



Universidad de las Ciencias Informáticas

Facultad 2

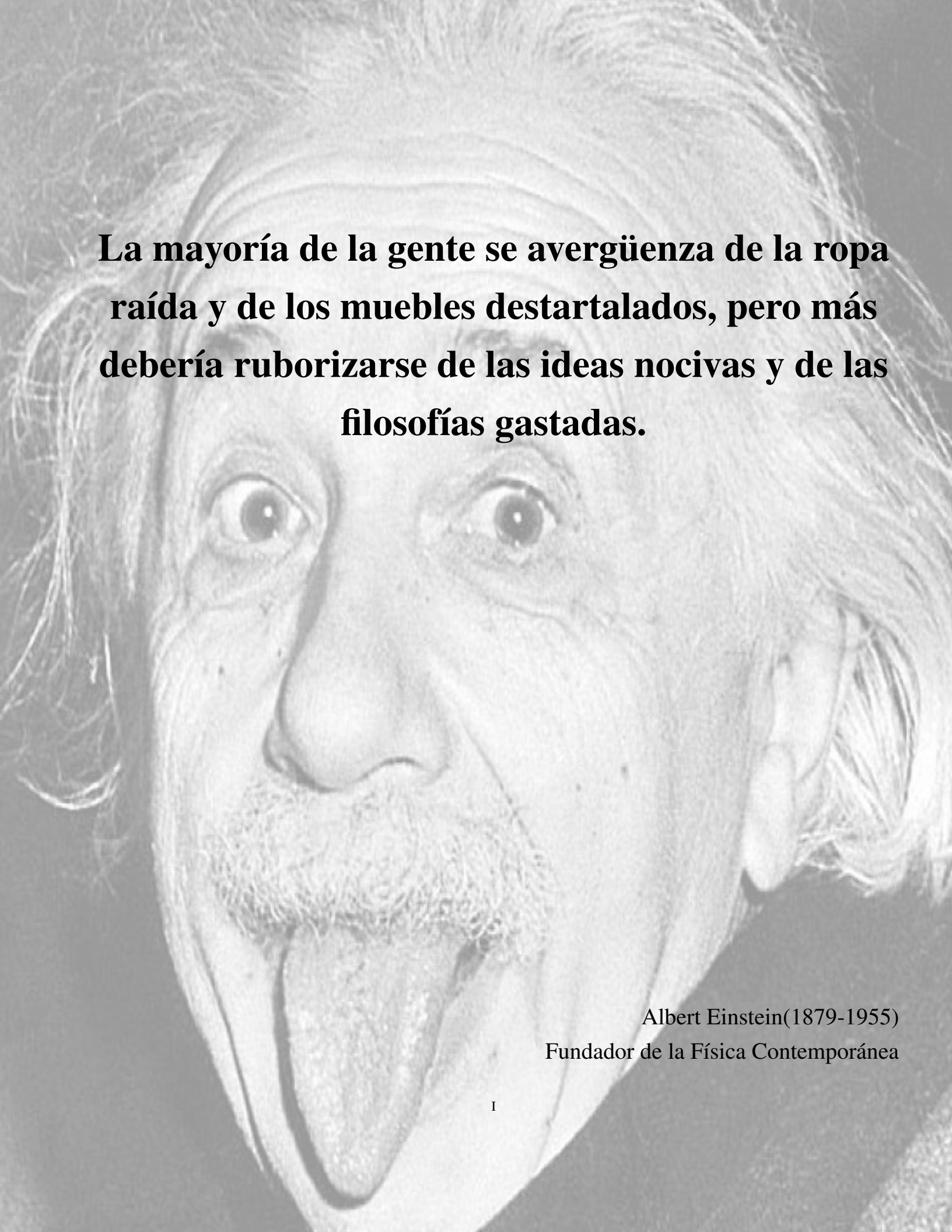
TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS  
INFORMÁTICAS

**Propuesta de un sistema para el respaldo de imágenes médicas en el Sistema de Adquisición,  
Almacenamiento, Transmisión y Visualización de Imágenes Médicas**

**Autores:** Alejandro Seijas Fonseca  
Carlos Asiel Pina Orozco

**Tutor:** MSc. Yasel Couce Sardiñas

**La Habana, 12 de junio de 2015**



**La mayoría de la gente se avergüenza de la ropa raída y de los muebles destartalados, pero más debería ruborizarse de las ideas nocivas y de las filosofías gastadas.**

Albert Einstein(1879-1955)  
Fundador de la Física Contemporánea

## **Datos de contacto**

### **Tutor:**

MSc. Yasel Couce Sardiñas (yaselc@.uci.cu): Graduado de Licenciatura en Ciencias de la Computación en el año 2005 en la Universidad Central de Las Villas. Posee Categoría Docente de profesor Asistente y Categoría Científica de Máster en Ciencias. Ha impartido las asignaturas Sistemas Operativos, Seguridad Informática e Inteligencia Artificial en la Facultad 7 desde el curso 2005 – 2006.

## **Resumen**

El Sistema de Adquisición, Almacenamiento, Transmisión y Visualización de Imágenes Médicas (PACS) basado en la nube o PACS en la nube es un concepto relativamente nuevo en el ámbito de la informática aplicada a la salud, el cual se encuentra total o parcialmente localizado en la nube y es accedido, tanto por los usuarios como por los administradores, a través de Internet. Estos sistemas prometen ofrecer independencia tanto de los dispositivos que se utilicen como del lugar del cual se acceda a ellos gracias a las tecnologías que utilizan en su implementación.

La utilización de la nube como medio de almacenamiento de las imágenes digitales médicas, brinda ventajas en cuanto a costos, escalabilidad, estabilidad e interoperabilidad, lo cual hace del uso de la plataforma de computación en la nube una alternativa factible para proporcionar un nuevo esquema de respaldo para el sistema alas PACS, un Sistema de Adquisición, Almacenamiento, Transmisión y Visualización de Imágenes Médicas desarrollado en el Centro de Informática Médica (CESIM) de la Universidad de las Ciencias Informáticas (UCI) en Cuba.

La presente investigación propone un esquema de respaldo de imágenes digitales médicas basado en servicios en la nube que disminuye el costo del almacenamiento y aumenta la disponibilidad e interoperabilidad de los estudios generados por el sistema alas PACS.

**Palabras clave:** PACS, nube, respaldo, escalabilidad, estabilidad, interoperabilidad, imágenes digitales médicas, alas PACS.

## Índice General

<b>Índice de figuras</b>	<b>IX</b>
<b>Índice de tablas</b>	<b>X</b>
<b>Introducción</b>	<b>1</b>
<b>1. Fundamentación teórica del sistema para el respaldo de imágenes médicas</b>	<b>7</b>
1.1. Respaldo de información . . . . .	7
1.1.1. Respaldo en Disco duro . . . . .	7
1.1.2. Respaldo en Discos espejo de servidores . . . . .	8
1.1.3. Respaldo en línea . . . . .	8
1.2. PACS Internacionales. . . . .	9
1.2.1. La computación en la nube en imágenes médicas (Cloud computing in medical imaging). . . . .	10
1.2.2. Syngo.plaza. . . . .	10
1.2.3. IntelliSpace PACS. . . . .	10
1.3. PACS Nacionales. . . . .	11
1.3.1. alas PACS. . . . .	11
1.4. Computación en la nube (Cloud Computing). . . . .	11
1.5. Tipos de nubes según los servicios que brindan. . . . .	12
1.5.1. Infraestructura como Servicio (IaaS). . . . .	12
1.5.2. Plataforma como Servicio (PaaS). . . . .	12
1.5.3. Software como Servicio (SaaS). . . . .	12

1.6.	Tipos de Nubes según su infraestructura. . . . .	12
1.6.1.	Nube Privada. . . . .	12
1.6.2.	Nube Pública. . . . .	13
1.6.3.	Nube Híbrida. . . . .	13
1.6.4.	Nube Comunitaria. . . . .	13
1.7.	Proveedores de servicio de almacenamiento en la nube. . . . .	13
1.7.1.	Dropbox. . . . .	13
1.7.2.	Google Drive. . . . .	13
1.7.3.	SkyDrive. . . . .	14
1.7.4.	Amazon . . . . .	14
1.7.5.	Comparación entre Dropbox, Google Drive, SkyDrive y Amazon. . . . .	14
1.8.	Selección de la nube, proveedor y servicio a utilizar . . . . .	16
1.9.	Metodología de desarrollo de software . . . . .	16
1.9.1.	Programación Extrema (XP) . . . . .	16
1.10.	Lenguaje de modelado . . . . .	17
1.10.1.	Lenguaje Unificado de Modelado (UML) . . . . .	17
1.11.	Herramienta de Modelado . . . . .	17
1.11.1.	Visual Paradigm . . . . .	17
1.12.	Framework . . . . .	17
1.12.1.	Django . . . . .	17
1.13.	Bibliotecas gráficas . . . . .	18
1.13.1.	JQuery . . . . .	18
1.13.2.	HighCharts . . . . .	18
1.14.	Lenguajes de Programación . . . . .	18
1.14.1.	Python . . . . .	18
1.14.2.	Lenguaje de Marcado de Hipertexto (HTML) . . . . .	19
1.14.3.	CCS . . . . .	19
1.14.4.	JavaScript . . . . .	19

1.15. Entorno de Desarrollo . . . . .	20
1.15.1. PyCharm . . . . .	20
1.16. Sistema Gestor de Base de Datos . . . . .	20
1.16.1. MySQL . . . . .	20
1.17. Planificación de tareas . . . . .	21
1.17.1. Celery . . . . .	21
1.18. Interacción con Amazon Glacier . . . . .	21
1.18.1. Boto . . . . .	21
1.19. Conclusiones del capítulo . . . . .	21
<b>2. Características del sistema para el respaldo de imágenes médicas</b>	<b>23</b>
2.1. Sistema propuesto . . . . .	23
2.2. Escenario de funcionamiento del sistema . . . . .	23
2.3. Monitoreo de disco . . . . .	24
2.3.1. Espacio en disco . . . . .	25
2.3.2. Cambios en el árbol de directorios . . . . .	25
2.4. Proceso de respaldo y restauración de un estudio . . . . .	26
2.4.1. Respaldo de un estudio . . . . .	26
2.4.2. Restauración de un estudio . . . . .	27
2.5. Modelo del dominio . . . . .	27
2.5.1. Conceptos fundamentales . . . . .	28
2.5.2. Diagrama del modelo de dominio . . . . .	28
2.6. Especificación de los requisitos del software . . . . .	29
2.6.1. Requisitos funcionales . . . . .	29
2.6.2. Requisitos no funcionales . . . . .	31
2.7. Definición de los Casos de Uso del Sistema . . . . .	32
2.7.1. Actores del sistema . . . . .	33
2.7.2. Diagrama de casos de uso del sistema . . . . .	34

2.8.	Arquitectura . . . . .	34
2.8.1.	Estilo arquitectónico (Cliente-Servidor) . . . . .	34
2.8.2.	Patrón arquitectónico (Modelo Vista Controlador) . . . . .	35
2.9.	Análisis . . . . .	36
2.10.	Diseño . . . . .	37
2.10.1.	Modelo de Datos . . . . .	37
2.11.	Conclusiones del capítulo . . . . .	38
<b>3.</b>	<b>Implementación y prueba del sistema para el respaldo de imágenes médicas</b>	<b>39</b>
3.1.	Diagrama de componentes . . . . .	39
3.2.	Diagrama de despliegue . . . . .	41
3.3.	Prueba del sistema . . . . .	41
3.3.1.	Niveles de prueba . . . . .	41
3.3.2.	Métodos de Pruebas . . . . .	42
3.3.3.	Diseño de casos de prueba . . . . .	42
3.3.4.	Pruebas de Seguridad . . . . .	43
3.3.5.	Pruebas de Carga . . . . .	44
3.4.	Validación de las variables de la investigación . . . . .	45
3.4.1.	Aumento de la confiabilidad . . . . .	45
3.5.	Funcionalidades extras . . . . .	46
3.6.	Análisis económico . . . . .	47
3.6.1.	Información generada periodo cinco años . . . . .	47
3.6.2.	Costo de almacenar la información del esquema actual. . . . .	48
3.6.3.	Costo de almacenar la información con el SRIM. . . . .	49
3.6.4.	Costo de almacenar la información. . . . .	50
3.7.	Aporte . . . . .	52
3.8.	Conclusiones del capítulo . . . . .	52



<b>Conclusiones</b>	<b>54</b>
<b>Recomendaciones</b>	<b>55</b>
<b>Referencias bibliográficas</b>	<b>56</b>
<b>A. Primer apéndice</b>	<b>59</b>
A.1. Entrevista . . . . .	59
A.2. Características de Amazon . . . . .	60
A.2.1. Precios de almacenamiento . . . . .	60
A.2.2. Precios de las solicitudes . . . . .	60
A.2.3. Precios de transferencia de datos . . . . .	61
A.3. Características de Dropbox, Google Drive y SkyDrive . . . . .	62

## Índice de figuras

1.	Componentes principales de un PACS . . . . .	2
2.1.	Escenario de funcionamiento de la aplicación . . . . .	24
2.2.	Estructura del árbol de directorio . . . . .	26
2.3.	Diagrama del modelo de dominio . . . . .	29
2.4.	Diagrama de casos de uso del sistema . . . . .	34
2.5.	Flujo del estilo arquitectónico Cliente-servidor . . . . .	35
2.6.	Flujo del patrón MVC . . . . .	36
2.7.	Modelo de Datos . . . . .	37
3.1.	Diagrama de componentes . . . . .	40
3.2.	Diagrama de despliegue . . . . .	41
3.3.	Liberar espacio . . . . .	47
3.4.	Información generada en cinco años . . . . .	48
3.5.	Costo de almacenar la información del esquema actual . . . . .	49
3.6.	Costo de almacenar la información con el SRIM . . . . .	50
3.7.	Costo de almacenar la información . . . . .	51

## Índice de tablas

1.1. Comparación de proveedores de almacenamiento en la nube. . . . .	16
2.1. Actores del Sistema. . . . .	33
3.1. Plantilla casos de prueba. . . . .	43
3.2. Registro de tiempos de respuesta . . . . .	45
3.3. Validación de confiabilidad. . . . .	46
A.1. Precios de almacenamiento. . . . .	60
A.2. Precios de las solicitudes. . . . .	61
A.3. Precios de transferencia de datos. . . . .	61
A.4. Características de Dropbox, Google Drive y SkyDrive. . . . .	65

## **Introducción**

Durante los últimos años, las Tecnologías de la Información y las Comunicaciones (TIC), han alcanzado un desarrollo incuestionable aplicado a diferentes esferas de la sociedad, incluido la salud. El uso de la informática en la medicina ha producido grandes avances en los métodos de diagnóstico y tratamiento de las enfermedades, debido a la introducción de nuevas técnicas y estudios menos invasivos como las imágenes digitales médicas [1], que ganaron gran aceptación y adquieren cada vez un mayor desarrollo y aplicación. El considerable desarrollo de estos métodos, junto a la gran diversidad de equipos de adquisición de imágenes fabricados, trajo consigo el surgimiento de los Sistemas de Adquisición, Almacenamiento, Transmisión y Visualización de Imágenes Médicas (por sus siglas en inglés PACS) [2].

