

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

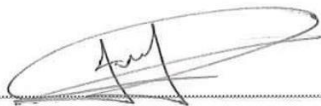
FACULTAD DE CIENCIAS

DESARROLLO DE UNA INTERFAZ GRÁFICA PARA MANEJO DE UNA UNIDAD
DE DISCOS MAGNÉTO-ÓPTICOS

QUE PRESENTA:

CÉSAR ADÁN RAMOS OROZCO

APROBADO POR:



L.C.C. JOSÉ MARTÍN OLGUÍN ESPINOZA
PRESIDENTE



L.C.C. ALBERTO LEOPOLDO MORÁN Y SOLARES
SECRETARIO



L.C.C. EVELIO MARTÍNEZ MARTÍNEZ
1ER VOCAL

AGRADECIMIENTOS

A MIS PADRES:

Por darme apoyo, cariño y estímulo.

A MI ASESOR Y SINODALES:

Por el tiempo dedicado a la revisión y del manuscrito y por su colaboración desinteresada.

A MIS MAESTROS:

Por su apoyo, conocimiento y experiencia durante la carrera.

A TODOS MIS AMIGOS, COMPAÑEROS Y FAMILIARES:

Que de alguna manera colaboraron en la realización y culminación de mis estudios profesionales.

CONTENIDO

1. INTRODUCCIÓN.	1
1.1 LAS INTERFACES DE USUARIO.	1
1.2 TECNOLOGÍAS DE ALMACENAMIENTO.	2
1.3 ANTECEDENTES.	4
2. X WINDOW.	2
2.1 ¿QUÉ ES X WINDOW?	2
2.2 XT INTRINSICS.	6
2.3 HERRAMIENTAS PROPIETARIAS.	7
2.3.1 OPEN LOOK.	7
2.3.2 OSF/MOTIF.	8
2.3.2.1 MOTIF TOOLKIT.	9
2.3.2.1.1 ¿CÓMO TRABAJA TODO ESTO JUNTO?	11
3. UNIDADES MAGNÉTICO-ÓPTICAS.	13
3.1 ¿CÓMO FUNCIONAN?	13
4. INTERFAZ DE USUARIO.	15
4.1 ADOPTAR LA PERSPECTIVA DEL USUARIO.	16
4.2 UTILIZAR METÁFORAS DEL MUNDO REAL.	17
4.3 MANIPULACIÓN DIRECTA.	18

Contenido

4.4 OPTIMIZAR EL TIEMPO DE RESPUESTA.	19
4.5 CONSISTENCIA.	19
4.6 PROCURAR UNA INTERFAZ NATURAL.	21
4.6.1 ASEGURAR UNA NAVEGACIÓN FÁCIL.	21
4.6.2 REDUCIR LOS MOVIMIENTOS DEL RATÓN.	22
4.6.3 PROPORCIONAR FIGURAS Y DIBUJOS NATURALES.	22
4.7 COMUNICACIÓN CON EL USUARIO.	24
4.7.1 PROVEER RETROALIMENTACIÓN AL USUARIO.	24
4.7.2 MENSAJES Y PREVENCIÓN DE ERRORES.	25
4.8 UTILIZAR DESTRUCCIÓN EXPLÍCITA.	25
4.9 DAR CONTROL AL USUARIO.	26
4.10 MANTENER LA FLEXIBILIDAD.	26
 5. MANUALES DE USUARIO.	 28
 5.1 TIPOS DE MANUALES.	 28
 6. INGENIERÍA DE SOFTWARE.	 30
 6.1 DISEÑO ESTRUCTURADO.	 33
6.1.1 LOS DIAGRAMAS DE FLUJO DE DATOS	35
6.1.2 DIAGRAMAS DE ESTRUCTURAS.	37
6.1.3 DICCIONARIOS DE DATOS.	38
6.1.4 REPRESENTACIONES LÓGICAS DE PROCEDIMIENTOS.	38
 7. DESARROLLO.	 40
 7.1 DEFINICIÓN DE REQUERIMIENTOS.	 40
7.2 ESPECIFICACIÓN DE SOFTWARE.	42
7.3 DISEÑO.	52
7.3.1 DIAGRAMAS DE FLUJO DE DATOS.	52
7.3.2 DIAGRAMAS DE ESTRUCTURAS (DE).	57
7.3.3 DICCIONARIO DE DATOS.	62
7.3.4 SEUDOCÓDIGO.	68
 8. PRUEBAS DEL SISTEMA.	 73

Contenido

8.1 DISEÑO DE CASOS DE PRUEBA.	73
8.2 CASOS DE PRUEBA.	78
8.3 RESULTADOS DE LOS CASOS DE PRUEBA.	83
 9. CONCLUSIONES.	 91
9.1 POSIBLES MEJORAS AL PROGRAMA.	91
9.2 CONCLUSIONES GENERALES.	92
 APÉNDICE.	 94
 BIBLIOGRAFÍA:	 113
GLOSARIO	115

1. Introducción.

1.1 Las interfaces de usuario.

El hombre para comunicarse con sus semejantes, utiliza un conjunto de reglas, esperas y respuestas culturalmente definidas que permiten que la comunicación se realice de manera fluida. Pero cuando la comunicación se efectúa entre un hombre y una máquina estas reglas están ausentes (Davis, 1985).

Las primeras personas que usaron la computadora fueron técnicos que comprendían el hardware y el software y captaban con facilidad los procedimientos y mandos crípticos. Sin embargo, a medida que un público más amplio empezó a computarizarse, tropezaba con el lenguaje de cómputo y buscaba accesos menos complicados a los programas. A medida que aumenta la presencia de los sistemas de cómputo debido a los reducidos costos del hardware, cada vez son más las personas que utilizan las computadoras como un instrumento de uso diario. El diseño de interfaz de usuario debe reflejar esto y garantizar que la computadora sea un instrumento útil y poderoso.

Las primeras computadoras no contaban con muchos recursos de hardware y era necesario optimizarlos y los sistemas computacionales eran diseñados de manera que la

comunicación entre el usuario y la computadora fueran fáciles de implementar. Pero hoy en día es diferente, ahora las computadoras cuentan con muchos más recursos que antes, hoy una computadora personal es tan poderosa como lo fue en los 70's una minicomputadora. También nuestra percepción del elemento humano en sistemas basados en computadora ha venido cambiando. Los campos de interfaz de usuario están expandiéndose rápidamente. Esta expansión es originada por tres factores: Expectación de los usuarios, costo del desarrollo de interfaz de usuario, y el surgimiento de nuevas tecnologías de entrada/salida (Bass *et al*, 1991).

1.2 Tecnologías de almacenamiento.

Las computadoras emplean dispositivos de almacenamiento para guardar la información que necesita para realizar sus tareas. Utilizan la memoria principal para guardar datos que van a utilizar de forma inmediata, como programas, resultados intermedios y salidas. Esta memoria esta diseñada para permitir un rápido acceso a la información.

Sin embargo la memoria primaria es costosa y escasa, es necesario tener áreas de almacenamiento adicionales para guardar grandes cantidades de información de manera económica. Además la memoria principal pierde la información cuando la corriente

eléctrica que la alimenta es interrumpida, lo cual la hace impráctica para el almacenamiento permanente de información.

Para guardar información de manera más durable se utiliza la memoria secundaria. Actualmente existen muchas opciones de almacenamiento secundario disponible como por ejemplo los discos flexibles, discos duros, unidades de cintas magnéticas, unidades para cartuchos removibles y los discos ópticos entre otros.

Aunque estos métodos de almacenamiento apoyan en forma efectiva a la mayor parte de las instalaciones de cómputo, hay ocasiones en las que su capacidad de almacenamiento no basta. Por ejemplo cuando se utilizan aplicaciones que requieren grandes cantidades de datos o para almacenar información que consumen mucho espacio de almacenamiento. En estos casos es necesario utilizar algún dispositivo de almacenamiento con una capacidad de cientos de miles de millones de caracteres.

Unos de los dispositivos que ofrece un gran volumen de almacenamiento son las unidades de discos magneto-ópticos. Estos dispositivos usan medio magnético para guardar información bajo control de láser en cartuchos removibles. Un dispositivo de este tipo pone gigabytes de almacenamiento a disposición del usuario.

1.3 Antecedentes.

En el Centro de Cómputo Universitario Unidad Ensenada (CECUUE) de la Universidad Autónoma de Baja California, se cuenta con una unidad de discos magneto-ópticos. Esta unidad es un dispositivo periférico que se conecta a una computadora y que permite almacenar y recuperar información en discos magneto-ópticos.

La compañía fabricante de la unidad de discos proporciona un software para el manejo del dispositivo. Este software consiste en un conjunto de mandos en línea y de bibliotecas para crear programas de control de la unidad. Los mandos en línea no ofrecen una interfaz amigable para el usuario, su aprendizaje implica mayor esfuerzo y el usuario esta propenso a cometer un mayor número de errores.

Esta memoria de servicio social tratan de las actividades llevadas a cabo durante la realización de un sistema computacional para manejar una unidad de discos magnético-ópticos desde estaciones de trabajo bajo el sistema operativo Solaris 2.4.

El objetivo principal que se planteó al iniciar el proyecto fue desarrollar un sistema que proporcione una interfaz gráfica al usuario (GUI) con la cual pueda manipular archivos de imágenes contenidas en discos magneto-ópticos desde una estación de trabajo.

2. X Window.

2.1 ¿Qué es X Window?

X Window surge a partir del proyecto Athena iniciado en 1984 en el Laboratory for Computer Science del Instituto Tecnológico de Massachusetts (MIT) en Cambridge, Massachusetts. La meta del proyecto era lograr que todos los programas estuvieran disponibles interactivamente para los usuarios sobre cualquier estación de trabajo, en cualquier lugar del campus. Tomaron como punto de partida el sistema de ventanas W de la Universidad de Stanford (Reiss *et al*, 1993).

El sistema X Window es una herramienta de software utilizada para el desarrollo de interfaz gráficas de usuario (GUI) sobre estaciones de trabajo. Se basa en la arquitectura cliente/servidor, en la cual los clientes o programas de aplicación, se comunican con los servidores, o unidades de visualización a través de la red. Una parte esencial del sistema X Window es el enlace físico entre el cliente y el servidor. A través de este enlace se realiza la comunicación entre el cliente y el servidor. El protocolo de red se encarga de transportar los mensajes desde el servidor al cliente y viceversa. Estos mensajes deben obedecer el protocolo X. Este protocolo define el número de bytes exactos que se necesitan para realizar todas las operaciones de X Window, incluyendo el dibujo y

movimiento de una ventana o la reacción cuando se pulsa un botón del ratón. El protocolo X es utilizado a través de una biblioteca de funciones que proporcionan unas 300 funciones en lenguaje C llamada Xlib.

2.2 Xt Intrinsics.

Xt Intrinsics se sitúa entre el programador de aplicaciones y las llamadas a las funciones Xlib, permitiendo a los programadores dar ordenes a la computadora con un lenguaje de más alto nivel que las llamadas de bajo nivel de Xlib. Xt Intrinsics define una colección de funciones y estructuras de datos que son estándar para el sistema X Window.

Xt Intrinsics ofrece un marco de trabajo orientado a objetos a partir de componentes de interfaz llamados *widgets*. Xt utiliza el paradigma conocido como encapsulamiento. Como programador, sólo se tiene acceso a funciones que crean, manejan y destruyen *widgets*, a través de ciertas variables públicas de los *widgets* llamadas recursos. Como resultado, la implementación interna de un *widget* puede ser cambiada sin necesidad de cambiar la Interfaz para Programación de Aplicaciones (API). Uno de los beneficios de utilizar este paradigma es que obliga a pensar en una aplicación en un sentido más abstracto y generalizado.

2.3 Herramientas propietarias.

Las casas de software desarrollan sus propias bibliotecas de funciones como extensiones de la biblioteca Xt Intrinsics. Las herramientas en propiedad incluyen características de personalización que prometen atractivas salidas, facilidades de uso, y un rápido desarrollo de la aplicación.

Cada herramienta propietaria produce un modelo de interfaz gráfica de usuario que permiten a los diseñadores crear una serie de aplicaciones con una apariencia estándar. Actualmente, las dos herramientas propietarias más comúnmente usadas son Open Look de Sun Microsystems y AT&T y OSF/MOTIF que pertenece a un consorcio de compañías entre ellas IBM, HP y DEC (Reiss *et al*, 1993).

2.3.1 Open Look.

El Open Look Intrinsics Toolkit (OLIT), basado en Xt Intrinsics, proporciona una interfaz de "*look and feel*", basada en las especificaciones desarrolladas por Sun, AT&T y Xerox. Está limitado a estaciones de trabajo Sun y ya está saliendo del mercado (Johnson *et al*, 1993).

2.3.2 OSF/Motif.

Motif fue desarrollado por la Open Software Foundation (OSF), un consorcio de industrias que incluyen a Digital Equipment Corp. (DEC), Hewlett-Packard (HP), e International Bussines Machines (IBM).

Motif es un conjunto de lineamientos que especifican cómo debe integrarse y presentarse (*look and feel*) una interfaz gráfica de usuario. Este término describe como una aplicación debe verse en la pantalla (*look*) y como el usuario interactúa con ella (*feel*).

La especificación Motif está dividida en dos partes principales (Heller *et al*, 1994):

El modelo de salida. Describe como deben verse los objetos sobre la pantalla. Incluye las forma de los botones, el uso de efectos tridimensionales, el uso de cursores y mapas de bits y la posición de las ventanas. Aunque se dan algunas recomendaciones concernientes al uso de los tipos de letra y otras características del ambiente de trabajo (*desktop*), Motif es flexible en estas recomendaciones.

El modelo de entrada. Especifica como interactúa el usuario con los elementos de la pantalla.

Para crear una interfaz de usuario basadas en el estilo Motif hay dos opciones: Escribir uno mismo la aplicación completa utilizando llamadas directas a X Window, y asegurarse de que todas las características de la interfaz obedecen la especificación Motif, o utilizar una biblioteca de funciones. Una biblioteca de funciones es una colección de funciones previamente escritas que implementan todas las características y especificaciones de un GUI en particular. La biblioteca de funciones que OSF/Motif proporciona es la herramienta Motif Toolkit.

2.3.2.1 Motif Toolkit.

Xt define ciertas clases base de *widgets*, cuya conducta puede ser aumentada o modificada por otras clase de *widgets*. Estas clases base de *widgets* proporcionan un fundamento común para todo el conjunto de *widgets* basados en Xt. Motif Toolkit es una extensión de las bibliotecas Xt, que permite crear interfaces de usuario siguiendo los lineamientos de OSF Motif. Proporciona *widgets* para la creación de elementos comunes de interfaz de usuario tales como etiquetas, botones, menús, cajas de diálogos, barras deslizantes, ventanas para entrada de texto y áreas de despliegue.

Un *widget* opera independientemente de la aplicación, excepto por las interacciones de prearranque que configuran el *widget*. Por ejemplo, un “*widget button*” conoce como

dibujarse a sí mismo, como resaltar cuando el cursor del ratón pasa sobre él, y como responder a un click del ratón.

Para la creación y destrucción de un *widget* se utilizan ciertas funciones que son específicas del *widget*. La configuración de los *widgets* se hace a través de variables públicas llamadas recursos. Estos recursos pueden ser modificados al tiempo de ejecución. Para cada clase de *widget* existen muchos recursos. Algunos de estos recursos proveen un control fino sobre la apariencia tridimensional de los *widgets* de Motif, estos recursos no deben modificarse si se desea que la interfaz sea visualmente consistente con las aplicaciones Motif.

