100 - 93.22\% = \mathbf{6.7\%}$$

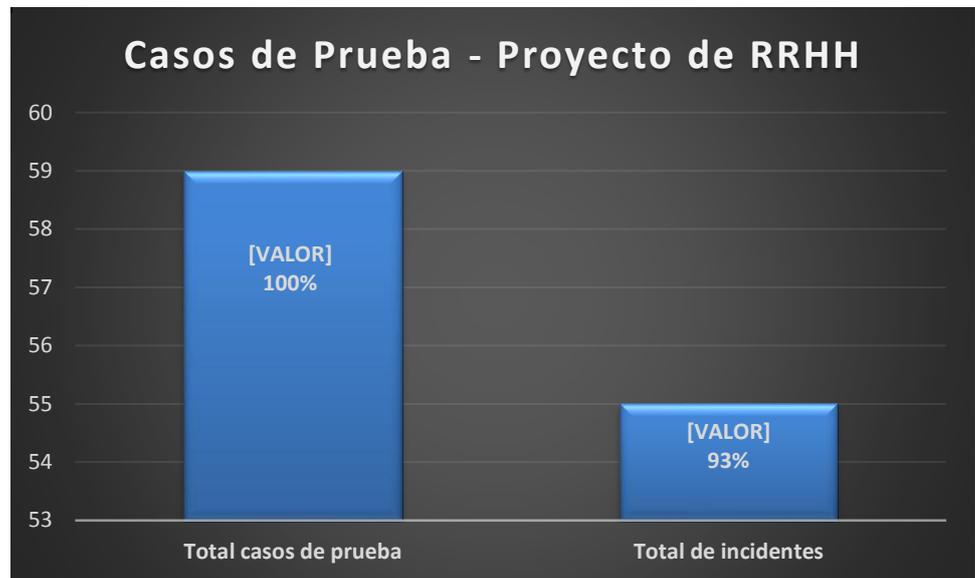


Figura N° 34: Gráfico de porcentaje de código duplicado para el Proyecto de Recursos Humanos

Según el gráfico de barras, el **93%** representa el total de incidentes reportados durante el proceso de puesta en producción, lo cual indica que del 100% solo el **6.7%** de pruebas funcionales se realizaron en el proceso de desarrollo. De otra manera, de los 59 casos de pruebas solo 7 casos se probaron completamente de los requerimientos funcionales que conforman el Sistema de RRHH.

Esto de hecho representa una mala calidad respecto al producto de software ya se ha sacrificado las pruebas funcionales por diferentes razones como falta de tiempo o una mala revisión de la funcionalidad y que en

conclusión ocasiona malestar en los usuarios finales, trayendo una mala imagen a la OSIE.

- PROYECTO DE TRÁMITE DOCUMENTARIO:

- a) **Métrica Índice de Mantenibilidad**

$$IM = \text{MAX}(0, (171 - 5.2 * \ln(V) - 0.23 * (CC) - 16.2 * \ln(\text{LOC})) * 100 / 171)$$

Dónde:

V : Volumen de Halstead

CC : Complejidad Ciclomática

LOC: Líneas de Código

El índice de mantenibilidad de todo el Proyecto de Trámite Documentario es de: **75.00**

Éste valor cae en el rango de 20-100, lo cual nos permite afirmar que es un producto de software que es de fácil mantenimiento.

Métrica: Complejidad Ciclomática

El valor que hemos calculado es de un promedio de 17.73 para todo el Proyecto de Trámite Documentario, el cual según la tabla de valores, vemos que es muy malo.

Métrica: Acoplamiento de Clases

El valor que hemos calculado es de un promedio de 1 para todo el Proyecto de Trámite Documentario, el cual según la tabla de valores, vemos que es Excelente.

Métrica: Profundidad de Herencia

El valor que hemos calculado es de un promedio de 6.31 para todo el Proyecto de Trámite Documentario, el cual según la tabla de valores vemos que es bueno.

Vamos a resumir en una tabla los valores obtenidos para el Proyecto de Trámite Documentario:

Complejidad Ciclomática	17.73
Acoplamiento de Clases	1
Profundidad de Herencia	6.31
Líneas de Código	6,728
Índice de Mantenibilidad	75

Tabla N° 15: Tabla resumen de valores de métricas para el Proyecto de Trámite Documentario

b) Métrica: Porcentaje de Código Duplicado

El cálculo se hace a través de la siguiente fórmula:

$$\%LCD = (NLR / LDC) * 100$$

$$\%LCD = (4,131/6,728)*100$$

$$\%LCD = 61.4\%$$

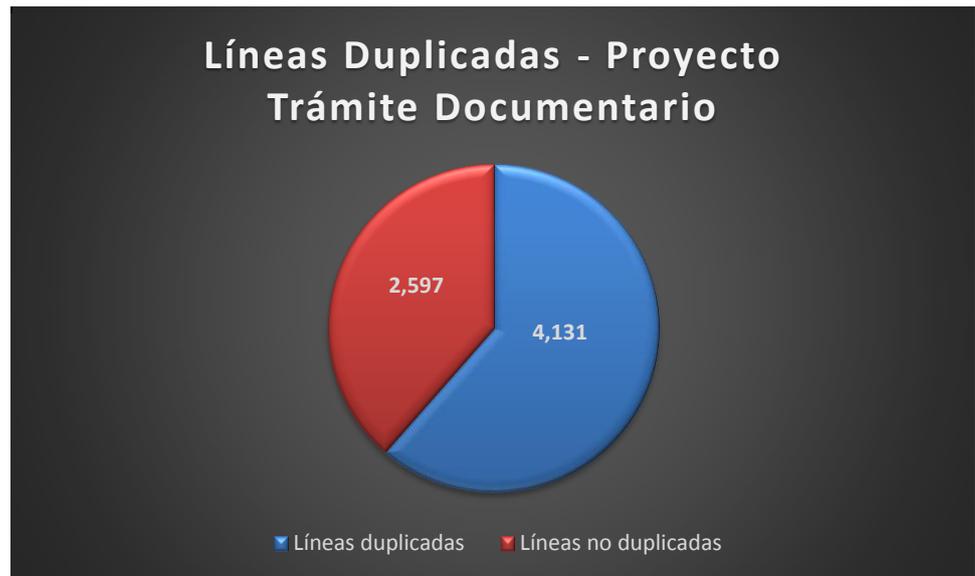


Figura N° 35: Porcentaje de Código Duplicado para el Proyecto de Trámite Documentario

Como se puede observar en el gráfico (Figura 35), el porcentaje de código repetido es de **61%**, lo cual muestra que más de la mitad en todo el proyecto ha sido copiado y pegado de líneas de código fuente del total del software, lo que nos predice que posteriormente habrá muchos problemas para su mantenimiento y por tanto requiere de tiempo para poder hacer cambios en el sistema.