Los PACS, según [3], constituyen almacenes lógicos de imágenes las cuales pueden ser recuperadas desde programas habilitados para tal fin, según nuestra necesidad, ya sea de forma inmediata para estudios actuales o de forma retardada para estudios almacenados en dispositivos de almacenamiento secundario. El objetivo principal de un PACS es permitir el funcionamiento del servicio de imágenes sin la necesidad de la impresión de placas radiológicas [4], ni de papeles para la información clínica asociada a las imágenes, supliendo los altos costos en tiempo y dinero que esto significa.

Estos sistemas permiten que los especialistas no tengan que estar presentes en la captura de imágenes y estas puedan ser vistas desde cualquier estación de trabajo del hospital para el diagnóstico del paciente, ya sea individual o simultáneamente, gracias a que estas imágenes se almacenan en un servidor central y están disponibles en todo momento.

Los sistemas PACS, están conformados por cuatro componentes principales [Figura 1], los cuales tienen la capacidad de interactuar entre sí y con otros sistemas [5].

**Sistema de adquisición:** Compuesto por los equipos encargados de la obtención de las imágenes médicas.

**Sistema de almacenamiento:** Encargado de la persistencia de la información del paciente, de los estudios y otros artefactos similares que se generen durante el ciclo de vida del paciente en la institución.

**Sistema de visualización:** Lo componen las estaciones de trabajo para visualización de los estudios y la confección de los reportes asociados a estos.

**Infraestructura de red:** Son las redes y otros dispositivos en las estaciones, que permiten la comunicación entre todas las partes que conforman el PACS.

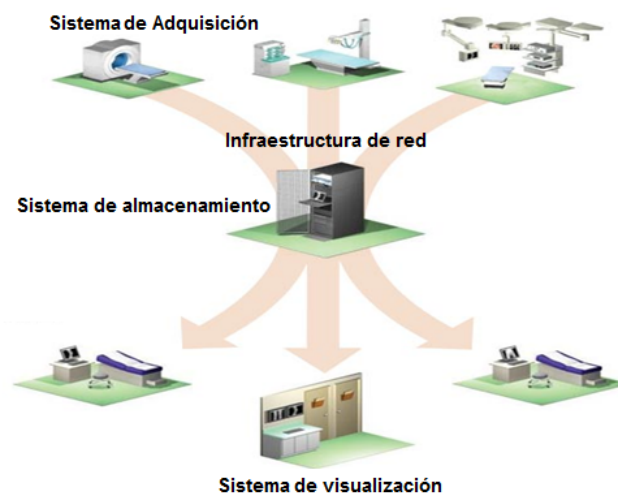


Figura 1: Componentes principales de un PACS

De los componentes, el Sistema de Almacenamiento es considerado como el más crítico, debido a que en él se encuentra toda la información referente a los estudios de los pacientes, información que aumenta considerablemente con el objetivo de proporcionar a los especialistas en medicina, imágenes de mayor calidad y resolución y al mismo tiempo se debe proteger pues constituye parte del patrimonio de la institución de salud y base para futuros estudios.

El sistema alas PACS [6] es una solución que forma parte de esta gran familia de Sistemas de Adquisición, Almacenamiento, Transmisión y Visualización de Imágenes Médicas, ha sido desarrollado en el Centro de Informática Médica (CESIM) de la Universidad de las Ciencias Informáticas (UCI) en Cuba. En la actualidad se encuentra instalado en diversos hospitales de La Habana y de Venezuela.

El sistema alas PACS cuenta con las siguientes características:

- Las imágenes se almacenan en un servidor local, ubicado en cada institución de salud.
- El almacenamiento de los estudios realizados por cada institución de salud, se realiza en primera instancia en el propio servidor, y en caso de que se sobrepase la capacidad del mismo, se realiza la copia de los estudios en soporte DVD, para su posterior almacenamiento por un período de cinco años.
- Cuenta con el sistema alas PACS DICOM Mail, el cual permite la transmisión de imágenes médicas de una forma personalizada mediante una interfaz similar a la de los correos electrónicos, permitiendo a los médicos y especialistas el envío de estudios, series e imágenes médicas según lo establecido por el estándar internacional para el manejo de imágenes médicas DICOM en su versión 3.0

En los ambientes donde se encuentra desplegado el sistema, se identifican las siguientes características <sup>1</sup>:

- El servidor de imágenes de cada institución de salud ronda una capacidad de almacenamiento de cinco a veinte terabytes.
- Las instituciones de salud cuentan con una media de cuatro equipos de obtención de las imágenes médicas.
- La información generada por un estudio a un paciente tiene una media de trescientos megabytes.
- En un día de trabajo se pueden realizar treinta estudios por equipo.
- Velocidad de conexión a Internet de un gigabit por segundo.
- Generan anualmente un aproximado de 2.9 a 12.5 terabytes de información.
- Saturan la capacidad de almacenamiento del servidor en un período de 0.4 a dos años

---

<sup>1</sup>Estos datos son recogidos a partir de la entrevista [A.1] realizada a profesionales que brindan el soporte al alas PACS en hospitales de Venezuela

Por las características de este esquema se evidencian deficiencias como:

- Baja interoperabilidad entre los hospitales y demás instituciones de salud, debido a la información referente a los pacientes se encuentra almacenada únicamente en la institución que se realizó el estudio y su acceso por otras instituciones debe realizarse a través del alas PACS DICOM Mail o del soporte DVD, trayendo consigo que las consultas de segundas opiniones fuera de la institución dependan de un especialista que envíe los respectivos estudios, además la información referente a los pacientes es de difícil acceso para otras instituciones.
- Incremento del costo de almacenamiento de la información en el servidor de imágenes de la institución de salud, debido al aumento en cantidad y calidad de los estudios realizados y por ende de la infraestructura requerida para soportar este crecimiento.
- Elevado costo de un mecanismo de recuperación ante desastres tales como la rotura o avería de las unidades de almacenamiento o del propio servidor de imágenes, además se puede incurrir en la pérdida parcial o total de la información, ya que esta no está respaldada en tiempo real o bajo períodos muy prolongados.
- Baja o limitada disponibilidad de los estudios realizados, la cual está en dependencia de la capacidad de almacenamiento del servidor local.
- El esquema de respaldo a utilizar recae completamente en la responsabilidad de la institución, donde depende si esta cuenta con personal especializado o no.
- La restauración del sistema se realiza de forma manual, copiando uno a uno los DVD donde se encuentra respaldada la información.

Por lo antes expuesto se identifica como **problema a resolver**: la baja confiabilidad del esquema de persistencia de imágenes digitales médicas en los servidores del Sistema de Adquisición, Almacenamiento, Transmisión y Visualización de Imágenes Médicas.

Se ha definido el **objeto de estudio**: esquemas de respaldo de la información; y el **campo de acción**: esquemas de respaldo de imágenes médicas.

El **objetivo general** de la investigación es desarrollar un sistema para el respaldo de imágenes médicas basado en servicios en la nube que permita aumentar la confiabilidad del esquema de persistencia en el Sistema de Adquisición, Almacenamiento, Transmisión y Visualización de Imágenes Médicas.

### **Objetivos específicos:**

1. Caracterizar el sistema de respaldo de los Sistemas de Adquisición, Almacenamiento, Transmisión y Visualización de Imágenes Médicas a partir de la bibliografía especializada en los mismos.
2. Analizar y diseñar un sistema para el respaldo de imágenes médicas en el alas PACS.
3. Implementar el sistema definido.
4. Realizar las pruebas correspondientes al sistema implementado.

### **Métodos Teóricos:**

**Analítico – sintético:** utilizado para procesar la información de las características del sistema alas PACS y de los ambientes donde se encuentra desplegado, a través de la entrevista [A.1] realizada.

**Inductivo – deductivo:** utilizado en la aplicación de casos de pruebas al sistema, con el objetivo de obtener conocimientos generalizadores para arribar a conclusiones a partir de las respuestas proporcionadas por este.

**Análisis histórico – lógico:** utilizado para conocer los antecedentes y la actualidad de los sistemas PACS.

**Modelación:** utilizado para elaborar diferentes diagramas que brindaron información clara sobre el tema de estudio.



**Métodos Empíricos:**

**Observación:** utilizado para conocer el comportamiento del sistema alas PACS en los ambientes donde se encuentra desplegado.

**Entrevista (no estructurada):** utilizada para obtener conocimiento acerca del sistema alas PACS y los ambientes donde se encuentra desplegado.

El desarrollo de este trabajo está organizado en tres capítulos, que a continuación se describen:

**CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA:** En este capítulo se realiza un análisis de los sistemas de respaldo de los Sistemas de Adquisición, Almacenamiento, Transmisión y Visualización de Imágenes Médicas existentes a nivel nacional e internacional, haciendo una valoración de las principales características que componen un esquema de respaldo. Además se describen las técnicas, tecnologías, metodología y los lenguajes de programación a tener en cuenta para modelar e implementar el sistema.

**CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA:** Se describe el proceso de negocio que tiene lugar en el respaldo de la información de los PACS. Se concretan las características del sistema. Se definen y se describen además los actores y trabajadores del negocio, los casos de uso del negocio y la descripción de los mismos. Los Requerimientos funcionales y no funcionales del sistema también están explícitos, así como la definición de los actores y casos de uso del sistema, con la especificación de los más significativos.

**CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA:** Muestra el despliegue y los componentes utilizados a través de diagramas. Se definirán los tipos de pruebas, los casos de prueba y los resultados de estas que permiten evaluar los resultados obtenidos, así como la validación de las variables de la investigación, el análisis económico de la solución existente con el sistema propuesto y el aporte del mismo.

# Capítulo 1

## Fundamentación teórica del sistema para el respaldo de imágenes médicas

El presente capítulo tiene como objetivo abordar los diferentes elementos que brindan la base teórico metodológica para el desarrollo de sistema para el respaldo de imágenes digitales médicas, además se realiza un análisis de algunos sistemas existentes en el mundo relacionados con la investigación en curso.

### 1.1. Respaldo de información

El respaldo de información [7] es la copia de los datos importantes de un dispositivo primario en uno ó varios dispositivos secundarios, ello para que en caso de que el primer dispositivo sufra una avería electromecánica ó un error en su estructura lógica, sea posible contar con la mayor parte de la información necesaria para continuar con las actividades rutinarias y evitar pérdida generalizada de datos.

La importancia radica en que todos los dispositivos de almacenamiento masivo de información tienen la posibilidad de fallar, por lo tanto es necesario que se cuente con una copia de seguridad de la información importante, ya que la probabilidad de que dos dispositivos fallen de manera simultánea es muy difícil.

#### 1.1.1. Respaldo en Disco duro

Los discos duros actualmente tienen varios terabytes y el tiempo medio de acceso es muy bajo (menos de veinte milisegundos).

Ventajas:

- Alta capacidad de almacenamiento a muy bajo costo en comparación con otras formas de respaldo (memorias).
- Rápido acceso a los datos con respecto a discos ópticos (CDs, DVDs, etc).
- La transferencia de información es rápida y muy eficiente.
- En condiciones normales de funcionamiento puede durar muchos años.

Desventajas:

- En ciertas condiciones anormales como exceso de calor, movimiento o golpes, la información almacenada en un disco duro puede alterarse o perderse.
- Consumen más energía eléctrica que otros medios de almacenamiento, dado que tienen que mantener los discos internos girando miles de veces por minuto.

### **1.1.2. Respaldo en Discos espejo de servidores**

Se trata de discos duros que se van autocopiando mientras se trabaja de manera normal, conforme el disco duro principal de una computadora va modificando su información, una computadora anexa va clonando las acciones de este. Esto se logra mediante una aplicación especial instalada en ambas computadoras, así si el principal falla, se autoactiva el otro como primario mientras se resuelve la avería del sistema.

### **1.1.3. Respaldo en línea**

También llamado respaldo remoto o almacenamiento en la nube, los sitios de almacenamiento en línea permiten almacenar información en los servidores de una compañía. Amazon, Dropbox, SkyDrive y muchos otros sitios proporcionan estos servicios de forma gratuita; por lo general, para obtener más espacio de almacenamiento se tiene que pagar una tarifa [8, 9].

Ventajas:

- Acceso desde cualquier parte, a la información que se almacena.

- El uso de estos generalmente se da como una copia de seguridad a archivos para que se mantengan a salvo de cualquier situación.
- Se ahorra espacio.
- Basta un navegador web o una aplicación específica que nos permita acceder a los servidores donde se encuentran los datos.
- Ahorro de recursos en el ordenador, ya que se utilizan los de los servidores (almacenamiento, memoria, CPU, software, etc.).
- No hay que preocuparse por actualizaciones de software o hardware.
- No hay problemas de capacidad de almacenamiento o pérdida de información: capacidad ilimitada y seguridad total.

#### Desventajas:

- El espacio de almacenamiento (gratuito) es limitado, por lo que normalmente hay que pagar una tarifa si necesitamos más espacio.
- Privacidad, debido a que se pone la información (sensible en muchos casos), en servidores fuera de la organización, dejando como responsable de los datos al proveedor de servicio.
- Dependencia, ya que el cliente se vuelve dependiente no sólo del proveedor del servicio, sino también de su conexión a Internet, debido a que el usuario debe estar permanentemente conectado para poder alcanzar al sistema que se encuentra en la nube.

## **1.2. PACS Internacionales.**

A continuación se describen dos de los principales Sistemas de Adquisición, Almacenamiento, Transmisión y Visualización de Imágenes Médicas, pertenecientes a los proveedores de equipos imagenológicos líderes a nivel internacional, basados en la comparación realizada por [10], de cuarenta PACS internacionales teniendo en cuenta treinta variables.

### **1.2.1. La computación en la nube en imágenes médicas (Cloud computing in medical imaging).**

Compuesto por tres componentes principales: una aplicación de visualización de imágenes, un motor de flujo de trabajo, y un sistema de archivamiento de imágenes [11].

El archivo de imágenes basado en la nube brinda varias oportunidades de mejora. La oportunidad más provechosa es la agregación de registros de imágenes dentro y entre las organizaciones de salud que abarcan varios lugares.

Su esquema de persistencia se basa en el respaldo en línea, lo que garantiza existencia de estudios previos para la realización de comparaciones, incluso si fueron hechas en otro lugar; estudios innecesarios, junto con la dosis de radiación innecesaria pueden reducirse ya que las imágenes de los estudios de otras ubicaciones se encuentran disponibles; el uso de CD/DVD puede ser reducido o eliminado, lo que a su vez reduce el coste y permite una atención más oportuna; se pueden compartir imágenes para referencias, segundas opiniones y consultas de subespecialidad.

### **1.2.2. Syngo.plaza.**

Syngo es un software para la salud desarrollado por la división Siemens Healthcare [12]. Cuenta con una aplicación PACS ágil, syngo.plaza, la cual es modular y escalable, lo que ofrece posibilidades de integración para su futuro crecimiento.

