



Universidad Tecnológica de la Mixteca

**SISTEMA PARA LA DIGITALIZACIÓN, PROCESAMIENTO
E IMPRESIÓN DE IMÁGENES EN LINUX**

TESIS

PARA OBTENER EL TÍTULO PROFESIONAL DE:

INGENIERO EN COMPUTACIÓN

PRESENTA:

HUGO ENRIQUE MARTÍNEZ CORTÉS

ASESOR INTERNO:

M.C. FRANCISCO DE ASIS LÓPEZ FUENTES

ASESOR EXTERNO

DR. JOSÉ JAVIER BÁEZ ROJAS

Agradecimientos

Agradezco de manera muy especial al Doctor José Javier Báez Rojas por todo su apoyo, paciencia y comprensión, durante el desarrollo del presente trabajo de tesis, así como al M.C. Francisco de Asis López Fuentes por todos sus consejos y atenciones. A mis sinodales, los Profesores Gabriel Gerónimo Castillo, Luis Anselmo Zarza y José Antonio Moreno, por su tiempo y recomendaciones.

Al equipo de Yukunitzá S. C. por el gran respaldo que me brindaron durante todo el proceso de realización de esta tesis, y a quienes les expreso una enorme gratitud y les reitero mi más profundo compromiso.

A todos mis amigos y familiares, en los que siempre encontrare el aliento para seguir adelante.

Agradezco también al INAOE (Instituto Nacional de Astrofísica Óptica y Electrónica) por todas las facilidades otorgadas para la realización del presente trabajo de tesis.

U. T. M. 12072

Resumen

En el presente trabajo de tesis se describen e implementan los mecanismos necesarios para la operación de un sistema de impresión en el sistema operativo Linux, capaz de controlar dispositivos de alto y bajo formato. En particular para la impresora de formato largo o plotter de la marca HP modelo DesingJet 1055CM así como impresoras láser y de inyección de tinta. Para esto se utilizan los sistemas de impresión LPD y CUPS que presentan múltiples características y propiedades convenientes, que les han permitido ampliar su espectro de difusión en los sistemas UNIX.

En este trabajo también se describen e implementan los mecanismos necesarios para la operación de un sistema de digitalización de imágenes en el sistema operativo Linux, para esto se utiliza el sistema SANE que permite en particular el control y administración de dispositivos encargados de la digitalización de imágenes.

Adicionalmente se desarrolla una aplicación para la segmentación de imágenes en color, utilizando el sistema de color modificado I1I2I3. Esta aplicación esta diseñada con base en una interfaz gráfica de usuario construida a partir de las herramientas proporcionadas en el paquete GTK y aprovecha además las características de modularidad presentes en el sistema GIMP. Este sistema incluye además, numerosas herramientas y módulos para la manipulación y creación de imágenes.

Como tarea final se integran los módulos de administración y control de los dispositivos encargados de la digitalización e impresión de imágenes, a la estructura funcional del sistema GIMP.

U. T. M. 12072

Índice general

| | |
|--|-----------|
| CAPITULO 1 INTRODUCCIÓN | 1 |
| 1.1 Antecedentes | 4 |
| 1.2 Objetivos | 6 |
| 1.3 Justificación | 7 |
| CAPÍTULO 2 PRESENTACIÓN DE LINUX Y DEL SOFTWARE LIBRE | 9 |
| 2.1 Software libre | 9 |
| 2.2 Sistema operativo Linux | 11 |
| 2.2.1 Características de Linux | 13 |
| 2.3 Conceptos de software libre | 14 |
| 2.3.1 La filosofía del software libre | 14 |
| 2.3.2 Elección del software libre | 15 |
| 2.3.2.1 Argumentación de mercado. | 17 |
| 2.3.2.2 Argumentación financiera | 18 |
| 2.3.2.3 Argumentación técnica | 20 |
| 2.3.2.4 Argumentación de soluciones | 20 |
| 2.3.2.5 Argumentación de servicios | 22 |
| 2.3.3 Falsas ideas sobre el software libre | 22 |
| 2.3.4 Problemas reales del software libre. | 24 |
| CAPÍTULO 3 SISTEMA DE DIGITALIZACIÓN DE IMÁGENES EN LINUX | 26 |
| 3.1 Generalidades | 27 |
| 3.1.1 Introducción al sistema SANE. | 27 |
| 3.1.2 Módulos que integran la distribución SANE | 29 |
| 3.2 Implementación | 31 |
| 3.2.1 Instalación del sistema de digitalización de imágenes para Linux utilizando SANE | 31 |
| 3.2.1.1 Instalación del escáner | 31 |
| 3.2.1.1.1 Pseudo-controladores. | 33 |
| 3.2.1.1.2 Sistema ppSCSI. | 33 |

| | | |
|--|---|-----------|
| 3.2.1.2 | Instalación de los módulos necesarios para la operación del escáner en el kernel de Linux | 36 |
| 3.2.1.3 | Instalación de SANE. | 40 |
| CAPÍTULO 4 SISTEMAS DE IMPRESIÓN EN LINUX | | 43 |
| 4.1 | Procedimientos generales para la configuración de un dispositivo de impresión bajo Linux | 44 |
| 4.2 | Generalidades | 45 |
| 4.2.1 | Impresoras soportadas en Linux. | 45 |
| 4.2.1.1 | Impresoras Postscript | 46 |
| 4.2.1.2 | Impresoras sin soporte para Postscript | 47 |
| 4.2.1.3 | Compatibilidad de las impresoras con Linux | 47 |
| 4.2.2 | Funcionamiento de un sistema de gestión para colas de impresión | 48 |
| 4.2.3 | Software para la gestión de colas de impresión | 49 |
| 4.2.3.1 | Sistema LPRng | 50 |
| 4.2.3.2 | Sistema LPD | 51 |
| 4.2.3.2.1 | GPR | 52 |
| 4.2.3.2.2 | PDQ | 52 |
| 4.2.3.2.3 | Configuración de dispositivos de impresión en el sistema LPD | 53 |
| 4.2.3.2.3.1 | Impresión hacia un sistema LPD remoto | 55 |
| 4.2.3.2.3.2 | Impresión hacia un servidor samba o Windows | 56 |
| 4.2.3.2.3.3 | Impresión hacia una impresora con un servidor de impresión LPD integrado | 56 |
| 4.2.3.2.3.4 | Configuración de una impresora utilizando la herramienta Printtool | 57 |
| 4.2.3.2.4 | Protocolo LPD | 61 |
| 4.2.3.2.4.1 | Protocolo de solicitud y respuesta | 61 |
| 4.2.3.2.4.2 | Solicitudes lpq | 62 |
| 4.2.3.3 | Sistema CUPS | 63 |
| 4.2.3.3.1 | Estructura de CUPS | 64 |
| 4.2.3.3.2 | Configuración de dispositivos de impresión en el sistema CUPS | 66 |
| 4.3 | Implementación | 71 |
| 4.3.1 | Requerimientos | 71 |
| 4.3.2 | Procedimiento para la configuración de una impresora en Linux | 72 |
| 4.3.3 | Configuración del sistema de impresión LPD para las impresoras HP Laserjet 6L y HP DeskJet 693C | 73 |
| 4.3.4 | Configuración del sistema de impresión LPD para la impresora HP DesingJet 1055CM | 75 |

| | |
|--|------------|
| 4.3.5 Implementación de un pseudo-servidor de impresión utilizando el sistema LPD | 76 |
| 4.3.6 Configuración del sistema de impresión CUPS para la impresora HP LaserJet 6L y HP DeskJet 693C | 81 |
| 4.3.7 Configuración del sistema de impresión CUPS para la impresora HP DesingJet 1055CM | 82 |
| CAPÍTULO 5 MÓDULO PARA LA SEGMENTACIÓN DE IMÁGENES EN COLOR EN UIMP | 83 |
| 5.1 Generalidades | 84 |
| 5.1.1 Introducción a GIMP | 84 |
| 5.1.1.1 Características principales de GIMP | 85 |
| 5.1.1.2 Versiones de GIMP | 86 |
| 5.1.2 Sistema de color modificado I1I2I3 | 87 |
| 5.2 Implementación | 88 |
| 5.2.1 Fundamentos para programar un módulo de GIMP | 88 |
| 5.2.1.1 Función principal | 89 |
| 5.2.1.2 El procedimiento query | 91 |
| 5.2.1.2.1 El argumento params | 93 |
| 5.2.1.2.1.1 El tipo INT32 | 95 |
| 5.2.1.2.1.2 El tipo IMAGE | 95 |
| 5.2.1.2.1.3 El tipo DRAWABLE | 96 |
| 5.2.1.2.2 El parámetro return_vals. | 96 |
| 5.2.1.3 El procedimiento run | 96 |
| 5.2.1.3.1 Valores de retorno | 97 |
| 5.2.1.4 Procedimientos para la manipulación de imágenes en GIMP | 98 |
| 5.2.1.4.1 Sistema de coordenadas | 99 |
| 5.2.1.4.2 Región de píxeles | 101 |
| 5.2.1.4.3 Aplicación de los cambios realizados sobre una región de píxeles | 103 |
| 5.2.1.5 Estructura general de un módulo programado para GIMP | 104 |
| 5.2.2 Implementación de la interfaz gráfica | 105 |
| 5.2.2.1 GTK | 105 |
| 5.2.2.2 Diseño | 106 |
| CONCLUSIONES | 111 |
| REFERENCIAS | 113 |

| | |
|--|------------|
| APÉNDICE A. Código fuente del script para aplicar el parche patch-2.2.15pre14-ppscsi1 al kernel de Linux. | 115 |
| APÉNDICE B. Código fuente del script <i>fil_virtual</i> para el servidor de impresión de la sección 4.3.5 | 118 |
| APÉNDICE C. MANUAL DE USUARIO | 120 |
| A.1 Requerimientos del módulo. | 120 |
| A.2 Compilación del módulo | 120 |
| A.3 Instalación del módulo | 121 |
| A.4 Ejecución del módulo | 121 |
| A.5 Pantalla principal | 123 |
| GLOSARIO | 127 |

Lista de figuras

| | | |
|------------|--|-----|
| Figura 3.1 | Arquitectura para un sistema encargado de la digitalización de imágenes utilizando SANE y un scanner SCSI de puerto paralelo . . . | 32 |
| Figura 3.2 | Arquitectura del sistema ppSCSI | 34 |
| Figura 4.1 | Funcionamiento de un sistema de gestión para colas de impresión . . . | 49 |
| Figura 4.2 | Estructura del sistema CUPS | 64 |
| Figura 4.3 | Infraestructura de cómputo que incluye dispositivos de impresión y equipos de cómputo conectados por medio de una red de área local. . . | 71 |
| Figura 4.4 | Infraestructura de cómputo que incluye un servidor de impresión . . . | 77 |
| Figura 5.1 | Representación gráfica de los argumentos de la función <code>gimp_drawable_mask_bounds</code> | 100 |
| Figura 5.2 | Representación gráfica de los argumentos de la función <code>gimp_drawable_offset</code> | 101 |
| Figura 5.3 | Diagrama de casos de uso para la herramienta de segmentación | 106 |
| Figura 5.4 | Diagrama de iteración para aplicar una fórmula del sistema I1I2I3 | 107 |
| Figura 5.5 | Diagrama de iteración para binarizar la imagen de trabajo | 108 |
| Figura 5.6 | Diagrama 2 de iteración para binarizar la imagen de trabajo | 108 |
| Figura 5.7 | Diagrama 3 de iteración para binarizar la imagen de trabajo | 109 |
| Figura 5.8 | Diagrama de iteración para cambiar las propiedades de la imagen en color | 110 |
| Figura A.1 | Jerarquía de opciones para la invocación del módulo encargado de la segmentación de imágenes. | 122 |
| Figura A.2 | Pantalla Principal. | 122 |
| Figura A.3 | Carpeta <i>AJUSTAR UMBRAL</i> | 123 |
| Figura A.4 | Resultado de la selección del umbral sobre el histograma de la imagen de trabajo | 124 |
| Figura A.5 | Carpeta <i>MODIFICAR PROPIEDADES DE LA IMAGEN</i> | 125 |
| Figura A.6 | Previsualización de los cambios realizados a las áreas segmentadas de la imagen en color | 126 |

Lista de tablas

| | | |
|------------|--|-----|
| Tabla 4.1 | Parámetros utilizados en la descripción de una entrada en el archivo /etc/printcap | 54 |
| Tabla 4.2 | Descripción del tipo de solicitud enviada al servidor LPD | 61 |
| Tabla 4.3 | Descripción de las respuestas retornadas por el servidor LPD. | 62 |
| Tabla 4.4 | Valores de configuración para los dispositivos presentes en la infraestructura de la figura 4.3 | 72 |
| Tabla 4.5 | Valores de configuración para la instalación de la impresora HP Laserjet 6L con Printtool | 74 |
| Tabla 4.6 | Valores de configuración para la instalación de la impresora HP DeskJet693C con Printtool. | 74 |
| Tabla 4.7 | Valores de configuración para la instalación (Remote Unix LPD Queue) de la impresora HP DesingJet 1055CM con la herramienta Printtool. | 75 |
| Tabla 4.8 | Valores de configuración para la instalación (Direct to port printer) de la impresora HP DesingJet 1055CM con la herramienta Printtool | 76 |
| Tabla 4.9 | Valores de configuración para los clientes Linux utilizando la herramienta Printtool | 80 |
| Tabla 4.10 | Valores de configuración para la instalación de la impresora HP Laserjet 6L con la herramienta web de la distribución CUPS | 81 |
| Tabla 4.11 | Valores de configuración para la instalación de la impresora HP Deskjet 693C con la herramienta web de la distribución CUPS | 81 |
| Tabla 4.12 | Valores de configuración para la instalación de la impresora HP DesingJet 1055CM con la herramienta web de la distribución CUPS | 82 |
| Tabla 4.13 | Valores de configuración para la instalación (socket) de la impresora HP DesingJet 1055CM con la herramienta web de la distribución CUPS | 82 |
| Tabla 5.1 | Definición de los argumentos para la función gimp_install_procedure | 92 |
| Tabla 5.2 | Argumentos requeridos de acuerdo a la posición del módulo en el menú de GIMP.. | 94 |
| Tabla 5.3 | Descripción de los argumentos para la función gimp_pixel_rgn_init | 102 |
| Tabla 5.4 | Configuración de los atributos dirty y shadow | 103 |

Capítulo 1

Introducción

El grado de desempeño de un sistema de cómputo depende en gran medida del rendimiento del sistema operativo subyacente, debido a que esta capa de software se encarga de la administración de los recursos del sistema (memoria, tiempo de procesador, dispositivos periféricos y de almacenamiento, entre otros) e implementa los servicios necesarios (llamadas al sistema) para la ejecución de los programas de aplicación de alto nivel. El grado de eficiencia y eficacia que se obtiene en la ejecución de las tareas realizadas en un sistema de cómputo se encuentra íntimamente ligada a su vez, con el desempeño de los diversos programas de aplicación que trabajan sobre el sistema operativo. En consecuencia la elección de las aplicaciones más apropiadas, así como de un sistema operativo estable, robusto y de alto desempeño, constituyen aspectos de gran importancia y trascendencia para el mejoramiento de la eficiencia y eficacia en la ejecución de las tareas realizadas en un sistema de cómputo. En este sentido de nada serviría una aplicación con un alto desempeño si el sistema operativo sobre el cual se ejecuta es inestable y administra los recursos del sistema de manera ineficiente, esto se debe a la dependencia directa que los programas de aplicación tienen sobre los servicios ofrecidos por el módulo operativo. Una situación similar se presenta cuando se tiene un sistema operativo estable y altamente eficiente en la gestión de los recursos del sistema, pero las aplicaciones presentan un pobre desempeño y no aprovechan las cualidades heredadas por el sistema operativo subyacente. En resumen es muy importante utilizar un sistema operativo de alto desempeño en conjunción con programas de aplicación con las mismas cualidades.

En un ámbito en donde se realizan múltiples actividades de investigación y desarrollo, frecuentemente surgen tareas poco convencionales cuyos requerimientos no se satisfacen por completo con la utilización de una herramienta o aplicación determinada, en consecuencia en muchos de los casos es necesario utilizar la combinación de diferentes aplicaciones para cubrir las diversas etapas que involucra la ejecución completa de las tareas en cuestión.

En el peor de los casos ninguna de las aplicaciones disponibles puede incluir las herramientas necesarias para la ejecución de una tarea determinada, en tales situaciones generalmente se utilizan lenguajes o herramientas de programación, que permiten escribir aplicaciones en donde se implementan los algoritmos requeridos para la realización de estas tareas. Uno de los principales inconvenientes que se acentúa con este hecho se desprende

de la utilización de aplicaciones propietarias (cuyo código fuente no se encuentra libremente disponible), debido a que no es posible integrar nuevos módulos a la estructura general de estos programas, con lo cual se limita su ámbito de aplicación y su funcionalidad en situaciones específicas. En consecuencia los módulos desarrollados por los usuarios deben utilizarse de manera aislada ampliando con ello el número de aplicaciones y fases requeridas para la ejecución de una tarea determinada.

Una alternativa al problema planteado consiste en la utilización de software libre, que se caracteriza básicamente por seguir una filosofía de código abierto, esto significa que las aplicaciones que pertenecen a esta categoría son distribuidas con su código fuente libremente disponible. De este hecho se desprenden numerosas cualidades que distinguen a las aplicaciones de código abierto, como su interoperabilidad, portabilidad, confiabilidad y en especial su alto desempeño. Utilizando software libre es posible integrar nuevos módulos a las aplicaciones existentes, debido a que estas son diseñadas utilizando una arquitectura abierta, que muestra los detalles de su implementación y ofrece los mecanismos necesarios para incorporar nueva funcionalidad a su núcleo operativo. En consecuencia cada usuario tiene la posibilidad de satisfacer sus necesidades computacionales específicas, utilizando para ello herramientas de programación que le permitan escribir aplicaciones que se puedan integrar en programas más robustos y con mayor funcionalidad. Un argumento adicional a favor de la utilización de software libre es la disponibilidad del código fuente, lo cual favorece la adaptabilidad, escalabilidad, estudio y la transmisión de los productos de software. El problema principal que se suscita de este enfoque corresponde al hecho de que es necesario conocer con detalle los mecanismos apropiados para integrar y programar tales aplicaciones, lo cual generalmente constituye un problema importante sobre todo para usuarios inexpertos con pocos conocimientos de programación.

En particular uno de los sistemas más representativos del software libre, que ha experimentado un gran crecimiento y desarrollo en los últimos años es el sistema operativo Linux, su aplicación se ha extendido rápidamente hacia múltiples sectores como el educativo, empresarial, industrial y de investigación. Este fenómeno se debe en gran medida a sus múltiples cualidades, expresadas en su estabilidad, eficiencia, amplitud de soluciones y rendimiento. Tales características se derivan principalmente de la filosofía del software libre y constituyen argumentos importantes, que muestran la conveniencia de utilizar este sistema operativo como plataforma para el desarrollo de aplicaciones, así como para la realización de tareas con un propósito determinado, mismas que se pueden asociar de manera especial y sin ningún problema con los requerimientos del procesamiento digital de imágenes. Sin embargo el sistema operativo Linux y el software abierto en general presentan algunas desventajas en comparación con algunos sistemas propietarios (como

Windows), que en múltiples ocasiones constituyen argumentos importantes en detrimento de la utilización de este sistema operativo, en particular para la ejecución de las tareas asociadas con la manipulación y creación de imágenes.

Una de las principales desventajas de Linux consiste en la existencia de múltiples aplicaciones para una misma solución, lo cual resulta claramente inadecuado en ambientes en donde existen usuarios inexpertos, pues dificulta el proceso de selección de los sistemas más apropiados. En el ámbito de los sistemas de impresión por ejemplo, existen diferentes opciones como los sistemas LPD (Line Printing Daemon), LPRng o CUPS (Common UNIX Printing System), los cuales presentan características propias y se adaptan mejor a determinadas situaciones. En este sentido la elección de la solución más conveniente toma mayor importancia, debido a que un sistema que involucra tareas asociadas con la manipulación y creación de imágenes, depende indudablemente de la elección apropiada del sistema de impresión subyacente.

Otra de las desventajas más importantes de Linux que ha persistido hasta el momento actual, corresponde al hecho de que no presenta soporte para múltiples dispositivos de hardware, o los soporta de manera limitada. Esto tiene especial trascendencia en ambientes en donde existen sistemas enfocados al procesamiento digital de imágenes, pues una fase muy importante para el funcionamiento de estos sistemas consiste básicamente en la digitalización e impresión de imágenes. Afortunadamente en la actualidad existen múltiples dispositivos encargados de tales operaciones que tienen compatibilidad con Linux, aunado a esto, también existen diversas aplicaciones que operan sobre este sistema operativo y ofrecen soluciones para su administración. Además muchos de los fabricantes de tales dispositivos están poniendo su atención y sus recursos para ofrecer soluciones orientadas a Linux, como la empresas IBM, HP, Compaq y EPSON, entre otras, de tal manera que en un futuro cercano su espectro de compatibilidad con los diversos dispositivos de hardware se ampliará considerablemente. Sin embargo, en la actualidad resulta muy importante establecer un mecanismo apropiado y la información necesaria para realizar la selección, instalación y configuración, particularmente de los sistemas de impresión y digitalización de imágenes.

1.1 Antecedentes

Los institutos de diseño, minería e hidrología son áreas que integran al conjunto de disciplinas tecnológicas con presencia en la Universidad Tecnológica de la Mixteca (UTM). A estos institutos se encuentran adscritos una amplia gama de profesionistas encargados de múltiples actividades de educación e investigación, en sus diversas áreas de estudio. Sin duda para la culminación exitosa de estas actividades es fundamental contar con las herramientas tecnológicas apropiadas, que permitan mejorar la eficiencia del proceso de ejecución de estas tareas, mismas que normalmente requieren un alto grado de precisión y detalle. Un aspecto notable surge a partir del hecho de que estas actividades involucran en la mayoría de los casos como elemento primordial, una fase de manipulación y creación de imágenes, esto se observa por ejemplo en el trazo de planos arquitectónicos e industriales, o en la manipulación de mapas cartográficos o planos geológicos, entre muchas otras actividades.

En este sentido es indudable que la presencia y operación de un sistema de cómputo constituye uno de los elementos de mayor importancia y trascendencia para alcanzar con éxito, los objetivos de eficiencia y eficacia requeridos especialmente en las actividades relacionadas con la manipulación y creación de imágenes digitales. Desde su nacimiento los sistemas de cómputo han jugado un papel decisivo en el mejoramiento progresivo del desempeño de las actividades relacionadas con la manipulación de información digital, de esta manera su aplicación se ha expandido hacia múltiples sectores y actualmente son utilizados con éxito en diversas disciplinas tecnológicas. Por supuesto el campo del procesamiento de imágenes no se ha quedado al margen de esta situación, y por el contrario es una de las áreas que ha encontrado un mayor beneficio en el uso de sistemas de cómputo. Actualmente existen numerosas aplicaciones de software y dispositivos de hardware desarrollados especialmente para su aplicación en este campo, lo cual ha representado una aportación muy valiosa para la aplicación apropiada de sus técnicas y procedimientos.

Los institutos de diseño, minería e hidrología de la UTM, cuentan desde luego con sistemas de cómputo para la ejecución de sus diferentes actividades y proyectos, en este sentido aprovechan los grandes beneficios derivados de su operación. La infraestructura de cómputo en el instituto de diseño se basa particularmente en estaciones de trabajo que operan sobre el sistema operativo Windows y en dispositivos periféricos de alto desempeño como digitalizadores de imágenes e impresoras de alto y bajo formato, además existe una infraestructura de red que permite que todas las estaciones de trabajo compartan y se comuniquen con los dispositivos periféricos presentes en la red. Adicionalmente con el propósito de realizar las actividades relacionadas con la manipulación y creación de imágenes se utilizan diversas aplicaciones, como los programas CorelDraw, PaintShoPro,

AutoCad y Photoshop, por citar a los más importantes, estos son programas comerciales propietarios, diseñados para trabajar sobre el sistema operativo Windows.

Una de las actividades cuya ejecución se facilita con la operación de un sistema de cómputo consiste particularmente en la manipulación de mapas. Estos mapas generalmente contienen información de diversos tipos diferenciada por colores, de esta manera un tarea frecuente consiste en modificar el color o aislar las distintas regiones presentes en el mapa, además de agregar información o hacer composiciones, entre otras actividades. Este proceso involucra diferentes etapas que van desde la digitalización de los mapas utilizando un escáner, la fase de procesamiento de la imagen realizada con uno o varios programas y finalmente la fase de impresión de la imagen procesada, en un dispositivo de alto o bajo formato. Para la ejecución de cada etapa generalmente se utilizan diversas aplicaciones, lo cual resulta poco práctico si consideramos que en estos casos con frecuencia es necesario transportar información de una aplicación a otra, lo cual requiere en la mayoría de las ocasiones un cambio en el formato de la imagen o la compresión de la misma, con su correspondiente pérdida de información. Además un aspecto importante que surge al realizar la manipulación de planos se desprende del hecho de que las imágenes con la información de los planos generalmente son de un gran tamaño, en algunos casos de varios megabytes, lo cual dificulta claramente su manejo en un sistema de cómputo, sobre todo si consideramos que la plataforma de trabajo corresponde a Windows, pues este sistema operativo es inestable y administra los recursos del sistema de manera poco eficiente, incluyendo por supuesto a la memoria principal del sistema de cómputo.

Linux por su parte, es un sistema maduro, estable y presenta un alto desempeño, sin embargo requiere un proceso más complejo para la instalación y operación de los dispositivos de impresión y digitalización de imágenes, lo cual representa un gran inconveniente sobre todo si se compara con la facilidad que implica la ejecución de estas actividades en el ambiente Windows.

1.2 Objetivos

- Establecer, describir e implementar los mecanismos necesarios para la operación de un sistema de impresión en el sistema operativo Linux, en particular para la impresora de formato largo o plotter de la marca HP modelo DesingJet 1055CM así como impresoras láser y de inyección de tinta.
- Establecer, describir e implementar los mecanismos necesarios para la operación de un sistema de digitalización de imágenes para el sistema operativo Linux, en particular para escáners con un adaptador de bus paralelo a SCSI.
- Mostrar la conveniencia de Linux y las aplicaciones de software libre como una solución alternativa para la ejecución de tareas asociadas con el procesamiento digital de imágenes
- Desarrollar una aplicación para la segmentación de imágenes en color, utilizando el sistema de color modificado I1, I2, I3. Esta aplicación debe aprovechar los beneficios del ambiente gráfico y las características de modularidad presentes en las aplicaciones de código abierto que operan en Linux. Además se debe integrar en este sistema, los módulos de administración y control de los dispositivos encargados de la digitalización e impresión de imágenes.

1.3 Justificación

Linux es un sistema operativo estable, robusto y de alto desempeño capaz de satisfacer las necesidades relacionadas con la manipulación y la creación de imágenes digitales, además constituye el software que ha experimentado el mayor crecimiento y difusión en los últimos años en el campo de los sistemas operativos, de esta manera representa la propuesta central así como la plataforma de desarrollo para las tareas realizadas en este trabajo de tesis.

Con el propósito de justificar la utilización del sistema operativo Linux, resulta imprescindible mostrar los procedimientos e información necesaria para configurar un sistema de impresión capaz de controlar dispositivos de formato grande y pequeño, así como un sistema para la operación y control de dispositivos encargados de la digitalización de imágenes. En especial debido a que esto representa uno de los principales argumentos en detrimento de la utilización de Linux, dada su incompatibilidad con múltiples dispositivos de hardware y la ausencia de un mecanismo más eficiente para su instalación y operación. Estos módulos de software son importantes no sólo para la operación de un sistema enfocado al procesamiento y creación de imágenes, sino además constituyen elementos de importancia fundamental en la operación normal de un sistema de cómputo. En particular el presente trabajo se centra en el plotter HP DesingJet 1055CM, debido a que es este dispositivo se utiliza en el instituto de diseño de la UTM.

Por otra parte, la segmentación de imágenes en color representa uno de los procesos más importantes y útiles para la manipulación de mapas temáticos. Con el propósito de mejorar la ejecución de este proceso se desarrolló la tesis¹ titulada “Segmentación de imágenes de color utilizando el sistema de color modificado I1I2I3”, dando como resultado un conjunto de aplicaciones para realizar este proceso de manera semi-automática. Sin embargo estas aplicaciones fueron diseñadas con base en un formato de línea de comandos, lo cual limita su manejabilidad y eficiencia, además de que constituyen aplicaciones de propósito específico que se ejecutan de manera aislada. En este sentido se propone desarrollar una aplicación o módulo de software con una interfaz gráfica de usuario que permita facilitar el proceso de segmentación de imágenes en color, utilizando de igual manera el sistema de color modificado I1, I2, I3. Este programa deberá integrarse como un módulo adicional en la estructura operacional de la aplicación GIMP², la cual contiene una gran cantidad de herramientas y módulos prediseñados que son útiles para la ejecución de tareas asociadas con la manipulación y creación de imágenes. Además se propone integrar en el ambiente de GIMP los módulos encargados de la impresión y digitalización de

¹ Este sistema de color se describe con mayor profundidad en la sección 5.1.2

² Esta aplicación se trata en el apartado 5.1.1

imágenes. En consecuencia se reduce el número de aplicaciones involucradas en el proceso de la segmentación.

Con el desarrollo de esta aplicación se pretende describir los mecanismos así como la información necesaria para implementar módulos que se pueden integrar al sistema GIMP. Esto facilita la adaptación, escalabilidad y el proceso de desarrollo de aplicaciones, con la tarea de satisfacer las necesidades específicas que surgen en un ambiente en donde se manipulan, crean o procesan imágenes. Adicionalmente la necesidad de presentar esta información se debe a que la documentación para este tema en particular es escasa y confusa, además de estar redactada en el idioma inglés.

El desarrollo de los capítulos siguientes se encuentra dividido en un marco teórico y en una parte de implementación. Esto se hace con el propósito de mantener el contexto de la información presentada en cada capítulo, pues su contenido aunque esta relacionado, no es mutuamente dependiente.

Capítulo 2

Presentación de Linux y del software libre

Es preciso definir algunos conceptos importantes que se emplean en la redacción de este trabajo de tesis y que tienen una importancia fundamental en el entendimiento y el desarrollo de los conceptos mas generales presentados en los capítulos posteriores. En este sentido es importante detallar el concepto que enmarca a Linux, por medio del estudio de sus propiedades y características que le han conferido la reputación de un sistema operativo poderoso, robusto, estable y altamente funcional.

2.1 Software libre

Uno de las ideas básicas que requieren un tratamiento especial es precisamente el concepto de software libre.

En la clasificación de software libre se incluyen a todos los productos de software que son distribuidos con su código fuente totalmente disponible. Frecuentemente se asocia este concepto con la idea de software gratuito que se distribuye sin ningún costo, sin embargo esta no es una verdad absoluta, debido a que existen algunos productos de software que pertenecen a la categoría de software libre y son distribuidos en forma comercial, a un precio determinado.

En general la idea subyacente del software libre es la disponibilidad del código fuente con el propósito de potenciar el estudio, la adaptabilidad, la escalabilidad y la transmisión de los productos de software [1].

Existen un conjunto de lineamientos para el uso del software libre que se derivan de las licencias usadas para su desarrollo y distribución, es decir, el usuario se encuentra limitado en el uso de un determinado producto de software por las restricciones impuestas en los términos de la licencia bajo la cual dicho producto es distribuido, en este sentido para las licencias mas abiertas (como las BSD¹), terceras personas, entendidas estas como las personas que no desarrollaron el producto original, pueden apropiarse del código e incluso pueden incorporarlo en productos comerciales sin ningún problema y sin importar si realizaron modificaciones o no al código fuente, además, los creadores originales del producto generalmente reciben únicamente una mención en la leyenda de los derechos

¹ Berkeley Software Distribution

reservados del producto. Por supuesto existen otro tipo de licencias que son mucho mas restrictivas, como la licencia GPL², la cual establece que todo producto de software derivado o construido a partir de software GPL hereda la licencia de este último, es decir el “nuevo producto” continua siendo software libre GPL. Una restricción notable impuesta por este tipo de licencia establece que todo producto de software GPL no puede ser distribuido en forma comercial.

Es preciso señalar que la definición de software libre introducida en esta sección no engloba todo el significado de este concepto, pues el software libre no se encuentra asociado únicamente con el acceso o la disponibilidad del código fuente, sino además esta ligado estrechamente con un conjunto de lineamientos que se deben cumplir para preservar el espíritu de la “libertad del software”. Para entender con mayor precisión todos los detalles asociados con el concepto que distingue al software libre se puede consultar el sitio web del GNU³ project [2], el sitio web del Open Source project [3], o el sitio del FreeBSD project [4], en donde existe una amplia información del software libre en general, así como de las licencias GPL, LPGL⁴ y BSD, además se realiza una descripción más precisa de sus lineamientos e ideas más importantes.

El sistema operativo Linux punto central en el desarrollo de este trabajo de tesis se encuentra regido por una licencia GPL, en este sentido resulta muy importante para la justificación de este trabajo, clarificar los términos utilizados en el ambiente del software libre, así como describir las ventajas y características que presentan los productos de software que corresponden a esta categoría, en especial del sistema operativo Linux.

Frecuentemente se confunde el concepto de software libre con el de software gratuito e incluso con el concepto de software abierto, sin embargo cada uno de ellos promueve una idea fundamentalmente distinta. El software gratuito como su nombre lo indica se distribuye sin ningún costo, sin embargo esto no implica que el código fuente asociado deba ser proporcionado con el producto, lo cual generalmente no sucede. Algunos ejemplos de aplicaciones que entran en esta categoría son el navegador Explorer y la aplicación Eudora.

Por el contrario el software libre establece que cada producto de software debe distribuirse con su código fuente asociado, o al menos debe hacerse disponible públicamente (por supuesto bajo las restricciones comerciales apropiadas), pero no establece ninguna restricción asociada con la comercialización de tal producto, por lo cual

² GNU Public License: Licencia Pública General

³ Acrónimo recursivo para “Gnu No es Unix”

⁴ Lesser General Public License: Library GPL

es frecuente encontrar software libre que no es distribuido en forma gratuita. Los sistemas operativos Linux y FreeBSD, el servidor web Apache, y los compiladores C++ y GNU C son ejemplos de software que entran en esta categoría.

En el caso del software abierto, se puede afirmar que su filosofía promueve en cierto sentido la estandarización de la interfaz o arquitectura de comunicación de los productos de software, con el objetivo de posibilitar y facilitar la intercomunicación y la interoperabilidad de diferentes aplicaciones, independientemente de la plataforma de hardware o software para la cual fueron desarrolladas y sobre las cuales operan. A partir de la descripción anterior se puede inferir con relativa facilidad que las aplicaciones que pertenecen a la categoría de software abierto no proporcionan necesariamente su código fuente, únicamente buscan difundir su interfaz de comunicación y no, los procedimientos implementados en su código. Ejemplos de aplicaciones que son representativas de esta categoría son el Micro Kernel Mach y el lenguaje PostScript⁵.

2.2 Sistema operativo Linux

Linux es un clón del sistema operativo UNIX que corre en múltiples arquitecturas y que soporta un amplio rango de aplicaciones. Constituye en este sentido una implementación versátil de UNIX, distribuida gratuitamente bajo los términos de la Licencia GNU. Las dos características más importantes de Linux que lo diferencian del resto de los sistemas operativos en el mercado, constituye en primer lugar el hecho de que es un sistema libre, y se distribuye con su código fuente libremente accesible [5].

Linux ha experimentado un gran crecimiento y desarrollo en los últimos años, y es frecuente encontrarlo en gran cantidad de sectores, como en el sector empresarial en donde es instalado en ambientes distribuidos que operan sobre grandes infraestructuras de redes, realizando tareas de gran importancia y trascendencia para estas empresas. El mundo de Internet representa el sector con el grupo de soluciones más amplio y maduro que se fundamentan en el sistema operativo Linux, pues el crecimiento de ambos se ha presentado en forma paralela a lo largo de los años. En el ámbito educativo existe también una gran aceptación por este sistema operativo, pues en múltiples universidades de todo el mundo Linux es la plataforma más popular para dar cursos de programación y de diseño de sistemas operativos. También se usa en los sectores de la salud, el sector industrial, de las telecomunicaciones, entre otros, además un número cada vez mayor de usuarios en todo el mundo están usando Linux en sus equipos de cómputo simplemente para programar, entretenerse, o conocerlo a fondo.