a) Métrica de Porcentaje de Pruebas Funcionales

$$\%PF = 100 - (ND/NCPE) * 100$$

$$\%PF = 100 - (40/72) * 100$$

$$\%PF = 100 - 56 = 44\%$$

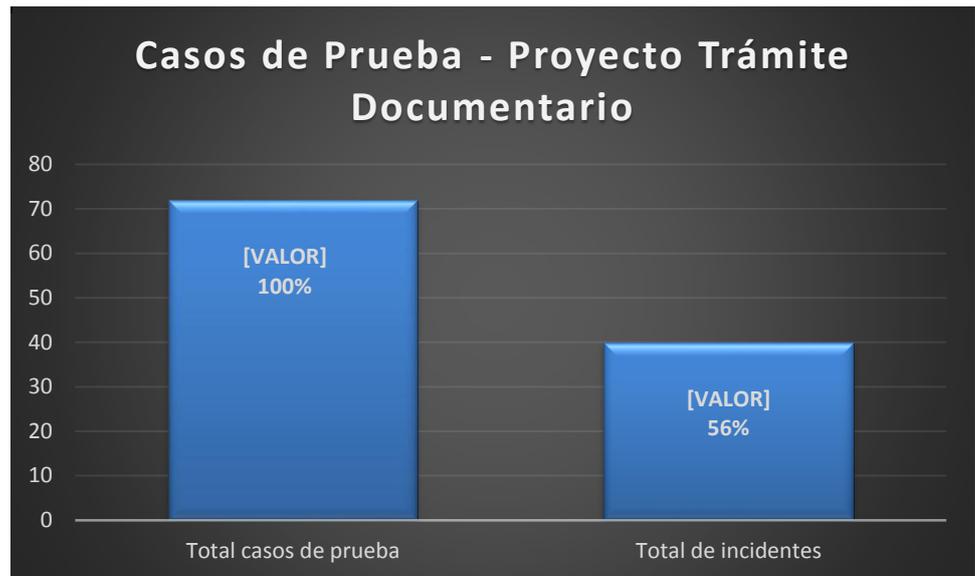


Figura N° 36: Porcentaje de Pruebas Funcionales para el Proyecto de Trámite Documentario

Según el grafico de barras, el **61%** representa el total de incidentes reportados durante el proceso de puesta en producción, lo cual indica que del 100% solo el **44%** de pruebas funcionales se realizaron en el proceso de desarrollo. De otra manera, de los 72 casos de pruebas solo 32 casos se probaron completamente de los requerimientos funcionales que conforman el Sistema de Trámite Documentario.

Esto de hecho representa una mala calidad respecto al producto de software ya se ha sacrificado las pruebas funcionales por diferentes razones como falta de tiempo o una mala revisión de la funcionalidad y que en conclusión ocasiona malestar en los usuarios finales, trayendo una mala imagen a la OSIE.

5.1.4 PROCESO: INTERPRETAR Y EVALUAR LA MEDICIÓN

En ésta fase se interpreta los resultados antes de aplicar los criterios para la calidad de software y posterior a la aplicación y recomendaciones para el mejoramiento de la calidad.

Proyecto	Métrica de Calidad	Valor
Recursos Humanos	• Índice de Mantenibilidad	77
	• Porcentaje de Código Duplicado	75%
	• Porcentaje de Pruebas Funcionales	6.7%
Trámite Documentario	• Índice de Mantenibilidad	75
	• Porcentaje de Código Duplicado	61%
	• Porcentaje de Pruebas Funcionales	44%

Tabla N° 16 : Tabla de Resultados del Cálculo de las Métricas para los Proyectos

INTERPRETACIÓN

En ambos proyectos donde se aplicaron el Modelo de Medición Propuesto y las métricas de calidad interna, los valores del índice mantenibilidad es alto y es muy bueno para realizar actualizaciones al código fuente, sin embargo vemos que el porcentaje de código duplicado y el porcentaje de pruebas funcionales son muy malos, lo que nos hace pensar que a mediano y largo plazo nos generará incidentes y errores por la mala calidad del producto de software.

El índice de mantenibilidad se calcula en base a la complejidad ciclomática que mide las condiciones del software, líneas de código y el volumen de Halstead. Pero si evaluamos la calidad respecto a pruebas rigurosas y duplicidad de código, entonces tenemos una visión más amplia en cuanto a calidad interna y nos permite evaluar cómo se está construyendo el producto.

CAPITULO VI: RESULTADOS

En este capítulo, se muestran los resultados de la evaluación del modelo PSM y GQM propuesto, a través de los resultados obtenidos de la aplicación del Modelo de Medición propuesto en el Área de Desarrollo de la OSIE.

6.1. Propuesta del Modelo de Medición

Como resultado del estudio de la metodología PSM y GQM, se propuso el modelo siguiente de métricas de calidad de software que se muestra en la siguiente figura:

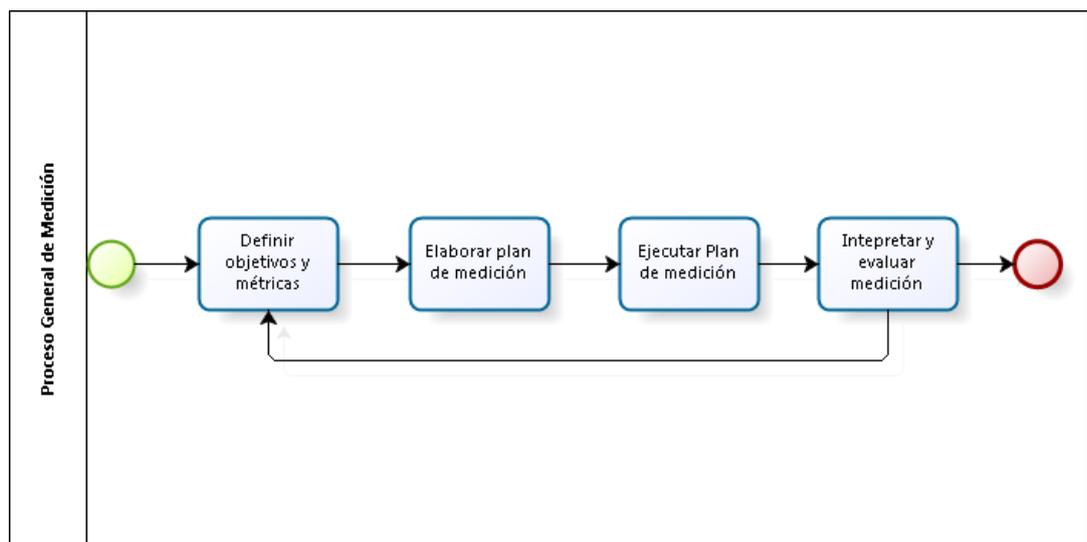


Figura N° 37: Proceso General del Modelo de Medición Propuesto

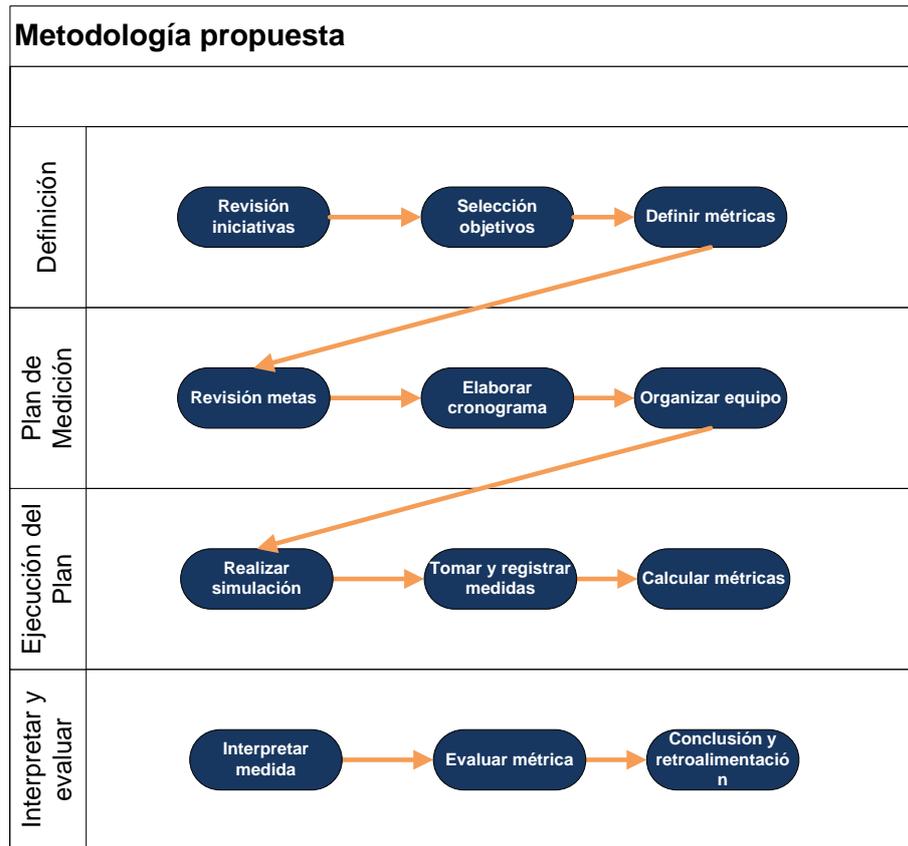


Figura N° 38: Modelo de Medición detallado

6.2. Despliegue del Modelo de Medición propuesto

En el siguiente gráfico se muestra el proceso de medición aplicando el Modelo de Medición propuesto y la herramienta de Code Metrics Values con Visual Studio 2015:

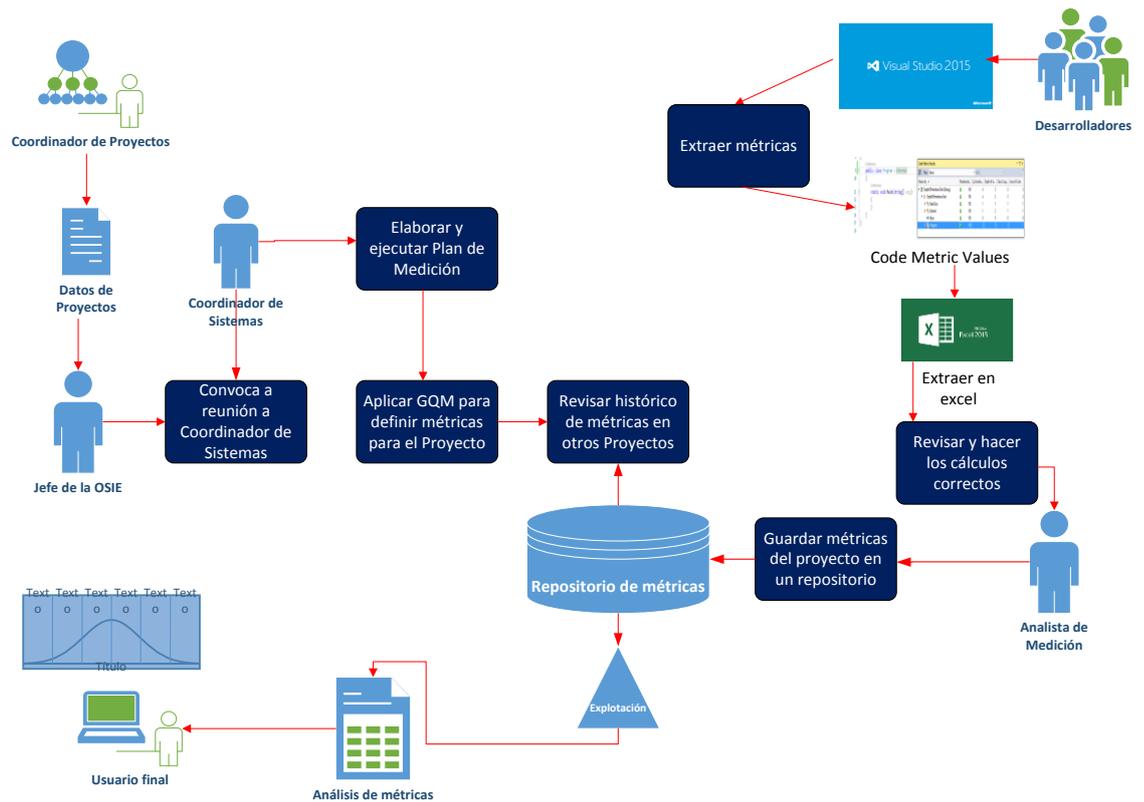


Figura N° 39: Proceso del Modelo de Medición propuesto

6.3. Validación del Modelo de Métricas de Calidad

VARIABLE INDEPENDIENTE: el Modelo de Medición para el proceso de desarrollo de Software

Los resultados de la evaluación del Modelo de Medición a través de las encuestas son las siguientes:

6.3.1 Cualidades evaluadas por expertos en el Modelo de Medición propuesta

Para poder obtener las siguientes puntuaciones de los atributos del Modelo de Medición propuesta, se implementó

una solución de medición y se expuso a cuatro expertos en el desarrollo de Sistemas de la OSIE, los cuales respondieron a la encuesta mostrada en el ANEXO 05, obteniendo los resultados mostrados en el ANEXO 06, en la siguiente tabla se muestra el promedio de la puntuación por atributo otorgada por los expertos del modelo propuesto.

Puntuación del Atributo por Expertos (Máxima Puntuación 100)					
Atributo	Experto 1	Experto 2	Experto 3	Experto 4	Promedio del Modelo Propuesto
Cumplimiento para CMMI Nivel 2	64	67	63	66	65,00
Facilidad de aplicación	70	71	73	75	72,25
Adaptación del modelo	82	83	86	81	83,00
Promedio de Atributos del Modelo					73,42

Tabla N° 17: Puntuación de atributos por expertos, en el Modelo de Medición propuesto

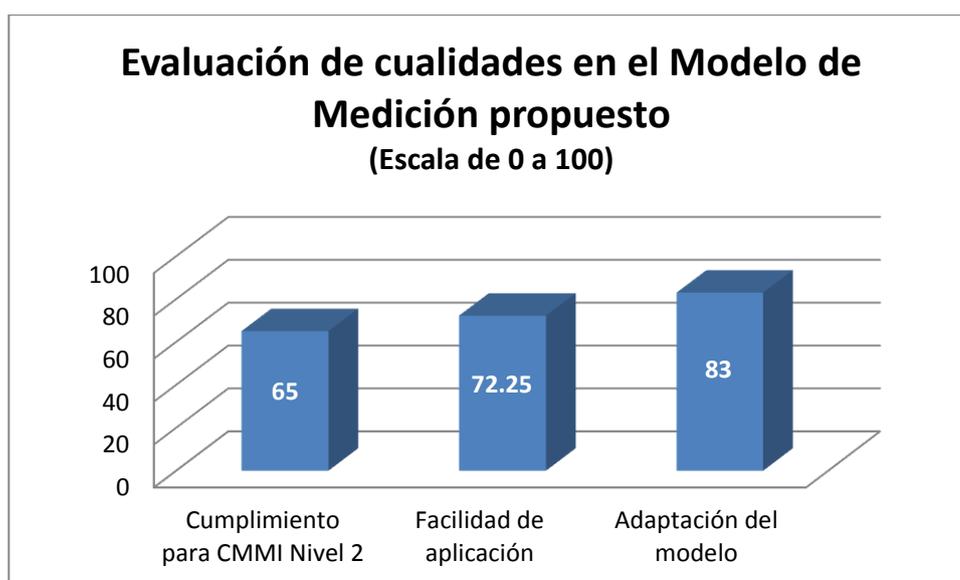


Figura N° 40: Puntuación promedio de Cualidades Evaluadas por expertos en el Modelo de Medición propuesto

Se puede aplicar el Alfa de Crombach a todos los resultados obtenidos y validarlos.

