



UNAP



FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
MAESTRÍA EN INGENIERÍA DE SISTEMAS CON MENCIÓN
EN GERENCIA DE LA INFORMACIÓN Y GESTIÓN DE
SOFTWARE

TESIS

DISEÑO DE UNA ARQUITECTURA PARA EL SISTEMA DE
GESTIÓN ACADÉMICA EN LA UNIVERSIDAD NACIONAL
DE LA AMAZONÍA PERUANA, 2018

PARA OPTAR EL GRADO ACADÉMICO DE MAESTRO EN
INGENIERÍA DE SISTEMAS CON MENCIÓN EN GERENCIA
DE LA INFORMACIÓN Y GESTIÓN DE SOFTWARE

PRESENTADO POR: ROBERTO MARTÍN TUESTA PEREYRA

ASESOR:

ING. AGRÓN. OSCAR RICARDO CASTILLO CORTÉS, MGR.

IQUITOS, PERÚ

2021

ACTA DE SUSTENTACIÓN



UNAP

Escuela de Postgrado "JOSÉ TORRES VÁSQUEZ"
Oficina de Asuntos Académicos



ACTA DE SUSTENTACIÓN DE TESIS 009-2020-OAA-EPG-UNAP

Con Resolución Directoral N° 0152-2020-EPG-UNAP, se autoriza la sustentación de la tesis: "DISEÑO DE UNA ARQUITECTURA PARA EL SISTEMA DE GESTIÓN ACADÉMICA EN LA UNIVERSIDAD NACIONAL DE LA AMAZONÍA PERUANA, 2018", teniendo como jurados a los siguientes profesionales:

| | |
|--|------------|
| Ing. Ind. Saul Flores Nunta, Dr. | Presidente |
| Ing. Tony Raúl Vásquez Vásquez, Mgr. | Miembro |
| Lic. Educ. Manuel Tuesta Moreno, Mgr. | Miembro |
| Ing. Oscar Ricardo Castillo Cortes, Mgr. | Asesor |

A los quince días del mes de febrero del 2020, a horas 10:30 a.m., en el Auditorio de la Escuela de Postgrado de la Universidad Nacional de la Amazonía Peruana, se constituyó el Jurado Evaluador y dictaminador, para presenciar y evaluar la sustentación de la tesis: "DISEÑO DE UNA ARQUITECTURA PARA EL SISTEMA DE GESTIÓN ACADÉMICA EN LA UNIVERSIDAD NACIONAL DE LA AMAZONÍA PERUANA, 2018" presentado por el señor ROBERTO MARTIN TUESTA PEREYRA, como requisito para obtener el **Grado Académico de Maestro en Ingeniería de Sistemas con mención en Gerencia de la Información y Gestión de Software**, que otorga la UNAP de acuerdo a la Ley Universitaria N° 30220 y el Estatuto de la Universidad Nacional de la Amazonía Peruana.

Después de haber escuchado la sustentación y luego de formuladas las preguntas, éstas fueron:

Abiertas satisfactoriamente

El Jurado, después de la deliberación correspondiente en privado, llegó a las siguientes conclusiones, la sustentación es:

1. Aprobado como: a) Excelente () b) Muy bueno () c) Bueno (X)

2. Desaprobado: ()

Observaciones: *Ninguna*

A Continuación, el Presidente del Jurado, da por concluida la sustentación, siendo las *11:50* a.m. del día quince de febrero del 2020; con lo cual, se le declara al sustentante *Apto* para recibir el **Grado Académico de Maestro en Ingeniería de Sistemas con mención en Gerencia de la Información y Gestión de Software**.

Ing. Ind. Saul Flores Nunta, Dr.
Presidente

Ing. Tony Raúl Vásquez Vásquez, Mgr.
Miembro

Lic. Educ. Manuel Tuesta Moreno, Mgr.
Miembro

Ing. Oscar Ricardo Castillo Cortes, Mgr.
Asesor

TESIS APROBADA EN SUSTENTACIÓN PÚBLICA EL DÍA 15 DE FEBRERO DEL 2020, EN EL AUDITORIO DE LA ESCUELA DE POSTGRADO DE LA UNIVERSIDAD NACIONAL DE LA AMAZONÍA PERUANA, EN LA CIUDAD DE IQUITOS-PERÚ.

Saul Flores

ING. IND. SAUL FLORES NUNTA, DR.

PRESIDENTE



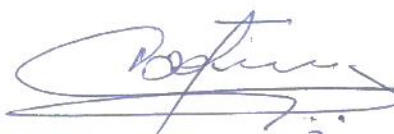
ING. SIST. TONY RAÚL VÁSQUEZ VÁSQUEZ, MGR.

MIEMBRO



LIC. EDUC. MANUEL TUESTA MORENO, MGR.

MIEMBRO



ING. AGRÓN. OSCAR RICARDO CASTILLO CORTÉS, MGR.

ASESOR

Esta Tesis se la dedico a:

A mis padres:

Nimia Andrea Pereira Cubas y Martín Tuesta Gómez, por haberme dado una educación plena y motivarme para seguir adelante para alcanzar mis metas.

A mi esposa:

Cynthia Lizeth Del Águila Guzmán por estar conmigo apoyándome en los momentos difíciles.

A mis hijos:

Tracy Nicole y Rodrigo Nicolás Tuesta Del Águila, por brindarme todo el amor siempre que estoy a su lado y gracias a ellos poder sentir la felicidad más grande que una persona pueda tener, son mi motivación para esforzarme cada día.

A Dios:

Por brindarme la vida, la salud y la perseverancia para continuar esforzándome en alcanzar este logro que es un paso más en mi carrera profesional.

AGRADECIMIENTOS

Expreso mi agradecimiento para la Universidad Nacional de la Amazonía Peruana, a mis docentes y compañeros de la maestría, pues ya que todos de alguna manera me apoyaron y aportaron para poder ampliar y profundizar mis conocimientos y convicciones profesionales.

De manera especial, reitero mi gratitud al:

- Ing. Saúl Flores Nunta, por su apoyo incondicional en este proceso importante para mi persona, en la elaboración de esta investigación, aportando a este proyecto con sus sabios conocimientos y su constante motivación para seguir adelante.
- Ing. Tony Raúl Vásquez Vásquez, por compartir su conocimiento y hacerme ver mis errores realizados en el desarrollo de este proyecto, el cual me sirvió para la culminación satisfactoria de mi trabajo.
- Lic. Manuel Tuesta Moreno, por apoyarme cuando lo necesité y compartir su experiencia en el desarrollo de investigaciones, el cual se ve reflejado en la culminación satisfactoria de mi trabajo.

Concluyo dando mi más sincero agradecimiento a todos los colaboradores de la Universidad Nacional de la Amazonía Peruana, ya que sin su aporte y tiempo no podría haber culminado exitosamente esta tesis.

ÍNDICE DE CONTENIDO

| | |
|---|------------|
| CARÁTULA | |
| CONTRACARÁTULA | i |
| ACTA DE SUSTENTACIÓN | ii |
| JURADO Y ASESOR | iii |
| DEDICATORIA | iv |
| AGRADECIMIENTOS | v |
| ÍNDICE DE CONTENIDO | vi |
| ÍNDICE DE TABLAS | ix |
| ÍNDICE DE GRÁFICOS | x |
| RESUMEN | xi |
| ABSTRACT | xii |
| INTRODUCCIÓN | 1 |
| CAPÍTULO I: MARCO TEÓRICO | 4 |
| 1.1. Antecedentes | 4 |
| 1.2. Bases Teóricas | 6 |
| 1.3. Definición de Términos Básicos | 17 |
| CAPÍTULO II: VARIABLES E HIPÓTESIS | 21 |
| 2.1. Variables y su Operacionalización | 21 |
| 2.1.1. Variable Independiente (X) | 21 |
| 2.1.2. Variable Dependiente (Y) | 21 |
| 2.1.3. Indicadores e Índices | 21 |

| | |
|--|-----------|
| 2.1.4. Operacionalización de las Variables | 23 |
| 2.2. Formulación de la Hipótesis | 24 |
| CAPÍTULO III: METODOLOGÍA | 25 |
| 3.1. Tipo y Diseño de la Investigación | 25 |
| 3.2. Población y Muestra | 25 |
| 3.3. Técnicas e Instrumentos | 26 |
| 3.4. Procedimientos de Recolección de Datos | 26 |
| 3.5. Técnicas de Procesamiento y Análisis de los Datos | 26 |
| 3.6. Aspectos Éticos | 27 |
| CAPÍTULO IV: RESULTADOS | 28 |
| 4.1. Análisis Descriptivo por Indicadores de la Variable Independiente antes de la aplicación del diseño propuesto. | 28 |
| 4.2. Análisis Descriptivo por Dimensiones de la Variable Independiente antes de la aplicación del diseño propuesto. | 30 |
| 4.3. Análisis Descriptivo por Indicadores de la Variable Dependiente antes de la aplicación del diseño propuesto. | 31 |
| 4.4. Análisis Descriptivo por Dimensiones de la Variable Dependiente antes de la aplicación del diseño propuesto. | 32 |
| 4.5. Análisis Descriptivo por Indicadores de la Variable Independiente después de la aplicación del diseño propuesto. | 33 |
| 4.6. Análisis Descriptivo por Dimensiones de la Variable Independiente después de la aplicación del diseño propuesto. | 35 |
| 4.7. Análisis Descriptivo por Indicadores de la Variable Dependiente después de la aplicación del diseño propuesto. | 36 |
| 4.8. Análisis Descriptivo por Dimensiones de la Variable Dependiente después de la aplicación del diseño propuesto. | 37 |

| | |
|--|-----------|
| CAPÍTULO V: DISCUSIÓN DE LOS RESULTADOS | 38 |
| CAPÍTULO VI: PROPUESTA | 40 |
| 6.1. Introducción: TOGAF como Marco Arquitectural de Referencia | 40 |
| 6.2. Descripción de las Fases de la Arquitectura Propuesta | 42 |
| CAPÍTULO VII: CONCLUSIONES | 69 |
| CAPÍTULO VIII: RECOMENDACIONES | 70 |
| CAPÍTULO IX: REFERENCIAS BIBLIOGRÁFICAS | 71 |
| ANEXOS | 73 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1. Dimensión: Nivel de Operatividad del Sistema – Pre Test | 28 |
| Tabla 2. Dimensión: Nivel de Actualización del Sistema – Pre Test | 29 |
| Tabla 3. Variable Independiente: Arquitectura de Software – Pre Test | 30 |
| Tabla 4. Dimensión: Nivel de Satisfacción del sistema – Pre Test | 31 |
| Tabla 5. Dimensión: Nivel de Eficiencia del sistema – Pre Test | 31 |
| Tabla 6. Variable Dependiente: Sist. de Gestión Académica – Pre Test | 32 |
| Tabla 7. Dimensión: Nivel de Operatividad del sistema – Post Test | 33 |
| Tabla 8. Dimensión: Nivel de Actualización del sistema – Post Test | 34 |
| Tabla 9. Variable Independiente: Arquitectura de Software – Post Test | 35 |
| Tabla 10. Dimensión: Nivel de Satisfacción del sistema – Post Test | 36 |
| Tabla 11. Dimensión: Nivel de Eficiencia del Sistema – Post Test | 36 |
| Tabla 12. Variable Dependiente: Sist. de Gestión Académica – Post Test | 37 |

ÍNDICE DE GRÁFICOS

| | |
|---|----|
| Gráfico 1. Diseño de la Arquitectura Propuesta | 44 |
| Gráfico 2. Visión de la Arquitectura | 45 |
| Gráfico 3. Galería de Actores | 47 |
| Gráfico 4. Ejemplo de asignación de competencias | 48 |
| Gráfico 5. Ejemplo de Vista Inicial | 50 |
| Gráfico 6. Galería de símbolos para casos de uso UML | 51 |
| Gráfico 7. Ejemplo de Casos de Uso | 52 |
| Gráfico 8. Vistas de la Arquitectura de Negocio | 53 |
| Gráfico 9. Símbolos específicos de la vista para organigramas | 54 |
| Gráfico 10. Símbolos específicos para la vista para procesos | 56 |
| Gráfico 11. Ejemplo de vista de procesos | 56 |
| Gráfico 12. Vistas de la Arquitectura de Datos | 60 |
| Gráfico 13. Galería de símbolos para la vista conceptual de datos | 61 |
| Gráfico 14. Ejemplo de modelado conceptual de datos | 61 |
| Gráfico 15. Vistas de la Arquitectura de Aplicaciones | 64 |
| Gráfico 16. Galería de símbolos para la vista de ingeniería de software | 65 |
| Gráfico 17. Ejemplo de vista de ingeniería de software | 65 |
| Gráfico 18. Vistas de Seguridad | 66 |

RESUMEN

El objetivo de esta tesis fue: Diseñar una arquitectura de referencia basada en el marco de trabajo TOGAF para el Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana. Esta investigación contaba con dos variables: La Variable Independiente (X): Arquitectura de Software y la Variable Dependiente (Y): Sistema de Gestión Académica. La investigación fue de tipo Cualitativo, Descriptivo y Explicativo, perteneciente al diseño no experimental, mientras que el diseño específico fue Transeccional Descriptivo.

La población estuvo conformada por los 06 trabajadores de la Universidad Nacional de la Amazonía Peruana (UNAP). La muestra estuvo conformada por el total de la población, es decir 6 personas, La técnica que se empleó en la recolección de los datos fue la encuesta y el instrumento fue el cuestionario.

Los resultados fueron: Del promedio (\bar{x}) de 6 (100%) encuestados, 4.2 (69.44%) encuestados manifestaron que se CUMPLE LA MEJORA A TRAVES DEL SISTEMA DE GESTIÓN ACADÉMICA, 1.6 (26.39%) encuestados manifestaron que se CUMPLE PARCIALMENTE LA MEJORA A TRAVES DEL SISTEMA DE GESTIÓN ACADÉMICA, mientras que 0.3 (4.17%) manifestó que NO SE CUMPLE LA MEJORA A TRAVES DEL SISTEMA DE GESTIÓN ACADÉMICA. Para determinar el impacto en que el Diseño de una Arquitectura de Software afecta positivamente se ha considerado ciertos atributos de calidad(indicadores) para la evaluación tanto de la Arquitectura propuesta (V.I.), como del Sistema de Gestión Académica(V.D.), demostrando de esta manera la hipótesis de la investigación que: Una arquitectura de referencia basada en el marco de trabajo TOGAF, permitirá mejorar el Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana.

Palabras Claves: Arquitectura, Software Web, Gestión Académica y UNAP.

ABSTRACT

The objective of this thesis was: Design a reference architecture based on the TOGAF framework for the Academic Management System of the National University of the Peruvian Amazon. This research had two variables: The Independent Variable (X): Software Architecture and the Dependent Variable (Y): Academic Management System. The research was of the Qualitative, Descriptive and Explanatory type, belonging to the non-experimental design, while the specific design was Descriptive Transectional. The population was made up of the 06 workers of the National University of the Peruvian Amazon (UNAP). The sample consisted of the total population, that is, 6 people. The technique used to collect the data was the survey and the instrument was the questionnaire. The results were: Of the average (\bar{x}) of 6 (100%) respondents, 4.2 (69.44%) respondents stated that THE IMPROVEMENT IS FULFILLED THROUGH THE ACADEMIC MANAGEMENT SYSTEM, 1.6 (26.39%) respondents stated that THE IMPROVEMENT IS PARTIALLY FULFILLED THROUGH THE ACADEMIC MANAGEMENT SYSTEM, while 0.3 (4.17%) stated that THE IMPROVEMENT IS NOT FULFILLED THROUGH THE ACADEMIC MANAGEMENT SYSTEM. To determine the impact on which the Design of a Software Architecture positively affects certain quality attributes (indicators) have been considered for the evaluation of both the proposed Architecture (VI), and the Academic Management System (VD), demonstrating this In this way, the research hypothesis that: A reference architecture based on the TOGAF framework will improve the Academic Management System of the National University of the Peruvian Amazon.

Keywords: Architecture, Web Software, Academic Management and UNAP.

INTRODUCCIÓN

La Universidad Nacional de la Amazonía Peruana es una entidad pública que brinda estudios superiores y ha sido creada el 14 de enero de 1961, ya con casi 49 años hoy en día cuenta con 14 facultades donde se evidencia que la gestión académica es deficiente, debido al retardo para atender diferentes necesidades y/o servicios que requiere la población estudiantil, como matrícula, solicitud de aplazados, consulta de horarios y notas, etc. Así como el desconocimiento del estado de su trámite documentario, como solicitud de certificado de estudios, grados y títulos académicos, etc. Además, actualmente no se cuenta con indicadores que permitan conocer en qué, donde y en que magnitud se está fallando, ya que no existen datos o son inconsistentes a nivel de las distintas dependencias, debido a que no existe integración de la mayoría de los procesos relacionados con la gestión académica.

En la actualidad la Universidad cuenta con un Software de Gestión Académica en un entorno web, que sistematiza algunos de los procesos involucrados y de acuerdo a las circunstancias se adecua el Software para dicho propósito o se altera los datos en bruto ya que el software no cumple con ciertos requisitos funcionales, lo que genera inconsistencia y pérdida de información para la toma de decisiones. Todo esto debido a que no se identificaron claramente los requisitos funcionales y no funcionales, no teniendo en consideración atributos de calidad para un Sistema de Gestión Académica de manera integral y en un entorno web, ya que la Universidad Nacional de la Amazonía Peruana cuenta con dependencias e infraestructura descentralizada.

EL problema se agrava, al darnos cuenta que se conoce muy poco acerca de representaciones formales para la descripción de arquitecturas de aplicaciones web. La arquitectura para aplicaciones web propone la descripción formal de todas las partes que integran la arquitectura de estos sistemas, como son su estructura, componentes, conectores e interfaces

haciendo uso de las herramientas existentes. Esto permitirá obtener arquitecturas que cumplan, de una mejor forma, con los requerimientos del sistema, dado que estos requerimientos se verán reflejados en la descripción de la arquitectura.

Es por ello que se ha considerado como problema general responder a la interrogante: ¿Cómo una arquitectura de referencia basada en TOGAF, mejoraría el Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana? A través de este proyecto se describirá una arquitectura de software de referencia para aplicaciones Web. En la actualidad no existen modelos de arquitectura de software que permitan representar la estructura y todas las características de distintas aplicaciones de software, en particular aplicaciones basadas en Web. El estudio de la arquitectura de software se centra en la forma en cómo son diseñados y construidos los sistemas de software. La arquitectura de un sistema de software define la estructura del sistema y está integrada de todos los componentes que involucran al sistema, así como sus interconexiones. La arquitectura de software es de especial importancia ya que la manera en que se estructura un sistema tiene un impacto directo sobre la capacidad de éste para satisfacer lo que se conoce como los atributos de calidad del sistema, como el desempeño, la usabilidad y la modificabilidad, de esta manera se garantiza cumplir con los requerimientos no funcionales. No olvidar que la arquitectura de software no puede ser pensada sin el contexto de los stakeholders. Esto puede parecer obvio, pero muchos sistemas por allí se realizan sin la debida atención a las necesidades de las personas de interés. La arquitectura debe ser creada sólo para satisfacer las necesidades de los stakeholders y de esta manera alcanzar los objetivos del negocio, para ello se debe identificar los requerimientos funcionales de los usuarios a través de reuniones y entrevistas con los stakeholders, así como los requerimientos no funcionales con el personal técnico de la Universidad.