Su esquema de persistencia se basa en el almacenamiento de conexión directa (DAS) y red de área de almacenamiento (SAN) para el almacenamiento a corto plazo y almacenamiento conectado a red (NAS) para almacenamiento a largo plazo, independiente del proveedor y recuperación ante desastres fuera del sitio.

### **1.2.3. IntelliSpace PACS.**

IntelliSpace PACS es un software de almacenamiento y transmisión de imágenes médicas desarrollado por la división Philips Healthcare [13].

Su esquema de persistencia se basa en el almacenamiento de conexión directa (DAS) y almacenamiento conectado a red (NAS) con datos siempre en línea; las imágenes son almacenadas localmente con copia de seguridad en el centro de recuperación de desastres fuera del sitio y el archivo a largo plazo, NAS basado en conjunto redundante de discos independientes (RAID) distribuido con recuperación ante desastres

### **1.3. PACS Nacionales.**

#### **1.3.1. alas PACS.**

Basado en cuatro componentes: alas PACS Viewer, alas PACS Server, alas PACS Worklist y alas PACS Report [6].

alas PACS Server (Servidor de imágenes médicas) tiene como función recibir las imágenes de los equipos médicos y permitir que sean recuperadas desde las Estaciones de Trabajo. El almacenamiento de los estudios realizados por cada institución de salud, se realiza en primera instancia en el propio servidor, y en caso de que se sobrepase la capacidad del mismo, se realiza la copia de los estudios en soporte DVD, para su posterior almacenamiento por un período de cinco años.

El respaldo de la información se realiza a través de la copia de los estudios en soporte DVD, incurriendo en un gasto adicional en cuanto a la compra de estos DVDs y de la infraestructura necesaria para su posterior almacenamiento y mantenimiento.

### **1.4. Computación en la nube (Cloud Computing).**

La computación en la nube, según [14], es un modelo que permite el acceso bajo demanda a través de la Red a un conjunto compartido de recursos de computación configurables (redes, servidores, almacenamiento, aplicaciones y servicios) que se pueden aprovisionar rápidamente con el mínimo esfuerzo de gestión o interacción del proveedor del servicio.

## **1.5. Tipos de nubes según los servicios que brindan.**

La nube ha sido categorizada en tres grupos principales de acuerdo al tipo de servicios que brindan: Infraestructura como Servicio (IaaS), Plataforma como Servicio (PaaS) y Software como Servicio (SaaS) [15].

### **1.5.1. Infraestructura como Servicio (IaaS).**

Tiene como objetivo ofrecerle al usuario recursos como capacidad de procesamiento, de almacenamiento o comunicaciones, para que los utilice en la ejecución de cualquier software; desde sistemas operativos hasta aplicaciones.

### **1.5.2. Plataforma como Servicio (PaaS).**

Tiene como objetivo permitirle al usuario desplegar sus propias aplicaciones en la infraestructura de la nube del proveedor, que ofrece la plataforma de desarrollo y las herramientas de programación.

### **1.5.3. Software como Servicio (SaaS).**

Tiene como objetivo ofrecerle al usuario la capacidad de que las aplicaciones suministradas se desenvuelvan en una infraestructura de la nube, siendo las aplicaciones accesibles a través de un navegador web, como en el correo electrónico Web.

## **1.6. Tipos de Nubes según su infraestructura.**

De acuerdo la infraestructura de la nube, se encuentra agrupada en cuatro grupos principales: Nube Privada, Nube Pública, Nube Híbrida y Nube Comunitaria [15].

### **1.6.1. Nube Privada.**

Los servicios no son ofrecidos al público en general. La infraestructura es íntegramente gestionada por una organización.

### **1.6.2. Nube Pública.**

La infraestructura es operada por un proveedor que ofrece servicios al público en general.

### **1.6.3. Nube Híbrida.**

Resultado de la combinación de dos o más nubes individuales que pueden ser privadas, compartidas o públicas. Permite enviar datos o aplicaciones entre ellas.

### **1.6.4. Nube Comunitaria.**

Ha sido organizada para servir a una función o propósito común. Es preciso compartir objetivos comunes (misión, políticas, seguridad). Puede ser administrada bien por las organizaciones constituyentes, bien por terceras partes.

## **1.7. Proveedores de servicio de almacenamiento en la nube.**

Se tomó una muestra de los proveedores más populares, estables y utilizados, cuáles brindan el enfoque más rentable y las mejores opciones de precio-capacidad.

### **1.7.1. Dropbox.**

Es muy fácil de usar y permite sincronizar archivos, compartir carpetas y documentos, subir las fotos del móvil automáticamente y consultar un histórico con los cambio [16].

### **1.7.2. Google Drive.**

Totalmente integrado con los servicios: Gmail, Google+, Google Docs y por supuesto, con el sistema operativo móvil Android. También permite sincronizar carpetas, subir fotos del móvil a Google+, compartir y editar documentos entre varios usuarios. Es bastante fácil de usar y compatible con un gran número de dispositivos y aplicaciones [16].



### 1.7.3. SkyDrive.





Similar al Google Drive, pero integrado a Microsoft. Asociada a los servicios de Windows Live (como tu cuenta de Hotmail o Skype), está perfectamente integrado con los servicios de Microsoft (como Office o Windows 7-8, tanto en su versión móvil como de sobremesa), ofrece las opciones de sincronizar carpetas, subir fotos desde el móvil y editar documentos [16].

### 1.7.4. Amazon

El Servicio de Almacenamiento de Amazon provee de un ilimitado volumen de datos (objetos) a almacenar, más de 5000 TB, el tamaño de estos objetos puede oscilar entre un byte y cinco gigabytes, cuenta con servicios especializados (Estándar y Glacier) con diferentes precios de almacenamiento, solicitudes y transferencias realizadas, [Anexo A.2], según las especificidades del producto a desarrollar [17].

### 1.7.5. Comparación entre Dropbox, Google Drive, SkyDrive y Amazon.

La capacidad máxima de almacenamiento que brindan Dropbox, Google Drive y SkyDrive es de quinientos gigabytes, dieciséis terabytes y ciento siete gigabytes respectivamente, según sus planes de mejora, además están enfocados en prestar un servicio de almacenamiento general, que incluye aplicaciones para la visualización y edición de los archivos y seguridad basada en un código de contraseña [Anexo A.3], en contraposición, considerando las opciones de seguridad, protección de datos, la capacidad prácticamente ilimitada de almacenamiento, los diferentes servicios de almacenamiento, así como los precios que Amazon como proveedor brinda, se evidencia la superioridad y ventajas que supone establecer Amazon Glacier como el servicio de almacenamiento en la nube a emplear en nuestra solución, la tabla 1.1 muestra un resumen de dicha comparación en forma tabular.

Proveedores	 Dropbox	 Google Drive	 SkyDrive	 Amazon
Continúa en la próxima página				

<p>Precios de Mejora</p>	<p><b>100GB</b> a \$9.99/mes o \$99/año</p> <p><b>500GB</b> a \$49.99/mes o \$499/año</p>	<p><b>25GB</b> \$2.49</p> <p><b>16TB</b> \$799.99</p>	<p><b>27GB</b> \$10/año</p> <p><b>107GB</b> \$50/ año</p>	<p><b>1TB/Mes</b> \$0.010/GB</p> <p><b>Más 5000TB/Mes</b> \$0.010/GB</p>
<p>Características</p>	<p>Sincronización selectiva de carpetas, historial de versiones, enlace para compartir.</p>	<p>Sincronización selectiva de carpetas, historial de versiones, editor de documentos en línea.</p>	<p>Historial de versiones, aplicaciones Web de Microsoft Office (Word, PowerPoint, Excel, OneNote), edición simultánea de documentos.</p>	<p>Infraestructura de almacenamiento diseñada para realizar copias de seguridad, los datos se almacenan en “archivos” de un tamaño máximo de cuarenta terabytes, se establecen políticas de acceso para cada almacén y usuario, los datos almacenados en Glacier son inmutables, además de contar con un historial de versiones.</p>
<p>Continúa en la próxima página</p>				

Seguridad	Código de Contraseña de cuatro dígitos.	Código de Contraseña.	Código de verificación.	Usuario y contraseña brindados por Amazon, además de la seguridad que el cliente implemente a través del API de desarrollo de Amazon.
-----------	---	-----------------------	-------------------------	---

Tabla 1.1: Comparación de proveedores de almacenamiento en la nube.

## **1.8. Selección de la nube, proveedor y servicio a utilizar**

Teniendo en cuenta las opciones de seguridad, protección de datos, la capacidad prácticamente ilimitada de almacenamiento, el API de desarrollo, la abundante bibliografía así como los precios provistos por Amazon, se decide emplear la infraestructura como servicio (IaaS) provista por la nube pública de Amazon y el servicio Glacier como servicio de respaldo a emplear en la aplicación.

## **1.9. Metodología de desarrollo de software**

### **1.9.1. Programación Extrema (XP)**

La metodología Programación Extrema (XP) se basa en la simplicidad, la comunicación y la reutilización del código, siendo así una metodología ágil de desarrollo de software. Como requisito para alcanzar el éxito del proyecto se incorpora al usuario final como parte del equipo. Es una metodología con reconocido éxito y se utiliza en proyectos con entregas a cortos plazos. XP se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y fácil adaptabilidad ante los cambios [18].

## **1.10. Lenguaje de modelado**

### **1.10.1. Lenguaje Unificado de Modelado (UML)**

UML es un lenguaje capaz de abstraer cualquier tipo de sistema mediante la utilización de diagramas, permite realizar presentaciones gráficas que contienen la información relevante del sistema. El modelado visual se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimientos sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener y controlar la información sobre tales sistemas [19].

## **1.11. Herramienta de Modelado**

### **1.11.1. Visual Paradigm**

Visual Paradigm es una herramienta de modelado que utiliza el Lenguaje Unificado de Modelado (UML), permite la generación de código e ingeniería inversa. Con una clase de diseño bien especificada, Visual Paradigm puede generar código hasta en quince lenguajes de programación, es una herramienta multiplataforma, tiene una interfaz intuitiva y es de fácil aprendizaje para los desarrolladores. Permite la generación automática de diagramas a partir de descripciones de casos de uso, por ejemplo, diagramas de secuencia, permitiendo la agilidad en el trabajo del analista. Permite además hacer descripción de los casos de uso dando una gran variedad de plantillas predeterminadas permitiendo personalizarlas. Con Visual Paradigm los analistas pueden generar la documentación necesaria de los artefactos obtenidos hasta el momento en el proyecto [20].

## **1.12. Framework**

### **1.12.1. Django**

Django es un framework de alto nivel basado en Python [21] que facilita el desarrollo de aplicaciones web dinámicas. Es un framework que abstrae a los programadores de los problemas comunes del desarrollo web y acelera las tareas más frecuentes en la programación. Permite mostrar y validar formularios de manera muy

simple, manipulando el código del formulario y adaptándolo a las necesidades de la aplicación. El mismo convierte los datos enviados por los usuarios, en estructuras de datos que pueden ser manipuladas fácilmente. A través de plantillas ayuda a separar el contenido de la presentación evitando tener que manipular la lógica de negocio cuando se necesite realizar cambios de apariencia en la página. Permite lidiar con trescientas peticiones web por segundo. Su Mapeo Relacional de Objeto (ORM) es muy poderoso, permitiendo definir los modelos de datos enteramente en Python. Además provee un soporte para el uso de plantillas, filtros, etiquetas y formularios [22, 23].

## **1.13. Bibliotecas gráficas**

### **1.13.1. JQuery**

Es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol del Modelo de Objetos del Documento (DOM), manejar eventos, desarrollar animaciones y agregar interacción con JavaScript y XML Asíncronicos (AJAX) a páginas web [24].

### **1.13.2. HighCharts**

Permite añadir gráficas animadas a un sitio web en una gran variedad de formatos (área, barra, columna, pastel, entre otras). Está implementado en Javascript por lo que es sencillo de incorporar a un sitio web. Es compatible con cualquier navegador que soporte Javascript, por lo que funciona en Chrome, Firefox, Safari, Internet Explorer ya sea con Linux, Windows, OSX, Android, entre otros [25].

## **1.14. Lenguajes de Programación**

### **1.14.1. Python**

Python es un lenguaje de programación de alto nivel su filosofía se basa en una sintaxis muy limpia y que favorezca un código legible. Es un lenguaje que soporta orientación a objetos y herencia así como capacidad

de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, SQLite, ODBC, entre otras, programación imperativa y en su menor medida programación funcional, es también multiplataforma, es lenguaje interpretado y no es fuertemente tipificado. Puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux [21].

#### **1.14.2. Lenguaje de Marcado de Hipertexto (HTML)**

Lenguaje de Marcado de Hipertexto (HTML), creado en 1989 por Tim Berners-Lee. Especializado en la descripción de documentos en pantalla. El proyecto inicial se basaba en una colección de etiquetas que permitían describir documentos de texto y vínculos de hipertexto que hacían posible el desplazamiento en forma jerárquica entre diferentes documentos. La facilidad de su uso y la particularidad de no ser propiedad de nadie, hizo de HTML el sistema idóneo para compartir información a través de Internet. Inicialmente su intención era que las etiquetas fueran capaces de marcar la información de acuerdo a su significado, pero por diversos motivos los creadores de los navegadores Web fueron añadiendo más etiquetas HTML, dirigidas a controlar la representación de la información contenida en el documento [26].

#### **1.14.3. CCS**

Hojas de Estilo en Cascada (CSS), es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla. Es el lenguaje que permite adicionar estilos (fuentes, colores, espaciados) a los documentos web [27].

#### **1.14.4. JavaScript**

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

Además posibilita la creación de efectos llamativos en las páginas web e interacción con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estas acciones, de modo que el mayor recurso con que cuenta este lenguaje es el navegador [28].

## **1.15. Entorno de Desarrollo**

### **1.15.1. PyCharm**

PyCharm es un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés) desarrollado por la compañía JetBrains, está basado en IntelliJ IDEA, el IDE de la misma compañía pero enfocado hacia Java y la base de Android Studio [29].

Características:

- Autocompletado, resaltador de sintaxis, herramienta de análisis y refactorización.
- Integración con frameworks web como: Django, Flask, Pyramid, Web2Py.
- Frameworks javascripts: jQuery, AngularJS.
- Debugger avanzado de Python y Javascript.
- Integración con lenguajes de plantillas: Mako, Jinja2, Django Template.
- Soporta entornos virtuales e intérpretes de Python 2.x, 3.x, PyPy, Iron Python y Jython.

## **1.16. Sistema Gestor de Base de Datos**

### **1.16.1. MySql**

MySQL es un gestor de base de datos sencillo de usar y rápido. Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. Sin embargo, bajo constante desarrollo, MySQL hoy en día ofrece un rico y muy útil conjunto de funciones. La conectividad, velocidad y seguridad, hace que MySQL sea altamente conveniente para acceder a bases de datos en Internet [30].

## **1.17. Planificación de tareas**

### **1.17.1. Celery**

La planificación y ejecución de tareas se realiza mediante Celery.

