

UNIVERSIDAD NACIONAL

Escuela de Informática
Licenciatura

**PROPUESTA DE DOCUMENTACIÓN DE ARQUITECTURA INTEGRAL DE
SOFTWARE PARA LA SECCIÓN DE SEGURIDAD INSTITUCIONAL DE LA SEDE
CENTRAL DE LA UNIVERSIDAD NACIONAL DE COSTA RICA**

Para optar por el grado de Licenciatura en

INFORMÁTICA CON ÉNFASIS EN SISTEMAS DE INFORMACIÓN

Por: YENDRY HERNÁNDEZ MONTEZUMA

IGNACIO AVALOS BONILLA

NOVIEMBRE, 2021

Agradecimiento

Gracias a la vida por este nuevo triunfo y a las personas que indirectamente creyeron y me apoyaron en la realización de este proyecto.

Mi más cordial agradecimiento a la Universidad Nacional de Costa Rica, que fue la casa que me acogió para poder realizarme profesionalmente.

A nuestro tutor, al Dr. Fulvio Lizano, por el tiempo dedicado, sus conocimientos brindados y su paciencia. De igual modo mi agradecimiento a los lectores de nuestro proyecto al M. SC. Majid Bayani y la M. SC. Gabriela Garita, por su acertado asesoramiento, gracias a su apoyo se pudo concluir el presente trabajo.

A todas las personas que me apoyaron e hicieron posible que este trabajo se realizara con éxito.

Yendry Hernández Montezuma

Dedicatoria

A Dios, por haberme permitido llegar hasta este punto y culminar con éxito este trabajo, gracias por su infinita bondad y amor.

A mi familia por haberme apoyado en todo momento, a mi padre Julián Hernández por su ejemplo de valentía y perseverancia, que me ha infundado siempre. Y a mi madre Julia Montezuma por sus consejos y valores. A mis hermanos Brayan y Leticia, que me han ayudado a ser una persona de bien.

Y a César Parral, que ha sido una persona que me aconsejado y guiado para desarrollarme profesionalmente.

Yendry Hernández Montezuma

Agradecimiento

En primer lugar, quiero agradecer a mi tutor Fulvio Lizano, quien con sus conocimientos y su apoyo me guió a través de cada una de las etapas de este proyecto para alcanzar los resultados que buscaba.

Un agradecimiento al profesor Majid Bayani, que con su ayuda y enseñanzas he podido arribar a estos resultados de manera exitosa.

También quiero agradecer a la Universidad Nacional de Costa Rica por brindarme todos los recursos y herramientas que fueron necesarios para llevar a cabo el proceso de investigación.

José Ignacio Ávalos Bonilla

Dedicatoria

Primeramente, a Dios, luego a mi familia, a mi novia y amigos que han sido parte de este proceso.

Ustedes han sido siempre el motor que impulsa mis sueños y esperanzas, quienes estuvieron siempre a mi lado en los días y noches más difíciles durante mis horas de estudio.

Siempre han sido mis mejores guías de vida. Hoy cuando concluyo mis estudios, les dedico a ustedes este trabajo como una meta más conquistada.

Gracias a mi compañera de trabajo durante este proyecto, por la ayuda brindada para culminar esta meta con éxito.

TABLA DE CONTENIDOS

TABLA DE CONTENIDOS.....	III
LISTA DE FIGURAS	V
LISTA DE TABLAS.....	VIII
LISTA DE ANEXOS	X
CAPÍTULO I: INTRODUCCIÓN	2
1.1 Antecedentes.....	3
1.2 Planteamiento del Problema	4
1.3 Justificación	5
1.4 Objetivos del Proyecto.....	9
1.4.1 <i>Objetivo general</i>	9
1.4.2 <i>Objetivos específicos</i>	9
CAPÍTULO II. MARCO TEÓRICO	11
2.1 Sección de Seguridad Institucional (SSI)	11
2.1.1 <i>Estructura Organizacional</i>	11
2.2 Generalidades de la Arquitectura de Software	12
2.2.1 <i>Historia de la Arquitectura de Sistema (AS)</i>	14
2.2.2 <i>El presente y el futuro de la Arquitectura de Software</i>	15
2.2.3 <i>Importancia de la Arquitectura de Software</i>	16
2.2.4 <i>Estructura de una Arquitectura de Software</i>	17
2.2.5 <i>Estilos</i>	17
2.2.6 <i>Frameworks y Vistas</i>	19
2.2.7 <i>Lenguajes de descripción arquitectónica (ADL)</i>	21
2.2.8 <i>Procesos y metodologías</i>	22
2.2.9 <i>Patrones de diseño de la arquitectura de software</i>	23
2.2.10 <i>Ciclo del desarrollo de la Arquitectura de Software</i>	29
2.3 Referentes Teóricos de Instrumentos.....	33
2.3.1 <i>Revisión Literaria</i>	33
2.3.2 <i>Encuesta</i>	36
2.3.3 <i>Prototipo</i>	38
2.3.4 <i>Método de Análisis de Compensación de Arquitecturas (ATAM)</i>	39
2.4 Descripción de Tres Arquitecturas de Software	45
2.4.1 <i>Arquitectura del Modelo Kruchten 4+1</i>	45
2.4.2 <i>Arquitectura del Modelo de Gestión de Procesos de Negocio (BPM)</i>	54
2.4.3 <i>Arquitectura de Servicios Web (WS)</i>	71

CAPÍTULO III. METODOLOGÍA.....	97
3.1 Tipo de Investigación.....	97
3.2 Población y Muestra	97
3.2.1 Población.....	97
3.2.2 Muestra	98
3.3 Descripción de Instrumentos	109
3.3.1 Revisión de Literatura (RL)	109
3.3.2 Encuesta.....	111
3.3.3 Prototipo de documentación de una arquitectura integral.....	113
3.3.4 Método de Análisis de Compensación de Arquitecturas (ATAM)	115
3.4 Procedimientos para Analizar la Información del Diagnóstico	117
CAPÍTULO IV. PROPUESTA DE SOLUCIÓN	119
4.1 Diagnóstico	119
4.2 Propuesta de Solución.....	119
4.2.1 Diseño de la Arquitectura de Software propuesto	120
4.2.2 Alcances	121
4.2.3 Objetivos y restricciones.....	121
4.2.4 Acrónimos, Siglas y definiciones	122
4.2.5 Glosario de términos y definiciones	123
4.2.6 Descripción de la documentación de la Arquitectura de Software	126
4.2.7 Arquitectura del modelo de Gestión de Procesos de Negocio (BPM).....	127
4.2.8 Business Process Modeling Notation (BPMN)	127
4.2.9 Arquitectura del Modelo 4+1	129
4.2.10 Arquitectura General propuesta	146
4.2.11 Método de Usabilidad.....	147
4.2.12 Vista de Datos (E-R)	159
4.2.13 Especificaciones preliminares de requerimientos	161
4.2.14 Calidad y rendimiento de procesos.....	166
4.3 Validación de la Propuesta	168
4.3.1 Método de Análisis de Compensación de Arquitecturas, ATAM.....	168
4.3.2 Ejecución de ATAM, Fase de Presentación.....	172
4.3.3 Ejecución de ATAM, Fase de Investigación y Análisis.....	173
4.3.4 Ejecución de ATAM, Fase de Pruebas.....	176
4.3.5 Ejecución de ATAM, Fase de Informes.....	209
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES	211
5.1 Conclusiones.....	211
5.2 Limitaciones	211
5.3 Trabajos Futuros	213
REFERENCIAS	¡ERROR! MARCADOR NO DEFINIDO.
ANEXOS.....	227

Lista de Figuras

<i>Figura 1.</i> Esquema General de la Ejecución de Pasos en el Logro del Proyecto. -----	8
<i>Figura 2.</i> Estructura Organizacional de la Sección de Seguridad Institucional. -----	12
<i>Figura 3.</i> Ciclo de Vida de la Arquitectura de Software (AS). -----	30
<i>Figura 4.</i> Flujo de Trabajo ATAM. -----	40
<i>Figura 5.</i> Modelo de “4+1” Vistas.-----	46
<i>Figura 6.</i> Estructura de Documentación de Arquitectura Modelo Kruchten 4+1. -----	47
<i>Figura 7.</i> Componentes de la Vista Lógica.-----	48
<i>Figura 8.</i> Componentes de la Vista de Procesos.-----	49
<i>Figura 9.</i> Componentes de la Vista de Desarrollo.-----	51
<i>Figura 10.</i> Componentes de la Vista Física. -----	52
<i>Figura 11.</i> Estructura de Documentación para Gestión de Procesos de Negocio. (BPM). -----	54
<i>Figura 12.</i> Arquitectura de Gestión de Proceso de Negocio (BPM). -----	59
<i>Figura 13.</i> Notación para el Modelado BPMN. -----	63
<i>Figura 14.</i> Modelo de Madurez. -----	70
<i>Figura 15.</i> Estructura de Documentación de Arquitectura de Servicios Web.-----	72
<i>Figura 16.</i> Esquema de Jerarquía de los Web Services.-----	74
<i>Figura 17.</i> Modelo Básico de los WS. -----	76
<i>Figura 18.</i> Actores, Objetos y las Operaciones de la Arquitectura de los SW.-----	78
<i>Figura 19.</i> Estructura del Funcionamiento y los Componentes de los SW.-----	80
<i>Figura 20.</i> Ejemplo de Acceso de un WSDL. -----	82
<i>Figura 21.</i> Estructura de la Interoperabilidad de los Servicios Web. -----	85
<i>Figura 22.</i> Estructura de Orquestación en los Servicios Web.-----	88
<i>Figura 23.</i> Estructura de Coreografía en los Servicios Web. -----	89
<i>Figura 24.</i> Coreografía Versus Orquestación. -----	90
<i>Figura 25.</i> Ejemplo de Orquestación de los Servicios Web (WS-BPEL)-----	93

<i>Figura 26. Actividades del WS-CDL. -----</i>	<i>96</i>
<i>Figura 27. Departamentos Encuestados -----</i>	<i>101</i>
<i>Figura 28. ¿El mecanismo utilizado por la SSI-UNA se ajusta a las necesidades de los funcionarios? -----</i>	<i>103</i>
<i>Figura 29. ¿Considera usted que el mecanismo utilizado por la SSI-UNA es eficiente y eficaz? -----</i>	<i>104</i>
<i>Figura 30. ¿Con cuánta frecuencia usted solicita el servicio Reserva o permiso de ingreso a parqueos institucionales y otras áreas de interés de la SSI-UNA? -----</i>	<i>107</i>
<i>Figura 31. ¿Cuánto ha sido la duración en tiempo para realizar una solicitud a la SSI-UNA? -----</i>	<i>108</i>
<i>Figura 32. Ejecución del Instrumento Revisión Literaria (RL).-----</i>	<i>111</i>
<i>Figura 33. Ejecución del Instrumento Encuesta-----</i>	<i>113</i>
<i>Figura 34. Ejecución del Instrumento Prototipo-----</i>	<i>114</i>
<i>Figura 35. Ejecución del Instrumento (ATAM). -----</i>	<i>116</i>
<i>Figura 36. Documento de Arquitectura de Software.-----</i>	<i>120</i>
<i>Figura 37. Diagrama del Modelado de Negocio de BMP. -----</i>	<i>128</i>
<i>Figura 38. Diagrama de la Vista Lógica del Sistema. -----</i>	<i>131</i>
<i>Figura 39. Diagrama de Clases que Muestra la Vista Lógica del Sistema. -----</i>	<i>132</i>
<i>Figura 40. Diagrama de Secuencia que muestra la vista Lógica del Sistema.-----</i>	<i>135</i>
<i>Figura 41. Diagrama de Comunicación que Muestra la Vista Lógica del Sistema. -----</i>	<i>136</i>
<i>Figura 42. Diagrama de actividad que muestra la vista de Procesos del Sistema. -----</i>	<i>138</i>
<i>Figura 43. Diagrama Componentes Muestra Vista de Desarrollo del Sistema. -----</i>	<i>141</i>
<i>Figura 44. Diagrama de Paquetes que muestra la vista de Desarrollo del Sistema. -----</i>	<i>142</i>
<i>Figura 45. Diagrama de Paquetes que Muestra la Vista Física del Sistema. -----</i>	<i>144</i>
<i>Figura 46. Diagrama de Componentes que Muestra la Vista Física del Sistema. -----</i>	<i>145</i>
<i>Figura 47. Arquitectura General propuesta.-----</i>	<i>147</i>
<i>Figura 48. Esquema Sobre el Método de Usabilidad de la ISO/IEC 9241-11. -----</i>	<i>148</i>
<i>Figura 49. Diagrama de Entidad-Relación de la Base de Datos-----</i>	<i>160</i>
<i>Figura 50. Seis Dimensiones de los Requerimientos. -----</i>	<i>161</i>
<i>Figura 51. Influencias Mutuas entre la Arquitectura y su Entorno.-----</i>	<i>162</i>

<i>Figura 52. Concepto de Árbol de Utilidad (UT).</i> -----	180
<i>Figura 53. Árbol de Utilidad.</i> -----	182

Lista de Tablas

Tabla 1. Marco de Referencia Arquitectónicos -----	20
Tabla 2. Patrones de diseño de arquitectura de software -----	25
Tabla 3. Fases del Método de Análisis de Compensación de Arquitecturas (ATAM) ---	42
Tabla 4. Resumen Sobre los Pasos del ATAM -----	43
Tabla 5. Tabla Comparativa de Vistas, Kruchten 4 +1 -----	53
Tabla 6. Ejemplo de documentación KPI-----	68
Tabla 7. Términos de Orquestación y Coreografía-----	91
Tabla 8. Información de la Muestra Relacionada con la Encuesta -----	100
Tabla 9. Acrónimos, siglas y sus definiciones -----	122
Tabla 10. Términos y Definiciones-----	124
Tabla 11. Las diez heurísticas de Nielsen -----	149
Tabla 12. Tabla de los Requerimientos Funcionales -----	163
Tabla 13. Tabla de los Requerimientos No funcionales -----	165
Tabla 14. Indicadores de Rendimiento y Calidad -----	167
Tabla 15. Fases y Pasos Ejecutadas ATAM-----	169
Tabla 16. Involucrados en la Evaluación del ATAM -----	170
Tabla 17. Pasos de ATAM Asociados con las Partes Interesadas -----	171
Tabla 18. Descripción Atributos de Calidad no Observables y Observables Vía Ejecución -----	179

Tabla 19. Plantilla para la Documentación y Análisis del Enfoque Arquitectónico -----	184
Tabla 20. Análisis Propuesta Arquitectónica de Escenario 01 por Medio de ATAM ---	185
Tabla 21. Análisis Propuesta Arquitectónica de Escenario 02 por Medio de ATAM ---	186
Tabla 22. Análisis Propuesta Arquitectónica de Escenario 03 por Medio de ATAM ---	188
Tabla 23. Análisis Propuesta Arquitectónica de Escenario 04 por Medio de ATAM ---	189
Tabla 24. Análisis Propuesta Arquitectónica de Escenario 05 por Medio de ATAM ---	190
Tabla 26. Análisis Propuesta Arquitectónica de Escenario 06 por Medio de ATAM ---	192
Tabla 26. Propuesta Arquitectónica de Escenario 07 por Medio de ATAM-----	194
Tabla 27. Propuesta Arquitectónica de Escenario 08 por Medio de ATAM -----	196

Tabla 28. Propuesta Arquitectónica de Escenario 09 por Medio de ATAM-----	197
Tabla 29. Propuesta Arquitectónica de Escenario 10 por Medio de ATAM-----	199
Tabla 30. Propuesta Arquitectónica de Escenario 11 por Medio de ATAM-----	200
Tabla 31. Propuesta Arquitectónica de Escenario 12 por Medio de ATAM-----	202
Tabla 32. Propuesta Arquitectónica de Escenario 13 por Medio de ATAM-----	203
Tabla 33. Propuesta Arquitectónica de Escenario 14 por Medio de ATAM-----	205
Tabla 34. Propuesta Arquitectónica de Escenario 15 por Medio de ATAM-----	206
Tabla 35. Votación de Atributos de Calidad -----	208

Lista de Anexos

Anexo 1.	Plantilla de encuesta ejecutada.-----	227
Anexo 2.	Respuesta de entrevistado sobre problema al realizar solicitud al SSI. -----	230
Anexo 3.	Respuesta a entrevistado sobre mecanismo utilizado por la SSI.-----	232
Anexo 4.	Respuesta dos entrevistado sobre mecanismo utilizado por la SSI.-----	233
Anexo 5.	Respuesta uno de entrevistado describiendo flujo de trabajo de la SSI. -----	234
Anexo 6.	Respuesta dos de entrevistado describiendo flujo de trabajo de la SSI. -----	235
Anexo 7.	Primer Correo a Arquitectos de Evaluación. -----	236
Anexo 8.	Segundo Correo a Arquitectos de Evaluación.-----	237
Anexo 9.	Captura de Pantalla de Sitio Web sobre Presentación de ATAM.-----	238
Anexo 10.	Captura Pantalla de Sitio Web sobre Modelo de Negocio. -----	239
Anexo 11.	Captura de Pantalla de Sitio Web sobre Presentación de Arquitectura.-----	240
Anexo 12.	Captura de Pantalla de Sitio Web sobre Formulario de Encuesta.-----	241
Anexo 13.	Captura de Pantalla de Sitio Web sobre árbol d Utilidad y Escenarios.-----	242
Anexo 14.	Captura de Pantalla de Sitio Web sobre votación de Escenarios. -----	243
Anexo 15.	Captura de Pantalla de Sitio Web sobre Menú principal. -----	244
Anexo 16.	Captura de Pantalla de Sitio Web sobre Nosotros.-----	245

Capítulo I: Introducción

El objetivo de este capítulo es servir de introducción a esta tesis. Durante su desarrollo se expondrá los antecedentes, el planteamiento del problema, la justificación, el objetivo general y los objetivos específicos.

Este proyecto está estructurado en 5 capítulos, para su mayor comprensión se describe cada capítulo con su contenido.

Capítulo I, Introducción: En este apartado se presenta los antecedentes, el planteamiento del problema, la justificación, el objetivo general y los objetivos específicos.

Capítulo II, Marco Teórico: En este capítulo se muestra el marco teórico que corresponde a las bases teóricas y científicas de este documento este se divide en cuatro apartados. Primer apartado, expone información sobre la Sección de Seguridad Institucional, su organigrama, así como la misión y visión. Segundo apartado corresponde a las generalidades de la Arquitectura de Software.

La tercera sección se refiere a los referentes teóricos de los instrumentos metodológicos utilizados para la ejecución de este proyecto. La cuarta sección describe tres arquitecturas de software, el modelo 4+1, el modelo de Gestión de los Procesos de Negocio, (BPM) y la arquitectura de los Servicios Web (WS).

Capítulo III, Marco Metodológico: Este capítulo se divide en 4 secciones: descripción del tipo de la investigación de este trabajo, introducir la población y la muestra, para compendiar los datos de las opiniones y flujo de trabajo de los involucrados directos con la Sección de

Seguridad Institucional, seguidamente se describe como se ejecutaron cada uno de los instrumentos y por último se define los procedimientos para analizar la información del diagnóstico.

Capítulo IV, Propuesta de solución: Este capítulo se divide en 3 secciones, primero el diagnóstico que corresponde a recopilar y tratar información relevante de la Sección de Seguridad Institucional, luego se presenta la propuesta de solución y por último la validación de la propuesta ejecutada.

Capítulo V, Conclusiones y Recomendaciones: En este capítulo se desarrolla tres secciones: las conclusiones observadas mediante el análisis de la investigación realizada, las limitaciones del proyecto y por último los trabajos futuros que se observó al finalizar este proyecto.

1.1 Antecedentes

Una documentación de arquitectura integral de Software abarca la definición de un producto a través de un conjunto de estructuras compuestas por elementos con propiedades y relaciones entre ellos (Clements, Garlan, Little, Nord, & Stafford, 2003).

La documentación de un diseño arquitectónico constituye la fase más detallada, dentro del ciclo de vida y de desarrollo de un software. Además, hace que se tome en cuenta aspectos esenciales como recurso, tiempo y espacio con el fin de poder programar, organizar y desarrollar un prototipo de software que cumpla con todas las expectativas del usuario. La arquitectura es un artefacto para el análisis inicial para garantizar que un enfoque de diseño produzca un sistema aceptable (Fernández, 2006).

Esta documentación de arquitectura integral no solo permitirá el aporte de un documento estructurado para el desarrollo de un sistema en la Sección de Seguridad de la Universidad Nacional, si no ofrecerá un conocimiento sobre esta disciplina que en los últimos años ha venido incrementado su importancia.

1.2 Planteamiento del Problema

Este apartado incluye la descripción del problema que atraviesa la Sección de Seguridad Institucional de la Universidad Nacional, (SSI-UNA) con el único objetivo de identificar los inconvenientes actuales.

Estos servicios se realizan de forma manual utilizando correos electrónicos y formularios en línea; arriesgando con ello la confidencialidad y el tiempo de respuesta. Además, estos flujos de trabajo no son nada amigables con el medio ambiente, ya que deben de imprimirse cantidades de solicitudes de reservas para informar al Centro de Monitoreo¹.

El Centro de Monitoreo está conformado por oficiales de seguridad y operadores de acceso vehicular el cual se encargan de conocer el tipo de solicitud de reserva² (ya aprobada por la

¹ Centro de Monitoreo: Sección que pertenece a la Universidad Nacional y que trabaja en conjunto con la Sección de Seguridad.

² Tipo de solicitud de reserva: Si la solicitud de reserva es a parqueos institucionales u otras áreas, (gimnasios, auditorios).

Sección de Seguridad Institucional), gestionarla y brindar el ingreso a diferentes puntos de la institución.

Según la coordinadora de la Sección de Seguridad Institucional “en años anteriores la comunidad universitaria solicitaba los permisos por medio de un documento formal, que se recibía en la recepción y luego se sacaban copias para distribuir e informar a los oficiales de seguridad, sin embargo, el trámite era muy tedioso porque debíamos recoger la firma del recibido para garantizarnos que el oficial de seguridad estaba informado” (M. Muñoz, comunicación personal, 23 de agosto de 2019).

A lo largo de los años ellos han venido buscando métodos que sean menos tediosas y más eficientes, porque lo que optaron por el uso del software GoogleForm y formularios. Pero el problema persiste ya que el flujo de procesos es grande y los métodos que se utilizan no son eficientes y no aportan fluidez.

Debido a lo anterior, se identificó la necesidad de crear una solución efectiva, con el fin de diseñar una Arquitectura de Software Integral que brinde las características y funcionalidades que puedan satisfacer de manera eficiente las necesidades de las unidades involucradas.

En la siguiente sección se presentará una justificación detallada del proyecto como parte necesaria de la investigación relacionada a la propuesta actual.

1.3 Justificación

La justificación del proyecto viene dada por cinco dimensiones que se discuten a continuación:

- *Actitud Individual:* Es la mejora de cada funcionario que requiere el servicio, de solicitud de reservas o permisos de ingreso a parqueos institucionales y otras áreas de interés. El acceso de una tecnología en la cual está centrado este diseño arquitectural permite el adecuado manejo eficaz de estas solicitudes incrementando eficiencia operativa.
- *La interoperabilidad entre departamentos:* Este proyecto busca contribuir indirectamente con los servicios de solicitud de reservas o permisos de ingreso a los parqueos institucionales y otras áreas, solicitados por personas y entidades fuera de la Universidad Nacional.

Lo anterior se basa en que la propuesta tiene como objetivo el diseño de la arquitectura de software de un sistema que puede facilitar la interoperabilidad indirectamente entre departamentos.

- *Impacto económico:* Los recursos financieros y de tiempo con que cuenta la Sección de Seguridad Institucional de la UNA son limitados. Esto implica que contratar un producto como el propuesto en este proyecto, que sea algo poco realista. Por consiguiente, la propuesta fue diseñado con base en los principales desafíos relacionados con la rapidez, eficiencia e interoperabilidad.
- *Efecto Ambiental:* Esta propuesta tiene un impacto ambiental efectivo y positivo al realizar los trámites a través del sistema tradicional existente implicaría el uso de gran cantidad de papel diariamente.
- *Aspectos Técnico:* Este proyecto genera un impacto a partir del uso de los resultados. En este caso los beneficiarios son la Universidad Nacional de Costa Rica, la comunidad estudiantil y los investigadores.

Esta documentación arquitectural contiene conocimientos de alto nivel, por lo que su relevancia servirá para generar un ejemplo de cómo se documenta una arquitectura de software, que sirva en los procesos académicos de formación a futuros profesionales en computación.

La considerable información y la situación con la que manejan, gestionan y controlan los ingresos de personas y vehículos la Sección de Seguridad Institucional, despertó el interés de desarrollar este proyecto como parte de ofrecer una herramienta que pueda sustentar estas necesidades.

Esta documentación arquitectural contiene conocimientos amplios y es muy rigurosa, por lo que su relevancia servirá para generar un ejemplo de cómo se documenta una arquitectura de software, que sirva en los procesos académicos de formación a futuros profesionales en computación.

La considerable información y la situación con la que manejan, gestionan y controlan los ingresos de personas y vehículos la Sección de Seguridad Institucional, es la causa principal que llevó a cabo esta investigación.

En la *Figura 1*, se muestra el “Esquema General de la Ejecución de Pasos en el Logro del Proyecto”. Este esquema muestra los métodos y pasos para la ejecución y el desarrollo del proyecto.

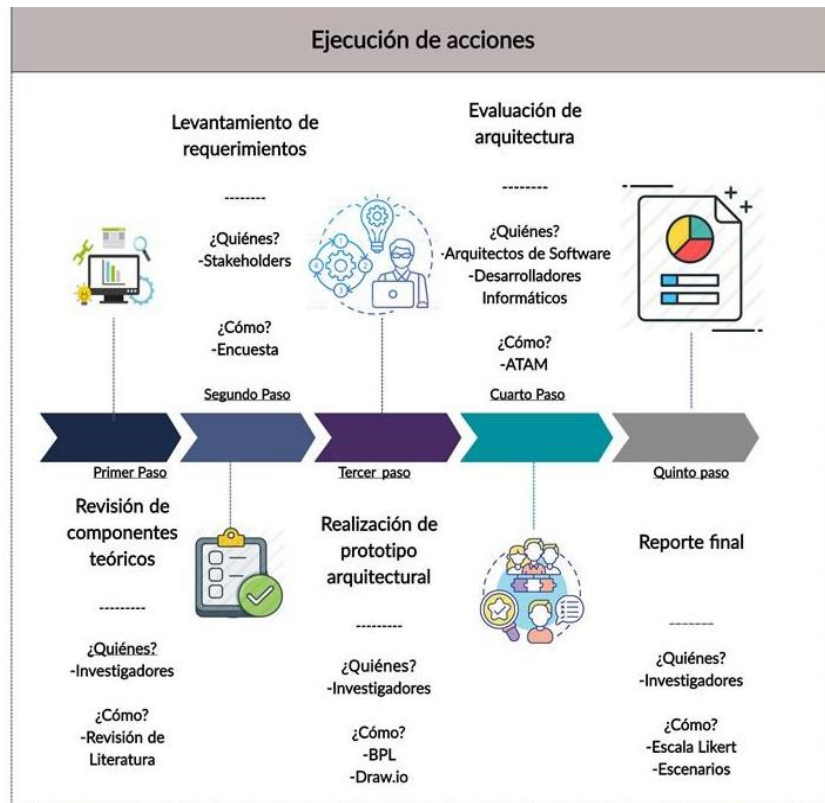


Figura 1. Esquema General de la Ejecución de Pasos en el Logro del Proyecto.

Fuente: Elaboración propia.

Inicialmente, se analizó una revisión literaria con el fin de conocer arquitecturas de software y poder implementarlas cumpliendo con los requerimientos. Se aplicó una encuesta para levantar requerimientos y las necesidades de los involucrados. Además, se elaboró un prototipo de arquitectura integral por medio de diagramas, figuras y software de simulaciones. El

paso final fue la evaluación por medio de un método adecuado de evaluación de arquitecturas llamado ATAM (Método de Análisis de Compensación de Arquitecturas). Basado en los resultados obtenidos se aplicó la escala de Likert, priorizando los escenarios del enfoque arquitectónico para evaluar los atributos de calidad arquitectura propuesta.

1.4 Objetivos del Proyecto

A continuación, se presentan los objetivos del proyecto.

1.4.1 Objetivo general

Desarrollar una propuesta de documentación de Arquitectura de Software Integral para la Sección de Seguridad Institucional, que permita la gestión y control de acceso de personas y vehículos a la Sede Central de la Universidad Nacional de Costa Rica.

1.4.2 Objetivos específicos

Para el logro del objetivo general se formulan los siguientes objetivos específicos.

- 1- Revisar los componentes teóricos disponibles relacionados con la documentación de Arquitectura de Software, con el fin de contar con una base y apoyar el proceso de investigación, para desarrollarla en la Sección de Seguridad Institucional de la Sede Central de la Universidad Nacional de Costa Rica.
- 2- Analizar el flujo de trabajo, mediante una encuesta, para lograr determinar cuáles son los inconvenientes que atraviesa la Sección de Seguridad Institucional, Sede Central de la Universidad Nacional de Costa Rica.

- 3- Diseñar un prototipo de Arquitectura Integral de Software, para establecer procesos de trabajos competentes, para la Sección de Seguridad Institucional de la Sede Central de la Universidad Nacional de Costa Rica.
- 4- Evaluar el prototipo de una documentación de Arquitectura Integral de Software, mediante algunos principios de ATAM (Architecture Tradeoff Analysis Method), para determinar la calidad de la Arquitectura Integral.

Capítulo II. Marco Teórico

Este capítulo contiene un conjunto de teorías, conceptos y aportes científicos que sustentan esta investigación que consta de cuatro secciones.

La primera parte incluye información sobre la Sección de Seguridad Institucional de la Universidad Nacional, (SSI-UNA), donde se llevó a cabo dicho proyecto.

En la segunda parte se desarrollaron los conceptos y las generalidades de una Arquitectura de Software, la historia de la Arquitectura de Software, así como el presente y el futuro de esta. En la tercera sección se describe los referentes teóricos de los instrumentos metodológicos ejecutados en el desarrollo de este trabajo. Finalmente, la cuarta sección se basa en describir los principales conceptos de las tres arquitecturas elegidas, enfatizando lo referido a la estructura de la documentación de cada arquitectura.

2.1 Sección de Seguridad Institucional (SSI)

La Sección de Seguridad Institucional (SSI) de la Universidad Nacional es una unidad encargada de gestionar las reservas y los permisos de ingresos a parqueos institucionales y otras áreas de interés, realizadas por los funcionarios de alguna Escuela de esta institución o por los estudiantes.

2.1.1 Estructura Organizacional

En la *Figura 2*, “Estructura Organizacional de la Sección de Seguridad Institucional, se presenta el organigrama de la SSI.

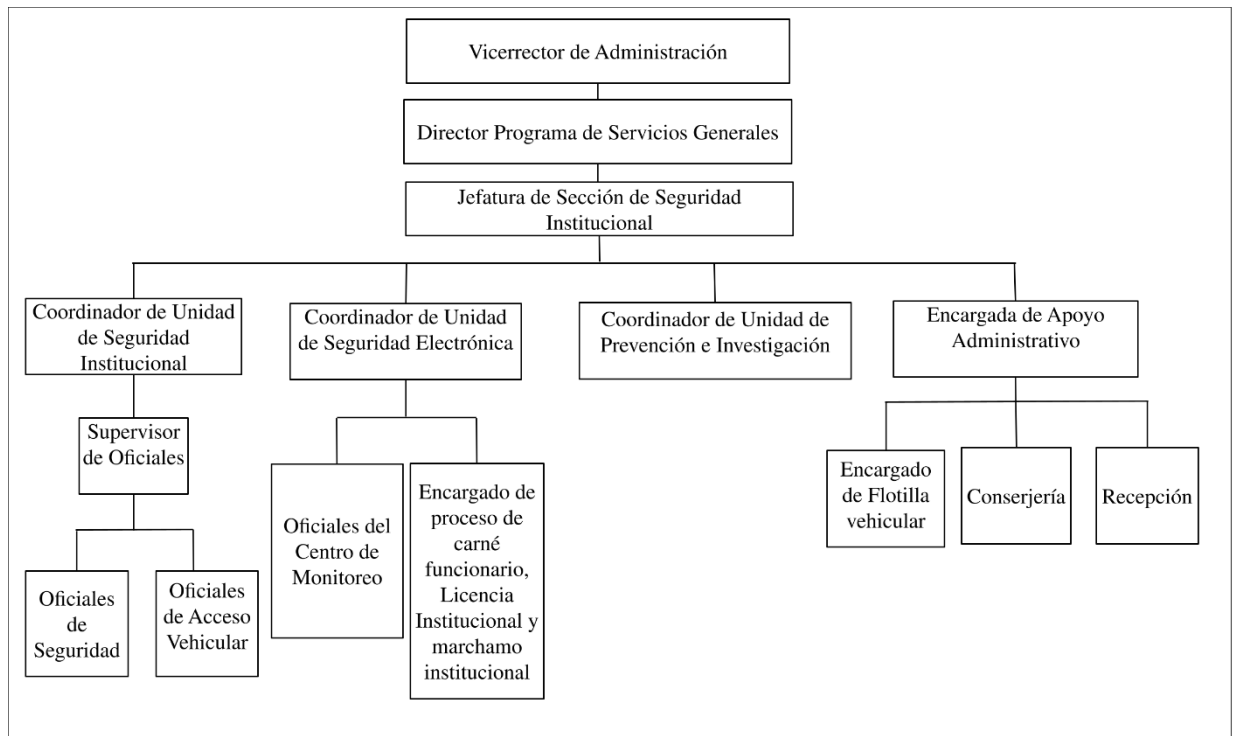


Figura 2. Estructura Organizacional de la Sección de Seguridad Institucional.

Fuente: Elaboración propia con base en la estructura organizacional de la Sección de Seguridad Institucional de la Universidad Nacional.

2.2 Generalidades de la Arquitectura de Software

La Arquitectura de Software es todo un componente de diseño abstracto. Para entender con la mayor precisión lo que significa la Arquitectura de Software se tomó inicialmente la definición textual que indica: “La arquitectura de un sistema es a grandes rasgos una vista del sistema que incluye los componentes principales del mismo donde la conducta de estos componentes se percibe

desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema” (Clements, Bass, & Kazman, 2002, p.02).

“La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión del detalle inherente a la mayor parte de las abstracciones, donde constituye un puente entre el requerimiento y el código” (Clements, Bass, & Kazman, 2002, p.02).

También, basado en el estándar ANSI/IEEE 1471-2000 define que la “Arquitectura es definida por la práctica habitual como la organización fundamental de un sistema, expresada en términos de componentes, las relaciones entre ellos, el entorno y los principios que gobiernan su diseño y evolución” (IEEE, 2019).

Para el (Software Engineering Institute, [SEI], 2012), la Arquitectura de Software es una representación que ayuda a comprender cómo se comportará el sistema, ya que es el principal portador de cualidades como lo es el rendimiento, la modificabilidad y la seguridad, de las cuales ninguna se puede lograr sin una visión arquitectónica unificadora.

Por ende, según Rozanski & Woods: una arquitectura de sistemas es un conjunto de decisiones significativas acerca de la organización del desarrollo de un software, la selección de los elementos estructurales, sus interfaces y las conductas es lo que compone al sistema (Rozanski & Woods, 2005).

2.2.1 Historia de la Arquitectura de Sistema (AS)

La historia de la Arquitectura de Sistema se remonta hasta 1960 donde Dijkstra, Parnas y Brooks (1968) (citado en Clements, Bass, & Kazman, 2002) considera que debe de establecer una estructuración correcta de los sistemas de software antes de lanzarse a programar.

Seguidamente la AS, quedó en un estado de vida latente, no fue hasta principios de los 90, cuando empezó a despertar cierto interés.

Luego se inició una expansión por parte de Dewayne Perry de AT&T Bell Laboratories de New Jersey y Alexander Wolf de la Universidad de Colorado, que fundaron esta disciplina, y a esta se le unió David Garlan, Mary Shaw, Paul Clements, y Robert Allen.

Luego en 1972, Parnas fue el primero en introducir el concepto moderno de abstracción a través de un ensayo en el que defendía el modularidad en el diseño del software, como un medio de dotar al sistema de flexibilidad, tanto en el desarrollo como en el mantenimiento correctivo o adaptativo Parnas (1972).

Shaw y Clements (2006) hicieron una retrospectiva del desarrollo de la arquitectura de software desde finales de la década de 1980. Estos autores argumentan que, desde sus raíces la arquitectura de software ha madurado para abarcar un amplio conjunto de notaciones, herramientas y técnicas de análisis.

En el momento de su estudio, los citados autores indicaron que el futuro de la arquitectura de sistemas estaba marcado por la necesidad de ampliar las relaciones formales entre las decisiones

de diseño arquitectónico y los atributos de calidad, encontrar el lenguaje adecuado para representar las arquitecturas y la forma de asegurar la conformidad entre la arquitectura y el código.

La AS cada día cuenta con bases científicas muy sólidas, con el fin de crear los atributos apropiados de calidad de un sistema y luego desarrollar un software completo y funcional.

2.2.2 El presente y el futuro de la Arquitectura de Software

En la actualidad, la Arquitectura de Software está en constante desarrollo y consolidando diferentes métodos, técnicas y estándares de arquitectura en general.

Para algunos investigadores Krutchen P, Obbink H, Stafford J. (2006) la nueva era está dedicada a la arquitectura, donde se estudian las decisiones de diseño arquitectónico y su justificación.

Por ende, ellos recalcan que el impacto de la AS cada día es mayor con el fin de educar a los arquitectos en esta disciplina. Las herramientas específicamente diseñadas para los arquitectos estarán disponibles para respaldar sus tareas de diseño, representación, análisis e implementación.

Kruchten, Obbink, & Stafford, (2006), indica que la Arquitectura del Software será reconocida como una base clave para el desarrollo de software.

Por lo tanto, lo anterior se complementa en lo que dice este gran personaje, “La práctica debe siempre ser edificada sobre la buena teoría” Leonardo Da Vinci (s.f).

2.2.3 Importancia de la Arquitectura de Software

La Arquitectura de Software se ha posicionado en el contexto del desarrollo del software como uno de los artefactos de análisis inicial que permite analizar que el enfoque del diseño produzca un sistema aceptable (SEI, 2017). En cuanto a lo anterior, Cervantes, Velasco-Elizondo & Castro considera que la arquitectura es uno de los factores claves para el éxito del sistema (2016).

La arquitectura de software es un concepto fundamental, porque la manera en que se estructura un sistema tiene un impacto directo sobre la capacidad para satisfacer los atributos de calidad del sistema (Cervantes, et al., 2015).

Además, (Cervantes, et al., 2015) indica que la arquitectura es el portador principal de las cualidades del sistema, ya que contempla, la capacidad de modificación, la seguridad y lo robusto. Sin embargo, ninguna de estas cosas se puede lograr sin una visión arquitectónica unificadora.

Clements & Northrop, (1996) piensan que la arquitectura es el principal vehículo para la comunicación con las partes interesadas. Según ellos: una arquitectura permite alcanzar el logro de ciertos atributos de calidad específicos de un sistema. Asimismo, la calidad de un sistema también se ve impactada de forma positiva al estudiar su arquitectura puesto que por este medio es posible predecir ciertas cualidades que tendrá el sistema.

Por lo tanto, según Booch & Jacobson la AS permite disminuir la necesidad de volver hacer el trabajo, donde además facilita el mantenimiento y este en conjunto ayuda a bajar los gastos (Booch & Jacobson, 1999).

En conclusión, esta disciplina juega un papel fundamental en el torno al proceso de desarrollo de un software. Para justificar el valor de la arquitectura de software documentada se toma una cita del escritor Steuart Henderson Britt, leída en una publicación de (Kostochkina & Gordievich, 2020) donde dice: “hacer negocios sin publicidad es como hacerle un guiño a una chica en la oscuridad. Sabes lo que estás haciendo, pero ella no lo está viendo”. (p.7).

2.2.4 Estructura de una Arquitectura de Software

Una Arquitectura de Software está compuesta por un conjunto de estructuras, vistas, estilos y patrones de diseño.

Los tres tipos de estructuras en el desarrollo de una arquitectura de software:

- Estructura de módulos. **(Module)**
- Estructura de los Componentes y Conectores. **(Component and Connect)**
- Estructura de Asignación. **(Allocation)**

2.2.5 Estilos

Un estilo arquitectónico es un concepto descriptivo que define una forma de articulación u organización arquitectónica, este conjunto es catalogado por las formas básicas posibles de estructuras de software (Reynoso, 2004).

Klein, Kazman, Bass, Carriere, y Lipson, proponen una definición según la cual un estilo arquitectónico es una descripción del patrón de los datos y la interacción de control entre los componentes (1999).

El conjunto de los estilos cataloga las formas básicas posibles de estructuras de software, mientras que las formas complejas se articulan mediante composición de los estilos fundamentales. Un estilo es una especialización de elemento y tipos de relación, junto con un conjunto de restricciones sobre cómo se pueden usar (Aïcha Choutri, 2010).

La descripción de un estilo se puede formular en lenguaje natural o en diagramas, pero lo mejor es hacerlo en un lenguaje de descripción arquitectónica o en lenguajes formales de especificación (Aïcha Choutri, 2010).

Para tenerlo más claro (Clements et al., 2010) realiza una clasificación de estilos de arquitectura de la siguiente manera:

Estilo de módulos: Está compuesto por los estilos de descomposición, uso, generalización, en capas, aspectos y estilo de modelo de datos.

Estilo de Componentes y Conectores: Contiene estilos de tuberías y filtros, Cliente-Servidor, Peer to Peer, Servicio Orientada a la Arquitectura, Publish Suscribe, multicapas, y el de Compartir Datos.

Estilo de Asignación: Incluye estilo de despliegue, Install, Trabajo Asignado, y otras asignaciones.

Estilo híbrido: Esta combina otro estilo o más.

2.2.6 Frameworks y Vistas

Las vistas es un concepto fundamental en la documentación de una arquitectura de sistema. Una arquitectura de sistemas es una entidad compleja que no se puede describir en un plano dimensional simple (Clements, et al.; 2002).

Para entender más a fondo de dónde viene este término, se detalla una historia breve sobre las vistas.

David Parnas observó que el software consiste en muchas estructuras, dónde él definió una colección parcial mostrando una colección de partes del sistema y mostrando las relaciones que había entre las partes (Parnas, 1979). Clements & Northrop indicaron que los frameworks son estructuras o plantillas definidas para desarrollar otro proyecto de software con la misma arquitectura (Clements & Northrop, 1996).

En resumen, una vista es un conjunto de elementos y requerimientos que se extraen de las partes interesadas, aunque esta información es abstracta, pero se desarrolla hasta hacerla realidad.

La información contenida en la Tabla 1, “Marco de Referencia Arquitectónicos”, podemos observar de manera general seis marcos de referencias arquitectónicos; estos marcos la mayoría cuentan con tres o más vistas. Como expresa (Hilliard, 1999).

Tabla 1. Marco de Referencia Arquitectónicos

Zachman (Niveles)	TOGAF (Arquitecturas)	4+1 (Vistas)	[BRJ99] (Vistas)	POSA (Vistas)	Microsoft (Vistas)
Scope	Negocios	Lógica	Diseño	Lógica	Lógica
Empresa	Datos	Proceso	Proceso	Proceso	Conceptual
Sistema lógico	Aplicación	Física	Implementación	Física	Física
Tecnología	Tecnología	Desarrollo	Despliegue	Desarrollo	
Representación		Casos de uso	Casos de uso		
Funcionamiento					

Fuente: (Hilliard, 2012)

El marco de referencia para la arquitectura de Zachman identifica 36 vistas en la arquitectura, con seis niveles (los que se muestra en la imagen anterior). Este marco es muy rígido y los últimos años no ha tenido mucho sustento, pero cuenta con tres vistas (conceptual, lógico y física); que han sido tomadas en otros marcos arquitectónicos (Hilliard, 2012).

Richard Hilliard en su resumen de los marcos arquitectónicos nos presenta el siguiente referente arquitectural llamado TOGAF (The Open Group); mostrando cuatro componentes principales, uno de los cuales es un framework de alto nivel que a su vez define cuatro vistas (Hilliard, 1999).

En (Kruchten P, 1995) propuso su conocido modelo “4+1”, que define cuatro vistas diferentes de la arquitectura de software.

Seguidamente el marco BRJ desarrolló su modelo arquitectónico con cinco vistas conectadas que conforman la arquitectura de software, (Booch & Jacobson, 1999).

Buschmann, Henney y Schmidt (2007) presentan una lista de vistas conocida como POSA, este marco es muy parecido al de Philippe Kruchten, pero este no hace mucho énfasis en la vista de escenarios.

Y por último en el resumen de marcos arquitectónicos se tiene la arquitectura de Microsoft define cuatro vistas o también llamado arquitectura , estas son de Negocios, de Aplicación, de Información y Tecnología (Platt, 2002).

2.2.7 Lenguajes de descripción arquitectónica (ADL)

Los Lenguajes de Descripción de Arquitecturas (ADLs), ocupan una parte importante del trabajo arquitectónico; varios investigadores han realizado muchas propuestas de Lenguajes de Descripción Arquitectónica.

Bass y otros, señalan que “las principales características de los ADL son componentes y conectores e incluyen reglas y recomendaciones para arquitecturas bien formadas. (...) Sin embargo, los ADLS solamente pueden ser comprendidos por expertos del lenguaje” (Clements, et al., 2002, p.140).

Una idea de Shaw y Garlan (1990) leída en una publicación de (Kicillof & Reynoso, 2004) señalan que en los ADLs los conectores sean desarrollados como entidades de primera clase y ellos afirman que un ADL genuino tiene que suministrar algunos componentes como la composición, abstracción, reusabilidad, configuración, heterogeneidad y análisis por lo que esto haría es excluir a todos los lenguajes de programación.

Los ADLs permite que se pueda modelar una arquitectura mucho antes que se inicie con la programación que la componen, además permite analizar su adecuación, determinar los diferentes puntos críticos y representar su comportamiento (Palmero, Martínez y Grass, 2018, pp. 143-157).

2.2.8 Procesos y metodologías

En los diferentes marcos las vistas estáticas se corresponden con las perspectivas particulares de los diferentes involucrados, mientras que las vistas dinámicas tienen que ver con etapas del proceso, ciclo de vida o metodología, caracterizadas como requerimiento, análisis, diseño (o construcción, o modelado), implementación, integración (prueba de conformidad, testing, evaluación) (Reynoso, 2004).

En la Arquitectura de software, podemos encontrar las siguientes metodologías: RUP, RAD, RDS, ARIS, PERA, CIMOSA, GRAI, GERAM, CMM. Sin embargo, a lo largo de los años se ha venido trabajando en esta área con el fin de elaborar nuevos métodos más específicos. (Reynoso, 2004).