El Objetivo General es “Diseñar una arquitectura de referencia basada en el marco de trabajo TOGAF para el Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana”. Y los Objetivos Específicos

son: “Crear una arquitectura de referencia, generando los componentes necesarios que serán utilizados para la descripción arquitectural”, “Evaluar la Arquitectura de Software antes y después del diseño propuesto” y “Evaluar el Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana antes y después de aplicar el diseño propuesto”.

CAPÍTULO I: MARCO TEÓRICO

1.1. Antecedentes

A medida que aumenta la complejidad de los sistemas de software surgen nuevos aspectos del desarrollo de aplicaciones que hasta ese momento no se habían tenido en cuenta, al menos de una forma explícita. De este modo, el proceso de desarrollo se ha ido convirtiendo gradualmente en una labor de ingeniería, en la que nuevas tareas, como la elaboración de especificaciones, el diseño del sistema, la construcción de prototipos, la integración y pruebas, la gestión de la configuración y otras muchas han ido cobrando importancia frente a las labores de programación que eran las que primaban en un inicio. La Ingeniería del Software ha ido respondiendo a esta situación con el desarrollo de nuevos modelos, notaciones, técnicas y métodos.

CANAL V. en su tesis "Un Lenguaje para la Especificación y Validación de Arquitecturas de Software" para optar el grado de Doctor Ingeniero en Informática, afirmó: Dentro de esta tendencia se encuadra el creciente interés por los aspectos arquitectónicos del software del que estamos siendo testigos en los últimos tiempos. Estos aspectos se refieren a todo lo relativo a la estructura de alto nivel de los sistemas: su organización en subsistemas y la relación entre estos; la construcción de aplicaciones vista como una actividad fundamentalmente composicional en la que se reutilizan elementos creados posiblemente por terceros; el desarrollo de familias de productos caracterizadas por presentar una arquitectura común; el mantenimiento y la evolución entendidos como sustitución de unos componentes por otros dentro de un marco arquitectónico, etc. En efecto, un aspecto crítico a la hora de desarrollar sistemas de software complejos es el diseño de su arquitectura, representada como un conjunto de elementos computacionales y de datos interrelacionados de un modo determinado.

COHALLEN J. en su tesis ".Modeling web application architectures with UML", propuso: metodologías que pretenden mejorar el desarrollo de

aplicaciones como lo exige la Web, sin embargo estas sólo se han preocupado de los aspectos funcionales, de la navegación y representación de la aplicación pero han omitido los aspectos de calidad o no funcionales y aspectos que permitan describir formalmente la arquitectura de dichas aplicaciones.

TAHUITON J. en su tesis "Arquitectura de software para aplicaciones Web" para optar al grado de Maestro en Ciencias de la Computación, afirmó: Se presenta una arquitectura de software para aplicaciones Web en donde se sigue un proceso de ingeniería de software. En este desarrollo, la arquitectura se descompone mediante distintas vistas o enfoques tales como, la vista lógica, la vista de procesos, la vista de desarrollo, la vista física y la vista de seguridad. Cada vista, en esta tesis, se desarrolla mediante el lenguaje de modelado unificado UML.

LIMÓN R. en su tesis "Las vistas arquitectónicas de software y sus correspondencias mediante la gestión de modelos" para optar al grado de Doctor en Informática, indica: Conforme los sistemas de software han sido más complejos la necesidad de anticiparse a la especificación detallada del sistema y de tener un mayor aprovechamiento de los recursos se hace más evidente, estos aspectos entre otros son capturados por la arquitectura de software. El análisis de las estructuras (vistas) que aquí se hace contribuye a mejorar esas decisiones tempranas del diseño, lo cual se materializa en la gestión de los modelos de las estructuras arquitectónicas.

La UNAP dentro de sus Proyectos de Inversión, elaboró un estudio a nivel de perfil: "Mejoramiento Informático en Gestión Académica e Investigación en la Universidad Nacional de la Amazonía Peruana, distrito de Iquitos, provincia de Maynas, región Loreto", da a conocer las deficiencias del actual Sistema de Gestión Académica indicando que: el Software de gestión académica ofrece prestaciones limitadas y en muchos casos deficientes lo que dificulta de sobremanera la gestión académica de la diferentes áreas usuarias, docentes, alumnos y aun mas, dificulta el cumplimiento de las metas institucionales con miras a un nivel de servicio académico aceptable y

acorde a las exigencias del mercado. El software de gestión académica de la UNAP fue concebido para atender tres principales grupos de procesos: Proceso de Admisión, Matrícula, Oferta Académica y Registro Académico.

1.2. Bases Teóricas

1.2.1. Arquitectura de Software

Bass (1998) afirma: “Sin pretender con ello establecer una definición completa ni definitiva, podemos considerar como arquitectura la estructura de alto nivel de un sistema de software, lo que incluye sus componentes, las propiedades observables de dichos componentes y las relaciones que se establecen entre ellos”. Esta definición se centra en aspectos puramente descriptivos, y determina que cualquier sistema de software, o al menos cualquiera que tenga una cierta complejidad, tiene una arquitectura, independientemente de si esta arquitectura está representada en algún lugar de forma explícita, o incluso de si quienes desarrollaron el sistema eran conscientes de ella. Otras definiciones incluyen también aspectos de proceso y, de este modo, Garlan & Perry (1995) afirman: “se debe añadir a la arquitectura de un sistema de software los principios y reglas que gobiernan su diseño y su evolución en el tiempo”.

“Teniendo en cuenta todos estos aspectos, e incorporando aún algunos otros, en la ya citada notación UML se define la arquitectura como el conjunto de decisiones significativas acerca de la organización de un sistema de software; la selección de los elementos estructurales a partir de los cuales se compondrá el sistema y sus interfaces, junto con la descripción del comportamiento de dichas interfaces en las colaboraciones que se producen entre los elementos del sistema; la composición de esos elementos estructurales y de comportamiento para formar subsistemas de tamaño cada vez mayor; y el estilo o patrón arquitectónico que guía esta organización: los elementos y sus interfaces, las colaboraciones y su composición”. (Rumbaugh, et Al., 1999)

Aparejado al concepto de arquitectura, surge la Arquitectura del Software, como la disciplina, inscrita dentro de la Ingeniería del Software, que se ocupa del estudio de la arquitectura de los sistemas de software, y también como una de las tareas del proceso de desarrollo, enmarcada dentro de las actividades propias del diseño. Así, de nuevo Garlan & Perry (1999) definen a: "La Arquitectura del Software, entendida como disciplina, como el nivel del diseño de software donde se definen la estructura y propiedades globales del sistema". "Según esto a Arquitectura del Software se centra en aquellos aspectos del diseño y desarrollo que no pueden tratarse de forma adecuada dentro de los módulos que forman el sistema". (Shaw & Garlan, 1996)

1.2.2. Necesidad de la Arquitectura

Diferentes aproximaciones de Ingeniería Web como OOH [10], JWE [11], WebML [12], OOHDM [13], WSDM [14] han sido probadas con éxito para la especificación de aspectos funcionales y navegacionales de una aplicación Web, es decir, expresan como una aplicación Web almacena, recupera y presenta la información a los usuarios. Sin embargo, la propia idiosincrasia de las aplicaciones Web requiere que se especifiquen no solamente aspectos funcionales, sino que también se establezcan restricciones sobre aspectos no funcionales como la distribución de los componentes, la escalabilidad del sistema, el mantenimiento, la conectividad con sistemas legados, etc. Para poder recoger estos aspectos, autores como Bachmann et al. [15] proponen el uso de técnicas bien conocidas de la disciplina de Arquitectura del Software para identificar y formalizar los subsistemas, conectores y componentes que deben constituir la aplicación. Sin embargo, por parte de los métodos de Ingeniería Web la definición de los aspectos de arquitectura ha sido ignorada o pospuesta hasta la implementación con la siguiente desventaja de no poder establecer de forma explícita cuáles son los componentes adecuados de la arquitectura de una forma independiente de la plataforma destino o poder especificar los artefactos de arquitectura en función de requisitos no funcionales. Existe por tanto un gap o salto entre las especificaciones funcionales representadas por los actuales métodos de desarrollo de Ingeniería Web y la implementación final obtenida. [16].

En conclusión, para asegurar la viabilidad de los métodos para el desarrollo de las aplicaciones Web, las diferentes propuestas deben incorporar de forma explícita los aspectos de arquitectura del software dentro de su proceso de desarrollo.

1.2.3. Descripción de la arquitectura

La Arquitectura del Software tiene como objetivo el conocimiento, análisis y reutilización de la arquitectura de los sistemas de software. Para ello, es necesario hacer explícita dicha arquitectura, lo que implica el uso de algún lenguaje. Podemos considerar diversas opciones a la hora de elegir una notación para representar la arquitectura de un sistema de software: notaciones diagramáticas o textuales, específicas o utilizadas para describir otros aspectos del software, formales o informales. (Canal D., 2000)

El Lenguaje Unificado de Modelado (UML) es una notación para especificar, visualizar y documentar sistemas de software desde la perspectiva de la orientación a objetos. Su objetivo es representar el conocimiento acerca de los sistemas que se pretenden construir y las decisiones tomadas durante su desarrollo, tanto lo referido a su estructura estática como a su comportamiento dinámico. UML postula un proceso de desarrollo iterativo, incremental, guiado por los casos de uso y centrado en la arquitectura (Booch et al., 1999).

La representación en UML de un sistema de software consta de cinco vistas o modelos parciales separados, aunque relacionados entre sí, denominadas vista de casos de uso, de diseño, de implementación, de procesos y de despliegue. Cada uno de estos modelos representa el sistema por medio de diversos diagramas. (Kruchten, 1995). Aunque no existe de forma explícita una vista arquitectónica, estas cinco vistas pretenden describir, en su conjunto, la arquitectura del sistema.

1.2.4. Buenas Prácticas para diseñar Arquitecturas Empresariales

TOGAF

Es un marco de referencia de arquitectura. En términos simples, TOGAF es una herramienta para asistir en la aceptación, creación, uso, y mantenimiento de arquitecturas. Está basado en un modelo interactivo de procesos apoyado por las mejoras prácticas y un conjunto reutilizable de archivos arquitectónicos existentes.

(Andrew Josey. TOGAF versión 9.1. Guía de Bolsillo. The Open Group, Pag 21)

1.2.5. Evaluación de una Arquitectura [20]

¿Por qué evaluar una Arquitectura?

Cuanto más temprano se encuentre un problema en un proyecto de software, mejor. El costo de arreglar un error durante las fases de requerimientos o diseño, es mucho menor al costo de arreglar ese mismo error en la fase de verificación. Dado que la arquitectura es un producto temprano de la fase de diseño, esta tiene un profundo efecto en el sistema y en el proyecto.

Una mala arquitectura puede llevar a un proyecto al fracaso. Todos los requerimientos de calidad pueden quedar insatisfechos.

La arquitectura también determina la estructura del proyecto: configuración, agenda y presupuesto, alcance, entre otros aspectos. Es mejor cambiar la arquitectura antes que otros artefactos, que están basados en ella, se establezcan.

Realizar una evaluación de la arquitectura es la manera más económica de evitar desastres.

¿Cuándo una Arquitectura puede ser evaluada?

Generalmente, la evaluación de la arquitectura ocurre después que esta ha sido especificada, pero antes que empiece la implementación. En un proceso iterativo y/o incremental, la evaluación se puede realizar al final de

cada ciclo. Sin embargo, uno de los atractivos de la evaluación de arquitecturas es que se puede efectuar en cualquier etapa de la vida de una arquitectura. En particular, existen dos variaciones útiles: temprana y tardía.

Evaluación temprana: La evaluación no tiene por qué esperar a que la arquitectura este totalmente especificada. Esta puede ser utilizada en cualquier etapa del proceso de creación de la arquitectura, para examinar las decisiones arquitectónicas ya tomadas y decidir entre las opciones que están pendientes. Por supuesto, la completitud y fidelidad de la evaluación es directamente proporcional a la completitud y fidelidad de la descripción de la arquitectura.

Evaluación tardía: Esta variación toma lugar no solo cuando la arquitectura está terminada, también cuando la implementación está completa. Este caso ocurre cuando la organización hereda un sistema legado. La técnica para evaluar un sistema legado es la misma que para evaluar un sistema recién nacido. Una evaluación es útil para entender el sistema legado, y saber si este cumple con los requerimientos de calidad y comportamiento.

En general, una evaluación debe realizarse cuando hay suficiente de la arquitectura como para justificarlo. Una buena regla sería: realizar una evaluación cuando el equipo de desarrollo empieza a tomar decisiones que dependen de la arquitectura y el costo de deshacerlas sobrepasa al costo de realizar una evaluación.

¿Quiénes están involucrados?

Hay dos grupos de personas involucrados en la evaluación de la arquitectura.

Equipo de evaluación: Estas son las personas que conducirán la evaluación y realizarán el análisis.

Stakeholders: Estos son los interesados en la arquitectura, y en el sistema que se construirá a partir de ella. Algunos de los stakeholders, pero no

todos, serán miembros del equipo de desarrollo, como ser implementadores, verificadores, entre otros. Un tipo especial de stakeholder es el decision maker (tomador de decisiones) del proyecto. Estos son los interesados en la salida de la evaluación y tienen el poder de toma de decisiones. Algunos de ellos son el arquitecto, los diseñadores y el director de proyecto. Cuando un stakeholder quiere que algo sea verdad en la arquitectura, una decision maker tiene el poder de asignar recursos para hacerlo realidad. Generalmente el cliente de la evaluación de la arquitectura es un decision maker, con un gran interés en la salida de la misma y poder sobre todo el proyecto.

En ocasiones el equipo de evaluación está formado por integrantes del plantel del proyecto, en cuyo caso también son stakeholders. Esto no está recomendado dado que habrá pérdida de la objetividad a la hora de ver la arquitectura.

¿Por qué cualidades puede ser evaluada una Arquitectura?

No es del todo cierto que podamos decir que el sistema alcanzará todas sus metas de calidad con solo mirar la arquitectura. Pero muchos atributos de calidad yacen directamente en el reino de la arquitectura. (Camacho, 2004)

Atributos de Calidad [21]

Según Barbacci et al. (1995) la calidad de software se define como el grado en el cual el software posee una combinación deseada de atributos. Tales atributos son requerimientos adicionales del sistema (Kazman et al., 2001), que hacen referencia a características que éste debe satisfacer, diferentes a los requerimientos funcionales.

Estas características o atributos se conocen con el nombre de atributos de calidad, los cuales se definen como las propiedades de un servicio que presta el sistema a sus usuarios (Barbacci et al. 1995).

A grandes rasgos, Bass et al. (1998) establece una clasificación de los atributos de calidad en dos categorías:

- **Observables vía ejecución:** aquellos atributos que se determinan del comportamiento del sistema en tiempo de ejecución.
 1. Disponibilidad: Es la medida de disponibilidad del sistema para el uso (Barbacci et al.,1995).
 2. Confidencialidad: Es la ausencia de acceso no autorizado a la información (Barbacci et al., 1995).
 3. Funcionalidad: Habilidad del sistema para realizar el trabajo para el cual fue concebido (Kazman et al., 2001).
 4. Desempeño: Es el grado en el cual un sistema o componente cumple con sus funciones designadas, dentro de ciertas restricciones dadas, como velocidad, exactitud o uso de memoria. (IEEE 610.12). Según Smith (1993), el desempeño de un sistema se refiere a aspectos temporales del comportamiento del mismo. Se refiere a capacidad de respuesta, ya sea el tiempo requerido para responder a aspectos específicos o el número de eventos procesados en un intervalo de tiempo. Según Bass et al. (1998), se refiere además a la cantidad de comunicación e interacción existente entre los componentes del sistema.
 5. Confiabilidad: Es la medida de la habilidad de un sistema a mantenerse operativo a lo largo del tiempo (Barbacci et al., 1995).
 6. Seguridad externa: Ausencia de consecuencias catastróficas en el ambiente. Es la medida de ausencia de errores que generan pérdidas de información (Barbacci et al., 1995).
 7. Seguridad interna: Es la medida de la habilidad del sistema para resistir a intentos de uso no autorizados y negación del servicio, mientras se sirve a usuarios legítimos (Kazman et al., 2001).

- **No observables vía ejecución:** aquellos atributos que se establecen durante el desarrollo del sistema.
 1. Configurabilidad: Posibilidad que se otorga a un usuario experto a realizar ciertos cambios al sistema (Bosch et al., 1999).
 2. Integrabilidad: Es la medida en que trabajan correctamente componentes del sistema que fueron desarrollados separadamente al ser integrados. (Bass et al. 1998)

3. Integridad: Es la ausencia de alteraciones inapropiadas de la información (Barbacci et al., 1995).
4. Interoperabilidad: Es la medida de la habilidad de que un grupo de partes del sistema trabajen con otro sistema. Es un tipo especial de integrabilidad (Bass et al. 1998)
5. Modificabilidad: Es la habilidad de realizar cambios futuros al sistema. (Bosch et al. 1999).
6. Mantenibilidad: Es la capacidad de someter a un sistema a reparaciones y evolución (Barbacci et al., 1995). Capacidad de modificar el sistema de manera rápida y a bajo costo (Bosch et al. 1999).
7. Portabilidad: Es la habilidad del sistema para ser ejecutado en diferentes ambientes de computación. Estos ambientes pueden ser hardware, software o una combinación de los dos (Kazman et al., 2001).
8. Reusabilidad: Es la capacidad de diseñar un sistema de forma tal que su estructura o parte de sus componentes puedan ser reutilizados en futuras aplicaciones (Bass et al. 1998).
9. Escalabilidad: Es el grado con el que se pueden ampliar el diseño arquitectónico, de datos o procedimental (Pressman, 2002).
10. Capacidad de Prueba: Es la medida de la facilidad con la que el software, al ser sometido a una serie de pruebas, puede demostrar sus fallas. Es la probabilidad de que, asumiendo que tiene al menos una falla, el software fallará en su próxima ejecución de prueba (Bass et al. 1998).

Si otro atributo de calidad, además de los mencionados antes, es importante para determinado proyecto, este también debe formar parte de la evaluación. En particular, el ATAM permite definir nuevos atributos de calidad a ser utilizados en la evaluación.

El punto es que los atributos de calidad no son cantidades absolutas, existen en un contexto con metas específicas.

No parece razonable, considerar que un sistema pueda alguna vez, por ejemplo, ser completamente confiable bajo toda circunstancia (pensar en problemas de energía, meteorológicos, entre otros). Dado esto, es importante que el arquitecto entienda perfectamente bajo qué circunstancias

un sistema debe ser confiable para ser considerado aceptable. Por lo tanto, el primer trabajo que debe realizar una evaluación de arquitectura es obtener las metas específicas de calidad ante las cuales la arquitectura será juzgada. Si algunas de estas metas no son específicas o son ambiguas, se debe pedir a los stakeholders que ayuden al equipo de evaluación a reescribirlas. El mecanismo a utilizar para representar estas metas es el de escenario. Un escenario es una pequeña descripción de la interacción de un stakeholder con el sistema. Los escenarios son parecidos a los casos de uso.

¿Cómo evaluar una Arquitectura?

