

Universidad de las Ciencias Informáticas

Facultad 10



Gestión y almacenamiento de imágenes digitales en los medios de prensa cubanos.

*Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas.*

Autores: Dayanis Rivero Rodríguez.

Yaneisy Calvera La O.

Tutor: Ing. Renier Socarras Pérez.

Consultante: Ing. Gilberto Lissabet Hernández.

Ciudad de La Habana 2010

Declaración de autoría

Declaramos ser autores del presente trabajo de diploma y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de junio del año 2010.

Yaneisy Calvera La O

Dayanis Rivero Rodríguez

Renier Socarras Pérez

Las grandes obras son hechas no con la fuerza, sino con la perseverancia.

Samuel Johnson

Agradecimientos

Yaneisy

Agradezco a mi madre y mi padre, por apoyarme tanto, por ser mis guías, mi sostén y siempre estar ahí brindándome su amor infinito.

A mi hermana por todos sus consejos y el cariño que siempre me ha entregado.

A mi cuña, porque siempre me ha dado buenos consejos y fuerzas para seguir.

A mis tutores Renier y Gilberto por la ayuda prestada en todo momento.

A mis amigos que de una forma u otra han manifestado su apoyo siempre que lo he necesitado.

A mi novio Ernesto Julio por aguantar mis malacrianzas, por estar siempre a mi lado y quererme tanto.

A Olner y Ivan, porque siempre me han brindado su cariño, ayuda y apoyo incondicional.

A la familia Domínguez Rodríguez, porque desde que los conocí no han hecho más que ayudarme y preocuparse por mí.

A Tato, porque aunque me cambia de vez en cuando se que me quiere y está siempre a mi lado cuando más lo necesito.

A Dayan, por ser más que mi amigo mi hermano.

A Jose Alejandro, por siempre escucharme y ser tan buen amigo.

A Dunia, Pánfilo, Yumi y Julio, por compartir conmigo momentos tan bonitos, de haberlo sabido antes me hubiera integrado al clan desde sus inicios.

Agradezco además a Dayanis y su familia, por siempre ayudarme cuando lo he necesitado y por apoyarme en los momentos en que he estado sola.

Un agradecimiento especial a Pyxel, por permitirme graduarme hoy.

Dedicatoria

Yaneisy

Dedico esta tesis a mi madre y mi padre, quienes se merecen esto y mucho más.

A mi hermana, por ser un ejemplo de profesional a seguir.

A mi bebé Eric David, para que siempre esté orgulloso de mí.

A mi cuña, siempre tan sincero.

A “tu tío”, de parte de su Ingeniera.

A mi abuela Carmen, quien siempre me ha dicho que el tiempo empleado en el trabajo luego se retribuye con buenos resultados.

A mis tías Betty, Nancy, Aleida, Ida, Magalis y Maria, las quiero mucho.

A mi familia en general, a mi abuelita Angelita y a todos, todos, todos, todos.....

A Cary, Anita y Ana, por siempre creer en mí y ayudarme tanto.

La dedico además a mis compañeros del 10101, especialmente a los que ya no están pero siempre recordaré con mucho cariño.

Resumen

Los centros de prensa cubanos cuentan con grandes archivos de imágenes digitales, fruto de largos años de trabajo, que son de vital importancia conservar. Actualmente estas imágenes están perdiéndose, ya que estos centros no cuentan con un espacio centralizado y seguro donde guardar sus archivos de información.

En la Universidad de las Ciencias Informáticas (UCI) se desarrolló un sistema Web llamado Ecúmene Pyxel (Pyxel), para almacenar y publicar las imágenes digitales de los centros de prensa cubanos. Este sistema permite realizar la recopilación de las imágenes a través de la Web de forma limitada, lo cual resulta complejo al querer realizar el almacenamiento masivo de archivos de imágenes. Se necesita contar con una vía que permita subir grandes cantidades de imágenes a Pyxel de la manera más cómoda y eficiente posible. Por tanto, el objetivo primordial de este trabajo es informatizar la gestión y almacenamiento de grandes volúmenes de imágenes digitales en los medios de prensa cubanos haciendo uso del sistema Web Pyxel.

El presente trabajo ofrece una alternativa de solución, la cual consiste en la creación de un FTP¹ que se integrará a dicho sistema, para poder recuperar y centralizar los archivos de imágenes que poseen los centros de prensa cubanos.

Palabras claves: archivos de imágenes, sistema Web, informatizar, centralizar.

¹ Protocolo de Transferencia de Ficheros.

Tabla de contenido

Resumen	IV
Introducción	1
Capítulo I Fundamentación Teórica	5
1.1 Conceptos asociados.	5
1.1.1 Módulo.	5
1.1.2 Imagen Digital.	5
1.1.3 Metadato.	6
1.1.4 IPTC.	6
1.1.5 Metadatos IPTC.	6
1.1.6 FTP.	6
1.1.7 Sistema Web.	7
1.1.8 PostgreSQL.	7
1.1.9 PgAdmin.	7
1.1.10 Licencia GPL.	7
1.1.11 Licencia Freeware.	7
1.2 Panorámica.	7
1.3 Herramientas para realizar la validación de imágenes digitales.	9
1.4 Metodología, lenguaje y herramientas a utilizar.	12
1.4.1 Metodología de desarrollo.	12
1.4.2 Python.	14
1.4.3 Geany.	16
1.4.4 Herramienta de Modelado UML.	16
1.4.5 Herramienta CASE.	17
Capítulo II Descripción de la solución propuesta.	19
2.1 Descripción del problema.	19
2.2 Solución propuesta.	20
2.3 Sistema Pyxel.	20
2.3.1 Estructura de Pyxel.	20
2.3.2 Principios de la arquitectura de Pyxel.	21
2.3.2.1 Sistema distribuido.	21

Tabla de contenido

2.3.2.2 Cero Acoplamiento.	22
2.3.2.3 REST y RDF.	23
2.3.3 Metodología de desarrollo y herramienta de modelado.	23
2.3.4 ORB de Pyxel.	23
2.3.5 Servicios en Pyxel.	24
2.3.6 Características de las imágenes digitales.	24
2.3.7 Subiendo imágenes digitales por la Web.	24
2.3.8 Sistema de Ayuda al usuario.	25
2.4 Características de los FTP.	25
2.4.1 Transferencia de archivos en FTP.	27
2.4.2. Tipos de acceso.	27
2.4.3 Modos de conexión del cliente FTP.	28
2.5 Clientes FTP.	29
2.6 Servidores FTP.	31
2.7 Sistema Gestor de Base de Datos.	32
2.8 PgAdmin III.	33
2.9 Modelo de Dominio.	34
2.9.1 ¿Por qué hacer Modelo de Dominio?	34
2.9.2 Conceptos fundamentales del Modelo de Dominio.	34
2.10 Especificación de los requisitos de software del sistema.	35
2.11 Definición y descripción de los Casos de Uso del Sistema.	36
2.11.1 Diagrama de Caso de Uso del Sistema.	38
2.11.2 Descripción de Caso de Uso del Sistema.	38
Capítulo III Análisis y diseño de la solución propuesta.	41
3.1 Análisis y diseño.	41
3.1.1 Modelo de Análisis.	41
3.1.2 Diagramas de Interacción.	43
3.1.3 Modelo de Diseño.	44
Capítulo IV Implementación y prueba.	47
4.1 Diagrama de componentes.	47

Tabla de contenido

4.1.1 Descripción de módulos y componentes.	48
4.2 ¿Cómo se realiza el almacenamiento de las imágenes desde el FTP hacia Pyxel?	51
4.2.1 Autenticar usuario.	51
4.2.2 Gestionar envío de imágenes.	52
4.3 Modelo de prueba.....	54
4.3.1 Pruebas aplicadas.....	57
Conclusiones	63
Recomendaciones	64
Referencias Bibliográficas.....	65
Bibliografía.....	67
Glosario de Términos.....	69

Introducción

El continuo desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) constituye la base para el progreso de las distintas esferas de la sociedad. Las mismas han determinado la evolución y supervivencia de las personas, pues proporcionan disímiles beneficios en la salud, la educación y la economía a nivel mundial.

Cuba ha estado inmersa en el profundo y novedoso proceso de transformaciones tecnológicas a partir de la cual se emprenden nuevos programas para elevar el nivel cultural tecnológico de la sociedad. Los medios de comunicación, especialmente los medios de prensa, se han visto inmersos en constantes cambios, destinados al mejoramiento y optimización de los procesos de producción, para proporcionarles a sus trabajadores un mejor servicio. La inmediatez de publicación es una de las necesidades que provocó llevar a cabo la informatización de la prensa en Cuba. Por esta razón, el primer cambio efectuado en los centros de prensa es la digitalización de la información, donde la mayoría de estos centros cuentan con su portal Web, al cual se puede acceder y encontrar en formato digital todas las noticias que día a día brindan a la población. Las facilidades que ofrecen estos cambios tecnológicos permiten a los trabajadores realizar su labor con mayor rapidez y eficiencia.

A pesar de que este proceso de informatización se está llevando a cabo arduamente, aún persisten problemas que son necesarios resolver.

Los centros de prensa guardan sus archivos de imágenes digitales en distintos medios de almacenamiento: como discos compactos, dispositivos de almacenamiento portátiles y ordenadores independientes, gran parte de estos medios son propiedad de los periodistas que laboran en la institución. Esto provoca que en muchas ocasiones las imágenes no estén disponibles para ser utilizadas en un momento determinado, pues no se encuentran en un lugar centralizado. Los soportes digitales donde son almacenadas, no constituyen un medio seguro y confiable, por lo que existe cierto descontrol y pérdida de las mismas.

Esos volúmenes de imágenes digitales, pertenecientes a los centros de prensa cubanos, son necesarios procesarlos en grandes paquetes y no existe un software competente para realizar dicha tarea.

Un claro ejemplo donde se ve reflejada esta situación es en la revista Bohemia, que cuenta con más de cien años de información en fotografías de toda índole, las cuales no pueden ser utilizadas por no existir una forma rápida y viable de recuperarlas, lo cual contribuye a que gran parte de ellas se hayan perdido completamente.

Otro de los problemas que afecta en gran medida a la prensa y sus periodistas, es la publicación de imágenes. Actualmente cuando un periodista necesita realizar alguna publicación, debe dirigirse al Centro de Información para la Prensa (CIPRE), portando las imágenes en un dispositivo de almacenamiento y hacer entrega del mismo a los informáticos, quienes se encargan de llevar a cabo la publicación. Este proceso genera pérdida de tiempo al difundir la información, pues la labor de estos trabajadores se intensifica debido al número de imágenes que se van a publicar y demás actividades que están encargados de efectuar. Dicha situación causa molestias a los periodistas, ya que deben esperar mucho tiempo para realizar la entrega de las imágenes.

Otros factores que intervienen negativamente cuando se desea realizar publicaciones es que si el personal capacitado para efectuarla se ausenta a su puesto de trabajo, no se realizaría la recogida de la información en el momento requerido y si un periodista que labore en otra provincia quiere publicar imágenes a nivel nacional, no cuenta con la posibilidad de hacerla llegar a tiempo al CIPRE.

Todos estos problemas surgen debido a que:

- Los medios de prensa cubanos no disponen de un espacio centralizado y seguro donde se puedan almacenar, procesar y recuperar grandes volúmenes de imágenes digitales.
- Los periodistas no cuentan con un procedimiento rápido y viable para realizar la entrega de las imágenes que posteriormente se publicarán.

De la situación anterior se deduce el siguiente **problema científico de la investigación**.

¿Cómo optimizar la gestión y almacenamiento de los grandes volúmenes de imágenes digitales de los medios de prensa cubanos?

Se define como **idea a defender** la siguiente: con la implementación de un módulo para la transferencia de imágenes, se logrará optimizar la gestión y almacenamiento de los grandes volúmenes de imágenes digitales en los medios de prensa cubanos.

Para lograr la gestión y almacenamiento de los archivos de imágenes digitales, es necesario sistematizar en el área donde se enmarca el **objeto de estudio**, constituido por la Gestión de imágenes digitales en los medios de prensa cubanos y el **campo de acción** se centra en la Gestión de imágenes digitales en el CIPRE.

Teniendo en cuenta lo planteado anteriormente el **objetivo general** es:

- Informatizar la gestión y almacenamiento de grandes volúmenes de imágenes digitales en los medios de prensa cubanos haciendo uso del sistema Pyxel².

De este objetivo general se derivan los siguientes **objetivos específicos**:

- Analizar el estado del arte de la gestión y almacenamiento de las imágenes digitales.
- Definir la estrategia a seguir para organizar el proceso de gestión y almacenamiento de grandes volúmenes de imágenes digitales de los medios de prensa cubanos.
- Desarrollar un módulo para el sistema Pyxel que soporte la estrategia propuesta.
- Integrar el módulo al sistema Pyxel.

Las **tareas** que se proponen para dar cumplimiento a los objetivos son:

- Realizar un levantamiento sobre la gestión y almacenamiento de las imágenes digitales.
- Desarrollar el proceso a seguir para realizar la gestión, validación y almacenamiento de grandes volúmenes de imágenes digitales de los medios de prensa cubanos.
- Implementar el módulo para gestionar y almacenar las imágenes digitales en los medios de prensa cubanos utilizando el lenguaje de programación Python.
- Realizar pruebas a la solución propuesta.

Métodos teóricos utilizados.

- **Analítico-Sintético:** con el objetivo de analizar las teorías y documentos, para extraer los conceptos esenciales relacionados con el objeto de estudio.
- **Inductivo-deductivo:** para arribar a conclusiones que nos permitan implementar la solución,

² Sistema Web desarrollado en la UCI para gestionar el almacenamiento, publicación y recuperación de imágenes digitales a través de la Web.

partiendo de la información adquirida.

Estructura del contenido.

El trabajo consta de una introducción, cuatro capítulos bien estructurados, conclusión, recomendaciones, referencias bibliográficas y bibliografías consultadas.

Capítulo I. Fundamentación Teórica.

En este capítulo se exponen conceptos a tener en cuenta para lograr una mejor comprensión del contenido, se realiza un estudio sobre los sistemas para la gestión de imágenes existentes en el mundo, se describen las herramientas para realizar la validación de las imágenes digitales y se escoge la metodología y lenguaje a utilizar para la creación del producto.

Capítulo II. Descripción de la solución propuesta.

En este capítulo se describe la propuesta de solución al problema científico. Se escogen las herramientas necesarias para crear la solución y se realiza la descripción de los casos de usos, así como la descripción de los requerimientos que debe cumplir.

Capítulo III. Análisis y diseño de la solución propuesta.

En este capítulo se realiza el análisis y diseño de la solución propuesta que se integrará con el sistema Pyxel.

Capítulo IV. Implementación y Prueba.

En este capítulo se realiza la implementación y prueba de la solución propuesta, tomando como base los resultados que se presentan en los capítulos anteriores.

Capítulo I Fundamentación Teórica

Con el desarrollo de este capítulo se pretende alcanzar conocimientos sobre los archivos de imágenes digitales y los conceptos asociados a estos; que permitirán desempeñar la parte práctica con la calidad requerida. Se ofrece una panorámica general relacionada con la existencia de programas utilizados para el almacenamiento y gestión de imágenes digitales, con el objetivo de realizar un estudio que permita conocer las funcionalidades que sirvan de aporte a la solución. Además, se explica la metodología, lenguaje de programación y herramientas a utilizar.

1.1 Conceptos asociados.

En el presente epígrafe se muestran los conceptos que son necesarios conocer para lograr una correcta comprensión del contenido de la tesis.

1.1.1 Módulo.

Un módulo es un segmento, conjunto de instrucciones, subalgoritmo o procedimiento, que puede definirse dentro de un algoritmo con el fin de ejecutar una tarea específica y puede ser llamado o invocado desde el algoritmo principal cuando sea necesario [1].

1.1.2 Imagen Digital.

Las imágenes digitales son fotos electrónicas tomadas de una escena o escaneadas de documentos - fotografías, manuscritos, textos impresos e ilustraciones. Se realiza una muestra de la imagen digital y se confecciona un mapa de ella en forma de cuadrícula de puntos o elementos de la figura (píxeles). A cada píxel se le asigna un valor tonal (negro, blanco, matices de gris o color), el cual está representado en un código binario (ceros y unos). Los dígitos binarios (bits) para cada píxel son almacenados por una computadora en una secuencia, y con frecuencia se los reduce a una representación matemática (comprimida). Luego la computadora interpreta y lee los bits para producir una versión analógica para su visualización o impresión [2].

La imagen digital es un producto del desarrollo de la informática que tiene como antecesor a la fotografía, (que toma como punto de partida un objeto del mundo real) y a la pintura, (donde la imagen ha sido creada por un artista) [3].