Según el escritor Carlos Billy Reynoso en (Reynoso, 2004) algunos de esos métodos son los siguientes:

- Architecture Based Design (ABD)
- Software Architecture Analysis Method (SAAM)
- Quality Attribute Workshops (QAW)

- Quality Attribute-Oriented Software Architecture Design Method (QASAR)
- AttributeDriven Design (ADD)
- Architecture Tradeoff Analysis Method (ATAM)
- Active Review for Intermediate Design (ARID)
- Cost-Benefits Analysis Method (CBAM)
- FamilyArchitecture Analysis Method (FAAM)
- Architecture Level Modifiability Analysis (ALMA)
- Software Architecture Comparison Analysis Method (SACAM)

2.2.9 Patrones de diseño de la arquitectura de software

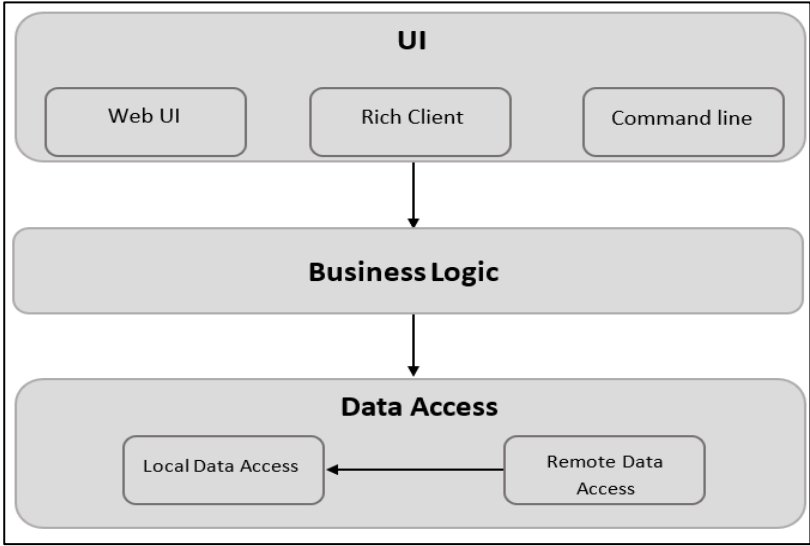
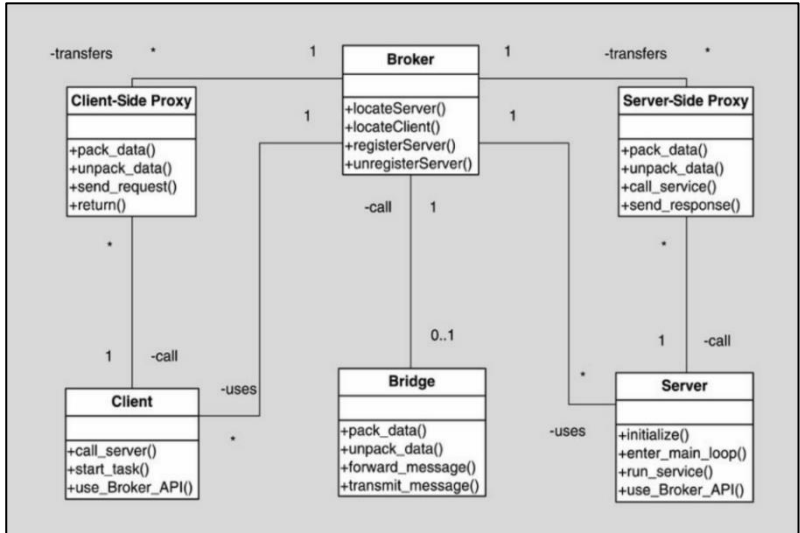
Cómo se mencionó anteriormente las estructuras arquitectónicas se pueden dividir en tres categorías principales, la estructura de módulos (modules), la de componentes y Conectores, (C&C) y la de asignación (Allocation).

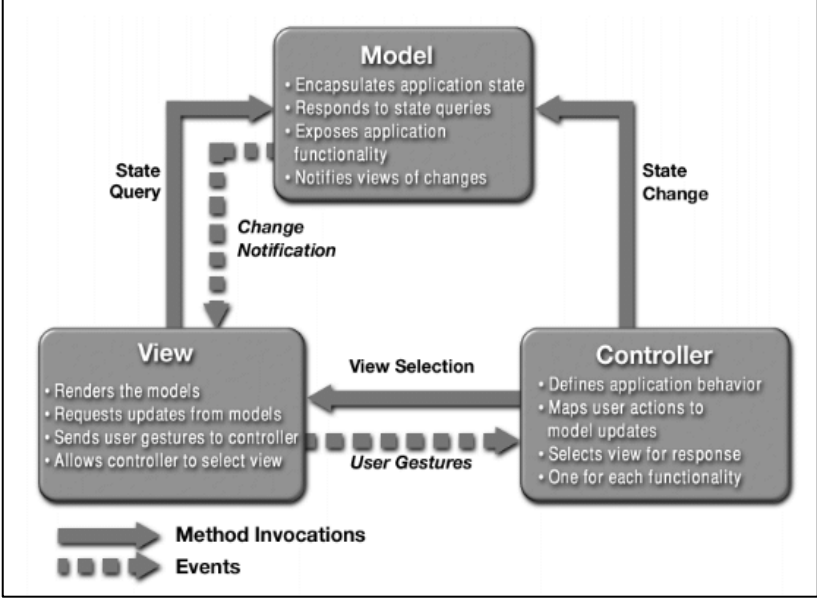
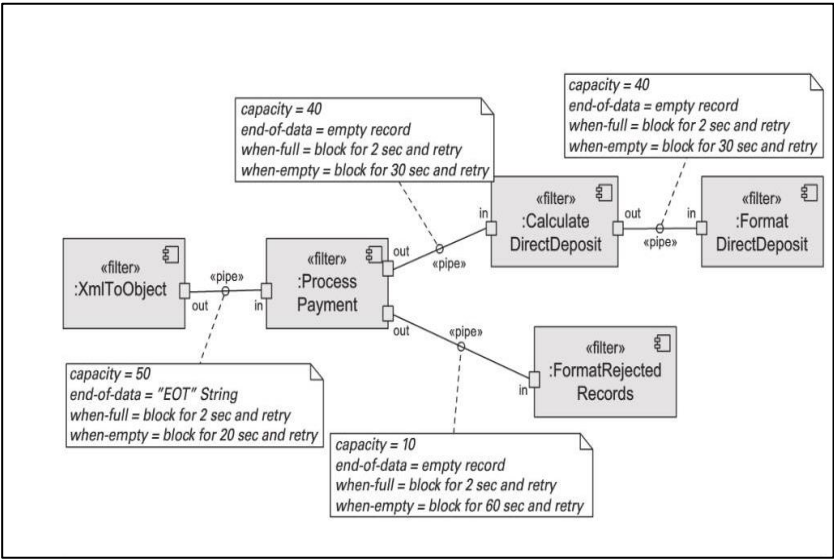
- **Estructura de los módulos:** Incorporan decisiones sobre cómo se puede estructurar el sistema como un conjunto de unidades de datos que se deben construir o adquirir.
- **Estructura de Componentes y Conectores:** Incorporan decisiones sobre cómo se estructurará el sistema como un conjunto de elementos que tienen un comportamiento en tiempo de ejecución (componentes) e interacciones (conectores).

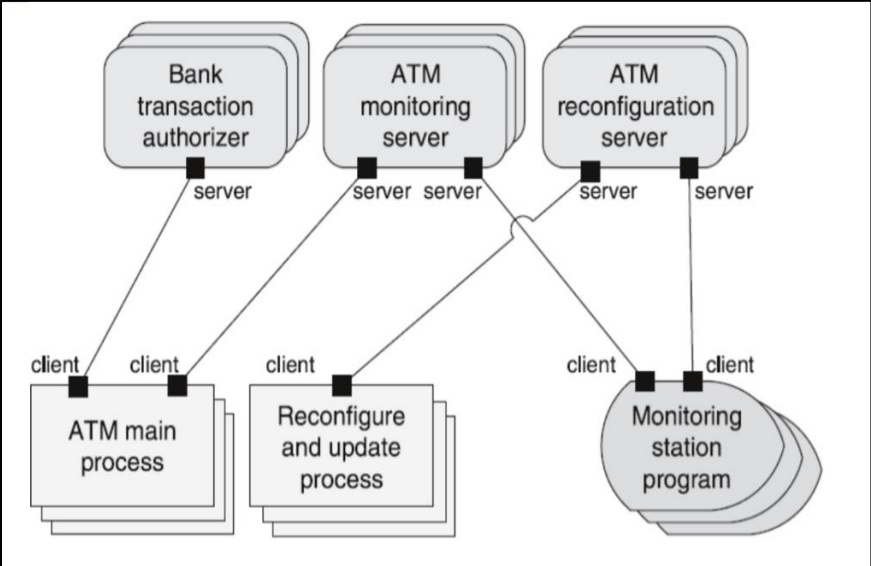
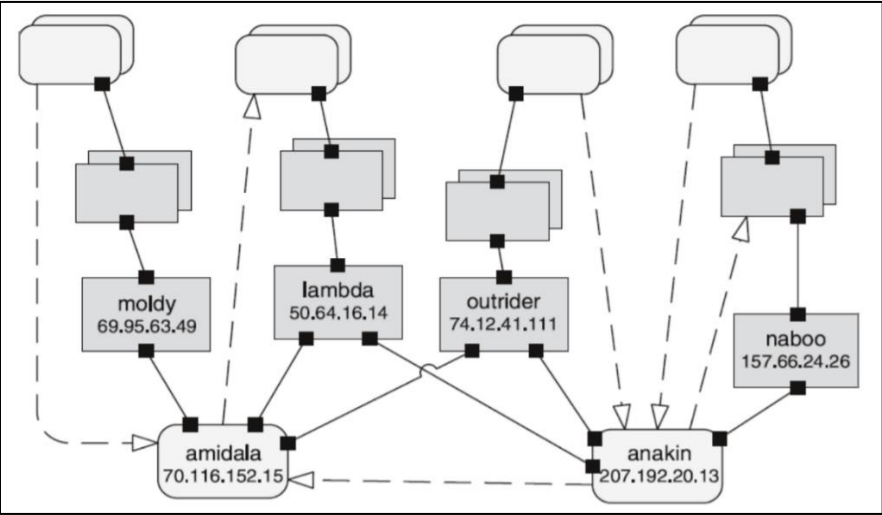
- **Estructura de Asignaciones:** Representan decisiones sobre cómo el sistema se relacionará con las estructuras que no son de software en su entorno, en los que se ejecuta el software (como CPU, sistemas de archivos, redes, equipos de desarrollo). (Clements, Garlan, Little, Nord, & Stafford, 2003).

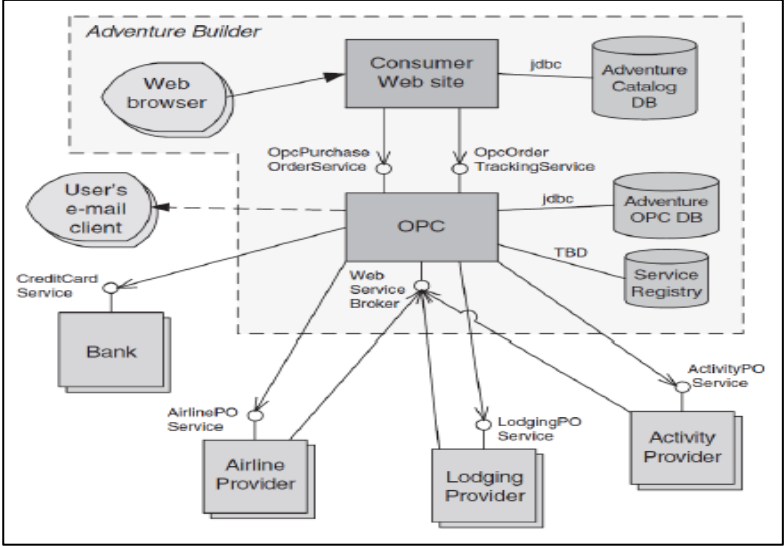
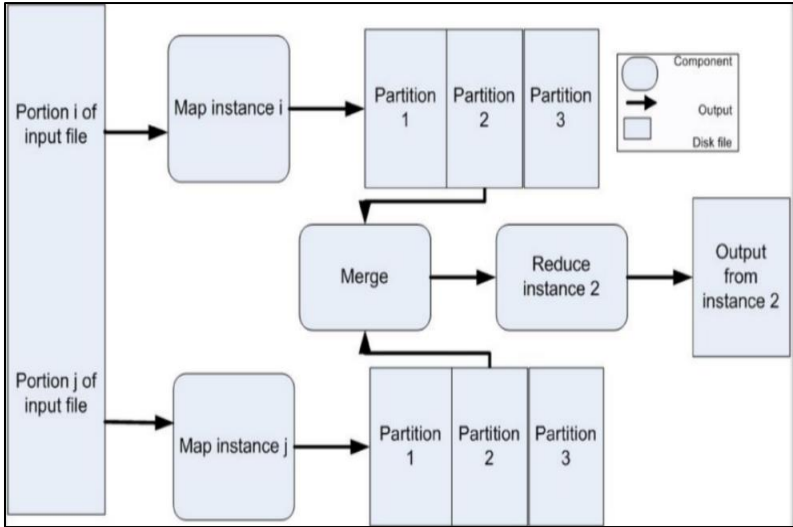
En la Tabla 2, “Patrones de Diseño de Arquitectura de Software”, podemos observar un resumen de 9 estructuras de patrones de la Arquitectura de Software, con la respectiva estructura a la que pertenecen y un ejemplo de ellas.

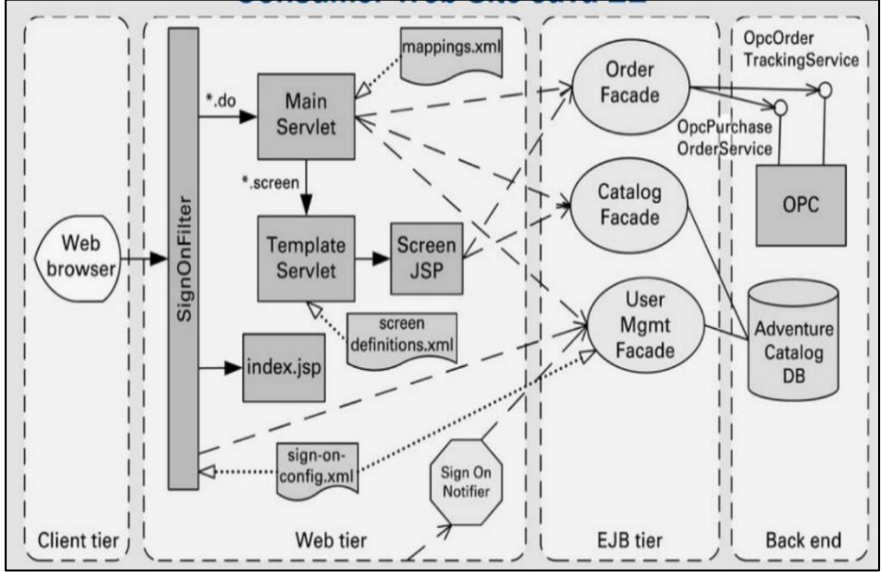
Tabla 2. Patrones de diseño de arquitectura de software

Patrón de diseño	Ejemplo
Layered Pattern (Module)	 <pre> graph TD subgraph UI WebUI[Web UI] RichClient[Rich Client] CommandLine[Command line] end subgraph BusinessLogic[Business Logic] end subgraph DataAccess[Data Access] LocalDataAccess[Local Data Access] RemoteDataAccess[Remote Data Access] end UI --> BusinessLogic BusinessLogic --> DataAccess RemoteDataAccess --> LocalDataAccess </pre>
Broker Pattern (C&C)	 <pre> classDiagram class ClientSideProxy { +pack_data() +unpack_data() +send_request() +return() } class Broker { +locateServer() +locateClient() +registerServer() +unregisterServer() } class ServerSideProxy { +pack_data() +unpack_data() +call_service() +send_response() } class Client { +call_server() +start_task() +use_Broker_API() } class Bridge { +pack_data() +unpack_data() +forward_message() +transmit_message() } class Server { +initialize() +enter_main_loop() +run_service() +use_Broker_API() } ClientSideProxy "1" -- "*" Broker : -transfers Broker "1" -- "*" ClientSideProxy Broker "1" -- "*" ServerSideProxy : -transfers ServerSideProxy "*" -- "1" Broker Client "1" -- "*" ClientSideProxy : -call Client "*" -- "1" Broker : -uses Broker "1" -- "0..1" Bridge : -call Bridge "0..1" -- "1" Broker Bridge "*" -- "1" Server : -uses Server "1" -- "*" ServerSideProxy : -call </pre>

Patrón de diseño	Ejemplo
MVC Pattern (C&C)	 The diagram illustrates the MVC (Model-View-Controller) pattern with the following components and interactions: Model: - Encapsulates application state - Responds to state queries - Exposes application functionality - Notifies views of changes View: - Renders the models - Requests updates from models - Sends user gestures to controller - Allows controller to select view Controller: - Defines application behavior - Maps user actions to model updates - Selects view for response - One for each functionality Interactions: State Query: A solid arrow from View to Model. Change Notification: A dashed arrow from Model to View. State Change: A solid arrow from Controller to Model. View Selection: A solid arrow from Controller to View. User Gestures: A dashed arrow from View to Controller. Legend: Method Invocations: Represented by solid arrows. Events: Represented by dashed arrows.
Pipe and Filter Pattern (C&C)	 The diagram illustrates the Pipe and Filter pattern for processing payments. It consists of the following filters and their connections: Filter 1: «filter» :XmlToObject - Capacity: 50 - End-of-data: "EOT" String - When-full: block for 2 sec and retry - When-empty: block for 20 sec and retry Filter 2: «filter» :Process Payment - Capacity: 40 - End-of-data: empty record - When-full: block for 2 sec and retry - When-empty: block for 30 sec and retry Filter 3: «filter» :Calculate DirectDeposit - Capacity: 40 - End-of-data: empty record - When-full: block for 2 sec and retry - When-empty: block for 30 sec and retry Filter 4: «filter» :Format DirectDeposit - Capacity: 40 - End-of-data: empty record - When-full: block for 2 sec and retry - When-empty: block for 30 sec and retry Filter 5: «filter» :FormatRejected Records - Capacity: 10 - End-of-data: empty record - When-full: block for 2 sec and retry - When-empty: block for 60 sec and retry Flow: The data flows from left to right through the filters, connected by pipes. The output of one filter is the input for the next.

Patron de diseño	Ejemplo
Client Server Pattern (C&C)	
Peer to Peer Pattern(C&C)	

Patrón de diseño	Ejemplo
Service Oriented Architecture Pattern (C&C)	
Map-Reduce-Pattern (Allocation)	

Patrón de diseño	Ejemplo
Multi-tierPattern	 The diagram illustrates a Multi-tierPattern architecture divided into four distinct tiers, each enclosed in a dashed box: Client tier: Contains a Web browser. Web tier: Contains a SignOnFilter, Main Servlet, Template Servlet, Screen JSP, and index.jsp. It also includes configuration files like mappings.xml, screen definitions.xml, and sign-on-config.xml. EJB tier: Contains Order Facade, Catalog Facade, and User Mgmt Facade. Back end: Contains OPC (Order Processing Component), Adventure Catalog DB, and Sign On Notifier. Arrows indicate the flow of data and dependencies between these components across the different tiers.
Fuente: (Clements, Garlan, Little, Nord, & Stafford, 2003)	

2.2.10 Ciclo del desarrollo de la Arquitectura de Software

Los tres pasos principales para el desarrollo de una Arquitectura de Software son determinar los requisitos arquitecturales, diseñar la arquitectura y evaluar la arquitectura.

En la Figura 3, “Ciclo de vida de la Arquitectura de Software, (AS)” se muestra el ciclo de desarrollo de una Arquitectura de Software.

Primeramente, se debe de conocer el concepto de Arquitectura de Software e investigar el modelo de negocio donde se desarrollará el diseño arquitectural, esto con el fin de levantar y

conocer los requerimientos iniciales. Luego se va evaluando cada avance del diseño y con ello refinando los detalles. Es necesario recordar que estas reuniones se realizan las veces que sean necesarias hasta que el informe final esté listo.

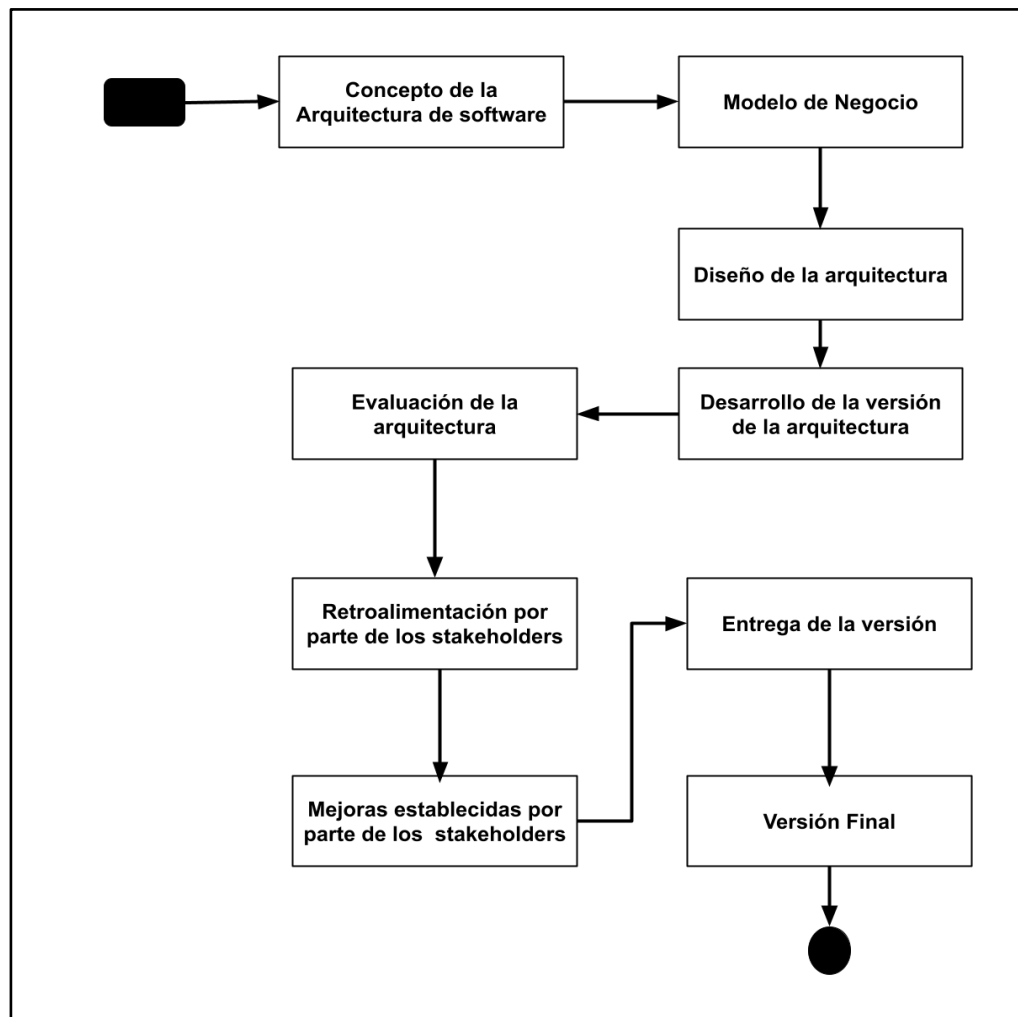


Figura 3. Ciclo de Vida de la Arquitectura de Software (AS).

Fuente: Elaboración propia basado en Arquitectura de Software. (Cervantes, 2015).

Como mencionamos anteriormente el ciclo de desarrollo de una AS tiene muchos sub-pasos, que son:

a. Requerimientos de la Arquitectura

En esta etapa se debe de elegir el método para determinar los requisitos arquitecturales que describen algunas funcionalidades o atributos de calidad de sistema de software. Este tipo de métodos son los que se describió anteriormente en la sección de Procesos y metodologías.

Las más utilizadas según investigaciones realizadas es el Quality Attribute Workshops (QAW), AttributeDriven Design (ADD), Architecture Tradeoff Analysis Method (ATAM), Active Review for Intermediate Design (ARID).

b. Diseño de la Arquitectura

Es la etapa donde se determina la estructura mediante la toma de decisiones de diseño.

En la sección anterior llamada *Patrones de diseño de la Arquitectura de Software*, se documentó estos patrones.

c. Documentación de la Arquitectura

Esta etapa se centra en la generación de documentos donde se escribe detalladamente la estructura de la Arquitectura.

La documentación de la Arquitectura se basa en elaborar un documento, con el propósito de comunicar la información de un proceso o de procesos, donde esto permite que la información persista. (Cervantes,2016).

d. Evaluación de la Arquitectura

La Evaluación de la Arquitectura del sistema, lo que permite es la detección del incumplimiento y los riesgos asociados.

Por lo que es importante que no se desvíe de manera significativa de los requerimientos funcionales, atributos de calidad y de restricciones.

De acuerdo con (Bass, Clements, Kazman, & Klein, 2008, p. 30), existen tres razones por las cuales la evaluación de la arquitectura de sistema es importante para los grandes y complejos sistemas de software:

La arquitectura sirve como medio de educación; consiste en introducir personas al equipo de trabajo, analistas, nuevos arquitectos o clientes, quien está buscando soluciones.

La arquitectura sirve como puente de comunicación con los involucrados; ya que todos deben estar involucrados y debe de haber un medio de comunicación primario.

La arquitectura sirve como la base para analizar y construir el sistema, reunir los objetivos de calidad del sistema.

Abonado a lo anterior, para algunos autores la documentación de Arquitectura fue creada para reunir las necesidades de los involucrados (Rozanski & Woods, 2005).

Existen muchos métodos de evaluación de arquitecturas. Hay algunos de ellos que analizan específicamente un atributo de calidad. Algunos de estos métodos son primeramente el método ALMA (Architecture Level Modifiability Analysis); el único atributo de calidad que se centra este método es en la modificación. PASA (Performance Assessment of Software Architecture), analiza el desempeño y SALUTA (Scenario based Architecture Level Usability Analysis); se basa en el uso del sistema.

Existen otros métodos que abarcan el análisis de muchos atributos de calidad y cuentan con fases muy elaboradas en el proceso de evaluación. Estos son algunos de estos métodos, ATAM (Architecture Trade-off Analysis Method), SAAM (Software Architecture Analysis Method) y ARID (Active Reviews for Intermediate Design).

2.3 Referentes Teóricos de Instrumentos

En esta sección se presentan las técnicas utilizadas en este proyecto. Cada técnica se aborda por medio de una breve descripción, los usos normalmente dados, las principales fortalezas y debilidades.

2.3.1 Revisión Literaria

De acuerdo con los autores, Hernández, Fernández y Baptista definen la RL de la siguiente manera:

“La Revisión Literatura implica detectar, consultar y obtener la bibliografía (referencias) y otros materiales que sean útiles para los propósitos del estudio, de donde se tiene que extraer y recopilar la información relevante y necesaria para enmarcar nuestro problema de investigación”. (Hernández, Fernández, & Baptista, 2014, p.94)

Una Revisión de Literatura (RL) crea una base científica para el avance del conocimiento. Además, facilita el desarrollo de la teoría, cierra las áreas donde hay una gran cantidad de investigación y descubre las áreas donde se necesita más investigación (Webster & Watson, 2002).

Además, otra características e importancia de una RL, es que usa las ideas contenidas en la literatura para justificar un enfoque particular sobre un tema específico, es la selección de métodos utilizados en el desarrollo de la RL. (Hart, 2018).

Según Danhke, (1989) uno de los propósitos de la revisión Literatura es analizar y discernir si la teoría existente y la investigación anterior sugieren una respuesta a las preguntas de investigación o proporciona una dirección al planteamiento del estudio actual.

Para comenzar a planear un estudio de este tipo Cooper sugiere el uso de su taxonomía sobre RL Cooper's (1988).

La propuesta de Cooper (1988), cuenta con un proceso de varias etapas, las cuales son:

- Formulación del problema.
- Recolectar datos.
- Evaluar los datos.

- Analizar e interpretar los datos.
- Presentación.

Para una mayor comprensión una RL puede dividirse en tres grandes e importantes etapas.

a-Buscar y evaluar fuentes de información: En esta etapa se debe considerar los siguientes aspectos; un criterio de búsqueda, que establezca el alcance de la revisión de literatura, tales como: el rango de fechas, paradigmas específicos, el interés del investigador, las regiones geográficas, los medios de publicación (revistas, procedimientos, etc.) y el tipo de fuente (primaria, secundaria, ambas).

b- Lectura y extracción de información relevante: Según Levy y Ellis (2006), esta etapa abarca muchas actividades. Inicialmente, se debe de conocer la información que se busca, para luego definirla, describirla e identificarla. Posteriormente se debe de comprender lo que se investiga, para analizar y hacer una separación de la información.

Se prosigue con el resumen de la información y para finalizar se debe de realizar el proceso de evaluación, donde se examiné la información recopilada y se establezca un juicio de valor.

c- Escribir la Revisión Literaria: Es importante señalar que en el proceso de la ejecución de este instrumento se debe de realizar otras actividades sumamente importantes como lo

es investigar con un experto sobre el tema que se desea investigar además de identificar palabras claves o términos de búsqueda, como el uso de ³tesauros.

Por cada aspecto o idea se debe agregar al menos 3 referencias. De cada referencia que se usó se identificó los fundamentos, objetivos, una breve metodología, resultados en el documento, resumen de la información en la RL y luego se debe de parafrasear o citar textualmente, con su respectivo proceso.

2.3.2 Encuesta

Una encuesta es un método de configuración independiente del entorno (Wynekoop, Senn, & Conger, 1992) orientado a recopilar datos utilizando diferentes técnicas, como correos electrónicos o cuestionarios en línea, entrevistas telefónicas, formularios, etc. Después de la recopilación de datos, es posible utilizar técnicas estadísticas para ayudar en la interpretación de los principales hallazgos (Gable, 1994).

Las encuestas tienen varias fortalezas y debilidades. Las principales fortalezas son:

- Alternativa práctica y económica.
- Ayuda a verificar hipótesis.

³ Tesauro es lista controlada y estructurada de términos para el análisis temático y la búsqueda de documento.

Tomado de <http://vocabularies.unesco.org/browser/thesaurus/es/>

- Confianza en la generalización.

Una encuesta es un enfoque práctico y económico. Este método permite la recopilación de datos de un pequeño o gran número de participantes utilizando diversos métodos de manera relativamente discreta. Además, una encuesta ayuda a verificar una hipótesis previamente estudiada en contextos particulares (Casas Anguita, Donado Campos, & Repullo Labrador, 2002).

Las encuestas ayudan a realizar contextos alternativos y los resultados son comparados con estudios anteriores, con ello se verifica las hipótesis y una exploración de teorías generales. Por lo tanto, el método de encuesta proporciona un alto nivel de confianza en la generalización de sus resultados (Jick, 1979).

Esta confianza en la generalización se basa en el uso de muestras representativas que permiten el descubrimiento de relaciones entre las variables consideradas en las encuestas. Como consecuencia de esta capacidad de generalización, las encuestas pueden aumentar la previsibilidad (Vidich & Shapiro, 2017).

Las principales debilidades de las encuestas son:

- Alcance limitado.
- Muestreo difícil.
- Exploración profunda limitada.

El alcance de las encuestas es limitado porque este método ayuda a comprender un fenómeno particular en un contexto y tiempo específico (Gable 1994). De cierta manera, estos resultados representan una "imagen" que es desactualizada en el momento en que ha terminado.

La discusión principal es cómo definir el tamaño de la muestra, como debería ser la composición de la muestra y el método de selección de los participantes. Posteriormente, las encuestas evitan la exploración profunda (Vidich & Shapiro, 2017).

2.3.3 Prototipo

Los prototipos han adquirido mucho protagonismo y visibilidad en los últimos tiempos. El desarrollo de software es uno de los casos con más auge, donde el lanzamiento de versiones no estables de programas se ha convertido en algo común (Kelty, 2008).

Es un producto que se construye para ser ensayado y evaluado intensivamente para observar si se le han incorporado todas las características deseadas. Un prototipo es como construir un sistema corregido (Kenneth E & Kendall, 2005).

Cómo se mencionó anteriormente el prototipo es una representación de un producto, donde se puede evaluar, luego probar para explorar su funcionamiento y uso y con ello generar un diseño de calidad.

Según Kenneth E & Kendall, (2005), una de las ventajas de la técnica de prototipo es:

- Poder modificar el sistema en las primeras etapas del desarrollo.
- Poder suspender el desarrollo del sistema si este no es funcional.

- Poder desarrollar un sistema que se asemeje a los requerimientos de los involucrados.

Algunas de sus desventajas son:

- Dificultad de manejar prototipos en sistemas muy grandes.
- Decisiones de implementación poco convenientes donde el desarrollador elige un lenguaje de programación poco conveniente.

2.3.4 Método de Análisis de Compensación de Arquitecturas (ATAM)

Uno de los métodos y herramientas más maduros para la evaluación de Arquitectura de Software, es el Método de Análisis de Compensación de Arquitectura (ATAM). Para este proyecto se eligió ATAM entre otros métodos de evaluación de AS como SAAM, CBAM, AIMA y FAAM.

El método ATAM permite validar una arquitectura mientras aún es un diseño en papel antes de que se haya convertido en código (Clements, Garlan, Little, Nord, & Stafford, 2003).

Las principales características de ATAM son las siguientes, está basada en escenarios que se enfoca en evaluar la arquitectura en los atributos de calidad. La ejecución de ATAM debe realizarse en un lapso corto con los recursos disponibles y los requisitos más relevantes. Uno de los propósitos de este método es no crear un análisis preciso de los atributos si no identificar si el enfoque arquitectónico es el preciso (Clements, Garlan, Little, Nord, & Stafford, 2003).

Kazman, Klein, & Clements, (2000) indica textualmente que “el propósito del ATAM es evaluar las consecuencias de las decisiones arquitectónicas a la luz de los requisitos de los atributos de calidad” (p.2).

El proceso de la evaluación de este método permite descubrir los riesgos, los puntos sensibles (Sensitive Point) y los puntos de compromiso o compensación (Tradeoff Point) ligado esto a las decisiones arquitectónicas.

Como se muestra en la *Figura 4*, “Flujo de Trabajo ATAM”, los componentes del modelo de negocio con la documentación de Arquitectura de software trabajan en conjunto, se toma los escenarios, los atributos de calidad y las decisiones arquitectónicas, para realizar un análisis. Esta evaluación da como resultado los puntos de sensibilidad, de compensación, los puntos de riesgo y no riesgo, si estos datos arrojados muestran un punto de riesgo se debe realizar nuevamente el proceso de evaluación. ATAM busca medir los atributos de calidad, para ello el mecanismo utilizado para obtener esta información es el escenario.

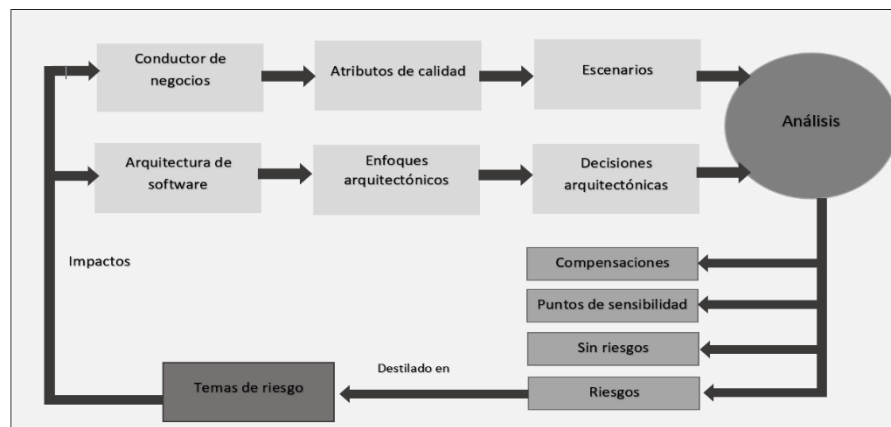


Figura 4. Flujo de Trabajo ATAM.

Fuente: Elaboración propia basado en *Evaluating Software Architecture* Paul Clements, Rick Kazman and Mark.

Un escenario describe la interacción entre un stakeholder (interesados) y el sistema, siendo el stakeholder el involucrado en el desarrollo, o el cliente del sistema. Los escenarios permiten ver las necesidades de un sistema dentro de su desarrollo (Clements, Garlan, Little, Nord, & Stafford, 2003).

Dentro de los escenarios se puede encontrar escenarios de casos de uso que involucran usos típicos del sistema como escenarios de crecimiento que se basa en todo el proceso desde su inicio hasta los posibles cambios en un futuro y escenarios de exploración que abarcan los cambios extremos al sistema.

La información que los escenarios proveen es evaluada en ATAM utilizando dos mecanismos diferentes en distintos momentos y con la participación de diferentes interesados e involucrados, estos son los árboles de utilidad y la lluvia de ideas estructuradas.

En la Tabla 3, “Fases del Método de Análisis de Compensación de Arquitecturas (ATAM)”. Se puede observar las cuatro fases con sus nueve respectivos pasos que corresponde la evaluación del Método de Análisis de Compensación de Arquitectura.

Tabla 3. Fases del Método de Análisis de Compensación de Arquitecturas (ATAM)

Fases	Pasos
Presentación	1-Presentar el ATAM.
	2-Presentar las metas de Negocio.
	3-Presentar la arquitectura.
Investigación y Análisis	4-Identificación de los enfoques arquitectónicos.
	5-Generar árbol de utilidad.
	6-Analizar el enfoque arquitectónico.
Pruebas	7-Lluvia de ideas y priorización de escenarios.
	8-Analizar enfoque arquitectónico del paso 6 con el ranking del paso 7.
Informes	9-Presentación de Resultados.

Fuente: Elaboración propia basado en (Clements, Kazman, & Klein, Evaluating software architectures, 2003)

Se puede observar las cuatro fases con sus nueve pasos respectivos que corresponde la evaluación del Método de Análisis de Compensación de Arquitectura.

Para entender la evaluación en la Tabla 4, “Resumen sobre los pasos del ATAM”, se describe lo que este método realiza en cada paso.

Tabla 4. Resumen Sobre los Pasos del ATAM

Pasos	Definiciones
1-Presentación de ATAM	Se presenta el método a los evaluadores y Stakeholders.
2-Presentación de los conductores del negocio:	Se presenta las metas de negocio que motivan el esfuerzo de desarrollo y que deberán ser brindados por la arquitectura.
3-Presentación de la arquitectura:	Presentar la arquitectura haciendo énfasis en cómo se abordan los objetivos de negocio
4-Identificación de enfoques arquitectónicos	Los arquitectos presentan los enfoques arquitectónicos para este dominio / arquitectura para hacer frente a las metas del sistema y los objetivos de negocio
5-Generar árbol de utilidad de atributo de calidad:	Los factores de calidad (fiabilidad, disponibilidad rendimiento) son identificados y especificados hasta el nivel de escenarios, con estímulos y respuestas y después priorizados.
6- Analizar enfoques arquitectónicos	Los Stakeholders y el arquitecto analizan como los enfoques arquitectónicos afectan a los factores identificados en el paso anterior.

7- Lluvia de ideas y priorización de escenarios	Si es necesario se generan los nuevos escenarios más detallados y se priorizan junto a los del árbol de utilidad.
8- Analizar los enfoques arquitectónicos	Se repite el paso 6 pero considerando los escenarios de alta prioridad definidos en el punto y se decide que cambios se han de acometer sobre la arquitectura.
9-Presentar los resultados.	Se presenta el informe de resultados.

Fuente: Elaboración propia basada en (Clements, Kazman, & Klein, Evaluating software architectures, 2003)

2.4 Descripción de Tres Arquitecturas de Software

En este apartado se documenta los conceptos de tres arquitecturas puntuales, enfatizando la estructura de documentación de cada arquitectura. Las arquitecturas que se describirán a continuación son: Krutchen 4+1, BMP (Gestión de Procesos de Negocios) y la arquitectura de Servicios Web, (WS).

En el Capítulo IV se elige y se desarrolla el prototipo de las arquitecturas elegidas que cumplieron con los requerimientos, esto se realiza por medio de una documentación de Arquitectura.

2.4.1 Arquitectura del Modelo Kruchten 4+1

Este modelo fue diseñado por Philippe Kruchten (Kruchten P, 1995) lo define como “El modelo de vista 4 + 1, donde organiza una descripción de una arquitectura de software utilizando cinco vistas simultáneas, cada una de las cuales aborda un conjunto específico de inquietudes” (p. 45).

Los arquitectos capturan sus decisiones de diseño en cuatro vistas y utilizan la quinta vista para ilustrarlas y validarlas” (Kruchten, P. B, (1995).

En la *Figura 5*, se muestra el “Modelo 4+1 Vistas”, modelo arquitectónico que consta de 5 vistas. Cada vista se describe en lo que se llama “diagrama” que usa su notación particular.

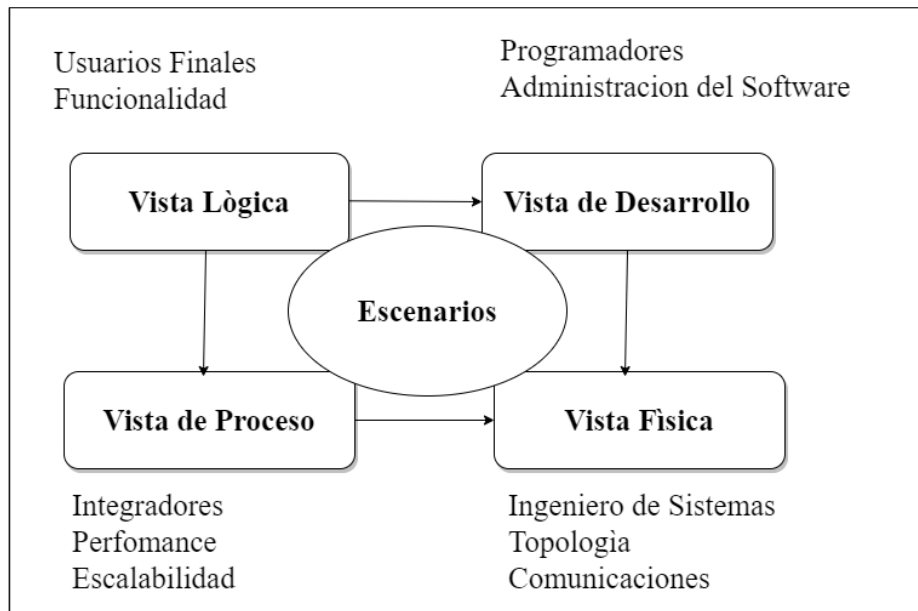


Figura 5. Modelo de "4+1" Vistas.

Fuente: (Kruchten P. , Modelo de "4+1" vistas, 1995)

En la *Figura 6*, se puede observar la "Estructura de Documentación de Arquitectura Modelo Kruchten 4+1", donde se muestra el desglose que abarca una documentación de Arquitectura de Software basado en el modelo 4+1.⁴

⁴ La estructura de documentación del modelo 4+1 se desarrolla en el capítulo IV "Propuesta de Solución"

Historial de cambios
Tabla de contenidos
Lista de Figuras
1.Alcances
2.Referencias
3.Arquitectura de Software
4. Objetivos de la Arquitectura y Restricciones
5. Arquitectura de la vista Lógica
6. Arquitectura de la vista de procesos
7.Arquitectura de la vista de Desarrollo
8.Arquitectura de la vista Física
9. Escenarios
10. Tamaño y performance
11.Calidad
Apéndices
A. Acrónimos y Abreviaciones
B. Definciones
C. Principios de diseño

Figura 6. Estructura de Documentación de Arquitectura Modelo Kruchten 4+1.

Fuente: PB Kruchten Software, IEEE 12 (6), 42-50.The 4 + 1 view model of architecture.

2.4.1.1 Arquitectura de la Vista Lógica (Modelo 4+1)

Se basa principalmente en los requerimientos funcionales y lo que el sistema pueda proveer con su funcionalidad y servicio a los usuarios. Este permite la creación de clases y objetos mediante encapsulamiento y herencia.

Como lo establece Kruchten, la vista lógica describe el modelo de objetos del diseño cuando se utiliza un método de diseño orientado a objetos (Kruchten P. , 1995).

La descomposición de la vista permite que se analice y se documente cada aspecto con mayor detalle con respecto a las funcionalidades de cada clase y objeto.

Para la documentación de esta vista es recomendable incluir:

- Diagramas de clases.
- Diagramas de secuencias.
- Diagrama de comunicación.

En la *Figura 7*, se muestra los “Componentes de la Vista Lógica”, esta es la notación que se utiliza en la vista lógica se desprende de la notación Booch & Jacobson, (1999) la cual considera sólo los elementos de relevancia para la vista.

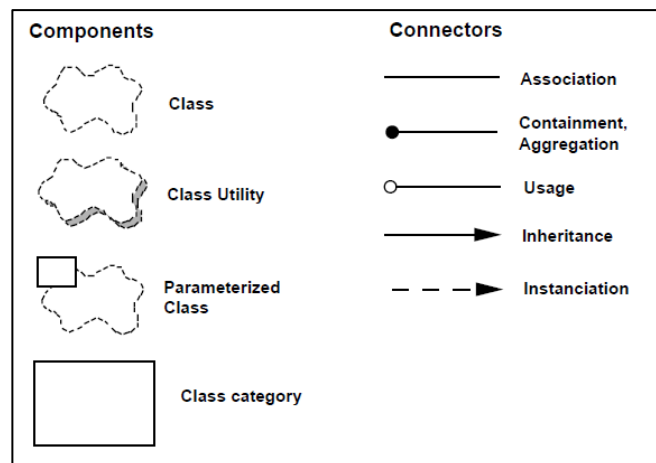


Figura 7. Componentes de la Vista Lógica.

Fuente: (Kruchten P.; 1995).

2.4.1.2 Arquitectura de la Vista de procesos. (Modelo 4+1)

En esta vista se busca documentar los procesos que se encuentran en el sistema, además de especificar su flujo de trabajo paso a paso y las funciones que se cumplen por los componentes dentro de esos procesos.

Kruchten destaca lo siguiente, la vista de proceso describe los aspectos de concurrencia y sincronización del diseño” (Kruchten, P. B, (1995) y sin dejar de lado los aspectos como la disponibilidad, integridad, la tolerancia a fallos, el rendimiento y la escalabilidad que los procesos puedan representar.

- Para la documentación de esta vista es recomendable incluir el Diagrama de Actividad.

En la *Figura 8*, se muestra los “Componentes de la Vista de Procesos.” Esta es la notación que se utiliza para la documentación de la arquitectura de la vista de procesos.

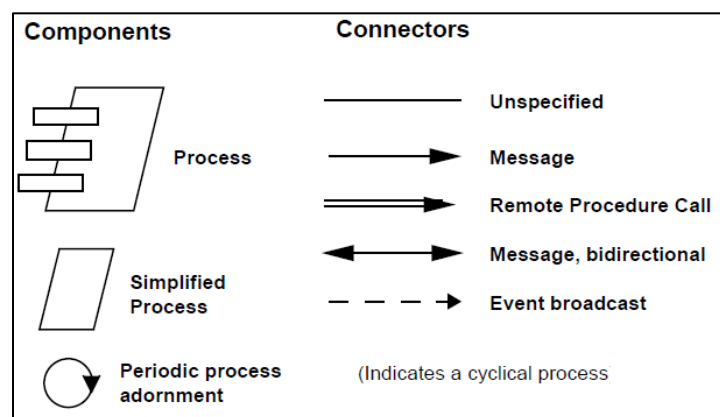


Figura 8. Componentes de la Vista de Procesos.

Fuente: (Kruchten P. , 1995)

2.4.1.3 Arquitectura de la Vista de desarrollo. (Modelo 4+1)

Esta vista incluye requisitos internos relativos a la facilidad de desarrollo, administración del software, reutilización y elementos comunes, y restricciones impuestas por las herramientas o el lenguaje de programación. Se describen los módulos del software creado por los desarrolladores. Estos módulos se crean por capas por lo que cada una debe ser especificada para su documentación.

La vista de desarrollo apoya los requisitos, la evaluación de costos, la planificación, el monitoreo de progreso del proyecto, y también funciona como una base para analizar la portabilidad y seguridad (Kruchten P, 1995).

Para la documentación de esta vista es recomendable incluir los siguientes:

- Diagrama de componentes.
- Diagramas de paquetes.

En la *Figura 9*, se muestra los “Componentes de la Vista de Desarrollo.” Esta es la notación que se utiliza para la documentación de la arquitectura de la vista de desarrollo.