Para manejar la interacción entre los *widgets* y la aplicación se utilizan los recursos “*callback*”. Se utiliza un recurso “*callback*” para cada tipo de interacción que el *widget* soporta. La aplicación asocia una función con un recurso *callback* en que está interesada. La función es invocada cuando el usuario ejecuta ciertas acciones en el *widget*. Por ejemplo: un “*widget PushButton*” proporciona un recurso *callback* para cuando el usuario activa el botón.

Existen varias razones para utilizar Motif toolkit:

- Motif es uno de los principales estándares de interfaces en el mundo Unix.
- Motif proporciona buenos efectos de 3D.
- Usar un *toolkit* ayuda a desarrollar aplicaciones mas rápidamente.
- Motif es convenientemente bueno con los estándares X (*windows manager* y *resource files*).
- La observación de los lineamientos Motif es necesaria para la fácil venta de los productos de grandes firmas y organizaciones (Johnson *et al*, 1993).

2.3.2.1.1 ¿Cómo trabaja todo esto junto?

Una inteface Motif es creada utilizando bibliotecas de Xt Intrinsics y Motif Xm. Xt proporciona funciones para la creación de recursos de configuración de los *widgets* y Xm provee los *widgets* en sí, además de un conjunto de rutinas de utilería y funciones que son apropiadas para la creación de grupos de *widgets* que son usados colectivamente como un simple componente del GUI.

Una aplicación puede necesitar también hacer llamadas a las funciones Xlib. En una aplicación, se pueden hacer llamadas de bajo nivel al sistema operativo, al sistema de

archivos o a un dispositivo específico. De este modo, la aplicación completa puede hacer llamadas a diferentes bibliotecas dentro del sistema.

La aplicación puede interactuar con todas las capas del sistema de ventanas, el sistema operativo, y otras bibliotecas según necesite. Sin embargo el código que implementa la interfaz de usuario debe restringirse a las bibliotecas Motif, Xt, y Xlib tanto como sea posible. Esto es con el fin de facilitar la portabilidad de la interfaz de usuario a través de múltiples plataformas y sistemas operativos.

3. Unidades Magnético-ópticas.

3.1 ¿Cómo funcionan?

Las unidades magneto-ópticas (MO) usan medio magnético para guardar información bajo control de láser. Utilizan cartuchos cubiertos con medio magnético de lectura/escritura. La superficie del disco magnético en un cartucho MO es ligeramente diferente al material magnético convencional. Las MO proporcionan almacenamiento de datos más durable que los discos duros y los discos flexibles convencionales.

Los discos MO usan los cambios de polaridad que suceden en el material magnético cuando es calentado. Cuando el rayo láser calienta el material magnético del disco resulta muy fácil cambiar la polaridad del material. Después de que el material se enfría, la polaridad queda fija en la nueva orientación. Dependiendo de la dirección en la cual es polarizado el material magnético es interpretado como un 1 o un 0 lógico.

Una unidad MO usa un cartucho removible que es escrito un lado a la vez. Para acceder información del otro lado del disco se tiene que quitar el cartucho, darle la vuelta y volverlo a poner en la unidad (Norton, 1994).

Las unidades MO se usan principalmente para respaldo de información en ambientes de redes.

4. Interfaz de usuario.

El proceso de diseñar y construir interfaces de usuario es crítico para la realización de un sistema que satisfaga las necesidades actuales y futuras del usuario. Un sistema con una interfaz mal diseñada puede tener consecuencias desagradables. En el mejor de los casos una interfaz inadecuada produce un costo de adiestramiento elevado y un mayor número de errores, y en el peor de ellos, resultará rechazada y el sistema es abandonado por completo. Se han identificado tres respuestas características que originan un diseño de interfaz deficiente (Sommerville, 1988).

- El usuario se rehusa a utilizar el sistema de cómputo. Después de la prueba inicial, lo juzga inadecuado para sus necesidades.
- El usuario aprende “mecánicamente” unos cuantos mandos y se limita a ellos en el uso del sistema. Puesto que el usuario no comprende realmente el sistema, no puede aprovechar todas sus características.
- El usuario se relaciona con el sistema a través de un intermediario que se convierte en el experto local del sistema.

Para ser efectiva, una interfaz debe permitir al usuario interactuar con la aplicación de una manera simple y natural. El usuario ante todo quiere realizar su trabajo. Una interfaz de usuario necesita ser diseñada de manera que el usuario pueda realizar su trabajo de manera sencilla y rápida. El usuario desea adquirir dominio de la aplicación. Lograr que el usuario adquiera dominio del sistema de manera sencilla es básico para una aplicación. Para lograr esto se deben aplicar algunos principios de interfaz de usuario. No siempre se es capaz de aplicar todos estos principios ya sea debido a la naturaleza particular de la aplicación, sus componentes, el manejador de ventanas del sistema, o en respuesta a las necesidades del cliente. A continuación se mencionan y se discuten estos principios.

4.1 Adoptar la perspectiva del usuario.

Un diseño efectivo empieza por adoptar el punto de vista del usuario. El diseñador del sistema, tiende a ver la aplicación como la implementación de funciones. En contraste, el usuario ve una aplicación en términos de su interfaz. Un buen diseño de interfaz resuelve los problemas del usuario, haciendo su trabajo más fácil y ofreciendo nuevas capacidades. Las dos maneras más efectivas para entender el trabajo del usuario son: involucrarlo en el diseño, o ser usuario uno mismo.

Si el sistema es construido para reemplazar existentes métodos manuales o sistemas, Bass y Coutaz (1991) recomiendan tener contacto directo con los usuarios potenciales de la aplicación. Esto incluye entrevistas, preguntas, observaciones del usuario en su trabajo, y sesiones de “pensar en voz alta”. Las sesiones de “pensar en voz alta” es una técnica en donde el operador explica sus acciones a tiempo que las va ejecutando (Bass *et al*, 1991).

Cuando la tarea a realizar es nueva, no siempre es posible contactar usuarios potenciales. En tales situaciones, los operadores pueden ser modelados basándose en una clasificación general, tal como la propuesta por Shneiderman (1992) en la que clasifica a los usuarios en tres categorías: usuario novato, usuario con conocimientos intermedios, y usuario frecuente.

4.2 Utilizar Metáforas del mundo real.

Utilizar metáforas del mundo real. Las metáforas son usadas para describir un nuevo modelo en términos de un modelo conocido. Una buena interfaz de usuario permite transferir habilidades de experiencias del mundo real. Cuando se diseña un nuevo componente, se debe considerar como es utilizado un control similar del mundo real, para así poder incorporar la metáfora en el nuevo componente. Esto ayuda a hacer obvio el uso

del nuevo componente porque ayuda al usuario a relacionar el componente de la interfaz con los objetos o conceptos del mundo real (Heller *et al*, 1994).

Actualmente existen dos tipos principales de metáforas para interacción; el modelo de metáforas del mundo real y las metáforas de conversación. Las metáforas del mundo real imitan electrónicamente objetos del mundo real. Un ejemplo es la metáfora del escritorio en el que los iconos representan *folders* y documentos y el ratón es la extensión electrónica de la mano del usuario. Las metáforas de conversación están basadas sobre la descripción lingüística de las acciones a ser ejecutadas sobre un sistema de objetos. Los ejemplos de metáforas de conversación incluyen mandos de texto tales como el *Shell* de Unix (Bass *et al*, 1991).

4.3 Manipulación directa.

El usuario debe ser capaz de manipular directamente elementos y funciones de la interfaz de usuario. La manipulación directa reduce la cantidad de información que el usuario necesita memorizar. También se deben soportar métodos de interacción con la aplicación utilizando únicamente el teclado. Estos métodos pueden ser usados por usuarios avanzados para ejecutar algunas tareas más rápidamente.

4.4 Optimizar el tiempo de respuesta.

El tiempo de respuesta de un sistema interactivo es quizás el factor de mayor influencia en la opinión del usuario. Shneiderman (1992) lo define como el número de segundos que pasan desde que el usuario inicia una actividad (usualmente presionando la tecla enter o un botón del ratón) hasta que la computadora empieza a presentar los resultados en la pantalla o impresora.

No es posible proporcionar un tiempo de respuesta aceptable estándar porque las expectativas del usuario varían según la complejidad de la operación que se va a ejecutar. Sin embargo, es necesario implementar operaciones que puedan significar grandes intervalos de tiempo de respuesta, se debe dar al usuario información de como avanza la tarea. Esto tiene más importancia cuando los tiempos de respuesta para la misma operación son variables, porque dependen por ejemplo de la carga del sistema. Si no existe una indicación de cuanto tardará la tarea, el usuario no puede utilizar con facilidad el tiempo de espera hasta que el sistema responde.

4.5 Consistencia.

La consistencia ayuda al usuario a utilizar habilidades familiares a nuevas situaciones. Una interfaz consistente implica que cuando un usuario consume tiempo para

aprender acerca de un mando de la interfaz, ese conocimiento se puede aplicar a los demás mandos del sistema (Sommerville, 1988). En un sistema bien diseñado, cuando el usuario se confronta con una situación nueva, el mecanismo de retroalimentación del usuario dice “ esta tarea es parecida a esta otra tarea que yo ya realicé antes” y el usuario entonces es capaz de transferir el aprendizaje obtenido durante la realización de la tarea conocida en esta nueva situación (Baecker *et al*, 1987).

Cuando unos componentes trabajan de manera consistente con los otros, el usuario va perdiendo el miedo a intentar realizar nuevas funciones, las cuales le parecen familiares, confortables, y apropiadas. Se debe mantener la consistencia dentro de un sistema y entre los sistemas que se ejecutan en la misma máquina. Para guardar la consistencia se pueden seguir las siguientes guías:

- Los componentes similares operan de manera similar y tienen usos similares.
- Las mismas acciones deben tener siempre los mismos resultados.
- La posición del cursor del ratón no debe cambiarse arbitrariamente.

Para guardar la consistencia entre diferentes aplicaciones existen estas reglas:

- Los componentes deben tener una apariencia similar.
- La interacción con el usuario debe ser de manera familiar.
- Los componentes de la interfaz deben organizarse de manera familiar.

4.6 Procurar una interfaz natural.

Las tareas que realiza el usuario deben fluir de manera natural. El usuario debe ser capaz de anticipar el progreso natural de cada tarea. Por medio de esta anticipación, él es capaz de completar las tareas más rápidamente. Cada objeto de la pantalla necesita tener una apariencia distinta de manera que el usuario pueda reconocerlo fácilmente y entenderlo rápidamente. Al mismo tiempo, el estilo de la interfaz debe unificar estos elementos y asegurar una apariencia atractiva y consistente sobre cualquier resolución de la pantalla.

4.6.1 Asegurar una navegación fácil.

La navegación rápida y sencilla dentro del área de trabajo del sistema, da al usuario un sentido de dominio de la aplicación y de su trabajo.

Los elementos de la interfaz debe ser arreglados de acuerdo a su uso. Un arreglo óptimo de estos elementos ayuda al usuario en el proceso de toma de las decisiones y reduce la posibilidad de error.

4.6.2 Reducir los movimientos del ratón.

Reducir el número de movimientos del ratón para simplificar las acciones del usuario. Esto ayuda a lograr una interfaz natural porque, desde el punto de vista del usuario, su trabajo involucra una serie de pensamientos, intenciones, y tareas (algunos ya predefinidos y otros que van apareciendo durante el proceso) donde todas se relacionan para decidir un resultado o acción. Esta serie de pensamientos es interrumpida cuando el usuario hace movimientos innecesarios del ratón.

4.6.3 Proporcionar figuras y dibujos naturales.

Minimizar el contraste entre los objetos de la pantalla en orden para dirigir la atención del usuario. El uso apropiado del contraste ayuda al usuario a distinguir objetos que se encuentran en el segundo plano de una ventana. Objetos muy oscuros sobre una pantalla brillante, objetos muy brillantes sobre una pantalla muy oscura, y los colores

brillantes llaman la atención del usuario. Shneiderman (1992) ofrece algunas líneas guías para el uso del color.

- Usar el color de manera conservadora.
- Limitar el número de los colores usados.
- Reconocer el poder del color como una técnica de codificación.
- Se debe ser consistente en la codificación del color.
- Debe estar atento a las expectativas comunes de los usuario acerca de la codificación del color.
- La codificación del color debe estar bajo control del usuario.
- Utilizar los cambios de colores para indicar los cambios de estado.
- Considerar el desplegado en monitores monocromáticos al elegir los colores.

Utilice los colores como un aspecto redundante de la interfaz que es usado para proveer diferenciación adicional entre varios objetos de la pantalla. La diferenciación también se obtiene de las figuras y el tamaño de los objetos. Por ejemplo, en muchas

partes del mundo, el símbolo de alto es un octágono rojo. La gente reconoce el símbolo de alto por la figura y el color.

4.7 Comunicación con el usuario.

Comunicar al usuario las acciones de la aplicación. Una aplicación debe permitir al usuario saber que ocurre con la aplicación pero sin revelar detalles de implementación. La comunicación conveniente entre el usuario y la aplicación incrementa la satisfacción del mismo. Existen tres líneas de comunicación: Proveer retroalimentación al usuario, prevención de errores, y mensajes.

4.7.1 Proveer retroalimentación al usuario.

La retroalimentación le permite al usuario saber que la computadora ha recibido su entrada. Dar retroalimentación al usuario siempre que éste selecciona un componente o elemento de un menú, remarcando este componente o elemento de alguna manera. Además, si ciertas operaciones toman más de algunos segundos en realizarse, se debe permitir al usuario saber que la computadora está trabajando en la operación desplegando un mensaje o cambiando el apuntador usual por un apuntador de trabajo como un reloj por ejemplo.

4.7.2 Mensajes y prevención de errores.

Anticipar los errores que podrían ocurrir. Una manera de reducir la pérdida en productividad debido a los errores es mejorar los mensajes de error que la computadora provee. Estos deben ser específicos, escritos en tono positivo y ser constructivos (Shneiderman, 1992).

Muchos usuarios se sienten más a gusto en el aprendizaje de como usar un software de aplicación cuando utilizan un método de intento y error. El soporte de una función “deshacer” (*undo*) en el proceso de aprendizaje minimiza el costo de un error. Una función “deshacer” le permite al usuario retractar la acción previa, y fomenta un espíritu de exploración y experimentación que son esenciales.

4.8 Utilizar destrucción explícita.

La destrucción explícita significa que, cuando un acción tiene consecuencias negativas irreversibles, ésta debe requerir que el usuario exprese explícitamente su deseo de llevarla a cabo.

Los avisos previenen al usuario de operaciones destructivas inadvertidas, permitiendo a este conservar el control de la aplicación. Los avisos también alientan al

usuario a experimentar sin miedo a pérdidas. Las operaciones que pueden causar una pérdida de datos seria e irrecuperable deben avisar al usuario de las consecuencias y solicitar una confirmación explícita.

4.9 Dar control al usuario.

Conforme los usuarios adquieren experiencia en el uso de la computadora su deseo de controlarla aumenta. No les satisface ser pasivos y aceptar a la computadora como el compañero predominante de la relación (Sommerville, 1988). Es necesario pues, proporcionarles una interfaz que les permita lograr que la computadora sea una herramienta que pueden controlar. El usuario puede obtener el control cuando una aplicación es flexible, y utiliza los descubrimientos progresivos.

4.10 Mantener la flexibilidad.

Proporcionar al usuario múltiples caminos de acceso a las funciones de la aplicación y llevar a cabo sus tareas incrementa su sentido de control. La flexibilidad habilita al usuario a seleccionar el mejor método para acceder a una función basado en el criterio que el elija, como nivel de experiencia, preferencia personal, situación particular o por simple hábito.