Celery es una cola de tareas/trabajos asíncronas basado en el paso de mensajes distribuidos, centrado en la operación en tiempo real, pero también admite la programación. Las unidades de ejecución, llamadas tareas se ejecutan simultáneamente en un único o varios servidores de trabajo utilizando el multiprocesamiento. Las tareas se pueden ejecutar de forma asíncrona (en el fondo) o de forma síncrona (espere hasta que esté listo) [31].

## **1.18. Interacción con Amazon Glacier**

### **1.18.1. Boto**

La Interacción con Amazon Web Services (AWS) se realiza a través del API Boto.

Boto es una interfaz desarrollada en Python, principalmente utilizando Python 2.6.6 y Python 2.7.1, para los servicios web de Amazon, ofrece soporte para otros servicios públicos como Google Storage, además de sistemas de nubes privadas como Eucalyptus, Open Stack y Open Nebula, no requiere de las bibliotecas o paquetes que no sean los que se distribuyen con Python [32].

## **1.19. Conclusiones del capítulo**

En el capítulo se realizó un estudio de los Sistemas de Adquisición, Almacenamiento, Transmisión y Visualización de Imágenes Médicas, lo que permitió analizar el estado actual de los mismos en lo referente a los esquemas de respaldo de la información, y la necesidad de automatizar este proceso. Con el análisis de los PACS existentes se concluyó que sus esquemas de respaldo están basando en servicios en la nube o distribuidos, mientras el las PACS sigue empleando el soporte DVD, tecnología demasiado cara y obsoleta,



indicando lógicamente que el camino a seguir está en la nube. Se abordaron además todas las herramientas y tecnologías escogidas para el desarrollo del sistema para el respaldo de imágenes médicas así como la metodología de desarrollo, que en su conjunto permiten el desarrollo de la solución.

# Capítulo 2

## Características del sistema para el respaldo de imágenes médicas

En este capítulo se define la visión general del sistema y su alcance. Se describen el modelo de dominio y los requerimientos funcionales y no funcionales de la aplicación a desarrollar. Se representa además el diagrama de casos de uso del sistema con las correspondientes especificaciones de cada uno de ellos.

### 2.1. Sistema propuesto

El presente trabajo propone la automatización de los procesos de respaldo de imágenes digitales médicas. Para lograr esto, se propone un sistema que mediante la gestión de las unidades de almacenamiento permita aumentar la confiabilidad del esquema de persistencia de imágenes digitales médicas en los servidores del alas PACS y al mismo tiempo se espera que la propuesta contribuya a disminuir los costos actuales del proceso de respaldo y a aumentar la interoperabilidad de las instituciones que utilizan dicho sistema.

### 2.2. Escenario de funcionamiento del sistema

La aplicación, Sistema para el respaldo de imágenes médicas (SRIM), estará instalado, en el servidor de imágenes, en cada institución sanitaria, en la que esté desplegado el alas PACS, el SRIM será el responsable de interactuar con el servidor de almacenamiento en la nube, en nuestro caso hemos seleccionado el proveedor Amazon y su servicio Glacier por todas las características que fueron mencionadas en el anterior capítulo. Otro de los componentes con los que deberá interactuar el sistema, será el servidor de imágenes médicas, el cual gestiona y centraliza todos los estudios realizados en una institución de salud.

La interoperabilidad entre instituciones se garantizará al estar conectadas al mismo servicio de respaldo, teniendo acceso a los estudios respaldados por otras instituciones sanitarias, sin la dependencia del envío de un correo electrónico o la transferencia de un soporte DVD.

La figura 2.1 muestra el escenario de funcionamiento del SRIM, así como una representación de lo que sería la interoperabilidad entre varias instituciones conectadas al mismo servicio de respaldo.

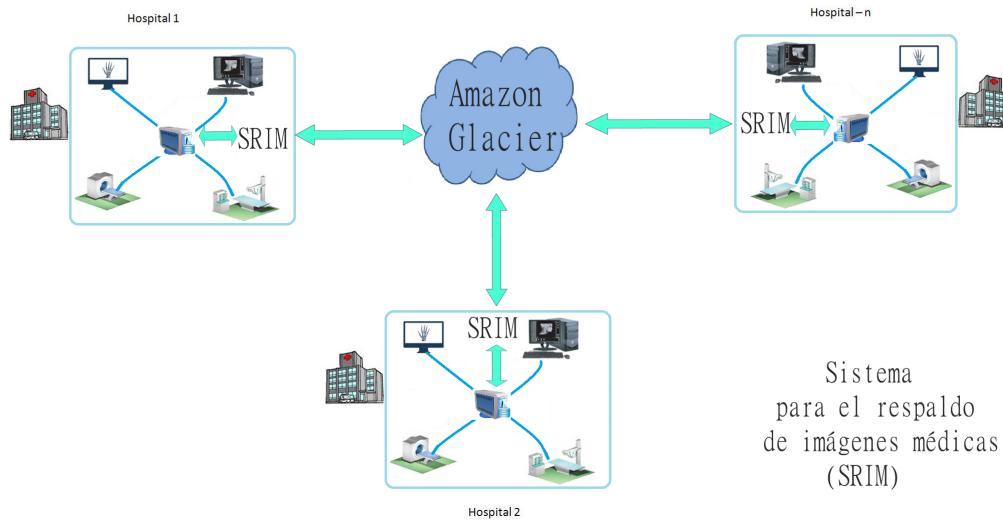


Figura 2.1: Escenario de funcionamiento de la aplicación

### 2.3. Monitoreo de disco

El monitoreo del disco se enfoca en mantener el control del espacio en disco y de los cambios en el árbol de directorios, siendo procesos de suma importancia, pues garantizan la autonomía del sistema, abstrayendo a los administradores de estos procesos.

Para el monitoreo de las unidades de almacenamiento local se emplean las librerías: tarfile, os y time y los comandos: listdir, realpath, sleep, exists, chdir, open, extractall, isdir y getsize propios de Python [33, 34].

### **2.3.1. Espacio en disco**

Monitorizar el espacio que los estudios ocupan en el servidor de almacenamiento local, tarea, aunque no muy compleja de realizar, si debe tenerse en cuenta, porque de lo contrario el espacio libre en disco alcanzaría niveles críticos y esto podría provocar la paralización de todo el alas PACS, debido a que no se podrían almacenar más estudios, para garantizar su ejecución permanente se impone que no sea muy costosa ni en tiempo ni en procesamiento y que según los niveles de alerta establecidos en la configuración, notifique a los administradores del sistema ante la ocurrencia de una situación excepcional.

### **2.3.2. Cambios en el árbol de directorios**

Monitorizar los cambios realizados en el árbol de directorios del servidor de almacenamiento local para mantener actualizado el estado de los estudios; a la hora de hacer un control de los cambios en el árbol de directorio, se debe evitar recorrer recursivamente el directorio raíz, debido a la estructura que posee y de esta forma realizar un proceso muy costoso en tiempo y procesamiento. Pero a la vez se requiere una precisión muy alta, de forma que ningún cambio en esta estructura de directorio se pierda, lo cual se traduciría en nuevos estudios que no serían respaldados y antiguos estudios eliminados que no podrían ser restaurados en caso necesario.

La figura 2.2 muestra la estructura del árbol de directorio del servidor de almacenamiento local, observándose en el directorio raíz una carpeta para cada modalidad médica, en su interior contiene una carpeta para cada paciente, dentro de esta, una carpeta para cada estudio realizado a el paciente correspondiente a dicha modalidad médica, dentro de cada estudio una carpeta para cada serie del estudio y dentro de cada serie las imágenes correspondientes al estudio.



Figura 2.2: Estructura del árbol de directorio

## 2.4. Proceso de respaldo y restauración de un estudio

El proceso de respaldo y restauración tiene como unidad central a los estudios, siendo estos los que se respaldan y/o restauran.

### 2.4.1. Respaldo de un estudio

Para el respaldo de un estudio, primero se compacta el mismo, centrándose como directorio principal la carpeta correspondiente al estudio, la cual incluye los subdirectorios de las series del estudio y las imágenes correspondientes al estudio, posteriormente se crea una conexión con el proveedor de servicios en la nube (Amazon), se verifica que exista un contenedor para la modalidad médica correspondiente al estudio, en

el caso que no exista, se crea y se procede a subir el estudio compactado, incluyendo como información adicional o metadatos la estructura del árbol de directorio del servidor de imágenes a la que pertenece el estudio, concluyendo satisfactoriamente la subida se elimina el estudio compactado, si el proceso se realiza de forma automática, a través de una planificación, en la cual se especificó que después de respaldar un estudio este se eliminase, se procede a borrar el estudio sin compactar.

#### **2.4.2. Restauración de un estudio**

Para la restauración de un estudio, primero se crea una conexión con el proveedor de servicios en la nube (Amazon), se verifica que exista un contenedor para la modalidad médica correspondiente al estudio y que exista el estudio compactado, procediendo a su descarga hacia una carpeta temporal, concluida la descarga se procede a descompactar el estudio, verificando que exista el árbol de directorios correspondiente a al paciente(modalidad médica/paciente), en caso de no existir se crea a partir de los metadatos del estudio, una vez finalizado correctamente el proceso de descompactar, se elimina el estudio compactado de la carpeta temporal.

### **2.5. Modelo del dominio**

El modelo de dominio es una representación estática de los objetos reales representados en el entorno del proyecto. Este permite mostrar de manera visual los principales conceptos que se manejan en el dominio del sistema en desarrollo. El modelo de dominio o modelo conceptual, como también se le conoce, es una representación de las clases conceptuales del mundo real y no de componentes de software. Este puede considerarse como una recopilación de información representada visualmente sobre las abstracciones relevantes, información y conceptos del dominio, lo que facilitará una mejor comprensión acerca de los conceptos utilizados por los usuarios, los conceptos con los que trabajan y con los que deberá trabajar la aplicación a desarrollar.

### **2.5.1. Conceptos fundamentales**

Para facilitar la comprensión del modelo de dominio a continuación se proporciona un marco conceptual de las definiciones identificadas.

**Unidad de almacenamiento local:** Dispositivo de hardware empleado para el almacenamiento local de información.

**Almacenamiento en la nube:** Servicio de almacenamiento brindado por la entidad proveedora de servicios en la nube.

**Planificación:** Conjunto de acciones almacenadas que brindan la información necesaria para garantizar la autonomía del sistema.

**Estudio:** Imagen o conjunto de imágenes generadas mediante el uso de un equipo de adquisición de imágenes médicas, almacenadas jerárquicamente de acuerdo a las categorías manejadas por el sistema PACS.

**Reloj:** Sistema autónomo con independencia en la ejecución de acciones, ante tareas planificadas que influyen directamente sobre los recursos que maneja el sistema.

**Cliente:** Personal vinculado con el control y administración del sistema.

### **2.5.2. Diagrama del modelo de dominio**

La figura 2.3 muestra las relaciones de los principales conceptos asociados al modelo de dominio, en el cuál se observa que el cliente tiene la posibilidad de descargar los estudios que se encuentren en el almacenamiento en la nube, así como la de respaldar los estudios que se encuentren en una unidad de almacenamiento local. También este puede definir las planificaciones de descarga y respaldo de estudios, las cuales son ejecutadas de forma autónoma por el reloj.

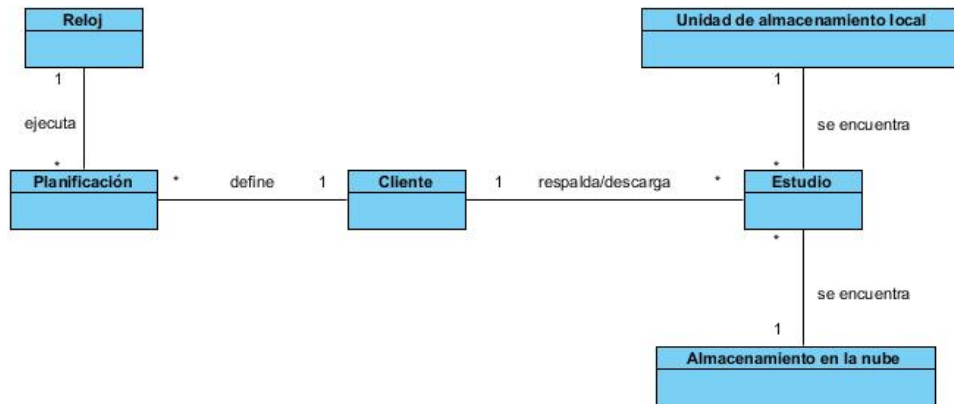


Figura 2.3: Diagrama del modelo de dominio

## 2.6. Especificación de los requisitos del software

Un requisito de software se define como una condición o capacidad que es necesitada por los clientes y debe encontrarse en un sistema o componente para satisfacer un contrato, norma, especificación u otro documento impuesto formalmente. Los requisitos del software tienen varias clasificaciones entre las que se encuentran:

**Requisitos funcionales:** describen la funcionalidad o los servicios que se espera que el sistema proveerá, sus entradas, salidas y excepciones.

**Requisitos no funcionales:** se refieren a las propiedades emergentes del sistema como la fiabilidad, el tiempo de respuesta, la capacidad de almacenamiento, la capacidad de los dispositivos de entrada/salida, y la representación de datos que se utiliza en las interfaces del sistema.

### 2.6.1. Requisitos funcionales

1. Listar cuentas en el servidor de almacenamiento en la nube
2. Gestionar cuentas en el servidor de almacenamiento en la nube
  - 2.1 Adicionar configuración de cuenta en el servidor de almacenamiento en la nube
  - 2.2 Modificar configuración de cuenta en el servidor de almacenamiento en la nube



- 2.3 Eliminar configuración de cuenta en el servidor de almacenamiento en la nube
- 2.4 Buscar configuraciones de cuenta en el servidor de almacenamiento en la nube
- 3. Listar unidades de almacenamiento local
- 4. Gestionar monitoreo de unidades de almacenamiento local
  - 4.1 Adicionar configuración de monitoreo de unidad de almacenamiento local
  - 4.2 Modificar configuración de monitoreo de unidad de almacenamiento local
  - 4.3 Eliminar configuración de monitoreo de unidad de almacenamiento local
  - 4.4 Buscar configuraciones de monitoreo de unidad de almacenamiento local
- 5. Gestionar planificación de tareas programadas
  - 5.1 Listar planificaciones
  - 5.2 Adicionar planificación
  - 5.3 Modificar planificación
  - 5.4 Eliminar planificación
  - 5.5 Buscar planificaciones
- 6. Gestionar estudios
  - 6.1 Listar estudios
  - 6.2 Buscar estudios
  - 6.3 Descargar estudio del almacenamiento en Glacier
  - 6.4 Subir estudio al almacenamiento en Glacier
- 7. Gestionar usuarios
  - 7.1 Listar usuarios
  - 7.2 Buscar usuarios

7.3 Adicionar usuario

7.4 Editar datos del usuario

7.5 Eliminar usuario

7.6 Cambiar privilegios al usuario

8. Gestionar registros históricos

8.1 Listar registros históricos

8.2 Buscar registros históricos

8.3 Eliminar registro histórico

#### **2.6.2. Requisitos no funcionales**

##### **Fiabilidad**

1. El sistema debe estar en funcionamiento de forma permanente.
2. El sistema debe contar con un grupo de políticas de accesibilidad a las diferentes funcionalidades del mismo en dependencia del nivel de autorización que presente un usuario determinado.
3. El sistema debe ser capaz de recuperarse ante un fallo.