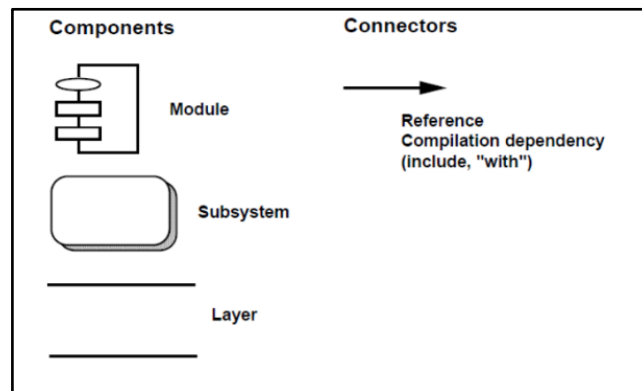


Figura 9. Componentes de la Vista de Desarrollo.

Fuente: (Kruchten P, 1995)

2.4.1.4 Arquitectura de la Vista Física. (Modelo 4+1)

La definición que establece Kruchten deja muy claro que esta vista se basa en el hardware y software y así como los componentes y sus conexiones físicas a diversos servicios con el cual se crea la perspectiva del proyecto.

Kruchten P, (1995) indica que “La vista física describe la asignación del software al hardware y refleja su aspecto distribuido” (p.45).

El software se ejecuta sobre una red de computadores o nodos de procesamiento (o solamente nodos). Los diversos elementos identificados como (redes, procesos, tareas y objetos) deben ser mapeados sobre los variados nodos. Por lo tanto, el mapeo del software en los nodos requiere ser muy flexible y tener un impacto mínimo sobre el código fuente en sí (Van Vliet, 2008).

Para la documentación de esta se recomienda incluir los siguientes:

- Diagrama de paquetes.
- Diagrama de Componentes.

En la *Figura 10*, se muestra los “Componentes de la Vista Física.” Esta es la notación que se utiliza para la documentación de la arquitectura de la vista física.

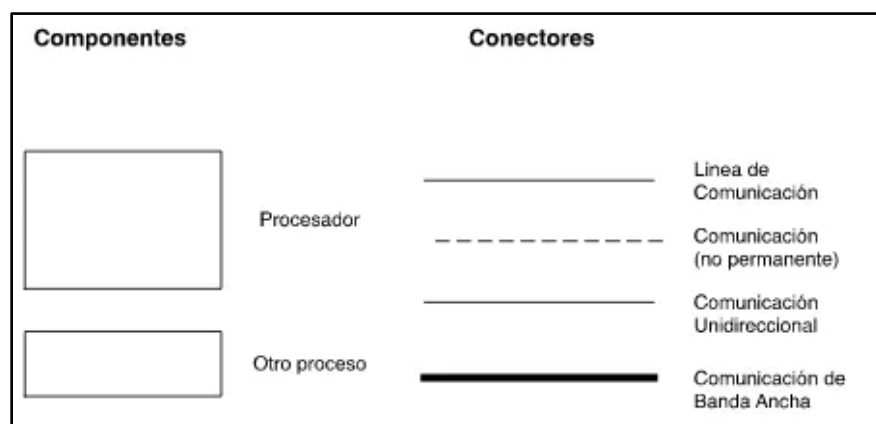


Figura 10. Componentes de la Vista Física.

Fuente: (Kruchten P., 1995).

2.4.1.5 Arquitectura de la Vista de Escenarios. (Modelo 4+1)

Las vistas trabajan en conjunto generando escenarios de casos de usos donde se pueden apreciar el aporte que cada vista ofrece y los diferentes procesos y relaciones que implica.

Estos escenarios nos permiten descubrir elementos arquitectónicos durante el diseño de arquitectura tal como se describirá más adelante.

En la Tabla 5, “Tabla Comparativa de vistas, Kruchten 4 +1” se muestran los factores para determinar cada vista y el entorno en que se desarrollan.

CAPÍTULO II

MARCO TEÓRICO

Tabla 5. Tabla Comparativa de Vistas, Kruchten 4 +1

Vista	Lógica	Procesos	Desarrollo	Física	Escenarios
Componentes	Clase	Tarea	Módulo, Subsistema	Nodo	Paso, Script
Conectores	Asociación Herencia contención	Rendez-vous Mensaje Broadcast, RPC	Dependencia de compilación, sentencia “with”, “include”	Medio de comunicación LAN, WAN, bus	
Contenedores	Categoría de clase	Proceso	Subsistema (biblioteca)	Subsistema físico	Web
Stakeholders	Usuario final	Diseñador Integrador	Desarrollador, administrador	Diseñador de sistemas	Usuario desarrollador
Intereses	Funcionalidad	Perfomance Disponibilida d Tolerancia a fallas Integridad	Organización, Reúso, portabilidad, líneas de productos	Escalabilidad Perfomance, Disponibilidad	Comprensibilida d
Herramientas	Rose	UNAS/SAL ES DADS	Apex, SoDA	UNAS, Openview, DADS	Rose

Fuente: (Kruchten P. , 1995)

2.4.2 Arquitectura del Modelo de Gestión de Procesos de Negocio (BPM)

La Arquitectura de Gestión de Procesos de Negocio, en inglés, Business Process Management, (BPM); se concentra en un enfoque integral entre los procesos, personas y tecnologías de la información (Deutsch, Hull, Batra, & Chen, 2016).

Para conocer a fondo el enfoque de este modelo arquitectónico se debe de conocer las tres dimensiones que la componen; la de procesos; como la dimensión de transformación, la de negocio, como dimensión de valor y la de gestión como la dimensión de capacitación. (Garimella, Michael & Bruce, 2008).

En la *Figura 11*, se muestra la “Estructura de documentación para Gestión de Procesos de Negocio. (BPM)”, donde se observa el desglose que abarca una documentación de Arquitectura de Software basado en la Arquitectura de Gestión Procesos de Negocio.

Historial de cambios
Tabla de contenido
Lista de Figuras y tablas
1- Definición de procesos
2- Modelo de negocio
3- Arquitectura de procesos
4- Modelado de procesos
5- Monitoreo de los procesos
6- Calidad y rendimiento de los procesos (KPI)
7- Modelo de madurez
8- Conclusiones
9- Referencias

Figura 11. Estructura de Documentación para Gestión de Procesos de Negocio. (BPM).
Fuente: (Benedict, y otros, 2019).

Los componentes fundamentales de esta arquitectura se detallan a continuación.

2.4.2.1 Procesos

Según la (Real Academia Española, 2001), el término de procesos se define como un “conjunto de las fases sucesivas de un fenómeno natural o de una operación artificial.” Por su parte Deutsch, lo define como un conjunto de tareas realizadas en una secuencia específica para lograr un objetivo común de negocio (Deutsch et al., 2016).

Por otra parte, con una definición más específica y científica, Hammer y Champy señalan que un proceso es como un “conjunto estructurado y medible de actividades que se desarrollan en una organización con el objetivo de conseguir un resultado concreto para un cliente o mercado específico” (Hammer & Champy, 1993. pp. 730-755).

Para los autores lo definen como todas las acciones que se realizan en una organización u empresa, con el fin de lograr un producto o servicio establecido.

2.4.2.2 Proceso de Negocio

Según Hammer & Champy (1993) define el concepto de Proceso de Negocio como “Un conjunto de actividades que toman uno o más tipos de ⁵inputs y crean output que será el valor de un cliente” (p.750).

La dimensión de negocio es la dimensión de valor, tanto para los clientes como para los interesados. Por lo que BPM, facilita los objetivos de negocio de la organización, haciendo mejoras en rendimiento e incentivando la innovación. Así como la satisfacción del cliente y elevar los niveles de eficiencia del personal.

2.4.2.3 Gestión de Proceso de Negocio. (BPM)

La Gestión de Procesos de Negocio, en inglés Business Process Management, es un modelo que abarca todos los procesos operativos y de negocios de la organización con el objetivo de buscar identificar, diseñar, ejecutar, documentar, monitorear, controlar y medir los procesos de negocios que una organización implementa (Garimella, Lees, & Williams, 2012).

Por su parte Dumas define el Modelo de Gestión de Procesos de Negocio como el arte y la ciencia de vigilar la forma como se realiza el trabajo en una empresa u organización, con el fin de

⁵ El término output (salidas) señala el producto que parte de la empresa o industria mientras que inputs (entradas) son el conjunto de factores o recursos que se necesita para cumplir con la producción.

asegurar resultados significativos y tomar ventaja de las oportunidades de mejora “ (Dumas, La Rosa, Mendling, & Reijers, 2013).

Asimismo, Harmon (2010) citado en (Rodríguez Zamuria, 2017) lo define como una disciplina de gestión centralizada en la mejora del rendimiento corporativo por medio de la gestión de procesos de negocio.

En resumen, BPM, es un diagrama que representa una secuencia de actividades, eventos, acciones, procesos y enlaces o puntos de conexión. Un modelo de procesos de negocio está conformado por procesos de Tecnologías de Información (TI) y procesos de personas.

En la *Figura 12*, se observa la “Arquitectura de Gestión de Proceso de Negocio (BPM)” donde se muestra las principales partes y las relaciones entre si de una arquitectura BPM.

La Arquitectura de la Gestión de Procesos funciona de la siguiente manera: el eje principal es la máquina de ejecución, esta se encarga de ejecutar los procesos por medio del lenguaje de orquestación del servicio BPEL. Posteriormente, los analistas de negocios y los analistas técnicos grafican estos procesos de flujos de trabajo por medio de una notación gráfica que permite el modelado de procesos de negocio, BPMN. Esta notación crea diagramas BPMN donde genera un lenguaje denominado BPEL XML.

Los involucrados dentro del proceso tienen flujos de trabajo que están conectados por medio de las interfaces programadas. De esta forma el usuario puede ejecutar las actividades pendientes, estos realizan estas actividades por medio de interfaces programadas (*Standard Worklist Interfaces*).

Estas interacciones pueden ser internas y externas. Las interacciones externas son comunicaciones con los procesos de otra empresa, esta acción es realizada por los Web Services (WS), y estas son ejecutadas por coreografías o colaboraciones del modelo de negocio B2B, esta se encarga de proporcionar contenidos web entre sitios.

Los encargados del manejo de arquitecturas BPM usan una consola que les permite comunicarse y monitorear el estado de los procesos de máquina, por medio de un lenguaje de administración como interfaz. Esta máquina de ejecución está en constante monitoreo de los procesos, utilizando una base de datos; Además, no usa el lenguaje de administración si no que se conecta de forma directa para realizar ⁶queries (consultas) con los propósitos particulares.

Para las aplicaciones internas que envuelven interacciones complejas con los participantes externos (por ejemplo: un proceso de negocio B2B) estos procesos se modelan con la notación BPMN que capturan las comunicaciones requeridas del proceso local. Las aplicaciones internas se

Es un lenguaje estándar de una Base de datos, que significa consulta que permite administrar datos de las tablas de una base de datos. Consultado en <https://es.wikipedia.org/wiki/SQL>

encuentran en la red de la empresa, estas ingresan acceden usando tecnologías de integración tales como Java, J2EE, XML y Web Services.

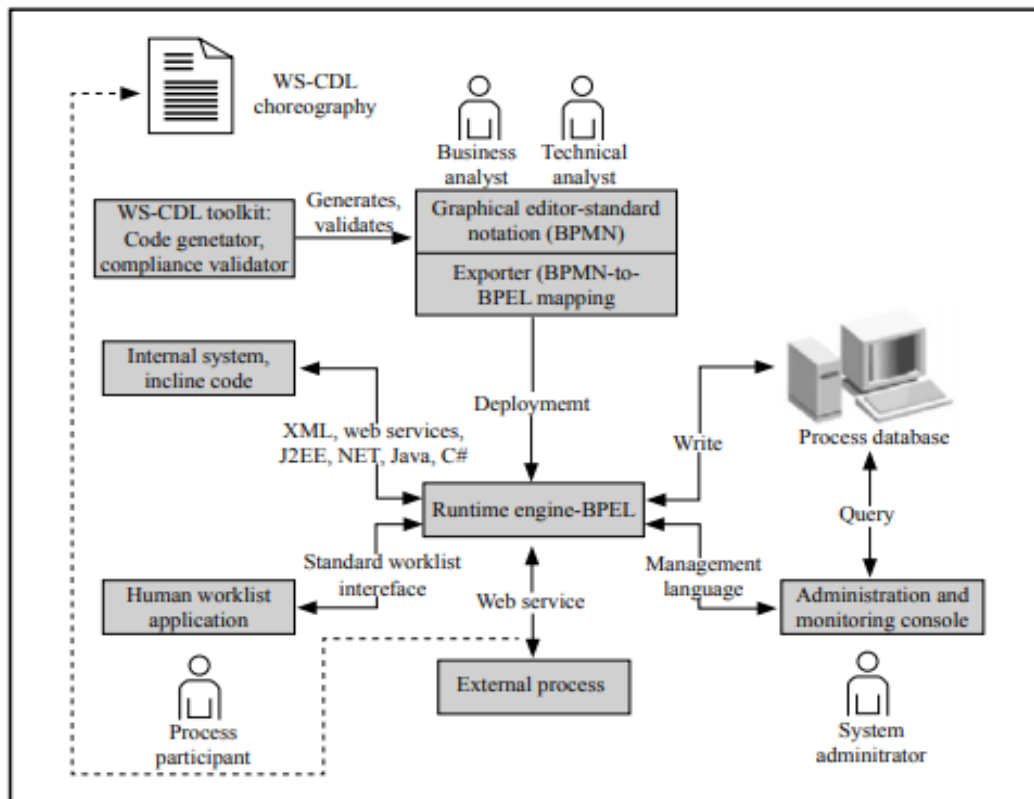


Figura 12. Arquitectura de Gestión de Proceso de Negocio (BPM).

Fuente: (Chong, Macías, Marchan, & Villacres, 2006).

2.4.2.4 Modelo de negocio

La arquitectura de negocio de la empresa permite tener una definición propia de la empresa en donde se tienen claros sus objetivos y metas (Magretta, 2002) define el modelo de negocio como “historia que explica cómo la empresa trabaja” (p.13).

Por su parte según Amit & Zott, (2001) lo define como un modelo de negocio que evidencia el contenido, la estructura y el gobierno de las transacciones designadas para crear valor al explotar oportunidades de negocio.

Esta arquitectura para su documentación debe de abarcar los siguientes puntos: los puntos de encuentro con los clientes, el punto de encuentro con los proveedores, colaboradores, los problemas y oportunidades de mejora, los procesos, datos y flujos de información, la organización, los sistemas informáticos y los indicadores de gestión y calidad.

2.4.2.5 Arquitectura de procesos

Esta arquitectura representa todos los procesos que se manejan en la organización, en donde los procesos están alineados para lograr los objetivos de la empresa de una manera exitosa.

En esta arquitectura se deben de establecer tres puntos:

- **El entorno de los procesos:** En un entorno de procesos se incluyen los flujos de un producto que son mostrados a los clientes. Cada entorno requiere recursos de personal y gestión de recursos.

- **La metodología de los procesos:** Para lograr que esta arquitectura de procesos tenga un preciso desempeño es necesario utilizar las metodologías adecuadas para los procesos como SCOR, Lean y Six Sigma. Estas metodologías se permiten mejorar los procesos.
- **El ciclo de vida de los procesos:** Un proceso debe ser establecido para que tenga éxito. Esto es porque los directores de los procesos analizan cada una de sus fases para conocer el rendimiento del proceso para tener claro las fases es importante documentar mediante las metodologías como CMMI, IDEAL, Lean y Six Sigma.

2.4.2.6 Modelado de procesos

El modelado de procesos es una técnica para la documentación de procesos de un sistema, sus entradas, sus salidas y sus formas de almacenamiento de datos (Alarcón, 2006).

La arquitectura de Procesos de Gestión de Negocios se puede dividir en dos notaciones gráficas, para su diseño, Business Process Modeling Notation (BPMN) y Business Process Execution Language (BPEL).

Business Process Modeling Notation (BPMN)

La primera versión de Business Process Modeling ((BPMN), fue desarrollada por el Institute Bussiness Process Management Initiative, (BPMI); en el 2004. El objetivo fue crear una notación gráfica que pudiera automatizar los procesos, a través del diseño gráfico.

Luego el siguiente año, tuvieron que trasladarla al Object Management Group (OMG), porque BPMI no establecía estándares tecnológicos. Esto ayudó a que BPM, se popularizara. (Hitpass, 2017).

Para su documentación es requerido utilizar BPMN Business Process Modeling cuya principal función es proveer de una notación gráfica que permita automatizar los procesos.

Elementos de BPMN

El modelado BPMN, está compuesto por notaciones y elementos gráficos. Estos elementos y notaciones se dividen en 5 categorías, las cuales son:

- **Objetos de flujo:** Sus elementos se basan en el comportamiento de los procesos. Existen tres tipos: Eventos, Actividades y Compuertas (Gateway).
- **Datos:** Están clasificados en cuatro elementos: Objetos de Datos, Datos de Entrada, Datos de Salida y Datos de Almacenamiento.
- **Objetos de conexión:** Estos objetos permiten establecer conexiones entre objetos de flujos. Están compuestos por los flujos de secuencia, flujos de asociación, líneas de mensajes.
- **Swimlanes:** Permiten organizar las actividades de flujos en diferentes categorías donde se le asigna roles y funciones. Sus elementos están compuestos por Pool y Lanes.
- **Artefactos:** Los artefactos proveen información adición de los procesos que se manejan. En sus elementos se encuentran Objetos de datos, anotaciones y grupos.

En la *Figura 13*, se puede apreciar la “Notación para el Modelado BPMN”, donde se muestran sus categorías con sus respectivos elementos para el modelado BPMN.

Categoría	Elemento	Descripción	Gráfica
Flujos	Evento	Es algo que sucede durante el curso del proceso de negocio. Afectan al flujo del proceso. Normalmente tienen una causa [disparador] o un impacto [resultado]. Dependiendo de cuando afectan al flujo serán eventos iniciales, intermedios o finales.	
	Actividad	Es un término genérico para el trabajo que realiza una compañía. Puede ser atómica (tarea) o compuesta [sub-proceso]. Para indicar la no atomicidad se coloca un signo + en la esquina del símbolo de actividad.	
	Gateway	Se utiliza para controlar la convergencia o divergencia de flujos. Representa una decisión para mezclar o unir caminos.	
Conexiones	Flujo de secuencia	Se utiliza para mostrar el orden o secuencia en que las actividades se realizan en un proceso.	
	Flujo de mensajes	Se utiliza para mostrar el flujo de mensajes entre dos participantes separados.	
	Asociación	Se utiliza para mostrar entradas y salidas de actividades.	
Swimlanes	Pool [fondo común]	Representa un participante en un proceso. Actúa como contenedor gráfico para particionar un conjunto de actividades.	
	Lane [sendero]	Es una sub-partición dentro de un pool y puede extenderse a todo lo largo o ancho del pool. Se utilizan para organizar y categorizar actividades.	
Artefactos	Objetos de datos	Mecanismo para mostrar como los datos son requeridos y producidos por las actividades. Se conectan a las actividades por asociaciones.	
	Grupos	Se utiliza para documentación o para propósitos de análisis, pero no afecta al Flujo de Secuencias.	
	Anotaciones	Mecanismo para que quien está modelando provea información adicional para el lector del diagrama.	
Datos	Objeto de datos	Objeto de datos: unidad de información. Puede asociarse a un estado.	
	Colección de objetos de datos	Objeto de datos que representa una colección de elementos.	
	Entrada de datos	Datos de entrada requeridos por un proceso cuando se inicia.	
	Recopilación de entrada de datos	Objeto de datos que representa una colección de elementos.	
	Salida de datos	Datos de salida requeridos por un proceso cuando se inicia.	
	Recopilación de salida de datos	Objeto de datos que representa una colección de elementos.	
	Almacén de datos	Almacén de datos: contenedor de objetos de datos persistentes.	

Figura 13. Notación para el Modelado BPMN.
Fuente: (Hitpass, 2017).

2.4.2.7 Diagramas en el modelo BPMN

Dentro del modelo BPMN podemos encontrar los diferentes tipos de diagramas (Manning, 2020).

- **Diagrama de coreografía:** Muestra las interacciones entre dos o más participantes. También puede ampliarse con subcoreografías.
- **Diagrama de colaboración:** Muestra las interacciones entre dos o más procesos, usando más de un grupo. En un diagrama de colaboración, se pueden utilizar las combinaciones de piscinas, procesos y coreografías.
- **Diagrama de conversación:** En general, esta es una versión simplificada de un diagrama de colaboración. Muestra un grupo de intercambios de mensajes relacionados en un proceso de negocios. Puede ampliarse con subconversaciones.

2.5.7.2. Business Process Execution Language (BPEL)

Business Process Execution Language en español llamado un Lenguaje de Ejecución de Proceso de Negocio, lo que hace es especificar los servicios de los procesos de negocio así como otorgar orquestación de servicio web, se caracteriza porque permite desarrollar procesos de negocio a partir de Servicios Web (SW) preexistentes y ofrecerlos como otro SW -BPEL.

Para la autora Bazán (Octubre 2007), los procesos de negocio especificados vía BPEL prescriben el intercambio de mensajes entre WS. Estos mensajes son mensajes WSDL. Algunos de estos mensajes pueden incluirse en lo que se denomina “contexto de negocio” de los procesos

de negocio, estos son un conjunto de mensajes WSDL llamado “contenedor” y representa los datos que son importantes para la correcta ejecución del proceso de negocio.

Para la documentación de BPEL, Hitpas (2017) establece dos partes la estructura y los componentes.

La estructura está compuesta por lo siguiente:

- **Variables:** Variables utilizadas durante el proceso. Pueden guardar el estado de un determinado mensaje.
- **PartnerLink (Socio):** Conjunto de servicios web involucrados en el proceso de negocio.
- **Operaciones:** Tipos de servicios web utilizados y operaciones invocadas.
- **Tipos de puertos:** Tipos de conexiones de servicios web requeridas.
- **Sequence (Secuencia):** Define la lógica de orquestación (pp1-5).

Los componentes están compuestos por lo siguiente:

- **Diseñador BPEL:** Aplicación con interfaz gráfica utilizada para definir el proceso de negocio.
- **Motor BPEL:** Aplicación que ejecuta una plantilla de proceso de negocio compatible con un estándar BPEL.
- **Plantilla de flujo de proceso:** Archivo generado por el diseñador BPEL para ser ejecutada por el motor BPEL. Contiene la especificación BPEL (pp1-5).

2.4.2.8 Monitoreo de los procesos

La principal función de esta arquitectura es poner en acción los procesos, personas y flujos de trabajo en un tiempo no muy extenso debido a que los proyectos BPM se implementan en periodos cortos de tiempo.

Durante la arquitectura de gestión los procesos no deben de ejecutarse una única vez, sino que deben de hacerlo en repetidas ocasiones, eso genera una buena gestión de procesos.

Los procesos se van deteriorando con el tiempo lo que hace que la empresa busque una mejora para los mismo o bien en caso de necesitar nuevas funciones estas se deben agregar.

Para tener un seguimiento y documentación de este, BAM permite a los directores de procesos tener claro cómo se gestionan los procesos, analizando sus condiciones y eventos durante la ejecución de los procesos.

El seguimiento y documentación de BAM se dan en los siguientes pasos:

1. Los procesos deben estar definidos y documentados con sus etapas y fases además de los indicadores de rendimiento. Teniendo esto en cuenta se puede establecer una tabla o Dashboard donde permita redefinir los procesos por prioridad y con un responsable del equipo asociado.
2. Una vez que los procesos están definidos, se debe configurar para introducirlo en el sistema BAM donde se analizará el funcionamiento de los procesos y se registraron los datos más relevantes que surgieron en el mismo.

3. Seguidamente se debe monitorear y observar el comportamiento del proceso en su entorno, en esta etapa se documentará cada indicador o mejora nueva que se haya realizado durante el monitoreo.
4. Business Activity Monitoring se basa en el aprendizaje de los procesos estos permite observar si el proceso se encuentra estable o no, o sea que el mismo cumple a totalidad con su función asignada.
5. Retroalimentación de los procesos BAM y reinicio de ciclo.

2.4.2.9 Indicador Clave de rendimiento (KPI)

Los indicadores de rendimiento y calidad de los procesos en inglés (Key Performance Indicators) permiten tener un mayor control de las actividades de la empresa. En el caso de BMP permite darles un rastreo a los procesos en donde el administrador del proceso pueda conocer el progreso de este y así conocer cuáles aspectos se pueden mejorar.

Para definir y documentar estos indicadores es necesario lo siguiente:

- Describir el objetivo específico que debe cumplir el proceso.
- Definir una fórmula o la lógica a utilizar para medir este indicador.
- Establecer la unidad de medición que tendrá el proceso.
- Establecer el periodo en que se va a medir su indicador.
- El indicador debe tener un responsable asociado para su respectiva medición.
- Finalmente indicar el proyecto al que se está analizando el proceso.

La información contenida en la Tabla 6 Ejemplo de documentación KPI, muestra un ejemplo de un indicador Clave de Rendimiento KPI.

Tabla 6. Ejemplo de documentación KPI

Definición	Objetivo	Fórmula	Periodicidad	Proyecto	Responsable
Proceso de ingreso de vehículos	50 vehículos	Cantidad de vehículos semanal	Mensual día 30 de cada mes	Sistema de seguridad	Técnico

Fuente: Elaboración propia

2.4.2.10 Modelo de Madurez - Business Process Maturity Model (BPMM)

El Modelo de Madurez en inglés Business Process Maturity Model, (BPMM) proporciona una guía para tener un control sobre los procesos y mejora de procesos de desarrollo de software. El principal objetivo de BPMM es tener una referencia de pasos a tomar para la mejora continua de los procesos en cinco niveles de madurez, permitiendo la mejora de los procesos en cada uno de ellos.

El Object Management Group define BPMM cómo un modelo conceptual basado en las “mejores prácticas” que están en uso en un dominio. Estos describen elementos esenciales de los procesos eficaces para los dominios seleccionados (Object Management Group, 2008).

Además, recalca el Object Management Group que estos elementos de proceso suministran una base para el control cuantitativo del proceso, que es la base para la mejora continua del Proceso (Object Management Group, 2008).

Estos dominios a lo que se refiere el Object Management Group son, marketing, operaciones de banca, manufactura, finanzas y las operaciones de TI.

El modelo de madurez permite describir cómo se encuentra la empresa desde la perspectiva de BPM (Object Management Group, 2008).

De acuerdo con la *Figura 14*, donde se muestra el “Modelo de Madurez“, se puede apreciar los 5 niveles de madurez que la componen. Todos los niveles de madurez excepto el inicial tienen áreas de procesos que indican donde la organización debe centrarse para mejorar sus procesos y lograr determinado nivel de madurez.

Cada área de proceso contiene un grupo de objetivos y de mejores prácticas para satisfacer el área específica y lograr los objetivos. Las mejores prácticas muestran lo que se debe de hacer, pero no como se debe de hacer. Por lo que las organizaciones son libres de definir sus métodos y enfoques para satisfacer las metas y objetivos de cada área de proceso.



Figura 14. Modelo de Madurez.

Fuente: (Object Management Group [OMG], 2016)

El (OMG,2008) describe los 5 niveles de la BPMM como se detallan a continuación:

Nivel 1.Inicial: Aquí los procesos se realizan en ocasiones de una forma inconsistente, con resultados difíciles de predecir.

Nivel 2.Gestionado: La gestión estabiliza el trabajo dentro de la unidades de trabajo, satisfaciendo los compomisos principales del grupo de trabajo.

Nivel 3. Estandarizado: Estos procesos se sintetizan a partir de las mejores prácticas identificadas en los grupos de trabajo. Las guías de adaptación se proporcionan para apoyar a las diferentes necesidades de negocio.

Nivel 4. Predecible: Las capacidades habilitadas por procesos estandar son explotados y proporcionados a las unidades de trabajo. El desempeño de los procesos se gestionan de forma estadística durante el flujo de trabajo, con el fin de predecir los resultados a partir de los estados intermedios.

Nivel 5. Innovación: Las acciones de mejoras productivas y oportunistas, buscan innovarse donde puedan eliminar las diferencias entre la capacidad actual de la organización y capacidad para lograr objetivos de negocio y competitividad.

2.4.3 Arquitectura de Servicios Web (WS)

A finales de los años 90, se comienza a estandarizar el uso de HTTP para la implementación de interoperabilidad entre componentes distribuidos. El gran desarrollo del uso de las tecnologías web para los servicios dio como resultado la creación de un estándar. A partir de ahí, la estandarización lleva a ⁷W3C definir una arquitectura de Servicios Web.

Las evidentes ventajas del uso de las tecnologías web para los servicios de software llevan a establecer un estándar.

⁷ W3C :World Wide Web Consortium, es un consorcio dedicado a la generación de estándares de calidad.
<http://www.w3.org>

La *Figura 15*, muestra la “Estructura de la Documentación de una Arquitectura de Servicios Web”. donde se observa el desglose que abarca una documentación de Arquitectura de Software basado en la Servicios Web (WS).

Tabla de Contenidos
Historial de cambios
Tabla de contenidos
Lista de figuras y tablas
1- ¿Qué es un Servicios Web?
2- Arquitectura de Servicios Web
3- Objetivos de la Arquitectura y Restricciones
4- Arquitectura de los Servicios Web
5- Estructura de la documentación de la Arquitectura Web Services
6- Estructura de la Interoperabilidad
7- Protocolos y estándares de los Servicios Web (WS)
8- Orquestación y Coreografía de Servicios Web
9- Descripción General de WS-BPEL
10- Descripción General de WS-CDL
11- Estructura de la Orquestación
12- Conclusiones
13- Referencias

Figura 15. Estructura de Documentación de Arquitectura de Servicios Web.

Fuente: (Pascual, 2019).

2.4.3.1 Servicios Web

Un Servicio Web, o en inglés Web Service, es un método de comunicación entre dos aparatos electrónicos en una red. Es una recopilación de protocolos abiertos y estándares usados para el intercambio de datos entre aplicaciones o sistemas (Lázaro, 2018).

Con relación a los Servicios Web, López, Echeverría, Fierro, & Jeder, (2007) manifiestan lo siguiente:

Es una pieza de software que conforma una serie de estándares de intercambio de información. (...) Estos estándares permiten el intercambio de operaciones entre diferentes tipos de computadoras, apartándose del problema del hardware que utilicen (...) de los sistemas (...), o de los lenguajes de programación en los que estén escritos.” (p.559).

Web Services es una tecnología de SOA (Arquitectura Orientada al Servicio), que está dentro de los estilos de Llamada y Retorno y dentro de la vista de Componentes y Conectores (C&C). Esta representación de estructura arquitectónica es la más reciente y es la más utilizada por las empresas.

Para una mayor comprensión en la *Figura 16*, “Esquema de jerarquía de los Web Services” se muestra detalladamente, el escalafón donde nace los Servicios Web.

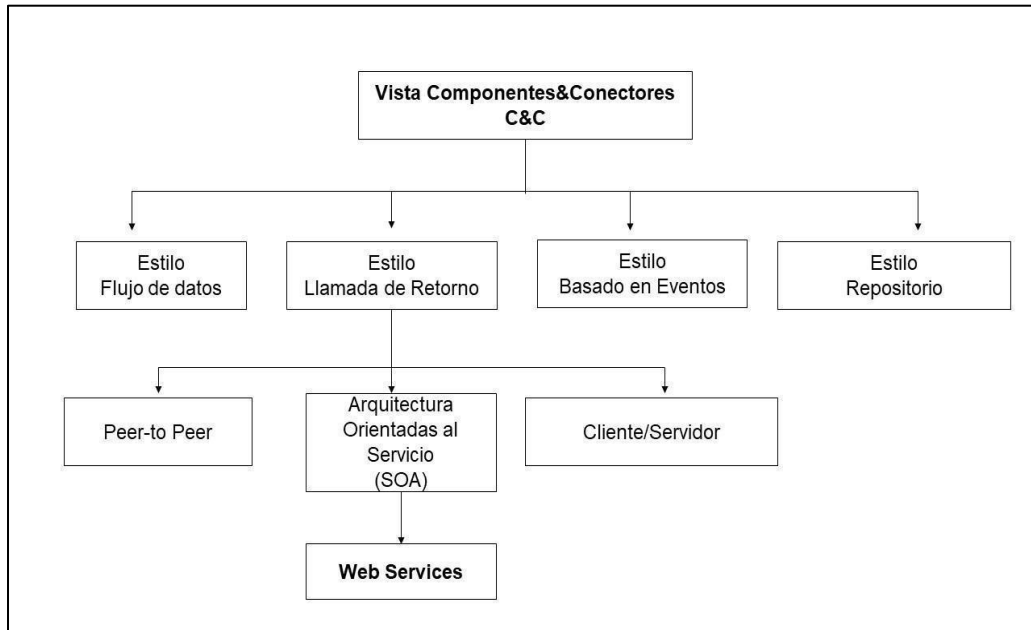


Figura 16. Esquema de Jerarquía de los Web Services.

Fuente: Elaboración propia basada en (Clements, Garlan, Little, Nord, & Stafford, 2003)

Web Services surgió en forma paralela a la idea de SOA, (Arquitectura Orientada a Servicios); lo cual ha ordenado su uso sistemático, combinando protocolos, perfiles, especificaciones y estándares.

Una Arquitectura Orientada al Servicio es una estrategia tecnológica por la cual las aplicaciones hacen uso de los servicios disponibles en una red. Por lo que los Web Services encajan perfectamente como una arquitectura en sí. (Veintimilla Reyes, Espinoza Mejía, & Cisneros, 2014).

La arquitectura basada en Web Services cuenta con varias funcionalidades y un conjunto bien definido de opciones. Además, cuenta con sistemas de software diseñados para soportar la interacción interoperable de máquina a máquina a través de una red, esta interoperabilidad se obtiene a través de una serie de estándares abiertos basados en XML, como WSDL, SOAP y UDDI. Estos estándares proporcionan un enfoque común para definir, publicar y usar servicios web.

Otras de las funciones de los Web Services es que ellas permiten a las organizaciones intercambiar datos sin necesidad de conocer los detalles de sus respectivos Sistemas de Información, así como la automatización de los procesos.

2.4.3.2 Estructura de los Servicios Web (WS)

De acuerdo con García Sánchez (2008), señala que la arquitectura básica del modelo Web Services describe a un consumidor, un proveedor y ocasionalmente un corredor (broker). Sus operaciones son las de publicar, encontrar y enlazar.

En la *Figura 17*, podemos observar el “Modelo Básico de los WS”, donde se muestra un ejemplo de un proveedor público, anunciando sus servicios en un corredor. Luego un consumidor se conecta al corredor encontrando los servicios deseados y una vez que lo hace se realiza un lazo entre el consumidor y el proveedor. Cada sujeto realiza ya sea un rol o todos los roles.

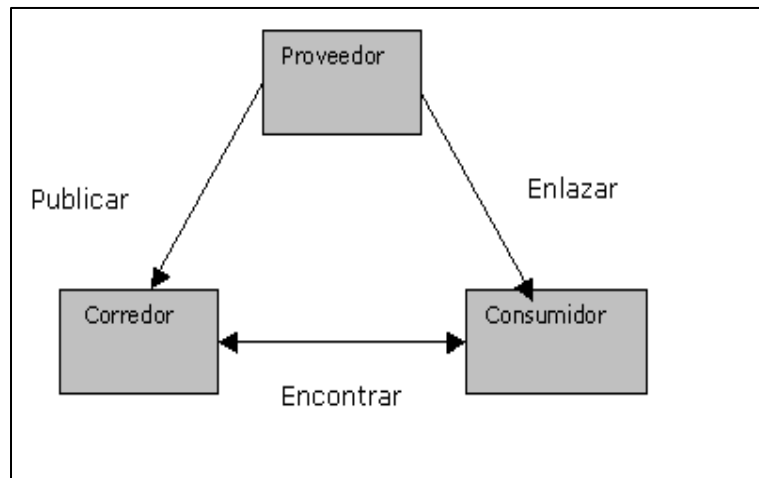


Figura 17. Modelo Básico de los WS.

Fuente: (Consortio World Wide Web (W3C), 2004).

2.4.3.3 Roles de la Arquitectura de los Servicios Web

La arquitectura de Servicios Web está estructurada por roles, objetos y operaciones, en la *Figura 19*, se muestra todos estos elementos. Para Kreger, (2001) los roles principales de la Arquitectura de Software son los que se describen a continuación.

- **Proveedor de Servicio (Service Provider).** Este crea un servicio web y luego publica el servicio con un registro de servicios basado en un estándar llamado Descripción de Descubrimiento e Integración Universal, UDDI.
- **Registro de Servicios (Service registry).** Este es un registro de búsqueda de Descripciones de Servicio. El Registro de Servicios es una función opcional en la arquitectura, porque un proveedor de servicios puede enviar la descripción directamente a los solicitantes de

servicios. Y estos pueden obtener una descripción del servicio de otras fuentes como un archivo local, un sitio FTP o un sitio web.

- **Solicitante de servicio (Service Requestor).** Esta es la aplicación que busca, invoca o inicia una interacción con un servicio. La función de solicitante de servicio puede ser desempeñada por un navegador dirigido por una persona o un programa sin una interfaz de usuario (pp. 6-7).

2.4.3.4 Operaciones en una Arquitectura de Servicios Web

Para que una aplicación aproveche los servicios web, deben producirse tres comportamientos:

El comportamiento de publicación, el de búsqueda y el de innovación están basadas en las descripciones de los servicios. Este comportamiento puede ocurrir de forma individual o repetida. Según (Kreger, Heather, 2001) estas operaciones son:

Publicar (Publish). Para que sea accesible, se debe publicar una descripción del servicio para que el solicitante del servicio pueda encontrarlo. El lugar donde se publica puede variar según los requisitos de la aplicación.

Encontrar (Find). En la operación de búsqueda, el solicitante del servicio (Service Requestor) recupera una descripción del servicio directamente. Esta operación busca o realiza una consulta para el tipo de Servicio requerido, (Service Discovery).

La operación de búsqueda puede estar involucrada en dos fases para el solicitante del servicio, (Service Requestor) en el momento del diseño para recuperar la descripción de la interfaz

del servicio para el desarrollo del programa, y en el tiempo de ejecución para recuperar el enlace del servicio y la descripción de la ubicación para la invocación.

Invocación (Bind). Un servicio necesita ser invocado. En esta operación el servicio solicitante (Service Requestor) invoca una interacción en el servicio en el tiempo de ejecución utilizando los detalles de la vinculación del servicio como la descripción para localizarlo, contactarlo e invocarlo.

La información contenida en la *Figura 18*, muestra los “Actores, Objetos y las Operaciones de la Arquitectura de los SW” descritos anteriormente.

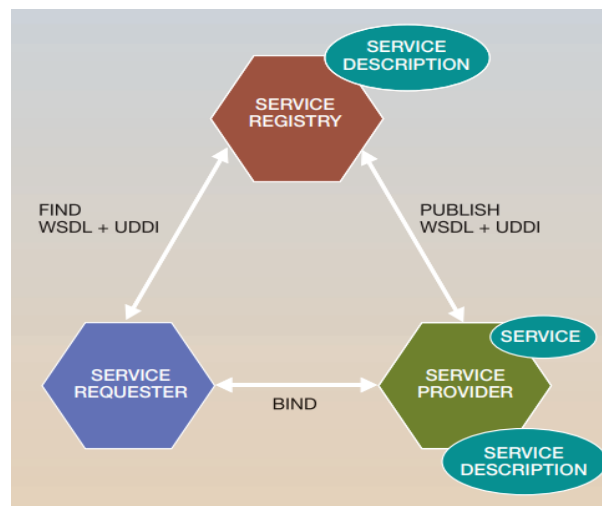


Figura 18. Actores, Objetos y las Operaciones de la Arquitectura de los SW.

Fuente: (Kreger, 2003).

2.4.3.5 Protocolos y estándares de Servicios Web (WS)

Una arquitectura Services Web cuenta con diferentes componentes para su ejecución. Web Services, al estar dentro de los estilos de Componentes y Conectores, (C&C), cuenta con Conectores y Componentes (Clements, Garlan, Little, Nord, & Stafford, 2003).

Como componentes de software que dan prestaciones de servicios entre sí. Sus estándares de los Servicios Web son: WSDL, XML, UDDI, SOAP. Son publicados y localizados vía UDDI (Universal Description Discovery and Integration), se describen usando WSDL (Web Service Description Language) e invocados vía SOAP (Simple Object Access Protocol) y el XML (Extensible Markup Languages) como Lenguaje de descripción de datos obteniendo interoperabilidad, ya que es independiente de tecnologías, plataformas y la representación física de los datos (Bazán, Octubre 2007).

Los Web Services son:

- Publicados y localizados vía UDDI (Universal Description, Discovery and Integration).
- Descriptos usando WSDL (Web Service Description Language).
- Invocados vía SOAP (Simple Object Access Protocol) sobre HTTP.

En la *Figura 19*, “Estructura del Funcionamiento y los Componentes de los SW”, observamos donde el usuario por medio de una petición Web Services, realiza una publicación en la UDDI, el cliente o implementación recibe un mensaje final por medio del componente SOAP (XML).

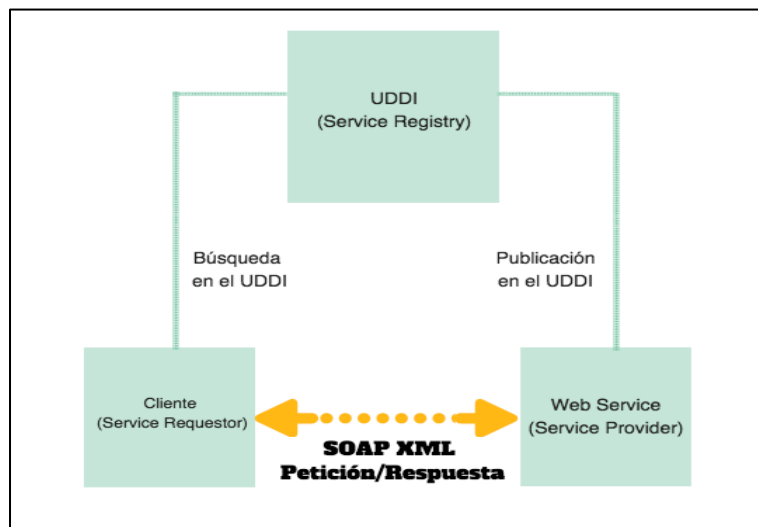


Figura 19. Estructura del Funcionamiento y los Componentes de los SW.

Fuente: (Lázaro, 2018).

a- XML (EXtensible Markup Language)

De acuerdo con Lázaro (2018), XML (EXtensible Markup Language), es un lenguaje de marcado para describir contenido de forma separada de su formato. Dentro de este estándar

hay otros estándares para la ⁸definición y transformación de lenguaje y otro para la ⁹presentación de lenguaje.

La función básica de los documentos de definición y transformación de lenguaje es describir estructura de datos, para usar una estructura común y los documentos de transformación son de tipo XML en el que se especifica cómo se preparan los datos para presentarlos, a través de la pantalla u otros medios (Lázaro, 2018).

b- WSDL (Web Services Description Language)

Es un Lenguaje de Descripción de Servicios Web, basado en XML, su función es describir servicios web y el camino para acceder a ellos. (Wei-Tek , Yamin Wang, & Dong, 2002).

En la *Figura 20*, “Ejemplo de Acceso de un WSDL”, muestra el acceso de un Web Services a través del estándar WSDL, se establece una comunicación entre el proveedor del servicio y el usuario donde el proveedor del servicio indica que funciones se pueden invocar, mediante UDDI se proporciona la agencia de descubrimiento de servicios, seguidamente el servidor de aplicaciones

⁸ DTD (Document Type Definition) o XSD (XML Schema Definition): Estándares para la definición y transformación de lenguajes.

⁹ XSL-FO (eXtensible Stylesheet Language Formatting Objects.) y XSLT (Extensible Stylesheet Language Transformations): Estándares para la presentación de lenguajes.

web, entrega el proveedor de servicios y finalmente la aplicación desempeña el papel de proveedor de servicios, permitiendo la recepción y el envío de mensajes.

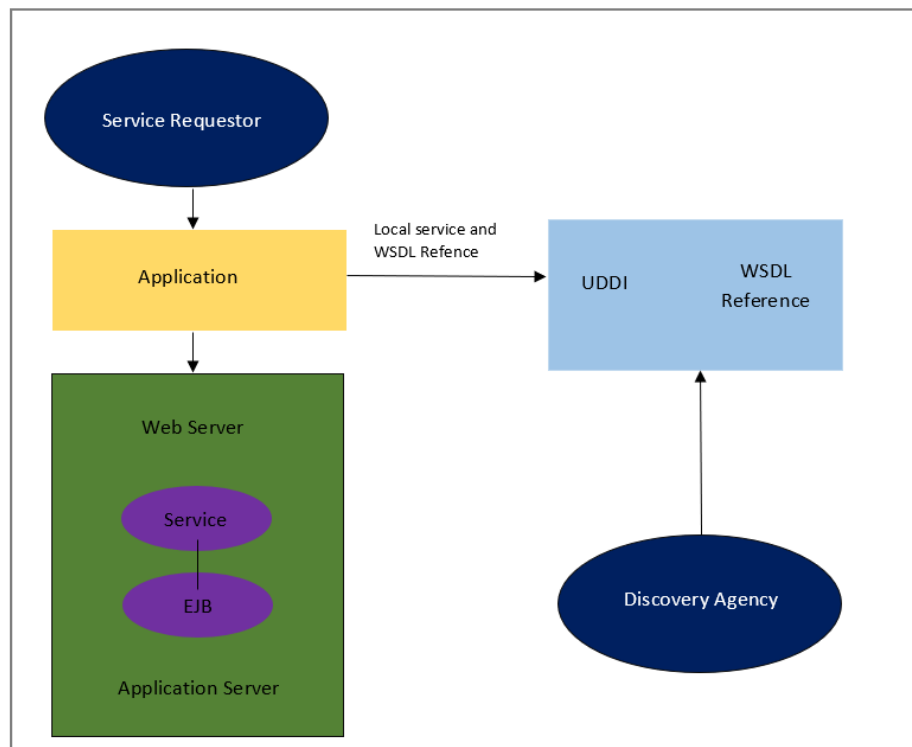


Figura 20. Ejemplo de Acceso de un WSDL.

Fuente: (Barry & Dick, 2013).

c- SOAP (Simple Object Access Protocol)

Su función principal es recibir y enviar mensajes entre aplicaciones sin enfrentamientos de interoperabilidad. Es un protocolo de comunicación que define cómo dos objetos en diferentes

procesos pueden conectarse por medio de intercambio de datos XML (Extensible Markup Languages) (Sandeep & Webber, 2004).

Este protocolo de servicio web intercambia los datos estructurados utilizando XML y generalmente HTTP y SMTP en entornos descentralizados y distribuidos. Estos se deben intercambiar entre quien realiza la solicitud (cliente) y quien la responde (servicio).

Este Protocolo Simple de Proceso a Objetos, utiliza documentos WSDL (Lenguaje de Descripción de Servicios Web) para distribuir un modelo de descripción de servicio web. Esto describe cómo deben de presentarse las solicitudes SOAP (del lado del cliente) y las respuestas (del lado del servidor). Además, los servicios web SOAP tienen estándares de seguridad y direccionamiento.

d- UDDI (Universal Description Discovery)

Este estándar permite configurar un servicio de registro o directorio donde los proveedores de servicios publican sus servicios, como un servicio de páginas amarillas de servicios web.

UDDI es un lenguaje de descripción de los Servicios Web, se comunica a través de SOAP. En UDDI, las especificaciones técnicas como las descripciones de WSDL pueden registrarse y luego utilizarse para calificar la descripción del registro de un servicio compatible (Sandeep & Webber, 2004).

Los datos de la implementación en WSDL son particularmente adecuada para este propósito, ya que puede proporcionar especificaciones genéricas implementadas por grandes clases de servicios (Sandeep & Webber, 2004).

El propósito de un registro UDDI es la representación de datos y metadatos acerca de Servicios Web. Un registro UDDI es un estándar para clasificar, manejar y catalogar servicios web, con el propósito de ser descubiertos y consumidos por otras aplicaciones (Sandeep & Webber, 2004).