Las mediciones que se realizan sobre una arquitectura de software pueden tener distintos objetivos, dependiendo de la situación en la que se encuentre el arquitecto y la aplicabilidad de las técnicas que emplea. Los objetivos que menciona Bosch (2000) son tres: cualitativos, cuantitativos y máximos y mínimos teóricos.

La medición cualitativa se aplica para la comparación entre arquitecturas candidatas y tiene relación con la intención de saber la opción que se adapta mejor a cierto atributo de calidad. Este tipo de medición brinda respuestas afirmativas o negativas, sin mayor nivel de detalle. La medición cuantitativa busca la obtención de valores que permitan tomar decisiones en cuanto a los atributos de calidad de una arquitectura de software. El esquema general es la comparación con márgenes establecidos, como lo es el caso de los requerimientos de desempeño, para establecer el grado de cumplimiento de una arquitectura candidata, o tomar decisiones sobre ella.

Este enfoque permite establecer comparaciones, pero se ve limitado en tanto no se conozcan los valores teóricos máximos y mínimos de las mediciones con las que se realiza la comparación. Por último, la medición de máximo y mínimo teórico contempla los valores teóricos para efectos de la comparación de la medición con los atributos de calidad especificados. El conocimiento de los valores máximos o mínimos permite el establecimiento claro del grado de cumplimiento de los atributos de calidad.

En líneas generales, el planteamiento anterior de Bosch (2000) presenta los objetivos para efectos de la medición de los atributos de calidad. Sin embargo, en ocasiones, la evaluación de una arquitectura de software no

produce valores numéricos que permiten la toma de decisiones de manera directa (Kazman, 2001).

Una de las diferencias principales entre los planteamientos de Bosch (2000) y Kazman et al. (2001) es el enfoque que utilizan para efectos de la evaluación. El método de diseño de arquitecturas planteado por Bosch (2000) tiene como principal característica la evaluación explícita de los atributos de calidad de la arquitectura durante el proceso de diseño de la misma. El autor afirma que el enfoque tradicional en la industria de software es el de implementar el sistema y luego establecer valores para los atributos de calidad del mismo. Este enfoque, según su experiencia, tiene la desventaja de que se destina gran cantidad de recursos y esfuerzo en el desarrollo de un sistema que no satisface los requerimientos de calidad. En este sentido, Bosch (2000) plantea las técnicas de evaluación: basada en escenarios, basada en simulación, basada en modelos matemáticos y basada en experiencia.

Por su parte, Kazman et al. (2001) proponen que resulta de poco interés la caracterización o medición de atributos de calidad en las fases tempranas del proceso de diseño, dado que estos parámetros son, por lo general, dependientes de la implementación. Su enfoque se orienta hacia la mitigación de riesgos, ubicando dónde un atributo de calidad de interés se ve afectado por decisiones arquitectónicas. En su estudio, Kazman et al. (2001) presentan tres métodos de evaluación de arquitecturas de software, que son Architecture Trade-off Analysis Method (ATAM), Software Architecture Analysis Method (SAAM) y Active Intermediate Designs Review (ARID).

Tanto Bosch (2000) como Kazman et al. (2001) indican la importancia de la especificación exhaustiva de los atributos de calidad como base para efectos de la evaluación de una arquitectura de software. Descripciones tales como “el sistema debe ser robusto” o “el sistema debe exhibir un desempeño aceptable” resultan ambiguos, puesto que lo que se entiende de ellos puede ser diferente para los distintos involucrados con el sistema (Kazman et al., 2001). El punto es entonces definir los atributos de calidad en función de sus metas y su contexto, y no como cantidades absolutas, según Kazman y sus colegas.

De los planteamientos de evaluación establecidos por Bosch (2000) y Kazman et al. (2001), se tiene que la evaluación de las arquitecturas de software puede ser realizada mediante el uso de diversas técnicas y métodos. En este sentido, resulta interesante estudiar las distintas opciones que existen en la actualidad para llevar a cabo esta tarea.

Según Bosch (2000), las técnicas utilizadas para la evaluación de atributos de calidad requieren grandes esfuerzos por parte del ingeniero de software para crear especificaciones y predicciones. Estas técnicas requieren información del sistema a desarrollar que no está disponible durante el diseño arquitectónico, sino al principio del diseño detallado del sistema.

En vista de que el interés es tomar decisiones de tipo arquitectónico en las fases tempranas del desarrollo, son necesarias técnicas que requieran poca información detallada y puedan conducir a resultados relativamente precisos (Bosch, 2000). Las técnicas existentes en la actualidad para evaluar arquitecturas permiten hacer una evaluación cuantitativa sobre los atributos de calidad a nivel arquitectónico, pero se tienen pocos medios para predecir el máximo (o mínimo) teórico para las arquitecturas de software. Sin embargo, debido al costo de realizar este tipo de evaluación, en muchos casos los arquitectos de software evalúan cualitativamente, para decidir entre las alternativas de diseño (Bosch, 2000). Bosch (2000) propone diferentes técnicas de evaluación de arquitecturas de software, a saber: evaluación basada en escenarios, evaluación basada en simulación, evaluación basada en modelos matemáticos y evaluación basada en experiencia.

En este sentido, Bosch (2000) plantea las técnicas de evaluación: basada en escenarios, basada en simulación, basada en modelos matemáticos y basada en experiencia.

¿Qué resultado produce la evaluación de una Arquitectura?

En términos concretos, la evaluación de la arquitectura produce un informe, la forma y contenido del mismo varía según el método utilizado. En particular, produce repuestas a dos tipos de preguntas:

- ¿Es esta arquitectura adecuada para el sistema para la cual fue diseñada?

- ¿Cuál de dos o más arquitecturas propuestas es la más adecuada para el sistema?

Decimos que una arquitectura es adecuada cuando cumple dos criterios:

- El sistema resultante de ella cumple con los objetivos de calidad. No todas las propiedades de calidad del sistema son resultado directo de la arquitectura, pero muchas lo son.
- El sistema puede ser construido con los recursos con que se cuenta: el plantel, el presupuesto, el sistema legado (si hay), entre otros. Esto es, la arquitectura es construible.

Un resultado que también produce la evaluación de una arquitectura es la captura y priorización de las metas que la arquitectura debe cumplir para poder ser considerada adecuada.

La evaluación de una arquitectura no produce resultados cuantitativos. No es de interés, por ejemplo, evaluar la performance en cantidad de transacciones por segundo a esta altura, dado que el sistema no está construido aún. Lo que interesa, en un espíritu de mitigación de riesgos, es aprender como un atributo de calidad es afectado por una decisión de diseño arquitectónico, para que de esta manera se pueda estudiar con cuidado dicha decisión.

Una evaluación arquitectónica dice si una arquitectura es adecuada respecto a un conjunto de metas, y problemática con respecto a otro conjunto de metas. En ocasiones, las metas pueden ser contradictorias entre ellas, o algunas ser más importantes que otras. El director de proyecto es quien deberá tomar la decisión si la arquitectura evalúa bien o mal en las distintas áreas. La evaluación ayuda a encontrar debilidades, no dirá “sí” o “no”, “bien” o “mal”, “6 en 10”, dirá dónde están los riesgos.

1.3. Definición de Términos Básicos

1.3.1. Objetivo SMART

Es un conjunto de metas y que el nombre en sí es una regla para definir las, para ello se debe responder a estos cinco:

- **Specific** (Específico)
- **Measurable** (Medible)
- **Achievable** – (Alcanzable)
- **Realistic** (Realista)
- **Timely** (Definido en un plazo de tiempo determinado)

1.3.2. Software Web

Se le conoce así a las páginas, sitios o portales que brindan servicios de interacción con el usuario y son dinámicas; además detrás de ellas hay un repositorio de datos que con ayuda de las nuevas tecnologías no hace mucho se comenzó el desarrollo de aplicaciones Web.

Esto produjo una evolución de los mecanismos para vincular los interfaces con bases de datos y esencialmente hacer todas las aplicaciones accesibles desde un navegador Web. Hoy en día podemos ver aplicaciones Web que aprovechan al máximo los beneficios de la Web. Como se ha descrito anteriormente, una aplicación Web utiliza una página Web para introducir datos (por ejemplo, formularios), enviarlos al servidor, procesar dichos datos (de acuerdo a las políticas del negocio) y mostrar los resultados por medio de otra página Web.

1.3.3. Sistema de Gestión Académica Universitaria.

Para lograr competitividad y por tanto, una mejor calidad en los procesos que determinan el actuar en las distintas unidades académicas de las instituciones Universitarias es necesario un control efectivo del cumplimiento de los objetivos estratégicos. Cabe resaltar que las Instituciones Universitarias, dadas por su origen, historia, localización y objetivos, presentan una gran diversidad la cual requiere de estrategias distintas de desarrollo acordes con dichas misión y visión. (Palacio, 2006).

1.3.4. Arquitectura de Software

Es resultado de la toma de decisiones a la hora de diseñar un sistema, dichas decisiones son seleccionadas a partir de los requerimientos funcionales y no funcionales de los interesados en el sistema, con el fin de alcanzar la mayor calidad del software como producto final. La arquitectura de software se produce durante las etapas de requerimientos y diseño del proceso de desarrollo del software.

Los Atributos de Calidad considerados para evaluar la Arquitectura de Software propuesta son: Configurabilidad, Integrabilidad, Integridad, Interoperabilidad, Modificabilidad, Mantenibilidad, Portabilidad, Reusabilidad, Escalabilidad y Capacidad de Prueba.

1.3.5. Sistema de Gestión Académica

Es el software de gestión académica de la Universidad Nacional de la Amazonía Peruana (UNAP).

Los Atributos de Calidad considerados para evaluar el Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana (UNAP) son: Disponibilidad, Confidencialidad, Funcionalidad, Desempeño, Confiabilidad, Seguridad externa y Seguridad interna.

CAPÍTULO II: VARIABLES E HIPÓTESIS

2.1. Variables y su Operacionalización

2.1.1. Variable Independiente (X)

Arquitectura de Software

2.1.2. Variable Dependiente (Y)

Sistema de Gestión Académica

2.1.3. Indicadores e Índices

2.1.3.1. De la Variable (X):

Dimensión: Operatividad

1. **Configurabilidad:** Indicador que permite determinar la posibilidad que se otorga a un usuario experto a realizar ciertos cambios al sistema.
2. **Integrabilidad:** Indicador que permite determinar la medida en que trabajan correctamente componentes del sistema que fueron desarrollados separadamente al ser integrados.
3. **Integridad:** Indicador que permite determinar la ausencia de alteraciones inapropiadas de la información.
4. **Interoperabilidad:** Indicador que permite determinar la medida de la habilidad de que un grupo de partes del sistema trabajen con otro sistema. Es un tipo especial de integrabilidad.
5. **Capacidad de Prueba:** Indicador que permite determinar la medida de la facilidad con la que el software, al ser sometido a una serie de pruebas, puede demostrar sus fallas. Es la probabilidad de que, asumiendo que tiene al menos una falla, el software fallará en su próxima ejecución de prueba.

Dimensión: Actualización

6. **Modificabilidad:** Indicador que permite determinar la habilidad de realizar cambios futuros al sistema.
7. **Mantenibilidad:** Indicador que permite determinar la capacidad de someter a un sistema a reparaciones y evolución. Capacidad de modificar el sistema de manera rápida y a bajo costo.
8. **Portabilidad:** Indicador que permite determinar la habilidad del sistema para ser ejecutado en diferentes ambientes de computación. Estos ambientes pueden ser hardware, software o una combinación de los dos.
9. **Reusabilidad:** Indicador que permite determinar la capacidad de diseñar un sistema de forma tal que su estructura o parte de sus componentes puedan ser reutilizados en futuras aplicaciones.
10. **Escalabilidad:** Indicador que permite determinar el grado con el que se pueden ampliar el diseño arquitectónico, de datos o procedimental.

2.1.3.2. De la Variable (Y):

Dimensión: Satisfacción

11. **Disponibilidad:** Indicador que determina la disponibilidad del sistema para el uso.
12. **Confidencialidad:** Indicador que determina la ausencia de acceso no autorizado a la información.
13. **Confiabilidad:** Indicador que determina la medida de la habilidad de un sistema a mantenerse operativo a lo largo del tiempo.

Dimensión: Eficiencia

14. **Desempeño:** Indicador que determina el grado en el cual un sistema o componente cumple con sus funciones designadas,

dentro de ciertas restricciones dadas, como velocidad, exactitud o uso de memoria.

15. **Funcionalidad:** Indicador que determina la habilidad del sistema para realizar el trabajo para el cual fue concebido.

16. **Seguridad Externa:** Indicador que determina la ausencia de consecuencias catastróficas en el ambiente. Es la medida de ausencia de errores que generan pérdidas de información.

17. **Seguridad Interna:** Indicador que determina la medida de la habilidad del sistema para resistir a intentos de uso no autorizados y negación del servicio, mientras se sirve a usuarios.

NOTA: Se debe indicar que para ambas variables los valores de medición (índices) serán en relación a las Técnicas utilizadas:

Objetivos SMART:

a) Cumple b) Cumple parcialmente c) No cumple

2.1.4. Operacionalización de las Variables

| Variables | Dimensión | Indicadores | Valores de Medición | Instrumentos de Medición |
|------------------------------|---------------|---------------------|--|--------------------------|
| Arquitectura de Software | Operatividad | Configurabilidad | - Cumple - Cumple Parcialmente - No cumple | Cuestionario |
| | | Integrabilidad | | |
| | | Integridad | | |
| | | Interoperabilidad | | |
| | | Capacidad de Prueba | | |
| | Actualización | Modificabilidad | | |
| | | Mantenibilidad | | |
| | | Portabilidad | | |
| | | Reusabilidad | | |
| | | Escalabilidad | | |
| Sistema de Gestión Académica | Satisfacción | Disponibilidad | | |
| | | Confidencialidad | | |
| | | Confiabilidad | | |
| | Eficiencia | Desempeño | | |
| | | Funcionalidad | | |
| | | Seguridad externa | | |
| | | Seguridad interna | | |

Fuente: Elaboración Propia

2.2. Formulación de la Hipótesis

2.2.1. Hipótesis de la Investigación (H1)

Una arquitectura de referencia basada en el marco de trabajo TOGAF, permitirá mejorar el Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana.

2.2.2. Hipótesis Nula (H0)

Una arquitectura de referencia basada en el marco de trabajo TOGAF, no permitirá mejorar el Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana.

CAPÍTULO III: METODOLOGÍA

3.1. Tipo y Diseño de la Investigación

3.1.1. Tipo de Investigación

De acuerdo a la clasificación que hacen Sampieri, Fernandez y Lucio (2006), los proyectos investigativos se pueden clasificar en cualitativos, cuantitativos y mixtos de acuerdo a su metodología y orientación. Para efectos de este proyecto se considera que el tipo de investigación es **Cualitativo, Descriptivo y Explicativo**. Porque además de hacer un diagnóstico de la arquitectura actual del sistema de gestión académica, se efectuará una exposición de los detalles de la nueva arquitectura de software.

3.1.2. Diseño de Investigación

El diseño de la investigación es no experimental. Por lo tanto, el diseño de la investigación es ***Transeccional Descriptivo***.

3.2. Población y Muestra

3.2.1. Población de estudio

La población estuvo representada por todo el personal del Vicerrectorado Académico de la Universidad Nacional de la Amazonía Peruana.

3.2.2. Muestreo o selección de la muestra

Para la muestra se empleó la técnica del muestreo no probabilístico intencional por criterio (Corral et al, 2015, p.162), porque se recibió recomendaciones de expertos. De tal manera que se consideró una muestra de seis (6) personas, a quienes se les aplicó el cuestionario del pre y post test para determinar el nivel de mejora del Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana.

3.3. Técnicas e Instrumentos

Para el desarrollo de la investigación se utilizará la entrevista y la encuesta estructurada y cerrada como mecanismo de recolección de datos.

Se coordinará con las personas interesadas e involucradas en los procesos de gestión académica y de la administración del aplicativo actual para realizarles las entrevistas y/o encuestas, indicando la fecha, hora y lugar donde se realizará.

En cuanto a la construcción del instrumento, los cuestionarios se diseñaron y fueron llenados por los interesados.

En el Anexo N° 02 se presenta el cuestionario de preguntas.

3.4. Procedimientos de Recolección de Datos

Se efectuarán encuestas hacia el personal previamente seleccionado que sienta un impacto relevante (stakeholders) al desarrollar la investigación planteada. El grupo de interés sería:

- **Comisión del Sistema Académico actual:**
 - Lic. Adm. Carlos Fernando Aguilar Hernández
 - Dr. Rodil Tello Espinoza
 - Dra. Ruth Vílchez Ramírez
 - Ing. Luis Pita Astengo
 - Ing. Ytala Milagros Guadalupe Pizango
 - Ing. Mary Tenazoa Rivera

3.5. Técnicas de Procesamiento y Análisis de los Datos

Posterior a la captura de datos y de acuerdo a los objetivos planteados se procederá a un análisis de los mismos cuyos resultados se presentarán a la Universidad Nacional de la Amazonía Peruana y Jefaturas correspondientes para su contrastación, siguiendo las siguientes tareas:

- ✓ Recopilación y tabulación de datos.
- ✓ Elaboración de tablas de frecuencias.
- ✓ Análisis y consistencia de datos.

- ✓ Interpretación de los datos y la validación de la hipótesis.

Por lo tanto, una arquitectura de referencia basada en el marco de trabajo TOGAF, permitirá mejorar el Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana.

Para el análisis de los datos se usarán herramientas como el Microsoft Excel y también se tendrá en cuenta la observación directa del juicio de expertos para ayudar a comprender ciertos resultados del negocio.

Dentro de las perspectivas de la investigación se tiene en consideración la importancia que a futuro se realice seguimiento de los aspectos contemplados en esta investigación para ver su evolución y continuar con el desarrollo de la misma.

Para finalizar el procesamiento de la información se procederá a documentar el proceso y generar el presente informe.

3.6. Aspectos Éticos

Previo al procedimiento de recolección de datos se realizó las coordinaciones con los miembros de la Comisión del Sistema Académico de la UNAP, de manera que se pueda contar con el consentimiento correspondiente y posteriormente aceptado por el titular de la Vicerrectoría Académica de la UNAP se procedió a la recolección de los datos para lo cual me comprometí a garantizar la confidencialidad de la información brindada en las encuestas para elaborar la presente investigación y así poder respaldar la veracidad de los resultados.