⁵ Este término es explicado en la sección 4.2.1.1

El código fuente de Linux está siempre accesible, lo que ha permitido que una gran cantidad de programadores alrededor del mundo participen en su diseño y programación, los cuales se comunican por Internet, intercambiando código, comentando fallos, y arreglando los problemas que se presentan, todo esto en un entorno abierto. En consecuencia cualquier usuario con conocimientos de programación es bienvenido a sumarse al esfuerzo de desarrollo de Linux, bajo la premisa de que existe un interés primordial para producir un sistema de software que se fundamenta en la filosofía del software libre. En Linux (en comparación por ejemplo con los sistemas propietarios UNIX), no existe una cultura consolidada para los conceptos de desarrollo organizado, sistemas de control de versiones, información de errores estructurada o análisis estadísticos.

Normalmente, Linux es distribuido junto con otro software en lo que se conoce como distribución, en donde se empaqueta el Kernel o núcleo estándar de Linux, con una colección de utilerías y paquetes con licencia GNU. Las distribuciones se crearon con la finalidad de facilitar el proceso de instalación de un sistema bajo Linux, de tal forma que solo es necesario utilizar el CD-ROM o los disquetes de instalación de la distribución para tener un sistema completo con programas de aplicación incluidos.

Existen varias empresas que se dedican a producir las diferentes distribuciones de Linux, que suministran a precios bastante accesibles, en la actualidad las más importantes son:

- Red Hat Linux
- Caldera OpenLinux
- Debian GNU/Linux
- SuSE
- ESWare
- Slackware
- Eurielec
- MkLinux
- LinuxPPP

2.2.1 Características de Linux

Linux implementa la mayor parte de las características que se encuentran en otros sistemas UNIX, y ofrece además otras cualidades muy importantes. Las más sobresalientes se mencionan a continuación [5]:

- **Multitarea:** Linux puede ejecutar varios procesos al mismo tiempo. Para ello los recursos de procesamiento se distribuyen entre los programas (procesos) en ejecución, asignando niveles de prioridad a cada uno de ellos.
- **Multiusuario:** diferentes usuarios pueden estar trabajando en la misma máquina al mismo tiempo. Los distintos usuarios pueden conectarse a través de redes locales y terminales tontas, o mediante accesos remotos.
- **Multiprocesamiento Simétrico:** por medio de esta característica Linux puede sacar provecho de una plataforma de hardware con varios procesadores, distribuyendo entre ellos las tareas de ejecución de los distintos procesos activos.
- **Multiplataforma:** corre en muchas plataformas distintas, no sólo en aquellas basadas en los procesadores Intel.
- **Soporta la carga de ejecutables por demanda:** Linux sólo lee del almacenamiento secundario (disco), aquellas partes de un programa que están siendo usadas actualmente.
- **Política de copia en escritura (copy on write) para la compartición de páginas entre ejecutables:** esto significa que varios procesos pueden usar la misma zona de memoria para ejecutarse. Cuando alguno intenta escribir en esa memoria, la página (4Kb de memoria) se copia a otro lugar. Esta política de copia en escritura tiene dos beneficios: aumenta la velocidad y reduce el uso de memoria.
- **Memoria virtual usando paginación a disco:** este mecanismo utiliza almacenamiento secundario para mantener páginas de la memoria principal. Esto permite aumentar el tamaño virtual de la memoria principal del sistema, y en consecuencia es posible ejecutar programas de gran tamaño.
- **Memoria unificada:** la memoria se gestiona como un recurso unificado para los programas de usuario y para el caché de disco, de tal forma que toda la memoria libre puede ser usada para caché y éste puede a su vez ser reducido cuando se ejecuten grandes programas.

2.3 Conceptos de software libre

Una vez que se han abordado las ideas generales asociadas con el software libre, es importante profundizar en las ideas que sustentan su filosofía para establecer los beneficios principales que surgen de ella, así como sus desventajas o puntos débiles. Esto es importante para la justificación asociada con la utilización de un sistema de software determinado, en este caso del sistema operativo Linux.

2.3.1 La filosofía del software libre

En esencia el software libre pretende compartir el conocimiento asociado con el desarrollo de las aplicaciones de software, de esta manera promueve y facilita la participación de todos los usuarios o programadores, interesados en aportar su trabajo y su conocimiento, en un esfuerzo común, que busca el desarrollo continuo y la rápida evolución de los productos de software. Además los preceptos promovidos por el software libre permiten que un usuario pueda modificar o ajustar un sistema software a sus necesidades específicas, lo cual amplía el espectro de aplicabilidad de dichos productos de software, esto representa sin duda, una característica muy importante y ventajosa.

Frecuentemente la ideología del software libre es ignorada en el ambiente de la computación y del desarrollo de software, pues en muchas ocasiones se mantienen ocultos los detalles de la implementación de los productos de software, lo cual limita sin duda, toda la potencialidad que un usuario puede explotar de su sistema de cómputo. Este problema fue tratado por primera vez a principios de los 80's por Richard Stallman, quien decidió fundar el proyecto GNU, proyecto que actualmente se constituye como el precursor del movimiento del software libre. Las principales ideas establecidas por Richard Stallman son las siguientes[1]:

- **Libertad:** para copiar, difundir o modificar un programa.
- **Igualdad:** toda persona debería tener los mismos derechos sobre el software. En este sentido el distribuidor no debería tener privilegios sobre los usuarios a los cuales distribuye sus productos.
- **Cooperación:** este modo de trabajar busca fortalecer la comunicación de toda la comunidad de desarrolladores y así producir software cada vez más rentable y útil para todos.

Estos conceptos que expresan en gran medida la filosofía del software libre tuvieron una gran aceptación y una gran difusión en todo el mundo, sobre todo en el ambiente

científico y en el ambiente educativo. Adicionalmente la ideología del software libre comenzó a adquirir una gran cantidad de partidarios en todo el mundo, los cuales desde luego tenían diferentes ideas, concepciones, métodos y técnicas, pero todos creían básicamente en la difusión y la libertad del conocimiento en el área del desarrollo del software y en general de las ideas promovidas por la filosofía del software libre.

2.3.2 Elección del software libre

La razón de presentar los argumentos asociados con la conveniencia de utilizar software libre se encuentra íntimamente ligada con los argumentos a favor de la utilización del sistema operativo Linux como plataforma para la realización de tareas y el desarrollo de aplicaciones con un propósito específico, con la premisa de satisfacer las necesidades que presenta cada usuario en particular. Esta afirmación es válida debido a que Linux se constituye en si mismo como software libre y por lo tanto hereda todas sus características y cualidades más sobresalientes. La elección del software libre permite una pluralidad de soluciones sobre todo en el mundo de las computadoras personales, en donde se presenta una monopolización cada vez mayor de instituciones motivadas por intereses primordialmente comerciales.

La elección del software libre debe necesariamente estar influenciada por sus propias cualidades, y en este sentido es importante describir las más sobresalientes [1]:

- **Acceso al código fuente:** esta cualidad es posiblemente la más importante de todas y constituye el factor de mayor peso en la elección de una aplicación de software libre. La disponibilidad del código fuente permite y facilita el estudio, el entendimiento, la adaptación, corrección, distribución y refinamiento del software.
- **Confiabilidad:** la calidad de un producto de software libre se deriva de la participación de una gran cantidad de personas quienes aportan su inteligencia y su experiencia para producir aplicaciones altamente confiables. El grado de refinamiento de una aplicación de software libre mejora continuamente con el paso del tiempo, resultado del gran número de correcciones y pruebas realizadas por los programadores y usuarios de las aplicaciones correspondientes. Además en el mundo del software libre no existen presiones comerciales o de mercado que obliguen a los desarrolladores del producto a liberar una aplicación aún cuando está no ha sido probada exhaustivamente y por lo tanto no se encuentra en un estado “satisfactorio”.
- **Portabilidad:** si uno de los objetivos fundamentales consiste en desarrollar software que sea útil y que satisfagan una necesidad específica, entonces su ámbito

de aplicación debe ampliarse en forma natural, hacia otras plataformas o ambientes de cómputo diferentes, a los que fueron inicialmente considerados. De esta forma al incrementar su disponibilidad, también se mejora su portabilidad y su confiabilidad.

- **Universalidad:** una de las cualidades más importantes del software libre es el carácter universal en el formato de los datos utilizados. El hecho de distribuir el código fuente asegura que los usuarios puedan entender la estructura de los datos y puedan escribir módulos que posibiliten la interacción de aplicaciones que emplean formatos en esencia distintos.
- **Desempeño:** en el movimiento del software libre participan una gran cantidad de profesionistas y científicos con una mentalidad bastante crítica y con una actitud natural de perfeccionamiento y optimización en el desempeño de los algoritmos utilizados en las aplicaciones de software, los cuales generalmente son el producto de investigaciones avanzadas, y son aplicados en diferentes situaciones para probar su efectividad, de esta forma el software libre tiene en general un buen desempeño. En el ambiente del software libre es una tarea común refinar grandes porciones de código con el propósito de mejorar continuamente el desempeño de las aplicaciones. Además es una práctica común distribuir versiones inestables o iniciales de un producto de software para que su funcionamiento pueda ser exhaustivamente probado por prácticamente todos los usuarios con la disponibilidad para hacerlo, todo esto con la finalidad de obtener un producto estable, con un grado elevado de desempeño.
- **Interoperabilidad:** esta cualidad es uno de los efectos principales que surge ante la necesidad de desarrollar aplicaciones útiles y de alto desempeño, en este sentido de muy poco nos serviría un ambiente en donde sus aplicaciones pudieran interactuar entre sí, pero con la limitación de operar con otras aplicaciones desarrolladas para plataformas de hardware y software esencialmente distintas.
- **Trabajo colectivo:** la ideología del software libre promueve la participación colectiva de numerosos usuarios en las tareas asociadas con el desarrollo, utilización y verificación de un producto de software, de esta forma se produce un fenómeno de reducción en el tiempo de respuesta ante cualquier suceso que se presenta durante el ciclo de vida de la aplicación, lo cual resulta especialmente importante sobre todo ante la detección de errores que requieren de la generación rápida de soluciones.

Las cualidades del software libre constituyen argumentos importantes que muestran la conveniencia de utilizar Linux como plataforma para el desarrollo de aplicaciones, así como para la realización de tareas con un propósito determinado, tareas que pueden

asociarse sin ningún problema a los requerimientos del procesamiento digital de imágenes. Sin embargo es necesario presentar argumentos adicionales desde diferentes perspectivas, con el objetivo de apoyar la justificación de Linux como una plataforma altamente conveniente para el desarrollo del presente trabajo de tesis y por supuesto de las tareas subyacentes realizadas.

2.3.2.1 Argumentación de mercado

En esta sección se introducen argumentos directamente asociados con la distribución de mercado en el ramo de los sistemas operativos, con el objetivo de mostrar el crecimiento de Linux y su rápida difusión en diferentes áreas.

En estudios realizados por la empresa International Data Corporation (IDC), se puede comprobar y observar el gran crecimiento que ha registrado Linux en el mercado de los sistemas operativos, sobre todo en los últimos años.

Según IDC, Linux experimentó un crecimiento record en 1998, pues alcanzó el 17.2% de la distribución del mercado de los sistemas operativos para servidores, lo cual representó una tasa de crecimiento del 212% en el año, es decir duplicó en el periodo de doce meses el número de organismos o instituciones que eligieron a Linux como la plataforma mas apropiada para operar sobre sus servidores. Este fenómeno fue realmente notable y significativo, sobre todo si se considera que en 1997 Linux era colocado en la ignorada clasificación de otros⁶, pues sus números en ese momento lo situaban en tal posición [1].

IDC atribuyó el crecimiento de Linux en 1998 a su gran desempeño, a su bajo costo y al fuerte sentimiento anti-Microsoft que predominaba en esos días, fenómeno que se mantiene de hecho hasta el momento actual. La tendencia en el crecimiento de Linux se confirmó en 1999, pues se acreditó el 24% del total del mercado, con un crecimiento del 39%, lo cual representó mas de cuatro de veces el crecimiento de su más cercano perseguidor [1].

En el año 2000 Linux se apropió del 28% del segmento mercado de los sistemas operativos, lo cual representa un crecimiento significativo para un mercado tan competitivo. Según IDC, el ritmo de crecimiento de Linux se mantendrá a un nivel sostenido durante los próximos años, con lo que se convertirá en un serio rival de Microsoft y su sistema operativo Windows.

⁶ En esta categoría se agrupan los elementos menos significativos

IDC estima que las adquisiciones totales de Linux crecerán mucho más rápido en comparación con cualquier otro ambiente operativo cliente o servidor hasta el 2004. IDC estima que las “ventas” de Linux se incrementarán a una tasa compuesta de crecimiento anual (CAGR) del 25% hasta el 2004, comparado con el 10% CAGR que se estima para todos los sistemas operativos cliente en conjunto, y con el 12% CARG previsto para todos los ambientes operativos servidor combinados.

En forma adicional, la agencia de búsqueda NetCraft Inc.[7], afirma que el 29% de los servidores Web en Internet se basan en el sistema operativo Linux, haciéndolo el más difundido para la gestión de sitios web, los resultados muestran la importancia tomada por Apache en más de 6 millones de sitios operacionales. La agencia IOS Counter, encargada de calcular las estadísticas asociadas con los servidores en Internet, confirmó a los sistemas operativos Linux y FreeBSD como las plataformas dominantes para los servidores en Internet.

Estos datos son altamente ilustrativos del gran crecimiento y del fenómeno de rápida expansión que ha experimentado Linux en el mercado de los sistemas operativos en los últimos años, todo esto sin considerar que en el ámbito del software libre resulta muy difícil establecer una estadística lo suficientemente exacta para describir la situación actual de un sistema como Linux, pues potencialmente cualquier persona puede tener acceso a una copia de este software y puede instarlo en el número de computadoras que desee, sin prácticamente ninguna restricción.

2.3.2.2 Argumentación financiera

Indudablemente al hablar de los aspectos financieros a favor de la utilización del software libre, las conclusiones son obvias. Como primer argumento se puede afirmar que los precios para adquirir un producto de software libre son muy bajos en comparación con los costos que representa adquirir un producto de software comercial. Puede parecer confuso hablar de costos en el ámbito del software libre, sin embargo incluso si el producto esta disponible libremente en Internet, se producen costos asociados con el sólo hecho de descargarlos, además los distribuidores de tales productos generalmente establecen un precio que aunque es bajo en comparación con su contraparte comercial, sí es significativo en muchos de los casos.

Otro de los argumentos financieros a favor del software libre consiste en la ausencia del concepto de licencia, ya sea por equipo de cómputo o por servicios suplementarios. Esta característica constituye una ventaja importante si se compara con el software comercial,

pues los costos no tienen que incrementarse con la incorporación de nuevas unidades de cómputo o con la adquisición de nuevos módulos para los productos de software ya presentes, en una arquitectura o entidad en general.

Los productos del software libre producen además mejores resultados al calcular el índice de costo total de propiedad (TCO⁷) de un sistema determinado. El TCO provee una metodología (basada en ciclo de vida) para comprender y determinar el costo promedio en la industria de un grupo concreto de bienes, que en este caso podemos enfocarlo a las soluciones de cómputo basadas en software libre. Estos resultados se deben en cierta medida a que los costos derivados de la administración del software libre son bajos en general, por ejemplo, si consideramos a los sistemas Linux o FreeBSD, estos pueden ser manipulados y administrados totalmente en forma remota, ya sea a través de una terminal de línea de comandos (generalmente con la aplicación telnet) o por medio de una interfaz gráfica, este hecho favorece la eficiencia y la sencillez asociada con las tareas de administración de estos sistemas, con la correspondiente reducción de costos. Adicionalmente estos sistemas constituyen verdaderas aplicaciones multiusuario, lo cual aumenta invariablemente el grado de eficiencia en las operaciones de administración. Si a esto le añadimos el soporte nativo para el protocolo PPP⁸ y la existencia de sistemas para la implementación de conexiones seguras, entonces es posible que la administración pueda ser realizada por una entidad externa, efectuando outsourcing.

La reducción de costos se puede presentar aún en aquellos casos en donde el software libre no satisface completamente las necesidades requeridas para una solución computacional determinada. En este caso únicamente será necesario adquirir software comercial para completar las tareas que no hayan sido cubiertas por su contraparte libre, generando con ello una solución heterogénea, que es posible implantar gracias a las cualidades de desempeño, portabilidad e interoperabilidad del software libre.

Como argumento final en relación a la reducción de costos, se puede afirmar que gracias al buen desempeño que presentan los productos de software libre estos pueden operar sobre plataformas de hardware consideradas como "obsoletas", calificativo aplicable tomando como punto de referencia los requerimientos exigidos por otros sistemas operativos o programas de cómputo. En este sentido es posible implantar un sistema con un rendimiento apropiado sobre una plataforma anticuada, todo esto a un costo bajo, y con la posibilidad de escalar progresivamente hacia un sistema de mayor potencia.

⁷ Total Cost of Ownership: costo total de propiedad.

⁸ Point to Point Protocol: protocolo de punto a punto.

2.3.2.3 Argumentación técnica

Una de las características más sobresalientes que presenta una solución basada en software libre consiste en su elevado índice de confiabilidad, factor que le permite operar en condiciones normales durante periodos largos de tiempo, lo cual resulta claramente ventajoso pues las labores de administración, mantenimiento, control y supervisión se hacen menos complicadas e intensivas. En este punto esta por demás afirmar que el sistema operativo Linux presenta dicho comportamiento, pues se han reportado numerosos casos en donde los sistemas bajo esta plataforma, han logrado un periodo de funcionamiento en condiciones normales de varios meses e incluso años de operación continua.

Uno de los efectos más notables que se producen como resultado de la gran portabilidad del software libre (reforzado por el establecimiento de normas y estándares), consiste en la producción de sistemas que heredan las mismas cualidades de portabilidad con otras plataformas de hardware y software, lo cual favorece la migración de una aplicación libre hacia plataformas con un mayor desempeño, y con un mayor potencial de evolución. Esta migración generalmente se hace necesaria cuando el desempeño o los servicios proporcionados por el software libre ya no satisfacen los estándares previstos y se vuelven insuficientes para cubrir con las necesidades computacionales requeridas.

El último argumento técnico en favor del software libre consiste en su gran modularidad, esta cualidad favorece la implantación de soluciones que se ajustan a las necesidades y a la situación específica que guarda una arquitectura o solución en particular. Si la arquitectura soporta por ejemplo una gran cantidad de servicios y tiene un grado elevado desempeño, entonces es posible implantar una solución de software completa es decir con todos los módulos necesarios para proporcionar cada uno de los servicios solicitados, y si por el contrario la plataforma de hardware es más limitada entonces la solución en software se puede adaptar, eliminando los módulos que no se requieran o que no tengan una utilidad específica.

2.3.2.4 Argumentación de soluciones

Este aspecto es posiblemente el más importante de todos porque refleja el espectro de aplicabilidad y utilidad de los productos que se fundamentan en la ideología del software libre. En este sentido de muy poca utilidad sería una aplicación que no cumpliera satisfactoriamente con los requerimientos o las demandas para los cuales fue desarrollado, especialmente en los diferentes ámbitos en donde su aplicabilidad es fundamental como en el sector empresarial, y en gran medida en los sectores educativo, científico y tecnológico.

Además una aplicación sin un propósito útil y sin un buen desempeño, invariablemente esta destinada al fracaso, al olvido y a su consecuente extinción.

La aplicabilidad del software de código abierto se ha enfocado históricamente a la construcción de servidores para Internet o Intranets, debido al paralelismo presente en el desarrollo de ambos, de esta forma en las soluciones de software libre para el internet se tienen servidores web, servidor FTP⁹, servidor DNS¹⁰, servidor de correo electrónico, servidor de servicios en línea para grupos de discusión, servidor proxy, servidor cache para el web. Todos estas herramientas se encuentran disponibles en la distribución estándar de Linux, y se pueden instalar sólo aquellos que sean necesarios para cumplir con un conjunto de tareas específicas, dependiendo de los servicios ofrecidos y de los requerimientos de una solución determinada.

El segundo dominio preferencial para el software libre es el dominio asociado con los servicios de impresión y de archivo, en donde los clientes pueden ser múltiples, como los clientes Unix, clientes de Windows, los clientes Novell o los clientes MacIntosh. Todos estos paquetes de software también se encuentran disponibles en la distribución estándar de Linux.

El área de la computación constituye otro de los dominios en los cuales un sistema como Linux puede ofrecer soluciones altamente satisfactorias y productivas, incluyendo el soporte de múltiples procesadores, la implementación de clusters con múltiples nodos, el soporte de interfaces de red de alta velocidad, la implementación de servidores de fax centralizados, servidores de archivo/respaldo, servidores de base de datos, entre muchas otras aplicaciones.

Desde la perspectiva de una configuración cliente existen también un gran número de soluciones basadas en software comercial y en productos que se fundamentan en la ideología del software libre. En el ámbito del Internet se tienen herramientas como los navegadores gráficos para el web, o navegadores en modo texto, terminales gráficas para consultar el correo, o su contraparte en modo texto. Adicionalmente existen una gran variedad de herramientas destinadas a la realización de las múltiples tareas que son indispensables para un sistema de cómputo de propósito general, como los visores de archivos PDF, las herramientas para la manipulación de imágenes, procesadores de texto, las suites comerciales de office, herramientas para la administración del sonido, aplicaciones para copiar discos compactos, emuladores comerciales y libres para varios

⁹ File Transfer Protocol: Protocolo de Transferencia de archivos.

¹⁰ Domain Name Service: Servicio de Nombres de Dominios

sistemas, compiladores e intérpretes para todos los lenguajes, o ambientes gráficos, entre muchas otras aplicaciones.

2.3.2.5 Argumentación de servicios

Uno de los aspectos más resaltados en detrimento de la utilización y expansión del software libre constituyó durante mucho tiempo la falta de una infraestructura de servicios totalmente desarrollada que proporcionara soporte, mantenimiento y asesoría técnica a los productos de código abierto. Este hecho tomó una mayor relevancia sobre todo en el ámbito empresarial, pues situaciones asociadas con una prolongada interrupción o un malfuncionamiento frecuente de los sistemas de cómputo, constituían un factor de riesgo de elevada probabilidad, pues no existía un respaldo técnico que pudiera subsanar o resolver de manera eficiente y eficaz los diversos problemas que sugieran durante la operación normal del sistema. Este hecho representaba para las empresas una situación con el potencial para detonar un ambiente de grandes pérdidas, principalmente en el ámbito económico, por lo que no estaban dispuestas a tomar un riesgo de tal magnitud.

Sin embargo el panorama ha cambiado de manera favorable en los últimos años y en la actualidad existen muchos proveedores de servicios o empresas dedicadas a la manufactura de hardware, como Hewlett Packard, Compaq o IBM que controlan estas soluciones y proporcionan soporte para ellas. Además existe una gran cantidad de información y recursos en Internet dedicadas a los productos de software libre, como son las listas de correo especializadas, los numerosos grupos de noticias en línea, y la gran cantidad de recursos bibliográficos presentes en el web, como manuales, reportes, tutoriales, entre otros.

2.3.3 Falsas ideas sobre el software libre

Existen diversas concepciones erróneas en relación a las cualidades del software libre que no tiene un fundamento válido en la actualidad, en este sentido resulta importante clarificar estos argumentos erróneos.

- **No hay soporte, ni entrenamiento:** como ya se mencionó en la sección previa, existen en la actualidad un gran número de empresas que ofrecen soporte y asesoría técnica, además de cursos especializados de capacitación para los productos de software libre. Por ejemplo la empresa RedHat proporciona hoy en día, soporte para sus soluciones, que se basan en la distribución de Linux que esta misma empresa produce. Además existen en el mercado diversos organismos que imparten cursos genéricos de administración de redes, así como de Unix, que constituyen un

fundamento importante como ayuda para el aprendizaje de las tareas de administración de un sistema como Linux. En una sección previa se mencionó también que existen en el Internet una gran cantidad de recursos de información enfocados al software libre, lo cual permite y facilita que cualquier usuario pueda aprender por sus propios medios o en forma autónoma los conceptos asociados con la administración y uso de una aplicación determinada.

- ***No existe suficiente documentación:*** esta es una idea completamente equivocada, debido a que en la actualidad existen numerosas fuentes de información dedicadas específicamente al software libre, una de las más importantes corresponde al *Proyecto de Documentación de Linux*, el cual contiene más de 300 documentos para Linux, muchos de los cuales están traducidos al español y se encuentran libremente disponibles. Adicionalmente existen una gran cantidad de sitios web que contienen numerosos manuales y tutoriales, así como grupos de noticias, y múltiples listas de discusión, fuentes que generalmente satisfacen completamente las necesidades de información en este ámbito. Existen también algunos editores como O'Reilly y SSC, que se especializan en la publicación de libros enfocados específicamente al software libre, los cuales son escritos en la mayoría de los casos por los mismos programadores de las aplicaciones.
- ***Un producto de software libre tiene un desempeño pobre:*** el hecho de que las aplicaciones de software libre tengan un precio muy bajo en comparación con su contraparte comercial, no significa que sean de menor calidad o que presenten un desempeño muy limitado, esta cualidad por el contrario se deriva únicamente de la ideología y los conceptos de código abierto, y de hecho la gran mayoría de las aplicaciones que se fundamentan en esta filosofía como ya se ha mencionado en secciones anteriores, han demostrado en la práctica un excelente grado de desempeño y confiabilidad, que incluso ha superado en gran medida y en muchos de los casos al desempeño que presentan los productos comerciales.
- ***Linux resulta difícil de instalar:*** este hecho también constituyó durante los primeros días de Linux, uno de los principales argumentos que limitaron su expansión, sin embargo gracias a las distribuciones RedHat, Mandrake, LinuxPPP, entre otras, la situación ha mejorado considerablemente, y en la actualidad la instalación completa del sistema operativo Linux no es más difícil que el proceso de instalación de otros sistemas operativos como Unix o Windows NT. En este punto resulta muy importante resaltar que el soporte de hardware en Linux no es totalmente satisfactorio pues existen varias empresas en la actualidad que ocultan los detalles de comunicación y control de sus dispositivos y no ofrecen controladores para Linux, sin embargo este comportamiento ha comenzado a cambiar en favor de Linux pues las empresas han reconocido el enorme potencial

que tiene este sistema operativo y su gran difusión y penetración en los diferentes sectores de mercado, los cuales representan indudablemente una gran cantidad de clientes potenciales para la adquisición de sus dispositivos. Para verificar la compatibilidad de un dispositivo de hardware con Linux se puede consultar el documento [6], así como las múltiples páginas de las empresas que los producen y distribuyen, en donde generalmente existe suficiente información al respecto.

- ***El software libre no es adecuado para tareas pesadas que requieren un alto desempeño:*** si bien es cierto que en este aspecto existen aplicaciones comerciales que en ciertos casos presentan mejores registros de rendimiento en comparación con Linux, también es cierto que esta situación constituye uno de los principales desafíos que está siendo atacado por la comunidad de programadores que intervienen en el desarrollo de Linux, y desde luego las próximas versiones de su Kernel contarán con características cada vez más potentes que le permitirán incrementar consistentemente su desempeño integral.
- ***El escritor de una aplicación de software libre no se hace responsable de su mal funcionamiento:*** en realidad las licencias de software protegen a los desarrolladores de las responsabilidades asociadas con los problemas que se presentan en sus productos de software, sin embargo esto no significa que dichas aplicaciones se encuentren aisladas, pues siempre existen desarrolladores que están dispuestos a ayudar cuando se presenta algún problema y tratan de corregirlo en el menor tiempo posible, situación que es muy común en la comunidad del software libre.

2.3.4 Problemas reales del software libre

Por supuesto no solamente existen ventajas derivadas de la ideología del software libre, también existen diversos problemas que se presentan en las aplicaciones que se fundamentan en esta concepción del software, muchas de las cuales no tienen una importancia significativa, pues su presencia es generalmente temporal gracias a la rápida atención de la comunidad de programadores, sin embargo otros problemas se desprenden directamente de la filosofía del software libre y por lo tanto son más difíciles de eliminar.

Una de los efectos problemáticos más evidentes del software libre se derivan del desarrollo de múltiples aplicaciones para una misma solución, lo cual generalmente dificulta el proceso de selección de la opción más apropiada por parte de los usuarios inexpertos, por ejemplo para la configurar un servidor de impresión también es posible elegir entre diversas aplicaciones como el LPD, LPRng o CUPS entre otros, de la misma manera existen diversas alternativas en la selección de una distribución de Linux, como RedHat, LinuxPPP, SuSE, Slackware, Mandrake o Debian, entre las más representativas.

Esto desde luego no representa una gran desventaja sobre todo para los usuarios expertos, pues con base en su conocimiento y en su experiencia son capaces de discernir en forma eficiente entre las distintas opciones que existen y elegir la más apropiada de todas para cubrir la necesidades específicas que se les presentan.

Otro de los problemas, que surge también en el ambiente del software libre es la necesidad de contar con una fuerte base de conocimientos en el área de Internet y en los aspectos relacionados con la administración de los sistemas Unix, con el objetivo de poder administrar de forma eficiente y eficaz tales soluciones, pues en realidad el poder disponible en una aplicación es proporcional a la capacidad de sus administradores para explotarlo. Este hecho resulta especialmente importante sobre todo en un ambiente de continuo desarrollo en donde las aplicaciones evolucionan constantemente y surgen aplicaciones que ofrecen características más novedosas y potentes, ante esta situación los administradores se ven obligados a ampliar sus conocimientos frecuentemente con el objetivo de extraer el mayor beneficio de las nuevos desarrollos. Un argumento a favor en este sentido constituye el hecho de que el conocimiento y las habilidades adquiridas no se desechan, pues realmente en la mayoría de los casos no es necesario re-aprender todo los aspectos asociados con una versión u otra, o con una aplicación u otra, especialmente en el ámbito del Internet. Otro de los factores importantes relacionados con este tema es el hecho de que la documentación para las aplicaciones de software libre se encuentran escritos en su mayoría en el idioma inglés, de tal manera que resulta útil contar con una base sólida en el conocimiento técnico de este lenguaje.

Un argumento final que representa un problema para el software libre en la actualidad, es la falta de convencimiento de muchos administradores y empresas que todavía no creen en las cualidades y ventajas que ofrecen las aplicaciones de código abierto, misma que ya han sido presentadas en su mayoría en las secciones previas de este documento. Afortunadamente este no es un fenómeno generalizado y existen importantes empresas que han enfocado su atención a este tipo de aplicaciones, como por ejemplo los esfuerzos emprendidos por las importantes compañías HP, Compaq o IBM, para ofrecer en sus productos el soporte necesario del sistema operativo Linux. De tal manera que esta situación mejora día a día y en un futuro no muy lejano indudablemente Linux constituirá una de las alternativas más predominantes en el mercado de los sistemas operativos y contará con el respaldo de una gran de número de empresas encargadas de la producción de hardware y software, así como de los profesionistas, usuarios y administradores en el área de la computación.

Capítulo 3

Sistema de digitalización de imágenes en Linux

En esta sección se describen los procedimientos necesarios para la operación de un escáner bajo el entorno del sistema operativo Linux. Desafortunadamente en la actualidad el número de escáners con compatibilidad para Linux es limitado y con frecuencia aquellos que son soportados tienen una funcionalidad moderada. Sin embargo la situación está cambiando favorablemente pues los fabricantes de tales dispositivos están centrando su atención al creciente y amplio grupo de usuarios de Linux, que representan un mercado potencial muy importante que difícilmente resulta conveniente ignorar.

Una gran variedad de escáners operan utilizando un protocolo propietario, esto significa que estos dispositivos no utilizan un protocolo o lenguaje de comunicación estándar, y por el contrario implementan uno propio, cuyo mecanismo de operación es conocido únicamente por el fabricante del dispositivo. Es técnicamente posible comunicarse desde Linux con cualquier dispositivo conectado a los puertos de comunicación de un sistema de cómputo, como los puertos USB, paralelo, serial o una interfaz SCSI, sin embargo para realizar tareas de control y administración sobre estos dispositivos, también es necesario conocer los detalles del lenguaje de comunicación que utilizan. Desafortunadamente muchos fabricantes presentan una actitud bastante radical y no ofrecen ningún tipo de información sobre los protocolos que utilizan sus productos. En estos casos la única solución consiste en utilizar técnicas de ingeniería inversa para dilucidar los detalles del protocolo subyacente, antes de poder escribir controladores para Linux. Desafortunadamente la aplicación de estas técnicas poco convencionales, requieren de una gran cantidad de conocimientos técnicos del dispositivo, así como del Kernel de Linux, en especial de la arquitectura de los diferentes subsistemas de comunicación, por tal razón, la mayoría de los dispositivos de hardware que requieren de la aplicación de ingeniería inversa, no son soportados en Linux.

Con el propósito de investigar el estado y el grado de soporte para múltiples modelos de escáners en Linux, se puede revisar el sitio [9], en donde se mantiene información actualizada acerca de nuevos desarrollos, proyectos en ejecución y noticias relacionadas con la liberación de la documentación para los protocolos propietarios de comunicación empleados por los escáners. Este sitio se enfoca básicamente en dispositivos de digitalización de imágenes que utilizan el puerto paralelo para comunicarse con el sistema operativo.

En forma adicional se puede revisar el sitio [10], en donde existe una larga lista de escáners que son soportados por el sistema operativo Linux. Este sitio constituye uno de los recursos más completos que existen en la red en donde se puede encontrar información relacionada con la compatibilidad, instalación y operación de los dispositivos de digitalización de imágenes bajo Linux. En este sentido antes de adquirir un escáner que será usado en una plataforma con este sistema operativo es conveniente consultar la información presente en este sitio para realizar la elección más apropiada.

3.1 Generalidades

3.1.1 Introducción al sistema SANE

SANE¹ es el acrónimo para “ahora es fácil el acceso a un escáner”. SANE implementa una interfaz para programas de aplicación (API²) que provee un “acceso estandarizado” a dispositivos de adquisición digital de imágenes, incluyendo escáners de cama plana, manuales, cámaras de video, cámaras de imagen fija y tarjetas de video [10]. El código fuente para el SANE API esta escrito en lenguaje C, diseñado para ejecutarse sobre plataformas UNIX, incluyendo desde luego al sistema operativo Linux. El código fuente de SANE se rige bajo las condiciones de la licencia GPL, de esta forma el SANE API sigue los fundamentos del software libre, logrando con ello llevar al dominio público los esfuerzos en su desarrollo, así como su difusión y utilización. El SANE API constituye una capa de software estandarizada para la adquisición de imágenes en sistemas UNIX, esta característica facilita el desarrollo de aplicaciones enfocadas al procesamiento digital de imágenes en estos sistemas.

En la actualidad existen múltiples paquetes de software para Linux que incluyen controladores para dispositivos de hardware encargados de la adquisición digital de imágenes. En la mayoría de los casos estos paquetes son suficientes para satisfacer los requerimientos de una determinada aplicación y un determinado dispositivo, sin embargo frecuentemente resulta difícil determinar el paquete de software más apropiado para controlar un dispositivo específico. Desde luego esta variedad de soluciones, nos ofrece una gama de múltiples posibilidades de las cuales podemos elegir la que mejor se adapte a nuestras necesidades, sin embargo en muchas ocasiones este proceso de elección puede resultar complicado, sobre todo si nuestras opciones presentan características y propiedades dispares, o simplemente no se ajustan en su totalidad a nuestros requerimientos.

¹ Acrónimo para: “Scanner Access Now Easy”

El principio bajo el cual se desarrolló SANE está orientado precisamente a proporcionar una solución a este dilema. La idea básica detrás del desarrollo de SANE es simple: Si se tiene una interfaz general y perfectamente definida para programas de aplicación (API), será fácil escribir aplicaciones, independientemente de los controladores de los dispositivos subyacentes, y viceversa. Esto se puede explicar desde dos puntos de referencia, esto es, en el primer caso el programador de un nuevo controlador, únicamente se tendría que preocupar por ajustar su desarrollo y por incluir en su controlador toda la funcionalidad establecida en la especificación de la interfaz de SANE, sin tener que preocuparse por la necesidad de desarrollar una aplicación para su controlador. En el otro caso, el programador de una nueva aplicación, encontrará un beneficio substancial con este enfoque, por el hecho de que le permite aislar su desarrollo de los requerimientos de administración operativa de bajo nivel asociada con los dispositivos de hardware que en determinado momento tenga que controlar. De esta forma "SANE permite mantener una misma interfaz en forma consistente, independientemente del dispositivo particular en operación".