2.4.3.6 Estructura de la Interoperabilidad de Servicios Web

El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), define los estándares de la interoperabilidad como la habilidad de dos o más sistemas o componentes para intercambiar y utilizar la información intercambiada. (IEEE, 2019).

El primer ejemplo para desarrollar es el propuesto por, Newcomer (2002), que muestra una arquitectura que está caracterizada por un principio significativo que tiene los Web Services y es la interoperabilidad. Permite que diferentes servicios web distribuidos se ejecuten en una variedad de plataformas de software y arquitecturas de hardware.

La interoperabilidad es el principal propósito de utilizar los Web Services. La interoperabilidad de los Servicios Web se basa en el XML. Un protocolo XML simple y estándar, desplegado de forma generalizada, es capaz de ser llevado a través de los principales protocolos de comunicación web, asegurando la interoperabilidad al nivel más básico (Ballinger, 2003).

Representación unificada de las aplicaciones dentro de los sistemas heterogéneos. Un lenguaje de definición de interfaz proporciona una representación de aplicaciones común, independiente de la plataforma.

En la *Figura 21*, se muestra la “Estructura de la Interoperabilidad de los Servicios Web”, donde se observa que la interoperabilidad puede desarrollar componentes de software con funciones propias e independientes. Estas funcionalidades pueden ser compartidas con otras aplicaciones y servicios.

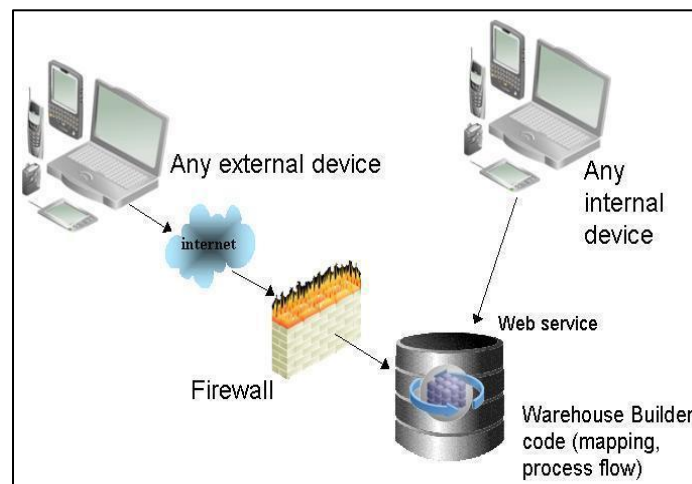


Figura 21. Estructura de la Interoperabilidad de los Servicios Web.

Fuente: (W3C, 2004).

2.4.3.7 Orquestación y Coreografía de Servicios Web

Orquestación

Un proceso se puede considerar como una orquestación de servicios cuando este es controlado por una única entidad. El proceso interacciona con los servicios de componentes y la lógica requerida para llevar correctamente las interacciones (Arroyo , Ch Peley, & Fino, 2009).

Dentro de la orquestación de Servicios Web, BPEL, en inglés, (Business Process Execution Language), en español, Lenguaje de Proceso de Negocio, constituye un lenguaje estándar para la integración y automatización de los procesos (Erl , y otros, 2008).

Los principios de Erl , y otros, (2008) señalan que los Servicios se pueden ejecutar en diferentes plataformas que cumplan dichos estándares. Esto permite la composición; donde debe ser construido de forma que pueda ser utilizado para construir servicios de alto nivel.

En los Servicios Web, el protocolo para la orquestación es ¹⁰(WS-BPEL) y para la coreografía es ¹¹(WS-CDL).

¹⁰ WS-BPEL: Lenguaje de Ejecución de Procesos de Negocio de Servicios Web. (Business Process Execution Language). <https://es.wikipedia.org/wiki/WS-BPEL>.

¹¹ WS-CDL: Lenguaje para la descripción de Coreografías de Servicios Web (Web Services Choreography Description Language). Fuente: <https://es.wikipedia.org/wiki/WS-CDL>

Abonado a lo anterior la orquestación de Servicios Web pretende proporcionar una aproximación abierta, basada en estándares, para conectar Servicios Web de forma conjunta, con el objetivo de crear procesos de negocio de alto nivel.

De acuerdo con Weerawarana y otros, el Lenguaje de Proceso de Negocio, (BPL), se define como un modelado, que ejecuta procesos empresariales como composiciones de servicios. La arquitectura de plataforma de servicios web le brinda una visión interna de la plataforma que cambiará la forma en que entrega las aplicaciones (Weerawarana , Curbera , Leym, Storey , & Ferguson, 2005).

BPEL, es un lenguaje de programación estructurado en diferentes tipos de bloques (recursivos,etc.), pero restringe definiciones y declaraciones al nivel superior (Erl , y otros, 2008).

Para que los servicios web logren alcanzar todos los requerimientos de información planteados es necesario que combinen varios servicios web, con el fin de generar un servicio que al final cumplan con todas las expectativas del usuario.

En la *Figura 22*, “Estructura de Orquestación en los Servicios Web”, esta información hace referencia a que la estructura de BPEL es un lenguaje de ejecución de procesos de negocio.

Un proceso de negocio (Business Process) es un flujo de trabajo, conformado por una serie de pasos o actividades. Las actividades del proceso de negocio pueden requerir de aplicaciones o la intervención humana para lograrlo. Un proceso de negocio tiene una duración larga.

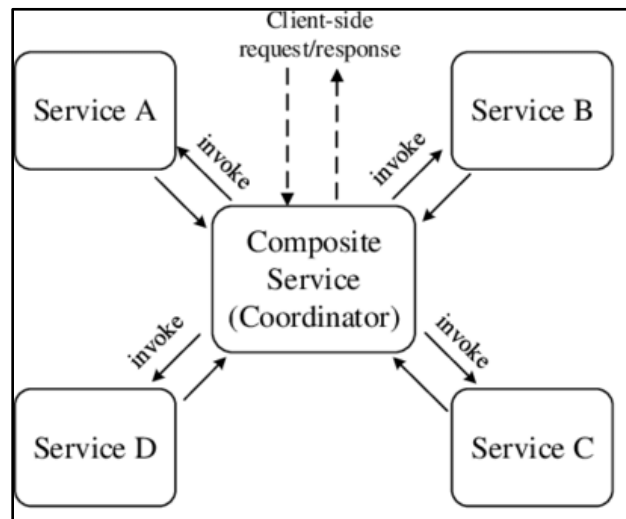


Figura 22. Estructura de Orquestación en los Servicios Web.

Fuente: (Li, Gui, Hofer, & Li, Web Orchestration, 2009).

Coreografía

La coreografía puede entenderse como un proceso público y no ejecutable; es público porque define el comportamiento común; y es no ejecutable porque no está pensado para ser llevado a cabo, sino para actuar como una forma de negocio que imponen reglas de interacción que deben cumplirse por las entidades participantes (Peltz, 2003).

Hay autores que se refieren a la Coreografía como las encargadas de combinar varios servicios e interacciones para satisfacer las necesidades del usuario es el caso de (Giraldo, Guzmán, & Ovalle, 2008) que indican que:

“Para que estas interacciones se realicen se hace por medio de la coreografía, ya que esta se encarga de representar y describir la secuencia y condiciones en la que los datos son

intercambiados por los Servicios Web. La coreografía de servicios web describe las interacciones pertinentes durante la integración entre los servicios, con el fin de lograr los objetivos propuestos (p.144).

Dumas y Oaks también señalan que un modelo de coreografía describe una colaboración entre una colección de servicios para lograr un objetivo común. Se capturan las interacciones y dependencias en las que se involucran los servicios participantes para lograr este objetivo (Barros, Dumas , & Oaks, 2005).

De acuerdo con la *Figura 23*, “Estructura de Coreografía en los Servicios Web” se muestra un modelo no centralizado punto a punto y todo el flujo de trabajo se define intercambiando los mensajes.

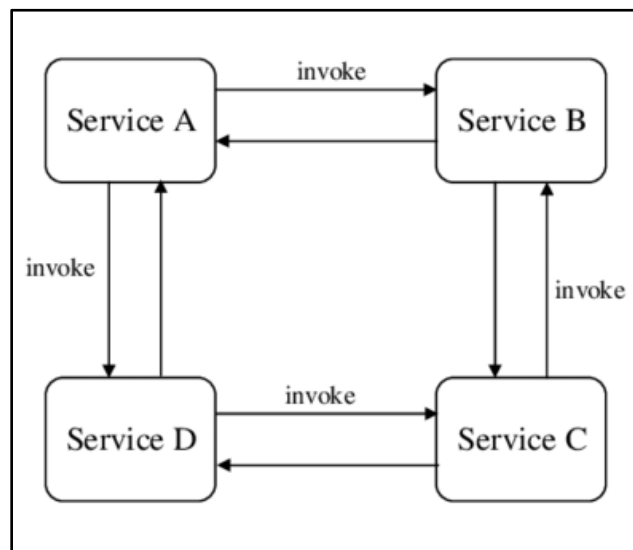


Figura 23. Estructura de Coreografía en los Servicios Web.

Fuente: (Li, Gui, Hofer, & Li, Web Service Choreography, 2019)

Orquestación versus Coreografía

Para entender profundamente los conceptos de la Orquestación y la Coreografía en la Figura 24, “Coreografía versus Orquestación”, se observa la diferencia entre una Coreografía y una Orquestación en la arquitectura de los Servicios Web. La Orquestación se refiere a un proceso ejecutable, mientras que la Coreografía lo que hace es rastrear las secuencias de mensajes entre las partes.

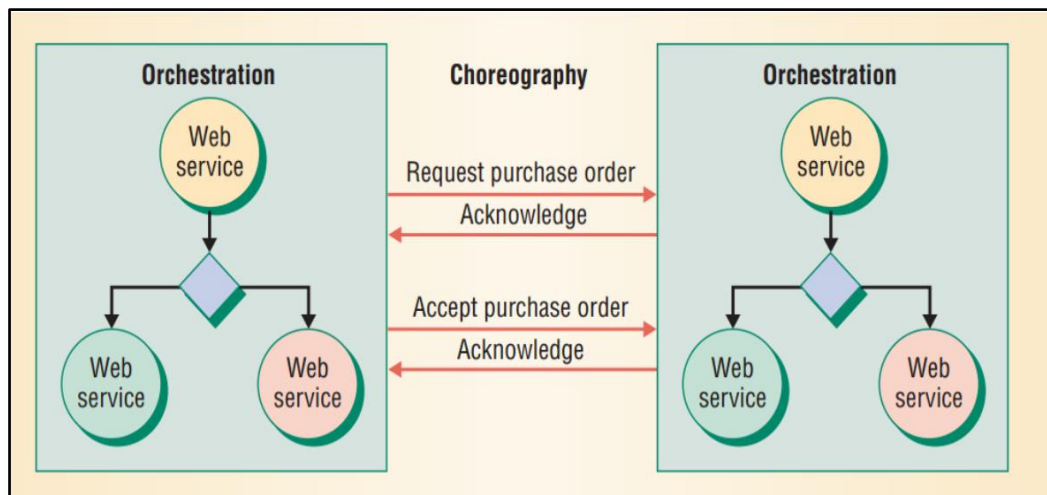


Figura 24. Coreografía Versus Orquestación.

Fuente: Basado en (Peltz, Orchestration versus choreography, 2003)

El significado de estos dos términos tiende a parecerse mucho, para poder diferenciarlos tomaremos las definiciones de Marquez Solis, (2007). En la Tabla 7, “Términos de Orquestación y Coreografía ” podemos apreciar estas definiciones entre orquestación y coreografía. Además, los lenguajes utilizados en cada servicio.

Tabla 7. Términos de Orquestación y Coreografía

Orquestación	Coreografía
Compone servicio para generar un nuevo proceso de negocio	Compone un servicio para permitir la colaboración entre negocios de forma en que múltiples participantes colaboren entre pares (peer to peer) como parte de una operación o transacción de negocio duradera.
La orquestación debe verse como algo atómico en sí mismo porque se ve el proceso de negocio como tal.	Muchos participantes pueden colaborar porque existen muchos flujos de control diferentes.
Se puede usar para preparar la información que se intercambia en la ejecución de una coreografía	Se describen interacciones con estado y duraderas.
En un proceso jerárquico el proceso web organiza la manera de invocar a otros mediante flujos de control que determina qué y cómo invocarlo.	Es posible realizar una coreografía mediante un conjunto de orquestaciones de cada uno de los componentes que la integran.

Fuente: Adaptado de (Marquez Solis, 2007).

2.4.3.8 Descripción general del WS-BPEL

El Lenguaje de Ejecución de Procesos de Negocio de Servicios Web, WS-BPEL, en inglés, (Business Process Execution Language). Se rige por un vocabulario de XML para especificar procesos de negocio. Este permite composición recursiva. El WS-BPEL es similar al lenguaje de programación, en la elaboración de bucles, condicionales, variables, gestión fallos y este interpretado por un motor de ejecución de BPEL.

En la *Figura 25*, “Ejemplo de Orquestación de los Servicios Web (WS-BPEL)” se muestra un ejemplo del proceso de Orquestación de los WS-BPEL para manejar una orden de compra. Las flechas discontinuas representan la secuencia. La agrupación libre de secuencias representa secuencias concurrentes.

Las flechas sólidas representan los enlaces de control que se hacen al recibir la orden de compra de un cliente, el proceso inicia con tres tareas simultáneamente: calcular el precio final de la orden, seleccionar un remitente y programar la producción y el envío de la orden. Los tres procesos pueden procesarse al mismo tiempo, existen dependencias de control y datos entre las tres tareas.

Luego se requiere el precio de envío para finalizar el cálculo del precio, y la fecha de envío es necesaria para el programa. Cuando se completen las tres tareas, el procesamiento de la factura puede continuar y la factura se envía al cliente, utilizando la sincronización entre actividades simultáneas. Cabe mencionar que esto no es una notación gráfica de los WS-BPEL.

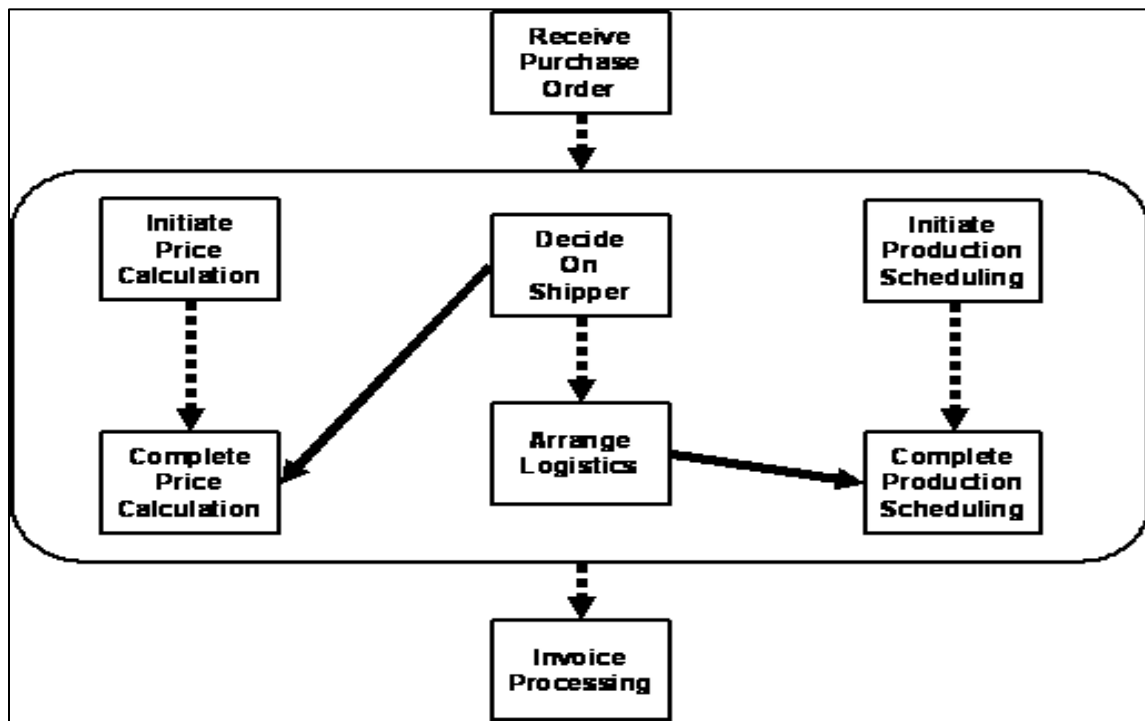


Figura 25. Ejemplo de Orquestación de los Servicios Web (WS-BPEL)

Fuente: (Organization for the Advancement of Structured Information Standards, 2007)

Descripción general del WS- CDL

El Lenguaje para la descripción de Coreografías de Servicios Web, WS-CDL, en inglés (Web Services Choreography Description Language); constituye un acuerdo entre un conjunto de partes sobre cómo debe ocurrir una colaboración determinada.

A diferencia de una orquestación, no hay un director en la coreografía, es un conjunto de relaciones entre pares.

El WS-CDL, está constituida por tres tipos de actividades, que se detallan a continuación (Barros, Dumas , & Oaks, 2005).

- **Primer tipo:** Actividad de Secuencia- Actividad Paralela y actividad de elección.
- **Segundo tipo:** Actividad WorkUnit.
- **Tercer tipo:** Interacción NoAction y SilentAction, Asignar y Perform.

A-Primer Tipo

- **Actividad de Secuencia:** Describe una o más actividades que se ejecutan en orden secuencial.
- **Actividad paralela:** Describe una o más actividades que se pueden ejecutar en cualquier orden o al mismo tiempo.
- **Actividad de elección:** Describe la ejecución de una actividad elegida entre un conjunto de actividades alternativas o competitivas.

B-Segundo tipo

- **Actividad WorkUnit:** Describe la ejecución condicional.

C-Tercer Tipo

- **Actividad NoAction y SilentAction:** Describen puntos en una coreografía donde un rol (nombrado) no realiza ninguna acción o realiza una acción detrás de escena que no afecta al resto de la coreografía.
- **Actividad Asignar:** Se utiliza para transferir el valor de una variable a otra variable dentro de un rol.
- **Actividad Interpretar:** Se utiliza para llamar a otra coreografía que se realizará dentro del contexto de la coreografía en ejecución. La coreografía llamada puede definirse dentro del mismo paquete que la persona que llama, o puede ser de un paquete completamente separado que se ha importado.
- **Actividad de Perform:** Tiene un mecanismo de vinculación que permite vincular las variables de los llamantes a las variables libres en la coreografía llamada.

En la *Figura 26*, “Actividades del WS-CDL” Se puede observar el paquete de las actividades que constituye el WS-CDL y que fueron detalladas anteriormente.

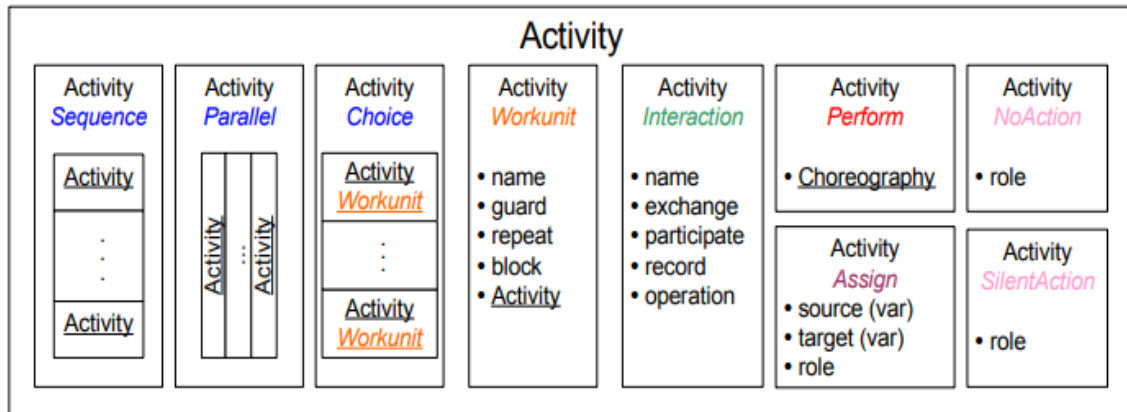


Figura 26. Actividades del WS-CDL.

Fuente: (Barros, Dumas , & Oaks , WS-CDL Activities, 2005)

Capítulo III. Metodología

Esta sección está compuesta por cuatro secciones: el tipo de investigación, la población meta y muestra, la descripción de los instrumentos utilizados y por último el procedimiento que se utilizó para analizar información obtenida.

3.1 Tipo de Investigación

La investigación seguida en este proyecto corresponde a un trabajo de campo hecho para desarrollar una propuesta de documentación de Arquitectura de Software Integral para la Sección de Seguridad Institucional de la UNA.

El estudio de campo suele ser utilizado con gran éxito en múltiples ocasiones en el contexto de los sistemas de información (Choi , Lee , & Yoo, 2010; El Eman & Madhavji, 1995, March; Huff & Munro, 1985; Sanders & Courtney, 1985).

3.2 Población y Muestra

3.2.1 Población

Una población es el conjunto de todos los casos que concuerdan con una serie de especificaciones (Lepkowski, Tucker, & Brick, 2008).

Pastor R, señala que una población es un conjunto de elementos que contienen ciertas características que se pretenden estudiar (2019).

La población utilizada en la investigación es finita porque se enfoca específicamente en los departamentos o escuelas que realizan solicitud de reserva o permiso de ingreso a parqueos Institucionales y otras áreas de interés, otorgados por la Sección de Seguridad Institucional.

Para calcular la muestra se tomó a la cantidad de personas y escuelas de la Universidad Nacional de Costa Rica que realizan la solicitud de reserva o permiso de ingreso a parqueos Institucionales y otras áreas de interés, con mayor frecuencia.

3.2.2 Muestra

Una muestra es un conjunto de elementos que contienen ciertas características que se pretenden estudiar. (Del Pino, 2008).

Esta medición se realizó mediante el método ¹²encuesta, de una muestra de la población. Esta encuesta se dividió en tres secciones: demografía, percepción y flujo de trabajo de la ¹³SSI – UNA. El objetivo fue medir que tan frecuente realizan estas solicitudes de ¹⁴reservas, el tiempo que toma esta solicitud desde el inicio hasta su aprobación, si el mecanismo utilizado es eficiente y si se ajusta a las necesidades.

¹² El formulario encuesta se observa en el apartado Anexos.

¹³ SSI-UNA: Sección de Seguridad Institucional de la Universidad Nacional.

¹⁴ Solicitud de reserva o permiso de ingreso a parqueos Institucionales y otras áreas de interés.

3.2.3 Sección I: Demografía

En la Tabla 8: “*Información de la muestra*”, observamos datos demográficos de la muestra de la población de estudio, realizada por el método encuesta. Se observa que la mayoría de los entrevistados /as son auxiliares de servicios secretariales, (55,6%, 5 de 9) habiéndose también entrevistado a profesionales ejecutivos y oficiales de seguridad (22%, 2 de 9 en cada caso).

En cuanto a la experiencia de trabajo en la Universidad Nacional, la mayoría de los entrevistados tiene entre 5 y 10 años (44,5%, 4 de 9). Siendo seguido por las personas con más de 10 años de experiencia (33%, 3 de 9). Adicionalmente, dos de los entrevistados (22% reportaron tener más de 1 año y menos de 5 años de experiencia.

Tabla 8. Información de la Muestra Relacionada con la Encuesta

Experiencia	Departamento	Rol desempeñado
Menos de 10 años	Escuela de arte escénico	Auxiliar de servicios secretariales
Menos de 5 años	Promoción estudiantil, área deportiva	Auxiliar de servicios secretariales
Menos de 10 años	Escuela de música	Auxiliar de servicios secretariales
Menos de 10 años	Instituto IRET	Auxiliar de servicios secretariales
Menos de 10 años	Instituto IRET	Auxiliar de servicios secretariales
Más de 10 años	CIDE	Profesional ejecutivo
Más de 10 años	Danza	Profesional ejecutivo
Más de 10 años	Centro de monitoreo	Oficial de seguridad
Menos de 5 años	Centro de monitoreo	Oficial de seguridad

Fuente: Elaboración propia.

Seguidamente los datos estadísticos contenidos en la Figura 27,” Departamentos encuestados”, observamos que la totalidad de los entrevistados pertenecen al IRET (Instituto Regional de Estudios en Sustancias Tóxicas) y al centro de Monitoreo (22%, 2 de 9, cada uno de ellos), además se entrevistaron a los funcionarios de la Escuela de Música, Arte Escénico, Escuela de Danza. CIDE (Centro de Investigación en Docencia y Educación) y el departamento de Promoción Estudiantil (11%, 1 de 9 respectivamente).

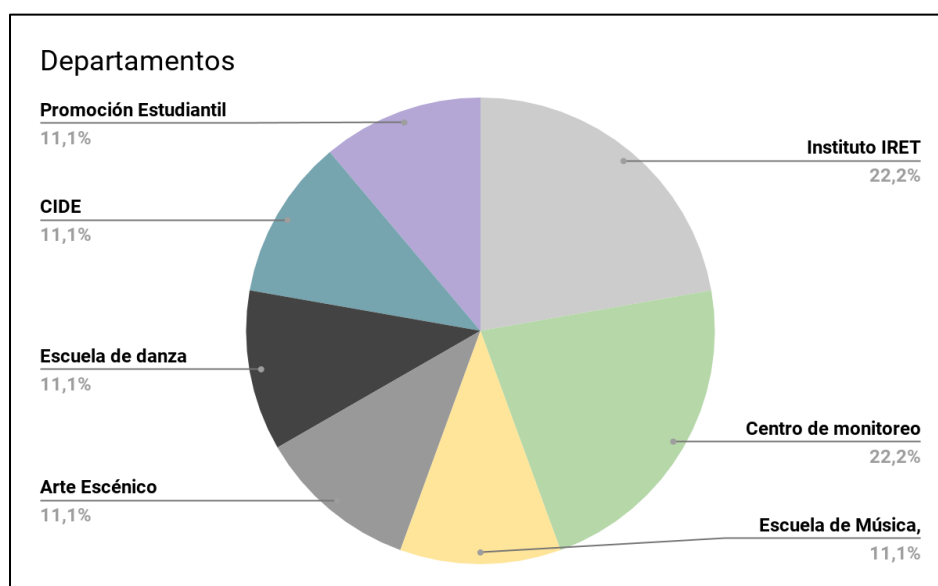


Figura 27. Departamentos Encuestados

Fuente: Elaboración propia

3.2.4 Sección II: Percepción

La segunda sección de la encuesta trata de los elementos de percepción. Todos los funcionarios entrevistados tenían conocimientos acerca del modelo actual o el mecanismo utilizado por la SSI-UNA, para la realización de solicitudes o permisos

El siguiente elemento de percepción analizado fue la opinión de los encuestados sobre si el mecanismo utilizado para la realización de solicitud de reserva o permisos de ingresos a parques Institucionales y otras áreas de interés, se ajustaba a las necesidades de los funcionarios/as.

En la Figura 28, “¿El mecanismo utilizado por la SSI-UNA se ajusta a las necesidades de los funcionarios?” observamos que el 67% (6 de 9) manifestó que el sistema actual o mecanismo utilizado para la realización de dicha solicitud no se ajusta a las necesidades de los funcionarios y un 33% (3 de 9), manifiestan que este mecanismo si se ajusta a las necesidades. Se podría inferir que esta respuesta tiene validez, en el tanto que la anterior pregunta el 100% de los entrevistados reportó conocer el mecanismo para realizar reservas y permisos de ingresos.

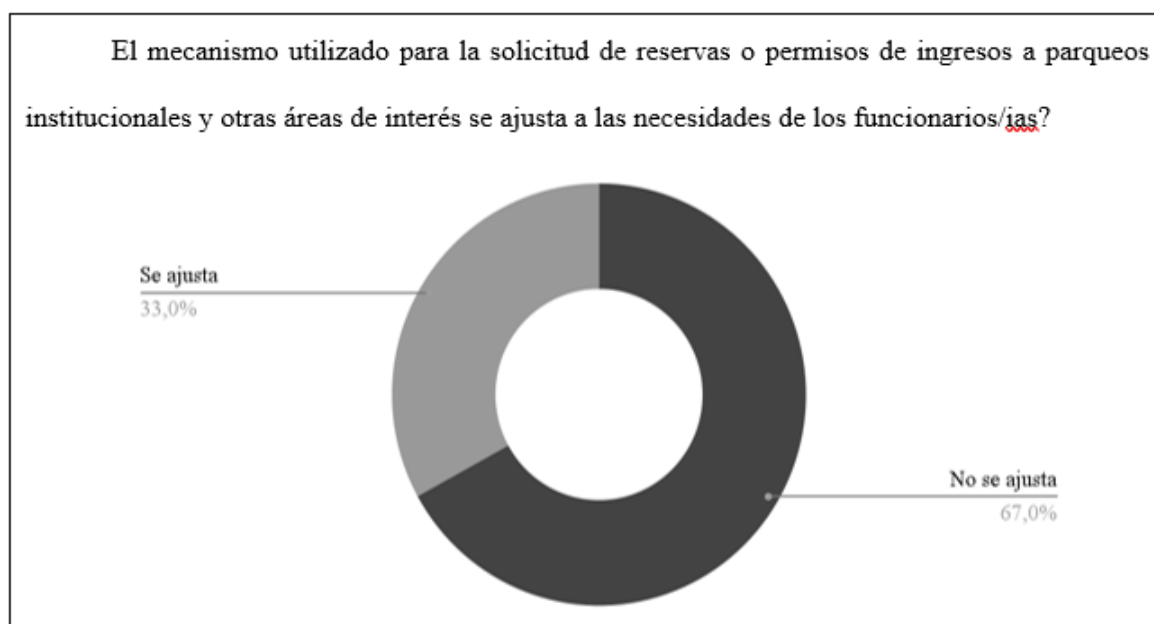


Figura 28. ¿El mecanismo utilizado por la SSI-UNA se ajusta a las necesidades de los funcionarios?

Fuente: Elaboración propia.

También se le preguntó a los entrevistados su opinión sobre la importancia de tener otro mecanismo para el proceso de reserva o permisos de ingreso a parqueos institucionales y a otras áreas. Aquí el 67% (6 de 9) respondió afirmativamente. En este sentido, la percepción de los consultados pareciera ser firme, puesto que no solo conocen el procedimiento (100%), sino que también la mayoría indica que el mismo no se ajusta a las necesidades de los funcionarios. El último elemento de percepción investigado en esta sección consiste en la pregunta sobre la eficiencia del sistema actual utilizado por la SSI-UNA.

El último elemento de percepción indagado en esta sección consistía en preguntar si el sistema actual o mecanismo utilizado por la SSI-UNA, para el proceso de reserva o permisos de ingresos a parqueos institucionales y a otras áreas, es eficiente y eficaz para las siguientes poblaciones: funcionarios, estudiantes y personas externas.

En la Figura 29, “¿Considera usted que el mecanismo utilizado por la SSI-UNA es eficiente y eficaz?” observamos en los datos arrojados que el 78% (7 de 9) señalaron que el proceso actual no es eficiente, y un 22% de los entrevistados opinan que si es un sistema eficaz.

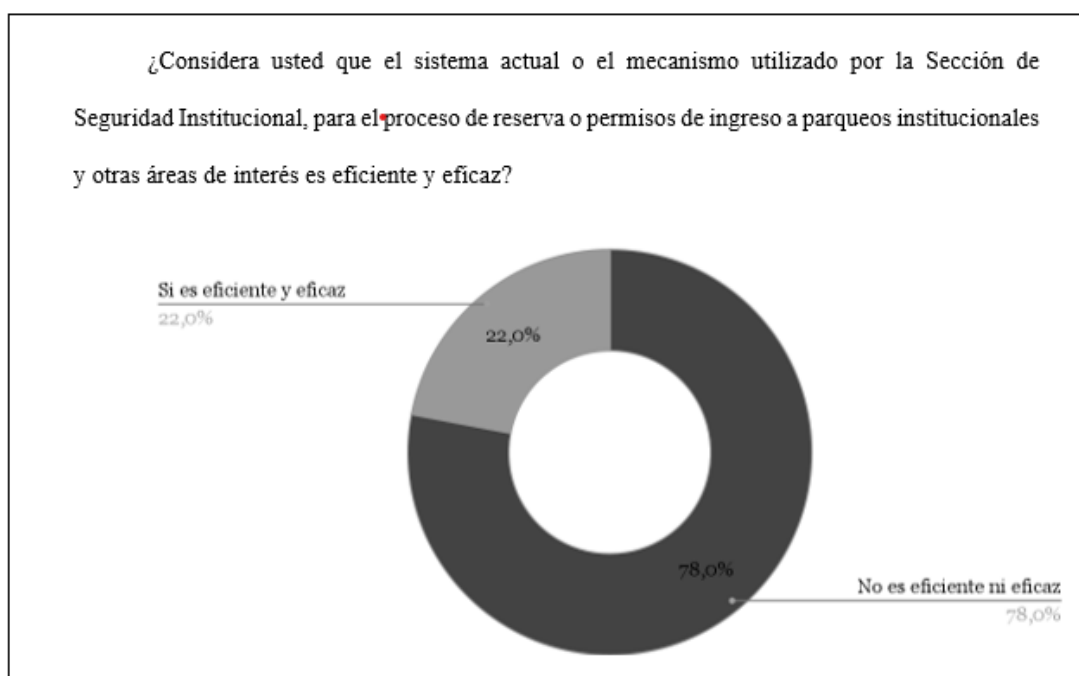


Figura 29. ¿Considera usted que el mecanismo utilizado por la SSI-UNA es eficiente y eficaz?

Fuente: Elaboración propia

La gran mayoría de los encuestados aportó razones específicas de esta percepción. Como ejemplo uno de las entrevistadas manifiesta lo siguiente; “es terriblemente tedioso tener que estar pidiendo permiso para ingresar en dichos vehículos cuando uno sale a hacer compras de la misma oficina donde uno trabaja” (Ver Anexo 2).

Otro participante declaró que “realizan solicitudes de un día para otro y esto no lo pueden gestionar, a veces no logran ver el correo electrónico a tiempo y cuando la persona va a ingresar no lo dejan. Además, la parte académica no tiene claridad los días hábiles de tiempo para realizar la solicitud ante la sección de seguridad” (Ver Anexo 3).

Finalmente, un tercer ejemplo de razonamiento para esta pregunta es la declaración de un funcionario: “Muchas veces es difícil porque aun enviando correo electrónico para ingreso, los funcionarios tienen problemas a la hora de ingresar” (Ver Anexo 4).

Considerando los datos obtenidos discutidos anteriormente, se ratifica que se necesita otro mecanismo o sistema para realizar dichos procesos con el fin de cumplir con todas las necesidades para las tres poblaciones: funcionarios, estudiantes y personas externas.

3.2.5 Sección III: Flujo de trabajo de la SSI-UNA

La tercera sección trata del flujo de solicitud de reserva o permisos de ingresos a parqueos institucionales y otras áreas de interés. Respecto a la primera pregunta acerca del recibimiento de alguna capacitación sobre el proceso, el (67% 6 de 9) refirió que nunca ha recibido alguna

capacitación. El 22% (2 de 9), indicó que ocasionalmente ha recibido capacitación y una persona indicó que casi nunca.

La segunda interrogante de esta sección plantea a los entrevistados describir en pocas palabras el proceso para la solicitud de reserva y permiso de ingresos a la institución. El 67% (6 de 9), detallaron un flujo de trabajo donde resalta la solicitud por medio del correo electrónico. Por ejemplo, uno de los entrevistados manifestó, “Envío correo con la información y se espera la respuesta de la Sección de Seguridad y se le copia a la persona que solicita el permiso esto con 3 días hábiles de tiempo” (Ver anexo 5).

Mientras que otro participante indicó que “se envía un correo a institucional con los nombres y cédulas, horario, asunto y lugar donde se realizará la actividad” (Consulta el Anexo 6).

La tercera pregunta de esta sección pretende indagar sobre la herramienta utilizada el entrevistado para realizar una solicitud de reserva o permiso de ingreso. En esta pregunta los encuestados podían indicar la utilización de más de una herramienta. De forma totalmente concordante con la pregunta anterior el 89% (8 de 9), indicó que ha utilizado el correo electrónico. Otras herramientas utilizadas fueron: llamada telefónica (33%, 3 de 9), trámites físicos (22% 2 de 9) y, finalmente solo un funcionario reportó haber utilizado el sistema web.

De forma similar en la quinta no existe uniformidad sobre la frecuencia, con la que se ha presentado algún problema con la solicitud de reservas y permisos. El 33% (3 de 9), mencionaron que nunca u ocasionalmente respectivamente. 2 de 9 (22%), indicaron que casi nunca y un funcionario (11%) indicó que siempre ha tenido problemas con el dicho proceso.

En otro orden de cosas, como se puede leer en la *Figura 30*, “¿Con cuánta frecuencia usted solicita el servicio Reserva o permiso de ingreso a parqueos institucionales y otras áreas de interés de la SSI-UNA?”

Los resultados obtenidos muestran que un 56% aseguró que más de una vez por semana.

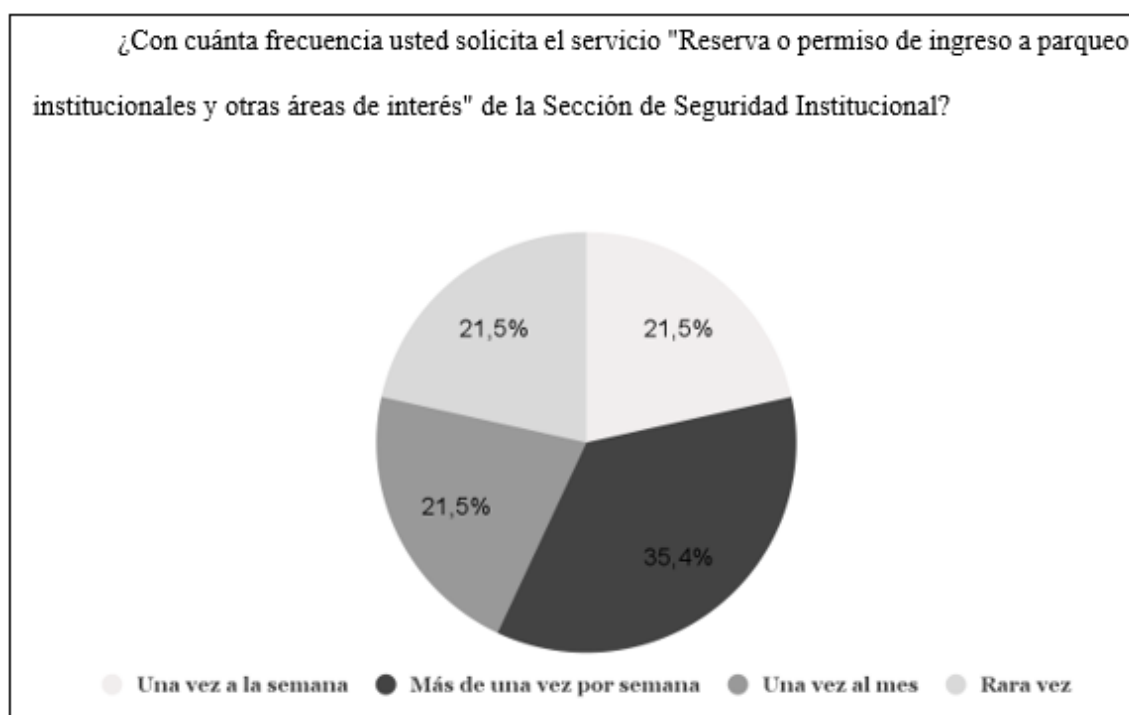


Figura 30. ¿Con cuánta frecuencia usted solicita el servicio Reserva o permiso de ingreso a parqueos institucionales y otras áreas de interés de la SSI-UNA?

Fuente: Elaboración propia.

La pregunta número 7, qué inquirió si alguna vez los funcionarios habían tenido problemas solicitando el servicio, permite clarificar esta situación; el 67% (6 de 9), dice que sí.

Finalmente se preguntó qué tipo de problemas se habían presentado a la hora de solicitar el servicio de reserva y permiso de ingreso. Los entrevistados afirmaron la existencia de más de un problema. 3 de 9 (33%), confirman que habían tenido problemas con comunicaciones con la SSI y falta de claridad del proceso, (respectivamente). Problemas con documentos mal estructurados y con el software fueron reportados por un funcionario. Finalmente, un 44% (4 de 9), refirió otro tipo de problemas.

Para concluir las preguntas de flujo de trabajo de la SSI, en la *Figura 32*. ¿Cuánto ha sido la duración en tiempo para realizar una solicitud a la SSI-UNA? En estos datos no se aprecia la uniformidad en las respuestas puesto que 3 de 9 de entrevistados (33%), respondieron-un día y 2 participantes (22%), reportaron unas horas y una semana respectivamente.

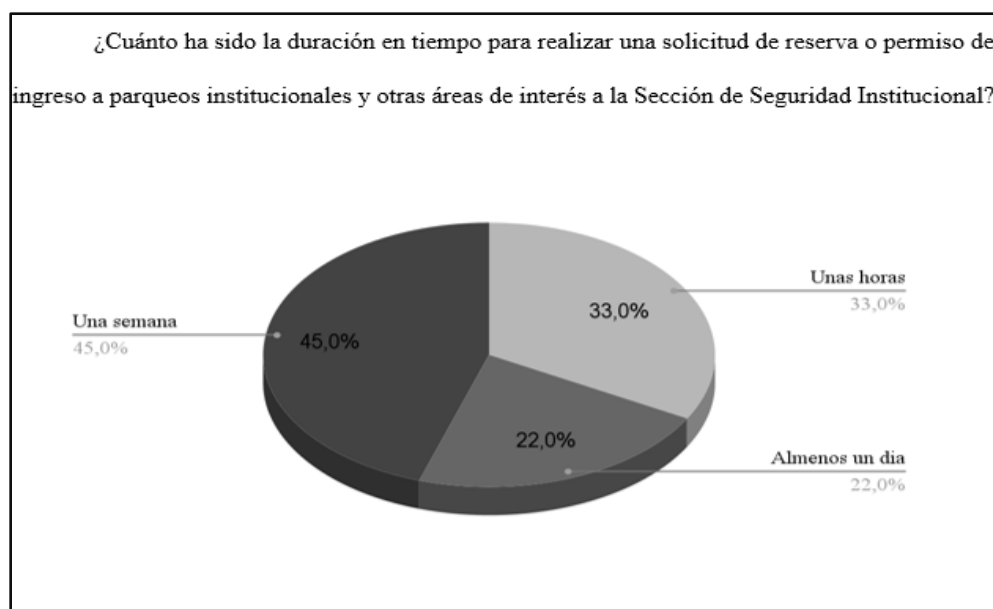


Figura 31. ¿Cuánto ha sido la duración en tiempo para realizar una solicitud a la SSI-UNA?
Fuente: Elaboración propia.

3.3 Descripción de Instrumentos

En esta sección se presenta la ejecución de los cuatro instrumentos metodológicos para llevar a cabo el desarrollo de este proyecto.

En el marco teórico se encuentra todo el referente teórico de los cuatro instrumentos metodológicos descritos anteriormente.

Los cuatro instrumentos metodológicos definidos son:

- Revisión Literatura (RL).
- Encuesta.
- Prototipo, Resultado Documento de Arquitectura de Software SAD.
- Método de Análisis de Compensación de Arquitecturas (ATAM).

3.3.1 Revisión de Literatura (RL)

Se realizó un referente literario sobre la documentación de Arquitectura de Software, sus principales definiciones, los componentes, patrones y estilos que la componen y el ciclo de desarrollo de esta.

Seguidamente, se realizó una Revisión Literaria para recopilar información sobre los 4 instrumentos metodológicos ejecutados para desarrollar este proyecto.

Luego se efectuó una Revisión Literaria, para extraer información de fuentes a tres Arquitecturas de Software, las cuales fue arquitectura del modelo 4+1, arquitectura del modelo BPM y la arquitectura de Servicios Web.

En la siguiente sección “Procedimientos para analizar la información del diagnóstico” se detalla en datos porcentuales que tipo de fuente bibliográfica se utilizó para sentar las bases de este proyecto y el tipo de publicación.

En la Figura 32, “Ejecución del Instrumento Revisión Literaria” (RL), observamos los pasos ejecutados para aplicar este instrumento metodológico. Este se ajustó a las etapas de la taxonomía de Cooper (1998), primeramente, se buscó referencias bibliográficas para luego evaluarlas, con el objetivo de encontrar referentes confiables y especialmente reportes científicos.

El segundo proceso que se realizó fue leer para comprender lo investigado, en este proceso primeramente se realizó una clasificación de referencias para diferenciarlas fácilmente. Seguidamente se trabajó en el resumen examinando cada investigación u aporte científico para luego realizar una toma de decisiones y realizar un juicio de valor en base a lo que se estaba investigando.

En la última etapa se inició con la escritura de la RL, aportando por cada definición o idea al menos 3 referencias, con el objetivo de sustentar la investigación.

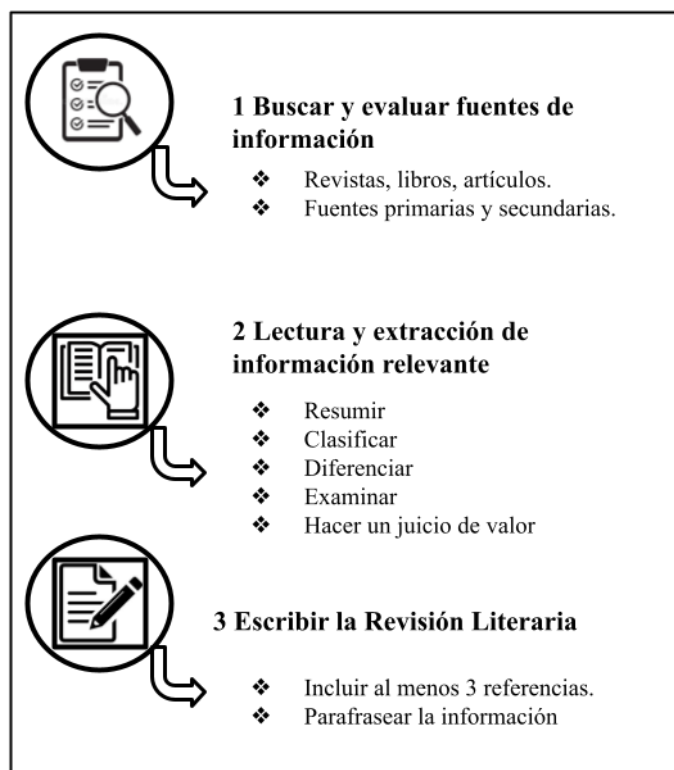


Figura 32. Ejecución del Instrumento Revisión Literaria (RL).

Fuente: Elaboración propia.

3.3.2 Encuesta

El objetivo de la ejecución de este instrumento, utilizando información tomada en base a lo manifestado por la directora de la Sección de Seguridad Marjorie Fernández (2020), es conocer el modelo de negocio, el proceso para obtener una reserva de este tipo con el fin de realizar un levantamiento de los requerimientos. Para ver la encuesta realizada esta se encuentra en el apartado Anexos, en el anexo 1a-1d.

En la Figura 33, “Ejecución del Instrumento Encuesta”, se observan los pasos que se abarcó para llevar a cabo el cumplimiento de este método.

La ejecución de este instrumento se realizó por medio de 4 etapas: Primeramente, la etapa de planeación que correspondió en la fijación de objetivos bien definidos y en determinar la población. Segunda etapa sobre la realización que consistió en la elaboración del cuestionario que correspondió a varias versiones en papel y luego la búsqueda de la herramienta adecuada para diseñarla. En la etapa de distribución se recogió la información los datos presencialmente, visitando las Escuelas y los Departamentos ¹⁵elegidos. En esta etapa se recopiló todas las respuestas para ser analizadas y con ello graficar la información. Esto con el fin de demostrar los resultados obtenidos. En la etapa final, la fase de decisión corresponde a la toma de decisiones y con ello establecer un juicio de valor.

¹⁵ Las escuelas elegidas para la encuesta se muestran en la sección población y muestra de este capítulo.