CAPÍTULO IV: RESULTADOS

4.1. Análisis Descriptivo por Indicadores de la Variable Independiente antes de la aplicación del diseño propuesto.

Tabla 1. Dimensión: Nivel de Operatividad del Sistema – Pre Test

| INDICADORES | Cumple | | Cumple Parcialmente | | No Cumple | | Total | |
|---|--------|--------|---------------------|--------|-----------|--------|-------|------|
| | fi | % | fi | % | fi | % | fi | % |
| CONFIGURABILIDAD | 0 | 00.00% | 4 | 66.67% | 2 | 33.33% | 6 | 100% |
| INTEGRABILIDAD | 1 | 16.67% | 3 | 50.00% | 2 | 33.33% | 6 | 100% |
| INTEGRIDAD | 2 | 33.33% | 2 | 33.33% | 2 | 33.33% | 6 | 100% |
| INTEROPERABILIDAD | 1 | 16.67% | 3 | 50.00% | 2 | 33.33% | 6 | 100% |
| CAPACIDAD DE PRUEBA | 0 | 00.00% | 3 | 50.00% | 3 | 50.00% | 6 | 100% |
| PROMEDIO (\bar{x}): | 0.8 | 13.33% | 3.0 | 50.00% | 2.2 | 36.67% | 6.0 | 100% |

Fuente: Cuestionario aplicado al personal de la Comisión del Sistema Académico de la Universidad Nacional de la Amazonía Peruana

Se observa en la Tabla 1 que el nivel de operatividad del sistema en la aplicación de una arquitectura propuesta es lo siguiente:

Del promedio (\bar{x}) de 6 (100%) encuestados, 0.8 (13.33%) encuestados manifestaron que se cumple el nivel de operatividad del sistema en la aplicación de una arquitectura propuesta, 3.0 (50.00%) encuestados manifestaron que se cumple parcialmente el nivel de operatividad del sistema en la aplicación de una arquitectura propuesta, mientras que 2.2 (36.67%) encuestados manifestaron que no se cumple el nivel de operatividad del sistema en la aplicación de una arquitectura propuesta.

Tabla 2. Dimensión: Nivel de Actualización del Sistema – Pre Test

| INDICADORES | Cumple | | Cumple Parcialmente | | No Cumple | | Total | |
|---|--------|--------|---------------------|--------|-----------|--------|-------|------|
| | fi | % | fi | % | fi | % | fi | % |
| MODIFICABILIDAD | 2 | 33.33% | 2 | 33.33% | 2 | 33.33% | 6 | 100% |
| MANTENIBILIDAD | 2 | 33.33% | 2 | 33.33% | 2 | 33.33% | 6 | 100% |
| PORTABILIDAD | 0 | 00.00% | 3 | 50.00% | 3 | 50.00% | 6 | 100% |
| REUSABILIDAD | 2 | 33.33% | 3 | 50.00% | 1 | 16.67% | 6 | 100% |
| ESCALABILIDAD | 2 | 33.33% | 3 | 50.00% | 1 | 16.67% | 6 | 100% |
| PROMEDIO (\bar{x}): | 1.6 | 26.67% | 2.6 | 43.33% | 1.8 | 30.00% | 6.0 | 100% |

Fuente: Cuestionario aplicado al personal de la Comisión del Sistema Académico de la Universidad Nacional de la Amazonía Peruana

Se observa en la Tabla 2 que el nivel de actualización del sistema en la aplicación de una arquitectura propuesta es lo siguiente:

Del promedio (\bar{x}) de 6 (100%) encuestados, 1.6 (26.67%) encuestados manifestaron que se cumple el nivel de actualización del sistema en la aplicación de una arquitectura propuesta, 2.6 (43.33%) encuestados manifestaron que se cumple parcialmente el nivel de actualización del sistema en la aplicación de una arquitectura propuesta, mientras que 1.8 (30.00%) encuestados manifestaron que no se cumple el nivel de actualización del sistema en la aplicación de una arquitectura propuesta.

4.2. Análisis Descriptivo por Dimensiones de la Variable Independiente antes de la aplicación del diseño propuesto.

Tabla 3. Variable Independiente: Arquitectura de Software – Pre Test

| DIMENSIONES | Cumple | | Cumple Parcialmente | | No Cumple | | Total | |
|---|--------|--------|---------------------|--------|-----------|--------|-------|------|
| | fi | % | fi | % | fi | % | fi | % |
| OPERATIVIDAD | 1 | 16.67% | 3 | 50.00% | 2 | 33.33% | 6 | 100% |
| ACTUALIZACIÓN | 2 | 33.33% | 2 | 33.33% | 2 | 33.33% | 6 | 100% |
| PROMEDIO (\bar{x}): | 1.5 | 25.00% | 2.5 | 41.67% | 2.0 | 33.33% | 6.0 | 100% |

Fuente: Cuestionario aplicado al personal de la Comisión del Sistema Académico de la Universidad Nacional de la Amazonía Peruana

Se observa en la Tabla 3 que la aplicación de la arquitectura propuesta es lo siguiente:

Del promedio (\bar{x}) de 6 (100%) encuestados, 1.5 (25.00%) encuestados manifestaron que se cumple la aplicación de la arquitectura propuesta, 2.5 (41.67%) encuestados manifestaron que se cumple parcialmente la aplicación de la arquitectura propuesta, mientras que 2.0 (33.33%) encuestados manifestaron que no se cumple la aplicación de la arquitectura propuesta.

4.3. Análisis Descriptivo por Indicadores de la Variable Dependiente antes de la aplicación del diseño propuesto.

Tabla 4. Dimensión: Nivel de Satisfacción del sistema – Pre Test

| INDICADORES | Cumple | | Cumple Parcialmente | | No Cumple | | Total | |
|---|--------|--------|---------------------|--------|-----------|--------|-------|------|
| | fi | % | fi | % | fi | % | fi | % |
| DISPONIBILIDAD | 2 | 33.33% | 3 | 50.00% | 1 | 16.67% | 6 | 100% |
| CONFIDENCIALIDAD | 2 | 33.33% | 2 | 33.33% | 2 | 33.33% | 6 | 100% |
| CONFIABILIDAD | 1 | 16.67% | 2 | 33.33% | 3 | 50.00% | 6 | 100% |
| PROMEDIO (\bar{x}): | 1.7 | 27.78% | 2.3 | 38.89% | 2.0 | 33.33% | 6.0 | 100% |

Fuente: Cuestionario aplicado al personal de la Comisión del Sistema Académico de la Universidad Nacional de la Amazonía Peruana

Se observa en la Tabla 4 que el nivel de satisfacción del sistema de gestión académica es lo siguiente:

Del promedio (\bar{x}) de 6 (100%) encuestados, 1.7 (27.78%) encuestados manifestaron que se cumple el nivel de satisfacción del sistema de gestión académica, 2.3 (38.89%) encuestados manifestaron que se cumple parcialmente el nivel de satisfacción del sistema de gestión académica, mientras que 2.0 (33.33%) encuestados manifestaron que no se cumple el nivel de satisfacción del sistema de gestión académica.

Tabla 5. Dimensión: Nivel de Eficiencia del sistema – Pre Test

| INDICADORES | Cumple | | Cumple Parcialmente | | No Cumple | | Total | |
|---|--------|--------|---------------------|--------|-----------|--------|-------|------|
| | fi | % | fi | % | fi | % | fi | % |
| DESEMPEÑO | 2 | 33.33% | 3 | 50.00% | 1 | 16.67% | 6 | 100% |
| FUNCIONALIDAD | 2 | 33.33% | 3 | 50.00% | 1 | 16.67% | 6 | 100% |
| SEGURIDAD EXTERNA | 1 | 16.67% | 2 | 33.33% | 3 | 50.00% | 6 | 100% |
| SEGURIDAD INTERNA | 1 | 16.67% | 3 | 50.00% | 2 | 33.33% | 6 | 100% |
| PROMEDIO (\bar{x}): | 1.5 | 25.00% | 2.8 | 45.83% | 1.8 | 29.17% | 6.0 | 100% |

Fuente: Cuestionario aplicado al personal de la Comisión del Sistema Académico de la Universidad Nacional de la Amazonía Peruana

Se observa en la Tabla 5 que el nivel de eficiencia del sistema de gestión académica es lo siguiente:

Del promedio (\bar{x}) de 6 (100%) encuestados, 1.5 (25.00%) encuestados manifestaron que se cumple el nivel de eficiencia del sistema de gestión académica, 2.8 (45.83%) encuestados manifestaron que se cumple parcialmente el nivel de eficiencia del sistema de gestión académica, mientras que 1.8 (29.17%) encuestados manifestaron que no cumple el nivel de eficiencia del sistema de gestión académica.

4.4. Análisis Descriptivo por Dimensiones de la Variable Dependiente antes de la aplicación del diseño propuesto.

Tabla 6. Variable Dependiente: Sist. de Gestión Académica – Pre Test

| DIMENSIONES | Cumple | | Cumple Parcialmente | | No Cumple | | Total | |
|---|--------|--------|---------------------|--------|-----------|--------|-------|------|
| | Fi | % | fi | % | fi | % | fi | % |
| SATISFACCIÓN | 2 | 33.33% | 2 | 33.33% | 2 | 33.33% | 6.0 | 100% |
| EFICIENCIA | 2 | 33.33% | 2 | 33.33% | 2 | 33.33% | 6.0 | 100% |
| PROMEDIO (\bar{x}): | 2.0 | 33.33% | 2.0 | 33.33% | 2.0 | 33.33% | 6.0 | 100% |

Fuente: Cuestionario aplicado al personal de la Comisión del Sistema Académico de la Universidad Nacional de la Amazonía Peruana

Se observa en la Tabla 6 que la aplicación de la arquitectura propuesta es lo siguiente:

Del promedio (\bar{x}) de 6 (100%) encuestados, 2.0 (33.33%) encuestados manifestaron que se cumple la gestión a través del sistema de gestión académica, 2.0 (33.33%) encuestados manifestaron que se cumple parcialmente la gestión a través del sistema de gestión académica, mientras que 2.0 (33.33%) encuestados manifestaron que no se cumple la mejora a través del sistema de gestión académica.

4.5. Análisis Descriptivo por Indicadores de la Variable Independiente después de la aplicación del diseño propuesto.

Tabla 7. Dimensión: Nivel de Operatividad del sistema – Post Test

| INDICADORES | Cumple | | Cumple Parcialmente | | No Cumple | | Total | |
|---|--------|--------|---------------------|--------|-----------|-------|-------|------|
| | fi | % | fi | % | fi | % | fi | % |
| CONFIGURABILIDAD | 4 | 66.67% | 2 | 33.33% | 0 | 0.00% | 6 | 100% |
| INTEGRABILIDAD | 5 | 83.33% | 1 | 16.67% | 0 | 0.00% | 6 | 100% |
| INTEGRIDAD | 4 | 66.67% | 2 | 33.33% | 0 | 0.00% | 6 | 100% |
| INTEROPERABILIDAD | 5 | 83.33% | 1 | 16.67% | 0 | 0.00% | 6 | 100% |
| CAPACIDAD DE PRUEBA | 4 | 66.67% | 2 | 33.33% | 0 | 0.00% | 6 | 100% |
| PROMEDIO (\bar{x}): | 4.4 | 73.33% | 1.6 | 26.67% | 0.0 | 0.00% | 6.0 | 100% |

Fuente: Cuestionario aplicado al personal de la Comisión del Sistema Académico de la Universidad Nacional de la Amazonía Peruana

Se observa en la Tabla 7 que el nivel de operatividad del sistema en la aplicación de una arquitectura propuesta es lo siguiente:

Del promedio (\bar{x}) de 6 (100%) encuestados, 4.4 (73.33%) encuestados manifestaron que se cumple el nivel de operatividad del sistema en la aplicación de una arquitectura propuesta, 1.6 (26.67%) encuestados manifestaron que se cumple parcialmente el nivel de operatividad del sistema en la aplicación de una arquitectura propuesta, mientras que 0 (0.00%) es decir ningún encuestado manifestó que no se cumple el nivel de operatividad del sistema en la aplicación de una arquitectura propuesta.

Tabla 8. Dimensión: Nivel de Actualización del sistema – Post Test

| INDICADORES | Cumple | | Cumple Parcialmente | | No Cumple | | Total | |
|---|--------|--------|---------------------|--------|-----------|-------|-------|------|
| | fi | % | fi | % | fi | % | fi | % |
| MODIFICABILIDAD | 4 | 66.67% | 2 | 33.33% | 0 | 0.00% | 6 | 100% |
| MANTENIBILIDAD | 5 | 83.33% | 1 | 16.67% | 0 | 0.00% | 6 | 100% |
| PORTABILIDAD | 5 | 83.33% | 1 | 16.67% | 0 | 0.00% | 6 | 100% |
| REUSABILIDAD | 5 | 83.33% | 1 | 16.67% | 0 | 0.00% | 6 | 100% |
| ESCALABILIDAD | 5 | 83.33% | 1 | 16.67% | 0 | 0.00% | 6 | 100% |
| PROMEDIO (\bar{x}): | 4.8 | 80.00% | 1.2 | 20.00% | 0.0 | 0.00% | 6.0 | 100% |

Fuente: Cuestionario aplicado al personal de la Comisión del Sistema Académico de la Universidad Nacional de la Amazonía Peruana

Se observa en la Tabla 8 que el nivel de actualización del sistema en la aplicación de una arquitectura propuesta es lo siguiente:

Del promedio (\bar{x}) de 6 (100%) encuestados, 4.8 (80.00%) encuestados manifestaron que se cumple el nivel de actualización del sistema en la aplicación de una arquitectura propuesta, 1.2 (20.00%) encuestados manifestaron que se cumple parcialmente el nivel de actualización del sistema en la aplicación de una arquitectura propuesta, mientras que 0 (0.00%) es decir ningún encuestado manifestó que no se cumple el nivel de actualización del sistema en la aplicación de una arquitectura propuesta.

4.6. Análisis Descriptivo por Dimensiones de la Variable Independiente después de la aplicación del diseño propuesto.

Tabla 9. Variable Independiente: Arquitectura de Software – Post Test

| DIMENSIONES | Cumple | | Cumple Parcialmente | | No Cumple | | Total | |
|---|--------|--------|---------------------|--------|-----------|-------|-------|------|
| | fi | % | fi | % | fi | % | fi | % |
| OPERATIVIDAD | 4 | 73.33% | 2 | 26.67% | 0 | 0.00% | 6 | 100% |
| ACTUALIZACIÓN | 5 | 80.00% | 1 | 20.00% | 0 | 0.00% | 6 | 100% |
| PROMEDIO (\bar{x}): | 4.6 | 76.67% | 1.4 | 23.33% | 0.0 | 0.00% | 6.0 | 100% |

Fuente: Cuestionario aplicado al personal de la Comisión del Sistema Académico de la Universidad Nacional de la Amazonía Peruana

Se observa en la Tabla 9 que la aplicación de la arquitectura propuesta es lo siguiente:

Del promedio (\bar{x}) de 6 (100%) encuestados, 4.6 (76.67%) encuestados manifestaron que se cumple la aplicación de la arquitectura propuesta, 1.4 (23.33%) encuestados manifestaron que se cumple parcialmente la aplicación de la arquitectura propuesta, mientras que 0 (0.00%) es decir ningún encuestado manifestó que no se cumple la aplicación de la arquitectura propuesta.

4.7. Análisis Descriptivo por Indicadores de la Variable Dependiente después de la aplicación del diseño propuesto.

Tabla 10. Dimensión: Nivel de Satisfacción del sistema – Post Test

| INDICADORES | Cumple | | Cumple Parcialmente | | No Cumple | | Total | |
|---|--------|--------|---------------------|--------|-----------|-------|-------|------|
| | fi | % | fi | % | fi | % | fi | % |
| DISPONIBILIDAD | 5 | 83.33% | 1 | 16.67% | 0 | 0.00% | 6 | 100% |
| CONFIDENCIALIDAD | 4 | 66.67% | 2 | 33.33% | 0 | 0.00% | 6 | 100% |
| CONFIABILIDAD | 4 | 66.67% | 2 | 33.33% | 0 | 0.00% | 6 | 100% |
| PROMEDIO (\bar{x}): | 4.3 | 72.22% | 1.7 | 27.78% | 0.0 | 0.00% | 6.0 | 100% |

Fuente: Cuestionario aplicado al personal de la Comisión del Sistema Académico de la Universidad Nacional de la Amazonía Peruana

Se observa en la Tabla 10 que el nivel de satisfacción del sistema de gestión académica es lo siguiente:

Del promedio (\bar{x}) de 6 (100%) encuestados, 4.3 (72.22%) encuestados manifestaron que se cumple el nivel de satisfacción del sistema de gestión académica, 1.7 (27.78%) encuestados manifestaron que se cumple parcialmente el nivel de satisfacción del sistema de gestión académica, mientras que 0 (0.00%) es decir ningún encuestado manifestó que no se cumple el nivel de satisfacción del sistema de gestión académica.

Tabla 11. Dimensión: Nivel de Eficiencia del Sistema – Post Test

| INDICADORES | Cumple | | Cumple Parcialmente | | No Cumple | | Total | |
|---|--------|--------|---------------------|--------|-----------|--------|-------|------|
| | fi | % | fi | % | fi | % | fi | % |
| DESEMPEÑO | 4 | 66.67% | 2 | 33.33% | 0 | 0.00% | 6 | 100% |
| FUNCIONALIDAD | 5 | 83.33% | 1 | 16.67% | 0 | 0.00% | 6 | 100% |
| SEGURIDAD EXTERNA | 3 | 50.00% | 2 | 33.33% | 1 | 16.67% | 6 | 100% |
| SEGURIDAD INTERNA | 4 | 66.67% | 1 | 16.67% | 1 | 16.67% | 6 | 100% |
| PROMEDIO (\bar{x}): | 4.0 | 66.67% | 1.5 | 25.00% | 0.5 | 8.33% | 6.0 | 100% |

Fuente: Cuestionario aplicado al personal de la Comisión del Sistema Académico de la Universidad Nacional de la Amazonía Peruana

Se observa en la Tabla 11 que el nivel de eficiencia del sistema de gestión académica es lo siguiente:

Del promedio (\bar{x}) de 6 (100%) encuestados, 4 (66.67%) encuestados manifestaron que se CUMPLE el nivel de eficiencia del sistema de gestión académica, 1.5 (25.00%) encuestados manifestaron que se cumple parcialmente el nivel de eficiencia del sistema de gestión académica, mientras que 0.5 (08.33%) encuestados manifestaron que no cumple el nivel de eficiencia del sistema de gestión académica.

4.8. Análisis Descriptivo por Dimensiones de la Variable Dependiente después de la aplicación del diseño propuesto.

Tabla 12. Variable Dependiente: Sist. de Gestión Académica – Post Test

| DIMENSIONES | Cumple | | Cumple Parcialmente | | No Cumple | | Total | |
|---|--------|--------|---------------------|--------|-----------|-------|-------|------|
| | fi | % | fi | % | fi | % | fi | % |
| SATISFACCIÓN | 4.3 | 72.22% | 1.7 | 27.78% | 0.0 | 0.00% | 6.0 | 100% |
| EFICIENCIA | 4.0 | 66.67% | 1.5 | 25.00% | 0.5 | 8.33% | 6.0 | 100% |
| PROMEDIO (\bar{x}): | 4.2 | 69.44% | 1.6 | 26.39% | 0.3 | 4.17% | 6.0 | 100% |

Fuente: Cuestionario aplicado al personal de la Comisión del Sistema Académico de la Universidad Nacional de la Amazonía Peruana

Se observa en la Tabla 12 que la aplicación de la arquitectura propuesta es lo siguiente:

Del promedio (\bar{x}) de 6 (100%) encuestados, 4.2 (69.44%) encuestados manifestaron que se cumple la gestión a través del sistema de gestión académica, 1.6 (26.39%) encuestados manifestaron que se cumple parcialmente la gestión a través del sistema de gestión académica, mientras que 0.3 (4.17%) manifestaron que no se cumple la mejora a través del sistema de gestión académica.

