

**UNIVERSIDAD CENTRAL**  
**VICERRECTORÍA ACADÉMICA**  
**ESCUELA DE INGENIERÍA INFORMÁTICA**

**Implementación de Microservicios en el Ambiente de Desarrollo  
para la Gestión y Consulta de Audiencias Judiciales con  
Almacenamiento en el Servidor Isilon Utilizando el Protocolo S3  
en el Poder Judicial de Costa Rica.**

**MODALIDAD DE TESIS PARA OPTAR POR EL GRADO DE LICENCIATURA EN  
INGENIERÍA INFORMÁTICA CON ÉNFASIS EN GERENCIA INFORMÁTICA**

**ING. SIVIANI CASTILLO RODRÍGUEZ**

**LIC. MAURICIO TORRES CERDAS**

**SEDE CENTRAL SAN JOSÉ, COSTA RICA**

**NOVIEMBRE, 2024**

## Índice

<b>Índice .....</b>	<b>II</b>
<b>Tablas.....</b>	<b>VII</b>
<b>Figuras.....</b>	<b>VIII</b>
<b>Dedicatoria y agradecimiento.....</b>	<b>XI</b>
<b>Resumen.....</b>	<b>XII</b>
<b>Capítulo 1: Problema.....</b>	<b>1</b>
<i>Planteamiento del Problema .....</i>	<i>2</i>
<i>Variables del Problema .....</i>	<i>3</i>
<i>Causas del Problema.....</i>	<i>3</i>
Gestión Inadecuada de Datos .....	3
Deficiencias en el Servidor Isilon .....	4
Protocolos y Medidas de Seguridad Inadecuados .....	4
Problemas de Infraestructura de Red .....	4
<i>Efectos del Problema .....</i>	<i>4</i>
Rendimiento Deficiente del Sistema.....	4
Insatisfacción de los Usuarios .....	4
Riesgos de Seguridad .....	5
Carga Adicional para los Administradores de Sistemas .....	5
<i>Pregunta de Investigación .....</i>	<i>5</i>
<i>Objetivos .....</i>	<i>5</i>
Objetivo General.....	6
Objetivos Específicos .....	6
<i>Justificación .....</i>	<i>7</i>

Presupuesto e Infraestructura .....	7
Documentación Técnica y Conocimiento Especializado .....	7
<i>Antecedentes</i> .....	8
<i>Proyecciones</i> .....	11
Alcances .....	12
Limitaciones .....	13
<b>Capítulo II: Marco Teórico.....</b>	<b>14</b>
<i>Marco Referencia de la Institución</i> .....	15
Misión .....	16
Visión .....	16
<i>Sistemas Informáticos</i> .....	18
<i>Almacenamiento de Datos</i> .....	19
<i>Tipos de Almacenamiento de Datos</i> .....	19
Almacenamiento de Conexión Directa (DAS) .....	20
Red de Área de Almacenamiento (SAN) .....	21
Almacenamiento Conectado a la Red (NAS).....	23
Almacenamiento en Base de Datos SQL (Structured Query Language).....	26
<i>Dell EMC Isilon</i> .....	29
Clustrer .....	31
Multiprotocolo.....	31
OneFS.....	32
<i>S3 (Amazon Simple Storage Service)</i> .....	33
Buckets.....	33
API de REST de Amazon S3 .....	34
SDK de AWS .....	35
Puntos de Acceso de S3 .....	36

Listas de Control de Acceso (ACL).....	36
<i>Ingeniería de Software</i> .....	37
<i>Ciclo de Vida del Software</i> .....	38
Planificación.....	40
Análisis .....	40
Diseño .....	41
Desarrollo.....	43
Pruebas .....	44
<i>Paradigmas de Programación</i> .....	47
Paradigma Declarativo .....	47
Paradigma Imperativo .....	48
Programación Orientada a Objetos .....	49
Programación Lógica.....	52
<i>Arquitectura de Software</i> .....	53
Microservicios.....	56
Interfaz de Programación de Aplicaciones (API).....	59
<i>Análisis y Diseño del Sistema</i> .....	60
Requerimientos .....	60
Funcionales .....	61
No Funcionales .....	62
Casos de Uso.....	63
Diagramas de Casos de Uso .....	65
Diagramas de Secuencia .....	67
<i>Metodologías Ágiles de Programación</i> .....	69
Scrum.....	69
Programación Extrema (XP).....	71
Kanban .....	72

<i>Metodologías de Desarrollo Tradicionales</i> .....	74
En Cascada .....	74
Incremental.....	75
Evolutivo .....	76
Prototipo.....	77
Espiral .....	79
<b>Capítulo III: Marco Metodológico.....</b>	<b>80</b>
<i>Investigación</i> .....	81
Tipo de Investigación .....	81
<i>Alcance Investigativo</i> .....	82
<i>Enfoque de la Investigación</i> .....	82
<i>Fuentes de Información</i> .....	84
Fuentes de Información Primaria .....	84
Fuentes de Información Secundaria .....	85
<i>Observación</i> .....	86
<i>Entrevistas</i> .....	87
<b>Capítulo IV: Análisis De Resultados .....</b>	<b>92</b>
<i>Análisis de Factibilidad</i> .....	93
<i>Factibilidad Económica</i> .....	94
<i>Diagnostico Administrativo y Operativo</i> .....	94
<i>Diagnóstico Técnico</i> .....	95
Lenguaje de Programación .....	96
Servidor de Aplicaciones.....	96
Herramientas Aplicadas sobre el Recurso Humano.....	97
<i>Plan de riesgos</i> .....	98

<b>Capítulo V: Conclusiones y Recomendaciones .....</b>	<b>106</b>
<i>Conclusiones .....</i>	<i>107</i>
<i>Recomendaciones .....</i>	<i>108</i>
<b>Capítulo VI. Propuesta de desarrollo .....</b>	<b>111</b>
<i>Modelado de la Solución.....</i>	<i>112</i>
Mejora Propuesta en el Almacenamiento y Consulta de Audiencias .....	112
Exportación de Audiencias de SIGAO.....	116
Consulta de Audiencias desde Iphone GL.....	118
<i>Requerimientos.....</i>	<i>119</i>
<i>Reglas .....</i>	<i>120</i>
<i>Casos de Uso.....</i>	<i>120</i>
<i>Diagramas Casos de Uso .....</i>	<i>130</i>
<i>Diagramas de Secuencia.....</i>	<i>132</i>
<i>Base de Datos .....</i>	<i>135</i>
<i>Desarrollo y Pruebas de los Microservicios con Sistemas .....</i>	<i>136</i>
<b>Referencias .....</b>	<b>149</b>
<b>APÉNDICES.....</b>	<b>¡Error! Marcador no definido.</b>

**Tablas**

<i>Tabla 1 Diferencias entre Arquitectura de Software y Diseño de Software .....</i>	<i>54</i>
<i>Tabla 2 Arquitecturas Monolíticas vs. Microservicios.....</i>	<i>58</i>
<i>Tabla 3 Variables .....</i>	<i>90</i>
<i>Tabla 4 Determinación de Brechas .....</i>	<i>97</i>
<i>Tabla 5 Plan de Gestión de Riesgos para el Proyecto .....</i>	<i>99</i>
<i>Tabla 6 Matrix de Riesgos de Probabilidad e Impacto .....</i>	<i>103</i>
<i>Tabla 7 Análisis de Riesgos por Probabilidad, Impacto y Nivel de Riego.....</i>	<i>104</i>
<i>Tabla 8 Identificación de Requerimientos.....</i>	<i>119</i>
<i>Tabla 9 Caso de Uso Conversión de Audiencias de SIGAO .....</i>	<i>120</i>
<i>Tabla 10 Caso de Uso Consulta de Audiencias.....</i>	<i>123</i>
<i>Tabla 11 Caso de Uso Consulta de audiencias desde Iphone GL.....</i>	<i>124</i>
<i>Tabla 12 Caso de Uso Agregar Nuevas Audiencias.....</i>	<i>126</i>
<i>Tabla 13 Caso de Uso Inventariado de Audiencias del ISILON .....</i>	<i>128</i>
<i>Tabla 14 Caso de Uso Exportación de Audiencias en SIGAO.....</i>	<i>129</i>
<i>Tabla 15 Diccionario de Datos .....</i>	<i>135</i>

## Figuras

<i>Figura 1 Valores Institucionales Poder Judicial</i> .....	16
<i>Figura 2 Organigrama del poder Judicial</i> .....	17
<i>Figura 3 Diagramas de Tipos de Almacenamiento de Datos</i> .....	19
<i>Figura 4 Arquitectura NAS (Almacenamiento de Conexión Directa)</i> .....	22
<i>Figura 5 Arquitectura NAS (Almacenamiento de Conexión Directa)</i> .....	24
<i>Figura 6 Structured Query Language (SQL)</i> .....	26
<i>Figura 7 Ejemplo Diagrama de Base de Datos SQL</i> .....	27
<i>Figura 8 Arquitectura NAS de Expansión Horizontal de Isilon</i> .....	30
<i>Figura 9 Ejemplo de Protocolos</i> .....	31
<i>Figura 10 Integración de Aplicación con Isilon API REST y Protocolo S3</i> .....	35
<i>Figura 11 Historia de la ingeniería de Software</i> .....	38
<i>Figura 12 Ciclo de Vida del Software</i> .....	39
<i>Figura 13 Componentes Claves del Análisis</i> .....	41
<i>Figura 14 Elementos del Diseño del Software</i> .....	42
<i>Figura 15 Ejemplos de Buenas Prácticas de la Fase de Desarrollo</i> .....	44
<i>Figura 16 Clasificación de las pruebas de Software</i> .....	45
<i>Figura 17 Las Tres Estructuras Básicas de la Programación Estructurada</i> .....	49
<i>Figura 18 Programación Orientada a Objetos</i> .....	50
<i>Figura 19 Ejemplo de Arquitectura de software</i> .....	53
<i>Figura 20 Tipos de Arquitectura de Software</i> .....	55
<i>Figura 21 Arquitectura de Microservicios</i> .....	57
<i>Figura 22 Beneficios de los Microservicios</i> .....	58

<i>Figura 23 Concepción de un Requerimiento .....</i>	<i>60</i>
<i>Figura 24 Ciclo de Vida del Producto .....</i>	<i>61</i>
<i>Figura 25 Sistema de Casos de Uso.....</i>	<i>64</i>
<i>Figura 26 Mapa Conceptual de Casos de Uso.....</i>	<i>65</i>
<i>Figura 27 Diagrama de Caso de Uso .....</i>	<i>66</i>
<i>Figura 28 Componentes y sus Símbolos de Diagrama de Casos de Uso.....</i>	<i>67</i>
<i>Figura 29 Diagrama de Secuencia .....</i>	<i>68</i>
<i>Figura 30 Proceso Scrum.....</i>	<i>70</i>
<i>Figura 31 Proceso de Programación Extrema .....</i>	<i>72</i>
<i>Figura 32 Tablero kanban .....</i>	<i>74</i>
<i>Figura 33 Modelo Cascada.....</i>	<i>75</i>
<i>Figura 34 Modelo Incremental.....</i>	<i>76</i>
<i>Figura 35 Modelo Evolutivo .....</i>	<i>77</i>
<i>Figura 36 Modelo de Prototipo .....</i>	<i>78</i>
<i>Figura 37 Modelo Espiral .....</i>	<i>79</i>
<i>Figura 38 Proceso Cuantitativo.....</i>	<i>83</i>
<i>Figura 39 Proceso Cualitativo .....</i>	<i>84</i>
<i>Figura 40 Tipologías de Entrevistas .....</i>	<i>87</i>
<i>Figura 41 Estudio o Análisis de Factibilidad .....</i>	<i>93</i>
<i>Figura 42 Plan de Riegos de Software .....</i>	<i>98</i>
<i>Figura 43 Diagrama Propuesta en el Almacenamiento y Consulta de Audiencias .....</i>	<i>112</i>
<i>Figura 44 Diagrama Utilitario para Inventariar las Audiencias .....</i>	<i>114</i>
<i>Figura 45 Exportación de Audiencias de SIGAO.....</i>	<i>116</i>

<i>Figura 46 Consulta de Audiencias desde Iphone GL.....</i>	<i>118</i>
<i>Figura 47 Diagrama Caso Conversión de audiencias de SIGAO .....</i>	<i>130</i>
<i>Figura 48 Diagrama Caso de Consulta de Audiencias.....</i>	<i>131</i>
<i>Figura 49 Diagrama Caso de Exportación de Audiencias de SIGAO.....</i>	<i>131</i>
<i>Figura 50 Diagrama de Secuencia Servicio Conversión de Audiencias.....</i>	<i>132</i>
<i>Figura 51 Diagrama de Secuencia Conversión de Audiencias Microservicio S3.....</i>	<i>133</i>
<i>Figura 52 Diagrama de Secuencia de Consulta de Audiencias.....</i>	<i>134</i>
<i>Figura 53 Diagrama de Base de Datos.....</i>	<i>135</i>
<i>Figura 54 Interfaz Swagger Microservicio Consulta Audiencias .....</i>	<i>136</i>
<i>Figura 55 Interfaz Swagger Microservicio Encargado de Guardar las Audiencias.....</i>	<i>137</i>
<i>Figura 56 Worker Services Inventariado Audiencias.....</i>	<i>138</i>
<i>Figura 57 Prueba Reproducción Audiencia de Audio Gestión en Línea.....</i>	<i>139</i>
<i>Figura 58 Prueba Reproducción Audiencia de Video Gestión en Línea .....</i>	<i>140</i>
<i>Figura 59 Audiencias de Prueba Sistema SIGAO .....</i>	<i>141</i>
<i>Figura 60 Prueba Reproducción Audiencia de Video SIGAO .....</i>	<i>142</i>
<i>Figura 61 Prueba Reproducción Audiencia de Audio SIGAO .....</i>	<i>143</i>
<i>Figura 62 Prueba Exportación de Audiencia Video SIGAO .....</i>	<i>144</i>
<i>Figura 63 Prueba Exportación de Audiencias Audio SIGAO .....</i>	<i>145</i>
<i>Figura 64 Pruebas de Carga Consulta de Audiencias .....</i>	<i>146</i>
<i>Figura 65 Pruebas de Carga Guardar Audiencias .....</i>	<i>146</i>
<i>Figura 66 Pruebas Unitarias Microservicios Encargado de Guardar las Audiencias.....</i>	<i>147</i>
<i>Figura 67 Pruebas Unitarias Microservicios Encargado de Consultar las Audiencias.....</i>	<i>148</i>

### **Dedicatoria y agradecimiento**

Este trabajo se lo dedico primero a Dios, por ser mi guía en cada paso de este camino y a mi familia, que ha sido un pilar fundamental en todo momento. Agradezco su presencia constante y su apoyo incondicional en cada etapa de mi vida.

A lo largo de este recorrido, he contado con el apoyo incondicional de muchas personas que han sido clave en mi desarrollo personal y profesional. Agradezco profundamente a quienes me han acompañado en las distintas etapas, brindándome su confianza, consejo y motivación. Su respaldo ha sido fundamental para superar los desafíos y alcanzar este logro. Gracias a todos por ser parte de este camino.

## Resumen

El presente proyecto se enfoca en proporcionar una solución tecnológica para mejorar la gestión de audiencias judiciales de audio y video en el Poder Judicial de Costa Rica. Las tecnologías que actualmente se utilizan para realizar estos procesos presentan problemas de eficiencia y organización en el servidor *Isilon*, lo que genera tiempos prolongados de respuesta y fallos al consultar archivos, afectando tanto a los administradores del sistema como a los usuarios finales.

El objetivo principal de la investigación consiste en desarrollar e implementar el protocolo S3 de Amazon con el servidor *Isilon* para almacenar y consultar las Audiencias. Esto permite que estas tecnologías optimicen la organización de la información, para la recuperación de los archivos, mejorando tanto la seguridad como la eficiencia, además, permitirá a la institución hacer frente a la problemática que se está presentando actualmente. Con tal propósito, se realizaron pruebas de integración y de carga para asegurar que estas aplicaciones funcionen correctamente en el entorno de desarrollo.

Los resultados observados mostraron mejoras en el tiempo de respuesta al consultar y almacenar las Audiencias, lo que aumentará la satisfacción de las personas usuarias.

También, reducirá el trabajo del personal que tiene que atender los reportes relacionados con la problemática actual.

Esta implementación basada en microservicios y el *protocolo S3* representa una mejora considerable en la gestión de los datos de audiencias judiciales en el Poder Judicial de Costa Rica. La solución se enfoca en los problemas actuales de rendimiento, organización y asegura un buen funcionamiento del sistema con tecnologías de punta, garantizando un servicio más eficiente y seguro para las personas usuarios internas y externas.

## Capítulo 1: Problema

## Planteamiento del Problema

En el entorno actual de rápida evolución tecnológica, la capacidad de almacenar y consultar eficientemente grandes volúmenes de datos de audio y video es crucial para las organizaciones que buscan mantenerse competitivas y cumplir con las expectativas de las personas usuarias. Sin embargo, en el Poder Judicial la tecnología que se está utilizando actualmente para gestionar las audiencias de audio y video, no cumplen con las expectativas esperadas en cuanto a la eficiencia, seguridad y capacidad de crecimiento. Por lo cual esto tiene repercusiones significativas en la organización. Este capítulo se va a enfocar en definir el problema que se está presentando actualmente con el almacenamiento y consulta de audiencias con el objetivo de identificar las causas y los efectos que justifican la necesidad de abordar las deficiencias tecnológicas que se están presentando.

En específico, el problema principal surge al realizar consultas al *servidor Isilon*, donde se experimentan tiempos prolongados para localizar y responder a las peticiones de los diversos sistemas jurisdiccionales. En algunos casos, el servidor no responde debido a problemas operativos significativos.

Este problema se origina en la gestión inadecuada del consolidado que almacena los archivos de audio y video en *streaming* de los diferentes circuitos judiciales desde el año 2019. La falta de organización estructurada de los datos impide una respuesta rápida del servidor. La ausencia de una organización por carpetas u otro método eficiente dificulta la localización eficaz de los archivos de las audiencias, prolongando así los tiempos de búsqueda y afectando la disponibilidad del sistema para atender las solicitudes de manera oportuna.

El problema mencionado anteriormente es relevante porque afecta directamente a las personas administradoras de los sistemas, quienes enfrentan dificultades en el

almacenamiento adecuado de las audiencias de audio y video, especialmente durante los momentos de mayor demanda. Una preocupación adicional es la capacidad limitada para consultar eficientemente las audiencias ya almacenadas en el *servidor Isilon*. Durante períodos de alta actividad, estas consultas experimentan demoras significativas, lo resulta en un mal rendimiento del sistema. Esto genera frustración a las personas usuarias internas como externas de la organización. Abordar estos problemas es crucial para mejorar la velocidad, seguridad y confiabilidad del sistema, aspectos fundamentales para mantener la competitividad y la satisfacción de las personas usuarias.

### **Variables del Problema**

- Falta de un protocolo moderno y eficiente para almacenar datos de audio y video.
- Problemas con el servidor que se encarga de resolver las consultas de audio y video de las audiencias.
- No existe un protocolo adecuado o medidas de seguridad adecuadas para la consulta y almacenamiento de las audiencias.
- Problemas relacionados con la red institución que afecten la transmisión y accesos de los datos. Así con no tener el ancho de banda adecuado en algunas zonas del país para las consultas que se realizan, por lo cual se producen cuellos de botella.

### **Causas del Problema**

#### ***Gestión Inadecuada de Datos***

La falta de una estructura organizativa adecuada en el almacenamiento de archivos de audio y video desde 2019 impide una rápida localización y acceso a las audiencias. La ausencia de una organización por carpetas u otra forma dificulta la gestión y recuperación de los datos cuando se realizan consultas.

### ***Deficiencias en el Servidor Isilon***

Problemas operativos significativos en el *servidor Isilon*, encargado de almacenar y servir los archivos de audio y video, contribuyen a tiempos prolongados de respuesta e incluso a la falta de respuesta en algunos casos.

### ***Protocolos y Medidas de Seguridad Inadecuados***

La falta de un protocolo moderno y eficiente para el almacenamiento consulta y seguridad de los datos de audio y video aumenta el riesgo de problemas operativos y compromisos de seguridad.

### ***Problemas de Infraestructura de Red***

Limitaciones en la red institucional, incluyendo falta de ancho de banda adecuado en algunas zonas del país, provocan cuellos de botella durante las consultas de audiencias. La transmisión lenta o interrumpida de datos afecta negativamente la experiencia del usuario y la disponibilidad del sistema.

## **Efectos del Problema**

### ***Rendimiento Deficiente del Sistema***

Durante los períodos de alta actividad, las consultas a las audiencias almacenadas experimentan demoras significativas o no pueden completarse, lo que impacta negativamente en la productividad y eficiencia de los sistemas.

### ***Insatisfacción de los Usuarios***

Tanto los usuarios internos como externos del Poder Judicial experimentan frustración debido a la lentitud o falta de respuesta del sistema al consultar audiencias. Esta insatisfacción puede deteriorar la percepción y confianza en el sistema.

### ***Riesgos de Seguridad***

La falta de medidas adecuadas de seguridad aumenta el riesgo de accesos no autorizados o pérdida de datos sensibles relacionados con las audiencias judiciales. Esto puede comprometer la integridad y confidencialidad de la información crítica manejada por el Poder Judicial.

### ***Carga Adicional para los Administradores de Sistemas***

Las dificultades en el almacenamiento de las audiencias de audio y video durante los momentos de alta demanda aumentan la carga de trabajo para las personas administradoras del sistema.

El Poder Judicial presenta una problemática significativa en la gestión eficiente de audiencias de audio y video debido a deficiencias tecnológicas y operativas. Los problemas en el *servidor Isilon*, seguridad y confiabilidad del sistema. Estos problemas no solo generan frustración entre los usuarios internos y externos, sino que también comprometen la eficiencia operativa y la integridad de los datos judiciales.

### **Pregunta de Investigación**

¿Cómo se puede realizar un desarrollo de software con el protocolo S3 de una forma eficiente utilizando un servidor Isilon como almacenamiento e implementado el uso de microservicios para el almacenamiento y consulta de audiencias de audio y video para solucionar los problemas de rendimiento y seguridad?

### **Objetivos**

A continuación, se presentan los objetivos de esta investigación, los cuales están diseñados para abordar de manera efectiva el problema que se está presentado con el almacenamiento y consulta de audiencias de audio y video en el contexto del Poder Judicial:

### ***Objetivo General***

Implementar microservicios en el ambiente de desarrollo del Poder Judicial de Costa Rica en el segundo semestre del año 2024, utilizando el protocolo S3 en un servidor Isilon para el almacenamiento y consulta de audiencias de audio y video, mejorando el rendimiento y seguridad.

### ***Objetivos Específicos***

A continuación, se mencionan los objetivos específicos del estudio.

1. Evaluar el estado actual, analizando la forma de almacenamiento y consulta de audiencias de audio y video, para la identificación de deficiencias y áreas de mejora.
2. Diseñar una solución basada en microservicios utilizando el protocolo S3 para las deficiencias identificadas en el sistema actual.
3. Desarrollar microservicios de almacenamiento y consulta usando el protocolo S3 en el *servidor Isilon* que permite a las personas usuarias la reproducción de las audiencias orales.
4. Realizar pruebas de microservicio de almacenamiento y consulta por medio de la revisión de del código, pruebas de integración y la utilización de pruebas unitarias para que la solución dada no presente problemas en el ambiente de producción.

## **Justificación**

La implementación de un sistema basado en microservicios utilizando el protocolo S3 en el *servidor Isilon* del Poder Judicial de Costa Rica se justifica por diversos factores clave que aseguran la viabilidad y el éxito del proyecto. A continuación, se detallan estos aspectos.

### ***Presupuesto e Infraestructura***

La inversión en este proyecto es mínima debido a varios factores. El desarrollo de los microservicios utilizando el protocolo S3, será realizado como parte de un proyecto de graduación, lo cual elimina costos asociados al desarrollo. Además, el protocolo S3 de Amazon es gratuito cuando se utiliza con un *servidor Isilon* localmente, sin costos de licenciamiento ni tarifas asociadas. La infraestructura existente incluye el *servidor Isilon*, ya instalado y en uso, eliminando la necesidad de inversión adicional en *hardware* o infraestructura de almacenamiento. Los microservicios no tienen un costo adicional, ya que se desarrollarán utilizando tecnologías y herramientas disponibles.

La única inversión significativa se relaciona con el personal necesario para realizar las pruebas y la implementación en producción, quienes dedicarán tiempo a estas actividades como parte de sus responsabilidades habituales, sin gastos adicionales fuera del presupuesto operativo actual del departamento de TI. Aunque el desarrollo no tiene costo, se debe considerar el tiempo del personal de TI para pruebas exhaustivas y la implementación en el entorno de producción. Por lo cual en temas de presupuesto este proyecto es factible, ya que no conlleva una gran inversión económica para la institución.

### ***Documentación Técnica y Conocimiento Especializado***

En cuanto a la documentación técnica relacionada con el protocolo S3 de Amazon AWS, se dispone de una gran variedad de recursos detallados y actualizados, que incluyen guías de implementación, mejores prácticas y ejemplos. Adicionalmente, el Poder Judicial

ofrece *wikis internas* con información relacionada con el desarrollo de *software*. Esta documentación facilita la implementación del protocolo S3 en el Poder Judicial.

El departamento de sistemas del Poder Judicial cuenta con personal calificado que tiene varios años de experiencia en desarrollo dentro de la institución. Este personal proporciona un apoyo técnico abundante tanto en temas relacionados con el desarrollo como en asuntos de negocio. Además, están familiarizados con nuevas tecnologías, como los microservicios.

Existen también departamentos especializados, como QA, infraestructura, bases de datos y telemática, cuyos conocimientos especializados en sus respectivas áreas pueden ser de gran ayuda en caso de que surjan problemas. Esta estructura organizacional garantiza que el proyecto de ser necesario cuente con el soporte técnico necesario para su éxito.

### **Antecedentes**

Uno de los principales antecedentes es de Amazon (2024) define el protocolo S3 como:

Amazon Simple Storage Service (Amazon S3) es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes en el sector. Puede usar Amazon S3 para almacenar y recuperar cualquier cantidad de datos en cualquier momento, desde cualquier parte.

Amazon AWS es crucial para esta investigación porque ofrece una estructura detallada sobre cómo implementar el protocolo S3 para el almacenamiento de objetos, lo cual mejorará los sistemas judiciales. Además, proporciona una guía exhaustiva que facilita la implementación del protocolo S3 y amplía el conocimiento sobre conceptos esenciales para integrar este protocolo en microservicios. Esto es fundamental para asegurar la eficiencia y seguridad en la gestión de la información de las audiencias de audio y video del Poder

Judicial. Lo más importante es que estos llevan años en el mercado ofreciendo el servicio de almacenamiento de archivos usando el protocolo S3.

Otro antecedente importante es (Dell Technologies, 2023) menciona “La plataforma de almacenamiento NAS de escalamiento horizontal combina hardware modular con software unificado para aprovechar datos no estructurados. El sistema operativo OneFS impulsa la plataforma para ofrecer un pool escalable de almacenamiento con un espacio de nombres global” (p.04), Dell Technologies proporciona una extensa documentación que aborda tanto las capacidades como las limitaciones de los servidores Isilon, información crucial para la investigación sobre la implementación de microservicios utilizando el protocolo S3.

Este documento presenta directrices y recomendaciones para configurar *OneFS*. Se proporcionan directrices de configuración para protocolos, características del sistema de archivos, componentes de *software* y *hardware*, y configuraciones de red.

Un primer trabajo de investigación fue el de López (2017) que comento lo siguiente:

El objetivo del presente estudio fue identificar las tecnologías, metodología y arquitectura que utiliza la CGTIC para el desarrollo de aplicaciones web y la correspondiente identificación de las tecnologías existentes para el desarrollo e implementación de microservicios, utilizando como base de la investigación un enfoque cualitativo, con un tipo de investigación descriptiva y diseño documental.  
(p.10)

El estudio menciona e investiga la implementación de microservicios, lo cual es una tendencia significativa en el diseño de arquitecturas de *software* moderna, lo cual va a ayudar al desarrollo del proyecto que se está investigando y se va a implementar, ya que la información proporcionada por Ibarra en su investigación ayudara a comprender mejor el tema de los microservicios.

En un segundo trabajo (Chicaiza, 2020) menciona:

Desde el principio, la ingeniería de software buscó desarrollar, crear y mantener aplicaciones informáticas que satisfagan las necesidades de las organizaciones y los requisitos técnicos de los consumidores. Por lo tanto, la arquitectura de microservicios nació como un enfoque para mejorar las arquitecturas tradicionales (monolíticas), enfocadas en la estabilidad, la confiabilidad, un alto grado de tolerancia a fallas y una escala simple. (p.12)

Al comparar este enfoque con estudios e investigaciones similares previas como la anterior, se puede explorar cómo diferentes criterios abordan la implementación de microservicios y sus beneficios en contraste con las arquitecturas tradicionales. Además, se puede analizar cómo esta investigación contribuye al entendimiento y la adopción de mejores prácticas en el desarrollo de *software moderno*, destacando su impacto potencial en la eficiencia, la agilidad y la capacidad de respuesta de los sistemas informáticos en entornos organizacionales.

En un tercer trabajo de investigación final de la Universidad de Costa Rica (González, 2023) menciona lo siguiente:

La facilidad de implementación de aplicaciones web contenerizadas y la facilidad de administración que ofrecen las plataformas orquestadoras como Kubernetes ha impulsado el diseño de sistemas por microservicios. Los contenedores que implementan microservicios pueden ser objetivos de ataques a nivel de seguridad y debido a la homogeneización del lenguaje de programación dichos ataques son más propensos al éxito. (p.07)

Esta investigación realizada abordar desafíos de seguridad que pueden afectar los microservicios en relación con la homogeneización del lenguaje de programación, lo cual

puede facilitar los ataques. Por lo cual este estudio es importante para comprender temas de seguridad y los riesgos que se pueden presentar con la implementación de un microservicio como es el caso de la implementación que se está investigando en estos documentos sobre los microservicios que se van a implementar.

En otro trabajo de investigación final de la Universidad de Costa Rica (González, 2023) menciona lo siguiente:

Los microservicios han surgido como un estilo arquitectónico que ofrece muchas ventajas, pero también plantea desafíos. Uno de estos desafíos gira alrededor de las pruebas, puesto que una aplicación puede tener cientos o miles de servicios que funcionan juntos, y cada uno de ellos requiere ser probado a medida que evolucionan. Para superar este desafío, la automatización adquiere un papel clave, y junto con ella, el uso de herramientas de pruebas eficientes y eficaces. (p.07)

El estudio proporciona una visión clara sobre los desafíos que enfrentan los sistemas basados en microservicios, especialmente en términos de pruebas. Además, los resultados y conclusiones del proyecto de investigación ofrecen recomendaciones de herramientas que podrían ser utilizadas para entornos en específico y también la información relacionada con errores comunes y la optimización de soluciones tecnológicas. Esto permitirá contextualizar mejor los problemas y desafíos de la investigación sobre el almacenamiento y consulta de audiencias de audio y video en el Poder Judicial.

### **Proyecciones**

A continuación, se destallarán las proyecciones, este apartado está dividido en alcances y limitaciones de proyecto de investigación.

### *Alcances*

- Asegurar un entendimiento del protocolo S3, incluyendo sus características, beneficios, limitaciones, recopilar y analizar experiencias de implementación en entornos similares para adoptar las mejores prácticas, evaluando su compatibilidad y limitaciones de su implementación en el *servidor Isilon* mucha de esta información será impórtate para agregarla en el marco teórico del proyecto de investigación.
- Evaluar el estado actual de la forma en la que se guardan y consultan las audiencias para identificar deficiencias y áreas de mejora, lo que proporcionará una base clara para diseñar soluciones más eficientes. Esto se plasmará en un diagrama de situación actual del sistema.
- Diseñar una solución basada en microservicios utilizando el protocolo S3, enfocándose en resolver las deficiencias identificadas en el sistema actual, asegurando una arquitectura escalable y eficiente para el almacenamiento y consulta de audiencias. En este caso se realizará un diagrama con la solución por desarrollar y permitirá un mejor entendimiento de la solución propuesta.
- Desarrollar microservicios de almacenamiento y consulta usando el protocolo S3 en el *servidor Isilon* permitiendo a las personas usuarias la reproducción de las audiencias orales, para lo cual se tiene que utilizar el ambiente de desarrollo del Poder Judicial donde se tiene que crear los microservicios en *net core*.
- Realizar pruebas de los microservicios de almacenamiento y consulta, incluyendo revisión de código, pruebas de integración y unitarias, para asegurar que la solución funcione correctamente en el ambiente de producción sin problemas. En este caso se crea el proyecto de prueba en el microservicio, además se crean evidencias como capturas de pantallas o videos.

- Documentar detalladamente los resultados, recomendaciones y conclusiones obtenidas durante el desarrollo e implementación del microservicio de almacenamiento y consulta, para que las personas beneficiadas de la tesis comprendan y utilicen correctamente los microservicios desarrollados. Este entregable se va a ver plasmado en la documentación de la tesis.

### ***Limitaciones***

A continuación, se mencionan algunas limitaciones que pueden afectar a la investigación o implementación del desarrollo de este trabajo de investigación.

**Complejidad Tecnológica.** La implementación de nuevas tecnologías como el protocolo S3 y la adopción de un enfoque basado en microservicios pueden requerir una curva de aprendizaje, lo que podría retrasar la implementación o aumentar el riesgo de errores durante el proceso de desarrollo.

**Infraestructura Existente.** Dependencia de la infraestructura actual del Poder Judicial de Costa Rica, lo que puede limitar la implementación de algunas soluciones avanzadas.

**Desafíos por Restricciones del S3 e Isilon.** La implementación del protocolo S3 o el *servidor Isilon* puede enfrentar obstáculos si ciertos puertos no están permitidos o si hay bloqueos de consultas por parte del departamento de telemática. Esto requeriría solicitar aprobaciones y generar reportes para obtener los permisos necesarios, lo que podría causar retrasos en los tiempos de desarrollo.

## **Capítulo II: Marco Teórico**

## **Marco Referencia de la Institución**

Con la Independencia de Costa Rica, el 15 de setiembre de 1821, los costarricenses se organizaron políticamente y constituyeron un gobierno propio. Para el 1° de diciembre de 1821, los representantes de diferentes ciudades y pueblos de aquel entonces formularon el Pacto Social Fundamental Interino de Costa Rica, conocido como el Pacto de Concordia, considerado como el primer documento constitucional de Costa Rica. En este, se estableció la Junta Suprema Gubernamental para ejercer las funciones de gobierno y se creó también un Tribunal para administrar la justicia pronta y rectamente, conforme a las leyes para los pueblos indo-españoles, conocidas como Leyes de Indias promulgadas por España para regir las posesiones españolas en América, el cual se convirtió en el primer cimiento de la Corte Suprema. (Poder Judicial, 2024)

Como se puede ver, esta institución ya tiene muchos años desde su nacimiento y destaca históricamente por su Pacto de Concordia que fue fundamental para formar el sistema de gobierno y Justicia de nuestro país. Desde ese momento se sentaron los bases para que el Poder Judicial se encargara de administrar una justicia pronta y rectamente en relación con las leyes establecidas.

Considerando el rol tan importante que tiene el Poder Judicial en cuanto a los temas de justicia en nuestro país, la Dirección de Tecnología tiene un papel fundamental en apoyar los objetivos estratégicos de la institución, hoy en día las Tecnologías de la Información se han convertido en un elemento indispensable para el cumplimiento de nuestras leyes, facilitando la resolución de casos y manteniéndose a la vanguardia para tener la información sensible de los costarricenses de manera segura. Este acompañamiento por parte del departamento de TI permite al Poder Judicial responder a las demandas de una sociedad cada

vez más digitalizada, garantizando que la justicia sea más pronta, accesible, equitativa, segura y eficiente.

### ***Misión***

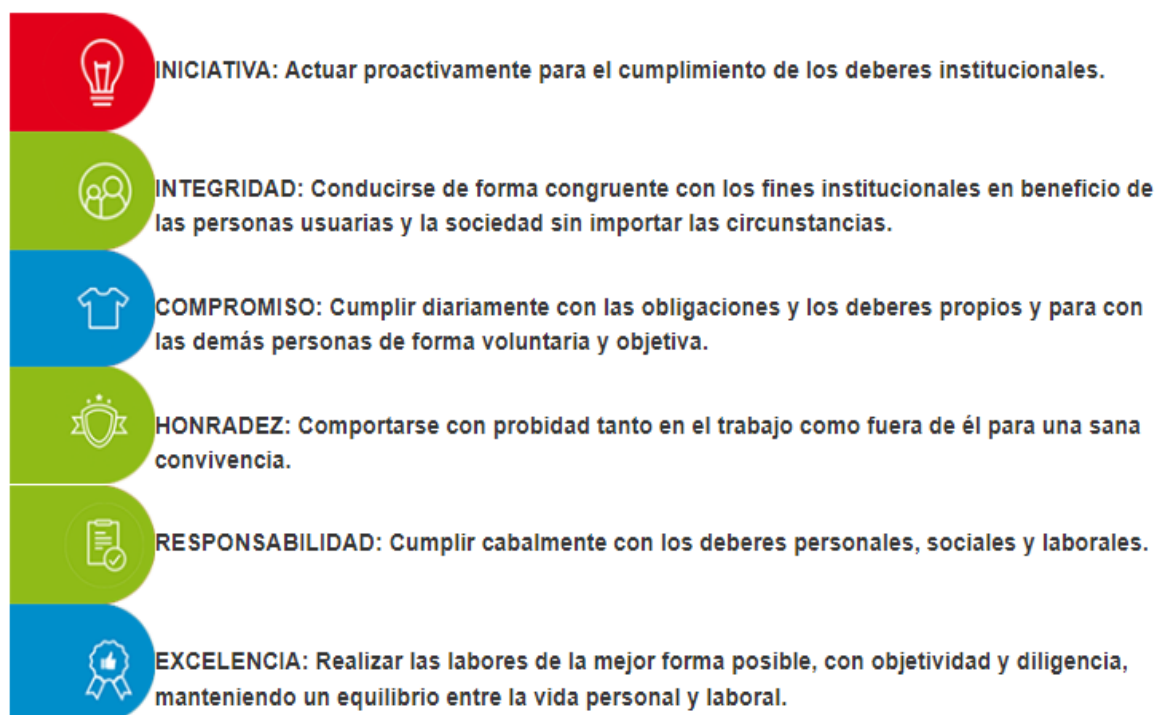
“Administrar justicia pronta, cumplida y accesible, de conformidad con el ordenamiento jurídico para contribuir con la democracia, la paz social y el desarrollo sostenible del país.” (Poder Judicial, 2024)

### ***Visión***

“Ser un Poder Judicial que garantice al país, pleno acceso a la justicia, que resuelva los conflictos de manera pacífica, eficaz, eficiente, transparente y en apego a la ley, con personas servidoras comprometidas con su misión y valores, consciente de su papel en el desarrollo de la sociedad.” (Poder Judicial, 2024)

## **Figura 1**

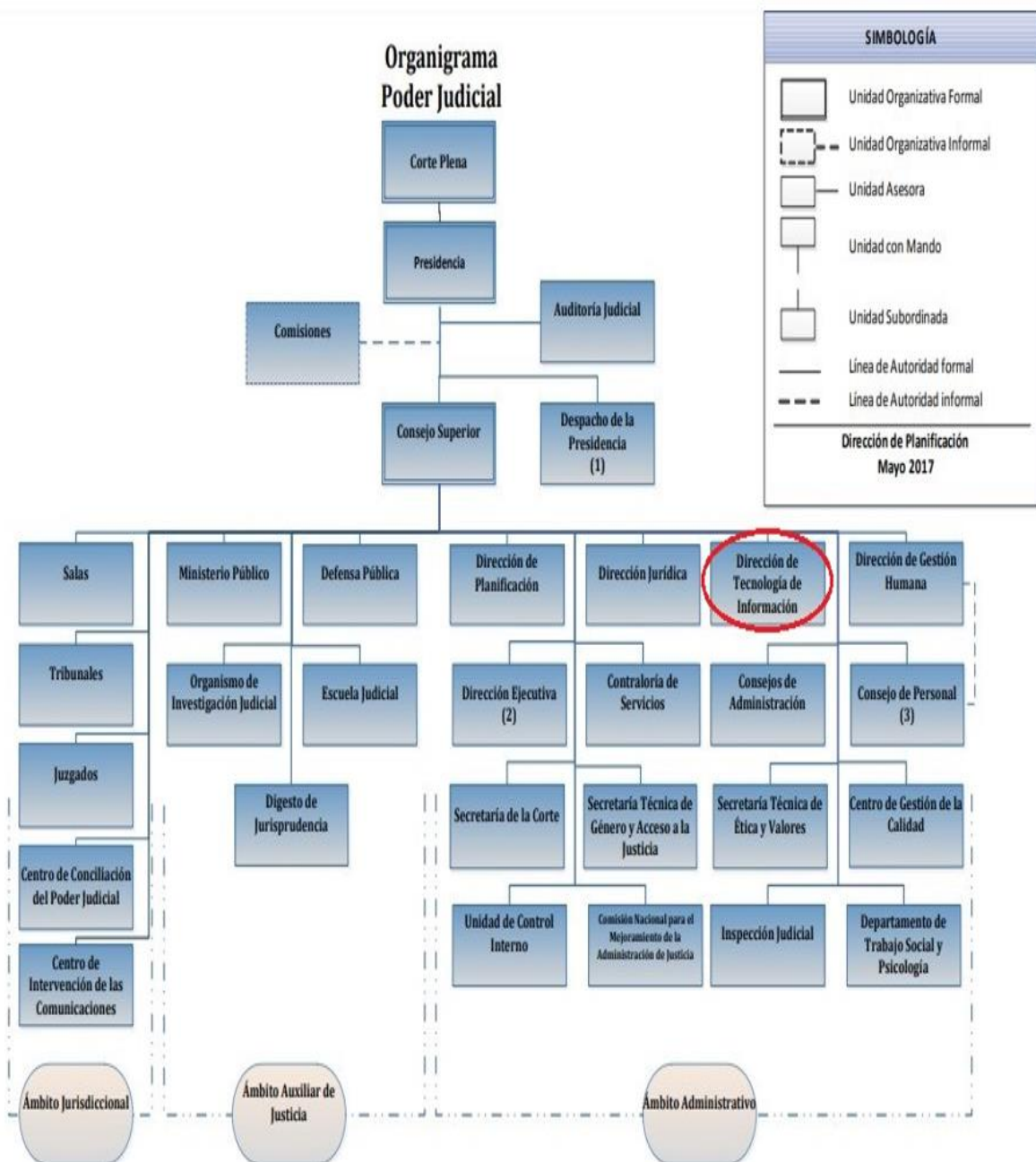
### *Valores Institucionales Poder Judicial*



Fuente: (Poder Judicial, 2024)

A continuación, se observa un organigrama del Poder Judicial, el cual representa la estructura jerárquica de la institución, destacando las diferentes áreas y departamentos que la conforman.