Una aplicación también debe ser configurable. Permitir al usuario configurar y seleccionar preferencias personales aumenta su sentido de control.

Se debe diseñar la aplicación para que las funciones más comunes y necesarias sean presentadas primeramente y en un orden lógico. Y hacer las funciones menos frecuentes más sofisticadas y ocultas a la vista inmediata, pero fácilmente disponibles.

5. Manuales de usuario.

5.1 Tipos de manuales.

Todos los usuarios de sistemas computacionales interactivos requieren alguna capacitación. Muchos usuarios pueden aprender de otras personas que conocen el sistema, pero los manuales de capacitación muchas veces son necesarios.

Los manuales impresos son el tipo más tradicional de manuales de usuario. Estos son documentos que describen las características del sistema. Existen muchas variaciones de estos manuales:

- Listado alfabético y descripción de los mandos.
- Referencia rápida con una presentación concisa de la sintaxis.
- Como iniciar, notas que permiten a los usuarios ansiosos probar algunas características del sistema.
- Tutoriales para los usuarios principiantes.

- Manual de referencia que cubre todas las características del sistema.

Otro tipo de manuales son los que no están impresos y se llaman manuales en línea.

También existen diferentes tipos de manuales en línea (Shneiderman, 1992):

- Manual de usuario en línea: Es una versión electrónica del manual de usuario tradicional. La simple conversión a un documento electrónico puede hacer el texto más fácilmente disponible.
- Ayuda en línea donde el usuario selecciona u oprime una tecla y entonces se presenta en pantalla un texto explicativo del mando.
- Tutorial en línea el cual utiliza un medio electrónico para enseñar al principiante a utilizar un sistema. Se puede valer para esto de simulaciones del sistema trabajando, del desplegado de animaciones, y de sesiones interactivas con el usuario.
- Demostraciones en línea para que los usuarios potenciales que desean un panorama general del software puede beneficiarse de una demostración en línea que los lleve a través de una sesión de uso del software.

6. Ingeniería de software.

Con la aparición de computadoras de la tercera generación se crearon máquinas con un poder de cómputo que aventajaba de manera considerable a sus antecesoras de la segunda generación. Esto hizo posible el desarrollo de sistemas computacionales mucho más grandes y complejos.

La construcción de estos primeros grandes sistemas mostraron lo inadecuado de las metodologías de desarrollo existentes hasta entonces. No era lo mismo la construcción de software relativamente pequeño, al desarrollo de grandes sistemas donde el número de detalles de cada aspecto del software resultaban difíciles de recordar para cualquier individuo. No podía hacerse una simple ampliación a escala de las técnicas aplicables a los sistemas pequeños. Muchos proyectos se entregaron tarde, muchos sobrepasaron el presupuesto inicial, resultaron poco confiables y eficientes. Incluso varios proyectos iniciados nunca fueron terminados. Había una gran necesidad de técnicas y metodologías nuevas que permitieran controlar la complejidad inherente a los grandes sistemas de software.

En 1968 se llevó a cabo una reunión de trabajo en Alemania Oriental para analizar estos problemas a los que llamaron “la crisis de software”. En esta reunión se utilizó por primera vez el término “Ingeniería de Software” como una disciplina encaminada a mejorar la construcción de sistemas computacionales.

La ingeniería de software es la disciplina tecnológica y administrativa dedicada a la producción sistemática de productos de programación, que son desarrollados y modificados a tiempo y dentro de un presupuesto definido (Fairley, 1988).

Durante la década de los 70's se dieron grandes avances en la ingeniería de software: se desarrollaron técnicas de análisis para determinar los requisitos; se crearon nuevas metodologías para el desarrollo de programas; desde entonces a la fecha han nacido nuevas técnicas de instrumentación, así como nuevos lenguajes de programación; se han utilizado técnicas para la validación de programas y, se han mejorado los procedimientos de mantenimiento de los programas.

Además las técnicas de administración se han adaptado para la ingeniería de software. Se han explorado nuevas técnicas de dinámica de grupos y comunicación de proyectos.

También han evolucionado las técnicas cuantitativas para la estimación de costos y la confiabilidad de un producto, se han descubierto los principios fundamentales de análisis, diseño, instrumentación y pruebas. Se han desarrollado herramientas automáticas de desarrollo para mejorar la calidad de los programas, la productividad de los programadores y el control y administración de proyectos(Fairley, 1988).

Actualmente (1997) existen varias metodologías para el desarrollo de sistemas, están por ejemplo el ciclo de vida del software, el diseño orientado al flujo de datos, el diseño orientado a objetos y el diseño estructurado.

Para el desarrollo de un producto de software la ingeniería han identificado en general seis fases: La definición de requisitos, la especificación de programas, el diseño, la implementación, las pruebas y el mantenimiento (Bass *et al*, 1991).

La definición de requisitos: cubre la elaboración de un documento que contiene una descripción del problema que se solucionará y de las restricciones que existen en su resolución. También se debe incluir una descripción-de la situación actual y de las metas que debe lograr el nuevo sistema.

La especificación de software: durante esta etapa se analiza la definición de requisitos y se diseñan los componentes de software para proporcionar los servicios al usuario.

Diseño: es el proceso mediante el cual se traducen la especificación de software en una representación del software. Inicialmente, la representación describe una visión general del sistema. Refinamientos posteriores conducen a una representación de diseño que se acerca mucho al código fuente.

Implementación: es la expresión del diseño en términos de un conjunto de lenguajes de programación y herramientas.

Pruebas: es el proceso de establecer la existencia de errores en el programa.

Mantenimiento: esencialmente involucra hacer cambios y arreglar los efectos colaterales.

6.1 Diseño estructurado.

El enfoque básico en el diseño estructurado es la conversión sistemática de los diagramas de flujos de datos en cartas de estructuras. El desarrollo y descripción de un diseño de software estructurado incluye varias etapas (Sommerville *et al*, 1988):

1. Estudiar y comprender el problema. Sin esta comprensión, es imposible el diseño efectivo del software.
2. Identificar las características generales de, al menos, una posible solución. En esta etapa suele ser útil identificar varias soluciones y evaluar cada una de ellas. Se debe elegir la solución más simple posible. Es de particular importancia no considerar detalles de aplicación de bajo nivel, de los cuales tenga conocimiento el diseñador, que interfieran en la elección de una solución.
3. Construir un diagrama de flujo de datos que muestre transformaciones generales de los datos del sistema. Si esto parece imposible, quizá no se haya comprendido de manera adecuada el problema.
4. Mediante el diagrama de flujo de datos, construir un diagrama de estructura que muestre las unidades de programa relacionadas con la solución.
5. Describir cada abstracción utilizada en la solución mediante un lenguaje de descripción. Es probable que en las primeras etapas del diseño, esto consista exclusivamente en una descripción en lenguaje natural.

Dos versiones similares del Análisis Estructurado de Sistemas (SSA) han sido descritas por Gane y Sarson y por DeMarco (1988).

SSA utiliza un lenguaje gráfico para construir los modelos de los sistemas. Hay cuatro elementos básicos en SSA: diagramas de flujo de datos, diagrama de estructuras, diccionarios de datos y representaciones lógicas de procedimientos.

6.1.1 Los diagramas de flujo de datos

Los diagramas de flujos de datos (DFD) son diagramas que se utilizan para describir un diseño de sistemas de alto nivel; muestran como se transforman los datos al pasar de un componente a otro (Sommerville, 1988).

En un DFD se utilizan cuatro elementos principales; las entidades, los procesos, el almacenamiento de datos y los flujos de datos.

Las entidades externas son usualmente cosas o personas que representan una fuente o destino de transacciones, por ejemplo: clientes, empleados, aviones etc. Una entidad externa puede ser simbolizada por una cuadro sólido. Con los lados izquierdo y superior remarcados para hacer notar que símbolo va fuera del diagrama (Gane *et al*, 1979).

Los procesos son elementos donde se realiza alguna transformación de datos. Pueden considerarse como una caja negra independiente que transforma sus entradas en salidas. Para describir la función de cada proceso y, para una fácil referencia, se da a cada proceso un identificador único. Los procesos pueden ser representados por un rectángulo con las esquinas redondeadas, opcionalmente puede ser dividida en tres áreas (Gane *et al*, 1979).

La descripción del proceso debe ser una sentencia imperativa, idealmente un verbo activo (extraer, calcular, verificar) seguida por una clase de objeto, entre más simple mejor (extraer cuenta, calcular saldo etc.) (Gane *et al*, 1979).

El almacenamiento de datos es una estructura conceptual; durante el diseño uno puede identificar la necesidad de definir datos así como lugares donde guardarlos. Los almacenes de datos pueden ser simbolizados por un par de líneas horizontales paralelas, cerradas por un lado. Cada almacén puede ser identificado por una "D" y un número arbitrario. El nombre debe ser elegido de manera que sea descriptivo (Gane *et al*, 1979).

Los flujos de datos son representados por una flecha, preferentemente horizontal y/o vertical, con una punta de flecha mostrando la dirección del flujo. Por claridad y especialmente para facilitar el dibujo de los diagramas, una flecha con doble punta puede ser usada en lugar de dos flechas (Gane *et al*, 1979).

6.1.2 Diagramas de estructuras.

Los diagramas de estructuras (DE) son gráficas de jerarquía que muestran la relación estructural de los componentes de un sistema de software (Sommerville, 1988).

Los DE describen el sistema como una jerarquía de partes y lo muestran gráficamente como un árbol. Estos diagramas documentan como se pueden aplicar los elementos de un diagrama de flujo de datos como una jerarquía de unidades de programa. Un DE muestra las relaciones entre las unidades de programa sin incluir ninguna información acerca del orden de activación de esas unidades. Se traza mediante tres símbolos:

1. Un rectángulo con el nombre de la unidad.
2. Una flecha que conecta los rectángulos.
3. Una flecha con un círculo, con el nombre de los datos que se pasan entre los elementos del diagrama de estructura. Las flechas con círculos suelen dibujarse paralelas a las flechas que conectan los rectángulos del diagrama (Sommerville, 1988).

6.1.3 Diccionarios de datos.

El diccionario de datos es un documento que proporciona detalles de todas y cada una de las entidades relevantes para el sistema que se describe. Las entidades que se incluyen en el diccionario de datos comprenden elementos que se mencionan en el sistema que se documenta, como procedimientos, registros, archivos, módulos, etc., y también las entidades del sistema del mundo real modelados por el programa (Sommerville, 1988).

La definición de un elemento en el diccionario de datos incluye el nombre del dato, y atributos tales como el diagrama donde se utiliza, su propósito, procedencia, subunidades, y cualquier nota aclaratoria necesaria (Fairley, 1988).

6.1.4 Representaciones lógicas de procedimientos.

Las representaciones lógicas del procedimiento se utilizan para especificar la secuencia de procesos en términos entendibles para el usuario y el diseñador, para esto se utilizan tablas de decisión y el pseudocódigo (Fairley, 1988).

Con el uso del pseudocódigo el diseñador describe las características del sistema usando frases cortas y precisas en español, las cuales se encuentran estructuradas por

medio del uso de palabras claves como si-entonces-sino, mientras-repetir y fin. Con estas palabras claves y el uso de la indentación se puede describir el flujo de control del programa, mientras que cada frase en español describe las acciones a ejecutar. Usando la estrategia jerárquica hacia abajo, cada frase en español puede ser expandida en un pseudocódigo más detallado en el nivel inferior, hasta que la definición de las especificaciones llegan al nivel del lenguaje de instrumentación (Fairley, 1988).

7. Desarrollo.

7.1 Definición de requerimientos.

7.1.1. Descripción funcional.

Objetivo general.

Desarrollar un sistema que proporcione una interfaz gráfica de usuario a través de la cual sea posible manipular archivos contenidos en discos magnético-ópticos desde las estaciones de trabajo.

Objetivos específicos:

- a) Sea capaz de leer y presentar en pantalla el directorio de archivos de cualquier disco magnético-óptico contenido en la unidad.
- b) Cambiar cualquier disco contenido en la unidad magneto-óptica.
- c) Permitir al usuario navegar entre los diferentes directorios.
- d) Copiar y borrar cualquier archivo en el directorio de trabajo del usuario.
- e) Renombrar archivos del directorio de trabajo del usuario.

- f) Crear y borrar directorios en el directorio de trabajo del usuario.
- g) Ordenar el despliegue en la pantalla de los archivos según su tamaño, fecha, o nombre.
- h) Visualizar en la pantalla información de los archivos tal como su tamaño, fecha y/o atributos.

7.1.2. Entradas y salidas.

A continuación se listan las entradas y las salidas del sistema.

Entradas:

- 7.1.2.1 Número del disco que se desea leer.
- 7.1.2.2 Nombre del archivo que se desea leer.
- 7.1.2.3 Operación que desea realizar.

Salidas:

- 7.1.2.4 Información del archivo.
- 7.1.2.5 Información del disco.
- 7.1.2.6 Resultado de la operación realizada.
- 7.1.2.7 Archivo transferido.

7.1.3. Contexto de uso de la aplicación.

- El sistema debe implementarse de tal forma que pueda ser utilizado a través de una red utilizando los servicios TCP/IP.
- Los usuarios podrán utilizar el sistema desde estaciones de trabajo con algún sistema operativo Unix y con el sistema X Window.

7.1.4. Restricciones de acceso y seguridad.

- El sistema deberá contar con un mecanismo de seguridad que permita el acceso solo a los usuarios autorizados.
- El contenido de los discos magneto-ópticos no podrán ser modificados por los usuarios del sistema.
- Los usuarios tampoco podrán insertar o retirar discos del dispositivo.
- Sólo un usuario a la vez podrá utilizar la unidad de discos magneto-ópticos

7.2 Especificación de software.

Especificación de software

El presente documento es una especificación de las características que el sistema debe tener y los servicios que debe proporcionar al usuario.

La primera parte del documento muestra la especificación de la interfaz de usuario. Se muestran todos sus elementos con una breve enunciado que indica su propósito.

La segunda sección hace referencia a la especificación de requerimientos y se muestra en una tabla una asociación de cada requerimiento específico con los elementos de la interfaz que dará el servicio.

En la última parte se muestra un diagrama de alto nivel del sistema y se asocia con las entradas y salidas mencionadas en la especificación de requerimientos.

I. Interfaz de usuario.

La interfaz con la que el usuario interactúa con la aplicación es la mostrada en la figura 1. Esta es una interfaz basada en ventanas, con la cual puede comunicarse a través del ratón o del teclado. A continuación se listan sus elementos:

1. *Textedit* para escribir el nombre del archivo del disco magnético-óptico.
2. *List* para desplegar la lista de archivos del disco magnético-óptico.

3. *Textedit* para escribir el directorio del disco magnetico-óptico.
4. *List* para desplegar los directorios del disco magnetico-óptico.
5. *OptionMenu* para seleccionar el disco.
6. *Textedit* para escribir el nombre del archivo del usuario.
7. *List* para desplegar la lista de archivos del usuario.
8. *Textedit* para escribir el directorio del usuario.
9. *List* para desplegar la lista de directorios del usuario.
10. Menú Archivos.
11. Menú Visualizar.
12. Menú Ayuda.
13. Botón para copiar archivos.
14. Botón para borrar archivos.
15. Botón para renombrar archivos.
16. Botón para crear directorios.
17. Botón para borrar directorios.