CAPÍTULO V: DISCUSIÓN DE LOS RESULTADOS

Se observa en la Tabla 3 acerca de la arquitectura del sistema sin la aplicación del diseño propuesto lo siguiente: del promedio (\bar{x}) de 6 (100%) encuestados, 1.5 (25.00%) encuestados manifestaron que se cumple, 2.5 (41.67%) encuestados manifestaron que se cumple parcialmente, mientras que 2.0 (33.33%) encuestados manifestaron que no se cumple. Y se observa en la Tabla 9 acerca de la arquitectura del sistema, después de aplicar el diseño propuesto que el 4.4 (73.33%) encuestados manifestaron que se cumple, 1.6 (26.67%) encuestados manifestaron que se cumple parcialmente, mientras que 0 (0.00%) es decir ningún encuestado manifestó que no se cumple. Es decir, existe una diferencia significativa de 48.33% que afirman que se cumple los indicadores que nos permiten evaluar un sistema software a nivel arquitectónico, estos indicadores están distribuidos en dos dimensiones: Operatividad (Configurabilidad, Integrabilidad, Integridad, Interoperabilidad, Capacidad de Prueba) y Actualización (, Modificabilidad, Mantenibilidad, Portabilidad, Reusabilidad, Escalabilidad).

Se observa en la Tabla 6 acerca del sistema de gestión académica sin la aplicación del diseño propuesto lo siguiente: del promedio (\bar{x}) de 6 (100%) encuestados, 2.0 (33.33%) encuestados manifestaron que se cumple, 2.0 (33.33%) encuestados manifestaron que se cumple parcialmente, mientras que 2.0 (33.33%) encuestados manifestaron que no se cumple, es decir que se encontró igual opinión en los niveles de cumplimiento, lo cual refleja demasiada incertidumbre.

Finalmente se observa en la Tabla 12 acerca del sistema de gestión académica después de aplicar el diseño propuesto que el 4.2 (69.44%) encuestados manifestaron que se cumple, 1.6 (26.39%) encuestados manifestaron que se cumple parcialmente, mientras que 0.3 (4.17%) encuestados manifestaron que no se cumple. Es decir, existe una diferencia significativa de 36.11% que afirman que se cumple los indicadores que nos permiten evaluar un sistema software en ejecución, estos indicadores están

distribuidos en dos dimensiones: Satisfacción (Disponibilidad, Confidencialidad, Confiabilidad) y Eficiencia (Desempeño, Funcionalidad, Seguridad Externa, Seguridad Interna). A pesar de ello aún existe un porcentaje muy bajo de 4.17% que afirman que no se cumple aún en la actualidad los indicadores de Seguridad Externa y Seguridad Interna.

CAPÍTULO VI: PROPUESTA

6.1. Introducción: TOGAF como Marco Arquitectural de Referencia

The Open Group Architectural Framework (TOGAF) es un marco genérico para el desarrollo de arquitecturas empresariales. Hasta su versión 7, sólo abordaba la arquitectura tecnológica mientras que a partir de su versión 8, publicada en diciembre de 2002, ya cubre cualquier faceta. La versión inicial de TOGAF se desarrolló en 1995, basándose en TAFIM-Technical Architecture Framework for Information Management, resultado de proyectos y mucho esfuerzo financiados por el Departamento de Defensa (DoD) de los Estados Unidos.

TOGAF se puede dividir en cuatro grandes bloques:

- ✓ Un entorno de alto nivel, basado en algunos de los conceptos clave. El entorno gira en torno a la arquitectura empresarial, que está compuesta por cuatro arquitecturas fuertemente relacionadas: arquitectura de negocio, arquitectura de datos, arquitectura de aplicación y arquitectura tecnológica.
- ✓ Architecture Development Method (ADM), metodología para el desarrollo de la arquitectura, que es el núcleo de TOGAF y describe paso a paso el desarrollo de una arquitectura empresarial para sistemas relacionados con Tecnologías de la Información.
- ✓ La Arquitectura Base TOGAF, compuesta por un Modelo de Referencia Técnico (TRM), el Open Group Standards Information Base (SIB) y el Building Blocks Information Base (BBIB).
- ✓ El repositorio de recursos de TOGAF, conjunto de herramientas y técnicas disponibles para su uso cuando la implantación de TOGAF o de su metodología-ADM lo requiera (vistas de arquitecturas, escenarios de negocio, ADML, casos de estudio, etc.).

TOGAF no describe, sino prescribe y en su última versión, ya se adhiere al IEEE 1471-2000 con su taxonomía de vistas. El uso de un marco arquitectural como TOGAF aumenta la velocidad y simplifica el proceso de desarrollo de una arquitectura software, asegurando una mejor cobertura de la solución diseñada y facilitando que la arquitectura seleccionada permita un futuro crecimiento en respuesta a las necesidades del entorno.

Por lo que se refiere a la metodología de desarrollo de arquitecturas, ADM, está enfocada para el desarrollo de arquitecturas empresariales y cubre diferentes vistas arquitecturales.

La estructura básica de su ciclo de vida lo puede ver a detalle en el Anexo N° 03, con un modelo de ciclo de vida que trata el desarrollo del proceso desde el inicio hasta la implantación o el mantenimiento. Esta estructura es iterativa y adaptable en función de la madurez del sistema, pero marca unas pautas muy concretas dentro de la generalidad de una metodología a este nivel, indicando los resultados que se deben obtener en cada fase y siendo independiente de lenguajes y herramientas de modelado.

El núcleo de esta metodología está en las fases que van de la A a la D; así se recoge la visión de la arquitectura en la fase A y se define cómo recoger las competencias de todos los actores del sistema en las fases B, C y D. Es en estas fases donde se toman las primeras decisiones significativas de diseño y se identifican un conjunto de vistas asociadas a los actores que son las que posteriormente se modelarán en el proceso de desarrollo del sistema:

- ✓ Vistas de la arquitectura de negocio, que abordan las competencias de los usuarios y gestores.
- ✓ Vistas de la arquitectura de datos, que cubren los objetivos fijados por los diseñadores y administradores de las bases de datos y los ingenieros de sistemas responsables del desarrollo e integración de los componentes de datos del sistema.
- ✓ Vistas de la arquitectura de aplicaciones, utilizadas por los ingenieros de software y sistemas responsables del desarrollo e integración de los diversos componentes software del sistema.

- ✓ Vistas de la arquitectura tecnológica, que abordan los objetivos de los encargados de compras, personal de operaciones y administradores y gestores del sistema.

La metodología de referencia presentada en este capítulo se ha nutrido del análisis de las experiencias previas de arquitecturas software, tanto genéricas como específicas en el ámbito académico. Así, la metodología tiene como bases principales TOGAF del Open Group. Esta metodología de referencia que se propone **recomienda, pero no obliga** a seguir los pasos indicados, buscando siempre la practicidad y considerando la arquitectura el medio y no el fin.

Durante todo el capítulo se ha separado la arquitectura de referencia en dos partes claves: la información y la presentación, haciendo que las estructuras de información y las relaciones que se establecen entre las diferentes vistas no dependan de ninguna herramienta de visualización o presentación de esta información, diferenciando claramente el qué (información) se propone hacer del cómo (forma) se sugiere hacerlo. Las herramientas de representación de las vistas que se han utilizado durante la tesis es el Microsoft Visio 2016, IBM Rational Rose 7.0 y Erwin 7.2.

6.2. Descripción de las Fases de la Arquitectura Propuesta

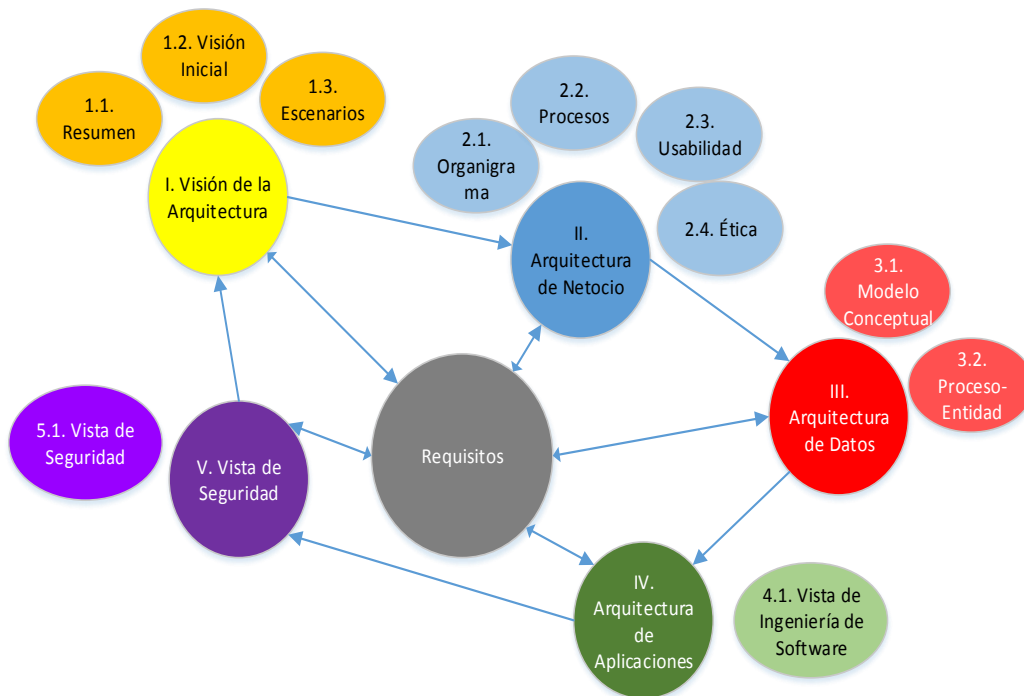
La metodología ofrecida por TOGAF (The Open Group Architecture Framework) para el desarrollo de arquitecturas, más conocido como ADM (Architecture Development Method) ya ha sido resumida a través del Gráfico 6.1, la cual no es más que un conjunto de vistas recomendadas por el estándar de la IEEE 1471-2000. En esta tesis se diseña una Arquitectura tomando como referencia sólo las primeras 4 fases comunes y más críticas; además en estas fases es donde se puede agregar valor con el único objetivo de estandarizar y recomendar unas vistas que deben ser contempladas antes de iniciar la fase de desarrollo de cualquier sistema de gestión académica de una Universidad.

Por lo tanto, empezaremos brindando un marco de referencia para el diseño de la Arquitectura de Software aplicado al Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana, que se representa en el Gráfico 6.2, y que se divide en las siguientes fases:

- I. Visión de la arquitectura.** Aquí se da a conocer a los actores del sistema, los procesos en los que participan dichos actores del sistema, los objetivos del sistema, y los escenarios presentes del sistema de gestión académica.
- II. Arquitectura de negocio.** Diseñada para aclarar los requisitos y las competencias de los usuarios finales y de los encargados de la gestión académica; además se definen las vistas que nos permiten tener una clara idea del organigrama y del personal involucrado y sus funciones, los procesos de negocio, la usabilidad y el aspecto ético.
- III. Arquitectura de datos.** Aquí se modelan las necesidades de los administradores de Bases de Datos, además durante la conceptualización de datos, se desea diseñar las relaciones entre las entidades identificadas.
- IV. Arquitectura de aplicaciones.** Es esta fase se documentan los modelos o paradigmas de programación, lenguajes de programación, entornos de desarrollo integrados (IDE) que usarán los ingenieros de software, es por ello que se describirá solamente la vista de ingeniería software.
- V. Vista de seguridad.** Esta es una vista en la cual se centra la atención directamente a los actores del sistema, en la funcionalidad que podrá realizar cada uno de ellos, por lo cual se pueda definir el nivel de seguridad apropiado para cada uno de ellos.

Un mayor detalle de la arquitectura con las vistas propuestas para cada fase se presenta en el siguiente esquema:

Gráfico 1. Diseño de la Arquitectura Propuesta



Fuente: Elaboración Propia en Visio 2016

6.2.1. Visión de la Arquitectura

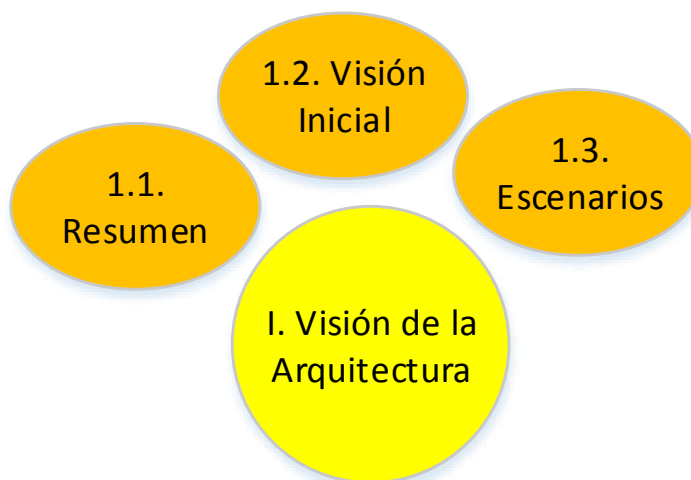
La visión de la arquitectura es el primer paso para el diseño de la arquitectura de un sistema; aquí aprovecharemos en hacer evidente las ventajas que tiene un diseñador e ingeniero de software y de sistemas.

Como ya se dijo antes lo principal es dar a conocer las funciones y los requisitos de los actores participantes del sistema enmarcado en los objetivos a cumplir.

Es por eso que, durante el desarrollo de esta fase, el arquitecto software del sistema deberá:

- ✓ En primer lugar, brindar un resumen de toda la información referida al sistema.
- ✓ Hay que identificar a los actores del sistema y brindar una descripción de sus competencias, requisitos y objetivos a cumplir.
- ✓ Se debe mostrar los escenarios de uso del sistema.

Gráfico 2. Visión de la Arquitectura



Fuente: Elaboración Propia en Visio 2016

6.2.1.1. Resumen:

Es en esta sección se ofrece un resumen de lo que se considerará al definir y diseñar la arquitectura del sistema. Se especifican qué vistas van a ofrecerse para dar una idea clara del sistema a cada tipo de usuario, con qué nivel de detalle va a ser tratada cada vista y el horizonte temporal con el que se está planificando la arquitectura de dicho sistema. Un ejemplo de Visión de la Arquitectura para el Sistema de Gestión Académica es:

| ÁMBITO DEL PROYECTO | SISTEMA DE GESTIÓN ACADÉMICA DE LA UNIVERSIDAD NACIONAL DE LA AMAZONÍA PERUANA | |
|-------------------------------------|--|------------------|
| | TRATADA | NIVEL DE DETALLE |
| I. Visión de la Arquitectura | X | |
| 1.1. Resumen | | Medio-Alto |
| 1.2. Visión Inicial | | Medio-Alto |
| 1.3. Escenarios | | Medio-Alto |
| II. Arquitectura de Negocio | X | |
| 2.1. Vista de Organigrama | | Medio |
| 2.2. Vista de Procesos | | Medio-Alto |
| 2.3. Vista de Usabilidad | | Medio-Alto |
| 2.4. Vista de Ética | | Medio |
| III. Arquitectura de Datos | X | |
| 3.1. Vista de Modelo Conceptual | | Medio-Alto |

| | | |
|---|---|------------|
| 3.2. Vista de Proceso-Entidad | | Medio-Alto |
| IV. Arquitectura de Aplicaciones | X | |
| 4.1. Vista de Ingeniería de Software | | Medio |
| V. Vista de Seguridad | X | |
| 5.1. Vista de Seguridad | | Medio |

Fuente: Elaboración Propia

6.2.1.2. Visión Inicial:

En esta sección se identifica a todos los usuarios que interactuarán con el sistema de gestión académica. Posteriormente se evaluará las funciones y responsabilidad de cada uno para poder a proceder a la recolección de requisitos, sus competencias y sus objetivos.

Los grupos de actores más comunes en el área educativo universitario que ya se han identificado son:

- ✓ Alumno: persona que requiere del servicio universitario.
- ✓ Docente: persona dedicada a la enseñanza académica.
- ✓ Cajero: persona que es responsable de los cobros por diferentes conceptos.
- ✓ Director de Escuela: persona encargada de dirigir los procesos académicos en cada Escuela.
- ✓ Coordinador de Horarios: persona encargada de gestionar los horarios en cada Facultad.
- ✓ Soporte Técnico: persona que está a cargo de los programas y/o dispositivos para garantizar su correcto funcionamiento sin interrupción.
- ✓ Administrador Académico: persona encargada de las actividades académicas y administrativas dentro del Vicerrectorado Académico.
- ✓ Administrador del Sistema: persona autorizada para tener el control total de la información y todos los dispositivos informáticos de la institución.

Gráfico 3. Galería de Actores



Fuente: Elaboración Propia en Visio 2016

Requisitos de usuario: Los requisitos de usuario varían sensiblemente en función del tipo de usuario o actor al que se asocian, aunque siempre se dan unos bloques comunes de datos:

- Datos personales: nombres, apellidos, dirección, teléfono, sexo, fecha de nacimiento, estado civil, etc.
- Datos generales: código, correo electrónico, usuario, contraseña, foto, lugar de procedencia, lugar de nacimiento, fecha de ingreso, nivel de estudios, situación, etc.
- Comunicación con el sistema: experiencia previa con tecnologías, aceptación del usuario y facilidad de uso de diferentes modos para la introducción de datos, modos preferidos para la recepción de mensajes del sistema, etc.

Competencias: Las competencias son los intereses relativos al desarrollo del sistema, su funcionamiento u otros aspectos que son críticos o importantes de alguna manera para uno o más actores. Así pues, cada actor tendrá sus competencias que después se documentarán en vistas. Ejemplos de competencias

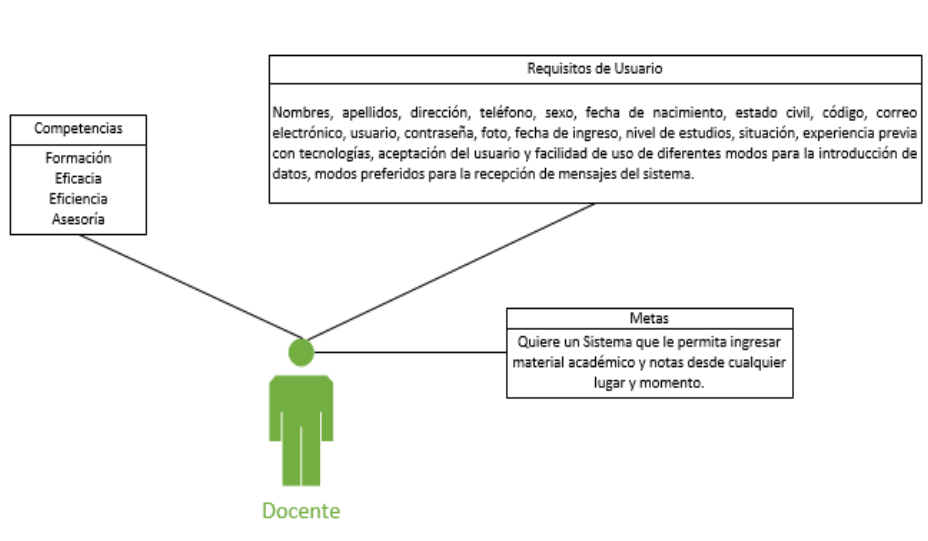
son para cada tipo de usuario o actores del Sistema de Gestión Académica UNAP:

| Tipo de Usuario | | Competencia |
|-----------------------|---------|---|
| Alumno | | Aprendizaje, formación, ayuda, ... |
| Docente | | Formación, eficacia, eficiencia, Asesoría, ... |
| Cajero | | Satisfacción del usuario, ética, eficiencia, ... |
| Jefatura Académica | | Configuración, ayuda, gestión de cambios, ... |
| Soporte Académico | Técnico | Mantenimiento, comunicaciones, escalabilidad, rendimiento, gestión de errores, ayuda... |
| Administrador Sistema | del | Funcionalidad, comunicaciones, seguridad, instalación, usabilidad, interoperabilidad, datos, formación, mantenimiento, escalabilidad, rendimiento, uso energético, gestión de errores, ayuda... |

Fuente: Elaboración Propia

En el siguiente Gráfico, se muestra a modo de ejemplo cómo se está caracterizando un usuario y se le está vinculando sus competencias.