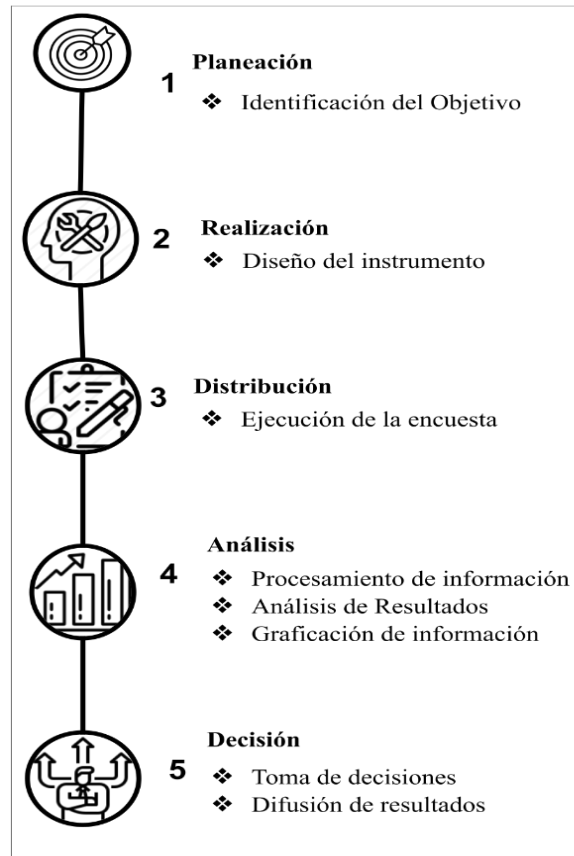


Figura 33. Ejecución del Instrumento Encuesta

Fuente: Elaboración propia.

3.3.3 Prototipo de documentación de una arquitectura integral

En el instrumento metodológico de prototipo dio como resultado el Documento de Arquitectura de Software (SAD).

En la *Figura 34*, “Ejecución del Instrumento Prototipo” se observa que, para determinar las especificaciones de requerimientos, se utilizó la técnica de encuesta, posterior a esto se realizó un estudio para determinar el tipo de solución y tecnología que se usará, para luego documentar la Arquitectura de Sistema Integral.

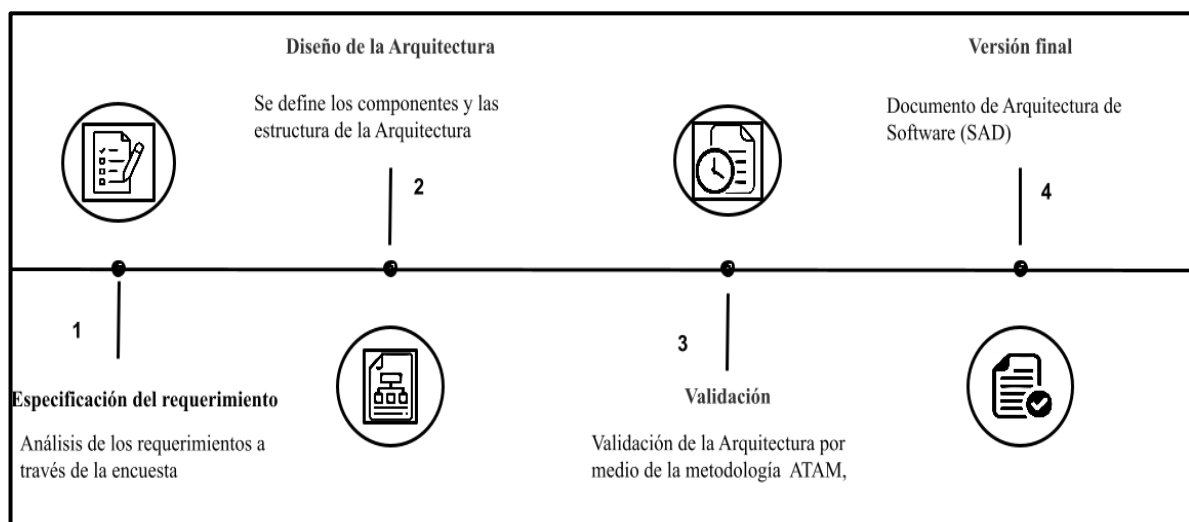


Figura 34. Ejecución del Instrumento Prototipo

Fuente: Elaboración propia

En la fase de diseño de la arquitectura, se definió la estructura y las responsabilidades de los componentes que comprenderán la Arquitectura de Software. En la fase de validación; en este punto se probó la arquitectura, típicamente pasando a través del diseño contra los requerimientos actuales y cualquier posible requerimiento a futuro, en esta fase utilizamos el método ATAM, que es una técnica de escenarios desarrollado en el SEI (Software Engineering Institute).

3.3.4 Método de Análisis de Compensación de Arquitecturas (ATAM)

El referente teórico del Método de Análisis de Compensación de Arquitecturas, (ATAM), se desarrolla en el marco teórico. Para efectos de este trabajo se tomaron los siguientes siete principios de este método. Cabe mencionar que para llevar a cabo la evaluación se realizó por medio de un ¹⁶sitio web, por la situación mundial del virus SARS-CoV-2 y con ello evitar la comunicación física.

A continuación, se describen los 7 principios de ATAM (Metodología de Análisis de Compensación de Arquitectura), elegidos y ejecutados en este trabajo.

- Presentación del ATAM.
- Presentación de modelo de negocio.
- Presentación de la propuesta de arquitectura.
- Identificación de los atributos arquitecturales de calidad.
- Generación y análisis del Árbol de Utilidad.
- Revisión y votación de los atributos de calidad.
- Presentación y correlación de los resultados obtenidos.

¹⁶ WWW.evaluacionarquitecturas.atwebpages.com

Los pasos ejecutados en este método se contemplan en la *Figura 35* “Ejecución del Instrumento Método de Análisis de Compensación de Arquitecturas. (ATAM). Basados en lo que señala Clements, Kazman, & Klein, (2003), primeramente, se realizó la presentación de ATAM, posteriormente, la presentación del Modelo de Negocio, de la Sección de Seguridad Institucional, seguidamente la identificación de los atributos arquitecturales de calidad por medio de una encuesta.

Los resultados obtenidos de esta encuesta fueron analizados por los autores y con ello se realizó la generación y el análisis del Árbol de Utilidad, (UT). El siguiente paso que corresponde al paso 6, se realizó con el grupo de evaluación, para la revisión y votación de los atributos de calidad y para concluir se presentó la correlación de los resultados obtenidos.

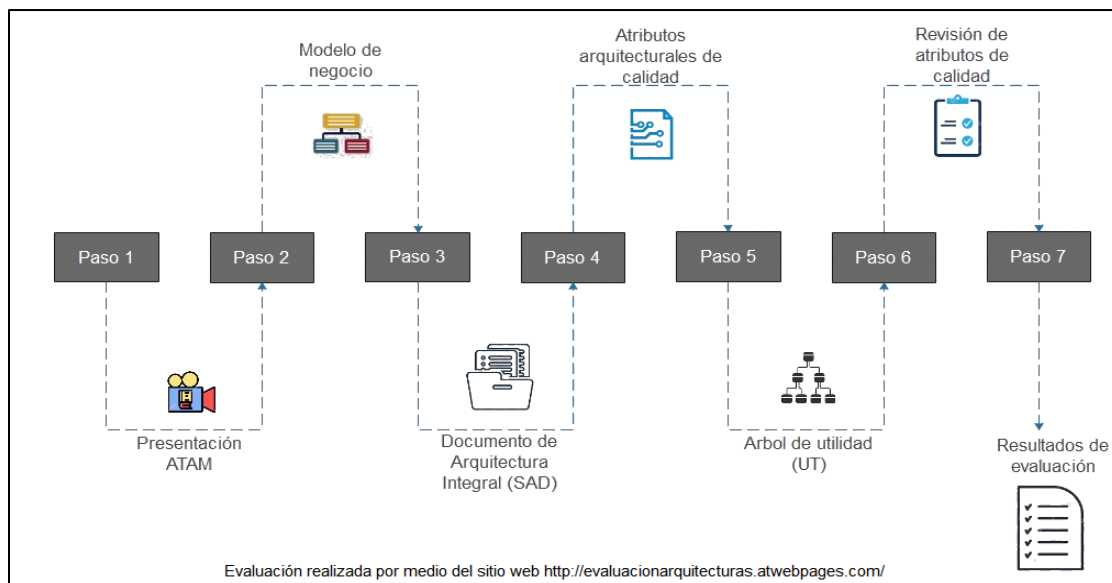


Figura 35. Ejecución del Instrumento (ATAM).

Nota. En el Capítulo 4 en la sección 4.3, sobre “Validación de la propuesta”, se desarrolla toda la ejecución del Método de Análisis de Compensación de Arquitecturas.

Fuente: Elaboración propia.

3.4 Procedimientos para Analizar la Información del Diagnóstico

A continuación, se muestra los procedimientos que se tomaron para analizar la información, mismos que se detallan con base en la técnica correspondiente para revisión de literatura, encuesta y ATAM.

El análisis de la información utilizada indica que las fuentes bibliográficas en este artículo fueron diversas. Se utilizaron 116 cantidades de referencias en el MT 78 referencias bibliográficas. El 97.4% (113) son fuentes primarias y el 2.6% fuentes secundarias. El medio de publicación utilizado fue de revistas en un 16.3% (19), conferencias de alto nivel en un 5.17% (6), referencias de libros un 45.6% (53), tesis y sitios de internet como blogs reconocidos ambos para un 15.4% consistentes en una referencia por cada uno de estos medios de publicación. El análisis de la antigüedad de estas referencias bibliográficas indica que sólo el 19.8% (23) fueron inferiores al año 2000, el 53.44% (62) corresponden al periodo del año 2000 al 2010, para el periodo comprendido entre el 2010 y el 2021, año de publicación del artículo, se utilizaron un 26.7% (31) de referencias bibliográficas.

Respecto a la encuesta se analizaron los datos de 9 personas en donde el 22% (2) personas tiene menos de 5 años trabajando en la Universidad, el 44.4% (4) tiene menos de 10 años, pero más de 5 años de estar en trabajando en la Universidad y el 33.3% (3) tiene más de 10 años

Finalmente, en cuanto a ATAM, se analizaron los datos de 7 arquitectos que participaron en la evaluación, en donde el 71% (5) tiene más de 5 años de experiencia, pero menos de 10 años y el 29% (2) tienen una experiencia mayor a 10 años.

Capítulo IV. Propuesta de Solución

4.1 Diagnóstico

Este capítulo corresponde a la propuesta de solución, el cual está dividido en tres secciones; el diagnostico, la propuesta de solución y por último la validación de la propuesta.

Para la elaboración del diagnóstico de este proyecto se describieron y se caracterizaron los problemas y con ello se pudo identificar la solución que tuvo mayor impacto.

4.2 Propuesta de Solución

El documento se desarrolla y organiza en base a la plantilla elaborado por Rational Software Corporation, llamado Software Architecture Document (SAD). (Rational Software Corporation, 2001).

Se adaptó las características particulares del proyecto desarrollado, el cual habitualmente se organiza por vistas, a través de las cuales pueden especificarse los distintos aspectos técnicos y funcionales, así como también las decisiones involucradas en cada decisión. El Documento de Arquitectura de Software, proporcionó una visión general de la arquitectura del sistema de software.

En el año 2003, Bass y otros dejó claro que la arquitectura de software lo que haces es mostrar las estructuras de un sistema, que están compuestas de elementos con propiedades visibles

de forma externa y con relaciones existentes entre ellos (Bass, Clements, & Kazman, Software architecture in practice, 2003).

4.2.1 Diseño de la Arquitectura de Software propuesto

Como ya mencionamos esta propuesta corresponde a un documento integral, que contempla varias secciones y apartados. La información contenida en la *Figura 36*, “Documento de Arquitectura de Software.” se puede observar la visión general de las partes que contempla el Documento de Arquitectura de Software.

Alcances
Planteamiento del problema
Objetivos y restricciones
Acrónimos, Siglas y definiciones
Glosario de términos y definiciones
Descripción de la documentación de la Arquitectura de Software
Arquitectura del modelo de Gestión de Procesos de Negocio (BPM)
Business Process Modeling Notation (BPMN)
Arquitectura del Modelo 4+1
La Arquitectura Lógica
Arquitectura de Procesos
La Arquitectura de desarrollo
Arquitectura Física
Método de Usabilidad
Vista de Datos (E-R)
Especificaciones preliminares de requerimientos
Calidad y rendimiento de procesos

Figura 36. Documento de Arquitectura de Software.

Fuente: Elaboración propia.

4.2.2 Alcances

En relación con los alcances de este trabajo: Documentación de Arquitectura de Software permite conocer como está organizado los componentes del sistema. Esto con el fin de brindar el desempeño óptimo por medio de los requerimientos funcionales y no funcionales que cada usuario necesita.

4.2.3 Objetivos y restricciones

En este apartado se incluye los objetivos y restricciones, con el fin de proporcionar un marco de referencia para la realización de la documentación de la arquitectura. Además de las limitantes que se pueden encontrar.

Los objetivos de la documentación de esta Arquitectura para la Sección de Seguridad Institucional de la Universidad Nacional; es conocer las necesidades de los involucrados en el proceso.

Las principales restricciones de una documentación de Arquitectura es limitar el rango de alternativas del arquitecto, así como encontrar algunas restricciones que imponen condiciones a la arquitectura que no son negociables.

4.2.4 Acrónimos, Siglas y definiciones

En este apartado se presentan los acrónimos¹⁷, las siglas¹⁸ y el significado de cada una de ellas, con el fin de guiar y facilitar la comprensión de estas en el documento.

En la Tabla 9, “Acrónimos, siglas y sus definiciones” se muestra una lista de las principales siglas y acrónimos, que se encontrarán en el documento.

Tabla 9. Acrónimos, siglas y sus definiciones

Acrónimos	Significado
BD	Base de Datos
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Modeling Notation
MVC	Modelo Vista Controlador
ORM	Object Relational Model

¹⁷ Tipo de sigla que se pronuncia como una palabra.

¹⁸ También se llama acrónimo a la sigla que se pronuncia como una palabra, ejemplo: RAM, IBM, ovni, sida.

SAD	Software Architecture Document
SSI	Sección de Seguridad Institucional
SSI-UNA	Sección de Seguridad Institucional de la Universidad Nacional
UNA	Universidad Nacional

Fuente: Elaboración propia.

4.2.5 Glosario de términos y definiciones

En este apartado se muestra el glosario de términos¹⁹ y la definición correspondiente.

En la Tabla 10, “Términos y Definiciones” se muestra una lista de las principales palabras y expresiones, que a simple vista son difíciles de comprender. Esto se hace con el fin de guiar al lector y que se sienta familiarizado con el documento.

¹⁹ Conjunto de términos o vocablos propios de determinada profesión, ciencia o materia.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Tabla 10. Términos y Definiciones

Término	Definiciones
Arquitectura de Software	Representa la estructura y los componentes de un Software, así como las propiedades visibles externamente y las relaciones entre ellas.
Base de datos	Conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.
Data Center	Denominado Centro de Proceso de Datos, al espacio donde se concentran los recursos necesarios para el procesamiento de la información de una organización.
Diagrama	Representación técnica de un diseño de arquitectura sobre las funciones, comportamientos y relaciones de un futuro sistema informático.
Documentación de Arquitectura de Software	Documentos que representa y abarcan los lineamientos, las funcionalidades y representaciones de un sistema de software.
Firewall	En español un cortafuego es la parte de un sistema o red informáticos que controla el acceso de una computadora a la red y de elementos de la red a la computadora, por motivos de seguridad.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Heurística	Conjunto de técnicas y métodos para resolver un problema.
Modelo 4+1	Modelo diseñado por el ingeniero Philippe Kruchten para describir la arquitectura de un software.
Motor de Base de Datos	Es el servicio principal para almacenar, procesar y proteger los datos.
Requerimientos funcionales	Son los comportamientos del futuro sistema, las actividades, procesos y servicios que este proveerá. Ejemplo, manipulación de datos, reportes, entre otros.
Requerimientos No funcionales	Son las características generales y sus restricciones, ofrecidos por el sistema informático. Ejemplo, seguridad, funcionabilidad, entre otros.
Server	Un servidor web (server) es un ordenador de gran potencia que se encarga de “prestar el servicio” de transmitir la información pedida por sus clientes.
Servidor BD	Programa que provee servicios de base de datos a otros programas u otras computadoras.
Stakeholders	Es cualquier individuo o grupo que forma parte de forma directa e indirectamente en el proceso y desarrollo de este producto.
Switch	Un switch o conmutador es un dispositivo de interconexión utilizado para conectar equipos en red

CAPÍTULO IV

PROPUESTA DE SOLUCION

	formando lo que se conoce como una red de área local (LAN).
Vistas	Representación de un área de interés y perspectiva de sistema en alto nivel.
Workstation	En español estación de trabajo es un computador de altas prestaciones destinado para trabajos técnico o científicos.

Fuente: Elaboración propia.

4.2.6 Descripción de la documentación de la Arquitectura de Software

Este documento tiene la finalidad de realizar una presentación, por medio de la selección de arquitecturas acordes a la finalidad de los procesos de negocios y los flujos de trabajo, donde permita analizar la efectividad del diseño y con ello cumplir con los requisitos establecidos.

Se eligieron dos arquitecturas; el modelo BPM que permite definir y coordinar las actividades con claridad de la organización y el modelo 4+1 que es popular, porque ayuda a describir un sistema de software basado en el uso de múltiples puntos de vista.

4.2.7 Arquitectura del modelo de Gestión de Procesos de Negocio (BPM)

En este capítulo se incluye el modelo de Gestión de Procesos de negocio, con sus siglas en inglés BPM (Business Process Management); se describe y se representa por medio de imágenes con su debida notación²⁰.

Este modelo de arquitectura es un enfoque disciplinado orientado a los procesos de negocio²¹. Además, es una técnica que documenta procesos de un sistema, sus entradas, sus salidas y sus formas de almacenamiento de datos (Alarcón, V. F. 2006).

4.2.8 Business Process Modeling Notation (BPMN)

Esta sección incluye la notación del modelo de arquitectura de Gestión de Procesos de Negocio, llamado Modelo y Notación de Procesos de Negocio, BPMN (Business Process Modeling Notation) y su representación.

“La principal función del Modelo y Notación de Procesos de Negocio es proveer de una notación gráfica que permita automatizar los procesos”. Rolón & Piattini. (2006) (pp. 419-432).

²⁰ Notación: Es el conjunto específico de símbolos y sus reglas de uso en la descripción de una cosa.

²¹ Proceso de negocio: Conjunto de funciones secuenciales que entrega valor a un cliente. Los procesos son iniciados por eventos externos claramente definidos.

Ejemplo del Modelado de Negocio

En la Figura 37, Diagrama del Modelado de Negocio de BPM”, se incluye los elementos generales que modelan el sistema desde la solicitud de reserva o permiso de ingreso hasta su aprobación de las siguientes perspectivas la de usuario, de la unidad de seguridad y de la unidad de monitoreo.

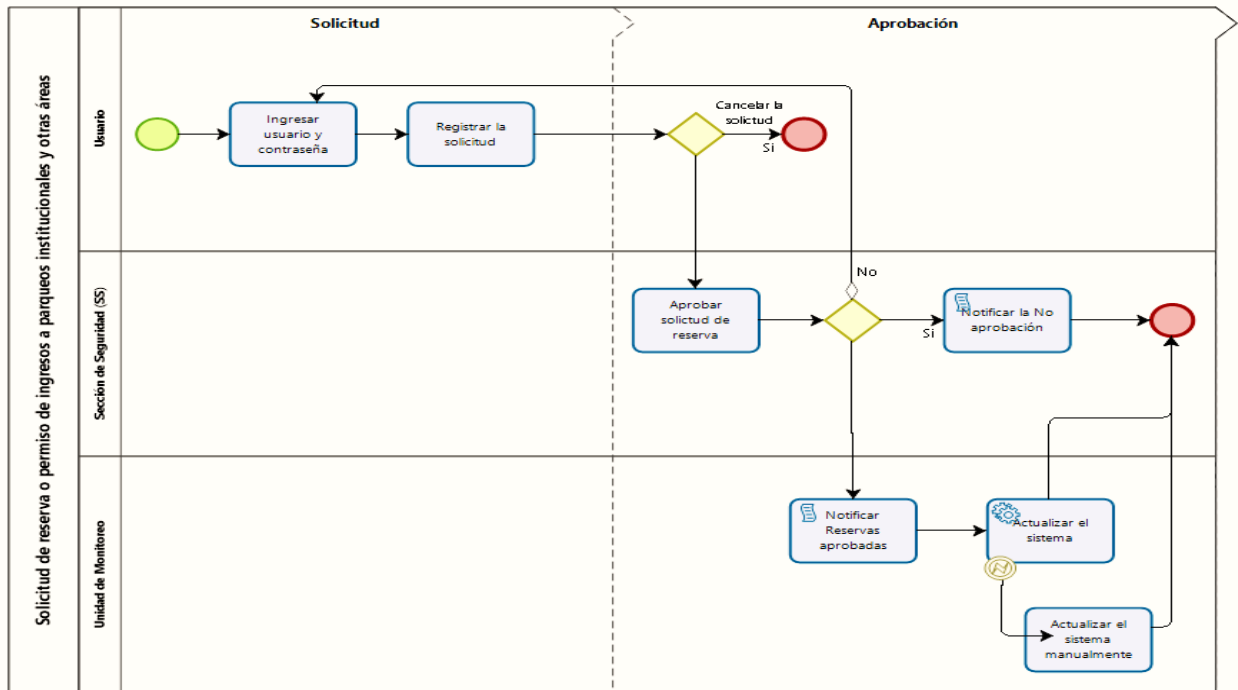


Figura 37. Diagrama del Modelado de Negocio de BPM.

Fuente: Elaboración propia.

4.2.9 Arquitectura del Modelo 4+1

El modelo 4+1 fue diseñado por Philippe Kruchten que describe la arquitectura del software usando vistas concurrentes. Cada vista se refiere a un conjunto de intereses de diferentes stakeholders del sistema. Para este documento de arquitectura integral se eligió las siguientes 4 vistas, Kruchten, (1995).

La **vista lógica** describe el modelo de objetos del diseño cuando se usa un método de diseño orientado a objetos.

- La **vista de procesos** describe los aspectos de concurrencia y sincronización del diseño.
- La **vista física** describe el mapeo del software en el hardware y refleja los aspectos de distribución.
- La **vista de desarrollo** describe la organización estática del software en su ambiente de desarrollo.

Arquitectura Lógica. (Descomposición Orientada a Objetos)

Esta sección se basa principalmente en los requerimientos funcionales de la arquitectura Lógica, así como descomponer el sistema en un conjunto de abstracciones, en forma de objetos o clases de objetos.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Como lo establece Kruchten, la vista lógica describe el modelo de objetos del diseño cuando se utiliza un método de diseño orientado a objetos (Krutchen, P. B, (1995).

La documentación de la vista lógica se realiza por medio del diagrama de clases, el diagrama de secuencia y el diagrama de comunicación.

Ejemplo de la Vista Lógica

En la *Figura 38*, “Diagrama de la Vista Lógica del sistema” se contempla los módulos generales que van a operar para el sistema.

Un módulo de autenticación que implementa la asignación de privilegios de los usuarios, solamente se puede acceder a este módulo con su clave y su usuario otorgado por la institución.

En el módulo de acceso los usuarios con ese privilegio tendrán acceso a los recursos del sistema. En el modelo solicitud solamente los usuarios podrán solicitar reservas o permisos de ingresos a parqueos y otras áreas de interés de la Universidad Nacional.

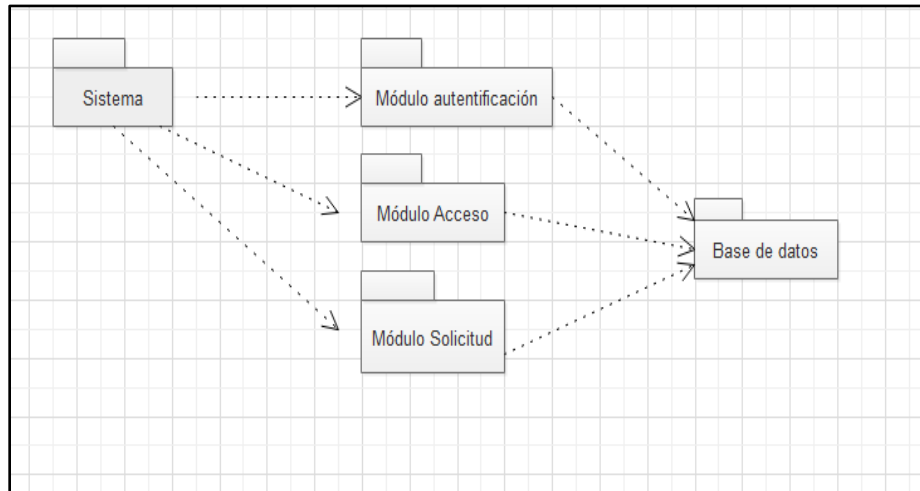


Figura 38. Diagrama de la Vista Lógica del Sistema.

Fuente: Elaboración propia

Diagrama de Clase.

En esta sección se incluye la descripción del diagrama de clases y la representación de este diagrama.

El diagrama de clases es un conjunto de clases y sus relaciones lógicas: la asociación, el uso, composición, herencia, etc. Los conjuntos de clases relacionadas se pueden agrupar en categorías de clase. Kruchten, P. (1995).

CAPÍTULO IV

PROPUESTA DE SOLUCION

Ejemplo del diagrama de clases.

En la *Figura 39*, “Diagrama de clases que muestra la vista Lógica del Sistema” se observan las descripciones de los objetos que comparten iguales atributos, métodos y relaciones.

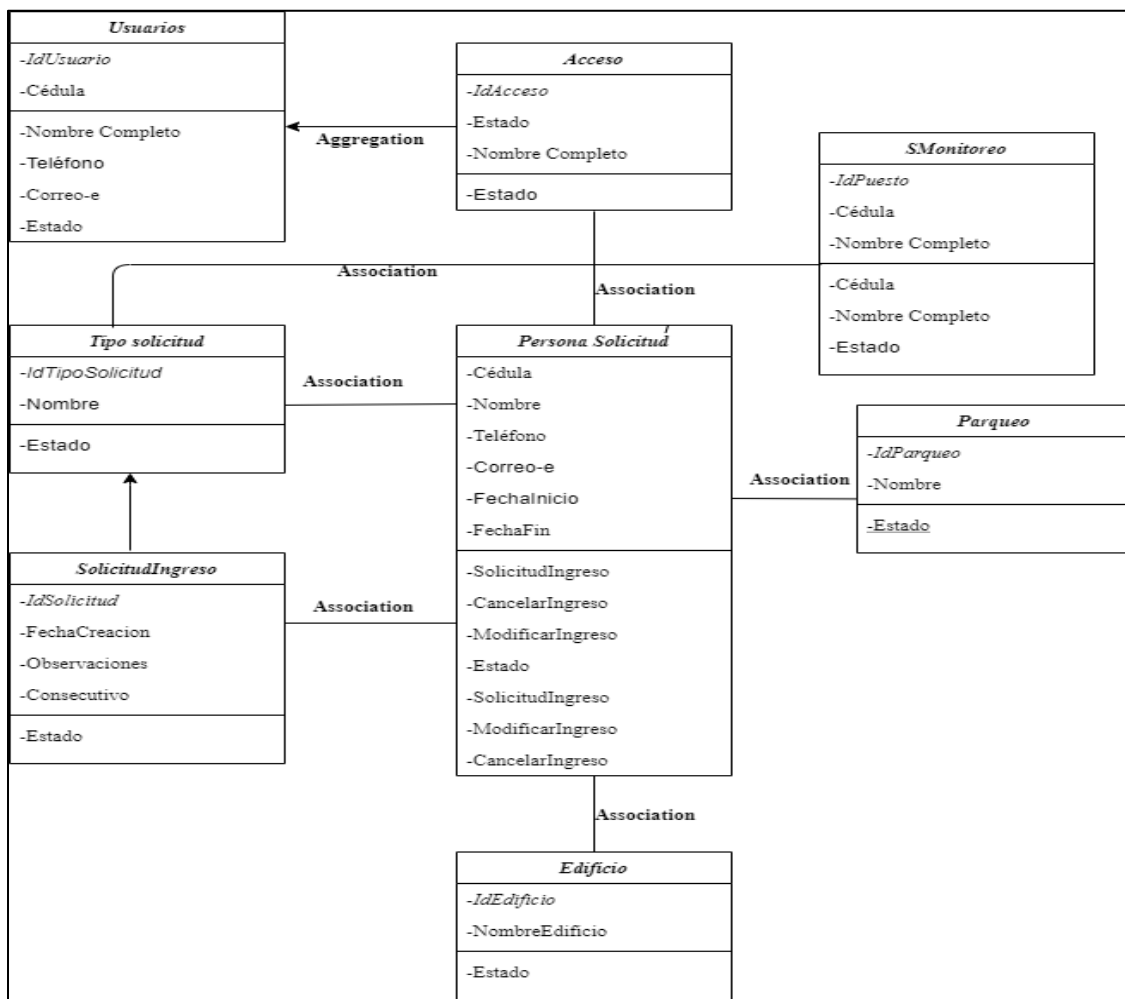


Figura 39. Diagrama de Clases que Muestra la Vista Lógica del Sistema.

Fuente: Elaboración propia.

En la clase *SolicitudIngreso* su relación es la agregación o composición, el objeto base se construye a partir del objeto incluido, en este caso con *el TipoSolicitud*. Además, como se observa con la clase *PersonaSolicitud* tiene una relación de asociación con *Acceso* este tipo de relación permite la interacción entre sí.

Diagrama de Secuencia

En esta sección se incluye la descripción del diagrama de Secuencia y la representación de este diagrama.

El diagrama de secuencia describe la dinámica del sistema por medio de un grupo de objetos mostrando de forma secuencial los envíos de mensajes entre ellos. A esta instancia u objeto se le asocia una línea de vida que muestra las reacciones y acciones.

Durante la recepción de un mensaje los objetos se vuelven activos y ejecutan el método del mismo nombre. Un envío de mensaje es por tanto una llamada a un método.

Ejemplo del diagrama de Secuencia

El diagrama de secuencia muestra la interacción entre objetos en el sistema.

En la *Figura 40*, “Diagrama de Secuencia que muestra la vista Lógica del Sistema” se muestra el diagrama compuesto por 6 objetos o instancias que tiene interacción con el sistema, por medio de una secuencia de mensajes.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Esta interacción se realiza desde la solicitud de la reserva hasta su aprobación por parte de la Sección de Seguridad y luego se hace de conocimiento por la unidad de monitoreo, para ser ejecutada físicamente.

La primer instancia usuario es el primer emisor, encargada de solicitar una reserva o permiso de ingreso ya sea a parqueos institucionales y otras áreas de interés, dónde quien recibe esta petición es el sistema o software. Este acceso únicamente se realiza por medio de la clave unificada²².

²² La Clave Unificada es una clave de acceso proporcionada por la institución a todos sus administrativos y académicos.

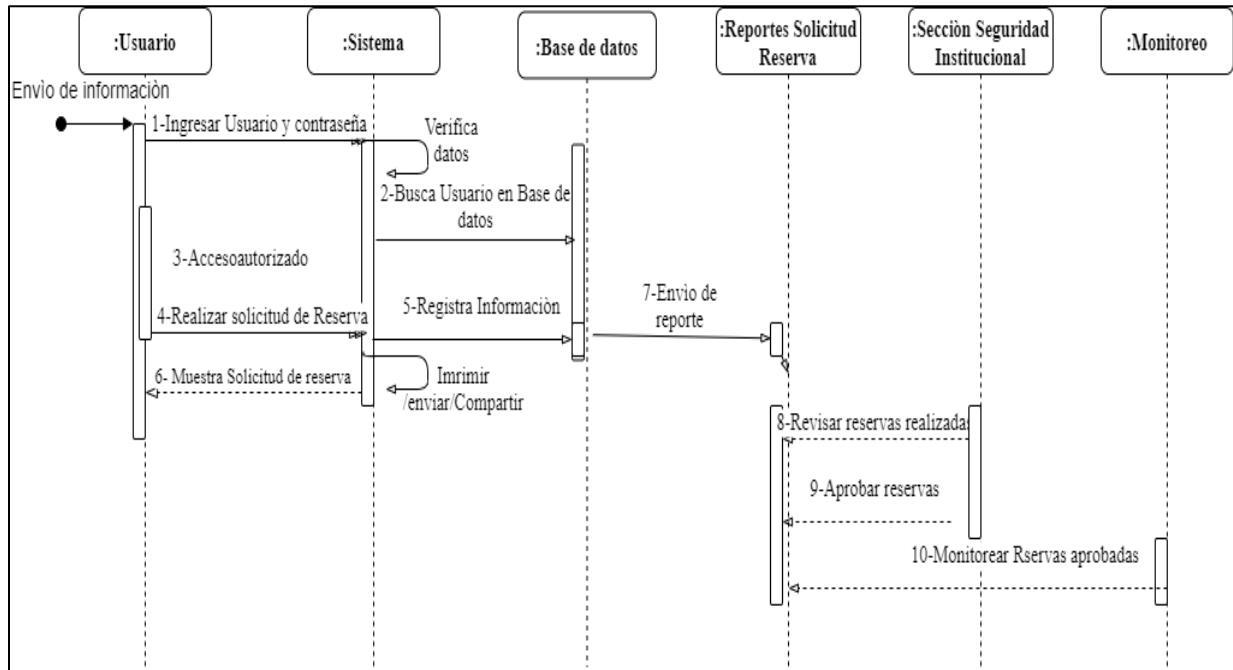


Figura 40. Diagrama de Secuencia que muestra la vista Lógica del Sistema.

Fuente: Elaboración propia.

Diagrama de Comunicación

En esta sección se incluye la descripción del diagrama de Comunicación y la representación de este.

En los modelados de comunicación la colocación de los objetos es más flexible y permite mostrar de forma más clara cuáles son las colaboraciones entre estos. En este tipo de diagrama, la comunicación entre objetos se denomina vínculo o enlace, el cual está asociado con el o los mensajes que intercambian los objetos. Brey, G. A., & Blanco, S. (2005).

Ejemplo del diagrama de Comunicación

En la *Figura 41*, “Diagrama de Comunicación que muestra la vista Lógica del Sistema” se puede observar los mensajes enumerados entre cada objeto, desde la validación del usuario hasta el proceso de reservar el ingreso a parqueos institucionales y otras áreas de interés. Luego se muestra la relación que realiza los objetos de la Unidad de Monitoreo y la Sección de Seguridad Institucional con el sistema.

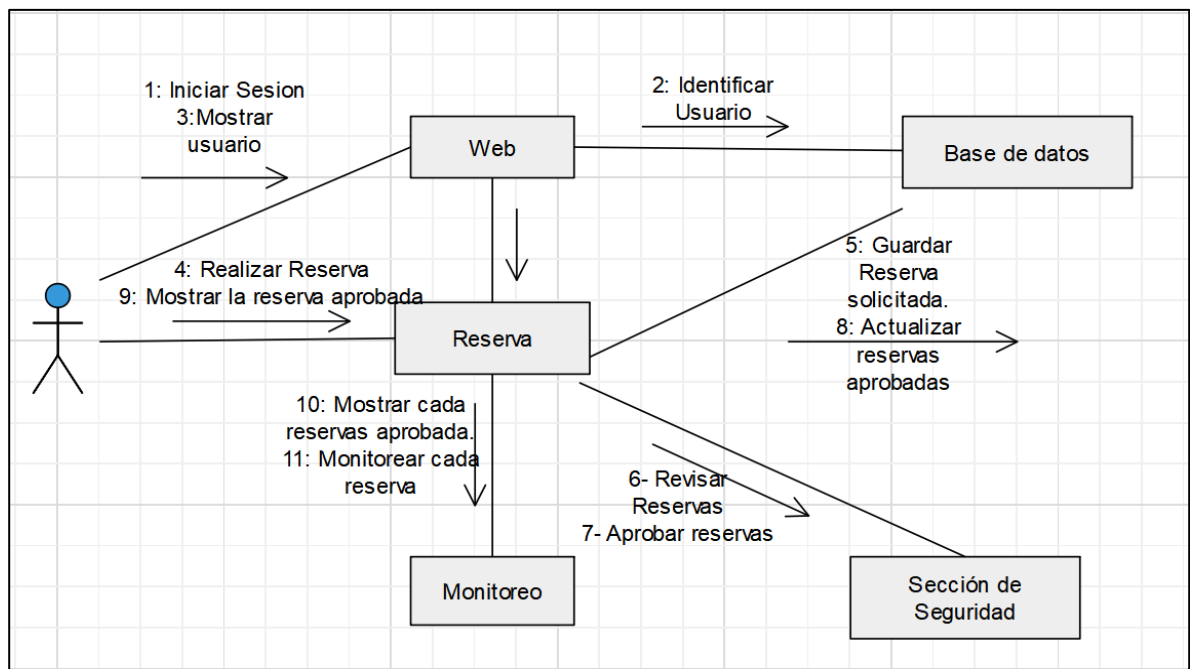


Figura 41. Diagrama de Comunicación que Muestra la Vista Lógica del Sistema.

Fuente: Elaboración propia.

Arquitectura de Procesos (La descomposición de procesos)

En esta sección se basa principalmente en representar los múltiples procesos que un sistema informático debe realizar.

Kruchten destaca lo siguiente “La vista de proceso describe los aspectos de concurrencia y sincronización del diseño” (Kruchten, P. B., (1995) y sin dejar de lado aspectos como disponibilidad, integridad, la tolerancia a fallos, el rendimiento y la escalabilidad que los procesos puedan representar.

La documentación de la vista de procesos se realiza por medio del diagrama de actividad.

Ejemplo del diagrama de actividad de la Vista de Procesos

En la *Figura 42*, “Diagrama de actividad que muestra la vista de Procesos del Sistema”, el diagrama de actividad se realiza con el fin de facilitar una mayor comprensión en los procesos que realizará el sistema futuro.

Está compuesto por cuatro grupos o calles, estas son las organizaciones o responsables de las actividades, además de algunas bifurcaciones o caminos alternativos, que se encontrarán durante el proceso.

Por ejemplo, en la primera columna de usuario, si el solicitante, no completó algún espacio del formulario, se le debe de enviar un mensaje, para proceder a llenar los espacios faltantes, hasta que se complete la acción correctamente y se prosiga a la siguiente actividad.

CAPÍTULO IV

PROPUESTA DE SOLUCION

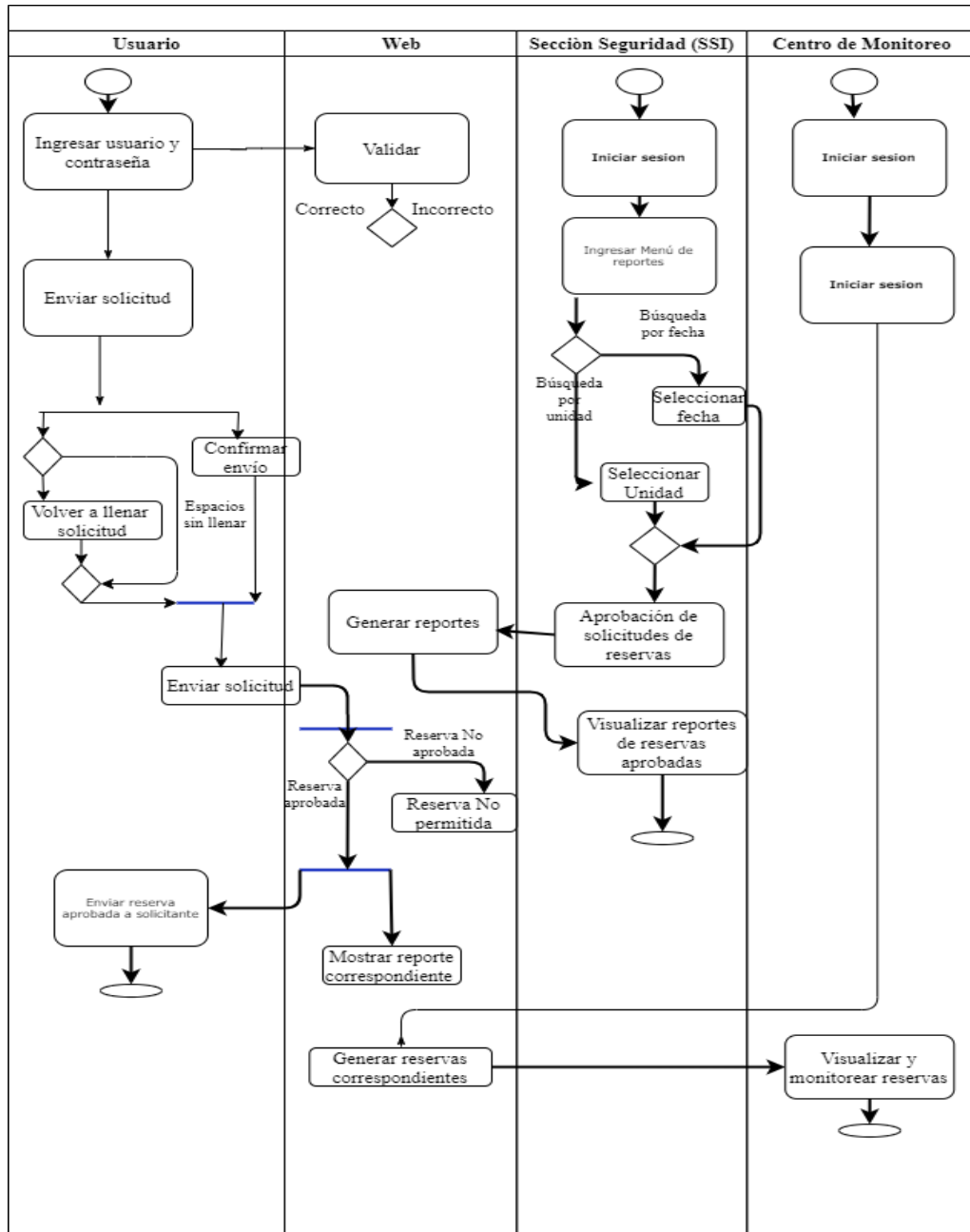


Figura 42. Diagrama de actividad que muestra la vista de Procesos del Sistema.

Fuente: Elaboración propia

Arquitectura de desarrollo (Descomposición del Subsistema)

Esta sección se basa en representar la arquitectura de desarrollo, donde se presenta y se realiza una descomposición del sistema en diferentes librerías, subsistemas y componentes.

La vista de desarrollo apoya la asignación de requisitos, la evaluación de costos, la planificación, el monitoreo de progreso del proyecto y también funciona como base para analizar la reutilización, la portabilidad y seguridad.

La documentación de la vista de desarrollo se realiza por medio del diagrama de Componentes y el diagrama de paquetes.

Ejemplo del diagrama de componentes de la Vista de Desarrollo

En la *Figura 43*, “Diagrama de componentes que muestra la vista de desarrollo del Sistema”, donde se muestra los cinco paquetes generales que modelaran el sistema; el paquete de presentación, modelo, control, servidor y base de datos.

El paquete de presentación está compuesto por las interfaces de usuario, en este caso Interfaz Administrador e Interfaz Usuario. Luego el paquete modelo está compuesto por un ORM (Object Relational Model) que son los modelos utilizados en la arquitectura del sistema.

Seguidamente, el paquete control, es la lógica interna encargada de controlar las solicitudes del usuario, para este caso el subsistema de usuario, solicitudes, reservación e informes.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Finalmente, el paquete servidor integrado por el servidor a utilizar y el paquete de base de datos, con los componentes servidor BD, motor de BD y la base de datos como tal.

El flujo que se maneja está basado en MVC en donde a través de las interfaces de paquete de presentación el usuario realiza solicitudes al sistema, este a su vez se comunica con el control.

Dependiendo de la petición este ingresa a un subsistema y realiza la comunicación con el servidor. El servidor se comunica con el modelo necesario y el modelo con la base de datos donde la base de datos a través del motor de BD extrae los datos.

CAPÍTULO IV

PROPUESTA DE SOLUCION

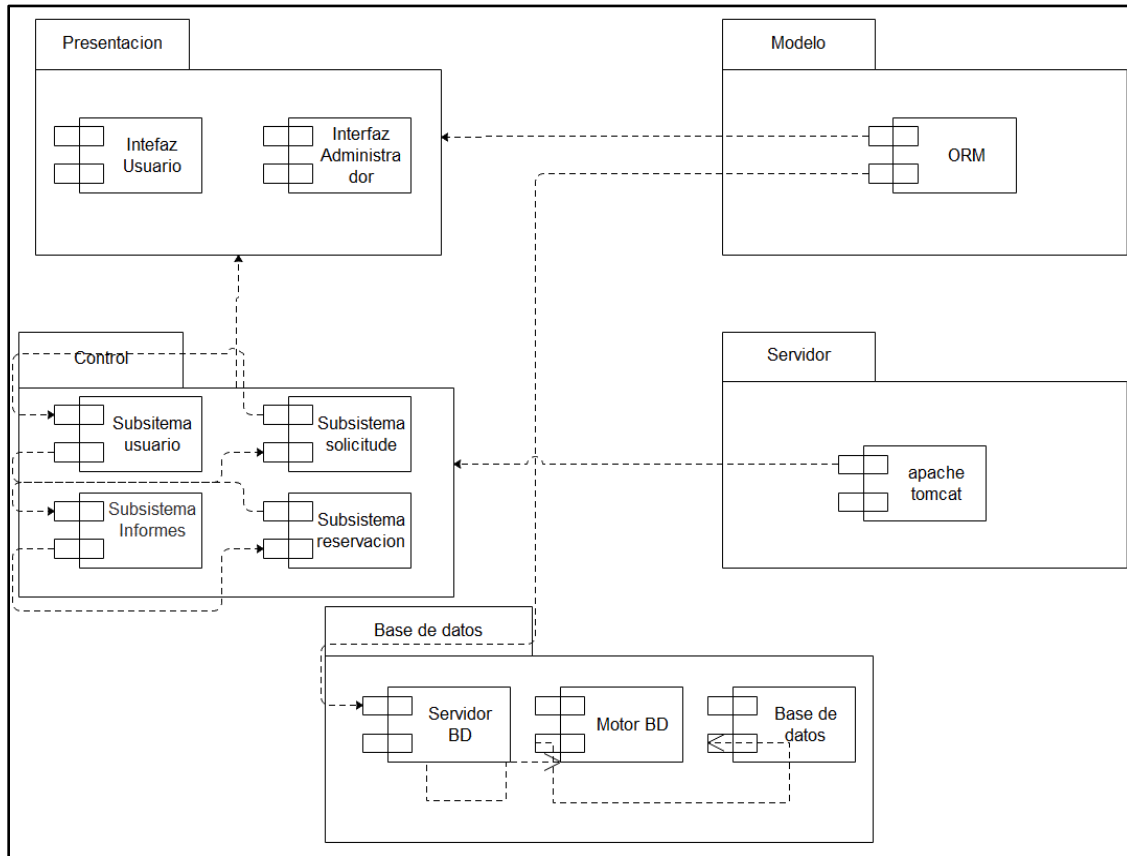


Figura 43. Diagrama Componentes Muestra Vista de Desarrollo del Sistema.

Fuente: Elaboración propia.

Ejemplo del diagrama de paquetes de la Vista de Desarrollo

En la Figura 44, “Diagrama de paquetes que muestra la vista de desarrollo del Sistema.”, donde se muestran los paquetes o agrupaciones lógicas y sus dependencias del futuro sistema informático.

