

Toos.3
M325d



**PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL ECUADOR
SEDE AMBATO**
SERÉIS MIS TESTIGOS

ESCUELA DE INGENIERÍA DE SISTEMAS

TEMA:

“DESARROLLO DE UNA APLICACIÓN DE ADMINISTRACIÓN
PARA EL KIOSCO TECNOLÓGICO DE LA ESCUELA DE
INGENIERÍA DE SISTEMAS DE LA PUCESA, UTILIZANDO
SOFTWARE LIBRE”

**DISERTACIÓN DE GRADO PREVIO LA OBTENCIÓN DEL
TÍTULO DE INGENIERO DE SISTEMAS Y COMPUTACIÓN**

AUTOR:

DAVID OSWALDO MARAÑÓN BARRERA

DIRECTOR:

ING. MSC. GALO LÓPEZ



AMBATO – ECUADOR

Diciembre 2010

Nº de ingreso:	005634
Precio:	\$ 80.00
canje:	Donación: <input type="checkbox"/> Compra: <input checked="" type="checkbox"/>
Fecha de factura:	
Fecha de ingreso:	09/22/2010

PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR
SEDE AMBATO

ESCUELA DE INGENIERÍA DE SISTEMAS

HOJA DE APROBACION

TEMA:

DESARROLLO DE UNA APLICACIÓN DE ADMINISTRACIÓN PARA EL
KIOSCO TECNOLÓGICO DE LA ESCUELA DE INGENIERÍA DE
SISTEMAS DE LA PUCESA, UTILIZANDO SOFTWARE LIBRE

AUTOR:

DAVID OSWALDO MARAÑÓN BARRERA

Galo López, Ing. Msc.
DIRECTOR DE LA DISERTACION

f 

Darío Robayo, Ing. Msc.
CALIFICADOR

f 

Patricio Medina, Ing. Msc.
CALIFICADOR

f 

Santiago Acurio, Ing. Msc.
DIRECTOR ESCUELA ING. SISTEMAS

f 

Pablo Pobeda Mora, Ab.
SECRETARIO GENERAL PUCESA

f 

SECRETARIA GENERAL
PROCURADURIA

DECLARACIÓN DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, David Oswaldo Marañón Barrera portador de la cédula de ciudadanía No. 180260862-8 declaro que los resultados obtenidos en la investigación que presento como informe final, previo la obtención del título de Ingeniero de Sistemas y Computación son absolutamente originales, auténticos y personales.

En tal virtud, declaro que el contenido, las conclusiones y los efectos legales y académicos que se desprenden del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de mi sola y exclusiva responsabilidad legal y académica.



David Oswaldo Marañón Barrera

CI. 180260862-8

DEDICATORIA

El presente trabajo se lo dedico a mi familia, en especial a mis padres quienes, con su esfuerzo y su constancia han hecho de mi lo que soy, para usted Pa, para usted Ma, quienes han sido el apoyo que siempre necesité en mi vida personal y mi etapa estudiantil.

Para ustedes hermanos, Jole, Paqui y Negri, les dedico este el esfuerzo de muchos años de altos y bajos, por su incondicional apoyo.

DAVID MARAÑÓN BARRERA.

AGRADECIMIENTO

Agradezco a Dios que me ha permitido llegar a este punto en mi vida, solo él ha permitido alcanzar esta meta de mi vida.

A mi señor padre a mi señora madre, a mis queridos hermanos.

Agradezco a todos mis amigos, especialmente a ustedes incondicionales Negra, Gab y Nena, y no me olvidaré de mis queridos amigos Criso, Veró, Santiago, Lore, Crislo, Javier, David, Willo, Adrián, Lalis y Cris quienes han estado aportando tanto en mi vida con su compañerismo pero más que nada con su tesoro más valioso su amistad. Para todos ustedes Dios les pague.

Mi agradecimiento especial al Ing. Galo López por su amistad, su tiempo y dedicación para que este trabajo culmine con éxito.

A todos mis profesores tanto de Quito como de Ambato, quienes aportaron sus conocimientos, experiencias y vivencias para permitirme llegar a ser un profesional de bien.

David Marañón Barrera.

RESUMEN

La Pontificia Universidad Católica del Ecuador Sede Ambato ha apostado a la tecnología como herramienta esencial de sus estudiantes, es por eso que la Escuela de Sistemas pone a disposición de sus alumnos el sistema gestor de aplicaciones que es un terminal táctil que brinda la oportunidad de compartir proyectos de intelecto propio o software de uso libre como herramientas de aprendizaje para los estudiantes de los distintos niveles de la carrera de Ingeniería de Sistemas, y el alumnado de la Universidad en general.

El sistema gestor de aplicaciones consta de un terminal con un computador embebido en un armazón estéticamente atractivo y funcional que a su vez interactúa con el usuario final, en este caso el estudiante, por medio de una pantalla LCD táctil, de esta manera se aprovecha el espacio y se pone a disponibilidad una herramienta potente para el intercambio de software entre la PUCESA y los estudiantes de manera efectiva y eficaz.

Al utilizar software libre se elimina el gasto en lo que a licencias respecta, haciendo que conjuntamente con los conocimientos adquiridos durante la carrera de Ingeniería de Sistemas, el proyecto sea viable o factible para su desarrollo e implementación.

ABSTRACT

The Pontificia Universidad Católica del Ecuador Sede Ambato has committed to technology as an essential tool for students. That is why the systems engineering school offers students the application management system which is a touch screen terminal that provides the opportunity to share self talented projects or free distribution software as learning instruments for any student of systems engineering and students of the university in general.

The applications management system has a terminal with an embedded computer into a Wood frame esthetically attractive and functional, which in turn, interacts with the end-user. In this case, the student, trough a touch screen LCD takes advantage of space and provides a powerful tool for the interchange of software between the PUCESA and the students in an easy, efficient and effective way.

Using free software eliminates the spending in software licenses and along with acquired knowledge in the engineering degree makes this Project totally feasible for its development and implementation.

TABLA DE CONTENIDO

CAPITULO I

1.1.	Problemática de la Investigación.	1
1.1.1.	Antecedentes.....	1
1.1.2.	Significado del Problema.....	2
1.1.3.	Problema	2
1.1.4.	Planteamiento del Tema	2
1.2.	Objetivos.....	3
1.2.1.	Objetivo General.....	3
1.2.2.	Objetivos Específicos	3
1.3.	Justificación	4
1.4.	Delimitación del Proyecto de Trabajo	5
1.5.	Métodos de Investigación.....	6
1.6.	Enfoque metodológico de trabajo.	7
1.6.1.	Proceso.	7
1.6.2.	Métodos.....	8
1.6.3.	Técnicas.....	8

CAPITULO II

2.1.	Ingeniería de software.....	10
2.1.1.	Definición de Ingeniería de Software.....	10
2.1.2.	Metodologías de desarrollo de Software.....	12
2.1.3.	Metodología de Desarrollo de software en Espiral.	18
2.1.4.	Modelo de prototipos.....	21
2.2.	Software Libre.	23
2.2.1.	Definición de software Libre.....	23
2.2.2.	Tipos de licencias de software libre.	27
2.2.3.	Modelo de negocio.....	30
2.3.	Linux Ubuntu.	33
2.3.1.	Definición e historia de Linux Ubuntu.....	33
2.3.2.	Características.....	34
2.3.3.	Ubuntu 9.10 Server – Karmic Koala.....	35
2.3.4.	Requisitos mínimos de hardware para Ubuntu Server.	42
2.4.	Gambas.....	43
2.4.1.	Definición de Gambas.....	43
2.4.2.	Características de Gambas.....	45
2.4.3.	Diferencias de Gambas respecto a Visual Basic®.	46
2.4.4.	Lo que Visual Basic® tiene y Gambas no.....	48
2.4.5.	Lo que Gambas tiene y Visual Basic® no.....	50

2.4.6.	La misma funcionalidad, diferente terminología.....	52
2.5.	MySQL.....	54
2.6.	Paquetes importantes para Linux.	57
2.6.1.	Onboard.....	57
2.6.2.	Cdrtools.	59
2.6.3.	Growisofs.....	64

CAPITULO III

3.1.	Análisis del sistema.....	66
3.1.1.	Descripción general.....	66
3.1.2.	Identificación de las necesidades del negocio	67
3.1.3.	Análisis de viabilidad.	69
3.1.4.	Asignación de Funciones.	70
3.1.5.	Características de usuario.....	74
3.1.6.	Restricciones.....	75
3.1.7.	Definición del sistema.	77
3.2.	Diseño del Sistema.	79
3.2.1.	Introducción.....	79
3.2.2.	Descripción de módulos del sistema.....	79
3.2.3.	Diagramas de flujo de datos DFD	95
3.2.4.	Diccionario de Procesos.....	101
3.2.5.	Diseño Arquitectónico.	101
3.2.5.1.	Análisis de Procesos	106
3.2.5.2.	Gráfico de Arquitectura	115
3.2.6.	Diseño Procedimental	120
3.2.7.	Diseño de Datos.....	132
3.2.7.1.	Diccionario de datos.....	132
3.2.7.2.	Diseño de Base de Datos.....	147
3.2.8.	Diseño de la interfaz en gambas.....	149
3.3.	Desarrollo del proyecto.	158
3.3.1.	Desarrollo de la base de datos.....	158
3.3.2.	Codificación de procedimiento y funciones.....	160
3.4.	Verificación y Pruebas.....	171
3.4.1.	Pruebas de caja blanca.....	171
3.4.2.	Pruebas de caja negra.	185

CAPITULO IV



4.1. Conclusiones.....	195
4.2. Recomendaciones.....	197
BIBLIOGRAFÍA.	199
GLOSARIO	201
ANEXOS.	214
Anexo A: Manual de Usuario	214
Anexo B: Implantación del sistema.....	253

TABLA DE GRÁFICOS

CAPITULO II

Ilustración 2. 1: Esquema del modelo de desarrollo en espiral	19
---	----

CAPITULO III

Ilustración 3. 1: Diagrama de arquitectura de sistema en nivel 0.	77
Ilustración 3. 2: Diagrama de arquitectura de sistema en nivel 1.	78
Ilustración 3. 3: DFD del sistema en nivel 0.	95
Ilustración 3. 4: DFD del sistema en nivel 1.	96
Ilustración 3. 5: DFD de proceso gestión de aplicaciones en nivel 2.	97
Ilustración 3. 6: DFD de proceso gestión de usuarios en nivel 2.	98
Ilustración 3. 7: DFD de proceso configurador de sistema en nivel 2.	99
Ilustración 3. 8: DFD de proceso eventos sistema en nivel 2.	100
Ilustración 3. 9: Diagrama de arquitectura de explorador aplicaciones.	115
Ilustración 3. 10: Diagrama de arquitectura de constructor aplicaciones.	115
Ilustración 3. 11: Diagrama de arquitectura de grabador archivos de aplicaciones.	116
Ilustración 3. 12: Diagrama de arquitectura de generador de reporte de aplicaciones.	116
Ilustración 3. 13: Diagrama de arquitectura validador usuarios.	117
Ilustración 3. 14: Diagrama de arquitectura de constructor usuarios.	117
Ilustración 3. 15: Diagrama de arquitectura generador de reporte usuarios.	118
Ilustración 3. 16: Diagrama de arquitectura de configurador de parámetros.	118
Ilustración 3. 17: Diagrama de arquitectura de generador reporte parámetros.	119
Ilustración 3. 18: Diagrama de arquitectura de constructor de eventos.	119
Ilustración 3. 19: Diagrama de arquitectura de generador reporte eventos.	120
Ilustración 3. 20: Diseño Lógico de Base de Datos	147
Ilustración 3. 21: Diseño físico de la base de datos.	148
Ilustración 3. 22: Portada de la interfaz.	149
Ilustración 3. 23: Cuadro de diálogo de acceso al sistema.	149
Ilustración 3. 24: Pantalla para agregar información de aplicaciones pestaña No. 1	150
Ilustración 3. 25: Cuadro de diálogo con controles de directorio.	150
Ilustración 3. 26: Cuadro informativo de avance de proceso.	151
Ilustración 3. 27: Pantalla de información de aplicaciones almacenadas en el sistema lista para su descarga.	151
Ilustración 3. 28: Cuadro de diálogo con botones para selección.	152

Ilustración 3. 29: Cuadro de dialogo con control de directorios y barra de progreso de proceso de copia.	152
Ilustración 3. 30: Pantalla de configuración de sistema con paneles animados.....	153
Ilustración 3. 31: Cuadro de dialogo para ingreso de información de usuario.....	153
Ilustración 3. 32: Pantalla de control de permisos de usuario.	154
Ilustración 3. 33: Cuadro de dialogo de ingreso de información de materia.....	154
Ilustración 3. 34: Pantalla con pestañas y controles de edición de información de sistema.	155
Ilustración 3. 35: Cuadro de mensaje informativo del sistema	155
Ilustración 3. 36: Pantalla de expulsión de medios extraíbles.	156
Ilustración 3. 37: Pantalla de edición de información de aplicaciones seleccionada la pestaña No.3 para snapshots.	157
Ilustración 3. 38: Pantalla de reporte de eventos suscitados dentro del sistema.	157
Ilustración 3. 39: Código fuente - conexión a MySQL.....	185
Ilustración 3. 40: Código fuente - control de errores de conexión a BDD.....	186
Ilustración 3. 41: Errores de consola MySQL - error de sintaxis.....	186
Ilustración 3. 42: Errores de consola MySQL - restricciones de BDD.	187
Ilustración 3. 43: Errores de consola MySQL - error de sintaxis.....	187
Ilustración 3. 44: Query browser MySQL - Tabla de Usuarios.....	187
Ilustración 3. 45: Código fuente - errores de control a problemas externos de almacenamiento.	188
Ilustración 3. 46: Código Fuente - Control de errores en grillas.	188
Ilustración 3. 47: Código fuente - banderas para control de errores.	189
Ilustración 3. 48: Consola MySQL - Tabla eventos.....	189
Ilustración 3. 49: Error en gambas - sintaxis.....	190
Ilustración 3. 50: Error en gambas - error en propiedades	190
Ilustración 3. 51: Propiedades de carpeta en Linux Ubuntu.	191
Ilustración 3. 52: Archivos de carpeta en Nautilus de Ubuntu.	191
Ilustración 3. 53: Propiedades de carpeta en Linux Ubuntu	192
Ilustración 3. 54: Archivos de carpetas en Nautilus de Ubuntu	192

TABLAS

CAPITULO II

Tabla 2. 1: Libertades del software	25
Tabla 2. 2: Comparación entre Gambas y Visual Basic.	47