Gráfico 4. Ejemplo de asignación de competencias



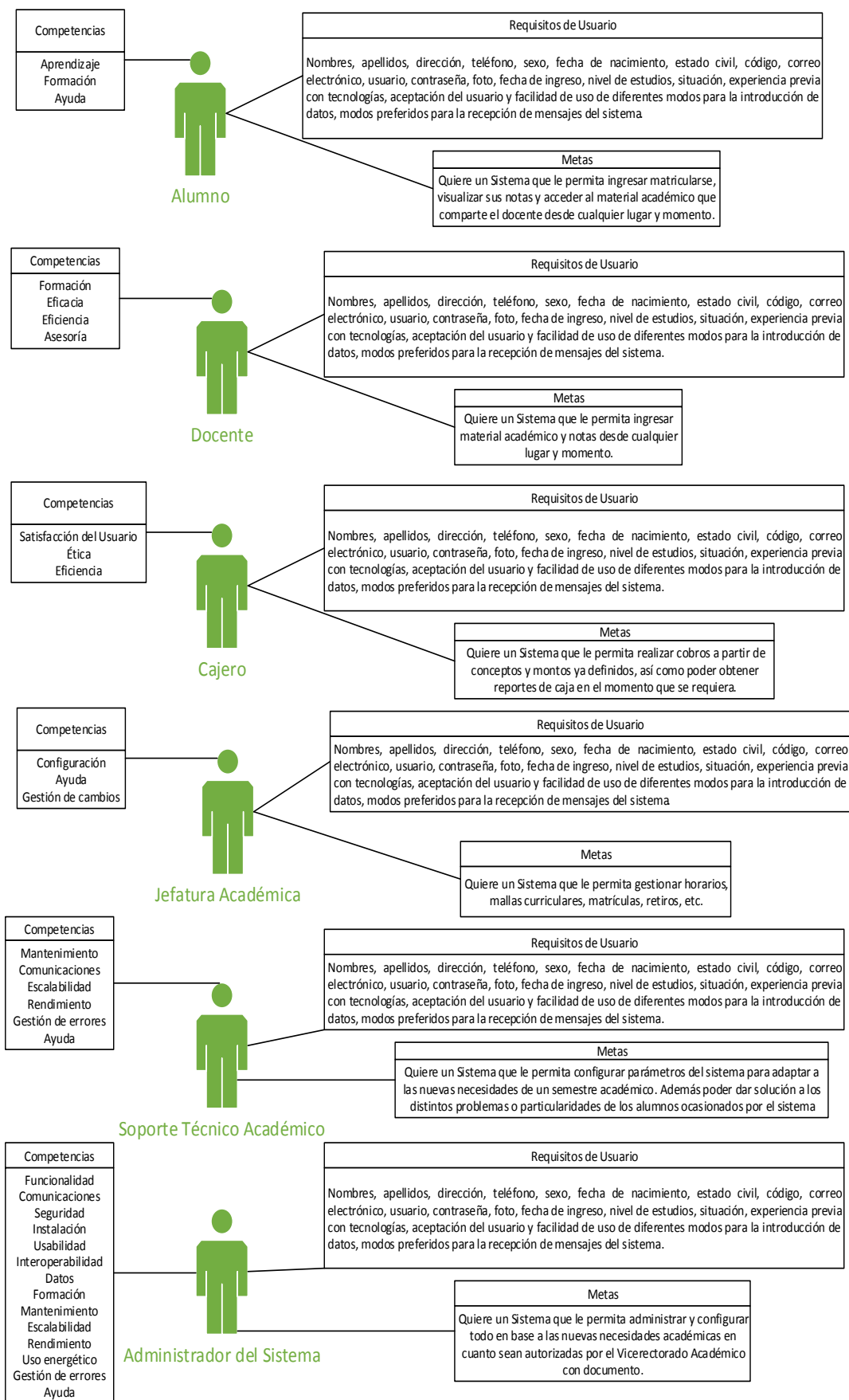
Fuente: Elaboración Propia en Visio 2016

Metas: Ahora es momento de describir las metas de la arquitectura propuesta, deben estar alineados a los objetivos definidos por los actores. Los objetivos, deberán convertirse en metas SMART (Specific, Measurable, Actionable, Realistic, Time-bound). La representación gráfica de esta visión inicial se debe reflejar en un documento, teniendo como elementos principales a los actores. Los actores indicados anteriormente tendrán propiedades asociadas a ellos dependiendo de la responsabilidad y función de cada uno de ellos.

Las propiedades de los actores no son más que lo que se requiere de los usuarios a nivel de competencias, objetivos y/o metas. Estas propiedades deberán ser parte de la gráfica en la que los actores se mostrarán relacionándose así mismo los requisitos de los usuarios, las competencias y los objetivos o metas que se hayan definido.

A continuación, se muestra un ejemplo de la vista inicial:

Gráfico 5. Ejemplo de Vista Inicial



Fuente: Elaboración Propia en Visio 2016

6.2.1.3. Escenarios:

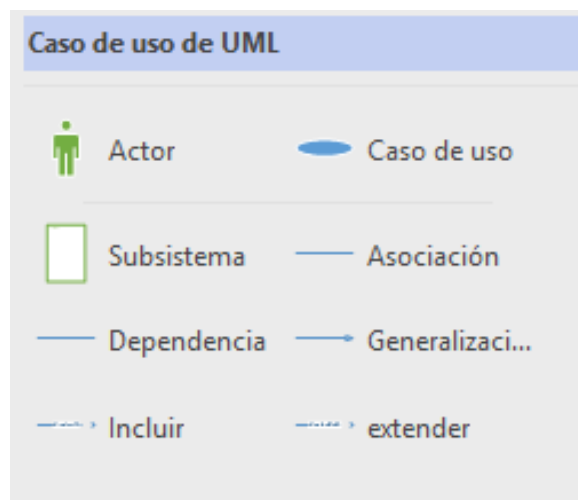
Dentro de esta fase se ha considerado utilizar los Escenarios, ya que se considera que la visión arquitectural es necesario describir:

- ✓ Un proceso de negocio.
- ✓ El entorno donde se está contextualizando el negocio.
- ✓ Los actores y otros elementos que participan dentro del escenario.
- ✓ El resultado deseado tras terminada la ejecución de manera correcta.

Un buen escenario de negocio nos permite representar una necesidad de negocio de importancia, permitiendo a los proveedores hacer entender a los clientes el valor que tiene una aplicación desarrollada a medida de sus necesidades. Esto permite facilitar la manera como se comunican los proveedores de software con la problemática del cliente y las soluciones informáticas ofrecidas por las que las desarrollan e implementan.

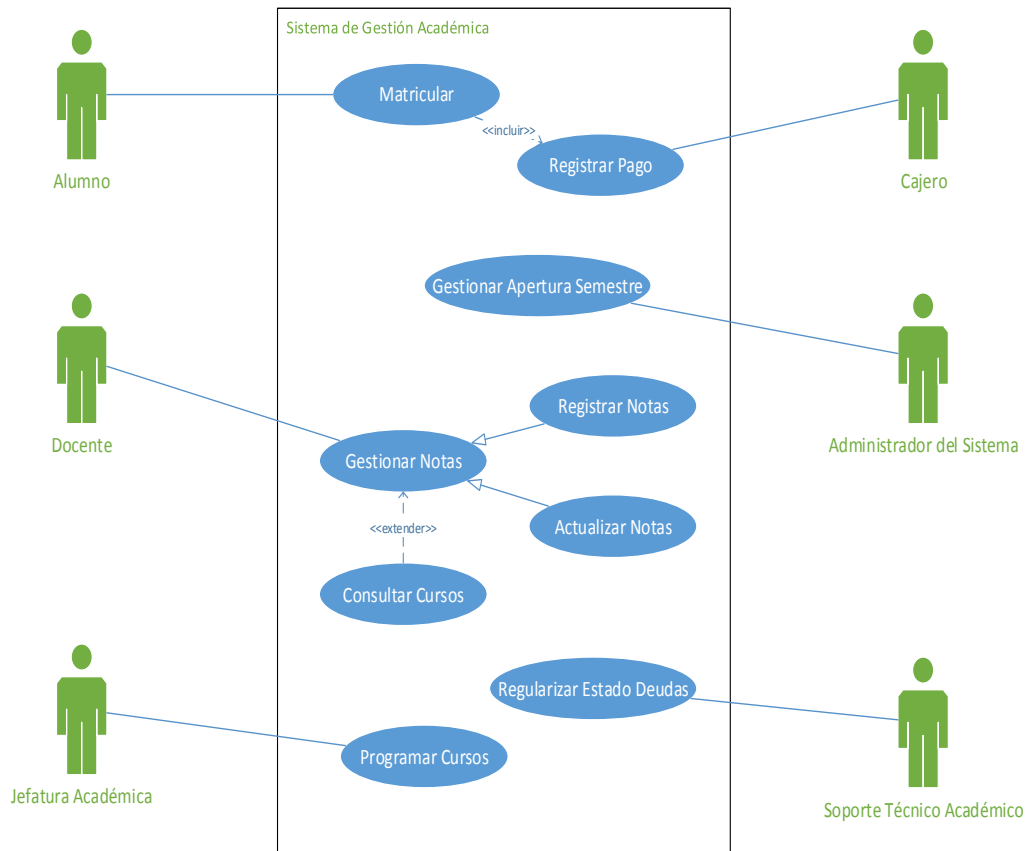
Representación: Para representar los escenarios se diseña los casos de uso UML, utilizando una galería de símbolos que mostramos a continuación:

Gráfico 6. Galería de símbolos para casos de uso UML



Fuente: Visio 2016

Gráfico 7. Ejemplo de Casos de Uso

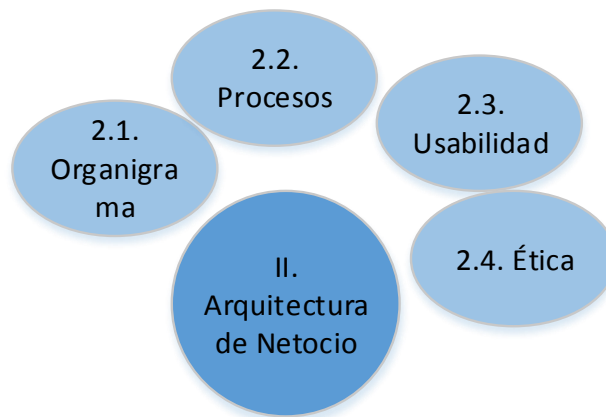


Fuente: Elaboración Propia en Visio 2016

6.2.2. Arquitectura de Negocio

La Arquitectura de Software debe incluir el contexto del negocio, siguiendo las pautas plasmadas en algunos documentos como el estándar IEEE 1471-2000 y el marco de referencia TOGAF. Esta arquitectura cubre los requisitos y las competencias de todos los actores involucrados en los procesos de gestión académica. Con respecto a la Arquitectura de negocio, se definen las vistas más importantes que nos permitirán tener una clara idea del organigrama del personal involucrado, los procesos, la usabilidad y ética. Además, es sabido que la arquitectura de negocio que incluye elementos propios del dominio del negocio aparezcan en las vistas posteriores de esta arquitectura. Si hay nuevos requerimientos y por ende nuevos procesos del negocio esto conllevará a trabajo adicional.

Gráfico 8. Vistas de la Arquitectura de Negocio



Fuente: Elaboración Propia en Visio 2016

6.2.2.1. Vista de Organigrama:

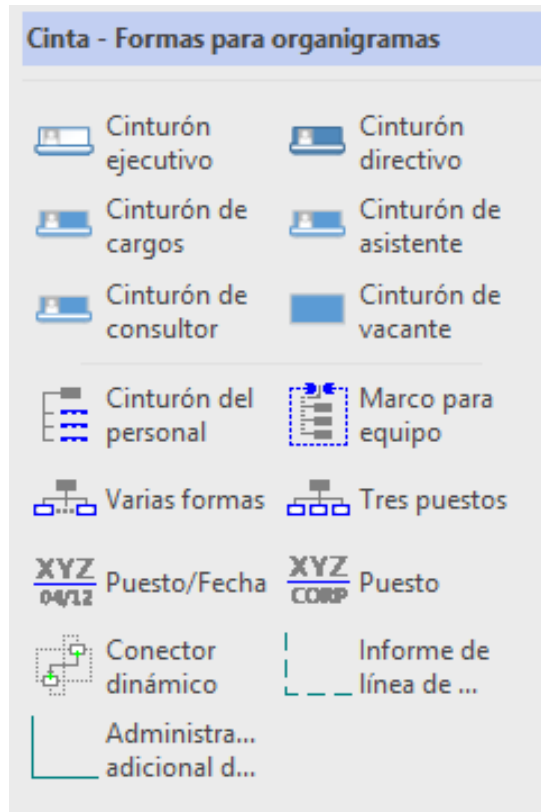
La vista de organigrama facilita la documentación de las estructuras organizativas que proveen o reciben servicios académicos y administrativos de la Universidad, los actores asociados a cada una de ellas, su jerarquía y las relaciones existentes entre ellas. Además, es útil el conocer la distribución física de los actores que están involucrados en el sistema, dado que ello puede servir para detectar problemas de eficiencia o distribuir de una manera más racional los recursos y/o las relaciones existentes entre los actores.

En esta vista aparecen los actores identificados en la fase previa, que pueden ser relacionados entre sí mediante bien enlaces de jerarquía bien mediante enlaces de comunicación. Además, los diferentes actores pueden ser agrupados con cuadros organizativos, cuando están trabajando en una misma empresa o departamento o por ubicación física, cuando actores de las mismas o diferentes organizaciones se encuentran en un mismo espacio físico.

Representación: Para representar este punto de vista se usarán los actores identificados previamente, además se utilizará la

galería de símbolos para organigramas que mostramos a continuación:

Gráfico 9. Símbolos específicos de la vista para organigramas



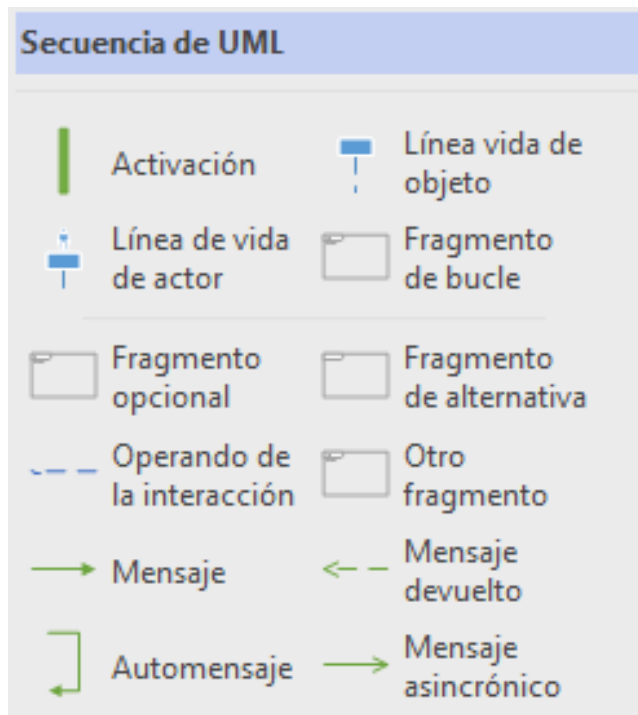
Fuente: Visio 2016

6.2.2.2. Vista de Procesos:

La vista de procesos muestra todos los procesos de los actores involucrados especificando qué se hace, sobre qué y cuál es el resultado de esa interacción. De tal manera que se logra tener una visión bastante clara de qué es lo que hace realmente el sistema a detalle.

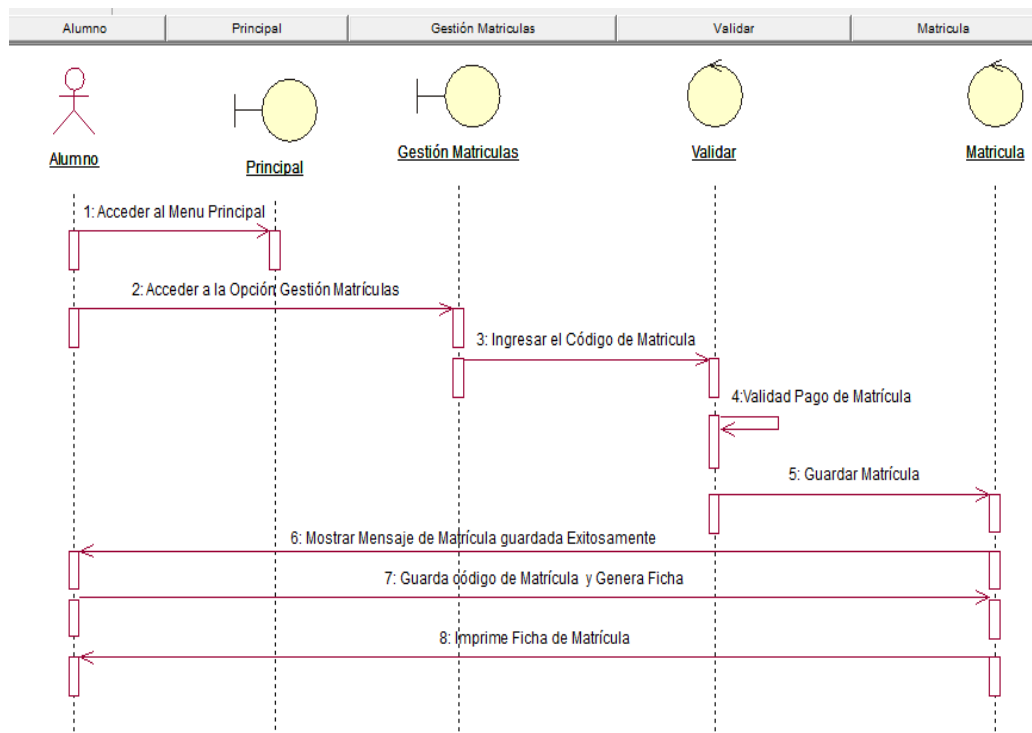
Representación: Existen muchas formas en las que se puede apreciar las interacciones entre actores y los procesos; es por ello que aquí se ha elegido mostrar a través del diagrama de secuencia, lo cual mostramos a continuación:

Gráfico 10. Símbolos específicos para la vista para procesos



Fuente: Visio 2016

Gráfico 11. Ejemplo de vista de procesos



Fuente: Elaboración Propia en IBM Rational Rose 7.0

6.2.2.3. Vista de Usabilidad:

La usabilidad está referida a la facilidad que tiene el usuario para aprender a operar, ingresar datos, interpretar resultados de un sistema. Algunos de los aspectos a tener en cuenta para mejorar la usabilidad son: Disminución del tiempo para realizar cada una de las operaciones, minimizar la cantidad de fallas en tiempo de ejecución, disminuir el tiempo utilizado para aprender a utilizar el sistema y de esta manera lograr aumentar la satisfacción del usuario del sistema. Cabe recalcar que los aspectos de usabilidad del sistema y del entorno no solo afectan a los actores de tipo persona, sino también a los actores de tipo hardware y software. Es por ello que se debe considerar preparar las interfaces de usuario y otros mecanismos de uso para facilitar la administración y respuesta por parte del sistema, como son directivas de contraseñas, reducción de la capacidad de uso de memoria, desarrollo de prototipos, manuales de usuario, gestión de los accesos, tratamiento de errores, optimización algorítmica y el uso de estándares o buenas prácticas.