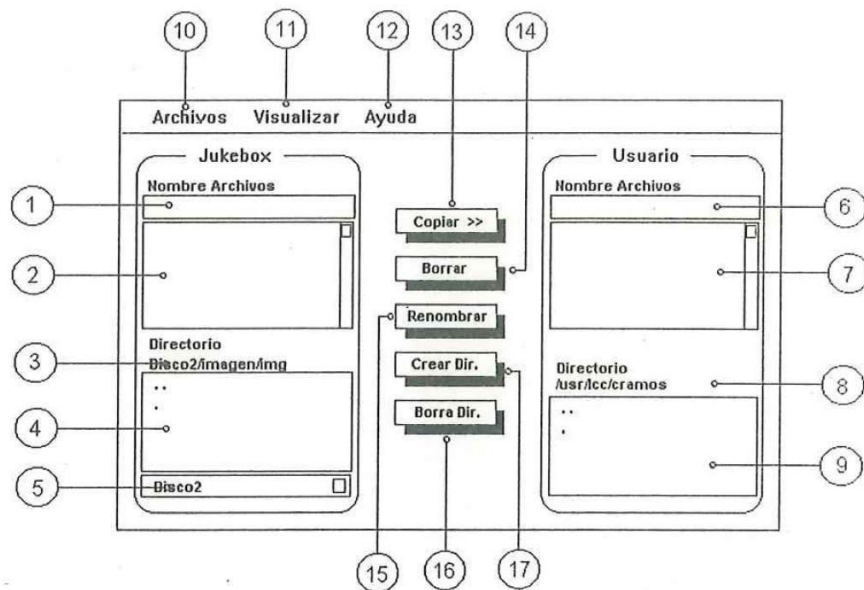


Figura 1. Interfaz de usuario.

Los menús Archivos, Visualizar y Ayuda son mostrados en las Figuras 2, 3 y 4, respectivamente. Las opciones de los menús se listan a continuación:

Menú Archivo:

18. Copiar Archivo.

19. Borrar Archivo.

20. Renombrar Archivo.

21. Crear Directorio.

22. Borrar Directorio.

23. Actualizar Directorio.

24. Salir.

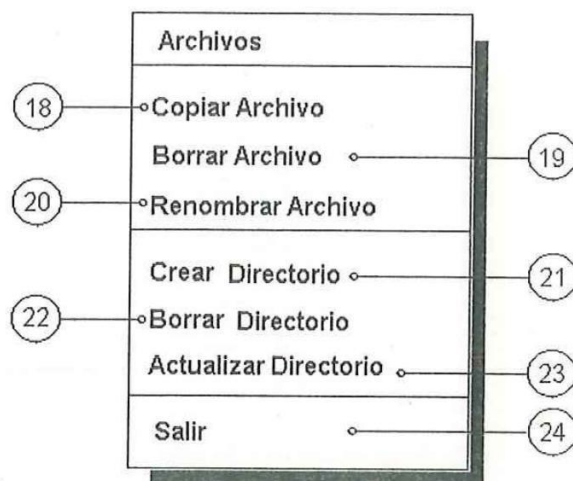


Figura 2. Menú Archivos.

Menú Visualizar:

25. Jukebox Tamaño.

26. Jukebox Fecha.

27. Jukebox Atributos.

28. Usuario Tamaño.

29. Usuario Fecha.

30. Usuario Atributos.

31. Ordenar Jukebox: Nombre, Tamaño, Fecha.

32. Ordenar Usuario: Nombre, Tamaño, Fecha.

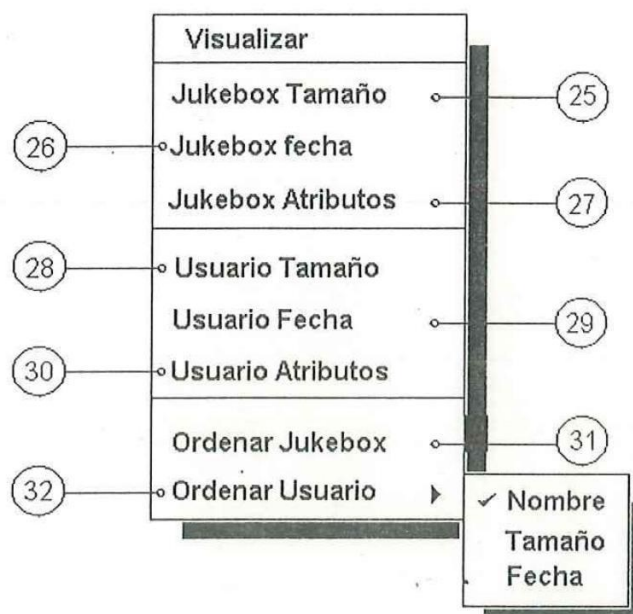


Figura 3. Menú Visualizar.

Menú Ayuda:

33. Contenido.

34. Acerca de



Figura 4. Menú Ayuda

III. Especificación funcional.

En la Tabla I se muestra una relación de cada uno de los requisitos funcionales del sistema con un elemento de la interfaz de usuario.

Tabla I. Relación entre los requerimientos y la especificación de la interfaz.

Requerimiento	Especificación de la interfaz
a) Sea capaz de leer y presentar en pantalla el directorio de archivos que contenga imágenes de cualquier disco magnetico-óptico contenido en la unidad.	Elemento de interfaz (2)
b) Cambiar cualquier disco contenido en la unidad.	Elemento de interfaz (5)
c) Permitir al usuario navegar entre los diferentes directorios.	Elemento de interfaz (4), (3), (9), (8)
d) Copiar y borrar cualquier archivo de imagen en el directorio de trabajo del usuario.	Elemento de interfaz (13), (18), (14), (19)
e) Renombrar archivos del directorio de trabajo del usuario.	Elemento de interfaz (15), (20)
f) Crear y borrar directorios en el directorio de trabajo del usuario.	Elemento de interfaz (17), (21), (16), (22)
g) Ordenar el despliegue en la pantalla de los archivos según su tamaño, fecha o nombre.	Elemento de interfaz (31), (32)
h) Visualizar en la pantalla información de los archivos tal como su tamaño, fecha y/o atributos.	Elemento de interfaz (25), (26), (27), (28), (29), (30)

IV. Entradas y salidas.

En la Figura 5 se muestra un diagrama de flujo de datos del sistema. El proceso 1.1 representa la interfaz de usuario mostrada en la Figura 1. Por medio de ella el usuario interactúa con la aplicación. El proceso 1.2 representa el software que realizará las tareas solicitadas por el usuario como copiar y borrar archivos, crear directorios etc. El cuadro Jukebox representa al dispositivo magneto-óptico el cual atenderá las solicitudes de lectura de los discos. El cuadro Sistema Operativo representa al sistema operativo que atenderá solicitudes de creación de archivos y directorios entre otras. El cuadro Usuario representa al usuario del sistema.

Los datos que fluyen desde el usuario al proceso 1.1 representan las entradas mencionadas en la especificación de requerimientos (7.1.2.1, 7.1.2.2, 7.1.2.3). Las salidas (7.1.2.4, 7.1.2.5, 7.1.2.6, 7.1.2.7) son los datos que van del proceso 1.1 al usuario, del proceso 1.2 al Jukebox y del proceso 1.2 al sistema operativo. Estas relaciones de entrada y salida con los flujos de datos son mostradas en la tabla II y III.

Tabla II. Entradas

Entradas	Flujo de datos
7.1.2.1 Número del disco que se desea leer.	Solicitudes del usuario
7.1.2.2 Nombre del archivo que se desea leer.	
7.1.2.3 Operación que desea realizar.	

Tabla III. Salidas

Salidas	Flujo de datos
7.1.2.4 Información del archivo.	Respuestas al usuario
7.1.2.5 Información del disco	Respuestas al usuario
7.1.2.6 Resultado de la operación realizada	Respuestas al usuario Solicitudes al dispositivo Solicitudes al S. O.
7.1.2.7 Archivo transferido	Solicitudes al S. O.

7.3 Diseño.

7.3.1 Diagramas de flujo de datos.

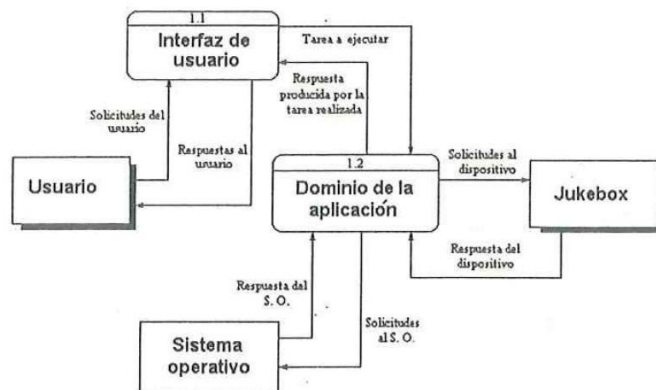


Figura 5. Diagrama de Flujo de datos (DFD) general.

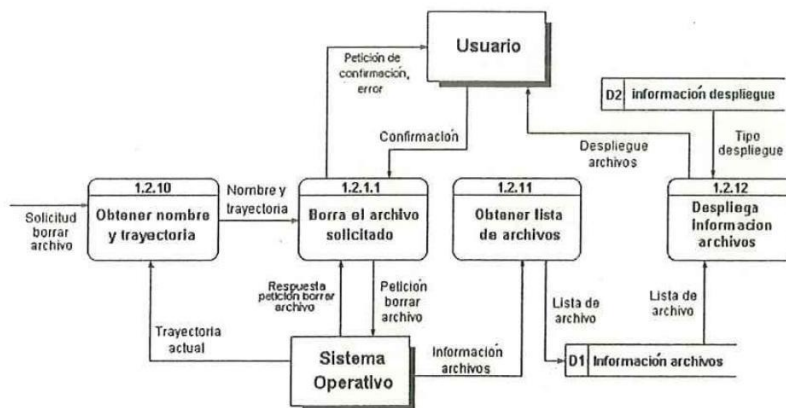


Figura 6. DFD 1.2.1. Borrar archivo.

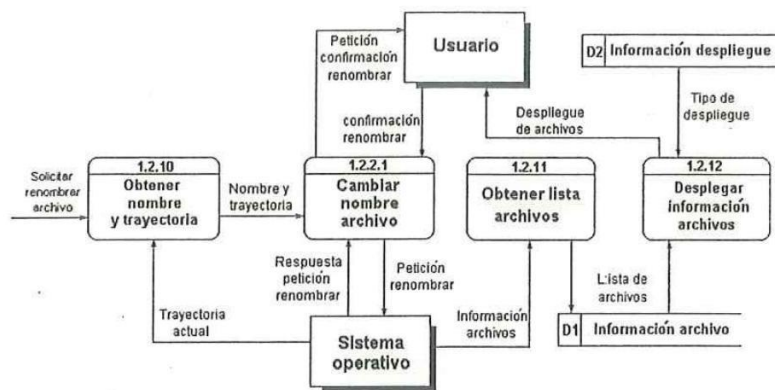


Figura 7. DFD 1.2.2 Renombrar archivo.

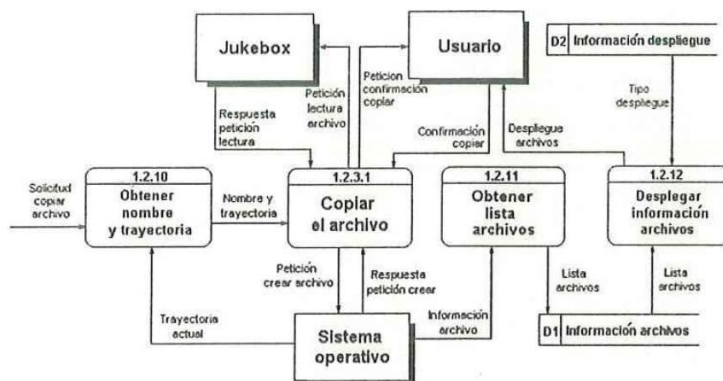


Figura 8. DFD 1.2.3 Copiar archivo.

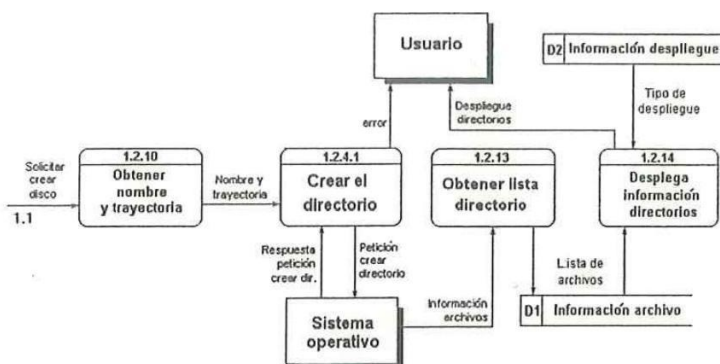


Figura 9. DFD 1.2.4 Crear directorio.



Figura 10. DFD 1.2.5 Borrar directorio.



Figura 11. DFD 1.2.6 Ordenar archivos.

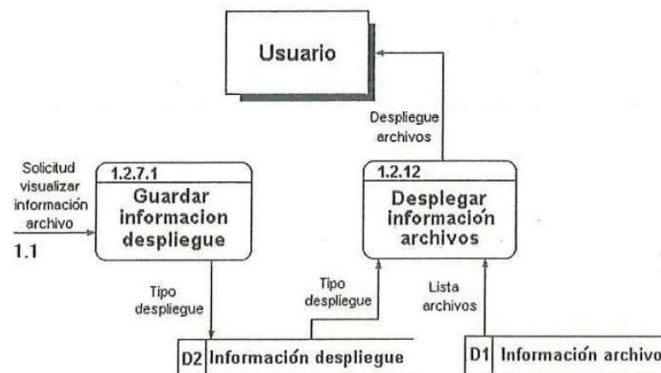


Figura 12. DFD 1.2.7 Visualizar información archivo.

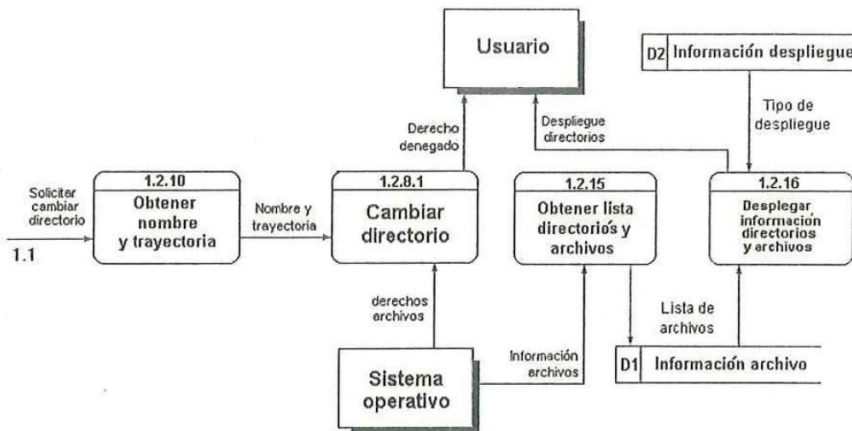


Figura 13. DFD 1.2.8 Cambiar directorio.