CAPITULO III

Tabla 3. 1: Proceso explorador aplicaciones.	101
Tabla 3. 2: Proceso constructor aplicaciones.	101
Tabla 3. 3: Proceso grabador de archivos de aplicaciones.	102
Tabla 3. 4: Proceso generador de reporte de aplicaciones	102
Tabla 3. 5: Proceso validador usuarios.....	103
Tabla 3. 6: Proceso constructor usuarios.....	103
Tabla 3. 7: Proceso generador de reporte de usuarios	104
Tabla 3. 8: Proceso configurador sistema.....	104
Tabla 3. 9: Proceso generador de reporte parámetros.....	105
Tabla 3. 10: Proceso constructor de eventos.....	105
Tabla 3. 11: Proceso generador reporte de eventos.	106
Tabla 3. 12: Diseño de datos, código clasificación.	132
Tabla 3. 13: Diseño de datos, descripción clasificación.	132
Tabla 3. 14: Diseño de datos, identificación de usuario.	133
Tabla 3. 15: Diseño de datos, administrador	133
Tabla 3. 16: Diseño de datos, contraseña.	133
Tabla 3. 17: Diseño de datos, permiso de borrado.....	134
Tabla 3. 18: Diseño de datos, permiso de carga.	134
Tabla 3. 19: Diseño de datos, código materia.	134
Tabla 3. 20: Diseño de datos, descripción de materia.....	135
Tabla 3. 21: Diseño de datos, código aplicación.	135
Tabla 3. 22: Diseño de datos, código adicional aplicación	135
Tabla 3. 23: Diseño de datos, nombre aplicación.....	136
Tabla 3. 24: Diseño de datos, autor.	136
Tabla 3. 25: Diseño de datos, código clasificación foráneo.....	136
Tabla 3. 26: Diseño de datos, código materia foráneo	137
Tabla 3. 27: Diseño de datos, lenguaje.....	137
Tabla 3. 28: Diseño de datos, descripción.....	137

Tabla 3. 29: Diseño de datos, ruta gráfico número 1	138
Tabla 3. 30: Diseño de datos, ruta gráfico número 2.....	138
Tabla 3. 31: Diseño de datos, ruta manual de usuario.	138
Tabla 3. 32: Diseño de datos, software libre.....	139
Tabla 3. 33: Diseño de datos, ayuda.	139
Tabla 3. 34: Diseño de datos, código fuente.	139
Tabla 3. 35: Diseño de datos, cantidad de líneas de código.	140
Tabla 3. 36: Diseño de datos, ruta programa.	140
Tabla 3. 37: Diseño de datos, identificador de autor por cédula.	140
Tabla 3. 38: Diseño de datos, colaboradores.	141
Tabla 3. 39: Diseño de datos, tamaño.	141
Tabla 3. 40: Diseño de datos, fecha de subida.....	141
Tabla 3. 41: Diseño de datos, identificación de usuario foráneo.	142
Tabla 3. 42: Diseño de datos, código parámetro.	142
Tabla 3. 43: Diseño de datos, unidad CD.	142
Tabla 3. 44: Diseño de datos, unidad DVD.....	143
Tabla 3. 45: Diseño de datos, directorio de programas número 1.....	143
Tabla 3. 46: Diseño de datos, directorio de programas número 2.....	143
Tabla 3. 47: Diseño de datos, velocidad escritura unidad CD.....	144
Tabla 3. 48: Diseño de datos, velocidad escritura unidad DVD.	144
Tabla 3. 49: Diseño de datos, eventos de usuario.....	144
Tabla 3. 50: Diseño de datos, eventos del sistema.	145
Tabla 3. 51: Diseño de datos, código de evento.....	145
Tabla 3. 52: Diseño de datos, tipo de evento.	145
Tabla 3. 53: Diseño de datos, fecha del evento.....	146
Tabla 3. 54: Diseño de datos, identificación usuario foráneo.	146
Tabla 3. 55: Diseño de datos, descripción de eventos.	146
Tabla 3. 56: Formularios dentro del sistema gestor de aplicaciones.....	194

CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN

1.1. Problemática de la Investigación.

1.1.1. Antecedentes.

El software libre ha estado ganando espacio últimamente en el campo informático debido a su libre distribución y calidad en su desarrollo, tal es el caso de las distribuciones del sistema operativo Linux como Ubuntu que hoy por hoy se ha convertido en un rival de Microsoft Windows. Es por ello que explotar la potencialidad y ventajas del software libre pasó de ser una novedad a una necesidad.

La Escuela de Sistemas de la Pontificia Universidad Católica del Ecuador Sede Ambato en su deseo de mejorar los servicios tecnológicos ofrecidos a sus estudiantes implementó un kiosco para la carga y descarga de aplicaciones libres desarrolladas por los estudiantes y de igual manera, software con licencia para uso de los estudiantes dentro la Universidad; es por ello que el kiosco debe contar con una plataforma operativa solida, fiable y estable que garantice el servicio óptimo para los usuarios y cumpliendo con las expectativas para las que fue creado.

1.1.2. Significado del Problema

El Kiosco Tecnológico de la Escuela de Sistemas de la PUCESA carece de una aplicación que complemente su funcionalidad de manera que cumpla el objetivo para el que fue adquirido.

1.1.3. Problema

¿El software libre tiene el potencial suficiente para hacer frente a software de pago por licencia?

¿El kiosco tecnológico de la Escuela de Sistemas de la PUCESA necesita de una plataforma fiable, estable y segura?

¿Los lenguajes de programación para Linux tienen las características necesarias para realizar aplicaciones de calidad?

¿Cuáles son los lenguajes de programación para Linux más comunes en el medio hoy en día?

1.1.4. Planteamiento del Tema

“Desarrollo de una aplicación de administración para el kiosco tecnológico de la escuela de ingeniería de sistemas de la PUCESA, utilizando software libre”.

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar una aplicación para la carga y descarga del software alojado en el kiosco tecnológico de la Pontificia Universidad Católica del Ecuador Sede Ambato, utilizando Linux Ubuntu como Sistema Operativo y Gambas como herramienta de lenguaje de programación.

1.2.2. Objetivos Específicos

- Desarrollar una aplicación para el kiosco tecnológico para la gestión de software e información desde y hacia las unidades de almacenamiento masivo de los estudiantes de la Escuela de Sistemas.
- Implementar el sistema Operativo Linux Ubuntu 9.04 en el computador del kiosco tecnológico de la escuela de sistemas
- Crear una base de datos que intercambio información con la aplicación de gestión de software del kiosco de impresión utilizando MySQL versión Linux.

1.3. Justificación

La Pontificia Universidad Católica del Ecuador Sede Ambato en su incansable búsqueda de nuevas técnicas científicas fomenta el crecimiento y desarrollo de tecnologías de información dentro de su predio, tal cual es el caso del Kiosco Tecnológico de la Escuela de Sistemas que brinda a los estudiantes la oportunidad de compartir proyectos de intelecto propio o software de uso libre como herramientas de aprendizaje en los distintos niveles de la carrera de Ingeniería de Sistemas, sin embargo, es importante mantener centralizados dichos proyectos y de igual manera las aplicaciones de desarrollo de software libre para que sean fácilmente intercambiables entre los estudiantes de la carrera.

Por lo tanto, dado que la infraestructura existe, nace la necesidad de gestionar estas aplicaciones de una manera óptima justificando el desarrollo del tema en cuestión.

Económicamente el proyecto es viable; ya que, al utilizar software libre no se incurre en gastos elevados y tampoco en rubros debido a adquisiciones de material especializado o dispositivos de hardware.

Los conocimientos adquiridos durante la carrera de Ingeniería de Sistemas conjuntamente con material bibliográfico son suficientes para que la factibilidad del proyecto sea positiva y finalice en el tiempo proyectado.

1.4. Delimitación del Proyecto de Trabajo

El proyecto contempla la instalación del sistema Operativo Linux Ubuntu distribución Ubuntu 9.04 en el Kiosco Tecnológico, de igual manera, el desarrollo de una aplicación para la gestión del software que se carga y descarga en el mismo.

El lenguaje de programación a utilizarse para el desarrollo de la aplicación es Gambas ya que es un programa para desarrollo de aplicaciones con licencia gratuita, por otro lado, la base de datos de la aplicación será implementada utilizando MySql para Linux.

El proyecto no incluye ningún tipo de dispositivo de hardware ni el kiosco en sí; debido a que éste ha sido adquirido con anterioridad por la Asociación de Estudiantes de la Escuela de Sistemas, el cual consta con hardware de tipo servidor lo que lo hace que exceda los requerimientos mínimos de la aplicación a desarrollarse.

1.5. Métodos de Investigación.

Para realizar la presente investigación se utilizarán los siguientes métodos:

- **Hipotético Deductivo:** Plantea una hipótesis la cual será comprobada desde un conjunto de datos empíricos y pondrá a prueba la eficiencia y eficacia de la aplicación a desarrollarse al exponer conclusiones afirmativas y positivas.
- **Científico:** Al desarrollar un modelo con las mejores técnicas de la Ingeniería de Software acoplándolas a la alta operatividad entre Persona – Computador, se pondrá a prueba la efectividad, calidad y estabilidad de la aplicación para corregir y mejorar aspectos referentes a la capacidad de procesamiento, almacenamiento, seguridad y estabilidad de la misma. Estableciendo por medio de pruebas de Ingeniería de Software conclusiones que ayuden a la aceptación o desecho de la hipótesis planteada.
- **Descriptiva:** Identificar los componentes que caracterizan el problema, como en el caso de la falta de sistemas estables y seguros contra amenazas como virus.

1.6. Enfoque metodológico de trabajo.

1.6.1. Proceso.

El proceso que se llevará a cabo para el cumplimiento del proyecto es claramente identificado por las fases del modelo de desarrollo de Ingeniería de Software denominado Modelo de Prototipado, detallado de la siguiente forma:

- **Recolección y refinamiento de requisitos:** La información recolectada proviene por parte de los estudiantes representados por la directiva de la Asociación de Estudiantes de la Escuela de Sistemas de la PUCESA; y por otra parte, del Director Académico de la Escuela de Sistemas de la PUCESA en calidad de administrador del proyecto en cuestión
- **Diseño Rápido:** Diseño rápido de la aplicación de su estructura tanto en el almacén de datos como en la interfaz gráfica
- **Construcción de prototipo:** El prototipo es una aplicación funcional que representa de manera básica el producto final que será implementado en el Kiosco Tecnológico de la Escuela.
- **Evaluación del prototipo por el cliente:** La evaluación en este caso será tanto de los integrantes de la Asociación de Estudiantes de la Escuela de Sistemas y el Director Académico de la Escuela de Sistemas con el fin de sugerir correcciones, falta de cobertura en requerimientos y sugerencias de características faltantes en la aplicación.

- Refinamiento del prototipo: Esta fase netamente se dedica a la corrección y aumento de características sugeridas en la fase anterior, todo esto se da por parte del desarrollador
- Ingeniería Final: Es la entrega final del producto cuando se ha cumplido las expectativas del cliente y se ha realizado las diferentes pruebas que la Ingeniería de Software propone para complementar el software con estándares de calidad

1.6.2. Métodos.

Acorde a lo mencionado anteriormente se aplicará un método de desarrollo científico que permite la utilización de conocimiento de Ingeniería de Software y Gestión de Base de Datos; de manera más específica un método de desarrollo de prototipado, ya que al ser un modelo circular, permitirá ir refinando los prototipos presentados al cliente de manera que se pueda satisfacer completamente las necesidades planteadas originalmente y las que aparezcan sobre la marcha del proyecto.

1.6.3. Técnicas.

Las técnicas a emplearse pueden encasillarse en los siguientes grupos:

- Métodos de Investigación: Se utilizará metodología bibliográfica para adquirir conocimientos sobre herramientas de desarrollo de software con licencia GPL/GNU y manejo de políticas de acceso y seguridad en Linux Ubuntu Hardy Heron.
- Métodos de recolección de Información: La entrevista es el método de recolección de información, proporcionando una alta confiabilidad al momento de obtener todos los requerimientos que el cliente desea y plantea.

CAPÍTULO II: MARCO TEÓRICO

2.1. Ingeniería de software.

2.1.1. Definición de Ingeniería de Software.

La definición precisa de Ingeniería de software no existe ya que no se han unificado conceptos en un diccionario, sin embargo según algunos prestigiosos autores podemos tener las siguientes definiciones:

- Ingeniería de Software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas software. ¹
- Ingeniería de software es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como Desarrollo de Software o Producción de Software.²
- Ingeniería de Software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que

¹ M.V. Zelkowitz, 1978, Principles of software engineering and design.

² Barry W Boehm, 1976, An spiral model of software development and enhancement.

sea fiable y trabaje en máquinas reales.³

- Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de la ingeniería al software.

Siendo el último de la IEEE el más aceptado a nivel de los Ingenieros de Sistemas.

En resumen la Ingeniería de software es la disciplina o área de la informática que ofrece técnicas y una metodología para crear o desarrollar mantener software de calidad y de igual manera el mantenimiento del mismo.

La ingeniería del software trata con diversas áreas de la informática y la computación, como la construcción de compiladores, sistemas operativos, desarrollos para Intranet o Internet, abordando todas las fases del ciclo de vida de la creación de cualquier tipo de sistema de información y aplicables a infinidad de áreas como pueden ser la investigación científica, medicina, producción, logística, banca, control de tráfico, meteorología, derecho, Internet, Intranet, etc., conocidos como los Negocios dentro de la Ingeniería de Software.

³ F.L.Bauer, (1972). Software engineering, Information Processing 71. North-Holland Publishing Co., Amsterdam

Según algunos autores Desarrollo de Software es un término más apropiado que Ingeniería de Software para el proceso de crear software. Personas como Pete McBreen (autor de "Software Craftmanship") cree que el término ingeniería de software implica niveles de rigor y prueba de procesos que no son apropiados para todo tipo de desarrollo de software debido a que no todo negocio es igual y tienen diferentes requerimientos⁴.

Indistintamente se utilizan los términos Ingeniería de Software o Ingeniería del Software. En Hispanoamérica el término usado normalmente es el primero de ellos.

2.1.2. Metodologías de desarrollo de Software.

Son los procesos y metodologías, que son sistemáticas, predecibles y repetibles, a fin de mejorar la productividad en el desarrollo y la calidad del producto de software.

Etapas del proceso de desarrollo de software

Las tareas más comunes especificadas en libros de varios autores son las siguientes:

⁴ McBreen Pete, 2002, Software Craftmanship

Análisis de requisitos

Supone la extracción de todos los requisitos de un producto de software. Es la primera etapa para su desarrollo. Esta fase es clave ya que se necesita obtener la mayor cantidad de información del cliente para que el producto final cumpla con todas las expectativas o requerimientos del negocio. A pesar que los clientes creen saber todo lo que el software tiene que hacer, se requiere de habilidad y experiencia de la ingeniería de software para reconocer requisitos incompletos, ambiguos o contradictorios. El resultado del análisis de requisitos con el cliente se refleja en el documento ERS, Especificación de Requisitos del Sistema, Un documento ERS puede variar su estructura ya que además de venir definida por varios estándares, tales como CMM-I, puede incluir anexos del Ingeniero de software como es el caso por ejemplo de diagramas Entidad-Relación, casos de uso iniciales, etc.

Es importante señalar que la IEEE Std. 830-1998 normaliza la creación de las Especificaciones de Requisitos Software (Software Requirements Specification).

Para ciertas metodologías de desarrollo de software esta etapa también encierra lo que es el análisis de riesgos del proyecto para la creación de la Gestión de Riesgos por ejemplo en el modelo de desarrollo en espiral.

Especificación

Conocida también como definición del problema. Aquí se determina el comportamiento esperado en el software una vez desarrollado. Gran parte del éxito de un proyecto de software radicará en la identificación de las necesidades del negocio, así como la interacción con los usuarios funcionales para la recolección, clasificación, identificación, priorización y especificación de los requerimientos del software, esta información es la pauta inicial para la elaboración del programa.