SANE, no se ha significado como el primer intento por desarrollar una interfaz universal o estándar para el control de dispositivos encargados de la digitalización de imágenes. Algunos esfuerzos encaminados hacia el mismo objetivo se han realizado en el desarrollo de PINT y TWAIN, sin embargo cada uno de ellos presenta deficiencias en algún sentido. PINT por ejemplo, es una interfaz primitiva para Linux que opera a nivel del Kernel, enfocada al control exclusivo de escáners manuales, limitando con ello su aplicabilidad con otros tipos de dispositivos para la adquisición digital de imágenes. Por otro lado, TWAIN es un paquete de software que se fundamenta en muchos de los principios que soportan a SANE, sin embargo existen diferencias entre ellos, en particular se puede recalcar que TWAIN incorpora la interfaz gráfica de usuario que permite la manipulación del dispositivo en el propio controlador, en este sentido un controlador TWAIN está diseñado utilizando las especificaciones de un GUI API³ en particular (que puede ser de Win32 o el API de Mac). Esta característica de TWAIN es particularmente ineficiente en ambientes de red, puesto que frecuentemente el controlador del dispositivo debe ejecutarse en una máquina y la aplicación en otra, por ejemplo, si se tiene una LAN⁴ en donde múltiples computadoras se comunican entre sí, pero sólo algunas de ellas tienen acceso a dispositivos de adquisición digital de imágenes, entonces resulta muy conveniente el acceso transparente a tales recursos desde las computadoras remotas. SANE por el contrario traslada la interfaz gráfica al nivel de la aplicación, es decir, establece una clara

² Application Programming Interface: Programa de Aplicación de Interfaz.

³ Graphical User Interface API: API para Interfaz de Usuario Gráfica.

⁴ Local Area Network: Red de Área Local

separación entre el controlador y la interfaz de usuario, dando paso con ello al soporte para la adquisición transparente de imágenes a través de una red [10].

SANE se fundamenta en la idea de facilitar el acceso a un escáner, para lograr este objetivo busca reducir la complejidad asociada con la implementación del API y al mismo tiempo soportar todas las características requeridas por los dispositivos de adquisición de imágenes más modernos. En particular SANE debería ser lo suficientemente robusto para controlar dispositivos tales como: escáners, cámaras de video, cámaras de imagen fija, tarjetas de video, así como dispositivos virtuales que incluyen por ejemplo a los filtros de imágenes.

3.1.2 Módulos que integran la distribución SANE

En general la distribución SANE incluye controladores para los siguientes dispositivos [10]:

- *Connectix QuickCam (de color y monocromático).*
- *Algunos escáners Epson SCSI.*
- *Escáners Hewlett-Packard ScanJet SCSI.*
- *Escáners Microtek SCSI.*
- *Escáners Mustek SCSI de cama plana.*
- *Dispositivos PINT:* PINT es una interfaz del Kernel-Unix para NetBSD, OpenBSD y SunOS. Un controlador PINT en SANE permite el acceso a cualquier escáner para el cual existe soporte bajo PINT.
- *La mayoría de los escáners UMAX SCSI.*

La distribución SANE incluye además las siguientes aplicaciones:

- *El programa scanimage:* este programa permite realizar tareas de administración, control y supervisión sobre los dispositivos instalados en el sistema y con soporte para SANE, tales como escáners o cámaras. Esta diseñado con base en una interfaz de línea de comandos, es decir, los dispositivos son controlados por medio de opciones presentadas a través de una línea de comandos. Normalmente, una vez que estas opciones son procesadas, el programa scanimage realiza la digitalización de la imagen. Los datos de la imagen son escritos hacia la salida estándar en los formatos soportados por SANE, específicamente el formato PBM para imágenes binarias, el formato PGM para imágenes en escala de grises, el

formato PPM para imágenes de color y el formato TIFF que se utiliza para imágenes en color, en escalas de grises o binarias.

- **El programa xscanimage:** constituye la versión gráfica del programa scanimage. Este programa se distribuye como una aplicación individual y como una extensión de GIMP⁵. El programa xscanimage permite la previsualización y la digitalización de imágenes individuales y puede ser invocado utilizando la línea de comandos o a través del ambiente gráfico del sistema GIMP. En el primer caso, xscanimage se ejecuta con una aplicación aislada y almacena las imágenes digitalizadas en cualquiera de los formatos soportados en SANE, ha saber, PBM, PGM o PPM. Cuando xscanimage es invocado a través del sistema GIMP, las imágenes digitalizadas son transferidas al entorno de ejecución de esta aplicación, para su procesamiento correspondiente. En este sentido es posible utilizar las herramientas y módulos instalados en GIMP para realizar tareas asociadas con la manipulación de estas imágenes. La aplicación xscanimage también tiene acceso a los dispositivos de adquisición de imágenes por medio de la interfaz implementada por SANE. La lista de dispositivos disponibles depende del hardware instalado y de su configuración.
- **La aplicación xcam:** que incluye una interfaz gráfica de usuario enfocada al control de cámaras que producen un grupo continuo de imágenes (tal como la Connectix QuickCam).
- Otro de los componentes más importantes de la distribución SANE corresponde al demonio de red llamado saned. Este demonio posibilita el acceso transparente hacia dispositivos remotos a través de una red, por medio de este mecanismo es posible controlar de manera local cualquier dispositivo soportado por SANE (que puede ser una cámara) localizado en un sitio remoto, todo esto por supuesto, utilizando las facilidades que otorga el Internet. El demonio **saned** permite el acceso de clientes remotos hacia dispositivos de adquisición de imágenes disponibles en una máquina local.

La distribución de SANE se encuentra dividida en dos paquetes:

- **sane-backends:** incluye a los backends o controladores para el escáner, además de la aplicación scanimage, el demonio para la impresión en red saned, y la documentación del SANE-API.
- **sane-frontends:** incluye a las aplicaciones gráficas xscanimage y xcam.

⁵ Esta aplicación se trata en el apartado 5.1.1

3.2 Implementación

3.2.1 Instalación del sistema de digitalización de imágenes para Linux utilizando SANE

Este proceso se puede realizar en dos fases:

- Instalación de los módulos necesarios para el soporte del escáner en el Kernel de Linux.
- Instalación del sistema SANE.

3.2.1.1 Instalación del escáner

Esta fase consiste básicamente en la instalación del escáner en el núcleo del sistema operativo Linux.

Para realizar esta etapa se utilizó el escáner de la marca SCANPORT modelo SCANPAQ 4200, que incluye una interfaz paralela. Este dispositivo emplea un conjunto de comandos SCSI, y contiene un adaptador de bus paralelo a SCSI construido a partir del chip OnSpec 90c26.

Existen algunas empresas que han fabricado escáners con base en una interfaz SCSI, sin embargo, en lugar de reconstruir complemente la interfaz de estos dispositivos, una mejor alternativa ha consistido en utilizar una interfaz paralela a SCSI. En este sentido únicamente es necesario proporcionar un controlador adicional para este dispositivo, equivalente al controlador de una tarjeta SCSI. Esta estrategia resulta mucho más barata que un rediseño completo del hardware y software para el escáner.

Es importante mencionar que en general un escáner SCSI constituye la mejor opción, debido a que su desempeño es mayor en comparación con escáners de puerto paralelo, y además existe una mayor variedad de opciones que son compatibles con Linux. La instalación de los dispositivos SCSI con un adaptador de bus paralelo bajo el sistema operativo Linux, constituye un proceso más elaborado y que requiere la ejecución de fases adicionales en comparación con la instalación de un escáner de puerto paralelo o SCSI. Como argumento adicional se puede mencionar que existen múltiples escáners que cuentan con esta arquitectura híbrida y que son ampliamente utilizados en la actualidad, los cuales se pueden consultar en el sitio [9].

En general se puede afirmar que si existe un controlador o pseudo-controlador para el adaptador de puerto paralelo a SCSI, y además existe un controlador o backend de SANE para la versión SCSI de este escáner, entonces es posible usar alguna de las aplicaciones o frontends de SANE para controlar este dispositivo bajo Linux.

En particular la operación del escáner SCANPAQ 4200 sobre una plataforma Linux, debe funcionar de acuerdo a la siguiente configuración:

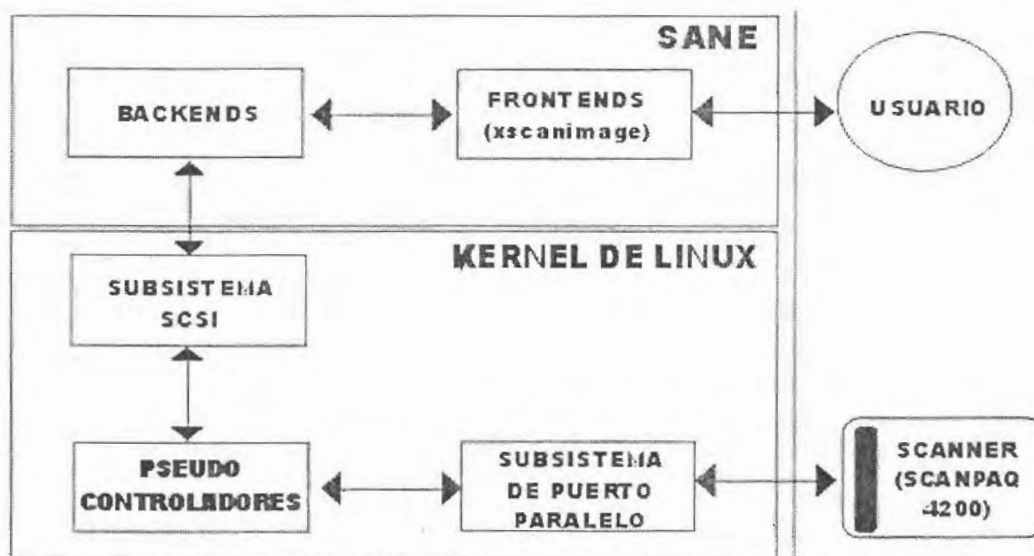


Figura 3.1: Configuración para un sistema encargado de la digitalización de imágenes utilizando SANE y un escáner SCSI de puerto paralelo.

En la figura 3.1 se describe la interacción de los diferentes módulos que intervienen en la operación de un escáner SCSI de puerto paralelo bajo Linux. Algunos de estos módulos se ejecutan sobre el Kernel, como los componentes de SANE, sin embargo su función exacta se describirá con mayor detalle en la siguiente sección. Los módulos restantes se ejecutan a nivel del Kernel y son necesarios para el control de bajo nivel del escáner. En particular los subsistemas SCSI y de puerto paralelo, son incluidos en las distintas distribuciones de Linux en la forma de módulos que se pueden integrar de manera dinámica al Kernel, en este sentido no será necesario instalar estos subsistemas en la mayoría de los casos. Para la implementación del subsistema de puerto paralelo se utilizan los módulos *parport* y *parport_pc*, y para el control del subsistema de puerto SCSI se utiliza el módulo *sg (scsi generic)*. En resumen la tarea principal consiste básicamente en realizar la instalación de los módulos que corresponden a los pseudo-controladores.

3.2.1.1.1 Pseudo-controladores

El lenguaje SCSI se puede conceptualizar como la unión de un conjunto de comandos y un conjunto de buses de hardware encargados de transportar los datos asociados con los comandos. Estos buses se pueden clasificar de la siguiente forma [12]:

- Buses que son utilizados exclusivamente para el protocolo SCSI.
- Buses que son compartidos con otros protocolos, por ejemplo con el protocolo USB.
- Buses que no se ajustan con los diferentes estándares definidos para el protocolo SCSI.

Particularmente en la última categoría se presentan algunos ejemplos interesantes, como los escáners SCSI o los dispositivos ZIP. La característica particular de estos dispositivos consiste en que emplean una suite de comandos SCSI, sin embargo no se comunican a través de un puerto SCSI sino a través de un puerto paralelo.

Para el control de tales dispositivos es necesario emplear pseudo-controladores. Estos módulos se comunican con otros subsistemas de Linux, como el subsistema SCSI o el subsistema de puerto paralelo, con el propósito de enviar los comandos SCSI hacia los dispositivos físicos controlados por estos subsistemas.

3.2.1.1.2 Sistema ppSCSI

El sistema *ppSCSI* es una arquitectura diseñada para dispositivos que utilizan comandos SCSI a través de un puerto paralelo. El módulo ppSCSI proporciona la funcionalidad necesaria para facilitar la implementación de los diferentes protocolos de comunicación utilizados por estos dispositivos.

Cada módulo desarrollado con base en la arquitectura ppSCSI, se registra con el módulo principal conocido como ppSCSI.o. Este proceso consiste en la transferencia de una lista de puntos de entrada que son comunes a todos los controladores que implementan estos protocolos. En la figura 3.2 se muestra la arquitectura del sistema ppSCSI.

Para el funcionamiento del escáner es necesario instalar un módulo encargado la traducción de los datos SCSI originados en el escáner (a través del puerto paralelo), y dirigidos hacia el subsistema SCSI de Linux, o viceversa. Para realizar este proceso el nuevo módulo debe traducir y entender el protocolo utilizado por el adaptador SCSI integrado en el escáner. Una vez que el módulo requerido se encuentra instalado, es posible

utilizar el backend más apropiado de SANE para establecer el mecanismo de comunicación con el escáner y posibilitar la digitalización de imágenes. Afortunadamente existen un conjunto de módulos ppSCSI que implementan los protocolos utilizados por diferentes adaptadores SCSI, los cuales se encuentran integrados en algunos modelos de escáners.

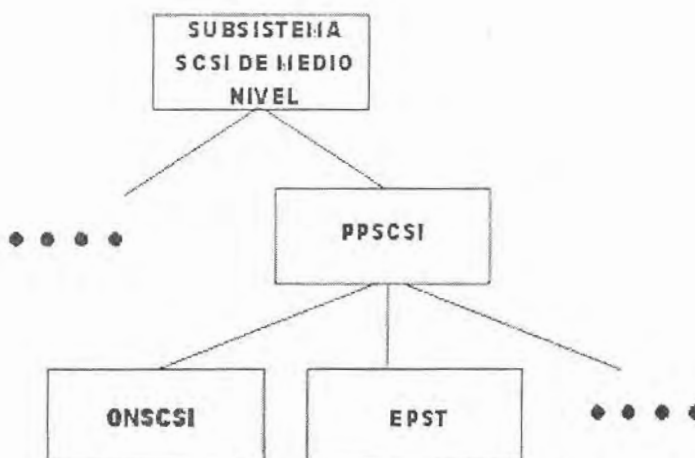


Figura 3.2: Arquitectura del sistema ppSCSI [12]

Existen algunos fabricantes que han construido adaptadores que pueden ser usados para conectar un bus SCSI externo hacia el puerto paralelo de una computadora. A continuación se detallan los adaptadores que tienen soporte en Linux o se encuentran en su etapa de desarrollo [11].

- **Iomega:** la empresa Iomega tiene dos adaptadores SCSI para puerto paralelo. Uno de ellos es ampliamente conocido como PPA-3 o el VPI0, que es usado en algunos modelos viejos de unidades ZIP de puerto paralelo. El otro adaptador es conocido como Iomega MatchMaker o VPI2, y es usado en las unidades ZIP-Plus y ZIP-250. Estos dispositivos son soportados por los controladores imm y ppa.
- **Adaptec:** Adaptec utiliza dos interfaces de bus paralelo a SCSI, que fueron desarrollados originalmente por Trantor. Ha saber, los dispositivos mini-SCSI plus APA-348 y el APA-358. El APA-348 esta basado en el NCR-5380 mientras que el APA-358 esta basado en el NCR-53c400. Estos adaptadores también han sido distribuidos bajo la marca NEC. Adicionalmente Adaptec también distribuye el APA-1350, que se basa en el adaptador Shuttle EPST. El grupo de controladores ppSCSI, ofrece el soporte para estos adaptadores.
- **Shuttle Technology:** la empresa Shuttle Technology contruyó un adaptador paralelo a SCSI, que es distribuido por muchas compañías, incluyendo a H45 Technology. Este adaptador es incluido en las versiones de puerto paralelo de algunos discos SCSI y unidades de CD-ROM. También es usado en muchos modelos de escáners

de la empresa Hewlett-Packard, incluyendo el HP-5200C. El grupo de controladores ppSCSI, incluye el soporte para este adaptador.

- **OnSpec:** el chip 90c26 de la empresa OnSpec Electronics, incluye un adaptador SCSI integral. Este adaptador es distribuido por muchas compañías incluyendo Rancho y Phase-II. Particularmente es usado también en muchos escáners de puerto paralelo, incluyendo algunos modelos de la empresa MicroTek. El grupo de controladores ppSCSI, incluye el soporte para este adaptador.
- **Shining Technology:** esta empresa distribuye actualmente el adaptador SparCSI, sin embargo también ha sido distribuido como LinkSys ParaSCSI o Winbond WBS-11A. Este adaptador combina un dispositivo KingByte y un dispositivo NCR-5380. El grupo de controladores ppSCSI, incluye el soporte para este adaptador.
- **Belkin:** el adaptador Belkin F5B001, es un dispositivo basado en un FPGA, que debe ser programado por un software antes de usarlo. Un controlador ppSCSI para este dispositivo se encuentra en desarrollo.

El grupo de controladores ppSCSI para adaptadores SCSI de puerto paralelo esta disponible en el siguiente sitio ftp:

ftp://ftp.torque.net/pub/ppSCSI-0.91.patch

Este parche esta diseñado para el Kernel de Linux versión 2.2.10, sin embargo también pueden ser aplicado a versiones más recientes. Los controladores deben ser compilados e instalados como módulos que se pueden integrar al Kernel de forma dinámica. Los módulos incluidos en esta suite son [13]:

- **ppscsi:** módulo con la librería principal para ppSCSI.
- **t348:** módulo con el protocolo para el APA-348 y T348.
- **t358:** módulo con el protocolo para el APA-358 y T358.
- **epsa2:** módulo con el protocolo para el Shuttle EPSA-2.
- **epst:** módulo con el protocolo para el Shuttle EPST y APA-1350.
- **onscsi:** módulo con el protocolo para el OnSpec 90c26.
- **sparcsi:** módulo con el protocolo para el SparCSI y ParaSCSI.

Estos módulos se pueden utilizar con una gran variedad de dispositivos incluyendo CD-ROMs, discos duros, cintas de almacenamiento y escáners.

3.2.1.2 Instalación de los módulos necesarios para la operación del escáner en el Kernel de Linux

El escáner modelo SCANPAQ 4200 incluye un adaptador OnSpec 90c26, en este sentido es necesario aplicar el parche `ppSCSI-0.91.patch` al Kernel de Linux. Este parche integra los módulos de la distribución ppSCSI a la estructura del Kernel, con lo cual es posible establecer los mecanismos de control necesarios para la operación de este escáner en el entorno del sistema operativo Linux. Es importante mencionar que el Kernel de trabajo para esta tesis corresponde a la versión 2.2.16.

El proceso completo para el soporte del escáner en el Kernel de Linux incluye las siguientes fases:

1. Aplicación del parche ppSCSI-0.91.patch (con la suite de protocolos ppSCSI) al Kernel 2.2.x de Linux.

Un inconveniente importante que surge en esta fase, se debe a que el parche `ppSCSI-0.91.patch` se encuentra diseñado para su aplicación en el Kernel 2.2.10 de Linux, de tal manera que es necesario modificar el código fuente del parche para aplicarlo con éxito en otras versiones del Kernel.

Con el propósito de facilitar la ejecución de esta tarea, se desarrolló un programa o script para el shell `bash`. El objetivo de este programa consiste en adaptar y aplicar el parche `ppSCSI-0.91.patch` a un Kernel en particular. La ejecución del script comprende las siguientes etapas.

- a) El primer paso consiste en definir al directorio `/usr/src` como el directorio actual de trabajo. En este directorio se encuentran la estructura de archivos fuente para el Kernel del sistema operativo Linux, en consecuencia cualquier modificación al Kernel debe ser realizada desde esta ubicación. Es importante mencionar que pueden existir otras versiones del Kernel instaladas bajo el mismo directorio `/usr/src`. Lógicamente si el directorio no se encuentra presente o no se tienen los permisos de escritura correspondientes, entonces el programa termina y se muestran los mensajes de error apropiados.
- b) En la siguiente fase, el programa verifica si el archivo `ppSCSI-0.91.patch` se encuentra presente en el sistema, con este propósito revisa el directorio `/usr/src`, así como el directorio de trabajo presente durante la invocación del script. Si el parche no se encuentra en la ruta indicada, entonces el programa termina.
- c) Con el directorio de trabajo definido como `/usr/src`, el programa revisa las entradas registradas en este directorio y detecta las versiones del Kernel presentes

- en el sistema Linux. Esta información se muestra en la forma de un menú de opciones, que el usuario puede utilizar para elegir el Kernel al que se le aplicará el parche. Si no existe ninguna versión del Kernel el programa termina.
- d) A continuación el programa debe verificar la presencia de la liga simbólica *linux*, la cual debe hacer referencia al Kernel seleccionado en el paso anterior, en caso de que esto no suceda debe realizar los ajustes correspondientes. Este ajuste es importante debido a que esta liga indica generalmente la versión del Kernel de Linux que se encuentra instalada y sobre la que se aplicarán los cambios necesarios.
 - e) Como paso final se crea el archivo *parche_ppSCSI.patch*, que constituye la versión adaptada para la versión del Kernel seleccionada en el paso 3. Este archivo se crea a partir del archivo *ppSCSI-0.91.patch*, cambiando las entradas etiquetadas como *linux-2.2.10* por el nombre de la versión correspondiente al Kernel seleccionado. Finalmente se aplica este nuevo parche a la estructura del Kernel apropiado.

La implementación de este script se puede consultar en el apéndice A.

2. Realizar la reconfiguración del núcleo de Linux, así como la compilación e instalación de los módulos recién integrados al Kernel.

Para realizar la configuración de los módulos presentes en el núcleo de Linux se puede utilizar el comando *make config*, que ejecuta un script de configuración situado en el directorio */usr/src/linux*, este script se ejecuta en modo texto realizando una serie de preguntas para definir la configuración del núcleo de Linux. Existen desde luego algunas alternativas más sencillas y confortables para realizar esta configuración. El comando *make menuconfig* es probablemente el más utilizado, sin embargo también es posible emplear el comando *make xconfig*, que presenta una interfaz gráfica altamente funcional. Estas dos alternativas presentan numerosas ventajas en comparación con la primera opción, la más sobresaliente de ellas consiste en la facilidad de cancelar los cambios de manera directa sin necesidad de volver a iniciar la ejecución del script.

Las opciones de configuración para el núcleo de Linux son presentadas de manera jerárquica por los comandos *make menuconfig* y *make xconfig*. Generalmente es posible establecer las opciones apropiadas eligiendo 'y' (sí) o 'n' (no). Los controladores de dispositivos presentan una opción adicional etiquetada como 'm' (módulo), esto significa que el sistema compilará estos controladores como módulos que pueden integrarse de

forma dinámica al Kernel de Linux. Para versiones del Kernel mayores a la 2.0.x, existe una opción '?', que proporciona una breve descripción de los parámetros de configuración.

El interés particular de esta fase consiste en realizar la configuración de la arquitectura ppSCSI en el Kernel de Linux. Esta tarea implica la configuración de los controladores ppSCSI, como módulos que se puede integrar de manera dinámica al núcleo de Linux, es decir, deben ser configurados bajo la opción 'm' en la estructura jerárquica presentada por los scripts de configuración. Los controladores ppSCSI se encuentran ubicados en la siguiente posición dentro de esta estructura.

- **SCSI support**
 - **SCSI-low-level-drivers**
 - **ParallelPort SCSI adapter**

Una vez concluida la fase anterior es recomendable ejecutar el comando *make dep* así como el comando *make clean*. El primero asegura que todas las dependencias de un módulo, particularmente los archivos cabecera o librerías, se encuentren en la ubicación correcta. El script *make clean* elimina los archivos objeto así como otros archivos innecesarios, producidos por una versión anterior del Kernel.

En este punto es posible realizar la compilación de los módulos dinámicos utilizando el comando *make modules*, este proceso genera ligas hacia los módulos recién compilados en el archivo `/usr/src/linux/modules`. Finalmente se debe ejecutar el comando *make modules_install*, que se encarga de instalar los módulos en el directorio `/lib/modules/x.y.z`, en donde x.y.z. corresponde a la versión del Kernel utilizada.

Es importante mencionar que en esta fase no es necesario recompilar el Kernel de Linux, debido a que todos los módulos ppSCSI son configurados para integrarse dinámicamente sobre el núcleo en ejecución, de tal manera que el Kernel no sufre ningún cambio.

3. La siguiente fase consiste en la integración de los módulos necesarios para la operación del escáner en el Kernel de Linux, incluyendo a los módulos de los subsistemas de alto nivel, al módulo *ppscsi* y al módulo con el protocolo para el adaptadores SCSI correspondiente.

Para realizar esta fase se requieren algunas utilerías que operan específicamente sobre los módulos presentes en el Kernel. Las distintas distribuciones de Linux incluyen estas utilerías por defecto, de tal manera que en la mayoría de los casos no es necesario instalarlas. Sin embargo se pueden obtener el paquete *modutils-x.y.x.tar.gz* que incluye la

versión de código fuente para estas utilerías (es recomendable obtener la versión más reciente). El proceso de instalación de este paquete es similar al de SANE, de tal manera que los pasos necesarios para su instalación no serán detallados. El paquete *modutils* incluye a las utilerías *insmod*, *rmmmod*, *ksyms*, *lsmod*, *genksys*, *modprobe* y *depmod*, las cuales son instaladas en el directorio */sbin*. Las utilerías de interés para esta fase son:

- *insmod*: inserta un módulo en el Kernel en ejecución.
- *modprobe*: verifica si un módulo se encuentra cargado en el Kernel.

Los controladores de alto nivel requeridos por un escáner con un adaptador de bus paralelo a SCSI, incluyen a los módulos del subsistema de puerto paralelo, particularmente a los módulos *parport.o* y *parport_pc.o*, así como el módulo *sg.o* correspondiente al subsistema SCSI.

Para verificar si estos módulos se encuentran integrados al Kernel en ejecución se puede utilizar el comando *modprobe <módulo>*, el cual genera la información relativa al estado del módulo. Si los módulos de alto nivel no están cargados entonces se puede utilizar el comando *insmod <módulo>*, para integrarlos al núcleo de Linux. La secuencia requerida se muestra a continuación:

- a) *insmod sg*
- b) *insmod parport*
- c) *insmod parport_pc*

A continuación se debe cargar el módulo *ppSCSI* y finalmente el módulo con el controlador para el tipo de adaptador utilizado por el escáner, que en este caso corresponde al dispositivo *OnSpec 90c26*.

- d) *insmod ppscsi*
- e) *insmod onscsi*

4. *El paso final consiste en utilizar las aplicaciones compatibles con la interfaz de SANE para el control del escáner.*

Una vez que los módulos necesarios para la ejecución del escáner han sido integrados con éxito al Kernel en ejecución, la tarea restante consiste en utilizar las herramientas compatibles con la interfaz de SANE, para controlar estos dispositivos y realizar el proceso de digitalización de imágenes bajo el sistema operativo Linux. Las herramientas más utilizadas para este propósito incluyen a *scanimage*, *xsane* y *xscanimage*. Desde luego todo esto es válido, si la aplicación SANE se encuentra instalado en el sistema (proceso que se describe en la siguiente sección).

3.2.1.3 Instalación de SANE

Para instalar el SANE API primero es necesario adquirir la última distribución de este paquete. El sistema SANE se distribuye en dos formatos, como un binario precompilado o como un paquete con el código fuente.

La versión más actualizada de la distribución precompilada de SANE esta disponible⁶ para diferentes plataformas en el sitio:

<http://www.mostang.com/sane/source.html/>

La versión más estable de la distribución que incluye el código fuente esta disponible en el sitio:

<ftp://ftp.mostang.com/pub/sane/>

Es importante mencionar en este punto, que resulta imprescindible adquirir el backend o controlador apropiado para el modelo de escáner a utilizar, sin embargo en la mayoría de la ocasiones esto no es necesario debido a que SANE es distribuido con una amplia gama de controladores.

La distribución de SANE incluye programas que operan como interfaces gráficas de usuario, que en caso de ser requeridas pueden ser compiladas por la herramienta gcc, bajo la premisa de que el paquete GIMP, o como mínimo la distribución GTK⁷ se encuentran instalados en el sistema operativo, en caso contrario será necesario adquirirlos, compilarlos e instalarlos, antes de realizar el proceso de compilación de SANE. Los paquetes GIMP y GTK están disponibles en el directorio:

<ftp://ftp.gimp.org/>

El paquete GTK incluye un grupo de programas y rutinas que facilitan el desarrollo de interfaces en sistemas UNIX. Es importante señalar que la distribución SANE se puede compilar sin las librerías GIMP/GTK, pero en este caso no se instalaran las interfaces gráficas incluidas en la distribución, eliminando con ello las ventajas que surgen de la utilización de una interfaz de usuario con elementos gráficos estructurados. En este sentido resulta altamente recomendable instalar como mínimo el paquete GTK.

El primer paso para la instalación del sistema encargado de la digitalización de imágenes bajo Linux, consiste en adquirir e instalar la última distribución de SANE.

⁶ Hasta la fecha de edición de esta tesis

⁷ Véase la sección 5.2.2.1

La versión de código fuente más estable⁸ de la distribución SANE, es la versión 1.0.4. En esta distribución los archivos fuente están divididos en dos paquetes:

- *sane-backends*
- *sane-frontends*

En general el proceso para la instalación de ambos paquetes es similar, y comprende las etapas que se describen a continuación:

1. El primer paso consiste en desempaquetar y en su caso descomprimir los paquetes de la distribución, utilizando el comando:

```
tar -xvzf sane-backends-1.0.4.tgz ó tar -xvzf sane-frontends-1.0.4.tgz
```

2. La siguiente fase inicia al ingresar al directorio raíz de la estructura de archivos fuente que se generó en el paso anterior (en este caso el directorio *sane-1.0.4*), a continuación se debe ejecutar el script *./configure*, el cual básicamente verifica si la distribución de SANE puede ser compilada en el sistema actual. Durante el proceso de ejecución de este script su salida se dirige hacia la consola del sistema y hacia un archivo llamado *config.log*, este proceso se detiene automáticamente si encuentra algún error que pudiera impedir la compilación exitosa de la distribución de SANE. Al terminar la ejecución del script se pueden revisar los archivos de ayuda incluidos en esta distribución, con el propósito de realizar una configuración más precisa en una plataforma determinada. Finalmente es importante mencionar que es posible utilizar el comando *configure -help* en el directorio raíz, para obtener un resumen de las opciones disponibles para la ejecución del script *configure*, sin embargo en la mayoría de las ocasiones las opciones por defecto, son apropiadas para realizar la fase de configuración.
3. Nuevamente en el directorio raíz se debe ejecutar el comando *./make*, el cual se encarga de realizar la compilación de los archivos fuente de la distribución SANE. Durante el proceso de compilación se presentan numerosas advertencias, sin embargo estas corresponden a situaciones que deber ser revisadas mas cuidadosamente o que representan potencialmente una inconsistencia o redundancia. Para observar el conjunto completo de advertencias se debe ejecutar el comando *./configure --enable-warnings*.

Para realizar la compilación del paquete con los frontends de SANE y para incluir la aplicación *xscanimage*, es necesario tener instalada la librería GTK o al menos la distribución GDK, antes de realizar el proceso de compilación. De no ser

⁸ Hasta la fecha de edición de esta tesis

así, será necesario adquirirlos, desempaquetarlos e instalarlos siguiendo básicamente el mismo procedimiento para la distribución SANE.

4. El siguiente paso consiste en reingresar al sistema Linux en la forma del usuario *root* utilizando el comando *su* o *su root*, de esta forma se adquieren los permisos del administrador y es posible acceder a toda la estructura de directorios. A continuación se debe ejecutar el comando *make install*, el cual se encarga básicamente de realizar la instalación de los archivos ejecutables, librerías, documentos y manuales de SANE, en la estructura de directorios del sistema operativo Linux.

Una vez que la aplicación *xscanimage* ha sido instalada, es posible integrarla en GIMP como un módulo externo, por medio del siguiente comando:

```
In -s /usr/local/bin/xscanimage /usr/local/lib/gimp/1.1/plugin-ins
```

Este hecho hace posible la invocación del programa *xscanimage* desde el menú principal del entorno de ejecución de GIMP. El programa *xscanimage* efectúa entonces el proceso de digitalización de la imagen y transfiere los datos resultantes hacia GIMP, para su procesamiento correspondiente. Esta situación es conveniente debido a que no es necesario utilizar diferentes aplicaciones para realizar la digitalización y el procesamiento de una imagen, ya que todo esto puede ser ejecutado en la mayoría de los casos, únicamente con el sistema GIMP.

5. Como paso final se debe ejecutar el comando *./find-scanner -v* presente en el directorio *tools*. Este comando produce una lista de los dispositivos SCSI instalados y reconocidos por el núcleo de Linux. Para este mismo propósito se puede utilizar el comando *scanimage -L*. Esta herramienta es útil para verificar si un escáner ha sido instalado exitosamente, en cuyo caso es posible utilizar las herramientas que permiten el control y administración de los escáners, como la aplicación *scanimage* o su versión gráfica *xscanimage*.

La instalación de la versión compilada del sistema SANE es realizada en el directorio */usr/local*, sin embargo esto puede ser reconfigurado durante la fase de compilación.

El sistema SANE no depende de las distintas plataformas sobre la que se ejecuta. En este sentido existen diversos módulos binarios de SANE diseñados para múltiples plataformas, por ejemplo, existen RPMs para las distribuciones de Linux como RedHat, SuSe, Caldera, entre otros. También se distribuyen paquetes para el sistema operativo Solaris. Estos módulos trabajan independientemente del procesador utilizado y funcionan para múltiples versiones de los sistemas operativos que los soportan.

Capítulo 4

Sistemas de impresión en Linux

El objetivo final del presente capítulo consiste en establecer y describir los mecanismos necesarios para construir una solución completa para la impresión de archivos de datos en Linux, pues esta tarea es un requerimiento frecuente y esencial de numerosas aplicaciones de propósito específico, sobre todo de aquellas que involucran algún tipo de procesamiento digital sobre imágenes.

En este capítulo se describen los principales sistemas de impresión que operan o son compatibles con el sistema operativo Linux. En este sentido durante el desarrollo de las secciones siguientes se presentarán los conceptos, propiedades y características que describen a cada uno de los sistemas de impresión más importantes que se pueden instalar y administrar bajo una plataforma Linux.

En este punto es necesario mencionar que uno de los problemas principales que presenta Linux en la actualidad, es que no tiene una compatibilidad total con los diferentes dispositivos de impresión que existen en el mercado (en comparación con el soporte que tiene Windows). Esto se debe principalmente a que todavía existen algunas empresas dedicadas a la producción de impresoras, que no ofrecen controladores para Linux ni la información necesaria para producirlos. Por supuesto como se describió en el capítulo anterior este fenómeno está cambiando favorablemente, y muchas de estas empresas ya están poniendo su atención de manera progresiva en las soluciones de impresión para Linux, motivadas principalmente por el gran crecimiento de esta sistema operativo y su rápida expansión en el mercado de los sistemas operativos. Tal situación se presenta en empresas importantes como Compaq, Hewlett Packard e IBM que ha invertido numerosos recursos con el propósito de ofrecer una amplia gama de soluciones para Linux. Con base en estos argumentos es válido decir que en futuro no muy lejano el ámbito de soluciones de impresión para Linux, se ampliará completamente, de tal manera que para contar con una solución totalmente satisfactoria no será necesario realizar un proceso de selección riguroso y detallado de los dispositivos de impresión compatibles con este sistema. Desde luego la situación actual no es tan desfavorable debido a que Linux presenta un soporte importante para los dispositivos de impresión más comunes, de esta manera únicamente es necesario realizar un proceso de selección de los dispositivos compatibles y configurar los mecanismos de impresión subyacentes.