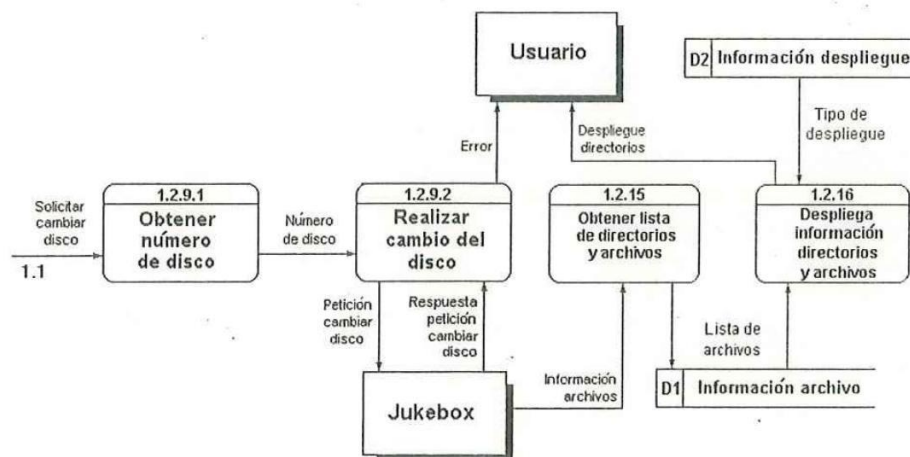


Figura 14. DFD 1.2.9 Cambiar disco.

7.3.2 Diagramas de Estructuras (DE).

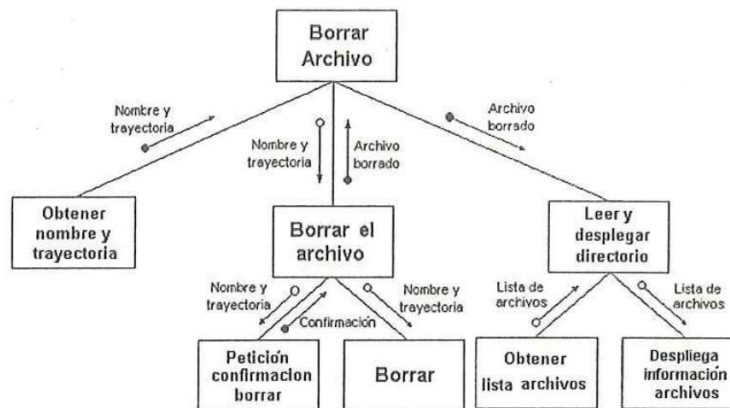


Figura 15. DE 1.2.1 Borrar archivo.

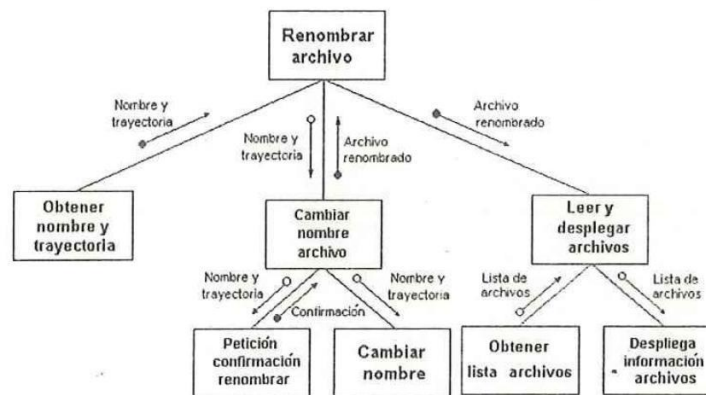


Figura 16. DE 1.2. 2 Renombrar archivo.

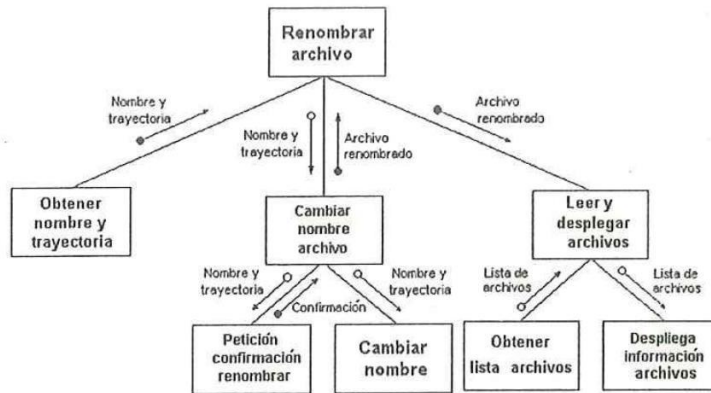


Figura 16. DE 1.2. 2 Renombrar archivo.

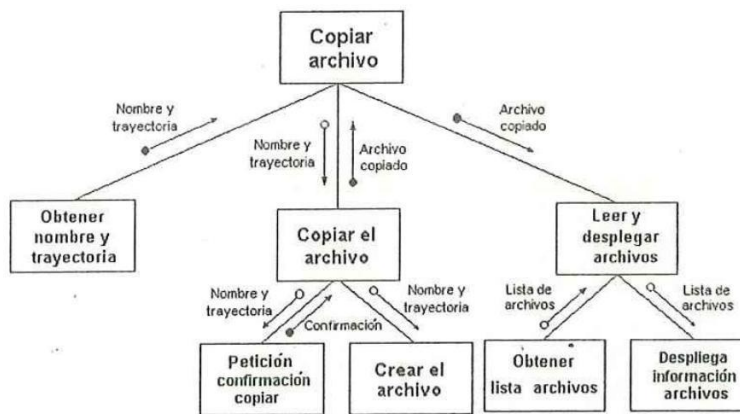


Figura 17. DE 1.2.3 Copiar archivo.

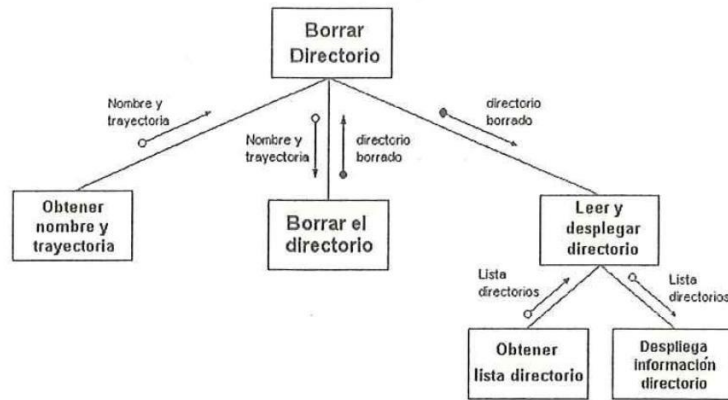


Figura 19. DE 1.2.5 Borrar directorio

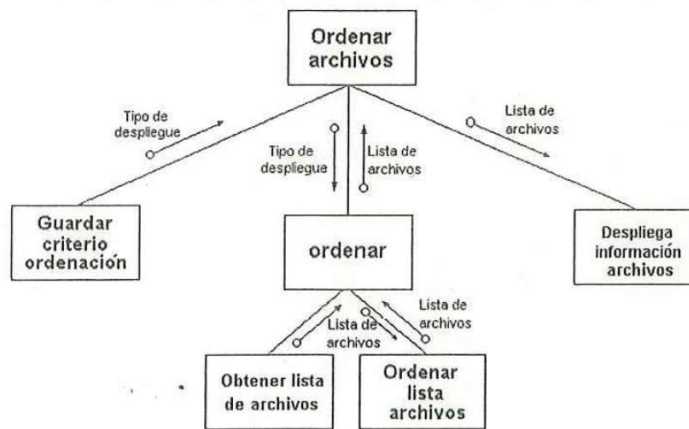


Figura 20. DE 1.2.6 Ordenar archivos.



Figura 21. DE 1.2.7 Visualizar información archivo.



Figura 22. DE 1.2.8 Cambiar directorio.



Figura 23. DE 1.2.9 Cambiar disco.

7.3.3 Diccionario de datos.

Tabla IV. Diccionario de datos.

Clave	Nombre	Tipo de elemento	Descripción
1.2.1.1	Borrar el archivo solicitado	proceso	Proceso que borra un archivo
1.2.8.1	Cambiar directorio	proceso	Proceso que cambia el directorio actual
1.2.2.1	Cambiar nombre archivo	proceso	proceso que cambia el nombre a los archivos
	confirmación	datos	confirmación del usuario para realizar una tarea
	Confirmación borrar	datos	Confirmación del usuario de borrar directorio
	Confirmación copiar	datos	Confirmación del usuario de copiar el archivo
	Confirmación renombrar	datos	Confirmación del usuario para renombrar
1.2.3.1	Copiar archivo	proceso	proceso que copia un archivo
1.2.4.1	Crear el directorio	proceso	Proceso para crear un directorio
	Derecho archivos	datos	Derecho de acceso de los archivos
	Derecho denegado	datos	Aviso de error al usuario de que no puede acceder el directorio

TablaIV (continuación) Diccionario de datos.

Clave	Nombre	Tipo de elemento	Descripción
1.2.14	Desplegar información directorios	proceso	Proceso que despliega en la pantalla la lista de directorios
1.2.16	Desplegar información directorios y archivos	proceso	Proceso que despliega en pantalla la lista de nombres de archivos y directorios e información de estos
1.2.12	Despliega información archivos	proceso	Proceso que formatea la lista de archivos y la manda al usuario a través de la interface
	Despliegue de archivos	datos	despliegue de la lista de archivos al usuario
1.2	Dominio de la aplicación	proceso	Proceso que realiza todas las operaciones de la aplicación
	error	datos	mensaje de error al usuario
1.1.6.1	Guardar criterio de ordenación	proceso	Proceso que guarda en D2 el criterio de ordenación que el usuario selecciona
1.2.7.1	Guardar información de despliegue	proceso	proceso que guarda el tipo de despliegue que el usuario selecciona
D1	Información archivos	datos	Lugar donde se almacena los nombres e información de los archivos
	información de los archivos	datos	información que el sistema operativo envía a la aplicación acerca de los archivos del usuario
D2	Información despliegue	Almacena miento de datos	Se guarda la información de la manera de desplegar los archivos

TablaIV (continuación) Diccionario de datos.

Clave	Nombre	Tipo de elemento	Descripción
1.1	Interface de usuario	proceso	Proceso que gestiona la interface de usuario
	Jukebox	entidad	Dispositivo de discos magnético-ópticos
	Lista de archivos	datos	Una estructura de datos que contiene el nombre e información de los archivos
	Nombre y trayectoria	datos	Una cadena de caracteres conteniendo el nombre y la trayectoria de un archivo o directorio
1.2.11	Obtener lista de archivos	proceso	Proceso que obtiene la lista de archivos del directorio del usuario
1.2.13	Obtener lista de directorio	proceso	Proceso que obtiene la lista de directorio del directorio actual del usuario
1.2.15	Obtener lista de directorios y archivos	proceso	Proceso que obtiene del sistema operativo nombre e información de los archivos y directorios
1.2.10	Obtener nombre y trayectoria	proceso	Proceso que obtiene el nombre y la trayectoria de un archivo o directorio
1.2.9.1	Obtener numero de disco	proceso	Proceso que obtiene el numero de disco que el usuario desea cargar en la lectora del dispositivo
1.1.6.2	Ordenar lista de archivos	proceso	Proceso que ordena la lista de archivos contenida en D1 según el criterio guardado en D2
	Petición borrar archivo	datos	Petición al sistema operativo de borrar un archivo

TablaIV (continuación) Diccionario de datos.

Clave	Nombre	Tipo de elemento	Descripción
	Petición borrar archivo	datos	Petición al sistema operativo de borrar un directorio
	Petición cambiar disco	dato	Petición al dispositivo de cargar un disco en la lectora
	Petición confirmación copiar	datos	Solicitud al usuario de que confirme su solicitud de copiar cuando ya existe un archivo con ese nombre
	Petición confirmación renombrar	datos	Petición al usuario para que confirme que realmente quiere ejecutar la tarea
	Petición crear archivo	datos	Solicitud al sistema operativo para crear un archivo
	Petición crear directorios	datos	Petición al sistema operativo de crear un directorio
	Petición de confirmación	datos	Se solicita una confirmación al usuario
	Petición lectura archivo	datos	Petición al dispositivo para leer un archivo
	Petición renombrar	datos	Petición al sistema operativo de renombrar un archivo

TablaIV (continuación) Diccionario de datos.

Clave	Nombre	Tipo de elemento	Descripción
1.2.9.2	Realizar cambio del disco	proceso	Proceso que cambia el disco de la lectora de la unidad
	Respuesta petición renombrar	datos	Resultado de la operación renombrar
	Respuesta petición archivo	datos	Resultado de la operación de lectura
	Respuesta petición borrar archivo	datos	Respuesta del sistema operativo a la petición borrar un archivo
	Respuesta petición borrar directorio	datos	Resultado de la operación borrar directorio
	Respuesta petición cambiar disco	dato	Resultado de la operación cambiar disco
	Respuesta petición crear	datos	Resultado de la operación crear archivo
	Respuesta petición crear directorio	datos	Resultado de la operación crear directorio
	Respuesta producida por la tarea realizada	datos	Son las respuestas de las tareas realizadas que se envían a la inteface de usuario para su despliegue al usuario

TablaIV (continuación) Diccionario de datos.

Clave	Nombre	Tipo de elemento	Descripción
	Respuestas al usuario	datos	Son todas las salidas que genera la aplicación a través de la interface
	Respuestas del dispositivo	datos	Los mensajes que el dispositivo mandaba a la aplicación como respuesta a una petición
	Respuestas del S. O.	datos	Los mensajes que el sistema operativo manda a al sistema como respuesta a una petición
	Sistema Operativo	entidad	El sistema operativo sobre el cual se ejecuta la aplicación
	Solicitud confirmación borrar dir.	Datos	Solicitud de confirmación al usuario de borrar directorio
	Solicitudes al dispositivo	datos	Solicitudes que el sistema hace al dispositivo magnético-óptico
	Solicitudes del S. O.	datos	Todas las solicitudes que el sistema hace al sistema operativo
	Solicitudes del usuario	datos	Solicitudes que el usuario hace a la aplicación a través de la interface de usuario
	Tarea a ejecutar	datos	Petición de ejecución de alguna tarea del sistema
	Tipo de despliegue	datos	Estructura de datos que contiene tipo el información que va desplegarse de los archivos así como el orden en que se presentarán
	Trayectoria actual	datos	Trayectoria actual del directorio de trabajo del usuario
	Usuario	entidad	El usuario de la aplicación

7.3.4 Seudocódigo.

1.2.1 Borrar archivo

```
{  
    Obtener nombre y trayectoria;  
    Petición confirmación borrar;  
    Borrar;  
    Obtener lista archivos;  
    Desplegar información archivos;  
}
```

1.2.2 Renombrar archivo

```
{  
    Obtener nombre y trayectoria;  
    Petición confirmación renombrar;  
    Cambiar nombre;  
    Obtener lista archivos;  
    Despliega información archivos;  
}
```

1.2.3 Copiar archivo

```
{  
    Obtener nombre y trayectoria;  
    Petición confirmación copiar  
    Crear el archivo;  
    Obtener lista de archivos;
```

```
        Despliega información archivos;
    }

    1.2.4 Crear directorio
    {

        Obtener nombre y trayectoria;
        Crear el directorio;
        Obtener lista directorio;
        Despliega información directorio;
    }

    1.2.5 Borrar directorio
    {

        Obtener nombre y trayectoria;
        Borrar el directorio
        Obtener lista directorio;
        Despliega información directorio;
    }

    1.2.6 Ordenar archivos
    {

        Guardar criterio ordenación;
        Obtener lista archivos;
        Ordenar lista archivos;
        Despliega información archivos;
    }
```


1.2.7 Visualizar información archivos

```
{  
    Guardar información despliegue;  
    Despliega información archivos;  
}
```

1.2.8 Cambiar directorio

```
{  
    Obtener nombre y trayectoria;  
    Revisar derechos;  
    Realizar cambio dir;  
    Obtener lista directorios y archivos;  
    Desplegar información directorios y archivos;  
}
```

1.2.9 Cambiar disco

```
{  
    Obtener numero de disco;  
    Realizar cambio del disco;  
    Obtener lista directorios y archivos;  
    Desplegar información directorios y archivos;  
}
```

1.3.1 Obtener lista archivos

```
{  
    Abrir directorio;
```

```
repetir hasta (leer todos los elementos archivos)
{
    Leer elemento archivo;
    Insertar datos del elemento archivo en la lista de
    archivos;
}
Cerrar directorio;
}
```

1.3.2 Desplegar información archivos

```
{
    repetir hasta (lista = fin)
    {
        Leer elemento de la lista;
        Desplegar elemento en la pantalla;
    }
}
```

1.3.3 Obtener lista directorio

```
{
    Abrir directorio;
    Repetir hasta (leer todos los elementos directorios)
    {
        Leer elemento directorios;
        Insertar datos del elemento directorio en la lista;
    }
}
```

```
Cerrar directorio;  
}
```

1.3.4 Despliega información directorio

```
{  
    repetir hasta (lista = fin)  
    {  
        Leer elemento de la lista;  
        Desplegar elemento en la pantalla;  
    }  
}
```

1.3.5 Ordenar lista archivos(Lista lista)

```
{  
    Quicksort (lista);  
}
```

1.3.6 Obtener lista directorios y archivos

```
{  
    Obtener lista archivos;  
    Obtener lista directorio;  
}
```

8. Pruebas del sistema.

8.1 Diseño de casos de prueba.

Para diseñar los casos de prueba se utilizó el método de particiones equivalentes. En la columna “Nombre del campo” se hace referencia a los elementos de la interfaz de usuario desde donde se hace la entrada de datos. Los números entre parentesis corresponden a los números asociados a cada elemento de la interfaz en el documento de especificación de software. Los números entre corchetes sirven para numerar los casos de prueba. Los casos para los cuales no se encontraron entradas inválidas se indicaron como “no aplica”.