**Figura 2**  
*Organigrama del poder Judicial*



Fuente: (Poder Judicial, 2024)

Como se observa en la figura anterior, la Dirección de Tecnología de información DTI se encuentra dentro de ámbito administrativo y es la encargada de gestionar las iniciativas tecnológicas dentro del Poder Judicial. Además, DTIC tiene a su cargo varios departamentos como es el caso de Telemática, Sistemas Jurisdiccionales, Sistemas Administrativos, Soporte Técnico, Apoyo a la Gestión e Informática Gestión, los cuales se tienen que alinear con los planes estratégicos de la Dirección de TI.

### **Sistemas Informáticos**

Antes de abordar otros conceptos relacionados con el desarrollo del presente capítulo se tiene que entender qué es un sistema informático. El siguiente autor nos ofrece esta definición:

"Un sistema informático es aquel sistema que aúna por un lado la parte física de la informática y por otra, la parte digital o no tangible de la informática." (Llamas, 2020). Esto implica que un sistema informático está estrechamente relacionado con la parte física, que incluye dispositivos esenciales como discos duros, memorias, procesadores y redes cableadas, entre otros. Estos componentes físicos proporcionan la infraestructura básica sobre la cual operan los sistemas informáticos.

También es indispensable la parte lógica o digital, para que el sistema informático funcione de forma correcta. La parte lógica o no tangible es aquella en la que no podemos tocar los sistemas operativos, aplicaciones y programas.

"La función principal de un sistema informático es el procesamiento de datos almacenados. Gracias a que la utilidad de los sistemas informáticos, son adaptables a casi cualquier sector o actividad económica, pueden ser utilizados casi sin restricciones." (Llamas, 2020), entonces se entiende que el objetivo principal de los sistemas de información es la

manipulación de la información de la forma correcta, por lo cual realizar procesos como almacenar, procesar, modificar o eliminar los datos de una forma eficiente y segura.

### Almacenamiento de Datos

Teniendo clara la definición de sistemas informáticos hay que entender el almacenamiento de datos, ya que es una de las funciones más importantes de estos sistemas. “El almacenamiento de datos son medios magnéticos, ópticos o mecánicos que registran y preservan información digital para operaciones en curso o futuras”. (IBM, 2024), se puede entender que la información necesita de elementos físicos de almacenamiento donde se guardan los datos y se pueda mantener la información segura para ser accesible en el momento en que los sistemas informáticos la necesiten.

### Tipos de Almacenamiento de Datos

#### Figura 3

*Diagramas de Tipos de Almacenamiento de Datos*



En la actualidad, con el crecimiento de la información se ha ido creando diferentes tipos de almacenamiento de datos, por eso la correcta elección de dónde o cómo guardar la información es crucial para el rendimiento y la eficiencia de las aplicaciones. Los datos se pueden almacenar de diferentes formas por los cual las empresas o las personas encargadas de gestionar los sistemas tiene que analizar cuáles se adecuan más a sus necesidades.

Los tipos de almacenamiento más conocidos en el mundo de la informática *Direct Attached Storage (DAS)*, *Network Attached Storage (NAS)* y *Storage Area Network (SAN)* en español, respectivamente, sería almacenamiento de conexión directa, almacenamiento conectado a la red y red de área de almacenamiento. Estas tecnologías ofrecen características únicas en cuanto la gestión de datos, la accesibilidad y la escalabilidad. A continuación, se detallarán cada uno de estos tipos de almacenamiento para comprender sus diferencias, y cómo puede impactar en la infraestructura de TI en una organización.

### ***Almacenamiento de Conexión Directa (DAS)***

“Es una tecnología de almacenamiento de datos que conecta un dispositivo de almacenamiento a un equipo a través de una interfaz SCSI, Fibre Channel o SAS. DAS se puede utilizar para acceder y administrar datos y archivos.” (HUAWEI, 2024). En otras palabras, este tipo de almacenamiento es directo a un dispositivo en específico, en este caso se puede decir que un ejemplo es un disco duro de una computadora o una unidad de estado sólido externo conectado a una computadora por lo cual este disco solo puede ser accedido por esa computadora y además este tipo de almacenamiento no tiene un sistema operativo.

### **Ventajas**

- Puedes conectarlo rápidamente a tu computadora no necesita de configuración ni conocimientos avanzados.
- Los datos se acceden rápidamente porque están directamente conectados.

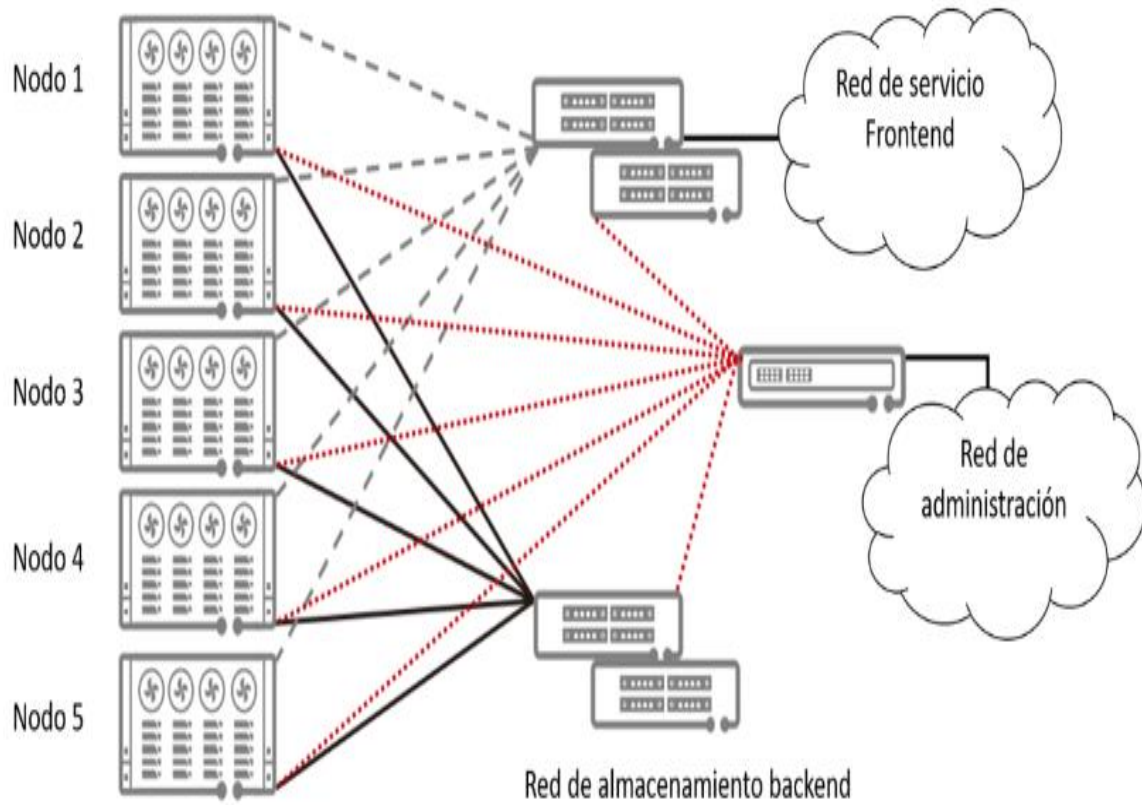
## **Desventajas**

- No puedes compartir fácilmente los datos con otras computadoras en una red.
- Si necesitas más almacenamiento, tendrás que añadir más dispositivos, lo que puede ser desordenado.

El almacenamiento DAS es una opción simple y rápida, ya que conecta directamente dispositivos como discos duros o unidades de estado sólido a una computadora sin necesidad de configuración avanzada. Sin embargo, su limitación para compartir datos en red y la necesidad de agregar más dispositivos para ampliar la capacidad pueden hacer que su uso sea menos práctico en entornos donde se requiere mayor flexibilidad y escalabilidad.

## ***Red de Área de Almacenamiento (SAN)***

“El almacenamiento SAN puede ser una red de varios dispositivos de diferentes tipos, incluido el almacenamiento SSD y flash, el almacenamiento híbrido, el almacenamiento en el cloud híbrido, el software y los dispositivos de copia de seguridad y el almacenamiento en el cloud.” (IBM, 2024) Se puede entender que este tipo de almacenamiento SAN es una forma avanzada de guardar los datos en una red dedicada. Entonces es totalmente diferente al almacenamiento DAS, ya que es un sistema especialidad en una red separada que conecta varios dispositivos de almacenamiento. Además, este tipo de almacenamiento permite la conexión de varios dispositivos como computadoras o celulares al mismo tiempo para acceder a la información de una forma rápida.

**Figura 4***Arquitectura NAS (Almacenamiento de Conexión Directa)*

Fuente: (HUAWEI, 2024)

**Ventajas**

- El acceso a los datos es más rápido, por lo cual es una buena opción para aplicaciones que requieren mucha velocidad de respuesta a la hora de gestionar los datos.
- Es flexible y escalable esto porque se puede ampliar la cantidad de datos que se pueden almacenar según las necesidades de la organización.
- Ayuda a las empresas a centralizar el almacenamiento de la información en un solo lugar, lo que permite que varios usuarios o servidores puedan acceder a la información.

- En comparación con DAS, SAN es más seguro, esto porque tiene tipos de seguridades avanzadas, lo cual permite que los datos estén protegidos, esto porque son redes exclusivas para el almacenamiento, lo que reduce considerablemente los daños que pretenden las personas mal intencionadas.

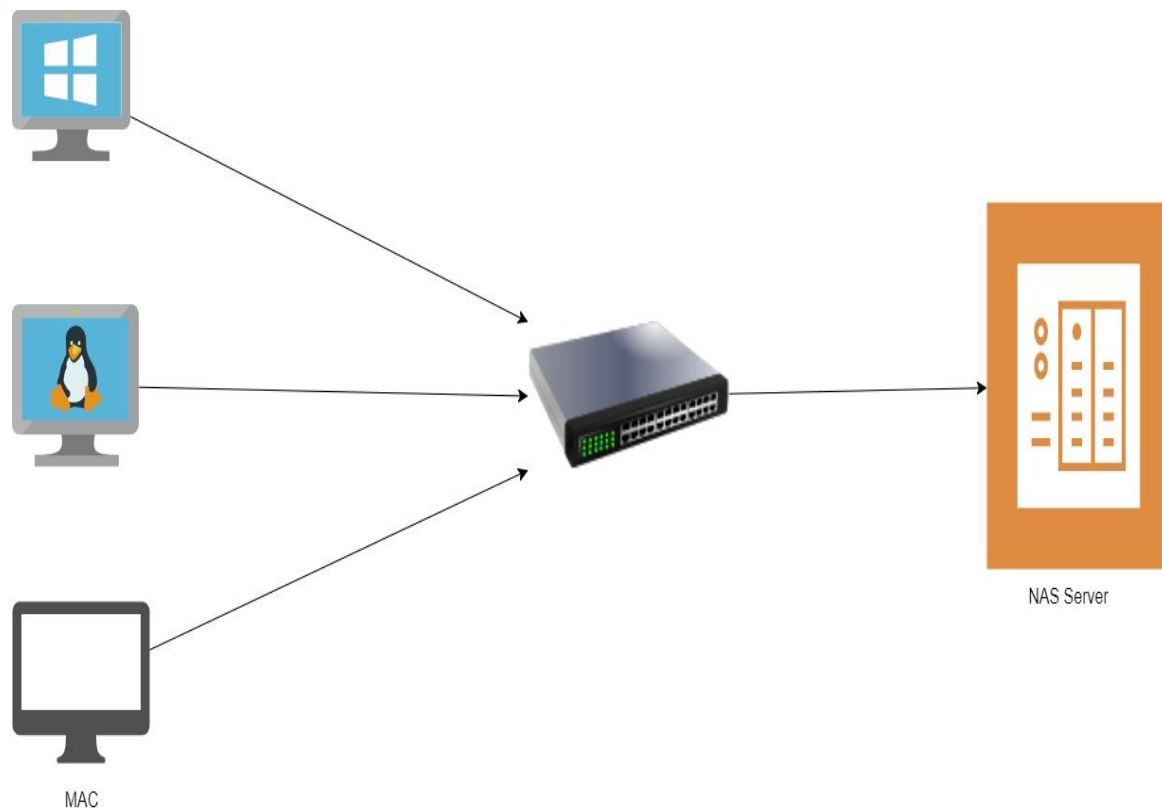
### **Desventajas**

- Puede ser más caro de implementar que otros tipos de almacenamiento debido a la infraestructura, la seguridad y los dispositivos especializados que se necesita para su implementación.
- Requiere una configuración y gestión más compleja comparadas con otras soluciones de almacenamiento.

El almacenamiento SAN es una solución avanzada que ofrece acceso rápido y centralizado a los datos mediante una red dedicada que conecta múltiples dispositivos de almacenamiento. Su flexibilidad, escalabilidad y seguridad lo hacen ideal para aplicaciones que demandan alta velocidad. No obstante, su implementación requiere una inversión considerable y una gestión compleja debido a la infraestructura especializada necesaria.

### ***Almacenamiento Conectado a la Red (NAS)***

“El almacenamiento conectado a red (NAS) es un almacenamiento de archivos dedicado que permite que varios usuarios y dispositivos de los clientes heterogéneos recuperen datos de una capacidad de disco centralizada. Los usuarios de una red de área local (LAN) acceden al almacenamiento compartido a través de una conexión Ethernet estándar” (J. Bigelow y otros, 2024). En este se puede observar que el tipo de almacenamiento NAS es similar a SAN, esto porque proporcionar un tipo de almacenamiento que está centralizado, es más seguro y permite la conexión de varios usuarios, así como dispositivos al mismo tiempo. Pero NAS se enfoca en dar acceso a nivel de archivos a través de una red local.

**Figura 5***Arquitectura NAS (Almacenamiento de Conexión Directa)****Ventajas***

- El almacenamiento NAS puede ser flexible y escalable esto porque se puede agregar más discos duros para aumentar el almacenamiento según las necesidades de la organización.
- Por lo general la administración y configuración de este tipo de almacenamiento ofrece interfaces de usuario amigables, por lo cual las personas encargadas lo pueden gestionar de una forma sencilla.
- La seguridad que ofrece este tipo de almacenamiento es buena, además tiene la posibilidad de realizar respaldos para recuperar la información.

- Al ser un almacenamiento centralizado todas las personas que tengan accesos a la red y tengan permisos para acceder a los datos lo podrán hacer de forma remota.

### **Desventajas**

- El almacenamiento de tipo NAS tiene una dependencia fuerte con la estabilidad y el rendimiento de la red, si la red está lenta, sobrecargada, no tiene el ancho de banda recomendado u otros problemas, este tipo de almacenamiento se verá afectado a la hora de guardar, modificar o acceder a los datos.
- Este tipo de almacenamiento puede ser caro de implementar, ya que se necesita una inversión inicial para la compra del equipo, configuración y la capacitación del departamento de Tecnología informática.
- NAS es un almacenamiento que tiene que administrar la empresa que lo adquiere, por lo cual, el personal de TI tiene que estar capacitado en temas de seguridad, además de que la infraestructura de la organización debe ser segura para que se pueda implementar este tipo de almacenamiento, más si los datos son confidenciales. A pesar de que NAS no es tan complicado de implementar, requiere de mantenimiento, actualizaciones y estar monitoreando su correcto funcionamiento.

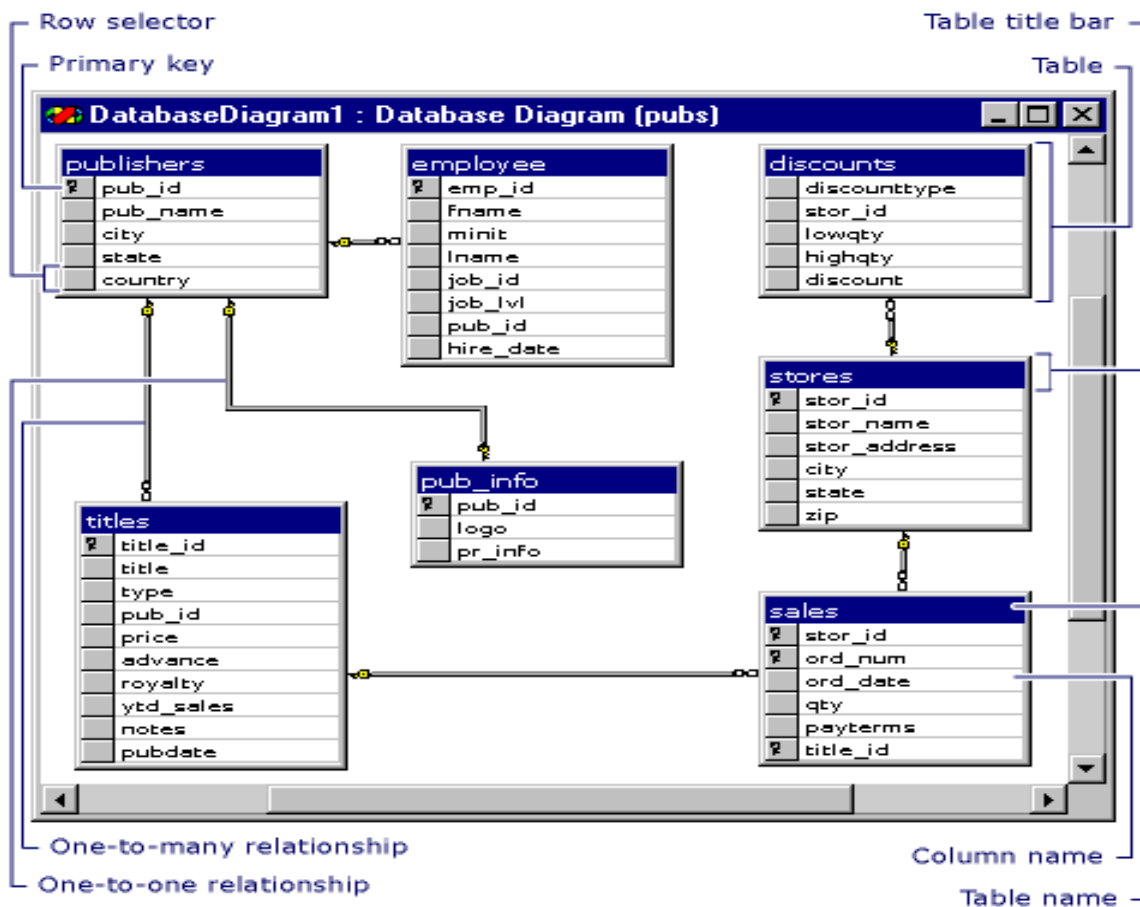
El uso de NAS es muy útil cuando se necesita un acceso centralizado y compartido a los datos, facilitando la gestión de información en una empresa. Es ideal para organizaciones donde varios usuarios deben acceder a archivos desde distintas ubicaciones, ya que permite escalar el almacenamiento según sea necesario. Sin embargo, es importante considerar las limitaciones de la red, porque NAS depende de su estabilidad y rendimiento. Además, hay



tipo de bases de datos relacionales, Google las define como; “una colección de información que organiza datos en relaciones predefinidas, en la que los datos se almacenan en una o más tablas (o "relaciones") de columnas y filas, lo que facilita la visualización y la comprensión de cómo se relacionan las diferentes estructuras de datos entre sí.” (Google Cloud, 2024)

**Figura 7**

*Ejemplo Diagrama de Base de Datos SQL*



Fuente: (Learn Microsoft, 2023)

El lenguaje de programación SQL no solo permite realizar operaciones de selección, inserción, actualización y eliminación de datos, sino también operaciones administrativas sobre las bases de datos. Por tanto, se trata de un lenguaje completo de bases de datos que va más allá de la recuperación de la información.” (Universidad Europea, 2021).

Como se observa SQL se entiende como un sistema gestor de datos que además permite operaciones administrativas como la creación y modificación de estructuras propias de este lenguaje como tablas, vistas, *Jobs*, *ETLs*, Procedimientos almacenados e índices, también de la administración de usuarios, roles y permisos, esto por mencionar algunas de sus funciones. Por lo cual este lenguaje asegura un control sobre el acceso y la integridad de los datos.

### **Ventajas**

- *SQL* es bastante aceptado en la industria porque mantiene un estándar muy similar a los diferentes sistemas de gestión de bases de datos relacionales que existen en el mercado.
- Su lenguaje es fácil de entender en cuanto a su sintaxis, lo que lo hace que la gestión por parte de la persona Administradora de Base de datos sea más sencilla.
- Al ser un lenguaje tan reconocido en la actualidad, se puede encontrar una cantidad de recursos, tutoriales, guías y ejemplos disponibles en la red lo que hace que entender y solucionar problemas en *SQL* sea más sencillo para las personas administradoras de las bases de datos.

### **Desventajas**

- A pesar de que *SQL* es reconocido y tiene mucha documentación para su implementación y gestión, en algunos casos las consultas o los procesos que se programan por parte de las personas administradoras de base de datos no son los óptimos y pueden tener problemas a la hora de optimizar. En algunos casos, las bases de datos no están bien diseñadas o por su extensa cantidad de datos pueden presentar inconvenientes.

- El tema de los costos es relevante para las organizaciones porque acceder a todas las funcionalidades de *SQL Server* implica gastos. Por ejemplo, si se quiere utilizar *SQL* en la nube, como en Azure, se incurre en costos basados en el uso, como el número de consultas realizadas o el tipo de plan contratado para la organización.
- *SQL* cuando se emplea localmente, dependiendo del enfoque que se dé o la cantidad de datos que se tengan pueden ser un problema, ya que va a necesitar servidores potentes y una buena infraestructura. Además, al ser este lenguaje pertenece a *Microsoft*, se integra bien con estos productos, pero pueden presentarse problemas de compatibilidad con *software libre*.

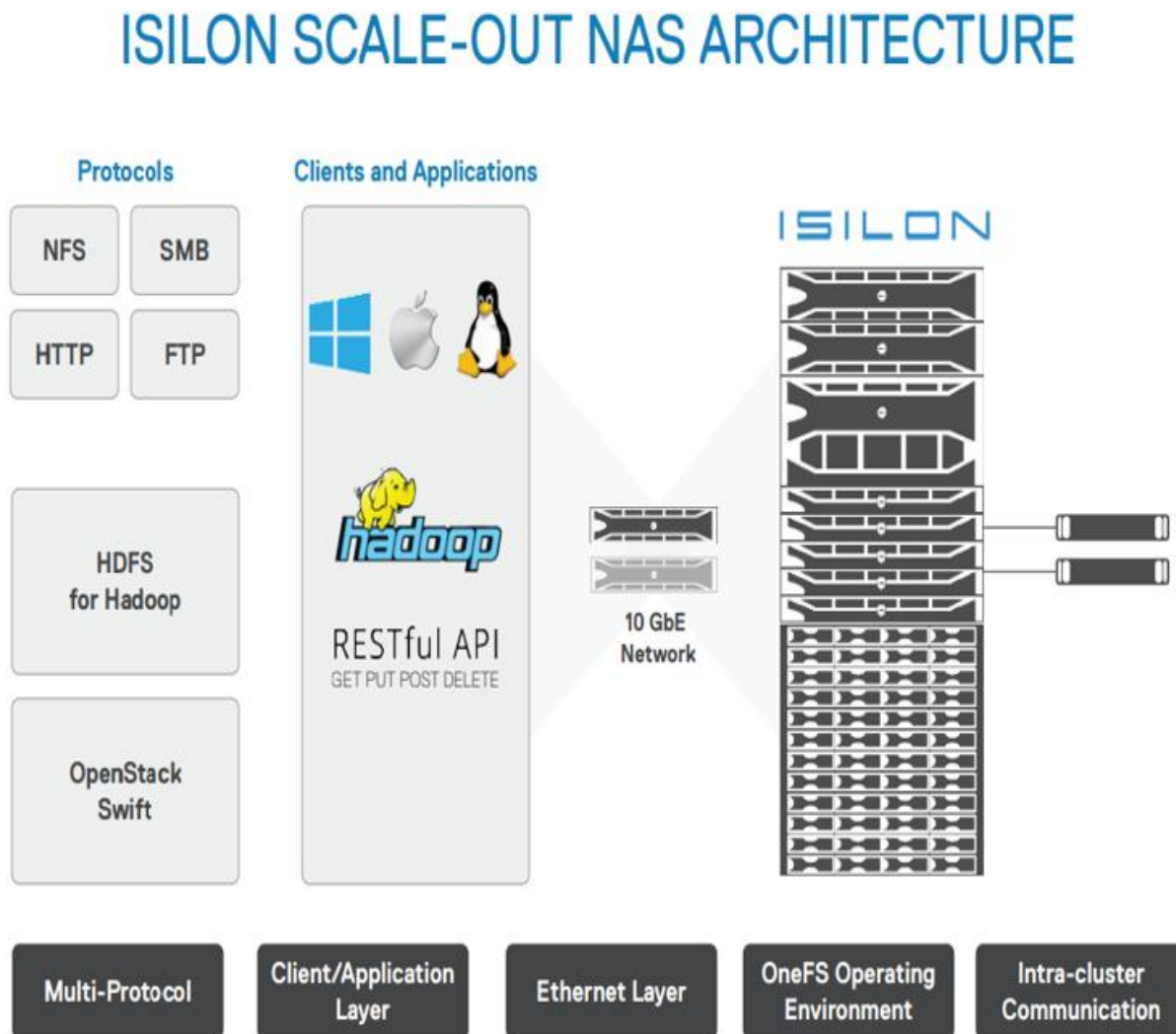
En la actualidad las bases de datos *SQL* tiene gran popularidad en muchos sectores como la banca, el comercio, la contabilidad, entre otros, esto por su capacidad de ser flexible, integrar y expandir los sistemas según la necesidad, el negocio de la organización. Este lenguaje permite que los datos sean accesibles, seguros y organizados de manera eficiente.

### ***Dell EMC Isilon***

*Isilon* es una plataforma líder en almacenamiento *NAS* escalable, ideal para una variedad de cargas de trabajo de datos no estructurados, archivos compartidos, contenido multimedia y análisis de datos. Ofrece soluciones de almacenamiento *all-flash*, híbridas y de archivo que se integran en un único *clúster* y funcionan con el sistema operativo *OneFS*. Su diseño modular permite escalar de decenas de *terabytes a petabytes* en un solo clúster (DELL, 2019).

Figura 8

Arquitectura NAS de Expansión Horizontal de Isilon



Fuente: (DELL, 2019)

Como se puede observar en la figura anterior, en la infraestructura de *Isilon NAS* para conectar todo se necesita de una red *Ethernet*, donde se pueden conectar diferente tipo de usuarios como *Windows, Mac o Linux o RESTful API*. Además, se puede ver que se usa comunicación entre nodos para mantener los datos sincronizados, gestiona todo con *OneFS*, soporta varios protocolos de acceso. A continuación, se va a detallar cada concepto de estos.

## Clustrer

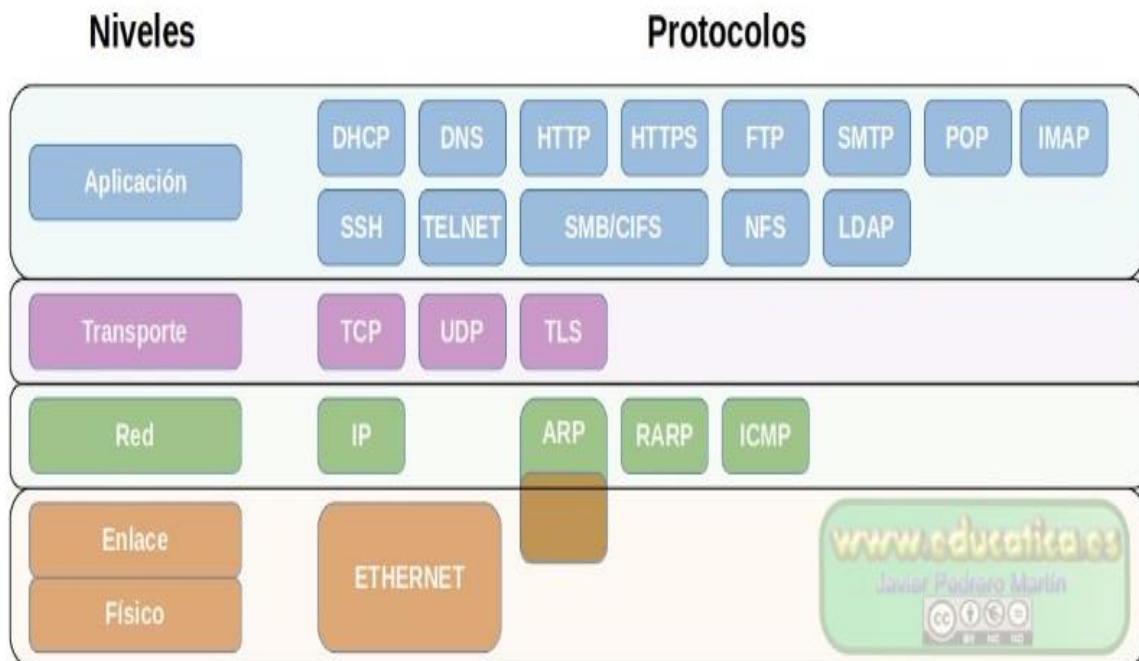
Un clúster de servidores es un grupo de servidores y otros recursos independientes que funcionan como un único sistema. Los servidores suelen estar situados muy cerca unos de otros y están interconectados por una red dedicada. De este modo, los clusters proporcionan un recurso de procesamiento de datos centralizado. Un cliente habla con el grupo de servidores como si fuera una sola máquina” (x5servers, 2024).

En el contexto de un *Isilon*, se puede entender que es un *cluster* de uno o varios servidores de almacenamiento de esta marca que se ven como un solo recurso, en este caso la conexión que usan estos nodos para conectarse entre sí, a esto se le llama *intra-cluster communication* o en español la comunicación intra-cluster.

## Multiprotocolo

### Figura 9

*Ejemplo de Protocolos*



Fuente: (EDUCATICA, 2024)

En un entorno multiprotocolo como es el caso de un servidor *Isilon*, tanto usuarios de *UNIX* como de *Windows* pueden acceder al mismo archivo usando diferentes protocolos, compartiendo la misma estructura de directorios. El problema es cómo se autentican las identidades y qué permisos se aplican. Antes, los usuarios solo usaban un proveedor de autenticación, pero en un sistema multiprotocolo, como *LDAP* (Protocolo Ligero de Acceso a Directorio) y *Active Directory* conectados a un solo *NAS*, esto cambia. Un usuario de *UNIX*, por ejemplo, podría autenticarse con *Active Directory*, y si tiene cuentas en ambos sistemas, *OneFS* las puede mapear y vincular (Delltechnologies, 2023).

El sistema operativo *OneFS* es compatible no solo con los protocolos *SMB* y *NFS*, sino también con otros como *HTTP* (Protocolo de Transferencia de Hipertexto), *HDFS* (Sistema de Archivos Distribuido de Hadoop), *S3* (*Amazon Simple Storage Service*) y *FTP* (Protocolo de Transferencia de Archivos). Debido a esta amplia gama de protocolos admitidos, se vuelve necesario contar con un modelo de permisos unificado, independientemente del protocolo utilizado, los permisos se gestionan de manera uniforme, evitando posibles conflictos y simplificando la administración en entornos donde múltiples protocolos están en uso simultáneamente (Delltechnologies, 2023).

### ***OneFS***

*OneFS* ofrece la inteligencia detrás de los sistemas de almacenamiento de escalamiento horizontal. Combina las tres capas de arquitecturas de almacenamiento tradicionales (sistema de archivos, administrador de volúmenes y protección de datos) en una capa de software unificada, con lo que se crea un sistema de archivos inteligente que abarca todos los nodos dentro de un clúster (DELL, 2022).

En relación con el texto anterior se entiende que *OneFS* es un sistema operativo que permite que administrar los nodos dentro de un *cluster* de almacenamiento *Isilon*, por lo cual permite que los servidores se vean como una sola, esto puede simplificar de alguna manera la administración del servidor por parte de los administradores del servidor.

### **S3 (*Amazon Simple Storage Service*)**

En el contexto de esta investigación, el protocolo S3 (*Amazon Simple Storage Service*) es bastante importante porque se tiene que integrar con el *servidor Isilon* existente en la infraestructura del Poder Judicial y su uso en entornos de microservicios lo que lo convierte un elemento clave, explorando su importancia y aplicación en entornos de almacenamiento. A continuación, se explicará en detalle este protocolo.

“Amazon S3 es un servicio de almacenamiento de objetos que almacena datos como objetos dentro de buckets. Un *objeto* es un archivo y cualquier metadato que describa ese archivo” (Amazon Web Services, 2024). Entonces se puede decir que *un objeto de Amazon S3* es la unidad básica que implementan los servicios AWS (*Amazon Web Services*) para guardar la información, para esto necesita de dos elementos, un archivo y un metadato.

Un archivo es cualquier dato que se pueda almacenar, en este caso puede ser un documento, una imagen, un audio o video. Por otra parte, hay un metadato que es la información de dicho archivo el cual se puede decir que describe las características del archivo, pueden ser información como nombre, tipo de archivo, fecha de creación o cualquier otro dato importante que necesite este protocolo para administrar los archivos.

### ***Buckets***

“Un bucket es un contenedor para objetos almacenados en Amazon S3. Puede almacenar cualquier cantidad de objetos en un bucket y puede tener hasta 100 buckets en su

cuenta” (Amazon Web Services, 2024). Entonces, se entiende que los objetos están formados por los dos elementos mencionados anteriormente: el archivo y sus metadatos. *Los Buckets* a su vez son contenedores, que almacenan los objetos dentro de un servidor dedicado al almacenamiento, como un *servidor Isilon*. Esto permite que los datos estén bien organizados por medio de esta estructura y que el acceso sea eficiente.

Es importante mencionar que el protocolo S3 no se limita únicamente a su uso en la nube, también puede ser implementado localmente en un *servidor Isilon*. Sin embargo, es fundamental tener en cuenta la compatibilidad con el sistema operativo *OneFS*, esto porque dependiendo de la versión puede que no sea compatible con el S3.

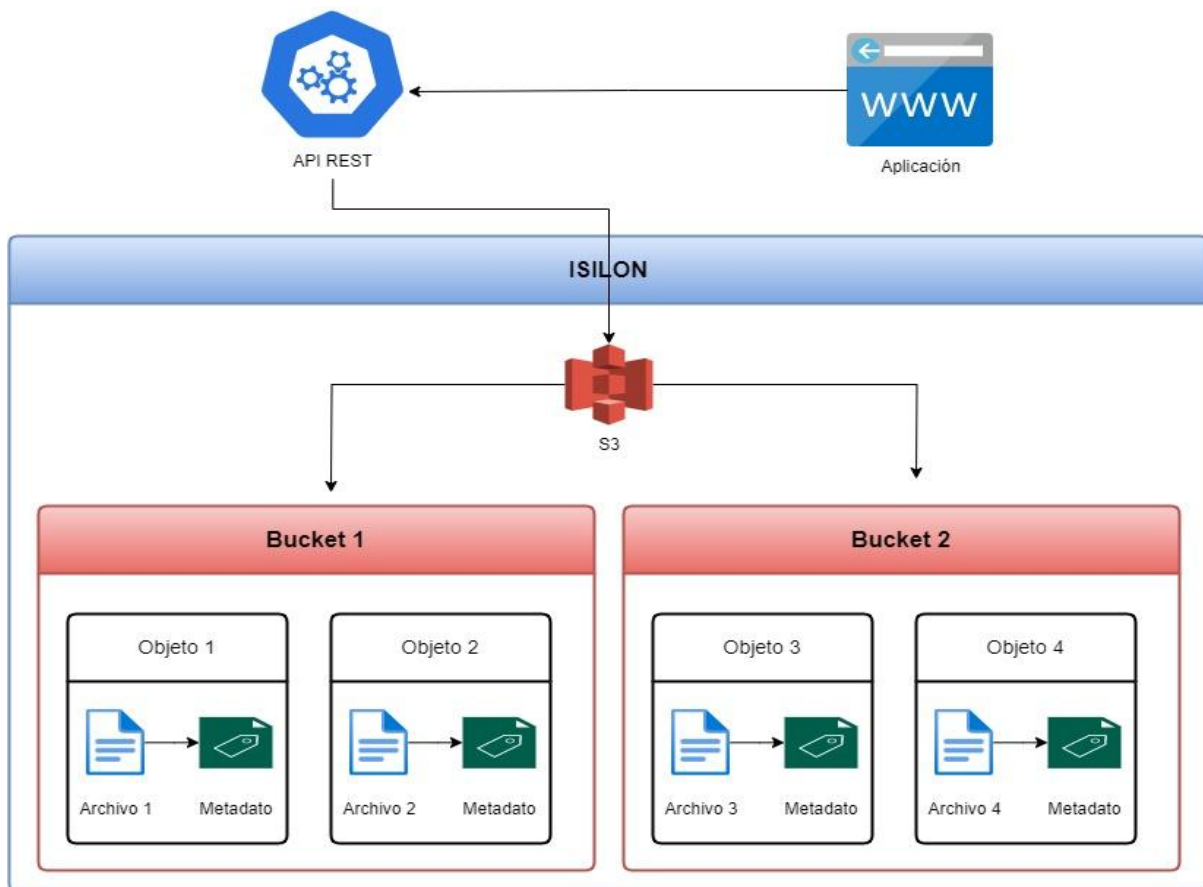
### ***API de REST de Amazon S3***

La arquitectura de Amazon S3 está diseñada con un lenguaje de programación neutro y utiliza interfaces admitidas por AWS para almacenar y recuperar objetos. Puede acceder a S3 y AWS mediante programación mediante la API de REST de Amazon S3. La API de REST es una interfaz HTTP para Amazon S3.” (Amazon Web Services, 2024).