Capítulo I Fundamentación Teórica

Abordados estos conceptos se puede definir como imagen digital a toda imagen que sea generada por un ordenador o creada a través de un medio electrónico, cómo una cámara o escáner.

1.1.3 Metadato.

Los metadatos son datos estructurados que describen información, describen el contenido, la calidad, la condición y otras características de los datos [4].

Los metadatos son información acerca de algo o bien que contiene. El “algo” puede ser un objeto físico como un libro o un envase, o bien un contenido digital como una página web en línea [5].

Finalmente, se puede arribar a la conclusión de que los metadatos no son más que datos que describen a otros datos.

1.1.4 IPTC³.

IPTC es un organismo internacional dedicado a promover estándares que permiten el intercambio de información entre las principales agencias de noticias del mundo. Una de estas normativas es la definición de unos campos de información que acompañan a cada imagen, y que permiten guardar datos sobre el autor, fecha y lugar de la toma, etc [6].

1.1.5 Metadatos IPTC.

Es un estándar para incluir los datos descriptivos de una fotografía o imagen digital. Estos se escriben dentro del fichero de la imagen digital.

1.1.6 FTP.

El FTP es la herramienta que te permite, a través de la red, copiar ficheros de un ordenador a otro. Y ello, sin importar en absoluto donde están localizados estos ordenadores, ni si usan o no el mismo sistema operativo [7].

FTP es un protocolo de la familia TCP/IP⁴ utilizado para transferir ficheros entre ordenadores que ejecutan TCP/IP. Este protocolo permite realizar la transferencia de archivos y manipular directorios en máquinas

³ International Press Telecommunications Council.

remotas, aunque no está diseñado para permitir el acceso a otro ordenador para ejecutar programas, resulta la mejor utilidad para manipular ficheros.

1.1.7 Sistema Web.

Un sistema Web es una aplicación Web desarrollada para satisfacer necesidades específicas y resolver problemas mediante el análisis de la lógica de los procesos o también llamado "lógica de negocios", automatizando procesos en línea [8].

1.1.8 PostgreSQL.

Es un Sistema Gestor de Base de Datos Relacionales Orientadas a Objetos.

1.1.9 PgAdmin.

PgAdmin es una herramienta de propósito general para diseñar, mantener y administrar las bases de datos de Postgres [9].

1.1.10 Licencia GPL.

GPL es el acrónimo de Licencia Pública General de GNU. Fue creada por la Free Software Foundation (FSF) y está orientada a proteger la libre distribución, modificación y uso de software libre.

1.1.11 Licencia Freeware.

El término Freeware es utilizado para los programas de software que se distribuyen sin costo y se encuentran disponibles para ser utilizados por tiempo ilimitado. Incluye una licencia de uso, la cual permite que se pueda distribuir pero con las restricciones de no modificar la aplicación, no venderla y dar cuenta de su autor. Se puede desautorizar el uso en compañías con fines comerciales, o requerir pagos si se le va a dar uso comercial.

1.2 Panorámica.

El surgimiento y desarrollo de Internet ha posibilitado a las personas acceder fácilmente a la información desde cualquier parte del mundo. Esto representa un gran desafío para los desarrolladores de sistemas

⁴ Protocolo de control de transmisión / Protocolo de Internet: es un conjunto de protocolos de red en los que se basa Internet y que permiten la transmisión de datos entre redes de computadoras

Capítulo I Fundamentación Teórica

para el trabajo con la Web, ya que los avances en la tecnología demandan cada vez aplicaciones más rápidas y ligeras.

Hoy en día las formas de almacenamiento de información están sufriendo un proceso de digitalización constante. Las imágenes digitales, por supuesto, no escapan a este cambio. Miles de ellas son producidas por distintos medios y son difundidas ya sea por impresión en papel, transmisión electrónica y almacenamiento en sistemas Web diseñados para su publicación.

Los sistemas Web para el almacenamiento y publicación de imágenes digitales han tenido una gran aceptación por los usuarios de Internet, pues cubren la necesidad que tienen de organizar sus imágenes. Existen muchos sistemas que brindan estos y otros servicios que los hacen más cómodos y eficaces.

En la búsqueda de respuestas a las necesidades que presentan los centros de prensa de todo el país, se estudiaron diferentes sistemas para la gestión y almacenamiento de imágenes digitales. El análisis se realizó con el objetivo de conocer su funcionamiento y las características que pudieran servir de aporte para la implementación del módulo que requiere Pyxel.

A continuación se ofrece una panorámica general de los sistemas para el almacenamiento y gestión de imágenes digitales más utilizados, a los cuales se les realizó el estudio.

Dr. Upload.

Es un sistema de almacenamiento de imágenes digitales gratuito. Permite almacenar imágenes de hasta 10 MB, los formatos que soporta son JPG, GIF, PNG, BMP y PSD. Su uso es sencillo y simple, solo se debe seleccionar el archivo, subirlo y luego el sistema genera una serie de enlaces para compartirla. Una de sus limitaciones es que no cuenta con la posibilidad de redimensionar o subir imágenes digitales vía URL⁵.

Picasa.

Es un sistema que permite organizar, editar y subir fotos de forma rápida y sencilla a través de la Web. Organiza las fotos en carpetas y álbumes. Se puede modificar el nombre, la fecha en la que se produjo,

⁵ Uniform Resource Locator: protocolo para especificar direcciones en Internet.

Capítulo I Fundamentación Teórica

lugar donde se tomaron las fotos y activar y desactivar distintas funciones. Existen versiones del mismo para la mayoría de los sistemas operativos existentes.

Flypicture.

Es un sistema de almacenamiento en línea para imágenes digitales y archivos. Entre sus funcionalidades posibilita compartir y almacenar de forma ilimitada. El tamaño máximo del archivo que permite subir es de 3 MB.

Arkiva.

Es un servicio de almacenamiento en línea gratuito que ofrece 1 Gb para almacenar y compartir cualquier tipo de fichero digital. Además, permite la edición en línea de documentos e impresión de fotos.

Los sistemas analizados no cuentan con funcionalidades que permitan efectuar el almacenamiento de grandes volúmenes de imágenes digitales, pues han sido creados para realizar trabajos sencillos y almacenar pocas imágenes a través de la Web. Concluyendo que, ninguno de los sistemas analizados presentan las características ideales para ser utilizados como guía en la solución. Se decide realizar la implementación de un nuevo producto que responda a las necesidades de los centros de prensa cubanos.

1.3 Herramientas para realizar la validación de imágenes digitales.

Las imágenes que serán almacenadas en el sistema Pyxel tienen que cumplir con el estándar propuesto por sus desarrolladores, el cual consiste en que deben contener los metadatos IPTC que permitan su identificación.

Validar las imágenes digitales para luego subirlas a Pyxel es el proceso que se lleva a cabo para incluirle los metadatos IPTC, haciendo uso de herramientas que permitan su manipulación. La inclusión de estos metadatos en las imágenes se realiza con el objetivo de facilitar su búsqueda una vez almacenadas en el sistema.

Existen herramientas para la creación, implementación y gestión de metadatos. Estas herramientas van desde la simple creación del metadato a complejas herramientas que integran todo el proceso de organización, descripción y búsqueda de información, las cuales pueden ser software libre o comercial.

Capítulo I Fundamentación Teórica

Las siguientes aplicaciones poseen características que serán analizadas, para optar por una de ellas para realizar la validación de las imágenes digitales.

Adobe Photoshop.

Adobe Photoshop es un software diseñado para la creación gráfica, edición de imágenes digitales y retoques fotográficos. Ofrece múltiples herramientas con funciones y capacidades de gran calidad incluyendo la automatización de tareas.

A pesar de que Photoshop es un programa profesional con múltiples funcionalidades, excelentes herramientas para la edición de imágenes digitales, una gran biblioteca de efectos, filtros y capas, requiere de un equipo con mucha capacidad, pues ocupa un considerable espacio en disco. Si el ordenador donde se encuentra instalado, está muy lleno o no es muy potente, puede darse el caso de que no trabaje bien. Además, es un software privativo por lo que se debe pagar para obtener las actualizaciones.

Xmp Manager.

Xmp Manager es una herramienta gráfica para administrar metadatos XMP en GNU/Linux. Actualmente es un software experimental y las características implementadas son mínimas, por lo que puede que existan fallos cuando vaya a utilizarse.

Xmp Manager es de licencia GPL pero soporta y permite editar solamente algunos de los metadatos que fueron seleccionados por los desarrolladores de Pyxel.

PhotoMe.

PhotoMe es una poderosa herramienta para mostrar y editar los metadatos de las imágenes. Gracias a su buen diseño y manejo intuitivo, es posible analizar y modificar los metadatos EXIF e IPTC-NAA. Es una utilidad gratuita para ver toda la información adicional de una foto e incluso de la cámara con que fue tomada. Esto es posible si la foto contiene la información EXIF. PhotoMe lee los formatos JPG, TIFF y RAW.

PhotoMe es de licencia Freeware y no permite editar todos los metadatos que requieren las imágenes que serán procesadas por Pyxel.

Capítulo I Fundamentación Teórica

Mapivi.

Mapivi es una herramienta de gestión de fotografías con un enfoque en imágenes jpeg. Admite la adición, edición y búsqueda de IPTC/IIM, datos EXIF y comentarios jpeg. Mapivi es de licencia GPL, permite editar algunos de los metadatos que requieren las imágenes que Pyxel procesará, además es una herramienta que necesita que se le incorporen nuevas funcionalidades que lo hagan más comprensible y cómodo para el uso de los usuarios.

GIMP.

GIMP es una herramienta de manipulación fotográfica multiplataforma. Esta aplicación está diseñada para operar con las imágenes digitales, posibilitando su creación, composición y retoques fotográficos. Ofrece otras funcionalidades, incluyendo la creación de gráficos, cambio de tamaño, modificación de colores y edición avanzada de vídeo. GIMP es de licencia GPL pero solamente soporta metadatos IPTC, no permite la edición de los mismos.

XnViewMP.

XnViewMP es un visor de imágenes compacto, rápido y compatible con una enorme cantidad de formatos gráficos. Su organización es bastante tradicional, con una vista de miniaturas, un árbol de directorios y un panel inferior con pestañas. Las imágenes se pueden clasificar con etiquetas, categorías o puntuaciones. XnViewMP es de licencia Freeware y permite la edición de todos los metadatos IPTC que necesitan las imágenes a procesar por Pyxel.

Herramienta	Licencia	Trabajo con Metadatos IPTC
XnViewMP	Freeware	Edita y soporta todos
GIMP	GPL	Soporta
Mapivi	GPL	Edita y soporta algunos
PhotoMe	Freeware	Soporta
Xmp Manager	GPL	Soporta
Photoshop	Privativo	Edita y soporta todos

Tabla 1.1 Comparación de herramientas para validación de imágenes.

Selección de la herramienta.

Las imágenes a ser procesadas por Pyxel deben tener incluidos (como anteriormente se expuso), los metadatos IPTC que fueron seleccionados por los desarrolladores de la aplicación, por cumplir con las expectativas que se requieren, para que el sistema las procese y publique adecuadamente. En este proceso de selección se tuvo en cuenta que la herramienta debe permitir incluir y editar estos metadatos IPTC. De esta forma, quedan descartados todos aquellos sistemas que solamente editan y/o soportan otro tipo de metadato.

Los centros de prensa cuentan con ordenadores que poseen poco espacio en disco. Atendiendo a las características de estas computadoras, los programas a ser instalados deben ser pequeños y consumir pocos recursos para poder correr eficientemente.

Teniendo en cuenta esto y tras realizar un estudio sobre las funcionalidades que brindan dichas herramientas, se pudo arribar a la conclusión de que XnViewMP es el programa ideal para utilizar, pues la inclusión de metadatos en las imágenes que maneja, se hace de forma fácil y rápida. Además, la validación de imágenes digitales que se realizará antes de subirlas a Pyxel es sencilla, por lo que no es necesario instalar programas pesados para realizar operaciones tan elementales que un software ligero como este es capaz de hacer. XnViewMP distribuido bajo licencia Freeware es un programa de adquisición gratuita, disponible para su uso por tiempo ilimitado y aunque no se puede modificar, ni vender la aplicación, resulta muy útil, pues solo se utilizará para realizar la inclusión de los metadatos a las imágenes que posteriormente se procesarán.

1.4 Metodología, lenguaje y herramientas a utilizar.

Definir las herramientas, el lenguaje y la metodología a seguir para la creación de un sistema informático, es el primer paso que se realiza y sin duda el más importante para llevar a cabo la implementación de un sistema de forma coherente y sistemática.

1.4.1 Metodología de desarrollo.

Una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar un producto y que identifica claramente las etapas en las que se divide, las tareas que se

Capítulo I Fundamentación Teórica

llevarán a cabo, los requerimientos que deben cumplirse, las herramientas y técnicas que se emplearán y la gestión y control del proyecto en general.

Para lograr un control estricto de las acciones que se ejecutarán, se hará uso de la metodología Rational Unified Process (RUP), pues permite llevar a cabo el proceso de desarrollo práctico, brindando la base de los conocimientos, un lenguaje de modelado y una noción de cómo desarrollar el software. Esta le permite al desarrollador obtener una definición acertada del sistema desde un inicio para evitar reconstrucciones innecesarias en etapas posteriores.

¿Qué es RUP?

RUP es una metodología de desarrollo de software que intenta integrar todos los aspectos a tener en cuenta durante todo el ciclo de vida del software, con el objetivo de hacer abarcables tanto pequeños como grandes proyectos software. Además, Rational proporciona herramientas para todos los pasos del desarrollo así como documentación en línea para sus clientes [10].

Las características principales de RUP son:

- **Guiado o manejado por casos de uso:** la razón de ser de un sistema software, es servir a usuarios, ya sean humanos u otros sistemas; un caso de uso es una facilidad que el software debe proveer a sus usuarios. Los casos de uso reemplazan la antigua especificación funcional tradicional y constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo, incluyendo el diseño, la implementación y las pruebas del sistema.
- **Centrado en la arquitectura:** la arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros, por plataformas software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. Los casos de uso guían el desarrollo de la arquitectura y la arquitectura se realimenta en los casos de uso, los dos juntos permiten conceptualizar, gestionar y desarrollar adecuadamente el software.
- **Iterativo e incremental:** para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini proyecto, cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo.

RUP está constituido por cuatro fases que representan el ciclo de desarrollo del software y nueve flujos de trabajo, seis ingenieriles y tres de soporte que intervienen dentro de cada fase.

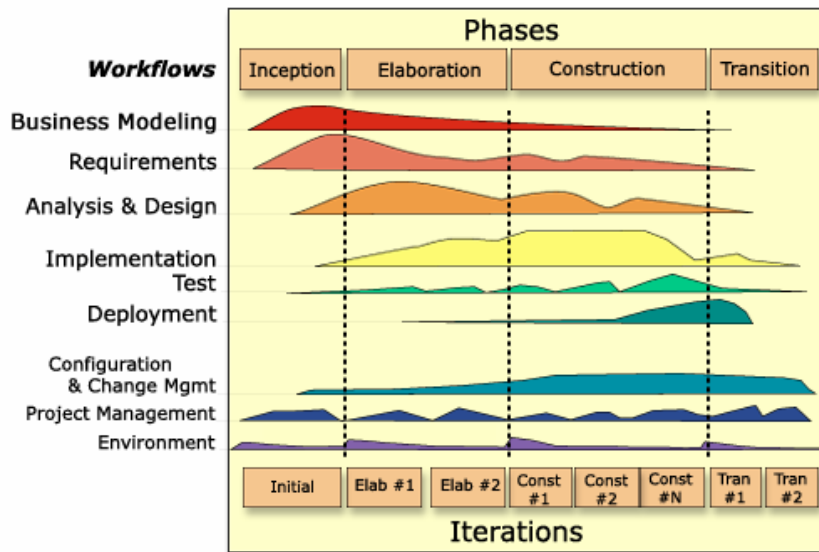


Figura 1.1 Fases, Flujos e iteraciones de RUP

1.4.2 Python.

El lenguaje de programación que será utilizado para la implementación de la funcionalidad que se quiere añadir a Pyxel, es Python, debido a que el sistema Pyxel desde sus inicios fue creado haciendo uso del mismo, por las facilidades que ofrece, por ser libre y por otras características que se mencionan a continuación.

¿Qué es Python?

Python es un lenguaje de programación creado por Guido Van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses "Monty Python". Es un lenguaje similar a Perl, pero con una sintaxis muy amplia y que favorece un código legible. Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos [11].