Tabla V. Diseño de casos de prueba.

Dominio de entrada	Nombre del campo	Condición de entrada	Caso válido	Caso inválido
Copiar archivo	(1) Textedit	Cadena de entre 1 y 200 caracteres	[1] $1 \leq \text{cadena} \leq 200$	[2] cadena = 0 [3] cadena > 200
	(13) Botón copiar archivo ó	Oprimir el botón copiar	[4] Oprimir el botón copiar	no aplica
	(18) Opción copiar archivo	Seleccionar opción copiar archivo	[5] Seleccionar opción copiar archivo	no aplica

Tabla V (continuación) Diseño de casos de prueba.

Dominio de entrada	Nombre del campo	Condición de entrada	Caso válido	Caso inválido
Borrar archivo	(6) Textedit	Cadena de entre 1 y 200 caracteres	[6] $1 \leq \text{cadena} \leq 200$	[7] cadena = 0 [8] cadena > 200
	(14) Botón borrar archivo ó	Oprimir el botón borrar	[9] Oprimir el botón borrar	no aplica
	(19) Opción borrar archivo	Seleccionar opción borrar archivo	[10] Seleccionar opción borrar archivo	no aplica
Renombrar archivo	(6) Textedit	Cadena de entre 1 y 200 caracteres	[11] $1 \leq \text{cadena} \leq 200$	[12] cadena = 0 [13] cadena > 200
	(15) Botón renombrar archivo ó	Oprimir el botón renombrar	[14] Oprimir el botón renombrar	no aplica
	(20) Opción renombrar archivo	Seleccionar opción renombrar archivo	[15] Seleccionar opción renombrar archivo	no aplica
Crear directorio	(8) Textedit	Cadena de entre 1 y 200 caracteres	[16] $1 \leq \text{cadena} \leq 200$	[17] cadena = 0 [18] cadena > 200
	(17) Botón crear dir. ó	Oprimir el botón crear dir.	[19] Oprimir el botón crear dir.	no aplica
	(21) Opción crear directorio	Seleccionar opción crear directorio	[20] Seleccionar opción crear directorio	no aplica

Tabla V (continuación) Diseño de casos de prueba.

Dominio de entrada	Nombre del campo	Condición de entrada	Caso válido	Caso inválido
Borrar directorio	(8) Textedit	Cadena de entre 1 y 200 caracteres	[21] $1 \leq \text{cadena} \leq 200$	[22] cadena = 0 [23] cadena > 200
	(16) Botón crear dir. ó	Oprimir el botón borrar dir.	[24] Oprimir el botón crear dir.	no aplica
	(22) Opción crear directorio	Seleccionar opción borrar directorio	[25] Seleccionar opción crear directorio	no aplica
Visualizar información tamaño de los archivos del dispositivo	(25) Jukebox tamaño	Seleccionar opción Jukebox tamaño	[26] Seleccionar opción Jukebox tamaño	no aplica
Visualizar información fecha de los archivos del dispositivo	(26) Jukebox fecha	Seleccionar opción Jukebox fecha	[27] Seleccionar opción Jukebox fecha	no aplica
Visualizar información atributos de los archivos del dispositivo	(27) Jukebox atributos	Seleccionar opción Jukebox atributos	[28] Seleccionar opción Jukebox atributos	no aplica
Visualizar información tamaño de los archivos del usuario	(28) Usuario tamaño	Seleccionar opción Usuario tamaño	[29] Seleccionar opción Usuario tamaño	no aplica

Tabla V (continuación) Diseño de casos de prueba.

Dominio de entrada	Nombre del campo	Condición de entrada	Caso válido	Caso inválido
Visualizar información fecha de los archivos del usuario	(29) Usuario fecha	Selecciona opción usuario fecha	[30] Selecciona opción usuario fecha	no aplica
Visualizar información atributos de los archivos del usuario	(30) Usuario atributos	Selecciona opción usuario atributos	[31] Selecciona opción usuario atributos	no aplica
Ordenar lista archivos del jukebox por nombre	(31) ordenar jukebox; opción nombre	Seleccionar opción ordenar jukebox; opción nombre	[32] Seleccionar opción ordenar jukebox; opción nombre	no aplica
Ordenar lista archivos del jukebox por tamaño	(31) ordenar jukebox; opción tamaño	Seleccionar opción ordenar jukebox; opción tamaño	[33] Seleccionar opción ordenar jukebox; opción tamaño	no aplica
Ordenar lista archivos del jukebox por fecha	(31) Ordenar jukebox; opción fecha	Seleccionar opción ordenar jukebox; opción fecha	[34] Seleccionar opción ordenar jukebox; opción fecha	no aplica
Ordenar lista archivos del usuario por nombre	(32) Ordenar usuario; opción nombre	Seleccionar opción ordenar usuario; opción nombre	[35] Seleccionar opción ordenar usuario; opción nombre	no aplica

Tabla V (continuación) Diseño de casos de prueba.

Dominio de entrada	Nombre del campo	Condición de entrada	Caso válido	Caso inválido
Ordenar lista archivos del usuario por tamaño	(32) Ordenar usuario; opción tamaño	Seleccionar opción ordenar usuario; opción tamaño	[36] Seleccionar opción ordenar usuario; opción tamaño	no aplica
Ordenar lista archivos del usuario por fecha	(32) Ordenar usuario; opción fecha	Seleccionar opción ordenar usuario; opción tamaño	[37] Seleccionar opción ordenar usuario; opción tamaño	no aplica
Seleccionar disco	(5) Disco	Seleccionar opción disco	[38] Seleccionar disco contenido en la lectora	[39] Seleccionar disco ausente en la unidad
Ayuda en línea	(33) Contenido	seleccionar opción contenido	[40] seleccionar opción contenido	no aplica
Información del sistema	(34) Acerca de..	Seleccionar opción acerca de...	[41] Seleccionar opción acerca de...	no aplica

8.2 Casos de prueba.

Los casos de prueba que se crearon a partir del diseño de casos de prueba son mostrados a continuación. Las entradas subrayadas son mandos ejecutados sobre la interfaz como por ejemplo seleccionar una opción del menú u oprimir un botón. Las entradas escritas en cursivas son entradas de texto desde el teclado. Los números entre corchetes hacen referencia a los casos de prueba de la sección anterior.

Tabla VI. Casos de prueba.

Dominio de entrada	Caso válido	Caso inválido
Copiar archivo	[1] <i>"archivo1"</i>	[2] " "
		[3] <i>"111111111122222222223333333333444444444555555555566666666667777777777888888888899999999990000000000111111111122222222223333333333444444444555555555566666666667777777777888888888899999999990000000000x"</i>
	[4] <u>Oprimir el botón copiar</u>	no aplica
	[5] <u>Seleccionar opción copiar archivo</u>	no aplica

Tabla VI (continuación) Casos de prueba.

Dominio de entrada	Caso válido	Caso inválido
Borrar archivo	[6] "Archivo2"	[7] " "
		[8] "111111111122222222223333333333444 4444444555555555556666666666777777 777788888888889999999999000000000 111111111122222222223333333333444 4444444555555555556666666666777777 7788888888889999999999000000000x"
	[9] <u>Oprimir el botón borrar</u>	no aplica
	[10] <u>Seleccionar opción borrar archivo</u>	no aplica
Renombrar archivo	[11] "NombreArchivo"	[12] " "
		[13] "111111111122222222223333333333444 4444444555555555556666666666777777 777788888888889999999999000000000 111111111122222222223333333333444 4444444555555555556666666666777777 7788888888889999999999000000000x"
	[14] <u>Oprimir el botón renombrar</u>	no aplica
	[15] <u>Seleccionar opción renombrar archivo</u>	no aplica

Tabla VI (continuación) Casos de prueba.

Dominio de entrada	Caso válido	Caso inválido
Crear directorio	[16] "NuevoDir"	[17] " "
		[18] "111111111122222222223333333333444 444444455555555555666666666677777 777788888888889999999999000000000 111111111122222222223333333333444 444444455555555555666666666677777 7788888888889999999999000000000x"
	[19] <u>Oprimir el botón crear dir</u>	no aplica
	[20] <u>Seleccionar opción crear directorio</u>	no aplica
Borrar directorio	[21] "NuevoDir"	[22] " "
		[23] "111111111122222222223333333333444 444444455555555555666666666677777 777788888888889999999999000000000 111111111122222222223333333333444 444444455555555555666666666677777 7788888888889999999999000000000x"
	[24] <u>Oprimir el botón borrar dir</u>	no aplica
	[25] <u>Seleccionar opción borrar directorio</u>	no aplica
Visualizar información tamaño de los archivos del dispositivo	[26] <u>Seleccionar opción Jukebox tamaño</u>	no aplica

Tabla VI (continuación) Casos de prueba.

Dominio de entrada	Caso válido	Caso inválido
Visualizar información fecha de los archivos del dispositivo	[27] <u>Seleccionar opción Jukebox fecha</u>	no aplica
Visualizar información atributos de los archivos del dispositivo	[28] <u>Seleccionar opción Jukebox atributos</u>	no aplica
Visualizar información tamaño de los archivos del usuario	[29] <u>Seleccionar opción Usuario tamaño</u>	no aplica
Visualizar información fecha de los archivos del usuario	[30] <u>Selecciona opción usuario fecha</u>	no aplica
Visualizar información atributos de los archivos del usuario	[31] <u>Selecciona opción usuario atributos</u>	no aplica
Ordenar lista archivos del jukebox por nombre	[32] <u>Seleccionar opción ordenar jukebox; opción nombre</u>	no aplica
Ordenar lista archivos del jukebox por tamaño	[33] <u>Seleccionar opción ordenar jukebox; opción tamaño</u>	no aplica
Ordenar lista archivos del jukebox por fecha	[34] <u>Seleccionar opción ordenar jukebox; opción fecha</u>	no aplica
Ordenar lista archivos del usuario por nombre	[35] <u>Seleccionar opción ordenar usuario; opción nombre</u>	no aplica

Tabla VI (continuación) Casos de prueba.

Dominio de entrada	Caso válido	Caso inválido
Ordenar lista archivos del usuario por tamaño	[36] <u>Seleccionar opción</u> <u>ordenar usuario; opción tamaño</u>	no aplica
Ordenar lista archivos del usuario por fecha	[37] <u>Seleccionar opción</u> <u>ordenar usuario; opción tamaño</u>	no aplica
Seleccionar disco	[38] <u>Seleccionar disco</u> <u>contenido en la lectora</u>	[39] <u>Seleccionar disco ausente en la</u> <u>unidad</u>
Ayuda en línea	[40] <u>seleccionar opción</u> <u>contenido</u>	no aplica
Información del sistema	[41] <u>Seleccionar opción acerca</u> <u>de...</u>	no aplica

8.3 Resultados de los casos de prueba.

Tabla VII. Resultados de los casos de prueba.

Dominio de entrada	Caso de prueba	Respuesta del sistema
Copiar archivo	[1] "archivo1" [4] <u>Oprimir el botón copiar</u>	El archivo fue copiado
	[1] "archivo1" [5] <u>Seleccionar opción copiar</u>	El archivo fue copiado
	[2] " " [4] <u>Oprimir el botón copiar</u>	No fue posible oprimir el boton copiar archivo porque estaba desactivado
	[2] " " [5] <u>Seleccionar opción copiar</u>	No fue posible seleccionar la opcion porque estaba desactivada
	[3] "111111111222222222333333333444 4444445555555555666666666777777 7777888888888999999999000000000 111111111222222222333333333444 4444455555555556666666667777777 7788888888999999990000000000x" [4] <u>Oprimir el botón copiar</u>	No fue posible probar este caso porque el programa no puede leer directorios con nombres de mas de 200 caracteres (vease las conclusiones)
	[3] "111111111222222222333333333444 4444445555555555666666666777777 7777888888888999999999000000000 111111111222222222333333333444 4444455555555556666666667777777 7788888888999999990000000000x" [5] <u>Seleccionar opción copiar</u>	No fue posible probar este caso porque el programa no puede leer directorios con nombres de mas de 200 caracteres

Tabla VII (continuación) Resultados de los casos de prueba.

Dominio de entrada	Caso de prueba	Respuesta del sistema
Borrar archivo	[6] "Archivo2" [9] <u>Oprimir botón borrar</u>	El archivo fue borrado
	[6] "Archivo2" [10] <u>Seleccionar opción borrar archivo</u>	El archivo fue borrado
	[7] " " [9] <u>Oprimir botón borrar</u>	No fue posible oprimir el botón borrar porque estaba deshabilitado
	[7] " " [10] <u>Seleccionar opción borrar archivo</u>	No fue posible seleccionar la opción porque estaba deshabilitada
	[8] "111111111222222222233333333334444444444555555555566666666667777777777888888888899999999990000000001111111112222222222333333333344444444445555555555666666666677777777778888888888999999999900000000000000000x" [9] <u>Oprimir botón borrar</u>	No fue posible probar este caso porque el programa no puede leer directorios con nombres de mas de 200 caracteres
	[8] "1111111112222222222333333333344444444445555555555666666666677777777778888888888999999999900000000001111111112222222222333333333344444444445555555555666666666677777777778888888888999999999900000000000000000x" [10] <u>Seleccionar opción borrar archivo</u>	No fue posible probar este caso porque el programa no puede leer directorios con nombres de mas de 200 caracteres

Tabla VII (continuación) Resultados de los casos de prueba.

Dominio de entrada	Caso de prueba	Respuesta del sistema
Renombrar archivo	[11] "NombreArchivo" [14] <u>Oprimir botón renombrar</u>	El archivo fue cambiado de nombre
	[11] "NombreArchivo" [15] <u>Seleccionar opción renombrar archivo</u>	El archivo fue cambiado de nombre
	[12] "" [14] <u>Oprimir botón renombrar</u>	No fue posible oprimir el botón porque estaba deshabilitado
	[12] "" [15] <u>Seleccionar opción renombrar archivo</u>	No fue posible seleccionar la opción porque estaba deshabilitada
	[13] "111111111222222222233333 333334444444444555555555566666 6666677777777778888888888999999 999900000000001111111112222222 22233333333334444444455555555 556666666666777777777888888888 89999999999000000000x" [14] <u>Oprimir botón renombrar</u>	No fue posible probar este caso porque el programa no puede leer directorios con nombres de mas de 200 caracteres

Tabla VII (continuación) Resultados de los casos de prueba.