Para poder identificar que los datos que obtuvimos como RESULTADO son fiables el Alfa de Crombach debe tener un valor cerca de 1, caso contrario no serán datos útiles para poder aplicar análisis estadístico y en ese sentido se sugeriría una normalización de datos.

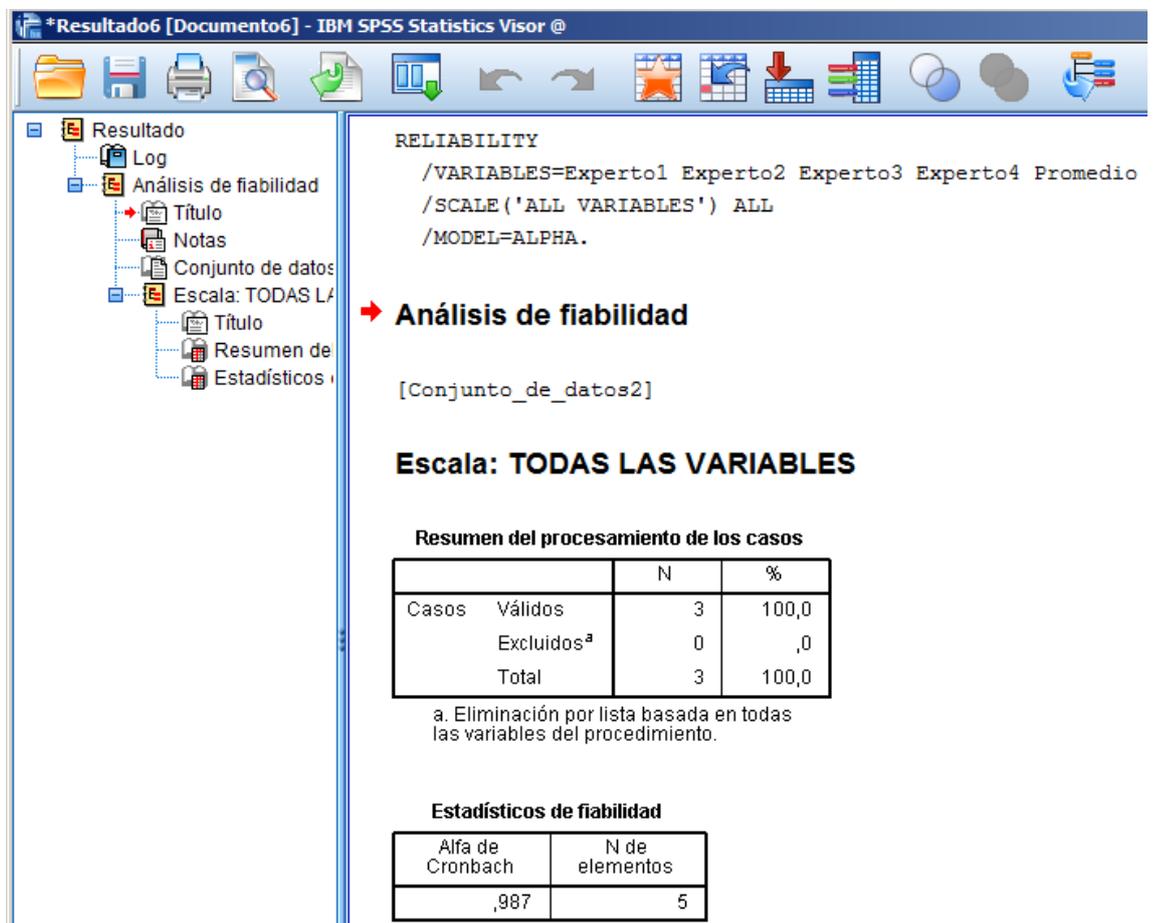


Figura N° 41: Resultado del Alfa de Crombach con IBM SPSS

En el presente caso el Alfa de Crombach tiene un valor de 0.987 en las cinco columnas o variables SPSS.

El Alfa de Crombach nos permite medir la fiabilidad (es decir, si los datos escalares son factibles de aplicar el Alfa de Crombach), de una escala de medida que muestra indirectamente una cualidad observable en una población.

VARIABLE DEPENDIENTE: Métricas de Calidad para el producto de software.

Los resultados de ésta variable fueron tomados en un solo tiempo, lo que nos da un diagnóstico actual de cómo se encuentran los productos de software desarrollados por la OSIE.

1. Índice de Mantenibilidad

Proyecto	Valor	Calificación
Recursos Humanos	76.90	Excelente
Trámite Documentario	75.00	Excelente

Tabla N° 18 : Resultado de la evaluación del indicador-Índice de Mantenibilidad

2. Porcentaje de Código Duplicado

Proyecto	Valor	Calificación
Recursos Humanos	75%	Mucho riesgo Necesita Refactoring
Trámite Documentario	61%	Mucho riesgo Necesita Refactoring

Tabla N° 19: Resultado de la evaluación del indicador-Porcentaje de Código Duplicado

3. Porcentaje de Pruebas Funcionales

Proyecto	Valor	Calificación
Recursos Humanos	7%	Alto Riesgo No se hizo las pruebas funcionales de manera rigurosa
Trámite Documentario	44%	Mediano Riesgo No se hizo las pruebas funcionales de manera rigurosa

Tabla N° 20 : Resultado de la evaluación del indicador-Pruebas Funcionales

CAPITULO VII: DISCUSIÓN DE RESULTADOS

En el presente capítulo, se somete a discusión, mediante estadística descriptiva, los resultados obtenidos de la evaluación de los atributos del Modelo de Medición propuesto para el presente proyecto.

Asimismo se somete a discusión, mediante estadística descriptiva, los resultados obtenidos de la evaluación de los indicadores para la calidad de software, de la aplicación del Modelo de Medición propuesto, para exponer que la hipótesis: “La propuesta de Modelo de Medición con métricas para el proceso de desarrollo de software, permitirá mejorar la calidad del producto de software en el Área de Desarrollo de la Oficina de Sistemas e Ingeniería de la Información de la Universidad Privada Antenor Orrego, año 2013” ha sido demostrada.