##### **Seguridad**

4. Debe existir un registro de todos los sucesos.
5. Comunicación con el servidor y el inicio de sesión, debe ser cifrada, usando SSL.

##### **Diseño e Implementación**

6. Debe ser implementado en el framework Django.
7. Compatibilidad con los diferentes navegadores existentes: Chrome 37, Firefox 33, Internet Explorer 8, Opera 28 y Safari 5 o versiones superiores.

### **Interfaz de usuario**

8. El Cliente debe tener una interfaz gráfica, visualmente atractiva para el usuario.
9. El Sistema debe facilitar al usuario mensajes que lo ayuden a llevar a cabo la tarea que realiza.
10. Se debe hacer uso de botones, con imágenes que indiquen de modo intuitivo la función que realizan.
11. Los controles visuales deben mostrar mensajes que indiquen su función.
12. Se requiere que el Cliente tenga una interfaz gráfica que permita la interacción con el sistema brindado toda la información referente a los recursos que maneja el sistema.
13. Se requiere una interfaz ligera, que permita de manera sencilla configurar los parámetros de conexión y monitoreo del sistema.

### **Funcionamiento**

14. Usar como sistema operativo Linux distribución Ubuntu 14.04.

### **Usabilidad**

15. Los parámetros de funcionamiento del sistema deben ser configurables.

### **Hardware**

Requerimientos para la máquina servidora.

16. Memoria RAM de 2 GB o superior
17. Disco Duro de 120 GB
18. Procesador Intel Celeron de 2 GHz o superior

## **2.7. Definición de los Casos de Uso del Sistema**

Un caso de uso es, en esencia, una interacción típica entre un usuario y un sistema de cómputo [35]. El mismo capta alguna funcionalidad visible para el usuario y logra un objetivo discreto para el mismo. Los

casos de uso pueden considerarse como un documento narrativo que describe la secuencia de eventos de un actor que utiliza un sistema para completar un proceso [36].

Durante el levantamiento de requisitos para el presente trabajo se identificaron los actores y casos de uso que guiarían el desarrollo del producto final. En esta sección se enumerarán y describirán cada uno de ellos.

### **2.7.1. Actores del sistema**

La tabla que se muestra a continuación ofrece un resumen de los actores del sistema.

Actor	Descripción
Administrador	Encargado de realizar la configuración general del sistema.
Usuario	Encargado de realizar las búsquedas en Glacier.
Cliente	Usuario encargado de realizar las búsquedas en Glacier.
Reloj	Inicia los procesos automáticos del sistema.

Tabla 2.1: Actores del Sistema.

### 2.7.2. Diagrama de casos de uso del sistema

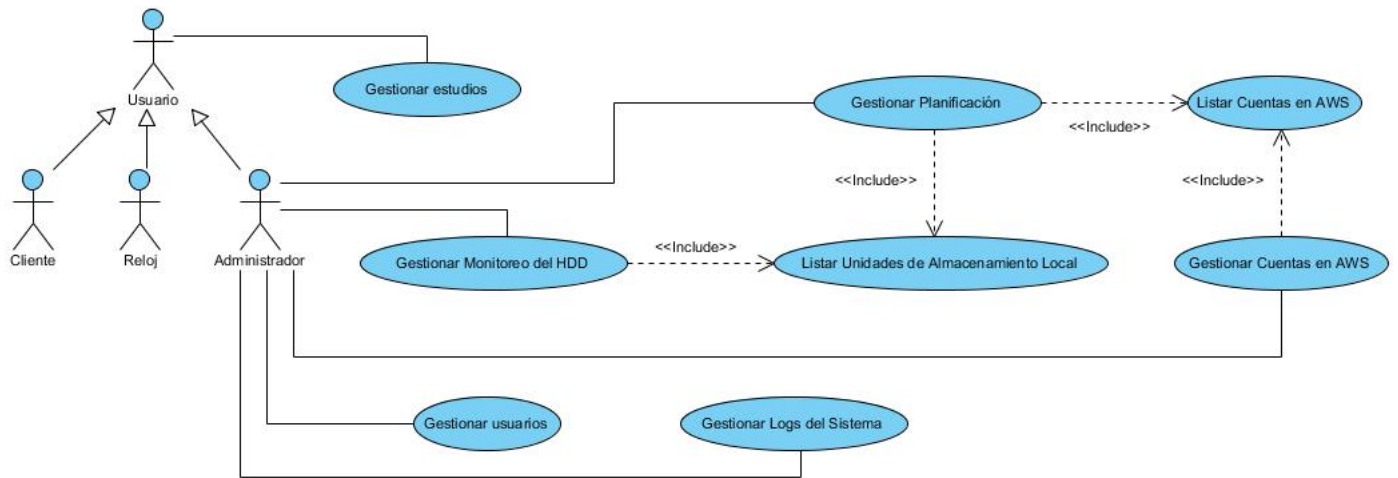


Figura 2.4: Diagrama de casos de uso del sistema

## 2.8. Arquitectura

La Arquitectura de un Software indica la estructura, funcionamiento e interacción entre las partes del mismo. Es un nivel de diseño que se centra principalmente en aspectos más allá de la estructura de datos de la computación.

### 2.8.1. Estilo arquitectónico (Cliente-Servidor)

Para el desarrollo del sistema se definió la utilización del estilo Cliente-Servidor.

El esquema cliente-servidor es un modelo de computación. En él, el procesamiento requerido para ejecutar una aplicación o conjunto de aplicaciones relacionadas, se divide entre dos o más procesos que cooperan entre sí. Usualmente la mayoría del trabajo pesado se hace en el proceso llamado servidor y el (los) proceso(s) cliente(s) sólo se ocupa de la interacción con el usuario (aunque esto puede variar).

Los principales componentes del esquema cliente-servidor son entonces los clientes, los servidores y la infraestructura de comunicaciones.

En el entorno de la aplicación desarrollada (como en el de las aplicaciones web en general) el cliente puede ser visto como el navegador web, el cual es usado por el usuario para acceder a la aplicación y el servidor en este caso se trata de un servidor web, aplicación que soporta diferentes tipos de protocolos web con el objetivo de dar respuesta a los pedidos del cliente [37].

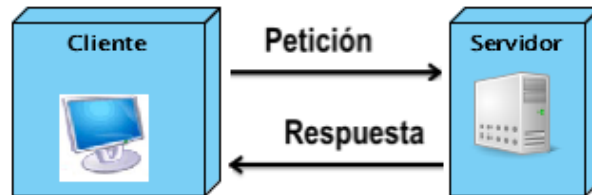


Figura 2.5: Flujo del estilo arquitectónico Cliente-servidor

### 2.8.2. Patrón arquitectónico (Modelo Vista Controlador)

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Este patrón se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio y el controlador es el responsable de recibir los eventos de entrada desde la vista [38].

El propósito del MVC es separar los datos y lógica de negocio de la lógica de presentación.

Django es un framework de desarrollo web que sigue el MVC, aunque los desarrolladores del proyecto prefieren llamarlo Modelo-Plantilla-Vista (Model-Template-View “MTV”) [22].

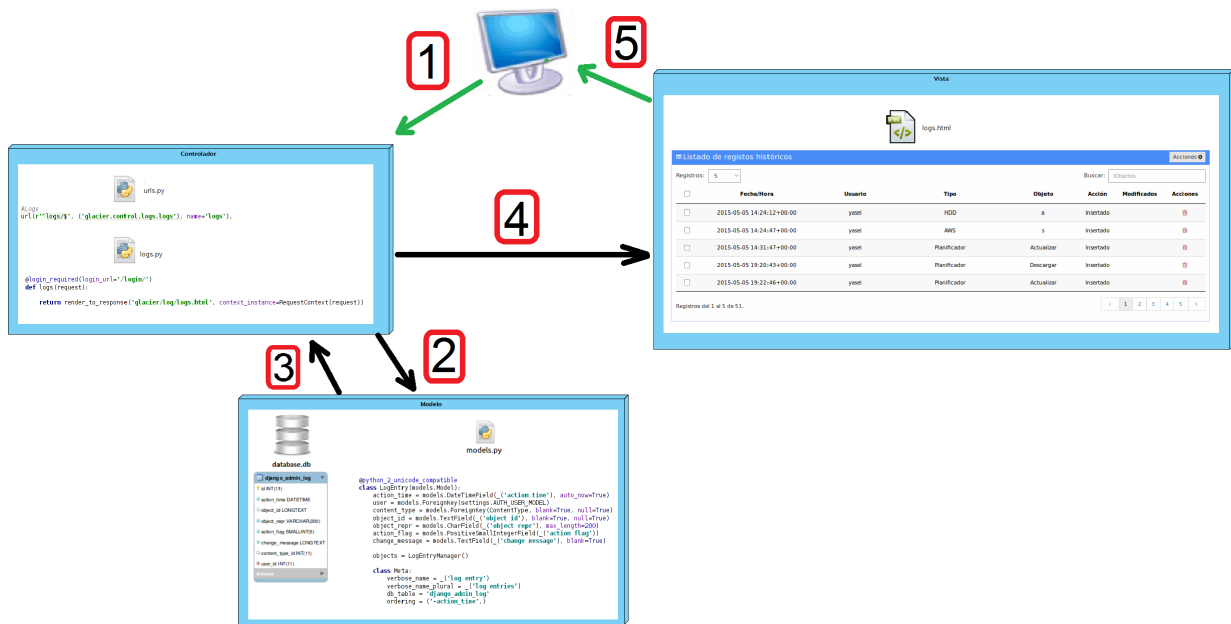


Figura 2.6: Flujo del patrón MVC

Leyenda:

- (1) El cliente envía una petición al controlador vía URL.
- (2) El controlador solicita los datos al modelo.
- (3) El modelo devuelve los datos.
- (4) El controlador selecciona la vista a mostrar.
- (5) Se devuelve la vista (página) que carga los datos del modelo seleccionado.

## 2.9. Análisis

Durante la fase de análisis se examinaron, refinaron y estructuraron los requisitos, con el fin de conseguir una comprensión más precisa que ayudará a una correcta estructuración del sistema como un todo. Además se obtuvieron las primeras propuestas de las clases encargadas de garantizar las funcionalidades que el cliente necesita.

Para el desarrollo de la aplicación fue necesaria la elaboración de los diagramas de clases del análisis asociados a la misma, para mejorar la visión de los programadores en cuanto a la estructura del sistema.

## 2.10. Diseño

En el desarrollo de un sistema informático el diseño constituye un elemento fundamental. Tiene como objetivo modelar el sistema y encontrar su forma para que soporte todos los requisitos, tanto funcionales como no funcionales. Aquí es donde se define cómo va a quedar el sistema en un primer momento.

### 2.10.1. Modelo de Datos

La figura 2.7 muestra el modelo de datos del SRIM, evidenciándose que el Sistema se centra en el control y ejecución automática de tareas asociadas a los estudios, siendo estas tablas las de mayor cantidad de peticiones y entradas.

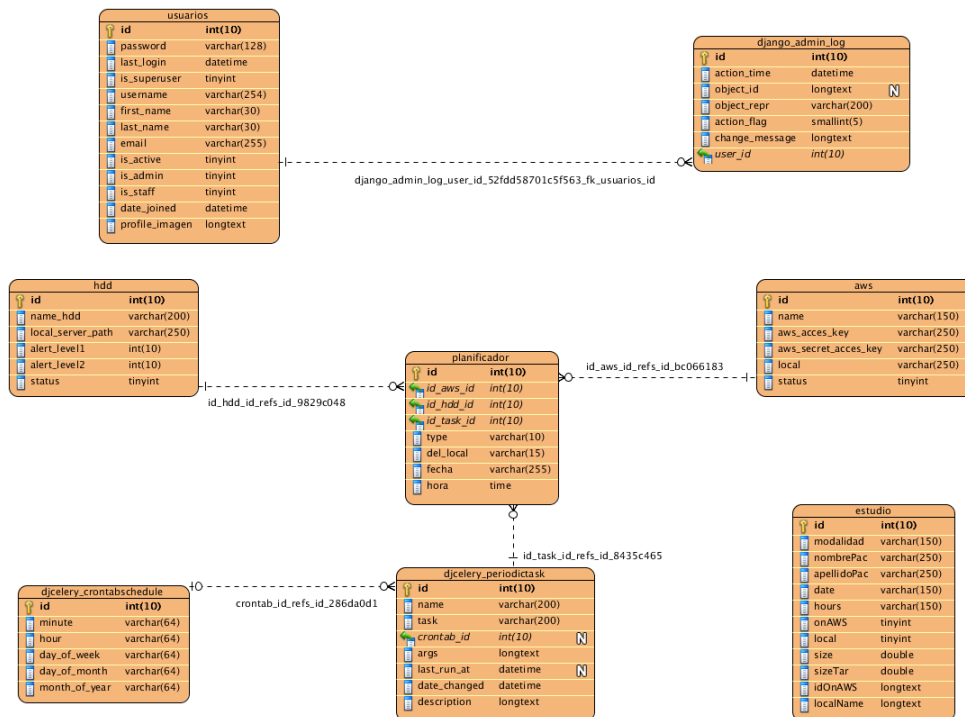


Figura 2.7: Modelo de Datos



## **2.11. Conclusiones del capítulo**

En este capítulo se elaboró una propuesta de solución. Además, se realizó el Modelo de Dominio donde se especificaron los conceptos u objetos que se relacionan en el problema a resolver. También, se detallaron los requisitos funcionales y no funcionales del componente propuesto y se definieron los casos de uso y actores del sistema.

# Capítulo 3

## Implementación y prueba del sistema para el respaldo de imágenes médicas

En este capítulo se describe la implementación y las pruebas que se le realizarán al sistema. Las pruebas al sistema permitirán verificar los resultados de la implementación ya que son un elemento crítico para la garantía de la calidad del software.

### 3.1. Diagrama de componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Normalmente contienen componentes, interfaces y relaciones entre ellos y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo.

Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. En cuanto a los componentes, sólo aparecen tipos de componentes, ya que las instancias específicas de cada tipo se encuentran en el diagrama de despliegue. Los diagramas de componentes muestran los componentes software que constituyen una parte reusable, sus interfaces, y sus interrelaciones, en muchos aspectos se puede considerar que un diagrama de componentes es un diagrama de clases a gran escala. Cada componente en el diagrama debe ser documentado con un diagrama de componentes más detallado, un diagrama de clases, o un diagrama de casos de uso [39].

La figura 3.1 muestra los principales componentes del sistema siguiendo el patrón arquitectónico MVT utilizado por Django. El contenedor Model muestra los componentes correspondientes al modelo de los datos, el View los correspondientes a las vistas que se encuentran definidas y el Template los correspondientes a las plantillas que definirán como se mostrarán las vistas.