Se entiende que la *API REST de Amazon* permite realizar operaciones como consultar, actualizar, agregar y eliminar objetos de los *buckets* lo cual facilita a los programadores la integración de este protocolo, para gestionar y automatizar los recursos almacenados.

**Figura 10**

*Integración de Aplicación con Isilon API REST y Protocolo S3*



### **SDK de AWS**

AWS tiene la opción de instalar los SDK (*Kits de Desarrollo de Software*) los cuales son compatibles con varias plataformas y lenguajes de programación como, por ejemplo, *Java, Python, Ruby, .NET, iOS y Android*. Estos Kits facilitan el desarrollo de aplicación para las personas programadoras porque ofrecen bibliotecas y código que permite usar el *API REST de Amazon S3* para gestión de forma segura los *buckets* que se encuentra en los servidores de almacenamiento. Además, este Kit de desarrollo ayuda con temas de gestión de errores, los reintentos automáticos, la encriptación de los datos, entre otras funcionalidades que ya bien integradas para automatizar mejor la conexión entre empresa y el servicio de almacenamiento de datos (Amazon Web Services, 2024).

### ***Puntos de Acceso de S3***

Los puntos de acceso en *Amazon S3*, también conocidos como *endpoints* de red, son diseñados con políticas de acceso específicas que determinan cómo se puede interactuar con los datos a través de este tipo de conexión. Estos puntos de acceso están vinculados a los *buckets* y permiten realizar operaciones con objetos de S3, como las acciones de *GetObject* y *PutObject*. Una ventaja de este tipo de conexión es que facilita el acceso a los datos, especialmente en escenarios donde se manejan grandes cantidades de datos compartidos dentro de Amazon S3 (Amazon Web Services, 2024).

Estos *endpoints* tienen su propia política individual, lo que permite un control sobre cómo se manejan las solicitudes de acceso a los *buckets*. Además, se puede configurar para bloquear el acceso público, lo cual garantiza que cada organización o cliente pueda acceder a sus datos y no se permitan conexiones que no son necesarias, que pueden atentar con la información que se encuentra guardada en los servidores de almacenamiento. También se puede hacer que solo se acepte únicamente la conexión desde una nube virtual privada (VPC), lo cual ayuda a mejorar la seguridad porque de alguna manera es una capa más que protege la información (Amazon Web Services, 2024).

### ***Listas de Control de Acceso (ACL)***

En cuanto a la seguridad, además del control de los *endpoints* existe en Amazon S3 se puede utilizar las Lista de control de Acceso, las cuales permiten otorgar permisos de lectura y escritura a las personas usuarias que estén autorizadas a manipular los *buckets* y los objetos. Pero según la documentación, estas listas solo se deben usar si es necesario controlar el acceso a los objetos de forma Individual. La recomendación es utilizar *las IAM Identity and Access Management*, esto permite gestionar de una mejor forma los accesos, los permisos de las personas usuarias y los recursos utilizados dentro de la infraestructura de AWS (Amazon Web Services, 2024)

Amazon S3 es un protocolo o servicio muy completo para el almacenamiento en la nube y para una implementación local con un *servidor Isilon*, es altamente escalable, seguro y flexible, permite a las empresas gestionar sus datos de manera eficiente. Gracias a su arquitectura basada en la *API REST* y la integración con los *SDK de AWS*, los desarrolladores pueden gestión los datos en los servidores de almacenamiento fácilmente, aprovechando los Kits de desarrollo que automatizan procesos complejos y la gestión de errores. Esto convierte a S3 en una solución ideal para una amplia gama de aplicaciones.

### **Ingeniería de Software**

“La ingeniería del software es una disciplina de la ingeniería cuya meta es el desarrollo costearle de sistemas de software. Este es abstracto e intangible.” (Sommerville, Ingeniería del software Séptima edición, 2005). Es muy interesante esta definición de este autor porque nos ayuda a entender que la meta de esta ingeniería es crear sistemas que sean eficientes y económicamente factibles. Además, se entiende que el *software* no es producto que se pueda tocar, pero sí se puede manipular, ya que al final los sistemas están formados por código y lógica, lo cual hace destacar contar otros productos que ofrecen ingenierías que son más tradicionales, lo cuales son concretos y tangibles.

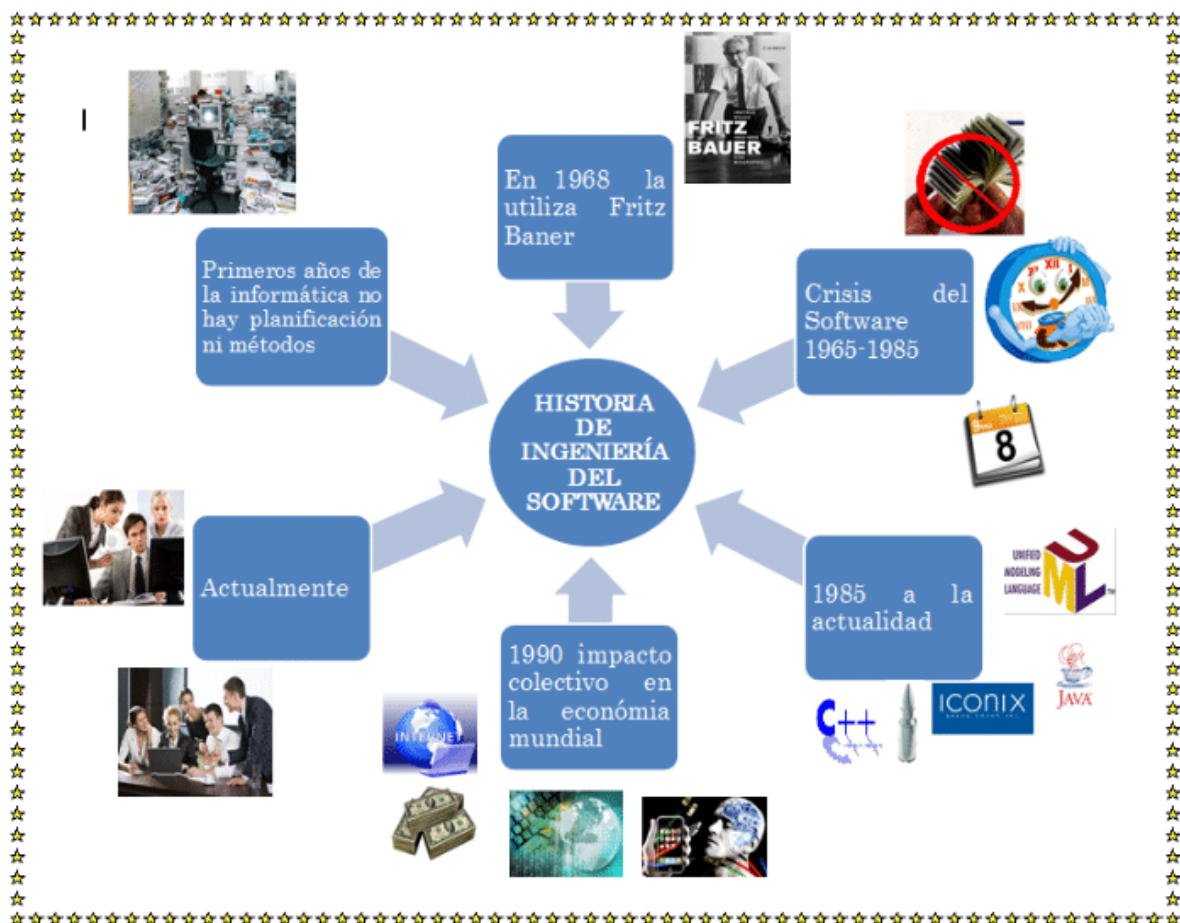
Es importante mencionar que la ingeniería de *software* son un conjunto de procesos que permiten el desarrollo de sistemas para que sean seguros, eficientes, confiables, de alta calidad y sobre todo que cumplan con las necesidades solicitadas por las organizaciones o en algunos casos solucionar problemas que ya se están presentando en un software en específico.

Para entregar un producto de calidad, la ingeniería de *software* debe integrar conocimientos como la gestión de proyectos, el uso de metodologías ágiles, la aplicación de principios y buenas prácticas. De este modo, las etapas de análisis, planificación, diseño, desarrollo, revisión y mantenimiento se establecen como los pilares fundamentales para

satisfacer los requisitos de las personas u organizaciones que solicitan cambios o mejoras en sus negocios.

**Figura 11**

*Historia de la ingeniería de Software*



Fuente: (Mi Carrera Universitaria, 2024)

### Ciclo de Vida del Software

“También denominado el proceso del desarrollo de software, el ciclo de vida del software nos permite entablar una serie de procedimientos mediante las cuales se procede para la realización del mismo.” ( OKHOSTING SC, 2024).

El ciclo de vida del software se basa en descomponer un proceso complejo en varias etapas, lo que permite mantener un orden lógico y metódico para el cumplimiento de los

requisitos y objetivos establecidos en el desarrollo del *software*. Este enfoque facilita la organización y la planificación del trabajo, sino que también asegura que el producto final cumpla con las expectativas esperadas por la persona usuaria o la organización.

Además, al entender y aplicar correctamente estas fases, es posible dividir de manera más efectiva las tareas entre los colaboradores y grupos de trabajo de un proyecto, lo que optimiza el tiempo de desarrollo y mejora la eficiencia al crear aplicaciones.

### Figura 12

*Ciclo de Vida del Software*



Fuente: (Evotic, 2024)

En la figura anterior se ilustran las principales etapas del ciclo de vida de *software*, las cuales se van a explicar detalladamente a continuación para tener una comprensión más clara del proceso de desarrollo.

## ***Planificación***

La primera etapa del ciclo de vida del *software* es la planificación y tal vez una de las más importantes, ya que es la base de todo el desarrollo, donde se busca estructurar el proyecto de manera efectiva. En esta fase, es fundamental contar con un gestor de proyectos con experiencia en el área, quien se encargará de definir los objetivos del proyecto, el personal necesario, los alcances, los tiempos para cada una de las siguientes etapas.

Además, en esta fase se realiza una estimación del presupuesto requerido para llevar a cabo el proyecto. Hoy en día, la planificación es apoyada por aplicaciones que facilitan la organización de tareas, la gestión del tiempo y la asignación de responsabilidades a las personas encargadas de cada tarea.

En esta etapa también se consideran los posibles riesgos asociados al desarrollo del *software*. Esta previsión permite a las empresas establecer un margen de seguridad tanto en el tiempo como en el presupuesto, lo que ayuda a evitar retrasos en la fecha de entrega del proyecto. Al anticipar estos riesgos y prepararse adecuadamente, se reduce la probabilidad de que surjan imprevistos que puedan comprometer el éxito del proyecto.

## ***Análisis***

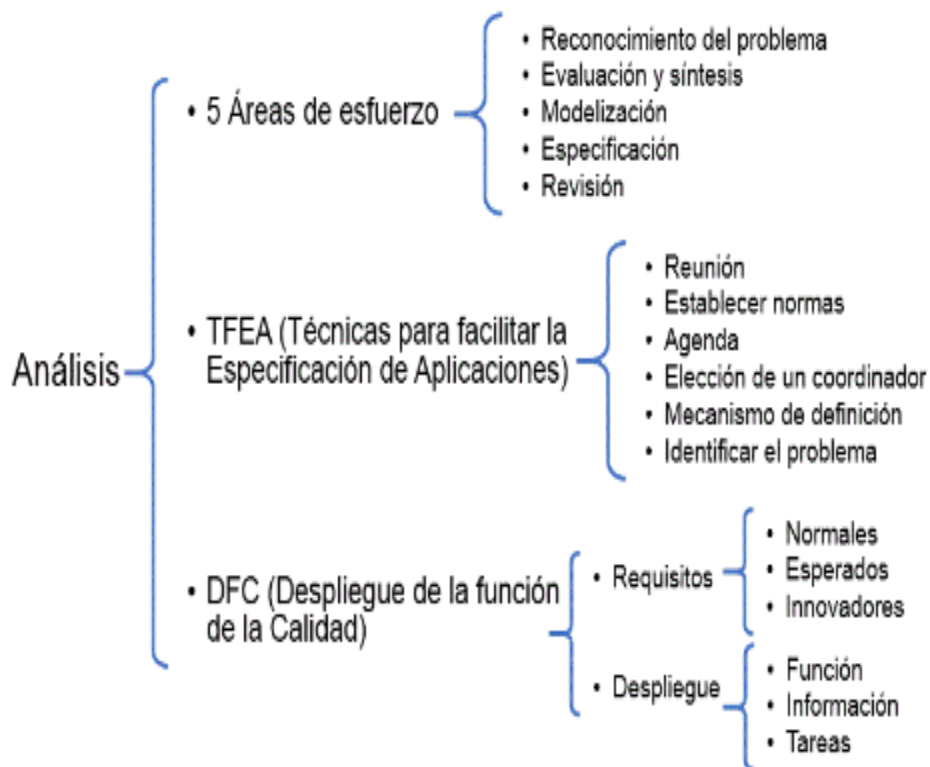
Esta es la segunda etapa del ciclo de vida del *software* y una de las más delicadas de todas, ya que en ella se definen los requisitos funcionales y no funcionales del sistema. Generalmente, los responsables de este proceso son los analistas de sistemas, quienes desempeñan un papel crucial en la correcta interpretación de las necesidades del proyecto.

En esta etapa se realizan una serie de reuniones con las partes interesadas, como usuarios, clientes, el gestor del proyecto, desarrolladores y demás personal involucrado, para comprender a fondo las necesidades del sistema y del negocio. Todo este proceso es cuidadosamente documentado para asegurar que no se pierda ningún detalle importante.

Una vez que se tiene una visión clara del negocio y de los requisitos, se procede a definir las funcionalidades del sistema, las cuales se dividen en historias de usuario. Estas historias permiten especificar de manera precisa lo que el usuario final espera del sistema. Es fundamental que sean claras, sin lugar a ambigüedades, ya que cualquier error o malentendido en esta fase podría resultar en un desarrollo incorrecto, dado que estas son la guía de implementación por parte de los desarrolladores.

**Figura 13**

*Componentes Claves del Análisis*



Fuente: (UAEH, 2024)

**Diseño**

“El diseño arquitectónico y detallado de software determina la calidad de los productos de software que se entregan. Es por esto por lo que la industria del software ha

venido sistematizando heurísticas de diseño por medio de lo que se ha llamado principios y patrones de diseño” (UCR, 2020).

Teniendo en cuenta la referencia anterior, se puede observar que esta fase también es bastante amplia, ya que se enfoca en el diseño del software. A lo largo de los años, ha sido necesario ser han creado buenas prácticas, junto con principios y patrones de diseño, que son soluciones que ya se han implementado, lo que ayuda a solucionar problemas en el desarrollo de software.

### Figura 14

#### *Elementos del Diseño del Software*



Fuente: (Universidad Nacional Autónoma de México, 2022)

En la figura anterior se pueden observar varios elementos del diseño de *software*. El arquitectónico se encarga de la estructura y los patrones del sistema, mientras que el diseño detallado se enfoca en la implementación de los componentes, incluyendo la interfaz de usuario. Además, se incluye el diseño de la base de datos, que define las relaciones, entidades y otros objetos clave para su implementación.

El versionado de la base de datos abarca el control de versiones, migraciones y respaldos. Finalmente, la documentación de todos estos elementos asegura que el personal esté al tanto del desarrollo del sistema y proporciona un respaldo para resolver posibles errores futuros.

### ***Desarrollo***

En la etapa del ciclo de vida del software, es donde se lleva a cabo la implementación, para esto se tiene apoyo de las fases anteriores donde ya se tiene un diseño del sistema o la aplicación, así como su arquitectura y las historias de usuario para guía de los desarrolladores.

En esta fase se tiene que realizar varias tareas como la escritura del código fuente del software para la creación del back-end y el *Font-end*, configuraciones varias en los servidores y las herramientas de desarrollo, la creación de la interfaz de usuario, las pruebas unitarias y la solución de errores todo lo anterior para cumplir con los requisitos funcionales y no funcionales del producto que se va a entregar.

Como se puede observar en la siguiente figura 14, uno de los puntos clave al crear código es tener claras las buenas prácticas tanto de los lenguajes utilizados como de los marcos de trabajo aplicados durante el desarrollo. Además de esto, es crucial seguir las mejores prácticas en el desarrollo de bases de datos y mantener estándares de documentación o comentarios que faciliten la comprensión del código.

El código limpio y bien documentado no solo facilita la resolución de problemas en entornos de producción, sino que también contribuye a la eficiencia en el mantenimiento y evolución del software. Por lo cual, los desarrolladores y las empresas dedicadas a este trabajo deberían aprovechar las herramientas disponibles en el mercado para mejorar la calidad del código.

**Figura 15***Ejemplos de Buenas Prácticas de la Fase de Desarrollo*

Fuente:(Universidad Nacional Autónoma de México, 2022)

### ***Pruebas***

“Las pruebas de software permiten controlar la calidad y funcionalidad de cualquier producto que se desarrolle; son la mejor garantía de que este no presenta fallos y se comporta adecuadamente” (Universidad Internacional de La Rioja, 2023).

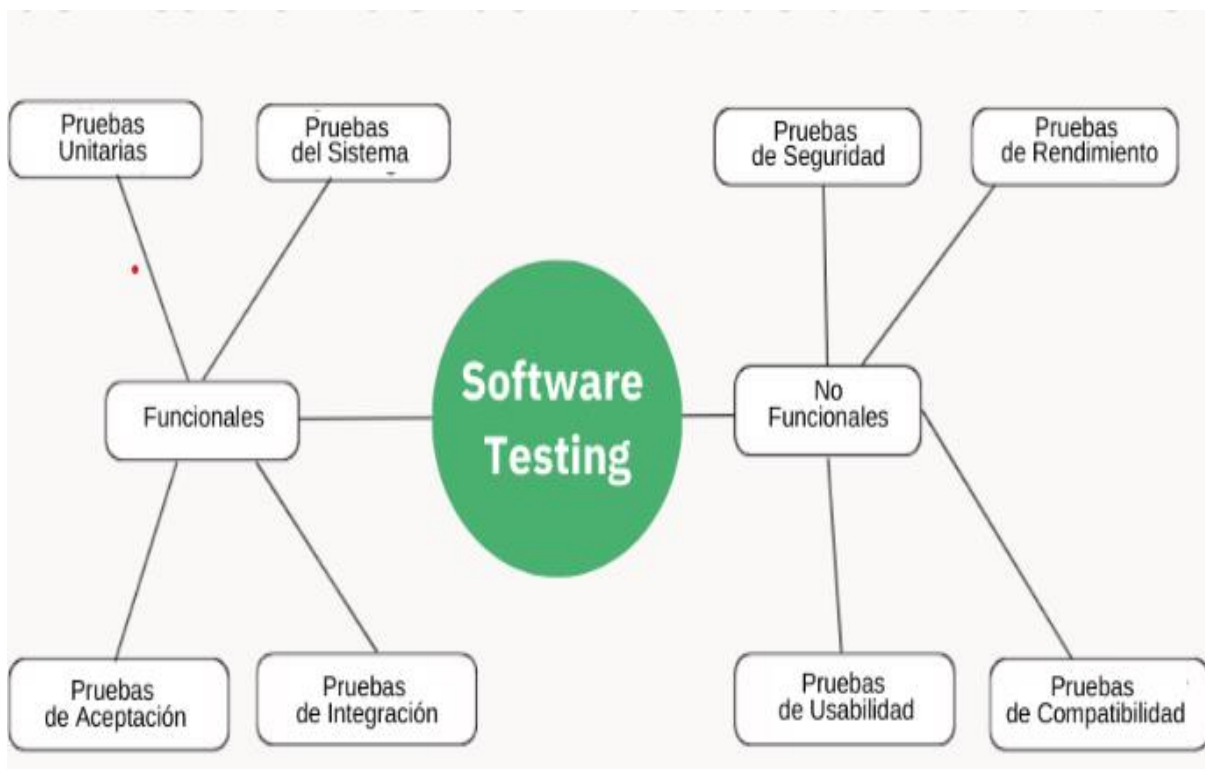
Según la cita anterior, las pruebas de software garantizan que la aplicación o sistema desarrollado no presenten fallos en producción. Sin embargo, las pruebas son esenciales para

validar los requerimientos del software, utilizando historias de usuario para verificar que el desarrollo cumpla con sus criterios de aceptación.

En muchas empresas de desarrollo, existe un departamento dedicado exclusivamente a realizar estas pruebas, conocido como QA (*Quality Assurance*) o Aseguramiento de la Calidad en español. Estas personas se dedican a realizar diferentes tipos de pruebas a los sistemas para aprobar o no la puesta en producción de estas aplicaciones.

### Figura 16

*Clasificación de las pruebas de Software*



Fuente: (QAlified , 2023)

En la figura anterior se ilustran los diferentes tipos pruebas de software, los cuales se van a explicar detalladamente a continuación para tener una comprensión más clara del proceso de desarrollo.

**Pruebas Unitarias.** Estas pruebas las crean las desarrolladoras en fase de desarrollo, Se encargan en verificar individualmente el correcto funcionamiento de los métodos de un software. Su objetivo es detectar errores en las primeras etapas del desarrollo y permiten mantener la integridad del código.

**Pruebas de Integración.** Estas pruebas permiten verificar las funcionalidades que interactúan entre sí dentro del software. Estas pruebas identifican errores y aseguran que los módulos del sistema que se relacionan funcionen correctamente.

**Pruebas del Sistema.** A diferencia de las pruebas de integración, estas son más holísticas, ya que se enfocan en verificar el sistema por completo en un entorno real simulado, en otras palabras, en un ambiente reproductivo que es lo más parecido al de producción.

**Pruebas de Aceptación.** Una vez que se realizan las pruebas del sistema, se pueden realizar las de aceptación, que se enfocan en verificar que se cumpla con los requisitos y expectativas de la persona usuaria.

**Pruebas de Seguridad.** Estas pruebas de seguridad son muy importantes para evitar algún tipo de vulnerabilidad en el software. Estas pruebas aseguran la protección de datos, temas legales, verificar la versión que los sistemas estén actualizados en cuanto base de datos, paquetes, *framework*. Todo lo anterior ayuda a evitar incidentes que puedan atentar contra la integridad y confianza hacia la empresa dueña de la aplicación.

**Pruebas de Rendimiento.** Las pruebas de rendimiento son las encargadas de verificar que el software pueda soportar sin que deje de funcionar en las horas de más concurrencia, por lo cual se realizan test de carga o de rendimiento, los cuales se realizan con herramientas que permiten simular estos escenarios.

**Pruebas de Usabilidad.** Estas pruebas se enfocan en evaluar la interfaz de usuario de los sistemas o aplicaciones, donde el objetivo es verificar que se pueda utilizar de forma sencilla el software y que se pueda entender. Además, verifican que el sistema cumpla en cuanto a los términos de accesibilidad, intuición y satisfacción del cliente.

**Pruebas de Compatibilidad.** Estas pruebas se centran en verificar que el sistema se puede ejecutar en diferentes entornos, dispositivos, Sistemas operativos y diferentes exploradores, según sea el caso. Lo que permite que el software sea más accesible para la persona usuaria.

### **Paradigmas de Programación**

Laprovittera (2024) afirma lo siguiente:

Define un modelo o patrón con reglas y pautas que todo desarrollador debe seguir para resolver una situación problemática. La palabra Paradigma no tiene una relación directa con la programación. Existen numerosos paradigmas de programación: modular, procedimental, estructurado, orientado a eventos, a objetos, etcétera.

Con base en lo anterior se puede interpretar que son diferentes enfoques para escribir el código de un programa. Esto quiere decir que hay diferentes tipos de paradigmas en donde cada uno tiene su propio enfoque y estilo, debido a que ofrecen diferentes formas para poder estructurar el programa, depende del problema que se busca resolver se elige el tipo de paradigma que se debe usar.

#### ***Paradigma Declarativo***

“Modelos de Desarrollo: Funcional, Lógico y de Flujo de Datos. Se construye señalando hechos, reglas, restricciones, ecuaciones, transformaciones y otras propiedades derivadas del conjunto de valores que configuran la solución” (Universidad de Salamanca , 2007, pág. 4). Quiere decir que este paradigma se centra en cómo se debe realizar una tarea,

básicamente es decirle al programa lo que tiene que hacer, ya que no se le debe especificar cómo hacerlo, debido a que el lenguaje de programación se encarga de hacerlo sin la necesidad de especificarle los pasos. Los lenguajes más comunes de este paradigma son *SQL, HTML, Prolog, Lisp, Haskell*.

### ***Paradigma Imperativo***

El paradigma imperativo es un “modelo abstracto que consiste en un gran almacenamiento de memoria donde la computadora almacena una representación codificada de un cálculo y ejecuta una secuencia de comandos que modifican el contenido de ese almacenamiento” (Universidad de Salamanca , 2007, pág. 4). Lo anterior hace referencia a que se especifica el paso a paso de cómo debe realizarse el programa, el programador es quien indica las acciones que se deben realizar. Los lenguajes más comunes usados en este paradigma son *C, C++, Java, Python*.

Por otro lado, es importante destacar que la programación imperativa se puede clasificar en tres estilos diferentes de programación, los cuales son el procedimental, modular y el estructurado.

**Programación Procedimental.** “En este paradigma de programación, la atención se centra en el procesamiento, es decir, en las acciones necesarias para resolver un determinado problema.” (Universidad de Málaga , 2007). El propósito de este estilo es que el código del programa sea más claro, optimizando el rendimiento al reducir repeticiones que son innecesarias, y a la vez va a facilitar la comprensión, el mantenimiento y la detección de errores.

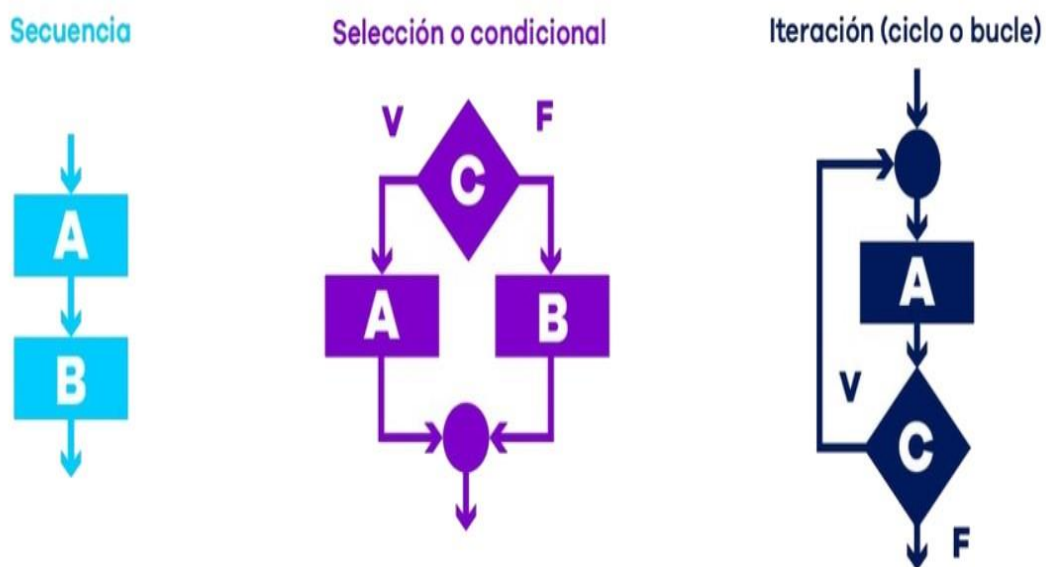
**Programación Modular.** “La programación modular está basada en la técnica de diseño descendente, que como ya vimos consiste en dividir el problema original en diversos subproblemas que se pueden resolver por separado, para después recomponer los resultados y

obtener la solución al problema” (Universitat de Valencia, 2004). En otras palabras, se encarga de dividir el programa en módulos, cada parte se encarga de una tarea en específico, esto con el propósito de trabajarlos de forma dividida, con el fin de que la comprensión del código se vuelva más sencilla.

**Programación Estructurada.** “La programación estructurada es una teoría orientada a mejorar la claridad, calidad y tiempo de desarrollo utilizando únicamente subrutinas o funciones” (Unir FP, 2022). Este paradigma se centra en mejorar la calidad, claridad y acelerar el tiempo de desarrollo, utilizando tres bloques básicos los cuales son Secuencial, de selección y de iteración.

### Figura 17

*Las Tres Estructuras Básicas de la Programación Estructurada*



Fuente: (Unir FP, 2022)

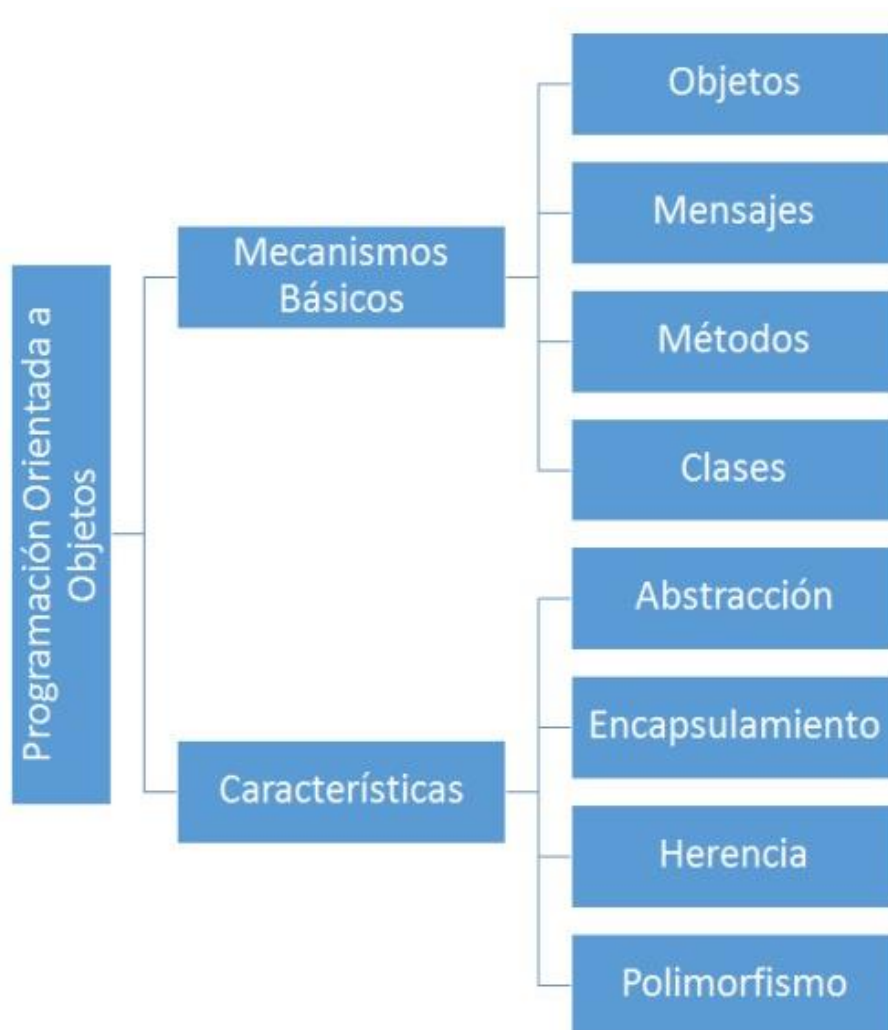
### *Programación Orientada a Objetos*

“Es un paradigma de programación que define los programas en términos de “clases de objetos”, objetos que son entidades que combinan estado (propiedades o datos),

comportamiento (procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto)". (García, 2021). En este caso hace énfasis en los objetos que son reales

**Figura 18**

*Programación Orientada a Objetos*



Fuente: (García, 2021)

A continuación, se presentarán los componentes fundamentales de la programación orientada a objetos.

**Clases.** Son los datos definidos por el usuario, es donde se establece un modelo para la estructura de métodos y atributos. Es donde se define como se verán y qué harán, ya que

las clases incluyen campos representativos como atributos y métodos que definen comportamientos.

**Objetos.** Un objeto es un elemento real, el cual es una copia creada a partir de la clase, esto quiere decir que tiene las mismas características y habilidades que se definieron anteriormente en la clase.

**Métodos.** Los métodos son los comportamientos que tendrá el objeto. Esto quiere decir que es donde se define las acciones que deben realizar. Un claro ejemplo es que tengamos un objeto que es un carro y el método sería arrancar, ya que esa es la acción que queremos que haga el carro.

Es importante destacar que hay principales lenguajes usados en la programación orientada a objetos, a continuación, se podrán observar algunos de ellos.

**Java.** Es un lenguaje multiplataforma que permite realizar aplicaciones para todo tipo de entornos, para web, dispositivos móviles, aplicaciones de escritorio, servidor, etc (Universidad Nacional , 2024). Este lenguaje está orientado a objetos, ya que se basa en la creación de objetos que interactúan entre sí. Asimismo, es el lenguaje principal para el desarrollo de aplicaciones Android.

**C++.** “Es un lenguaje compilado que fue desarrollado en 1980 por Bjarne Stroustrup en los laboratorios At&T como una extensión orientada a objetos del lenguaje C” (Universidad de Valladolid , 2021). Se centra en organizar los códigos en clases y objetos, de esta forma facilita la creación del sistema. Asimismo, es un lenguaje muy conocido por trabajar cerca del hardware, permitiendo un mejor rendimiento y un control sobre la memoria.

**C#.** “Es un lenguaje de programación moderno, innovador, de código abierto, multiplataforma orientado a objetos y uno de los 5 principales lenguajes de

programación de GitHub” (Microsoft , 2024 ). Este lenguaje es perteneciente de Microsoft y es parte de su plataforma.NET actualmente tiene mucha popularidad en el mercado por lo cual muchas empresas lo utilizan en la actualidad.

**Python.** Es reconocido por múltiples fuentes de la industria como uno de los tres lenguajes de programación más utilizados, no solo en el desarrollo de aplicaciones, sino también en la automatización de tareas en infraestructura de información y en el análisis de datos (Universidad cenfotec , 2022). Es muy destacado por su versatilidad y amplios conjuntos de bibliotecas, esto lo hace adecuado para una gran variedad de aplicaciones, desde desarrollo web hasta la automatización. Es un lenguaje simple que le permite a los programadores escribir menos código para poder hacer más, en la actualidad es uno de los más usados para la implementación de inteligencia artificial con determinadas bibliotecas que ayudan con este tema.

**PHP.** “Es un lenguaje de programación que se ejecuta en el servidor Web permitiendo generar contenido interactivo y dinámico al usuario cliente.” (Universidad Veracruzana , 2017) Este tipo de lenguaje se incrusta en el *código HTML* y los programas se ejecutan en el servidor web, suele ser usado para la elaboración de aplicaciones y páginas web que sean interactivas, ya que es fácil conectarse a la base de datos.

### ***Programación Lógica***

“Un programa lógico consta de un conjunto de fórmulas lógicas que expresan propiedades satisfechas por un cierto problema” (Cardona Taltavull, 2015, pág. 10). Se basa en el uso de la lógica para resolver distintos problemas, esto quiere decir que, en lugar de dar instrucciones detalladas, se definen las reglas y relaciones. El sistema usa esas reglas para encontrar la solución por sí solo.

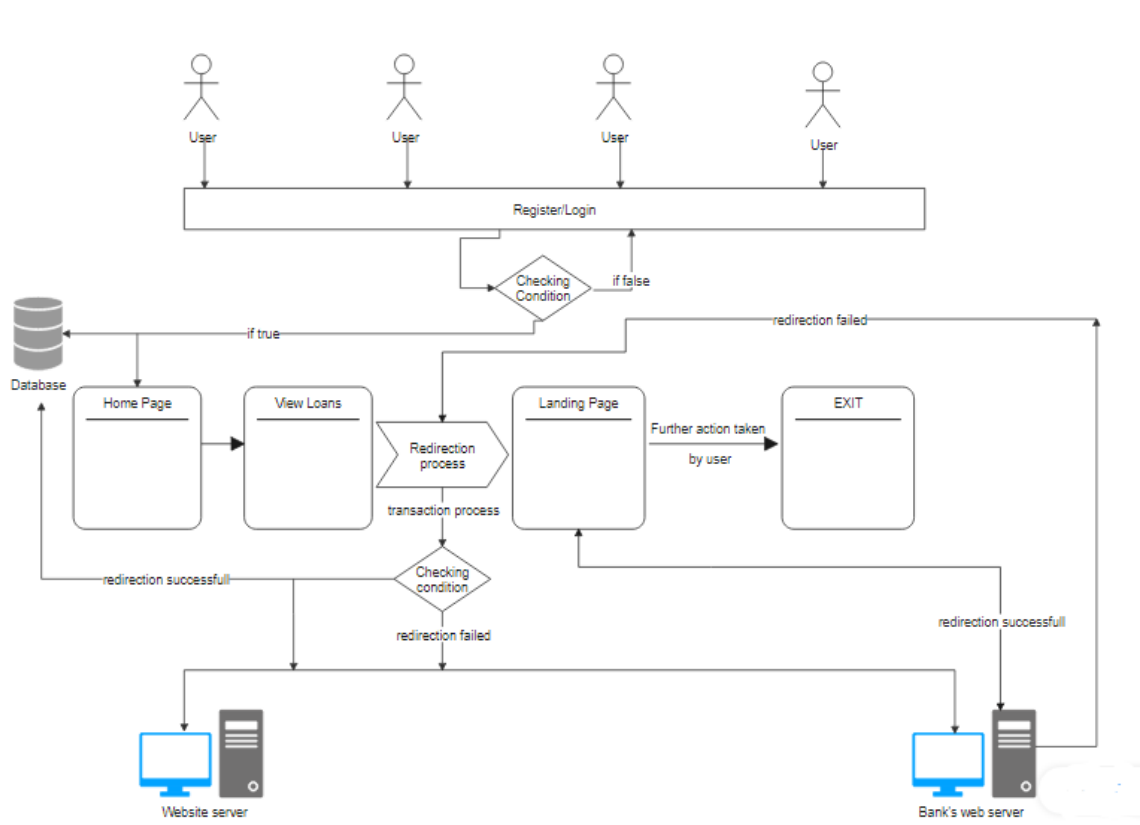
## Arquitectura de Software

La arquitectura de software y sistemas se refiere al diseño estructurado y la organización de componentes de software y hardware que forman un sistema informático o una aplicación, así como definir la forma en que los componentes interactúan entre sí. (Universidad Internacional de La Rioja, 2024).

En relación con la definición anterior se puede observar que la arquitectura es el diseño de la estructura de un software que permite diseñar y visualizar de una forma holística el sistema en relación con el tipo de arquitectura que se está usando, cómo se integran los componentes entre sí. En la siguiente figura se puede observar un ejemplo de arquitectura de software.

**Figura 19**

*Ejemplo de Arquitectura de software*



Fuente: (Edraw, 2024)

Es importante entender que la arquitectura y el diseño de software están relacionados, pero son conceptos distintos. La arquitectura de software define la estructura de alto nivel del sistema, mientras que el diseño aborda los detalles específicos de los componentes. Esto permite comprender mejor su papel en el desarrollo de software y cómo contribuyen a la creación de sistemas eficientes y mantenibles. En la siguiente tabla se puede observar las diferencias entre estos enfoques.