7.1. ANÁLISIS DEL PROCESO DE MEDICIÓN

Del estudio del Proceso de Medición en el Área de Desarrollo-OSIE de la Universidad Privada Antenor Orrego, se identificaron los actores y actividades más relevantes que permitan realizar una buena medición:

Actores y actividades

- Jefe de la OSIE: establece el alineamiento de las métricas con los objetivos estratégicos de la universidad.
- Coordinador de Sistemas: establece la línea base de cada proyecto y la metodología de desarrollo
- Analista de Medición: establece la recolección de datos de las métricas definidas
- Jefe de la OSIE: convocar a reunión para evaluar y aprobar las métricas propuestas para cada proyecto

Propuesta

Luego de identificar las actividades más representativas del proceso, se realiza una propuesta de las siguientes actividades para mejorar el proceso:

- Jefe de la OSIE: establece el alineamiento de las métricas con los objetivos estratégicos de la universidad. Además puede obtener métricas históricas de un repositorio y aplicarlos en proyectos similares.
- Coordinador de Sistemas: establece la línea base de cada proyecto y la metodología de desarrollo. Puede obtener históricos de las métricas y hacer un seguimiento respecto a la calidad del producto de software
- Analista de Medición: establece la recolección de datos de las métricas definidas. Puede hacer una recolección más fácil con herramientas como Code Metrics Values de Visual Studio 2015
- Jefe de la OSIE: evalúa y aprueba las métricas propuestas para cada proyecto

7.2. ANÁLISIS DEL MODELO DE MEDICIÓN PROPUESTO

El Modelo de Medición propuesto proporciona un marco de trabajo eficiente que permite el proceso de medición de las etapas del desarrollo del ciclo de vida de los sistemas y poder medir a través de métricas de calidad así como almacenarlos para su futuro análisis y retroalimentación en futuros proyectos.

Las metodologías tomadas como PSM y GQM se han obtenido las mejores prácticas que son las más relevantes y ventajosas para el modelo propuesto tales como:

- Los mejores procesos de PSM para poder tener un marco de trabajo desde la planeación hasta la interpretación de los datos

- Las mejores prácticas de GQM para la definición de las métricas de calidad
- Se está basando en herramientas de desarrollo propietario de Microsoft, que es la que mantiene un contrato de Campus Agreement con la universidad
- Los atributos o características del Modelo de Medición propuesto ha sido validado por expertos en el tema pertenecientes a la misma área de la OSIE
- El Modelo de Medición propuesto ha sido validado a través de su aplicación en el área de Desarrollo de la OSIE, la misma que tuvo resultados positivos que garanticen el modelo propuesto es de calidad.

7.3. ANÁLISIS DEL DESPLIEGUE DEL MODELO DE MEDICIÓN

De los resultados que se obtuvieron en el despliegue del Modelo de Medición, se puede observar una reducción de tiempo y costo, así como un incremento en la calidad del producto de software en el área de desarrollo de la OSIE, por lo que tuvo una mejora significativa.

7.4. ANÁLISIS DE LA VALIDACIÓN DEL MODELO DE MEDICIÓN PROPUESTO

Según la tabla de resultados obtenidos de la evaluación de los expertos de los atributos del Modelo de Medición Propuesto, expuestos en el ANEXO 06, se puede apreciar que el Modelo propuesto resulta mucho más ventajoso en cada uno de los atributos:

- a) **Cumplimiento para CMMI Nivel 2:** con un marco de trabajo para la medición y métricas se puede lograr la certificación de CMMI.
- b) **Facilidad de aplicación:** es aplicable en casi todas las actividades del ciclo de vida del desarrollo de software. Se puede aplicar a los procesos como al producto de software.
- c) **Adaptación del modelo:** se adapta a cualquier metodología de desarrollo, por su flexibilidad y sencillez de las fases que contiene el Modelo propuesto. Se exceptúa la fase de Diseño.

Dando un ponderado a cada criterio para un proyecto de este tipo se encontró que los atributos más resaltantes favorable y desfavorablemente son: los atributos denominados Cumplimiento para CMMI Nivel 2, Facilidad de aplicación y Adaptación del modelo.

Atributos	Puntaje Final	Ponderado Acorde al Proyecto	Relevancia Final
Cumplimiento para CMMI Nivel 2	65,73	3	197,19
Facilidad de aplicación	71,10	3	213,30
Adaptación del modelo	82,4	2	164,80

Tabla N° 21: Relevancia Final de cada atributo del Modelo de Medición propuesto

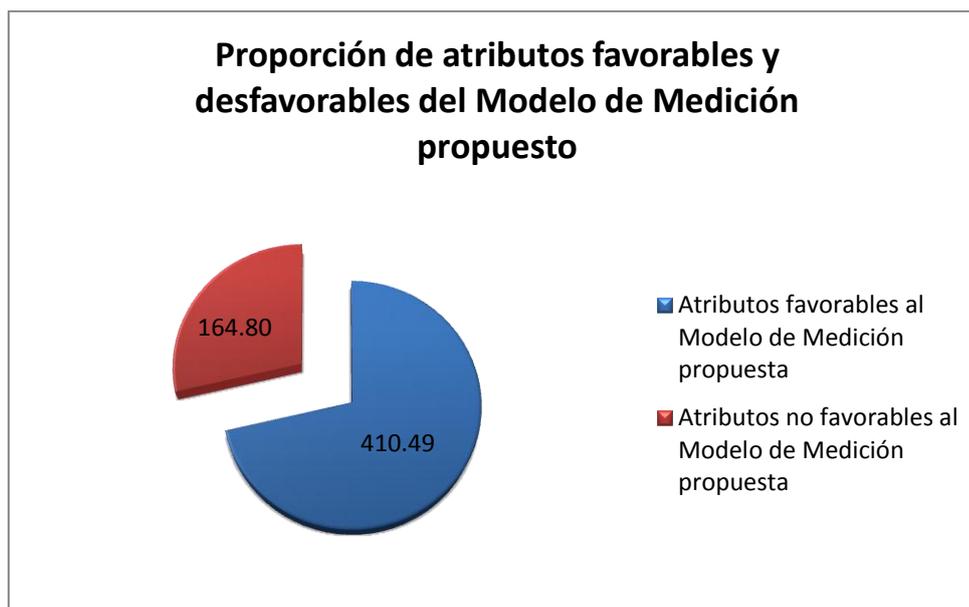


Figura N° 42: Proporción de los atributos favorables y desfavorables del Modelo de Medición Propuesto

ANALISIS DE LOS RESULTADOS OBTENIDOS DE LA EVALUACIÓN DE LOS INDICADORES DE LAS METRICAS CON EL MODELO DE MEDICION PROPUESTO

En la tabla de resultados que muestran los datos obtenidos a través de sus respectivos indicadores de las métricas con la aplicación del modelo de medición propuesto al área de desarrollo de la OSIE, se puede medir a través de la comparación si los resultados indican que el modelo produce una elevación de la calidad, indicando si los resultados antes y después son significativamente diferentes.

Para el presente proyecto no hace falta hacer Test estadísticos muy sofisticados como la PRUEBA DE T-STUDENT, Chi-cuadrado, o la prueba de Fisher, para demostrar que realmente mejora la calidad en el área de desarrollo de la OSIE con la aplicación del Modelo de Medición propuesto, ya que los resultados obtenidos están bastantes claros, en la siguiente tabla y figura se puede apreciar a través de los resultados de la evaluación de los indicadores, que si existe una mejora significativa.

Indicadores	Unidad de Medida	Valor
Índice de Mantenibilidad	Escalar	76.90
Porcentaje de Código Duplicado	Porcentaje	75%
Porcentaje de Pruebas Funcionales	Porcentaje	7%

Tabla N° 22: Valor final de cada indicador de las Métricas con el Modelo de Medición propuesto Proyecto de Recursos Humanos

Indicadores	Unidad de Medida	Valor
Índice de Mantenibilidad	Escalar	75
Porcentaje de Código Duplicado	Porcentaje	61%
Porcentaje de Pruebas Funcionales	Porcentaje	44%

Tabla N° 23: Valor final de cada indicador de las Métricas con el Modelo de Medición propuesto Proyecto de Trámite Documentario

De los resultados obtenidos de las tablas y figuras anteriores se deduce, que los valores mostrados en ambos casos, son significativamente y que nos dan un diagnóstico más concreto de la calidad del producto de software. Lo cual nos indica que el Modelo propuesto es significativamente relevante para OSIE, ya que se puede apreciar claramente una disminución de la calidad del producto interno en cuanto a duplicidad de código fuente y falta de una rigurosa pruebas funcionales y por tanto un incremento en tiempo y costo por mantenimiento y un decremento por tanto en la calidad del software.