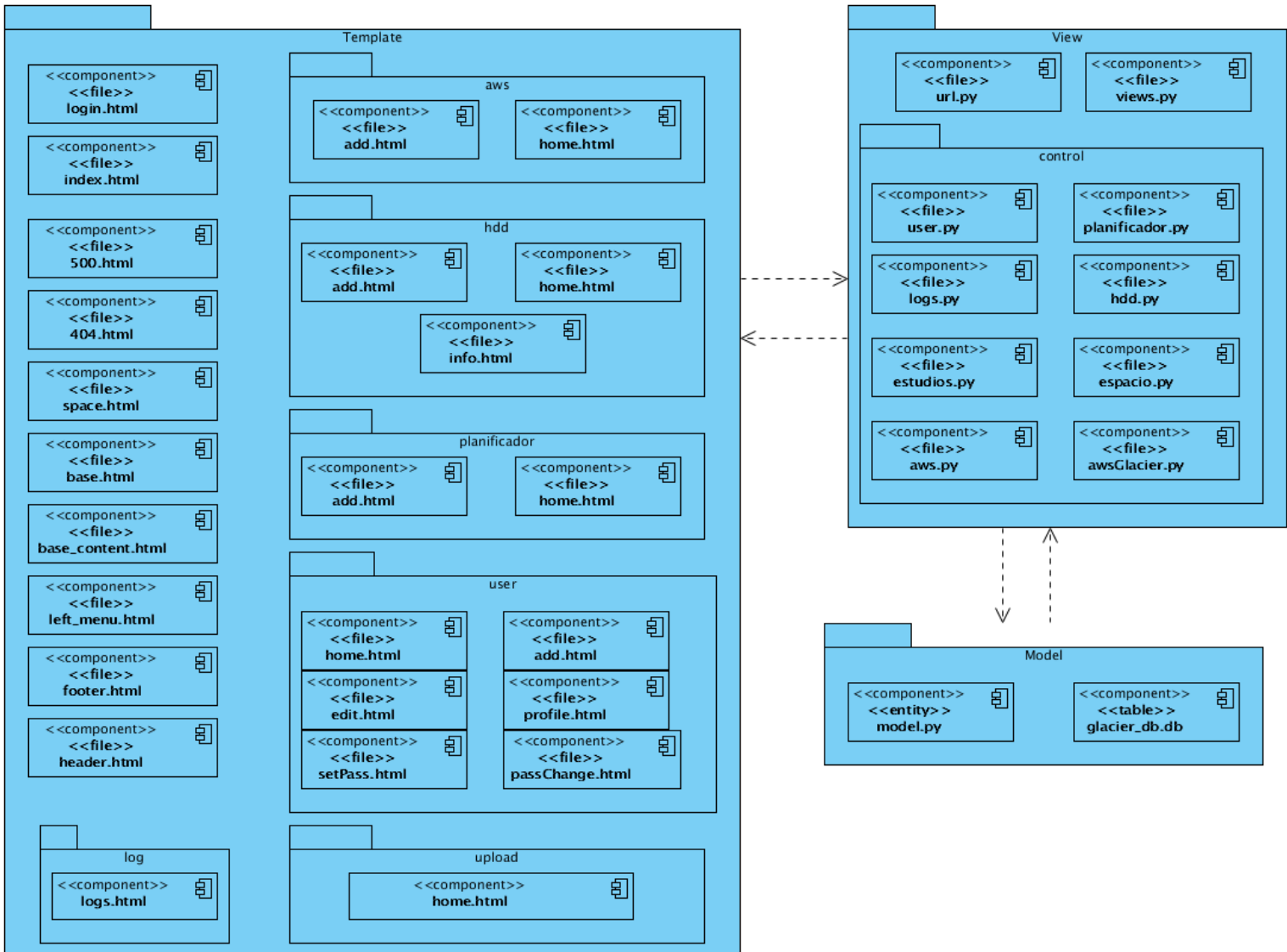


Figura 3.1: Diagrama de componentes

### 3.2. Diagrama de despliegue

La figura 3.2 muestra los principales elementos del diagrama de despliegue del sistema, siendo el núcleo del mismo la interacción que se establece entre los servidores: de aplicación, de imágenes del alas PACS y Amazon Glacier, garantizando el respaldo y la descarga de los estudios de las instituciones sanitarias así como la interoperabilidad entre las mismas.

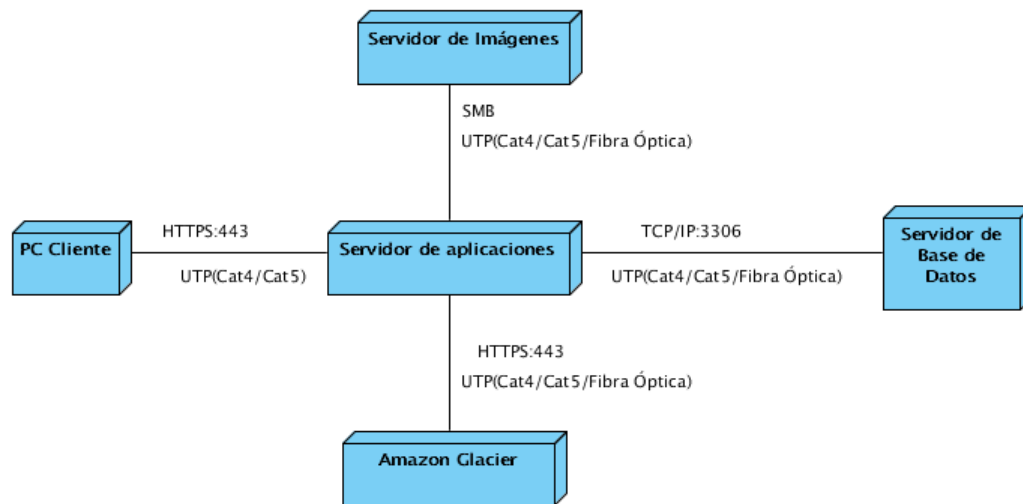


Figura 3.2: Diagrama de despliegue

### 3.3. Prueba del sistema

Las pruebas de software constituyen un elemento crítico para garantizar la calidad final del producto que se está desarrollando. El objetivo de estas es comprobar que la aplicación funciona correctamente y cumple con los requisitos establecidos [40].

#### 3.3.1. Niveles de prueba

Entre los niveles de prueba se encuentran las pruebas unitarias, de integración, de funcionalidad, de sistema, de aceptación y de instalación. En el proceso de pruebas de la aplicación se utilizaron los siguientes niveles:

**Pruebas unitarias:** se realizan con el objetivo de detectar errores en los datos, lógica y algoritmos utilizados en la implementación del sistema. En este nivel se prueban fragmentos de código por separado para asegurar que funcionen correctamente, esto facilita la documentación del código y que los errores sean más fáciles de encontrar.

**Pruebas de integración:** luego de realizar las pruebas unitarias se prueban todos los elementos como un todo con el objetivo de encontrar errores de comunicación entre las distintas partes del software y darle solución a estos.

**Pruebas funcionales:** el objetivo de estas pruebas es detectar errores en la implementación de los requisitos.

### **3.3.2. Métodos de Pruebas**

Los métodos de prueba son utilizados con el propósito de descubrir fallos y para descubrir fallos en el funcionamiento del software. En las pruebas realizadas a la aplicación se hizo uso del método de Caja Negra y Caja blanca.

**Pruebas de Caja Negra:** estas pruebas se centran en los requisitos funcionales. Su objetivo es saber qué es lo que hace el sistema sin dar importancia a cómo lo hace y permiten encontrar funciones incorrectas o ausentes y errores de interfaz, estructuras de datos, inicialización y terminación [41].

**Caja blanca:** como parte de las pruebas unitarias se utilizaron las pruebas de caja blanca en las que se probó el correcto funcionamiento del código. Estas pruebas se realizan con acceso al código fuente de la aplicación y se trabaja con entradas, salidas y el conocimiento interno del sistema. El método se llevó a cabo por el equipo de desarrollo y no se generó documentación de las mismas [41].

### **3.3.3. Diseño de casos de prueba**

Teniendo como guía la descripción de los casos de uso del sistema generados durante el flujo de análisis y diseño se crearon los diseños de casos de prueba, siguiendo la plantilla 3.1, correspondientes a estos. Los casos de prueba constituyen la guía fundamental por la cual se regirán las pruebas de software.

<b>Código:</b> (código de identificación)	<b>Historia de Usuario:</b> (CUS al que pertenece)
<b>Nombre:</b> (acción a desarrollar)	
<b>Clase:</b> (válida y/o no válida)	
<b>Descripción:</b> (descripción del objetivo de la prueba)	
<b>Condiciones de ejecución:</b> (requisitos para su ejecución)	
<b>Entrada/ Pasos de ejecución:</b> (pasos de ejecución)	
<b>Resultado esperado:</b> (respuesta del sistema)	
<b>Evaluación de la prueba:</b> (resultado de la prueba)	

Tabla 3.1: Plantilla casos de prueba.

### 3.3.4. Pruebas de Seguridad

El uso de Django garantizara la protección automática de muchos de los errores comunes de seguridad, brindando facilidades y herramientas para combatir vulnerabilidades como inyecciones SQL, Cross-Site Scripting (XSS), inyección de cabeceras de email, exposición de mensajes de error, entre otras facilidades. El sistema provee además un sistema de autenticación donde se solicita nombre de usuario y contraseña verificando que los datos introducidos sean válidos y garantizando que usuarios no autorizados no accedan a los datos que se manejan en el sistema [23].

Para la realización de las pruebas de Seguridad al Sistema, primeramente, se identificaron los distintos tipos usuarios que están definidos en el sistema y sus respectivos niveles de acceso o privilegios definidos; luego se creó un usuario de cada tipo y se procedió con las pruebas [42, 43].

Las primeras pruebas fueron intentos de violación de la seguridad del sistema, tratando de acceder a éste con usuarios no existentes o utilizando usuarios reales con contraseñas erróneas. Estas pruebas de acceso

arrojaron resultados favorables al sistema, ya que no se logró acceder con usuarios no registrados previamente ni con falsas contraseñas, demostrando que el Sistema cuenta con un mecanismo de autenticación seguro, garantizando que solamente trabajen con el programa personas autorizadas previamente [42, 43].

Luego de acceder al sistema con un usuario específico comenzó la segunda fase de la prueba, donde se intentó acceder a funciones no permitidas para ese determinado usuario, como son: creación, modificación o eliminación de usuarios, modificación de los permisos de usuarios existentes, liberar espacio de la unidad de almacenamiento, entre muchas otras funciones de las que depende el correcto funcionamiento del sistema. Los resultados de las pruebas fueron los esperados, los usuarios registrados en el Sistema tienen acceso solamente a las funcionalidades predefinidas por los administradores del sistema, impidiendo en todo momento que se lleven a cabo funcionalidades por personas no autorizadas.

### **3.3.5. Pruebas de Carga**

Para las pruebas de carga se utilizó una consulta correspondiente a la funcionalidad Gestionar Planificaciones y variando la cantidad de conexiones simultáneas para conocer el comportamiento del servidor de Bases de Datos según la cantidad de clientes del sistema.

Los resultados arrojados se muestran en la Tabla 3.2, donde se registraron los datos utilizados, y el tiempo que se tomó el sistema para dar respuesta a todos los clientes simulados. Los tiempos de respuestas se expresaron en milisegundos (ms).

Iteración	Conexiones	Peticiones	Tiempo de respuesta (ms)
Primera	100	1	203
	100	100	380
Continúa en la próxima página			

Segunda	200	1	242
	200	100	411
Tercera	300	1	377
	300	100	671

Tabla 3.2: Registro de tiempos de respuesta

### 3.4. Validación de las variables de la investigación

A continuación se evalúa la variable: confiabilidad

#### 3.4.1. Aumento de la confiabilidad

Para validar la confiabilidad se emplearán algunas de sus aristas como son:

- Respaldo
- Consistencia de los archivos
- Desempeño del sistema de archivos
- Disponibilidad de la información
- Interoperabilidad

Se consideraron las aristas a las que deben responder ambos sistemas. Esta comparación se resume en la tabla 3.3.

	alas PACS	SRIM
Continúa en la próxima página		



<b>Respaldo</b>	La copia en soporte DVD	Respaldo en línea contratado (ilimitado)
<b>Consistencia de los archivos</b>	Método actual no garantiza que la información respaldada se mantenga correcta con el paso del tiempo, dependiendo la durabilidad del manejo y condiciones de almacenamiento de la unidad DVD	La información respaldada se mantiene correcta con el paso del tiempo, pues Amazon por si sólo incluye en el servicio de Glacier, el respaldo de la información, los datos son inmutables.
<b>Desempeño del sistema de archivos</b>	Acceso a los datos locales inmediato Acceso a los datos en soporte DVD indefinido, existen demasiados factores	Acceso a los datos locales inmediato Acceso a los datos respaldados de tres a cinco horas
<b>Disponibilidad de la información</b>	Período máximo de cinco años (en dependencia de la capacidad de almacenamiento)	Ilimitado (24 horas los 365 días, período superior a cinco años)
<b>Interoperabilidad</b>	A través del alas PACS Dicom Mail (envío de correos electrónicos)	A través del propio Sistema

Tabla 3.3: Validación de confiabilidad.

### 3.5. Funcionalidades extras

Las siguientes funcionalidades se implementaron con el objetivo de brindar y garantizar el total control sobre la unidad de almacenamiento de los estudios.

- Para el control sobre los estudios, se mantiene una vigilancia permanente, intervalo de espera de un segundo, sobre el directorio donde se encuentran almacenados estos, permitiendo actualizar automáticamente el Sistema en caso de que se creen y/o eliminen estudios.

- Se mantiene permanentemente la observación sobre la salud de la unidad de almacenamiento, en cuanto al porciento de la capacidad disponible, parámetros configurables por los usuarios, en caso que se sobrepase alguno de los dos niveles (1: Advertencia y 2: Alerta Crítica), el Sistema enviará automáticamente cada dos horas el respectivo correo electrónico a todos los administradores, registrados en el Sistema, detallando la hora y fecha de la detección, el nombre establecido para la unidad de almacenamiento, su ruta y el porciento de la capacidad que tiene disponible así como el porciento del(de los) nivel(es) sobrepasado(s), brindando la posibilidad de liberar espacio a través de Sistema.
- Para liberar espacio [Figura 3.3] el Sistema tiene en cuenta la antigüedad de los estudios que se encuentran en la unidad de almacenamiento, realiza una gráfica de barras horizontal en la cual se organizan los estudios según su fecha de creación en cuatro categorías: menor de tres meses, entre tres meses y tres años, entre tres años y cinco años y mayor de cinco años, además permite seleccionar a través de una barra de desplazamiento horizontal el espacio a liberar en megabytes, alertando al administrador, que de continuar con dicha acción se borrarán los estudios mientras no se supere en total los megabytes seleccionados anteriormente, comenzando por los estudios de mayor antigüedad.