CAPÍTULO IV

PROPUESTA DE SOLUCION

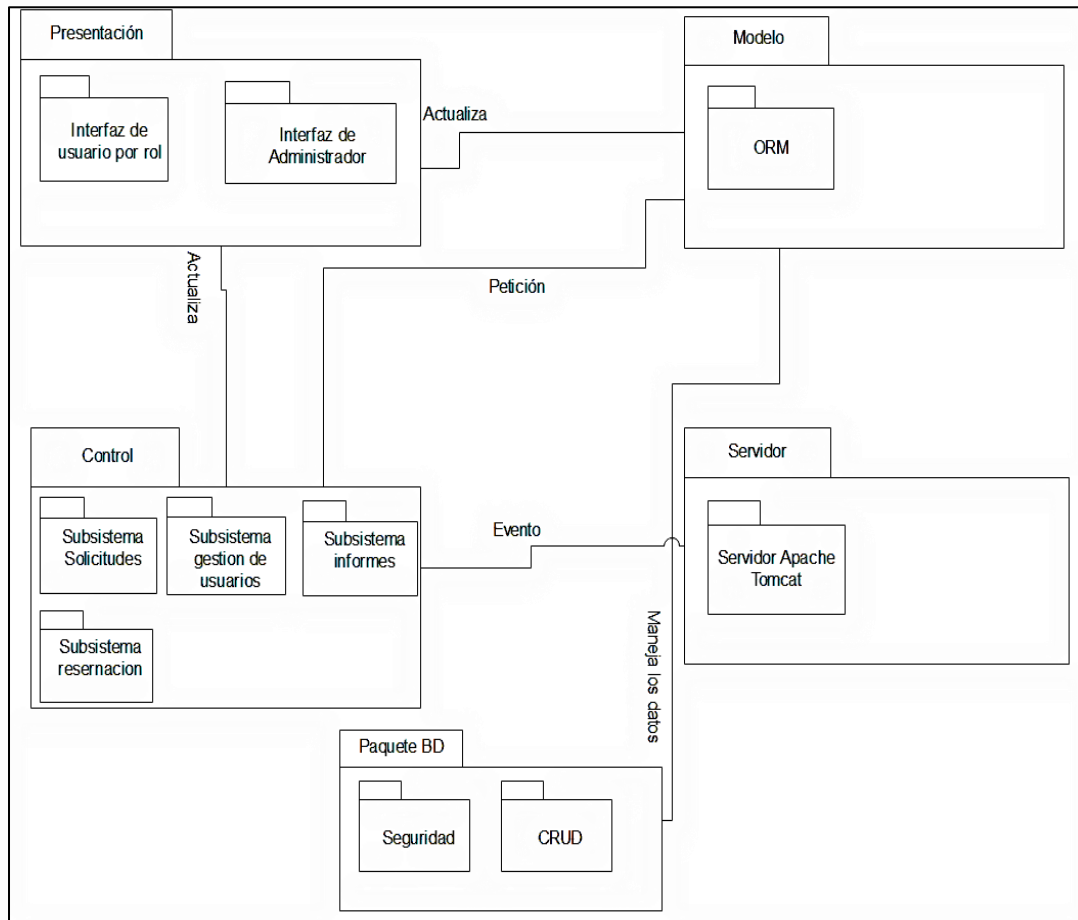


Figura 44. Diagrama de Paquetes que muestra la vista de Desarrollo del Sistema.

Fuente: Elaboración propia.

Arquitectura Física (Mapeo del software al hardware)

La definición que establece Kruchten deja muy claro que esta vista está basada en hardware y software. La vista física describe la asignación del software al hardware y refleja su aspecto distribuido. (Kruchten, P. B., (1995).

Además, los componentes y las conexiones físicas a diversos servicios con el cual se crea la perspectiva del proyecto.

El software se ejecuta sobre una red de computadores o nodos de procesamiento (o solamente nodos). Los diversos elementos identificados como (redes, procesos, tareas y objetos) deben ser mapeados sobre los variados nodos. Por lo tanto, el mapeo del software en los nodos requiere ser muy flexible y tener un impacto mínimo sobre el código fuente en sí. (Kruchten, P. B., (1995).

Ejemplo del diagrama de paquetes de la vista física.

En la *Figura 45*, “Diagrama de paquetes que muestra la vista física del Sistema”, esta presentación modela los procesos entre los distintos componentes del sistema, además está compuesta por dos paquetes que son: la interfaz de red y la Workstation.

El paquete de interfaz de red está compuesto por firewall, server, red local y el paquete Workstation, el cual está integrado por el equipo y el hardware del equipo.

CAPÍTULO IV

PROPUESTA DE SOLUCION

La interfaz de red es la que comunica la red local externa en este caso WAN, para que esta red sea segura es necesario tener la seguridad con Firewall, una vez que la red es segura se tiene acceso a los servidores ya sea desde un datacenter entre otros.

La Workstation está compuesta por el equipo que administra el sistema, este equipo está compuesto por servidores, switch y los componentes que son necesarios para su administración.

Al tener una interfaz de red segura esto permitiría que se tenga un acceso autorizado al equipo o servidor donde se encuentre el sistema. Esto permite que el sistema pueda tener acceso por medio de un dominio seguro en donde los ataques al mismo no tengan consecuencias y los usuarios puedan utilizarlo.

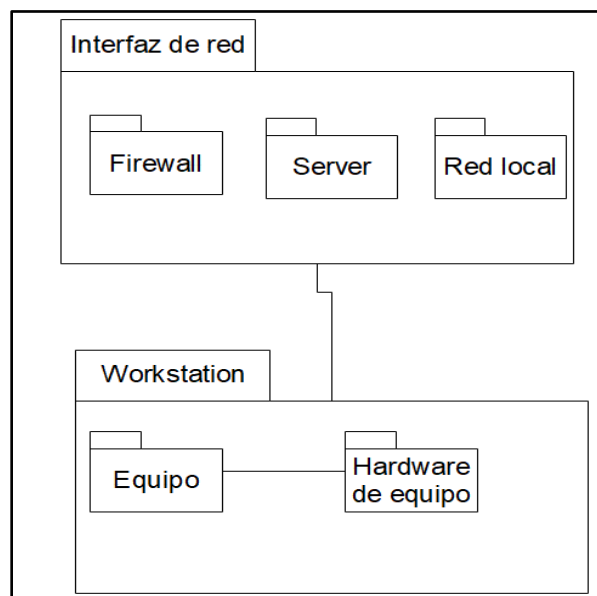


Figura 45. Diagrama de Paquetes que Muestra la Vista Física del Sistema.

Fuente: Elaboración propia.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Ejemplo del diagrama de componentes de la vista física. En la *Figura 46*, “Diagrama de componentes que muestra la vista física del Sistema”, este diagrama está realizado con el fin de modelar la vista física es decir los equipamientos del hardware.

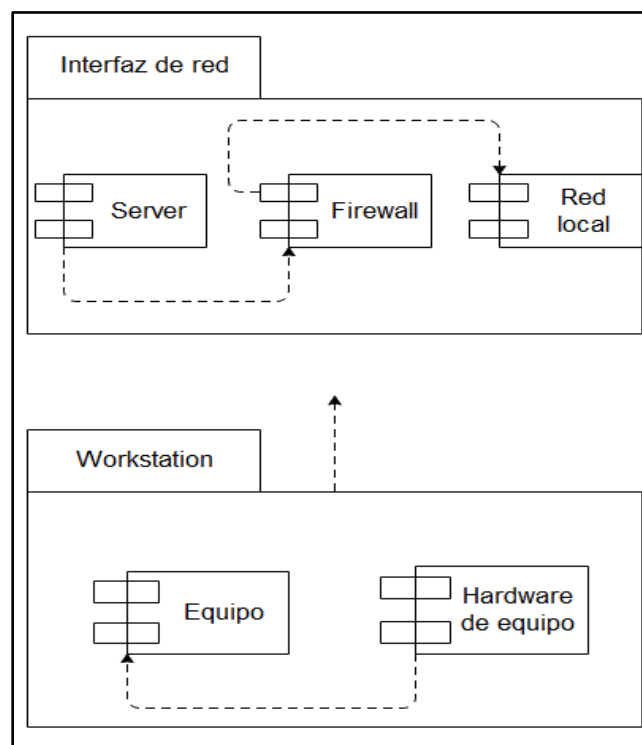


Figura 46. Diagrama de Componentes que Muestra la Vista Física del Sistema.

Fuente: Elaboración propia.

4.2.10 Arquitectura General propuesta

En la *Figura 47*, “Arquitectura General Propuesta” muestra la estructura y la forma que debe de operar, la arquitectura de software propuesta. Primeramente, el usuario ingresa a la interfaz gráfica, este se autentica. Esta solicitud pasa por un mecanismo de verificación de seguridad, seguidamente al servidor de autenticación. Al realizar todo este proceso si la autenticación es correcta, le devolverá a la interfaz de usuario, la página principal del sistema, donde procederá a realizar las solicitudes de reserva que requiera.

Posteriormente, el usuario, se dirige al apartado de solicitudes, para realizar una solicitud de reserva o permisos de ingreso a parqueos institucionales y otras áreas de interés. Realizado esta acción, este automáticamente se dirige a un controlador de esta solicitud, este la procesa y devuelve los datos a la aplicación.

Seguidamente, el servidor de la base de datos se encarga de administrar esta solicitud y la base de datos comisiona las funciones de guardar y registrar toda la información ingresada, y ser consultada cuando sea necesario por su administrador o por un modelo cliente-servidor.

Finalmente, este proceso de solicitud se devuelve al servidor de bases de datos y esta muestra en la interfaz de usuario, el estado de su solicitud.

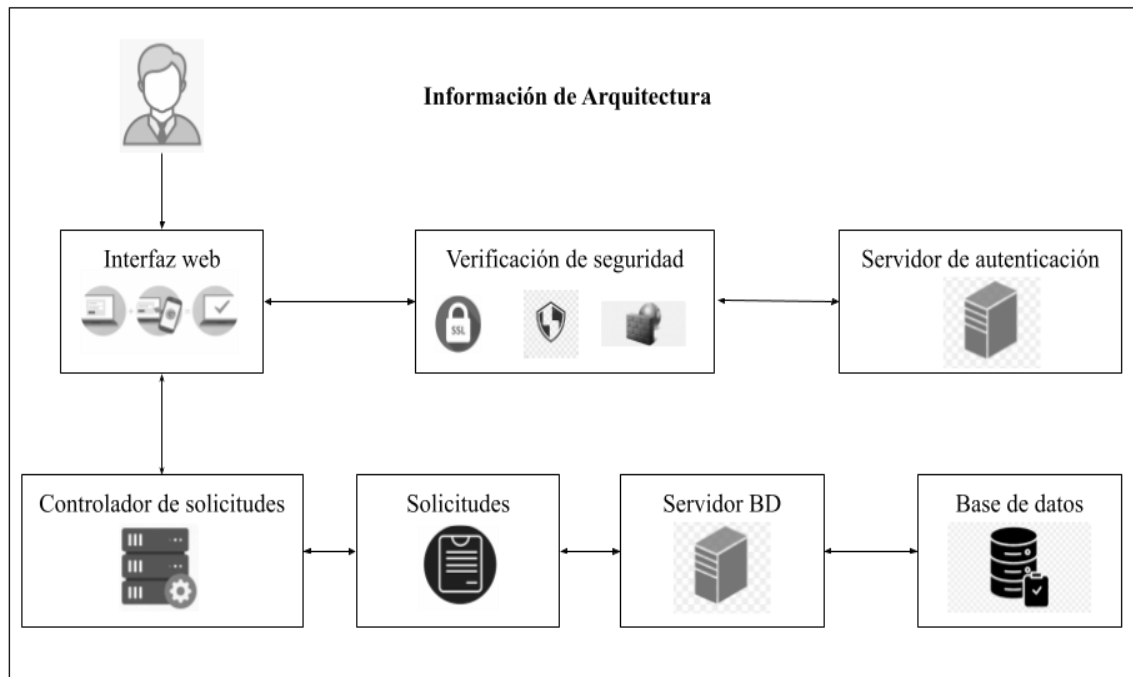


Figura 47. Arquitectura General propuesta.

Fuente: Elaboración propia

4.2.11 Método de Usabilidad

La presente sección muestra la definición y la implementación del método de Usabilidad. La Usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado, y ser atractivo para el usuario, en condiciones específicas de uso. ISO/IEC9126.

La Interfaz de Usuario (IU) es la puerta hacia la funcionalidad del sistema, es necesario diseñarla para que sea usable para los usuarios. Hay que tomar en cuenta que un mal diseño limita

CAPÍTULO IV

PROPUESTA DE SOLUCION

su uso. Un diseño de software puede ser aceptable que sea fácil de utilizar, eficiente y que tenga facilidad de aprendizaje por parte de los usuarios.

En la *Figura 48*, “Esquema sobre el Método de Usabilidad de la ISO/IEC 9241-11”; donde se muestra que el usuario realiza las tareas específicas en escenarios específicos con total efectividad.

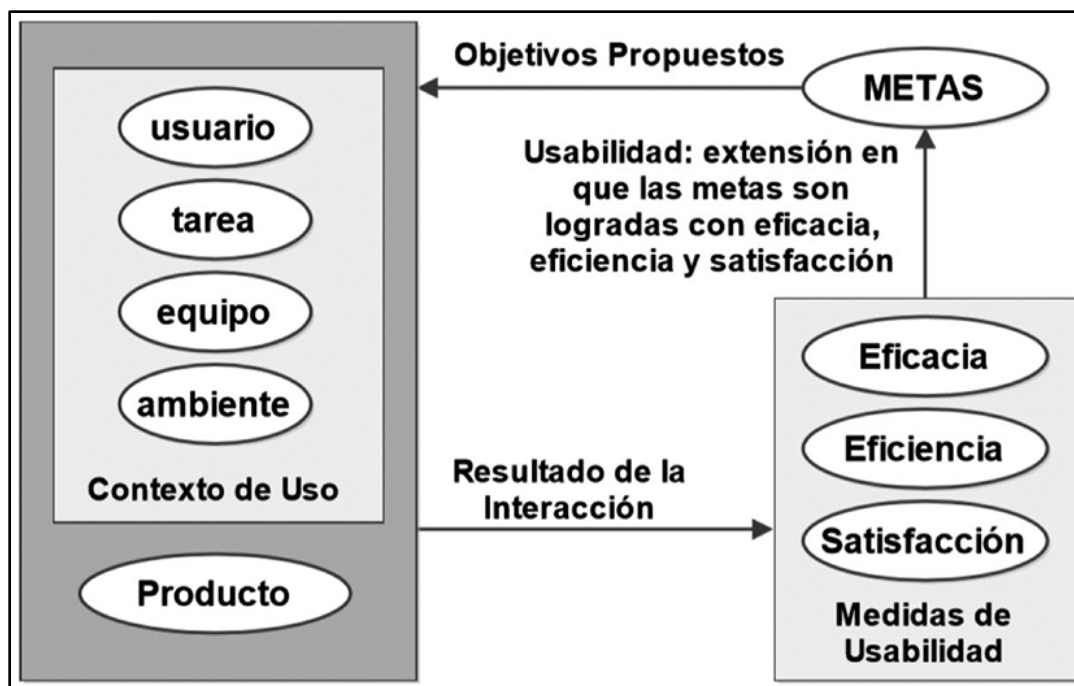


Figura 48. Esquema Sobre el Método de Usabilidad de la ISO/IEC 9241-11.

Nota. Usabilidad es la eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico. Fuente: ISO/IEC 9241

CAPÍTULO IV

PROPUESTA DE SOLUCION

La Tabla 11, “Las diez heurísticas de Nielsen”, dividido en tres columnas, el nombre de la heurística, la definición y su implementación en el futuro sistema.

La primera heurística llamada visibilidad del estado del sistema, tiene como finalidad que el usuario debe estar informado de lo que está sucediendo durante la interacción del usuario-sistema, por medio de mensajes o habilitando botones.

Últimamente en la tercera columna se narra la implementación o como esta heurística debe reflejarse; ejemplo, el botón de carga de un archivo debe estar habilitado, hasta que se haga clic en él; otro ejemplo sería que se debe mostrar un mensaje de respuesta cuando se realiza una acción, esto por mencionar algunos.

Tabla 11. Las diez heurísticas de Nielsen

Heurística	Definición	Implementación
1)-Visibilidad del estado del sistema.	Esta heurística se refiere a que el sistema siempre debe mantener a los usuarios informados sobre lo que está sucediendo, a través de comentarios apropiados	En el sistema a implementar se deberán programar mensajes de advertencia de procesos en marcha, además de indicaciones claras de la acción que el sistema está realizando en algún momento. Algunos

CAPÍTULO IV

PROPUESTA DE SOLUCION

dentro de un tiempo ejemplos adicionales de funcionalidades que cumplen con esta heurística son:

- El botón de carga de un archivo está habilitado, hasta que se hace clic. Luego se reemplaza con un indicador de progreso hasta que el archivo haya terminado de cargarse.
- El mensaje de progreso y el indicador se muestran mientras se carga la aplicación al inicio.
- Se muestra un mensaje de respuesta cuando se realiza una acción.
- La seguridad de la contraseña se muestra cuando se ingresa la contraseña.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Heurística	Definición	Implementación
2)-Coincidencia entre el sistema y el mundo real.	Esta heurística describe que el sistema debe hablar el idioma de los diferentes usuarios, utilizando palabras, frases y conceptos familiares para el usuario, en lugar de términos técnicos referentes al sistema.	Se debe implementar un lenguaje coloquial y natural, donde sea entendible por cualquier tipo de usuario (funcionarios, estudiantes, personas externas). Además de utilizar los términos actuales, con los que se realizan estos procesos, (ejemplo reserva o permiso de Ingresos a parqueos institucionales y otras áreas); con el fin de que los usuarios se sientan familiarizados con ello.

CAPÍTULO IV

PROPUESTA DE SOLUCION

- 3)- Control del usuario y libertad Esta heurística se refiere a la capacidad que tienen los usuarios de elegir tener una salida de emergencia cuando seleccionan algunas funciones del sistema por error. Esta “salida de emergencia” debe estar claramente señalada sin tener que pasar por un diálogo.
- Se debe habilitar la funcionalidad de salida de cualquier programa, módulo, ventana, entre otros, de forma abrupta en cualquier momento. También se deben habilitar las funcionalidades de deshacer y rehacer. Algunos ejemplos adicionales de funcionalidades que cumplen con esta heurística son:
- La función de búsqueda es fácil de abrir, ingresar información, ejecutar o cancelar.
 - Se indica claramente dónde está la persona y hacia dónde puede ir mostrando la selección en cada menú.
 - Los botones Deshacer y Rehacer están disponibles en la barra de herramientas, y también se puede

CAPÍTULO IV

PROPUESTA DE SOLUCION

acceder con los atajos de teclado estándar.

- 4)- Consistencia y Esta heurística se refiere que Se debe de basar en los estándares de la los usuarios no deben institución, siguiendo los esquemas e preguntarse si las diferentes interfaces de otros sistemas. Los palabras, las situaciones o usuarios principales que realizan estas acciones significan lo mismo. solicitudes son funcionarios de esta Universidad, por lo tanto, este sistema debe de tener una consistencia con otros sistemas. (como por ejemplo el sistema de Solicitudes de Vacaciones). Utilizando las mismas barras de herramientas, de estilo con las mismas opciones de menú principal: ingresar, modificar, eliminar entre otros.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Heurística	Definición	Implementación
5)- Prevención de errores.	Esta heurística se refiere a que existan mensajes de error de diseño cuidadoso, sería importante evitar que ocurra un problema en primer lugar.	En el sistema a desarrollar se debe deshabilitar el botón de actualización después de hacer clic, por lo que la persona no puede realizar la acción de actualizar dos veces por accidente.
	Se debe eliminar las condiciones propensas a errores o verificarlas y presentarlas a los usuarios una opción de confirmación antes de comprometerse con la acción.	También se debe de tomar en cuenta los botones de prevención, por ejemplo, el botón de aceptar que se marque a la izquierda de la pantalla, y más grande, que el botón de cancelar, además que el botón de cancelar se puede mostrar como un enlace.
		Otra recomendación para el futuro sistema es habilitar las funciones de recomendación automática, esto

CAPÍTULO IV

PROPUESTA DE SOLUCION

ayuda a reducir las faltas de ortografía.

Por ejemplo, añadir los nombres de parqueos institucionales, nombres de gimnasios, teatros y otras áreas de interés.

Además de describir breve y con claridad los botones y barras de herramientas, así como los cuadros de textos para ubicar al usuario.

Heurística	Definición	Implementación
6)- Reconocimiento en lugar de recordar	Esta heurística señala que se debe de minimizar la carga de memoria del usuario estableciendo visibilidad a los objetos, las acciones y las opciones. El usuario no debería tener que recordar	El sistema debe estar familiarizado con el usuario de tal forma que el usuario reconozca cuando está navegando en el sitio web. Por ejemplo, se deben diseñar las interfaces de forma similar a las demás interfaces de otros sistemas, ya en operación en la

CAPÍTULO IV

PROPUESTA DE SOLUCION

información de una parte de institución, esto con el fin de facilitar el una caja de texto a otra. Las reconocimiento a los usuarios. instrucciones de uso del sistema deben ser visibles o fácilmente recuperables siempre que sea apropiado.

7)-Flexibilidad y eficiencia de uso. La séptima heurística dice que El sistema deberá de ser flexible y el sistema puede atender tanto eficiente con el usuario o sea que debe a usuarios inexpertos como ser entendible por cualquier perfil que experimentados. Así como tiene la institución. permitir a los usuarios Además, sería deseable que el usuario personalizar las acciones tenga a mano la solicitud realizada, por frecuentes. lo que se debe habilitar una opción de imprimir la solicitud de reserva, enviar por correo, enviar como archivo por WhatsApp y guardar en la nube y localmente.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Heurística	Definición	Implementación
8)- Diseño estético y minimalista.	Esta heurística nos señala que los diálogos no deben contener información irrelevante o rara vez necesaria. Además, el diseño visual debe respetar los principios de contraste, repetición, alineación y proximidad.	El sistema no deberá tener elementos “de más” para el manejo de los usuarios. El contenido debe ser sutil y la interfaz debe ser fresca, simple y permitir al usuario reconocer solo lo que necesita.
9)- Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de errores.	En esta heurística los mensajes de error deben expresarse en lenguaje sencillo (sin códigos), indicar con precisión el problema y sugerir constructivamente una solución.	El futuro sistema debe ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores. Por ejemplo, cuando un usuario esté realizando algún proceso en el sistema (ingresos, modificaciones de solicitudes de reserva, etc.), en cada ventana se debe proporcionar comentarios inmediatos sobre el resultado de las

CAPÍTULO IV

PROPUESTA DE SOLUCION

acciones o con instrucciones específicas (por ejemplo: “registro actualizado”, “Presione el botón de aceptar para proceder”, entre otros

- 10)- Ayuda y documentación. En la última heurística suele ser necesario proporcionar el usuario, que sean breves, detallados, ayuda y documentación. y fáciles de entender, con ilustraciones Cualquier información de este y descripciones de las posibles tipo debería ser fácil de inquietudes de los usuarios. Estos buscar, centrada en la tarea del manuales también deben ser accesibles usuario, enumerar los pasos en la forma de ayuda en línea para concretos que se deben llevar cualquier tipo de dispositivo a cabo y no ser demasiado (Computadoras o dispositivo móvil). grande.

Fuente: Elaboración propia basada en (Nielsen, 1994).

4.2.12 Vista de Datos (E-R)

Este capítulo incluye la representación de la vista de datos mediante la representación de entidades y relaciones que componen el sistema de software.

En la *Figura 49*, “Diagrama de Entidad-Relación de la Base de Datos” se incluyen las clases que conformarán el sistema futuro, así como sus relaciones.

La principal función del Diagrama de Entidad-Relación de la Base de Datos permite modelar y representar el sistema de software futuro, mediante la percepción de un mundo real, asociando las funciones y requerimientos (Osorio Rivera, 2008).

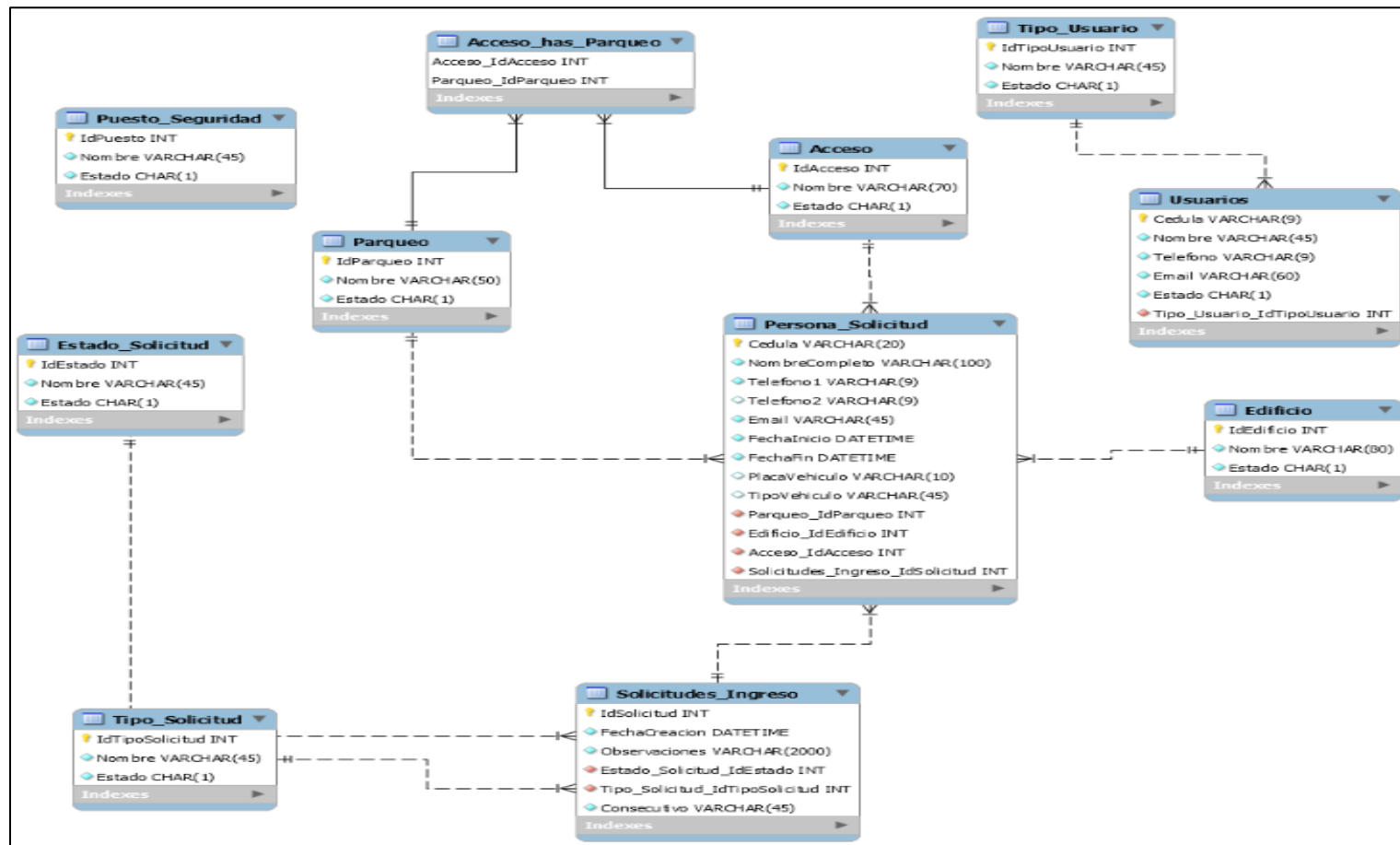


Figura 49. Diagrama de Entidad-Relación de la Base de Datos
Fuente: Elaboración propia.

4.2.13 Especificaciones preliminares de requerimientos

En esta sección se muestran las especificaciones generales de los requerimientos del sistema.

En la *Figura 50*, “Seis Dimensiones de Requerimientos” se muestra las principales seis categorías de deben ser abordadas.

Las especificaciones y obtención de los requisitos; es el proceso de determinar lo que realmente los usuarios finales necesitan que haga el futuro sistema propuesto, con la claridad de documentar dicha información. Los requisitos de un proyecto son la base de una solución que será diseñada e implementada (Tsui , Karam , & Bernal, 2016).

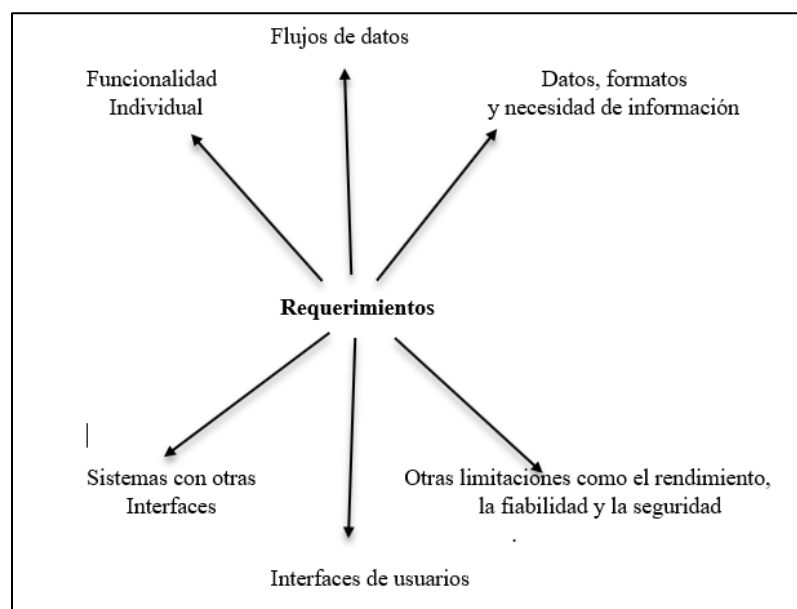


Figura 50. Seis Dimensiones de los Requerimientos.

Fuente: (Tsui , Karam , & Bernal, 2016).

En la *Figura 51*, “Influencias Mutuas entre la Arquitectura y su Entorno, se presenta los requerimientos como una influencia determinante en la concepción de la arquitectura de software del sistema.

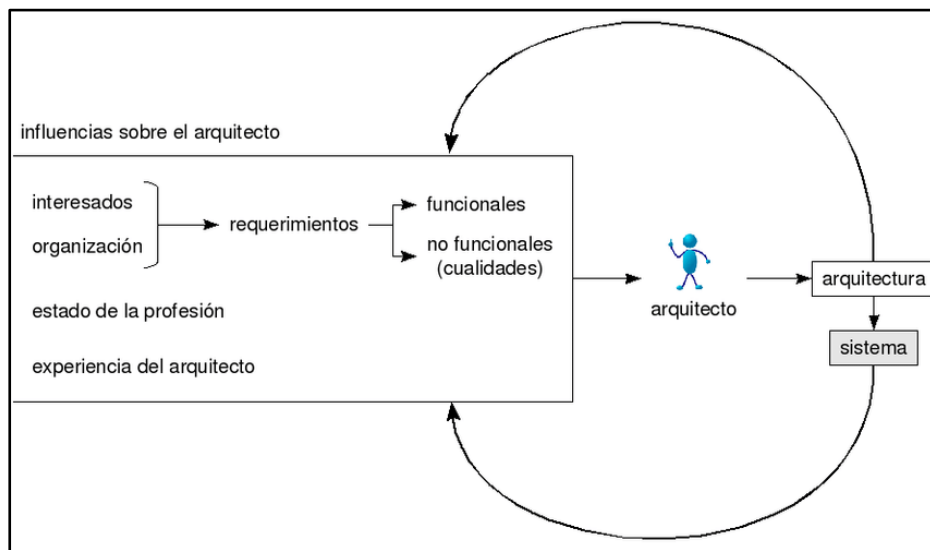


Figura 51. Influencias Mutuas entre la Arquitectura y su Entorno.

Fuente: (Tsui , Karam , & Bernal, 2016).

4.2.13.1 Requerimientos Funcionales

Seguidamente se muestran los requerimientos, que se dividen en dos grandes clases; Requerimientos funcionales y No funcionales.

CAPÍTULO IV

PROPUESTA DE SOLUCION

La Tabla 12 “Tabla de los requerimientos funcionales” muestra en la primera columna el requerimiento funcional y en la segunda columna las declaraciones de los servicios o comportamiento que este brindará.

Tabla 12. Tabla de los Requerimientos Funcionales

Requerimiento	Comportamiento o servicio
Ingresar usuario	Los usuarios que cuenten con el id otorgado por la Universidad Nacional, donde lo hacen acreedores en dicha institución, solamente ellos podrán acceder al sistema.
Cancelar o posponer reserva realizada	Los usuarios pueden cancelar o posponer solicitud de reserva o permiso de ingreso a parqueos institucionales y otras áreas de interés cuando este lo amerite.
Eliminar reserva realizada	Los usuarios pueden eliminar una solicitud de reserva o permiso de ingreso a parqueos institucionales y otras áreas de interés, cuando ya no lo requieran.
Aprobar reservas	Solamente los personeros de la SSI-UNA podrán aprobar las reservas de permisos de ingresos y vehículos a los parqueos.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Mostrar Reservas realizada	Solamente los usuarios de la SSI-UNA podrán acceder a la lista de reservas realizadas.
Mostrar reservas aprobadas	Los usuarios de la Unidad de Monitoreo, los de la Sección de Seguridad y el usuario que realizó la solicitud de reserva, podrán ver el estado de la solicitud.
Buscar reserva realizada	Los usuarios de la Sección de Seguridad de podrán únicamente buscar reservas realizadas por el usuario por medio de la fecha o nombre de usuario.
Buscar reserva aprobadas por unidad	Los usuarios de la Sección de Seguridad podrán únicamente buscar reservas aprobadas por medio del nombre de la Unidad o edificio.

Fuente: Elaboración propia.

4.2.13.2 Requerimientos No funcionales

La Tabla 13, “Tabla de los requerimientos No funcionales” presenta los requerimientos No Funcionales, también llamados cualidades del sistema). la primera columna se presenta el requerimiento No Funcional y en la segunda columna las características del software.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Tabla 13. Tabla de los Requerimientos No funcionales

Requerimiento No Funcional	Características
Acceso desde un cliente móvil	Los requisitos no funcionales que se requieren cumplir son: La aplicación web debe ser accesible desde celulares, tabletas y cualquier dispositivo móvil. Ajustando la visualización a la resolución del dispositivo móvil.
Seguro y confiable	El sistema debe brindarle al usuario seguridad y confianza en el manejo de sus datos.
Basados en estándares de diseño de la institución	El sistema debe estar diseñado para que el usuario reconozca como parte de un sitio de esta institución.
Alta disponibilidad	El sistema debe estar disponible 24/7, para realizar solicitudes de reservas o permisos de ingresos a parqueos institucionales y otras áreas de interés.
Mantenibilidad	En el momento que se necesite agregar un módulo, ventana o algún componente este debe realizarse de manera sencilla sin tener ningún inconveniente.

Fuente: Elaboración propia.

4.2.14 Calidad y rendimiento de procesos

Los indicadores de rendimiento y calidad de los procesos permiten tener un mayor control de las actividades de la empresa. En el caso de BMP permite darles un rastreo a los procesos en donde el administrador del proceso pueda conocer el progreso de este y así conocer cuáles aspectos se pueden mejorar.

Para definir y documentar estos indicadores es necesario lo siguiente.

- Describir el objetivo específico que debe cumplir el proceso.
- Definir una fórmula o la lógica a utilizar para medir este indicador.
- Establecer la unidad de medición que tendrá el proceso.
- Establecer el periodo en que se va a medir su indicador. Se deben tener fechas concretas para su medición.
- El indicador debe tener un responsable asociado para su respectiva medición.
- Finalmente indicar el proyecto al que se está analizando el proceso.

En la Tabla 14, “Indicadores de Rendimiento y Calidad” podemos observar los Indicadores de rendimiento y calidad. Donde primeramente se define un indicador de rendimiento y calidad, luego se especifica el objetivo, seguidamente se especifica la fórmula de dicho indicador, luego se indica el tiempo el lugar del proyecto y los responsables.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Tabla 14. Indicadores de Rendimiento y Calidad

Definición	Objetivo	Fórmula	Periodicidad	Proyecto	Responsable
Proceso de ingreso de vehículos	50 vehículos	Cantidad de vehículos semanal	Mensual día 30 de cada mes	Sistema de seguridad	Técnico
Aprobación de reserva	Aprobar las solicitudes emitidas	Solicitudes diarias	Solicitudes semanalmente	Sistema de seguridad	Usuario Administrador
Solicitudes sin aprobar	Verificación de espacios disponibles para la aprobación de reservas	Solicitudes emitidas por día	Solicitudes semanalmente	Sistema de seguridad	Usuario Administrador
Acceso al sistema por medio de dispositivo móvil	Evaluar las solicitudes emitidas por medio de dispositivo móvil	Cantidad de solicitudes emitidas por mes	Trimestral	Sistema de seguridad	Técnico
Revisión de bitácoras	Verificar bitácoras para tener control del proceso utilizado por parte de los usuarios del sistema	Transacciones emitidas por los usuarios del sistema	Mensual día 30 de cada mes	Sistema de seguridad	Técnico

Fuente: Elaboración propia.

4.3 Validación de la Propuesta

En este apartado se valida la propuesta desarrollada, esta validación se realizó por medio de los principios del Método de Análisis de Compensación de Arquitecturas, (ATAM). (Kazman, Klein, & Clements, 2000).

4.3.1 Método de Análisis de Compensación de Arquitecturas, ATAM

El Architecture Tradeoff Analysis Method (ATAM) es una metodología para evaluar Arquitecturas de Software basada en los atributos de calidad especificados para el sistema, desarrollada por el Software Engineering Institute (SEI).

Para este proyecto se tomaron las cuatro fases del Método de Análisis de Compensación de Arquitecturas y se combinaron algunos pasos por lo que los pasos ejecutados fueron siete, como se muestra en la Tabla 15, “Fases y pasos ejecutadas ATAM”.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Tabla 15. Fases y Pasos Ejecutadas ATAM

Fases	Pasos
Presentación	1-Presentar el ATAM 2-Presentar las metas de Negocio 3-Presentar la arquitectura
Investigación y Análisis	4-Identificación de los enfoques arquitectónicos, por medio de un formulario 5- Generación y análisis de los atributos del árbol de calidad
Pruebas	6- Priorización de escenarios y analizar atributos de calidad: Revisión y votación de estos. (Likert)
Informes	7-Presentación y correlación de resultados obtenidos

Fuente: Elaboración propia.

Una de las características de ATAM, es que debe de ejecutarse con arquitectos de software e informáticos, con diferentes años de experiencia en el campo. En la siguiente lista se refleja a 7 involucrados en esta evaluación.

Tabla 16. Involucrados en la Evaluación del ATAM

Nombre del arquitecto o Informático de Software	Años de experiencias como arquitectos e Informáticos de Software
Gonzáles María Fernanda	Más de 5 años de experiencia
Loaiza Mora Gabriela	Mas de 10 años de experiencia
López Jonathan	Más de 5 años de experiencia
Paniagua Soto Pablo Roberto	Más de 10 años de experiencia
Peralta Varela María José	Más de 5 años de experiencia
Sánchez Diego	Más de 5 años de experiencia
Vargas José Pablo	Más de 5 años de experiencia

Fuente: Elaboración propia.

El desarrollo tuvo una duración en tiempo de tres meses aproximadamente. La ejecución de ATAM, se realizó en tres bloques, se coordinó virtualmente por medio de correos electrónicos y por medio del sitio, donde se encontraba la información y los pasos requeridos. Estos bloques fueron divididos de la siguiente manera; primeramente, se ejecutaron los pasos 1,2,3 y 4, (consulta el Anexo 7 para ver correo). Luego se prosiguió con los pasos 5 y 6 (Consulta el Anexo 8 para ver correo) y finalmente el paso 7.

CAPÍTULO IV

PROPUESTA DE SOLUCION

La información contenida en la Tabla 17, “Pasos de ATAM Asociados con las Partes Interesadas”, muestra el grupo de involucrados que le correspondió ejecutar cada paso o actividad.

Tabla 17. Pasos de ATAM Asociados con las Partes Interesadas

Paso	Actividad	Grupo o parte interesada
1	Presentar el ATAM	Desarrolladores de proyecto
2	Presentar las metas de Negocio	Desarrolladores de proyecto/representante modelo de negocio.
3	Presentar la arquitectura.	Desarrolladores de proyecto
4	Identificación de los enfoques arquitectónicos, por medio de un formulario.	Grupo de Arquitectos e informáticos
5	Generación y análisis de los atributos del árbol de calidad.	Grupo de Arquitectos e informáticos
6	Priorización de escenarios y analizar atributos de calidad: Revisión y votación de estos.	Grupo de Arquitectos e informáticos

7	Presentación y correlación de resultados obtenidos	Todos los involucrados
---	--	------------------------

Fuente: Elaboración Propia.

4.3.2 Ejecución de ATAM, Fase de Presentación

Fase de Presentación

1. Presentar el ATAM: Este paso se presentó el Método de Análisis de Compensación de Arquitectura por medio de una video-presentación, con el fin de que los grupos involucrados, conocieran el método, el proceso y el objetivo de su ejecución. (Consulta el Anexo 9 para ver captura de pantalla).

2. Presentación del Modelo de Negocio: El segundo paso de este método corresponde a la presentación del proceso de negocio de la Sección de Seguridad Institucional, con el fin de conocer el flujo de trabajo. (Consulta el Anexo 10 para ver captura de pantalla).

3. Presentación de la Arquitectura: En este paso se presentó por medio de una video-presentación el Documento de Arquitectura de Software desarrollado. La finalidad de este paso es presentar el prototipo de Arquitectura de Software desarrollado. Para el grupo de evaluadores de este proyecto, se le realizó a parte de la video presentación, se le compartió el Documento de Arquitectura de Software, con el fin de estudiarlo, evaluarlo y mostrar sus evaluaciones. Consulta el Anexo 11 para ver captura de pantalla).

4.3.3 Ejecución de ATAM, Fase de Investigación y Análisis

Fase de Investigación y Análisis

Esta fase corresponde a la identificación de los enfoques arquitectónicos, por medio de una encuesta analítica en línea, con el fin de obtener resultados analíticos por los arquitectos evaluadores.

4. Identificación de los enfoques arquitectónicos, por medio de un formulario: En este paso se realizó un formulario compuesto de preguntas. Estas preguntas son relacionadas con los diferentes atributos de calidad, (Kazman et al, 2000) con el fin de poder evaluarlos y compararlos con la arquitectura propuesta. (En el anexo 12, se muestra captura de pantalla de este formulario).

Los resultados obtenidos en este paso sirvieron para poder priorizar escenarios y así generar el Árbol de Utilidad.

Cabe destacar que en este paso solamente se identifican y se capturan los atributos de calidad y enfoques arquitectónicos, por medio del equipo de arquitectos. Estos atributos son analizados en el paso siguiente.

Los enfoques arquitectónicos definen la estructura del sistema y describen las formas en que el sistema puede crecer, responder a los cambios, resistir ataques e integrarse con otros sistemas.

CAPÍTULO IV

PROPUESTA DE SOLUCION

De las repuestas realizadas por los 8 arquitectos de software, podemos apreciar que hubo similitud con el atributo de calidad, rendimiento y modificabilidad.

Dentro del formulario se logra apreciar que los arquitectos tienen una visión clara de los alcances del sistema, tal es el caso de que con la experiencia con las que cuenta los arquitectos involucrados en esta evaluación.

Además, con los conceptos brindados se pueden definir los principales atributos de calidad con la que debe contar una buena arquitectura.

A continuación, se relata textualmente algunas de estas percepciones:

“Las fallas en mi experiencia se catalogan en dos: Fallas funcionales (no cumplen las expectativas del cliente) y Fallas técnicas. Para la primera se mitiga haciendo un proceso detallado de pruebas con el cliente y para la segunda ya existen herramientas que te permiten validar fallas técnicas”.

Dentro de estas percepciones también se encontraron recomendaciones y opiniones que contribuyeron a la construcción del Árbol de Utilidad (UT), el cual es desarrollado en la siguiente fase.

“Los componentes críticos deben estar dentro de la red militarizada. Uso de certificados (https) y mecanismos de token. Ahora son elementos que conllevan un costo, su uso es recomendable, pero se debe evaluar el riesgo real para validar si realmente se necesitan”.

Otro arquitecto relató lo siguiente;

“Cómo arquitecto en sistemas considero que, si se puede agregar nuevas funcionalidades, porque un buen producto siempre debe de actualizarse frecuentemente, para un óptimo desempeño (...) pero una funcionalidad que se le podría anexar es el compartimiento de una nueva red social o el funcionamiento por medio de un robot en su momento...”

En la siguiente respuesta podemos identificar como el atributo de modificabilidad es un aspecto importante dentro del mantenimiento de una arquitectura, el siguiente relato de este involucrado en esta evaluación, narra lo siguiente:

“Al realizar un cambio de una funcionalidad a un sistema las consecuencias que pueden surgir es perjudicar la funcionalidad de otro módulo, por lo que se debe ser muy quisquilloso en no enlazar ciertas funcionalidades que tengan relaciones complejas que perjudiquen la agregación de otras funcionalidades”.

Otro aspecto evaluado por los arquitectos fue la estimación de tiempo, en la implementación de una funcionalidad. Ellos recalcan que es difícil de estimar la duración, lo justifican con lo siguiente;

“Considero que los factores que influyen en la estimación de tiempo en el momento de agregar una funcionalidad es la falta de conocimiento o un diseño inadecuado”.

Acerca del término de redundancia muchos arquitectos definen que este término es una práctica que debe ser aplicada de forma idónea, para obtener resultados satisfactorios. Esto lo justifica la siguiente respuesta.

“Considero que la redundancia es una buena práctica, donde se debe estudiar muy bien donde, para que estos elementos no influyan negativamente. Y si no se hace así lastimosamente no sería una excelente opción”.

4.3.4 Ejecución de ATAM, Fase de Pruebas

Fase de Pruebas

Esta fase corresponde a la fase de pruebas de los atributos de calidad, estos se realizan por medio del concepto del Árbol de Utilidad, luego de ejecutado el UT, se prosigue a analizar los atributos de calidad, por medio de una plantilla llamada plantilla para la documentación y análisis del enfoque arquitectónico.

5. Generación y Análisis de los atributos del árbol de calidad: Esta sección contiene la descripción de los principales conceptos como; atributos de calidad, escenarios y árbol de utilidades. Estos términos han sido ejecutados y desarrollados basándose en el paso 5 de dicho método. (Ver Anexo 13).

- **Atributos de Calidad.** Los atributos de calidad son puntos que describen el comportamiento esperado de un sistema dentro del entorno para el que fue proyectado. Un atributo de

calidad es una característica no funcional de un componente del sistema. (Barbacci, y otros, 2002)

- Escenarios y árboles de utilidades. Los escenarios son una herramienta simple y poderosa para describir requisitos. Es una declaración breve que describe una interacción de una de las partes interesadas con el sistema (Richards, 2015).

El propósito del árbol de utilidad²³ es priorizar los atributos de calidad identificando riesgos, no riesgos, puntos de sensibilidad y puntos de concesión para luego evaluarlos con la arquitectura de Software frente a los requisitos. (R. Kazman, M. Klein, and P. Clements, 2000). Para la realización del árbol de utilidad, (UT) y el análisis arquitectónico se deben de seleccionar ciertos atributos de calidad.

Para la evaluación de esta arquitectura, su elección se basó en los siguientes aspectos. Primeramente, se analizaron los datos obtenidos de la encuesta en línea a través del sitio Web (paso 4), que participaron los arquitectos de software nombrados anteriormente.

²³ Un UT es un esquema que presenta los atributos de calidad de un sistema de software, estableciendo escenarios que especifican el nivel de prioridad de cada uno.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Seguidamente los autores Carrascoso P, Chaviano, & Céspedes, (2009), los atributos más relevantes para evaluar una arquitectura de software son (modificabilidad, desempeño, disponibilidad y seguridad).

Por último; sin dejar de ser el menos importante Kazman, Bass, & Klein, establecen la siguiente clasificación de los atributos de calidad para la evaluación de arquitecturas, (2006).