4.1 Procedimientos generales para la configuración de un dispositivo de impresión bajo Linux

El objetivo de esta sección consiste en presentar una visión general de los pasos y las condiciones que surgen ante la necesidad de configurar un dispositivo de impresión bajo el sistema operativo Linux.

El procedimiento más simple y sencillo para configurar un dispositivo de impresión bajo Linux consiste en utilizar las herramientas que son provistas en las diferentes distribuciones de este sistema operativo, sin embargo esta opción requiere necesariamente que la propia herramienta incluya, y además ofrezca, el soporte adecuado para el controlador del dispositivo correspondiente. Cuando ambos requerimientos se satisfacen entonces la configuración del dispositivo se convierte en una tarea realmente sencilla. Si el procedimiento anterior no resulta, entonces el siguiente paso consistirá en investigar si existe en Linux algún grado de soporte para dicho dispositivo de impresión, con este propósito se pueden revisar las distintas fuentes de información al respecto, como son las listas de compatibilidad de las impresoras con Linux, grupos de noticias, listas de discusión, o los recursos en Internet que mantienen las empresas dedicadas a la manufactura de tales dispositivos. Si no existe la referencia de un controlador para Linux que trabaje con determinada impresora o si explícitamente se describe que no hay soporte para tal dispositivo, entonces puede resultar muy complicado configurar un sistema de impresión que trabaje con dicho dispositivo, a menos por supuesto que se construya el propio controlador, sin embargo esta tarea frecuentemente es sumamente complicada, sobre todo si consideramos de manera especial que las empresas manufactureras no ofrecen la información necesaria para construir tales módulos de control.

En el caso en donde se encuentran las referencias de un controlador que trabaja con el dispositivo de impresión en cuestión, entonces será necesario revisar si este controlador se encuentra instalado o no. En este sentido generalmente se debe instalar la versión más reciente del paquete Ghostscript, con el propósito de contar con los controladores de dispositivos de impresión más recientes y así reducir la probabilidad asociada con la falta de un controlador determinado. Una vez que el controlador se encuentra instalado apropiadamente, entonces es posible utilizar nuevamente las herramientas distribuidas con Linux para realizar la configuración del dispositivo correspondiente. Si esta opción falla, entonces el siguiente paso consistirá en seleccionar un sistema de impresión diferente, y si esta opción también falla entonces será necesario construir un módulo de configuración apropiado para el dispositivo de impresión.

4.2 Generalidades

Cada uno de los pasos descritos en la sección anterior se detallarán en las secciones siguientes, en la medida en que los diferentes conceptos involucrados se vayan desarrollando.

4.2.1 Impresoras soportadas en Linux.

En realidad el Kernel de Linux permite la comunicación con cualquier impresora que se encuentre conectada al puerto serial, al puerto paralelo o al puerto usb, así como con cualquier impresora conectada a la red. Desafortunadamente esto no es suficiente, pues para poder enviar los comandos de control a la impresora es necesario conocer el lenguaje o protocolo de comunicación que utiliza. Desde esta perspectiva las impresoras incompatibles con Linux corresponden principalmente a los dispositivos denominados como *Windows* o *GDI*. Reciben este apelativo debido a que todo o parte del lenguaje de control de la impresora y los detalles de diseño de los mecanismos de impresión que utilizan no están documentados o no se encuentran libremente accesibles. Generalmente el fabricante de este tipo de impresoras, distribuye controladores para el sistema operativo Windows, por tal motivo a estos dispositivos también se les conoce como *winprinters*, además el fabricante también suele proporcionar controladores para Windows NT, Windows 2000, OS/2 y otros sistemas operativos, excepto para Linux.

La mayoría de estas impresoras no trabajan con software libre, solo algunas de ellas lo hacen, pero frecuentemente con un grado de funcionalidad limitado. El desarrollo de los controladores para tales dispositivos constituye un reto difícil de lograr, sobre todo por la necesidad de aplicar técnicas de ingeniería inversa que permitan descifrar los detalles del lenguaje de control subyacente, sin embargo algunos esfuerzos en este sentido han rendido sus frutos, y en la actualidad es posible utilizar de forma limitada algunos de estos dispositivos en Linux. Para conocer con mayor precisión los detalles asociados con estos dispositivos se puede consultar la lista de compatibilidad de impresoras para el sistema operativo Linux.

Algunas otras impresoras se encuentran en un nivel intermedio, en relación con el grado de soporte que presentan en Linux, como son las impresoras NEC, las cuales implementan de forma limitada el lenguaje de impresión estándar o PCL¹, de esta manera los controladores de software que utilizan este protocolo solo pueden imprimir hasta con

¹ Printing Control Language: Lenguaje de Control de Impresora

una resolución de 300 dpi, pues los detalles necesarios para lograr una impresión de 600 dpi se encuentran ocultos por NEC.

4.2.1.1 Impresoras Postscript

En Linux la mejor opción consiste en adquirir una impresora con el soporte nativo para el lenguaje Postscript, esto se debe a que muchas de sus aplicaciones producen trabajos de impresión en código Postscript.

Postscript en forma resumida es un lenguaje de descripción de páginas. En este sentido el lenguaje Postscript cuenta con una serie de funciones que le permiten realizar todo tipo de trazos y gráficos. Se puede decir que es un lenguaje de descripción de alto nivel para objetos gráficos, los elementos básicos de este lenguaje son las líneas, coordenadas, curvas y otras funciones. A partir de la representación abstracta de los objetos presentes en un documento Postscript es posible construir una representación gráfica del mismo, y se puede enviar de manera directa a una impresora con el soporte nativo para este lenguaje, lo cual resulta ventajoso debido a que sólo se transmite una pequeña cantidad de información a la impresora, hecho que agiliza todo el proceso de impresión. La velocidad de impresión se incrementa debido a que se trasmite una cantidad menor de información utilizando el lenguaje Postscript y a que los dispositivos que lo interpretan son rápidos. Por ejemplo si se tiene un trabajo que contiene únicamente un triángulo, y este trabajo se representa como un arreglo de datos binarios entonces la información proporcionada al dispositivo de impresión será considerablemente mayor en comparación con la información necesaria para representar el mismo triángulo en lenguaje Postscript, pues únicamente se emplearían algunas funciones y coordenadas. Lógicamente las velocidades de impresión para cada tipo de trabajo se verán afectadas directamente por la cantidad de información que presentan, y en este sentido los resultados de su procesamiento favorecerán notablemente al documento Postscript.

Otra de las cualidades mas importantes que presenta el lenguaje Postscript consiste en su independencia respecto a la resolución de los dispositivos de impresión, de tal manera que la resolución resultante de un documento Postscript, corresponderá a la resolución soportada por el dispositivo de impresión subyacente y no al contenido del documento. Como argumento adicional en favor del lenguaje Postscript se puede decir que se ha convertido en un estándar en el ambiente del software libre y en la industria de la publicación, lo cual ha permitido su rápida difusión e implementación en numerosos productos de hardware y software dentro de estos ámbitos [14].

4.2.1.2 Impresoras sin soporte para Postscript

A pesar de las ventajas que ofrecen las impresoras con soporte nativo para el lenguaje Postscript, estos dispositivos son en general caros y corresponden principalmente a sistemas de impresión Laser, en este sentido para muchos usuarios resulta más conveniente adquirir una impresora sin el soporte directo para Postscript. Afortunadamente muchos de estos dispositivos también son soportados bajo Linux utilizando el paquete Ghostscript, que es un intérprete para código Postscript de libre distribución [14].

La mayoría de las distribuciones de Linux incluyen generalmente una versión atrasada de Ghostscript debido a las restricciones impuestas por su licencia, sin embargo es posible obtener versiones más recientes desde las fuentes apropiadas, localizadas generalmente en las distribuciones de Linux.

La empresa Adobe ha desarrollado un nuevo lenguaje de impresión llamado PrintGear. Este lenguaje comprime los elementos de una imagen para optimizar la salida de texto y de elementos gráficos que son utilizadas comúnmente en el ámbito empresarial. El lenguaje PrintGear hereda algunas características de Postscript pero no es compatible con este último. Afortunadamente algunas impresoras PrintGear soportan otro lenguaje como PCL (Printer Control Lenguaje) de forma nativa y por lo tanto estos dispositivos pueden operar bajo el sistema Linux.

Adobe también ofrece una implementación de Postscript llamada PressReady sin embargo está sólo opera sobre una plataforma Windows.

4.2.1.3 Compatibilidad de las impresoras con Linux

Con el propósito de establecer el grado de compatibilidad de un dispositivo de impresión con el sistema operativo Linux, se pueden revisar distintas fuentes de información como son:

- La base de datos *Printer Compatibility Database* [15], con información de las impresoras con compatibilidad para Linux, que contiene gran cantidad de información relativa a impresoras específicas, además de una extensa información de sus controladores y de sus especificaciones básicas, también ofrece un conjunto de herramientas para la construcción de scripts o filtros.

- El sitio *Ghostscript Printer Compatibility* [16], con información de compatibilidad para diferentes impresoras con Ghostscript, este sitio contiene una lista de las impresoras soportadas por Ghostscript, además de un conjunto de ligas hacia otras páginas.
- Por último los *Grupos de discusión Dejanews* [17], incluyen información con los testimonios o referencias de diferentes usuarios con relación a la configuración y operación de distintos dispositivos de impresión en el sistema operativo Linux.

4.2.2 Funcionamiento de un sistema de gestión para colas de impresión

Con el objetivo de implementar una solución de impresión exitosa, es necesario conocer con mayor detalle el funcionamiento del software para la gestión de colas de impresión. Todos los sistemas que pertenecen a esta categoría trabajan esencialmente de la misma manera, sin embargo algunos de ellos pueden presentar variaciones en el orden de los pasos o pueden excluir una o dos etapas del proceso completo de impresión.

En forma resumida se puede mencionar que un sistema de gestión de colas de impresión es un programa que acepta trabajos de impresión desde un programa o una interfaz de red, los almacena en una cola de impresión y finalmente los envía directamente hacia una impresora o hacia otro sistema de impresión para su procesamiento.

En un sistema de gestión para colas de impresión generalmente existen aplicaciones que permiten enviar o eliminar tareas de impresión, así como verificar el estado de los trabajos presentes en la cola de impresión, además permiten efectuar funciones de carácter administrativo como pausar o iniciar el proceso de impresión. Un sistema de gestión de colas de impresión es en realidad una aplicación cliente/servidor. Los programas cliente son utilizados para enviar los trabajos al sistema de impresión el cual se encarga de realizar las operaciones necesarias para su procesamiento. Con el propósito de llevar a cabo estas operaciones, el servidor puede requerir otros programas (filtros) que transforman los datos de entrada en un formato entendido por los dispositivos de impresión instalados. El servidor realiza además, múltiples funciones administrativas y de configuración.

En la figura 4.1 se muestra la secuencia general de pasos que se producen durante el procesamiento de un trabajo de impresión a través de la red, en el que interviene un sistema de gestión para colas de impresión.

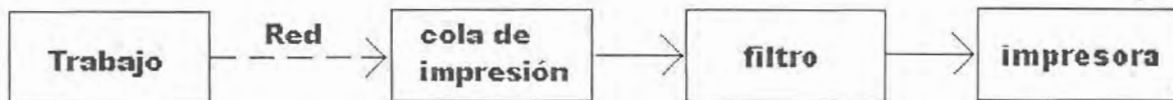


Figura 4.1. Funcionamiento de un sistema de gestión para colas de impresión

La secuencia de pasos es el siguiente [14]:

1. El usuario envía un trabajo de impresión en conjunto con las opciones seleccionadas. Los datos del trabajo corresponden generalmente a código Postscript, sin embargo también pueden corresponder a otros tipos de archivo.
2. El sistema de impresión transmite el trabajo en conjunto con las opciones seleccionadas a través de la red en dirección del sistema de impresión remoto, que se encarga de integrar el trabajo en la cola de impresión.
3. El sistema de impresión espera a que la impresora se encuentre disponible.
4. El sistema de impresión aplica al trabajo las opciones seleccionadas por el usuario, y traduce sus datos al lenguaje nativo de la impresora, el cual no corresponde generalmente al lenguaje Postscript. A esta etapa se le conoce como la fase de filtrado y representa una de las etapas más importantes y que requieren mayor atención, para el procesamiento exitoso de los trabajos de impresión,
5. Finalmente el trabajo es impreso. A partir de este instante el sistema de impresión elimina los restos del trabajo procesado. Si se produjo un error durante el procesamiento del trabajo, el sistema lo notifica al usuario de alguna manera (por ejemplo, a través de un correo electrónico).

4.2.3 Software para la gestión de colas de impresión

Hasta hace algún tiempo no existían muchas alternativas para la instalación y operación de un sistema encargado de la gestión de colas de impresión bajo una plataforma Linux, pues la única opción fue durante muchos años el sistema LPD², software que aún en la actualidad se sigue integrando en muchas distribuciones de Linux y en otros sistemas Unix.

En la actualidad existen una gran variedad de sistemas de impresión, los cuales presentan características y cualidades propias, por tal motivo resulta importante describir sus características más importantes, para elegir el más apropiado de acuerdo a la situación y a los requerimientos específicos de cada usuario o sistema de cómputo.

² Line Printing Daemon: Demonio de Impresión en Línea

A continuación se describen los sistemas de impresión con mayor desempeño y difusión en el ámbito del sistema operativo Linux.

4.2.3.1 Sistema LPRng

Algunas distribuciones de Linux incluyen al sistema LPRng, que es una implementación de software para la gestión de colas de impresión mucho más reciente que el sistema LPD, y con características más ventajosas.

El software para la gestión de colas de impresión llamado LPRng, es un sistema robusto, seguro, escalable y portable. Una de las características más sobresalientes del sistema LPRng es que proporciona un sistema de configuración altamente flexible que le permite desempeñarse óptimamente en ambientes con un solo usuario, así como en ambientes más complejos que incluyen a sistemas de impresión en red.

LPRng es más fácil de administrar para instalaciones de gran tamaño (que incluyen a las configuraciones con más de una impresora, con cualquier impresora serial o con cualquier impresora no basada en el protocolo LPD). LPRng incluye algunos ejemplos de configuración para las impresoras en red más comunes, lo cual facilita la configuración para dispositivos adicionales.

El sistema LPRng tiene tres componentes principales [18]:

- El demonio de impresión LPD y las aplicaciones cliente `lpr`, `lpq`, `lpc` y `lprm`, que son utilizadas para realizar operaciones de control, administración y monitoreo, de los trabajos de impresión.
- El filtro de impresión IFHP (`ifhp`) que es utilizado para convertir los trabajos hacia el formato nativo de una impresora en particular.
- La herramienta LPRngTool que presenta una interfaz gráfica de usuario que facilita la configuración y el monitoreo del sistema de gestión de colas de los trabajos de impresión.

4.2.3.2 Sistema LPD

LPD, es el demonio de impresión en línea desarrollado originalmente para Unix BSD. El sistema LPD se convirtió en el estándar de los sistemas Unix por muchos años y continúa siendo útil como el sistema básico para la gestión de colas de impresión. LPD es también el nombre asignado al protocolo de impresión en red, definido por el RFC 1179 [28]. Este protocolo de red no sólo es implementado por el demonio LPD, sino también por todos los servidores de impresión en red, así como por impresoras en red y por muchos de los sistemas de gestión de colas de impresión que existen en el mercado [14].

El sistema LPD esta compuesto por los siguientes módulos:

- *lpd*: es el demonio de impresión. Constituye el proceso principal que controla todo el procesamiento de los trabajos de impresión.
- *lpr*: este programa se comunica con LPD para enviar un nuevo trabajo hacia la cola de impresión.
- *lpq*: muestra el estado de los trabajos que se encuentran activos en la cola de impresión.
- *lpc*: constituye el comando de control del sistema LPD. Por medio de este programa es posible administrar la cola de impresión, debido a que permite detener, iniciar, pausar, reordenar o verificar el estado de las trabajos presentes en esta área de almacenamiento.
- *lprm*: remueve un trabajo de la cola de impresión.

En general el sistema LPD opera de la siguiente forma:

- El sistema LPD inicia su ejecución durante el proceso de arranque del sistema. A partir de este momento el demonio LPD se coloca en un estado de espera por solicitudes de conexión y se encarga de administrar las colas de impresión.
- Un usuario envía un trabajo por medio del comando *lpr*, o por medio de un programa como GPR o PDQ. LPR se comunica con el demonio LPD por medio de la red y le envía el archivo de datos en conjunto con el archivo de control que contiene las opciones del usuario.
- Cuando la impresora se encuentra disponible, el proceso LPD principal crea un proceso LPD hijo para controlar el trabajo de impresión, mientras el proceso padre vuelva a esperar nuevas solicitudes de conexión.

- El proceso LPD hijo ejecuta los filtros apropiados para este trabajo y envía los datos resultantes hacia la impresora.

LPD es un sistema viejo de tal manera que para satisfacer los requerimientos de los dispositivos de impresión modernos, es necesario utilizar elementos adicionales como son: los scripts o filtros para los trabajos de impresión, y las aplicaciones (front-ends) que muestran las opciones de configuración de cada uno de los dispositivos de impresión instalados.

Existen diversas aplicaciones que presentan una interfaz de usuario, a través de la cual muestran, y obtienen las opciones de configuración para cada trabajo y dispositivo de impresión instalado bajo el sistema LPD, además se encargan de iniciar el procesamiento de las tareas de impresión con base en la configuración seleccionada. Las aplicaciones más importantes que entran en esta clasificación son los sistemas GPR y XPDQ.

4.2.3.2.1 GPR

El sistema GPR fué desarrollado por Thomas Hubbell, y utiliza esencialmente código del sistema CUPS para filtrar los trabajos Postscript, además ofrece mayores facilidades para el control de las opciones asociadas con los trabajos de impresión. Algunas de estas opciones son implementadas directamente por el sistema GPR, mientras que otras son implementadas por las mismas impresoras o por el sistema de filtros que utiliza el software para la gestión de las colas de impresión. El sistema GPR trabaja con los sistemas LPD o LPRng.

4.2.3.2.2 PDQ

PDQ es un sistema de impresión que implementa una sintaxis propia para la configuración de los controladores que operan sobre los dispositivos de impresión. Esto incluye la posibilidad de declarar diferentes opciones de impresión, las cuales se muestran en una herramienta gráfica o de línea de comandos que los usuarios utilizan para seleccionar dichas opciones. Esta situación es altamente conveniente debido a que el usuario puede interactuar con una herramienta gráfica altamente funcional para definir opciones como la resolución, el tipo de papel o el número de hojas.

Existen en la distribución del sistema PDQ una gran cantidad de controladores o archivos de especificación de opciones para los dispositivos de impresión mas comunes, sin embargo en muchos de los casos puede ser necesario escribir algunos controladores

nuevos, con el propósito de utilizar una impresora no incluida en la distribución. El sistema PDQ también presenta algunas desventajas, la más notable de ellas constituye el hecho de que debe procesar un trabajo completo antes de enviarlo a la impresora. En consecuencia el sistema PDQ es ineficiente para trabajos de gran tamaño.

4.2.3.2.3 Configuración de dispositivos de impresión en el sistema LPD

En este apartado se describen los mecanismos necesarios para configurar un sistema de impresión en red utilizando el sistema LPD.

Con el propósito de agregar una cola de impresión al sistema LPD, se deben insertar una entrada en el archivo */etc/printcap*. El sistema de impresión revisa el archivo *printcap* cada vez que se procesa un trabajo de impresión, lo cual permite la incorporación y eliminación dinámica de los dispositivos de impresión. Cada entrada en este archivo es utilizada para describir una impresora. La impresora por defecto es normalmente el dispositivo *lp*, sin embargo esto puede ser reconfigurado utilizando la variable de ambiente *PRINTER*. Adicionalmente es posible definir de manera explícita el dispositivo de impresión a utilizar, empleando la opción *-Pnombre_impresora* en las aplicaciones que forman parte del sistema LPD.

Una entrada típica en el archivo */etc/printcap* tiene la siguiente estructura:

```
- # Ljet6L
  lp|lj|laserjet:\
    :sd=/var/spool/lpd/lj:\
    :mx#0:\
    :lp=/dev/lp0:\
```

Esta entrada define una impresora llamada *lp*, *lj* o *laserjet*, cuyo directorio para administrar los trabajos de impresión se describe como */var/spool/lpd/lj*, además se indica que no existe ningún límite en el tamaño máximo de cada trabajo. Finalmente se define que la impresora subyacente corresponde al dispositivo */dev/lp0*.

En la tabla 4.1 se muestra un lista de los parámetros más importantes y comunes que se utilizan para describir una entrada en el archivo */etc/printcap*.

Tabla 4.1 Parámetros utilizados en la descripción de una entrada en el archivo /etc/printcap. [21]

| NOMBRE | VALOR POR DEFECTO | DESCRIPCIÓN |
|--------|-------------------|--|
| af | NULL | Nombre del archivo con los datos de contabilidad |
| if | NULL | Nombre del filtro de entrada para los trabajos de impresión. |
| lo | "/dev/console" | Nombre del archivo para el registro de errores. |
| lp | "/dev/lp" | Nombre del dispositivo destino para enviar el trabajo de impresión |
| mx | 1000 | Tamaño máximo de un trabajo de impresión. El valor cero corresponde a un tamaño "ilimitado". |
| of | NULL | Nombre del filtro de salida para un trabajo de impresión |
| pl | 66 | Longitud de página (en líneas) |
| pw | 132 | Ancho de página (en caracteres) |
| px | 0 | Ancho de página en pixeles (horizontal) |
| py | 0 | Largo de página en pixeles (vertical) |
| rm | NULL | Nombre de la máquina con la impresora remota |
| rp | "lp" | Nombre de la impresora remota |
| sb | False | Etiqueta (solo una línea) |
| sc | False | Suprime múltiples copias |
| sd | "/usr/spool/lpd" | Directorio para la cola de impresión |

En muchas ocasiones es necesario realizar ajustes o conversiones de formato de los datos enviados al sistema de impresión, esto se hace con frecuencia para que los archivos puedan ser procesados apropiadamente en los dispositivos de impresión subyacentes. Esta tarea se lleva a cabo por medio de scripts o filtros, los cuales son integrados al sistema LPD utilizando la sentencia *if* en el archivo /etc/printcap. La mayoría de estos filtros realizan tareas sencillas, como la integración de caracteres especiales a la estructura de los archivos, o la aplicación y detección de formatos especiales, sin embargo existen algunos filtros mucho más robustos y completos llamados *filtros mágicos*, los cuales se encargan de identificar el formato de los archivos que se van a imprimir, para después invocar automáticamente a los programas apropiados que se encargan de convertir estos datos hacia un formato que la impresora pueda entender. Existen muchos filtros mágicos que ya han sido implementados y que pueden ser configurados con herramientas sencillas y eficientes. Los filtros mágicos más conocidos y robustos se describen a continuación [14]:

- **LPD-O-Matic:** es un filtro diseñado para usar la información generada por la base de datos *Printer Compatibility Database* [15] asociada con la operación de diversos dispositivos de impresión en Linux. Este filtro soporta una gran cantidad de controladores de distribución libre para diversas impresoras, en donde se incluyen a los controladores Ghostscript y a los controladores Uniprint. Utilizando este sistema de filtros es posible realizar la especificación de opciones, así como

el procesamiento local y remoto de los trabajos de impresión, además facilita el uso de múltiples controladores.

- **Magicfilter:** es también un filtro mágico que opera sobre los datos presentes en un trabajo de impresión, este sistema de filtros es incluido particularmente en la distribución Debian. Para configurar un dispositivo de impresión en este sistema, se puede utilizar el programa `magicfilterconfig`, el cual realiza una serie de preguntas para determinar la configuración de la impresora que habrá de integrar finalmente en el fichero `/etc/printcap`. El sistema Magicfilter incluye diversos filtros diseñados para diferentes impresoras, estos filtros se encuentran almacenados generalmente en el directorio `/etc/magicfilter`.
- **ASP Filter:** es un filtro diseñado para su operación en una gran variedad de sistemas Unix, incluyendo desde luego al sistema operativo Linux. Este filtro soporta potencialmente todos los controladores incluidos en el paquete Ghostscript. Un aspecto importante es que Apsfilter funciona con los sistemas LPD y LPRng. Con este sistema es posible reconocer e imprimir archivos con diversos formatos como ASCII, DVI, PS, PDF, GIF, TIFF, BMP, entre otros. Este paquete es incluido frecuentemente como parte de la distribución SuSE para la configuración de las impresoras.
- **RHS-Printfilters:** es un sistema de filtros construido por la empresa Red Hat. Este paquete contiene un conjunto de filtros de impresión diseñados para su utilización en la herramienta gráfica para la configuración de impresoras Printtool, también de Red Hat. Estos filtros proporcionan un mecanismo que facilita el manejo de numerosos formatos de archivo. Muchas distribuciones (incluyendo LinuxPPP y Debian) distribuyen la combinación `rhs-printfilters-printool` como la solución para la configuración y operación de dispositivos de impresión. Este sistema de filtros soporta una gran cantidad de controladores Ghostscript y Uniprint.

4.2.3.2.3.1 Impresión hacia un sistema LPD remoto

Con el objetivo de imprimir hacia otra máquina utilizando el protocolo LPD es necesario escribir una entrada en el archivo de configuración `/etc/printcap`, como se muestra a continuación:

```
# Ljet6L
lp|lj|laserjet:\
    :sd=/var/spool/lpd/lj:\
    :mx#0:\
    :rm=direccion.de.maquina.remota:\
    :lp=/dev/null:\
    :sh:
```


Esta entrada establece la configuración para la comunicación con una impresora en red, en este registro se incluye la referencia hacia un directorio local (`/var/spool/lpd/lj`), esto se hace con el propósito de mantener un área de almacenamiento, para los trabajos de impresión que se encuentran en un estado de espera, producido generalmente cuando los dispositivos de impresión se encuentra ocupados o están fuera de línea.

También es posible utilizar el comando `rpl` para enviar un trabajo directamente a la cola de impresión presente en una máquina remota, sin la necesidad de realizar la configuración del sistema LPD local, para realizar estas tareas. Este comando es útil cuando no se utilizan con frecuencia algunos dispositivos de impresión, de tal manera que es irrelevante la configuración de una entrada en el sistema LPD. El comando `lpr` utiliza el protocolo TCP/IP para enviar datos hacia los servidores LPD presentes en la red.

4.2.3.2.3.2 Impresión hacia un servidor samba o Windows

Es posible imprimir hacia una impresora controlada por un sistema Windows utilizando el programa `smbclient` incluido con la distribución `samba`. Esta distribución contiene además un script llamado `smbprint`, que se utiliza para enviar los archivos hacia un servicio de impresión SMB con base en el protocolo TCP/IP. En este caso es necesario crear una entrada en el archivo `/etc/printcap` similar a la que se muestra a continuación:

```
lp|impresora-smb-remota:\
:lp=/dev/null:\
:sd=/var/spool/lpd/lp:\
:if=/usr/local/sbin/smbprint:\
:sh:
```

Los aspectos más importantes que se pueden resaltar en la definición de la entrada anterior, se refieren particularmente a la necesidad de definir un directorio local para la cola de impresión, así como introducir una referencia hacia el script `smbprint`. Este script se encarga de enviar los trabajos de impresión directamente hacia el dispositivo samba, sin involucrar al sistema LPD.

4.2.3.2.3.3 Impresión hacia una impresora con un servidor de impresión LPD integrado

Muchas impresoras HP incluyen una tarjeta que funciona como servidor de impresión y como una interfaz de red (por ejemplo una tarjeta HP JetDirect). Utilizando esta interfaz es posible enviar directamente un trabajo hacia el dispositivo de impresión, utilizando el protocolo LPD. En general estas impresoras integran una implementación del

protocolo LPD, y proporcionan una o más colas de impresión que reciben los trabajos para ser procesados. Para una impresora HP con tales características se puede utilizar una entrada en el archivo `/etc/printcap` como la que se muestra a continuación.

```
lj-5|remote-hplj:\
:lp=/dev/null:\
:sh:\
:sd=/var/spool/lpd/lj-5:\
:rm=direccion.de.la.impresora:\
:rp=raw:
```

Los detalles más importantes de esta configuración se refieren en particular a la necesidad de definir la dirección IP asignado al dispositivo de impresión en la red, además de seleccionar su cola de impresión etiquetada como *raw* la cual acepta datos PCL o Postscript.

4.2.3.2.3.4 Configuración de una impresora utilizando la herramienta Printtool

En el ámbito de las soluciones para impresión, la herramienta de configuración mas difundida en las distintas distribuciones de Linux es la herramienta gráfica Printtool, que trabaja en combinación con el sistema de filtros RHS-Printfilters. Printtool incluye una interfaz gráfica funcional que facilita y agiliza la configuración de un dispositivo de impresión bajo el sistema operativo Linux. Generalmente el proceso de configuración consiste únicamente en la selección de las opciones más apropiadas para la operación del dispositivo de impresión correspondiente. Con estos datos Printtool genera nueva entrada en el archivo `/etc/printcap`, con la configuración para la nueva cola de impresión.

El proceso de configuración de una impresora con la herramienta Printtool involucra la ejecución de las siguientes etapas:

1. *Ejecutar la aplicación Printtool.*

La ejecución de la herramienta Printtool se puede hacer desde una terminal ejecutando el comando `Printtool`, sin embargo otra alternativa menos susceptible a errores consiste en hacerlo desde una consola gráfica de Linux. Para ejecutar esta herramienta es necesario contar con los permisos del usuario `root`, en consecuencia en muchas de las ocasiones el usuario debe reingresar al sistema en la forma del administrador, para realizar esta acción es conveniente usar el comando `su` o `su root`.

2. *Agregar una impresora.*

Una vez que la herramienta Printtool se encuentra en ejecución, se debe presionar el botón *Add a printer entry*, en la pantalla principal para agregar una nueva impresora. Al realizar esta acción se muestra una ventana con el mismo título, la cual incluye una serie de opciones que permiten seleccionar la configuración para la impresora.

3. *Seleccionar el tipo de configuración para la impresora.*

Las alternativas de elección para esta fase incluyen:

- **Local Printer:** esta opción se debe elegir cuando el dispositivo de impresión se encuentra conectado a la máquina local.
- **Remote Unix (LPD) Queue:** se utiliza cuando los trabajos de impresión son enviados a un sistema remoto de impresión LPD. Esto se presenta normalmente en configuraciones en donde la impresora se encuentra controlada por una computadora remota utilizando un sistema de impresión LPD, o cuando la impresora incluye una interfaz de red con el soporte nativo para el protocolo LPD.
- **SMB/Windows 95/NT Printer:** se utiliza con impresoras remotas, las cuales se encuentran controladas por servidores bajo Windows o SAMBA.
- **NetWare Printer (NCP):** esta opción es útil para la comunicación con dispositivos de impresión instalados en un ambiente Novell o sistemas compatibles.
- **Direct to port printer:** se aplica cuando la impresora puede recibir peticiones de impresión, a través de un puerto específico implementado en su interfaz de red.

Para ejecutar esta etapa se debe seleccionar la opción apropiada y en seguida se presiona el botón *OK*, para continuar con el proceso de configuración.

4. *Especificar los parámetros del dispositivo de impresión.*

En esta fase el usuario debe establecer los parámetros para la configuración de la impresora, los cuales dependen de la opción seleccionada en la fase anterior. Estos datos corresponden a los campos que integran a la nueva entrada en el archivo `/etc/printcap`. Algunos parámetros requeridos son:

- **Names:** incluye una lista con los nombres (separados por una barra vertical) que identifican a la cola de impresión para un dispositivo determinado. La entrada con el alias `lp` constituye la impresora por defecto.

- **Printer Device:** determina el nombre del archivo de dispositivo asociado con la impresora.
- **Spool Directory:** establece el directorio de trabajo para la cola de impresión. Este directorio debe corresponder preferentemente a un subdirectorío de /var/spool/lpd.
- **File Size Limit:** determina el tamaño máximo de un trabajo de impresión. El valor 0 indica que no existe un límite para esta restricción.
- **Remote Host:** determina el nombre de la máquina remota que ejecuta el sistema LPD, y que recibe los trabajos de impresión. Esta opción se presenta cuando se elige el tipo de configuración Remote Unix (LPD) Queue.
- **Hostname of Printer Server:** tiene el mismo significado que la opción anterior sin embargo se aplica con la configuración SMB/Windows 95/NT Printer.
- **IP number of Server:** indica la dirección IP de la máquina ejecutando el sistema SAMBA.
- **Printer Queue:** constituye una referencia de la cola de impresión en el sistema remoto LPD.
- **Printer Name:** corresponde al nombre de red del dispositivo de impresión conectado a una máquina bajo Windows.
- **Printer Device:** cuando se realiza la configuración de una impresora local, es necesario determinar la referencia de este dispositivo en el sistema Linux. Por defecto esta opción mantiene la referencia hacia el dispositivo primario detectado por Linux.
- **Input Filter:** establece el filtro que se utilizará para transformar los trabajos de impresión en un formato entendido por una impresora en particular. Para elegir el filtro más apropiado se debe oprimir el botón *select*.

5. *Seleccionar el filtro apropiado para la impresora.*

En esta fase se presenta una ventana de diálogo dividida en varias secciones. Estas secciones muestran diferentes opciones de configuración que son útiles para la operación del dispositivo de impresión subyacente.

- **Printer Type:** presenta una lista con todos los modelos de impresoras para los cuales existen controladores soportados por Printool y el sistema de filtros RHS-Printfilter.
- **Driver Description:** muestra información básica del controlador, incluyendo la descripción de los detalles asociados con su aplicación en otros dispositivos de impresión.

- **Resolution and Paper Size:** estas opciones permiten establecer la resolución, así como el tamaño de papel, que por defecto utilizará la impresora.
- **Color Depth/Uniprint Mode:** este parámetro se utiliza para establecer opciones adicionales de resolución, particularmente para impresoras en color.
- **Printing Options:** opciones de configuración adicionales para los trabajos de impresión. Estas opciones incluyen:
 - **Send EOF after job to eject page:** al activar esta opción la impresora expulsa la página una vez que el trabajo se encuentra terminado.
 - **Fix stair-stepping of text:** se debe seleccionar esta opción para eliminar el efecto que se produce cuando el texto no incluye el carácter de retorno de carro, y en consecuencia cada nueva línea es impresa a partir de la posición relativa del último carácter de la línea anterior, y no desde el inicio de la misma, produciéndose un efecto de escalera indeseable.
 - **Fast text printing (non-PS printers only):** con esta opción los archivos de texto son enviados hacia la impresora y no tienen que ser procesados por Ghostscript, lo cual agiliza el proceso de impresión.
 - **8, 4, 2, 1 pages per output page:** esta opción involucra al comando *mpage*, que permite la impresión de múltiples páginas de texto llano, en una sola página.
 - **Margins:** esta opción sirve para establecer el ancho del margen de la página, en los casos en donde se selecciona la opción anterior.
 - **Extra GS options:** define opciones adicionales para el paquete Ghostscript, disponibles para la impresora en cuestión.

6. Terminar la ejecución de printtool.

Una vez que las opciones de configuración han sido seleccionadas y aplicadas, la nueva cola de impresión se muestra en la pantalla principal de Printtool. En este punto se debe reiniciar el demonio LPD utilizando el menú con esta descripción, presente en la pantalla principal del entorno gráfico de Printtool. A partir de este momento es posible imprimir páginas de prueba para verificar el funcionamiento de la nueva cola de impresión y en su caso eliminar o editar la configuración de la nueva entrada en el sistema de impresión.

4.2.3.2.4 Protocolo LPD

En esta sección se describe de manera general el comportamiento del servidor de impresión LPD de acuerdo con las especificaciones expresadas en el RFC 1179 [28].

El RFC 1179 establece que el servidor LPD debe atender conexiones TCP/IP en el puerto 515. El archivo `/etc/services` o la base de datos de servicios TCP/IP tiene usualmente la siguiente entrada:

```
printer 515/tcp spooler # line printer spooler
```

El RFC 1179 establece explícitamente que todas las conexiones hacia el puerto 515 debe originarse desde los puertos 721-731. La razón de esta restricción se debe al concepto de puertos privilegiados o reservados en los sistemas UNIX.