Capítulo I Fundamentación Teórica

Python tiene muchas características de los lenguajes compilados, por lo que se puede decir que es semi interpretado. En Python como en otros lenguajes el código fuente se traduce a un pseudo código llamado bytecode que genera archivos .pyc o pyo, que se ejecutarán en sucesivas ocasiones.

Tipado dinámico.

La característica de tipado dinámico se refiere a que no es necesario declarar el tipo de dato que contendrá una variable, sino que su tipo se determina en tiempo de ejecución, dependiendo del valor que se le asigne.

Fuertemente tipado.

No está permitido tratar una variable como si fuera de un tipo distinto al que tiene, por lo que es necesario convertir de forma explícita dicha variable al nuevo tipo antes de utilizarla.

Multiplataforma.

El intérprete de Python está disponible para muchas plataformas (UNIX, Linux, Windows, Mac OS, etc.), por lo que si no se utilizan librerías específicas de cada plataforma, el programa podrá correr en todos los sistemas sin grandes cambios.

Python combina un remarcable poder con una sintaxis muy clara. Tiene interfaces a muchas llamadas al sistema y bibliotecas, así como también a varios sistemas de ventanas, y es extensible en C o C++. También es utilizable como un lenguaje de extensión para aplicaciones que necesiten interfaces programables. Finalmente, Python permite dividir los programas en módulos reutilizables desde otros programas creados con Python. Además, posee una colección de módulos estándar que se pueden utilizar como guía para comenzar a aprender Python.

Python posee otras características:

- Propósito general: se pueden crear todo tipo de programas.
- Interpretado: no se debe compilar el código antes de su ejecución.
- Orientado a objetos: la programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.
- Funciones y librerías: dispone de muchas funciones incorporadas al lenguaje y existen muchas

librerías que se pueden importar en los programas para tratar temas específicos.

- Sintaxis clara: tiene una sintaxis muy visual, gracias a la notación con márgenes de obligado cumplimiento que posee. Permite la escritura de programas cuya lectura resulta más fácil.

1.4.3 Geany.

Geany es un editor de texto desarrollado en GTK2 con características básicas de un entorno de desarrollo integrado (IDE). Fue desarrollado para proporcionar un pequeño y rápido IDE, que tiene sólo unas pocas dependencias de otros paquetes. Es compatible con muchos tipos de archivos. Geany solo requiere las bibliotecas de ejecución GTK2. Incluye una terminal completamente funcional con la que se puede navegar por carpetas o ejecutar partes de código. También cuenta con compilador y ejecuta algunos tipos de archivos, como por ejemplo los scripts.

Algunas de las características que posee:

- Resaltado de sintaxis.
- Código desplegable.
- Cierre automático de etiquetas XML y HTML.
- Interfaz simple dividida en paneles.
- Permite compilar y ejecutar.
- Ofrece soporte de internacionalización.
- Permite crear y administrar proyectos.
- Incorpora un panel lateral que permite acceder a partes específicas del código que se está editando, como variables y funciones.

1.4.4 Herramienta de Modelado UML.

Lenguaje Unificado de Modelado (UML) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Permite la modelación de sistemas con tecnología orientada a objetos, es un lenguaje consolidado, fácil de aprender y permite una comunicación fluida entre los diversos actores. Un modelo UML describe lo que supuestamente hará el sistema, pero no dice cómo implementar dicho sistema.

Funciones:

- **Visualizar:** permite expresar de una forma gráfica un sistema, de manera que otro lo pueda entender.
- **Especificar:** permite especificar cuáles son las características de un sistema antes de su construcción.
- **Construir:** a partir de los modelos especificados se pueden construir los sistemas diseñados.
- **Documentar:** los propios elementos gráficos sirven como documentación del sistema desarrollado.

1.4.5 Herramienta CASE⁶

Las herramientas de modelado visual permiten representar un producto de software de forma completa, a través de diagramas que se desarrollan durante los ciclos de vida del proyecto. Una de las más usadas actualmente por las facilidades que ofrece es el Visual Paradigm.

Visual Paradigm

Conociendo que RUP utiliza un único lenguaje de modelado UML se hará uso del Visual Paradigm como herramienta CASE para el modelamiento UML. Esta herramienta profesional soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. Entre sus principales características se destaca por ser muy potente y de fácil uso e instalación, permite dibujar todo tipo de diagramas UML, así como generar código fuente desde dichos diagramas.

Visual Paradigm incluye los objetos más recientes de UML, como diagramas de casos de uso, de clases, componentes, etc. Ofrece soporte para Rational Rose y permite generar reportes y documentación en HTML y PDF.

Principales características:

- Producto de calidad.
- Fácil de instalar y actualizar.
- Soporta varios idiomas.

⁶ Ingeniería de Software Asistida por Ordenador: son diversas aplicaciones informáticas destinadas a aumentar la productividad del software.

Capítulo I Fundamentación Teórica

- Licencia gratuita y comercial.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Disponibilidad en múltiples plataformas.
- Diseño centrado en casos de uso y enfocado al negocio.

En este capítulo se ha cumplido con el objetivo principal, mostrar de forma general los aspectos a tener en cuenta para la implementación de lo que se quiere.

Se hace mención de los sistemas para la gestión de imágenes digitales más utilizados en el mundo, los cuales no han sido creados para uso de instituciones que contengan grandes volúmenes de imágenes digitales, ya que no cuentan con una funcionalidad que permita gestionar estos archivos. Por esta razón, no pueden ser tomados como guía, para implementar la funcionalidad que permitirá a Pyxel conservar las imágenes digitales existentes en los medios de prensa cubanos. También se describen algunos de los programas para el trabajo con los metadatos de las imágenes, obteniendo como resultado la selección del que será utilizado para realizar la validación. Se cuenta también con la metodología de desarrollo y demás herramientas que facilitarán la creación del producto.

Capítulo II Descripción de la solución propuesta

Actualmente en Cuba es necesario hacer uso del software libre, para lograr una completa independencia de las compañías que crean programas de software propietarios. Es muy importante encontrar alternativas libres que garanticen más calidad, fiabilidad y eficiencia que las propietarias. Por esta razón, en su mayoría las herramientas que se utilizarán para la solución de la problemática serán las que posean como principal característica licencia GPL.

En el presente capítulo se define el proceso de selección de la herramienta, características y funcionamiento. De igual forma se exponen los requisitos con los que debe contar la solución y se realiza una descripción detallada de los casos de uso que la conforman.

2.1 Descripción del problema.

En el capítulo anterior se realizó una profunda comparación de las herramientas existentes en el mundo, que permiten realizar la gestión de imágenes digitales y se arribó a la conclusión de que ninguna posee funcionalidades o características, que sirvan de ejemplo para construir la solución a la problemática por la cual surge esta tesis. Por esta razón, se mantiene vigente la necesidad que tienen los centros de prensa y los periodistas de contar con una herramienta que permita guardar archivos de imágenes digitales.

En la UCI se desarrolló un sistema Web llamado Pyxel (mencionado anteriormente), que permite gestionar el almacenamiento, publicación y recuperación de las imágenes digitales. Este sistema actualmente está siendo probado por sus desarrolladores y permite realizar la subida de no más de cinco imágenes a la vez a través de la Web, lo cual implica que de ser utilizado por los centros de prensa y los periodistas les ocasione molestias al intentar almacenar grandes archivos de imágenes digitales, pues realizar esta tarea les consumiría mucho tiempo.

Se propone encontrar una alternativa que solucione los problemas que afectan a la prensa y a los periodistas en cuanto al almacenamiento de las imágenes digitales, dicha alternativa será integrada al sistema Pyxel, el cual será implantado en el CIPRE. El objetivo de la misma es proporcionar una vía por la cual se pueda realizar la subida de muchas imágenes a dicho sistema en el menor tiempo posible. Se

Capítulo II Descripción de la solución propuesta

procede a realizar una valoración sobre las formas de transferir información para luego plantear la solución.

2.2 Solución propuesta.

Antes de que existiera la red, la transferencia de archivos se realizaba utilizando un dispositivo de almacenamiento para llevar la información de un ordenador hacia otro. Con el desarrollo de las TIC se ha ido alcanzando una mayor cultura en las formas de comunicación y actualmente la transferencia de ficheros puede realizarse a través de correo electrónico, por conexión entre computadoras y servicios de entrega en línea para uso de los usuarios. Estas formas de traslado de archivos son muy eficientes cuando:

- la información no es de gran tamaño.
- la red es local.
- existe acceso a Internet.

Teniendo en cuenta que los centros de prensa cubanos necesitan un espacio centralizado y seguro para recopilar los grandes volúmenes de imágenes digitales que poseen, no resulta eficaz realizar la transferencia de la información al sistema Pyxel de la forma antes mencionada. Por esta razón, se hará uso del servicio FTP como alternativa de solución al problema, pues es ideal para transferir grandes bloques de datos por la red, el tipo de conexión que utiliza es directa, es sencillo y cómodo en su forma de utilización e independiente del sistema operativo que utilice el ordenador. Concluyendo así, la propuesta para dar solución a la problemática planteada anteriormente es la creación de un módulo que permita subir grandes cantidades de imágenes vía FTP al sistema Pyxel, quien las procesará para posterior utilización por parte de los periodistas. El módulo a desarrollar se integrará a dicho sistema, con el objetivo de aportar una solución eficiente y satisfactoria a la prensa cubana.

2.3 Sistema Pyxel.

2.3.1 Estructura de Pyxel.

Pyxel es un sistema diseñado para almacenar, recuperar y publicar imágenes digitales que utilizan el estándar IPTC. Las imágenes digitales son recibidas a tamaño completo y este sistema se encarga de

Capítulo II Descripción de la solución propuesta

generar versiones para la Web y en forma de miniatura. Su interfaz es abierta y es posible la construcción de sistemas que interactúen con él y aprovechen toda su funcionalidad.

Las imágenes digitales que serán procesadas por Pyxel son preparadas anteriormente utilizando un editor de imágenes que permita incluirle los metadatos IPTC. Luego Pyxel extrae los metadatos para registrar las imágenes en su catálogo y usarlos para facilitar la búsqueda de las mismas. Una vez almacenadas no pueden ser sobre-escritas, modificadas, ni borradas, pues las mismas pueden ser usadas por otros sistemas sin necesidad de notificar a Pyxel.

La interfaz web de Pyxel muestra la detección de problemas con el procesamiento de alguna imagen en particular.

2.3.2 Principios de la arquitectura de Pyxel.

La arquitectura lógica o de software consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción de un software para un sistema. Además, establece los fundamentos para que analistas, diseñadores y programadores trabajen en una línea común que permita alcanzar los objetivos del sistema.

2.3.2.1 Sistema distribuido.

El sistema Pyxel está compuesto por varios subsistemas y estos se construyen a partir de módulos. Por lo general, los subsistemas están diseñados para operar independientemente, es decir, pueden correr en varias máquinas y comunicarse por la red. Esto permite que el sistema sea más completo, flexible y tolerante a los fallos que pueda presentar uno de los subsistemas sin que esto lo afecte completamente. Se puede modificar o reemplazar uno de los subsistemas existentes sin tener que detener el sistema completo.

Este principio intenta:

- Ganar flexibilidad en el diseño de cada subsistema.

Subsistemas de Pyxel.

- pyxel.web.application: implementa la interfaz web de Pyxel.
- pyxel.queue: recibe los ficheros que se suben por FTP o Web y asigna un identificador a las

Capítulo II Descripción de la solución propuesta

imágenes.

- `pyxel.processors`: chequea la cola de Pyxel donde se encuentran las imágenes, extrae los metadatos y genera los perfiles de salida.
- `pyxel.storages.failures`: almacena temporalmente las imágenes que no pudieron ser procesadas.
- `pyxel.catalog`: implementa el catálogo de Pyxel.
- `pyxel.storages.images`: implementa el servicio de almacenar los perfiles generados y también recuperados.
- `pyxel.config`: contiene toda la configuración del sistema.

2.3.2.2 Cero Acoplamiento.

Los módulos que componen a Pyxel son independientes en el sentido de que ningún módulo puede tener acceso directo a cualquier otro, con excepción del módulo al que llama y sus propios submódulos. Sin embargo, los resultados producidos por un módulo pueden ser utilizados por cualquier otro módulo cuando se transfiera a ellos el control.

Pyxel se ha diseñado con la máxima de acoplar sólo los módulos necesarios. Esto quiere decir que aunque en el diseño conceptual de Pyxel, se dibuje una dependencia entre un subsistema y otro, se cumple que:

- La dependencia sólo se establece en la interfaz del subsistema. El subsistema dependiente sólo depende que se implementen un conjunto preestablecido de mensajes y tipos de datos.
- Para ganar acceso al subsistema del cual depende, el subsistema dependiente no lo importa directamente; sino que le pide a `pyxel.config` (contiene toda la configuración de Pyxel) que le de acceso al subsistema del cual depende.

Por esta razón, cualquier paquete que forme parte de Pyxel y que implemente uno o más subsistemas debe declarar los subsistemas que implementa.

Este principio intenta lograr:

- Que un subsistema pueda ser reemplazado por otra implementación de los mismos servicios u otra versión.
- Ganar la máxima reutilización.

Capítulo II Descripción de la solución propuesta

2.3.2.3 REST y RDF.

Pyxel es un sistema REST⁷ y debe emplear RDF⁸, lo cual no significa que el intercambio entre los subsistemas se basa en REST y RDF.

2.3.3 Metodología de desarrollo y herramienta de modelado.

Desde sus inicios Pyxel utiliza como metodología de desarrollo a RUP, pues es una de las más utilizadas y difundidas en la actualidad, por contar con características como:

- Flexibilidad.
- Capacidad de adaptación para los distintos tipos de proyectos.
- Robustez.
- Define claramente las actividades que se realizan por roles.
- Define la creación de artefactos de vital importancia para la construcción de proyectos.

La herramienta CASE utilizada es Visual Paradigm, pues permite realizar una modelación absoluta de los procesos del negocio y del sistema y resulta de gran utilidad para los desarrolladores de proyectos.

2.3.4 ORB⁹ de Pyxel.

Pyxel es un sistema distribuido (como se explicó anteriormente), cada uno de los subsistemas que lo componen necesita de los servicios de otros subsistemas y para establecer esta comunicación utiliza como ORB a Pyro¹⁰, lo que permite tener objetos remotos y llamar a sus métodos de forma casi natural.

⁷ Siglas de Resource Description Framework: este es un modo de representar información sobre recursos. Todos los recursos en este framework están identificados por una URI.

⁸ Siglas de Representational State Transfer: es un estilo arquitectónico para sistemas basado en red que propone que las comunicaciones sean independientes unas de otras.

⁹ Intermediario de Peticiones de Objetos.

¹⁰ Es un avanzado y potente sistema de tecnologías de objetos distribuidos, escrito completamente en Python y está diseñado para ser muy fácil de usar.

Capítulo II Descripción de la solución propuesta

2.3.5 Servicios en Pyxel.

Los servicios son procesos que están corriendo permanentemente. La manera que tienen ellos de comunicarse entre sí en Pyxel es pasándose mensajes. Cada servicio es implementado en una clase, los mensajes que recibe son los métodos implementados en esa clase. Los servicios son registrados por su nombre en el Pyro y este a su vez se encarga también del protocolo y transporte de los datos hacia y fuera del servicio.

2.3.6 Características de las imágenes digitales.

Formato de las imágenes digitales.

El sistema Pyxel ha sido probado con ficheros JPG y PNG exitosamente. Actualmente la versión existente del producto está siendo explotada por sus desarrolladores, para determinar el conjunto de formatos que se pueden utilizar, se asume que cualquier imagen que permita que se le introduzcan los metadatos IPTC puede ser procesada por Pyxel.

Tipos de Metadatos IPTC.

Existen 3 versiones de los metadatos IPTC.

- IPTC IIM.
- IPTC Core Schema.
- IPTC Extensions Schema.

Los metadatos que incluyen las imágenes digitales que serán procesadas por Pyxel se encuentran en los formatos establecidos por la IPTC. Algunos de ellos son:

Title	Source
Headline	Creator
DateCreated	Description

2.3.7 Subiendo imágenes digitales por la Web.

Las imágenes digitales para ser subidas a Pyxel se preparan utilizando la herramienta seleccionada. Luego Pyxel las almacena a tamaño completo, las procesa extrayendo los metadatos IPTC (en cualquier versión), genera las versiones para la Web y las publica.

Capítulo II Descripción de la solución propuesta

2.3.8 Sistema de Ayuda al usuario.

Con el objetivo de facilitarles a los periodistas la inclusión de los metadatos IPTC en las imágenes se desarrolló un manual de usuario, haciendo uso de la herramienta XnViewMP. En el documento adjunto **Validar imágenes usando XnViewMP.pdf** se explican detalladamente los metadatos IPTC que se le incluyen a las imágenes, así como los pasos a seguir para validarlas.