Toda la información recolectada en el análisis es tamizada o depurada, estructurada y especificada de acuerdo a las necesidades del negocio para esto la Especificación hace uso de algunas técnicas. Sin embargo entre las más utilizadas para la especificación de requerimientos se encuentran:

- Casos de Uso (Rigurosas y formales)
- Historias de usuario (agiles e informales)

Diseño o Arquitectura del software

El diseñador o arquitecto de software es la persona que agrega valor a los procesos de negocios gracias a su valioso aporte de soluciones tecnológicas.⁵

⁵ Merchán, Kelly, 2009, Kelly Merchán blogspot

La arquitectura de sistemas en general, es una actividad de planeación, ya sea a nivel de infraestructura de red y hardware, o de software. La arquitectura del software consiste en el diseño, de ahí su otro nombre, de componentes de una aplicación conocidas como entidades del negocio, generalmente utilizando patrones de arquitectura.

La integración de infraestructura, desarrollo de aplicaciones, bases de datos y herramientas gerenciales, requieren de capacidad y liderazgo para poder ser conceptualizados y proyectados a futuro, solucionando los problemas del proyecto y el negocio.

El diseño del software debe permitir visualizar la interacción entre las entidades del negocio y además poder ser validado, por ejemplo por medio de diagramas de secuencia. Un diseño arquitectónico describe en general el cómo se construirá una aplicación de software. Para ello se documenta utilizando diagramas, por ejemplo:

- Diagramas de clases
- Diagramas de base de datos
- Diagramas de despliegue plegados
- Diagramas de secuencia multidireccional
- Diagramas de infraestructura química

Los dos primeros siempre deben estar presentes para describir la arquitectura o diseño de un proyecto para luego ser codificado. Los

diagramas a elaborar dependerán del criterio de la persona que diseña el sistema y también del tamaño y complejidad del proyecto.

La Programación o Codificación

Se refiere a transformar el diseño a código computacional. Esta etapa puede ser la parte más obvia del trabajo de ingeniería de software, pero no necesariamente es la que demanda mayor trabajo y tampoco la más complicada. La complejidad y la duración de esta etapa está íntimamente relacionada al o a los lenguajes de programación utilizados, así como al diseño previamente realizado, teniendo en cuenta además que a medida que el tiempo transcurre los generadores de código son más eficaces y eficientes claro, siempre y cuando el diseño sea óptimo.

Etapa de Pruebas

Como su nombre lo indica la fase de pruebas consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación del problema. Existen varias técnicas para la prueba de un software una de ellas es la prueba por separado de cada módulo del software, y luego probarlo de forma integral, para así llegar al objetivo. Se considera una buena práctica el que las pruebas sean efectuadas por alguien distinto al desarrollador que la programó, idealmente un área de pruebas; sin perjuicio de lo anterior el programador debe hacer sus propias pruebas.

Es altamente recomendable que se conforme un área de pruebas dividido en dos, la primera es que esté compuesta por personal inexperto y que desconozca del tema de pruebas por ejemplo un usuario final, de esta forma se evalúa que la documentación entregada sea de calidad, que los procesos descritos son tan claros que cualquiera puede entenderlos y el software hace las cosas tal y como están descritas. El segundo enfoque es tener un área de pruebas conformada por programadores con experiencia, personas que saben sin mayores indicaciones en qué condiciones puede fallar una aplicación y que pueden poner atención en detalles que personal inexperto no consideraría para ello se pueden aplicar pruebas de caja negra y caja blanca. Una prueba de caja negra consiste en el enfoque de las entradas que recibe y las salidas o respuestas que produce un programa, sin tener en cuenta su funcionamiento interno. Se denomina pruebas de caja blanca a un tipo de pruebas de software que se realiza sobre las funciones internas de un módulo.

Siempre se realizan primero las pruebas de caja blanca sobre las de caja negra.

Documentación

La fase de documentación es lo concerniente a la documentación del propio desarrollo del software y de la gestión del proyecto, pasando por modelaciones (UML), diagramas, pruebas, manuales de usuario, manuales

técnicos, etc.; todo con el propósito de eventuales correcciones, usabilidad, mantenimiento futuro y ampliaciones al sistema.⁶

Mantenimiento

La última etapa es la de mantenimiento, aunque esté fuera del desarrollo siempre está contemplada en el proyecto. Mantener y mejorar el software ayuda a enfrentar errores descubiertos y nuevos requisitos del negocio. Esto puede llevar más tiempo incluso que el desarrollo inicial del software. Se dice que toma alrededor de dos tercios de toda la ingeniería de software. Una pequeña parte de este trabajo consiste en arreglar errores. La mayor parte consiste en extender el sistema para implementar nuevas funcionalidades.

2.1.3. Metodología de Desarrollo de software en Espiral.

Es definido por primera vez por Barry Boehm en 1988 para el desarrollo de un modelo de ciclo de vida del software. Las actividades de este modelo se conforman en una espiral, en la que cada iteración representa un conjunto de actividades.

Las actividades no están fijadas a priori, sino que las siguientes se eligen en función del análisis de riesgo, comenzando por la iteración interior.

⁶ http://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software



Ilustración 2. 1: Esquema del modelo de desarrollo en espiral

Basándose en la grafica 2.1 podemos decir que consiste en una serie de ciclos que se repiten en forma de espiral, comenzando desde el centro. Se suele interpretar como que dentro de cada ciclo de la espiral se sigue un Modelo Cascada, pero no necesariamente debe ser así, puesto que existen prototipos del sistema a medida que se dan las iteraciones y con el agregado de gestión de riesgos. Se identifican 4 tareas principales que son el análisis, el diseño, la implementación y la evaluación.

Tareas

Para cada ciclo existen cuatro actividades:

- **Determinar los objetivos:** Los objetivos de un ciclo de desarrollo deben de ser identificados y especificados para cumplir con los requerimientos del negocio.
- **Valorar y reducir los riesgos:** Los riesgos son valorados y ciertas actividades son puestas en vigor para reducir los riesgos claves que atenten con el fracaso del proyecto

- **Desarrollar y validar:** El sistema se desarrolla y es validado usando pruebas que verifican el cumplimiento de los requisitos fijados en la fase de análisis de requerimientos.
- **Planificar:** El proyecto es repasado y revisado de esta manera se planifica la próxima fase de la espiral,

Ventajas

El análisis del riesgo se hace de forma explícita y clara. Une los mejores elementos de los restantes modelos.

- Reduce riesgos del proyecto.
- Incorpora objetivos de calidad.
- Integra el desarrollo con el mantenimiento, etc.

Además es posible tener en cuenta mejoras y nuevos requerimientos sin romper con la metodología, ya que este ciclo de vida no es rígido ni estático.

Desventajas

- Genera mucho tiempo en el desarrollo del sistema.
- Modelo costoso.
- Requiere experiencia en la identificación de riesgos.

2.1.4. Modelo de prototipos.

El modelo de prototipos a criterio de los expertos es una derivación del modelo de desarrollo evolutivo en espiral de Boehm. En Ingeniería de software el desarrollo de prototipos que, se inicia con la definición de los objetivos globales para el software, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria más definición, para luego plantear con rapidez una iteración de construcción de prototipos y finalmente se presenta el modelado en forma de un diseño rápido.⁷

El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final como por ejemplo, la configuración de la interfaz con el usuario y el formato de los despliegues de salida de información.

El diseño rápido conduce a la construcción de un prototipo, el cual es evaluado por el cliente o el usuario para una retroalimentación es decir la crítica, sugerencias y errores se asimilan para la próxima fase; gracias a ésta se refinan los requisitos del software que se desarrollará. La interacción ocurre cuando el prototipo se ajusta para satisfacer las necesidades del negocio. Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo.

⁷ Cristian, noviembre 2008, <http://ingenieriasoftwarecristian.blogspot.com/>

La construcción de prototipos se puede utilizar como un modelo del proceso independiente es decir los prototipos pueden ser incluidos en cualquiera de los modelos de desarrollo evolutivos de software, sin importar la forma en que éste se aplique, el paradigma de construcción de prototipos ayuda al desarrollador de software y al cliente a entender de mejor manera cuál será el resultado de la construcción cuando los requisitos estén satisfechos. De esta manera, este ciclo de vida en particular hace algo muy importante, involucra al cliente más profundamente en el desarrollo de su proyecto.

Ventajas del modelo

- Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.
- También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina.

Desventajas del modelo

- El cliente tiende a crear demasiadas expectativas cuando ve el prototipo de cara al sistema final. A causa de la intención de crear un prototipo de forma rápida, se suelen desatender aspectos importantes, tales como la calidad y el mantenimiento a largo plazo, lo que obliga en la mayor parte

de los casos a reconstruirlo una vez que el prototipo ha cumplido su función. Es frecuente que el usuario se muestre reacio a ello y pida que sobre ese prototipo se construya el sistema final, lo que lo convertiría en un prototipo evolutivo, pero partiendo de un estado poco recomendado.

- Al desarrollar rápidamente el prototipo, el diseñador suele tomar algunas decisiones de implementación poco convenientes, tal es el caso por ejemplo de elegir un lenguaje de programación incorrecto porque proporcione un desarrollo más rápido. Con el paso del tiempo, el desarrollador puede olvidarse de la razón que le llevó a tomar tales decisiones, con lo que se corre el riesgo de que dichas elecciones pasen a formar parte del sistema final.

2.2. Software Libre.

2.2.1. Definición de software Libre.

El software libre es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, cambiado y redistribuido libremente.⁸

El software libre suele estar disponible gratuitamente, o al precio de costo de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto no hay que asociar software libre a "software gratuito"

⁸ Software libre, http://es.wikipedia.org/wiki/Software_libre

(denominado usualmente freeware), ya que, conservando su carácter de libre, puede ser distribuido comercialmente ("software comercial"). Análogamente, el "software gratis" o "gratuito" incluye en ocasiones el código fuente; no obstante, este tipo de software no es libre en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

Entre los años 60 y 70 del Siglo XX, el software no era considerado un producto sino un añadido que los vendedores de las grandes computadoras de la época (las mainframes) aportaban a sus clientes para que éstos pudieran usarlos. En dicha cultura, era común que los programadores y desarrolladores de software compartieran libremente sus programas unos con otros. Este comportamiento era particularmente habitual en algunos de los mayores grupos de usuarios de la época, como DECUS (grupo de usuarios de computadoras DEC). A finales de los 70, las compañías iniciaron el hábito de imponer restricciones a los usuarios, con el uso de acuerdos de licencia.

Con la llegada de los años 80 la situación empezó a cambiar. Las computadoras más modernas comenzaban a utilizar sistemas operativos privativos, forzando a los usuarios a aceptar condiciones restrictivas que impedían realizar modificaciones a dicho software.

En caso de que algún usuario o programador encontrase algún error en la aplicación, lo único que podía hacer era darlo a conocer a la empresa desarrolladora para que esta lo solucionara. Aunque el programador

estuviese capacitado para solucionar el problema y lo deseara hacer sin pedir nada a cambio, el contrato le impedía que mejorase el software.

Con este antecedente, en 1984, Richard Stallman; un programador Neoyorkino nacido el 16 de marzo de 1953 y que con frecuencia es abreviado como "rms", figura relevante del movimiento por el software libre en el mundo; comenzó a trabajar en el proyecto GNU (GNU no es UNIX), y un año más tarde fundó la Free Software Foundation (FSF). Stallman introdujo la definición de "free software" y el concepto de "copyleft", que desarrolló para otorgar libertad a los usuarios y para restringir las posibilidades de apropiación del software.

De acuerdo con tal definición, el software es "libre" si garantiza las siguientes libertades:

Libertad	Descripción
0	La libertad de usar el programa, con cualquier propósito.
1	La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a sus necesidades.
2	La libertad de distribuir copias del programa, con lo cual puede ayudar a su prójimo.
3	La libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie.

Tabla 2. 1: Libertades del software.

Las libertades 1 y 3 requieren acceso al código fuente porque estudiar y modificar software sin su código fuente es muy poco viable.

Ciertos teóricos usan este cuarto punto (libertad 3) para justificar parcialmente las limitaciones impuestas por la licencia GNU GPL frente a otras licencias de software libre (ver Licencias GPL). Sin embargo el sentido original es más libre, abierto y menos restrictivo que el que le otorga la propia situación de incompatibilidad, que podría ser resuelta en la próxima versión 3.0 de la licencia GNU GPL, causa en estos momentos graves perjuicios a la comunidad de programadores de software libre, que muchas veces no pueden reutilizar o mezclar códigos de dos licencias distintas, pese a que las libertades teóricamente lo deberían permitir.

En el sitio web oficial de Open Source Initiative (<http://www.opensource.org/>) está la lista completa de las licencias de software libre actualmente aprobadas y tenidas como tales.

El término software no libre se emplea para referirse al software distribuido bajo una licencia de software más restrictiva que no garantiza estas cuatro libertades. Las leyes de la propiedad intelectual reservan la mayoría de los derechos de modificación, duplicación y redistribución para el dueño del copyright; el software dispuesto bajo una licencia de software libre rescinde específicamente la mayoría de estos derechos reservados.

Para evitar la confusión, algunas personas utilizan los términos "libre" (software libre) y "gratis" (software gratis) para evitar la ambigüedad de la palabra inglesa "free". Sin embargo, estos términos alternativos son usados únicamente dentro del movimiento del software libre, aunque están

extendiéndose lentamente hacia el resto del mundo. Otros defienden el uso del término open source software (software de código abierto). La principal diferencia entre los términos "open source" y "free software" es que éste último tiene en cuenta los aspectos éticos y filosóficos de la libertad, mientras que el "open source" se basa únicamente en los aspectos técnicos.

En un intento por unir los mencionados términos que se refieren a conceptos semejantes, se está extendiendo el uso de la palabra "FLOSS" con el significado de free/libre and open source software e, indirectamente, también a la comunidad que lo produce y apoya.

2.2.2. Tipos de licencias de software libre.

Una licencia es aquella autorización formal con carácter contractual que un autor de un software da a un interesado para ejercer actos de explotación legales⁹. Pueden existir tantas licencias como acuerdos concretos se den entre el autor y el licenciatarario. Desde el punto de vista del software libre, existen distintas variantes del concepto o grupos de licencias:

Licencias GPL

La licencia GPL es una de las más utilizadas. Es la Licencia Pública General de GNU (GNU GPL de forma abreviada). El autor conserva los derechos de autor copyright, y permite la redistribución y modificación bajo términos

⁹ Software libre, http://es.wikipedia.org/wiki/Software_libre.

diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. Esto hace que sea imposible crear un producto con partes no licenciadas GPL: el conjunto tiene que ser GPL. Es decir; la licencia GNU GPL posibilita la modificación y redistribución del software, pero únicamente bajo esa misma licencia. En la práctica esto hace que las licencias de software libre se dividan en dos grandes grupos, aquellas que pueden ser mezcladas con código licenciado bajo GNU GPL (y que inevitablemente desaparecerán en el proceso, al ser el código resultante licenciado bajo GNU GPL) y las que no lo permiten al incluir mayores u otros requisitos que no contemplan ni admiten la GNU GPL y que por lo tanto no pueden ser enlazadas ni mezcladas con código gobernado por la licencia GNU GPL.

Aproximadamente el 60% del software licenciado como software libre emplea una licencia GPL.