4.2.3.2.4.1 Protocolo de solicitud y respuesta

A partir del establecimiento de una conexión, es posible enviar una solicitud al servidor LPD. La solicitud consiste de un octeto que indica el tipo de solicitud, seguido por el nombre de la impresora (ó cola de impresión), a continuación se coloca el conjunto de opciones para la solicitud, y finalmente el caracter LF (o caracter de nueva línea). El formato de la solicitud es el siguiente:

```
\NNNprinter[options]\n
```

La opción NNN describe el tipo de operación ejecutada por el demonio LPD. La descripción de estas opciones se muestra en la tabla 4.2

Tabla 4.2 Descripción del tipo de solicitud enviada al servidor LPD

| NNN | OPERACIÓN | PROGRAMA |
|-----|--|----------|
| 1 | Iniciar la impresión. | lpc |
| 2 | Transferir un trabajo de impresión. | lpr |
| 3 | Imprimir el estado de la cola de impresión, en el formato corto. | lpq |
| 4 | Imprimir el estado de la cola de impresión, en el formato largo. | lpq |
| 5 | Remover un trabajo de impresión. | lprm |

Una vez que la solicitud ha sido enviada, entonces se debe esperar una respuesta, que tiene en general, el formato mostrado en la tabla 4.3.

Tabla 4.3 Descripción de las respuestas retornadas por el servidor LPD

| CÓDIGO | DESCRIPCIÓN |
|---------|------------------------------------|
| \000\n | Éxito. |
| \NNN\n | Falla (NNN es el código de error). |
| texto\n | Texto o información de estado. |

4.2.3.2.4.2 Solicitudes lpq

El protocolo RFC1179 especifica que las solicitudes para la impresión del estado de la cola de impresión pueden ser enviados al servidor LPD. La solicitud lpq tiene el siguiente formato:

```
\003printer [id]*\n corto
\004printer [id]*\n largo
```

El servidor de impresión LPD retorna el estado de la cola de impresión y cierra la conexión de datos.

El RFC 1179 no establece ningún formato para el valor del estado. De esta manera, las diferentes implementaciones de este protocolo tienen la libertad de aumentar o modificar el formato correspondiente. Aún el formato para los datos de estado de la implementación BSD LPR ha sido cambiado para diferentes versiones.

Los valores id son usados para seleccionar los trabajos a mostrar. Es importante notar que debido a que no existe identificación del solicitante de la información, entonces las restricciones de información son casi imposibles.

La respuesta es una cadena ASCII que describe el estado de la cola de impresión. El fin de línea es representado con caracteres de control ASCII LF.

4.2.3.3 Sistema CUPS

CUPS es una implementación del protocolo de impresión para internet (IPP³), este protocolo es un estándar RFC 2567 [29] al estilo HTTP⁴ que constituye el sustituto para el protocolo LPD. La implementación de CUPS es distribuida bajo una licencia de tipo GPL. CUPS proporciona una capa portable de impresión para sistemas operativos basados en UNIX. CUPS soporta los protocolos Line Printer Daemon (LPD), Server Message Block (SMB), y AppSocket [14].

CUPS presenta una gran cantidad de características muy ventajosas, como son el manejo sensible de opciones, la implementación de una interfaz gráfica de usuario, así como interfaces de línea de comandos; además incluye un sistema de filtros con un gran soporte para el lenguaje Postscript. El sistema CUPS es recomendado para instalaciones grandes y con requerimientos importantes de seguridad, sin embargo también es apropiado para ambientes pequeños o para instalaciones de gran tamaño pero con usuarios confiables.

Al igual que otros sistemas, CUPS puede ser utilizado con la mayoría de los controladores existentes. Desafortunadamente resulta un poco difícil configurar un controlador arbitrario, de tal forma que la mejor estrategia consiste en encontrar los archivos PPD así como el script o filtros apropiados, para lograr que el controlador funcione adecuadamente. Existen al menos cuatro conjuntos de controladores que se pueden utilizar con CUPS [14].

- **CUPS-O-Matic:** es un sistema basado en el web capaz de generar archivos PPD para su utilización con cualquier controlador para los dispositivos de impresión cuyos detalles se registraron en la base de datos de impresión de Linux. Los archivos PPD se utilizan en conjunto con un script llamado cupsomatic. CUPS-O-Matic utiliza controladores de software libre y en la actualidad solo algunos controladores son soportados.
- **CUPS Drivers y KUPS:** el proyecto de controladores para CUPS se ha enfocado en concentrar archivos PPD que son utilizados con impresoras Postscript, así como con el script ps2gs2raw. Estos archivos PPD utilizan controladores de software libre. KUPS es un programa de configuración para CUPS.
- **Archivos PPD Postscript:** CUPS puede utilizar directamente archivos PPD para impresoras Postscript, que son proporcionados por los distribuidores de estos dispositivos. Con frecuencia estos archivos se encuentran en los controladores de

³ Internet Printing Protocol: Protocolo de Impresión de Internet

⁴ Hypertext Transfer Protocol: Protocolo de Transmisión de Hipertexto

estas impresoras para Windows, o se pueden obtener en los sitios web de los fabricantes. La empresa Adobe también distribuye archivos PPD para muchas impresoras Postscript.

- **ESP Print Pro:** la empresa Easy Software Products, Inc. distribuye el sistema CUPS en conjunto con una colección de controladores propietarios. Estos controladores operan sobre muchas de las impresoras más comunes. A pesar de que estos controladores son relativamente caros, no son muy eficientes y en realidad presentan un desempeño muy pobre, sin embargo permiten la operación de numerosas impresoras bajo el sistema operativo Linux, lo cual resulta más importante en la mayoría de las situaciones.

4.2.3.3.1 Estructura de CUPS

El sistema de impresión CUPS está diseñado alrededor de un proceso central de impresión que se encarga del procesamiento de los trabajos de impresión, ejecuta comandos de administración, proporciona información del estado de los trabajos de impresión hacia programas locales y remotos e informa al usuario de los eventos que se presentan durante el procesamiento de las tareas de impresión. CUPS incluye los módulos que se muestran en la figura 4.2.



Figura 4.2 Estructura del sistema CUPS [19]

Los módulos de la figura 4.2 realizan las siguientes funciones [19]:

- **PROCESO ORGANIZADOR:** el proceso administrador constituye una aplicación servidor que maneja solicitudes HTTP. Este proceso también actúa como un servidor web para la documentación, monitoreo de estado y administración. El proceso organizador mantiene una lista de impresoras disponibles sobre la red y procesa los trabajos de impresión utilizando los filtros y backends apropiados.

- **CUPS API:** este módulo contiene un conjunto de funciones muy convenientes para el procesamiento de los trabajos de impresión. Estas funciones se utilizan para obtener información de la impresora, iniciar tareas de impresión, acceder a los recursos de la red por medio de los protocolos HTTP o IPP, y para manipular archivos PPD.
- **COMANDOS BERKELEY Y SYSTEM V CUPS:** proporciona interfaces de línea de comandos System V y Berkeley para procesar trabajos y verificar el estado de los dispositivos de impresión.
- **FILTROS:** un programa que actúa como filtro lee los datos desde la entrada estándar o desde un archivo y produce el formato de salida deseado. Todos los filtros deben soportar un conjunto común de opciones incluyendo el nombre del dispositivo de impresión, el identificador del trabajo, el nombre del usuario, el título del trabajo, el número de copias y las opciones para el trabajo. Los datos filtrados son enviados hacia la salida estándar. Los filtros son proporcionados para muchos formatos de archivo.
- **IMAGENES CUPS:** la librería para el tratamiento de imágenes en CUPS proporciona funciones para administrar imágenes de gran tamaño, conversión de diferentes espacios de colores así como funciones para manejar la escala de las imágenes durante el proceso de impresión.
- **BACKENDS:** un programa backend es un filtro especial que envía los datos de impresión hacia un dispositivo o conexión de red. En CUPS existen backends para los puertos paralelo, serial, usb, y para los protocolos LPD, IPP, SMB y AppSocket.

Finalmente resulta importante mencionar que existe un programa complementario del sistema CUPS llamado XPP, el cual ofrece una interfaz gráfica de usuario altamente eficiente que soporta toda la funcionalidad de CUPS, incluyendo una interfaz para la administración de los trabajos durante el proceso de impresión. Adicionalmente se pueden utilizar las herramientas KUPS y QtCUPS

4.2.3.3.2 Configuración de dispositivos de impresión en el sistema CUPS

En esta sección se describen los procedimientos necesarios para la configuración de dispositivos de impresión en el sistema CUPS. La distribución de CUPS incluye una herramienta de configuración para el Web y una herramienta basada en un esquema de línea de comandos. El proceso de configuración requiere la ejecución de las siguientes tareas:

1. Ejecución del sistema CUPS.

Cuando CUPS es instalado a partir de una versión precompilada RPM, el demonio del sistema CUPS se ejecuta inmediatamente después de la instalación, y durante el proceso de arranque del sistema operativo Linux. Con el propósito de verificar si el demonio cupsd se encuentra en ejecución, se puede utilizar el siguiente comando (como el usuario root):

```
/etc/rc.d/init.d/cups status
```

Si el sistema CUPS no se encuentra activo entonces será necesario ejecutar el siguiente comando para iniciar la ejecución del demonio:

```
/etc/rc.d/init.d/cups start
```

Una vez que CUPS se encuentra en ejecución, es posible utilizar un navegador Web para ejecutar la herramienta que permite la administración y el control del sistema de impresión subyacente. La invocación de esta herramienta se hace definiendo la entrada: *http://localhost:631*, en la barra de dirección del navegador. Esta referencia le indica al navegador que debe conectarse al puerto 631 en lugar del puerto 80, que es el puerto estándar para el Web. Si CUPS fue instalado correctamente entonces el navegador presentará la pantalla principal de la herramienta CUPS correspondiente.

La ventana principal de la herramienta web presenta una serie de opciones relacionadas con las tareas de administración, control y supervisión para los dispositivos y trabajos de impresión presentes en el sistema CUPS. El interés principal en este apartado corresponde específicamente al proceso de configuración de los dispositivos de impresión.

2. Agregar una impresora.

Para agregar una impresora en el sistema CUPS es necesario seleccionar la opción con la etiqueta *Do Administration Task*, que constituye la puerta de entrada hacia los módulos que permiten la ejecución de las tareas de administración. Para ingresar a esta sección el administrador deberá contar con los permisos del usuario root, para lograrlo debe introducir el password de root en una ventana de diálogo que presenta el sistema CUPS antes de permitir el acceso al área con las opciones de configuración.

Si el administrador ingresa correctamente se mostrará una pantalla bajo el título *Admin*. Esta ventana presenta una serie de opciones que se describen a continuación:

- **La opción *classes*:** incluye diferentes módulos que se utilizan para configurar un grupo de impresoras en lo que se denomina una clase. En este sentido si el procesamiento de un trabajo es encomendado a una clase y no a una impresora específica, entonces la primer impresora disponible en la clase, recibirá el trabajo. Esto sin duda mejora la eficiencia en el procesamiento de los trabajos de impresión, pues la carga se distribuye de manera apropiada en todas las impresoras presentes en una clase. En este sentido no es necesario implementar un servidor de impresión como ocurre para el sistema LPD.
- **La opción *Jobs*:** incluye módulos para administrar los trabajo de impresión en la cola del sistema.
- **La opción *Printers*:** incluye módulos para realizar tareas de administración sobre las impresoras instaladas.

Para agregar una impresora, se debe seleccionar el módulo Add Printer. Una vez seleccionado se muestra una pantalla que solicita la siguiente información:

- **Name:** el nombre con el que se identificará a la nueva impresora.
- **Location:** la máquina en donde la impresora se encuentra localizada.
- **Description:** una descripción general de la impresora.

Esta información se utiliza únicamente como referencia y no tiene una repercusión directa en la configuración y operación del dispositivo de impresión subyacente.

3. Selección del tipo de configuración para la impresora.

Una vez que se ha introducido toda la información requerida se debe presionar el botón *Continue* para avanzar a la siguiente fase. En la nueva pantalla se muestra información para elegir el tipo de configuración apropiado para la operación de la impresora. Las opciones en esta fase son:

- **Disk File:** para imprimir directamente hacia un archivo en el disco duro.
- **AppSocket/HP JetDirect:** se utiliza para enviar trabajos de impresión hacia un dispositivo presente en la red, utilizando el protocolo AppSocket que es usado comúnmente por algunas impresoras HP y Tektronix.
- **Internet Printing Protocol:** permite imprimir hacia un puerto específico utilizando el protocolo de impresión para internet o ipp.
- **LPD/LPR Host or Printer:** es útil para enviar trabajos hacia una cola de impresión utilizando el protocolo LPD.
- **Parallel Port #1:** para enviar trabajos de impresión hacia el puerto paralelo.
- **Serial Port #1:** para enviar trabajos de impresión hacia el puerto serial 1.
- **Serial Port #2:** para enviar trabajos de impresión hacia el puerto serial 2.
- **Serial Port #3:** para enviar trabajos de impresión hacia el puerto serial 3.
- **Unknown Network Device (smb):** es posible enviar trabajos a dispositivos de impresión que se encuentran compartidos e instalados en una máquina administrada por el sistema operativo Windows. Es muy común encontrar ambientes en donde existen servidores de impresión bajo Windows, en tales casos resulta necesario tener instalado el paquete Samba para poder imprimir hacia una impresora bajo esta configuración. El paquete Samba se incluye en las distribuciones de Linux, por tal motivo en la mayoría de los casos no es necesario instalarlo.

Una vez que el tipo de dispositivo ha sido seleccionado, se requiere presionar el botón con la etiqueta *Continue*, para continuar con el proceso de configuración.

4. Determinar la información del URI⁵ para el dispositivo de impresión.

En esta nueva etapa, se solicita información del identificador uniforme de recurso (URI) asociado al tipo de dispositivo de impresión, este parámetro identifica al nuevo dispositivo en el sistema de impresión CUPS. Para definir este parámetro la herramienta de configuración muestra algunos ejemplos en su interfaz web, que incluyen a los siguientes:

⁵ Uniform Resource Identifier: Identificador Universal/Uniforme de Recursos

- *file:/path/to/filename.prn*
- *http://hostname:631/ipp/port1*
- *socket://hostname[:port]*
- *lpd://hostname/queue*
- *smb://workgroup/server/sharename*
- *parallel:/dev/file*
- *serial:/dev/file*

En cada caso, se debe mantener el patrón que identifica al tipo de dispositivo, este patrón corresponde a la palabra colocada a la izquierda del símbolo ':'. Los datos restantes se deben sustituir para ajustarse a la configuración de un dispositivo en particular.

Para usar la opción *smb://*, se debe instalar el programa *smbpool*, en la forma de un *backend* para CUPS. El programa *smbpool* se distribuye con el paquete Samba. Para llevar a cabo la instalación, se deben ejecutar los siguientes comandos:

```
ln -s 'which smbpool' /usr/lib/cups/backends/smb
```

o el comando:

```
ln -s /usr/bin/smbpool /usr/lib/cups/backends/smb
```

Es importante mencionar que la referencia */usr/bin/smbpool* debe corresponder a la ubicación de la aplicación *smbpool* en la estructura de directorios de la distribución de Linux utilizada.

Una vez que el URI para el dispositivo en cuestión ha sido definido, se debe presionar el botón *continue* para avanzar a la siguiente fase de configuración.

5. Selección del filtro apropiado para la impresora.

Dependiendo del tipo de dispositivo utilizado, la siguiente etapa de configuración presenta un conjunto de controladores diseñados para múltiples modelos de impresoras. En este caso únicamente es necesario elegir el controlador apropiado para la operación de una impresora en particular. La distribución estándar de CUPS incluye controladores para algunos modelos de impresoras, sin embargo para ampliar el rango de estos controladores se puede obtener la última versión del paquete *cups-drivers* [23].

En esta fase también es posible utilizar el sistema de filtros CUPS-O-Matic que se puede obtener del sitio *LinuxPrinting.org* [20]. La instalación de este programa es muy sencilla, únicamente se debe copiar el archivo *cupsomatic* al directorio:

```
/usr/lib/cups/filters/cupsomatic
```

Para confirmar que este programa de filtros tiene permisos de ejecución se debe aplicar el comando:

```
chmod 555 /usr/lib/cups/filters/cupsomatic
```

Para finalizar la instalación del programa *cupsomatic* se debe reiniciar el demonio *cupsd*, aplicando el comando:

```
/etc/rc.d/init.d/cups restart.
```

El sistema CUPS-O-Matic se utiliza en combinación con archivos PPD. Estos archivos describen las capacidades para un modelo de impresora determinado, información que es utilizada por el sistema *cupsomatic* para el soporte de características específicas y operaciones de filtrado inteligente. Los archivos PPD para CUPS-O-Matic se pueden obtener del sitio *LinuxPrinting.org* [20], sin embargo también se distribuyen en el paquete *cups-drivers*.

Para instalar un archivo PPD, este se debe colocar en el directorio: */usr/share/cups/model*, y debe contar con todos los permisos de lectura, para este propósito se puede utilizar el comando:

```
chmod 444 /usr/share/cups/model/archivo.ppd
```

6. Realizar tareas de administración, control y supervisión.

En la fase final se muestra un mensaje indicando que la impresora ha sido agregada exitosamente. A partir de este momento es posible realizar tareas de control, administración y supervisión sobre los dispositivos de impresión y sus tareas asociadas. El módulo con la etiqueta *Manage printer* por ejemplo, permite la impresión de una página de prueba y otras tareas administrativas. También muestra el estado de las impresoras, así como de los trabajos de impresión.

4.3 Implementación

4.3.1 Requerimientos

La configuración de la infraestructura de cómputo de interés en este trabajo de tesis, se muestra en la figura 4.3.

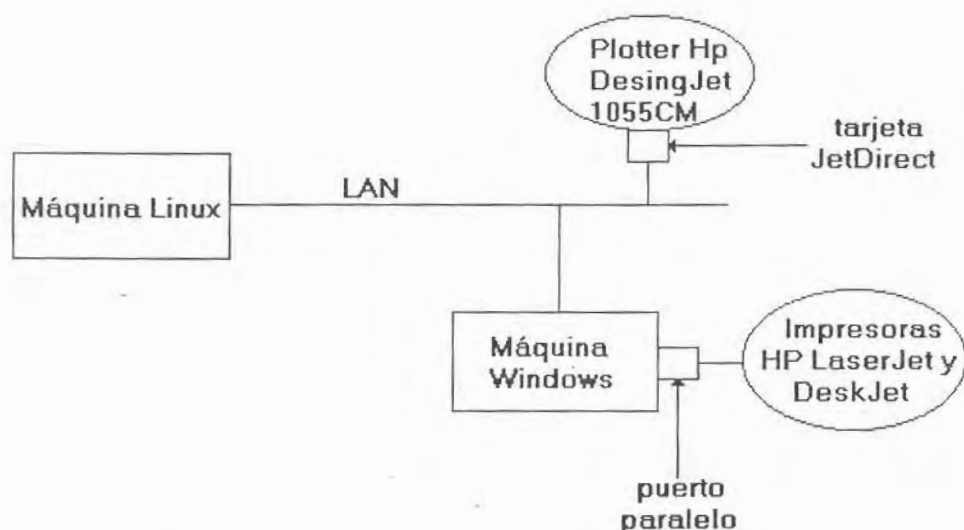


Figura 4.3. Infraestructura de cómputo que incluye dispositivos de impresión y equipos de cómputo conectados por medio de una red de área local.

Tomando como referencia esta configuración la tarea principal consiste en instalar y configurar un sistema de impresión en la máquina con la plataforma Linux, el cual se encargará de realizar el procesamiento, control y supervisión de los trabajos de impresión enviados desde esta máquina hacia los dispositivos de impresión presentes en la red. Estos dispositivos corresponden a impresoras de gran formato o plotters, en particular al plotter de la marca HP modelo Desing Jet 1055CM y a impresoras de formato pequeño como las impresoras HP LaserJet 6L y HP DeskJet 693C. También es importante mencionar que estos dispositivos de impresión se encuentran instalados utilizando diferentes configuraciones, el plotter DesingJet 1055CM incluye una tarjeta JetDirect que le permite conectarse directamente a la red y de esta forma operar de manera independiente y autónoma. Por el contrario las impresoras LaserJet 6L y HP DeskJet 693C se comunican a través del puerto paralelo hacia una máquina conectada a la LAN. Esta máquina funciona como el servidor de impresión para estos dispositivos y realiza diversas tareas de administración sobre ellos, utilizando módulos de control que operan sobre el sistema operativo Windows. En conclusión existe una infraestructura heterogénea que incluye dispositivos de impresión con características dispares, y con diferentes modos de instalación y operación, además de la presencia de sistemas operativos diferentes que

necesitan establecer mecanismos de intercomunicación para el control de las tareas y dispositivos de impresión. A pesar de todos estos requerimientos el sistema operativo Linux incluye las herramientas necesarias para instalar una solución lo suficientemente flexible y robusta para soportar el funcionamiento de un sistema impresión capaz de operar bajo tales circunstancias, inclusive ante la presencia de diferentes tipos de redes.

En la tabla 4.4 se muestran los valores de los parámetros de interés que presentan los dispositivos de la infraestructura de cómputo en estudio. Es importante mencionar que estos datos corresponden a un caso particular, sin embargo se pueden tomar como referencia para la instalación de dispositivos con una configuración similar.

Tabla 4.4 Valores de configuración para los dispositivos presentes en la infraestructura de la figura 4.3

| DISPOSITIVO | PARÁMETRO | VALOR |
|---------------------|---|---------------|
| Máquina Windows | Dirección IP | 192.168.1.101 |
| Máquina Windows | Grupo de trabajo | Wingroup |
| Máquina Windows | Nombre de la máquina | Pc-01 |
| HP DesingJet 1055CM | Dirección IP | 192.168.1.102 |
| Máquina Linux | Dirección IP | 192.168.1.103 |
| HP LaserJet 6L | Nombre del recurso compartido en el sistema operativo windows | Winlaserjet |
| HP DeskJet 693C | Nombre del recurso compartido en el sistema operativo windows | Windeskjet |

4.3.2 Procedimiento para la configuración de un dispositivo de impresión en linux

La mayoría de las distribuciones de Linux realizan la instalación y configuración de los módulos necesarios para el soporte de los servicios de impresión en el sistema operativo Linux, sin embargo en algunas ocasiones puede ser necesario realizar este proceso de forma manual, lo cual implica ejecutar los siguientes pasos:

1. El paso más obvio, que debe realizarse en primera instancia consiste en verificar la compatibilidad de una impresora con el sistema operativo Linux. Con este propósito se pueden revisar las distintas fuentes de información que existen al respecto⁶. En general si no existe alguna referencia que indique el soporte para un dispositivo determinado o si explícitamente se describe que este dispositivo no es compatible con Linux, entonces su instalación será una tarea sumamente complicada o no factible.

⁶ véase la sección 4.2.1.3

2. Instalar el soporte de impresión en el Kernel de Linux. Generalmente los dispositivos de impresión se comunican con Linux por medio del puerto paralelo, por tal motivo el Kernel incluye un controlador para este subsistema de comunicación. Este controlador está constituido por muchos módulos, que incluyen frecuentemente al módulo `parport_pc.o`, al módulo `parport.o`, y al módulo `parport_probe.o`, los cuales deben ser integrados de forma dinámica al Kernel de Linux, para el funcionamiento del puerto paralelo. Estos módulos se localizan en el directorio `/lib/modules` de la mayoría de las distribuciones de Linux. Es importante mencionar en este punto que para verificar si un módulo se encuentra instalado en el Kernel de Linux se puede utilizar el comando `modprobe < nombre_modulo >`, el cual genera mensajes que indican el estado del módulo correspondiente.
3. El paso final consiste en instalar y configurar el sistema de impresión. En esta fase se debe realizar la elección del sistema de impresión más apropiado para la configuración específica de una impresora en particular. En este ámbito existen diferentes alternativas, sin embargo las más importantes como ya se ha mencionado corresponden a los sistemas LPD, LPRng y CUPS, pues ofrecen las mejores características y constituyen sistemas robustos y altamente maduros. En el ámbito de los sistemas de filtros también es necesario realizar un proceso de elección debido a que existen diferentes alternativas como el `magicfilter`, `RHS-Printfilters`, `lpdomatic`, `cupsomatic` o el `aspfiler`, que se utilizan en combinación con otras herramientas para la configuración de los dispositivos de impresión.

4.3.3 Configuración del sistema de impresión LPD para la impresoras HP Laserjet 6L y HP DeskJet 693C

El primer paso para la instalación de este dispositivo consiste en establecer el grado de compatibilidad y soporte para estas impresoras en el sistema operativo Linux, con este propósito se deben revisar las distintas fuentes de información⁷ que existen al respecto. En particular estas impresoras utilizan el lenguaje PCL, que constituye un lenguaje de control estándar soportado por múltiples dispositivos de impresión así como por diferentes módulos de software (que realizan la conversión de datos hacia el formato PCL). El paquete Ghostscript incluye controladores para ambos modelos de impresoras, lo cual hace posible el procesamiento de trabajos de impresión remitidos desde el sistema de impresión de la computadora bajo Linux.

⁷ véase la sección 4.2.1.3

Dado que estas impresoras se encuentra conectada a una computadora remota⁸, administrada bajo el sistema operativo Windows, resulta imprescindible que el paquete SAMBA se encuentre instalado en el sistema Linux, debido a que SAMBA ofrece los mecanismos necesarios para lograr el acceso desde un sistema UNIX, hacia los servicios proporcionados por el sistema operativo Windows.

El procedimiento más sencillo y práctico para realizar la configuración de un dispositivo de impresión consiste en utilizar las herramientas que para este propósito se incluyen en las diferentes distribuciones de Linux. En particular la aplicación Printtool, es la herramienta más utilizada y difundida en los sistemas Linux, (incluyendo la distribución LinuxPPP) por tal razón constituye la herramienta seleccionada para realizar la configuración de las impresoras correspondientes. La instalación de la impresora HP modelo LaserJet 6L con base en la herramienta Printtool comprende la ejecución de los pasos presentados en la sección 4.2.3.2.3.4, por tal motivo únicamente es importante resaltar las opciones de configuración que son determinantes para la correcta operación de este dispositivo. En la tabla 4.5 se muestran los valores apropiados para las opciones de configuración de esta impresora, los cuales dependen de los datos de la tabla 4.4.

Tabla 4.5 Valores de configuración para la instalación de la impresora HP Laserjet 6L con Printtool

| OPCIONES | SELECCIÓN |
|--------------------------------|---|
| Tipo de configuración | Smb/windows 95/nt printer |
| Names | lp1 laserjet |
| Spool Directory | /var/spool/lpd/lp1 |
| Hostname of Printer Server | Winlaserjet |
| IP number of Server (optional) | 192.168.1.101 |
| Printer Name | Laserjet |
| Input Filter | Hp LaserJet 4/5/6 series non-postscript |

En la tabla 4.6 se muestran los valores apropiados para las opciones de configuración de la impresora HP modelo DeskJet 693C, utilizando la herramienta Printtool. Estos datos dependen de los valores presentados en la tabla 4.4.

Tabla 4.6 Valores de configuración para la instalación de la impresora HP DeskJet693C con Printtool

| OPCIONES | SELECCIÓN |
|--------------------------------|-------------------------------|
| Tipo de configuración | Smb/windows 95/nt printer |
| Names | lp3 deskjet |
| Spool Directory | /var/spool/lpd/lp3 |
| Hostname of Printer Server | Windeskjet |
| IP number of Server (optional) | 192.168.1.101 |
| Printer Name | Deskjet |
| Input Filter | HP DeskJet 670/680/690 series |

⁸ véase la figura 4.3

4.3.4 Configuración del sistema de impresión LPD para la impresora HP DesingJet 1055CM

Esta impresora soporta los lenguajes Adobe PostScript 3, HP-GL/2 y HP-GL. La característica más notable de esta impresora se desprende del soporte nativo que presenta para el lenguaje Postscript, en este sentido no existe prácticamente ninguna restricción importante para su operación en el entorno del sistema operativo Linux. Esta impresora tiene integrada una tarjeta JetDirect, que le permite conectarse directamente a la red, además este dispositivo puede funcionar como un servidor de impresión LPD.

La instalación de esta impresora de formato largo en el sistema LPD, se puede realizar utilizando la herramienta Printtool, aplicando básicamente dos tipos principales de configuración.

Debido a que esta impresora incluye un servidor de impresión LPD que acepta solicitudes de impresión de forma remota, la configuración más obvia consiste en elegir la opción Remote Unix (LPD) Queue, los parámetros más importantes requeridos por esta configuración se muestran en la tabla 4.7.

Tabla 4.7 Valores de configuración para la instalación (*Remote Unix LPD Queue*) de la impresora HP DesingJet 1055CM con la herramienta Printtool

| OPCIONES | SELECCIÓN |
|-----------------------|--|
| Tipo de configuración | Remote Unix (LPD) Queue |
| Names | lp2 Desingjet |
| Spool Directory | /var/spool/lpd/lp2 |
| Remote Host | 192.168.1.102 |
| Remote Queue | raw (Es el nombre por defecto para la cola de impresión del servidor LPD de la impresora) |
| Input Filter | En este caso existen dos alternativas, se puede utilizar el dispositivo Ghostscript dnj650c diseñado para una impresora DesinJet 650c, o se puede emplear la configuración para una impresora PostScript. En cualquiera de los dos casos, los filtros necesarios están presentes como parte del sistema RHS-Printfilters utilizado por printtool. |

El servidor de impresión integrado en el ploter HP 1055CM, puede recibir solicitudes de impresión, por medio del puerto 9100. En consecuencia es factible utilizar la configuración *Direct to port printer*, para la comunicación con este dispositivo desde el sistema Linux.

Tabla 4.8 Valores de configuración para la instalación (*Direct to port printer*) de la impresora HP DesingJet 1055CM con la herramienta Printtool

| OPCIONES | SELECCIÓN |
|-----------------------|--|
| Tipo de configuración | Direct to port printer |
| Names | lp2 Desingjet |
| Spool Directory | /var/spool/lpd/lp2 |
| Remote Host | 192.168.1.102 |
| Port Number | 9100 |
| Input Filter | Se puede utilizar el filtro para la impresora DesinJet 650c, o la configuración para una impresora PostScript. |

4.3.5 Implementación de un pseudo-servidor de impresión utilizando el sistema LPD

En ambientes con múltiples usuarios, en donde existen impresoras que no soportan el lenguaje Postscript de forma nativa, resulta altamente útil instalar un servidor de impresión dedicado. Este servidor debe recibir solicitudes de impresión desde clientes remotos, y puede efectuar el procesamiento ghostscript sobre los trabajos de impresión, aplicando los filtros necesarios para convertir estos datos, hacia un formato que los dispositivos de impresión puedan entender. Esta configuración presenta las siguientes ventajas:

- La configuración, administración y mantenimiento de los dispositivos de impresión se centraliza en el servidor de impresión. Este hecho agiliza y simplifica el proceso de configuración de los sistemas de impresión en los clientes remotos.
- El servidor de impresión puede controlar la carga de trabajo, asignando los tareas en espera a las impresoras libres, lo cual redundará en un mejor aprovechamiento de los recursos de impresión presentes en una infraestructura de cómputo determinada.
- Adicionalmente el servidor de impresión puede funcionar como la cola de impresión para dispositivos con un servidor LPD integrado, el cual no presenta la funcionalidad suficiente para efectuar operaciones de filtrado sobre los trabajos a procesar. Bajo esta configuración los clientes remotos pueden enviar con mayor agilidad sus trabajos de impresión, reduciendo el tiempo de espera asociado con la disponibilidad de la impresora.
- Finalmente al centralizar el acceso a los dispositivos de impresión, es posible establecer mecanismos de supervisión, control y seguridad, enfocados a mejorar el uso de tales recursos.

Una desventaja notable de la operación de un servidor de impresión centralizado surgen ante situaciones de malfuncionamiento. Bajo estas circunstancias los clientes no podrán enviar trabajos de impresión hacia los dispositivos configurados en la máquina servidor. Para reducir este efecto se puede configurar una entrada en el sistema de impresión de cada máquina cliente, que permita la comunicación con una impresora remota sin la intervención del servidor de impresión.



Figura 4.4. Infraestructura de cómputo que incluye un servidor de impresión

En el infraestructura de cómputo presentada en la figura 4.4, el servidor de impresión debe establecer los mecanismos de comunicación necesarios para enviar los trabajos de impresión hacia los dispositivos disponibles en la red. En esta configuración los recursos de impresión incluyen a una impresora (dispositivo lp1) con un servidor de impresión integrado (normalmente una tarjeta JetDirect), y con una conexión directa a la red, así como una impresora instalada en el puerto paralelo de una máquina remota administrada bajo el sistema operativo Windows (señalada en la figura con la etiqueta lp2). Bajo esta configuración, la función principal del servidor de impresión consiste en distribuir la carga de trabajo hacia los diferentes dispositivos de impresión, de tal manera que cuando un cliente envía un trabajo de impresión, el servidor debe seleccionar el dispositivo más apropiado para procesar dicho trabajo. El criterio principal de selección utilizado por el servidor corresponde en este caso a la disponibilidad de los dispositivos de impresión subyacentes, es decir, asigna los trabajos de impresión a los dispositivos libres, o que no encuentran realizando tareas de impresión.

En este caso el servidor de impresión constituye un pseudo-servidor debido a que realmente consiste en un script que se ejecuta sobre el demonio LPD y en donde se implementa particularmente las políticas de administración definidas para el servidor.

Para la operación del pseudo-servidor de impresión es necesario ejecutar las siguientes etapas:

1. Configuración de las impresoras en el sistema LPD del servidor.

El primer paso para la implementación del servidor de impresión consiste en configurar el sistema LPD del servidor de impresión. La ejecución de esta fase involucra la creación de las colas de impresión para cada una de las impresoras presentes en la infraestructura de cómputo, en este caso las impresoras lp1 y lp2. La información necesaria para llevar a cabo esta fase se describe en la sección 4.3.2.

2. Configuración de un dispositivo virtual en el sistema LPD del servidor.

La siguiente fase consiste en la configuración de una entrada o cola de impresión adicional en el sistema de impresión del servidor. Esta entrada funcionará como un dispositivo virtual que recibirá las peticiones de los clientes remotos, además constituye el módulo en donde se deben aplicar las políticas y procedimientos necesarios para distribuir la carga de trabajo hacia los dispositivos de impresión. La entrada correspondiente al dispositivo virtual debe tener la siguiente configuración en el archivo `/etc/printcap`.

```
imp_virtual:\
:sd=/var/spool/lpd/imp_virtual:\
:lp=/dev/null:\
:if=/var/spool/lpd/imp_virtual/fil_virtual:\
:mx#0:
```

El elemento más interesante e importante de la definición anterior corresponde a la declaración del filtro `fil_virtual` en el parámetro `if`. En realidad gran parte de la funcionalidad del servidor de impresión se encuentra en este script.

El script `fil_virtual` se encarga de aplicar las políticas y procedimientos de gestión a los trabajos de impresión enviados al servidor. En general, este filtro coloca los trabajos en la cola de impresión del servidor (situada en el directorio `/var/spool/lpd/imp_virtual`), hasta que cualquiera de los dispositivos de impresión configuradas en el servidor se encuentren disponibles.

El problema en este punto se centra, en encontrar un mecanismo apropiado para detectar cuando una impresora se encuentra disponible. Para satisfacer este requerimiento se desarrolló el programa `verpri` que se comunica con el demonio LPD, a partir de las especificaciones del RFC 1179⁹.

⁹ Véase la sección 4.2.3.2.4

Este programa examina la cola de impresión utilizada por el sistema LPD con el objetivo de extraer la información asociada con el estado de los dispositivos de impresión configurados en esta área.

El programa *verpri* establece una conexión con el demonio LPD, con base en las siguientes especificaciones:

- La comunicación se realiza utilizando el protocolo TCP/IP como se define en el RFC 1179.
- El mecanismo de comunicación se establece utilizando un socket.
- Una vez que la conexión ha sido establecida, el programa envía una solicitud para obtener el estado de la cola de impresión de un dispositivo determinado, en el formato corto. Esta solicitud tiene el formato: `\003printer \n`
- A partir de la información proporcionada por el demonio LPD, el programa retorna la información correspondiente, al estado del dispositivo de impresión seleccionado.
- La herramienta de desarrollo consiste en el lenguaje C.

En conclusión a partir de la información generada por el programa *verpri*, es posible detectar si una impresora se encuentra disponible, pues esta herramienta genera la información necesaria para determinarlo.