Representación: Para representar este punto de vista se utilizará una matriz, donde se incluirá los aspectos de usabilidad considerados, los actores ya sean personas, hardware o software participantes, los procesos, la acción a realizar para mejorar dicho aspecto y finalmente la recomendación indicando buenas prácticas o estándares reconocidos. A continuación, mostramos la matriz:

| ASPECTO DE USABILIDAD | ACTOR | PROCESO | ACCIÓN | RECOMENDACIÓN O ESTÁNDAR |
|-----------------------|-------|---------|--|---|
| Ayuda y Documentación | Todos | Todos | Manual de Usuario, Manual de Instalación y Configuración, Videos Tutoriales y Teléfono del Área de Soporte del Sistema Académico | Archivos con extensión: hlp, html y mp4 |

| | | | | |
|---|---------------------------|-----------|---|----------------------------------|
| Control del sistema por parte del usuario | Todos | Todos | El sistema hace lo que se le solicita | |
| Dispositivos | Todos | Todos | El tamaño de la pantalla del dispositivo móvil ha de ser mayor de 8x4 cm y táctil. | |
| Entrenamiento | Docente | Matrícula | Capacitación Grupal de 30' | |
| Entrenamiento | Administrativo | Todos | Capacitación Grupal de 1 hora | |
| Facilidad de uso | Todos | Todos | Cualquier operación con menos de 4 clicks | |
| Interfaces de usuario | Todos | Todos | Cumplimiento Norma W3C | Estándar W3C |
| Interoperabilidad | Todos | Todos | Uso de XML | XML |
| Mecanismos de acceso | Todos | Todos | Uso de Usuario y Contraseña | SSL, TLS, HTTPS y Encriptado MD5 |
| Reducción carga memoria | Todos | Todos | Generación de datos automáticamente y permite recuperar la contraseña cuando el usuario no lo recuerda. | |
| Tratamiento de errores | Administrador del Sistema | Todos | Documentación de los errores, alertas a soporte y capacidad de uso de versión antes del error. | |

Fuente: Elaboración Propia

6.2.2.4. Vista de Ética:

Para la vista de ética, empezamos diciendo que ya desde hace años el estado está asumiendo un papel más responsable en el sector de educación, ya que se evidencia una alta preocupación por el nivel de aprendizaje y la poca preocupación de brindar calidad de educación al estudiante por parte de las instituciones superiores, como los institutos tecnológicos, pedagógicos y universidades. En esta investigación se ha elegido la Universidad Nacional de la Amazonía Peruana, siendo la más importante en nuestra ciudad de Iquitos; pero tal vez la que más preocupa por aspectos como lo es el ambiente político, la falta de ética y la poca moral de algunas personas involucradas en la gestión académica de la universidad. Es por ello que se ha considerado algunos aspectos éticos a considerar para el Sistema de Gestión

Académica de la UNAP, que serán plasmados como principios aplicables al mismo. Estos principios han sido analizados y recogidos de documentos elaborados por diferentes organizaciones internacionales reconocidas dentro de este ámbito, como lo son la Autorización, Competencia para Uso, Relación Docente-Alumno, Responsabilidad del Docente, Rol del Alumno, Consentimiento del Alumno, Confidencialidad, Exactitud de la Información, Seguridad del Acceso a la Información.

Representación: Para representar la vista de ética, se usará una matriz en función de los objetivos del sistema (matrícula, notas, programación de cursos, gestión de deudas, etc.). Esta matriz será utilizada como plantilla para incluir los principios seleccionados y también puede incluir otros principios específicos del sistema, posteriormente se debe asociar a estos principios las acciones a realizar que nos permita garantizar el cumplimiento. A continuación, la plantilla:

| PRINCIPIO | ACCIONES A REALIZAR | CONSIDERACIONES |
|---------------------------------------|----------------------------|------------------------|
| Autorización | | |
| Competencia para Uso | | |
| Relación Docente-Alumno | | |
| Responsabilidad del Docente | | |
| Rol del Alumno | | |
| Consentimiento del Alumno | | |
| Confidencialidad | | |
| Exactitud de la Información | | |
| Seguridad del Acceso a la Información | | |

Fuente: Elaboración Propia

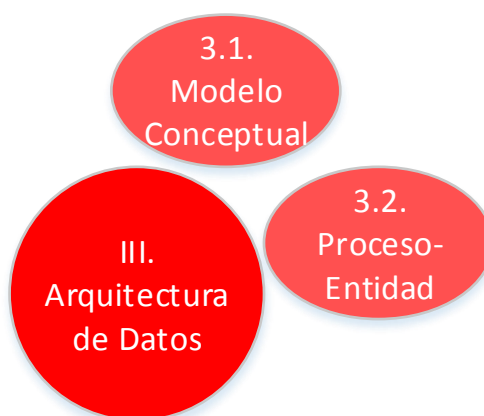
6.2.3. Arquitectura de Datos

Aquí es obvio que es competencia de los diseñadores y administradores de base de datos, que se encargarán de documentar el modelo conceptual de datos y las relaciones entre entidades.

Se debe tener claro que la intención aquí no es pretender llegar al nivel lógico o físico de los sistemas de almacenamiento, sino concentrarse en la definición de las entidades, sus atributos y sus relaciones entre ellos.

En esta vista es donde se detectará los posibles errores de relación entre entidades, datos faltantes, datos que no se están utilizando y estructuras de datos faltantes.

Gráfico 12. Vistas de la Arquitectura de Datos



Fuente: Elaboración Propia en Visio 2016

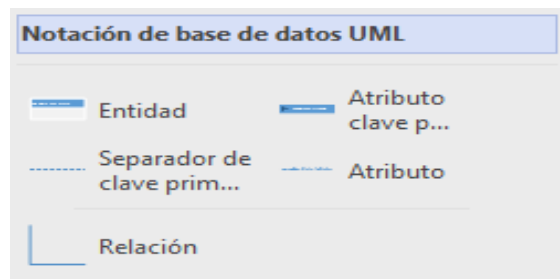
6.2.3.1. Vista del Modelo Conceptual:

En la vista del modelado conceptual de datos se identifica a las entidades, sus relaciones y las relaciones entre ellas. Para ello el analista debe tener claro que una entidad es en realidad cualquier cosa existente en el mundo real (concreto o abstracto). Ejemplo: Una persona, concepto o evento. Y las propiedades son las características específicas de una entidad que permitirán identificar a un objeto participante del mini mundo que es este sistema. Cabe recalcar que este modelo servirá como base para

la futura creación de la base de datos con la cual interactuará el sistema.

Representación: Para representar este punto de vista se usará la galería de símbolos para base de datos UML que mostramos a continuación:

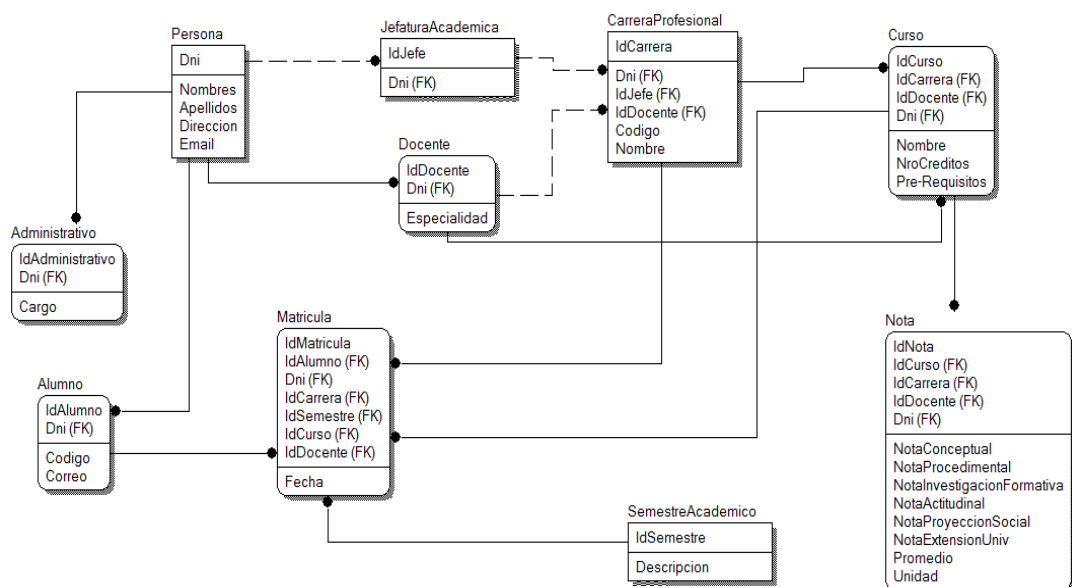
Gráfico 13. Galería de símbolos para la vista conceptual de datos



Fuente: Visio 2016

A continuación, se brinda un ejemplo de vista del modelo conceptual de datos elaborada en Erwin 7.2.

Gráfico 14. Ejemplo de modelado conceptual de datos



Fuente: Elaboración Propia en Erwin 7.2.

6.2.3.2. Vista Relación Negocio-Entidad:

En esta vista el objetivo es poder brindar una clara idea de cómo se relacionan las entidades identificadas en la vista conceptual de datos con el conjunto de procesos identificados en la vista de arquitectura de negocio. De tal forma, que garantizamos coherencia en la estructura de datos a implementar posteriormente y esto nos facilita identificar error en la etapa de diseño; que nos obligaría posiblemente a eliminar entidades innecesarias o incluirlas como parte de otras según su necesidad en el negocio.

Representación: Para representar este punto de vista en esta oportunidad se recomienda usar una tabla donde relacionaremos las entidades con los procesos, plasmando que operaciones CRUD (Create, Read, Update & Delete) realizarán cada uno de los procesos sobre las entidades. Un ejemplo de ello lo mostramos a continuación:

| PROCESOS | GESTIÓN DE MATRÍCULAS | GESTIÓN DE NOTAS | GESTIÓN DE PROGRAMACIÓN DE CURSOS | GESTIÓN DE DEUDAS |
|--------------------|-----------------------|------------------|-----------------------------------|-------------------|
| ENTIDADES | | | | |
| Alumno | R | R | | R |
| Docente | R | | CRU | |
| Administrativo | R | R | R | R |
| JefaturaAcademica | | | CRUD | R |
| Matricula | CRU | R | | CRU |
| CarreraProfesional | R | R | R | R |
| SemestreAcademico | R | R | CR | R |
| Curso | R | R | R | |
| Nota | | CRU | | |

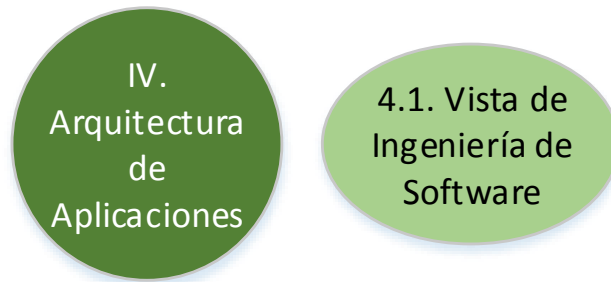
Fuente: Elaboración Propia

6.2.4. Arquitectura de Aplicaciones

En estas vistas se documenta las capacidades de los especialistas en software o ingenieros de sistemas, concentrando la atención a la vista de ingeniería de software (programación en capas, lenguajes de programación y entornos de desarrollo integrados) y la de estándares e interoperabilidad

de aplicaciones. El objetivo de esta fase es definir los principales tipos de aplicaciones a desarrollar y que darán solución a la problemática del negocio a través del ingreso, procesamiento, almacenamiento y salida de los datos. Además, aquí es donde se evidencia las competencias de los ingenieros de software para aplicar la programación orientada a objetos, por capas, portabilidad, interoperabilidad, migración y multiplataforma de ser necesario.

Gráfico 15. Vistas de la Arquitectura de Aplicaciones



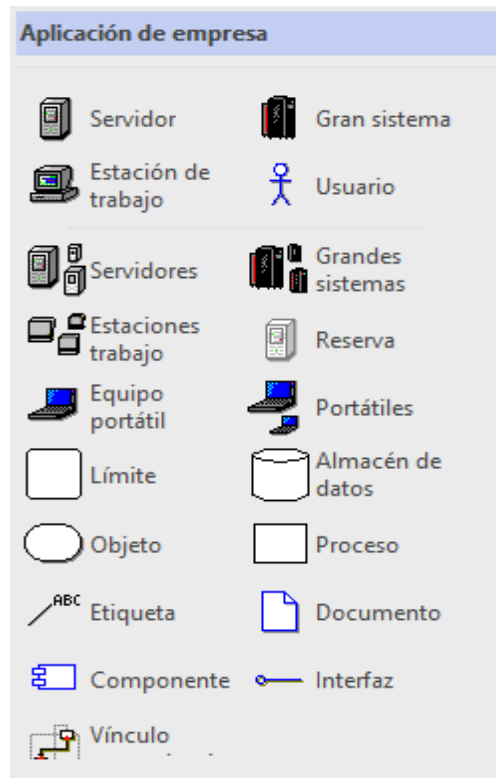
Fuente: Elaboración Propia en Visio 2016

6.2.4.1. Vista de Ingeniería de Software:

La vista de ingeniería de software como ya hemos mencionado antes, se centra en los aspectos relacionados al desarrollo del software, aplicando buenas prácticas para reducir tiempo de desarrollo, minimizar el esfuerzo, reducir la cantidad de líneas de código a escribir, disminuir los riesgos, etc. Aquí es donde plasmamos lo necesario que en un futuro nos facilitará, como la reutilización de código y mantenimiento del sistema.

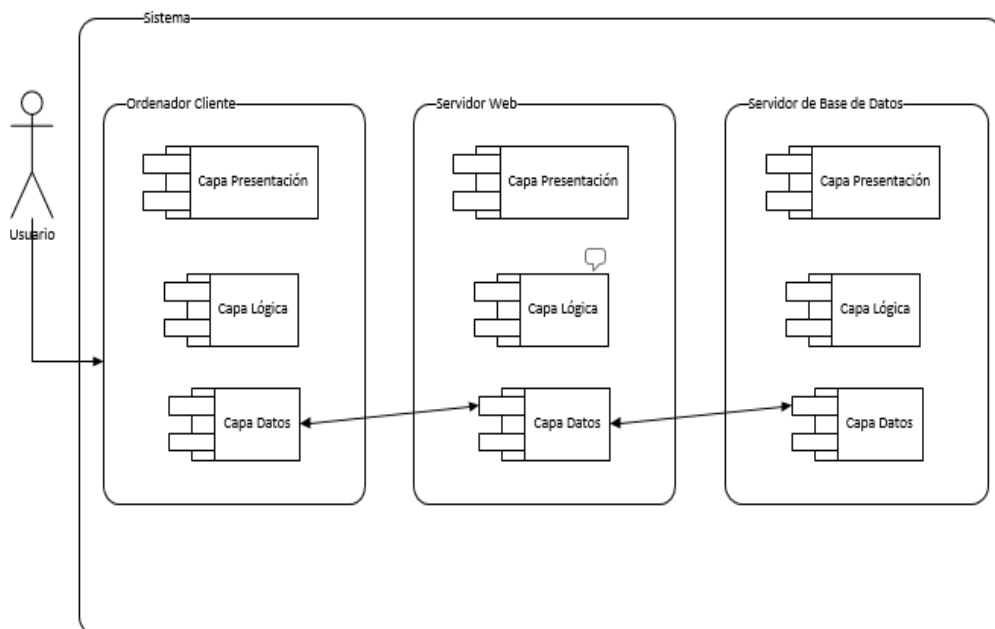
Representación: Para representar este punto de vista se utilizará la galería de símbolos para ingeniería de software, que mostramos a continuación:

Gráfico 16. Galería de símbolos para la vista de ingeniería de software



Fuente: Visio 2016

Gráfico 17. Ejemplo de vista de ingeniería de software

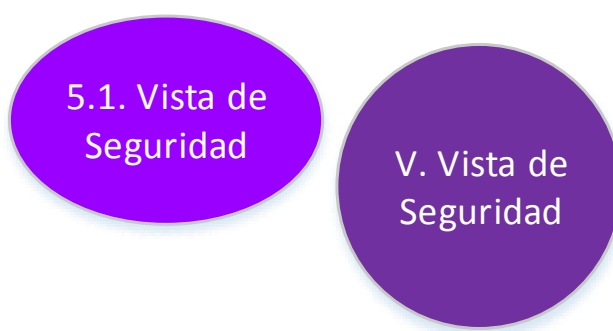


Fuente: Elaboración Propia en Visio 2016

6.2.5. Vista de Seguridad

En esta fase se podrá tratar las competencias que afectará a los actores involucrados y a toda la arquitectura en general relacionadas a la seguridad. Para ello se responderá a un conjunto de interrogantes relacionadas a la autenticación del usuario, registro todas las operaciones que afectan los datos, permisos y denegación de accesos, gestión de las cuentas de usuario, encriptación y/o el uso de certificados de seguridad.

Gráfico 18. Vistas de Seguridad



Fuente: Elaboración Propia en Visio 2016

Para poder enfocarnos en la arquitectura de seguridad de éste sistema, primero debemos saber es qué se está protegiendo, qué importancia tiene para los usuarios, cuáles son los riesgos y amenazas.

En esta oportunidad se está considerando que la no disponibilidad de datos o servicios en algún momento requerido, la no confidencialidad de los datos, la inconsistencia de datos y el bajo control de acceso a los recursos o servicios académicos a nivel de software y hardware son amenazas que ocasionarían un gran impacto en la gestión académica de la Universidad Nacional de la Amazonía Peruana.

Para responder a estas amenazas, la idea en esta vista es documentar una lista de criterios y/o servicios a considerar por el sistema.

- A través de la autenticación identificar específicamente a la persona que ingresa al sistema, control de ingresos, por periodos, restricciones a cierto tipo de contenido, acceso limitado a servicios o funciones del sistema.
- Configuración de la auditoría de la base de datos, para lograr tener un registro de cada operación CRUD, es decir registro de cada acción que ejecuta el usuario a través del sistema y genera manipulación de datos; para ello se considerará registrar además quién realizó la acción, en qué fecha y hora lo hizo; y en que terminal lo realizó.
- Considerar un módulo que permita configuración los permisos y denegación de acciones de forma dinámica a los distintos usuarios del sistema.
- Se debe tener en cuenta que el sistema debe tener un módulo que permita consultar, crear, actualizar y dar de baja cuentas de usuario tanto a nivel del sistema, como a nivel de base de datos.
- Adicionalmente a lo anterior considerar no solo la seguridad lógica sino también la física para evitar la posible falla o pérdida mal intencionada de los equipos donde residen los datos.
- Documentar como se gestionará la forma de respaldar la base de datos y los momentos en que se realizarán. No olvidar considerar las copias físicas ante posibles accidentes o desastres.
- La encriptación de datos hoy en día se ha hecho más que necesario, para ello se debe investigar, evaluar y elegir los estándares oportunos para un Sistema de información como éste.