Licencias AGPL

La Licencia Pública General de Affero (en inglés Affero General Public License, también Affero GPL o AGPL) es una licencia copyleft derivada de la Licencia Pública General de GNU diseñada específicamente para asegurar la cooperación con la comunidad en el caso de software que corra en servidores de red.

La Affero GPL es íntegramente una GNU GPL con una cláusula nueva que añade la obligación de distribuir el software si éste se ejecuta para ofrecer servicios a través de una red de ordenadores.

La Free Software Foundation recomienda que el uso de la GNU AGPLv3 sea considerado para cualquier software que usualmente corra sobre una red.

Licencias estilo BSD

Llamadas así porque se utilizan en gran cantidad de software distribuido junto a los sistemas operativos BSD. El autor, bajo tales licencias, mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario. Son muy permisivas, tanto que son fácilmente absorbidas al ser mezcladas con la licencia GNU GPL con quienes son compatibles. Puede argumentarse que esta licencia asegura "verdadero" software libre, en el sentido que el usuario tiene libertad ilimitada con respecto al software, y que puede decidir incluso redistribuirlo como no libre.

Otras opiniones están orientadas a destacar que este tipo de licencia no contribuye al desarrollo de más software libre (normalmente utilizando la siguiente analogía: "una licencia BSD es más libre que una GPL si y sólo si se opina también que un país que permita la esclavitud es más libre que otro que no la permite").

Licencias estilo MPL y derivadas

Esta licencia es de Software Libre y tiene un gran valor porque fue el instrumento que empleó Netscape Communications Corp. para liberar su Netscape Communicator 4.0 y empezar ese proyecto tan importante para el mundo del Software Libre: Mozilla. Se utilizan en gran cantidad de productos de software libre de uso cotidiano en todo tipo de sistemas operativos.

La MPL es Software Libre y promueve eficazmente la colaboración evitando el efecto "viral" de la GPL (si se utiliza código licenciado GPL, su desarrollo final tiene que estar licenciado GPL). Desde un punto de vista del desarrollador la GPL presenta un inconveniente en este punto, y lamentablemente mucha gente se cierra ante el uso de dicho código.

No obstante la MPL no es tan excesivamente permisiva como las licencias tipo BSD. Estas licencias son denominadas de copyleft débil. La NPL (luego la MPL) fue la primera licencia nueva después de muchos años, que se encargaba de algunos puntos que no fueron tenidos en cuenta por las licencias BSD y GNU. En el espectro de las licencias de software libre se la puede considerar adyacente a la licencia estilo BSD, pero perfeccionada.

2.2.3. Modelo de negocio.

El negocio detrás del software libre se caracteriza por la oferta de servicios adicionales al software como: la personalización, instalación, soporte

técnico, donaciones, patrocinios, y mantenimiento; en contraposición al modelo de negocio basado en licencias predominante en el software de código cerrado.

El software libre es una cuestión de libertad, no de precio. Para entender el concepto, se debe pensar en libre como en libre expresión, no como en una barra libre es decir de gratuidad.

Software libre no significa que no sea comercial. Un programa libre debe estar disponible para el uso comercial, la programación comercial y la distribución comercial. La programación comercial de software libre ya no es inusual; tal software libre comercial es muy importante. Puede haber pagado dinero para obtener copias de software libre, o puede haber obtenido copias sin costo. Pero sin tener en cuenta cómo obtuvo sus copias, siempre tiene la libertad de copiar y modificar el software, incluso de vender copias.

Sin embargo, las reglas sobre cómo empaquetar una versión modificada son aceptables si no limitan sustancialmente su libertad de publicar versiones modificadas, o su libertad de hacer y usar versiones modificadas en privado.

Las reglas que requieran la publicación del código fuente a los usuarios para versiones que haya puesto en uso público también son aceptables. También es aceptable para la licencia requerir que, si se distribuye una versión modificada y un programador anterior pide una copia, se debe enviar una, o que se identifique como propias sus modificaciones.

En el proyecto GNU, se utiliza copyleft para proteger legalmente estas libertades para todos. Pero también existe software libre sin copyleft. Según la práctica se ha determinado que existen razones importantes por las que es mejor usar copyleft, pero si el programa en desarrollo es software libre sin copyleft, sigue encasillándose en el marco ético.

En algunos casos las regulaciones de control de exportación y las sanciones comerciales pueden limitar las libertades de distribuir copias de programas intencionalmente. Los desarrolladores de software no tienen el poder de eliminar o pasar por alto estas restricciones, pero lo que pueden y deben hacer es rechazar imponerlas como condiciones para el uso del programa. De este modo, las restricciones no afectarán a las actividades ni a las personas fuera de las jurisdicciones de dichos gobiernos. Por ende, las licencias de software libre no deben requerir la obediencia a ninguna regulación de exportaciones como condición de cualquiera de las libertades esenciales.

La mayoría de las licencias de software libre están basadas en el copyright, y existen límites en los tipos de requisitos que pueden ser impuestos a través del copyright. Si una licencia basada en el copyright respeta la libertad en las formas antes mencionadas, es poco probable tener otro tipo de problema que no se haya anticipado. Sin embargo, algunas licencias de software libre están basadas en contratos, y los contratos pueden imponer un rango mucho más grande de restricciones posibles. Esto significa que existen muchas maneras posibles de que tal licencia pueda ser

inaceptablemente restrictiva y que no sea libre, por ejemplo cuando una empresa desarrolla su propio software contratando personal externo por medio de un contrato laboral y un contrato de desarrollo de software.

Finalmente enumerar todas las posibles formas en las que eso puede pasar es una tarea que resulta imposible. Si una licencia basada en un contrato restringe al usuario de un modo que no puedan hacer las licencias basadas en el copyright, y que no está mencionado aquí como legítimo, entonces se puede concluir que deja de ser software libre.

2.3. Linux Ubuntu.

2.3.1. Definición e historia de Linux Ubuntu.

Ubuntu es una distribución GNU/Linux que ofrece un sistema operativo orientado principalmente a computadoras personales, aunque también proporciona soporte para servidores. Es una de las más importantes distribuciones de GNU/Linux a nivel mundial. Se basa en la distro Debian GNU/Linux y concentra su objetivo en la facilidad y libertad de uso, la fluida instalación y los lanzamientos regulares. El principal patrocinador es Canonical Ltd., una empresa privada fundada y financiada por el empresario sudafricano Mark Shuttleworth.¹⁰

¹⁰ Vegas Gabriel, Historia de Ubuntu, <http://gabrielvegas.wordpress.com/ubuntu-history/>

El nombre de la distribución proviene del concepto zulú y xhosa de Ubuntu, que significa "humanidad hacia otros" o "yo soy porque nosotros somos". Ubuntu es un movimiento sudafricano encabezado por el obispo Desmond Tutu, quien ganó el Premio Nobel de la Paz en 1984 por sus luchas en contra del Apartheid en Sudáfrica. El sudafricano Mark Shuttleworth, padrino del proyecto, se encontraba muy familiarizado con la corriente. Tras ver similitudes entre los ideales de los proyectos GNU, Debian y en general con el movimiento del software libre, decidió aprovechar la ocasión para difundir los ideales de Ubuntu. El eslogan de la distribución "Linux para seres humanos" (en inglés "Linux for Human Beings"¹¹) resume una de sus metas principales: hacer de GNU/Linux un sistema operativo más accesible y fácil de usar.

La versión estable de Ubuntu Desktop o Server está disponible para su descarga en la página oficial de Ubuntu: www.ubuntu.com.

2.3.2. Características.

Ubuntu está basada en la distribución Debian GNU/Linux y soporta oficialmente dos arquitecturas de hardware: Intel x86, AMD64 y a partir de abril de 2009, con la liberación de Ubuntu 9.04 se ofreció soporte oficial para procesadores ARM.²²; sin embargo ha sido portada extraoficialmente a

¹¹ Mark Shuttleworth and Canonical Ltd, Slogan del lanzamiento del proyecto Ubuntu

cinco arquitecturas más: Mac PowerPC, HP PA-RISC, SPARC,1 IA-64,2 y Sony Playstation 3.

Al igual que casi cualquier distribución basada en GNU/Linux, Ubuntu es capaz de actualizar a la vez todas las aplicaciones instaladas en la máquina a través de repositorios, a diferencia de otros sistemas operativos propietarios, donde esto no es posible.

Esta distribución ha sido y está siendo traducida a más de 130 idiomas, y cada usuario es capaz de colaborar voluntariamente a esta causa, a través de Internet.

Los desarrolladores de Ubuntu se basan en gran medida en el trabajo de otros proyectos de software libre y código abierto, pero en especial en el de la comunidad de Debian.

2.3.3. Ubuntu 9.10 Server – Karmic Koala.

Edición destinada a la construcción de un servidor, aunque se puede incluir un entorno gráfico, casi todo se maneja desde la línea de comandos.

Desde el lanzamiento inicial de Ubuntu en el año 2004, se ha implementado en un sinnúmero de organizaciones, de hecho miles de organizaciones a

nivel mundial, entregando la clave en servicios de confiabilidad, predictibilidad y economización.

Ubuntu Server es:

- Eficiente en el control de energía, es un sistema operativo con bajo consumo de recursos tanto en disco como en memoria física.
- No hay licencia ni costos de mantenimiento.
- Un ecosistema expandido.
- Mantenimiento mínimo.
- Una comunidad de asociados creciente.

Construido en base a una distribución de Debian, la cual es conocida por sus instalaciones robustas, tiene una herencia fuerte respecto a la confiabilidad en el desempeño y una evolución predecible.

Beneficios

Los usuarios de Ubuntu Server disfrutan de un desempeño sin precedentes y la seguridad que Linux provee.

Debido a que Ubuntu se preocupa acerca del negocio se asegura que la versión Server de Ubuntu:

- Se integre fácilmente a sus redes actuales.
- Provea un costo total muy bajo de propiedad.
- Ofrece escenarios múltiples del ciclo de vida para que el usuario los escoja y los acople según sus necesidades.
- Sea provisto de un mantenimiento libre de por vida.
- Sea cocido con el soporte de Canonical Ltd a nivel mundial, para ayuda, consultoría y entrenamiento.

La mayoría de organizaciones quieren asegurarse de tener un servicio profesional detrás de ellas, es por eso que Canonical provee servicios excepcionales en la edición de servidor de Ubuntu.

El soporte multilinguaje está disponible para la atención al cliente desde un equipo de expertos dedicados, quienes están disponibles las 24 días los 7 días de la semana. Incluso los técnicos de soporte pueden ayudar con las implantaciones de Ubuntu Server y se los puede contactar vía telefónica o correo electrónico.¹²

Es importante recalcar que existe un entrenamiento dedicado en cómo implementar Ubuntu Server y su Nube Empresarial, la misma que está disponible a través de cursos estructurados que revelan los pasos a seguir para una instalación exitosa.

¹² www.ubuntu.com

Kernel o Núcleo

Ubuntu Server incluye un kernel especialmente configurado para satisfacer los requerimientos de todas las cargas de trabajo comunes encontradas típicamente en un servidor, de esta manera se pueda obtener lo mejor del hardware mientras se mantiene un consumo de energía y recursos óptimo.

Autenticación

Sea que ejecute Windows®, Unix, MacOS X® u otra distribución semejante a Unix, La edición servidor de Ubuntu viene con todas las herramientas para integrar fácilmente con los sistemas de administración de usuario actuales (Microsoft®, Solaris, Spark, Apple®, Unix), permitiendo que los clientes se autenticuen sin problemas, compartan documentos y accedan a los servicios.

Implementaciones

Ubuntu, ya sea usado para computadores personales o para servidores, provee herramientas de automatización de la instalación por lo tanto, permite la reproducción de diferentes escenarios de instalación independientes al hardware.

Servidor WEB

Sin importar el lenguaje que se utilice, PHP, Perl, Java Ruby o Python junto con Apache y MySQL, Postgresql, DB2 u Oracle, Ubuntu puede estar listo en minutos para empezar a servir a cualquier tipo de páginas web dinámicas.

Servidor de correo

El servidor de Correo de Ubuntu está compuesto de dos de los más renombrados paquetes de código abierto como son Postfix y Dovecot.

Postfix es un agente de transferencia pequeño que en otras palabras significa el programa que enruta y entrega los e-mails. Dovecot es el agente que se comunica con el cliente de correo electrónico usando protocolos IMAP o POP3.

Infraestructura

Ubuntu Server provee al usuario las herramientas necesarias para crear la infraestructura de su red. Estas herramientas permiten también añadir los servicios clave para crear y mantener sus redes haciendo uso de la capacidad integrada de Linux para enrutar las redes.

Administración de Energía

Las características de administración de energía permiten hacer que el sistema sea resistente y reduce las facturas de consumo de electricidad.

Seguridad

El equipo de seguridad de Ubuntu trabaja a la par con Debian y otros vendedores líderes de seguridad para asegurar que Ubuntu permanezca tan seguro como sea posible; cuando se descubren defectos, publican noticias de cuando hay parches disponibles.

JeOS

Ubuntu Server Edition JeOS es una variante eficiente del servidor configurado especialmente para accesorios virtuales.

Archivos e Impresión

Compartir archivos con Linux, Windows® y Mac® esta hecho de la manera más sencilla, simplemente se necesita poner un visto en la casilla correspondiente en el proceso de instalación.

Virtualización

Ubuntu Server Ofrece una gran variedad de tecnologías de virtualización, tanto para el uso de un anfitrión o de un invitado del sistema operativo. Desde soluciones gratuitas a no gratuitas, Ubuntu ofrece alternativas para satisfacer cada necesidad.

Paquetes

El sistema de paquetes de Debian es una de las mejores fortalezas de Ubuntu que está directamente heredado de esa distribución. No solo porque provee una implementación eficiente de paquetes y herramientas de mantenimiento sino también la diferencia en simplicidad y practicidad a otros sistemas de paquetes complicados.

Procesadores

Ubuntu Server Edition esta optimizado para trabajar con procesadores multinúcleo que pueden ser de 32 o 64 bits. El equipo encargado del kernel está trabajando de manera muy cercana con Intel y AMD para implementar en el sistema las últimas características y ventajas de los procesadores y de igual manera los controladores más actuales para manejar los chipsets.

Almacenamiento

La edición servidor de Ubuntu trae consigo el poder de compartir almacenamiento entre los servidores del usuario de una manera muy económica. Mas y mas organizaciones confían en el desempeño formidable de las redes Gigabit o 10 Gigabit para construir sus propios SAN(Storage Area Network) tanto para almacenamiento de datos o para almacenaje de maquinas virtuales.

Terminal server

Ubuntu ofrece una manera fácil de pasar de una instalación de escritorio a un servidor de terminal para la oficina o negocio para servir a múltiples clientes sin disco duro.

2.3.4. Requisitos mínimos de hardware para Ubuntu Server.

La edición servidor de Ubuntu está diseñada para ejecutarse en procesadores AMD x86, AMD_64 y EM_64T. Este requiere de un mínimo de 192 Mb en RAM y de 1 GB de espacio en disco. Dependiendo de las necesidades del usuario, se podría ejecutarlo incluso con menos recursos pero pasaría a ser un riesgo si se ignoran las recomendaciones de hardware mínimo.

2.4. Gambas.

2.4.1. Definición de Gambas.

Gambas es un lenguaje orientado a objetos con gran número de capacidades y un entorno de desarrollo basado en un intérprete de BASIC®. Se encuentra publicado bajo licencia GNU (General Public Licence). Gambas es un ambiente libre de desarrollo basado en un intérprete Basic con extensiones de objetos, similar a Visual Basic™ de Microsoft®¹³.