Por tanto, aceptaríamos la hipótesis de que “La propuesta de un modelo de medición con métricas para el proceso de desarrollo de software, permitirá mejorar la calidad del producto de software en el Área de Desarrollo de la Oficina de Sistemas e Ingeniería de la Información de la Universidad Privada Antenor Orrego, año 2013”, por

lo que podemos concluir que existe evidencia significativa como para decir, que el Modelo de Medición en el área de desarrollo de la OSIE.

CONCLUSIONES

- Teniendo en cuenta la metodología PSM tiene las siguientes fases: Planificación, Ejecución y Evaluación de la Medición, y la metodología GQM por su parte tiene las siguientes fases: Planeación, Definición, Colección de Datos e Interpretación. GQM se caracteriza por tener la fase de Definición de las métricas muy bien orientado a la práctica y de fácil aplicación a cualquier proceso de desarrollo de software así como al producto, es así como se propuso un modelo de medición con métricas tomando algunas fases de ambas metodologías y éstas son: Definir objetivos y métricas, Elaborar un plan de medición, ejecutar plan de medición e Interpretar y evaluar la medición para el proceso de medición propuesto, el cual demuestra que el marco metodológico de PSM y el GQM son perfectamente complementarios.
- Se definió las métricas apropiadas para la OSIE, teniendo en cuenta que dentro del Modelo de Medición propuesto existe una fase de Definición de Métricas aplicando GQM. Estas métricas de calidad interna del producto de software son: Índice de Mantenibilidad, Porcentaje de Código Duplicado y Porcentaje de Pruebas Funcionales.
- Se diseñó el proceso completo de medición contemplado en el Modelo de Medición propuesto, que incluye: Definir objetivos y métricas, Elaborar un plan de medición, ejecutar plan de medición e Interpretar y evaluar la medición.
- Se estableció dentro del Modelo de Medición propuesto, usar plantillas en Excel inicialmente para registrar las métricas por cada proyecto, de tal manera que quede visibilidad en toda la OSIE para futuras consultas y retro alimentación.
- Se analizaron los datos obtenidos de las métricas de calidad para el proceso de desarrollo de software en la fase de la Codificación, para luego con esta información contrastar la hipótesis, con lo que se demostró que los indicadores nos dan un diagnóstico más preciso del

estado de construcción del producto de software. Para el Proyecto de Recursos Humanos se obtuvo un valor para el índice mantenibilidad de 76.9, para el porcentaje de código duplicado de un 75% y para el porcentaje de pruebas funcionales de 7%. Para el proyecto de Trámite Documentario se obtuvo un valor para el índice de mantenibilidad de 75, para el porcentaje de código duplicado de un 61% y para el porcentaje de pruebas funcionales de un 44%, estos resultados hacen que el modelo de medición con métricas sea relevante en el proceso de medición para el Área de Desarrollo de la OSIE de la Universidad Privada Antenor Orrego.

RECOMENDACIONES

- Para implementar el proceso de medición se recomienda que las siguientes fases: Definir objetivos y métricas, Elaborar un plan de medición, Ejecutar Plan de Medición e Interpretar y Evaluar la Medición, se realicen de manera sostenida, esto permitirá conocer de manera más objetiva las métricas de calidad del producto de software y su impacto que tendrá en la puesta en producción y posterior a ésta.
- Se recomienda que para definir métricas apropiadas se use GQM ya que es una de las mejores prácticas que permiten alinear los objetivos de medición con los objetivos estratégicos de una organización.
- Se recomienda que toda área de desarrollo de software tenga un Modelo de Medición que les permita tener un Plan de Medición para poder tomar medidas del producto de software.
- Se recomienda que cada Proyecto de Desarrollo de Software tenga su Plan de Medición y métricas como parte de la documentación, esto nos servirá como histórico y retro alimenta a futuros proyectos de software.
- Se recomienda para futuros trabajos que el tiempo para aplicar el modelo de medición propuesto a una organización de desarrollo, sea de un estudio como mínimo de 6 meses, en lugar de tomar una sola medida, como se realizó en el presente trabajo de investigación, para que así nos permita hacer una evaluación más constante y precisa, y así obtener una mayor consistencia, validez e interpretación necesarias y tomar acciones antes de la puesta en producción de un producto de software.
- Se recomienda para trabajos futuros usar mediciones en cada uno de los sprint de SCRUM u otra metodología que use cualquier empresa de desarrollo de software. De esta manera se puede ir monitoreando y retroalimentando el proceso de desarrollo y hacer los ajustes antes de la puesta en producción del producto final. En la presente investigación se ha considerado solo una medición del producto final para encontrar

el estado actual de las métricas definidas con el modelo de medición propuesta. Se recomienda por tanto para trabajos futuros aplicarlo de forma más sostenida por un equipo de medición y usar más herramientas como NDEPEND y SONARQUBE para obtener mayor información de métricas del producto.

REFERENCIAS BIBLIOGRÁFICAS

- Association for Computing Machinery (2008). "CS2008 Curriculum Update: The Computing Curricula Computer Science Volume is complete and approved". Recuperado el 25 de Noviembre de 2014, de <http://acm.org/education/curricula/ComputerScience2008.pdf>
- Capers Jones (2008). Applied Software Measurement, Global Analysis of Productivity and Quality. McGraw-Hill; Third Edition.
- Card, D. (2003). Integrating Practical Software Measurement and the Balanced Scorecard Computer Software and Applications Conference, Annual International, IEEE Computer Society, doi: 10.1109/CMPSAC.2003.1245366
- Center for Systems and Software Engineering (2011). COCOMO II. University of Southern. Recuperado el 25 de Noviembre de 2014, de http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html
- Cenqua Pty Ltd. (2007). Clover, Code Coverage. Recuperado el 25 de Noviembre de 2014, de <http://www.cenqua.com/clover/doc/coverage/intro.html>
- Chirstof Ebert, Reiner Dumke (2007). Software Measurement: establish, extract, evaluate, execute. Springer-Verlag Berlin Heidelberg.
- D. Knuth (1998). "The Art of Computer Programming: Volumes 1-3". USA: Addison-Wesley Professional.
- Department of Computer Science (2010). SENG Focus Area: Evolutionary Software Engineering. University of Calgary. Recuperado el 25 de Noviembre de 2014, de http://www.cpsc.ucalgary.ca/cpsc_research/areas/evolutionary
- ISO/IEC 15939:2007 (2007). Software Engineering – Software measurement process.
- Jeff Tian (2005). Software Quality Engineering, Testing, Quality Assurance, and Quantifiable Improvement. IEEE Computer Society. John Wiley & Sons, Inc., Hoboken, New Jersey.
- Juan Palacio, Claudia Ruata (2011). Scrum Manager Gestión de Proyectos. Safe Creative. Revisión 1.4.0-Enero de 2011