**Tabla 1**

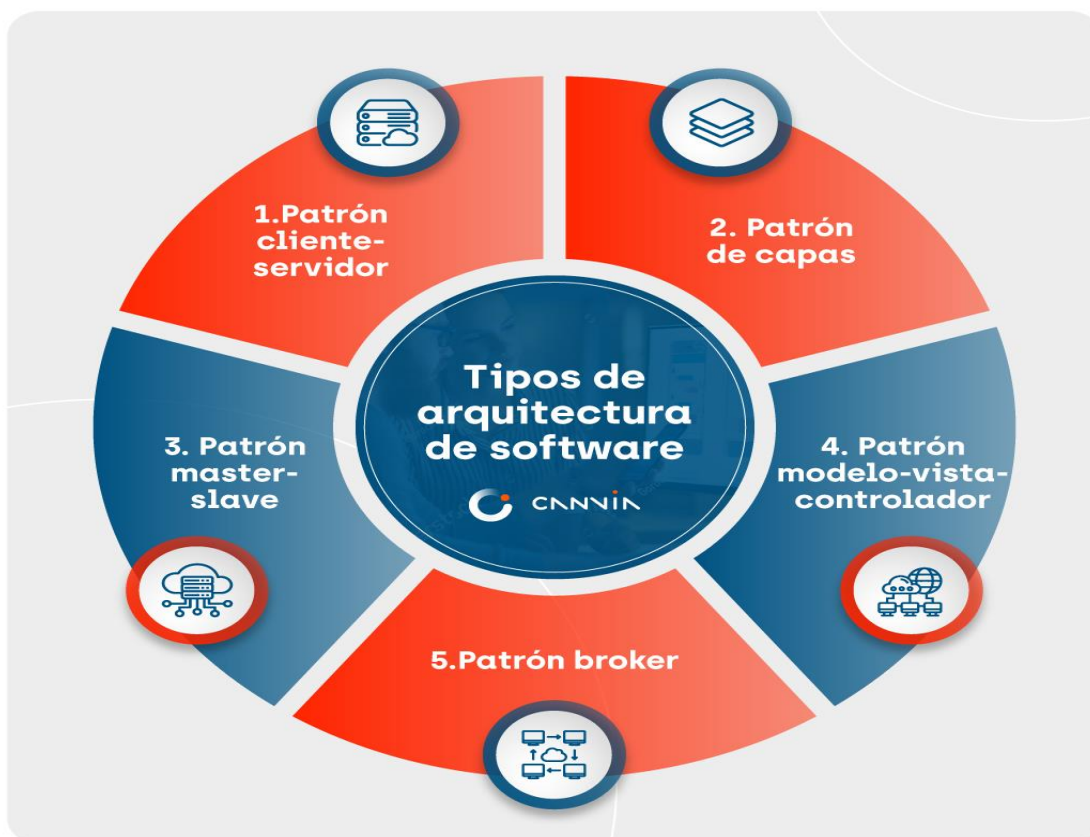
*Diferencias entre Arquitectura de Software y Diseño de Software*

<b>Criterio</b>	<b>Arquitectura de Software</b>	<b>Diseño de Software</b>
<b>Enfoque</b>	Estructura de alto nivel del sistema y decisiones tecnológicas clave.	Detalles internos de módulos y componentes específicos.
<b>Propiedades</b>	Propiedades fundamentales como escalabilidad y seguridad.	Propiedades detalladas como algoritmos y estructuras de datos.
<b>Proceso</b>	Decisiones estratégicas sobre componentes principales y relaciones.	Definición táctica de cómo se implementarán y comunicarán los módulos.

<b>Foco</b>	Gestión de la estructura global y coordinación de componentes.	Implementación concreta de funcionalidades y modularidad.
<b>Documentación</b>	Incluye diagramas de alto nivel como Diagramas de Componentes y Diagramas de Despliegue.	Incluye Diagramas de Clases, Diagramas de Secuencia y otros más detallados.

**Figura 20**

*Tipos de Arquitectura de Software*



Fuente: (Canvia, 2023)

En relación con la figura anterior, se puede observar que se mencionan algunos patrones de arquitectura de software. En la actualidad, existen muchos más patrones de desarrollo, lo que ofrece a las empresas dedicadas a esta área diversas opciones para la implementación, ajustándose a las necesidades específicas del software que desean desarrollar.

También es importante destacar que los patrones de diseño de software facilitan la comprensión de la estructura del software para los colaboradores de TI, proporcionando una guía clara sobre aspectos cruciales del desarrollo y el sistema. Además, la estandarización de los procesos de desarrollo contribuye a mejorar la eficiencia y coherencia en la creación de soluciones tecnológicas

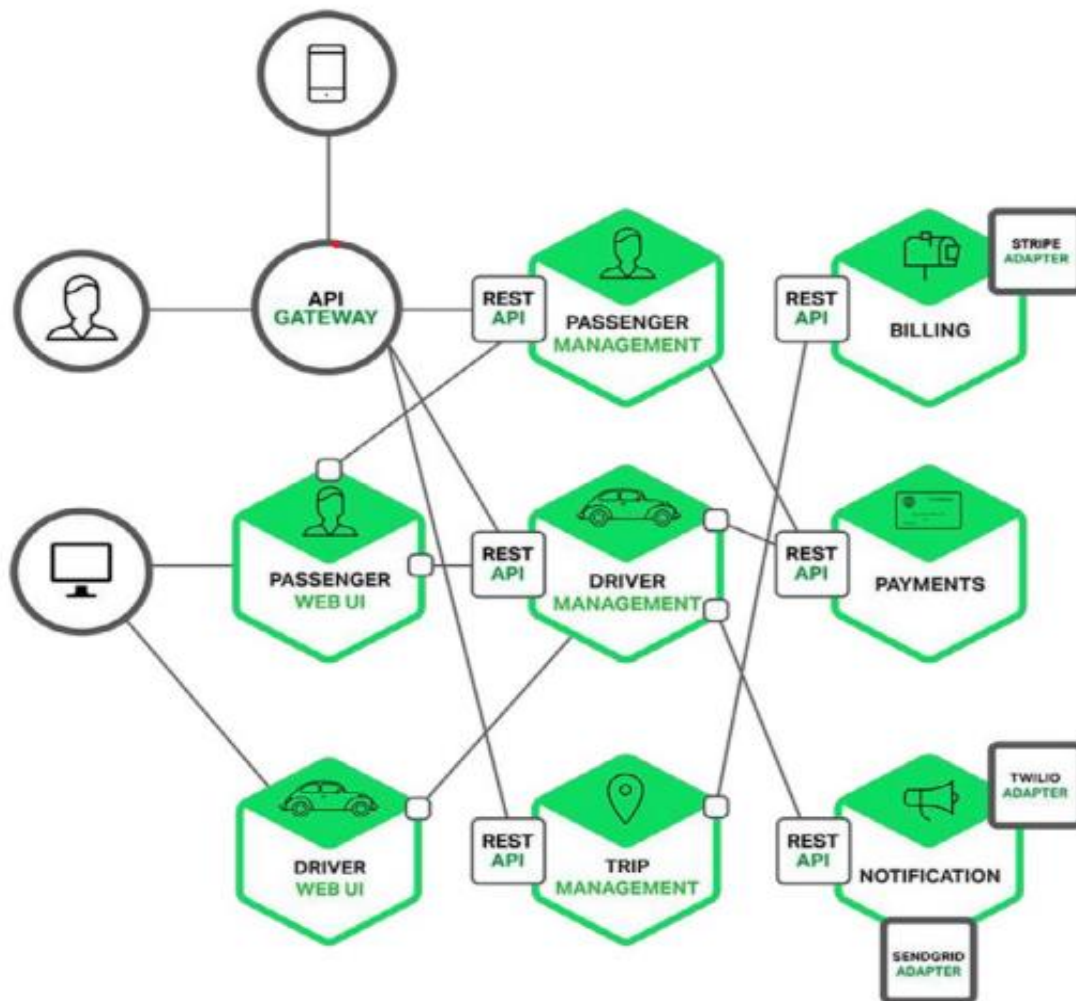
### ***Microservicios***

Según (Microsoft, 2024) define los microservicios como:

Una arquitectura de microservicios consta de una colección de servicios autónomos y pequeños. Cada uno de servicio es independiente y debe implementar una funcionalidad de negocio individual dentro de un contexto delimitado. Un contexto delimitado es una división natural de una empresa y proporciona un límite explícito dentro del cual existe un modelo de dominio.

En los últimos años, la arquitectura de microservicios ha ganado popularidad como una excelente opción para el desarrollo de software, ya que permite dividir los sistemas en secciones más pequeñas y enfocadas en el negocio. Esta división facilita un control más eficiente y un desarrollo más organizado.

Figura 21

*Arquitectura de Microservicios*

Fuente: (Barrios Contrera, 2018)

Además, si una sección del sistema falla, las demás podrían seguir funcionando con normalidad, lo que mantiene de alguna forma la continuidad del servicio que se da al cliente. También es más sencillo identificar y corregir errores específicos. Por lo cual, dividir la aplicación en microservicios permite vencer los desafíos de complejidad y rigidez que suelen afectar a las arquitecturas más tradicionales de Software.

Tabla 2

*Arquitecturas Monolíticas vs. Microservicios*

<b>Categoría</b>	<b>Arquitectura Monolítica</b>	<b>Arquitectura de microservicios</b>
<b>Código</b>	Una base de código única para toda la aplicación.	Múltiples bases de código. Cada microservicio tiene su propia base de código.
<b>Comprensibilidad</b>	A menudo confuso y difícil de mantener.	Mayor facilidad de lectura y mucho más fácil de mantener.
<b>Despliegue</b>	Implementaciones complejas con ventanas de mantenimiento y paradas programadas.	Despliegue sencillo ya que cada microservicio se puede implementar de forma individual, con un tiempo de inactividad mínimo, si no es cero.
<b>Lenguaje</b>	Tipicamente totalmente desarrollado en un lenguaje de programación.	Cada microservicio puede desarrollarse en un lenguaje de programación diferente.
<b>Escalamiento</b>	Requiere escalar la aplicación entera, aunque los cuellos de botella estén localizados.	Cada microservicio puede desarrollarse en un lenguaje de programación diferente.

Fuente: (López & Maya, 2017)

Figura 22

*Beneficios de los Microservicios*

Fuente: (López & Maya, 2017)

Como se muestra en la figura anterior, los microservicios ofrecen varios beneficios que facilitan el desarrollo de software. A continuación, se explicarán en detalle algunos de estos beneficios.

- La entrega continua es uno de los beneficios más importantes que ofrecen los microservicios, ya que estos se pueden desarrollar de forma independiente, lo que facilita la puesta en producción, así como su mantenimiento y solución de errores.
- Tienen la flexibilidad de que se desarrollen sistemas con diferentes tecnologías, ya que los microservicios son independientes, lo que permite la libertad de cambiar de lenguajes de programación para cada microservicio, esto permite innovar y mejorar el rendimiento de las aplicaciones.
- Permite dividir el trabajo de desarrollo en las principales partes del negocio, de esta forma los equipos pueden trabajar en varios requerimientos a la vez sin que se den problemas.
- En cuanto a la intensa modulación ya se ha mencionado la autonomía que tienen estos microservicios, por lo cual esto facilita el mantenimiento, la escalabilidad, la seguridad y el rendimiento del software.
- El tema de las aplicaciones heredadas hace referencias a que los microservicios pueden ayudar a facilitar migraciones de sistemas antiguos en la organización lo que permite ir haciendo por partes pequeñas.

### ***Interfaz de Programación de Aplicaciones (API)***

“Una API es un conjunto de definiciones y protocolos para crear e integrar *softclúster* aplicación. A veces se la denomina contrato entre un proveedor de información y un usuario

de información, que establece el contenido requerido del consumidor (la llamada) y el contenido requerido por el productor (la respuesta)” (Red Hat, 2020).

Entonces un API en español se define como una Interfaz de Programación de Aplicaciones, se puede entender como una herramienta que funciona en el *Back-End* de un sistema que interactúa con diferentes aplicaciones y con la interfaz que utiliza la persona usuaria, pero ella no sabe cómo funciona.

## **Análisis y Diseño del Sistema**

### ***Requerimientos***

"Un requerimiento se visualiza como una declaración abstracta de alto nivel de un servicio que debe proveer el sistema o como una restricción de éste”. (Sommerville, Universidad Veracruzana , 2015). Los requerimientos primeramente deben ser evaluados, en este caso se evalúa la viabilidad del proyecto, lo que luego se traduce en requisitos funcionales y no funcionales, siendo esto la base para el desarrollo del sistema por elaborar.

### **Figura 23**

#### ***Concepción de un Requerimiento***



Fuente: (Wong, 2018)

## Análisis de Factibilidad de un Requerimiento

“La factibilidad es la evaluación previa que se efectúa para determinar si un requerimiento será o no elaborado; muestra las restricciones a nivel tecnológico, funcional, de costos y de tiempo” (Wong, 2018, pág. 117). Antes de llevar a cabo el estudio de factibilidad es primordial saber las necesidades que tenga el usuario, por ello es muy importante programar reuniones con el usuario, con el propósito de plantear bien la solicitud. Cuando ya se tiene determinado este paso, se procede a especificar las características del hardware y software necesarios para el desarrollo del sistema. Además, se planifican los costos y plazos, y a la vez se determinará la metodología de ciclo de vida que se debe seguir para poder realizar su ejecución.

**Figura 24**

*Ciclo de Vida del Producto*



Fuente: (Wong, 2018)

### ***Funcionales***

Los requerimientos funcionales “describen el comportamiento y los datos que el sistema administrará. También, las capacidades que el sistema podrá realizar en términos de

comportamientos o acciones o respuestas de aplicación de tecnología de la información específicas de las operaciones”. (Wong, 2018). En otras palabras, son la guía detallada de lo que debe hacer un sistema de software para cumplir con las necesidades del usuario o del cliente.

**Ejemplos de Requisitos Funcionales.** El sistema debe ser capaz de registrar la información de todos los clientes de la tienda, gestionar el registro de las marcas de los productos y a la vez manejar la logística tanto del almacén como de la tienda. De igual forma, debe facilitar la generación de boletas de venta y facturas, y realizar automáticamente el cálculo del IGV. (Wong, 2018)

### ***No Funcionales***

Este tipo de requisitos están relacionados con la operación del software y hacen referencia al comportamiento de la solución, describen condiciones ambientales bajo las cuales la solución debe permanecer activa por determinados períodos de tiempo o cualidades que los sistemas deben tener a nivel operacional. (Wong, 2018)

Asimismo, se podría interpretar como las restricciones que se le imponen a un sistema, ya que debe definirse el comportamiento que va a tener, así como sus funciones en términos de rendimiento, capacidad, seguridad, velocidad entre otros.

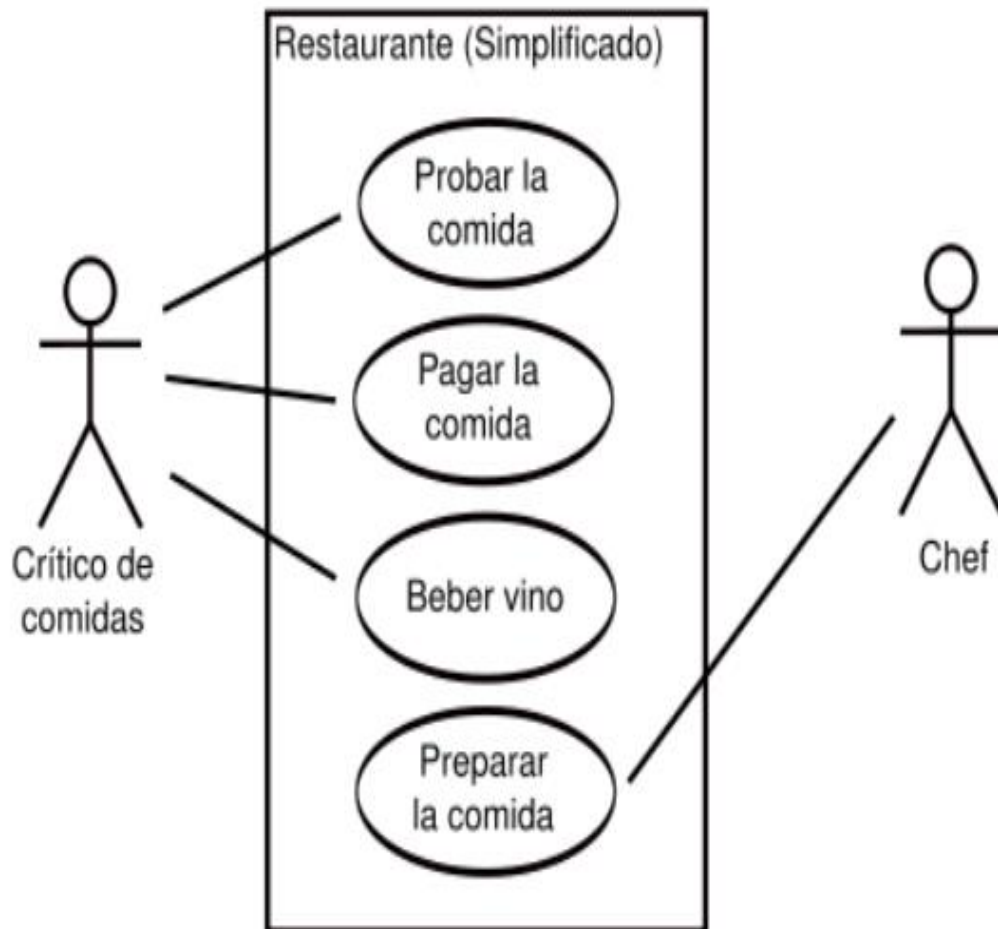
**Ejemplos Requisitos no Funcionales.** El sistema debe ser compatible con los navegadores de Explorer 7, Explorer 8 y Chrome, asegurando que funcione en todos los casos. Además, debe tener una disponibilidad ilimitada, esto quiere decir que debe estar habilitado en toda ocasión para que los usuarios puedan acceder en cualquier momento. Por otro lado, las interfaces de la aplicación móvil deben ser fáciles de usar, y la generación de reportes no deben superar los 30 segundos (Wong, 2018).

### *Casos de Uso*

“Un caso de uso representa una unidad funcional coherente de un sistema, subsistema o clase” (Vega M. , 2010, pág. 5). Con base en lo anterior se puede interpretar que es una secuencia de acciones que se realizan por medio del sistema, en donde describe cómo el sistema interactúa con usuarios externos o con otros sistemas, con la finalidad de lograr un objetivo en específico. Básicamente se trata de pasos que determinan las acciones que el actor hace para completar una tarea.

Hay seis principios básicos en el corazón de cualquier aplicación exitosa de los casos de uso:

- a) Mantenerlos simples al narrar historias
- b) Entender el panorama general
- c) Enfocarse en el valor
- d) Construir el sistema por partes
- e) Entregar el sistema en incrementos
- f) Adaptarse para cubrir las necesidades del equipo (Jacobson, 2013, pág. 5).

**Figura 25***Sistema de Casos de Uso*

Fuente: (Cevallos, 2015)

La función principal de los casos de uso es dejar claro qué debe hacer el sistema, y qué no debe hacer. Asimismo, facilitan la visión del comportamiento del sistema, el cual se centra en cumplir con las expectativas y requisitos de los usuarios.

Figura 26

## Mapa Conceptual de Casos de Uso



Fuente: (Jacobson, 2013).

## Diagramas de Casos de Uso

“El diagrama de casos de uso es un diagrama UML (lenguaje de modelado unificado) que, por medio de una representación gráfica del sistema, permite modelar los requerimientos funcionales de una aplicación, mostrándolos desde el punto de vista del usuario” (Martínez F. Y., 2022, pág. 5)

Estos diagramas tienen dos funciones que son importantes, las cuales se basan en:

- Capturar los requisitos funcionales del sistema
- Simplificar la construcción de los modelos de objetos

Por otro lado, es importante destacar los cuatro componentes fundamentales de los casos de uso. A continuación, se lograrán apreciar estos componentes.

**Sistemas.** Es lo que se está modelando para que sea usado por el actor con el propósito de cumplir su objetivo.

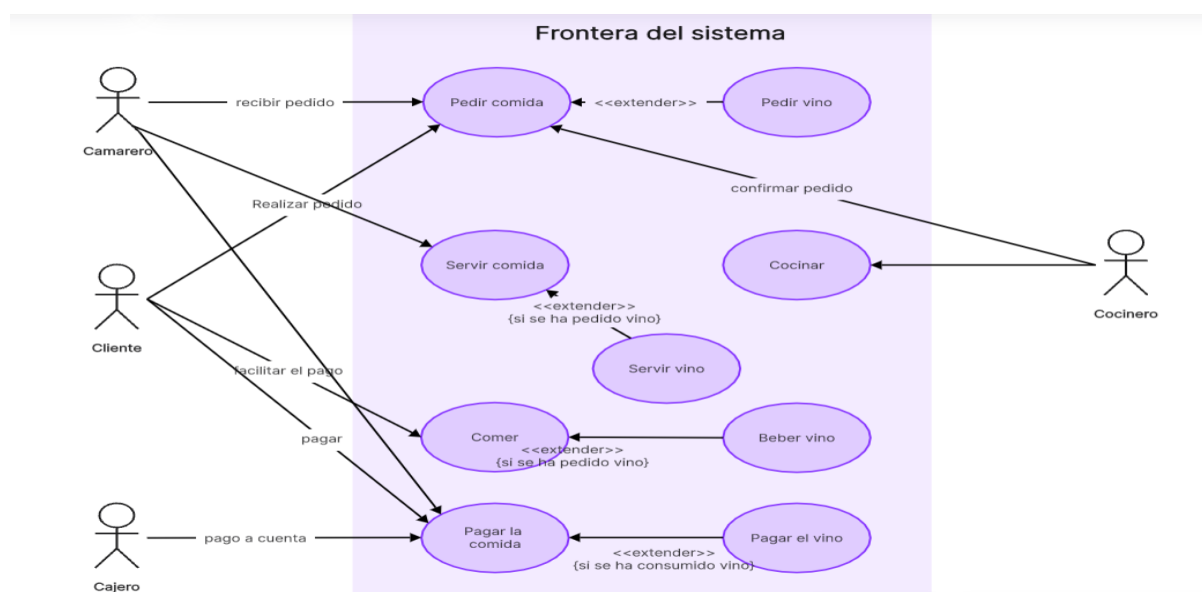
**Actores.** Los actores son quienes interactúan con el sistema, esto quiere decir que pueden ser personas, algún sistema o dispositivo que realice acciones en el sistema.

**Casos de usos.** Básicamente se centra en como los actores hacen uso del sistema para poder lograr un propósito en particular.

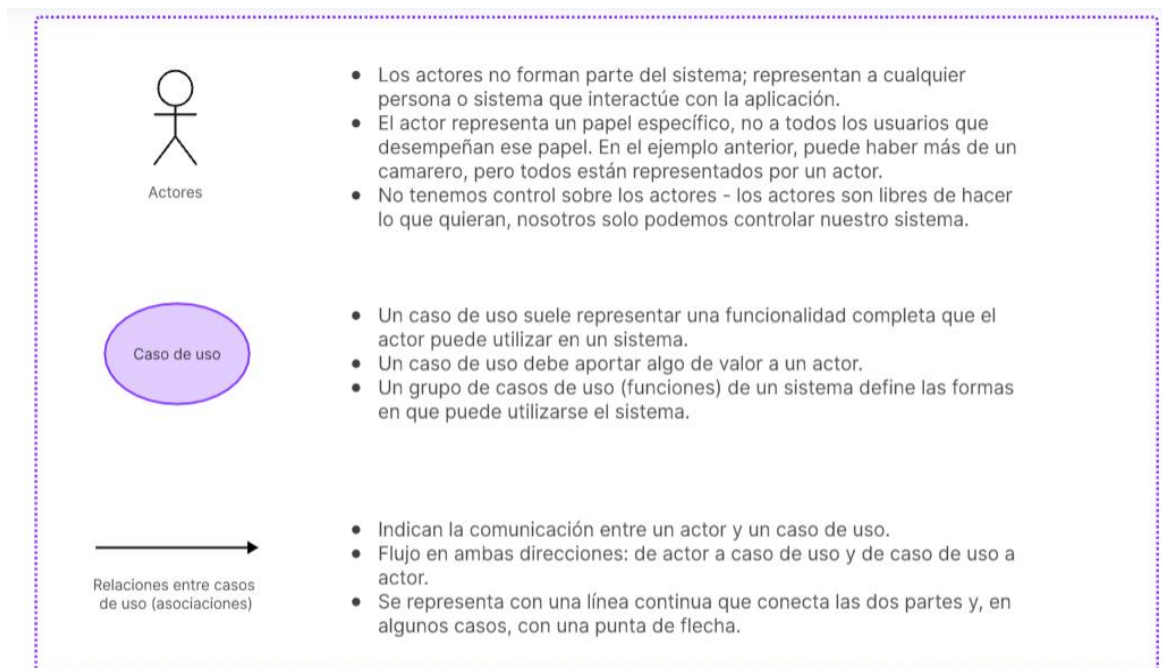
**Relaciones.** Muestran cómo interactúan los actores, los casos de uso en cómo están interactuando entre sí con el sistema.

**Figura 27**

*Diagrama de Caso de Uso*



Fuente: (Boardmix, 2024)

**Figura 28***Componentes y sus Símbolos de Diagrama de Casos de Uso*

Fuente: (Boardmix, 2024).

***Diagramas de Secuencia***

“El diagrama de secuencia es un tipo de diagrama de interacción cuyo objetivo es describir el comportamiento dinámico del sistema de información haciendo énfasis en la secuencia de los mensajes intercambiados por los objetos” (Lombeida, 2020). Esto hace énfasis en que es una representación visual en la cual se muestra cómo los objetos se comunican entre sí, con la finalidad de completar un objetivo en específico.

La función principal de un diagrama de secuencia es mostrar cómo los actores o componentes intercambien mensajes, asimismo está la documentación, que se encarga de explicar el proceso específico del sistema, por último, se puede destacar el diseño de interacciones que ayuda a planificar y comprender la comunicación entre los distintos

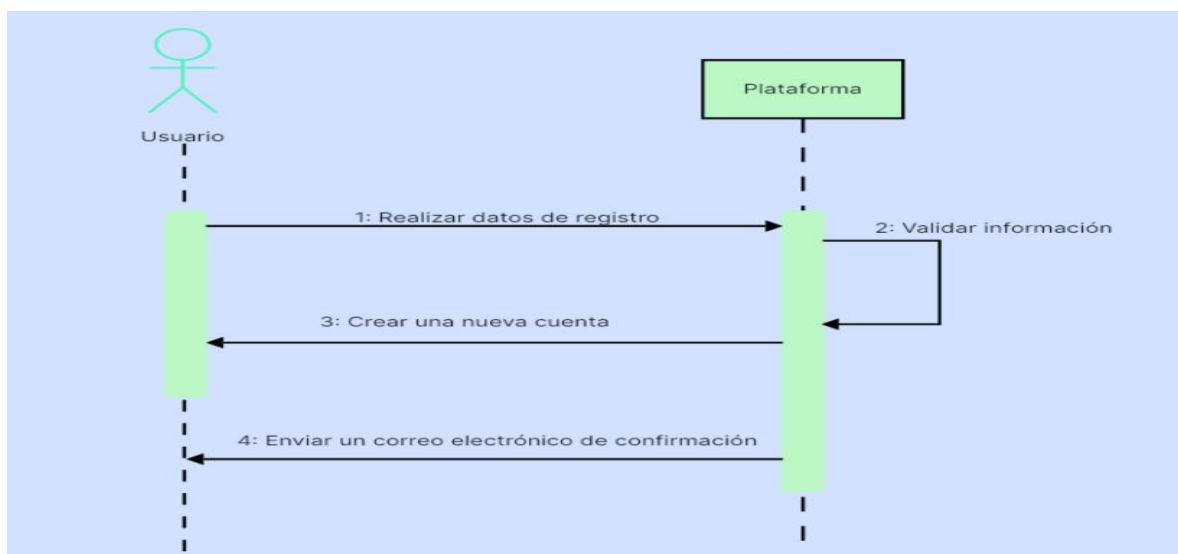
elementos del sistema. Esto con la finalidad de asegurar que todos tengan claro cómo se debe operar el sistema.

A continuación, se podrán observar los elementos de un diagrama de secuencia

- Rol de la clase: Describe cómo se va a comportar el objeto.
- Activación de foco: Cada cuadro de activación va a representar el tiempo que necesita un objeto para realizar una tarea.
- Mensaje: Se refiere a flechas horizontales que van a representar la comunicación entre los objetos.
- Objetos: Suele representarse con un rectángulo que suele tener un nombre o identificador en su interior, del cual se extiende una línea vertical hacia abajo.
- Destrucción de objetos: Los objetos se pueden destruir, básicamente se usa una flecha etiquetada que dice destruir la cual apunta a una X. (Lombeida, 2020).

**Figura 29**

*Diagrama de Secuencia*



Fuente: (Lopez N. , 2024)

## Metodologías Ágiles de Programación

Las metodologías ágiles son “una estrategia integral que impulsa a las organizaciones a gestionar los proyectos con rapidez y flexibilidad” (ADEN, 2024). Básicamente, las metodologías ágiles se enfocan en hacer que la entrega de un proyecto sea más eficiente, el cual se organiza en etapas que permiten entregar avances regulares y recibir retroalimentación constante. A diferencia de las metodologías tradicionales, que suelen ser más rígidas y menos adaptables a los cambios durante el desarrollo.

### *Scrum*

*Scrum* es una metodología ágil que proporciona un enfoque estructurado para planificar y controlar proyectos complejos, mediante la división de *sprints*. De esta forma permite gestionar la incertidumbre y adaptarse a los cambios de una forma más efectiva.

Por otro lado, la planificación se suele realizar de forma semanal, de esta forma al final de cada *sprint* se revisa el trabajo validado de la semana anterior, con el propósito de hacer las correcciones debidas, asimismo se planifican las actividades del siguiente sprint. (Lopez, 2018, pág. 56).

En *Scrum*, los roles juegan un papel primordial en la organización y ejecución del proyecto. Cada rol tiene un propósito claro y contribuye al éxito del equipo, garantizando que las responsabilidades estén bien definidas y que todos los miembros trabajen hacia un objetivo común. A continuación, se explican cómo se dividen las responsabilidades dentro de esta metodología.

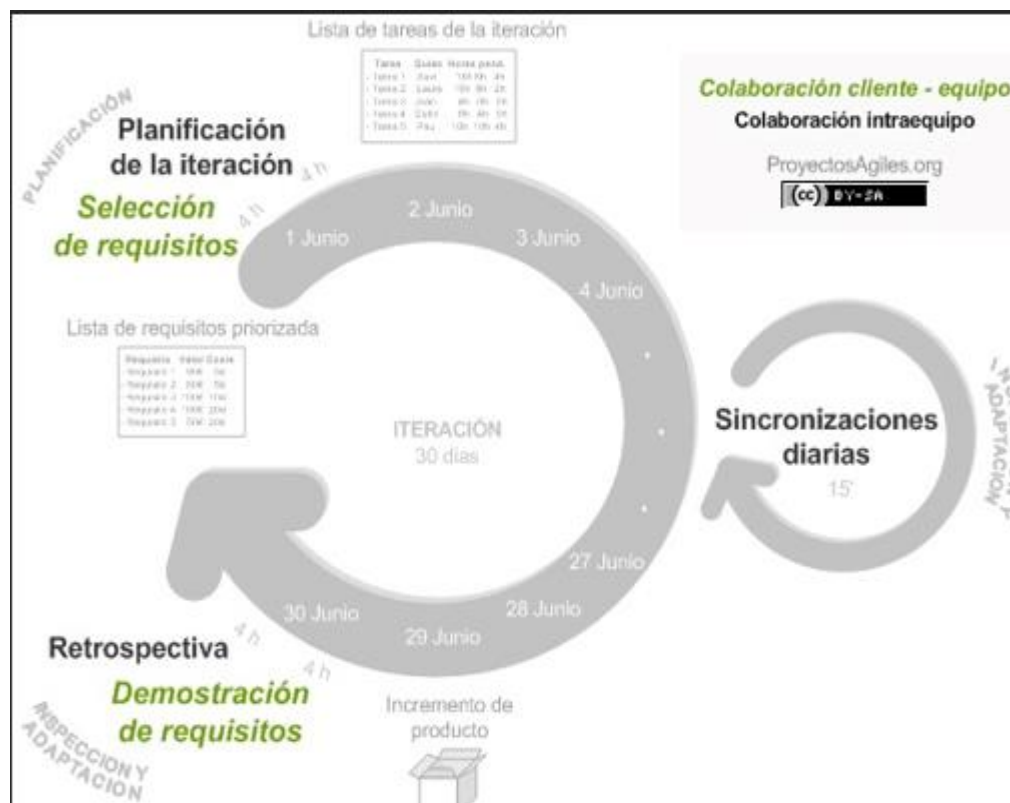
**Product Owner.** Se asegura de que el proyecto esté alineado con la estrategia del negocio. Asimismo, se encarga de redactar y organizar las historias del usuario en el producto *backlog*, de igual forma se mantiene al tanto de la velocidad del equipo para gestionar bien el tiempo.

**Scrum Máster o Facilitador.** Su función principal es estar presente en las reuniones y asegurarse de que el equipo cumpla con los tiempos y objetivos establecidos.

**Development Team Member.** Es el equipo encargado de elaborar el producto acorde con los requerimientos necesarios. Es importante resaltar que también se encargan de realizar las pruebas y las debidas correcciones según el *feedback* recibido.

**Figura 30**

*Proceso Scrum*



Fuente: (proyectosagiles, 2008)

### ***Programación Extrema (XP)***

Esta metodología se centra en la satisfacción del cliente, con la finalidad de que el equipo proporcione el software cuando lo requiera el cliente. Es importante destacar que este método les permite a los desarrolladores adaptarse a los cambios que sugiere el cliente durante las reuniones, incluso en las etapas finales del desarrollo. (Lopez, 2018, pág. 63)

En la XP, las funciones del equipo se distribuyen para maximizar la colaboración, adaptabilidad y entrega continua de valor. Cada miembro asume responsabilidades específicas, facilitando una interacción cercana y una comunicación constante. A continuación, se describen las responsabilidades clave en XP.

***Tracker.*** Se encarga del seguimiento del equipo y les aporta *feedback* acerca de mejoras que se pueden realizar.

***Customer.*** Cliente o usuario que redacta las historias de usuario, enfocándose en la que le ofrezca mayor valor al negocio. Además, se encarga de las pruebas funcionales para que el software cumpla con los requisitos

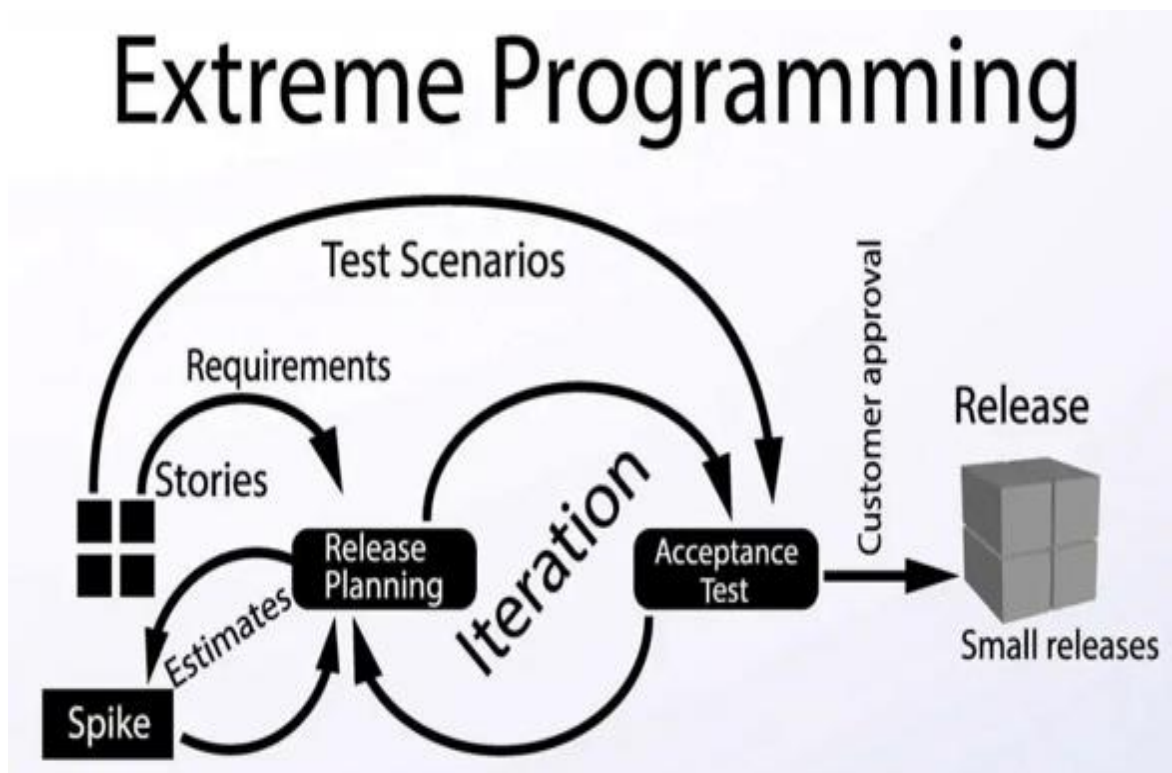
***Programmer.*** Se encarga de escribir el código para elaborar el software según las especificaciones y requisitos.

***Coach.*** Guía a los miembros del equipo, para que todo se lleve a cabo de una forma adecuada.

Entre otros roles se pueden encontrar al *Tester, Big Boss, Consultor, Manager, Doomsayer.*

Figura 31

Proceso de Programación Extrema



Fuente: (Bhaskar, 2024)

### ***Kanban***

*Kanban* es una metodología la cual es un marco de referencia, donde mejora la eficiencia de las empresas, debido a que las tareas son clasificadas en: tareas pendientes, haciendo en este momento o en progreso, realizadas. *Kanban* es una metodología *Lean* y *Agile*, creada para optimizar los procesos de trabajo.

“La metodología *Kanban* se implementa por medio de tableros *Kanban*. Se trata de un método visual de gestión de proyectos que permite a los equipos visualizar sus flujos de trabajo y la carga de trabajo” (Martins, 2024).