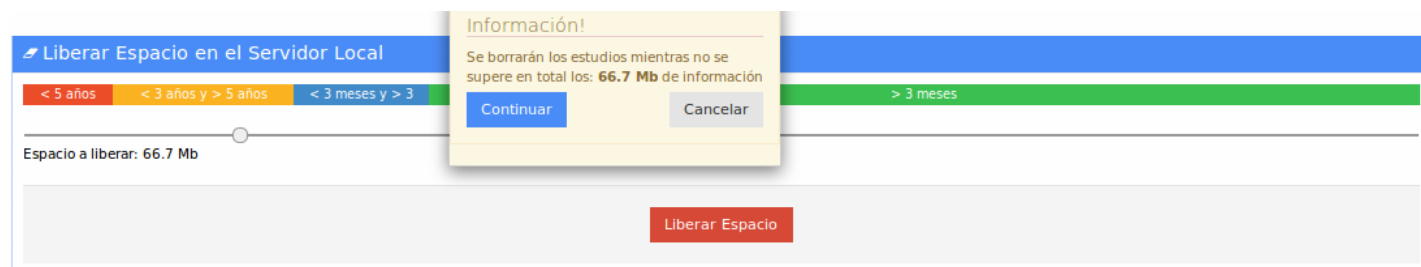


Figura 3.3: Liberar espacio

## 3.6. Análisis económico

### 3.6.1. Información generada periodo cinco años

Caso 1:

- El servidor de imágenes de cada institución de salud tiene una capacidad de almacenamiento de veinte terabytes.

- Las instituciones de salud cuentan con una media de cuatro equipos de obtención de las imágenes médicas.
- La información generada por un estudio a un paciente de trescientos megabytes.
- En un día de trabajo se pueden realizar treinta estudios por equipo.

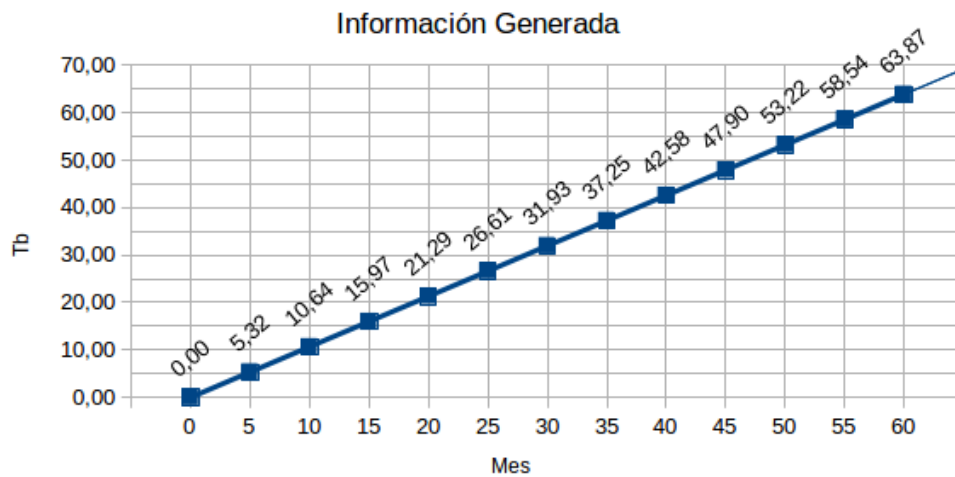


Figura 3.4: Información generada en cinco años

### 3.6.2. Costo de almacenar la información del esquema actual.

Caso 1 (36 000 megabytes y 3.4 años):  $44676000 \text{ megabytes} / 4812.8 \text{ megabytes} = 9283 \text{ DVD} \times 0.54 \text{ USD/DVD} = 5012.82 \text{ USD}$ .

- Costo de almacenar la información en soporte DVD de 83.55 USD/MES <sup>1</sup>.
- Un informático, salario mínimo de 1158.73 USD/MES.
- Un aire acondicionado Split Lg 24000 Btu de 34920.6 USD, el cual consume mensualmente 413.4 KWH/MES para un gasto de 12.82 USD.
- Diez estantes metálicos de 2.00m X 0.90m X 0.35m de 150 Kg a 548 USD/CU.

<sup>1</sup>Estos datos son obtenidos a partir de la búsqueda en Internet de sus precios en Venezuela

La figura 3.5 muestra el costo asociado al almacenamiento de la información de una institución sanitaria, 63.87 TB, empleando el esquema actual del alas PACS, el costo inicial se refiere a la compra del aire acondicionado y de los estantes.

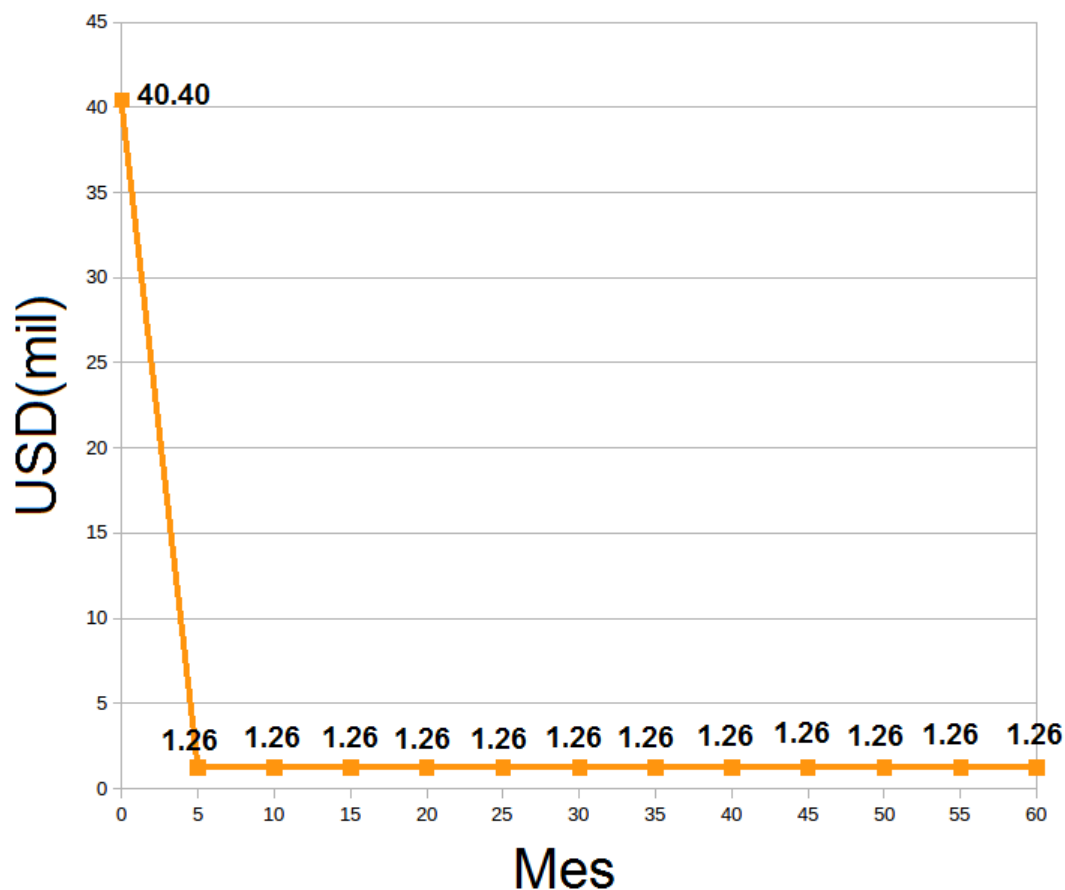


Figura 3.5: Costo de almacenar la información del esquema actual

### 3.6.3. Costo de almacenar la información con el SRIM.

La figura 3.6 muestra el costo asociado al almacenamiento de la información de una institución sanitaria, 63.87 TB, empleando el SRIM, para el cálculo del costo se tuvo en cuenta el volumen de almacenamiento facturado en un mes corresponde a la media del almacenamiento utilizado durante todo el mes, medido en gigabyte-meses (GB-meses). El tamaño de cada uno de sus archivos se calcula según la cantidad de datos que carga más unos 32 kilobytes de datos adicionales para la indexación y los metadatos (la descripción de su

archivo).

El uso del almacenamiento calculará según la ecuación:

$$\text{tamaño\_archivo (bytes)} \times \text{días\_almacenamiento} \times 24 \text{ horas} \\ = \text{almacenamiento (Bytes-Horas)}$$

La conversión a GB-mes (tomando meses de 30 días) según la ecuación:

$$\text{almacenamiento (Bytes-Horas)} \times (1 \text{ GB} / 1\,073\,741\,824 \text{ bytes}) \\ \times (1 \text{ mes} / 720 \text{ horas}) = \text{almacenamiento\_pagar (GB-Meses)}$$

El costo del mes se calcula a partir del *almacenamiento\_pagar (GB – Meses)* y la tabla de precios A.1.

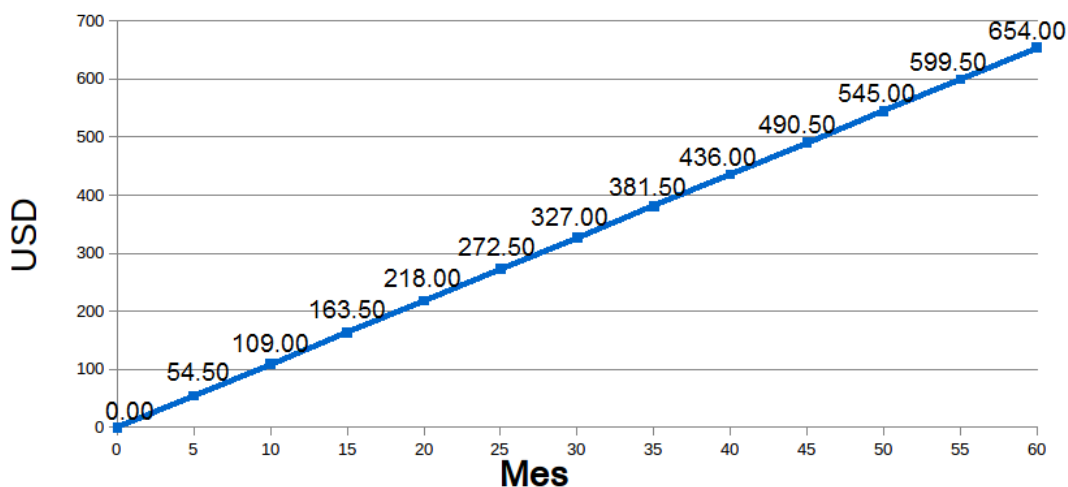


Figura 3.6: Costo de almacenar la información con el SRIM

### 3.6.4. Costo de almacenar la información.

La figura 3.7 muestra el costo asociado al almacenamiento de la información generada durante cinco años por una institución sanitaria, 63.87 TB, empleando el esquema actual del alas PACS y el SRIM, marcándose la notada diferencia entre los costos desde el inicio del proceso, concluyendo con un acumulado de 115706.60 USD para el esquema actual del alas PACS y de 19946.48 USD para el SRIM.

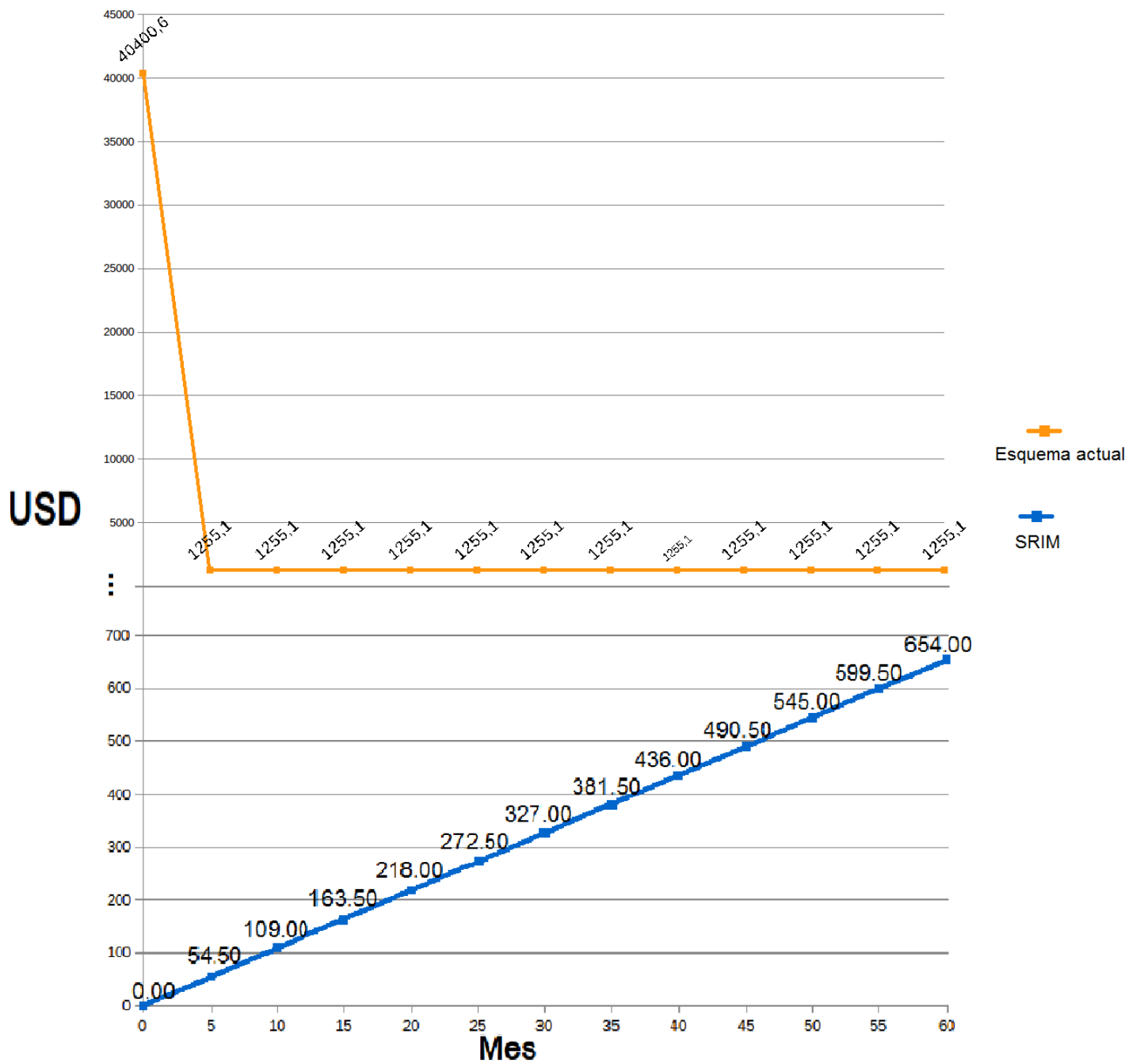


Figura 3.7: Costo de almacenar la información

### **3.7. Aporte**

Aumento de la capacidad de almacenamiento, en las instituciones sanitarias, teniendo la facilidad de almacenar la información generada por los estudios de los pacientes, por el período de tiempo superior a cinco años.

Aumento de la interoperabilidad en el sistema alas PACS, interconectando las instituciones sanitarias entre sí y con otras instituciones de interés (Universidades, centros de I+D, entre otros), permitiendo que la información sea de fácil acceso para las instituciones.

Los pacientes se podrán realizar consultas de segundas opiniones fuera de la institución; su información será de fácil acceso para las instituciones de salud para realizar el seguimiento de su(s) enfermedad(es) según el interés del paciente o de la nación, no se necesitará repetir los estudios por motivos de pérdida o daño por parte de la institución o por un cambio de institución sanitaria realizado por el paciente, disminuyendo los daños ocasionados a la salud del paciente.