- ❖ **Los observables en tiempo de ejecución:** aquellos atributos que se determinan del comportamiento del sistema en tiempo de ejecución (por ejemplo, disponibilidad, performance, seguridad).
- ❖ **No observables en tiempo de ejecución:** aquellos atributos que se establecen durante el desarrollo del sistema (por ejemplo, modificabilidad, portabilidad, portabilidad, escalabilidad).

En la Tabla 18, “Descripción de atributos de calidad no observables y observables vía ejecución”, se muestra la descripción de los atributos de calidad elegidos, para la generación del árbol de utilidad.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Tabla 18. Descripción Atributos de Calidad no Observables y Observables Vía Ejecución

Atributo de Calidad	Descripción
Modificabilidad	Es la habilidad de realizar cambios futuros al sistema (Booch & Jacobson, 1999).
Disponibilidad	Es la medida de disponibilidad del sistema para el uso (Klein, et al, 1995).
Rendimiento	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, 2007).
Seguridad	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, Jai , & Klein, 2001).
Interoperabilidad	Es la medida de la habilidad de que un grupo de partes del sistema trabajen con otro sistema. Es un tipo especial de integralidad (Bass, Clements, & Kazman,1998).

Fuente: Kazman, Bass, & Klein, 2006.

En la *Figura 52*, “Concepto de Árbol de Utilidad. (UT).” se muestran las salidas de la generación del árbol de utilidad proveen una lista priorizada de escenarios.

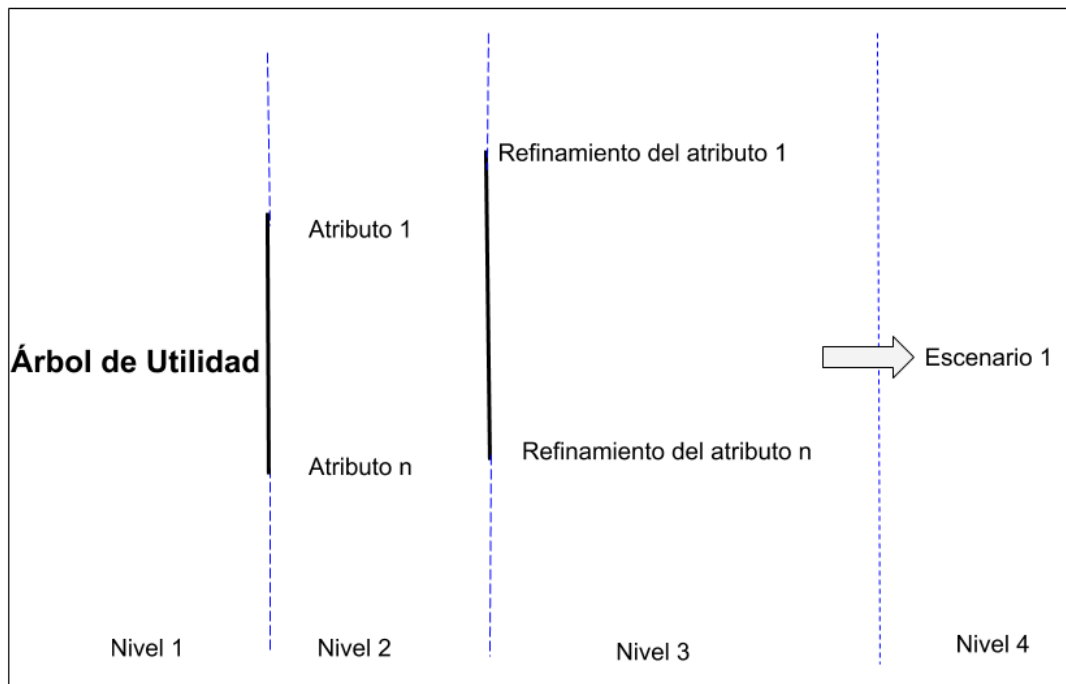


Figura 52. Concepto de Árbol de Utilidad (UT).

Fuente: Elaboración propia basada en (Bass L. , Clements, Kazman, & Klein, 2008)

En la *Figura 54*, “Árbol de Utilidad” se muestra la generación de los atributos de calidad. los cuales son: modificabilidad, disponibilidad, rendimiento, seguridad e interoperabilidad.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Un árbol de utilidad se compone inicialmente de una raíz principal. En la raíz principal se encuentra el término "Árbol de Utilidad". Los atributos de calidad se encuentran en el segundo nivel. En el tercer nivel se especifican los requisitos descritos en el nivel anterior y, finalmente, en el cuarto y último nivel se describen los miembros de Escenarios de Calidad y también sus prioridades.

Las prioridades descritas en un árbol de utilidad tienen el siguiente formato: (X, Y), donde X se refiere al grado de importancia de este escenario para el sistema, y la Y indica el nivel de dificultad para implementar el escenario. Tanto el grado de importancia como la dificultad de implementación se clasifican en: Alto (A), Medio (M) o Bajo (B). De esta forma, por ejemplo, en el escenario "Cambio de la Interfaz de usuario" se indica que tiene una importancia alta (A) y una dificultad baja (B)".

CAPÍTULO IV

PROPUESTA DE SOLUCION

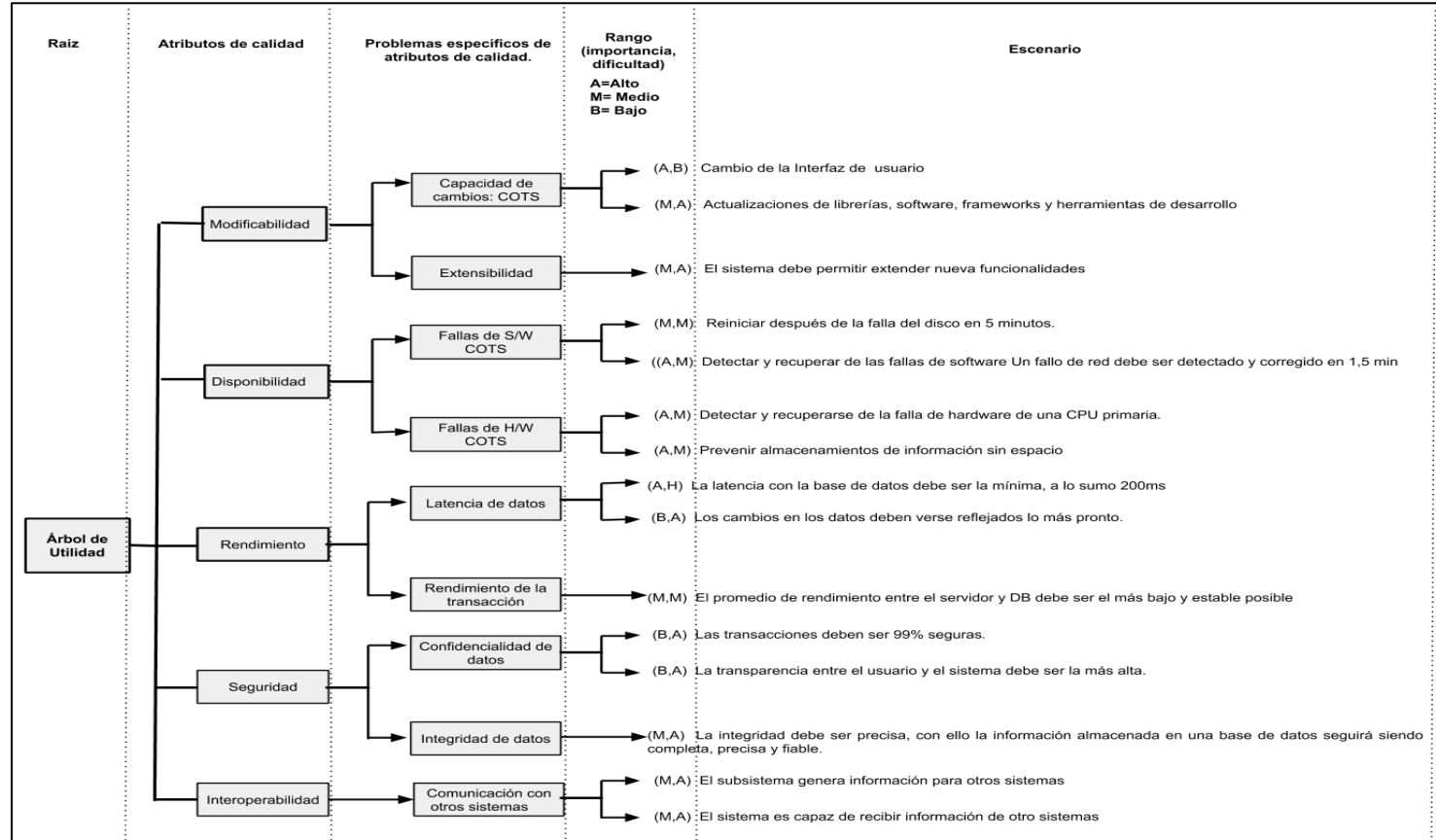


Figura 53. Árbol de Utilidad.
Fuente: Elaboración propia

Análisis de escenarios mediante ATAM

El análisis de escenarios se documenta por medio de una plantilla de análisis; que se muestra en la Tabla 19, “Plantilla para la documentación del enfoque arquitectónico”. Estos escenarios fueron tomados del árbol de utilidad, (UT). En esta plantilla se define el número de escenario con el atributo que corresponde; un escenario consta de tres partes: el estímulo, el ambiente y la respuesta al ambiente.

El ambiente describe qué sucede en el sistema al momento del estímulo. **El estímulo** describe lo que el involucrado en el desarrollo hace para iniciar la interacción con el sistema. **La respuesta** describe, a través de la arquitectura, cómo debería responder el sistema ante el estímulo.

Seguidamente las decisiones arquitectónicas se categorizan por una organización Arquitectural, en puntos de sensibilidad, compensación, riesgo y no riesgo. Por último, el razonamiento, por medio de una justificación cualitativa o cuantitativa y el diagrama arquitectónico si lo amerita.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Tabla 19. Plantilla para la Documentación y Análisis del Enfoque Arquitectónico

Análisis de enfoque arquitectónico				
Escenario #: <>	<un escenario del árbol de utilidad>			
Atributo:	<atributo de calidad: Rendimiento Seguridad, modificabilidad, etc>			
Ambiente:	Supuestos relevantes sobre el entorno en el que reside el sistema.			
Estímulo:	<Una declaración precisa del estímulo del atributo de calidad (p. ej., falla, amenaza, modificación,) encarnado por el escenario>			
Respuesta:	<una declaración precisa de la respuesta del atributo de calidad (por ejemplo, tiempo de respuesta, medida de la dificultad de modificación)>			
Categoría de Organización Arquitectural				
Decisiones arquitectónicas:	Puntos de			
	Sensibilidad	Compensación	Riesgo	No Riesgo
	<puntos de sensibilidad#>	<puntos de compensación#>	<puntos de riesgo#>	<puntos de no riesgo <#>
	< S# >	< C#>	< R #>	< NR #>
Razonamiento	<justificación cualitativa y / o cuantitativa de por qué la lista de decisiones arquitectónicas contribuye para cumplir con el requisito de respuesta del atributo de calidad>			

Fuente: (Kazman, Klein, & Clements, 2000).

CAPÍTULO IV

PROPUESTA DE SOLUCION

A continuación, se detallan los 15 escenarios principales, los cuales fueron definidos y priorizados para el proceso de evaluación de la arquitectura de software.

El escenario 01 se muestra en la Tabla 20, “Análisis Propuesta Arquitectónica de Escenario 01 por Medio de ATAM.”, del atributo de calidad Modificabilidad. Este escenario está relacionado con los cambios de la interfaz de usuario.

Tabla 20. Análisis Propuesta Arquitectónica de Escenario 01 por Medio de ATAM

Análisis de enfoque arquitectónico				
Escenario #: 01	Cambios de la interfaz de usuario			
Atributo:	Modificabilidad. (Capacidad de cambios: COTS)			
Ambiente:	Los cambios que se realizan en la UI deben ser adaptados fácilmente sin ningún problema.			
Estímulo:	Actualización y funcionamiento de módulos modificados.			
Respuesta:	La dificultad de modificación de un módulo se realiza de manera eficaz sin afectar en absoluto a los usuarios ni a otros módulos.			
Categoría de Organización Arquitectural				
Decisiones arquitectónicas:	Puntos de			
	Sensibilidad	Compensación	Riesgo	No Riesgo
	Cambio de una funcionalidad	S1 S2	C1	R1

CAPÍTULO IV

PROPUESTA DE SOLUCION

Razonamiento	S1: Duplicidad de funcionalidades.: S2: Afecta la interoperabilidad. C1: Mejora una funcionalidad, pero se pierde otra. R1. Pérdida de alguna funcionalidad del sistema.
---------------------	--

Fuente: Elaboración propia.

En la Tabla 21, “Análisis Propuesta Arquitectónica de Escenario 02 por Medio de ATAM.” se muestra el escenario 02, este escenario documenta la actualización de librerías, software, frameworks y herramientas de desarrollo.

Tabla 21. Análisis Propuesta Arquitectónica de Escenario 02 por Medio de ATAM

Análisis de enfoque arquitectónico	
Escenario #: 02	Actualización de librerías, software, frameworks y herramientas de desarrollo.
Atributo:	Modificabilidad. (Capacidad de cambios: COTS)
Ambiente:	Las actualizaciones que se realicen de librerías, software, frameworks y herramientas se deben de realizar periódicamente.
Estímulo:	Actualización y funcionamiento de librerías, software, frameworks y herramientas de desarrollo.
Respuesta:	La actualización de librerías, software, frameworks y herramientas de desarrollo, se realiza sin ningún problema sin afectar el funcionamiento del sistema.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Categoría de Organización Arquitectural				
Decisiones arquitectónicas:	Puntos de			
	Sensibilidad	Compensación	Riesgo	No riesgo
Actualización de librerías	S1	C1	R1	
Razonamiento	S1: Mejora el desempeño y la funcionalidad.			
	C1. Se actualizan las librerías, pero se deben realizar cambios en la lógica de programación del sistema.			
	R1: Insatisfacción del usuario y administrador del sistema.			
Fuente: Elaboración propia.				

En la Tabla 22, “Análisis Propuesta Arquitectónica de Escenario 03 por Medio de ATAM.” se muestra el escenario 03; el cual indica que el sistema debe permitir extender nuevas funcionalidades.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Tabla 22. Análisis Propuesta Arquitectónica de Escenario 03 por Medio de ATAM

Análisis de enfoque arquitectónico				
Escenario #: 03	El sistema debe permitir extender nuevas funcionalidades.			
Atributo:	Modificabilidad. (Extensibilidad)			
Ambiente:	Las nuevas extensiones de las funcionalidades no deben de afectar en absoluto el desempeño del sistema.			
Estímulo:	Nuevas funcionalidades deben actualizarse e integrarse sin ningún problema.			
Respuesta:	El sistema tiene nuevas funcionalidades que no afectan ningún módulo.			
Categoría de Organización Arquitectural				
Decisiones arquitectónicas:	Puntos de			
	Sensibilidad	Compensación	Riesgo	No Riesgo
Agregación de funcionalidades	S1	C1		R1
Razonamiento	S1: Mejora la funcionabilidad y la navegabilidad.			
	C1: Mejora la interoperabilidad, pero puede disminuir la navegabilidad.			
	NR1: Agregar un componente no perjudica en nada el sistema.			
Fuente: Elaboración propia.				

CAPÍTULO IV

PROPUESTA DE SOLUCION

El escenario 04 se muestra en la Tabla 23, “Análisis Propuesta Arquitectónica de Escenario 04 por Medio de ATAM.” en el atributo de disponibilidad. Este escenario documenta el reinicio después de la falla del disco en 5 minutos.

Tabla 23. Análisis Propuesta Arquitectónica de Escenario 04 por Medio de ATAM

Análisis de enfoque arquitectónico					
Escenario #: 04		Reiniciar después de la falla del disco en 5 minutos.			
Atributo:		Disponibilidad. (Fallas de S/W COTS)			
Ambiente:		El reinicio de la falla del disco no supera los 5 min.			
Estímulo:		El reinicio y las fallas en el disco se deben de realizar sin ningún problema.			
Respuesta:		El sistema primero asegura de que el estado de la copia de seguridad sea lo suficientemente actualizado antes de reanudar los servicios.			
Categoría de Organización Arquitectural					
Decisiones arquitectónicas:		Puntos de			
		Sensibilidad	Compensación	Riesgo	No Riesgo
Renovación de servicios		S1	C1	R1	
Razonamiento		S1: Mejora el rendimiento y la confiabilidad.			
		C1: Mejora la disponibilidad, pero disminuye el desempeño.			

CAPÍTULO IV

PROPUESTA DE SOLUCION

R1: Reiniciar un componente puede producir pérdidas de memoria o datos.

Fuente: Elaboración propia.

En la Tabla 24, “Análisis Propuesta Arquitectónica de Escenario 05 por Medio de ATAM” se muestra el escenario 05 el cual indica Detectar y recuperarse de la falla de Software

Tabla 24. Análisis Propuesta Arquitectónica de Escenario 05 por Medio de ATAM

Análisis de enfoque arquitectónico				
Escenario #: 05		Detectar y recuperarse de la falla de Software. Una falla de red debe de ser detectado y corregido en 1,5 min.		
Atributo:		Disponibilidad. (Fallas de S/W COTS)		
Ambiente:		En operaciones normales.		
Estímulo:		Fallas en la red.		
Respuesta:		La disponibilidad de respuesta debe ser de manera inmediata.		
Categoría de Organización Arquitectural				
Decisiones arquitectónicas:	Punto de			
	Sensibilidad	Compensación	Riesgo	No Riesgo
La falla de red no se detecta a tiempo.	S1	C1	R1	

CAPÍTULO IV

PROPUESTA DE SOLUCION

Razonamiento

S1: No se obtiene respuesta por parte de la red.

C1: Se obtiene respuesta en 1,5 minuto, pero esta no garantiza su disponibilidad.

R1: Al no recuperarse de una falla de red, el sistema colapsa.

Fuente: Elaboración propia.

En la Tabla 25, “Análisis Propuesta Arquitectónica de Escenario 06 por Medio de ATAM” se presenta el escenario 06. Este escenario documenta la acción de detectar y recuperarse de la falla de hardware de una CPU primaria.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Tabla 26. Análisis Propuesta Arquitectónica de Escenario 06 por Medio de ATAM

Análisis de enfoque arquitectónico				
Escenario #: 06	Detectar y recuperarse de la falla de hardware de una CPU primaria.			
Atributo:	Disponibilidad. (Fallas de H/W COTS)			
Ambiente:	En operaciones normales.			
Estímulo:	Fallas en el CPU			
Respuesta:	La disponibilidad de respuesta es inmediata.			
Categoría de Organización Arquitectural				
Decisiones arquitectónicas:	Puntos de			
	Sensibilidad	Compensación	Riesgo	No Riesgo
Respaldo del CPU	S1		R1	
No tener un canal de datos de respaldo	S2	C1	R2	
watchdog	S3			
Hearbeat	S4			
Razonamiento	S1: Detectar fallas en 2 segundos basado en frecuencias de watchdog y Hearbeat. S2: No se tiene un respaldo para recuperarse los datos S3: No hay respuesta del watchdog. C1: Contar con una canal de respaldo.			

CAPÍTULO IV

PROPUESTA DE SOLUCION

R1: Garantiza que no haya fallas mediante el uso de hardware y sistema operativo diferentes.

R2: El requisito de disponibilidad puede estar en riesgo debido a la falta de un canal de respaldo.

Fuente: Elaboración propia.

El escenario 07 se muestra en la Tabla 26. “Propuesta Arquitectónica de Escenario 07 por Medio de ATAM.”. Este escenario documenta la prevención de almacenamiento de información sin espacio.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Tabla 26. Propuesta Arquitectónica de Escenario 07 por Medio de ATAM

Análisis de enfoque arquitectónico				
Escenario #: 07	Prevenir almacenamiento de información sin espacio.			
Atributo:	Disponibilidad. (Fallas de H/W COTS)			
Ambiente:	En operaciones normales.			
Estímulo:	Los almacenamientos de información tienen poco espacio			
Respuesta	La disponibilidad de respuesta debe ser inmediata permitiendo el espacio suficiente de almacenamiento de datos.			
Categoría de Organización Arquitectural				
Decisiones arquitectónicas:	Puntos de			
	Sensibilidad	Compensación	Riesgo	No Riesgo
Respaldo del almacenamiento de datos.	S1	C1	R1	
Razonamiento	S1: La disponibilidad se puede ver afectada por los tiempos de respaldo de almacenamiento de datos. C1: Se tiene disponibilidad, pero la navegación puede ser lenta. R1: Pérdida de datos.			

Fuente: Elaboración propia.

CAPÍTULO IV

PROPUESTA DE SOLUCION

En la Tabla 27, “Propuesta Arquitectónica de Escenario 08 por Medio de ATAM.” se presenta el escenario 08, en el atributo de calidad rendimiento. Este escenario documenta; la latencia con la base de datos debe ser la mínima, a lo sumo 200ms.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Tabla 27. Propuesta Arquitectónica de Escenario 08 por Medio de ATAM

Análisis de enfoque arquitectónico				
Escenario #: 08	La latencia con la base de datos debe ser la mínima, a lo sumo 200ms.			
Atributo:	Rendimiento. (Latencia de datos)			
Ambiente:	La latencia que existe entre la base de datos y la aplicación tiene tiempos de respuesta que no superan los 200 milisegundos.			
Estímulo:	Transacciones (CRUD) a base de datos son recibidas en menos de 200ms.			
Respuesta:	Las transacciones de base de datos son procesadas y generan sus datos con un tiempo de respuesta de 200ms.			
Categoría de Organización Arquitectural				
Decisiones arquitectónicas:	Sensibilidad	Compensación	Riesgo	No Riesgo
Las transacciones se ven incompleta a nivel de información.	S1	C1	R1	
Razonamiento	S1: Las transacciones solo guardarían datos importantes se evitaría bitácoras para tener tiempo de respuesta bajo. C1: Mejora la velocidad de respuesta, pero no permite tener información en bitácoras sobre acciones realizadas por el usuario.			

CAPÍTULO IV

PROPUESTA DE SOLUCION

R1: Se desconoce quien realiza acciones desde un usuario específico.

Fuente: Elaboración propia.

El escenario 09 se muestra en la Tabla 28, “Propuesta Arquitectónica de Escenario 09 por Medio de ATAM”. Este escenario documenta los cambios en los datos deben verse reflejados lo más pronto.

Tabla 28. Propuesta Arquitectónica de Escenario 09 por Medio de ATAM

Análisis de enfoque arquitectónico	
Escenario #: 09	Los cambios en los datos deben verse reflejados lo más pronto.
Atributo:	Rendimiento. (Latencia de datos)
Ambiente:	Los cambios que se realizan de la información en base de datos deben verse reflejados de manera inmediata permitiendo al usuario ver dichos cambios de manera adecuada.
Estímulo:	Actualización de la información de base de datos en interfaz de usuario.
Respuesta:	La información recibida del usuario es procesada y actualizada de manera inmediata en la interfaz de usuario.
Categoría de Organización Arquitectural	
Decisiones arquitectónicas:	Punto de

CAPÍTULO IV

PROPUESTA DE SOLUCION

	Sensibilidad	Compensación	Riesgo	No Riesgo
La información no se presenta completa	S1	C1	R1	
Razonamiento	S1: Los datos no se muestran completos porque algunos se consideran innecesarios o redundantes. C1: Mejora la velocidad de respuesta, pero el usuario puede estar restringido de ver datos relevantes para él. R1: Insatisfacción del usuario al usar el sistema.			

Fuente: Elaboración propia.

En la Tabla 29, “Propuesta Arquitectónica de Escenario 10 por Medio de ATAM.” se presenta el escenario 10. Este escenario documenta el promedio de rendimiento entre el servidor de aplicación y DB debe ser el más bajo y estable posible.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Tabla 29. Propuesta Arquitectónica de Escenario 10 por Medio de ATAM

Análisis de enfoque arquitectónico	
Escenario #: 10	El promedio de rendimiento entre el servidor de aplicación y DB debe ser el más bajo y estable posible
Atributo:	Rendimiento. (Rendimiento de transacción)
Ambiente:	Las transacciones a base de datos están optimizadas permitiendo al sistema tener un rendimiento de respuesta inmediato; sin espera excesiva de tiempo y sin el mal uso de recursos de memoria y otros medios.
Estímulo:	Transacciones de base de datos son realizadas por el sistema con un rendimiento de respuesta inmediato.
Respuesta:	Las transacciones de bases de datos tienen un óptimo nivel de respuesta.

Categoría de Organización Arquitectural

Decisiones arquitectónicas:	Puntos de			
	Sensibilidad	Compensación	Riesgo	No Riesgo
Las transacciones se realizan por	S1	C1	R1	

CAPÍTULO IV

PROPUESTA DE SOLUCION

medio de C2
procedimientos
almacenados en la
base de datos.

Razonamiento

S1: Las transacciones no se realizan de manera adecuada.

C1: Mejora la velocidad de respuesta, pero no permiten la seguridad de una inyección de SQL.

C2: Mejora la respuesta, pero afecta la seguridad.

R1: Acceso a la información por terceros.

Fuente: Elaboración propia.

En el atributo de calidad de seguridad, se encuentra el escenario 11, mostrado en la Tabla 30, “Propuesta Arquitectónica de Escenario 11 por Medio de ATAM.” donde indica que las transacciones deben ser 99% seguras.

Tabla 30. Propuesta Arquitectónica de Escenario 11 por Medio de ATAM

CAPÍTULO IV

PROPUESTA DE SOLUCION

Análisis de enfoque arquitectónico	
Escenario #: 11	Las transacciones deben ser 99% seguras.
Atributo:	Seguridad. (Confidencialidad de datos)
Ambiente:	La seguridad de las transacciones alcanza su máximo nivel siendo este del 99%; donde los datos deben ser protegidos para evitar robo de información sensible para el usuario como para el sistema.
Estímulo:	El sistema recibe información sensible por parte del usuario.
Respuesta:	El sistema procesa la información de manera confidencial por medio de procesos de encriptación luego el mismo da una respuesta al usuario de manera confidencial.

Categoría de Organización Arquitectural				
Decisiones arquitectónicas:	Puntos de			
	Sensibilidad	Compensación	Riesgo	No Riesgo
Se realiza la encriptación de información recibida del usuario.	S1	C1	R1	
Razonamiento	S1: Una errónea encriptación de datos aumenta la posibilidad de robo de información. C1: Mejora la seguridad, pero disminuye la velocidad de respuesta. R1: Tiempos de respuesta excesivos en algoritmos de encriptación.			

Fuente: Elaboración propia.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Tabla 31. Propuesta Arquitectónica de Escenario 12 por Medio de ATAM

Análisis de enfoque arquitectónico				
Escenario #: 12	La transparencia entre el usuario y el sistema deben ser la más alta.			
Atributo:	Seguridad. (Confidencialidad de datos)			
Ambiente:	La información recibida por el usuario es transparente durante el procesamiento del sistema.			
Estímulo:	La seguridad que se le brinda al usuario es transparente por parte del sistema.			
Respuesta:	El sistema procesa la información de manera segura por medio de procesos transparentes y confidenciales.			
Categoría de Organización Arquitectural				
Decisiones arquitectónicas:	Puntos de			
	Sensibilidad	Compensación	Riesgo	No Riesgo
Se realiza el procesamiento de información seguro y confidencial por parte del sistema.	S1	C1	R1	

CAPÍTULO IV

PROPUESTA DE SOLUCION

Razonamiento	S1: Procesamiento lento debido a la verificación de seguridad de los datos. C1: Mejora la seguridad, pero disminuye la velocidad de respuesta. R1: Tiempos de respuesta excesivos durante el procesamiento de la información.
---------------------	---

Fuente: Elaboración propia.

El escenario 13 es representado en la Tabla 32, “Propuesta Arquitectónica de Escenario 13 por Medio de ATAM.”. Este documenta la integridad de datos debe ser precisa, la inconsistencia de la información debe ser evitada en su totalidad, ya que esta debe ser totalmente fiable.

Tabla 32. Propuesta Arquitectónica de Escenario 13 por Medio de ATAM

Análisis de enfoque arquitectónico	
Escenario #: 13	La integridad de datos debe ser precisa, la inconsistencia de la información debe ser evitada en su totalidad, ya que esta debe ser totalmente fiable.
Atributo:	Seguridad. (Integridad de datos)
Ambiente:	La integridad de datos es precisa, la inconsistencia de la información es vitada en su totalidad ya que esta es totalmente fiable.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Estímulo: La información que brinda el usuario es almacenada en el sistema de forma completa e íntegra.

Respuesta: Los datos son recibidos por el sistema y se procede a almacenar de manera íntegra evitando fallas de datos e inconsistencias de estos.

Categoría de Organización Arquitectural

Decisiones arquitectónicas:	Puntos de			
	Sensibilidad	Compensación	Riesgo	No riesgo
Los datos son excesivos por lo que no se logran almacenar todos en su totalidad.	S1	C1	R1	
Razonamiento	S1: Los datos incompletos causan inconsistencias de información del sistema.			
	C1: Mejora la velocidad de respuesta, pero no permiten la integridad de datos.			
	R1: Pérdida de datos e información.			

Fuente: Elaboración propia.

CAPÍTULO IV

PROPUESTA DE SOLUCION

En la Tabla 33, “Propuesta Arquitectónica de Escenario 14 por Medio de ATAM.” se puede observar el escenario 14, del atributo de Interoperabilidad, lo cual documenta el siguiente escenario; el subsistema genera información para otros sistemas.

Tabla 33. Propuesta Arquitectónica de Escenario 14 por Medio de ATAM

Análisis de enfoque arquitectónico				
Escenario #: 14		El subsistema genera información para otros sistemas.		
Atributo:		Interoperabilidad. (Comunicación con otros sistemas).		
Ambiente:		Durante la integridad con otros sistemas de la Universidad Nacional, el sistema utiliza estándares asignados.		
Estímulo:		El usuario ingresa al sistema para realizar una consulta el mismo puede consultar otros subsistemas para obtener información.		
Respuesta:		Se obtiene la información por parte del sistema, consultando otros subsistemas de la Universidad Nacional.		
Categoría de Organización Arquitectural				
Decisiones arquitectónicas:		Puntos de		
		Sensibilidad	Compensación	Riesgo
Utilización de estándares establecidos.	de	S1	C1	R1
	no		C2	

CAPÍTULO IV

PROPUESTA DE SOLUCION

Razonamiento	S1: En caso de no existir una buena comunicación con otros subsistemas se debe agregar nuevas funcionalidades.
	C2: Aumenta la disponibilidad de información, pero se ve afectado el tiempo de respuesta hacia el usuario.
	R1: Dependencia de otros subsistemas.

Fuente: Elaboración propia.

El escenario 15 se muestra en la Tabla 34, “Propuesta Arquitectónica de Escenario 15 por Medio de ATAM” documentando lo siguiente; el sistema es capaz de recibir información para otros sistemas.

Tabla 34. Propuesta Arquitectónica de Escenario 15 por Medio de ATAM

Análisis de enfoque arquitectónico	
Escenario #: 15	El sistema es capaz de recibir información para otros sistemas.
Atributo:	Interoperabilidad. (Comunicación con otros sistemas).
Ambiente:	El sistema está operando en su ambiente normalmente.
Estímulo:	El sistema automático y flexible para encargarse de distribuir la información.

CAPÍTULO IV

PROPUESTA DE SOLUCION

Respuesta: El sistema genera, presta servicios de información y procesa estos servicios sin ningún problema.

Categoría de Organización Arquitectural

Decisiones arquitectónicas:	Puntos de			
	Sensibilidad	Compensación	Riesgo	No Riesgo
El sistema puede ofrecer servicios sin ningún problema a otros subsistemas.	S1	C1		R1
		C2		
Razonamiento	S1: En caso de no existir una buena comunicación con otros subsistemas se debe agregar nuevas funcionalidades.			
	C2: Aumenta la disponibilidad de información, pero se ve afectado el tiempo de respuesta hacia el usuario.			
	R1: Dependencia de otros subsistemas.			

Fuente: Elaboración propia.

Para el siguiente paso, que corresponde a la fase de pruebas, se toman los atributos de calidad del paso anterior y se ponen a votación.

6. Priorización de escenarios y analizar atributos de calidad: Revisión y votación de estos.

CAPÍTULO IV

PROPUESTA DE SOLUCION

En este paso se tomaron los escenarios evaluados y sus atributos, y se pusieron a votación. Los arquitectos encargados de la evaluación realizaron su elección virtualmente, por medio de la escala de LIKERT.

En la Tabla 35, “Votación de Atributos de Calidad”, se observa el resultado obtenido. Se valoraron 15 escenarios, (mostrados en el paso 5), la escala de votación fue de muy importante a poco importante. (Ver anexo 14).

Tabla 35. Votación de Atributos de Calidad

Escenario	Puntaje
Prevenir almacenamientos de información sin espacio	40
Las transacciones deben ser 99% seguras.	39
Los cambios en los datos deben verse reflejados lo más pronto.	39
Cambio de la Interfaz de usuario	38
La transparencia entre el usuario y el sistema debe ser la más alta.	36
Actualizaciones de librerías, software, frameworks y herramientas de desarrollo	36
La integridad debe ser precisa, con ello la información almacenada en una base de datos seguirá siendo completa, precisa y fiable.	36
El promedio de rendimiento entre el servidor y DB debe ser el más bajo y estable posible	35
El sistema debe permitir extender nuevas funcionalidades	33
Detectar y recuperar de las fallas de software Un fallo de red debe ser detectado y corregido en 1,5 min	33
Detectar y recuperarse de la falla de hardware de una CPU primaria.	33
La latencia con la base de datos debe ser la mínima, a lo sumo 200ms	33
El subsistema genera información para otros sistemas	32
El sistema es capaz de recibir información de otros sistemas	32
Reiniciar después de la falla del disco en 5 minutos.	31

Fuente: Elaboración propia.

4.3.5 Ejecución de ATAM, Fase de Informes

Fase de Informes

Corresponde a la última fase del ATAM, donde se muestran los resultados obtenidos y se realiza un juicio de valor.

7. Presentación y correlación de resultados obtenidos

En la etapa final de la ejecución del Método de Análisis de Compensación de Arquitecturas (ATAM), corresponde a la presentación de un informe final, muy general sobre los resultados obtenidos durante las ejecuciones de todas sus etapas.

Primeramente, en el paso 4 se realizó un tipo encuesta como lo presenta este método, con el fin de identificar los atributos de calidad. Para este método de evaluación de arquitectura los atributos de calidad son de vital importancia en el diseño de una arquitectura de software.

Seguidamente, con la identificación de los atributos de calidad se logró construir el Árbol de Utilidad (UT), este mecanismo es implementado por este método con el fin de analizar los atributos seleccionados y realizar una evaluación de escenarios.

Por último, estos escenarios fueron sometidos a una votación en este caso se utilizó la escala de Likert.

En conclusión, se logró determinar cuáles fueron los enfoques arquitectónicos y atributos de calidad que favorecían la arquitectura evaluada.

CAPÍTULO IV

PROPUESTA DE SOLUCION

De los 15 escenarios evaluados estos fueron calificados según la importancia dentro la arquitectura, donde los que tienen una mayor calificación son más relevantes según los arquitectos, para el diseño arquitectural.

Capítulo V: Conclusiones y Recomendaciones

5.1 Conclusiones

La estrategia utilizada para el proyecto fue efectiva, ya que cumplieron los objetivos propuestos. Se obtuvo como producto resultante un diseño arquitectural de Software para la Sección de Seguridad Institucional, muy detallado y principalmente bajo los requisitos solicitados por los involucrados.

La documentación de la arquitectura es una actividad fundamental para poder realizar una comunicación adecuada del diseño. Es importante recalcar que no existen criterios precisos que permitan determinar el grado exacto de documentación que se debe realizar, y esto se debe a diversos aspectos que incluyen a los involucrados, el tipo de proyecto y la experiencia del equipo de desarrollo.

La metodología de trabajo desarrollada ha aportado una línea de investigación e innovación en el diseño de procesos de negocio y en su evaluación; con ello abre un camino para las investigaciones futuras.

5.2 Limitaciones

En el desarrollo de este proyecto se presentaron las siguientes limitaciones:

- Escasez bibliográfica sobre ciertos conceptos ha llevado a la búsqueda exhaustiva de la información sobre diferentes referencias.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

- Falta de comunicación de manera inmediata con los involucrado en la Sección de Seguridad, por lo cual se dio un retraso de la información solicitada para obtener la información.
- No se pretende clasificar los tipos de arquitecturas analizadas por parte de los diferentes autores.
- El estudio no abarca las fases de la implementación de la arquitectura integral.
- No se incluirán los elementos de otras ramas o disciplinas distinta a la Ingeniería en sistemas.
- El proyecto se enfocará únicamente en la Sección de Seguridad de la Universidad nacional.
- La investigación no se extenderá a otras secciones o departamentos fuera de la Universidad Nacional.
- El estudio se realizará con una muestra de los involucrados dentro del procesos de solicitud y reservación de espacios en la Sección de Seguridad.
- El período de tiempo de realización del proyecto abarca 2 años, sin embargo, por el tema mundial de la pandemia estos tiempos son afectados.
- Incluirá solamente a los involucrados en el área de la tecnología de información para recabar información actual de la tecnología utilizada por parte de ese departamento.
- El estudio sólo propondrá un modelo por lo que no se llegará a la implementación.

5.3 Trabajos Futuros

A partir de los resultados obtenidos con el desarrollo de la documentación de esta Arquitectura de Software se encuentran algunos posibles puntos de mejora que podrían facilitar y ayudar a futuras investigaciones. Estos se detallan a continuación:

- 1) Por medio de este proyecto se ha investigado los estudios más recientes donde se utilizó la metodología de revisión de bibliográfica y literatura en el cual a pesar de que se lograron los objetivos existe la posibilidad de mejora de la versión actual de la propuesta presentada en este proyecto.
- 2) Otra recomendación es extender la investigación sobre las arquitecturas basadas en servicios, específicamente en los servicios REST, que en los últimos años ha tomado fuerzas en las organizaciones. Esto con el fin de ayudar en el progreso de las organizaciones con sus modelos de negocio.
- 3) Es recomendable desarrollar cursos dentro del plan de estudio de la carrera de Ingeniería de Sistema; donde permita a los interesados tener una visión clara de la arquitectura de software para reforzar el aprendizaje de la población estudiantil.
- 4) Otra recomendación es promover investigaciones similares a este proyecto.
- 5) Basado en el método de evaluación de Arquitecturas de Software, aplicado en este proyecto se descubrieron varios puntos de sensibilidad, de compensación, de riesgos y no riesgos. Los cuales podrían ser utilizados como guía para tomar futuras decisiones arquitectónicas.

Referencias

- Aïcha Choutri, F. B. (2010). *A Tile Logic Based Approach for Software Architecture Description Analysis*. doi: 10.4236/jsea.2010.311126
- Alarcón, V. (2006). *Desarrollo de sistemas de información: Una metodología basada en el modelado*. Univ. Politèc. de Catalunya (Vol. 120). Catalunya.
- Amit, R., & Zott, C. (2001). Value creation in e-business. *Strategic management journal*, 22(6/2), 493.
- Arroyo , E., Ch Peley, R., & Fino, J. (2009). Caracterización del manejo de mensajes para la orquestación de servicios web. *Omnia*, 15(2), 116-124.
- Ballinger, K. (2003). *Net Web Services : Architecture and Implementation*. Addison-Wesley Professional.
- Barbacci, M. R., Ellison, R., Lattanze, A. J., Wood, W. G., Weinstock, C. B., & Stafford, , J. (2002). *Quality attribute workshops, 2nd Edition*. Pittsburgh, Estados Unidos: CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.
- Barros, A., Dumas , M., & Oaks, P. (2005). A critical overview of the web services choreography description language. *BPTrends Newsletter*, 3, 1-24.
- Barros, A., Dumas , M., & Oaks , P. (2005). WS-CDL Activities. *Imagen*. Obtenido de <https://www.bptrends.com/publicationfiles/03-05%20WP%20WS-CDL%20Barros%20et%20al.pdf>
- Barry, D. K., & Dick, D. (2013). Web Services Description Language. *[Imagen]*.

Bass, L., Clements, P., & Kazman, R. (1998). *Software Architecture in Practice*. Addison-Wesley

1.

Bass, L., Clements, P., & Kazman, R. (2003). *Software architecture in practice*. Addison-Wesley Professional.

Bass, L., Clements, P., Kazman, R., & Klein, M. (2008). Evaluating the Software Architecture Competence of Organizations. *Llevado a cabo en la Conference on Software Architecture (WICSA 2008)*. Vancouver, BC, Canada. doi:10.1109/WICSA.2008.12

Bazán, P. (Octubre 2007). BPEL: Una propuesta para el uso de Web Services. *Trabajo presentado en XIII Congreso Argentino de Ciencias de la Computación*. Buenos Aires.

Benedict, T., Kirchmer, M., Scarsig , M., Frantz, P., Saxena, R., Morris, D., & Hilty , J. (02 de Noviembre de 2019). Guide to the Business Process Management Common. *[Imagen]*.

Booch, G., & Jacobson, J. R. (1999). *The Unified Software Development Process*. Addison-Wesley Professional.

Buschmann , F., Henney , K., & Schmidt, D. C. (2007). *Pattern-oriented software architecture, on patterns and pattern languages* (Vol. 5). John wiley & sons.

Carrascoso P, J. A., Chaviano, E., & Céspedes, A. (2009). *Procedimientos para la evaluación de arquitecturas de software basadas en componentes*.

Casas Anguita, J., Donado Campos, J. d., & Repullo Labrador, J. R. (enero de 2002). La encuesta como técnica de investigación.

Cervantes, H. (13 de April de 2015). Proceso general de diseño de la arquitectura. *Figura*.

- Cervantes, H., Velasco-Elizondo, P., & Castro, L. F. (2016). *ARQUITECTURA DE SOFTWARE. CONCEPTOS Y CICLO DE DESARROLLO*. Mexico: Cengage Learning.
- Choi , S., Lee , H., & Yoo, Y. (2010). The impact of information technology and transactive memory systems on knowledge sharing, application, and team performance: a field study. *MIS quarterly*, 855-870. *MIS quarterly* , 855-870.
- Chong, J., Macías, V., Marchan, K., & Villacres, M. (2006). BPM: Business Process Modeling. *[Imagen]*. Obtenido de Recuperado de https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1511791
- Clements, P. C., & Northrop, L. M. (1996). *Software Architecture: An Executive Overview*. Pittsburgh, Pennsylvania . Obtenido de Recuperado de <https://apps.dtic.mil/sti/pdfs/ADA305470.pdf>
- Clements, P., Bass, L., & Kazman, R. (2002). *Software Architecture in Practice (SEI Series in Software Engineering)*.
- Clements, P., Garlan, D., Little, R., Nord, R., & Stafford, J. (2003). Documenting Software Architectures Views and Beyond. *25th International Conference on Software Engineering, 2003. Proceedings..* (págs. 740-741). IEEE.
- Clements, P., Kazman, R., & Klein, M. (2003). *Evaluating software architectures*. Beijing, China: Tsinghua University Press.
- Consortio World Wide Web (W3C). (2004 de Febrero de 2004). El modelo de Web Services. *[Imagen]*. Obtenido de <https://www.w3.org/>
- Consortio World Wide Web (W3C). (2004). Interoperability of Web Services. *[Imagen]*.

- Danhke, G. L. (1989). *Investigación y Comunicación*. México: McGraw-Hill.
- Del Pino, S. (2008). Población y muestra. *Revista Digital, Innovación y Experiencias Educativas*, 12.
- Deutsch, A., Hull, R., Batra, V., & Chen, Y. (2016). *Towards a shared ledger business collaboration language based on data-aware processes*. Springer, Cham.
- Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. (2013). *Fundamentals of Business Process Management* (Vol. I). Heidelberg: Springer.
- El Eman, K., & Madhavji, N. H. (1995, March). A field study of requirements engineering practices in information systems development. In *Proceedings of 1995 IEEE International Symposium on Requirements Engineering*, pp. 68-80.
- Erl, T., Karmarkar, A., Walmsley, P., Haas, H., Orchard, D. U., Liu, K., . . . Andre TostJames Pasley, A. T. (2008). *Web Service Contract Design and Versioning for SOA*. Prentice Hall.
- Fernández, L. (2006). Arquitectura de software. *Software Guru*, 2(3), 40-45.
- Gable, G. (1994). Integrating case study and survey research methods: an example in information systems. *European journal of information systems*,. *European journal of information systems*, 3(2), 112-126.
- García Sánchez, F. (2008). *Sistema basado en tecnologías del conocimiento para entornos de servicios web semánticos*.
- Garimella, K., Lees, M., & Williams, B. (2012). *BPM (Gerencia de Proceso de Negocio)*.

Garimella, K., Michael , L., & Bruce , W. (2008). *BPM (Gerencia de procesos de negocio).*"

Introducción a BPM. Recuperado el 07 de 08 de 2020, de https://d1wqtxts1xzle7.cloudfront.net/54146648/bpm.pdf?1502807809=&response-content-disposition=inline%3B+filename%3DBPM_GERENCIA_DE_PROCESOS_DE_NEGOCIO_Toma.pdf&Expires=1623105395&Signature=gE~0PcMkosfAW1hsj0h88KAtvoIqRLFYHpIbo~02xJzUr5nkjZEwDKdhmY7LL3I0

Giraldo, J., Guzmán, J. A., & Ovalle, D. A. (2008). Integración de procesos de negocio basados en servicios Web: Coreografía y satisfacción de restricciones. *Revista Ingenierías Universidad de Medellín*, 7(12), 141-155.

Hammer , M., & Champy, J. (1993). *Business process reengineering.* London.

Harmon, P. (2010). In Handbook on business process management 1In Handbook on business process management 1In Handbook on business process management 1In Handbook on business process management 1. En *The scope and evolution of business process management* (págs. 37-81). Springer, Berlin, Heidelberg.

Hart, C. (2018). *Doing a literature review: Releasing the research imagination.* London.

Hilliard, R. (1999). *Architecture Frameworks* . doi:10.1109/WICSA.2008.32

Hilliard, R. (22 de Octubre de 2012). *Architecture Frameworks. A documentation framework for architecture decisions.*

- Hitpass, B. (2012). Notación básica de BPMN. Obtenido de Recuperado de https://books.google.es/books?hl=es&lr=&id=Dm4-MGAy5vMC&oi=fnd&pg=PR1&dq=Bernhard+hitpass&ots=zXfLJ95s6L&sig=h3gMEtDRBggpyv_MhTrDoYPIZTg#v=onepage&q=Bernhard%20hitpass&f=false
- Hitpass, B. (2017). Notación para el modelado BPMN. [Imagen].
- Hitpass, B. (2017). *PM: Business Process Management: Fundamentos y Conceptos de Implementación 4a Edición actualizada y ampliada* (Cuarta ed.).
- Huff , S. L., & Munro, M. C. (1985). Information technology assessment and adoption: A field study. *MIS quarterly*, 327-340.
- IEEE. (2007). On non-functional requirements. . *15th IEEE International Requirements Engineering Conference (RE 2007)* (págs. 21-26). Delhi, India: IEEE. doi:10.1109/RE.2007.45
- IEEE. (25 de marzo de 2019). www.computer.org. Obtenido de www.computer.org
- Jick, T. (1979). Mixing qualitative and quantitative methods: Triangulation in action. *Administrative science quarterly*, 21(11), 602-611.
- Kazman, R., Bass, L., & Klein, M. (2006). The essential components of software architecture design and analysis. *Journal of Systems and Software*, , 79(8), 1207-1216.
- Kazman, R., Jai , A., & Klein, M. (2001). Quantifying the costs and benefits of architectural decisions. In *Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001*. 297-306. IEEE.