### **Prácticas básicas del método *Kanban*:**

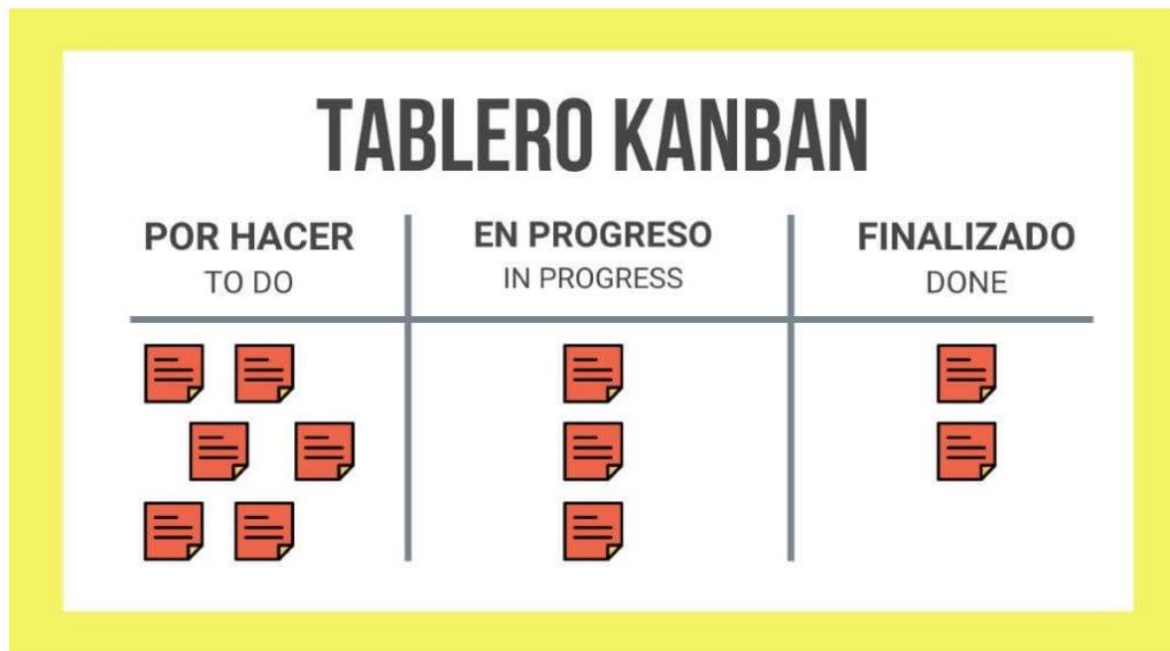
- Visualizar el flujo de trabajo

- Limitar el *WIP* (trabajo en curso)
- Gestionar el flujo
- Hacer explícitas las políticas de los procesos
- Implementar bucles de retroalimentación
- Mejorar en colaboración, evolucionar experimentalmente (NimbleWork, 2024).

### Implementación de Kanban

Primeramente, se debe visualizar el flujo de trabajo en un tablero con tres columnas que van a representar las etapas del proceso denominadas de la siguiente forma en pendiente, en proceso y finalizado. Cada tarea estará representada por tarjetas, las cuales se moverán por el tablero conforme avance su proceso.

Es importante destacar que se define la entrega de las tareas, asimismo se realizan reuniones regulares para recibir retroalimentación, con la finalidad de mejorar el proceso de forma colaborativa.

**Figura 32***Tablero kanban*

Fuente: (Reis Digital , 2022)

### **Metodologías de Desarrollo Tradicionales**

“Las metodologías tradicionales de desarrollo de software son orientadas por planeación. Inician el desarrollo de un proyecto con un riguroso proceso de elicitación de requerimientos, previo a etapas de análisis y diseño” (Navarro et al, 2013,pág.31).

Con base en lo anterior estas metodologías se centran en llevar una documentación exhaustiva de todo el proyecto. Una vez que se ha finalizado una fase, es más complejo realizar cambios, debido a que este proceso es de forma lineal, ya que una vez que se pasa a la etapa siguiente es muy complejo regresar a la anterior. A este método se le denomina modelo cascada.

### ***En Cascada***

La metodología cascada “se define como una secuencia de actividades, donde la estrategia principal es seguir el progreso del desarrollo de software hacia puntos de revisión

bien definidos (en inglés, *milestones o checkpoints*) mediante entregas programadas con fechas precisas (en inglés, *schedule*)” (Instituto Tecnológico PN, 2011, pág. 2).

Esta metodología sigue un proceso de forma lineal y secuencial, el cual proporciona una visión del desarrollo del software. En este modelo el proceso del software se representa por medio de varias fases que se complementan unas con otras, las cuales están de forma secuencial y separadas. Entre estas fases están los requisitos, diseño, implementación, pruebas y despliegue.

**Figura 33**

*Modelo Cascada*



Fuente: (Editorial IONOS, 2019)

### ***Incremental***

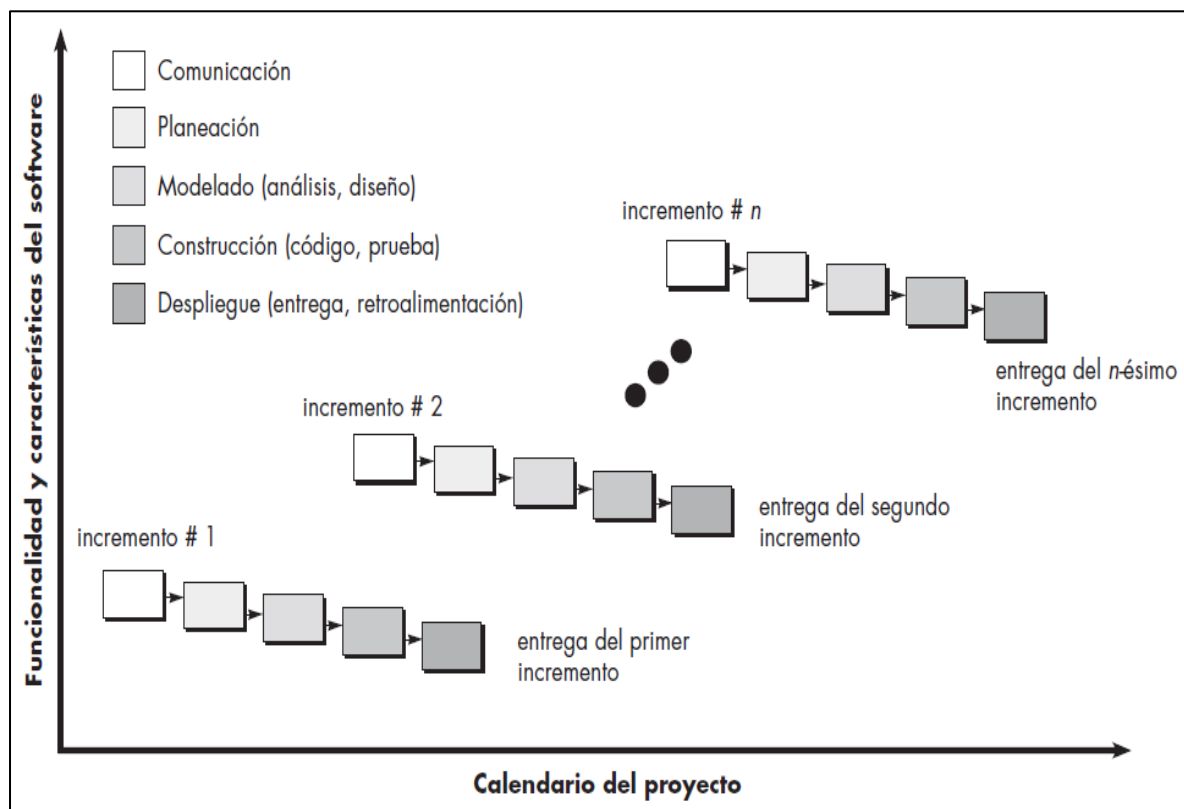
En este modelo el sistema se desarrolla y se entrega en pequeñas partes, ya que no se desarrolla todo el sistema de forma inmediata. Esto quiere decir que el proyecto se divide en

módulos que se van completando y entregando de forma gradual, en este proceso se puede recibir retroalimentación para realizar mejoras en las siguientes etapas.

“En esta metodología de desarrollo de software se va elaborando el producto final de manera progresiva. En cada etapa se añade una nueva funcionalidad, con la finalidad de ver resultados de una forma más rápida en comparación con el modelo en cascada” (Ginzo, 2021).

**Figura 34**

*Modelo Incremental*



Fuente: (Pressman, 2010)

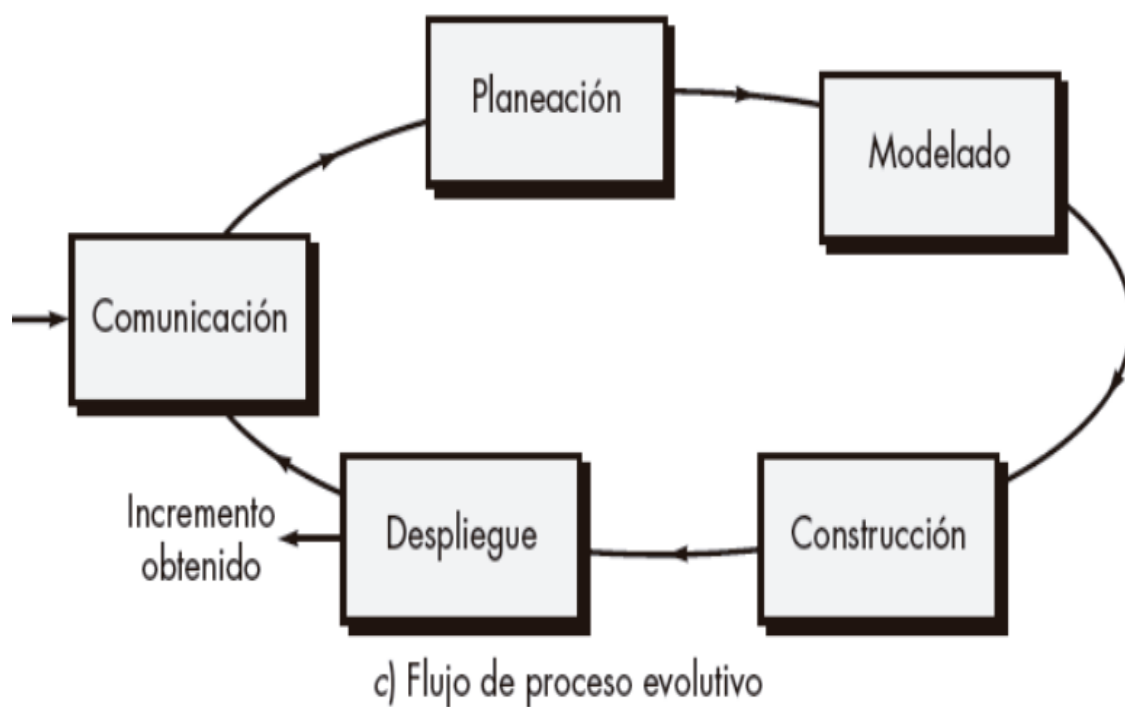
***Evolutivo***

“El modelo evolutivo es una extensión al modelo incremental en la que los incrementos se hacen de manera secuencial, en lugar de en paralelo” (Instituto Tecnológico

PN, 2011). A este modelo también se le conoce como RAD (Desarrollo Rápido de Aplicaciones). El desarrollo del sistema se basa en un proceso de mejoras continuas, por medio de pequeños cambios. Cabe destacar que se enfoca en la creación de prototipos, con la finalidad de realizar pruebas y avanzar de forma ágil y eficiente.

**Figura 35**

*Modelo Evolutivo*



Fuente: (Universidad Veracruzana , 2018)

### ***Prototipo***

“Un prototipo es una versión preliminar, intencionalmente incompleta o reducida de un sistema. El uso de prototipos es una estrategia que puede aplicarse en casi todas las actividades del proceso de software” (Instituto Tecnológico PN, 2011).

Los prototipos se suelen realizar con la finalidad de visualizar y probar las diferentes ideas del sistema del software, con el propósito de hacer pruebas para ver si funciona bien el

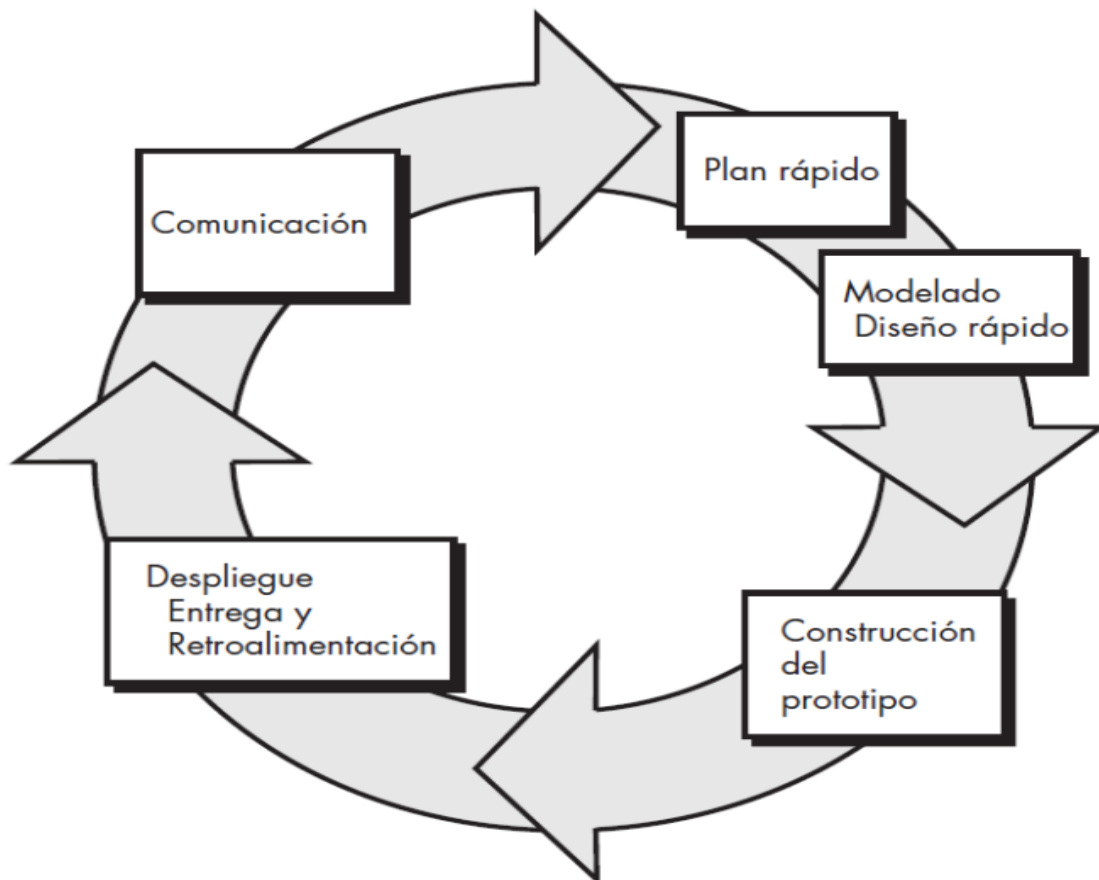
sistema o si va a requerir algún cambio o ajuste. Una vez que se realicen las pruebas se podrá desarrollar la versión final.

Hay diferentes tipos de prototipos entre ellos están:

- Prototipos de requisito
- Prototipos de análisis
- Prototipos de diseño
- Prototipos verticales
- Prototipos de factibilidad

**Figura 36**

Modelo de Prototipo



Fuente: (Pressman, 2010)

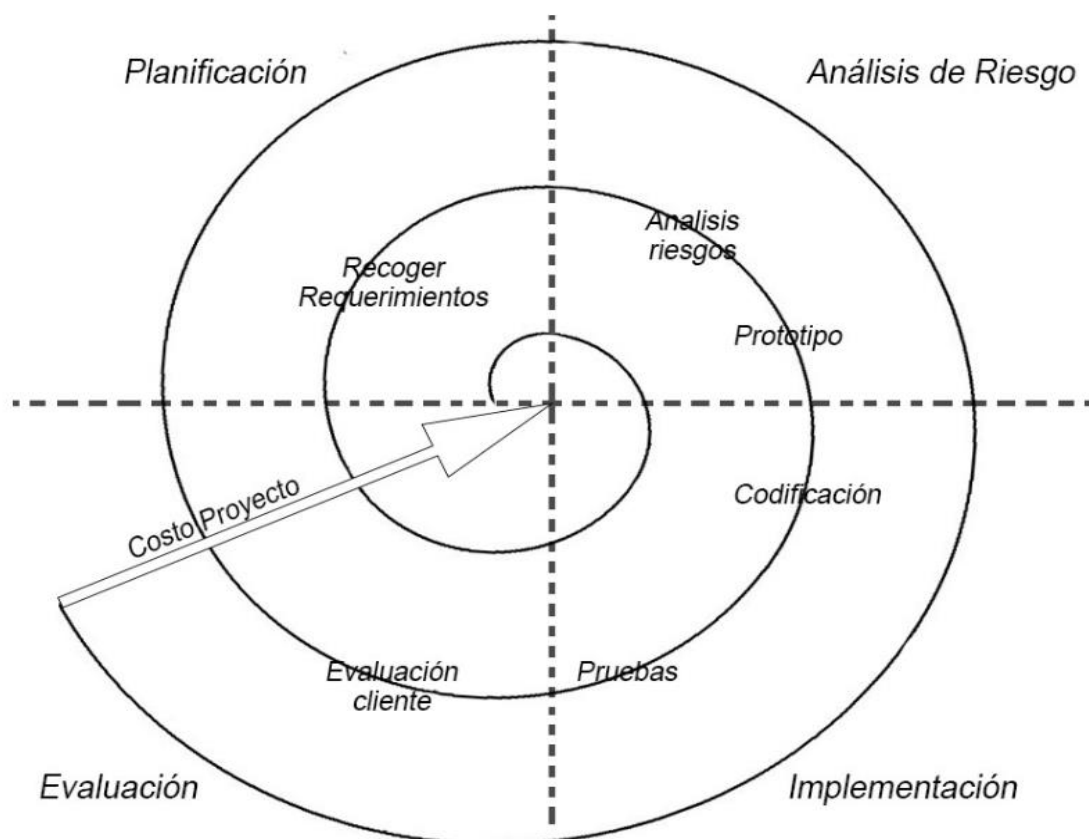
## *Espiral*

La metodología espiral se caracteriza por combinar dos modelos que son el incremental y la cascada, el cual tiene un enfoque en la gestión de riesgos. Se puede decir que este enfoque es una respuesta a los inconvenientes del desarrollo de cascada.

“El nombre de esta metodología se debe a su funcionamiento, ya que las etapas se procesan en forma de espiral. Cuanto más cerca del centro se está, más avanzado está el proyecto” (Ginzo, 2021).

### **Figura 37**

#### *Modelo Espiral*



Fuente: (Hernandez, 2022)

### **Capítulo III: Marco Metodológico**

## **Investigación**

“El proceso de investigación es un método sistemático utilizado para recopilar información y responder a preguntas concretas. El proceso garantiza que los resultados sean creíbles, de alta calidad y aplicables a un contexto más amplio” (ATLAS, 2024).

Con base en lo anterior se puede entender que por medio de este proceso se adquieren nuevos conocimientos y a la vez se confirman o desmienten hipótesis, asimismo se resuelven diferentes temas o problemáticas. Con el fin de recopilar y analizar los datos de esta forma se va a obtener más conocimiento acerca de un tema en específico. Este proceso suele tener varias etapas, como la formulación de una pregunta o alguna problemática, posteriormente se basa en la revisión de documentos, en donde se analizan y se obtienen datos e información, asimismo se presentan los resultados y las respectivas conclusiones a las que se llegaron por medio de la investigación.

### ***Tipo de Investigación***

El tipo de investigación es aplicada, debido a que se centra en la resolución de un problema práctico y específico dentro del entorno del poder Judicial de Costa Rica. Cabe destacar que la investigación básica se centra en ampliar el conocimiento teórico sin pensar en cómo usarlas de inmediato, en cambio la aplicada busca resolver un problema en específico de forma inmediata.

Para reforzar lo anterior, se puede entender que la investigación aplicada es:

También conocida como práctica o empírica, este tipo de investigación se caracteriza porque busca la aplicación o utilización de los conocimientos que se adquieren. La investigación aplicada se encuentra estrechamente vinculada con la investigación básica, pues depende de los resultados y avances de esta última ya que toda investigación aplicada requiere de un marco teórico. Sin embargo, en una

investigación empírica, lo que le interesa al investigador, primordialmente, son las consecuencias prácticas. Si una investigación involucra problemas tanto teóricos como prácticos, recibe el nombre de mixta (Universidad Veracruzana , 2014).

### **Alcance Investigativo**

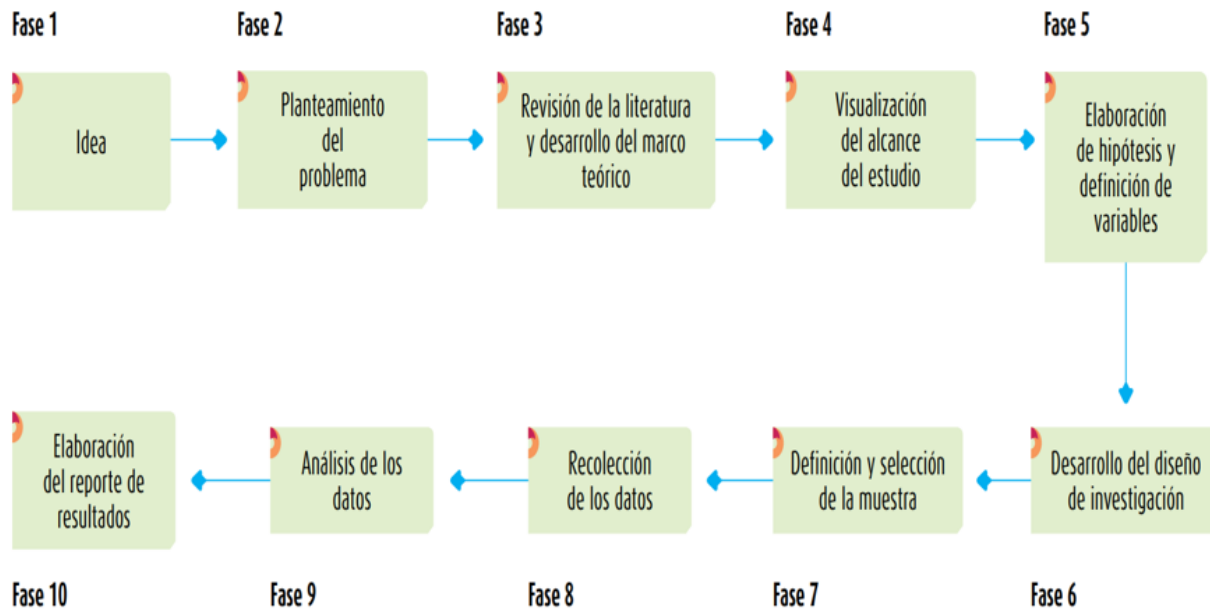
El alcance de esta investigación es descriptivo, ya que la investigación se centra en describir detalladamente el estado actual del sistema de almacenamientos y consulta de audiencias de audio y video en el Poder Judicial de Costa Rica. Básicamente se va a identificar y documentar cuáles son las deficiencias y problemas que está enfrentando el sistema, asimismo se identificarán los problemas que están afectando el rendimiento y la seguridad. Con base en esto se logrará tener una visión más clara del problema y así poder fundamentar el diseño de la solución propuesta.

### **Enfoque de la Investigación**

El enfoque es mixto debido a que se integran métodos cualitativos y cuantitativos, esto quiere decir que este enfoque proporciona una visión más detallada acerca de un problema o el fenómeno de estudio. A continuación, se explicarán un poco más a fondo en que consiste cada método.

La investigación cuantitativa es aquella en la que se recogen y analizan datos cuantitativos sobre variables y estudia las propiedades y fenómenos cuantitativos. Ente las técnicas de análisis se encuentran: análisis descriptivo, análisis exploratorio, inferencial univariable, inferencial multivariado, modelización y contrastación” (Universidad de Jaén , 2023).

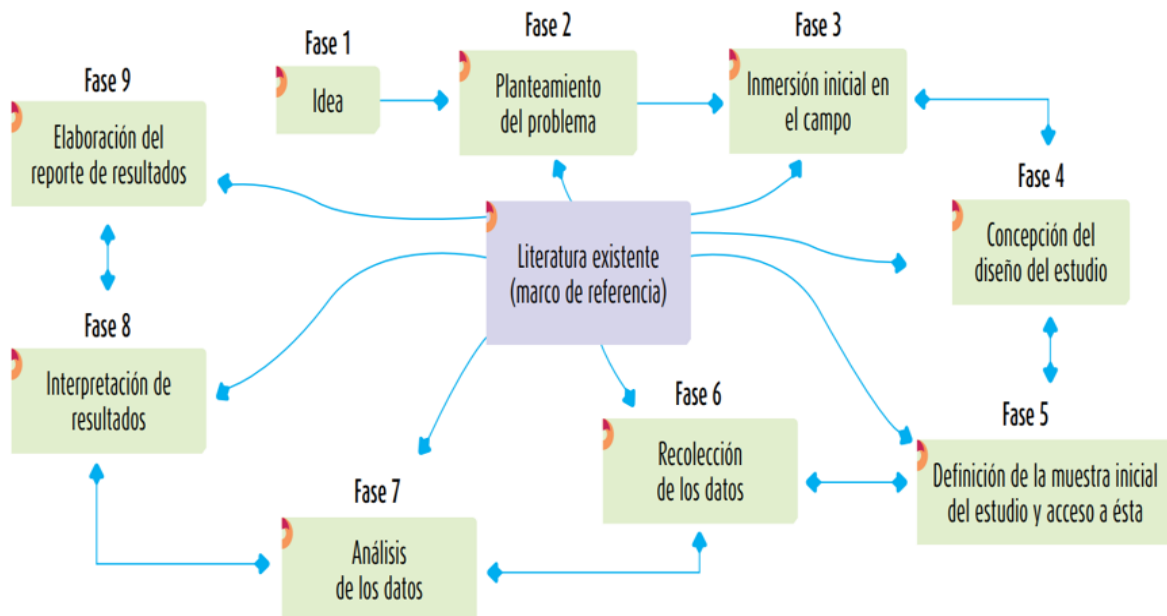
Esto quiere decir que se centra en la recolección y análisis de datos numéricos, que pueden ser medidos y cuantificados.

**Figura 38***Proceso Cuantitativo*

Fuente: (Sampieri, 2020, pág. 37)

La investigación de enfoque cualitativo es utilizada para explorar y comprender la profundidad y la complejidad de fenómenos sociales y humanos.

Este enfoque se centra en obtener una comprensión detallada de las experiencias y perspectivas de los participantes, a partir de la construcción de sentido y la interpretación, y suele involucrar un análisis de información para identificar patrones y temas emergentes” (Vega E. , 2024).

**Figura 39***Proceso Cualitativo*

Fuente: (Sampieri, 2020, pág. 39)

**Fuentes de Información**

Se considera una fuente de investigación a cualquier recurso que proporcione información acerca de un tema, lo cual es esencial para realizar una investigación y de esta forma obtener los datos necesarios, estas fuentes se dividen en fuentes primarias y secundarias.

***Fuentes de Información Primaria***

Para efectos de esta investigación la información se obtendrá por medio de las reuniones, en las cuales se entrevistarán administradores del sistema y personal técnico, con el fin de recopilar información acerca de cómo se gestiona el almacenamiento y la consulta

de audiencias de audio y video, analizando los problemas que enfrenta y obteniendo las opiniones acerca de las posibles mejoras, asimismo se hará uso de documentos y artículos que se encuentren en los sitios web.

En relación con las fuentes de información primaria, se pueden interpretar de la siguiente forma.

Esta fuente es aquella que otorga un testimonio o evidencia directa sobre el tema de investigación. Las fuentes primarias son escritas durante el tiempo que se está estudiando o por el investigador directamente envuelto en la investigación. La naturaleza y el valor de la fuente no puede ser determinado sin referencia al tema de estudio (U San Marcos , 2020, pág. 6).

Las fuentes que fueron usadas son las siguientes:

- Sitios Web: Sitios web especializados en el tema, así como documentación relacionada con la investigación.
- Artículos de investigación originales: Los artículos de investigación originales ofrecerán un soporte teórico y metodológico que será de gran soporte en la recolección de información.
- Entrevistas: Las entrevistas con personal técnico y usuarios del sistema actual son esenciales para identificar las principales deficiencias y necesidades en torno al almacenamiento y consulta de audiencias de audio y video.

### ***Fuentes de Información Secundaria***

Por otro lado, también se hará uso de las fuentes de información secundarias para la obtención de la información requerida para poder llevar a cabo la investigación. “Una fuente secundaria es aquella que interpreta y analiza fuentes primarias. Las fuentes secundarias están a un paso distanciadas de las fuentes primarias”

Las fuentes secundarias que fueron usadas son las siguientes:

- Tesis: Las tesis académicas que aborden temas de informática, ingeniería de software, eran recursos fundamentales para esta investigación
- Libros digitales: Estos libros proporcionaran información que será de utilidad para el proyecto.
- Documentos: Se toman en cuenta otras investigaciones acerca de temas relevantes que serán necesarios para el desarrollo del proyecto.

### **Observación**

“La observación es un elemento fundamental de todo proceso de investigación; en ella se apoya el investigador para obtener el mayor número de datos” (Sanjuán, 2011).

Básicamente se basa en observar y registrar los datos obtenidos en el entorno natural del objeto de estudio. Por medio de la observación, el investigador puede captar detalles que no siempre son accesibles mediante otros métodos de recolección de datos.

La observación se puede dividir en directa o indirecta, en donde la directa se basa en que el investigador se pone directamente en contacto con el fenómeno que está estudiando, interactuando directamente con el entorno o los sujetos de la investigación. Por otro lado, la observación indirecta se refiere a cuando el investigador obtiene información de un fenómeno o hecho de interés, por medio de las observaciones y registros realizados previamente por otra persona. (Sanjuán, 2011)

En el caso de esta investigación se va a monitorear y registrar el rendimiento del servidor Islon, antes y después de implementar los microservicios que utilizan el protocolo S3. Primeramente, se observarán diferentes factores como el tiempo de respuesta, la proporción de errores y la capacidad del sistema para atender solicitudes. Todo esto con la finalidad de identificar las debilidades del sistema actual, de esta forma se podrán

implementar las correcciones necesarias para mejorar el sistema. Una vez que se realicen estas mejoras se van a comparar con el anterior, de esta forma se podrán determinar los cambios en términos de velocidad, fiabilidad y eficiencia.

## Entrevistas

“La entrevista es una técnica de recogida de información que además de ser una de las estrategias utilizadas en procesos de investigación, tiene ya un valor en sí misma” (Bertomeu, 2016). En otras palabras, se puede interpretar que la entrevista consiste en preguntarle a una persona o a varias acerca de un tema en específico, con el fin de obtener datos de interés que complementan una investigación.

### Figura 40

#### *Tipologías de Entrevistas*

Criterios	Tipologías de entrevistas
<b>Según el momento</b>	Inicial De desarrollo Final
<b>Según el grado de estructuración</b>	Estructurada Semiestructurada No estructurada o en profundidad
<b>Según el número de participantes</b>	Individual Grupal

Fuente: (Bertomeu, 2016)

Las entrevistas se les hará a los líderes técnicos de los sistemas involucrados y personal técnico del Poder Judicial. Estas entrevistas serán semi-estructurado, esto quiere decir que incluirán tanto preguntas abiertas como cerradas.

En las cuales se debe tener en consideración los siguientes pasos para poder llevarla a cabo de una forma ordenada y clara:

**Inicio o Presentación:** El entrevistador debe saludar al entrevistado o a otros presentes. En esta primera etapa se le expone el tema de la entrevista y cómo se va a desarrollar.

**Desarrollo del Diálogo:** Esta parte es la más importante, debido a que comienzan las preguntas por parte del entrevistador al entrevistado que las responde. En este caso el entrevistador debe fluir con la serie de preguntas y orientarlas acorde con los propósitos establecidos.

**Cierre y Despedida:** Al finalizar el entrevistador hace un cierre de lo discutido y agradece la colaboración. Antes de dar por terminada la sesión, le da al entrevistado la oportunidad de despedirse, para así dar por concluida la entrevista.

El propósito de las entrevistas se basará en identificar el problema principal, de esta forma se podrá recolectar información sobre las principales deficiencias que enfrentan los usuarios y los técnicos con el sistema. Por otro lado, se podrán entender las necesidades y expectativas que tenga el personal en relación con la nueva solución que esperan, así como el *feedback* que proporcionen para mejorar la estrategia, de esta forma se podrá asegurar que la implementación de los microservicios con el protocolo S3 responda adecuadamente a las demandas operativas del Poder Judicial.

Las entrevistas se llevaron a cabo en reuniones que incluían la participación de la Jefatura, coordinadores, líderes técnicos, desarrolladores y analistas, todos juntos. Esto facilitó la comprensión del problema raíz relacionado con el almacenamiento y consulta de audiencias en el *servidor Isilon*. Además, permitió un levantamiento de requerimientos más preciso para la solución. Estas sesiones también ayudaron a identificar posibles problemas

que no se detectan mediante el análisis de observación, lo que subraya la importancia de contar con diversas perspectivas en este tipo de reuniones.

Esta colaboración es esencial para garantizar que el desarrollo del software se ajuste a los requisitos de todos los involucrados. Al reunir a todos los actores, se asegura que el proyecto esté alineado con las necesidades reales del negocio y se minimizan los errores que podrían surgir en las primeras fases de análisis. De esta forma se promueve un desarrollo coherente y bien fundamentado.

**Tabla 3***Variables*

<b>Objetivo General</b>	Implementar microservicios en el ambiente de desarrollo del Poder Judicial de Costa Rica en el segundo semestre del año 2024, utilizando el protocolo S3 en un servidor Isilon para el almacenamiento y consulta de audiencias de audio y video, mejorando el rendimiento y seguridad.				
<b>Objetivo Especifico</b>	<b>Variable</b>	<b>Indicador</b>	<b>Medida</b>	<b>Tipo de variable</b>	<b>Instrumento</b>
Evaluar el estado actual analizando la forma de almacenamiento y consulta de audiencias de audio y video, para la identificación de deficiencias y áreas de mejora.	Proceso de almacenamiento y consulta de audiencias de audio y video	Identificación de deficiencias y áreas de mejora.	Análisis de los métodos actuales de almacenamiento y consulta.	Cualitativa	Observación y entrevistas
Diseñar una solución basada en microservicios utilizando el protocolo S3 para las deficiencias identificadas en el sistema actual.	Diseño de la solución basada en microservicios con protocolo S3	Propuesta de arquitectura basada en microservicios y S3.	Especificación técnica y diagrama de la arquitectura propuesta.	Cualitativa	Observación

<p>Desarrollar microservicios de almacenamiento y consulta usando el protocolo S3 en el servidor Isilon permitiendo a las personas usuarias la reproducción de las audiencias orales.</p>	<p>Desarrollo de microservicios de almacenamiento y consulta.</p>	<p>Funcionalidad de reproducción de audiencias orales.</p>	<p>Número de microservicios desarrollados y su capacidad de reproducir audiencias orales correctamente.</p>	<p>Cualitativa</p>	<p>Desarrollo y observación.</p>
<p>Realizar pruebas de microservicio de almacenamiento y consulta por medio de la revisión del código, pruebas de integración y la utilización de pruebas unitarias para que la solución dada no presente problemas en el ambiente de producción.</p>	<p>Calidad del microservicio en ambiente de producción.</p>	<p>Porcentaje de errores detectados y corregidos</p>	<p>Número de pruebas unitarias, de integración y revisiones de código realizadas, y el número de problemas detectados y corregidos.</p>	<p>Cualitativa y cuantitativa</p>	<p>Observación</p>

## **Capítulo IV: Análisis de Resultados**

## Análisis de Factibilidad

Los estudios de factibilidad tienen por objetivo decidir sobre la posibilidad y conveniencia de realizar una idea de proyecto y determinar cuál es la mejor forma de hacerlo. Proveen a los responsables del proyecto de una base objetiva para decidir sobre su continuidad y escoger las mejores alternativas para su realización.

(Universidad del Valle , 2001)

El análisis revela que el sistema actual enfrenta problemas significativos debido a la falta de un registro y organización adecuada de los archivos de audio y video, lo que resulta en una respuesta lenta y errática del servidor. Esto afecta negativamente la productividad y eficiencia de los sistemas jurisdiccionales. Se propone el desarrollo de un sistema que permita organizar y gestionar de manera eficiente las audiencias, lo que evitará errores, y a la vez mejorará considerablemente la velocidad de respuesta y satisfacción de los usuarios.

### Figura 41

*Estudio o Análisis de Factibilidad*



Fuente:(El Profe Alegría, 2024)

## **Factibilidad Económica**

El servicio de Estudio de Factibilidad Económica y Financiera es un análisis exhaustivo que evalúa la viabilidad económica y financiera de tu proyecto o inversión. Proporcionamos información clave para tomar decisiones informadas y estratégicas, asegurando que asignes tus recursos de manera eficiente y maximices la probabilidad de éxito. (FUMERO , 2021)

En relación con la factibilidad económica, el desarrollo de los microservicios que utilizan el protocolo S3 no representa un costo adicional significativo para la institución. Al ser parte de un proyecto de graduación, los costos de desarrollo están eliminados. Además, el uso del protocolo S3 con el *servidor Isilon*, ya existente en la infraestructura, es gratuito, sin requerir licencias o inversión en hardware adicional.

La única inversión relevante es el tiempo del personal de TI necesario para realizar pruebas y llevar a cabo la implementación en producción, actividades que se realizarán como parte de sus responsabilidades regulares. Por estas razones el proyecto es altamente viable, ya que no implica una gran carga económica para la institución.

## **Diagnóstico Administrativo y Operativo**

Actualmente, el proceso que se utilizará para gestionar y consultar los archivos de las audiencias de audio y video de Poder Judicial está presentando problemas, lo que resulta en retrasos significativos al consultar esta información desde sistemas como SIGAO, Gestión en Línea, Nexus, entre otros. Estos retrasos afectan la eficiencia general de los sistemas, provocando tiempos de espera prolongados, en ocasiones, fallos en el acceso a los archivos necesarios.

Otro problema es la generación de reportes por las personas usuarias que requieren reinicios del servidor para poder consultar las audiencias. Además, se han generado reportes

para solucionar problemas con audiencias que no están guardadas en el servidor, lo que afecta la integridad y disponibilidad de los datos.

La falta de un proceso confiable para el manejo de los archivos de las audiencias causa retrasos y errores frecuentes, obligando al personal de TI a dedicar tiempo a solucionar problemas. Esto afecta negativamente la eficiencia operativa del Poder Judicial, ya que tanto las personas usarias internas como externas experimentan demoras en el acceso a la información.

### **Diagnóstico Técnico**

Al realizar el diagnóstico se puede observar que el principal problema técnico que se tiene es la sobrecarga de archivos en un solo directorio dentro del *servidor Isilon*, todas las audiencias grabadas de audio y video se guardan en una misma carpeta lo que hace que los sistemas no puedan consultar estas audiencias por el servidor donde se encuentran dura demasiado tiempo buscado los archivos los cual no están ordenados.

Para solucionar este problema, se propone realizar un inventario de las audiencias que se encuentran actualmente en el servidor y gestión de las audiencias por medio de una base de datos, mediante la implementación de un sistema basado en microservicios y el uso del protocolo S3 de Amazon. Esto permitirá manejar los archivos de audio y video de una manera eficiente, asegurando la escalabilidad y mejora en la respuesta del sistema. Además, se recomienda la creación de un *API controlador* que administre la conversión y almacenamiento de audiencias, optimizando el uso del *servidor Isilon*.

Para abordar este problema se va a implementar un microservicio para la consulta de audiencias y otro para que el servicio de conversión pueda guardar las audiencias que están previamente grabadas por el sistema *SIGAO*.

También se va a utilizar el protocolo S3, el cual según la documentación de Amazon se puede utilizar en un microservicio por medio de *un SDK* que permite una conexión directa con el *servidor Isilon* lo cual permite guardar, consultar o eliminar los archivos, además de que este servicio es más seguro que utilizar *FTP*.

### ***Lenguaje de Programación***

En cuanto al lenguaje de programación que se va a utilizar en este caso para el desarrollo de los microservicios es *Net Core 7* al ser parte de Microsoft permite una amplia documentación y va en relación con los nuevos proyectos que se están implementando en el Poder Judicial. En el servicio de convención de audiencia y en los sistemas se tiene que hacer cambios a esto sistemas si están en una tecnología de Visual Basic.

En cuanto a la base de datos que se va a utilizar para inventariar los archivos que se encuentran en *ípsilon* y guardar los archivos que va a enviar convertidor de audiencias se va a utilizar *SQL*.

### ***Servidor de Aplicaciones***

En relación con el servidor de desarrollo donde se va a implementar los microservicios y donde se encuentra el servicio de conversión de audiencias tiene las siguientes características.