2.4 Características de los FTP.

El servicio FTP constituye una de las herramientas más antiguas con las que se ha trabajado en Internet, incluso es anterior a la existencia de la World Wide Web. Fue utilizado por investigadores para trasladar información de un ordenador a otro. Luego se crearon bibliotecas de ficheros para que los usuarios pudieran acceder al mismo sin necesidad de disponer de una clave de acceso. Actualmente es una herramienta imprescindible que proporciona el intercambio de ficheros entre computadoras remotas, así como el acceso a miles de servidores denominados servidores FTP, que contienen información de todo tipo. El FTP permite manipular toda clase de archivos, tanto si son de tipo texto como si son ejecutables, independientemente de que estén comprimidos o empaquetados. Tampoco tiene importancia el sistema operativo en que han sido almacenados o al que van destinados; ya sea DOS, UNIX, Windows, Macintosh, VM, VMS, o cualquier otro.

El protocolo FTP define la manera en que los datos deben ser transferidos a través de una red TCP/IP.

EL objetivo del protocolo FTP es:

- Permitir que equipos remotos puedan compartir archivos.
- Permitir la independencia entre los sistemas de archivos del equipo del cliente y del equipo del servidor.
- Permitir una transferencia de datos eficaz.

Funcionamiento:

Para que pueda procederse a la utilización del FTP deben estar presentes dos partes. La primera de ellas dispondrá de un programa servidor y la segunda deberá tener a su disposición un software cliente. Este protocolo está basado en el modelo cliente-servidor, es decir, un equipo (software cliente) envía órdenes y el otro (programa servidor) espera solicitudes para llevar a cabo las acciones.

Capítulo II Descripción de la solución propuesta

Servicio FTP:

El servicio FTP utiliza para el envío de pedidos y respuestas los puertos 20 y 21 generalmente. Durante la conexión FTP se encuentran abiertos dos canales de transmisión, uno de comandos (canal de control) y uno de datos. El cliente y el servidor cuentan con dos procesos que permiten la administración de estos dos tipos de información.

DTP (Proceso de transferencia de datos): es el encargado de establecer la conexión y administración del canal de datos. El DTP del lado del cliente se denomina USUARIO DE DTP y del lado del servidor SERVIDOR DE DTP.

PI (Intérprete de protocolo): es el encargado de interpretar el protocolo y permitir que el DTP pueda ser controlado a través de los comandos que son recibidos por el canal de control.

- El SERVIDOR PI es responsable de escuchar los comandos que provienen de un USUARIO PI a través del canal de control en el puerto 21, también de establecer la conexión para el canal de control, de recibir los comandos FTP del USUARIO PI a través de este, de responderles y de ejecutar el SERVIDOR DTP.
- El USUARIO PI es responsable de establecer la conexión con el servidor FTP, de enviar los comandos FTP, de recibir respuestas del SERVIDOR PI y de controlar al USUARIO DTP.

Cuando un cliente FTP se conecta con un servidor FTP, el USUARIO PI inicia la conexión con el servidor a través del puerto 21, las órdenes FTP las genera el USUARIO PI y se transmiten al proceso servidor a través del canal de control, el servidor los interpreta, ejecuta su DTP y luego envía una respuesta estándar al PI de usuario a través del canal de control como respuesta a las órdenes. Estas órdenes FTP especifican parámetros para la conexión de datos (puerto de datos, modo de transferencia, tipo de representación y estructura) y la naturaleza de la operación sobre el sistema de archivos (almacenar, añadir, borrar, recuperar, etc). Luego el DTP de usuario u otro proceso, debe esperar a que el servidor inicie la conexión al puerto de datos especificado (puerto 20) para transferir los datos en función de los parámetros que se especificaron.

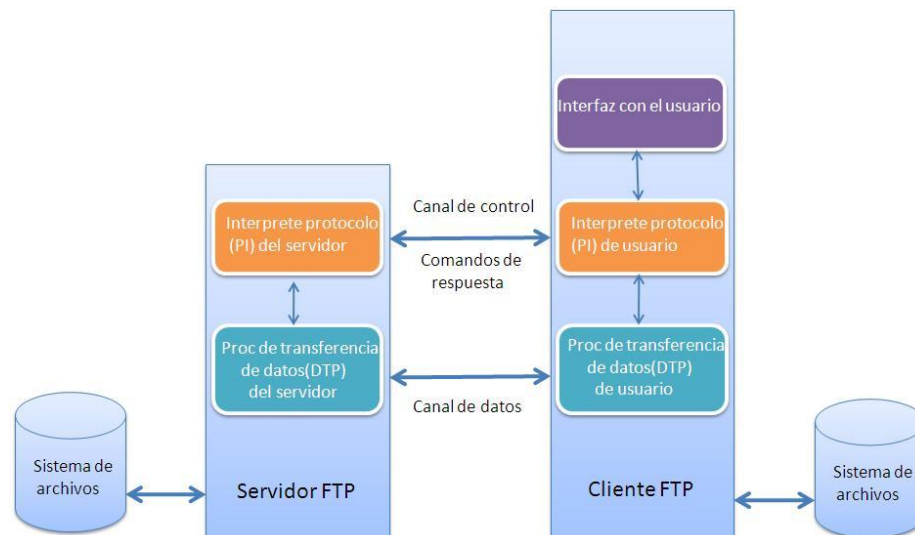


Figura 2.1 Conexión FTP.

2.4.1 Transferencia de archivos en FTP.

Para transportar archivos a través de la red, se debe tener en cuenta la opción más adecuada para conservar la información.

- type ascii: es utilizado para transferir archivos que solo contengan caracteres ASCII.
- type binary: es utilizado cuando se trata de archivos comprimidos, ejecutables para computadoras, imágenes, etc.

2.4.2. Tipos de acceso.

Acceso anónimo.

Los servidores FTP anónimos ofrecen sus servicios libremente a todos los usuarios y permiten acceder a sus archivos sin necesidad de autenticarse, aunque con menos privilegios que un usuario normal, solo podrá leer y copiar los archivos existentes.

Acceso de usuario.

Al iniciar una sesión FTP el usuario debe autenticarse, y así tiene privilegios de acceso a cualquier parte del sistema de archivos del servidor FTP, para modificar y subir archivos.

Capítulo II Descripción de la solución propuesta

Cliente FTP basado en Web.

Un cliente FTP basado en Web es un cliente al cual se puede acceder a través del navegador sin necesidad de tener otra aplicación para ello. Es decir, cuando el usuario accede mediante la Web a un servidor Web, este se conecta a través de FTP al servidor FTP y muestra el contenido que está almacenado en él.

Acceso de invitado.

Permite a cada usuario conectarse a la máquina mediante autenticación, evitando que tenga acceso a partes del sistema de archivos que no necesita para realizar su trabajo.

2.4.3 Modos de conexión del cliente FTP.

FTP admite dos modos de conexión del cliente. Estos se denominan Activo y Pasivo. Tanto en el modo Activo como Pasivo el cliente establece una conexión con el servidor mediante el puerto 21, que establece el canal de control.

Modo Activo.

El modo activo generalmente es conocido también como modo estándar. En este modo se implantan dos conexiones distintas, en la primera conexión se establece una comunicación para la transmisión de comandos a través del puerto aleatorio mayor que 1024 del cliente FTP hacia el puerto 21 del servidor FTP y por esa misma conexión se le notifica al servidor cual es el puerto del cliente FTP que está a la espera de los datos. Es decir, si se descarga algún archivo mediante la ayuda de algún cliente FTP, es el servidor el que inicia la transmisión de datos, desde el puerto 20 al puerto que aleatoriamente el cliente FTP le ha indicado.

Modo Pasivo.

El modo pasivo se realiza cuando el cliente FTP inicia la conexión con el servidor FTP mediante el envío del comando PASV, en este punto el cliente FTP establece una comunicación mediante un canal de control, el cual generalmente utiliza un puerto aleatorio mayor al 1024 para comunicarse con el servidor FTP a través del puerto 21. Al pasar al modo pasivo el cliente FTP pedirá al servidor que abra un puerto, el cual debe ser aleatorio y mayor al 1024, una vez recibida la respuesta, será el cliente el que establezca la conexión de datos al servidor a través del puerto especificado anteriormente.

Capítulo II Descripción de la solución propuesta

En el modo activo se abre una conexión para datos desde el servidor FTP al cliente FTP, esto es una conexión de afuera hacia adentro, por lo que si el cliente FTP se encuentra detrás de un firewall¹¹, este filtrará o bloqueará la conexión entrante.

En el modo pasivo es el cliente el que inicia tanto la conexión de control como la de datos, con lo cual el firewall no tendrá ninguna conexión entrante que filtrar o bloquear.

2.5 Clientes FTP.

Un cliente FTP es un programa que sirve para transferir los archivos desde un ordenador hasta el servidor donde serán alojados, utilizando el protocolo FTP para establecer la conexión. Aunque existen navegadores que vienen equipados con la función FTP, resulta más confiable establecer la conexión utilizando un programa cliente.

Actualmente existe una gran cantidad de clientes FTP que permiten realizar la transferencia de ficheros satisfactoriamente. Algunos de ellos son: CuteFTP, AceFTP, SmartFTP y FileZilla.

CuteFTP: es un popular cliente FTP con una interfaz sencilla que permite la gestión de archivos almacenados en un servidor FTP mediante la copia y pega de elementos. Ofrece otras opciones como la captura de URL y transmisión entre servidores (FXP). Su licencia es Shareware, se puede conseguir una versión de evaluación gratuita y sus capacidades son bastante avanzadas.

AceFTP: es un cliente FTP de sencillo manejo, potente, fiable y rápido. Incluye una serie de características destacables como la transferencia de archivos entre servidores, gestión de sitios favoritos y conexiones simultaneas. Es gratuito, pero incluye publicidad.

SmartFTP: es un potente cliente FTP, rápido y provisto de muchas características básicas y avanzadas. El programa ofrece transferencias seguras, confiables y eficientes. Contiene una interfaz similar al explorador de Windows, es fácil de usar, posee pestañas que permiten trabajar cómodamente con

¹¹ Conocido también como Cortafuegos, es un software utilizado en redes de computadoras para controlar las comunicaciones, permitiéndolas o prohibiéndolas.

Capítulo II Descripción de la solución propuesta

conexiones múltiples, admite reanudar transferencias erróneas, incluyendo envíos y descargas. Su licencia es Freeware y puede ser utilizado para la publicación y mantenimiento de páginas web.

FileZilla: excelente cliente FTP, que mediante una interfaz sencilla, similar al explorador de Windows permite fácilmente la gestión de ficheros vía FTP. Incluye una completa galería de funciones y características como administrador de sitios FTP, conexión proxy y conexiones simultaneas. Ha sido diseñado para soportar las máximas funcionalidades, cuidando siempre la velocidad y asegurando un entorno estable. Su licencia es GPL, es multiplataforma y cuenta con las mejores funcionalidades de los programas FTP. Posee como características:

- capacidad para continuar descargas interrumpidas.
- capacidad para mantener viva la conexión con el servidor FTP.
- soporte para trabajar junto con firewalls.
- posibilidad de trabajo con múltiples conexiones.
- cola de cargas y descargas.
- compatible con el cifrado fuerte utilizando FTP a través de SSL/TLS y SSH (FTPS y SFTP).

Aunque con la utilización de los clientes FTP mencionados anteriormente resulta extremadamente rápido y sencillo subir los archivos al servidor FTP, los más utilizados por las características que poseen son SmartFTP y FileZilla.

A continuación se establece una comparación entre ambos para determinar cuál será empleado en la solución.

- **Sistema operativo:** FileZilla es multiplataforma, de manera que el código se ha optimizado para diferentes sistemas. En el caso de SmartFTP, se puede intentar el uso en sistemas GNU/Linux mediante Wine, aunque por este motivo no se puede garantizar plena compatibilidad.
- **Licencia:** FileZilla está distribuido bajo licencia GPL, lo que permite a los usuarios la libre distribución, modificación y uso del software. SmartFTP está distribuido bajo licencia Freeware que aunque es completamente gratuita no permite realizar modificaciones al código.
- **Uso:** FileZilla dispone de una versión portable, lo cual hace práctico su uso. Aunque se creó la posibilidad de convertir en portable a SmartFTP, lo cierto es que podría resultar complicado para usuarios poco experimentados, pues requiere el copiado manual de ciertas librerías.

Capítulo II Descripción de la solución propuesta

El cliente seleccionado para posterior utilización es FileZilla, pues de acuerdo con las características que posee y la comparación realizada, resulta ser un excelente cliente FTP, multiplataforma y de código abierto.

2.6 Servidores FTP.

Los servidores FTP son almacenes de ficheros. Su misión es permitir a los usuarios recibir y enviar ficheros de todas las máquinas que sean servidores FTP.

Algunos de los servidores FTP existentes son: Pureftpd y Proftpd.

Pureftpd: es un servidor FTP muy seguro, potente, liviano, fácil de configurar e instalar, una de sus mejores características es la de gestión de usuarios virtuales y ancho de banda. La instalación recomendada para aprovecharlo al máximo es con el servidor de base de datos MySQL, el cual se encarga de gestionar los usuarios, los permisos y demás opciones. Es de libre distribución y existe para múltiples plataformas y en muchos idiomas.

Algunas de las características más notables son:

- Usuarios virtuales.
- Gestión de ancho de banda y espacio en disco por usuario.
- Directorios personales Chroot ().
- Estadísticas en tiempo real en texto, HTML o XML.
- Autenticación de usuarios con MySQL, PostgreSQL, LDAP.
- Opciones avanzadas de seguridad.

Proftpd: Professional FTP Daemon es uno de los servidores FTP más versátiles y seguros de los existentes hoy en día. Es altamente configurable, flexible, estable y su diseño es modular (lo que permite extender su funcionalidad ampliamente a extensiones como cifrado SSL/TLS, LDAP o SQL). ProFTPD es uno de los mejores servidores FTP que existen para sistemas Linux, también puede ser utilizado en otros sistemas operativos y puede tener múltiples servidores brindando servicio de FTP anónimo. Proftpd está bien documentado, usa un único fichero de configuración, puede ser fácilmente configurado como múltiples servidores FTP virtuales y tiene capacidades para ser enjaulado dependiendo del sistema de archivos que haya por debajo. Funciona sobre los sistemas operativos y emuladores siguientes:

Capítulo II Descripción de la solución propuesta

AIX	BSD/OS	GNU/Linux
Cvgwin	DG/UX	Mac OS X
Digital UNIX	FreeBSD	Solaris
HP-UX	IRIX	SunOS
Linux for IBM S/390	zSeries	Gentoo

Proftpd posee muchas características que lo convierten en el servidor FTP más completo y eficaz para usar:

- Está bien documentado y es uno de los mejores servidores FTP que existen para GNU/Linux.
- Principal archivo de configuración individual, con directivas y grupos de directivas, los cuales son muy intuitivos para cualquier administrador que haya usado alguna vez el servidor Web Apache.
- Diseñado para funcionar tanto como un servidor independiente o desde inetd, dependiendo de la carga del sistema.
- El código fuente está disponible para los administradores y desarrolladores de auditoría.
- Ofrece opciones avanzadas de seguridad.

2.7 Sistema Gestor de Base de Datos.

En el campo de los Sistemas de Gestión de Base de Datos (SGBD) existen muy buenos competidores. Programas de software como Oracle y Microsoft SQLServer brindan del lado propietario un amplio conjunto de facilidades que no están al alcance de todos, no solo por sus precios sino por la falta total de libertades que imponen sus fabricantes.

PostgreSQL es el gestor de base de datos de código abierto más avanzado hoy en día, ofrece control de concurrencia multi-versión, soporta casi toda la sintaxis SQL y cuenta con un amplio conjunto de enlaces con lenguajes de programación. Contiene prácticamente todo lo que tienen los gestores comerciales, lo cual lo convierte en una buena alternativa GPL.

PostgreSQL se destaca por su amplia lista de prestaciones que lo hacen capaz de competir con cualquier Sistema Gestor de Base de Datos comercial:

Capítulo II Descripción de la solución propuesta

- La Interfaz de Programación de Aplicaciones (API) de acceso al SGBD se encuentra disponible en C, C++, Java, Perl, PHP, Python, entre otros.
- Cuenta con un rico conjunto de tipos de datos, permitiendo además su extensión mediante operadores definidos y programados por el usuario.
- Su administración se basa en usuarios y privilegios.
- Sus conexiones abarcan TCP/IP, sockets Unix y sockets NT.
- Es altamente confiable en cuanto a estabilidad.
- Control de concurrencia multi-versión.
- Soporte para vistas, claves foráneas, integridad referencial, disparadores, procedimientos almacenados y subconsultas.