Dominio de entrada	Caso de prueba	Respuesta del sistema
	[13] "1111111112222222223333333333444444444555555555666666666777777777888888888999999999000000000111111111222222222333333333444444444555555555666666666777777777888888888999999999000000000x" [15] <u>Seleccionar opción renombrar archivo</u>	No fue posible probar este caso porque el programa no puede leer directorios con nombres de mas de 200 caracteres
Crear directorio	[16] "NuevoDir" [19] <u>Oprimir el botón crear dir</u>	El directorio fue creado
	[16] "NuevoDir" [20] <u>Seleccionar opción crear directorio</u>	El directorio fue creado
	[17] "" [19] <u>Oprimir el botón crear dir</u>	El sistema desplegó un mensaje de error del sistema operativo
	[17] "" [20] <u>Seleccionar opción crear directorio</u>	El sistema desplegó un mensaje de error del sistema operativo

Tabla VII (continuación) Resultados de los casos de prueba.

Dominio de entrada	Caso de prueba	Respuesta del sistema
	[18] "1111111112222222222333333333444 4444444555555555566666666677777 7777888888889999999990000000000 1111111112222222222333333333444 4444444555555555566666666677777 7788888888999999999000000000x" [19] <u>Oprimir el botón crear dir</u>	No fue posible probar este caso porque el programa no puede leer directorios con nombres de mas de 200 caracteres
	[18] "1111111112222222222333333333444 4444444555555555566666666677777 7777888888889999999990000000000 1111111112222222222333333333444 4444444555555555566666666677777 7788888888999999999000000000x" [20] <u>Seleccionar opción crear directorio</u>	No fue posible probar este caso porque el programa no puede leer directorios con nombres de mas de 200 caracteres
Borrar directorio	[21] "NuevoDir" [24] <u>Oprimir botón borrar dir</u>	El directorio fue borrado
	[21] "NuevoDir" [25] <u>Seleccionar opción borrar directorio</u>	El directorio fue borrado
	[22] " " [24] <u>Oprimir botón borrar dir</u>	No fue posible oprimir el botón porque estaba deshabilitado
	[22] " " [25] <u>Seleccionar opción borrar directorio</u>	No fue posible seleccionar la opción porque estaba deshabilitada

Tabla VII (continuación) Resultados de los casos de prueba.

Dominio de entrada	Caso de prueba	Respuesta del sistema
	[23] "111111111222222222333333333444444444555555555666666666777777777888888888999999999000000000111111111222222222333333333444444444555555555666666666777777777888888888999999999000000000x" [24] <u>Oprimir botón borrar dir</u>	No fue posible probar este caso porque el programa no puede leer directorios con nombres de mas de 200 caracteres
	[23] "111111111222222222333333333444444444555555555666666666777777777888888888999999999000000000111111111222222222333333333444444444555555555666666666777777777888888888999999999000000000x" [25] <u>Seleccionar opción borrar directorio</u>	No fue posible probar este caso porque el programa no puede leer directorios con nombres de mas de 200 caracteres
Visualizar información tamaño de los archivos del dispositivo	[26] <u>Seleccionar opción Jukebox tamaño</u>	La información del tamaño de los archivos del dispositivo fue desplegada
Visualizar información fecha de los archivos del dispositivo	[27] <u>Seleccionar opción Jukebox fecha</u>	La fecha de los archivos del dispositivo fue desplegada
Visualizar información atributos de los archivos del dispositivo	[28] <u>Seleccionar opción Jukebox atributos</u>	Los atributos de los archivos del dispositivo fueron desplegados
Visualizar información tamaño de los archivos del usuario	[29] <u>Seleccionar opción Usuario tamaño</u>	La información del tamaño de los archivos del usuario fue desplegada
Visualizar información fecha de los archivos del usuario	[30] <u>Selecciona opción usuario fecha</u>	La fecha de los archivos del usuario fue desplegada

Tabla VII (continuación) Resultados de los casos de prueba.

Dominio de entrada	Caso de prueba	Respuesta del sistema
Visualizar información atributos de los archivos del usuario	[31] <u>Selecciona opción</u> <u>usuario atributos</u>	Los atributos de los archivos del usuario fueron desplegados
Ordenar lista archivos del jukebox por nombre	[32] <u>Seleccionar opción</u> <u>ordenar jukebox; opción nombre</u>	Los archivos del dispositivo fueron ordenados por nombre
Ordenar lista archivos del jukebox por tamaño	[33] <u>Seleccionar opción</u> <u>ordenar jukebox; opción tamaño</u>	Los archivos del dispositivo fueron ordenados por su tamaño
Ordenar lista archivos del jukebox por fecha	[34] <u>Seleccionar opción</u> <u>ordenar jukebox; opción fecha</u>	Los archivos del dispositivo fueron ordenados por su fecha
Ordenar lista archivos del usuario por nombre	[35] <u>Seleccionar opción</u> <u>ordenar usuario; opción nombre</u>	Los archivos del usuario fueron ordenados por nombre
Ordenar lista archivos del usuario por tamaño	[36] <u>Seleccionar opción</u> <u>ordenar usuario; opción tamaño</u>	Los archivos del usuario fueron ordenados por su tamaño

Tabla VII (continuación) Resultados de los casos de prueba.

Dominio de entrada	Caso de prueba	Respuesta del sistema
Ordenar lista archivos del usuario por fecha	[37] <u>Seleccionar opción ordenar usuario; opción tamaño</u>	Los archivos del usuario fueron ordenados por su fecha
Seleccionar disco	[38] <u>Seleccionar disco contenido en la lectora</u>	No fue posible probar este caso porque la unidad de discos estaba descompuesta
	[39] Seleccionar disco ausente en la unidad	No fue posible probar este caso porque la unidad de discos estaba descompuesta
Ayuda en línea	[40] Seleccionar opción contenido	La ayuda fue desplegada
Información del sistema	[41] Seleccionar opción acerca de ...	La información del sistema fue desplegada

9. Conclusiones.

9.1 Posibles mejoras al programa.

Durante las pruebas del sistema se encontró que si se intentaba crear un archivo o directorio cuyo nombre excediera los 200 caracteres, el programa fallaba, esto se debe a que la longitud máxima que se definió para los nombres es de 200 caracteres. Sin embargo es necesario realizar una validación de las entradas de los nombres de los archivos y directorios de manera que no acepte los nombres que excedan los 200 caracteres.

Es posible aumentar la funcionalidad del sistema agregándole algunas características como:

- Múltiple selección de archivos para borrar o copiar.
- Borrar árboles de directorios.
- Soporte del mecanismo arrastrar y soltar (*Drag and drop*).
- Ayuda sensible al contexto.

9.2 Conclusiones generales.

El diseño de interfaces de usuario a evolucionado mucho desde los inicios de la computación a la actualidad. Ahora utilizamos programas con interfaces más fáciles de manejar. La tendencia es lograr que la interacción del hombre con la computadora sea lo más fácil y natural posible, de manera que cualquier persona pueda utilizarla sin mucha capacitación previa.

Las herramientas para el desarrollo de interfaces como Motif Toolkit son de gran ayuda en el desarrollo de sistemas interactivos. Ya no es necesario invertir tanto tiempo en la creación de la interface y se puede concentrar más el trabajo en el dominio de la aplicación. Además los lineamientos Motif ayudan a desarrollar sistemas con interfaces consistentes reduciendo mucho el esfuerzo de aprendizaje de manejo de sistemas computacionales nuevos.

La programación en Solaris (Unix) ofrece muchas ventajas como por ejemplo multitarea, una gran cantidad de memoria principal disponible, un sistema operativo robusto, y otras. El problema es que no se cuenta en el CECCUE con las herramientas de programación (como Visual WorkShop C++ y WorkShop Professional Pascal) que se tienen para otras plataformas como son Windows o DOS. Esto no ayuda en el desarrollo de programas para Unix.

Apéndice.

A continuación se presenta parte del código fuente del sistema.

1.2.1 Borrar archivo

```
void Boton2_pro(menu_item, client_data, call_data)
Widget menu_item;
XtPointer client_data;
XtPointer call_data;
{
    char *archivo2;
    char *cadena;
    int posicion;

    /* Obtener nombre y trayectoria */
    cadena = XmTextGetString(Usuario.Text1b);
    if(NULL == (archivo2 = (char*)malloc(NOMBRELONG)))
        MensajeError(Vprincipal, "Error al asignar memoria");
    sprintf(archivo2, "%s/%s", path, cadena);
    /*Petición confirmación borrar */
    if(1 == AskUser(Shell, "Desea borrar el archivo seleccionado", "SI", "NO",
2))
    {
        /* Borrar */
```



```

        if(-1 == unlink(archivo2))

            MensajeError(Shell, sys_errlist[errno]);

    }

    ...

    /* Obtener lista archivos y Desplegar información archivos */
    if(ERROR != CambiarDirectorioD(path, Usuario.ListaU2))

        CambiarDirectorioA(path, Usuario.ListaU1, Usuario.Textlb, listaA);

    ...

}

```

1.2.2 Renombrar archivo

```

void RenombrarA_pro(menu_item, client_data, call_data)

Widget menu_item;

XtPointer client_data;

XtPointer call_data;

{

    ...

    /* Obtener nombre y trayectoria */

    archivo = XmTextGetString(Usuario.Textlb);

    XmStringGetLtoR(param->value, XmFONTLIST_DEFAULT_TAG, &Sarchivo);

    if(Sarchivo[0] == '\0')

        MensajeError(Shell, "Debe introducir el nuevo nombre");

    else

    {

        sprintf(fuente, "%s/%s", path, archivo);

        sprintf(destino, "%s/%s", path, Sarchivo);
    }
}

```

```

if(NULL != (pArc = fopen(destino, "r")))
{
    /* Petición confirmación renombrar */

    if(2 == AskUser(menu_item, "El Archivo ya existe. Desea
sobrescribirlo?",

        "SI", "NO", 2))

        cond = 1;

fclose(pArc);
}

/* Cambiar nombre */

if(-1 == rename(fuente, destino))

    MensajeError(Shell, sys_errlist[errno]);

/* Obtener lista archivos Despliega información archivos */

if(ERROR != CambiarDirectorioD(path, Usuario.ListaU2))

    CambiarDirectorioA(path, Usuario.ListaU1, Usuario.Textlb,
listaA);

. . .
}

```

1.2.3 Copiar archivo

```

void Boton1_pro(menu_item, client_data, call_data)

Widget menu_item;

XtPointer client_data;

XtPointer call_data;

{

    char *archivo, cond=0;

    char *destino, *fuente;

```

```

FILE *arch;

/* Obtener nombre y trayectoria */
archivo = XmTextGetString(Jukebox.Text1);
sprintf(destino, "%s/%s", path, archivo);
sprintf(fuente, "%s/%s", pathD, archivo);
if(NULL != (arch = fopen(destino, "r")))
{
    /* Petición confirmación copiar */

    if(2 == AskUser(menu_item, "El archivo ya existe. Desea sobrescribirlo?", "SI", "NO", 2))
    . . .

    /* Crear el archivo */
    if(NULL != CopiarArchivo(fuente, destino))
        MensajeError(Shell, "Error al copiar el archivo");
    . . .

    /* Obtener lista de archivos y Despliega información archivos */
    if(ERROR != CambiarDirectorioD(path, Usuario.ListaU2))
        CambiarDirectorioA(path, Usuario.ListaU1, Usuario.Text1b, listaA);
    . . .
}

```

1.2.4 Crear directorio

```

void CrearDir_pro(menu_item, client_data, call_data)

Widget menu_item;

XtPointer client_data;

```

```
XtPointer call_data;

{

    /* Obtener nombre y trayectoria */

    if(NULL == (direc = (char*)malloc(NOMBRELONG)))

        MensajeError(Shell, "Error al asignar memoria");

    XmStringGetLtoR(param->value, XmFONTLIST_DEFAULT_TAG, &archivo);

    sprintf(direc, "%s/%s", path, archivo);

    /* Crear el directorio */

    if(-1 == mkdir(direc, 0xled))

        MensajeError(Shell, sys_errlist[errno]);

    . . .

    /* Obtener lista directorio y Despliega información directorio */

    CambiarDirectorioD(path, Usuario.ListaU2);

    . . .

}
```

1.2.5 Borrar directorio

```
void Boton5_pro(menu_item, client_data, call_data)

Widget menu_item;

XtPointer client_data;

XtPointer call_data;

{

    char *archivo;

    char *cadena;

    /* Obtener nombre y trayectoria */

    if(NULL == (archivo = (char*)malloc(NOMBRELONG)))
```

```

    MensajeError(Shell, "Error al asignar memoria");

    sprintf(archivo, "%s/%s", path, DirBorrar);

    /* Borrar el directorio */

    if(1==AskUser(Shell, "Desea borrar el Directorio seleccionado?",

        "SI", "NO", 2))

    {

        if(-1 == rmdir(archivo))

            MensajeError(Shell, sys_errlist[errno]);

        . . .

        /* Obtener lista directorio y Despliega información directorio */

        if(ERROR != CambiarDirectorioD(path, Usuario.ListaU2))

            CambiarDirectorioA(path, Usuario.ListaU1, Usuario.Text1b, listaA);

        . . .

    }

```

1.2.6 Ordenar archivos

```

char CambiarDirectorioA(directorio, Archivo, Texto, lista)

char *directorio;

Widget Archivo;

Widget Texto;

ListaArchivos *lista;

{

    . . .

    /* Obtener lista archivos */

    inicio = listaA;

```



```

/* Ordenar lista archivos */

Quicksort(inicio, nodo, VUsuario.orden);

. . .

/ * Despliega información archivos */

nodo = lista;
while(nodo->sig != NULL)
{
    cadena[0] = '\0';

    if(VUsuario.permisos)
    {
        sprintf(cadena, "0%-5o", nodo->modo&0777);

        sprintf(cadena, "%s %-25s", cadena, nodo->archivo);
    }
    else
        sprintf(cadena, "%-25s", nodo->archivo);

    if(VUsuario.longitud)
        sprintf(cadena, "%s %8d bytes", cadena, nodo->longitud);

    if(VUsuario.fecha)
        sprintf(cadena, "%s %s", cadena, asctime(nodo->fecha));

    entrada = XmStringCreateLocalized(cadena);
    XmListAddItemUnselected(Archivo, entrada, 0);

    nodo = nodo->sig;
}

```

```

    free(cadena);
}

1.2.7 Visualizar información archivos

void visual_pro(widget, client_data, call_data)
Widget widget;
XtPointer client_data;
XtPointer call_data;
{
    /* Guardar información despliegue */
    int opcion = (int)client_data;
    int condicion = 0;

    switch (opcion)
    {
        case 0: if(VUsuario.longitud)
                VUsuario.longitud = 0;
                else
                VUsuario.longitud = 1;
                condicion = 1;
                break;
        case 1: if(VUsuario.fecha)
                VUsuario.fecha = 0;
                else
                VUsuario.fecha = 1;
    }
}