Con Gambas se puede diseñar rápidamente un programa con interfaz gráfica de usuario (GUI) o librerías para otras aplicaciones de consola o con GUI, acceder a bases de datos MySQL, PostgreSQL, Firebird, ODBC y SQLite, aplicaciones piloto KDE con DCOP, además ofrece la capacidad de traducir el programa a cualquier lenguaje, crear aplicaciones de red fácilmente, hacer aplicaciones 3D con OpenGL para gráficos en 3 dimensiones, hacer aplicaciones web CGI, entre otras prestaciones.

Su arquitectura está ampliamente inspirada por el lenguaje Java. Un ejecutable Gambas es un archivo de clases compiladas y archivos con datos que se ejecuta a través de un intérprete creado al efecto.

Gambas se compone de:

¹³ Usuario de seudónimo caipirinho, España, Agosto de 2009. <http://www.taringa.net/posts/linux/3156102/Programador-Visual-Basic-en-Linux-Equot:Open-sourcerEamp;q.html>

- Un compilador.
- Un intérprete.
- Un archivador.
- Un intérprete de scripts.
- Un entorno de desarrollo.
- Muchos componentes que extienden su funcionalidad.

El compilador es un programa rápido y pequeño escrito en lenguaje C. Este compila cerca de 128000 líneas de código útil por segundo en un procesador AMD Athlon 2000; tal velocidad permite un proceso rápido de desarrollo en el ciclo código, compilación y pruebas.

No realiza prácticamente ninguna optimización en este momento, dejando dicho trabajo al intérprete para su realización durante la ejecución del programa. Gestiona y compila traducciones de cadenas empleando las herramientas de GNU gettext.

El intérprete es también un pequeño ejecutable escrito en lenguaje C, de menos de 180 Kilobytes de tamaño en disco, proporciona todas las características nativas del lenguaje, dando acceso indirecto prácticamente a todas las herramientas del estándar POSIX contenidas en la librería glibc para atender las llamadas de sistema.

Carga clases por demanda, optimizando el bytecode la primera vez que es ejecutado. El enlace entre clases se realiza también en tiempo de ejecución,

y lo más tarde posible. Así, incluso ejecutables de gran tamaño se inician rápidamente.

Finalmente, el archivador es un programa que crea un ejecutable Gambas a partir de la carpeta de un proyecto Gambas.

2.4.2. Características de Gambas.

Las características principales del lenguaje BASIC de Gambas son:

- Cerca de 250 palabras clave y funciones nativas para gestionar prácticamente cualquier cosa: aritmética, cadenas, errores, entrada-salida, archivos, fechas, etc.
- Gestión completa de errores.
- Control total de procesos, con gestión de pseudo-terminal.
- Soporte completo para recibir eventos de descriptores de archivo de entrada/salida.
- Soporte de lazo de eventos con temporizadores.
- Soporte nativo de cadenas en formato UTF-8.
- Soporte completo de internacionalización y traducciones.
- Capacidad de hacer llamadas a funciones externas alojadas en librerías de enlace dinámico.

Pero Gambas es también un lenguaje totalmente orientado a objetos, con:

- Objetos y clases.
- Propiedades, métodos, constantes y eventos.
- Símbolos públicos y privados.
- Gestión de métodos virtuales.
- Herencia simple.
- Constructores y destructores.
- Gestión de matrices, enumeradores, objetos ordenables.

La herencia en Gambas es un mecanismo totalmente dinámico, y permite:

- Crear versiones más especializadas de cualquier clase existente.
- Reimplementar clases y extenderlas.
- Reimplementar algunos métodos o propiedades de una clase.

Cualquier clase puede ser heredada, reimplementada o sobrecargada, incluso aquellas nativas creadas en C/C++.

Por último, una clase nativa denominada "Observer" que permite interceptar cualquier evento generado por un objeto.

2.4.3. Diferencias de Gambas respecto a Visual Basic®.

Gambas no intenta ser un clon de Microsoft® Visual Basic®, sin embargo utiliza BASIC® por ende, existen varias similitudes entre los dos lenguajes,

y algunas relaciones en las características; posiblemente hay más similitudes que diferencias, pero no se puede copiar un proyecto de Visual Basic® e intentar compilarlo bajo el entorno de Gambas.

La intención es ayudar a aquellos programadores que estén migrando a Gambas desde ese lenguaje.

Diferencias no referentes al lenguaje

- En Visual Basic® el código de cada formulario y el objeto formulario están incluidos en el mismo archivo. Gambas usa dos archivos separados para ello: archivo .form y un archivo .clase.
- Extensiones de los archivos:

Tipo de Archivo	Visual Basic	Gambas
Archivo de definición del proyecto	.vbp	.project (un por directorio)
Módulo	.bas	.module
Archivo de Class	.cls	.
Archivo de definición del Form	.frm	.form
Archivos de recursos binarios	.frx	Algún otro archivo almacenado en el directorio del proyecto.

Tabla 2. 2: Comparación entre Gambas y Visual Basic.

- Los proyectos de Gambas se definen con un directorio que tiene un archivo .project dentro, y todos los archivos en ese directorio. Visual Basic® permite tener múltiples archivos de proyecto en distintos directorios, o usar el mismo archivo fuente de un directorio en distintos

proyectos, lo que tiene sus ventajas y sus inconvenientes.

- Las medidas de la pantalla se hacen en Visual Basic® en "twips", que son un 1/1440 de pulgada; en Gambas se hacen en píxeles reales.
- Los controles de los Form son privados por defecto en los programas de Gambas. Se puede cambiar este comportamiento en las Propiedades del Proyecto, marcando el recuadro "Los controles del formulario son públicos".
- Las funciones de conversión Str\$, Val, CStr se comportan de forma distinta. Por ejemplo, Str\$ y Val usan la configuración de idioma puesta en Gambas, mientras que no hacen eso en Visual Basic®. En la documentación se detalla de una manera más extensa.

2.4.4. Lo que Visual Basic® tiene y Gambas no.

- No se puede editar el código en modo de depuración en Gambas; es necesario detener la ejecución del programa.
- En Gambas, los tipos de datos simples (Integer, String, etc.) se pasan por valor a los procedimientos y funciones. No se pueden pasar por referencia como en Visual Basic®. Hay que recordar que Visual Basic® pasa los parámetros por referencia si no se usa la palabra clave ByVal, así que se debe tener cuidado cuando se intente migrar un proyecto Visual Basic®. También se pasan siempre por referencia el contenido de los tipos de datos objeto array, collections y objects en ambos lenguajes. Pasar argumento por referencia será posible en las versiones futuras ya

en desarrollo.

- No existe nada parecido a una variable global para todo el proyecto en Gambas. Para simularlo se puede hacer una clase llamada por ejemplo "Global" y declarar las variables globales como variables públicas estáticas de esa clase, entonces se puede referir a ellas como `Global.nombre_de_la_variable` en todo el proyecto. No es una práctica de programación elegante, pero al menos funcionará como una variable global cuando se quiera usarla.
- En un módulo de Visual Basic®, a no ser que incluya la frase `Option Explicit`, no necesita declarar las variables antes de usarlas. Gambas se comporta como si `Option Explicit` estuviera siempre activo, lo que mejora mucho el código al ahorrar codificación al programador.
- No hay nada en Gambas equivalente a la propiedad `Index` de los controles en los formularios de Visual Basic®. Es muy fácil crear arreglos de controles, pero hay que hacerlo mediante código. No existe en la actualidad ninguna forma de hacerlo gráficamente. Por tanto, cuando se copie y pegue un control, en lugar de preguntar si desea hacer un arreglo de controles, lo que Gambas hace automáticamente es renombrar el control copiado con un nombre adecuado.
- El evento `MouseMove` solo ocurre en Gambas cuando hay pulsado un botón del ratón. La única excepción es el control `DrawingArea`, que tiene una propiedad `Tracking` para poder seguir los eventos del ratón incluso si no se ha pulsado ningún botón.
- En Visual Basic se pueden unir dos strings con el símbolo `+`. Como en Gambas el signo `+` sólo se usa para la suma matemática, se debe usar el

signo & cuando quiera unir dos strings.

- El símbolo de dos puntos ":" no sirve para separar el código. Se debe poner en una nueva línea para cumplir con esto.
- Algunos de los caracteres que se pueden usar en el código de Visual Basic® como identificadores, tales como el subrayado "_", no se aceptan en Gambas.
- En Gambas no se puede usar GOTO para capturar errores, En su lugar se utiliza CATCH, FINALLY o TRY .
- ENUM no puede ser usado para enumerar constantes de tipo Integer. En su lugar, se tiene que definir cada elemento ENUM como una constante.

Ejemplos

```
CONST SUMA AS Integer = 1
```

```
CONST RESTA AS Integer = 2
```

2.4.5. Lo que Gambas tiene y Visual Basic® no.

- Al contrario que con Visual Basic®, no se necesita compilar el soporte gráfico si se quiere escribir una aplicación de consola en Gambas. Tan solo se debe quitar la selección al componente gb.qt en las Propiedades del Proyecto y asegurarse de que se ha definido un Sub Main().
- En lugar de la palabra clave WITH EVENTS, se debe proveer un "nombre de evento" a los objetos que se desea que disparen el evento. Esto se puede encontrar en la documentación en línea de la instrucción NEW para más información.

- Gambas tiene el concepto de grupos de controles, lo que permite manejar eventos de cualquier número de controles distintos dentro de una única subrutina. Esto reduce código redundante y puede usarse para hacer la mayoría de las cosas que los index de los controles de Visual Basic® hacen, y algunas cosas que Visual Basic® no puede hacer.
- Mientras que en Visual Basic® es imposible ejecutar un programa de forma síncrona y recibir su salida sin aprender a hacer llamadas a la API de Windows® (Shell simplemente lanza el programa, sin esperar nada) Gambas permite hacerlo usando SHELL y EXEC, controlar el proceso que inicia usando la clase Process, e incluso leer y escribir en el proceso, lo que facilita añadir funcionalidades a aplicaciones y asistentes. Esto hace que sea muy fácil escribir interfaces gráficas con Gambas para casi cualquier programa de consola.
- Se puede hacer todo lo anterior con los dispositivos de Unix y archivos especiales también, como puertos serie o paralelo.
- Para hacer una ventana con formas extrañas, solo hay que asignar la propiedad ME.Picture y la propiedad ME.Mask de la ventana usando una imagen que tenga áreas transparentes. Visual Basic® necesita para eso llamadas a la API y algo más de trabajo en cuanto a código respecta.
- Se pueden crear controles y menús de forma dinámica, tan sólo se necesita instanciarlos con la instrucción NEW.
- Se puede insertar un formulario de Gambas dentro de otro: cuando se instancie el primero, se especifica el segundo como parent.
- Los controles tienen los eventos Enter y Leave, lo que le permite saber cuando el puntero del ratón entra en un control y cuando lo abandona. Se

pueden usar para implementar fácilmente efectos mouse-over.

- Se pueden leer datos de archivos binarios y manejar automáticamente la forma de colocar sus bytes (endianness), usando la propiedad `ByteOrder` de la clase `Stream`.
- Gambas usa internamente el juego de caracteres UTF-8, por lo que se puede internacionalizar fácilmente todo el proyecto.
- Gambas es Software libre y su entorno de desarrollo está escrito en Gambas, lo que le permite un grado máximo de personalización, sólo dependiendo del nivel de conocimientos del programador sobre BASIC®.

2.4.6. La misma funcionalidad, diferente terminología.

- End Sub/End Function: se utiliza la instrucción `END`.
- Exit Sub/Exit Function: se utiliza la instrucción `RETURN`. Además, en vez de poner el nombre de una variable igual a la de la función para retornar el valor, simplemente se puede devolver el valor deseado de la función como un simple parámetro en la instrucción `RETURN`.
- End (finalizar el programa): se utiliza la instrucción `QUIT`.
- Los arreglos se utiliza con corchetes a diferencia de los tradicionales paréntesis. Así por ejemplo se usa `DIM x[9] AS Float` en vez de `DIM x(9) AS Float`
- Los arreglos no tiene un elemento extra para el indexado como `1...n`, el índice siempre debe estar en `0..(n-1)`.
- On Error Goto: se utiliza la instrucción `TRY`, `CATCH` y `FINALLY`.

- MsgBox: se utiliza la instrucción Message. Normalmente se utiliza para información Message.Info.
- DoEvents: se utiliza la instrucción WAIT. WAIT también reemplaza la API utilizada en Windows® que incluye la función “sleep”.
- VScrollBar, HScrollBar: el ScrollBar de gambas reemplaza a las dos de VisualBasic®. Esta automáticamente se configura ya sea para utilizarla de manera vertical u horizontal, basta con ajusta el ancho y el alto y el control determina si es un scroll vertical y horizontal.
- El dialogo Open y Save (abrir y guardar respectivamente): Se puede utilizar los controles Qt o los controles de diálogos KDE en vez de los controles comunes de dialogo de Windows®. Alguna de las propiedades son nombradas de manera diferente y los filtros se especifica con un arreglo de tipo string como este ["Filtro 1 (*.jpg)" , "Filtro 2 (*.bmp)"].
- Validación del texto de entrada: en Visual Basic® ciertos eventos tienen un parámetro de cancelación que se puede configurar para prevenir que el evento sea manejado de manera normal después que el manejador (handler) a terminado, así de esta manera se puede escribir solo números o letras, realizar una validación de campo o forzar la entrada a ser mayúsculas o minúsculas. En gambas, esto es hecho utilizando la instrucción STOP EVENT.
- Los manejadores del teclado y el mouse no manejan parámetros. En vez de eso se utiliza miembros estáticos públicos de las clases Mouse y Key.
Por ejemplo:
 - Mouse.X y Mouse.Y para las coordenadas del mouse.
 - Mouse.Left Para conocer si se presiona el botón izquierdo del mouse.

- Key.Code Para obtener el código de una tecla.
 - Key.Shift para saber si la tecla SHIFT del teclado se ha presionado, etc.
-
- No se utiliza el método FORM.LOAD. Este es completamente diferente del evento LOAD de Visual Basic. En Gambas, este es tan solo un método estático que crea la instancia implícita del formulario.

2.5. MySQL.

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL de Oracle está desarrollado como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL (Licencia Pública General) para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C®.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propietario y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet.

Las características predominantes de MySQL son las siguientes:

- Soporte a multiplataforma.
- Procedimientos almacenados.
- Disparadores (triggers).
- Cursores.
- Vistas actualizables.
- Soporte a VARCHAR.
- INFORMATION_SCHEMA.
- Modo Strict.
- Soporte X/Open XA de transacciones distribuidas; transacción en dos fases como parte de esto, utilizando el motor InnoDB de Oracle.
- Motores de almacenamiento independientes (MyISAM para lecturas rápidas, InnoDB para transacciones e integridad referencial).
- Transacciones con los motores de almacenamiento InnoDB, BDB Y Cluster; puntos de recuperación (savepoints) con InnoDB.
- Soporte para SSL (Secure Socket Layer).
- Query caching (Almacenamiento en cache de consultas).
- Sub-SELECTs (o SELECTs anidados).
- Réplica con un maestro por esclavo, varios esclavos por maestro, sin

soporte automático para múltiples maestros por esclavo.