La aplicación de PostgreSQL contribuirá al buen desarrollo y éxito de la solución propuesta, por ser un programa capaz de almacenar grandes cantidades de datos diferentes. PostgreSQL es muy utilizado en proyectos basados en software libre que exigen prestaciones de alto nivel.

2.8 PgAdmin III.

Con el objetivo de proporcionar una herramienta gráfica que permita realizar el trabajo con PostgreSQL de una manera más sencilla, se hará uso de PgAdmin III; que ha sido diseñado para responder a las necesidades de los usuarios, para escribir consultas y desarrollar complejas Bases de Datos. Su interfaz gráfica soporta todas las características de PostgreSQL y permite realizar una fácil administración. PgAdmin III permite trabajar con casi todos los objetos de la Base de Datos, examinar sus propiedades y realizar tareas administrativas. Constituye una herramienta útil y a la vez didáctica, pues cuando se realiza alguna modificación a un objeto escribe las sentencias SQL correspondientes.

Otras características que incluye:

- Entradas SQL aleatorias.
- Pantallas de Información y “Ayudas” para bases de datos, tablas, índices, secuencias, vistas, programas de arranque, funciones y lenguajes.
- “Ayudas” para importar y exportar datos.
- “Ayuda” para migrar Bases de Datos.

Capítulo II Descripción de la solución propuesta

- Preguntas y respuestas para configurar Usuarios, Grupos y Privilegios.

2.9 Modelo de Dominio.

El Modelo de Dominio o Modelo Conceptual, es una representación visual de los conceptos u objetos del mundo real, significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema, conceptos del mundo real, no de los componentes de software. En este modelo no se muestra comportamiento. Las clases conceptuales pueden tener atributos pero no métodos [12].

El principal objetivo de este modelo es ayudar a los implicados en el desarrollo del producto a utilizar un vocabulario común que posibilite una mejor comprensión entre ellos, para poder entender y describir las clases más importantes dentro del contexto donde se ubica el sistema.

2.9.1 ¿Por qué hacer Modelo de Dominio?

Debido al conocimiento que se posee sobre el funcionamiento del sistema al que se le integrará el módulo, no es necesario realizar un Modelo de Negocio completo, por lo que se realizará un Modelo de Dominio donde se capturarán los tipos de objetos más importantes que existen y los eventos que suceden en el entorno.

2.9.2 Conceptos fundamentales del Modelo de Dominio.

Usuario: es la persona que hace uso de la aplicación para subir al FTP los archivos de imágenes, en este caso es el periodista.

FTP imágenes: es el servidor FTP al cual accede el usuario para almacenar los archivos de imágenes.

Sistema de archivo de imágenes: es un sistema de archivos virtual que crea carpetas temporales donde se almacenarán los archivos de imágenes que se encuentran en el servidor FTP antes de ser enviados a Pyxel.

Carpeta de imágenes: son las carpetas temporales que crea el sistema de archivos para almacenar los archivos de imágenes.

Imágenes: son los archivos de imágenes que se almacenan para posterior publicación.

Pyxel: es el sistema que procesa los archivos de imágenes.

Capítulo II Descripción de la solución propuesta

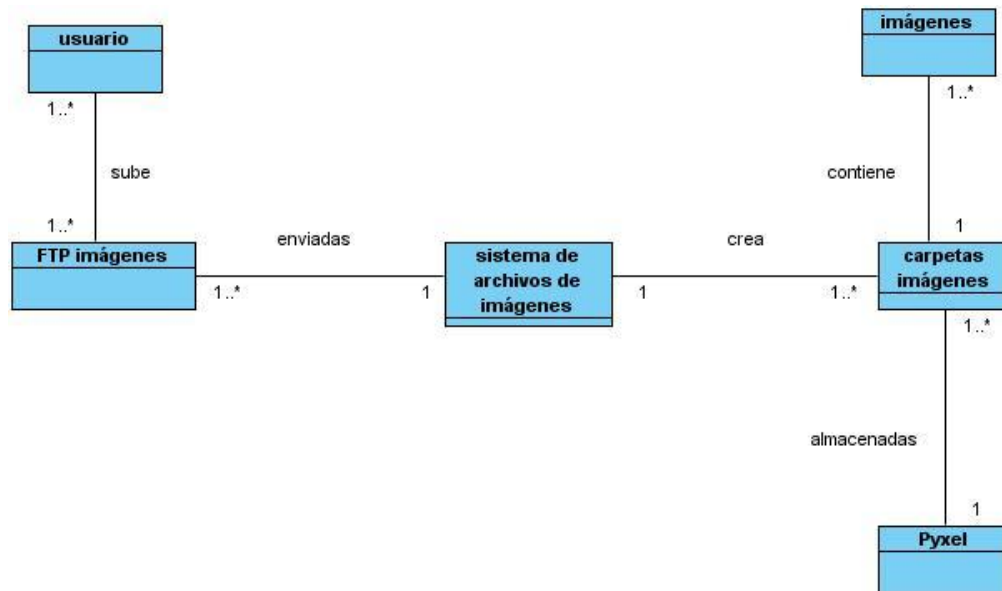


Figura 2.2 Diagrama de Dominio.

2.10 Especificación de los requisitos de software del sistema.

Una especificación de requisitos de software es una descripción completa del comportamiento del sistema a desarrollar. Estos deben ser especificados por escrito, abstractos, concisos y posibles de probar o verificar.

La captura de requisitos surge para lograr una comunicación efectiva entre los usuarios y el equipo de proyecto, con el objetivo de llegar a un entendimiento de lo que se debe hacer; esta es la clave del éxito en la producción de un software. La especificación de los requisitos del sistema evita que existan incongruencias entre lo que el usuario necesita, lo que interpreta cada miembro del equipo de proyecto y lo que realmente se obtiene.

Requisitos funcionales:

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir.

- **R1:** el sistema debe permitir que el usuario se autentique.
- **R2:** las imágenes que se suban al FTP no pueden ser borradas, descargadas o modificadas por los usuarios.
- **R3:** el sistema debe permitir que las imágenes que se almacenan en el sistema de ficheros virtual

Capítulo II Descripción de la solución propuesta

estén contenidas dentro de una carpeta que lleve por nombre el usuario que las subió.

- **R4:** el sistema debe borrar las imágenes que se encuentren en el servidor FTP, cuando estas sean almacenadas en el sistema de ficheros virtual.
- **R5:** el sistema debe borrar las imágenes que se encuentren en el sistema de ficheros virtual, una vez que estas sean almacenadas en Pyxel.

Requisitos no funcionales:

Los requisitos no funcionales son cualidades o propiedades que el producto debe tener.

- **Usabilidad:** la aplicación debe ser de fácil uso para cualquier periodista que tenga acceso al sistema.
- **Soporte:** una vez terminada la aplicación se entregará a los usuarios junto con la documentación que contiene las indicaciones para la instalación del software.
- **Seguridad:** el sistema tendrá acceso diferenciado con contraseña para cada usuario.
- **Software:** para poder instalar el software se debe disponer de un sistema operativo GNU/Linux con distribución Ubuntu o Debian.
- **Hardware:** los requerimientos mínimos del hardware son:
 - Para PC cliente:
 - Computadora con cualquier tipo de CPU a 1 MHz.
 - Memoria RAM de 128 MB.
 - Para PC servidor:
 - Computadora con cualquier tipo de CPU a 1MHz.
 - Memoria RAM de 256 MB.
 - Capacidad de disco duro 80 GB.
- **Restricciones en el diseño y la implementación:**
 - Lenguaje de programación: Python.
 - Herramienta de desarrollo: Geany.

2.11 Definición y descripción de los Casos de Uso del Sistema.

Los casos de uso son descripciones de la funcionalidad del sistema, constituyen la base para la implementación de las fases y disciplinas de RUP. Son artefactos narrativos que describen bajo la forma

Capítulo II Descripción de la solución propuesta

de acciones y reacciones el comportamiento del sistema desde el punto de vista del usuario. Los casos de usos deben hacer referencia a al menos un requerimiento y cada requerimiento debe quedar reflejado en al menos un caso de uso, cualquier modificación que exista en algún requerimiento puede afectar al caso de uso correspondiente.

Clasificación de los casos de uso:

- **Críticos:** más importantes para los usuarios porque cubren las principales tareas o funciones que el sistema ha de realizar. Definen la arquitectura.
- **Secundarios:** sirven de apoyo a los casos de uso críticos, involucran funciones secundarias y tienen un impacto más modesto sobre la arquitectura, pero deben implementarse pronto porque responden a requerimientos de interés para los usuarios.
- **Auxiliares:** no son claves para la arquitectura y completan casos de usos críticos o secundarios.
- **Opcionales:** responden a funcionalidades que pueden o no estar en la aplicación, pero que no son imprescindibles en las primeras versiones [13].

Definición de actor.

Un actor es una idealización de una persona externa, de un proceso, o de una cosa que interactúa con un sistema, un subsistema o una clase. Un actor caracteriza las interacciones que los usuarios exteriores pueden tener con el sistema. En tiempo de ejecución, un usuario físico puede estar limitado a los actores múltiples dentro del sistema. Diferentes usuarios pueden estar ligados al mismo actor y por lo tanto pueden representar casos múltiples de la misma definición de actor [14].

Los actores del sistema:

- No son parte de él.
- Pueden intercambiar información con él.
- Pueden ser un recipiente pasivo de información.
- Pueden representar el rol que juega una o varias personas, un equipo o sistema automatizado.

Actor	Descripción
Periodista	Persona que tendrá acceso al sistema como usuario para subir imágenes.

Tabla 2.1: Definición de actores.

Capítulo II Descripción de la solución propuesta

2.11.1 Diagrama de Caso de Uso del Sistema.

Un Diagrama de Casos de Uso del Sistema representa gráficamente a los procesos y su interacción con los actores.

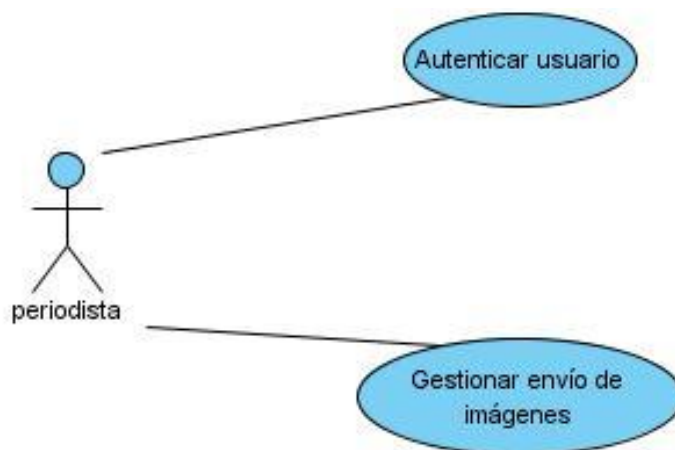


Figura 2.3 Diagrama de Caso de Uso del Sistema.

2.11.2 Descripción de Caso de Uso del Sistema.

Caso de Uso:	Autenticar usuario.	
Actor:	Periodista.	
Resumen:	El caso de uso comienza una vez que el periodista desea entrar al servidor FTP. El periodista accede al cliente FTP.	
Precondiciones:	El usuario que desee entrar debe encontrarse en la base de datos.	
Referencias		
Prioridad	Alta.	
Flujo Normal de Eventos		
Sección "Autenticar usuario"		
Acción del Actor	Respuesta del Sistema	
1- El actor accede al cliente FileZilla y en "Gestor de sitios" presiona "conectar".	1.1- El sistema muestra una ventana para la autenticación del actor.	
2- El actor entra su usuario y contraseña y presiona el botón "Aceptar".	2.1- El sistema muestra un certificado digital enviado por el servidor.	
3- El actor acepta el certificado	3.1- El sistema verifica que los datos sean válidos, en	

Capítulo II Descripción de la solución propuesta

enviado por el servidor, en caso contrario remitirse a CA1.	caso contrario remitirse al CA2. 3.2- El sistema permite entrar al FTP y muestra una carpeta donde se encuentran las imágenes que han sido subidas.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	CA1- Se muestra un mensaje de que no se confió en el certificado remoto y no se pudo conectar al servidor. CA2-Se muestra un mensaje de error para especificar que la autenticación es incorrecta y no se pudo conectar al servidor.
Pos condiciones	Autenticación satisfactoria.

Caso de Uso:	Gestionar envío de imágenes.
Actor:	Periodista.
Resumen:	El caso de uso comienza una vez que el periodista desea subir imágenes al FTP.
Precondiciones:	Las imágenes deben ser previamente preparadas con sus metadatos IPTC. El periodista debe haberse autenticado. El sistema de ficheros virtual debe estar montado en el ordenador.
Referencias	R1, R2, R3, R4, R5.
Prioridad	Alta.
Flujo Normal de Eventos	
Sección “Gestionar envío de imágenes”	
Acción del Actor	Respuesta del Sistema
1- El actor copia las imágenes a la carpeta del FTP.	2- El sistema de archivos virtual (Pyxel.Fuse) toma las imágenes que se encuentran en el servidor FTP. 2.1- Pyxel.Fuse almacena las imágenes dentro de una carpeta con el nombre del usuario que las subió. 2.2- El sistema elimina del FTP las imágenes que fueron copiadas a Pyxel.Fuse. 2.3- Pyxel.Fuse envía las imágenes a la cola de Pyxel. 2.4- El sistema elimina de Pyxel.Fuse los directorios que contienen las imágenes que fueron copiadas a la cola de Pyxel.

Capítulo II Descripción de la solución propuesta

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
Pos condiciones	Imágenes subidas satisfactoriamente.

En este capítulo se determinó que la solución a desarrollar se integrará al sistema Pyxel y consiste en la creación de un módulo que permita subir muchas imágenes vía FTP a dicho sistema. También se realizó un estudio de las herramientas necesarias para la creación de este módulo, obteniendo una selección de las mismas, teniendo en cuenta las características que las hacen ideales para su uso. Se realizó el modelo de dominio con el objetivo de brindarle a los desarrolladores un mejor entendimiento del contexto del problema. Además, se identificaron los requisitos funcionales y no funcionales y se describieron los casos de uso del sistema, los cuales constituyen un aspecto fundamental para el desarrollo de las fases posteriores.

Capítulo III Análisis y diseño de la solución propuesta

En el presente capítulo se realiza el análisis y diseño a la solución propuesta, con el objetivo de efectuar las transformaciones necesarias y lograr un diseño adecuado del sistema. También se modelan los artefactos necesarios que contribuyen a la posterior implementación, estos son los modelos de análisis y diseño; los cuales se encargan de describir partiendo de clases como deben funcionar los casos de uso del sistema.

3.1 Análisis y diseño.

Etapa de un sistema que captura los requisitos y el dominio del problema. El análisis se centra en lo que hay que hacer, mientras que el diseño se centra en cómo hacerlo. En un proceso iterativo, estas etapas no tienen por qué realizarse de forma secuencial. El resultado de esta etapa se representa mediante modelos del nivel de análisis, especialmente la vista de casos de uso y la vista estática [14].

El objetivo de este flujo de trabajo es traducir los requisitos a una especificación que describe cómo implementar el sistema. Luego de seleccionados los requisitos, se obtiene una vista externa del sistema, que describe lo que el cliente espera de él a través del diagrama de casos de uso. Partiendo de esto se debe profundizar en los casos de uso, de tal forma que permita reflejar una vista interna descrita con el lenguaje de los desarrolladores. En esta vista interna se especifican mejor los casos de uso y se determinan las clases necesarias para llevar a cabo las funcionalidades contenidas en ellos. Este proceso se desarrolla fundamentalmente en la fase de elaboración y se corresponde principalmente con el flujo de análisis y diseño.

3.1.1 Modelo de Análisis.

El análisis consiste en obtener una visión del sistema que se preocupa de ver “Qué” hace, de modo que sólo se interesa por los requisitos funcionales. El modelo de análisis debe lograr tres objetivos primarios:

- Describir lo que requiere el cliente.
- Establecer una base para la creación de un diseño de software.
- Definir un conjunto de requisitos que se puedan validar una vez que se construya el software.

Capítulo III Análisis y diseño de la solución propuesta

En el análisis se definen todas las clases que son relevantes al problema que se va a resolver.

Diagrama de clases de Análisis.

En el diagrama de clases de análisis se representan los conceptos fundamentales que se abarcan en el dominio del problema. Estos diagramas figuran las definiciones y relaciones entre las clases.

Clases de análisis: Son evidentes en el dominio del problema pues representan conceptos y relaciones del dominio. Poseen atributos y entre ellas se establecen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos.