```

```

        condicion = 1;

        break;

    case 2:if(VUsuario.permisos)

        VUsuario.permisos = 0;

    else

        VUsuario.permisos = 1;

    condicion = 1;

    break;

    case 3:if(VJukebox.longitud)

        VJukebox.longitud = 0;

    else

        VJukebox.longitud = 1;

    break;

    case 4:if(VJukebox.fecha)

        VJukebox.fecha = 0;

    else

        VJukebox.fecha = 1;

    break;

    case 5:if(VJukebox.permisos)

        VJukebox.permisos = 0;

    else

        VJukebox.permisos = 1;

    break;

}

```

```

/* Despliega información archivos */
if(1 == condicion)
{
    if(ERROR != CambiarDirectorioD(path, Usuario.ListaU2))
        CambiarDirectorioA(path, Usuario.ListaU1, Usuario.Text1b, listaA);
}
else
{
    CambiarDirectorioDJuke(pathD, Jukebox.ListaJ2);
    CambiarDirectorioAJuke(pathD, Jukebox.ListaJ1, Jukebox.Text1,
listaDA);
}
}

```

1.2.8 Cambiar directorio

```

void cambiar_pro(menu_item, client_data, call_data)
Widget menu_item;
XtPointer client_data;
XtPointer call_data;
{
    XmListCallbackStruct *cbs = (XmListCallbackStruct*) call_data;
    int posicion;
    char *cadena;
    Widget a;
    struct stat buf;

    /* Obtener nombre y trayectoria */

```

```

posicion = cbs->item_position;
cadena = ObtenerElemento(listaD, posicion);
if(cadena == NULL)
    MensajeError(Shell, "Error al obtenet elemento");
if(0 == strcmp(cadena, ".."))
    CortarTray(path);
else
{
    if(0 == strcmp(cadena, "."))
        return;
    else
        sprintf(path, "%s/%s", path, cadena);
}

/* Revisar derechos */
if(-1 == stat(path, &buf))
    MensajeError(Shell, sys_errlist[errno]);

if( 7 > ((buf.st_mode & 0x1c0)>>6))
{
    MensajeError(Shell, "Acceso negado");
    CortarTray(path);
    return;
}

. . .

/* Realizar cambio dir,

```


Obtener lista directorios y archivos y

Desplegar información directorios y archivos

```

*/
if(ERROR != CambiarDirectorioD(path, menu_item))
    CambiarDirectorioA(path, Usuario.ListaU1, Usuario.Text1b, listaA);
}

```

1.2.9 Cambiar disco

```

void seleccionaDisco_pro(widget, client_data, call_data)
Widget widget;
XtPointer client_data;
XtPointer call_data;
{
    Widget a;
    char *error;

    /* Obtener numero de disco */
    NumDisco = (int)client_data;
    . . .

    /* Realizar cambio del disco */
    system(desmontar);
    if((error = eoinsert(unidad, unidad, NumDisco+1, lado)) != NULL)
    {
        MensajeError(Shell, error);
        ResetCursor(VPrincipal);
        return;
    }
}

```

```

strcpy(pathD, montar_dir);

system(montar);

    /* Obtener lista directorios y archivos,
       Desplegar información directorios y archivos */
CambiarDirectorioDJuke(pathD, Jukebox.ListaJ2);
CambiarDirectorioAJuke(pathD, Jukebox.ListaJ1, Jukebox.Text1, listaDA);
. . .
}

/* Obtener lista archivos */
1.3.1 char ObtenerListaDir(directorio)
char *directorio;
{
    DIR *dir;
    . . .

    /* Abrir directorio */
    if( NULL == (dir = opendir(directorio)) )
    {
        MensajeError(Shell, "Error al abrir el directorio");

        return(ERROR);
    }
    . . .

    /* Repetir hasta (leer todos los elementos archivos) */
    while((a = readdir(dir)) != NULL)
    {

```

```

/* Leer elemento archivo */

sprintf(cadena, "%s/%s", directorio, a->d_name);

if(-1 == stat(cadena, &buf))

    MensajeError(Shell, sys_errlist[errno]);

. . .

if( (buf.st_mode & S_IFMT) == S_IFREG )/** ARCHIVO ***/

{

    /* Insertar datos del elemento archivo en la lista de archivos */

    strcpy(listaa->archivo, a->d_name);

    listaa->fecha = (struct tm*)malloc(sizeof(struct tm));

    CopiaEstructura(listaa->fecha, localtime(&buf.st_mtime));

    listaa->cfecha = (listaa->fecha->tm_year*10000 +

        ((listaa->fecha->tm_mon+1)*100) +

        listaa->fecha->tm_mday)+

        (((listaa->fecha->tm_hour+1)*10000) + ((listaa->

>fecha->tm_min+1)*100)

        + listaa->fecha->tm_sec) /1000000.0);

    listaa->longitud = buf.st_size;

    listaa->modo = buf.st_mode;

    listaa->elemento = elementoA++;

    if(NULL==(listaa->sig=(ListaArchivos*)malloc(sizeof(

        ListaArchivos))) )

        MensajeError(Shell, "Error al asignar memoria");

    la = listaa;

```

```

        listaa = listaa->sig;

        listaa->ant = la;

        listaa->sig = NULL;
    }
}

/* Cerrar directorio */
if( -1 == closedir(dir))
{
    MensajeError(Shell, "Error al cerrar el directorio");
    return(ERROR);
}

return(EXITO);
}

```

1.3.2 Desplegar información archivos

```

...

nodo = lista;

/* repetir hasta (lista = fin) */
while(nodo->sig != NULL)
{
    /* Leer elemento de la lista */

    if(VUsuario.permisos)
    {
        sprintf(cadena, "%5s", nodo->modo&0777);
    }
}

```

```

        sprintf(cadena, "%s  %-25s", cadena, nodo->archivo);
    }
    else
        sprintf(cadena, "%-25s", nodo->archivo);
    if(VUsuario.longitud)
        sprintf(cadena, "%s  %8d bytes", cadena, nodo->longitud);
    if(VUsuario.fecha)
        sprintf(cadena, "%s  %s", cadena, asctime(nodo->fecha));
    /* Desplegar elemento en la pantalla */
    entrada = XmStringCreateLocalized(cadena);
    XmListAddItemUnselected(Archivo, entrada, 0);
    nodo = nodo->sig;
}
free(cadena);
}

```

1.3.3 Obtener lista directorio

```

char ObtenerListaDir(directorio)
char *directorio;
{
    . . .

    /* Abrir directorio */

    if( NULL == (dir = opendir(directorio)) )
    {

```



```

MensajeError(Shell, "Error al abrir el directorio");

return(ERROR);

}

. . .

/* Repetir hasta (leer todos los elementos directorios) */
while((a = readdir(dir)) != NULL)
{
    /* Leer elemento directorios */
    sprintf(cadena, "%s/%s", directorio, a->d_name);
    if(-1 == stat(cadena, &buf))
        MensajeError(Shell, sys_errlist[errno]);
    if( (buf.st_mode & S_IFMT) == S_IFDIR )/** DIRECTORIO ***/
    {
        /* Insertar datos del elemento directorio en la lista de archivos */
        strcpy(listad->archivo, a->d_name);
        listad->longitud = (int)buf.st_size;
        listad->modo = buf.st_mode;
        listad->elemento = elementoD;
        if(NULL==(listad->sig=(ListaArchivos*)malloc(sizeof(ListaArchivos))))
            MensajeError(Shell, "Error al asignar memoria");
        ld = listad;
        listad = listad->sig;
        listad->ant = ld;

        listad->sig = NULL;
    }
}

```

```

    }

    . . .

    /* Cerrar directorio */
    if( -1 == closedir(dir))
    {
        MensajeError(Shell, "Error al cerrar el directorio");
        return(ERROR);
    }

    return(EXITO);
}

```

1.3.4 Despliega información directorio

```

char PresentaLista(Widget widget, ListaArchivos *inicio)
{
    XmString entrada;
    ListaArchivos *nod;
    char *cadena;
    nod = inicio;
    . . .

    /* repetir hasta (lista = fin) */
    while(nod->sig != NULL)
    {
        /* Leer elemento de la lista */
        sprintf(cadena, "%-25s", nod->archivo);

        /* Desplegar elemento en la pantalla */
        entrada = XmStringCreateLocalized(cadena);
    }
}

```

```

        XmlListAddItem(widget, entrada, 0);

        nod = nod->sig;
    }

    free(cadena);

    return(EXITO);
}

```

1.3.5 Ordenar lista archivos

```

void Quick(ListaArchivos *lb, ListaArchivos *ub, int tipo)
{
    ListaArchivos *j;

    if(lb->elemento < ub->elemento)
    {
        j = Particion(lb, ub, tipo);
        Quick(lb, j->ant, tipo);
        Quick(j->sig, ub, tipo);
    }
}

```

Bibliografía:

Shneiderman, Ben. 1992. Designing the user interface strategies for effective human-computer interaction. Ed. Addison-Wesley Publishing Company Inc. California, EUA. 51-86, 181-190, 278-299, 325-329, 439-464 pp.

Len Bass, Len y Coutaz, Joelle. 1991. Developing software for the user interface. Ed. Addison-Wesley Publishing Company.

Baecker, Ronald M. and Buxton, William A. S. 1987. Readings in human-computer interaction a multidisciplinary approach. Morgan Kaufmann Publishers, Inc. EUA. 190-210, 237-250, 286-295 pp.

Davis s. Williams. 1985. Sistemas operativos de la computación. Mexico. 344 pp.

Sommerville, Ian. 1988. Ingenieria de Software. Addison Wesley Iberoamericana. México. 14-27, 43-48, 73-99, 251- 257 pp.

Pressman, Roger S. 1988. Ingenieria del software. McGraw-hill. Mexico. 15-16, 185-190, 227-257 pp.

Heller, Dan & Ferguson, Paula M. 1994. Motif programming manual. O'Really & Associates, Inc. 6-20 pp.

Johnson. Eric F and Reichard, Kevin. 1993. Professional Graphics Programming in the X Window System. Mis Press a subsidiary of Henry Holf and Co., Inc. EUA.

Norton, Peter. 1993. Periféricos y accesorios para la IBM-PC, PS/2 y compatibles. Prentice-hall Hispanoamericana. S. A. 24-30 pp.

Norton, Peter. 1994. Toda la PC. 5ta edicion. Prentice-Hall Hispanoamericana. S. A. 215-224 pp.

Reiss, Levi y Radin, Joseph. 1993. Aplique X Windows McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. 1-17 pp.

Fairley, Richard E. 1988. Ingeniería de software. MacGraw-Hill. Mexico. 1-28, 32-38, 92-119, 133-135, 144-191 pp.

Gane, Chris and Sarson, Trish. 1979. Structured systems analysis. Prentice-Hall, Inc. New Jersey, EUA. 25-35 pp.

Glosario

3D. Tridimensional.

API. Interfaz para Programación de Aplicaciones.

Árbol de directorios. Un directorio puede contener otros directorios dentro de él (llamados subdirectorios). Los directorios y subdirectorios de un disco pueden ser imaginados en términos de una estructura conocida. Esta estructura es el árbol de directorios.

Arquitectura cliente/servidor. Define una relación entre el usuario de una estación de trabajo y un servidor posterior de archivos, impresión, comunicaciones u otro tipo de sistema proveedor de servicios.

Arrastrar y soltar (Drag and drop). Es una técnica del ratón en la cual se coloca el puntero sobre un elemento en el área de trabajo (desktop), se pulsa el botón del ratón para seleccionar un elemento y entonces se mantiene pulsado el botón del ratón mientras el elemento es movido. Cuando desee liberar un elemento o finalizar la selección, suelte el botón del ratón.

Ayuda sensible al contexto. Esta ayuda esta directamente relacionada con la situación de la ventana activa.

Barras deslizantes. Es una barra vertical u horizontal localizada en el lado derecho o en la parte inferior de una ventana, que le permite moverse dentro de esta.

Biblioteca de funciones. Una biblioteca básica de software de uso general que se suministra por algunos fabricantes para realizar trabajos de programación.

Caja de diálogo. Es un objeto de una interfaz de usuario que se utiliza para proporcionar información adicional y definir parámetros u opciones antes de ejecutar una orden.

Click del ratón. Es la acción de presionar y soltar el botón del ratón.

Críptico. Oscuro, enigmático.

Desktop. Es el área de la pantalla en la cual tienen lugar todas las operaciones de una interfaz gráfica de usuario basada en ventanas. El desktop ocupa toda la pantalla, y todas las ventanas e iconos aparecen sobre él.

Directorios. Una porción de la unidad de discos a la cual se le asigna un nombre. Se utilizan los directorios para separar los archivos en secciones manejables.

Disco magnéto-óptico. Utilizan medio magnético para almacenar información con ayuda del láser.

DOS. Sistema operativo en disco desarrollado por Microsoft para la Pc.

GUI. Acrónimo de Interfaz Gráfica de Usuario (por sus siglas en inglés).

Herramientas de programación. Son herramientas que ayudan y asisten en la programación, estas herramientas van desde las herramientas más elementales como ensambladores y depuradores, hasta compiladores y ligadores (*linked*), incluyendo editores interactivos de texto y sistemas de bases de datos, así como también lenguajes procesadores de diseño, analizadores de requisitos de programación y ambientes totales de desarrollo.

Instrumentación. Su objetivo principal es el escribir el código fuente y la documentación interna de modo que la concordancia del código con sus especificaciones sea fácil de verificar, y que se faciliten la depuración, pruebas y modificaciones.

Lenguaje C. Lenguaje de programación de alto nivel y algo estructurado. Fue desarrollado por Dennis Ritchie de los laboratorios At&T Bell.

List. Un widget de Motif que le permite elegir una opción de un grupo de acciones. Las "List" utilizan barras de desplazamiento cuando la lista de opciones disponible es demasiado larga.

Mandos. La parte de una palabra de instrucción que especifica la operación que se desea realizar.

Menús. Se utiliza en las interfaces de usuario para desplegar una lista de órdenes.

Minicomputadora. Estas unidades están caracterizadas por un rendimiento mas alto que el de las microcomputadoras. Tienen un conjunto de instrucciones de mayor contenido, precios mas altos y proliferación de lenguajes de programación, sistemas operativos y metodologías de operación en redes.

Multitarea. La capacidad de una computadora de realizar más de una operación a un tiempo.

OptionMenu. Es un *widget* que se maneja en las interfaces de usuario Motif para elegir una opción de un grupo de opciones.

Pensar en voz alta. Las sesiones de “pensar en voz alta” es una técnica donde el operador explica sus acciones a tiempo que las va ejecutando.

PushButton. Es un *widget* que se utiliza en las interfaces Motif para implementar un botón.

Software. Los elementos constituyentes del software son los programas, lenguajes y procedimientos de un sistema computacional.

Solaris. Sistema operativo basado en Unix de SunSoft que está diseñado para que se pueda ejecutar sobre diferentes configuraciones y diferentes plataformas informáticas.

Tablas de decisión. Se trata de una tabla específica de todas las contingencias que han de considerarse en la descripción de un problema, junto con las acciones que deben tomarse.

TCP/IP (Transmission Control Protocol/ Internet Protocol). Permite comunicarse entre varios sistemas interdependientes y de diferentes proveedores. Tiene fuerte capacidades de interconexión de red y sufre una oleada creciente de popularidad, principalmente debido a que tiene un desarrollo abierto.

Textedit. Es un elemento de las interfaces Motif que le permiten responder a una pregunta escribiendo o pegando caracteres en él.

Unix. Sistema operativo multitarea que soporta las redes interconectadas y los sistemas de archivos distribuidos. Fue desarrollado por Ken Thompson y Dennis Ritchie en los laboratorios AT&T Bell.

Windows. Interfaz gráfica de usuario basada en ventanas desarrollada por Microsoft.