Se podrán realizar investigaciones sobre enfermedades que afecten a sectores determinados de la población, garantizando la existencia de un histórico de datos superior a cinco años y la representatividad de toda la población, además de análisis estadísticos de mayor certeza, para la toma de decisiones pertinentes a la salud de la población.

Económicamente existirá un esquema de respaldo de almacenamiento y un mecanismo de recuperación ante desastres bajo en costes financieros, recursos humanos e infraestructura y se disminuirán los costes asociados a los estudios extras realizados por motivos de pérdida o daño.

### **3.8. Conclusiones del capítulo**

- Se obtuvo el Sistema para el respaldo de imágenes médicas, desarrollado con tecnologías libres, y provisto de elementos de seguridad y un entorno amigable, gracias a la utilización del framework Django.

- El sistema fue probado mediante pruebas de Caja Blanca y Caja Negra, y validado utilizando pruebas de aceptación.
- Se evaluaron las variables consideradas en la investigación, demostrando que la solución desarrollada contribuye al respaldo de imágenes médicas, permitiendo aumentar la confiabilidad del esquema de persistencia de imágenes digitales médicas en los servidores de las PACS.



## Conclusiones

La realización del presente trabajo ha posibilitado cumplir con los objetivos propuestos, por lo que se pueden plantear las siguientes conclusiones:

1. La confiabilidad de un sistema de persistencia de imágenes médicas es una de las principales aristas que deben caracterizar cualquier sistema que manipule estudios imagenológicos, considerando la sensibilidad de este tipo de información y el daño que podría constituir para la institución y el paciente su pérdida total o parcial.
2. El estudio del estado del arte realizado evidenció la creciente tendencia del uso de servicios de almacenamiento en la Nube por los Sistemas de Adquisición, Almacenamiento, Transmisión y Visualización de Imágenes Médicas, y a la misma vez, demostró la necesidad de actualizar el sistema alas PACS hacia un esquema de persistencia y respaldo de la información más fiable que el utilizado actualmente en los despliegues de la solución.
3. La solución presentada contribuye a un alto incremento de la confiabilidad del sistema alas PACS, así como una notable disminución en los costos de mantenimiento del esquema de respaldo, no solo de una institución hospitalaria, sino de un ambiente multientidad en la cual cooperan varias instituciones sanitarias.

## **Recomendaciones**

Luego de culminada la investigación y cumplido el objetivo general de la misma, en vista a mejorar el funcionamiento del sistema, se recomienda:

- Explorar la factibilidad de nuevos servicios de almacenamiento en la Nube, que permitan continuar abaratando los costes del esquema propuesto y a la misma vez brindar soporte para la integración al servicio de almacenamiento de preferencia del usuario o la institución.
- Implementar una interfaz de servicios web que permitan la interconexión entre el servidor de imágenes y el sistema propuesto, disminuyendo aún más la dependencia actual de la intervención de un administrador en el sistema para la realización de tareas simples como el restablecimiento de estudios.
- Incluir en el sistema un tercer actor, el Paciente, brindándole a estos la posibilidad de acceder a sus estudios y realizar operaciones de descarga o visualización de los mismo. En este caso se sugiere la implementación de un nuevo sistema, que se nutra de la información almacenada en la Nube, evitando la interacción directa y sobrecarga del sistema de respaldo.

## Referencias bibliográficas

- [1] T. Porter and T. Duff, “Compositing digital images,” in *ACM Siggraph Computer Graphics*, vol. 18, pp. 253–259, ACM, 1984.
- [2] H. Huang, *PACS and imaging informatics: basic principles and applications*. John Wiley & Sons, 2010.
- [3] F. B. i Rovira and M. C. Díaz, “Almacenamiento y transmisión de imágenes. pacs.” [http://www.seis.es/seis/is/is45/IS45\\_54.pdf](http://www.seis.es/seis/is/is45/IS45_54.pdf), 2004.
- [4] R. O. Aranegui, C. M. García, C. M. Martínez-Pereda, and J. L.-Q. Martínez, “Radiografía sin película. actualización en diagnóstico por la imagen en odontoestomatología,” *Científica Dental. Revista científica de formación continuada*, vol. 3, no. 1, pp. 11–20, 2006.
- [5] R. Everhart, “Flatpaneldr.com.” <http://www.flatpaneldr.com/index.php/what-is-a-pacs/>, Enero 2011.
- [6] L. G. Rodríguez, Y. D. Yero, H. R. González, P. M. Riesgo, R. B. Cáceres, A. J. Gamboa, and A. L. Morales, “Arquitectura del sistema para la transmisión, almacenamiento y visualización de imágenes médicas en instituciones hospitalarias: Alas pacs,” *Serie Científica*, vol. 2, no. 7, 2009.
- [7] “Informatica moderna.” <http://www.informaticamoderna.com/Backup.htm>.
- [8] J. S. Soto, “Computación en la nube,” *Investiga. TEC*, vol. 10, no. 10, pp. pág-4, 2013.
- [9] K. M. Soto, “Almacenamiento en la nube?,”
- [10] M. Healthcare, “Comparison chart.” <http://www.merge.com/MergeHealthcare/media/company/In%20The%20News/merge-pacs-comparison.pdf>.

- [11] G. C. Kagadis, C. Kloukinas, K. Moore, J. Philbin, P. Papadimitroulas, C. Alexakos, P. G. Nagy, D. Visvikis, and W. R. Hendee, "Cloud computing in medical imaging," *Medical physics*, vol. 40, no. 7, p. 070901, 2013.
- [12] "Siemens global website." <http://www.healthcare.siemens.com/medical-imaging-it>.
- [13] "Philips - united states." [http://www.healthcare.philips.com/us\\_en/products/healthcare\\_informatics/products/enterprise\\_imaging\\_informatics/isite\\_pacs/isite\\_pacs.wpd](http://www.healthcare.philips.com/us_en/products/healthcare_informatics/products/enterprise_imaging_informatics/isite_pacs/isite_pacs.wpd).
- [14] P. Mell and T. Grance, "The nist definition of cloud computing (draft)," *NIST special publication*, vol. 800, no. 145, p. 7, 2011.
- [15] L. J. Aguilar, "Computación en la nube: Notas para una estrategia española en cloud computing," *Revista del Instituto Español de Estudios Estratégicos*, vol. 1, no. 1, 2013.
- [16] B. Voo, "Hongkiat.com." [www.hongkiat.com/blog/dropbox-gdrive-skydrive/](http://www.hongkiat.com/blog/dropbox-gdrive-skydrive/).
- [17] Amazon, "Amazon web services." <http://aws.amazon.com/es/s3/pricing/>.
- [18] P. Letelier, "Métodologías ágiles para el desarrollo de software: extreme programming (xp)," tech. rep., 2006.
- [19] J. Rumbaugh, G. Booch, and I. Jacobson, "El lenguaje unificado de modelado," tech. rep., Madrid, 2000.
- [20] "Características de visual paradigm." [www.visual-paradigm.com](http://www.visual-paradigm.com).
- [21] R. G. Duque, "Python para todos," tech. rep., España, 2006.
- [22] A. Holovaty and J. Kaplan-Moss, "El libro de django," tech. rep., España, 2007.
- [23] "django.es." <http://django.es/>.
- [24] "The jquery project." <http://jquery.com/>.

- [25] “Highcharts.” <http://www.highcharts.com/products/highcharts>.
- [26] C. Minnick and C. Valentine, “Xhtml serie práctica,” tech. rep., Nueva York,, 2000.
- [27] J. E. Pérez, “Introducción a css,” tech. rep., 2009.
- [28] J. E. Pérez, “Introducción a javascript,” tech. rep., 2009.
- [29] “Jetbrains suite.” <https://www.jetbrains.com/>.
- [30] “Mysql.” <http://dev.mysql.com>.
- [31] “Celeryproject.” <http://www.celeryproject.org/>.
- [32] J. Munir, “Exploring the boto interface for resource management in cloud infrastructures,” 2012.
- [33] H. P. Langtangen, *Python scripting for computational science*. Springer, 2006.
- [34] G. V. Rossum, “An introduction to python for unix/c programmers,” 1993.
- [35] M. Flower, “Uml, gota a gota,” tech. rep., México,, 1999.
- [36] I. Jacobson, “Object-oriented software engineering: A use case driven approach,” tech. rep., 1992.
- [37] Y. R. Amador, K. C. Santo, and D. M. Viñas, “Lógica del negocio de la empresa en una capa intermedia.,” vol. 23, no. 1, 2002.
- [38] “Mvc definicion y características.” <http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas>.
- [39] V. H. Gil, “Introducción a uml,”
- [40] R. S. Pessman, *Ingeniería de Software un enfoque práctico.5ta Edición*. 1997.
- [41] G. J. R. Ivar Jacobson, *El Proceso Unificado de Desarrollo de Software*. 2000.
- [42] R. S. Pressman, *Ingeniería del Software. Un enfoque práctico. Quinta Edicion*. 2001.
- [43] P. G. Pérez, *Principios básicos del desarrollo seguro. División de Seguridad Lógica de Germinus*. 2005.

## **Anexo A**

### **Primer apéndice**

#### **A.1. Entrevista**

Con respecto al las PACS:

1. ¿Dónde se almacenan las imágenes obtenidas en una institución de salud?
2. Si el almacenamiento es interno, en un servidor para cada institución:
  - ¿En caso que se sature el servidor dónde se almacenan las imágenes?
  - ¿Cómo se comunican con otras instituciones?
  - ¿Por cuánto tiempo almacenan las imágenes?¿Por qué ese tiempo?

Con respecto a los ambientes donde se encuentra desplegado el sistema:

1. ¿Cuál es la capacidad de almacenamiento del servidor?
2. ¿Con cuántos equipos de obtención de imágenes médicas cuenta cada institución?
3. ¿Cuánta información genera un estudio?
4. ¿Cuántos estudios por equipo se pueden realizar en un día de trabajo?
5. ¿Cuál es la velocidad de conexión a Internet?
6. ¿Anualmente cuánta información genera una institución de salud?
7. ¿En qué tiempo se satura la capacidad de almacenamiento del servidor?

## A.2. Características de Amazon

### A.2.1. Precios de almacenamiento

	Estándar	Redundancia Reducida	Glaciar
Primer TB/mes	\$0.0300 / GB	\$0.0240 / GB	\$0.010 / GB
Siguientes 49 TB/mes	\$0.0295 / GB	\$0.0236 / GB	\$0.010 / GB
Siguientes 450 TB/mes	\$0.0290 / GB	\$0.0232 / GB	\$0.010 / GB
Siguientes 500 TB/mes	\$0.0285 / GB	\$0.0228 / GB	\$0.010 / GB
Siguientes 4000 TB/mes	\$0.0280 / GB	\$0.0224 / GB	\$0.010 / GB
Más de 5000 TB/mes	\$0.0275 / GB	\$0.0220 / GB	\$0.010 / GB

Tabla A.1: Precios de almacenamiento.

### A.2.2. Precios de las solicitudes

	Precios
Solicitudes PUT, COPY, POST o LIST	\$0.005 por cada 1000 solicitudes
Solicitudes de restauración y archivado en Glaciar	\$0.05 por cada 1000 solicitudes
Continúa en la próxima página	

Solicitudes de eliminación	Grátis(Glaciari \$0,03/ GB)
GET y todas las demás solicitudes	\$0.004 por cada 10000 solicitudes
Restauraciones de datos en Glaciari	Grátis(Glaciari +5 % del almacenamiento medio mensual a \$0.011/ GB)

Tabla A.2: Precios de las solicitudes.




### A.2.3. Precios de transferencia de datos

	Precios
Transferencia ENTRANTE de datos a Amazon S3	
Todas las transferencias entrantes de datos	\$0.000 / GB
Transferencia SALIENTE de datos de Amazon S3 a Internet	
Primer GB/mes	\$0.000 / GB
Hasta 10 TB/mes	\$0.120 / GB
Siguientes 40 TB/mes	\$0.090 / GB
Siguientes 100 TB/mes	\$0.070 / GB
Siguientes 350 TB/mes	\$0.050 / GB
Siguientes 524 TB/mes	Mutuo Acuerdo
Siguientes 4 PB/mes	Mutuo Acuerdo
Superior a 5 PB/mes	Mutuo Acuerdo

Tabla A.3: Precios de transferencia de datos.



**A.3. Características de Dropbox, Google Drive y SkyDrive**

Proveedores	 Dropbox	 Google Drive	 SkyDrive
Almacenamiento Gratis	2GB	5GB	7GB
Continúa en la próxima página			

Precios de Mejora	<p><b>100GB</b> a \$9.99/mes o \$99/año</p> <p><b>200GB</b> a \$19.99/mes o \$199/año</p> <p><b>500GB</b> a \$49.99/mes o \$499/año</p>	<p><b>25GB</b> \$2.49</p> <p><b>100GB</b> \$4.99</p> <p><b>200GB</b> \$9.99</p> <p><b>400GB</b> \$19.99</p> <p><b>1TB</b> \$49.99</p> <p><b>2TB</b> \$99.99</p> <p><b>4TB</b> \$199.99</p> <p><b>8TB</b> \$399.99</p> <p><b>16TB</b> \$799.99</p> <p>Mejora de almacenamiento aplicada a Picasa, 25 GB adicionales para Gmail.</p>	<p><b>27GB</b> \$10/año</p> <p><b>57GB</b> \$25/ año</p> <p><b>107GB</b> \$50/ año</p>
Continúa en la próxima página			

Características	<p>Sincronización selectiva de carpetas.</p> <p>Seguimiento de eventos.</p> <p>Historial de versiones.</p> <p>Enlace para compartir.</p> <p>Integración con Facebook.</p>	<p>Sincronización selectiva de carpetas.</p> <p>Seguimiento de eventos.</p> <p>Historial de versiones.</p> <p>Configuración de los permisos compartidos.</p> <p>Comentarios sobre los archivos.</p> <p>Editor de documentos en línea.</p> <p>Edición simultánea de documentos.</p>	<p>Seguimiento de eventos.</p> <p>Historial de versiones.</p> <p>Configuración de los permisos compartidos.</p> <p>Comentarios sobre los archivos.</p> <p>Aplicaciones Web de Microsoft Office (Word, PowerPoint, Excel, OneNote).</p> <p>Edición simultánea de documentos.</p> <p>Acceso remoto a los archivos en el PC.</p>
Límite del archivo a la subida.	Ilimitado a través de aplicación de escritorio, 300 MB en el sitio web.	10 GB a través de aplicaciones de escritorio y sitios web.	2 GB a través de aplicaciones de escritorio y 300 MB a través de sitios web.
Continúa en la próxima página			

---

Seguridad	Dos pasos de verificación.  Código de Contraseña de cuatro dígitos para Mobiles.	Dos pasos de verificación a través de todos los servicios de Google.	Código de verificación para el acceso remoto a los archivos.
-----------	--	--	--

Tabla A.4: Características de Dropbox, Google Drive y SkyDrive.