RUP propone clasificar las clases del análisis en:

CI_ (nombre de la clase): es una clase interfaz y modela la interacción entre el sistema y sus actores.

CC_ (nombre de la clase): es una clase controladora y coordina la realización de los casos de uso teniendo en cuenta las actividades de los objetos que implementan la funcionalidad del caso de uso.

CE_ (nombre de la clase): es una clase entidad y modela información que posee larga vida y que es a menudo persistente.



Figura 3.1 Diagrama de Clase de Análisis (Autenticar usuario).

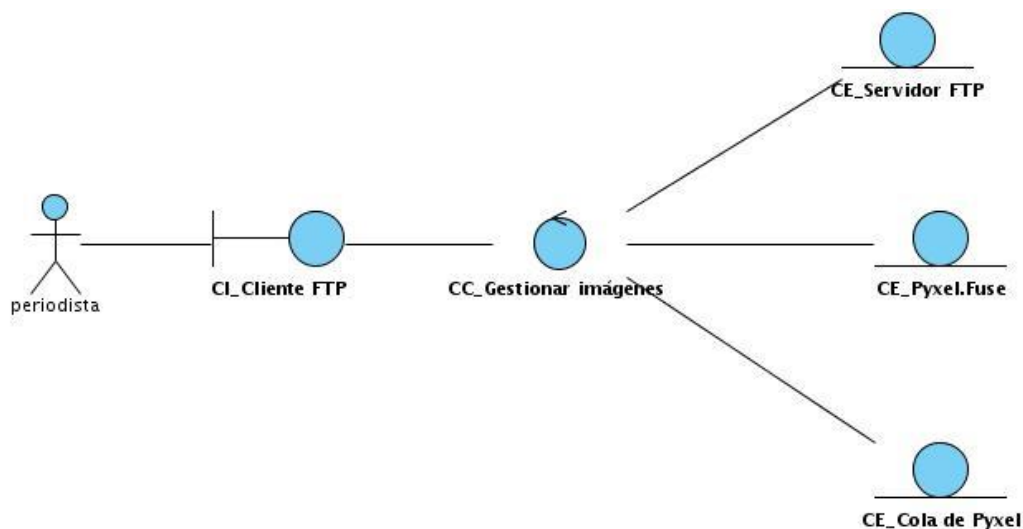


Figura 3.2 Diagrama de Clase de Análisis (Gestionar envío de imágenes).

3.1.2 Diagramas de Interacción.

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. Constan de un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

Los diagramas de interacción contienen:

- Objetos
- Mensajes
- Enlaces

Los diagramas de interacción se dividen en dos categorías:

Diagrama de Secuencia.

Es un diagrama de interacción que destaca el orden temporal de los mensajes y muestra las interacciones expresadas en función de secuencias temporales.

Diagrama de Colaboración.

Un diagrama de colaboración destaca la organización de los objetos que participan en una interacción y manifiesta las relaciones entre los objetos y los mensajes que intercambian.

Capítulo III Análisis y diseño de la solución propuesta

Teniendo en cuenta que los diagramas de colaboración revelan la relación existente entre los objetos y permiten comprender mejor los efectos de estos, se procede a realizar los correspondientes a los casos de uso definidos.



Figura 3.3 Diagrama de Colaboración (Autenticar usuario).

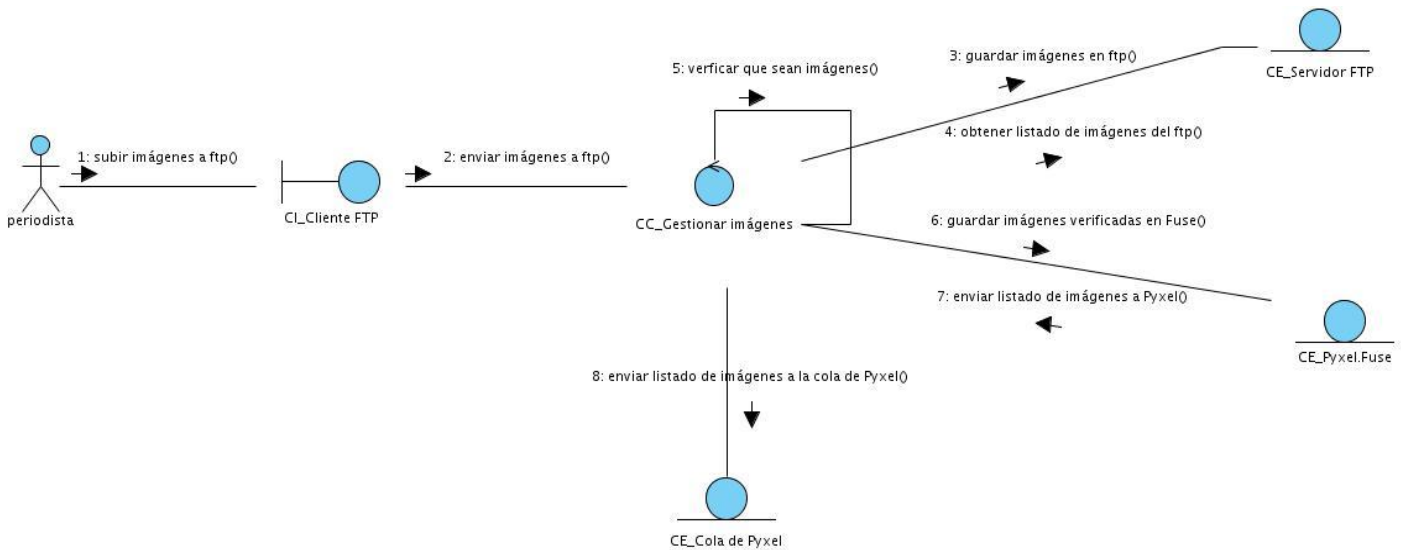


Figura 3.4 Diagrama de Colaboración (Gestionar envío de imágenes).

3.1.3 Modelo de Diseño.

El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales. Debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

Capítulo III Análisis y diseño de la solución propuesta

En el diseño, las decisiones estratégicas y tácticas se toman para resolver los requisitos funcionales y de calidad requeridos de un sistema [14].

Se puede plantear que el modelo de clases de diseño, es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales tienen impacto en el sistema a desarrollar. El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción.

Este modelo sirve de abstracción de la implementación y es utilizado como entrada fundamental de las actividades de implementación.

Diagrama de clases del diseño.

Un diagrama de clases del diseño muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Estos se utilizan para modelar la vista de diseño estática de un sistema.

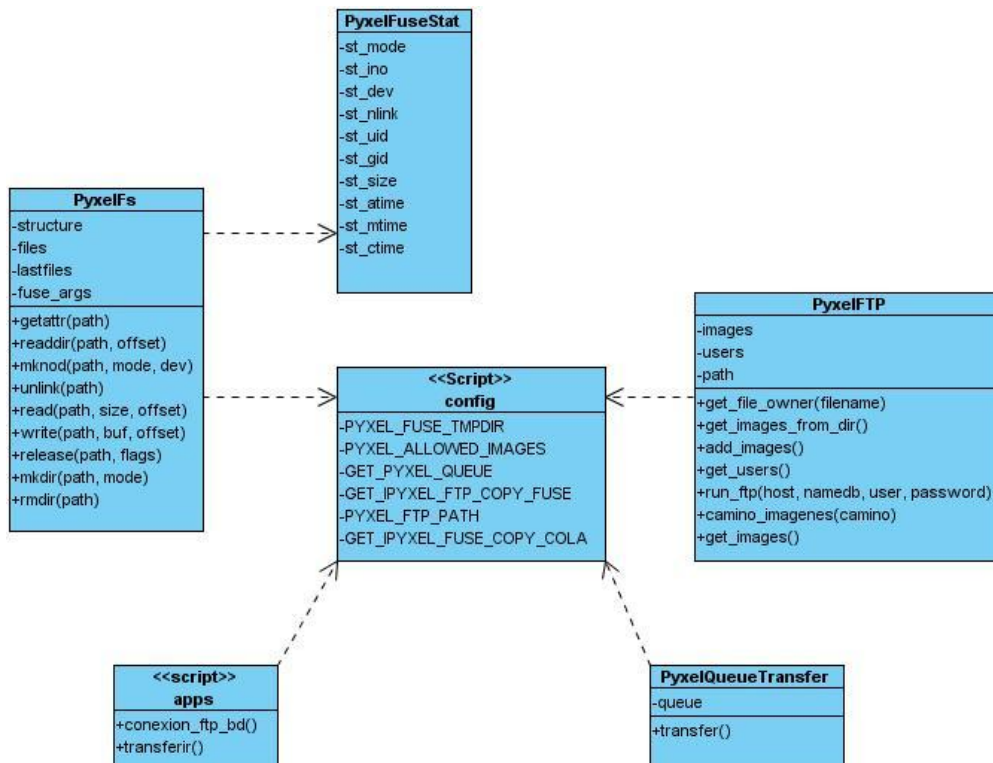


Figura 3.5 Diagrama de Clase de Diseño.

Capítulo III Análisis y diseño de la solución propuesta

La generación de los artefactos que RUP propone para el flujo de análisis y diseño, proporcionó una amplia visión del tamaño y complejidad que posee la solución a crear, así como obtener una mayor comprensión de lo que se propone realizar y definir los principios que guiarán la implementación y organización del módulo. Se analizaron los requisitos indispensables con los cuales debe contar este módulo para satisfacer las necesidades de los usuarios. También se conocen cuales son los mensajes que se intercambian entre los objetos para realizar las tareas necesarias, facilitando de esta forma, una mejor comprensión por parte de los desarrolladores.

Capítulo IV Implementación y prueba

En este capítulo se realiza la implementación y prueba de la solución propuesta. Se construyen los modelos necesarios para desarrollar el proceso de implementación con los diagramas de componentes definidos.

En el flujo de trabajo Implementación se pasa de los resultados de la fase de diseño a implementar el sistema partiendo de componentes tales como ficheros fuente, ejecutables, scripts, etc.

Los objetivos de la implementación son:

- Implementar las clases definidas durante el diseño. En concreto, se implementan dentro de componentes (ficheros) que contienen código fuente.
- Probar los componentes individualmente e integrarlos en uno o más ejecutables.
- Integrar los componentes en el sistema siguiendo un enfoque incremental.

4.1 Diagrama de componentes.

Los diagramas de componentes son utilizados para estructurar el modelo de implementación partiendo de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. Es un grafo de componentes unidos a través de relaciones que pueden ser de compilación y/o ejecución y además se pueden representar las interfaces de esos componentes. El uso más importante que tienen estos diagramas es mostrar la estructura de alto nivel del modelo de implementación, especificando los subsistemas de implementación y sus dependencias cuando se va a importar código y organizar los subsistemas de implementación en capas. Lo que distingue a este diagrama de los demás es su contenido.

Un componente es la parte modular de un sistema que encapsula implementación e interfaces y proporciona la realización de los mismos. Contiene clases y es un elemento de implementación que representa algo físico como los ficheros o archivos.

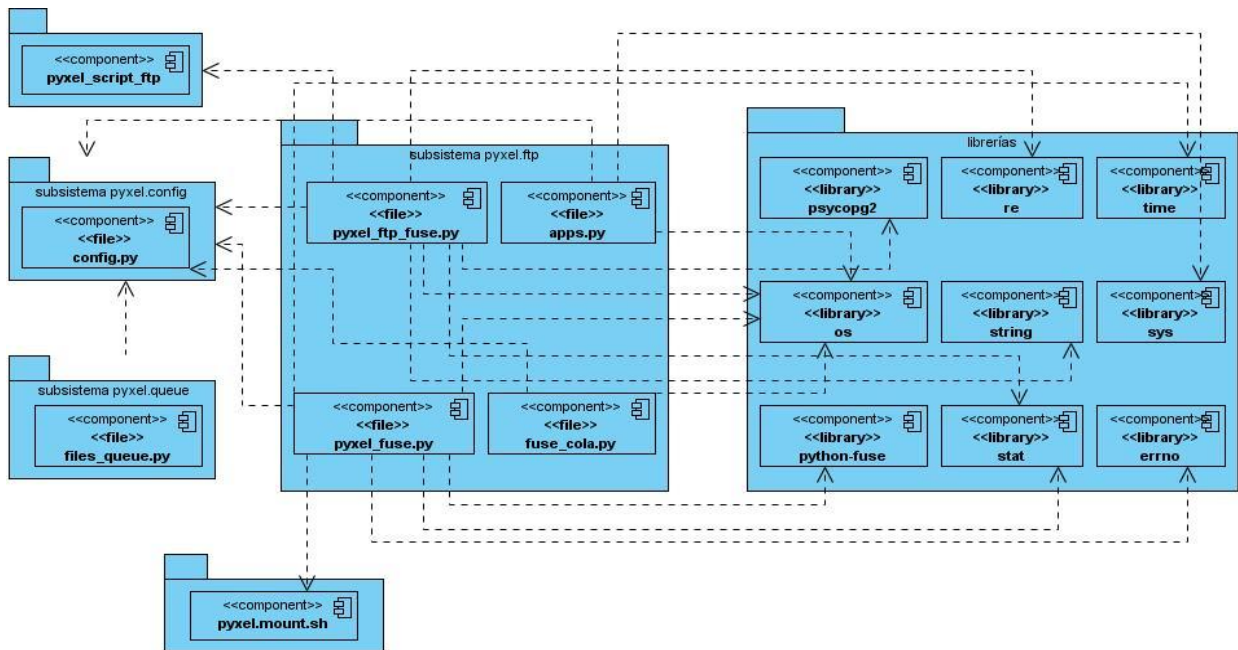


Figura 4.1 Diagrama de Componentes.

4.1.1 Descripción de módulos y componentes.

Con el objetivo de proporcionar un mejor entendimiento sobre el trabajo que realizan los componentes de la aplicación, a continuación se explican cada uno de los utilizados en la realización del módulo, el cual lleva por nombre “pyxel.ftp”.

Módulo **psycpg2**.

Es un controlador de PostgreSQL que está bajo desarrollo activo. Está diseñado para aplicaciones de subprocesos múltiples y gestiona su propia agrupación de conexiones. Una característica interesante que posee es que si se utiliza la matriz de tipo de datos PostgreSQL, **psycpg2** convertirá automáticamente un resultado que contenga ese tipo de datos en una lista de Python.

- **psycpg2.connect:** crea una sesión nueva de base de datos y devuelve un nuevo objeto de conexión.

Módulo Python-fuse.

Sistema de archivos en Espacio de usuario (Fuse) es un módulo cargable de núcleo para sistemas operativos de computador tipo Unix, que permite a usuarios no privilegiados crear sus propios sistemas de archivos sin necesidad de editar el código del núcleo.

Módulo os.

Este módulo proporciona una manera portable de utilizar funcionalidades dependientes del sistema operativo.

- **os.stat():** el valor de retorno es un objeto cuyos atributos corresponden a los miembros de la estructura: st_mode (bits de protección), st_ino (número de nodo i), st_dev (dispositivo), st_nlink (número de enlaces duros), st_uid (id del propietario), st_gid (grupo id del propietario), st_size (tamaño del archivo en bytes), st_atime (tiempo de acceso más reciente), st_mtime (tiempo de modificación de contenidos más reciente), st_ctime (dependiente de la plataforma, el tiempo del cambio de metadatos más reciente en Unix, o el tiempo de la creación en Windows).
- **os.listdir ():** devuelve una lista con todos los ficheros y/o directorios, dados por el camino que se pasa como parámetro. La lista está en orden arbitrario.
- **os.path.isdir ():** regresa verdadero si la ruta pasada como parámetro es un directorio existente.
- **os.path.exists ():** regresa verdadero si la ruta pasada como parámetro se refiere a una ruta existente. Devuelve falso para los enlaces simbólicos rotos.
- **os.mkdir ():** crea un directorio.
- **os.mknod ():** crea un nodo de sistema de archivos.
- **os.rmdir ():** es utilizado para borrar directorios de sistema de archivos. Se pasa como parámetro el directorio a eliminar.

Módulo re.

Este módulo proporciona la expresión regular que coincide con operaciones similares a las encontradas en Perl. Ambos patrones y cadenas a buscar pueden ser cadenas Unicode, así como cadenas de 8 bits.

- **re.compile ():** brinda la posibilidad de compilar una expresión regular para después poder aplicarla

a múltiples cadenas.

Módulo `string`.

El módulo `string` contiene una serie de constantes y clases útiles, así como algunas funciones que están disponibles como métodos en `string`.

- **`string.join ()`**: concatena una lista o tupla de palabras.

Módulo `errno`.

Este módulo hace disponibles los símbolos estándares del sistema `errno`. El valor de cada símbolo es el valor entero correspondiente.