- Linda M. Laird, M. Carol Brennan (2006). Software Measurement and Estimation: A practical approach, John Willey & Sons.
- McGarry, Jhon, C.J.B.L.E.C.J.D. & Hall, F. (2002). Practical Software Measurement: Objective Information for Decision Makers, Addison-Wesley.
- Oficina de Sistemas e Ingeniería de la Información (2012). Gestión del Área de Desarrollo de Sistemas UPAO.
- Practical Software Measurement: A Foundation for Objective Project Management, v.4.0b1 (2000). PSM User's Group. Recuperado el 22 de Noviembre de 2014, de <http://www.psmc.com>
- Project Management Institute (2013). Guía de los Fundamentos para la dirección de Proyectos (Guía PMBOK). Quinta Edición.
- Quantitative Software Management, Inc. (2011). SLIM Software Cost Estimation, Planning and Benchmarking Tools. Recuperado el 25 de Noviembre de 2014, de <http://www.qsm.com>
- R. Lytz (2010). On Board Software for the Boeing 777. Recuperado el 25 de Noviembre de 2014, de http://www.stsc.hill.af.mil/resources/tech_docs/gsam3/appenq.pdf
- Roger S. Pressman (2010). Software Engineering, A Practitioner's Approach. McGraw-Hill; Seventh Edition.
- Rini van Solingen, Egon Berghout (1999). The Goal/Question/Metric Method: a practical guide for quality improvement of software development. McGraw-Hill.
- Software Engineering Institute (2010). CMMI para Desarrollo, Versión 1.3, CMMI-DEV, V1.3, Carnegie Mellon University.
- Software Engineering Institute (2011). Standard CMMI Appraisal Method for Process Improvement A, Versión 1.3: Method Definition Document, Carnegie Mellon University.
- T. DeMarco (1982). Controlling Software Projects: Management, Measurement & Estimation, Yourdon Press, New York, USA, 1982.
- Victor R. Basili, D.M. Weiss (1984). A Methodology for Collecting Valid Software Engineering Data. IEEE Transactions on Software Engineering, Noviembre 1984

Victor R. Basili (1992). Software Modeling and Measurement: The Goal Question Metric Paradigm, University of Maryland, College Park, Md., Sept. 1992.

W.T. Ward (1991). Calculating the Real Cost of Software Defects, October 1991, Recuperado el 25 de Noviembre de 2014, de http://findarticles.com/p/articles/mi_m0HPJ/is_n4_v42/ai_11400873

ANEXOS

Anexo N° 01: Entrevista 01

Entrevistado: Genero Pajuelo Ureña	Fecha: 30-Abril-2016
Entrevistador: Armando Caballero Alvarado	Área o Departamento: Sección Remuneraciones
Objetivo: Conocer la situación actual de la universidad, con respecto a la información general de la UPAO.	
Dirigido a: Jefe de Sección de Remuneraciones	
Preguntas: <ol style="list-style-type: none"> 1. ¿Cuál es su función principal dentro del área? 2. ¿Cuánto tiempo de creación tiene la universidad? 3. ¿Cuántas sedes tiene la universidad en el país? 4. ¿Cuáles son las carreras profesionales que más estudiantes tiene? 	
Resumen: La UPAO es un institución que tiene 30 años de creado, cuenta con 23 carreras profesionales. Actualmente cuenta con dos sedes, Piura y Lima (Postgrado). Las carreras que cuentan con más estudiantes son: Medicina Humana, Derecho, Ingeniería Civil, Ingeniería Industrial, Administración y Arquitectura.	

Anexo N° 02: Entrevista 02

Entrevistado:	Jimmy Maco Elera	Fecha:	30-Abril-2016
Entrevistador:	Armando Caballero Alvarado	Área o Departamento:	OSIE
Objetivo:			
Conocer la problemática de la universidad, con respecto al uso de los sistemas de información.			
Dirigido a:			
Coordinador de Sistemas			
Preguntas:			
<ol style="list-style-type: none"> 1. ¿Cuál es su función dentro de la universidad? 2. ¿El área de sistemas cuenta con sus propios desarrolladores? 3. ¿Cuál es la problemática actual con respecto a los sistemas de información que se utilizan? 4. ¿Cuáles son los sistemas de información con los que cuenta actualmente la universidad? 			
Resumen:			
<p>Desde Enero del 2002 se inició la implantación de un ERP, Banner, el cual incluye la gestión Académica y Financiera. El sistema funcionaba de manera correcta sobre todo en la parte académica, pero el constante crecimiento de la universidad y la creciente demanda de funcionalidades nuevas, volvió limitado al sistema en cuanto a los módulos académicos y financieros. Se inició el 2010 el desarrollo de un sub sistema llamado Campus Virtual para la parte académica y en cuanto a lo financiero se dio mantenimientos y adecuaciones de acuerdo a las necesidades de los usuarios. Asimismo se comenzó a crear más aplicaciones tales como: Centros de Producción: Clínica Estomatológica, Librería, Centro de Salud y Panificadora, Biblioteca, Recursos Humanos, entre otros.</p>			

Anexo N° 03: Entrevista 03

Entrevistado:	Marcelino Carretero Obando	Fecha:	30-Abril-2016
Entrevistador:	Armando Caballero Alvarado	Área o Departamento:	OSIE
Objetivo:			
Conocer la problemática de la universidad con respecto a la calidad de los sistemas de información.			
Dirigido a:			
Jefe de la OSIE			
Preguntas:			
<ol style="list-style-type: none"> 1. ¿Cuál es su función dentro de la empresa? 2. ¿De qué manera se realiza la evaluación para la implementación de un nuevo sistema? 3. ¿Cuál es la metodología de desarrollo que usa la OSIE? 4. ¿Se evalúa la calidad dentro de todo el proceso de desarrollo de software? 5. ¿Cuánto tiempo demora en ponerse en producción un nuevo sistema? 6. ¿Cuántos incidentes recibe la OSIE después de una puesta en producción de un nuevo sistema? 			
Resumen:			
<p>La evaluación para implementación de un nuevo sistema se hace en función a la necesidad y a la alineación a las estrategias de la universidad. La OSIE tiene una metodología de desarrollo de software basado en SCRUM, con un manual en el cual incluye los criterios y nomenclaturas estándares para la denominación de objetos y asimismo para la documentación técnica. La evaluación de la calidad solo se hace de manera superficial y en pruebas de caja negra (pruebas funcionales). Los tiempos para la puesta en producción dependen de la complejidad y alcance de cada proyecto de software, que pueden durar en un intervalo de 3 meses a 9 meses. Los incidentes son altos una vez puesto en producción el nuevo sistema, llegando a obtener hasta 10 incidentes por día y luego va bajando en el tiempo.</p>			

Anexo N° 04: Entrevista 04

Entrevistado:	Víctor Jave Sacramento	Fecha:	30-Abril-2016
Entrevistador:	Armando Caballero Alvarado	Área o Departamento:	OSIE
Objetivo:			
Saber la problemática de la universidad con respecto al desarrollo de software			
Dirigido a:			
Desarrollador de sistemas.			
Preguntas:			
<ol style="list-style-type: none"> 1. ¿Cuál es su función principal dentro del área? 2. ¿Qué tipo de pruebas realiza en la fase de codificación de software? 3. ¿Cada cuánto tiempo hace éstas pruebas? 4. ¿Cómo mide que el código que está desarrollando es de calidad? 5. ¿Cuánto tiempo demora en documentar la parte técnica del sistema? 6. ¿Qué herramientas utiliza para hacer sus pruebas de calidad? 			
Resumen:			
<p>Las pruebas que se realiza en la fase de codificación del software, son las pruebas unitarias, pero se ejecuta muy esporádicamente por falta de tiempo y se deja para el final en las pruebas de caja negra (pruebas funcionales). No tenemos un indicador o métrica que no diga que el código que estamos desarrollando es de calidad. Adicionalmente documentamos el diseño del sistema: modelo de datos, y no documentamos más fases debido a la falta de tiempo. Las herramientas que usamos para nuestras pruebas están integradas y las que vienen con el Microsoft Visual Studio.</p>			