- **Edición:** *Windows Server 2016 Standard.*
- **Versión;** 1607
- **Procesador:** *Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz 2.19 GHz (2 processors).*
- **RAM:** 24.0 GB.
- **Tipo de Sistema:** 64-bit operating system, x64-based processor
- **Internet Information Services:** *(Versión 10.0.14393.0)*

- *Microsoft .NET Runtime: versión 7.0.4 (x64).*

### ***Herramientas Aplicadas sobre el Recurso Humano***

Para el levantamiento de los requerimientos y la aprobación de la propuesta de la implementación de microservicios en el ambiente de desarrollo para la gestión y consulta de audiencias judiciales con almacenamiento en el *servidor Isilon* utilizando el protocolo S3, fue necesario realizar reuniones con los líderes técnicos de los sistemas involucrados, así como con los coordinadores de sistemas jurisdiccionales y la jefatura. Además, se recurrió a la observación directa y a la revisión de documentación de los sistemas involucrados que se encuentran en producción.

Se realizó una investigación detallada sobre la implementación del protocolo S3, con el objetivo de garantizar que la solución propuesta sea la más adecuada y coherente con las expectativas de los usuarios finales. Todo este proceso permitió asegurar que no solo responda a las necesidades técnicas, sino que también sea lo esperado por las personas usuarias.

### **Tabla 4**

#### *Determinación de Brechas*

<b>Actual</b>	<b>Brecha</b>	<b>Deseado</b>
Los sistemas actuales presentan retrasos en las consultas de audiencias debido a la sobrecarga de archivos en directorios.	Necesidad de un microservicio que gestione mejora la organización de audiencias para evitar dichos retrasos.	Optimizar la consulta de las audiencias desde los diferentes sistemas.

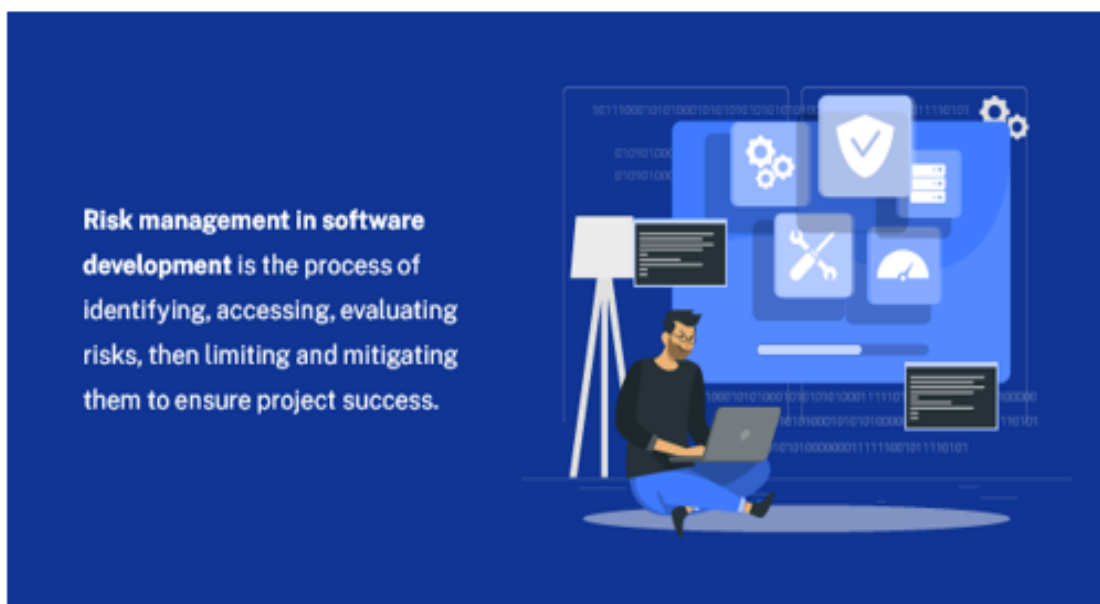
<p>Las audiencias almacenadas no se encuentran organizadas por carpetas, lo que dificulta la accesibilidad.</p>	<p>Falta de un método de organización adecuado de los archivos en el <i>servidor ISILON</i>.</p>	<p>Organización de audiencias mediante directorios o <i>buckets</i> <i>utilizado</i> el microservicio con la implementación del protocolo S3.</p>
---	--	---

### Plan de riesgos

La gestión de riesgos es el proceso de identificar, evaluar, monitorizar y responder a aquellos factores de riesgo que pueda encontrarse un proyecto empresarial, de forma que se defiendan los objetivos del negocio. Supone controlar y prever posibles amenazas de una forma proactiva, anticipándose, y no reactiva. (Cesce , 2022)

### Figura 42

#### *Plan de Riesgos de Software*



Fuente: (Gottdiener, 2022)

El plan de riesgos es una herramienta fundamental para los desarrolladores de software, ya que les ayuda a identificar y clasificar los riesgos potenciales, lo que les permite anticiparse a los problemas que puedan surgir durante el proceso de creación de aplicaciones. Gracias a este plan, los riesgos que se lleguen a presentar a lo largo del proyecto pueden ser controlados, ignorados o mitigados según sea necesario. Además, proporciona a las personas involucradas una guía sobre cómo actuar ante estas eventualidades.

Es importante destacar que los planes de riesgos requieren un mantenimiento constante, ya que pueden surgir nuevos riesgos no contemplados inicialmente. Sin embargo, en lugar de considerarlo un motivo de preocupación, debe verse como una oportunidad de aprendizaje para proyectos futuros.

**Tabla 5**

*Plan de Gestión de Riesgos para el Proyecto*

<b>ID</b>	<b>Riesgo</b>	<b>Respuesta al Riesgo</b>	<b>Acciones</b>
R-01	Incapacidad médica o enfermedad de la persona desarrolladora de este proyecto de tesis.	Aceptar	En caso de incapacidad temporal, se esperará a la recuperación de la persona, ya que es la única con el conocimiento necesario para llevar a cabo el proyecto. Si la incapacidad es permanente, el proyecto de tesis será reprobado y el Poder Judicial deberá reasignar el trabajo a sus analistas y desarrolladores internos.

R-02	Problemas con el levantamiento de requerimientos.	Evitar	<p>Para evitar este riesgo, se deberá asegurar la correcta documentación de los requerimientos, organizar reuniones frecuentes con las partes interesadas, para obtener todas las aprobaciones necesarias. Un manejo efectivo de los requisitos y una comunicación clara y constante con la organización evitarán futuros problemas o malentendidos durante el desarrollo.</p>
R-03	Los microservicios pueden presentar problemas de comportamiento o errores inesperados.	Aceptar	<p>Si los microservicios presentan problemas, se deberá aceptar el riesgo y proceder con la solución de los errores en el entorno de desarrollo. Es fundamental realizar pruebas, como pruebas unitarias, de integración, para identificar fallos antes de la implementación en producción y minimizar interrupciones.</p>
R-04	Mala comunicación o desinterés de los interesados en el proyecto.	Evitar	<p>Aunque es poco probable, debido al alto interés de la organización, se deberá mantener una comunicación constante y efectiva con los interesados mediante todos los canales posibles (reuniones, correos, llamadas). Si el desinterés surge, se tomarán medidas para identificar la causa y buscar la solución más adecuada.</p>

R-05	Cambios de requerimientos por parte de los interesados al finalizar el proyecto.	Transferir	Si los interesados solicitan cambios una vez finalizado el proyecto, se transferirá el riesgo a la empresa, ya que se supone que los acuerdos sobre los requerimientos y el alcance fueron previamente establecidos en las reuniones iniciales. No obstante, se podrían aceptar cambios menores que no afecten el cronograma, siempre y cuando se mantenga dentro de lo pactado.
R-06	Problemas para la integración de los microservicios con otros sistemas en producción o consultas de audiencias de personas usuarias.	Evitar y Transferir	Para evitar este riesgo, se realizarán pruebas de integración en el entorno de desarrollo, asegurando que los sistemas interactúan correctamente con los microservicios. Además, se dejará toda la documentación detallada sobre los cambios y procedimientos. No obstante, el riesgo en producción será transferido a la organización, ya que el alcance del proyecto no incluye soporte post-implementación.
R-07	Dificultades para implementar mejoras en distintos circuitos y sistemas, ya que es necesario coordinar una	Transferir	Este riesgo se transferirá a la empresa, ya que el alcance del proyecto se limita al desarrollo. No obstante, se sugerirá a la organización coordinar un plan detallado para la implementación en producción a

	misma fecha a nivel nacional.		nivel nacional, garantizando que las mejoras puedan ser aplicadas sin interrumpir el servicio a los usuarios finales en cada circuito.
R-08	Fuga o pérdida de información sensible, lo cual podría comprometer la seguridad y confidencialidad de los datos.	Evitar	Para mitigar este riesgo, se está implementando la encriptación de todos los datos sensibles en los archivos de configuración del microservicio, siguiendo estrictamente las políticas y recomendaciones de seguridad de la empresa. Además, se asegurará el uso del protocolo S3 para la transmisión segura de datos. También se gestionará con la universidad la confidencialidad de la tesis para prevenir la fuga de información sensible relacionada con esta implementación.
R-09	Atrasos en el cronograma de desarrollo por errores, investigación, permisos u otros imprevistos.	Evitar	Para anticiparse a posibles retrasos, se planificarán tiempos adicionales en el cronograma para absorber posibles demoras, ya sea por errores, investigaciones o temas administrativos. La implementación de una metodología ágil, como <i>Scrum</i> y <i>Kanban</i> , permitirá una

			<p>mejor gestión del proyecto y una mayor flexibilidad para enfrentar estos imprevistos sin afectar el plazo de entrega general.</p>
--	--	--	--

En la tabla anterior se pueden observar los posibles riesgos que pueden surgir durante el desarrollo e implementación de los microservicios de este proyecto. Para cada riesgo, se han definido acciones específicas que permitirán abordar y gestionar los problemas cuando se presenten.

Una adecuada planificación y control de estos riesgos contribuirán a aumentar la probabilidad de éxito del proyecto, ayudando a minimizar los inconvenientes que puedan afectar su desarrollo y brindando claridad sobre cómo actuar frente a estos desafíos.

**Tabla 6**

*Matrix de Riesgos de Probabilidad e Impacto*

		Impacto ¿Qué tan severos serían los resultados si ocurriera el riesgo?				
		Insignificante 1	Menor 2	Significativo 3	Mayor 4	Severo 5
Probabilidad ¿Cuál es la probabilidad de que ocurra el riesgo?	5 Casi seguro	Medio 5	Alto 10	Muy alto 15	Extremo 20	Extremo 25
	4 Probable	Medio 4	Medio 8	Alto 12	Muy alto 16	Extremo 20
	3 Moderado	Bajo 3	Medio 6	Medio 9	Alto 12	Muy alto 15
	2 Poco probable	Muy bajo 2	Bajo 4	Medio 6	Medio 8	Alto 10
	1 Raro	Muy bajo 1	Muy bajo 2	Bajo 3	Medio 4	Medio 5

**SafetyCulture**

Fuente: (SafetyCulture, 2024)

Como se observa en la matriz de riesgos de probabilidad e impacto de la imagen anterior, esta herramienta permite a las personas gestoras de riesgos analizar y priorizar aquellos que requieren acciones inmediatas o identificar cuáles son los más peligrosos para la organización, lo ayuda con la implementación y desarrollo del proyecto.

La información presentada en la matriz facilita una interpretación más sencilla, simplificando la clasificación de los riesgos y reduciendo la necesidad de análisis extensos, que a menudo no resultan tan claros o efectivos.

**Tabla 7**

*Análisis de Riesgos por Probabilidad, Impacto y Nivel de Riesgo*

Riesgo	Probabilidad	Impacto	Nivel del Riesgo
R-01	Moderado	Severo	15
R-02	Poco probable	Severo	10
R-03	Moderado	Menor	6
R-04	Raro	Significativo	3
R-05	Raro	Severo	5
R-06	Poco probable	Severo	10
R-07	Probable	Severo	20
R-08	Moderado	Severo	15
R-09	Probable	Significativo	12



## **Capítulo V: Conclusiones y Recomendaciones**

## Conclusiones

1. Para llevar a cabo la evaluación del estado actual en el Poder Judicial en relación con gestión de audiencias, se implementó una metodología que incluyó la observación y la realización de entrevistas con las personas involucradas en el proceso, como la jefa encargada del subproceso, coordinadores, líderes, desarrolladores y analistas. Gracias a esto se identificó el problema principal, la sobrecarga en las carpetas del servidor Isilon, lo cual generaba una serie de inconvenientes en el acceso y manejo de las audiencias.

Tras la identificación de este problema, se propusieron varias propuestas para que las personas anteriormente mencionadas aprobaran la más idónea, tomando en cuenta las necesidades del negocio como la infraestructura actual. Además, se definieron los requerimientos de alto nivel, este fue el primer paso para solucionar el problema actual.

2. El diseño de la solución se desarrolló siguiendo las buenas prácticas de arquitectura de software y los estándares establecidos por el Departamento de Sistemas Jurisdiccionales del Poder Judicial. Este enfoque permitió desarrollar una solución bien documentada, facilitando la comprensión de los procesos involucrados, sistemas relacionados y los cambios realizados en comparación con la implementación anterior.

Se documentaron detalladamente todos los componentes basados en microservicios. Los diagramas elaborados ilustran claramente el flujo de datos y la interacción entre los microservicios, utilizado para mejorar el almacenamiento y recuperación de los archivos de audio y video. El equipo encargado tendrá acceso a una documentación

clara que les permitirá entender mejor el sistema y garantizar su correcto funcionamiento a medida que las necesidades del Poder Judicial evolucionen.

3. Para el desarrollo de los microservicios, se implementaron funcionalidades que permiten que un servicio de Windows almacene archivos de audio o video en *streaming* en el *servidor Isilon* mediante el protocolo S3. Para ello, se utilizó el SDK de AWS y *.NET Core 7*, lo que garantiza la creación de un sistema con condiciones óptimas en términos de seguridad, escalabilidad, flexibilidad y tecnología de vanguardia. Además, se mejoró el proceso de consulta de las audiencias desde los sistemas que realizaban dichas consultas, optimizando su eficiencia y rendimiento.
4. En el ambiente de desarrollo, se llevaron a cabo pruebas de carga, integración, y pruebas en distintos dispositivos, además de evaluaciones de las interfaces con los diferentes sistemas involucrados. De igual manera, se realizaron pruebas unitarias en los microservicios, garantizando que el sistema cumpla con los requerimientos solicitados por los encargados de los sistemas del Poder Judicial y asegurando que no surjan problemas al implementar los cambios en el ambiente de producción.

### **Recomendaciones**

1. Para futuras mejoras en los sistemas, se sugiere optimizar los procesos relacionados con el levantamiento de requerimientos mediante la implementación de prácticas que garanticen reuniones más eficientes y productivas. Esto incluye el uso de agendas estructuradas, la participación activa de todos los interesados en los sistemas y el establecimiento de roles claros para cada participante.

Además, se recomienda registrar y documentar los acuerdos alcanzados durante estas sesiones, tanto en formato escrito como mediante grabaciones, para garantizar transparencia y un fácil acceso a la información relevante en el futuro.

También sería beneficioso diseñar un formulario o guía que facilite a los desarrolladores y analistas estructurar de manera más precisa los requerimientos levantados, asegurando claridad y especificidad desde el inicio. Esto permitirá agilizar el desarrollo de soluciones alineadas con las necesidades del negocio y minimizará posibles confusiones o malentendidos.

2. Se recomienda implementar un proceso de actualización de la documentación técnica entregada, con el objetivo de garantizar que cualquier cambio realizado en los microservicios, componentes o sistemas relacionados con la mejora implementada quede debidamente registrado. Esto facilitará que los equipos encargados de realizar futuras mejoras o implementar soluciones similares dispongan de la información completa, precisa y actualizada, lo cual es esencial para la toma de decisiones fundamentadas.

Además, se sugiere centralizar esta documentación en un repositorio accesible, como un sistema de gestión de documentación (*wiki* institucional o plataforma de control de versiones), que permita al personal autorizado de la institución consultarla fácilmente. De esta forma, se asegura la disponibilidad de la información en un formato estructurado y organizado, promoviendo la colaboración y evitando la duplicidad de esfuerzos.

3. Para asegurar el correcto funcionamiento del sistema, se recomienda llevar a cabo una revisión continua de los logs de errores en conjunto con los logs de los sistemas. Esta

práctica permitirá identificar y resolver problemas de manera oportuna, manteniendo la integridad y eficiencia del sistema.

4. Es fundamental replicar en el ambiente de pruebas (*QA*) las mismas pruebas realizadas durante el desarrollo, incluyendo pruebas de carga, integración, y pruebas unitarias, para garantizar la estabilidad y funcionalidad del sistema. Además, se debe ampliar el alcance de estas pruebas para incluir escenarios adicionales que puedan presentarse en un entorno real, contemplando casos extremos y excepcionales que podrían afectar el desempeño del sistema.

Se recomienda realizar pruebas exhaustivas con todos los sistemas involucrados, asegurando la correcta interacción y compatibilidad entre ellos. En particular, es crucial llevar a cabo pruebas específicas con el *worker services* utilizando datos reales del ambiente de producción, no limitándose únicamente a los datos de desarrollo. Esto permitirá detectar posibles problemas relacionados con la manipulación de datos reales, asegurando que el sistema funcione de manera eficiente en condiciones reales. Asimismo, es indispensable realizar pruebas utilizando las direcciones *HTTPS* de los servidores para garantizar que las configuraciones de seguridad y comunicación estén adecuadas y que no se presenten inconvenientes en la conexión o en el acceso a los servicios.

Para las pruebas de inventario, es esencial incluir tanto la carpeta de desarrollo como la carpeta consolidada de producción, verificando no solo la integridad de los datos, sino también los tiempos de procesamiento. Esto asegurará que el sistema maneje adecuadamente los volúmenes de datos y cumpla con los estándares de rendimiento establecidos.

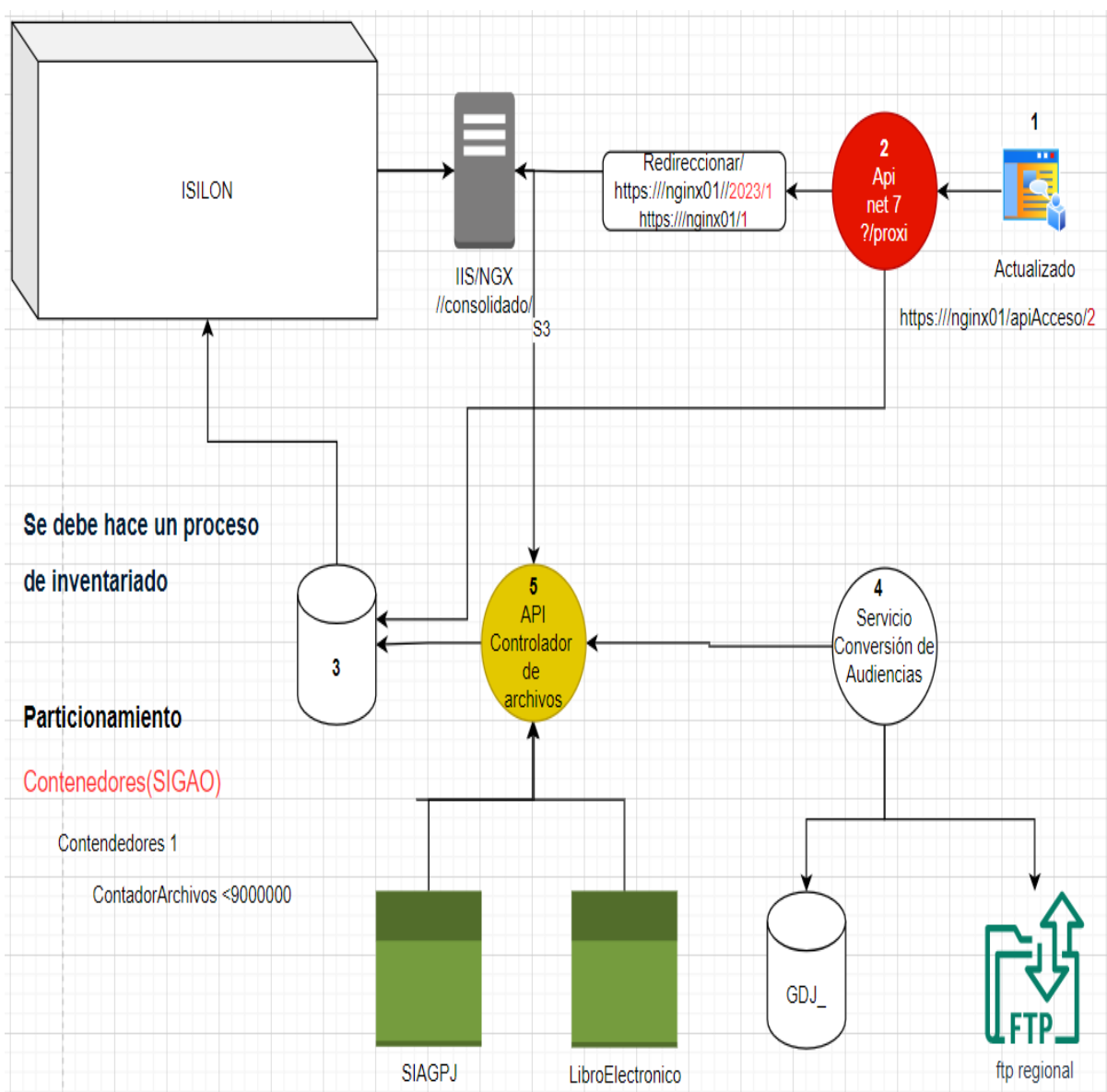
## **Capítulo VI. Propuesta de desarrollo**

## Modelado de la Solución

### Mejora Propuesta en el Almacenamiento y Consulta de Audiencias

**Figura 43**

*Diagrama Propuesta en el Almacenamiento y Consulta de Audiencias*

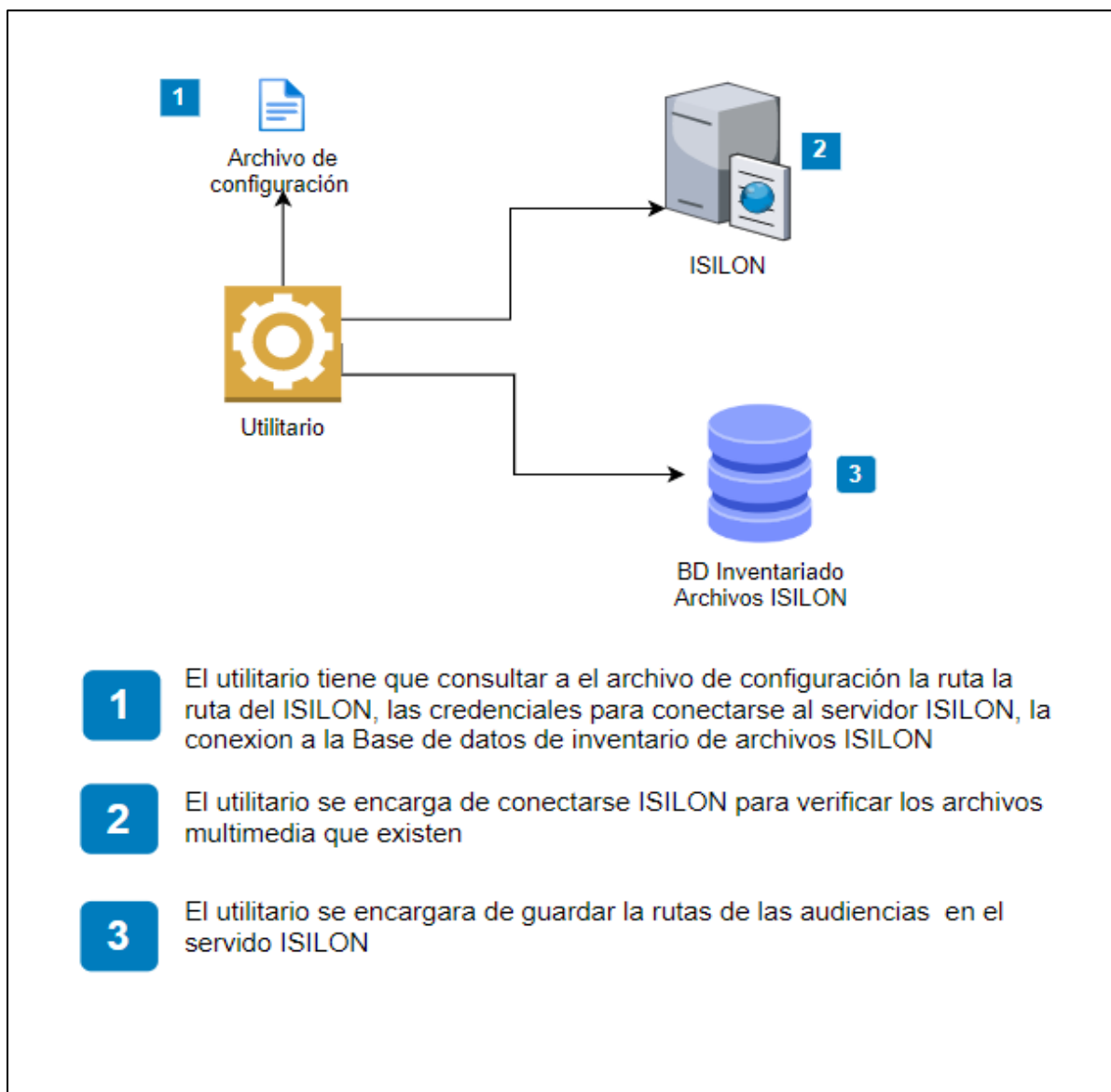


En el diagrama anterior se puede observar la solución que se va a aplicar para esta mejora, en la cual se pretende realizar cambios en la forma en la que se está administrando, almacenando y consultando las audiencias en el servidor ISILON, mediante una nueva

implementación que sea más eficiente, evitando de esta forma los problemas que se están presentando cuando las personas usuarias realizan consultas de audiencias.

Los siguientes puntos hacen referencia a la numeración que se encuentra en el diagrama, con los nuevos cambios que se tienen que realizar en esta nueva mejora:

1. Indica que todos los sistemas tienen que cambiar la ruta que se tenía antes del servidor NGINX por la nueva ruta API proxy, en este caso se tiene que cambiar esta ruta en el config o configuración de los sistemas. Es importante mencionar que este cambio se lleve a cabo en producción, llevará una previa coordinación para que se pueda implementar, ya que todos los sistemas tienen que hacer el cambio en la misma fecha, cabe la posibilidad de que no se puedan reproducir audiencias en caso de que un regional o un sistema no actualice.
2. Cuando los sistemas realicen el cambio mencionado en el punto número 1, el nuevo *API proxy* se encargará de redireccionar en relación con el parámetro que recibirá con el nombre del archivo, luego realizará una consulta en la nueva base de datos para buscar por medio de ese nombre la ruta en la que se encuentra guardada la audiencia. Después este *API* se encargará de indicarle al servidor *NGINX* cuál va a ser la ruta que va a utilizar para buscar la audiencia, ya sea en la raíz del consolidado o en alguna subcarpeta del *servidor ISILON*, para que finalmente el sistema pueda reproducir de forma normal la audiencia.
3. Esta nueva base de datos se encargará de guardar los identificadores y las rutas correspondientes de los archivos o audiencias. Es importante mencionar, que se tiene que hacer un inventariado de todos los archivos *del ISILON* que se encuentran en la raíz por medio de un utilitario, para registrar las rutas y los ID de estas audiencias.

**Figura 44***Diagrama Utilitario para Inventariar las Audiencias*

4. En el servicio de conversión de audiencia se van a realizar varios cambios en esta nueva implementación, en este caso, ya no se encargará de guardar las audiencias en la carpeta del regional en el *servidor ISILON*, esta función la va a asumir el nuevo API controlador de archivos que explica en el punto 5.

Se revisa la programación del Servicio de Conversión de Audiencias y se puede observar que al generar los archivos de *streaming de video*, se utilizan dos protocolos

DASH y HLS para generar los archivos de video que se envían al servidor ISILON, estos protocolos son los encargados de transmitir videos a través de internet.

DASH (*Dynamic Adaptive Streaming over HTTP*): este protocolo es el que se está utilizando actualmente, para la transmisión de videos de las audiencias en el *servidor ISILON*, a la hora que los diferentes sistemas realizan las consultas de los archivos *streaming de video*, lo que se hacen es que utilizan o invocan el archivo .MDP (*Media Presentation Description*), este archivo es de tipo XML, el cual tiene la información del contenido multimedia de las audiencias de video, que se encuentran *en streaming en el servidor ISILON*.

HLS (*HTTP Live Streaming*): Este protocolo también permite la entrega de contenido audio y video, el cual fue desarrollado por Apple en 2009, este lo que hace es que se crean varios archivos .M3U8 a la hora que se realiza el *proceso de streaming* de las audiencias de video.

El otro cambio a realizar es con los protocolos anteriormente mencionados, el Servicio de Conversión de Audiencias solo transformará las audiencias generando el archivo .MDP, el archivo de audio, el archivo de video y la imagen PNG, en otras palabras, utilizando el protocolo DASH. Además, usando también el protocolo HLS, que son 4 archivos del formato M3U8. Para esto se va a crear una llave de configuración en el config de este servicio, que va a permitir habilitar o deshabilitar el protocolo de HLS en el servidor de conversión.

5. El nuevo API controlador de archivos se va a encargar de administrar y almacenar los archivos en el *servidor ISILON*, quiere decir que este API se encargará de recibir los archivos que están convertidos en *streaming* por el servicio de conversión de audiencias, para luego guardarlos en una carpeta que se va a crear en la raíz del

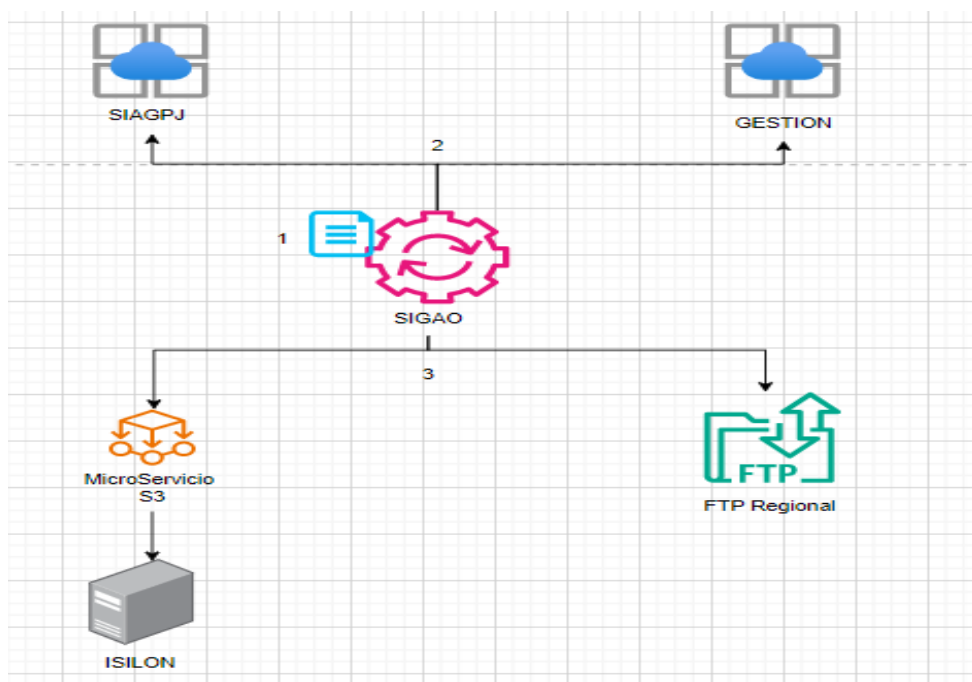
consolidado, en caso de que esta carpeta exceda los 900 000 archivos, este API controlador de archivos se encargará de crear un nuevo directorio o carpeta en la raíz del consolidado para seguir guardando las audiencias.

El nuevo API controlador de archivos se va a encargarse de administrar y almacenar los archivos en la nueva base de datos, que tendrá el inventario de los *archivos streaming*, donde se guardará el ID y la ruta donde se guardó el archivo, en caso de que el copiado de los archivos se haya realizado de forma correcta en el *servidor ISILON*. Es importante mencionar que el proceso de BTE de copiar los archivos *en streaming* de la carpeta del regional, a la raíz del consolidado en el *servidor ISILON* ya no se ocupará, esto porque el API controlador de archivos se encargará de guardar los archivos directamente en una carpeta del consolidado.

### ***Exportación de Audiencias de SIGAO***

**Figura 45**

*Exportación de Audiencias de SIGAO*



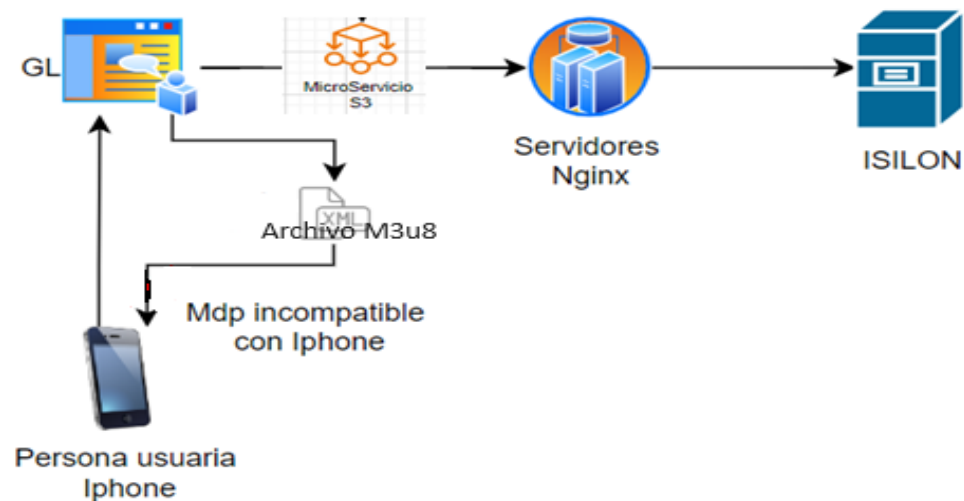
1. El sistema de *SIGAO* inicia y carga las configuraciones del archivo de conexiones.xml que se encuentran en el servidor, luego estas configuraciones se cargan en el config de la aplicación, donde se encuentran las rutas de los servidores de *streaming*, *SIAGPJ*, conexiones a BD entre otros.
2. Después de esto, el sistema se encarga de consultar las audiencias con relación a si el expediente es de *SIAGPJ* o de *Gestión*. A como se van cargando las audiencias, de una vez le coloca la información de la ruta FTP, si las audiencias se encuentran en el FTP de regionales, por otra parte, si la audiencia se encuentra en el servidor ISILON se asigna la ruta del microservicio del S3 a la audiencia.
3. Finalmente, cuando la persona usuaria exporta una audiencia la aplicación de *SIGAO* se encarga de validar el estado de la audiencia, si la audiencia tiene estado dos de publicación quiere decir que se encuentra en el *servidor ISILON*, en caso contrario la audiencia no se ha convertido a streaming por lo cual todavía se encuentra en el FTP del regional. En relación con esto, el sistema de *SIGAO* define si conectarse al FTP Regional o llamar al microservicio para descargar los archivos de la audiencia para la persona usuaria final.

Como se puede observar ya no se va a utilizar el *FTP del ISILON* que se tenía antes para descargar las audiencias de *SIGAO*, en este caso se tiene que llamar al microservicio de S3, el cual se va a encargar por medio del protocolo S3 a descargar los archivos de las audiencias, para que finalmente *SIGAO* le realice el tratamiento de exportación para luego colocar los archivos en la ruta designada por la persona usuaria.

## Consulta de Audiencias desde Iphone GL

**Figura 46**

Consulta de Audiencias desde Iphone GL



Los siguientes puntos hacen referencia a los nuevos cambios que se tienen que realizar en esta nueva mejora:

1. El sistema de Gestión en Línea se tiene que conectar ya no directamente a los servidores Nginx, sino que tiene que conectarse al microservicio de audiencias para la reproducción de las audiencias, este microservicio se encargará de redireccionar al Nginx.
2. En este caso el microservicio de S3 se encargará de devolver un archivo M3U8 el cual permite la reproducción de las audiencias de los *iPhone*, se tiene que validar que la persona usuaria esté realizando las consultas de las audiencias desde un *celular iPhone*.

## Requerimientos

**Tabla 8**

*Identificación de Requerimientos*

Modelado del negocio	Condiciones	Modelado de la solución
Trabajadores <ul style="list-style-type: none"> <li>• Funcionarios Judiciales.</li> </ul>	¿Se automatizan algunas de sus tareas? Si	Actores <ul style="list-style-type: none"> <li>• Sistema</li> </ul>
Actores <ul style="list-style-type: none"> <li>• Sistema.</li> </ul>	¿Si se automatizan tareas que antes realizaban trabajadores, utilizará el sistema directamente? Sí.	Actores <ul style="list-style-type: none"> <li>• Sistema</li> </ul>
Procesos <ul style="list-style-type: none"> <li>• Consulta de Audiencias</li> <li>• Conversión de audiencias</li> <li>• Agregar audiencias</li> <li>• Inventariado ISILON</li> </ul>	¿Cuáles actividades se automatizan? No,	Requerimientos <ul style="list-style-type: none"> <li>• Consulta de Audiencias</li> <li>• Conversión de audiencias</li> <li>• Agregar audiencias</li> <li>• Inventariado de audiencias en el ISILON</li> </ul>

1. Permitir a la persona usuaria poder realizar consultas de las audiencias en los sistemas de una forma más rápida, sin que se presenten problemas con el servidor donde se encuentra los archivos de streaming que se consultan de este sistema.
2. Mejorar la forma en la que se está administrando y almacenando las audiencias en *servidor ISILON*, para que no se exceda el límite de archivos por directorio recomendados según la documentación de DELL.
3. Crear un mecanismo que permita realizar un inventariado de los archivos streaming que se encuentran en el *servidor ISILON*, para guardar las rutas de las audiencias en la nueva base de datos de inventario de *archivos ISILON*.
4. Permitir la exportación de audiencias en el sistema de *SIGAO*.

## Reglas

Las reglas que rigen los procesos del *SIGAO* son:

Las consultas solo funcionan, si los archivos de las audiencias orales se encuentran convertidas en el *servidor ISILON*, esto porque en algunos casos pasa que las audiencias no se convierten por diferentes motivos y la persona usuaria no puede consultarlas.

## Casos de Uso

### Tabla 9

*Caso de Uso Conversión de Audiencias de SIGAO*

<b>Caso de Uso</b>	<b>CU01 – Cambio en el almacenamiento de la conversión</b>
<b>Actores</b>	Sistema

<b>Propósito</b>	Permite al servicio de conversión convertir los archivos a streaming
<b>Precondiciones</b>	<ol style="list-style-type: none"> <li>1. Tener audiencias pendientes para convertir a streaming para guardar en <i>servidor ISILON</i>.</li> <li>2. El despacho al cual pertenece la audiencia tiene que tener activa la conversión de audiencias.</li> <li>3. El servicio de conversión de audiencias únicamente va a buscar, las audiencias pendientes de los despachos que se tengan configurados en el archivo xml de despachos.</li> </ol>
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. El servicio realizará la consulta de las audiencias que se encuentren pendientes de convertir en la base de datos de Gestión y <i>SIAGPJ</i>. Para que el servicio de conversión realice este proceso se tiene que conectar al <i>API de SIAGPJ</i>.</li> <li>2. Después de esto, el servicio de conversión de audiencias se encarga de conectarse al FTP regional para obtener las audiencias que tiene que convertir a streaming.</li> <li>3. Después de obtener los archivos multimedia, el servicio de conversión de audiencias se encarga de validar los formatos permitidos para la conversión de audio y video.</li> <li>4. En caso de que el audio sea formato MP3 no se tiene que convertir a <i>streaming</i> y continuar el proceso en el punto 7.</li> <li>5. Si el formato es video, el servicio de conversión de audiencias se encarga de convertir el video a streaming, en este caso. Los</li> </ol>

archivos que se crean son un archivo MP4 de audio y otro de video, un archivo MDP, una imagen de previsualización del video formato PNG y cuatro archivos de protocolo DASH. En el archivo de configuración del servicio de conversión de audiencias se encuentra una llave que permite habilitar o deshabilitar el uso de protocolo HLS.