- **`errno.ENOENT`**: no existe el fichero o directorio.

Módulo `time`.

Este módulo proporciona diversas funciones relacionadas con el tiempo. Aunque siempre está disponible, no todas las funciones están disponibles en todas las plataformas.

Subsistema `pyxel.config`: el módulo `pyxel.config` contiene toda la configuración del sistema Pyxel. Además, funciona como el paquete que gestiona las dependencias entre los subsistemas de Pyxel.

Subsistema `pyxel.queue`: el subsistema `pyxel.queue` tiene como propósito recibir los ficheros que suben los usuarios de Pyxel, tanto por la interfaz web como por el FTP. Con este subsistema es con quien se integra la solución.

Subsistema `pyxel.ftp`: este módulo junto con el servidor FTP instalado, constituyen la interfaz FTP por la cual el usuario puede subir muchos ficheros de forma más directa a Pyxel. Los componentes que lo integran son:

- **`pyxel_fuse.py`**: contiene la clase encargada de crear el sistema de ficheros “Pyxel.Fuse”.
- **`pyxel_ftp_fuse.py`**: contiene la clase encargada de tomar las imágenes disponibles en el FTP y almacenarlas en “Pyxel.Fuse”.
- **`fuseCola.py`**: contiene la clase encargada de tomar las imágenes que se encuentran en el sistema de ficheros “Pyxel.Fuse” y las añade a la cola de imágenes de Pyxel.

- **apps.py:** es el script que establece mediante “pyxel.config” la relación con las clases implementadas en el paquete.

pyxel.mount.sh: el script “pyxel.mount.sh” se encuentra en la dirección /var/local/bin y permite montar el sistema de ficheros “Pyxel.Fuse” que se encuentra en el archivo “pyxel_fuse.py”.

pyxel_script_ftp.py: es el script encargado de parar el servidor FTP temporalmente, ejecutar el fichero “pyxel_ftp_fuse.py” y luego volver a levantar el servidor.

4.2 ¿Cómo se realiza el almacenamiento de las imágenes desde el FTP hacia Pyxel?

Luego de seleccionada la vía de solución y obtenidos los artefactos que facilitan el desarrollo del módulo en cuestión, se procede a explicar cómo se realiza el proceso de gestión de las imágenes desde que se encuentran en el FTP hasta que son almacenadas en Pyxel, en otras palabras se explica cómo se efectúa la implementación de “pyxel.ftp”.

4.2.1 Autenticar usuario.

Para almacenar las imágenes en Pyxel mediante el FTP, el usuario debe autenticarse para acceder al servidor y realizar el copiado de las imágenes. Primeramente se efectúa la instalación de las herramientas que lo permiten. Se debe contar con un cliente FTP para interactuar con el servidor donde se encontrarán almacenadas las imágenes temporalmente. El cliente seleccionado es FileZilla debido a las características y comodidades que ofrece. Una vez instalado, se procede a escoger el servidor, en este caso se utilizará Proftpd, pues se pudo comprobar que constituye uno de los más documentados, confiables y fáciles de configurar.

Para acceder al servidor, se le solicita al usuario sus datos de autenticación, estableciéndose de esta forma, la conexión con PostgreSQL, que contendrá una base de datos con la información de los usuarios que pueden acceder al sistema. El tipo de acceso que se realiza al servidor es “Acceso de usuario” y el modo de conexión del cliente es “Pasivo”.

El FTP se considera un método bastante inseguro, pues no encripta los archivos que se envían, ni las contraseñas con las que se accede al servidor. Por esto se hace uso del protocolo SSL/TLS¹², con el objetivo de mantener la confidencialidad e integridad de los datos durante la comunicación.

En el documento adjunto **Manual de configuración de Proftpd y PostgreSQL.pdf** se explica detalladamente el proceso a seguir para instalar y configurar el cliente, servidor y base de datos que se utilizan para la autenticación del FTP. Se declaran también las directivas que se colocan en el archivo de configuración del servidor para realizar la conexión con la base de datos y proporcionarles a los usuarios los permisos para copiar sus imágenes.

4.2.2 Gestionar envío de imágenes.

Realizada la autenticación de usuarios, se ejecutan una serie de actividades para enviar las imágenes digitales que se encuentran en el FTP hacia Pyxel.

Creación del sistema de ficheros virtual.

Se propone crear un sistema de fichero virtual. Este sistema de ficheros se nombrará “Pyxel.Fuse”, donde serán almacenadas las imágenes disponibles en el servidor FTP. El objetivo por el cual surge este sistema de archivos es para que se puedan añadir los nuevos ficheros directamente a la cola de Pyxel. La estructura que va a tener dicho sistema es:

Nombre de usuario/

-- imagen1.jpg

-- imagen2.jpg

Nota: la carpeta se llama Nombre de usuario y contiene dentro las imágenes que el usuario subió (imagen1.jpg e imagen2.jpg).

¹² Ambos protocolos sirven para proporcionar comunicaciones seguras en Internet, usando un modelo de autenticación y privacidad de la información entre extremos sobre Internet mediante criptografía.

Para la creación de este sistema de archivos se hará uso del módulo llamado “Python-fuse”, pues es realmente útil para la creación de este tipo de sistemas de archivos, su instalación es fácil, es estable y posee una librería con una API¹³ sencilla.

Procedimiento de envío de imágenes.



Figura 4.2 Proceso de envío de imágenes a Pyxel.

Para poder realizar el envío de las imágenes a Pyxel, el sistema de ficheros “Pyxel.Fuse” debe estar montado en el ordenador.

El procedimiento de envío de imágenes comienza a realizarse cuando se ejecuta el script “pyxel_script_ftp.py”, encargado de suspender temporalmente el servidor, con el objetivo de que en ese momento los usuarios no puedan acceder a él y así comenzar a realizar el procesamiento de la información sin interrupción y procurando realizar la copia de aquellas imágenes que se encuentran completas en el servidor. Terminado el proceso de almacenamiento de las imágenes en el sistema, el servidor FTP se iniciará nuevamente. Este script se ejecutará como tarea del Cron¹⁴ y se correrá dos veces al día para evitar que el servidor esté muy cargado de información.

Cuando se encuentre montado el sistema de ficheros y se ejecute el script mencionado anteriormente, se realizan dos procesos de forma secuencial:

Proceso de almacenamiento de imágenes en Pyxel.Fuse.

El sistema de ficheros Pyxel.Fuse, toma las imágenes que están disponibles en el servidor FTP, una vez que el usuario las copió.

¹³ Interfaz de Programación de Aplicaciones.

¹⁴ Viene instalado en la mayoría de las distribuciones de Linux, permite automatizar muchas tareas y programarlas cada cierta periodicidad.

Capítulo IV Implementación y prueba

Las imágenes copiadas en Pyxel.Fuse tomarán la estructura explicada anteriormente. Cuando estas se encuentren en el sistema de ficheros son borradas del FTP.

Proceso de almacenamiento de imágenes en Pyxel.

Almacenadas las imágenes en Pyxel.Fuse serán enviadas a Pyxel. Es aquí donde se establece la integración con dicho sistema.

Pyxel posee un subsistema llamado “pyxel.queue”, que contiene una cola donde son almacenadas las imágenes que se suben al sistema. A esta cola es donde serán enviadas las imágenes que se encuentran en Pyxel.Fuse. El acceso a la misma se realiza a través del subsistema “pyxel.config”, pues es el que gestiona las dependencias entre los subsistemas de Pyxel. Cuando las imágenes comienzan a enviarse a la cola de Pyxel, inmediatamente son borradas del sistema de ficheros Pyxel.Fuse.

4.3 Modelo de prueba.

La fase de prueba es una de las más costosas del ciclo de vida de un software. Deben realizarse pruebas de todos los artefactos que se generan durante la construcción de un producto, esto incluye especificación de requisitos, diagramas y código fuente.

Las pruebas de software son las encargadas de evaluar la calidad del producto que se desarrolla. Los principales aspectos a ser evaluados en un producto software son la Fiabilidad (resistencia a fallos), la Funcionalidad (hace lo que debe) y el Rendimiento (lleva a cabo su trabajo de manera efectiva).

Garantizar la calidad de un producto de software implica varios tipos de pruebas, las cuales poseen sus propias características y estrategias. Cuando se le aplica al software cada prueba este se va perfeccionando hasta que sea probado completamente y queda listo para ser entregado. La realización de pruebas no asegura la ausencia de defectos, sólo demuestra que existen errores en la aplicación.

Con la realización de estas pruebas se pretende encontrar y documentar los defectos que puedan afectar la calidad del software, validar los requerimientos y probar a su vez que fueron implementados correctamente.

Existen diferentes niveles y tipos de prueba, entre ellas se encuentran:

Capítulo IV Implementación y prueba

Pruebas de Integración: este tipo de prueba se realiza para comprobar la compatibilidad y funcionalidad de las interfaces entre las partes que componen un sistema, las cuales pueden ser módulos, aplicaciones individuales, aplicaciones cliente/servidor, entre otras. Este tipo de prueba se le realiza en su mayoría a aplicaciones distribuidas.

Pruebas de Validación: este tipo de prueba se realiza sobre el software para evaluar el cumplimiento de los requisitos que fueron especificados en la etapa de concepción del software.

Pruebas de Sistema: el software ya validado se integra con el resto del sistema donde algunos tipos de prueba a considerar son:

- **Rendimiento:** determina el tiempo de respuesta, el espacio que ocupa el módulo en disco o memoria, el flujo de datos que genera a través de un canal de comunicaciones, etc.
- **Resistencia:** determina hasta donde puede soportar el software ciertas condiciones extremas.
- **Robustez:** determina la capacidad que posee el software de soportar entradas incorrectas.
- **Seguridad:** determina los niveles de permisos de usuarios, las operaciones de acceso a las bases de datos y el sistema.
- **Usabilidad:** determina la calidad de un usuario en la forma que interactúa con la aplicación, se considera el grado de satisfacción del usuario y la facilidad de uso del sistema.
- **Instalación:** se determinan las operaciones de arranque y actualización del software.

Pruebas de Carga: está basada en las aplicaciones bajo cargas pesadas, generalmente usadas en sitios web y servidores con gran cantidad de datos, donde se determina en cuales puntos existen degradaciones del sistema.

Pruebas de Aceptación: estas son las pruebas que realiza el usuario, que determinan si el sistema cumple con lo deseado y se obtiene la conformidad del cliente. El propósito de esta etapa es comprobar si el producto software puede calificarse como adecuado o aceptable por los clientes antes de su puesta en marcha.

La mayoría de los desarrolladores de productos de software llevan a cabo un proceso denominado pruebas alfa y beta para descubrir errores que parezca que sólo el usuario final puede descubrir.

- **Prueba alfa:** esta prueba se lleva a cabo por un cliente en el lugar de desarrollo. En esta prueba se usa el software de manera neutral con el desarrollador como observador del usuario y registrando los errores y problemas de uso. Las pruebas “alfa” se llevan a cabo en un entorno controlado.
- **Prueba beta:** estas pruebas son llevadas a cabo por los usuarios finales del software en lugares de trabajo de los clientes. A diferencia de las pruebas “alfa”, el desarrollador no está presente. De esta forma, la prueba “beta” es una aplicación en vivo del software en un entorno que el desarrollador no puede controlar, por lo que el cliente registra todos los problemas que encuentra en el software y luego informa al desarrollador.

Prueba de Caja Blanca: está basada en la lógica interna de la aplicación y el código. Hace una cobertura de declaraciones de código, ramas, caminos y condiciones. La prueba de caja blanca se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental. Mediante las pruebas de caja blanca se pueden obtener casos de prueba que garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método, además de que se ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa, que se ejecuten los bucles en sus límites operacionales y se ejerciten las estructuras internas de datos para asegurar su validez.

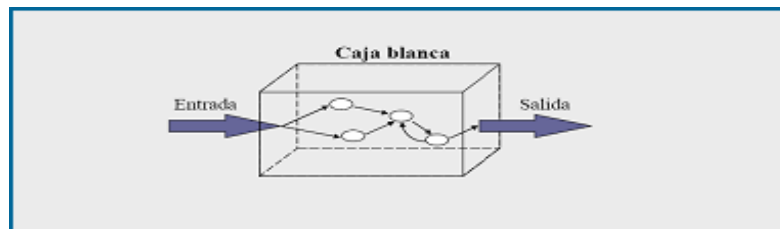


Figura 4.3 Representación de prueba de Caja Blanca.

Algunas técnicas de prueba de Caja Blanca son:

- Prueba de Condición.
- Prueba de Flujo de Datos.
- Prueba de Camino Básico.
- Prueba de Bucles.

Prueba de Caja Negra: estas pruebas se centran en lo que se espera de un módulo, es decir, intentan

encontrar casos en que el módulo no corresponde con su especificación, es por esto que son denominadas pruebas funcionales, a las cuales se le suministran datos como entrada para estudiar la salida, sin importar que pueda estar haciendo el módulo internamente. Es por esto que se especifica que no están basadas en el conocimiento del código o diseño interno, sino que determinan la funcionalidad del sistema.

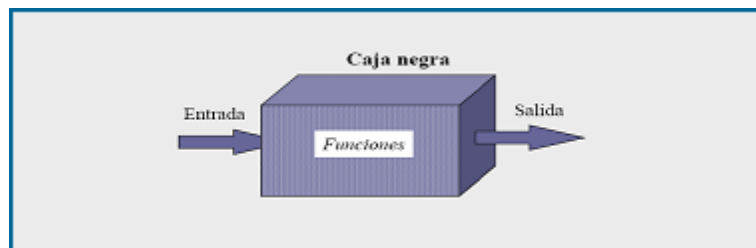


Figura 4.4 Representación de prueba de Caja Negra.

Para confeccionar los casos de prueba de Caja Negra existen distintas técnicas. Ellas son:

- Particiones de Equivalencia.
- Análisis de Valores Límites.
- Métodos basados en Grafos.
- Pruebas de comparación.

4.3.1 Pruebas aplicadas.

Las pruebas que se le realizarán al módulo son Pruebas de Caja Blanca y Caja Negra, con el objetivo de efectuar la comprobación de los caminos lógicos del software y funcionalidades del sistema.

Prueba de Caja Blanca.

Para efectuar las pruebas de caja blanca a un algoritmo primeramente se debe realizar el análisis de la complejidad ciclomática del mismo, pues este valor define la cantidad máxima de caminos independientes del programa y el límite superior para el número de casos de prueba que se deben llevar a cabo, para asegurar que se ejecute al menos una vez cada sentencia y cada condición.

La porción de código que se toma para realizar la prueba es la función encargada de obtener del servidor FTP solamente los archivos que sean imágenes y adicionarlas a una lista.

```

def get_images_from_dir(self):
    images = list()
    path = self.path
    content = os.listdir(path)

    for line in content:
        r = re.compile('.*\.((' + string.join(PYXEL_ALLOWED_IMAGES, '|') + '))')
        if r.match(line):
            if not os.path.isdir(path + line) and os.path.exists(path + line):
                owner = self.get_file_owner(path + line)
                fp = open(path + line)
                if owner != None:
                    images.append((owner, fp.read(), fp.name.encode('utf-8'), line))
                    if owner not in self.users:
                        self.users.append(owner)

    return images

```

Figura 4.5 Representación del algoritmo `get_images_from_dir ()`.

Seguidamente se construye el grafo de flujo asociado al código anterior.

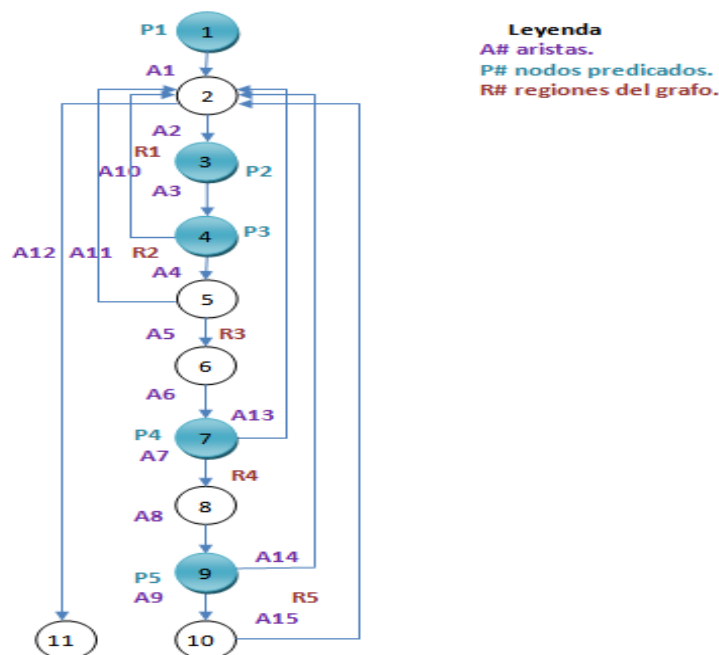


Figura 4.6 Grafo de flujo de la función `add_images ()`.