Anexo N° 05**ENCUESTA PARA EVALUAR EL MODELO DE METRICAS DE CALIDAD PROPUESTO****PREGUNTAS**

La puntuación puede ser de 0 a 100

- a) **Cumplimiento para el CMMI Nivel 2:** evalúe la fase de Construcción que es donde va dirigido el modelo propuesto en el cumplimiento de la exigencia de CMMI Nivel 2

	Ítem	Valor
1	Evalué la adaptación del modelo en la Actividad de Diseño de Estructura de Datos	
2	Evalué la adaptación del modelo en la Actividad de Implementación de Componentes	
3	Evalué la adaptación del modelo en la Actividad de Diseño de Interfaces	
4	Evalué la adaptación del modelo en la Actividad de Codificación (programación)	
5	Evalué la adaptación del modelo en la Actividad de Validación del producto (pruebas y verificación)	

- b) **Facilidad de aplicación:** evalúe la fase de Construcción del modelo propuesto a la facilidad de aplicación dentro de las actividades que involucran la construcción.

	Ítem	Valor
1	Evalué la adaptación del modelo en la Actividad de Diseño de Estructura de Datos	
2	Evalué la adaptación del modelo en la Actividad de Implementación de Componentes	
3	Evalué la adaptación del modelo en la Actividad de Diseño de Interfaces	
4	Evalué la adaptación del modelo en la Actividad de Codificación (programación)	
5	Evalué la adaptación del modelo en la Actividad de Validación del producto (pruebas y verificación)	

- c) **Adaptación del modelo:** evalúe la fase de Construcción del modelo propuesto en cuanto a la precisión que nos dá las métricas.

	Ítem	Valor
1	Evalué la adaptación del modelo en la Actividad de Diseño de Estructura de Datos	
2	Evalué la adaptación del modelo en la Actividad de Implementación de Componentes	
3	Evalué la adaptación del modelo en la Actividad de Diseño de Interfaces	
4	Evalué la adaptación del modelo en la Actividad de Codificación (programación)	
5	Evalué la adaptación del modelo en la Actividad de Validación del producto (pruebas y verificación)	

ANEXO N° 06

RESULTADOS DE LA ENCUESTA PARA EVALUAR EL MODELO PROPUESTO

RESUMEN DE RESULTADOS DE LAS ENCUESTAS EFECTUADAS						
	ATRIBUTOS DEL MODELO	EXPERTOS				
I	Cumplimiento para el CMMI Nivel 2	E1	E2	E3	E4	PROMEDIO
	1. Diseño de Estructura de Datos	50	55	45	45	48.75
	2. Implementación de Componentes	60	65	63	60	62.00
	3. Diseño de Interfaces	75	70	72	80	74.25
	4. Codificación (programación)	55	60	50	65	57.50
	5. Validación del producto	75	75	73	72	73.75
	PROMEDIO POR EXPERTO:	63	65	60.6	64.4	
	PROMEDIO TOTAL POR EXPERTO:	63,25				
II	Facilidad de aplicación	E1	E2	E3	E4	PROMEDIO
	1. Diseño de Estructura de Datos	82	74	80	85	80.25
	2. Implementación de Componentes	65	50	60	75	62.50
	3. Diseño de Interfaces	60	65	70	55	62.50
	4. Codificación (programación)	65	70	75	70	70.00
	5. Validación del producto	70	80	85	85	80
	PROMEDIO POR EXPERTO:	68.40	67.80	74	74	
	PROMEDIO TOTAL POR EXPERTO:	71,05				
III	Adaptación del Modelo	E1	E2	E3	E4	PROMEDIO
	1. Diseño de Estructura de Datos	75	70	85	74	76.00
	2. Implementación de Componentes	85	90	90	85	87.50
	3. Diseño de Interfaces	80	85	80	70	78.75
	4. Codificación (programación)	75	70	80	85	77.50
	5. Validación del producto	90	95	93	90	92.00
	PROMEDIO POR EXPERTO:	81	82	85.6	80.8	
	PROMEDIO TOTAL POR EXPERTO:	82,35				

ANEXO N° 07: Ficha de Entrevista

Entrevistado:	Fecha:
Entrevistador:	Área o Departamento:
<p>Objetivo General: Conocer cuánto tiempo se invierte en el desarrollo de un sistema de información.</p> <p>Objetivos específicos:</p> <ul style="list-style-type: none"> ➤ Identificar al personal implicado en el desarrollo ➤ Conocer el tiempo invertido en el análisis de requisitos de un sistema de información. ➤ Conocer el tiempo de demora en la fase de construcción del sistema de información ➤ Conocer el tiempo de demora en las pruebas de calidad. 	
<p>Dirigido a:</p> <p>Personal involucrado en el desarrollo de sistemas</p>	
<p>Preguntas:</p> <ol style="list-style-type: none"> 1. ¿Cuántas personas se involucran en el análisis de requerimientos? 2. ¿Cuánto tiempo se tardan en el análisis de requerimientos? 3. ¿Cuánto tiempo se invierte en priorizar los requerimientos? 4. ¿Cuántas reuniones se realizan mensualmente? 5. ¿Cuánto tiempo se demora en construir cada requerimiento? 6. ¿Cuánto tiempo demora en pruebas de calidad de cada requerimiento? 7. ¿Cuánto tiempo demora en pruebas integrales de calidad? 8. Una vez terminado la construcción del sistema ¿Cuánto tiempo tarda en lanzarla a producción? 	
<p>¿Qué se logró?</p>	
<p>Conclusiones:</p>	

ANEXO N° 09: Ficha de Análisis Documental**Datos de la Aplicación****a) Nombre de la Institución:****b) Fechas:**

Fecha de aplicación 1: __/__/__

Fecha de aplicación 2: __/__/__

Fecha de aplicación 3: __/__/__

Fecha de aplicación 4: __/__/__

c) Miembros que participaron:**d) Documentos de la organización revisados:**

Documentos	Tiene		Se revisó	
	Si	No	Si	No
Plan Estratégico				
Iniciativas Estratégicas				
Manual de Desarrollo				
..				

ANEXO 10

Plan de Medicion

1. **Objetivos de Medicion:** detalles y estado de las metas de la medicion asociados a la universidad en terminos de logros, progreso y calidad.
2. **Metricas por indicadores:** se enumeran las metricas que seran reevaluadas cada cierto tiempo durante el desarrollo para asi dar soporte a los objetivos planeados.
3. **Procedimiento de Recoleccion y Almacenamiento de Datos:** definicion de los procedimientos para recolectar datos, para asegurar que estan siendo obtenidos de manera correcta. Tambien se definen los procedimientos de almacenamiento de los datos y cómo éstos prodrían ser recuperados para asegurar que esten disponibles y accesibles para su uso posterior.
4. **Identificacion de actores relevantes:** definicion de los roles y actores relevantes que participarán en el proceso de medición.