6. El servicio de conversión de audiencias se encarga de guardar los archivos ya convertidos a streaming a una ruta temporal que está especificada en el archivo de configuración.
7. El servicio de conversión de audiencias se encargará de llamar al API controlador de archivos, este guardará los archivos de las audiencias en el servidor ISILON y este responderá con un “true”, si la operación se completa correctamente. Además, la ruta donde se guardó la audiencia se tiene que guardar en una base de datos centralizada.
8. Después de esto, el servicio de conversión se encarga de actualizar los estados en la base de datos que corresponda, para indicar que ya el archivo fue convertido a streaming, el estado de la audiencia pasa ha publicado. Este proceso se realiza por medio del API *SIAGPJ.Audiencias*.
9. Finalmente, el servicio de conversión de audiencias se encarga de eliminar los archivos que se convirtieron y que se encuentran en la ruta temporal de conversión.

<b>Excepciones</b>	
--------------------	--

**Tabla 10***Caso de Uso Consulta de Audiencias*

<b>Caso de Uso</b>	<b>CU02 – Consulta de audiencias</b>
<b>Actores</b>	<ul style="list-style-type: none"> <li>• Sistema</li> <li>• Persona usuaria</li> </ul>
<b>Propósito</b>	Permite a la persona usuaria realizar consultas de audiencias
<b>Precondiciones</b>	<p>Tener permisos para ingresar al sistema.</p> <p>La audiencia tiene que estar convertida en streaming en el ISILON y la audiencia tiene que estar en estado publicado en la base de dato de SIAGPJ o Gestión.</p>
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. La persona usuaria ingresa a los sistemas que permiten realizar la consulta de audiencias.</li> <li>2. La persona usuaria debe ingresar la información que se le solicita, en este caso el número de expediente y de esta forma podrá consultar las audiencias de dicho expediente.</li> <li>3. El sistema tiene que verificar en la base de datos correspondiente donde se encuentra la audiencia consultada, para verificar que la</li> </ol>

	<p>misma esté en estado publicado y devolver el nombre del archivo a consultar en el <i>servidor ISILON</i>.</p> <ol style="list-style-type: none"> <li>4. El sistema se encarga de buscar el archivo de configuración o la configuración, la ruta del nuevo API proxy, después de esto, el sistema va a llamar a este API para que el mismo se encargue de redireccionar a la ruta donde se encuentra la audiencia en el <i>servidor ISILON</i> para realizar cargar la audiencia en el sistema.</li> <li>5. El API proxy se encargará de consultar las audiencias por el nombre del archivo a la nueva base de datos de archivos que va a tener el inventario de las rutas y archivos del <i>servidor ISILON</i>, después de esto el API proxy redireccionará a la ruta donde se encuentra la audiencia.</li> <li>6. La persona usuaria podrá reproducir las audiencias de audio o video que estén relacionadas con el expediente.</li> <li>7. Finalmente, la persona usuaria puede cerrar la aplicación o utilizar las otras funcionalidades del sistema.</li> </ol>
<b>Excepciones</b>	

**Tabla 11**

*Caso de Uso Consulta de audiencias desde Iphone GL*

<b>Caso de Uso</b>	<b>CU03– Consulta de audiencias desde Iphone GL</b>
<b>Actores</b>	<ul style="list-style-type: none"> <li>• Sistema</li> </ul>

	<ul style="list-style-type: none"> <li>• Persona usuaria</li> </ul>
<b>Propósito</b>	<p>Permite a la persona usuaria realizar consultas de audiencias desde GL utilizando <i>Iphone</i></p>
<b>Precondiciones</b>	<p>Tener permisos para ingresar al sistema.</p> <p>La audiencia tiene que estar convertida en <i>streaming</i> en el <i>ISILON</i> y la audiencia tiene que estar en estado publicado en la base de dato de SIAGPJ o Gestión.</p>
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. La persona usuaria ingresa a GL para realizar la consulta de audiencias.</li> <li>2. La persona usuaria debe ingresar la información que se le solicita, en este caso el número de expediente y de esta forma podrá consultar las audiencias de dicho expediente.</li> <li>3. El sistema tiene que verificar en la base de datos correspondiente donde se encuentra la audiencia consultada, para verificar que la misma esté en estado publicado y devolver el nombre del archivo a consultar en el <i>servidor ISILON</i>.</li> <li>4. El sistema se encarga de buscar el archivo de configuración o la configuración, la ruta del nuevo <i>API proxy</i>, después de esto, el sistema va a llamar a este microservicio para que el mismo se encargue de redireccionar a la ruta donde se encuentra la audiencia en el <i>servidor ISILON</i> para realizar la carga la audiencia en el sistema.</li> </ol>

	<ol style="list-style-type: none"> <li>5. El <i>API proxy</i> se encargará de consultar las audiencias por el nombre del archivo a la nueva base de datos de archivos que va a tener el inventario de las rutas y archivos del servidor ISILON, después de esto el <i>API proxy</i> redireccionará a la ruta donde se encuentra la audiencia.</li> <li>6. En este caso, si el dispositivo que está utilizando la persona usuaria es un <i>Iphone</i> el microservicio tiene que devolver el archivo de tipo M3U8. Para la reproducción de las audiencias.</li> <li>7. La persona usuaria podrá reproducir las audiencias de audio o video que estén relacionadas con el expediente.</li> <li>8. Finalmente, la persona usuaria puede cerrar la aplicación o utilizar las otras funcionalidades del sistema.</li> </ol>
<b>Excepciones</b>	

**Tabla 12***Caso de Uso Agregar Nuevas Audiencias*

<b>Caso de Uso</b>	<b>CU04 – Agregar audiencias por medio del nuevo API</b>
<b>Actores</b>	Sistema
<b>Propósito</b>	Recibir el archivo que se quiere guardar en el <i>servidor ISILON</i>
<b>Precondiciones</b>	Tener una audiencia en estado publicado para que la tome el servicio de conversión de audiencias.

**Flujo**

1. El servicio de conversión de audiencias se encargará de llamar al API controlador de archivos, este API recibirá estos datos para empezar con el proceso de guardado de la audiencia en el *servidor ISILON*.
2. Antes de guardar los archivos, el API controlador de archivos se tiene que verificar en la nueva base de datos de administración de archivos del *servidor ISILON*, que no se haya superado el límite permitido de archivos por carpeta en el *servidor ISILON*, en caso de que ese límite se supere el API controlador de archivos se encargará de crear una nueva carpeta “Audiencias + el número contador de carpetas por audiencias” donde se tiene que empezar a guardar las nuevas audiencias.
3. También, se tiene que verificar que archivo no exista en servidor ISILON.
4. El API controlador de archivos se encargará de guardar los archivos de las audiencias en el *servidor ISILON*.
5. Si los archivos de las audiencias se guardan de forma correcta, el API controlador de archivos tiene que guardar en base de datos el nombre del archivo y la ruta en la cual quedó dicho archivo.
6. El API controlador de archivos indicará al servicio de conversión de audiencias que la operación de guardado de la audiencia se ejecutó de forma correcta.

**Excepciones**

**Tabla 13***Caso de Uso Inventariado de Audiencias del ISILON*

<b>Caso de Uso</b>	<b>CU05 – Inventariado de audiencias del ISILON</b>
<b>Actores</b>	Sistema
<b>Propósito</b>	Inventariar todas las audiencias que se encuentran en el <i>servidor ISILON</i> para guardar la información en la nueva base de datos
<b>Precondiciones</b>	<p>Tener archivos de las audiencias en el consolidado.</p> <p>Tener un repositorio configurado en la tabla Configuración Repositorio.</p>
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. Consultar la lista de archivos que se encuentran en la carpeta que se especificó en el <i>appconfig</i> con la ruta del consolidado.</li> <li>2. Luego crear la carpeta en el consolidado del <i>ISILON</i> con el nombre que se encuentra en la tabla <i>cofiguracionRepositorio</i> en este caso sería “Audiencias”.</li> <li>3. Posteriormente se tiene que copiar las audiencias de la raíz del consolidado a la carpeta de “Audiencias”.</li> <li>4. Validar que no se exceda el límite de archivos permitidos, si es el caso se tiene que crear una nueva carpeta para seguir guardando audiencias.</li> <li>5. Se tiene que validar cuando los archivos lleguen al límite de las audiencias permitidas por carpeta, verificar que no se esté copiando un archivo <i>streaming</i> de video, porque los mismos tienen 8 o 4</li> </ol>

	<p>archivos, los cuales no pueden quedar en carpetas por arparte, porque estos trabajan en conjunto para la reproducción de audiencias.</p> <ol style="list-style-type: none"> <li>6. Obtener la ruta y el nombre de los archivos de la lista de archivos.</li> <li>7. Crear la conexión a la base de datos que tiene el inventario de los archivos <i>ISILON</i>.</li> <li>8. Agregar los datos a la tabla archivo de la base de datos de inventario de archivos <i>ISILON</i>.</li> </ol>
<b>Excepciones</b>	

**Tabla 14**

*Caso de Uso Exportación de Audiencias en SIGAO*

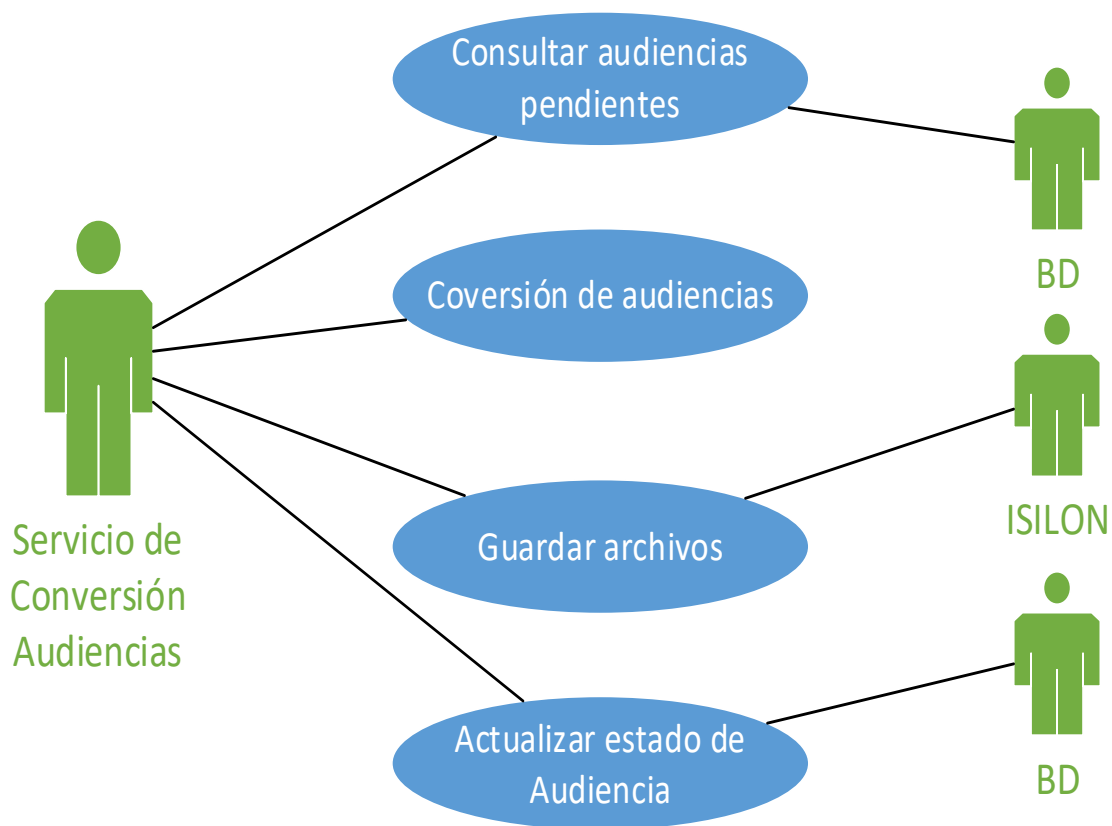
<b>Caso de Uso</b>	<b>CU06 – Exportación de audiencias</b>
<b>Actores</b>	Sistema
<b>Propósito</b>	Realizar la exportación de audiencias de audio y video.
<b>Precondiciones</b>	Tener una audiencia en publicada en el servidor <i>ISILON</i> .
<b>Flujo</b>	<ol style="list-style-type: none"> <li>1. Se consulta el expediente desde el sistema de <i>SIGAO</i>.</li> <li>2. El sistema carga las audiencias registradas.</li> <li>3. Se selecciona la audiencia y se pulsa el botón de Exportar.</li> <li>4. El sistema muestra la ventana de exportación.</li> <li>5. Se indica la ubicación donde se van a guardar las audiencias</li> </ol>

Excepciones	<p>6. El sistema debe consumir el nuevo microservicio del S3 en caso de que la audiencia esté publicada en el <i>servidor ISILON</i>, si no se debe consumir el FTP del regional, para poder descargar los archivos de las audiencias.</p> <p>7. El sistema descarga las audiencias</p>
-------------	---

**Diagramas Casos de Uso**

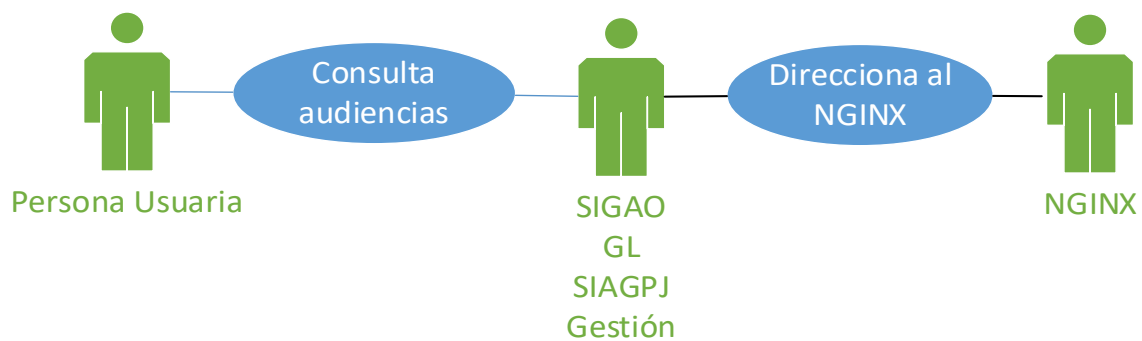
**Figura 47**

*Diagrama Caso Conversión de audiencias de SIGAO*



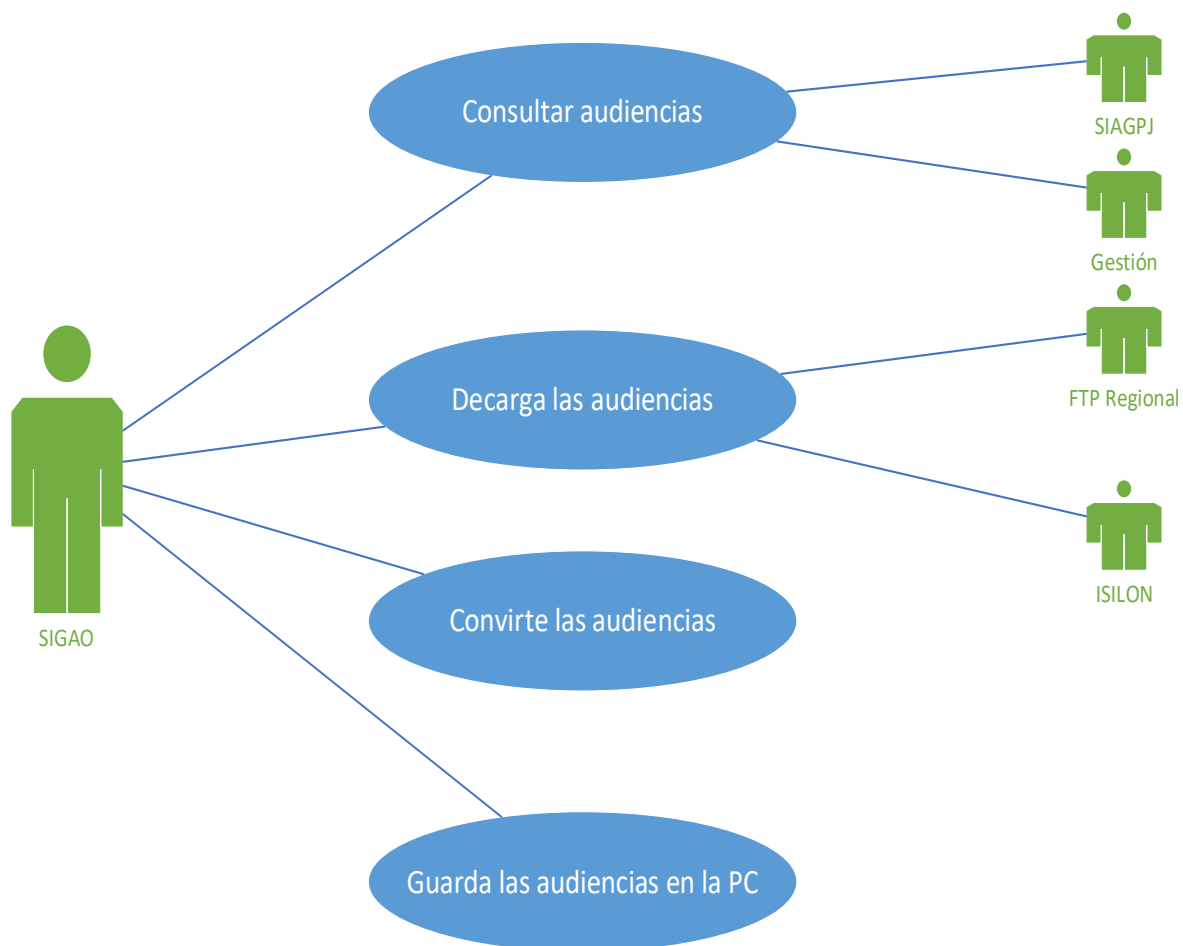
**Figura 48**

*Diagrama Caso de Consulta de Audiencias*



**Figura 49**

*Diagrama Caso de Exportación de Audiencias de SIGAO*



## Diagramas de Secuencia

**Figura 50**

*Diagrama de Secuencia Servicio Conversión de Audiencias*

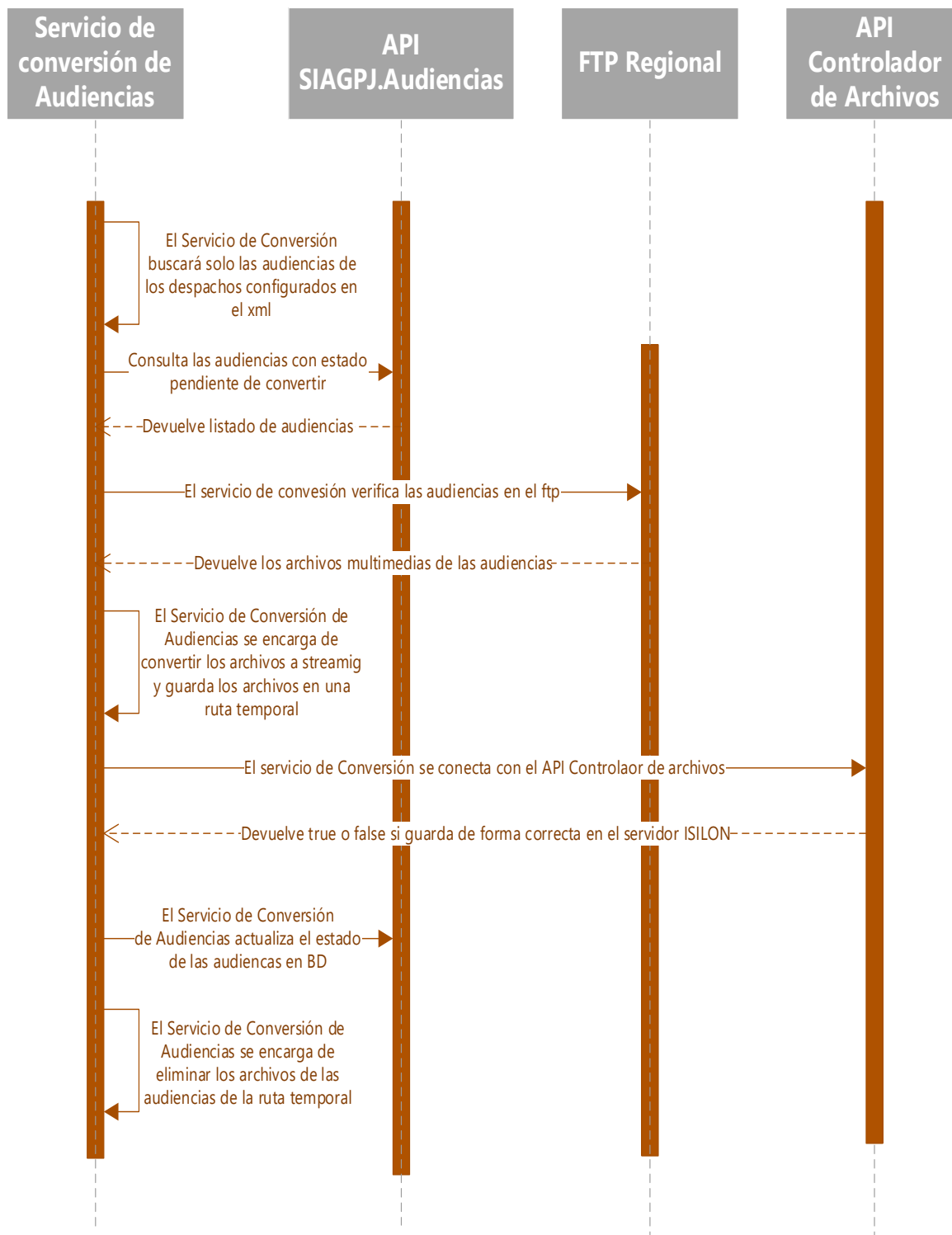


Figura 51

Diagrama de Secuencia Conversión de Audiencias Microservicio S3

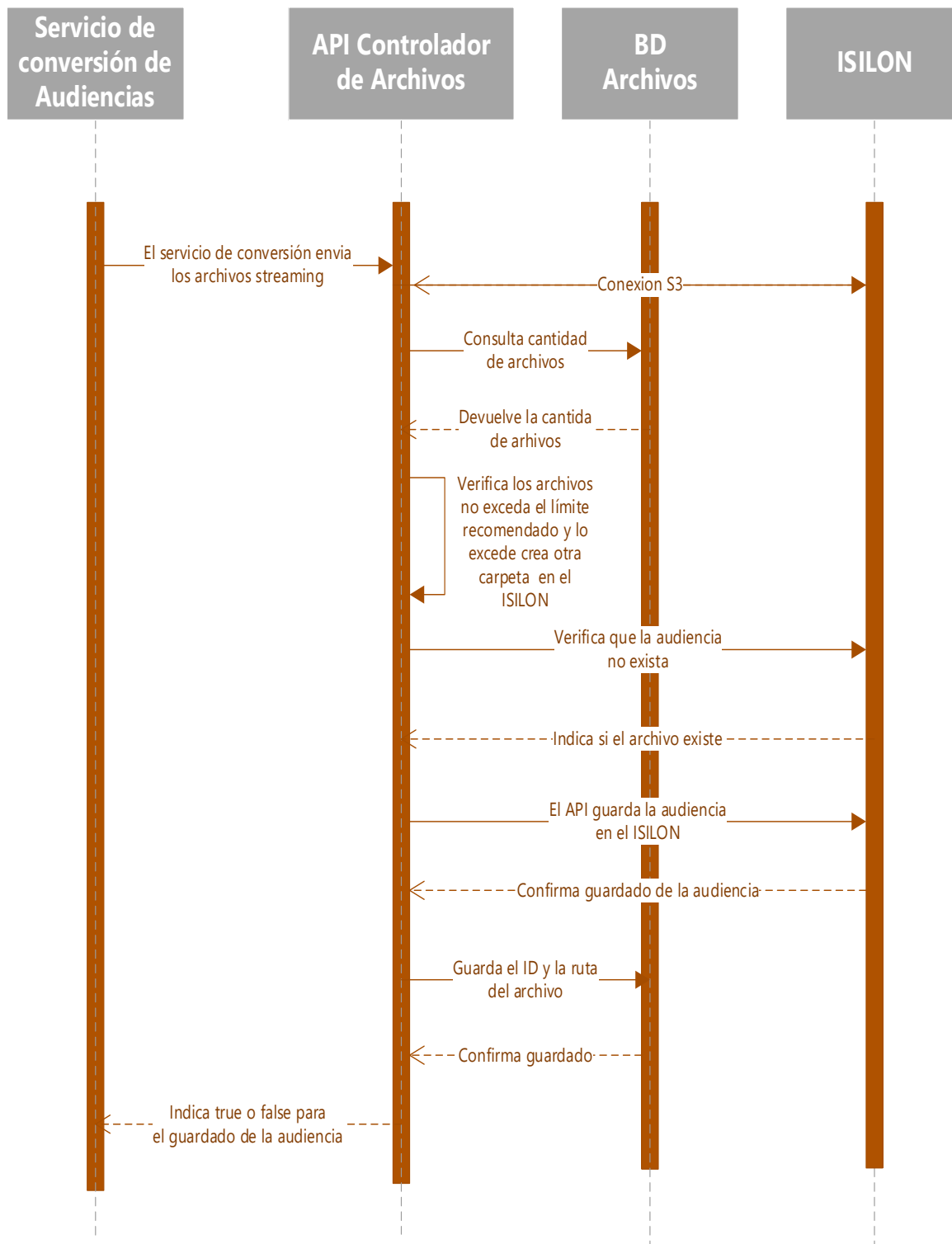
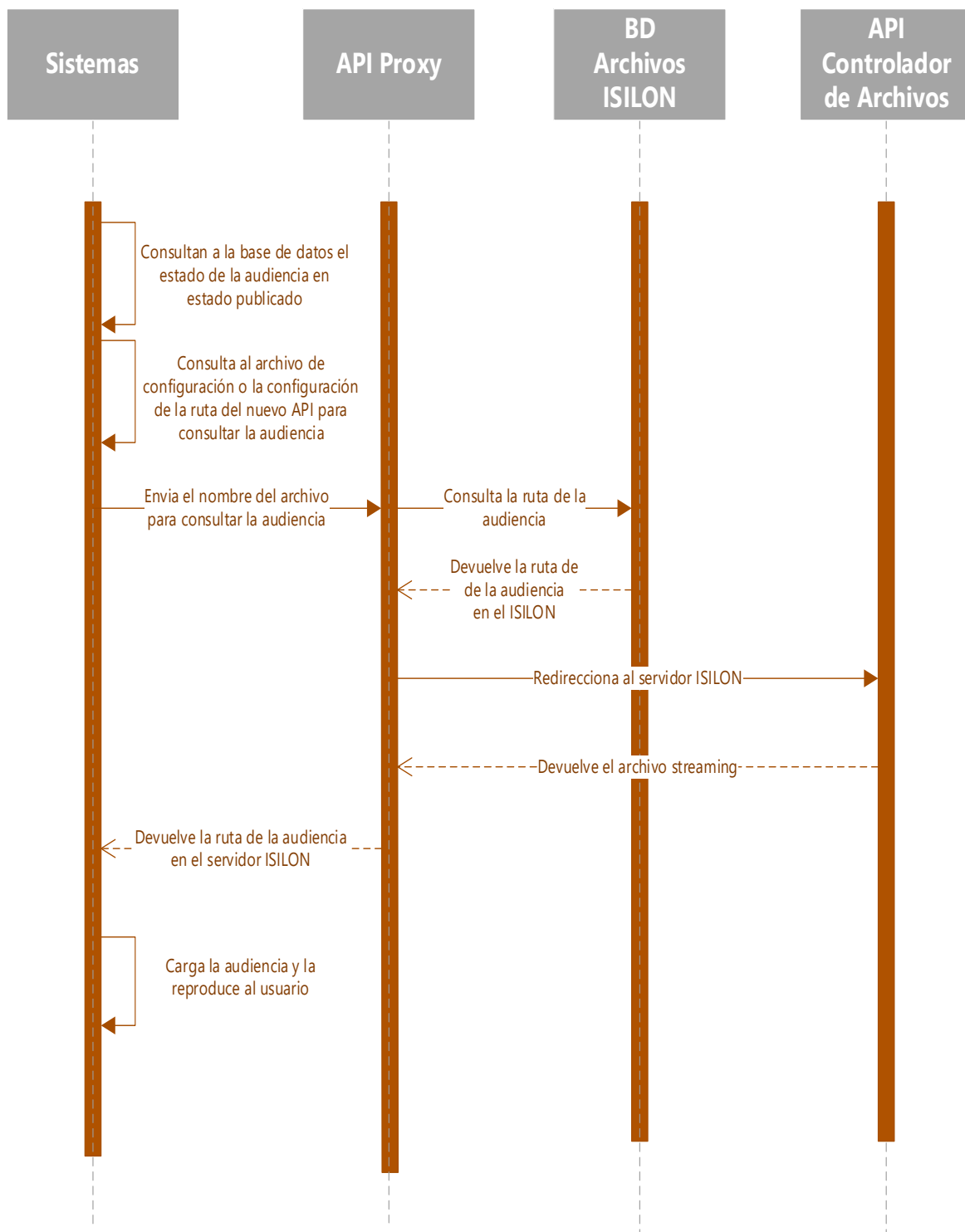


Figura 52

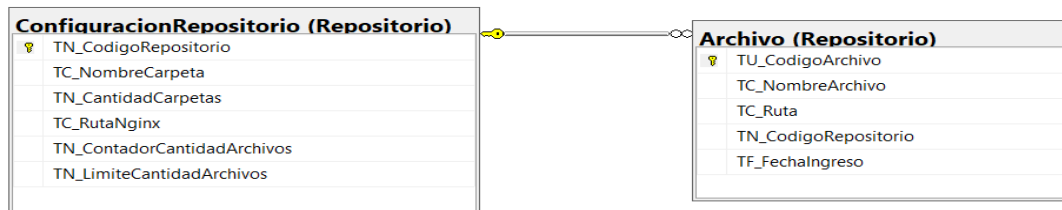
Diagrama de Secuencia de Consulta de Audiencias



## Base de Datos

**Figura 53**

*Diagrama de Base de Datos*



**Tabla 15**

*Diccionario de Datos*

Llave	Campo	Tipo	Tamaño	Nulo	Descripción
PK	TN_CódigoRepositorio	<i>SMALLINT</i>	-	NO	PK autoincrementable. Código para el repositorio.
	TC_NombreCarpeta	<i>VARCHAR (50)</i>	50	NO	Nombre de la carpeta creada en Isilon.
	TN_CantidadCarpetas	<i>SMALLINT</i>	-	NO	Cantidad de carpetas.
	TC_RutaNginx	<i>VARCHAR (50)</i>	50	NO	Ruta raíz, en este caso del servidor Isilon.
	TN_ContadorCantidadArchivos	<i>INT</i>	-	NO	Contador de archivos guardados en la carpeta.
	TN_Límite Cantidad Archivos	<i>INT</i>	-	Sí	Límite de archivos permitidos por carpeta.
PK	TU_Código Archivo	<i>UNIQUEIDENTIFIER</i>	-	NO	PK autoincrementable. Código único para el archivo.
	TC_Nombre Archivo	<i>VARCHAR (50)</i>	50	NO	Nombre del archivo.
	TC_Ruta	<i>VARCHAR (150)</i>	150	NO	Ruta donde se encuentra el archivo.
	TN_Código repositorio	<i>SMALLINT</i>	-	NO	Código del repositorio donde se guardará el archivo.
	TF_FechaIngreso	<i>DATETIME2(7)</i>	-	NO	Fecha de ingreso del archivo.

## Desarrollo y Pruebas de los Microservicios con Sistemas

Figura 54

Interfaz Swagger Microservicio Consulta Audiencias

Swagger UI

localhost:5045/swagger/index.html

Swagger  
Supported by SMARTBEAR

Select a definition SIAGPJ.StreamingAudiencias.WebApi v1

# Micro Servicios EF Core SIAGPJ <sup>v1</sup> OAS3

<http://localhost:5045/swagger/v1/swagger.json>

Este es un Web Api de SIAGPJ permite realizar consultas archivos streaming

[Contact Subproceso de Sistemas Jurisdiccionales](#)

Authorize

## Consulta

GET	/ReproducirAudiencia	Redirige a una consulta de streaming según el tipo de archivo y el nombre.	✓	🔒
GET	/audio.png	Acción que responde a las solicitudes HTTP GET en el endpoint "/audio.png". Se encarga de recuperar una imagen de audio desde el servidor Nginx para que aparezca en pantalla de SIGAO.	✓	🔒

Figura 55

Interfaz Swagger Microservicio Encargado de Guardar las Audiencias

Swagger UI

localhost:44373/swagger/index.html

Select a definition SIAGPJ.ControladorAudiencias.WebApi v1

# Micro Servicios EF Core SIAGPJ <sup>v1</sup> OAS3

<https://localhost:44373/swagger/v1/swagger.json>

Este es un Web Api de SIAGPJ para realizar guardar archivos streaming

[Contact Subproceso de Sistemas Jurisdiccionales](#)

Authorize

## Archivo

- POST** /api/ArchivosS3/IniciarCargaMultiparte Método que inicia la carga multipartes de un archivo específico y consulta el código de bucket(carpetas) y devuelve una entidad de tipo resultado donde va el nombre del bucket.
- POST** /api/ArchivosS3/SubirPartesArchivos Metodo encargado de subir archivos por partes
- POST** /api/ArchivosS3/CompletarMultipartes Metodo encargado de subir archivos por partes
- POST** /api/ArchivosS3/GuardarInformacionArchivoBD Metodo encargado de guardar información de los archivos a la base de datos por bucket

Figura 56

*Worker Servicios Inventariado Audiencias*

```

E:\Tesis\Woker\Inventariad x + v
info: InventariadoIsilonWokerService.Worker[0]
  Recuerda que antes de ejecutar este utilitario, debe haber configurado la tabla RepositorioConfiguracionRepositori
o. Presiona Enter para continuar...

Este equipo > ADATA SE880 (E:) > Tesis > Woker >

E:\Tesis\Woker\Inventariad x + v
VALUES (@p0, @p1, @p2, @p3, @p4),
(@p5, @p6, @p7, @p8, @p9),
(@p10, @p11, @p12, @p13, @p14),
(@p15, @p16, @p17, @p18, @p19),
(@p20, @p21, @p22, @p23, @p24),
(@p25, @p26, @p27, @p28, @p29),
(@p30, @p31, @p32, @p33, @p34),
(@p35, @p36, @p37, @p38, @p39),
(@p40, @p41, @p42, @p43, @p44),
(@p45, @p46, @p47, @p48, @p49),
(@p50, @p51, @p52, @p53, @p54),
(@p55, @p56, @p57, @p58, @p59),
(@p60, @p61, @p62, @p63, @p64),
(@p65, @p66, @p67, @p68, @p69),
(@p70, @p71, @p72, @p73, @p74),
(@p75, @p76, @p77, @p78, @p79),
(@p80, @p81, @p82, @p83, @p84),
(@p85, @p86, @p87, @p88, @p89),
(@p90, @p91, @p92, @p93, @p94);
UPDATE [Repositorio].[ConfiguracionRepositorio] SET [TN_ContadorCantidadArch
OUTPUT 1
WHERE [TN_CodigoRepositorio] = @p96;
*****
*
*           La ejecución del Worker Service ha finalizado correctamente.
*
*****
info: InventariadoIsilonWokerService.Worker[0]
  La ejecución del Worker Service ha finalizado correctamente.

```

**Figura 57***Prueba Reproducción Audiencia de Audio Gestión en Línea*

The screenshot shows a web browser window with the address bar displaying `localhost:6567/inicio/consultas/consultanumeroexpediente/detalleexpediente`. The main content area features a large audio waveform player with a blue waveform on a black background. Below the waveform is a playback control bar showing a progress indicator at `0:01 / 0:07`, a play/pause button, and a volume icon. To the right of the waveform, there is a search filter section labeled "etiquetas" with a dropdown menu showing "0 de 0" and navigation arrows. Below this, it states "Registros encontrados: 0".

Below the audio player, there are two document thumbnails. The left one is labeled "Video" and shows a screenshot of a web interface with the text "Definir Tipos para trámite Incorporar Documento Multimedia" and a timestamp of "20/10/2024 22:51:11". The right one is labeled "Audio 1" and shows the logo of the Poder Judicial of Costa Rica with the text "Definir Tipos para trámite Incorporar Documento Multimedia" and a timestamp of "20/10/2024 22:50:10". Both thumbnails have a blue play button icon.

In the bottom right corner, there is a "CHATBO" chatbot icon and a blue accessibility icon.

Figura 58

Prueba Reproducción Audiencia de Video Gestión en Línea

Sistema de Gestión en Línea

localhost:6567/inicio/consultas/consultanumeroexpediente/detalleexpediente

Despacho: JUZGADO PENAL DEL CIRCUITO JUDICIAL DE SAN JOSE-SUOEDALRIDESOBINIST

Salir Versión 3.0.0

Fecha: 20/10/2024 22:41:06

Poder Judicial - Grabador de Audios (v6c)

Realizar transcripción automática  Hacer grabación de prueba de equipo

Grabar Monitor

Monitor: [dropdown]

Circuito: SEGUNDO CIRCUITO JUDICIAL DE SAN JOSE [dropdown] Salir

Utilizar cámara de video

Tipo de audiencia: Definir Tipo para trámite Incorporar Documento Multimedia [dropdown]

Descripción de la grabación: Video

Etiquetas

N°	Tiempo	Nota	Privado
----	--------	------	---------

Opciones del grabador

00:00:00

Cámaras disponibles: [dropdown]

Integrar Webcam

Microfonos disponibles

Valor referencia (Realtek Audio)

Entrada de audio: [dropdown] Entrada de video: [dropdown]

Nivel de Audio: [slider]

0:00 / 0:10

Video Definir Tipos para trámite Incorporar Documento Multimedia

20/10/2024 22:51:11

Audio 1 Definir Tipos para trámite Incorporar Documento Multimedia

COSEVI

Figura 59

## Audiencias de Prueba Sistema SIGAO

Preparación de la Audiencia v3.8.0.0

Contexto

Despacho: JUZGADO PENAL DEL II CIRCUITO JUDICIAL DE SAN JOSE - SJOBDJURDES03INST

[Salir](#)

Versión 3.8.0.0

Preparación de la Audiencia

NUE:     Fecha:

Descripción:

Despacho Judicial:  Juez:

Intervieniente **Audiencias**

Buscar:  Ordenar:

Fecha	Descripción	Tipo de Audiencia	Duración
<input checked="" type="checkbox"/> 20/10/2024 10:51:11	Video	Definir Tipos para trámite Incor...	00:00:10.05
<input type="checkbox"/> 20/10/2024 10:50:10	Audio 1	Definir Tipos para trámite Incor...	00:00:07.59

Figura 60

Prueba Reproducción Audiencia de Video SIGAO

The screenshot displays the 'Preparación de la Audiencia' software interface. At the top, a header bar shows 'Preparación de la Audiencia v3.8.0.0'. Below this, there are two main panels: 'Contexto' on the left and 'Preparación de la Audiencia' on the right. The 'Contexto' panel contains the text: 'Despacho: JUZGADO PENAL DEL II CIRCUITO JUDICIAL DE SAN JOSE - SUOEDAMR020301NST' and 'Versión 3.8.0.0'. The 'Preparación de la Audiencia' panel shows a form with fields for 'NUE:' (24/10/2024 05:15 PE) and 'Fecha:' (20/10/2024 22:41:36). Below these fields, there are several sections: 'Opciones de grabación' with checkboxes for 'Realizar transcripción automática' and 'Hacer grabación de prueba de equipo'; 'Grabar Monitor'; 'Monitor' dropdown; 'Circuito' dropdown set to 'SEGUNDO CIRCUITO JUDICIAL DE SAN JOSE'; 'Utilizar addresses de audio'; 'Tipo de audiencia' dropdown set to 'Define Tipo para trámite Incorporar Documento Multimedia'; 'Descripción de grabación' dropdown set to 'Video'; and a table with columns 'MP', 'Tiempo', 'Nota', and 'Estado'. A video player is embedded in the center, showing a preview of the software interface. To the right of the video player is a 'Notas' section with the text 'Buscar en etiquetas:' and an empty search box. At the bottom of the video player, there is a progress bar and a volume control. Below the video player, a light blue box contains the following information: 'Tipo de audiencia: Definir Tipos para trámite Incorporar Documento Multimedia', 'Duración: 00:00:10', and 'Fecha de la audiencia: 20-10-2024 10:51 pm'.