En resumen la tarea principal consiste en programar el script `fil_virtual` utilizando la secuencia de pasos que se describe a continuación.

1. Revisar el estado de las impresoras configuradas en el sistema de impresión LPD del servidor, y terminar hasta que una impresora se encuentre disponible o se exceda el límite permitido de iteraciones de verificación.
2. Verificar el estado de cada una de las dispositivos configurados en la cola de impresión. En particular se debe verificar que los dispositivos predefinidos se encuentran instalados en el sistema.
3. Si ninguna de las impresoras se encuentra instalada, entonces se termina la ejecución del script. Esto se debe a que probablemente ninguna de las impresoras se encuentra disponible para recibir trabajos de impresión, en cuyo caso resulta inútil seguir revisando su estado.



4. Si todas las impresoras se encuentran ocupadas, entonces es necesario esperar algunos segundos antes de revisar nuevamente el estado de las impresoras, con la esperanza de que en este tiempo haya terminado el procesamiento del trabajo actual.
5. Si se excede el número de iteraciones de verificación permitidas, y ninguna impresora esta disponible entonces se finaliza la ejecución del script.
6. El paso final consiste en enviar el trabajo de impresión al dispositivo que se encuentre disponible.

La implementación de este script se puede consultar en el apéndice B.

3. Configuración del sistema LPD en las máquinas cliente

La última etapa de este proceso consiste en crear una nueva entrada en el sistema de impresión LPD (archivo `/etc/printcap`) de las máquinas cliente, con la configuración para el dispositivo virtual presente en el servidor remoto de impresión. Este nuevo registro debe tener la siguiente configuración:

```
lp|imp:\
:sd=/var/spool/lpd/imp\
:mx#0:\
:rm=direccion.servidor.impresion:\
:rp=imp_virtual:\
:lp=/dev/null:
```

Bajo esta configuración los trabajos de impresión serán enviados directamente a la cola de impresión `imp_virtual` en el servidor remoto, en este lugar se realiza el proceso de filtrado de los archivos y se envían los datos directamente a los dispositivos de impresión correspondientes. Para realizar esta configuración se puede utilizar la herramienta `Printtool` con los datos de la tabla 4.9, que se basan en la configuración de la tabla 4.4.

Tabla 4.9 Valores de configuración para los clientes Linux utilizando la herramienta `Printtool`

| OPCIONES | SELECCIÓN |
|-----------------------|-------------------------|
| Tipo de configuración | Remote Unix (LPD) Queue |
| Names | lp imp |
| Spool Directory | /var/spool/lpd/imp |
| Remote Host | 192.168.1.103 |
| Remote Queue | imp_virtual |

4.3.6 Configuración del sistema de impresión CUPS para la impresora HP Laserjet 6L y HP DeskJet 693C

El grado de compatibilidad de estos dispositivos de impresión con Linux ya ha sido tratado en las secciones anteriores, en particular, lo más importante de este asunto consiste en el soporte que presenta el paquete Ghostscript para estas impresoras, que permite y facilita su instalación utilizando la herramienta CUPS.

La instalación de la impresora HP modelo LaserJet 6L con base en la herramienta web de configuración CUPS comprende la ejecución de los pasos presentados en la sección 4.2.3.3.2, por tal motivo únicamente es importante resaltar las opciones de configuración que son determinantes para la correcta operación de este dispositivo. En la tabla 4.10 se muestran los valores apropiados para las opciones de configuración de esta impresora, los cuales dependen de los datos presentados en la tabla 4.4.

Tabla 4.10 Valores de configuración para la instalación de la impresora HP Laserjet 6L con la herramienta web de la distribución CUPS

| OPCIONES | SELECCIÓN |
|-----------------------|----------------------------------|
| Name | Laserjet |
| Location | Pc-01 |
| Description | Impresora HP LaserJet 6L |
| Tipo de configuración | smb |
| URI | smb://Wingroup/Pc-01/Winlaserjet |
| Model/Driver | HP LaserJet Series CUPS v1.1 |

En la tabla 4.11 se muestran los valores apropiados para las opciones de configuración de la impresora HP modelo DeskJet 693C, utilizando la herramienta web de configuración incluida con la distribución CUPS. Estos datos dependen de los valores presentados en la tabla 4.4.

Tabla 4.11 Valores de configuración para la instalación de la impresora HP DeskJet 693C con la herramienta web de la distribución CUPS

| OPCIONES | SELECCIÓN |
|-----------------------|---------------------------------|
| Name | Deskjet |
| Location | Pc-01 |
| Description | Impresora HP DeskJet 693C |
| Tipo de configuración | Smb |
| URI | smb://Wingroup/Pc-01/Windeskjet |
| Model/Driver | HP DeskJet Series CUPS v1.1 |

4.3.7 Configuración del sistema de impresión CUPS para la impresora HP DesingJet 1055CM

Esta impresora soporta de manera nativa el lenguaje Adobe PostScript 3, en consecuencia no existe prácticamente ninguna restricción importante para su operación en el entorno del sistema operativo Linux. Debido a que esta impresora incluye un servidor de impresión LPD que acepta solicitudes de impresión de forma remota, la configuración más obvia consiste en elegir el tipo de configuración *LPD*, los parámetros más importantes requeridos por esta configuración se muestran en la tabla 4.12.

Tabla 4.12 Valores de configuración para la instalación de la impresora HP DesingJet 1055CM con la herramienta web de la distribución CUPS

| OPCIONES | SELECCIÓN |
|-----------------------|-------------------------------|
| Name | DesingJet |
| Location | 192.168.1.102 |
| Description | Impresora HP DesingJet 1055CM |
| Tipo de configuración | lpd |
| URI | lpd://192.168.1.102 /raw |
| Model/Driver | HP DesingJet 1055CM PS3 |

El servidor de impresión integrado en el ploter HP 1055CM, puede recibir solicitudes de impresión por medio del puerto 9100. Es consecuencia es factible utilizar la configuración *socket*, para la comunicación con este dispositivo desde el sistema Linux.

Tabla 4.13 Valores de configuración para la instalación (socket) de la impresora HP DesingJet 1055CM con la herramienta web de la distribución CUPS

| OPCIONES | SELECCIÓN |
|-----------------------|-------------------------------|
| Name | DesingJet |
| Location | 192.168.1.102 |
| Description | Impresora HP DesingJet 1055CM |
| Tipo de configuración | AppSocket/HP JetDirect |
| URI | socket://192.168.1.102:9100/ |
| Model/Driver | HP DesingJet 1055CM PS3 |

Capítulo 5

Módulo de GIMP para la segmentación de imágenes en color

En este capítulo se describe el proceso de implementación de una aplicación gráfica, para realizar la segmentación de imágenes en color utilizando el sistema de color modificado I1I2I3. Uno de los objetivos más importantes de esta aplicación consiste en reducir el número de etapas o herramientas que normalmente se requieren para realizar el proceso completo de segmentación, desde la fase de digitalización hasta la fase de impresión de las imágenes procesadas. Para satisfacer este requerimiento es imprescindible contar con una plataforma de desarrollo altamente modular que permita la integración dinámica de nueva funcionalidad en su núcleo operativo. Adicionalmente, con el propósito de ampliar el espectro de aplicabilidad de esta solución, es importante que la plataforma incluya herramientas y módulos adicionales, encargados fundamentalmente de la implementación de diversos algoritmos asociados con la manipulación de imágenes. El último requerimiento importante consiste en la posibilidad de integrar los módulos encargados de la digitalización e impresión de imágenes, al entorno de la plataforma de trabajo, esto permite reducir los inconvenientes que surgen ante situaciones en donde se emplean diferentes programas para llevar a cabo estas tareas de manera independiente.

Las características de GIMP son suficientes para satisfacer los requerimientos impuestos por el desarrollo de una aplicación encargada de la segmentación de imágenes en color, objetivo particular de este trabajo de tesis. En consecuencia la parte de implementación de este capítulo muestra los detalles del proceso de implementación de esta aplicación en la forma de un módulo para GIMP. La necesidad de presentar esta información se debe fundamentalmente a la ausencia de documentación clara y suficiente en este tema en particular.

5.1 Generalidades

5.1.1 Introducción A GIMP

GIMP es el programa de manipulación de imágenes de la GNU. GIMP es distribuido libremente bajo los términos de la licencia GPL. En general GIMP constituye una pieza de software que puede ser utilizada para la ejecución de una amplia gama de tareas asociadas con la creación y manipulación de imágenes. GIMP es un programa robusto que incluye una gran cantidad de herramientas, módulos y extensiones, esto le confiere de numerosas habilidades y por ende facilita su adaptabilidad a una amplia gama de aplicaciones. En este sentido GIMP puede ser utilizado como una herramienta para dibujar imágenes, y al mismo tiempo funcionar como un programa experto en el retoque profesional de fotos; su flexibilidad le permite de la misma forma operar como un sistema en línea de procesamiento por lotes, y funcionar como un convertidor de formatos de imagen, entre muchas de sus múltiples facetas o posibles aplicaciones [24].

Las características mas sobresalientes e importantes de GIMP se derivan de su capacidad para expandirse y extenderse. GIMP esta diseñado para ampliar su funcionalidad por medio de módulos y extensiones, lo cual aumenta en gran medida su espectro de aplicabilidad, reduciendo con ello los casos en donde resulta difícil su utilización.

Su avanzada interfaz basada en scripts, facilita el desarrollo y la integración de módulos externos que implementan desde las tareas más simples hasta los procedimientos de manipulación de imágenes más complejos.

La capacidad de GIMP para expandirse se deriva principalmente de su arquitectura y de los principios bajo los cuales fue diseñado, esto es, GIMP es el resultado del esfuerzo de un conjunto de programadores de diversas partes del mundo, que respaldados en la filosofía del software libre, desarrollaron una herramienta basada en una arquitectura abierta, lo suficientemente flexible para permitir su continua y rápida expansión. Para entender esto, resulta útil mencionar que en un ambiente en donde existe un gran número de programadores dispuestos a participar en el desarrollo de un sistema software, es imperativo contar con una arquitectura abierta altamente modular, que haga más eficiente el proceso de desarrollo del software, en términos de tiempo y funcionalidad. Dicho esquema de modularidad permite la eliminación, expansión y modificación de los diferentes módulos que componen al sistema, así como la inserción de módulos externos, esto facilita claramente la integración de nueva funcionalidad y en consecuencia permite la expansión y extensión de un sistema software, tal como sucede en GIMP.

Para establecer los alcances y capacidades de GIMP resulta útil mencionar sus cualidades principales.

5.1.1.1 Características principales de GIMP

A continuación se muestra una lista reducida de las características principales de GIMP [24].

- Incluye una suite completa de herramientas de dibujo, incluyendo brochas, lápices, aerógrafo, clonación, entre otros.
- Administración de memoria basada en bloques, de esta forma el tamaño de la imagen se encuentra limitado sólo por el espacio disponible en disco.
- Soporte de capas y canales.
- Contiene una base de datos procedural para invocar funciones internas de GIMP desde programas externos.
- Capacidades avanzadas para el soporte de scripts.
- Soporte para múltiples niveles de hacer/ deshacer (limitado solo por el espacio en disco)
- Incluye herramientas de transformación para rotar, escalar, trasquilar y reflejar.
- Los formatos soportados por GIMP incluyen GIF, JPEG, PNG, XPM, TIFF, TGA, MPEG, PS, PDF, PCX, BMP entre otros.
- Carga, despliega, convierte y guarda hacia muchos formatos de archivo.
- Incluye herramientas de selección como rectángulo, elipse, mano alzada, tijeras inteligentes, entre muchas otras.
- Soporta Módulos que facilitan la incorporación de nuevos formatos de archivo y nuevos filtros de efectos.

Dados los argumentos anteriores GIMP se presenta como una opción apropiada para la implementación de nuevos métodos o procedimientos asociados con el procesamiento digital de imágenes debido principalmente a su características de modularidad y a la posibilidad de utilizar la diversa gama de herramientas, módulos y extensiones que ya se encuentran implementadas e integradas a la estructura de GIMP.

5.1.1.2 Versiones de GIMP

El GIMP fue desarrollado por Peter Mattis y Spencer Kimball. A partir de 1995 GIMP ha sufrido una gran transformación, producto de una evolución constante reflejada en sus diferentes versiones [24].

La versión 0.54 de GIMP fue liberada en febrero de 1996, y tuvo un gran impacto pues se significó ciertamente como el primer programa profesional de manipulación de imágenes en Linux. Fue el primer programa de distribución libre que pudo competir con los grandes programas comerciales de manipulación de imágenes.

La versión 0.60 fue liberada en julio de 1996. Las mejoras principales con respecto a la programación fueron las nuevas cajas de herramientas, GTK (GIMP ToolKit) y GDK (GIMP Drawing Kit), que eliminaron la dependencia en Motif.

En febrero de 1997, la versión 0.99 de GIMP entro en la escena. En esta versión se realizaron numerosos cambios a GIMP y se incorporaron nuevas características. La principal mejora de esta versión fue el desarrolló de una novedosa API y el PDB¹, que hicieron posible la escritura de scripts o macros para GIMP. GTK/GDK también sufrió algunos cambios y obtuvo la denominación de GTK+. Adicionalmente, la versión 0.99 utilizó una nueva forma de administración de la memoria basada en bloques, este hecho permitió cargar imágenes de gran tamaño en GIMP .

Una de las mejoras más significativas de esta versión fue el desarrollo de una nueva API, que permitió y facilitó la escritura e inserción de extensiones y módulos para GIMP. Esta novedosa característica determinó en gran medida el futuro de GIMP, pues con ello abrió las puertas a todos los programadores interesados en aportar sus esfuerzos al desarrollo de esta aplicación. A partir de ese momento surgieron muchas y novedosas extensiones y módulos para GIMP, transformándolo en un programa mucho más versátil. Uno de los ejemplos más notables de este fenómeno fue el desarrollo del módulo para SANE, que permitió la digitalización de imágenes directamente desde GIMP. Actualmente existen más de 150 módulos registrados para GIMP.

GIMP continua evolucionando, y frecuentemente surgen nuevas versiones con mejores características y funciones adicionales. El historial evolutivo de GIMP incluye a las versiones 1.0.X, 1.1X y más recientemente las versiones 1.2.X.

¹ Procedural Data Base

5.1.2 Sistema de color modificado I1I2I3

El sistema de color I1I2I3 representa tres características importantes del color, utilizando las formulas que se describen a continuación:

$$I1 = \frac{(R + G + B)}{3} \quad (5.1)$$

$$I2 = \frac{(R - B)}{2} \quad (5.2)$$

$$I3 = \frac{(2G - R - B)}{4} \quad (5.3)$$

Este sistema de color fue presentado en el artículo *Color Information for Region Segmentation*, por los doctores Yuichi Otha, Takeo Kanade y Toshiyuki Sakai en 1980 [25].

En la tesis denominada “Segmentación de imágenes en color modificando el sistema de color I1I2I3”[25], se propone una adaptación del sistema original agregando las siguientes fórmulas:

$$I2_1 = \frac{(G - R)}{2} \quad (5.4)$$

$$I2_2 = \frac{(B - G)}{2} \quad (5.5)$$

En este punto es importante señalar que el objetivo de esta tesis no consiste en analizar el desempeño del sistema de color modificado propuesto, debido a que su conveniencia es analizada y desarrollada en el trabajo de tesis [25]. En este sentido el propósito de este trabajo consiste particularmente, en desarrollar una aplicación gráfica que facilite la ejecución del proceso completo que implica la segmentación de imágenes, utilizando el sistema de color presentado en esta sección.

5.2 Implementación

El desarrollo de la herramienta gráfica propuesta, para la segmentación de imágenes en color utilizando el sistema de color modificado III2I3, se puede dividir en dos fases principales:

1. La implementación de los mecanismos necesarios para cumplir con las especificaciones impuestas por GIMP para la programación de módulos externos.
2. La implementación de la interfaz gráfica.

La primera fase involucra la definición de los mecanismos necesarios para la comunicación del módulo con el entorno operacional de GIMP, esto se hace utilizando la especificación de la interfaz correspondiente. Esta interfaz incluye diversos procedimientos y definiciones que son útiles particularmente, para la instalación del módulo y para la manipulación de las imágenes administradas por GIMP.

La segunda fase consiste en el desarrollo de la interfaz gráfica, la cual debe incluir los elementos gráficos apropiados para facilitar la selección y ajuste de los parámetros que son necesarios para realizar el proceso de segmentación.

5.2.1 Fundamentos para programar un módulo para GIMP

Existen un conjunto de características esenciales que son comunes a cualquier módulo escrito para GIMP, sin importar el propósito para el cual fue desarrollado o la función específica que realiza. Estas características se encuentran íntimamente relacionadas con los procedimientos y la estructura, que debe cumplir un programa para ajustarse a los requerimientos establecidos en la interfaz definida por GIMP, para la programación de módulos. En forma general para cumplir estos requerimientos el módulo debe registrarse a si mismo en la *Base de Datos de Procedimientos de GIMP (PDB)*, y debe incluir una función que pueda ser ejecutada en el momento en que GIMP invoque al módulo, o procedimiento previamente registrado. En las secciones siguientes se describe la secuencia de pasos en el cual esta funciones son invocadas, y los requerimientos estructurales mínimos que debe cumplir un módulo [26].

5.2.1.1 Función principal

Todo programa escrito en lenguaje C tiene una función *main()*, que constituye la función principal del programa y por consiguiente es el primer procedimiento en ejecutarse. La función *main()* de un módulo escrito para GIMP tiene la estructura siguiente:

MAIN()

En este caso *MAIN()* es una macro, definida en el archivo cabecera *libgimp/gimp.h*, esta macro se encarga de invocar a la función *gimp-main*, la cual se encarga a su vez de realizar diversas tareas de verificación e inicialización. En esta sentido la función *gimp-main* trata de asegurarse de que el módulo sea invocado desde la aplicación GIMP, además configura todos los aspectos asociados con el manejo de señales entre procesos, así como la comunicación entre GIMP y el módulo, entre otras tareas.

Durante su etapa de inicialización GIMP invoca a los módulos que se encuentran en los directorios predefinidos para su instalación. En ese momento cada uno de estos módulos debe registrarse a si mismo en la base datos procedural de GIMP, que constituye el recipiente en donde se realiza el registro de todos los procedimientos disponibles para GIMP, durante su fase de ejecución.

El mecanismo por medio del cual GIMP invoca a un módulo (o procedimiento) se basa en gran medida en la definición de la variable global *PLUG_IN_INFO*, que constituye una variable del tipo estructura *GimpPlugInInfo* que incluye la referencia de los procedimientos que serán invocados por GIMP para la operación del módulo, como se muestra en la definición siguiente.

```

/* esta estructura se encuentra definida en el archivo cabecera gimp.h*/
typedef void (* GimpInitProc) (void);
typedef void (* GimpQuitProc) (void);
typedef void (* GimpQueryProc) (void);
typedef void (* GimpRunProc) (gchar *name,
                             gint nparams,
                             GimpParam *param,
                             gint *nreturn_vals,
                             GimpParam **return_vals);

struct _GimpPlugInInfo
{
/* esta función es invocada al iniciar la ejecución o carga de la aplicación GIMP*/
  GimpInitProc init_proc;

/* esta función es invocada al terminar la ejecución de GIMP */
  GimpQuitProc quit_proc;

```

```
/* La siguiente función es muy importante debido a que con ella se definen en gran
medida las características y capacidades asociadas al módulo por registrar, en este
sentido se puede decir que constituye un elemento primordial durante el proceso de
instalación de un procedimiento en la PDB */
```

```
GimpQueryProc query_proc;
```

```
/* El procedimiento siguiente será invocado por GIMP para la ejecución del módulo*/
```

```
GimpRunProc run_proc;
```

```
};
```

En el esquema anterior se observa que la estructura *GimpPlugInInfo* contiene apuntadores que referencian a cuatro funciones que constituyen el medio o la interfaz principal de comunicación entre GIMP y el módulo. Las funciones *init_Proc* y *quit_proc* se ejecutan, en el mismo orden al iniciar y terminar la ejecución de GIMP, estas funciones no se emplean frecuentemente y por tal motivo se les asigna el valor NULL. La función *query_proc* se utiliza en el proceso de registro del módulo en el PDB, pues en ella se definen las características que describen e identifican al procedimiento que será invocado por GIMP. Finalmente La función *run_proc* constituye el procedimiento a ejecutar durante el llamado de GIMP para el procesamiento de un módulo previamente registrado.

Con base en la explicación anterior se alcanza un mejor entendimiento del proceso de interacción (registro - ejecución) de un modulo con el ambiente de GIMP, sin embargo resulta útil presentar una serie de ejemplos que muestren en los hechos la forma en que se realiza este proceso, que recae fundamentalmente en la definición de la variable *PLUG_IN_INFO*, misma que se muestra en el esquema siguiente:

```
/* Declaración de los prototipos de las funciones que incluirá la estructura
GimpPlugInInfo. Los nombres de la funciones son arbitrarios, sin embargo es
recomendable utilizar identificadores que sean representativos del propósito
específico que cumple la función en la definición de la estructura y en el modulo en
general. */
```

```
/* prototipo de la función asociada con la definición del tipo Gimp_Query_proc */
```

```
static void query (void);
```

```
/* prototipo de la función asociada con la definición del tipo Gimp_Run_proc, (los
parámetros que recibe esta función se detallarán en los tópicos siguientes)*/
```

```
static void run (gchar *name,
                gint nparams,
                GimpParam *param,
                Gint *nreturn_vals,
                GimpParam **return_vals);
```


Como se puede observar la función *gimp_install_procedure* requiere de un conjunto de argumentos de diversos tipos. Estos parámetros determinan en gran medida las características y capacidades del módulo que se instalará en el ambiente de ejecución de GIMP, por tal motivo resulta importante describir con mayor precisión el propósito específico de cada uno de estos argumentos, lo cual se puede observar en la tabla 5.1.

Tabla 5.1: Definición de los argumentos para la función *gimp_install_procedure*

| ARGUMENTOS | DESCRIPCIÓN |
|------------|---|
| name | Corresponde a una cadena de caracteres en la que se describe el nombre del módulo. Esta cadena identifica también al nuevo módulo en el ambiente de ejecución de GIMP. |
| blurb | Corresponde a una cadena de caracteres, en la que se realiza una pequeña descripción de la funcionalidad del módulo, así como también se suelen incluir algunos detalles asociados a sus características más generales. |
| help | Es una cadena de caracteres que como su nombre lo indica suele incluir un descripción de los procedimientos asociados al uso apropiado del módulo correspondiente. |
| copyright | Incluye una descripción relacionada con los términos de propiedad intelectual del módulo, así como algunos otros detalles, como la versión o la fecha de creación del módulo. |
| menu_path | <p>Es una cadena de caracteres que describe la ruta completa del lugar en el cual el procedimiento debe ser instalado dentro del ambiente de GIMP. Esta cadena inicia con uno de los tokens siguientes <ToolBox>, <Image>, <Load>, <Save>, enseguida se concatena la ruta hacia la posición del módulo. En el caso del módulo "Segmentar" que fue desarrollado como parte de esta tesis incluye la siguiente descripción para el parámetro menu_path</p> <p>"<Image>/Filters/Colors/segmentar..."</p> <p>Los datos definidos en este parámetro, le permiten a GIMP determinar además de la posición, el tipo de módulo que se instalará, lo cual a su vez establece los parámetros necesarios para cada uno de los tipos correspondientes.</p> <p>Es posible instalar procedimientos sin definir este parámetro, lo cual imposibilita una interacción directa entre el usuario y el módulo instalado, es decir, solamente estos procedimientos pueden ser invocados por medio de otros módulos internos. Este tipo de procedimientos naturalmente no se presentan con frecuencia en la definición de la mayoría de los módulos presentes en GIMP.</p> |

Tabla 5.1: Definición de los argumentos para la función `gimp_install_procedure` (continúa)

| ARGUMENTOS | DESCRIPCIÓN |
|----------------------------------|---|
| <code>image_types</code> | Este parámetro es muy importante ya que consiste en una lista de tokens o constantes que determinan el tipo de imágenes que puede aceptar el módulo, los formatos de interés en esta tesis son el RGB y el tipo GRAY que corresponde a un formato en escala de grises, sin embargo también se incluyen los tipos: RGBA, RGB*, GRAYA, GRAY*, INDEXED, INDEXEDA, e INDEXED*. El token RGB* incluye a los tipos RGB y RGBA, una analogía se presenta con el token GRAY* e INDEXED*. Se pueden utilizar espacios, tabuladores o comas para separar los elementos en la lista. Únicamente los tipos de imágenes definidos en la lista, pueden ser aceptados por el módulo, es decir, el módulo no puede ejecutarse utilizando imágenes de un tipo no incluido en este parámetro.. |
| <code>nparams</code> | Indica el número de parámetros que usa el procedimiento. |
| <code>nreturn_vals</code> | Indica el número de parámetros que retorna el procedimiento. |
| <code>params, return_vals</code> | Determinan el tipo de los parámetros que usa el procedimiento, así como de los valores retornados. |

La mayoría de los argumentos descritos en la tabla 5.1 son altamente descriptivos y resulta sencillo intuir su función, sin embargo los parámetros `params` y `return_vals` merecen una explicación adicional.

5.2.1.2.1 El argumento `params`

El argumento `params` es en realidad un apuntador o puntero que referencia a un arreglo de parámetros (variables) del tipo `GimpParamDef` (definido en `gimp.h`), de esta forma la variable `nparams` determina el número de elementos o parámetros en este arreglo. El nuevo tipo `GimpParamDef`, es una estructura cuya definición se presenta a continuación.

```
struct _GimpParamDef
{
    /* Determina el tipo asociado al argumento correspondiente, el cual es válido en el
     * ambiente de GIMP, los valores asociados con este campo se enumeran en el
     * archivo gimpenums.h */
    GimpPDBArgType type;
    gchar      *name;           // identificador del parámetro
    gchar      *description;    // descripción del parámetro
};
```

En este punto es necesario presentar un ejemplo que describa con mayor precisión el proceso de declaración del argumento `params`, lo cual se muestra a continuación.

```
static GparamDef params[] =
{
  { GIMP_PDB_INT32, "run_mode", "Interactivo, no-interactivo" },
  { GIMP_PDB_IMAGE, "image_id", "(sin uso)" },
  { GIMP_PDB_DRAWABLE, "drawable_id", "imagen sobre la cual se trabaja" },
  { GIMP_PDB_COLOR, "fgcolor", "color para dibujar" }
}
```

Ejemplo 5.1: declaración del argumento *params*

El ejemplo 5.1 describe la declaración típica del parámetro *params*, con esta definición se establece el tipo de datos que recibirá el módulo una vez que ha sido registrado en el PDB, esto depende de los datos proporcionados en el parámetro *menu_path*, es decir, como se había mencionado con anterioridad la posición del módulo indica a GIMP el tipo de parámetros que requiere. Es completamente permitido que el módulo ignore algunos parámetros dependiendo de su función específica, sin embargo es necesario que no omita la declaración de ninguno de ellos.

En la tabla 5.2 se muestran los parámetros requeridos por el módulo dependiendo de su localización tentativa en el ambiente gráfico de la aplicación GIMP.

Tabla 5.2: Argumentos requeridos de acuerdo a la posición del módulo en el menú de GIMP.

| LOCALIZACIÓN | # | TIPO DE ARGUMENTO | DESCRIPCIÓN |
|--------------|---|-------------------|---|
| <Toolbox> | 0 | INT32 | Modo de ejecución |
| <Image> | 0 | INT32 | Modo de ejecución |
| | 1 | IMAGE | Identificador de la imagen actual |
| | 2 | DRAWABLE | Identifica la zona activa en la imagen actual |
| <Load> | 0 | INT32 | Modo de ejecución |
| | 1 | STRING | N/A |
| | 2 | STRING | N/A |
| <Save> | 0 | INT32 | Modo de ejecución |
| | 1 | IMAGE | Identificador de la imagen actual |
| | 2 | DRAWABLE | Identificador de la zona activa en la imagen actual |
| | 3 | STRING | N/A |
| | 4 | STRING | N/A |

La definición de los argumentos presentados en la tabla 5.2 dentro de la declaración del arreglo referenciado por *params*, debe seguir el orden expuesto, es decir, la inserción de estos parámetros en el arreglo debe realizarse tomando los tokens de arriba hacia abajo, dependiendo de la configuración seleccionada. Es posible incluir parámetros adicionales,

sin embargo estos deben colocarse al final del arreglo en seguida de los parámetros obligatorios.

El siguiente paso consiste en describir la función de los parámetros introducidos en la tabla 5.2. En este punto es importante aclarar que solo se presentará la descripción de aquellos que son importantes para el desarrollo de este trabajo de tesis, la justificación se realizará en los tópicos siguientes.

5.2.1.2.1.1 El tipo INT32

Este parámetro indica al módulo el modo en el cual debe ejecutarse. En GIMP existen dos tipos de ejecución para un módulo: la interactiva y la no-interactiva.

En una ejecución interactiva el módulo debe enlazarse con el ambiente gráfico de GIMP, es decir, si el módulo necesita obtener valores, debe presentar elementos gráficos que el usuario pueda manipular directamente desde la aplicación GIMP, además los resultados del procesamiento realizado por el módulo deben reflejarse de igual manera en los elementos gráficos subyacentes, que generalmente corresponden a las imágenes.

En una ejecución no interactiva el módulo no necesita enlazarse con el ambiente gráfico de GIMP y únicamente recibe y envía los datos necesarios por medio de su interacción con otro módulo interno. Los datos que puede requerir un módulo corresponden a los parámetros adicionales que se pueden declarar en la definición del puntero *params*.

Desde el punto de vista del origen de la llamada para la ejecución del módulo se puede decir en general, que éste se debe ejecutar en forma interactiva cuando es invocado por el usuario a través de su referencia directa en el ambiente gráfico de GIMP, y que se debe ejecutar en forma no-interactiva cuando la petición de ejecución proviene de un módulo interno.

5.2.1.2.1.2 El tipo IMAGE

Constituye un identificador de la imagen actual de trabajo, es decir, proporciona una referencia de la imagen que se encontraba activa en el ambiente de ejecución de GIMP en el momento en que el módulo fue invocado, lo cual es válido, naturalmente, en el caso de una ejecución interactiva. En el caso de un procesamiento no-interactivo, el módulo

interno que realiza la petición de ejecución del módulo es el encargado de proporcionar los parámetros requeridos por este último, incluyendo la imagen de trabajo.

El procesamiento realizado por el módulo se aplica generalmente a la imagen referenciada por este parámetro, de tal manera que es en esta imagen a donde se aplican los efectos, realizados en la aplicación del módulo.

5.2.1.2.1.3 El tipo DRAWABLE

Este parámetro constituye también un identificador, pero en este caso a la zona activa de la imagen actual de trabajo. En el caso más general esta zona corresponde a la imagen completa, sin embargo en muchos de los casos solo hace referencia a un fragmento de la imagen original.

Frecuentemente todo el procesamiento realizado por el módulo se aplica realmente a esta zona de trabajo, y en consecuencia los resultados sólo se observan en la zona activa de la imagen de trabajo, permaneciendo intacta la zona no cubierta por los márgenes de esta zona activa.

5.2.1.2.2 El parámetro *return_vals*

El parámetro *return_vals*, constituye también un apuntador a un arreglo de variables del tipo *GimpParamDef*. En este caso GIMP no establece ninguna restricción asociada con la declaración de este parámetro, sin embargo el módulo debe retornar al menos un valor, que corresponde al código de estado de la ejecución del módulo. Éste código le indica a GIMP la situación final del módulo en el momento de terminar su ejecución. Es muy importante indicar que este valor no debe ser tomado en cuenta en la definición de los valores para los argumentos *nreturn_vals* y *return_vals*.

El proceso de declaración del parámetro *return_vals*, es similar al expuesto en el ejemplo 5.1.

5.2.1.3 El procedimiento *run*

Esta función es invocada por GIMP para la ejecución del procedimiento o módulo previamente registrado en el PDB (véase el procedimiento *Query*). En este sentido es válido afirmar que esta función representa la puerta de entrada para la ejecución del módulo,

cuando es invocado ya sea por el usuario directamente en su interacción con el ambiente gráfico de GIMP o por otro módulo interno presente en GIMP.

La declaración del procedimiento *run* se basa en la definición del tipo *GimpRunProc*, misma que se muestra a continuación.

```
typedef void (* GimpRunProc) (gchar *name,
                              gint nparams,
                              GimpParam *param,
                              gint *nreturn_vals,
                              GimpParam **return_vals);
```

Las tareas comúnmente asociadas con el procedimiento *run* incluyen:

- La validación del procedimiento adecuado en la invocación del módulo.
- La definición de los parámetros a retornar.
- La ejecución del módulo, tomando en cuenta el valor de los parámetros que la han sido proporcionados. En particular si el módulo es invocado en forma interactiva, es en este procedimiento en donde se realiza la ejecución de la interfaz gráfica del módulo.

A continuación se muestra la declaración del parámetro *GimpParam*

```
struct _GimpParam
{
  GimpPDBArgType type;
  GimpParamData data;
};
```

El tipo *GimpParamData* es una union que mantiene el valor de los argumentos pasados al procedimiento *run*.

5.2.1.3.1 Valores de retorno

Con base en las especificaciones de la interfaz para la programación de un módulo en GIMP se espera que el primer valor retornado corresponda al código de estado. Este valor es proporcionado por medio de una variable de tipo *GIMP_PDB_STATUS* correspondiente a su vez al tipo *GimpPDBArgType*, y con uno de los valores del tipo *GimpPDBStatusType* que se muestran en la siguiente definición:

```
typedef enum
{
    GIMP_PDB_EXECUTION_ERROR,
    GIMP_PDB_CALLING_ERROR,
    GIMP_PDB_PASS_THROUGH,
    GIMP_PDB_SUCCESS,
    GIMP_PDB_CANCEL
} GimpPDBStatusType;
```

5.2.1.4 Procedimientos para la manipulación de imágenes en GIMP

En esta sección se estudia gran parte de la arquitectura presente en GIMP para el soporte de las imágenes, incluyendo por supuesto la descripción de los procedimientos necesarios para su manipulación.

Indudablemente el elemento principal sobre el cual gira todo el funcionamiento de GIMP, son las imágenes, pues se constituyen como los objetos en donde realmente se aplica todo el procesamiento implementado por los módulos presentes en GIMP, además son los elementos que reflejan los resultados de dicha manipulación. En este sentido se puede afirmar que un módulo no sirve de mucho si no realiza ningún procesamiento sobre una imagen.

En GIMP existen los conceptos de máscaras, canales y capas, sin embargo cada uno de ellos se puede describir como un grupo de píxeles que pueden ser configurados para mantener un valor determinado (asociado a un color), es decir, para dibujar sobre ellos. En consecuencia todos estos objetos son modelados bajo el mismo concepto y en ese sentido pueden en el ser manipulados utilizando los mismos procedimientos. Para lograr la consistencia con esta afirmación GIMP introduce el concepto de *drawables*, categoría en la que incluye a las máscaras, canales y capas. En consecuencia se puede inferir que una imagen no es más que la unión de un conjunto de drawables.

Desde la perspectiva de un módulo las imágenes no constituyen el foco principal de atención debido a que los drawables contienen realmente el bloque de píxeles de trabajo. en cualquier caso la referencia más compleja del módulo hacia una imagen corresponde a un entero (`gint32`), que no es más que un identificador por medio del cual el módulo tiene acceso a la imagen.

El concepto drawable es implementado en GIMP por medio de una estructura cuya definición se presenta a continuación.

```

/* esta definición se encuentra en el archivo cabecera gimp.h */
struct _GimpDrawable
{
  gint32  id;           // drawable ID
  guint   width;       // ancho del drawble
  guint   height;      // alto del drawble
  guint   bpp;         // bytes por pixel del drawable
  guint   ntile_rows;  // número de bloques por renglón que abarca
  guint   ntile_cols;  // número de bloques por columna que abarca
  GimpTile *tiles;     // los bloques normales
  GimpTile *shadow_tiles; // los bloques de respaldo
};

```

En este punto no se requiere una descripción más detallada de los campos introducidos en el esquema anterior debido a que generalmente el programador de un módulo no tiene una interacción directa con estos datos, en su lugar utiliza un conjunto de funciones previamente definidas que se encargan de obtener y ajustar los valores de dichos campos o variables.