- Indexing y buscando campos de texto completos usando el motor de almacenamiento MyISAM.
- Embedded database library.
- Soporte completo para Unicode.
- Conforme a las reglas ACID usando los motores InnoDB, BDB y Cluster.
- Shared-nothing clustering a través de MySQL Cluster.
- Usa GNU Automake, Autoconf, y Libtool para portabilidad.
- Uso de multihilos mediante hilos del kernel.
- Usa tablas en disco b-tree para búsquedas rápidas con compresión de índice.
- Tablas hash en memoria temporales.
- El código MySQL se prueba con Purify (un detector de memoria perdida comercial) así como con Valgrind, una herramienta GPL.
- Completo soporte para operadores y funciones en cláusulas Select y Where.
- Completo soporte para cláusulas Group By y Order By, soporte de funciones de agrupación.
- Seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.
- Soporta gran cantidad de datos. MySQL Server tiene bases de datos de hasta 50 millones de registros.
- Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas.

El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2).

- Los clientes se conectan al servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows se pueden conectar usando named pipes y en sistemas Unix usando ficheros socket Unix.
- En MySQL 5.0, los clientes y servidores Windows se pueden conectar usando memoria compartida.
- MySQL contiene su propio paquete de pruebas de rendimiento proporcionado con el código fuente de la distribución de MySQL

2.6. Paquetes importantes para Linux.

2.6.1. Onboard.

OnBoard es un teclado virtual en pantalla escrito en Cairo de Phyton como un proyecto de Google Summer of Code.

Un teclado en pantalla simple es útil para aquellos que no pueden utilizar un teclado normal de computador, ya sea por una discapacidad, o porque se utiliza un computador que no tenga un teclado físico conectado a este, como son los terminales de información, o porque se quiere ingresar en caracteres de idiomas diferentes a la distribución del teclado existente.

Características de OnBoard.

- La configuración inicial básica es un teclado Qwerty que contiene la mayoría de teclas comúnmente utilizadas. Esto incluye a las letras en sí y a los modificadores de las letras como son las teclas Shift, Ctrl, Alt y algunas otras como la tecla de espacio, el tabulador, la coma y el punto. El teclado numérico puede o no estar incluido.
- Las teclas menos usadas se las ha ubicado en un segundo panel que se torna disponible al presionar una tecla de acceso.
- La distribución del teclado y la localización del mismo pueden ser fácilmente configuradas.
- El teclado está configurado por defecto para funcionar siempre al tope, incluso si la aplicación activa toma lugar en toda la pantalla. Esta característica puede deshabilitarse.
- El teclado puede ser anclado a una posición fija en la pantalla.
- La ventana del teclado puede ser modificada en cuanto al tamaño respecta como cualquier tipo de ventana y por su puesto su contenido se ajustará a la misma.
- Al presionar una tecla virtual se dispara la misma función que cuando se aplasta una tecla en un teclado físico.
- Teclas muertas: un simple clic en un modificador hará que éste permanezca como tecla muerta hasta que la siguiente tecla sea presionada; un doble clic en el modificado hará que el modificador permanezca como tecla muerta hasta que existe otro clic en el mismo.
- Puede cambiar con un botón el clic derecho o izquierdo del mouse.
- Botones Macros en el teclado principal que permite que se acceda a los

paneles secundarios del programa para más botones o plugins.

2.6.2. Cdrtools.

Cdrtools es un conjunto de programas de línea de comando que permite grabar CD, DVD y medios BlueRay BD.

Esta suite incluye los siguientes programas:

- cdrecord: Un programa para grabar CD,DVD o BD
- readcd: Un programa para leer un medio tipo CD,DVD o BD con características para clonación de discos.
- cdda2wav: El programa con más evolución para la extracción de sonidos de CD.
- mkisofs: Un programa que crea sistema de archivos híbridos de tipo ISO9660, JOLIET, HFS con la opción adicional de Rock Ridge.
- isodebug: Un programa para imprimir la información de depuración de mkisofs desde el medio óptico.
- isodump: Un programa para descargar la información de un medio ISO-9660.
- isoinfo: Un programa para el análisis y verificación de sistemas de archivos ISO9660, Joliet y Rock-Ridge.
- isovfy: Un programa para verificar las estructuras ISO-9660.
- rscsi: Un demonio remoto para habilitar SCSI.

Cdrtools es 100% libre verdaderamente OpenSource que permite que sigue las definiciones de la OSI (Open source initiative). Este se encuentra bajo continuo desarrollo desde febrero de 1996. La meta que cdrtools persigue es la permanente provisión de un conjunto de aplicaciones para escribir en medios CD, DVD, BD y productos similares como por ejemplo el HD-DVD.

Desde su inicio el trabajo del equipo de cdrtools ha sido permanente hasta la actualidad para favorecer los beneficios que reciben los usuarios finales.

Cdrecord

Cdrecord es utilizado ampliamente para grabar tanto datos como audio en discos compactos CD o en dispositivos DVD su creador es el alemán Joerg Schilling.

La sintaxis de la instrucción es la siguiente:

```
cdrecord [ general options ] dev=device [ track options ] track1...trackn
```

En donde DEV se refiere al dispositivo de grabación, existen opciones generales y también opciones respecto al modo de grabación de las pistas del disco. Para información más detallada se puede visitar el sitio web de cdrecord: <http://cdrecord.berlios.de/private/man/cdrecord/cdrecord.1.html>.

La mayoría de usuarios no necesitan preocuparse por el nombre de los dispositivos de grabación.

Si la opción "dev=" no está especificada, cdrecords implementa un soporte de auto objetivo y automáticamente busca el drive o unidad, en caso de que exactamente exista un tipo de unidad CD-ROM disponible en el sistema. En el caso de que exista más de un tipo de dispositivo CD-ROM puede extraerse el nombre de los posibles dispositivos de grabación con el comando: `cdrecord -scanbus`.

Es importante mencionar que la herramienta necesita privilegios administrativos para funcionar correctamente.

Cdrecord puede tomar una imagen ISO y grabarla en un medio óptico o puede copiar directamente de otro medio la imagen a grabarse.

Finalmente Cdrecord está completamente basado en comandos tipo SCSI pero esto no es un problema para manejar dispositivos normales de escritura de CD/DVD ya que estos están diseñados para utilizar estos comandos. Incluso los dispositivos ATAPI son dispositivos SCSI que utilizan una interface ATA para la transmisión de paquetes como un capa de transporte SCSI integrado en el transporte IDE (ATA). Es muy probable que se requiera especificar un capa de transporte en la línea de comandos si el núcleo del sistema operativo no implementa completamente un subsistema de

controladores de soporte que permitan acceder a la unidad vía comandos SCSI a través de la interfaz de usuario.

Mkisofs

Mkisofs es un programa pre-masterizado muy eficiente para generar sistemas de archivos híbridos tanto para ISO9660, JOLIET, HFS y Rock Ridge.

Sintaxis de la instrucción

```
mkisofs [ options ] [ -o filename ] pathspec [pathspec ...]
```

Donde “[options]” puede incluir opciones como Joliet, HFS, Rock Ridge, etc.; por otro lado “-o filename” será el path y nombre de la imagen ISO. Para un completo listado de las opciones disponibles se puede visitar el sitio web: <http://linux.die.net/man/8/mkisofs>.

Mkisofs es capaz de generar grabaciones SUSP (System Use Sharing Protocol) especificado por el protocolo de intercambio de Rock Ridge. Este es usado para describir un sistema de archivos en iso9660 para un anfitrión UNIX, y provee información para nombres largos, permisos POSIX, vínculos simbólicos, dispositivos de bloqueo y dispositivos de caracteres.

Si Joliet o HFS son especificadas en la línea de comandos entonces mkisofs creara un sistema de archivos adicionales para metadatos Joliet o

HFS. La opción por defecto es ISO9660 a menos que se especifique Joliet o HFS.

Mkisofts puede generar un verdadero modelo HFS híbrido para poder ser accedido en una computadores Macintosh® y también en computadores con sistemas ISO9660; recordando que HFS viene de Hierarchical File System y es el sistema nativo de archivos de las computadores Macintosh®.

Como una alternativa, mkisofts puede genera la extensión las extensiones de Apple a ISO9660 para cada archivo. Estas extensiones proveen a cada archivo banderas de información de su creador, tipo, etc. en computadoras Macintosh®. Además, mkisofs toma un snapshot de un árbol de directorios dado, y genera una imagen binaria la cual corresponde a un sistema de archivos ISO9660 o HFS cuando se escribe en un dispositivo de bloqueo.

Cada archivo escrito en el sistema ISO9660 debe tener un nombre de archivo en el formato 8.3 (8 caracteres, punto, 3 caracteres, todo en mayúsculas), incluso si Rock Ridge está en uso. Este sistema de archivos es usado en sistemas que no pueden utilizar extensiones Rock Ridge como por ejemplo MS-DOS®, y cada nombre de archivo en cada directorio debe ser diferente de los otros nombres de archivos en el mismo directorio. Mkisofs generalmente intenta corregir los nombre a la forma correcta al forzar que sean en mayúsculas y truncando el nombre de ser necesario, pero a veces esto no resulta de manera satisfactoria especialmente en casos donde los nombres truncados son semejantes; para esto mkisofts asigna el signo ~

seguido de un numerador para hacer que los archivos sean únicos ejemplo ARCHIV~1.FIL.

Eric Youngdale escribió la primera versión en el año de 1993 de esta herramienta. El copyright para las versiones antiguas le pertenece a Yggdrasil Computing, Incorporated. Jrg Schilling escribió la librería de transporte para la adaptación de capas de mkisofs y las nuevas partes de esta utilidad en 1999, esto hace que el copyright© de mkisofs 1999, 2000, 2001 sea de Jrg Schilling.

2.6.3. Growisofs.

Growisofs fue originalmente diseñada como un frontend de mkisofs para facilitar al agregamiento de datos a los volúmenes ISO9660 residentes en medios de acceso randómico como DVD+RW, DVD-RAM, archivos planos y particiones de discos duros. En el curso de del desarrollo se agrego el soporte para DVD de propósitos generales, es por eso que, hoy por hoy growisofs soporte no solo medios de acceso randómico sino también medios masterizados multisesión de DVD, DVD+R y DVD-R/-RW.

Por otra parte growisofs soporta la grabación de imágenes premasterizadas en formatos UDF, ISO9660 y cualquier otro sistema de archivos para todo tipo de medios DVD.

Sintaxis de la instrucción

```
Growisofs [-dry-run] [-dvd-compat][-overburn][-speed=1] -[Z|M] /dev/dvd  
          <mkisofs-options>
```

En donde podemos especificar parámetros como “[dvd-compat]” para máxima compatibilidad del medio en varios computadores con diferentes sistemas de archivos; “[overburn]” para proteger el buffer de lectura de la unidad y mantener la integridad de la grabación y los datos; “[speed=x]” que permite especificar la velocidad de grabación y “-[Z|M] /dev/dvd” que especifica la unidad de grabación a utilizarse. Para más información se puede ingresar al sitio web por medio del siguiente vínculo URL: http://gd.tuwien.ac.at/linuxcommand.org/man_pages/growisofs1.html el cual contiene una descripción de todos los parámetros y las diferentes opciones que soporta la línea de comando.

Al igual que `cdrecord` `growisofs` puede hacer uso de una imagen ISO construida con anterioridad de un medio óptico o de un árbol de directorios. Dicha imagen ISO se la especifica al final de la sintaxis de la instrucción `growisofs`.

La licencia de `growisofs` es distribuida bajo GNU GPL. Su creador es Andy Polyakov y la documentación de la utilidad es mantenida por Huub Reuver.

CAPÍTULO III: DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN

3.1. Análisis del sistema.

3.1.1. Descripción general.

La aplicación a desarrollarse es en resumen un Gestor de Aplicaciones, el cual permite la Carga desde un medio extraíble y Descarga hacia un medio extraíble de software con licencia libre o con licencia con fines educativos. Esta aplicación se alojará en un terminal con pantalla táctil y un CPU con características de servidor.

La aplicación será utilizada por el administrador para la carga de información y aplicaciones, gestión de aplicaciones y usuarios, la configuración del sistema; y por el usuario final (estudiantes y docentes) para la descarga de las aplicaciones alojadas en el servidor.

La aplicación debe ser intuitiva, con acceso a teclados virtuales y con procesos simplificados utilizando el potencial de la consola de Linux Ubuntu.

3.1.2. Identificación de las necesidades del negocio

Las necesidades provienen de dos entidades, por una parte el Presidente de la Escuela de Sistemas de la PUCESA y por otro lado el Director de la Escuela de Sistemas; en ambos caso se ha determinado por medio de una entrevista verbal las necesidades del negocio en cuatro grandes grupos:

a) Gestión de Carga de Información

- Incluye un módulo para cargar información relevante de la aplicación a cargarse en el sistema. Dicha información se almacenará en una base de datos.
- Las aplicaciones que se carguen al sistema se almacenan en el disco duro del Servidor.
- Las aplicaciones pueden ser cargadas desde un medio extraíble como memoria flash, disco duro externo o unidades ópticas CD-ROM y DVD-ROM.
- Solo un usuario administrador o con privilegios de carga puede subir la información.

b) Gestión de Descarga de Información

- Incluye un módulo para visualizar las aplicaciones disponibles para descargar desde el Servidor. La información desplegada se lee desde una base de datos.

- No existen restricciones para la descarga de la aplicación desde el terminal.
- Las aplicaciones pueden ser descargadas desde el disco duro del servidor hacia un medio extraíble como memoria flash, disco duro externo o unidades ópticas CD-R y DVD-R.

c) Gestión de Usuarios

- Por defecto existe un usuario administrador que gestiona todas las características del Sistema.
- Se pueden agregar usuarios con privilegios para cargar aplicaciones o crear usuarios sin privilegios.
- Se pueden eliminar usuarios excepto el usuario Administrador.
- Además, se permite modificar los privilegios de un usuario. Los privilegios determinan el rol que juega el usuario dentro del sistema.

d) Administración del Sistema

- La materia y la categoría son información predefinida para clasificar las aplicaciones a almacenarse por ende se precisa la gestión de dicha información de clasificación, es decir añadirla, quitarla y modificarla.
- La información de clasificación será agregada, modificada o eliminada solamente por el administrador.
- Los eventos de ingreso al sistema, cambio en aplicaciones y configuración son eventos que se realizarán sin que el usuario lo note,

sin embargo se necesita visualizarlos debido a que es parte importante de la auditoría del sistema y lo podrá hacer solo el administrador.

- Las aplicaciones que se agregan al sistema podrán ser modificadas o removidas por el usuario administrador solamente.
- Las memorias extraíbles conectadas al sistema deben tener una opción para removerlas de manera segura.

e) Configuración del Sistema

- Dentro de la configuración se encuentra principalmente la gestión de directorios tanto para directorios de almacenaje de aplicaciones como información del directorio de la aplicación.
- Admite la modificación de parámetros como la unidad y velocidad de grabación de CDs, la unidad y velocidad de grabación de DVDs.
- Permite modificar la cadena de conexión con la Base de Datos.
- La activación de la posibilidad de registrar los eventos del sistema es parte también de la configuración del sistema.
- La visualización de los eventos del sistema está incluida debido a que es parte importante de la Administración del sistema.