**Figura 61***Prueba Reproducción Audiencia de Audio SIGAO*


Preparación de la Audiencia v3.8.0.0

Contexto  
Despacho: JUZGADO PENAL DEL II CIRCUITO

Preparación de la Audiencia  
NUE:


Poder Judicial - Reproductor de Audiencias

# Audio 1



## Notas

Buscar en etiquetas:

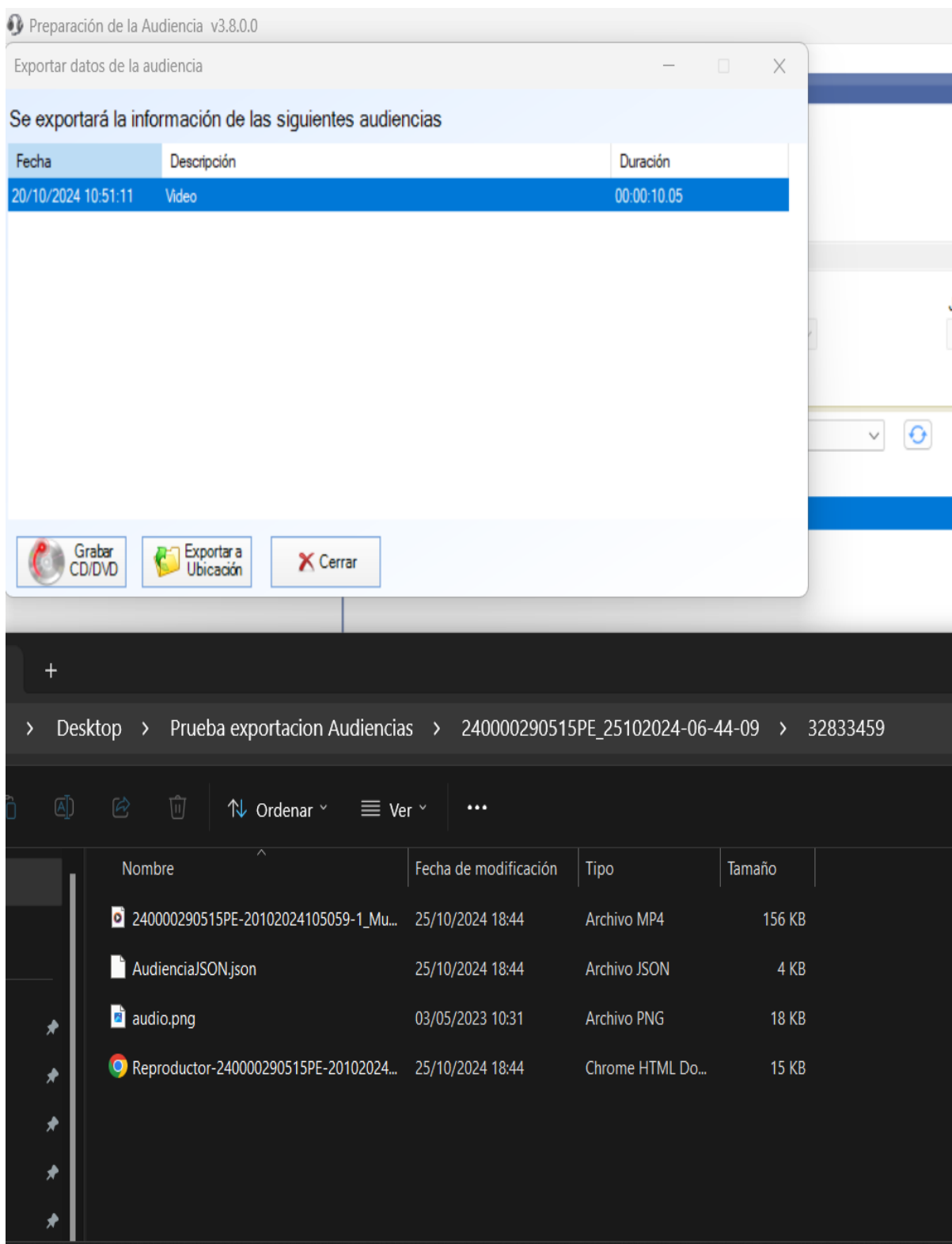


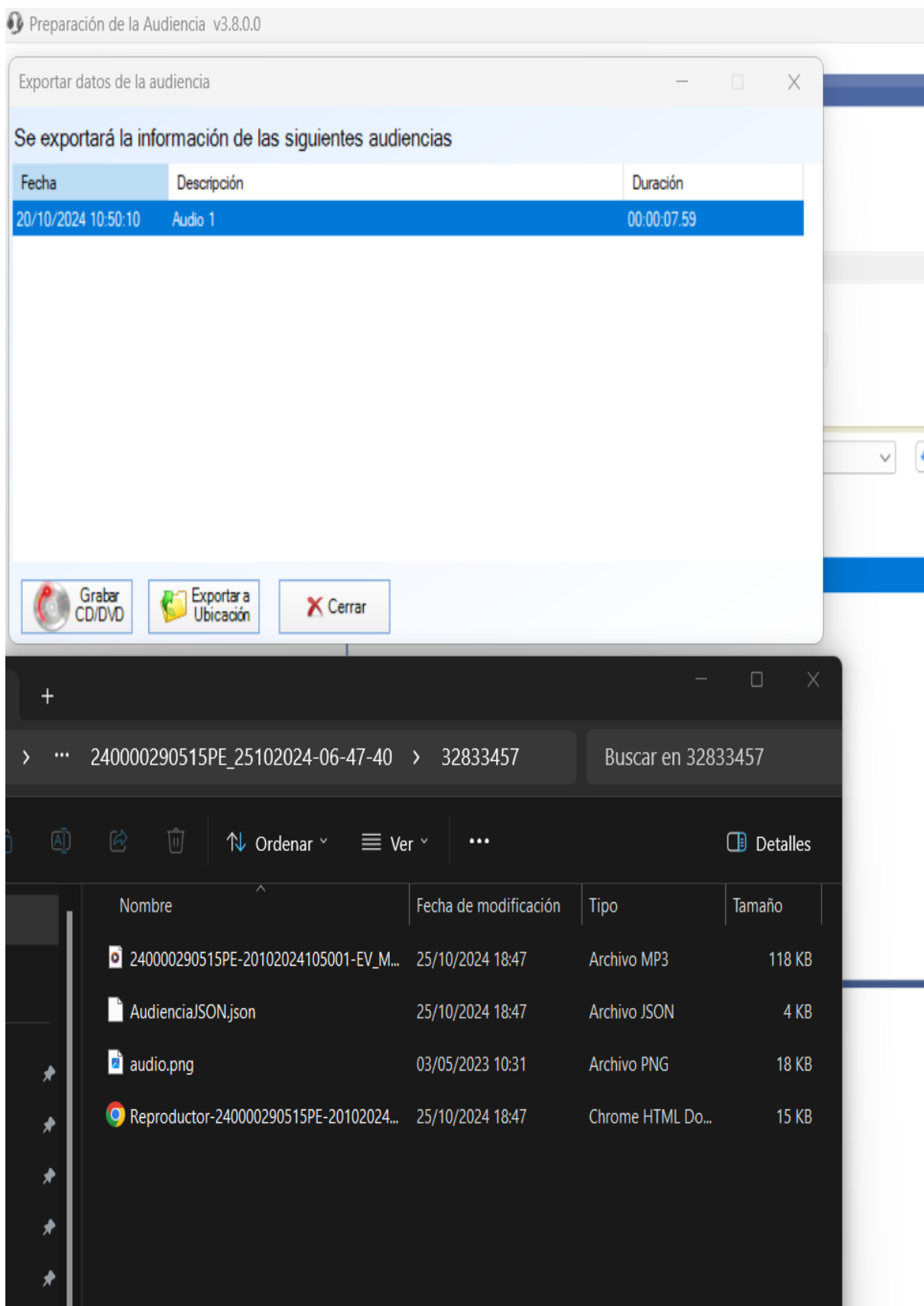
0:08 0:08

**Tipo de audiencia:** Definir Tipos para trámite Incorporar Documento Multimedia

**Duración:** 00:00:07

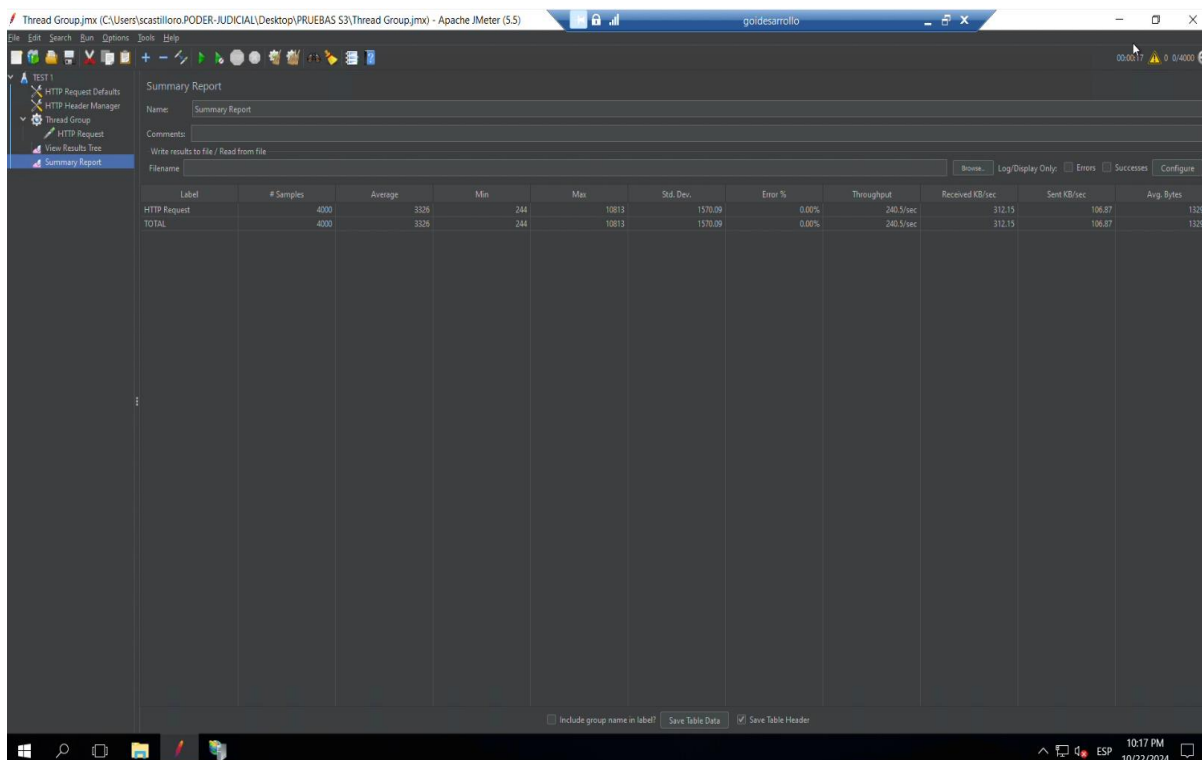
**Fecha de la audiencia:** 20-10-2024 10:50 pm

**Figura 62***Prueba Exportación de Audiencia Video SIGAO*

**Figura 63***Prueba Exportación de Audiencias Audio SIGAO*

**Figura 64**

*Pruebas de Carga Consulta de Audiencias*



**Figura 65**

*Pruebas de Carga Guardar Audiencias*

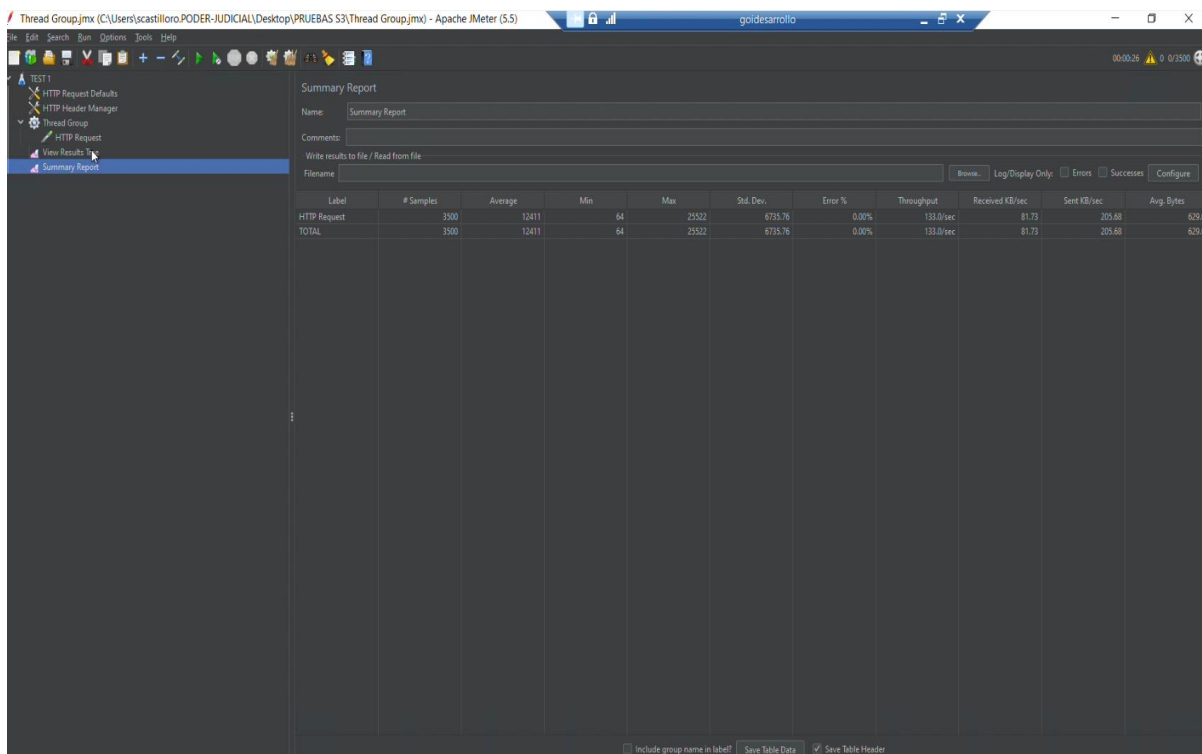


Figura 66

*Pruebas Unitarias Microservicios Encargado de Guardar las Audiencias*

SIAGPJ.ControladorAudiencias.WebApi.XunitTest

SIAGPJ.ControladorAudiencias.WebApi.XunitTest.Controller.IniciarCargaMultiparte()

1 us1no Amazon.S3.Model;

Explorador de pruebas

19 19 0

Serie de pruebas finalizada: 5 pruebas (Superadas: 5; Con errores: 0; Omitidas: 0) ejecutadas en 1 s

Prueba	Duración	Rasgos	Mensaje de error
SIAGPJ.ControladorAudiencias.WebApi.XunitTest (14)	164 ms		
SIAGPJ.StreamingAudiencias.WebApi.XunitTest (5)	141 ms		
SIAGPJ.StreamingAudiencias.WebApi.XunitTest.Controller (3)	69 ms		
ConsultaControllerTest (3)	69 ms		
ConsultamagenAudioNginx	66 ms		
RedireccionarConsultaStreaming	2 ms		
RedireccionarConsultaStreamingError	1 ms		
SIAGPJ.StreamingAudiencias.WebApi.XunitTest.Services (2)	72 ms		
ConsultaServicesTest (2)	72 ms		
RedireccionarConsultaStreaming	2 ms		
RedireccionarConsultaStreamingM3U8	70 ms		

Ejecutar Depurar

Resumen del grupo

SIAGPJ.StreamingAudiencias.WebApi.XunitTest

Pruebas en grupo: 5

Duración total: 141 ms

Salidas

5 Correcta

Figura 67

*Pruebas Unitarias Microservicios Encargado de Consultar las Audiencias*

SIAGPJ.ControladorAudiencias.WebApi.XunitTest

SIAGPJ.ControladorAudiencias.WebApi.XunitTest.Controller.IniciarCargaMultiparte()

usinq Amazon.S3.Model;

Explorador de pruebas

19 19 0

Serie de pruebas finalizada: 5 pruebas (Superadas: 5; Con errores: 0; Omitidas: 0) ejecutadas en 1 s

Prueba	Duración	Rasgos	Mensaje de error
SIAGPJ.ControladorAudiencias.WebApi.XunitTest (14)	164 ms		
SIAGPJ.StreamingAudiencias.WebApi.XunitTest (5)	141 ms		
SIAGPJ.StreamingAudiencias.WebApi.XunitTest.Controller (3)	69 ms		
ConsultaControllerTest (3)	69 ms		
ConsultalmagenAudioNginx	66 ms		
RedireccionarConsultaStreaming	2 ms		
RedireccionarConsultaStreamingError	1 ms		
SIAGPJ.StreamingAudiencias.WebApi.XunitTest.Services (2)	72 ms		
ConsultaServicesTest (2)	72 ms		
RedireccionarConsultaStreaming	2 ms		
RedireccionarConsultaStreamingM3U8	70 ms		

Ejecutar | Depurar

Resumen del grupo

SIAGPJ.StreamingAudiencias.WebApi.XunitTest

Pruebas en grupo: 5

Duración total: 141 ms

Salidas

5 Correcta

## Referencias

- OKHOSTING SC. (28 de 05 de 2024). *El Ciclo de Vida del Software*. okhosting.com.  
<https://okhosting.com/blog/el-ciclo-de-vida-del-software/>
- ADEN. (11 de Junio de 2024). *ADEN*. Metodologías ágiles: ¿Qué son y cuáles son las más utilizadas?.  
<https://www.aden.org/business-magazine/metodologias-agiles/>
- Amazon Web Services. (2024). *¿Qué es Amazon S3? AWS*.  
[https://docs.aws.amazon.com/es\\_es/AmazonS3/latest/userguide/Welcome.html](https://docs.aws.amazon.com/es_es/AmazonS3/latest/userguide/Welcome.html)
- ATLAS. (15 de Mayo de 2024). *ATLAS.ti*. Proceso de investigación . <https://atlasti.com/es/research-hub/proceso-de-investigacion>
- Barrios Contrera, D. A. (2018). Arquitectura de microservicios. *Publicación de la Facultad de Ingeniería: Tecnología, Investigación y Academia (TIA)*, 6 (1), 11. <https://doi.org/2344-8288>
- Bertomeu, P. F. (2016). *University of Barcelona*. La entrevista.  
<https://diposit.ub.edu/dspace/bitstream/2445/99003/1/entrevista%20pf.pdf>;
- Bhaskar. (26 de Abril de 2024). *nimble humanize work*. ¿Qué es la metodología ágil?.  
<https://www.nimblework.com/es/agile/metodologia-agil/>
- Boardmix. (24 de Enero de 2024). *Boardmix*. Guía completa sobre el diagrama de casos de uso.  
<https://boardmix.com/es/articles/case-use-diagram/>
- Canvia. (17 de 02 de 2023). *Arquitectura de software: Definición, elementos y tipos*. canvia.com.  
<https://canvia.com/arquitectura-software/>
- Cardona Taltavull, J. A. (23 de Enero de 2015). *University of Barcelona*. PROGRAMACIÓN LÓGICA.  
<https://diposit.ub.edu/dspace/bitstream/2445/64643/1/memoria.pdf>
- Cesce . (23 de Septiembre de 2022). *Cesce*. Gestión de riesgos: qué es y cómo elaborar un plan de gestión para tus proyectos . <https://www.cesce.es/es/w/asesores-de-pymes/gestion-de-riesgos>
- Cevallos, K. (4 de Junio de 2015). *Ingeniería del Software*. UML: Casos de Uso.  
<https://ingsoftwarekarlacevallos.wordpress.com/2015/06/04/uml-casos-de-uso/>
- Chicaiza, D. R. (05 de 03 de 2020). *DISEÑO DE UN PROTOTIPO DE UNA ARQUITECTURA BASADA EN*.  
<https://dspace.ups.edu.ec/bitstream/123456789/18532/1/UPS%20-%20ST004466.pdf>
- DELL. (11 de 01 de 2019). *SCALABLE DELL EMC ISILON STORAGE SOLUTIONS WITH DELL EMC OPEN NETWORK FABRICS*. i.dell.com.  
[https://i.dell.com/sites/csdocuments/Product\\_Docs/en/dell\\_emc\\_networking\\_solution\\_brief\\_fabrics\\_for\\_isilon.pdf](https://i.dell.com/sites/csdocuments/Product_Docs/en/dell_emc_networking_solution_brief_fabrics_for_isilon.pdf)
- DELL. (08 de 04 de 2022). *Dell EMC PowerScale*. delltechnologies.  
<https://www.delltechnologies.com/asset/es-pe/products/storage/briefs-summaries/h11757-ho-powerscale-onefs-at-a-glance.pdf>

- Dell Technologies. (01 de 06 de 2023). *PowerScale OneFS 9.4.0.0 Technical Specifications Guide*. dl.dell.com. <https://dl.dell.com/content/manual61682642-powerscale-onefs-9-4-0-0-technical-specifications-guide.pdf?language=en-us>
- Delltechnologies. (2023). *Dell PowerScale OneFS: Authentication, Identity Management, and Authorization*. www.delltechnologies.com. <https://www.delltechnologies.com/asset/en-us/products/storage/industry-market/h13115-wp-emc-isilon-onefs-multiprotocol-security-untangled.pdf>
- Editorial IONOS. (11 de Marzo de 2019). *IONOS*. El modelo en cascada: desarrollo secuencial de software. <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>
- Edraw. (16 de 08 de 2024). *Diagrama de Arquitectura de Software*. www.edrawsoft.com. [https://www.edrawsoft.com/es/software-architecture.html?gad\\_source=1&gl=1\\*1bu063g\\*up\\*MQ..&gclid=CjwKCAjwiaa2BhAiEiwAQBgYHuwoasy6kVLEaobWkTWDBbXдноExygVhLbUB0pL7FLVnQwiLbFUUhoCdmsQAvD\\_BwE](https://www.edrawsoft.com/es/software-architecture.html?gad_source=1&gl=1*1bu063g*up*MQ..&gclid=CjwKCAjwiaa2BhAiEiwAQBgYHuwoasy6kVLEaobWkTWDBbXдноExygVhLbUB0pL7FLVnQwiLbFUUhoCdmsQAvD_BwE)
- EDUCATICA. (2024). *5. Pila de protocolos TCP/IP*. www.educatica.es. <https://www.educatica.es/redes/5-pila-de-protocolos-tcp-ip/>
- El Profe Alegría. (16 de 06 de 2024). *El Profe Alegría*. Factibilidad Técnica, Económica y Operativa. <https://elprofealegria.com/factibilidad-tecnica-economica-y-operativa/>
- EVALUANDO SOFTWARE.COM. (2024). *Clasificación del almacenamiento de datos o Data Storage*. www.evaluandosoftware.com. <https://www.evaluandosoftware.com/cloud-computing/clasificacion-del-almacenamiento-datos-data-storage/>
- Evotic. (2024). *Ciclo de Vida del Software*. evotic.es. <https://evotic.es/software-a-medida/ciclo-de-vida-del-software/>
- FREEPIK. (2024). *Vectores SQL*. www.freepik.com. <https://www.freepik.com/vectors/sql>
- FUMERO . (2021). *FUMERO*. Estudios de factibilidad económica y financiera . <https://fumerocpa.com/estudios-de-factibilidad-economica-y-financiera/>
- García, M. R. (Marzo de 2021). *conalepveracruz*. Programación. <https://www.conalepveracruz.edu.mx/iniciobackup/wp-content/uploads/2021/03/Programaci%C3%B3n-orientada-a-objetos-M%C3%93DULO-PROFESIONAL.pdf>
- Ginzo. (28 de Diciembre de 2021). *Ginzo technologies*. Tipos de Metodología clásica en desarrollo de software . <https://ginzo.tech/metodologia-clasica-desarrollo-software/https://ginzo.tech/metodologia-clasica-desarrollo-software/>
- González, P. K. (2023). *www.kerwa.ucr.ac.cr*. DEFENSA DE BLANCO MÓVIL PARA DIVERSIFICACIÓN DE MICROSERVICIOS EN KUBERNETES. <https://www.kerwa.ucr.ac.cr/items/7cddf170-a653-4823-8959-9f089eed3109>
- Google Cloud. (22 de 05 de 2024). *¿Qué es una base de datos relacional?* cloud.google.com. <https://cloud.google.com/learn/what-is-a-relational-database?hl=es-419>

- Gottdiener, Z. (25 de Enero de 2022). *https://www.door3.com*. <https://www.door3.com/es/blog/understanding-risk-management-in-software-development-7-common-risks>
- Hernández, D. (1 de Julio de 2022). *Modelos del ciclo de vida del software*. Ciclo de vida del software. <https://www.linkedin.com/pulse/modelos-del-ciclo-de-vida-software-daniel-hernandez/>
- HUAWEI. (2024). *¿Qué es DAS, NAS y SAN? - Conceptos básicos | Almacenamiento*. [forum.huawei.com. https://forum.huawei.com/enterprise/es/Conceptos-b%C3%A1sicos-de-almacenamiento-%C2%BFQu%C3%A9-es-DAS-NAS-y-SAN/thread/667227040744620032-667212885295771648](https://forum.huawei.com/enterprise/es/Conceptos-b%C3%A1sicos-de-almacenamiento-%C2%BFQu%C3%A9-es-DAS-NAS-y-SAN/thread/667227040744620032-667212885295771648)
- IBM. (2024). *¿Qué es el almacenamiento de datos?* [www.ibm.com. https://www.ibm.com/es-es/topics/data-storage](https://www.ibm.com/es-es/topics/data-storage)
- Instituto Tecnológico PN. (2011). *Ingeniería de software*. Metodologías de desarrollo. <http://itpn.mx/recursosisc/6semestre/ingenieriadesoftware/Unidad%20II.pdf>
- J. Bigelow, S., Lutkevich, B., & Kranz, G. (2024). *¿Qué es el almacenamiento conectado a red (NAS)? Una guía completa*. [www.techtarget.com. https://www.techtarget.com/searchstorage/definition/network-attached-storage](https://www.techtarget.com/searchstorage/definition/network-attached-storage)
- Jacobson, I. (Agosto de 2013). *Ivar Jacobson International*. CASOS DE USO. [https://www.ivarjacobson.com/files/field\\_iji\\_file/article/use\\_case\\_2.0\\_-\\_spanish\\_translation.pdf](https://www.ivarjacobson.com/files/field_iji_file/article/use_case_2.0_-_spanish_translation.pdf)
- Learn Microsoft. (23 de 05 de 2023). *Diseñar diagramas de base de datos (Visual Database Tools)*. [learn.microsoft.com/. https://learn.microsoft.com/es-es/sql/ssms/visual-db-tools/design-database-diagrams-visual-database-tools?view=sql-server-ver16](https://learn.microsoft.com/es-es/sql/ssms/visual-db-tools/design-database-diagrams-visual-database-tools?view=sql-server-ver16)
- Llamas, J. (01 de 12 de 2020). *economipedia.com*. Sistema informático. <https://economipedia.com/definiciones/sistema-informatico.html>
- Lombeida, L. R. (23 de Agosto de 2020). *SlideShare*. Diagramas de secuencia . <https://es.slideshare.net/slideshow/diagramas-de-secuencia-238162313/238162313>
- López, A. (Septiembre de 2018). *agileexperience*. Estudio comparativo de metodologías tradicionales y ágiles para proyectos de Desarrollo de Software. <https://agileexperience.es/wp-content/uploads/2020/06/TFG-I-1015.pdf>
- López, D., & Maya, E. (23 de 07 de 2017). *Arquitectura de Software basada en Microservicios para Desarrollo de Aplicaciones Web*. Universidad Técnica del Norte. <https://documentos.redclara.net/bitstream/10786/1277/1/93%20Arquitectura%20de%20Software%20basada%20en%20Microservicios%20para%20Desarrollo%20de%20Aplicaciones%20Web.pdf>
- López, J. D. (25 de 10 de 2017). *ARQUITECTURA DE SOFTWARE BASADA EN MICROSERVICIOS PARA DESARROLLO DE APLICACIONES WEB DE LA ASAMBLEA NACIONAL*. [core.ac.uk. https://core.ac.uk/download/pdf/200329168.pdf](https://core.ac.uk/download/pdf/200329168.pdf)
- López, N. (13 de Enero de 2024). *Boardmix*. Descubra ejemplos de diagramas de secuencia: Uso práctico y planificación eficaz. <https://boardmix.com/es/examples/sequence-diagram-examples/>

- Martínez, F. Y. (25 de Diciembre de 2022). *Repositorio Institucional UCC*. ELABORACIÓN DE DIAGRAMAS DE CASO DE USO .  
<https://repository.ucc.edu.co/server/api/core/bitstreams/a7ea175f-de13-49fd-afba-c021baf7a6d0/content>
- Martínez, H. C. (2020). *EVALUACIÓN DE UNA HERRAMIENTA DE PRUEBAS END-TO-END PARA MICROSERVICIOS IMPLEMENTADOS EN JAVA Y NODE.JS*. www.kerwa.ucr.ac.cr .  
<https://www.kerwa.ucr.ac.cr/server/api/core/bitstreams/e630eb4a-4a37-47f0-b66a-a11fa1af54ad/content>
- Martins, J. (19 de Enero de 2024). *Asana* . ¿Qué es la metodología Kanban y cómo funciona?.  
<https://asana.com/es/resources/what-is-kanban>
- Mi Carrera Universitaria. (2024). *Ingeniería de software: Qué es, objetivos, características y más*. micarrerauniversitaria.com. <https://micarrerauniversitaria.com/c-ingenieria/ingenieria-de-software/>
- Microsoft . (2024 ). *Microsoft.net* . El lenguaje de programación moderno, innovador y de código abierto para crear todas sus aplicaciones.. <https://dotnet.microsoft.com/es-es/languages/csharp>
- Microsoft. (2024). *Descripción de los conceptos de NAS en Azure NetApp Files*. learn.microsoft.com. <https://learn.microsoft.com/es-es/azure/azure-netapp-files/network-attached-storage-concept>
- Microsoft. (2024). *Diseño de arquitectura de microservicios*. learn.microsoft.com. <https://learn.microsoft.com/es-es/azure/architecture/microservices/>
- Navarro. (Diciembre de 2013). *Redalyc*. Revisión de metodologías ágiles para el desarrollo de software. <https://www.redalyc.org/articulo.oa?id=496250736004>
- NimbleWork. (19 de febrero de 2024). *Nimble humanize Work* . ¿Qué es Kanban?.  
<https://www.nimblework.com/es/kanban/que-es-kanban/>
- Peiró, R. (01 de 10 de 2020). *API*. economipedia.com.  
<https://economipedia.com/definiciones/api.html>
- Poder Judicial. (2024). *INFORMACIÓN INSTITUCIONAL*. pj.poder-judicial.go.cr. <https://pj.poder-judicial.go.cr/index.php/informacion-institucional/historia-organizacion-y-funcionamiento>
- Pressman, R. S. (2010). *Ingeniería del software. Un enfoque práctico*. McGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V.
- Proyectosagiles. (2008). *proyectosagiles.org*. Qué es SCRUM. <https://proyectosagiles.org/que-es-scrum/>
- QAlified . (06 de 06 de 2023). *Comprendiendo el valor de los diferentes tipos de pruebas*. qalified.com. <https://qalified.com/es/blog/tipos-testing/#contacto>
- Red Hat. (08 de 05 de 2020). *¿Qué es una API REST?* www.redhat.com.  
<https://www.redhat.com/en/topics/api/what-is-a-rest-api>
- Reis Digital . (18 de Agosto de 2022). *Reis Digital* . El tablero Kanban: un ejemplo de cómo mejorar tu productividad. <https://reisdigital.es/ejemplos/tablero-kanban-ejemplo/>

- SafetyCulture. (15 de 01 de 2024). *Guía para entender la matriz de riesgo 5x5*. safetyculture.com. <https://safetyculture.com/es/temas/evaluacion-de-riesgos/matriz-de-riesgo/>
- Sampieri, D. R. (2020). *Metodología de la investigación*. Mc Graw Hill education.
- Sanjuán, D. L. (2011). *Facultad de Psicología UNAM*. La observación. [https://www.psicologia.unam.mx/documentos/pdf/publicaciones/La\\_observacion\\_Lidia\\_Diaz\\_Sanjuan\\_Texto\\_Apoyo\\_Didactico\\_Metodo\\_Clinico\\_3\\_Sem.pdf](https://www.psicologia.unam.mx/documentos/pdf/publicaciones/La_observacion_Lidia_Diaz_Sanjuan_Texto_Apoyo_Didactico_Metodo_Clinico_3_Sem.pdf)
- Sommerville, I. (2005). *Ingeniería del software Séptima edición*. Pearson Educación, S.A.
- Sommerville, I. (2015). *Universidad Veracruzana*. Capítulo 5 Requerimientos del software. [https://www.uv.mx/personal/fcastaneda/files/2015/08/F\\_Capitulo\\_5\\_Requerimientos\\_del\\_software.pdf](https://www.uv.mx/personal/fcastaneda/files/2015/08/F_Capitulo_5_Requerimientos_del_software.pdf)
- U San Marcos . (2020). *U San Marcos*. Fuentes de información y tipos de estudio . <https://repositorio.usam.ac.cr/xmlui/bitstream/handle/11506/1267/LEC%20MET%200007%202020.pdf?sequence=1&isAllowed=y>
- UAEH. (2024). *1.2.1 Análisis de Sistemas*. cidcame.uaeh.edu.mx. [http://cidcame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro17/121\\_analisis\\_de\\_sistemas.html](http://cidcame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro17/121_analisis_de_sistemas.html)
- UCR. (2020). *Diseño de Software*. www.ecci.ucr.ac.cr. <https://www.ecci.ucr.ac.cr/cursos/ci-0136>
- Unir FP. (21 de Junio de 2022). *unirfp*. ¿Qué es la programación estructurada?. [https://www.google.com/search?q=el+mes+6+cual+es+&a=gsb\\_tskspj3gunntmegikmoqw20k\\_00\\_00&x1=%7Bparam1%7D](https://www.google.com/search?q=el+mes+6+cual+es+&a=gsb_tskspj3gunntmegikmoqw20k_00_00&x1=%7Bparam1%7D)
- Universidad cenfotec . (23 de Marzo de 2022). *Ucenfotec*. Curso Python Fundamentos . <https://www.santanderopenacademy.com/es/blog/python-que-es.html>
- Universidad de Jaén . (24 de Octubre de 2023). *Universidad de Jaén*. . Enfoque cuantitativo. [https://web.ujaen.es/investiga/tics\\_tfg/enfo\\_cuanti.html](https://web.ujaen.es/investiga/tics_tfg/enfo_cuanti.html)
- Universidad de Málaga . (2007). *Universidad de Málaga*. METODOLOGÍA DE LA PROGRAMACIÓN. [http://www.lcc.uma.es/~program1/apuntes/metodologia/temaiv\\_MP.pdf](http://www.lcc.uma.es/~program1/apuntes/metodologia/temaiv_MP.pdf)
- Universidad de Salamanca . (2007). *audax*. LENGUAJES DE PROGRAMACIÓN . [http://audax.zam.usal.es/web/mlperez/programacion\\_archivos/trabajos\\_2006\\_2007/Tema\\_2\\_GR2.pdf](http://audax.zam.usal.es/web/mlperez/programacion_archivos/trabajos_2006_2007/Tema_2_GR2.pdf)
- Universidad de Valladolid . (2021). *Escuela de ingenierías industriales*. El lenguaje C++. [https://www2.eii.uva.es/fund\\_inf/cpp/temas/1\\_introduccion/introduccion.html](https://www2.eii.uva.es/fund_inf/cpp/temas/1_introduccion/introduccion.html)
- Universidad del Valle . (2001). *GYEPRO*. Facultad de Ciencias de la Administración Manual Resumido de Gestión de Proyectos . <https://gyepro.univalle.edu.co/documentos/gestion/factibilidad.pdf>
- Universidad Europea . (30 de 12 de 2021). *¿Qué es SQL y para qué sirve?* <https://universidadeuropea.com/>. <https://universidadeuropea.com/blog/lenguaje-programacion-sql/>
- Universidad Internacional de La Rioja. (07 de 12 de 2023). *La importancia de las pruebas de software*. mexico.unir.net. <https://mexico.unir.net/noticias/ingenieria/pruebas-software/>

- Universidad Internacional de La Rioja. (26 de 01 de 2024). *¿Qué es la arquitectura de software? Características y tipos*. ecuador.unir.net. <https://ecuador.unir.net/actualidad-unir/arquitectura-software/>
- Universidad Nacional . (2024). *Educación permanente*. Java básico. <https://www.formaciontic.una.ac.cr/index.php/oferta-de-capacitacion/programacion/231-java-basico>
- Universidad Nacional Autónoma de México. (2022). *Prácticas para el desarrollo de software*. www.red-tic.unam.mx. <https://www.red-tic.unam.mx/desarrollo-de-software>
- Universidad Veracruzana . (2014). *Universidad Veracruzana*. Tipos de investigación Inicio. <https://www.uv.mx/apps/bdh/investigacion/unidad1/investigacion-tipos.html>
- Universidad Veracruzana . (Agosto de 2017). *TECNOLOGIAS PARA LA INTEGRACION DE SOLUCIONES. SISTEMAS WEB* . <https://www.uv.mx/personal/ermeneses/files/2017/08/Clase13-PHP.pdf>
- Universidad Veracruzana . (Febrero de 2018). *Universidad Veracruzana* . PROCESOS PARA LA INGENIERIA DE SOFTWARE . <https://www.uv.mx/personal/ermeneses/files/2018/02/Clase8-Modelos de procesos de desarrollo de softwareII.pdf>
- Universitat de Valencia. (2004). *Universitat de València*. TEMA 5: Subprogramas, programación modular. [https://informatica.uv.es/iiguia/AED/oldwww/2004\\_05/AED.Tema.05.pdf](https://informatica.uv.es/iiguia/AED/oldwww/2004_05/AED.Tema.05.pdf)
- Vega, E. (30 de Mayo de 2024). *Medium*. La Investigación de Enfoque Cualitativo. <https://medium.com/@envervega/la-investigaci%C3%B3n-de-enfoque-cualitativo-dd3fe4ea3544>
- Vega, M. (Octubre de 2010). *Universidad de Granada*. Casos de uso. <https://lsi2.ugr.es/~mvega/docis/casos%20de%20uso.pdf>
- Wong, S. (117 de Sandra Wong Durand de 2018). *Repositorio continental*. Análisis y requerimientos de software. [https://repositorio.continental.edu.pe/bitstream/20.500.12394/4281/1/DO\\_FIN\\_103\\_MAI\\_UC0939\\_2018.pdf](https://repositorio.continental.edu.pe/bitstream/20.500.12394/4281/1/DO_FIN_103_MAI_UC0939_2018.pdf)
- X5servers. (2024). *Que son Clusters y Como Funcionan*. x5servers.com. <https://x5servers.com/que-son-clusters-y-como-funcionan/>