Por ejemplo, para obtener la referencia a un objeto de tipo *GimpDrawable* desde el campo *ID* perteneciente a otro objeto de tipo *drawable*, se utiliza la función:

```
GimpDrawable* gimp_drawable_get(gint32 drawable_ID)
```

La cual reserva memoria e inicializa un objeto de tipo *GimpDrawable*. Esta función retorna un puntero hacia el objeto recién creado. Esta función se aplica a cualquier tipo de identificador, es decir, si *drawable_ID* corresponde al identificador de un capa o de un canal, el apuntador al objeto reciente creado puede ser aplicado indistintamente en todos los casos en donde por supuesto sea válida su utilización, como por ejemplo, en la invocación de determinadas funciones.

5.2.1.4.1 Sistema de coordenadas

La posición relativa de cada uno de los pixeles que pertenecen a una imagen es uno de los conceptos más importantes e imprescindibles que deben ser definidos para la manipulación de una imagen en el cuerpo de un módulo. Frecuentemente es necesario determinar la posición de un objeto localizado en la imagen, en tal situación, es posible describir esta posición como la distancia que existe entre el objeto y un punto de referencia, que puede corresponder al origen de coordenadas de la imagen, al origen de la capa o al origen del área de trabajo actualmente seleccionada.

GIMP toma un enfoque más sencillo y con base en el concepto de drawable, define el origen de coordenadas como la esquina superior izquierda del drawable, de esta manera es posible calcular las coordenadas de cualquier punto localizado en el área comprendida por el drawable.

Para determinar la localización de una selección realizada en el drawable, y para determinar el desplazamiento del drawable con relación a la imagen se utilizan las funciones siguientes.

```
/* Esta función determina la posición del área rectangular asociada a la selección actual, tomando como referencia el punto de coordenadas definido por el drawable especificado. Esta función retorna un valor distinto de cero si existe una selección en el drawable*/
```

```
gint gimp_drawable_mask_bounds (gint32  drawable_ID,  
                                gint   *x1, gint   *y1,  
                                gint   *x2, gint   *y2);
```

```
/* Determina el desplazamiento del drawable referenciado por drawable_ID*/
```

```
void gimp_drawable_offsets  (gint32  drawable_ID,  
                              gint   *offset_x,  
                              gint   *offset_y);
```

Los argumentos `x1` y `y1` de la función `gimp_drawable_mask_bounds` corresponden sucesivamente a las coordenadas en el eje `x` y en el eje `y`, del punto localizado en la esquina superior izquierda del área seleccionada, en forma análoga los argumentos `x2` y `y2` corresponden a las coordenadas del punto localizado en la esquina inferior derecha del área subyacente. Esto se muestra en la figura 5.1.

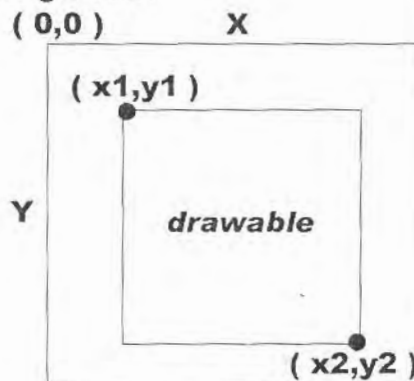


Figura 5.1: Representación gráfica de los argumentos de la función `gimp_drawable_mask_bounds`

Los argumentos `offset_x` y `offset_y`, que utiliza la función `gimp_drawable_offsets`, corresponden sucesivamente a la distancia en el eje `x`, y a la distancia en el eje `y` entre el origen de coordenadas del drawable y el origen de coordenadas de la imagen. Esto se muestra gráficamente en la figura 5.2.

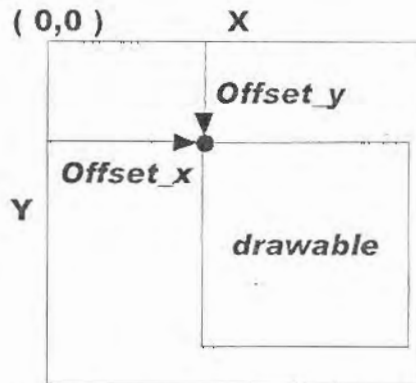


Figura 5.2: Representación gráfica de los argumentos de la función `gimp_drawable_offset`

5.2.1.4.2 Región de píxeles

Una vez que el sistema de coordenadas utilizado por GIMP ha sido descrito, el siguiente paso consiste en determinar el procedimiento a seguir para la manipulación de los píxeles presentes en un `drawable`. En este sentido GIMP introduce el concepto denominado *región de píxeles*, que representa el conjunto de píxeles que pueden ser manipulados directamente por el módulo.

En GIMP una región de píxeles es implementada como un tipo estructura identificado por el nombre `GimpPixelRgn`, la declaración de esta estructura no se detallará debido a que el programador de un módulo generalmente utiliza llamadas a funciones que requieren únicamente la referencia hacia un objeto de tipo `GimpPixelRgn`. Para su revisión, esta estructura se encuentra definida en el archivo cabecera `gimp.h`.

Con el objetivo de poder trabajar con una región de píxeles, es necesario llevar a cabo como primer paso un proceso de inicialización de dicha región, para tal propósito se utiliza la siguiente función:

```
void gimp_pixel_rgn_init (GimpPixelRgn* pr, GimpDrawable* drawable, int x,
                        int y, int width, int height, int dirty, int shadow);
```

En la tabla 5.3 se muestra una breve descripción de los argumentos utilizados por la función. `gimp_pixel_rgn_init`.

Tabla 5.3: Descripción de los argumentos para la función `gimp_pixel_rgn_init`.

| ARGUMENTO | DESCRIPCIÓN |
|-----------------------|---|
| <code>pr</code> | Este argumento es un apuntador que hace referencia a una estructura de tipo <code>GimpPixelRgn</code> , que corresponde a la región de pixeles que habrá de inicializarse. |
| <code>drawable</code> | Este argumento también es un puntero, pero este hace referencia a una estructura de tipo <code>GimpDrawable</code> , asociada al <code>drawable</code> sobre el cual habrá de obtenerse la región de pixeles. |
| <code>x</code> | Determina el punto origen a lo largo de la coordenada <code>x</code> , en el marco de referencia del <code>drawable</code> , a partir del cual se obtendrá la región de pixeles a inicializar. Para la definición de este punto se toma como referencia el origen de coordenadas del objeto referenciado por el argumento <code>drawable</code> . |
| <code>y</code> | Determina el punto origen sobre la coordenada <code>y</code> , en el marco de referencia del <code>drawable</code> , a partir del cual se obtendrá la región de pixeles a inicializar. Para la definición de este punto se toma como referencia el origen de coordenadas del objeto referenciado por <code>drawable</code> . |
| <code>width</code> | Determina el ancho (en pixeles) sobre la coordenada <code>x</code> , que tendrá la región de pixeles que habra de inicializarse. El ancho se toma a partir del punto definido por el argumento <code>x</code> . |
| <code>height</code> | Determina el alto (en pixeles) sobre la coordenada <code>y</code> , que tendrá la región de pixeles que habra de inicializarse. El alto se toma a partir del punto definido por el argumento <code>y</code> . |
| <code>dirty</code> | Un bloque de pixeles definido como <i>dirty</i> o <i>sucio</i> es aquel que ha sido modificado de alguna forma. En este sentido un bloque que no ha sido etiquetado como <i>sucio</i> no será actualizado, y por lo tanto se perderá cualquier cambio aplicado a este bloque. Cuando una región de pixeles es inicializada como <i>sucia</i> , la función encargada de procesar los datos en dicha región, considera que cada bloque de trabajo, es un bloque <i>sucio</i> , de esta forma es posible registrar cualquier cambio producido en esta región. |
| <code>shadow</code> | Un bloque de pixeles definido como <i>Shadow</i> o <i>respaldo</i> , indica a Gimp la necesidad de utilizar un buffer temporal para almacenar los cambios realizados en el bloque. Este enfoque ofrece algunas ventajas, como la posibilidad de mantener intacta la imagen original, además esto facilita y mejora la implementación del proceso de eliminación de los cambios realizados sobre la imagen. |

Para establecer el valor de los parámetros *Dirty* y *Shadow* presentados en la tabla anterior frecuentemente se utilizan las constantes booleanas `TRUE` y `FALSE`. Estos valores determinan la ausencia o la presencia de los atributos o características correspondientes en los bloques de pixeles, de la región que habrá de inicializarse. De esta forma, al tomar en cuenta únicamente estos dos parámetros se producen cuatro combinaciones diferentes asociadas a las distintas formas de inicialización que es posible aplicar a una región de pixeles.

El tipo de inicialización de una región de píxeles, determina el propósito o el tipo de operaciones que se pueden realizar sobre dicha región. Esta relación se muestra en la tabla 5.4.

Tabla 5.4: Configuración de los atributos dirty y shadow

| DIRTY | SHADOW | PROPÓSITO |
|-------|--------|---|
| TRUE | TRUE | Esta configuración es aplicable si se requiere escribir en la región de píxeles, además de registrar o cancelar los cambios producidos sobre la imagen, |
| TRUE | FALSE | En esta caso cualquier operación de escritura sobre la región de píxeles se realiza sobre la imagen original. |
| FALSE | TRUE | Esta configuración es útil cuando se requiere leer directamente de los bloques de respaldo que han sido escritos recientemente. |
| FALSE | FALSE | Esta configuración es aplicable en situaciones en donde únicamente se necesitan leer los datos de la región de píxeles. |

En GIMP existen un conjunto de funciones que obtienen los datos asociados a una imagen y los colocan en un buffer que ha sido creado previamente con ese propósito. Naturalmente estas funciones operan sobre la región de píxeles de una imagen y no con la imagen directamente. Las funciones *pixel_rgn_get_pixel*, *pixel_rgn_get_row*, *pixel_rgn_get_col* y *pixel_rgn_get_rect* constituyen un ejemplo de este tipo de funciones. El formato de los datos que serán almacenados en el buffer dependen del tipo de imagen del cual fueron sustraídos. Por ejemplo para un imagen en escala de grises se almacenará en el buffer un arreglo lineal de valores, en el caso de una imagen RGB, se almacenará la información correspondiente a los 3 canales que forman la imagen de manera secuencial en el buffer. En el archivo cabecera *gimp.h* se puede encontrar una descripción más detallada de estas funciones, así como de otras funciones asociadas con la manipulación de los datos presentes en una región de píxeles.

5.2.1.4.3 Aplicación de los cambios realizados sobre una región de píxeles

Una vez que se han aplicado los cambios sobre la imagen es necesario asegurarse que estos cambios se actualicen en el entorno de GIMP. La secuencia necesaria para realizar esta tarea, se muestra a continuación.


```

/* apuntador al área de trabajo de la imagen */
GimpDrawable *drawable;

/* ... */

/* La siguiente función se utiliza para eliminar posible bloques "sucios" en la imagen de
trabajo */
gimp_drawable_flush (drawable);

/* La siguiente función integra en la imagen, los datos escritos en los bloques de respaldo.
Si el segundo argumento es TRUE, esta acción podrá ser cancelada. Si no se utiliza esta
función, los datos escritos en los bloques de respaldo se perderán*/
gimp_drawable_merge_shadow (drawable->id, TRUE);

/* El siguiente procedimiento se utiliza para actualizar la porción modificada de la imagen
referenciada por drawable. En particular actualiza los componentes de previsualización de
la imagen */
gimp_drawable_update (drawable->id, x1, y1, width, height);

/* Esta función registra los cambios sobre la imagen en los visualizadores en pantalla */
gimp_displays_flush();

/* libera la memoria asignada al puntero drawable, incluyendo cualquier bloque utilizado.*/
gimp_drawable_detach(drawable);

```

5.2.1.5 Estructura general de un módulo programado para GIMP

En conclusión un módulo escrito para GIMP debe contener esencialmente los siguientes elementos estructurales:

1. **#include <libgimp/gimp.h>**
2. **PLUG_IN_INFO**, incluye la referencia de los procedimientos que serán invocados por GIMP para la operación del módulo.
3. **MAIN()**
4. **Una función query**, incluye una llamada a la función `gimp_install_procedure`, que se encarga de la instalación del módulo.
5. **Una función run**, constituye el procedimiento que será invocado para la ejecución del módulo. En este procedimiento se ejecuta la interfaz gráfica del módulo y se define un valor de retorno, con el estado de ejecución final del módulo.

5.2.2 Implementación de la interfaz gráfica

Para la implementación de la interfaz gráfica de la herramienta propuesta en esta tesis se utilizó el paquete GTK, debido a que es la plataforma de desarrollo utilizada por GIMP.

5.2.2.1 GTK

GTK (GIMP Toolkit) es una librería para crear interfaces gráficas de usuario. GTK es distribuido bajo los términos de la licencia LGPL, de tal manera que es posible desarrollar software libre o comercial, utilizando esta herramienta [27].

GTK es llamado el *GIMP toolkit* debido a que fue escrito originalmente para desarrollar el programa de manipulación de imágenes de la GNU, también conocido como GIMP, sin embargo ha sido utilizado en numerosos proyectos de desarrollo de software, incluyendo el proyecto GNOME². GTK esta construido sobre GDK (GIMP Drawing Kit) que constituye una capa de software de alto nivel para las funciones de bajo nivel que permiten el acceso a los procedimientos del entorno gráfico subyacente [27].

GTK es en esencia una interfaz orientada a objetos para la programación de aplicación. A pesar de que se encuentra escrita completamente en el lenguaje C, ha sido implementada usando la idea de clases y funciones virtuales (punteros a funciones).

Hay un tercer componente en la distribución de GTK llamado Glib, el cual contiene algunos substitutos para algunas funciones estándar, e incluye además un conjunto de funciones para manejar listas ligadas. Las funciones sustituto incluidas en Glib son usadas para aumentar la portabilidad de GTK. Debido a que algunas funciones funciones implementadas por GTK no se encuentran disponibles en otros sistemas Unix.

El desarrollo de un módulo para GIMP involucra la utilización de la interfaz de GTK para el lenguaje C. Existen implementaciones de GTK para muchos otro lenguajes incluyendo C++, Guile, Perl, Python, TOM, Ada95, C Objetivo, Pascal y Eiffel [27].

² GNU Network Object Model Environment

5.2.2.2 Diseño

La interfaz gráfica para esta herramienta debe contener los elementos gráficos necesarios, para facilitar la ejecución de las tareas involucradas en la segmentación de imágenes en color, utilizando el sistema de color modificado I1I2I3³. Las fases del proceso de segmentación son:

1. Aplicación de la fórmula⁴ apropiada del sistema de color modificado I1I2I3 a la imagen en color. Como resultado de esta fase se obtiene una imagen en tonos de gris, que constituye la imagen de trabajo.
2. Binarización de la imagen de trabajo. Para ejecutar esta fase es necesario seleccionar el umbral de binarización a partir del histograma de la imagen de trabajo. El resultado de la ejecución de esta etapa es una imagen binaria, que corresponde a las áreas segmentadas de la imagen en color.
3. Ajustar las propiedades de la imagen en color. En esta fase se aplican diferentes operaciones para modificar las propiedades de las áreas segmentadas de la imagen en color, utilizando como máscara la imagen binaria generada en el paso anterior. Estas operaciones incluyen básicamente el cambio de color de las áreas segmentadas y la eliminación del fondo de la imagen en color, sin modificar las áreas segmentadas.

Estas etapas se pueden modelar con los casos de uso, presentados en la figura 5.3.

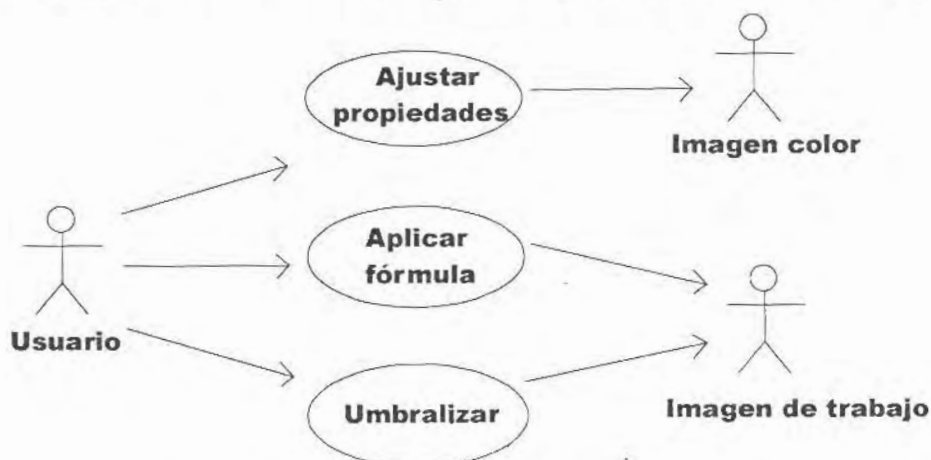


Figura 5.3: Diagrama de casos de uso para la herramienta de segmentación

³ véase la sección 5.1.2

⁴ estas fórmulas se describen en la sección 5.1.2

Con base en la metodología de UML⁵ se pueden generar diagramas de iteración para los casos de uso presentados en la figura 5.3. El diagrama de iteración correspondiente al procedimiento para aplicar la fórmula del sistema I1I2I3, se muestra en la figura 5.4

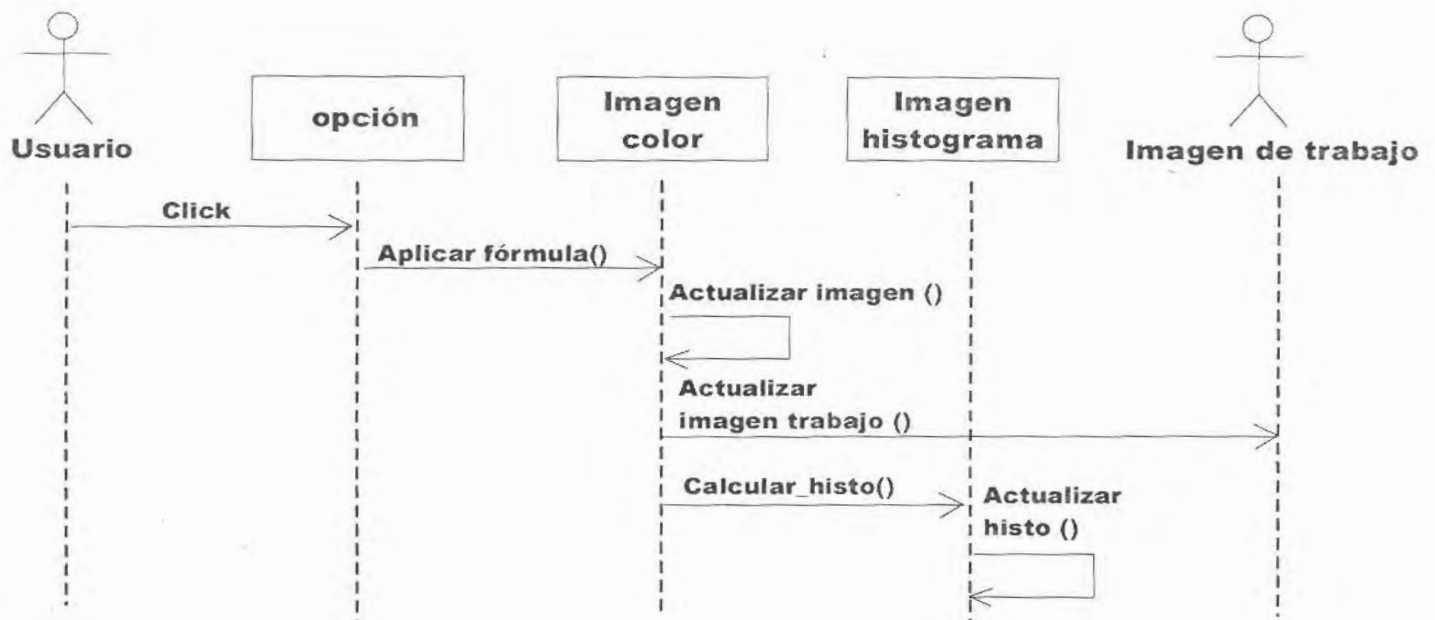


Figura 5.4: Diagrama de iteración para aplicar una fórmula del sistema I1I2I3

La secuencia de pasos para el diagrama de iteración de la figura 5.4 inicia cuando el usuario selecciona la fórmula del sistema I1I2I3 que se aplicará a la imagen en color, como resultado de la ejecución de este proceso se produce la imagen de trabajo, que tiene un formato en tonos de gris. A partir de este momento es posible calcular el histograma de la imagen de trabajo y presentarlo en forma gráfica en la imagen del histograma.

Para obtener el umbral de la imagen de trabajo se producen tres diagramas de iteración, que corresponden a las figuras 5.5, 5.6 y 5.7.

⁵ Unified Modelling Language: Lenguaje de Modelado Unificado

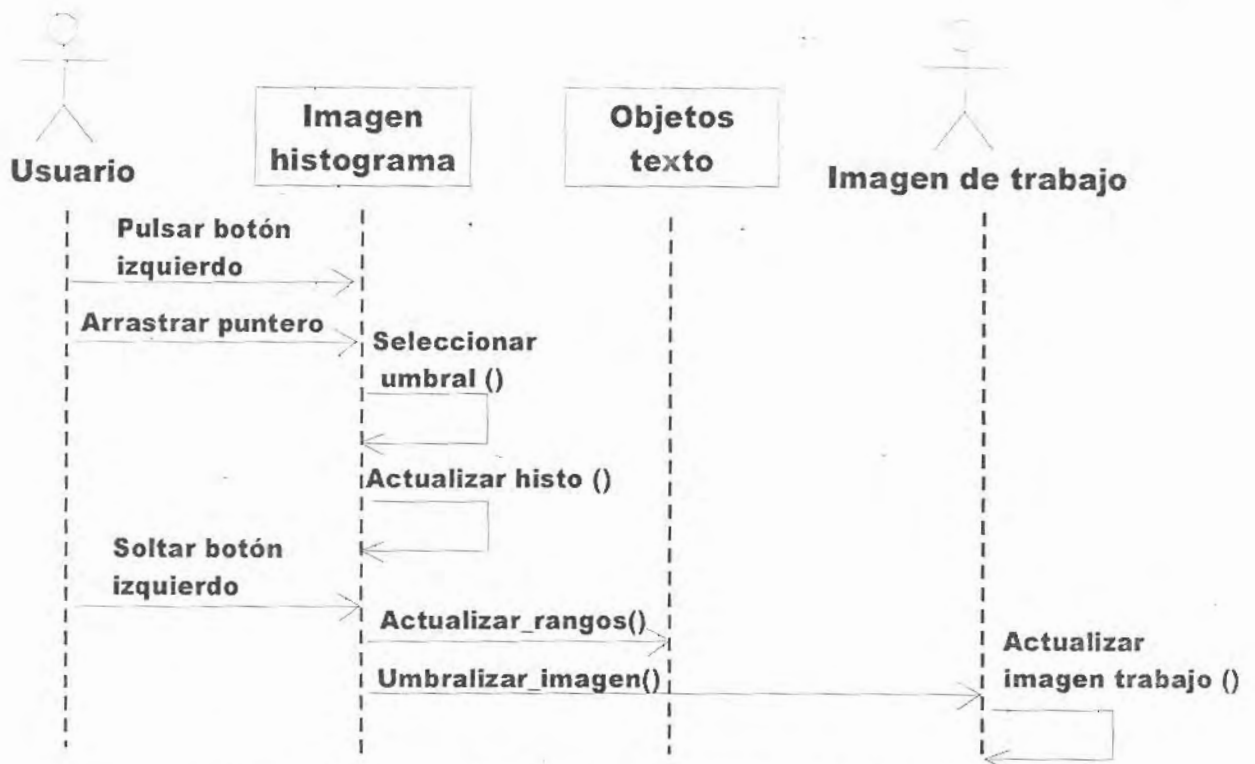


Figura 5.5: Diagrama de iteración para binarizar la imagen de trabajo

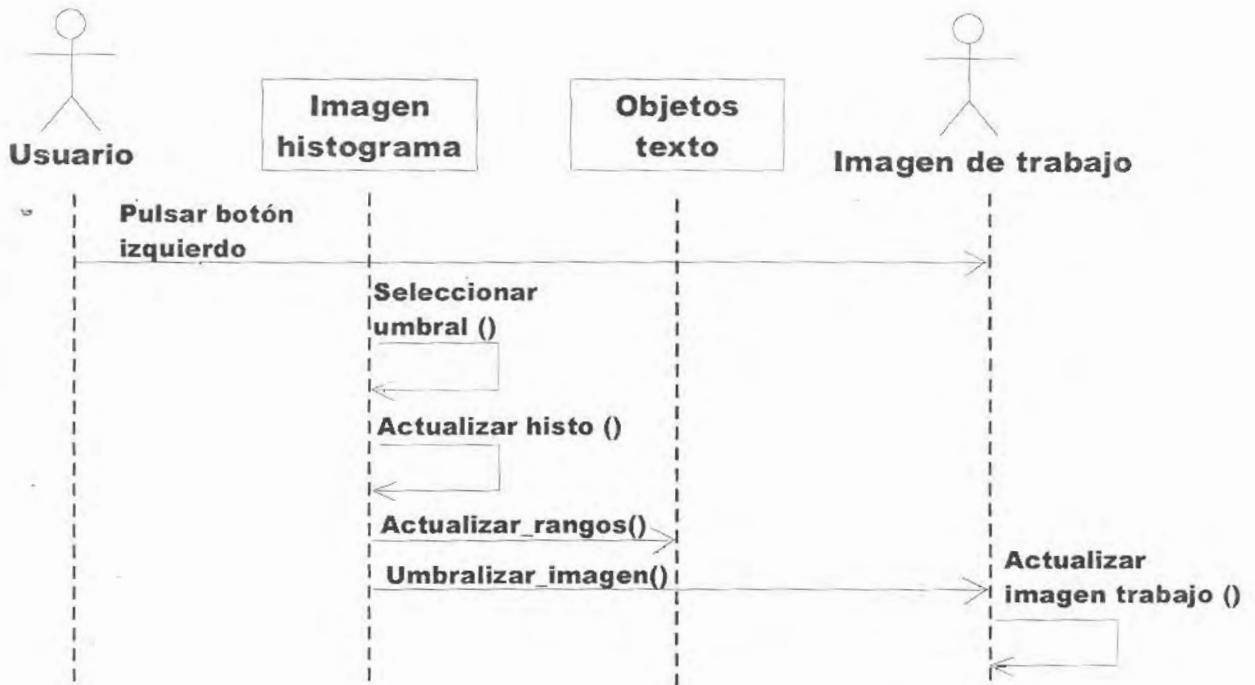


Figura 5.6: Diagrama 2 de iteración para binarizar la imagen de trabajo

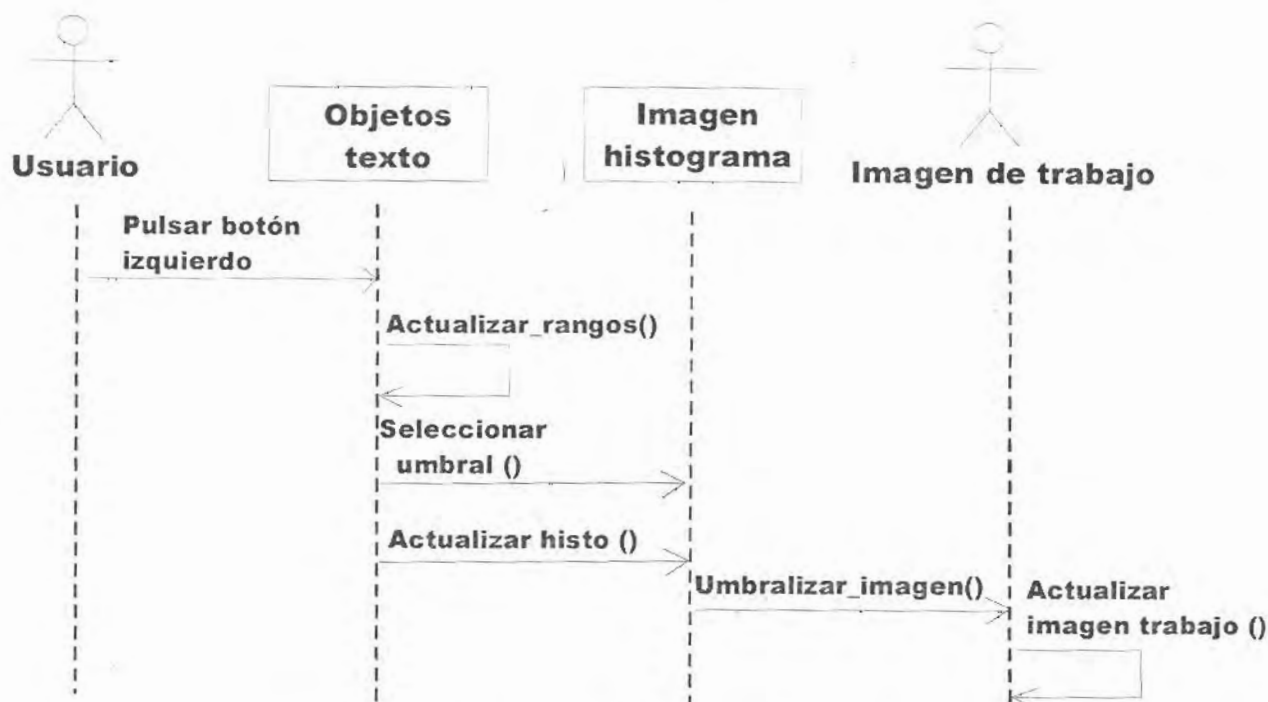


Figura 5.7: Diagrama 3 de iteración para binarizar la imagen de trabajo

Los diagramas de iteración de las figuras 5.5, 5.6 y 5.7 muestran tres métodos alternativos para seleccionar el umbral de binarización de la imagen de trabajo.

El primer método (figura 5.5) consiste en utilizar el puntero del mouse para definir el umbral directamente sobre la imagen del histograma. Cuando el usuario pulsa el botón derecho y arrastra el puntero sobre esta imagen, la porción seleccionada de la imagen es resaltada, para lograr este efecto es necesario actualizar el rango del umbral, así como el contenido de la imagen del histograma. Una vez que el usuario suelta el botón izquierdo el rango del umbral queda definido, y es presentado en dos objetos texto con la descripción de los rangos izquierdo y derecho, respectivamente. En seguida se realiza el proceso de binarización de la imagen de trabajo y se actualiza su contenido con la información binaria correspondiente.

El segundo método (5.6) consiste en seleccionar el rango del umbral directamente sobre la imagen de trabajo, es decir, cuando el usuario pulsa el botón izquierdo del mouse sobre la imagen de trabajo, el valor de pixel seleccionado se toma con el valor para el rango izquierdo y derecho del umbral. Este valor se presenta en los objetos texto correspondientes y se realiza el proceso de binarización sobre la imagen de trabajo, con la actualización subyacente.

El último método (5.7) consiste en definir el umbral de binarización utilizando los objetos texto con la descripción de los rangos correspondientes. En este caso se puede aumentar o disminuir el valor del rango para el umbral, pulsando el botón izquierdo del mouse sobre el objeto texto apropiado. Este método es más preciso debido a que el rango aumenta o disminuye en unidades fijas, normalmente en una unidad. Cuando el umbral ha sido redefinido, es necesario actualizar la porción resaltada de la imagen del histograma, y a continuación se realiza nuevamente el proceso de binarización y actualización de la imagen de trabajo.

En la figura 5.8 se muestra el diagrama de iteración del caso de uso para cambiar las propiedades de la imagen en color.

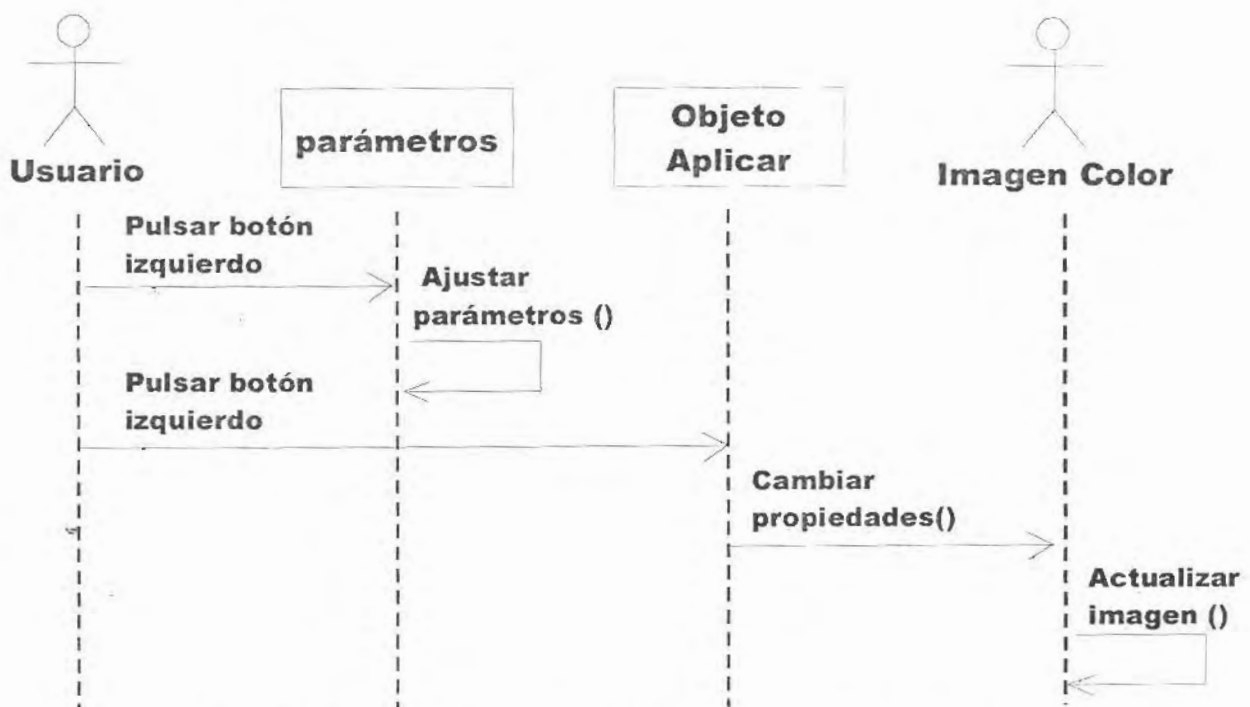


Figura 5.8: Diagrama de iteración para cambiar las propiedades de la imagen en color

El diagrama de iteración de la figura 5.8 inicia cuando el usuario selecciona la configuración de los parámetros que indican el tipo de operación que se realizará sobre la imagen de trabajo. Estas operaciones incluyen básicamente el cambio de color de las áreas segmentadas y la eliminación del fondo de la imagen, sin modificar las porciones segmentadas. Una vez que los parámetros de configuración han sido ajustados el usuario debe pulsar el botón izquierdo sobre el objeto aplicar para iniciar la ejecución de las operaciones correspondientes sobre la imagen en color y en seguida actualizar su representación gráfica.

Conclusiones

Linux presenta numerosas características altamente productivas, que le han permitido experimentar un gran crecimiento y una rápida difusión en el campo de los sistemas operativos, fenómeno que se ha visto reforzado por la presencia de una serie de argumentos financieros, de mercado, de servicios, técnicos y de soluciones, que justifican la utilización de Linux como un sistema operativo apropiado, para la ejecución de tareas con múltiples propósitos, incluyendo especialmente aquellas relacionadas con el procesamiento digital de imágenes. Un argumento adicional importante en este sentido, consiste en el hecho de que el diseño de Linux se basa en los preceptos de la filosofía del software libre, de tal manera que hereda de forma natural las cualidades asociadas con esta concepción del software, reflejadas en su portabilidad, confiabilidad, interoperatividad y su alto desempeño.

En la actualidad el soporte para dispositivos de hardware en Linux no es completo, sin embargo la gama de dispositivos compatibles con este sistema operativo es amplia, y se pueden encontrar numerosas fuentes y recursos de información al respecto.

En particular existen controladores en Linux para múltiples modelos de escáners de puerto paralelo, USB o SCSI, sin embargo, los dispositivos con un adaptador de bus paralelo a SCSI, requieren de un proceso de instalación más elaborado, que involucra generalmente, la aplicación de un parche al Kernel de Linux, en donde la implementación de un script, constituye un requerimiento frecuente para facilitar la ejecución de estas tareas.