Capítulo IV Implementación y prueba

Aplicación de las fórmulas para calcular la complejidad ciclomática. Para que el resultado sea correcto, debe ser el mismo para las tres fórmulas.

$$\begin{array}{l} 1. V(G) = (A - N) + 2. \\ 2. V(G) = P + 1. \\ 3. V(G) = R. \end{array} \quad \text{donde} \quad \left\{ \begin{array}{l} A: \text{aristas. } R: \text{regiones.} \\ N: \text{nodos.} \\ P: \text{nodos predicados.} \end{array} \right.$$

Sustituyendo los valores se obtiene que:

$$\begin{array}{l} 1. V(G) = (15 - 11) + 2 = 6. \\ 2. V(G) = 5 + 1 = 6. \\ 3. V(G) = 6. \end{array}$$

El cálculo efectuado mediante las tres fórmulas ha dado el mismo valor, por lo que se puede decir que la complejidad ciclomática del código es de 6, lo que significa que existen seis posibles caminos por donde el flujo puede circular, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado.

Representación de los caminos básicos por los que puede recorrer el flujo:

Camino básico # 1: 1 - 2 - 11.

Camino básico # 2: 1 - 2 - 3 - 4 - 2 - 11.

Camino básico # 3: 1 - 2 - 3 - 4 - 5 - 2 - 11.

Camino básico # 4: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 2 - 11.

Camino básico # 5: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 2 - 11.

Camino básico # 6: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 2 - 11.

Este algoritmo está conformado por la sentencia repetitiva (for) y por condicionales que se ejecutan una tras la otra, de esta manera, solo es necesario probar que el bucle y las condicionales se ejecutan completamente, garantizando una cobertura total.

Capítulo IV Implementación y prueba

Para realizar los casos de pruebas, es necesario cumplir con las siguientes exigencias:

Descripción: se hace la entrada de datos necesaria, validando que ningún parámetro obligatorio pase nulo al procedimiento o no se entre algún dato erróneo.

Condición de ejecución: se especifica cada parámetro para que cumpla una condición deseada y poder ver el funcionamiento del procedimiento.

Entrada: se muestran los parámetros que entran al procedimiento.

Resultados esperados: se expone el resultado que se espera que devuelva el procedimiento.

Caso de prueba para el camino básico #6.

Para el camino 6 que recorre la mayor cantidad de nodos del grafo de flujo se conformó el siguiente caso de prueba:

Descripción: Se quiere tomar del servidor FTP solamente aquellos archivos que sean imágenes. La función debe retornar una lista con todas las imágenes disponibles en el servidor.

Condición de ejecución: en la variable *path* se especifica la dirección del servidor donde se encuentran los ficheros que han sido almacenados, en la variable *content* se encuentra un listado con todos los ficheros disponibles en la dirección especificada en *path*.

Entrada: *path* = /home/ftp/upload / *content* = [image1.jpg, yane.txt]

Resultados esperados: se espera que el método devuelva una lista llamada *images* que contenga la imagen "image1.jpg".

Prueba de Caja Negra.

Para comprobar la calidad del producto desarrollado, se propone realizar prueba de caja negra al caso de uso "Autenticar usuario", con el objetivo de demostrar que el mismo cumple con las precondiciones y pos condiciones especificadas. En la realización de esta prueba, se tendrá en cuenta la técnica de Partición de

Capítulo IV Implementación y prueba

Equivalencia, pues la misma permite examinar los valores válidos e inválidos de las entradas existentes en el software.

Caso de Uso "Autenticar usuario"		
Entrada	Resultados esperados	Condiciones
El usuario deja en blanco los campos de la ventana de autenticación.	El sistema muestra un mensaje de error comunicando que no se ha dado ningún nombre de usuario.	El usuario debe llenar los campos de autenticación.
El usuario introduce caracteres no válidos en la ventana de autenticación y acepta el certificado digital.	El sistema muestra un mensaje de error comunicando que no se pudo conectar al servidor.	El usuario debe entrar los datos de autenticación correctos.
El usuario introduce los datos de autenticación válidos y no acepta el certificado digital que envía el servidor.	El sistema muestra un mensaje de que no se confió en el certificado remoto y no se pudo conectar al servidor.	El usuario debe aceptar el certificado digital.
El usuario introduce caracteres no válidos en la ventana de autenticación y no acepta el certificado digital.	El sistema muestra un mensaje de que no se confió en el certificado remoto y no se pudo conectar al servidor.	El usuario debe entrar los datos de autenticación correctos y aceptar el certificado digital.
El usuario introduce los datos válidos y acepta el certificado digital.	El sistema establece la conexión con el servidor FTP y muestra los datos que se encuentran en él.	Los datos entrados son correctos.

Tabla 4.1 Prueba para el caso de uso Autenticar usuario.

Capítulo IV Implementación y prueba

En este capítulo se realizó la implementación y prueba del producto, demostrando su adecuado funcionamiento y reflejando la calidad con que ha sido llevada a cabo la proyección del sistema. Se generó el diagrama de componentes, el cual muestra la relación entre los principales componentes del sistema, así como su distribución. Se brindó una base para el conocimiento de las librerías utilizadas en la implementación. Además, se realizó la explicación de las pruebas que fueron aplicadas al producto, las cuales permiten aumentar la calidad final de la solución, obteniendo como resultado un producto aceptable.

Conclusiones Generales

La realización de este trabajo responde a la necesidad de buscar una alternativa para los centros de prensa cubanos y sus periodistas que les permitiera efectuar la gestión y almacenamiento de sus archivos de imágenes digitales de manera rápida y eficiente.

Con la realización de este trabajo se logró:

- Efectuar un estudio de las herramientas necesarias para crear la solución, obteniendo una selección de las mismas, teniendo en cuenta las características que las hacen ideales para ser utilizadas.
- Utilizar la metodología y herramientas adecuadas para garantizar la implementación del módulo con la calidad requerida, haciendo uso del software libre.
- A partir de la descripción del problema y el levantamiento de requisitos se construyeron los casos de uso, aspecto fundamental que se tuvo en cuenta en el desarrollo de la solución.
- Se creó un manual para facilitar la inclusión de los metadatos en las imágenes que Pyxel procesa, utilizando la herramienta XnViewMP.
- Se creó un manual que contiene la configuración efectuada al servidor Proftpd para conectarse con bases de datos creada con PostgreSQL, con el objetivo de que sirva de guía al personal que desee realizarla.
- Dar cumplimiento de forma satisfactoria al objetivo general, implementándose el módulo “pyxel.ftp”, que permite gestionar grandes volúmenes de imágenes digitales haciendo uso del sistema Pyxel.

Como conclusión general, puede afirmarse que con la implementación del módulo “pyxel.ftp” se logró que los periodistas pudieran almacenar sus imágenes digitales en el sistema Pyxel, evitando así el tener que ir al CIPRE a hacer entrega de las mismas. De igual forma las imágenes existentes en los medios de prensa cubanos ya cuentan con un lugar centralizado y seguro que procese el almacenamiento de las mismas en grandes paquetes.

Recomendaciones

Por las tendencias a cambios que existen actualmente en las aplicaciones Web y para tratar de aumentar las ventajas de la misma, se recomienda:

- Extender el uso de la aplicación a otras instituciones que trabajen con imágenes digitales.
- Continuar con el perfeccionamiento del módulo para alcanzar una mayor seguridad en la autenticación y envío de archivos.
- Crear una vía óptima, mediante la cual se puedan tomar las imágenes del FTP, sin necesidad de deshabilitar temporalmente el servidor.

Referencias Bibliográficas

- [1] De Lobos María Elena. Modularidad. Procedimientos y Funciones. Marzo 2005. [Citado 17 Marzo 2010]. Disponible en la Web:
<<http://www.mailxmail.com/curso-aprende-programar/modularidad-procedimientos-funciones>>.
- [2] Cornell University. Tutorial de Digitalización de Imágenes - Terminología básica. 2000. [Citado 22 Abril 2010]. Disponible en la Web:
<<http://www.library.cornell.edu/preservation/tutorials/spanish/intro/intro-01.html>>.
- [3] Dr Valdivia Monteagudo Pedro. La imagen digital, una mirada interna. Diciembre 2008. [Citado 17 Marzo 2010]. Disponible en la Web:
<http://www.cecam.sld.cu/pages/rcim/revista_3/articulos_html/articulo_pedro.htm>.
- [4] Instituto Nacional de Estadística y Geografía México. Metadatos. Marzo 2003. [Citado 22 Abril 2010]. Disponible en la Web: <<http://antares.inegi.gob.mx/metadatos/metadat1.htm>>.
- [5] Universidad de las Américas Puebla. Introducción a los Metadatos. 2006. Disponible en la Web:
<<http://hosting.udlap.mx/estudiantes/jose.ferrercz/Introduccion%20a%20los%20metadatos.pdf>>.
- [6] EraDelPixel. EraDelPixel.com: IPTC. Enero 2010. [Citado 23 Abril 2010] Disponible en la Web:
<<http://www.eradelpixel.com/2010/01/iptc.html>>.
- [7] Dr Moliner. Universitat de València-Servei d'Informàtica. Enero 2010. [Citado 17 Marzo 2010]. Disponible en la Web:<<http://www.uv.es/ciuv/cas/web/ftp.html>>.
- [8] Develoweb S.R.L. Desarrollo de Páginas Web Perú - Desarrollo de tiendas virtuales, catálogos virtuales, sistemas web :Develoweb: 2010. [Citado 17 Marzo 2010]. Disponible en la Web:
<<http://www.develoweb.net/desarrollo-paginas-web.php>>.
- [9] El equipo de desarrollo de PostgreSQL. Manual del usuario de PostgreSQL. 2009.
- [10] Martínez Alejandro y Martínez Raúl. Guía a Rational Unified Process. 2009. Disponible en la Web:

Referencias Bibliográficas

<<http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-Guia%20RUP.pdf>>.

[11] Duque González Raúl. Python para todos.

Disponible en la Web: <<http://launchpadlibrarian.net/18980633/Python%20para%20todos.pdf>>.

[12] Pressman Roger. *Ingeniería de Software "Un enfoque práctico"*. 5ta ed [Madrid: Graw Hill], 2002.

[13] Universidad de las Ciencias Informáticas. Flujo de Trabajo de requerimientos. 2010.

[14] James Rumbaugh, Ivar Jacobson, Grady Booch. El Lenguaje Unificado de Modelado. Manual de Referencia.

Bibliografía

Alito. TODO SOFTWARE: Visualizador de imágenes ágil y sin complejos. Octubre 2009. [Citado 17 Marzo 2010]. Disponible en la Web:

<<http://www.todosoftware-alito.blogspot.com/2009/10/visualizador-de-imagenes-agil-y-sin.html>>.

Coletti Esteban Daniel. ProFTPD - un servidor FTP para profesionales.

Duque González Raúl. Python para todos. Disponible en la Web:

<<http://launchpadlibrarian.net/18980633/Python%20para%20todos.pdf>>.

Duttke Jens. PhotoME - Exif, IPTC & ICC Metadata Editor. 2008. [Citado 18 Marzo 2010]. Disponible en la Web: <http://www.photome.de/home_es.html>.

El equipo de desarrollo de PostgreSQL. Manual del usuario de PostgreSQL. 2009.

Enrico Tröger, Frank Lanitz, Nick Treleaven and Dominic Hopf. Geany: Home Page. 2006. [Citado 22 Abril 2010]. Disponible en la Web: <<http://www.geany.org/>>.

EraDelPixel. EraDelPixel.com: IPTC. Enero 2010. [Citado 23 Abril 2010]. Disponible en la Web:

<<http://www.eradelpixel.com/2010/01/iptc.html>>.

Genbeta. Dr. Upload, sistema de almacenamiento gratuito de imágenes de hasta 10 MB. Enero 2008. [Citado 19 Marzo 2010]. Disponible en la Web:

<<http://www.genbeta.com/web/dr-upload-sistema-de-almacenamiento-gratuito-de-imagenes-de-hasta-10-mb>>.

Grigio.org. Xmp Manager | Grigio.org - un blog crítico su Linux Ubuntu, Tecnología e altro. 2007. [Citado 18 Marzo 2010]. Disponible en la Web: <http://grigio.org/xmp_manager>.

Herrmann Martin. Mapivi - Martin's Picture Viewer. Septiembre 2009. [Citado 18 Marzo 2010]. Disponible en la Web: <<http://mapivi.sourceforge.net/mapivi.shtml>>.

Juszkiewicz Leo. Introducción al FTP. Diciembre 2005. [Citado 23 Marzo 2010]. Disponible en la Web:

<<http://www.desarrolloweb.com/articulos/2287.php>>.

- MasterMagazine. Definición de FTP - Significado y definición de FTP. 2004. [Citado 23 Marzo 2010]. Disponible en la Web: <<http://www.mastermagazine.info/termino/5086.php>>.
- Raja Prado Elena. Casi todas las pruebas del software. 2007. Disponible en la Web: <<http://www.sistedes.es/TJISBD/Vol-1/No-4/articles/pris-07-raja-ctps.pdf>>.
- Red Hat Enterprise Linux. FTP. 2005. [Citado 22 Marzo 2010]. Disponible en la Web: <<http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ftp.html>>.
- Softonic. PhotoME - Descargar. 2007a. [Citado 4 Abril 2010]. Disponible en la Web: <<http://photome.softonic.com/>>.
- Softonic. Xmp Manager (Linux) - Descargar. 2007b. [Citado 4 Abril 2010]. Disponible en la Web: <<http://xmp-manager.softonic.com/linux>>.
- Tellado Fernando. Arkiva – Comparte, almacena y más | Incubaweb. Julio 2008. [Citado 19 Marzo 2010]. Disponible en la Web: <<http://www.incubaweb.com/arkiva-comparte-almacena-y-mas/>>.
- Universidad de las Ciencias Informáticas. Preguntas frecuentes — Ayuda de Ecúmene Pyxel 1.0. 2010. [Citado 17 Marzo 2010]. Disponible en la Web: <<http://10.33.20.251:3389/help/faq/#qu-son-los-metadatos-iptc-y-para-qu-me-sirven>>.
- Vera Pablo. Flypicture: sistema de almacenamiento online para imágenes y archivos | Adictos a la red. Mayo 2007. [Citado 19 Marzo 2010]. Disponible en la Web: <<http://www.adictosalared.com/flypicture-sistema-de-almacenamiento-online-para-imagenes-y-archivos/>>.

Glosario de Términos

CIPRE: Centro de Información para la Prensa.

TIC: Tecnologías de la Información y las Comunicaciones.

API: Interfaz de Programación de Aplicaciones.

FTP: “File Transfer Protocol” en español “Protocolo de Transferencia de Ficheros”.

TCP/IP: Protocolo de control de transmisión / Protocolo de Internet, es un conjunto de protocolos de red en los que se basa Internet y que permiten la transmisión de datos entre redes de computadoras.

ORB: Intermediario de Peticiones de Objetos.

GPL: es el acrónimo de Licencia Pública General de GNU.

GIMP: herramienta de manipulación fotográfica multiplataforma.

RUP: Rational Unified Process. Metodología de desarrollo de software.

IDE: Entorno de Desarrollo Integrado.

URL: Uniform Resource Locator, protocolo para especificar direcciones en Internet.

UML: Lenguaje Unificado de Modelado.

CASE: Ingeniería de Software Asistida por Ordenador.

IPTC: IPTC es un organismo internacional dedicado a promover estándares que permiten el intercambio de información entre las principales agencias de noticias del mundo.

REST: siglas de Resource Description Framework, este es un modo de representar información sobre recursos. Todos los recursos en este framework están identificados por una URI.

RDF: siglas de Representational State Transfer, es un estilo arquitectónico para sistemas basado en red que propone que las comunicaciones sean independientes unas de otras.

SSL/TSL: ambos protocolos sirven para proporcionar comunicaciones seguras en Internet, usando un modelo de autenticación y privacidad de la información entre extremos sobre Internet mediante criptografía.

SGBD: Sistema Gestor de Base de Datos.

Pyxel: sistema Web para el almacenamiento y publicación de imágenes digitales.

Cron: viene instalado en la mayoría de las distribuciones de Linux, permite automatizar muchas tareas y programarlas cada cierta periodicidad.

Software: es un programa o aplicación programada para realizar tareas específicas.

PostgreSQL: es un Sistema de Base de Datos Relacionales Orientadas a Objetos.