3.1.3. Análisis de viabilidad.

El personal de desarrollo del proyecto cuenta con los recursos suficientes para iniciar y concluir el proyecto en lo que a recursos materiales concierne.

Por otra parte se cuenta con los recursos técnicos tanto en lo que a hardware y software respecta, ya que las herramientas de desarrollo son software libre y el equipo en el cual se implementará la solución es un dispositivo ya existente en la PUCESA, es por esto que se cumple con lo necesario para cubrir con las necesidades del cliente.

El software a desarrollar cumplirá todas las normas legales en lo relacionado a licencias de uso de programa. La licencia de la aplicación es propiedad de la Pontificia Universidad Católica del Ecuador Sede Ambato.

3.1.4. Asignación de Funciones.

a. Software

Aplicaciones

- Gestionar aplicaciones cargadas al sistema.
- Agregar Aplicaciones al sistema y copiar sus archivos asociados.
- Eliminar Aplicaciones del sistema y eliminar carpeta de archivos asociados.
- Modificar Datos de Aplicaciones.
- Copiar Archivos de Aplicaciones a medios extraíbles.
- Generar reporte de Aplicaciones almacenadas en el sistema.

Usuarios

- Gestionar usuarios

- Agregar Usuarios al sistema.
- Quitar Usuarios del sistema.
- Modificar datos de Usuarios del sistema.
- Grabar permisos de cada Usuario.
- Controlar el rol del Usuario dentro del sistema.
- Generar reporte de Usuarios del sistema.

Eventos del Sistema

- Registra Eventos de Usuario.
- Registra Eventos del Sistema.
- Generar reporte de Eventos del Sistema.

Configurador del Sistema

- Almacena parámetros operativos del sistema.
- Obtiene parámetros operativos del sistema.

b. Hardware

Pantalla táctil del terminal

- Mostrar información de CPU y sistema
- Recibir y captar movimientos del ratón y control táctil de teclado en pantalla.

c. Usuarios

Usuario final.

- Descargar Aplicaciones.

Usuario Editor de Aplicaciones.

- Cargar aplicaciones y sus archivos asociados al sistema gestor de aplicaciones.
- Modificar aplicaciones almacenadas en el sistema gestor de aplicaciones.

Usuario Administrador

- Ingreso y Modificación de Información del sistema.
- Ingreso y Modificación de información de usuarios del sistema.
- Determinar y asignar roles específicos de cada usuario del sistema.
- Auditar eventos del sistema.
- Ingreso, Modificación y Visualización de los parámetros del sistema.

d. Documentación

Documentación que informa sobre los siguientes aspectos

- Manual técnico
- Manual de usuario

e. Información

Este sistema va a manejar tanto el almacenamiento de archivos aplicaciones como la información de las mismas, que se han cargado en el sistema; por otra parte copiará de manera autónoma los archivos de las aplicaciones que

han sido seleccionadas por el usuario final para su descarga a medios extraíbles.

Finalmente por medio del módulo de administración del sistema se puede obtener reportes que genera la aplicación tanto en el aspecto de los eventos que se generan por el uso del sistema de gestión de aplicaciones como los usuarios y parámetros propios del sistema.

f. Procesos

Los procesos propios del sistema de gestión de aplicaciones del kiosco tecnológico están separados en 4 módulos claramente identificables. El primer módulo es el sistema gestor de aplicaciones, el cual se encarga en sí de la administración de la información de las aplicaciones cargadas al sistema y su información relacionada como es el caso del nombre de la aplicación, autores, vistas previas, manuales de usuario, etc. El segundo módulo se encarga de la administración de usuarios tanto para lo que es la gestión de usuarios, es decir, añadir, eliminar, modificar; y control de usuarios a nivel de acceso del sistema por medio de los permisos de usuario, asignándole por ende roles a cada usuario del sistema.

En cuanto al tercer módulo se puede decir que se encarga de los sucesos que ocurren en el sistema, almacenando cada evento para su posterior auditoria y de igual manera la generación de reportes; es así que se puede determinar cuándo y quién subió una aplicación al sistema, que aplicaciones

se encuentran en el sistema, cuantas veces se ha descargado determinada aplicación, accesos al sistema, etc.

Finalmente el cuarto módulo es netamente una característica del sistema que le permite obtener parámetros que han sido configurados por el Administrador para su correcto funcionamiento por ejemplo las unidades de DVD, rutas de almacenamiento de archivos, etc. Y claro está el administrador puede a cualquier momento hacer cambios de estos parámetros de operación del sistema, los cuales se almacenan en la Base de Datos.

3.1.5. Características de usuario.

Existen 3 roles dentro del sistema, descritos por su jerarquía se encuentra en primer lugar el Usuario Administrador quien puede tener acceso a reportes y configuración del sistema. Se lo identifica con el nombre de Administrador y posee una clave de paso para su identificación y control.

En segundo lugar se encuentra el Usuario Editor de Aplicaciones, el mismo que como su nombre lo indica puede cargar, modificar y eliminar aplicaciones; además puede, ver reportes de las aplicaciones almacenadas. Pueden existir uno o más editores del sistema y se los identifica con su respectivo nombre de usuario y clave de paso.

El último en jerarquía es el Usuario Final quien hace uso permanente del sistema, puede explorar las aplicaciones cargadas en el kiosco tecnológico para su posterior descarga a medios extraíbles. Este usuario no posee identificación ni claves de acceso.

Cabe señalar que el administrador puede asumir el rol de editor de aplicaciones, y de usuario final. El editor de aplicaciones puede asumir también el rol de usuario final pero no de administrador y finalmente el usuario final no puede asumir ningún rol de los restantes.

3.1.6. Restricciones.

En cuanto a las restricciones de usuario tenemos que el administrador es un único usuario dentro del sistema, es decir no hay 2 o más administradores.

El administrador no puede cambiar su nombre de usuario más sí su contraseña.

Para el almacenamiento de aplicaciones, la única restricción que determina la cantidad de aplicaciones almacenadas es directamente proporcional al tamaño de almacenamiento físico de los discos duros del servidor.

No se puede modificar los archivos de las aplicaciones almacenadas, de existir un error con los archivos cargados, debe eliminarse la aplicación del

registro de la base de datos y sus archivos almacenados en disco para su carga posterior con los archivos correctos puesto que la aplicación no controla los archivos almacenados por el usuario; es decir el contenido de la aplicación que se carga, lo que si hace es efectuar controles sobre sus procesos como la copia de archivos.

3.1.7. Definición del sistema.

DIAGRAMA DE ARQUITECTURA DEL SISTEMA - NIVEL 0

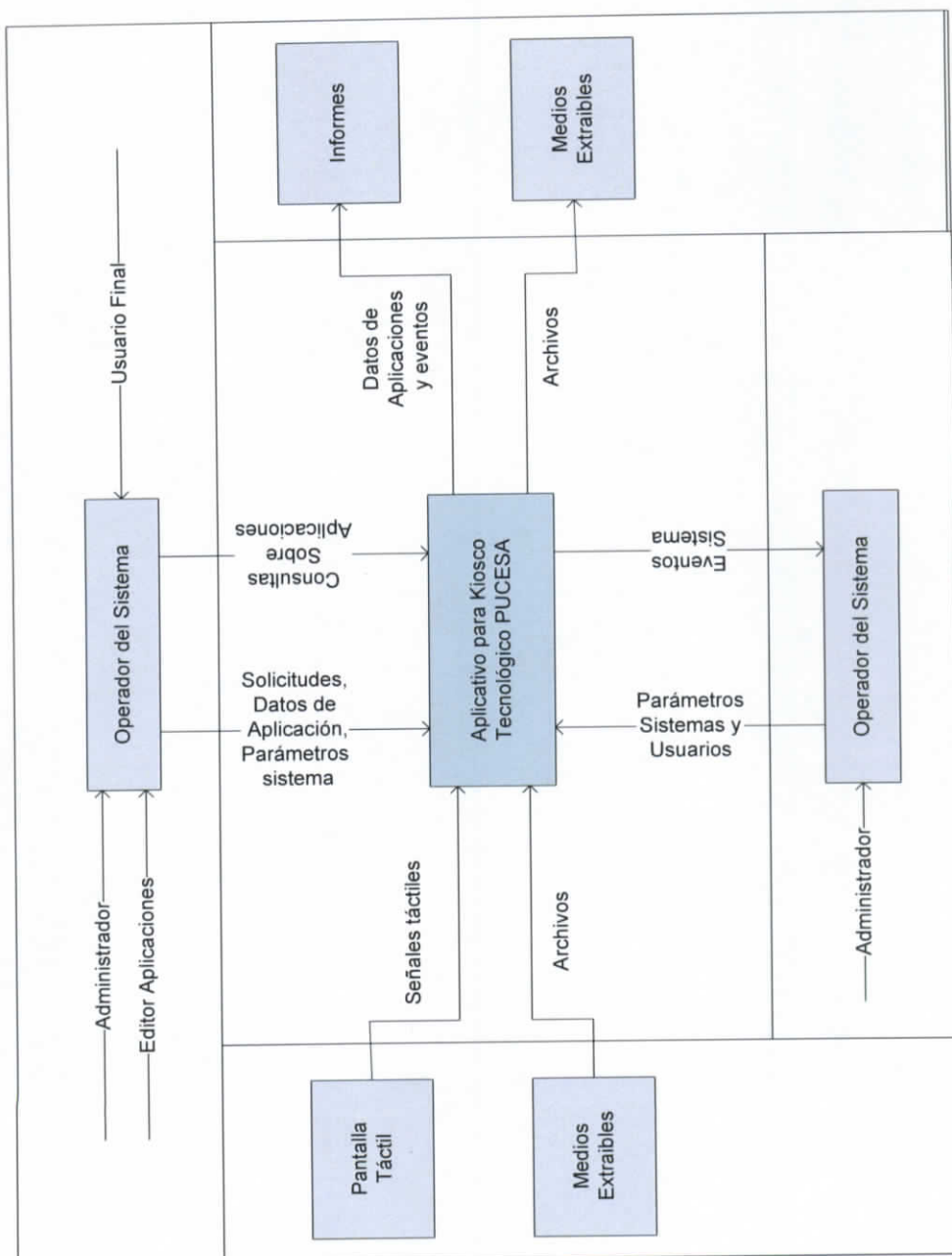
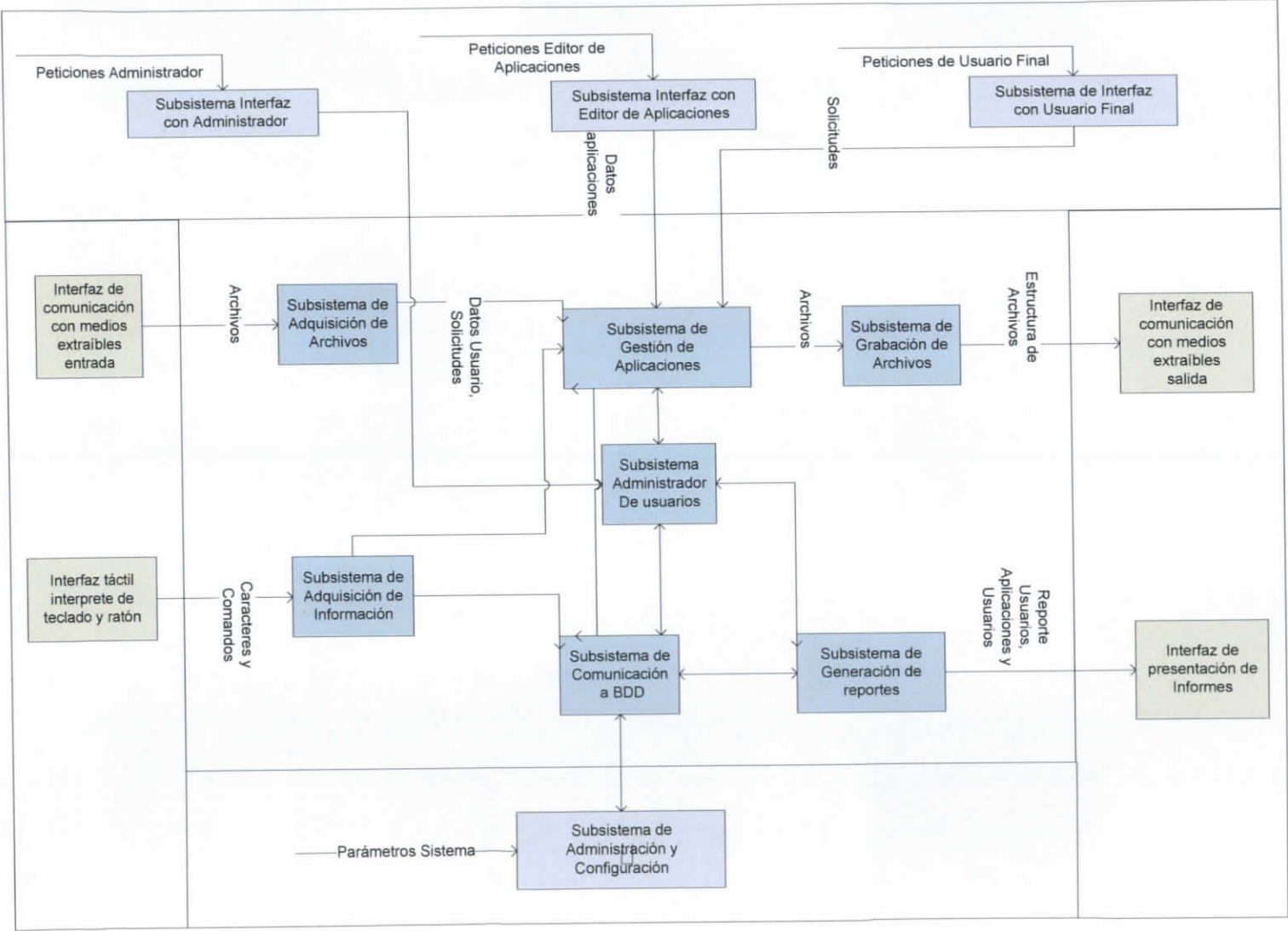


Ilustración 3. 1: Diagrama de arquitectura de sistema en nivel 0.

Ilustración 3. 2: Diagrama de arquitectura de sistema en nivel 1.



NIVEL 1

3.2. Diseño del Sistema.

3.2.1. Introducción.

La fase de diseño contempla tanto el diseño de la aplicación tanto en los procesos, subprocesos, flujo de información y los datos en sí que recibe, procesa y emite el sistema; y por otro lado, el diseño de la base de datos que almacena la información de uso del sistema y útil para el usuario.

Para esta etapa se procede con una descripción de los módulos detallados en el diagrama de arquitectura Nivel 1, uso de la técnica de DFD como análisis estructurado del sistema, detalle de procesos con diagrama de procesos, diccionario de datos y por último para la base de datos su descripción por medio del modelo lógico y físico.

3.2.2. Descripción de módulos del sistema.

a. Subsistema de Interfaz con el Administrador

Descripción:

Es la interfaz del software que comunica el sistema con el administrador para el ingreso de la información y configuración del sistema y usuarios.

Datos a manejar:

Este subsistema va a manejar datos numéricos, así como cadena de caracteres.

Fuente de entrada de datos:

Peticiones del Administrador

Receptor de datos de salida

Fuente de entrada de datos de:

- Subsistema de Administración de Usuarios.