En el campo de las soluciones de impresión, los sistemas LPD y CUPS cuentan con las características y herramientas suficientes para satisfacer las necesidades de impresión en Linux, para una gran cantidad de dispositivos de impresión, especialmente aquellos que utilizan un protocolo de comunicación estándar como el lenguaje Postscript o PCL. Estos lenguajes son utilizados particularmente por diversas impresoras de alto y bajo formato, de la marca HP.

El sistema de impresión LPD ha sido el estándar de los sistemas Unix por muchos años, y aún continúa integrándose en diversas distribuciones de Linux, sin embargo el sistema CUPS constituye una implementación más completa que se fundamenta en las especificaciones del protocolo IPP, que se plantea como el sustituto para el arcaico LPD.

En ambientes de cómputo de gran demanda, en donde existen múltiples usuarios y dispositivos de impresión, resulta altamente conveniente instalar un servidor de impresión

encargado de procesar las solicitudes de impresión realizadas desde clientes remotos. En este caso se implementó un servidor que realmente consiste en un programa que se ejecuta sobre el demonio LPD, y que implementan las políticas de administración definidas para el servidor.

El pseudo-servidor de impresión puede controlar la carga de trabajo, asignando las tareas en espera a los dispositivos de impresión libres, sin embargo no implementa ningún mecanismo de seguridad adicional al utilizado por el sistema LPD.

GIMP presenta una arquitectura altamente modular que permite y facilita la integración de nuevos módulos externos. En el proceso de implementación de estos módulos, es posible integrar una interfaz gráfica y utilizar las extensiones y herramientas instaladas en GIMP, todo esto en el ámbito del software libre, con la presencia de todas sus cualidades asociadas. En consecuencia se puede afirmar que GIMP constituye una plataforma altamente conveniente para la implementación de métodos o algoritmos asociados con el procesamiento digital de imágenes, realizados por cualquier programador con el conocimiento suficiente y necesario de la interfaz definida en GIMP y de las plataformas de desarrollo que lo soportan.

El módulo desarrollado para GIMP encargado de la segmentación de imágenes en color, utilizando el sistema de color modificado I1I2I3, presenta elementos gráficos (con base en la librería GTK) que facilitan la ejecución de este proceso en comparación con la limitada eficiencia que presentan los programas con una interfaz en modo texto.

El programa xscanimage se puede instalar como un nuevo módulo en el entorno funcional de GIMP. En forma adicional este programa para la manipulación de imágenes cuenta con un módulo de impresión, capaz de operar con los sistemas LPD y CUPS. En consecuencia es posible realizar el proceso completo de digitalización e impresión en el sistema GIMP, sin la necesidad de utilizar múltiples programas. Además se tiene la ventaja adicional de utilizar en el mismo ambiente de ejecución, las herramientas implementadas por un usuario en particular para realizar una tarea determinada.

En la práctica los requerimientos del procesamiento digital de imágenes, involucran la utilización de herramientas o librerías especializadas, esto desde luego, puede significarse como un argumento importante asociado con la dificultad de implementar un algoritmo determinado en GIMP, sin embargo, en la mayoría de los casos es posible encontrar en el mundo del software libre los módulos apropiados para satisfacer las necesidades de procesamiento requeridas.

Referencias

- [1] BRUNO CORNEC, *Hp how to*, 1997, Hewlett Packard.
- [2] GNU project, <http://www.gnu.org/>.
- [3] Open Source project, <http://www.opensource.org/>.
- [4] FreeBSD project, <http://www.freebsd.org/>.
- [5] LINUX Tutorial e implementación,
<http://atenea.udistrital.edu.co/grupos/glud/glud/manuales/Linux/http/mitutorlinux/contenidos.html/>.
- [6] International Data Corporation, <http://www.idc.com/>.
- [7] Netcraft Inc, <http://www.netcraft.com/>.
- [8] Hardware HOWTO,
<http://www.medasys-digital-systems.fr/mirror/linux/LDP/HOWTO/Hardware-HOWTO.html/>.
- [9] Parallel Ports escáner under Linux,
<http://www.buzzard.org.uk/jonathan/scanners.html/>.
- [10] DAVID MOSBERGER, *The SANE Scanner Interface*,
<http://panda.mostang.com/sane/lj98/>
- [11] Linux and parallel port SCSI adapters, <http://www.torque.net/parport/parscsi.html/>.
- [12] The Linux 2.4 SCSI subsystem HOWTO,
<http://www.linuxdoc.org/HOWTO/SCSI-2.4-HOWTO/llevelpseudo.html/>.
- [13] The ppSCSI suite of drivers, <http://www.torque.net/parport/ppscsi.html/>.
- [14] GRANT TAYLOR, *The Printing HOWTO*, versión 5.15, 2001.
<http://www.linuxprinting.org/howto/>
- [15] Printer Compatibility Database, <http://www.linuxprinting.org/database.html/>.
- [16] Ghostscript Printer Compatibility, <http://www.cs.wisc.edu/~ghost/doc/printer.htm>
- [17] Grupos de discusión Dejanews,
http://groups.google.com/googlegroups/deja_announcement.html
- [18] PATRICK A POWELL, *LPRng HOW TO*, 1996-2001
<http://www.lprng.com/LPRng-HOWTO/LPRng-HOWTO.html>
- [19] MICHAEL SWEET, *An Overview of the Common UNIX Printing System Version 1.1*, Easy Software Products, July 10, 2000, <http://www.cups.org/overview.html>

- [20] **LinuxPrinting.org**, <http://www.linuxprinting.org/howto/>
- [21] **Manual de printcap**, <http://www.linuxprinting.org/man/printcap.5.html>
- [22] **Manual de lpc**, <http://www.linuxprinting.org/man/lpq.1.html>.
- [23] **SOURCE FORGE: proyecto CUPS**, <http://sourceforge.net/projects/cups/>
- [25] **GIMP The GNU Image Manipulation Program**,
http://www.gimp.org/the_gimp.html
- [25] **GUERRERO RAMÍREZ MARÍA LUISA ANTONIETA**, *Segmentación de Imágenes Digitales en Color Modificando el sistema de Color III2I3*.
- [26] **KEVIN TURNER**, *Writing a GIMP Plug-In*, 1998-2000.
<http://gimp-plug-ins.sourceforge.net/doc/Writing/html/plug-in.html>
- [27] **TONY GALE, IAN MAIN**, *GTK+ 1.2 Tutorial*, Marzo 1 2001
<http://www.gtk.org/tutorial/>

RFC's

- [28] **L. MCLAUGHLIN III**, *Line Printer Daemon Protocol*, RFC 1179, Agosto 1990.
- [29] **F.D. WRIGHT**, *Design Goals for an Internet Printing Protocol*, RFC 2567, Abril 1999.

Apéndice A

Código fuente del script para aplicar el parche patch-2.2.15pre14-ppscsi1 al kernel de Linux.

```
#!/bin/bash

echo "*****"
echo -e "\n SCRIPT PARA APLICAR EL PARCHÉ PPSCSI AL KERNEL DE
LINUX\n"
echo "*****"

#Nombre del archivo con el parche para el kernel 2.2.10
PARCHE=patch-2.2.15pre14-ppscsi1

#Nombre del archivo con el parche modificado para el kernel de Linux elegido
FILE=parche_ppscsi

#Definición del directorio actual y del directorio /usr/src
DIRACT=`pwd`
DIR=/usr/src

#Verifica el acceso al directorio /usr/src
cd $DIR $> /dev/null
if test $? -ne 0
then echo -e "\n !!!No se encontro el directorio con los archivos fuente del kernel de
Linux!!!"
exit 2
elif !(test -w $DIR)
then echo -e "\n!!!No se puede escribir en el archivo, posiblemente tenga que
reingresar como el usuario root (con el comando SU!!!"
exit 2
fi

#Verifica si se encuentra el parche en los sitios apropiados
if test -r $PARCHE
then
DIRACT=$DIR
elif !(test -r ${DIRACT}/$PARCHE)
then echo -e "\n!!!No se encuentra el archivo $PARCHE\n -- coloquelo en $DIR
o $DIRACT"
exit 2
fi
```

```

#Verifica si se encuentra el parche en los sitios apropiados
echo -e "VERSIONES INSTALADAS DEL KERNEL DE LINUX:\n"
CONT=0
DLINUX=`ls /usr/src | grep linux`
for i in $DLINUX
do
  if !(test -h $i)
  then echo -e "\t $i ----- [SCONT]\n"
  DIRS=`echo -e "$DIRS $i"`
  CONT=`expr $CONT + 1`
  fi
done

#Verifica si se encuentra el parche en los sitios apropiados
if test $CONT -lt 1
  then echo -e "\n!NO SE ENCONTRARON LOS ARCHIVOS FUENTE PARA EL
KERNEL DE LINUX!"
  exit 2
fi

#Solicita al usuario la versión del kernel sobre la que se aplicara el parche
OPCION=-1;
until test $OPCION -gt -1 -a $OPCION -lt $CONT
do echo -e -n "\n  ELEGIR LA VERSION SOBRE LA QUE SE APLICARA EL
PARCHE: "
  read OPCION
done

#captura en la variable OPCION el número elegido
set $DIRS
while test $OPCION -gt 0
do shift
  OPCION=`expr $OPCION - 1`
done

#Verifica que el enlace simbólico "Linux" exista y que apunte a la versión apropiada
LINKDIR=`readlink $DIR/linux`
if (test $LINKDIR = "")
then
  ln -s $1 linux
elif !(test $LINKDIR = $1)
then
  until test $OPCION = "s" -o $OPCION = "n"
  do echo -e "\n !!! ATENCION: El enlace simbolico \"linux\" no apunta al
directorio $DIR/$1"
  echo -e -n "\n ¿DESEA CAMBIAR LA REFERENCIA DE ESTE
ENLACE SIMBOLICO (s/n)?: "
  read OPCION
  if (test $OPCION = "n")
  then
    echo -e -n "\n NO SE PUDO APLICAR EL PARCHE CON EXITO\n"

```

```

        exit 2
    else
        rm -f linux
        ln -s $1 linux
    fi
done
fi

```

#Adapta el parche original a la versión del kernel seleccionada por el usuario
sed '/linux-2.2.15pre14/s/'\$1'/g' \$SDIRACT/\$SPARCHE > \$SDIR/\$FILE

#Verifica la creación del nuevo parche

```

if !(test -r $SDIR/$FILE)
then
    echo -e "\n!!!NO SE PUDO CREAR EL NUEVO PARCHE $SDIR/$FILE \n"
fi

```

#Aplica el parche al kernel apropiado

```

echo -e "\n ---- APLICANDO EL PARCHE ---"
patch -p0 < $SDIR/$FILE

```

#Presenta el mensaje del estado correspondiente al finalizar la ejecución del parche.

```

if test $? -ne 0
then echo -e "\n !!!SE PRESENTARON PROBLEMAS AL APLICAR EL
PARCHE AL KERNEL DE LINUX!!!"
    exit 2
else
    echo -e "\n SE APLICO EL PARCHE CON EXITO"
fi

```

Apéndice B

Código fuente del script *fil_virtual* para el servidor de impresión de la sección 4.3.5

En este parámetro se define la lista de colas de impresión presentes en el sistema LPD del servidor

IMPRESORAS="lp1 lp2"

Este parámetro determina el número de veces que el script revisara el estado de cada una de las impresoras definidas en el parámetro anterior

ITERA=3

Si todas las impresoras se encuentran ocupadas, este parámetro determina el tiempo en segundos que el script debe esperar antes de realizar nuevamente el ciclo de verificación

ESPERA=5

LIBRE=0

CONT=0

#Revisa el estado de las impresoras predefinidas, hasta que una impresora se encuentre disponible o se exceda el límite permitido de iteraciones

until test \$LIBRE -ne 0 -o \$CONT -gt \$ITERA

do

DISPONIBLES=0

for IMP in \$IMPRESORAS

do

esta instrucción se utiliza para determinar si un dispositivo determinado se encuentra configurado en la cola de impresión.

./revpri \$IMP 0

if (test \$? -eq 0)

then

DISPONIBLES=1

./revpri \$IMP 1

para verificar si la impresora se encuentra disponible

if (test \$? -eq 0)

then

LIBRE=1

break

fi

fi

done

Si se presentaron errores durante la ejecución de lpq para todas las impresoras, entonces se termina la ejecución del script

```
if (test $DISPONIBLES -eq 0)
  then
    exit 2;
fi
```

si todas las impresoras se encuentran ocupadas entonces es necesario esperar algunos segundos, antes de revisar nuevamente, el estado de las impresoras.

```
if (test $LIBRE -eq 0)
  then
    sleep $ESPERA;
    CONT=`expr $CONT + 1`
  fi
done
```

#si se excedió el número de iteraciones permitidas, entonces se finaliza la ejecución del script

```
if (test $CONT -gt $ITERA)
  then
    exit 2;
fi
```

#Finalmente se envía el trabajo de impresión a la impresora disponible utilizando la utilería lpr

```
lpr -PSIMP
```


Apéndice C

Manual de Usuario

En este apartado se describen las funciones principales del módulo implementando en GIMP para la segmentación de imágenes en color, utilizando el sistema modificado I1I2I3.

A.1 Requerimientos del módulo

Debido a que la aplicación implementada en este trabajo de tesis constituye un módulo para GIMP, sus requerimientos corresponden a los de esta aplicación, es consecuencia el módulo puede ser utilizado prácticamente, en cualquier sistema Linux que tenga instalada la herramienta GIMP.

Adicionalmente para instalar y utilizar el módulo en GIMP, es necesario contar con una versión 1.1.x, en particular en este trabajo de tesis se utilizó la versión 1.1.26. En este punto es importante señalar que como consecuencia del rápido desarrollo de las nuevas versiones para GIMP, este módulo probablemente tendría que ser adaptado, para ajustarse a los requerimientos de las nuevas versiones.

A.2^a Compilación del módulo

La versión con el código fuente para el módulo de GIMP puede ser compilada utilizando el procedimiento que se describe a continuación. La versión precompilada del módulo, lógicamente no requiere la realización de este proceso.

Antes de realizar la compilación del módulo, se debe tener instalado un compilador de C, frecuentemente el compilador GCC. Además es necesario contar con el programa *make*. Estas herramientas se incluyen por defecto en las principales distribuciones de Linux, por lo tanto en muchos de los casos no es necesario instalarlos.

Para facilitar la compilación del módulo se puede utilizar el archivo *Makemod* que contiene un conjunto de definiciones, reglas y acciones para realizar la compilación directa del módulo. Para ejecutar el archivo *Makemod* se debe aplicar el siguiente comando:

make -f Makemod

La ejecución exitosa de este comando requiere la presencia de la distribución con el código fuente para GIMP, debido a que tanto el archivo `Makemod` y el programa fuente del módulo, deben ser colocados en el directorio `.../gimp-1.1.x/plug-ins/common/` presente en su estructura de archivos fuente.

Es necesario señalar que el programa `make` se encarga de ejecutar las acciones y comandos definidos en el archivo `Makemod`, para realizar la compilación del módulo.

A.3 Instalación del módulo

Una vez que se ha ejecutado el proceso de compilación de los archivos fuente del módulo (descrito en la sección anterior), se debe ejecutar el siguiente comando para realizar su instalación.

```
make -f Makemod install
```

Por otra parte, el proceso de instalación para la versión precompilada del módulo, requiere únicamente de la colocación del módulo correspondiente en el directorio `/usr/local/lib/gimp/1.1/plug-ins`.

A.4 Ejecución del módulo

La ejecución del módulo para la segmentación de imágenes en color, involucra la siguientes etapas:

1. Para utilizar el módulo implementado, primero es necesario ejecutar la herramienta GIMP, esto se puede hacer desde una terminal de línea de comandos, escribiendo directamente el comando `gimp`, o utilizando el acceso directo apropiado, presente en el entorno gráfico de Linux. Una vez que GIMP se encuentra en ejecución es posible utilizar todos sus módulos y herramientas.
2. El siguiente paso consiste en abrir la imagen en color. En esta fase GIMP crea una ventana independiente para la nueva imagen abierta.
3. A continuación, se debe pulsar el botón derecho del mouse sobre la imagen de trabajo. En ese momento se presenta un menú, en donde se debe seleccionar la opción "filtros..". Esta opción produce un submenú con nuevas opciones, en este punto se debe elegir la opción "colores..", a continuación se presenta otro submenú que incluye una opción etiquetada como "segmentar..", esta opción corresponde precisamente al nuevo módulo de segmentación. La jerarquía de opciones para el nuevo módulo se muestra en la figura A.1.



Figura A.1 Jerarquía de opciones para la invocación del módulo encargado de la segmentación de imágenes.

4. El siguiente paso lógico, consiste en seleccionar la opción “segmentar...”, lo cual presenta la pantalla principal del módulo. Esta pantalla incluye las opciones necesarias para realizar la segmentación de la imagen con base en el sistema de color modificado I1I2I3. En la figura A.2 se muestra esta pantalla principal



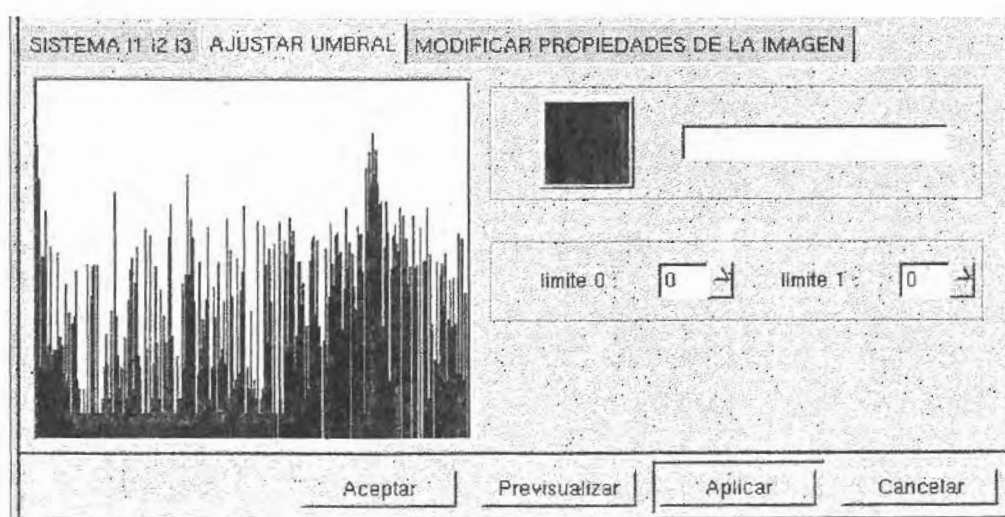
Figura A.2 Pantalla Principal

A.5 Pantalla principal

La interfaz del módulo esta dividida en varias secciones, en la parte superior se presentan dos áreas con imágenes. La imagen de la izquierda con el título *imagen a colorear*, corresponde a una previsualización de la imagen en color , y la imagen de la derecha bajo el título *imagen mascara* es la imagen de trabajo, en donde se observan los resultados de la aplicación de los parámetros utilizados por el algoritmo I1I2I3. Al iniciar la ejecución del módulo, la imagen de trabajo corresponde a la aplicación de la fórmula I1 sobre la imagen en color. Esta carpeta se muestra en la figura A.2.

En la parte inferior de la pantalla principal se presenta una carpeta dividida en 3 secciones, la primera de ellas corresponde a la aplicación de las fórmulas del sistema de color modificado I1I2I3 a la imagen en color, en este sentido las opciones incluyen a las fórmulas I1 (5.1), I2 (5.2), I3 (5.3), I2_1 (5.4) e I2_2 (5.5). En esta área se muestra un arreglo de botones etiquetados con estas opciones. Al presionar un botón determinado, la imagen de trabajo presenta los resultados que se obtienen al aplicar la fórmula correspondiente a la imagen original. Esto es útil para determinar la fórmula que genera los mejores resultados para el proceso de segmentación.

La segunda página de la carpeta situada en la parte inferior de la pantalla principal del módulo, corresponde a la sección bajo el título *AJUSTAR UMBRAL*, en donde se muestra el histograma de la imagen de trabajo. En esta sección se definen los rangos del umbral necesarios para realizar el proceso de binarización de la imagen de trabajo. Esta carpeta se muestra en la figura A3.



A.3 Carpeta *AJUSTAR UMBRAL*

Existen tres métodos alternativos para seleccionar el umbral de binarización de la imagen de trabajo.

El primer método consiste en utilizar el puntero del mouse para definir el umbral directamente sobre la imagen del histograma. Cuando el usuario pulsa el botón derecho y arrastra el puntero sobre esta imagen, la porción seleccionada de la imagen es resaltada, una vez que el usuario suelta el botón izquierdo el rango del umbral queda definido, y es presentado en dos objetos texto con la descripción *limite 0* y *limite 1*. En seguida se realiza el proceso de binarización de la imagen de trabajo y se actualiza su representación gráfica en la pantalla principal. Este proceso se muestra en la figura A.4.

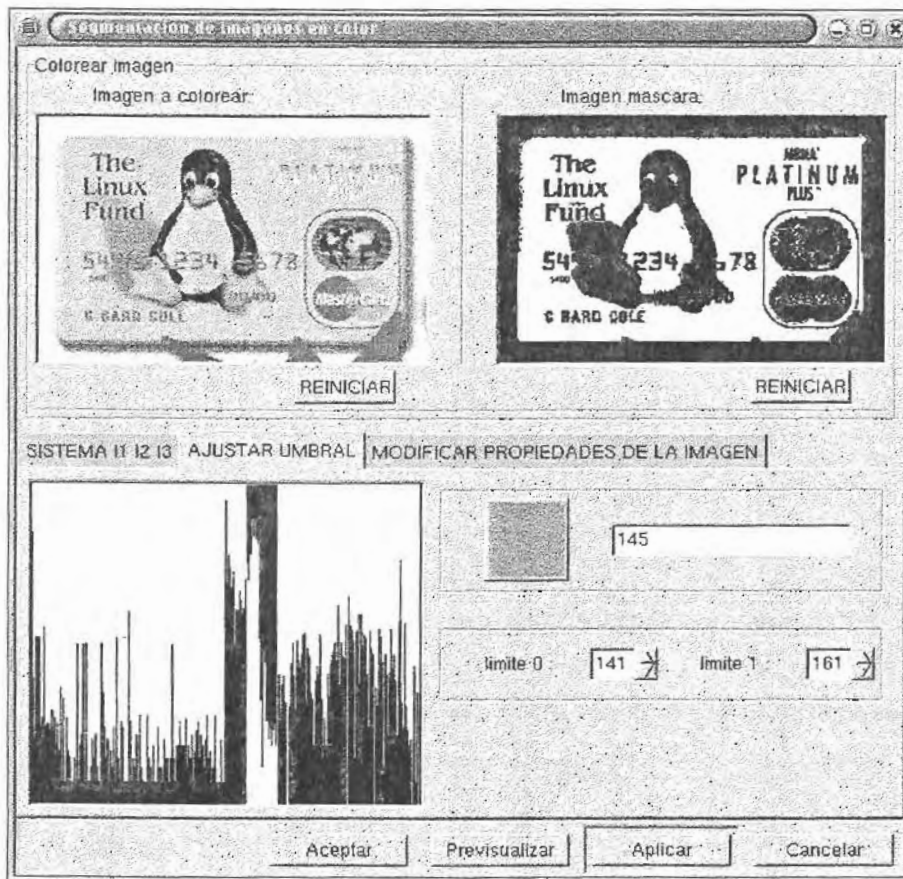


Figura A.4 Resultado de la selección del umbral sobre el histograma de la imagen de trabajo

El segundo método consiste en seleccionar el rango del umbral directamente sobre la imagen de trabajo, es decir, cuando el usuario pulsa el botón izquierdo del mouse sobre la imagen de trabajo, el valor de pixel seleccionado se toma como el valor para el rango izquierdo y derecho del umbral, y se muestra en el botón de color y el componente de texto colocados sobre los objetos con los límites del umbral. A continuación se realiza el proceso de binarización sobre la imagen de trabajo y la actualización subyacente.

El último método consiste en definir el umbral de binarización utilizando los objetos texto límite 0 y límite 1. En este caso se puede aumentar o disminuir el valor del rango para el umbral, pulsando el botón izquierdo del mouse sobre el objeto texto apropiado. Este método es más preciso debido a que el rango aumenta o disminuye en unidades fijas, normalmente en una unidad. Cuando el umbral ha sido redefinido, es necesario actualizar la porción resaltada de la imagen del histograma, y a continuación se realiza nuevamente el proceso de binarización y actualización de la imagen de trabajo.

La última carpeta de la pantalla principal es útil para MODIFICAR PROPIEDADES DE LA IMAGEN, es decir, para cambiar el color de la imagen segmentada o eliminar el fondo, con este propósito se utilizan los componentes presentes en esta carpeta, cuya representación gráfica se muestra en la figura A.5

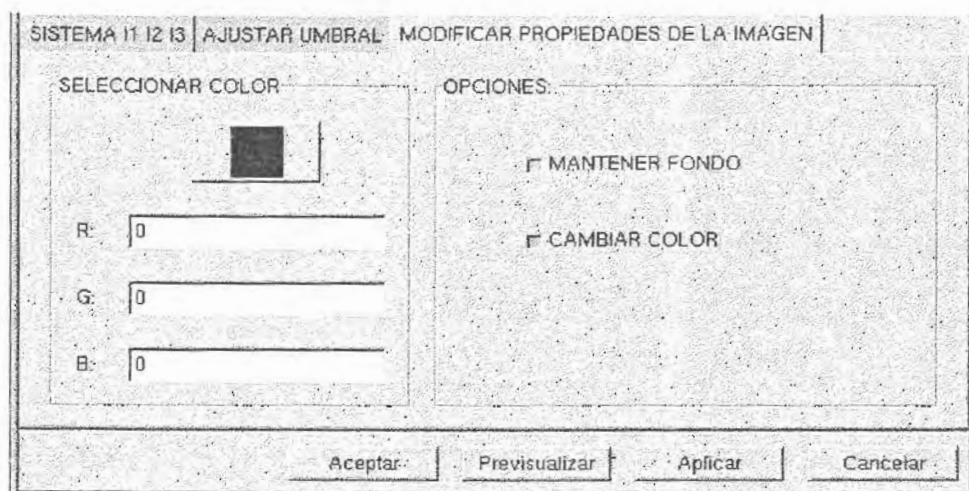
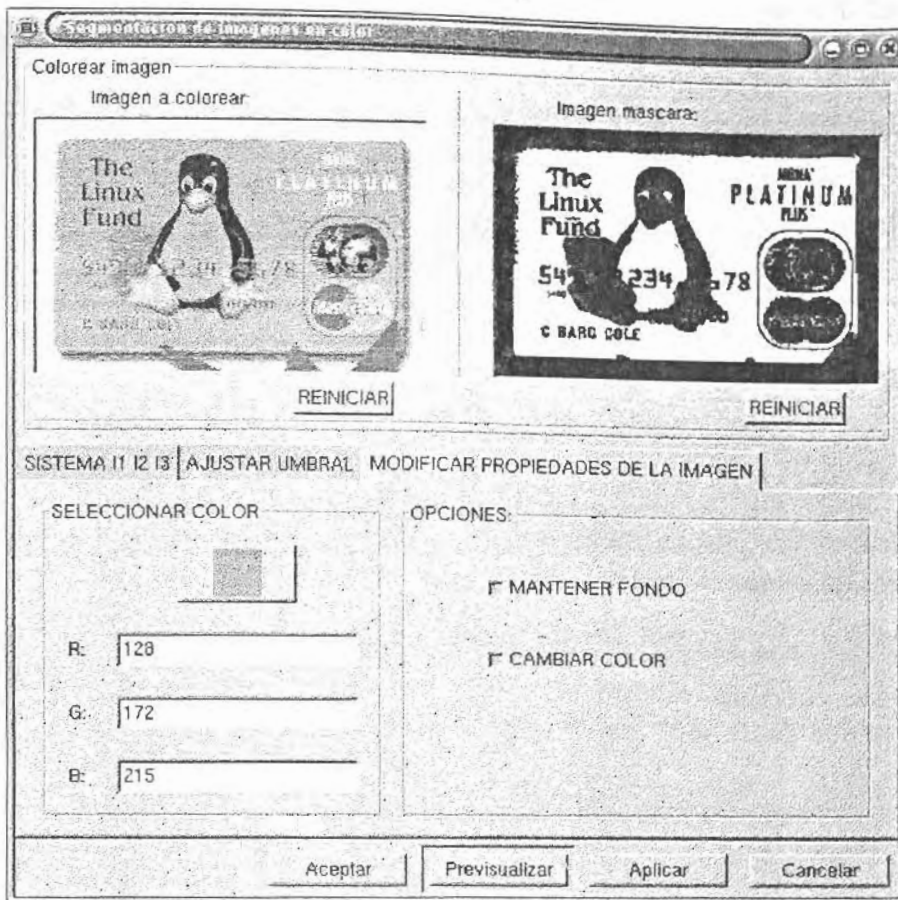


Figura A.5 Carpeta **MODIFICAR PROPIEDADES DE LA IMAGEN**

La sección de la izquierda de la figura A.5 es utilizada para definir el nuevo color que se utilizará para modificar las áreas segmentadas. La selección del color se realiza pulsando el botón de color, que invoca una ventana de diálogo en donde es posible elegir de manera interactiva el color deseado, los componentes RGB de este color se muestran en los objetos de texto correspondientes.

Cuando se han seleccionado las opciones de configuración apropiadas únicamente se presiona el botón con la etiqueta *Previsualizar*, situado en la parte inferior de la pantalla principal. Los cambios ahora se reflejarán en la imagen de color mostrando el resultado de la segmentación, esto se muestra en la figura A.6. Si estos cambios son los correctos únicamente se presiona el botón *Aceptar* para aplicarlos a la imagen principal que fue abierta en el entorno de ejecución del GIMP. El botón *Cancelar* por su parte, termina la ejecución del módulo sin realizar ningún cambio a la imagen principal.



FiguraA.6 Previsualización de los cambios realizados a las áreas segmentadas de la imagen en color.

Finalmente es importante mencionar que todos los cambios aplicados a las imágenes presentes en este modulo se pueden cancelar presionando los botones con la etiqueta *REINICIAR* que se encuentran en la parte inferior de la imagen correspondiente.

Glosario

API: Application Program Interface. Parte del sistema operativo que provee a las aplicaciones una interfaz de uso común o interfaz similar.

Backends: Programa de respaldo (programa de soporte en el sistema de cómputo), programa que efectúa las acciones de fondo.

Cola de impresión: Una cola de archivos en espera para su impresión.

Comando: Órdenes del sistema (ejecutables o scripts).

Controladores: Constituye una pieza de software que controla o administra un dispositivo de hardware, tal como una impresora o un escáner.

CUPS: Common UNIX Printing System. CUPS es una implementación del protocolo de impresión para internet (IPP), este protocolo es un estándar RFC 2567 al estilo HTTP que constituye el sustituto para el protocolo LPD.

Demonio: Proceso que se ejecuta en “el fondo”, realizando una operación específica en tiempos predefinidos o en respuesta a ciertos eventos.

DNS: Domain Name System. Sistema de directorios utilizado en Internet, a partir de un nombre se encuentra su dirección IP.

Frontends: La parte del programa que es responsable de la interfaz de usuario.

FTP: File Transfer Protocol. Protocolo de transferencia de archivos.

GIMP: GNU Image Manipulation Program. Es el programa de manipulación de imágenes de la GNU. GIMP es distribuido libremente bajo los términos de la licencia GPL.

GTK: GIMP Toolkit. Es una librería para crear interfaces gráficas de usuario. GTK es distribuido bajo los términos de la licencia LGPL.

GUI: Graphical User Interface. Es una interfaz basada en elementos gráficos que permite la interacción con programas, seleccionando iconos, botones, y ventanas.

Ingeniería inversa: Se llama ingeniería inversa al intento de descubrir el diseño a partir del comportamiento de un dispositivo.

Kernel: Núcleo o parte esencial de un sistema operativo. Provee los servicios básicos del resto del sistema.

LAN: Local Area Network. Red de Área Local. Una red de área local es un sistema de comunicación de alta velocidad de transmisión. Todas las computadoras, conectadas a una red local, pueden enviar y recibir información. Como su mismo nombre lo indica, una red local es un sistema que cubre distancias cortas.

Librería: Colección de rutinas precompiladas que un programa puede usar.

Licencia: Una licencia es el derecho que tiene su organización de usar o acceder a un producto de software.

Linux: Es un clón del sistema operativo UNIX que corre en múltiples arquitecturas y que soporta un amplio rango de aplicaciones. Constituye en este sentido una implementación versátil de UNIX, distribuida gratuitamente bajo los términos de la Licencia GNU.

LPD: Line Printing Daemon. Demonio de Impresión en Línea. Es el demonio de impresión en línea desarrollado originalmente para Unix BSD.

LPRng: Es una implementación de software para la gestión de colas de impresión mucho más reciente que el sistema LPD, y con características más ventajosas.

MicroKernel: Sostiene que el núcleo debe proporcionar la menor funcionalidad posible y que el grueso de los servicios debe obtenerse a través de los servidores a nivel usuario.

Módulo: Componente de software que se integra en alguna aplicación (por ejemplo GIMP) para añadir nuevas funciones.

Motif: Interfaz gráfica de usuario que realiza funciones de gestión de ventanas y contiene además; un conjunto de herramientas de alto nivel para el desarrollo de programas.

Outsourcing: Recurso externo. Sistema utilizado por empresas grandes de rentar los servicios de compañías chicas para efectuar proyectos en su lugar.

Postscript: Es un lenguaje de descripción de páginas. En este sentido el lenguaje Postscript cuenta con una serie de funciones que le permiten realizar todo tipo de trazos y gráficos.

PPSCSI: Es una arquitectura diseñada para dispositivos que utilizan comandos SCSI a través de un puerto paralelo.

Printtool: Es una herramienta con una interfaz gráfica que facilita y agiliza la configuración de un dispositivo de impresión bajo el sistema operativo Linux. Printtool trabaja en combinación con el sistema de filtros RHS-Printfilters.

Protocolo: Un juego de reglas que gobiernan la forma en que dos aplicaciones se comunican. Para que la comunicación funcione correctamente, los participantes deben ponerse de acuerdo en un conjunto de protocolos específicos, asegurando la compatibilidad entre los sistemas.

Proxy: Servidor Caché. El Proxy es un servidor de que conectado normalmente al servidor de acceso a la WWW de un proveedor de acceso va almacenando toda la información que los usuarios reciben de la WEB, por tanto, si otro usuario accede a través del proxy a un sitio previamente visitado, recibirá la información del servidor proxy en lugar del servidor real.

RFC: Request For Comments. Documentos de libre difusión que sirven como medio de comunicación a la comunidad Internet para el desarrollo de estándares, o para divulgación de información de utilidad.

SANE: Es el acrónimo para “ahora es fácil el acceso a un scanner”. SANE implementa una interfaz para programas de aplicación (API) que provee un “acceso estandarizado” a dispositivos de adquisición digital de imágenes

Script: Conjunto de comandos en un archivo que ordenados producen una salida concreta. Los Scripts no requieren ser compilados, ya que los ejecuta (interpreta) el shell a través de los comandos disponibles.

Servidor: Una computadora o aplicación que provee servicios, archivos o algún recurso central de información a un grupo de múltiples computadoras "clientes" que los solicitan a través de una red.

Shell: Intérprete de ordenes para un sistema operativo.

Socket: Es un mecanismo de comunicación implementado originalmente en la versión BSD del sistema operativo UNIX. Los sockets son utilizados como puertos de comunicación para enviar y recibir datos entre las computadoras.

Software Libre: En general la idea subyacente del software libre es la disponibilidad del código fuente con el propósito de potenciar el estudio, la adaptabilidad, la escalabilidad y la transmisión de los productos de software.

TCP/IP: Protocolo para la transmisión de datos en red y en especial en el Internet.

URL: Uniform Resource Locator. Identificador único para la localización de los recursos de Internet.

Utilerías: Programas dentro de un sistema de cómputo que tienen como objetivo ofrecer un mayor número de funciones básicas de las que contiene el sistema operativo.

Web: Sitio en el World Wide Web, WWW. Conjunto de páginas Web que forman una unidad de presentación, como una revista o libro. Un sitio está formado por una colección de páginas Web.