- Kazman, R., Klein, M., & Clements. (2000). *ATAM: Method for architecture evaluation*. Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst.
- Kelty, C. (2008). *The prototype: more than many and less than one* v v.
- Kenneth E, K., & Kendall, J. (2005). *Análisis y diseño de sistemas* (Sexta ed.). México: PEARSON EDUCACIÓN. Recuperado el Junio de 2020
- Klein, M. H., Barbacci, M., Longstaff, T. A., & Weinstock, C. B. (1995). *Quality Attributes*. UNIV PITTSBURGH PA SOFTWARE ENGINEERING INS: CARNEGIE-MELLON .
- Klein, M., Kazman, R., Bass, L., Carriere, J., & Lipson, M. B. (1999). "Attribute-based architecture styles". En Patric Donohoe (ed.) (Ed.), *Software Architecture* (págs. 225-243). Kluwer Academic Publishers.
- Kostochkina, O. V., & Gordievich, A. A. (2020). *Peculiarities of advertising in english*.
- Kreger, H. (2001). Introducción a los Servicios Web. *Actores, Objetos y las Operaciones de la Arquitectura de los Servicios Web*. Obtenido de Recuperado de
- Kreger, H. (Junio de 2003). Actores, objetos y operaciones de los Web Service. [Imagen].
- Kreger, Heather. (15 de Enero de 2001). *Web Services Conceptual Architecture (WSCA 1.0)*. Obtenido de https://www.researchgate.net/profile/Heather-Kreger/publication/235720479_Web_Services_Conceptual_Architecture_WSCA_10/link/s/563a67e008ae337ef2984607/Web-Services-Conceptual-Architecture-WSCA-10.pdf
- Kruchten, P. (Noviembre de 1995). Modelo de "4+1" vistas. *Figura*.
- Kruchten, P. (Noviembre de 1995). Notación para el diagrama de desarrollo. [Imagen].

- Kruchten, P. (Noviembre de 1995). Notación para el diagrama de procesos. *[Imagen]*.
- Kruchten, P. (Noviembre de 1995). Notación para el diagrama físico. *[Imagen]*.
- Kruchten, P. (Noviembre de 1995). Notación para la vista lógica. *[Imagen]*.
- Kruchten, P. (1995). Planos Arquitectónicos: El Modelo de 4+ 1 Vistas de la Arquitectura del Software. *IEEE software*, 12(6), 45-50.
- Kruchten, P. (Noviembre de 1995). Resumen del modelo de “4+1” vistas. *[Tabla]*.
- Kruchten, P. (1995). The 4+ 1 view model of architecture. *IEEE software*, 12(6), 45-50.
- Kruchten, P., Obbink, H., & Stafford, J. (2006). *The Past, Present, and Future of Software Architecture*. doi:10.1109/MS.2006.59
- Lázaro, D. (2018). Componentes de los Servicios Web. *[Imagen]*.
- Lázaro, D. (2018). *Introducción a los Web Services*. Recuperado el 20 de Setiembre de 2020, de Recuperado de <https://diego.com.es/introduccion-a-los-web-services>
- Lepkowski, J. M., Tucker, C., & Brick, J. M. (2008). *Advances in telephone survey methodology*.
- Li, Z., Gui, Z., Hofer, B., & Li, Y. (Julio de 2009). Web Orchestration. *[Imagen]*. doi:10.1007/978-981-32-9915-3_6
- Li, Z., Gui, Z., Hofer, B., & Li, Y. (2019). Web Service Choreography.
- López, G., Echeverría, A., Fierro, P., & Jeder, I. (Mayo,2007). Una propuesta de modelos de ciclo de vida (MCVS) para la integración de los procesos de negocio utilizando service oriented architecture (SOA). *IX Workshop de Investigadores en Ciencias de la Computación*, (págs. 559-563). La Plata, Buenos Aires.

- Magretta, J. (2002). Por qué importan los modelos de negocios. *Revista INCAE*, 12(3), 13.
- Manning , K. (13 de Agosto de 2020). *Tutorial y ejemplos de BPMN [Artículo del blog]*. Recuperado el 30 de Agosto de 2020, de <https://www.processmaker.com/es/blog/bpmn-2-0-tutorial-and-examples/>
- Marquez Solis, S. (2007). *Web semántica y servicios web semanticos*.
- Marquez Solis, S. (2007). Orquestación y Coreografía. *Tabla*.
- Michael, R., & Bruin, T. (2005). Towards a business process management maturity model. *ECIS 2005 Proceedings of the Thirteenth European Conference on Information Systems*, (págs. 1-12). Verlag and the London School of Economics.
- Newcomer, E. (2002). *Understanding Web Service: XML, WSDL, SOAP and UDDI*. Addison-Wesley Professional.
- Nicolás Kicillof, & Carlos Reynoso. (2004). *Lenguajes de Descripción de Arquitectura (ADL)*. Recuperado de <http://carlosreynoso.com.ar/archivos/carlos-reynoso-architectural-description-languages.pdf>.
- Nielsen, J. (01 de Noviembre de 1994). How to Conduct a Heuristic Evaluation. *Nielsen Norman Group*, 1, 1-8.
- Notes, A. S. (2015). Toward Simpler, not Simplistic, Quantification of Software Architecture and Metrics: Report on the Second International Workshop on Software Architecture and Metrics. *Software Engineering Institute*, 43-46. Obtenido de <https://www.sei.cmu.edu/>.

Object Management Group (OMG). (01 de Agosto de 2016). Modelo de madurez de procesos de negocio BPMM. *[Imagen]*.

Object Management Group. (2008). *OMG, Business Process Maturity Model (mpmn)*, 1.0.

Organization for the Advancement of Structured Information Standards. (11 de Abril de 2007). Purchase Order Process - Outline. *Imagen*. Obtenido de ORGANIZATION FOR THE ADVANCEMENT OF STRUCTURED INFORMATION STANDARDS (OASIS): <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>

Osorio Rivera, F. L. (2008). *Base de datos relacionales*.

Palmero, M. A., & Grass, N. S. (2018). Review of conceptual elements at representation of software reference. *Revista Cubana de Ciencias Informáticas*, 13(1), 143-157.

Parnas, D. L. (1979). *Designing software for ease of extension and contraction*.

Pascual, J. R. (Diciembre de 01 de 2019). Arquitectura de Servicios Web. *[Imagen]*. Obtenido de <https://www.disrupciontecnologica.com/arquitectura-de-servicios-web/>

Pastor R, B. F. (2019). Población y muestra. Pueblo continente. *Pueblo continente*, 30(1), 245-247.

Peltz, C. (2003). Orchestration versus choreography. *Imagen*.

Peltz, C. (2003). *Web services orchestration and choreography*. *Computer* (Vol. 36).

Platt, M. (2002). *Microsoft Architecture Overview*. Recuperado de <https://www.dragon1.com/downloads/microsoft%20architecture%20overview.pdf>.

- Raphael, A., & Zott, C. (2001). Value creation in e-business. *strategic management journal*, 22(6-7), 493-520. Recuperado el 08 de 08 de 2020, de Recuperado de <https://doi.org/10.1002/smj.187>
- Rational Software Corporation. (Febrero de 2001). *Software Architecture Document*. Obtenido de <https://www.ecs.csun.edu/~rlingard/COMP684/Example2SoftArch.htm>
- Real Academia Española. (2001). Madrid, España. Recuperado el 07 de 08 de 2020, de <https://www.rae.es/drae2001/proceso>
- Reynoso, C. B. (2004). *Introducción a la Arquitectura de Software*. Buenos Aires, Argentina.
- Richards, M. (2015). *Software Architecture Patterns* (Vol. 4). 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Incorporated.
- Rodríguez Zamuria, D. (2017). *Gestión por Procesos de Negocio (BPM)- (Maestría en Gestión de Tecnologías de Información y Comunicación)*. Managua, Nicaragua. Obtenido de Recuperado de <https://core.ac.uk/download/pdf/250143664.pdf>
- Rozanski, N., & Woods, E. (2005). *Software Systems Architecture: Working With Stakeholders* .
- Sandeep , C., & Webber, J. (2004). *Developing enterprise Web services: an architect's guide*. Prentice Hall Professional.
- Sanders , G., & Courtney, J. F. (1985). A field study of organizational factors influencing DSS success . *MIS quarterly*, 77-93.
- Sanders, P. A., & Hamilton, J. A. (2003). *A Process for Interoperability*. Joint Command.

- Software Engineering Institute. (Setiembre de 2012). Software Architecture in Practice, Third Edition. *Software Architecture in Practice, Third Edition*. Pittsburgh, Estados Unidos.
- Tsui , F. F., Karam , O., & Bernal, B. (2016). Essentials of software engineering. *Jones & Bartlett Learning*,.
- Van Vliet, H. (2008). *Software engineering: principles and practice* (Tercera ed.). Hoboken, NJ: John Wiley & Sons.
- Veintimilla Reyes, J., Espinoza Mejía, M., & Cisneros, F. (Octubre de 2014). Estudio y seleccion de una arquitectura orientada a servicios (SOA) que permita la integracion de sistemas informaticos legados. *Revista Tecnológica ESPOL*, 27(1), 14-21. Obtenido de Recuperado de https://www.researchgate.net/publication/267034826_Estudio_y_seleccion_de_una_arquitectura_orientada_a_servicios_SOA_que_permita_la_integracion_de_sistemas_informaticos_legados
- Vidich , A., & Shapiro, G. (2017). A comparison of participant observation and survey data. *In Sociological Methods*, 512-522.
- Webster, J., & Watson, R. (Junio de 2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. *Management Information Systems Research Center*, 26(2), págs. 13-23. Obtenido de Recuperado de <https://www.jstor.org/stable/4132319>
- Weerawarana , S., Curbera , F., Leym, F., Storey , T., & Ferguson, D. F. (2005). *Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more*. Prentice Hall PTR.

- Wei-Tek , T., Yamin Wang, R., & Dong, C. (2002). Extending WSDL to facilitate Web services testing. *7th IEEE International Symposium on High Assurance Systems Engineering, 2002. Proceedings.*, (págs. 171-172). Tokyo, Japón. doi: 10.1109/HASE.2002.1173119
- Wynekoop, J., Senn, J. A., & Conger, S. A. (1992). The implementation of CASE tools: An innovation diffusion approach. *In Proceedings of the IFIP WG8. 2 Working Conference on The Impact of Computer Supported Technologies in Information Systems Development*, (págs. 25-41).

Anexos

Anexo 1. Plantilla de encuesta ejecutada.

La siguiente encuesta tiene como finalidad recolectar información importante, para el proyecto que se elabora en la Sección de Seguridad Institucional de la Universidad Nacional de Costa Rica – Heredia. La información recopilada será exclusivamente para dicho fin. INSTRUCCIONES Conteste las siguientes interrogantes con responsabilidad y honestidad de acuerdo con las experiencias que ha vivido como funcionario de esta prestigiosa Universidad. Marque con una “X” su respuesta o responda con sus propias palabras cuando fuese necesario. SECCIÓN 01. DEMOGRAFÍA. 1- Cuál es el puesto en el que se desempeña para la Universidad Nacional. Marque con una “X”. - a- Director/ara o Coordinador/ara de Sección de Seguridad Institucional - b- Auxiliar de Servicios Secretariales - c- Profesional ejecutiva - d- Oficial de Seguridad - e- Operador/ ara de acceso vehicular - f- Otro. Detallar: _____. 2- Seleccione la Unidad, Instituto o Escuela a la que pertenece. Marque con una “X”. - a- Escuela de Música - b- Escuela de Arte Escénico - c- Escuela de Danza - d- Instituto IRET - e- CIDE - f- Centro de Monitoreo - g- Sección de Seguridad. - h- Otro. Especificar: _____.	3- Cuántos años lleva elaborando en la Universidad Nacional de Costa Rica. - a- Más de 1 año - b- Menos de 5 años - c- Menos de 10 años - d- 10 años adelante SECCIÓN 02. PERCEPCION. 1- Usted como funcionario/ria de la Universidad Nacional de Costa Rica conoce el modelo actual o el mecanismo utilizado por la Sección de Seguridad Institucional de la Universidad Nacional de Costa Rica, para la realización de solicitud de reserva o permiso de ingreso a parques Institucionales y otras áreas. Señale con una “X”. Si su respuesta es no, la encuesta a finalizado. - a- Sí, si lo conozco - b- No, no lo conozco. 2- Usted considera que el sistema actual o mecanismo utilizado por la Sección de Seguridad Institucional de la Universidad Nacional de Costa Rica, para la realización de solicitud de reserva o permiso de ingreso a parques Institucionales y otras áreas, se ajusta a las necesidades de los funcionarios/rias. Señale con una “X”. Si su respuesta es “Si”, pase a la pregunta n°4. - a- Sí - b- No 3- Considera usted que es importante y adecuado que la Sección de Seguridad Institucional, de la Universidad Nacional de Costa Rica, cuente con otro mecanismo para el proceso de reserva o permisos de ingreso a parques institucionales y a otras áreas. - a- Sí
--	--

Anexo 1-a: Plantilla de encuesta ejecutada.

b- No 4- Considera usted que el sistema actual o mecanismo utilizado por la Sección de Seguridad Institucional, de la Universidad Nacional de Costa Rica, para el proceso de reserva o permisos de ingreso a parqueos institucionales y a otras áreas, es eficiente y eficaz para las siguientes poblaciones: funcionarios, estudiantes y personas externas. a- Sí, ¿Por qué? ❖ Funcionarios de la Universidad: _____ _____ ❖ Estudiantes de la Universidad Nacional: _____ _____ ❖ Personas Externas a la Universidad Nacional: _____ _____ b- No. ¿Por qué? _____ _____ SECCION 03. FLUJO DE TRABAJO DE SOLICITUD DE RESERVA O PERMISO DE INGRESO A PARQUEOS INSTITUCIONALES Y OTRAS ÁREAS. 1- Alguna vez usted ha recibido capacitación por parte de la Sección de Seguridad Institucional de la Universidad Nacional de Costa Rica sobre el proceso de solicitud de reserva o permiso de ingreso a parqueos institucionales y otras áreas. Señale con una "X". a- Nunca b- Casi nunca c- Ocasionalmente d- Casi siempre e- Siempre	2- En pocas palabras, Describa el flujo de trabajo que realiza a la Sección de Seguridad Institucional de la Universidad Nacional, sobre la solicitud de reserva o permiso de ingreso a parqueos institucionales y otras áreas. a- _____ _____ _____ 3- Cuál herramienta usted ha utilizado para realizar una solicitud de reserva o permiso de ingreso a parqueos institucionales y otras áreas a la Sección de Seguridad Institucional de la Universidad Nacional. Señale con una "X". a- Correo Electrónico b- Sistema Web. c- Llamada telefónica. d- Trámites Físicos e- Otros (Especifique: _____). 4. Según su experiencia, ¿Cuánto ha sido la duración para hacer una solicitud a la Sección de Seguridad Institucional de la Universidad Nacional, de reserva o permiso de ingreso a parqueos institucionales y otras áreas? Señale con una "X". a- Unas horas b- Un día c- Una semana d- Más de 1 semana
---	--

Anexo 1-b: Plantilla de encuesta ejecutada.

5. Con cuánta frecuencia ha presentado problemas al realizar una solicitud de reserva o permiso de ingreso a parqueos institucionales y otras áreas de la Sección de Seguridad Institucional de la Universidad Nacional. Señale con una "X". a- Nunca b- Casi nunca c- Ocasionalmente d- Casi siempre e- Siempre 6. Cuánta frecuencia solicita este servicio "Reserva o permiso de ingreso a parqueos institucionales y otras áreas a la Sección de Seguridad Institucional de la Universidad Nacional. Señale con una "X". a- 1 vez a la semana. b- Más de 1 vez a la semana. c- 1 vez al mes d- Mas de 1 vez al mes e- En ocasiones esporádicas. 7. Cómo funcionario de la Universidad Nacional de Costa Rica. Alguna vez has presentado algún problema o inconveniente, solicitando el servicio de "Reserva o permiso de ingreso a parqueos institucionales y otras áreas a la Sección de Seguridad Institucional de la Universidad Nacional. Señale con una "X". Si tu pregunta es negativa, la encuesta ha finalizado. a- Sí. b- No.	8. Cómo funcionario de la Universidad Nacional de Costa Rica. Qué tipo de problema, se le ha presentado, solicitando el servicio de "Reserva o permiso de ingreso a parqueos institucionales y otras áreas a la Sección de Seguridad Institucional de la Universidad Nacional. Señale con una "X". a- Comunicación con la Sección de Seguridad b- Documentos mal estructurados c- Problemas de software d- Falta de claridad del proceso e- Falta de conocimiento de cómo se hace el proceso. f- Otros (Especifique: _____).
--	--

Anexo 1-c: Plantilla de encuesta ejecutada.

Esta sección es exclusivamente para funcionarios/as de la Sección de Seguridad Institucional de la Universidad Nacional de Costa Rica.

SECCION 04. OTROS FLUJOS DE TRABAJO RELACIONADOS CON LA SECCIÓN DE SEGURIDAD.

1- Recibe capacitación sobre el proceso para “Reserva o permiso de ingreso a parqueos institucionales y otras áreas a la Sección de Seguridad Institucional de la Universidad Nacional. Señale con una “X”.

f- Nunca
g- Casi siempre
h- De vez en cuando
i- Casi siempre
j- Siempre

2- ¿Considera usted necesario mejorar los procesos en la Sección de Seguridad de la Universidad Nacional para agilizar los servicios relacionados con solicitudes de personas, espacios y vehículos?

a-Sí. ¿Porqué?
b-No. ¿Porqué?

La encuesta ha concluido. Muchas gracias por su participación.

Página 7 de 7

Anexo 2. Respuesta de entrevistado sobre problema al realizar solicitud al SSI.

8. Cómo funcionario de la Universidad Nacional de Costa Rica. Qué tipo de problema, se le ha presentado, solicitando el servicio de "Reserva o permiso de ingreso a parqueos institucionales y otras áreas a la Sección de Seguridad Institucional de la Universidad Nacional. Señale con una "X".

a- Comunicación con la Sección de Seguridad

b- Documentos mal estructurados

c- Problemas de software

☒ Falta de claridad del proceso

e- Falta de conocimiento de cómo se hace el proceso.

☒ Otros (Especifique: No puede apertar en espacio para salir a hacer un mandado de la unidad y regresar.

Anexo 3. Respuesta a entrevistado sobre mecanismo utilizado por la SSI.

4- Considera usted que el sistema actual o mecanismo utilizado por la Sección de Seguridad Institucional, de la Universidad Nacional de Costa Rica, para el proceso de reserva o permisos de ingreso a parqueos institucionales y a otras áreas, es eficiente y eficaz para las siguientes poblaciones: funcionarios, estudiantes y personas externas.

a- Sí, ¿Por qué?

- ❖ Funcionarios de la Universidad: _____
- ❖ Estudiantes de la Universidad Nacional: _____
- ❖ Personas Externas a la Universidad Nacional: _____

b- No, ¿Por qué? a veces no logran ver el correo a tiempo y cuando la persona va a ingresar no lo dejan, además la parte académica no tiene claridad de que son 3 días hábiles de tiempo para hacer la solicitud ante la sección de seguridad

SECCION 03. FLUJO DE TRABAJO DE SOLICITUD DE RESERVA O PERMISO DE INGRESO A PARQUEOS INSTITUCIONALES Y OTRAS ÁREAS.

1- Alguna vez usted ha recibido capacitación por parte de la Sección de Seguridad Institucional de la Universidad Nacional de Costa Rica sobre el proceso de solicitud de reserva o permiso de ingreso a parqueos institucionales y otras áreas. Señale con una "X".

- ☒ a- Nunca
- b- Casi nunca
- c- Ocasionalmente
- d- Casi siempre
- e- Siempre

Página 3 de 7

Anexo 4. Respuesta dos entrevistado sobre mecanismo utilizado por la SSI.

4- Considera usted que el sistema actual o mecanismo utilizad por la Sección de Seguridad Institucional, de la Universidad Nacional de Costa Rica, para el proceso de reserva o permisos de ingreso a parqueos institucionales y a otras áreas, es eficiente y eficaz para las siguientes poblaciones: funcionarios, estudiantes y personas externas.

a- Si, ¿Por qué?

- ❖ Funcionarios de la Universidad: _____
- ❖ Estudiantes de la Universidad Nacional: _____
- ❖ Personas Externas a la Universidad Nacional: _____

☒ No. ¿Por qué? Muchas veces es difícil porque aún enviando correo para ingreso, la funcionarios tienen problemas a la hora de ingresar.

SECCION 03. FLUJO DE TRABAJO DE SOLICITUD DE RESERVA O PERMISO DE INGRESO A PARQUEOS INSTITUCIONALES Y OTRAS ÁREAS.

1- Alguna vez usted ha recibido capacitación por parte de la Sección de Seguridad Institucional de la Universidad Nacional de Costa Rica sobre el proceso de solicitud de reserva o permiso de ingreso a parqueos institucionales y otras áreas. Señale con una "X".

- ☒ a- Nunca
- b- Casi nunca
- c- Ocasionalmente
- d- Casi siempre
- e- Siempre

Página 3 de 7

Anexo 5. Respuesta uno de entrevistado describiendo flujo de trabajo de la SSI.

2- En pocas palabras, Describa el flujo de trabajo que realiza a la Sección de Seguridad Institucional de la Universidad Nacional, sobre la solicitud de reserva o permiso de ingreso a parqueos institucionales y otras áreas.

a- Envío correo con la información y se espera respuesta de la Sección de Seguridad y se le copia a la persona que solicita el ~~correo~~ permiso esto con 3 días hábiles de tiempo.

3-Cuál herramienta usted ha utilizado para realizar una solicitud de reserva o permiso de ingreso a parqueos institucionales y otras áreas a la Sección de Seguridad Institucional de la Universidad Nacional. Señale con una "X".

☒ a- Correo Electrónico

☐ b- Sistema Web.

☒ c- Llamada telefónica.

☒ d- Trámites Físicos

e- Otros (Especifique: _____).

4. Según su experiencia, ¿Cuánto ha sido la duración para hacer una solicitud a la Sección de Seguridad Institucional de la Universidad Nacional, de reserva o permiso de ingreso a parqueos institucionales y otras áreas? Señale con una "X".

a- Unas horas

b- Un día

☒ c- Una semana

d- Más de 1 semana

Página 4 de 7

Anexo 6. Respuesta dos de entrevistado describiendo flujo de trabajo de la SSI.

2- En pocas palabras, Describa el flujo de trabajo que realiza a la Sección de Seguridad Institucional de la Universidad Nacional, sobre la solicitud de reserva o permiso de ingreso a parques institucionales y otras áreas.

a- Se envía un correo a seguridad@unsa.cr con los nombres y cédulas de los visitantes, asunto y lugar donde se realizará la actividad.

3- Cuál herramienta usted ha utilizado para realizar una solicitud de reserva o permiso de ingreso a parques institucionales y otras áreas a la Sección de Seguridad Institucional de la Universidad Nacional. Señale con una "X".

☒ Correo Electrónico

b- Sistema Web.

c- Llamada telefónica.

d- Trámites Físicos

e- Otros (Especifique: _____).

4. Según su experiencia, ¿Cuánto ha sido la duración para hacer una solicitud a la Sección de Seguridad Institucional de la Universidad Nacional, de reserva o permiso de ingreso a parques institucionales y otras áreas? Señale con una "X".

a- Unas horas

☒ b- Un día

c- Una semana

d- Más de 1 semana

Página 4 de 7

Anexo 7. Primer Correo a Arquitectos de Evaluación.

Estimado/o usuario/a:

Esperando se encuentre bien y deseándole éxito en sus funciones;

Nosotros; Yendry Hernández Montezuma e Ignacio Ávalos Bonilla, estudiantes de Licenciatura en Sistemas de Información de la Universidad Nacional, estamos realizando un proyecto de tesis denominado *"Propuesta de documentación de Arquitectura Integral de Sistema para la Sección de Seguridad de la Sede Central de la Universidad Nacional"*.

Usted muy amablemente se ha ofrecido a colaborar con nosotros en el proceso de evaluación de la propuesta de arquitectura siguiendo un procedimiento especial que hemos diseñado para efectos de este proyecto de graduación. El proceso está sistematizado por medio de una aplicación web en la cual la evaluación será conducida a través de una serie de pasos que se le irán indicando a usted de forma oportuna. Apreciamos mucho su participación para brindarnos su valioso conocimiento.

La valoración de esta propuesta de arquitectura Integral está basada en el método ATAM (Método de Análisis de Compensación de Arquitectura).

Este proceso se va a dividir en tres bloques, usted debe completar los pasos 1,2,3,4. Luego los pasos 5 y 6 y finalmente el paso 7 que sería la presentación de los resultados.

Entre cada bloque va a ver un tiempo de 4 días hábiles. A partir del cuarto día le llegará un aviso para proseguir y los pasos siguientes (5,6) se activarán en ese momento.

Agradecemos desde ya su colaboración en el proceso en general, sobre todo en la ejecución de los pasos conforme les vayamos indicando.

A continuación, usted puede iniciar el proceso accediendo a la aplicación por medio de la siguiente información de acceso.

Enlace del sitio: <http://evaluacionarquitecturas.atwebpages.com/>

Usuario:

Contraseña:

Nuevamente, muchas gracias por su colaboración;

Atentamente.

Yendry Hernández Montezuma e Ignacio Ávalos Bonilla
Estudiantes de Licenciatura en Sistemas de Información-UNA

Anexo 8. Segundo Correo a Arquitectos de Evaluación.

Estimados usuarios/as:

Esperando se encuentre bien y deseándole éxito en sus funciones nuevamente les saludamos.

Este correo tiene la finalidad de indicarle a usted que puede proseguir con los pasos 5 y 6 de la evaluación de la arquitectura propuesta.

Nos permitimos recordarle la información de enlace para que pueda proseguir el proceso:

Enlace del sitio: <http://evaluacionarquitecturas.atwebpages.com/>

Usuario:

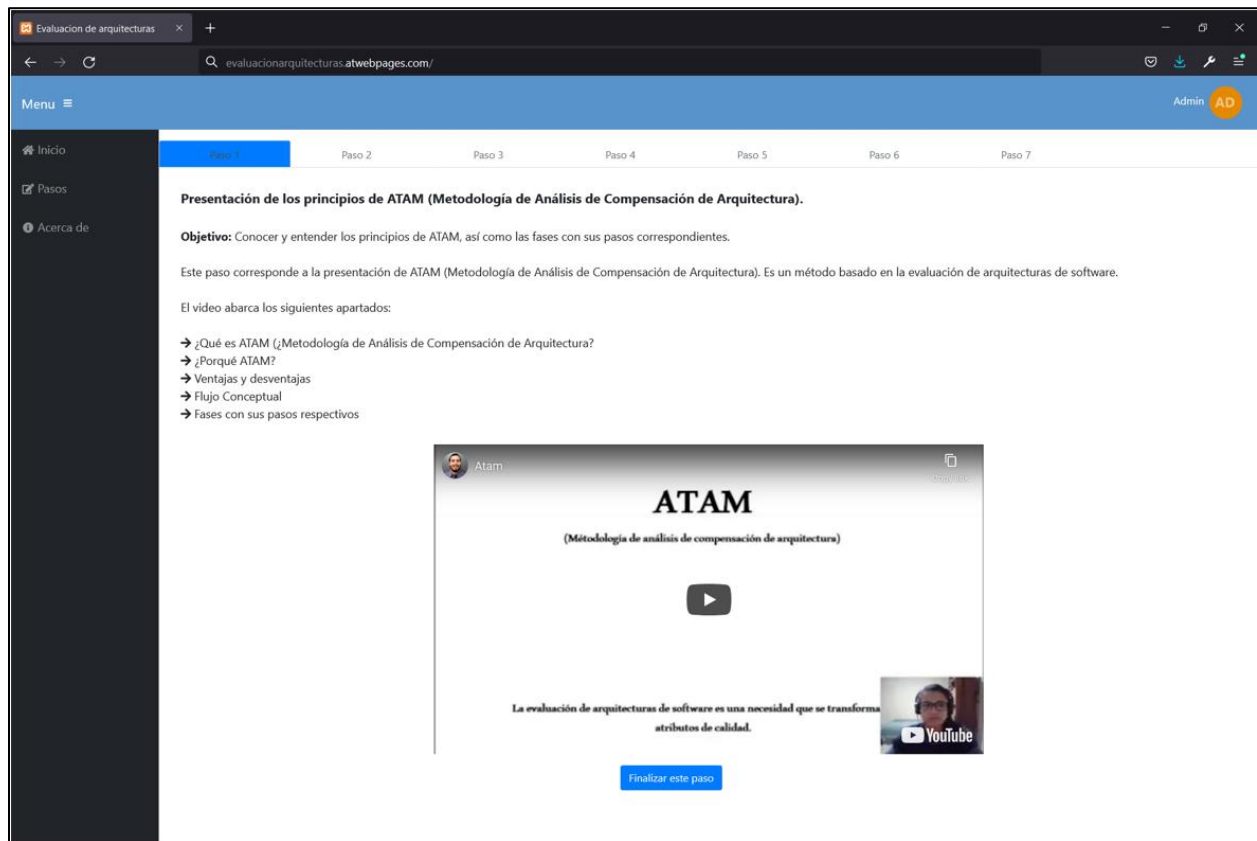
Contraseña:

Nuevamente, muchas gracias por su colaboración;

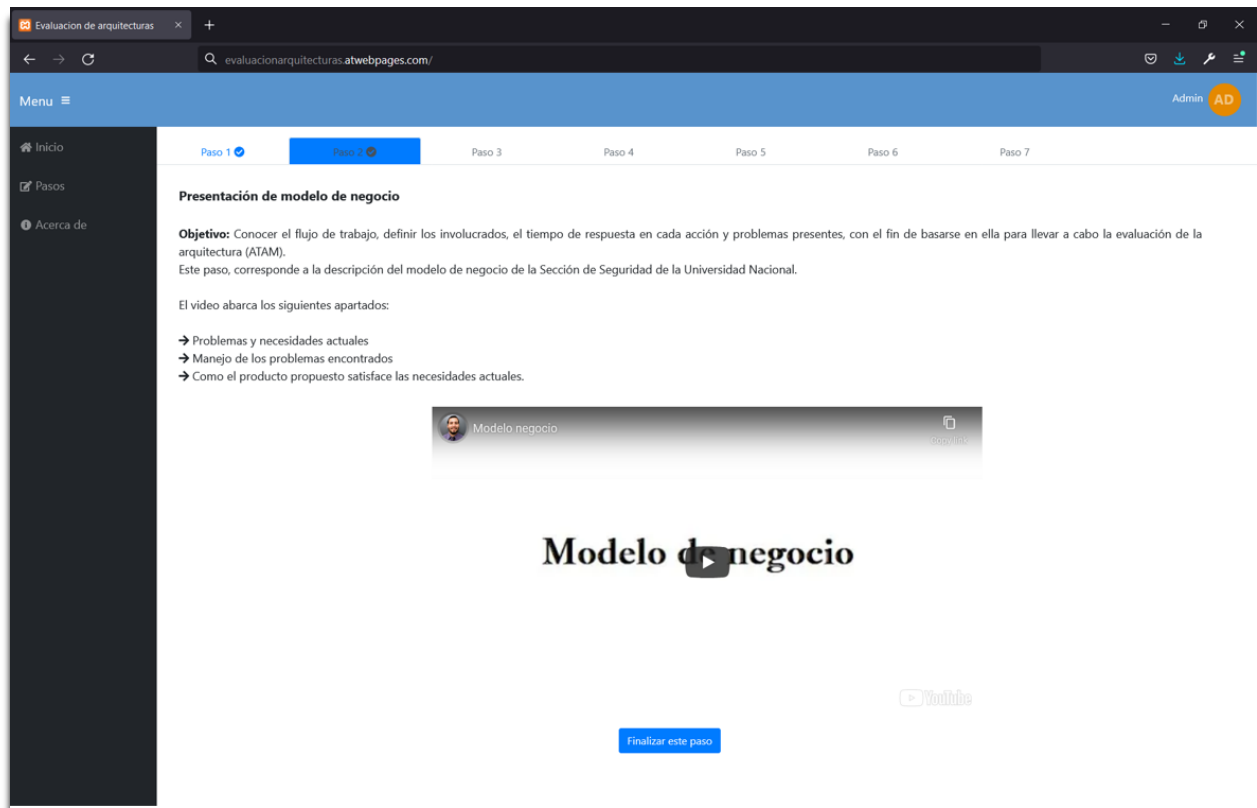
Atentamente.

Yendry Hernández Montezuma e Ignacio Ávalos Bonilla
Estudiantes de Licenciatura en Sistemas de Información-UNA

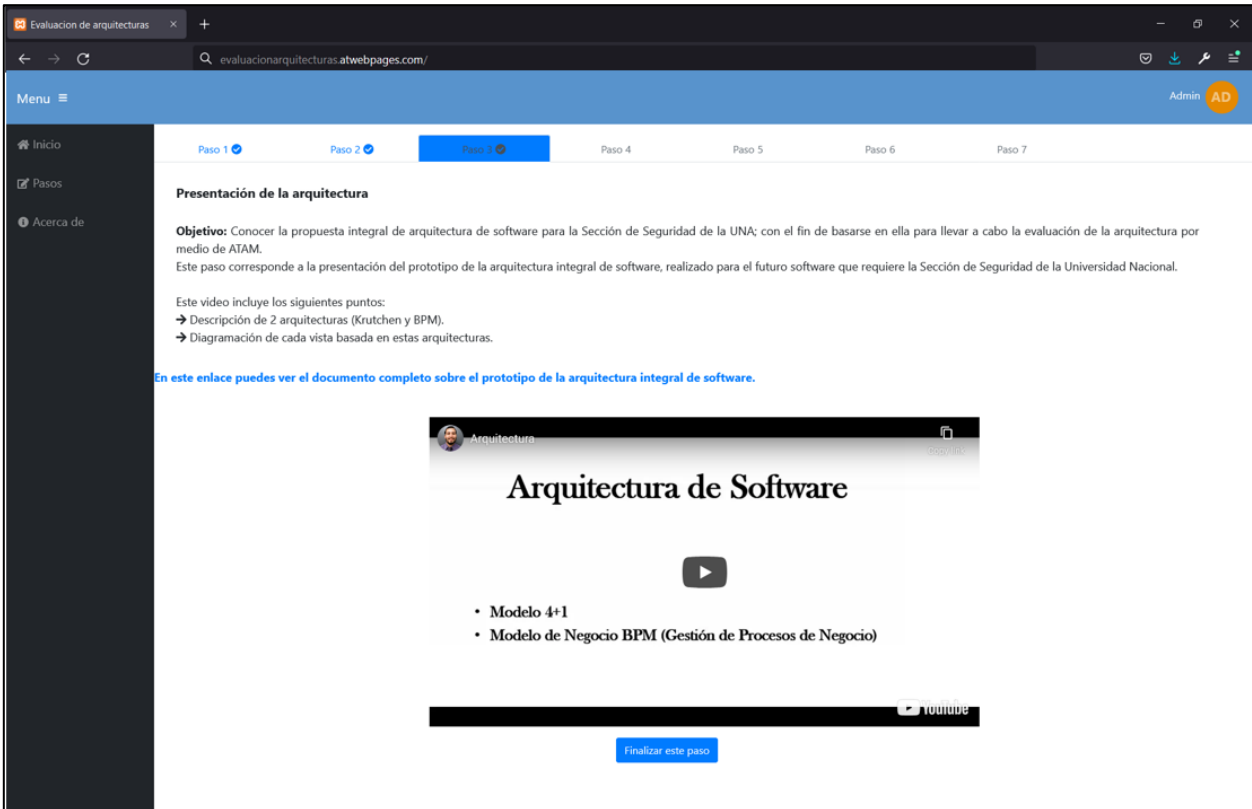
Anexo 9. Captura de Pantalla de Sitio Web sobre Presentación de ATAM.



Anexo 10. Captura Pantalla de Sitio Web sobre Modelo de Negocio.



Anexo 11. Captura de Pantalla de Sitio Web sobre Presentación de Arquitectura.



Anexo 12. Captura de Pantalla de Sitio Web sobre Formulario de Encuesta.

The screenshot shows a web browser window displaying a survey form. The browser's address bar shows the URL 'https://www.google.com/surveys/...'. The survey is titled 'Identificación de atributos de calidad' and is part of a 7-step process. The current step, 'Paso 3', is highlighted in blue. The survey is designed to capture and analyze quality attributes through a questionnaire. The form includes a list of questions with radio button options and text input fields. The questions are numbered 1 through 23. The form is displayed in a dark-themed browser window.

Identificación de atributos de calidad

Objetivo: Capturar y analizar los resultados acerca de los atributos de calidad. Con el fin de analizar la arquitectura y generar escenarios. Este paso corresponde a la extracción de resultados por medio de una encuesta con el fin de posteriormente ser evaluados y analizados, para la generación de los atributos de calidad y los escenarios.

1. Rango de edad:

- ☐ 25 años o 30 años
- ☐ 30 años a 35 años
- ☐ 35 años a 40 años
- ☐ 40 a 45 años
- ☐ Más de 45 años

2. Años de experiencia profesional:

- ☐ 1 a 5 años
- ☐ 5 a 10 años
- ☐ 10 a 15 años
- ☐ 15 a 20 años
- ☐ 20 a 25 años
- ☐ 25 años en adelante

3. Años de experiencia en arquitectura de software o tecnologías de la información:

- ☐ 1 a 5 años
- ☐ 5 a 10 años
- ☐ 10 años en adelante

4. Nivel académico:

- ☐ Bachiller
- ☐ Licenciado
- ☐ Maestr
- ☐ Doctorado

5. Si se usa redundancia en la arquitectura, ¿cómo elementos que fallen a la misma vez? La redundancia arquitectónica es el uso de elementos o componentes repetidos que permitan una rápida corrección de posibles fallos.

6. ¿Considera usted que la redundancia es una buena práctica para solucionar posibles fallos? (Marque)

7. Cuando se utiliza la redundancia se puede crear escenarios malos elementos para activar una falla, ¿el tiempo que se tarda en resolver las fallos de un componente redundante puede ser menor o mayor? ¿Qué herramientas se pueden pensar?

8. Como arquitecto o desarrollador de software, ¿considera usted que se debería agregar alguna nueva funcionalidad al futuro sistema? ¿Cuál? o ¿por qué?

9. ¿Qué se debe tomar en cuenta para que el sistema sea más seguro en el futuro?

10. De acuerdo a su experiencia, ¿el realizar un control de una funcionalidad del sistema que con frecuencia pueden surgir?

11. ¿La posibilidad que los componentes, conexiones e interfaces se vean afectados al eliminar datos obsoletos del repositorio de datos? Un repositorio de datos es un dispositivo centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos relacionados.

12. ¿Cuáles formas estándar usted que influyen en la reducción de tiempo para la implementación de una nueva funcionalidad al sistema?

13. ¿Cómo considera usted que se debe de identificar los fallos en el sistema?

14. ¿Se puede identificar tanto la falla de seguridad activa como la pasiva? Entendase seguridad activa, la que podemos medir y la seguridad pasiva, la que se realiza al recibir un ataque.

15. ¿Que atributo de calidad encuentra usted con la utilización de un sistema informático en lugar del uso de papel?

16. ¿Se está mejor definiendo mejor el sistema seguridad en un sistema?

17. Una encuesta generada dentro de los mecanismos respondidos, into acción se refiere a:

- ☐ Seguridad
- ☐ Rendimiento
- ☐ Interoperabilidad
- ☐ Infidencia

18. La mejora en los procesos en la toma de decisiones se refiere a:

- ☐ Interoperabilidad
- ☐ Infidencia
- ☐ Rendimiento
- ☐ Interoperabilidad
- ☐ Rendimiento

19. La disponibilidad de un sistema debe ser de un 99,99% de los usuarios que un usuario intente accederlo, la siguiente sería correspondiente a:

- ☐ Disponibilidad
- ☐ Interoperabilidad
- ☐ Interoperabilidad
- ☐ Disponibilidad

20. ¿Que atributo de calidad encuentra usted al acceder desde cualquier dispositivo con una excelente interfaz gráfica?

21. Con sus propias palabras y siendo breve, responda, ¿qué es para usted un sistema robusto de calidad?

22. Muestra alguna propiedad o estilo arquitectónico, controlado durante la presentación de la Descripción de la Arquitectura de Software (SAD) (Software Architecture Description).

[Volver Encuesta y Realizar Datos](#)

Anexo 13. Captura de Pantalla de Sitio Web sobre árbol d Utilidad y Escenarios.

Evaluación de arquitecturas

evaluacionarquitecturas.atwebpages.com/

Admin AD

Inicio

Pasos

Acerca de

Paso 1

Paso 2

Paso 3

Paso 4

Paso 5

Paso 6

Paso 7

Generación de los atributos del árbol de calidad

Objetivo: Buscar las debilidades conocidas de la propuesta, como los puntos sensibles y los puntos de compensación.

En este paso los atributos de calidad son representados en forma de escenarios. Con esto se pretende comprender los requisitos no funcionales identificados.

Item	Atributos de calidad	Requisitos específicos de atributos de calidad	Reglas (plantillas de calidad)	Escenarios
Árbol de Utilidad	Eficacia	Cantidad de control de calidad	RA1	Control de la calidad de la propuesta
		Calidad de control de calidad	RA2	Adaptación de la propuesta a los requisitos de calidad de la propuesta
	Eficacia	Cantidad de control de calidad	RA3	Adaptación de la propuesta a los requisitos de calidad de la propuesta
		Calidad de control de calidad	RA4	Adaptación de la propuesta a los requisitos de calidad de la propuesta
	Eficacia	Cantidad de control de calidad	RA5	Adaptación de la propuesta a los requisitos de calidad de la propuesta
		Calidad de control de calidad	RA6	Adaptación de la propuesta a los requisitos de calidad de la propuesta
	Eficacia	Cantidad de control de calidad	RA7	Adaptación de la propuesta a los requisitos de calidad de la propuesta
		Calidad de control de calidad	RA8	Adaptación de la propuesta a los requisitos de calidad de la propuesta
	Eficacia	Cantidad de control de calidad	RA9	Adaptación de la propuesta a los requisitos de calidad de la propuesta
		Calidad de control de calidad	RA10	Adaptación de la propuesta a los requisitos de calidad de la propuesta

Este paso ya fue finalizado!

Anexo 14. Captura de Pantalla de Sitio Web sobre votación de Escenarios.

Evaluación de arquitecturas

← → ↻

evaluacionarquitecturas.atwebpages.com/

Admin AD

Inicio

Pasos

Acerca de

Paso 1

Paso 2

Paso 3

Paso 4

Paso 5

Paso 6

Paso 7

Revision atributos de calidad

Objetivo: Evaluar los atributos de calidad por medio de

Este paso corresponde a la votación de los atributos de

al dar clic sobre los enlaces de la columna "Escenario" c

los votos emitidos.

Atributo de calidad	Escenario
Modificabilidad	Cambio de la l
Modificabilidad	Actualizaciones
Modificabilidad	El sistema debe
Disponibilidad	Remitir despu
Disponibilidad	Detectar y recup
Disponibilidad	Detectar y recup
Disponibilidad	Prevenir almacen
Rendimiento	La latencia con
Rendimiento	Los cambios en los datos deben verse reflejados lo más pronto.
Rendimiento	El promedio de rendimiento entre el servidor y DS debe ser el más bajo y estable posible
Seguridad	Las transacciones deben ser 99% seguras.
Seguridad	La transparencia entre el usuario y el sistema debe ser la más alta.
Seguridad	La integridad debe ser precisa, con ello la información almacenada en una base de
Interoperabilidad	datos seguirá siendo completa, precisa y fiable.
Interoperabilidad	El subsistema genera información para otros sistemas
Interoperabilidad	El sistema es capaz de recibir información de otro sistemas

Finalizar este paso

Escenario

Análisis de enfoque arquitectónico				
Escenario #: 3	Detectar y recuperarse de la falla de hardware de una CPU primaria.			
Atributo:	Disponibilidad. (Fallas de H/W CORS)			
Ambiente:	En operaciones normales.			
Estímulo:	Fallas en el CPU			
Respuesta:	La disponibilidad de respuesta debe ser de manera inmediata.			
Decisiones arquitectónicas:	Sensibilidad	Compensación	Riesgo	No Riesgo
Respaldo del CPU	S1		S1	
No tener un canal de datos de respaldo	S2	C1	S2	
Watchdog	S3			
Heartbeat	S4			
Razonamiento	S1: Detectar fallas en 2 segundos basado en frecuencias de watchdog y Heartbeat. C1: Contar con una canal de respaldo S1: Garantiza que no haya fallas mediante el uso de hardware y sistema operativo diferentes. S2: El requisito de disponibilidad puede estar en riesgo debido a la falta de un canal de respaldo.			

Paso importante

Sin importancia

Anexo 15. Captura de Pantalla de Sitio Web sobre Menú principal.

The screenshot shows a web browser window with the address bar displaying 'evaluacionarquitecturas.atwebpages.com/'. The page has a dark blue header with a 'Menu' button on the left and an 'Admin' profile icon on the right. The main content area is divided into two sections. The top section features a large banner with the title 'Propuesta de Documentación de Arquitectura Integral de Sistema para la Sección de Seguridad de la Sede Central de la Universidad Nacional.' and the subtitle 'INFORMÁTICA CON ÉNFASIS EN SISTEMAS DE INFORMACIÓN'. To the right of the text is an illustration of a person working on a laptop, surrounded by various digital devices and icons. Below the banner, the 'Descripción' section explains the purpose of the website, which is to support a project in Informatics with a focus on Information Systems. It mentions the 'Propuesta de Documentación de Arquitectura Integral de Sistemas para la Sección de Seguridad de la Sede Central de la Universidad Nacional.' and states that the website was created to facilitate communication and evaluation. The 'Explicación del proceso de evaluación' section provides instructions on how to use the website, including a link to the 'Pasos' page. Below this is a table with 7 steps, each with a description of the task and the type of output (e.g., video, form, document). At the bottom of the table is a blue button labeled 'Ir a la evaluación'.

Menu

Admin AD

Inicio

Pasos

Acerca de

Propuesta de Documentación de Arquitectura Integral de Sistema para la Sección de Seguridad de la Sede Central de la Universidad Nacional.

INFORMÁTICA CON ÉNFASIS EN SISTEMAS DE INFORMACIÓN

Descripción

Este sitio web nació con la finalidad de apoyar nuestro proyecto de Licenciatura en Informática con énfasis en Sistemas de Información llamado "Propuesta de Documentación de Arquitectura Integral de Sistemas para la Sección de Seguridad de la Sede Central de la Universidad Nacional." Esta documentación será evaluada por medio de una metodología llamada ATAM (Método de Análisis de Compensación de Arquitectura). La segunda finalidad de este sitio es poder establecer una comunicación con los involucrados del proyecto y así poder llevar a cabo dicha evaluación. Como es de conocimiento de todos y todas, la pandemia del VIRUS SARS-CoV-2, ha cambiado la forma de comunicación; por lo que nos impulsó a crear una comunicación virtual e innovadora.

Al final esto es parte del ser estudiante de la Universidad Nacional, con el fin de ir adquiriendo y desarrollando nuestras capacidades

Explicación del proceso de evaluación

Para iniciar con la evaluación diríjase a la pestaña [Pasos](#), la cual consta de 7 instrucciones detalladas. Para una mayor comprensión, se muestra una tabla con las instrucciones y la descripción de lo que se debe de realizar en cada paso.

Pasos	Descripción
Paso 1	Corresponde a la presentación de ATAM (Método de Análisis de Compensación de Arquitectura.) (Video - presentación)
Paso 2	Corresponde a la presentación del Modelo de Negocio. (Video - presentación)
Paso 3	Corresponde a la presentación de la arquitectura Integral de Software.(Video - presentación)
Paso 4	Corresponde a llenar el formulario en línea, con el fin de identificar los atributos arquitecturales de calidad. (Formulario en línea)
Paso 5	Corresponde a la generación del Árbol de Utilidad (Documento de texto)
Paso 6	Corresponde a la revisión y votación de los atributos de calidad (Escala de Likert).
Paso 7	Corresponde al último paso de la evaluación y comprende la presentación y correlación de los resultados obtenidos. (Documento reporte evaluación)

[Ir a la evaluación](#)

Anexo 16. Captura de Pantalla de Sitio Web sobre Nosotros.