- Además, garantizar la comunicación segura, a través de los protocolos o certificados SSL, TLS y HTTPS.
- Finalmente, no descartar el uso de firewalls, que nos permitirán evitar los accesos de personas no autorizadas y el uso de herramientas para detección de intrusos, plasmando las posibles acciones que se realizarán al momento de detectarlos.

Representación: Para representar este punto de vista se hará una tabla, donde en la primera columna aparecerán todas las funcionalidades relacionadas a la seguridad, y en las demás columnas se dará respuesta a las interrogantes: ¿Quién?, ¿Sobre qué?, ¿Cómo? y ¿Cuándo?

CAPÍTULO VII: CONCLUSIONES

A Nivel del Problema:

Se logró demostrar que la Aplicación de la Arquitectura de referencia propuesta (basada en TOGAF) cumple con mejorar el Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana. Según lo demuestran las Tablas N° 7, 8 y 9.

A Nivel de Objetivo:

Se logró Diseñar una Arquitectura de referencia basada en el marco de trabajo TOGAF para el Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana - 2019. Según lo demuestran las páginas del 47 al 72.

A Nivel de Hipótesis:

Una Arquitectura de referencia basada en TOGAF permitió mejorar el Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana. Según lo demuestran las Tablas N° 10,11 y 12.

Conclusión Final:

Es importante que la Arquitectura del Software esté diseñada antes de empezar el desarrollo y está demostrado a partir de esta investigación que es necesario tomarle muy en serio y tiempo al diseño de una buena Arquitectura de Software, además se sabe que esta propuesta personalizada del marco de referencia TOGAF, servirá como metodología específica que permitirá la descripción arquitectural de cualquier Sistema de Gestión Académica.

CAPÍTULO VIII: RECOMENDACIONES

1. A los administrativos y/o docentes que conforman la Comisión del Sistema de Gestión Académico de la Universidad Nacional de la Amazonía Peruana investigar más a fondo otras arquitecturas de software, tanto genéricas como específicas en el área de Educación.
2. Al personal de soporte técnico que apoyan al mantenimiento y/o configuración del Sistema de Gestión Académico de la Universidad Nacional de la Amazonía Peruana, utilizar esta arquitectura propuesta para que sea mejorada y/o actualizada en todas las fases del ciclo de vida que requiera el Sistema.
3. A los estudiantes de la Universidad de la Amazonía Peruana no dudar en informar los posibles errores o inconsistencias que manifieste el Sistema de Gestión Académico.
4. A los estudiantes de la Facultad de Ingeniería de Sistemas e Informática, continuar realizando estudios de buenas prácticas para diseño de Arquitecturas de Software.
5. A los directivos y altas autoridades de la Universidad Nacional de la Amazonía Peruana, promover eventos de capacitación sobre Diseño de Arquitectura de Software.
6. Y finalmente, considerar en mejorar en los aspectos de Seguridad Externa e Interna relacionados al Sistema de Gestión Académica, ya que ello influye en la eficiencia del mismo.

CAPÍTULO IX: REFERENCIAS BIBLIOGRÁFICAS

1. CANAL VELASCO, Carlos, Un Lenguaje para la Especificación y Validación de Arquitecturas de Software. Tesis para optar al grado de Doctor Ingeniero en Informática, Málaga - España 2000.
2. COHALLEN J. ".Modeling web application architectures with UML". USA 1999
3. TAHUITON MORA, Juan. Arquitectura de software para aplicaciones Web. Tesis para obtener el grado de Maestro en Ciencias en Computación. México 2011.
4. LIMÓN CORDERO, Rogelio Noé. Las vistas arquitectónicas de software y sus correspondencias mediante la gestión de modelos. Memoria para optar al grado de Doctor en Informática. España.
5. Bass, L., Clements, P., y Kazman, R. Software Architecture in Practice. Addison Wesley. 1998.
6. Garlan, D. y Perry, D. E. Special Issue on Software Architecture. IEEE Transactions on Software Engineering, 21(4). 1995.
7. Rumbaugh, J. et al. The Unified Modeling Language Reference Manual. Addison Wesley. 1999.
8. Garlan, D. y Perry, D. E. Special Issue on Software Architecture. IEEE Transactions on Software Engineering, 21(4). 1999.
9. Shaw, M. y Garlan, D. Software Architecture. Perspectives of an Emerging Discipline. Prentice Hall. 1996.
10. Cachero C. OO-H. Una extensión de los métodos OO para el modelado y generación automática de interfaces hipermediales. España 2003.
11. Koch N., A. Kraus, Cachero C., Meliá S. Integration of Business Process in Web Application Models. Journal of Web Engineering (JWE). Vol. 3, 2004.
12. Ceri S., Fraternali P., Matera M. Conceptual Modeling of Data-Intensive Web Applications (WebML), IEEE Internet Computing 6, No. 4, 20-30, July/August 2002.

13. Schwabe D., Almeida R., Moura I. OOHDM-Web: an environment for implementation of hipermedia applications in the WWW. SIGWEB News1. 8, 2, 18-34, June 1999.
14. Troyer, O.M.F. De, Leune, C.J. WSDM: a user centered design method for Web sites. In: Seventh International World Wide Web Conference, Springer-Verlag. Lecture Notes in Computer Science. 1998.
15. Bachmann F., Bass L., Chastek G., Donohoe P., Peruzzi F. The Architecture Based Design Method. CMU/SEI-2000-TR-001, 2000.
16. Meliá Santiago. Tesis Doctoral WebSA: Un Método de Desarrollo Dirigido por Modelos de Arquitectura para Aplicaciones Web. Tesis Doctoral.
17. Canal D. Tesis Doctoral: Un Lenguaje para la Especificación y Validación de Arquitecturas de Software, España 2000.
18. Booch, G. et al. La Guía del Usuario, Lenguaje de Modelado. Addison Wesley, 1999.
19. Kruchten, P. B. The 4+1 Vistas, Modelos de Arquitectura. IEEE Software, 1995.
20. Dávila M. et al., Evaluación de Arquitecturas de Software. Universidad de la República.
21. Camacho E. et al., Guía de estudio de Arquitecturas de Software, 2004. Pág. 9.
22. Palacio S. Katherine S. Modelo para el Diseño de un Sistema de Control de Gestión Académico – Administrativa en una Institución Universitaria aplicado en la División de Ingenierías de la Universidad del Norte. Colombia 2006. Pag. 25

ANEXOS

1. Matriz de Consistencia del Proyecto: “Diseño de una Arquitectura para el Sistema de Gestión Académica en la Universidad Nacional de la Amazonía Peruana, 2018”.

| PROBLEMA | OBJETIVOS | HIPÓTESIS | VARIABLES | INDICADORES | METODOLOGÍA |
|--|---|---|---|--|--|
| <p>Problema General</p> <p>¿Cómo una arquitectura de referencia basada en TOGAF, mejoraría el Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana?</p> | <p>Objetivo General</p> <p>Diseñar una arquitectura de referencia basada en el marco de trabajo TOGAF para el Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana.</p> <p>Objetivos Específicos</p> <ul style="list-style-type: none"> - Crear una arquitectura de referencia, generando los componentes necesarios que serán utilizados para la descripción arquitectural. - Evaluar la Arquitectura de Software antes y después del diseño propuesto. - - Evaluar el Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana antes y después de aplicar el diseño | <p>Hipótesis Principal</p> <p>Una arquitectura de referencia basada en el marco de trabajo TOGAF, permitirá mejorar el Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana.</p> <p>Hipótesis Nula</p> <p>Una arquitectura de referencia basada en el marco de trabajo TOGAF, no permitirá mejorar el Sistema de Gestión Académica de la Universidad Nacional de la Amazonía Peruana.</p> | <p>Variable Independiente (X)</p> <p>1.Arquitectura de Software</p> <p>Variable Dependiente (Y)</p> <p>2.Sistema de Gestión Académica</p> | <p>Dimensión: Operatividad</p> <p>1.1. Configurabilidad 1.2. Integrabilidad 1.3. Integridad 1.4. Interoperabilidad 1.5. Modificabilidad</p> <p>Dimensión: Actualización</p> <p>1.6. Mantenibilidad 1.7. Portabilidad 1.8. Reusabilidad 1.9. Escalabilidad 1.10. Capacidad de Prueba</p> <p>Dimensión: Satisfacción</p> <p>2.1. Disponibilidad 2.2. Confidencialidad 2.3. Funcionalidad</p> <p>Dimensión: Eficiencia</p> <p>2.4. Desempeño 2.5. Confiabilidad 2.6. Seguridad externa 2.7. Seguridad interna</p> | <p>Tipo de Investigación: Cualitativo, Descriptivo y Explicativo.</p> <p>Diseño de Investigación: Transeccional Descriptivo</p> <p>Población y Muestra: 06 personas</p> <p>Técnicas e Instrumentos de Recolección de datos:</p> <p>Técnica: Encuesta</p> <p>Instrumentos: Cuestionario</p> |

| | | | | | |
|--|------------|--|--|--|--|
| | propuesto. | | | | |
|--|------------|--|--|--|--|

2. Cuestionario Pre – Test

PRESENTACIÓN: Soy alumno de la Maestría de Ingeniería de Sistemas e Informática de la Universidad de la Amazonía Peruana, estoy realizando la Tesis “DISEÑO DE UNA ARQUITECTURA PARA EL SISTEMA DE GESTIÓN ACADÉMICA EN LA UNIVERSIDAD NACIONAL DE LA AMAZONÍA PERUANA, 2019”

OBJETIVO: Determinar en las personas la opinión con respecto a los atributos de calidad que deben considerarse para la arquitectura y el sistema de gestión académica de la UNAP antes de aplicar el diseño propuesto.

ENTREVISTADOS: Personas interesadas en la gestión académica y administrativa de la UNAP.

NOMBRE DEL ENCUESTADO:

MARCA CON UNA X SOLO UNA DE LAS ALTERNATIVAS, QUE RESPONDA LA PERSONA ENCUESTADA:

Nivel de Operatividad del sistema:

1. Con respecto a la posibilidad que se otorga a un usuario experto a realizar ciertos cambios al sistema actual. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

2. Con respecto a los componentes del sistema actual que fueron desarrollados separadamente al ser integrados trabajan correctamente. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

3. Con respecto a la ausencia de alteraciones inapropiadas de la información del sistema actual. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

4. Con respecto a la habilidad de que un grupo de partes del sistema actual trabajen con otro sistema. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

5. Con respecto a la facilidad con la que el software actual, al ser sometido a una serie de pruebas, puede demostrar sus fallas. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

Nivel de Actualización del sistema:

6. Con respecto a la habilidad de realizar cambios futuros al sistema actual. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

7. Con respecto a la capacidad de someter al sistema actual a reparaciones y evolución y/o modificar el sistema de manera rápida y a bajo costo. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

8. Con respecto a la habilidad del sistema actual para ser ejecutado en diferentes ambientes de computación. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

9. Con respecto al diseño del sistema actual, su estructura o parte de sus componentes puedan ser reutilizados en futuras aplicaciones. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

10. Con respecto a la posibilidad de ampliar el diseño arquitectónico, de datos o procedimental del sistema actual. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

Nivel de Satisfacción del sistema:

11. Con respecto a la disponibilidad del sistema actual para el uso. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

12. Con respecto a la ausencia de acceso no autorizado a la información en el sistema actual. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

13. Con respecto a la habilidad del sistema actual para mantenerse operativo a lo largo del tiempo. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

Nivel de Eficiencia del sistema:

14. Con respecto a la habilidad del sistema actual para realizar el trabajo para el cual fue concebido. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

15. Con respecto al sistema actual o componentes del mismo cumple con sus funciones designadas, dentro de ciertas restricciones dadas, como velocidad, exactitud o uso de memoria. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

16. Con respecto a la ausencia de consecuencias catastróficas en el ambiente en el que opera el sistema actual. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

17. Con respecto a la habilidad del sistema actual para resistir a intentos de uso no autorizados y negación del servicio mientras se sirve a usuarios. ¿Su opinión es?

Cumple
cumple

Cumple parcialmente

No

3. Cuestionario Post – Test

PRESENTACIÓN: Soy alumno de la Maestría de Ingeniería de Sistemas e Informática de la Universidad de la Amazonía Peruana, estoy realizando la Tesis “DISEÑO DE UNA ARQUITECTURA PARA EL SISTEMA DE GESTIÓN ACADÉMICA EN LA UNIVERSIDAD NACIONAL DE LA AMAZONÍA PERUANA, 2019”

OBJETIVO: Determinar en las personas la opinión con respecto a los atributos de calidad que deben considerarse para la arquitectura y el sistema de gestión académica de la UNAP después de aplicar el diseño propuesto.

ENTREVISTADOS: Personas interesadas en la gestión académica y administrativa de la UNAP.

NOMBRE DEL ENCUESTADO:

MARCA CON UNA X SOLO UNA DE LAS ALTERNATIVAS, QUE RESPONDA LA PERSONA ENCUESTADA:

Nivel de Operatividad del sistema:

18. Con respecto a la posibilidad que se otorga a un usuario experto a realizar ciertos cambios al sistema actual. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

19. Con respecto a los componentes del sistema actual que fueron desarrollados separadamente al ser integrados trabajan correctamente. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

20. Con respecto a la ausencia de alteraciones inapropiadas de la información del sistema actual. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

21. Con respecto a la habilidad de que un grupo de partes del sistema actual trabajen con otro sistema. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

22. Con respecto a la facilidad con la que el software actual, al ser sometido a una serie de pruebas, puede demostrar sus fallas. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

Nivel de Actualización del sistema:

23. Con respecto a la habilidad de realizar cambios futuros al sistema actual. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

24. Con respecto a la capacidad de someter al sistema actual a reparaciones y evolución y/o modificar el sistema de manera rápida y a bajo costo. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

25. Con respecto a la habilidad del sistema actual para ser ejecutado en diferentes ambientes de computación. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

26. Con respecto al diseño del sistema actual, su estructura o parte de sus componentes puedan ser reutilizados en futuras aplicaciones. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

27. Con respecto a la posibilidad de ampliar el diseño arquitectónico, de datos o procedimental del sistema actual. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

Nivel de Satisfacción del sistema:

28. Con respecto a la disponibilidad del sistema actual para el uso. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

29. Con respecto a la ausencia de acceso no autorizado a la información en el sistema actual. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

30. Con respecto a la habilidad del sistema actual para mantenerse operativo a lo largo del tiempo. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

Nivel de Eficiencia del sistema:

31. Con respecto a la habilidad del sistema actual para realizar el trabajo para el cual fue concebido. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

32. Con respecto al sistema actual o componentes del mismo cumple con sus funciones designadas, dentro de ciertas restricciones dadas, como velocidad, exactitud o uso de memoria. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

33. Con respecto a la ausencia de consecuencias catastróficas en el ambiente en el que opera el sistema actual. ¿Su opinión es?

Cumple Cumple parcialmente No
cumple

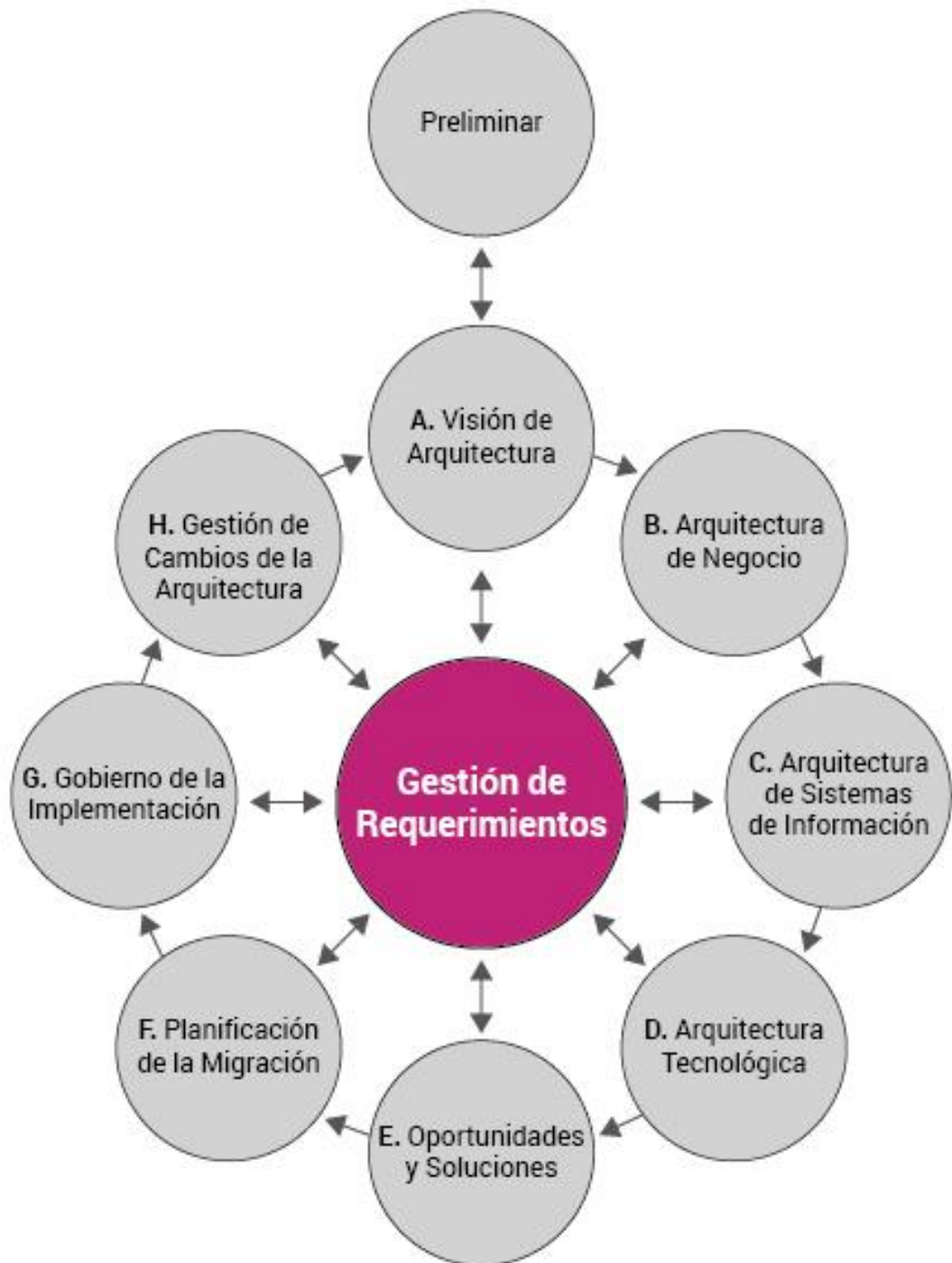
34. Con respecto a la habilidad del sistema actual para resistir a intentos de uso no autorizados y negación del servicio mientras se sirve a usuarios. ¿Su opinión es?

Cumple
cumple

Cumple parcialmente

No

4. Ciclo del Método de Desarrollo de Arquitecturas.



Fuente: TOGAF Versión 9.1 – Guía de Bolsillo