Restricciones del diseño

- Interfaz de Software amigable y sencilla aplicando paneles y teclado en pantalla.
- Comunicación con módulos exclusivos del sistema.
- Validación de datos de ingreso al sistema.

b. Subsistema de Interfaz con Editor de Aplicaciones

Descripción:

Es la interfaz del software que comunica el sistema con el editor de aplicaciones para el ingreso de la información de las aplicaciones cargadas y a cargarse en el sistema.

Datos a manejar:

Este subsistema va a manejar datos numéricos, así como cadena de caracteres.

Fuente de entrada de datos:

Peticiones del Editor de Aplicaciones.

Receptor de datos de salida

Fuente de entrada de datos de:

- Subsistema de Gestión de Aplicaciones.

Restricciones del diseño

- Interfaz de Software amigable y sencilla aplicando paneles y teclado en pantalla.
- No se comunica con módulos de configuración del sistema.
- Validación de datos de ingreso al sistema.

c. Subsistema de Interfaz con Usuario Final

Descripción:

Es la interfaz del software que comunica el sistema con el usuario final para la consulta de aplicaciones almacenadas en el sistema y su posterior descarga a medios extraíbles.

Datos a manejar:

Este subsistema va a manejar datos en formato de cadena de caracteres.

Fuente de entrada de datos:

Peticiones del Usuario Final

Receptor de datos de salida

Fuente de entrada de datos de:

- Subsistema de Gestión de Aplicaciones.

Restricciones del diseño

- Interfaz de Software amigable y sencilla, con menús y paneles intuitivos, teclado en pantalla.
- Restricción de acceso solo a módulos descarga de aplicaciones.

- Paneles de búsqueda de criterios.
- Descarga de Archivos a medios extraíbles por medio de interfaz intuitiva y procesos transparentes de grabación.

d. Subsistema de Interfaz táctil interprete de teclado y ratón.

Descripción:

Se encarga de recoger los toques del usuario en la matriz digitalizadora para convertirlos en acciones de teclado y mouse dentro del sistema operativo permitiendo interactuar al usuario con el sistema.

Datos a manejar:

Datos transparentes al sistema, código binario interpretado por el sistema operativo.

Fuente de entrada de datos:

Toques de la matriz digitalizadora como impulsos eléctricos.

Receptor de datos de salida

Fuente de entrada de datos de:

- Subsistema de Adquisición de Información

Restricciones del diseño

- Comunicación directa entre hardware y software por medio de controladores no privativos
- Transparencia total con el usuario.

e. Subsistema de Interfaz de Comunicación con Medios Extraíbles Entrada

Descripción:

Esta interfaz permite comunicar a los medios extraíbles, tanto como almacenamiento masivo USB como unidades ópticas a nivel de hardware con el sistema operativo.

Datos a manejar:

Este subsistema maneja datos transparentes al sistema, código binario para transmisión de entrada (input) con dispositivos de hardware externos.

Fuente de entrada de datos:

Archivos de unidades extraíbles ópticas y USB de almacenamiento masivo.

Receptor de datos de salida

Fuente de entrada de datos de:

- Subsistema de Adquisición de archivos.

Restricciones del diseño

- Comunicación directa entre hardware y software por medio de controladores no privativos.
- Transparencia total con el usuario.

f. Subsistema de Adquisición de Información.

Descripción:

El Subsistema de Adquisición de Información se encarga de traducir los comandos del usuario provenientes de la interfaz táctil a órdenes o acciones para ejecutarse en otros módulos del sistema.

Datos a manejar:

Datos transparentes al sistema, datos de caracteres ASCII y comandos de ratón.

Fuente de entrada de datos:

Comandos de teclado y mouse provenientes de la Interfaz táctil interprete de teclado y ratón.

Receptor de datos de salida

Fuente de entrada de datos de:

- Subsistema de Gestión de Aplicaciones.

Restricciones del diseño

- Comunicación directa entre el Sistema Operativo y la Aplicación
- Transparencia total con el usuario.

g. Subsistema de Adquisición de Archivos.

Descripción:

Esta interfaz adquiere los archivos de la interfaz de hardware de comunicación con medios extraíbles para almacenarlas en carpetas predestinadas para el almacenamiento de los archivos de las aplicaciones.

Datos a manejar:

Paquetes y Archivos de las aplicaciones a cargarse en el sistema, cadenas de caracteres para instrucciones de consola.

Fuente de entrada de datos:

Archivos provenientes desde la interfaz de comunicación con medios extraíbles.

Receptor de datos de salida:

Fuente de entrada de datos de:

- Subsistema de Gestión de Aplicaciones.

Restricciones del diseño:

- Selección de archivos solo para unidades extraíbles como almacenamiento masivo y unidades ópticas.
- Copia de archivos al almacenamiento físico del sistema de manera transparente con el usuario.

h. Subsistema de Administración y Configuración del Sistema.

Descripción:

El configurador del sistema establece los parámetros de operación del sistema, estos parámetros son preestablecidos por el Administrador a través de la interfaz gráfica de administración y configuración.

Datos a manejar:

Recibe cadena de caracteres y datos numéricos.

Fuente de entrada de datos:

Parámetros establecidos por el administrador.

Receptor de datos de salida:

Fuente de entrada de datos de:

- Subsistema de Comunicación a Base de Datos BDD.

Restricciones del diseño:

- Parámetros configurables solamente para el Usuario Administrador.
- Existen parámetros no configurables solo para visualización.

- La configuración se almacenara en la base de datos propia del sistema.
- Consta de una interfaz será sencilla y estructurada para el Administrador del sistema.

i. Subsistema de Gestión de Aplicaciones.

Descripción:

Es el módulo principal de la aplicación a desarrollarse, gestiona la carga y descarga de las aplicaciones almacenadas en el sistema, por una parte administrando la información de las aplicaciones y por otro lado administrando los archivos de las aplicaciones almacenadas en el sistema. Crea un catálogo para la descarga de las aplicaciones disponibles en el sistema.

Datos a manejar:

Cadenas de caracteres, paquetes y archivos de las aplicaciones a cargarse en el sistema.

Fuente de entrada de datos:

- Información de aplicación proveniente desde el Subsistema Interfaz con Editor de Aplicaciones.
- Solicitudes de Exploración provenientes desde el Subsistema de Interfaz con Usuario Final.
- Archivos de Aplicaciones provenientes del Subsistema de Adquisición de Archivos.
- Parámetros operativos del sistema desde el Subsistema de Comunicación a BDD.

- Control de paso de usuarios desde el Subsistema Administrador de Usuario.

Receptor de datos de salida:

Fuente de entrada de datos de:

- Subsistema de Grabación de Archivos.
- Subsistema Administrador de Usuarios
- Subsistema de Comunicación a BDD.

Restricciones del diseño:

- Comunicación con el usuario con una interfaz gráfica intuitiva y sencilla.
- Comunicación con el Sistema Operativo a partir de comandos del sistema Operativo.
- Almacenamiento catalogado por medio de registro en la base de datos.
- Comprobación de espacio disponible libre en disco duro de servidor.
- Acceso para carga de información solamente para usuarios con privilegio de Edición de Aplicaciones.
- Usuario final con derecho a descarga sin necesidad de autenticación.

j. Subsistema Administrador de Usuarios.

Descripción:

El módulo para la administración de Usuarios se encarga de dos funciones, la primera que es la gestión de usuarios es decir la creación, modificación y la eliminación de usuarios del sistema específicamente de la base de datos y por otra parte la función de autenticar a los usuarios en el sistema y otorgar los permisos en base al rol que tiene cada usuario.

Datos a manejar:

Cadenas de caracteres.

Fuente de entrada de datos:

- Solicitud de acceso para usuarios desde el Subsistema de Gestión de Aplicaciones.
- Datos de Usuario a partir del Subsistema Interfaz con Administrador.
- Datos de Usuario a partir del Subsistema de Comunicación a BDD.
- Solicitud de acceso para usuarios desde el Subsistema de Generación de Reportes.

Receptor de datos de salida:

Fuente de entrada de datos de:

- Subsistema de Gestión de Aplicaciones.
- Subsistema de Comunicación a BDD.
- Subsistema de Generación de Reportes.

Restricciones del diseño:

- Los usuarios pueden eliminarse, modificarse o ser añadidos únicamente por el usuario Administrador.
- No debe permitir la modificación del usuario Administrador.
- Debe validar a los usuarios dentro de la Aplicación para que cumplan el rol que se les ha sido asignado.
- Todos los datos de usuario deben almacenarse en la base de datos del sistema.

k. Subsistema de Comunicación a BDD.

Descripción:

Este subsistema es el encargado de recibir información y guardarla en una base de datos para su uso ulterior por parte del usuario y del sistema. La comunicación con el gestor de base de datos es parte de las funciones de este módulo.

Datos a manejar:

Cadenas de caracteres, datos numéricos, booleanos y fechas.

Fuente de entrada de datos:

- Información de las aplicaciones a almacenarse de parte del Subsistema de Adquisición de Información.
- Solicitudes del Subsistema de Gestión de Aplicaciones
- Datos de Usuario provenientes del Subsistema Administrador de Usuarios.
- Solicitud de información desde el Subsistema de Generación de Reportes.
- Parámetros de configuración del sistema por parte del Subsistema de Administración y Configuración.

Receptor de datos de salida:

Fuente de entrada de datos de:

- Subsistema de Gestión de Aplicaciones.
- Subsistema de Administración de Usuarios.
- Subsistema de Generación de Reportes.
- Subsistema de Administración y Configuración.

Restricciones del diseño:

- La comunicación con el Sistema Gestor de Base de Datos es transparente al Usuario.
- Los datos deben ser validados antes de enviarlos a la base de datos.
- Solo existe una sola base de datos.
- El límite del almacenamiento es una restricción del Sistema Gestor de Base de Datos.

I. Subsistema de Grabación de Archivos.

Descripción:

Es un subsistema del sistema que se comunica el sistema operativo del servidor de aplicaciones para permitir por medio de comandos consola la grabación de los archivos de la aplicación solicitada por el usuario final a un destino específico que puede ser discos de almacenamiento masivo USB o unidades ópticas.

Datos a manejar:

Archivos de las aplicaciones y cadena de caracteres para instrucciones de consola.

Fuente de entrada de datos:

Archivos de Aplicaciones desde el Subsistema de Gestión de Aplicaciones.

Receptor de datos de salida:

Fuente de entrada de datos de:

- Interfaz de comunicación con medios extraíbles.

Restricciones del diseño:

- El sistema debe ocupar los comandos necesarios y hacer uso de rutinas de grabación a partir del tipo de medio de salida escogido por el Usuario Final.
- El proceso de grabación muestra progresos pero es transparente al usuario.
- Solo se puede descargar una aplicación a la vez.
- Control de espacio disponible en medios extraíbles USB.
- En el caso de unidades ópticas debe escoger automáticamente el medio adecuado para la grabación a partir del tamaño de la aplicación a descargarse.

m. Subsistema de Generación de Reportes.

Descripción:

El módulo para la Generación de Reportes como su nombre lo indica obtiene información sobre 3 aspectos como es las aplicaciones almacenadas dentro del sistema, los usuarios del sistema con sus respectivos eventos y los parámetros operacionales del sistema para formatear dicha información y presentarla al usuario de manera visual.

Datos a manejar:

Cadenas de caracteres y datos numéricos.

Fuente de entrada de datos:

- Control de paso de usuarios desde el Subsistema Administrador de Usuarios.

- Datos de Usuario a partir del Subsistema de Comunicación a BDD.
- Datos de Aplicaciones almacenadas en el sistema a partir del Subsistema de Comunicación a BDD.
- Datos sobre eventos acontecidos en el sistema generado por otros módulos.

Receptor de datos de salida:

Fuente de entrada de datos de:

- Subsistema de Administración de Usuarios.
- Subsistema de Comunicación a BDD.
- Interfaz de presentación de informes.

Restricciones del diseño:

- Los tipos de reportes disponibles serán de acuerdo al tipo de Usuario, es decir del rol que le pertenece al usuario que ejecuta una solicitud de reporte.
- Los reportes son formateados para ser presentados en formularios visuales para el usuario a través de la interfaz de presentación de informes.
- Debe permitir filtrado de criterios en reportes donde se incluya fechas.

n. Subsistema de Interfaz de Comunicación con Medios Extraíbles Salida

Descripción:

Esta interfaz permite comunicar a los medios extraíbles, tanto como almacenamiento masivo USB como unidades ópticas a nivel de hardware con el sistema operativo.

Datos a manejar:

Este subsistema maneja datos transparentes al sistema, código binario para transmisión de salida (output) con dispositivos de hardware externos.

Fuente de entrada de datos:

Estructuras de Archivos desde el Subsistema de Grabación de Archivos.

Receptor de datos de salida

Medios físicos de Unidades ópticas o unidades de almacenamiento masivo USB.

Restricciones del diseño

- Comunicación directa entre hardware y software por medio de controladores no privativos.
 - Transparencia total con el usuario.
- o. Interfaz de presentación de Informes.

Descripción:

La interfaz de presentación de informes es en sí la interfaz visual que permite al usuario visualizar tanto la interfaz gráfica del sistema como los datos o reportes que el mismo contenga.

Datos a manejar:

Datos de comunicación en código binario transparente al usuario entre el CPU y la pantalla LCD.

Fuente de entrada de datos:

- Reporte de aplicaciones almacenadas en el servidor desde el Subsistema de Generación de Reportes.

- Reporte de parámetros operativos del sistema desde el Subsistema de Generación de Reportes.
- Reporte de usuarios del sistema desde el Subsistema de Generación de Reportes.

Receptor de datos de salida:

Pantalla LCD para mostrar la interfaz grafica al Usuario.

Restricciones del diseño:

- Los informes que se presentan en pantalla deben ser coherentes y de fácil navegación en caso de existir mucha información
- Todo lo que respecta a interfaz gráfica de la aplicación debe estar ajustada a la resolución de salida de imagen del hardware de presentación de video (LCD).

3.2.3. Diagramas de flujo de datos DFD

DIAGRAMA DE FLUJO DE DATOS

NIVEL 0

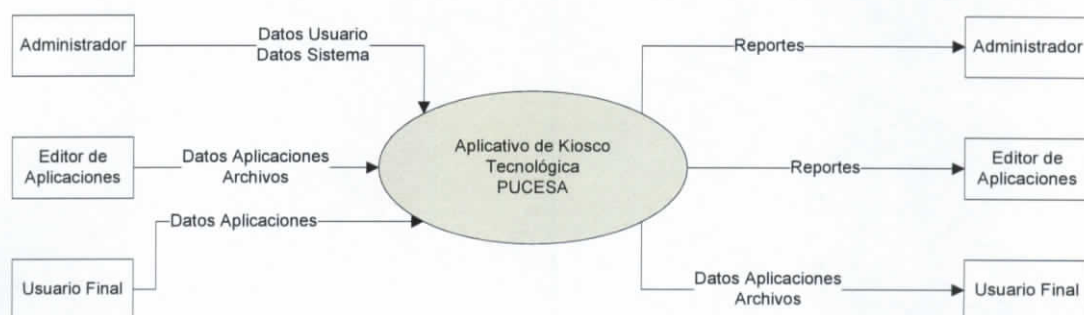


Ilustración 3. 3: DFD del sistema en nivel 0.

NIVEL 1

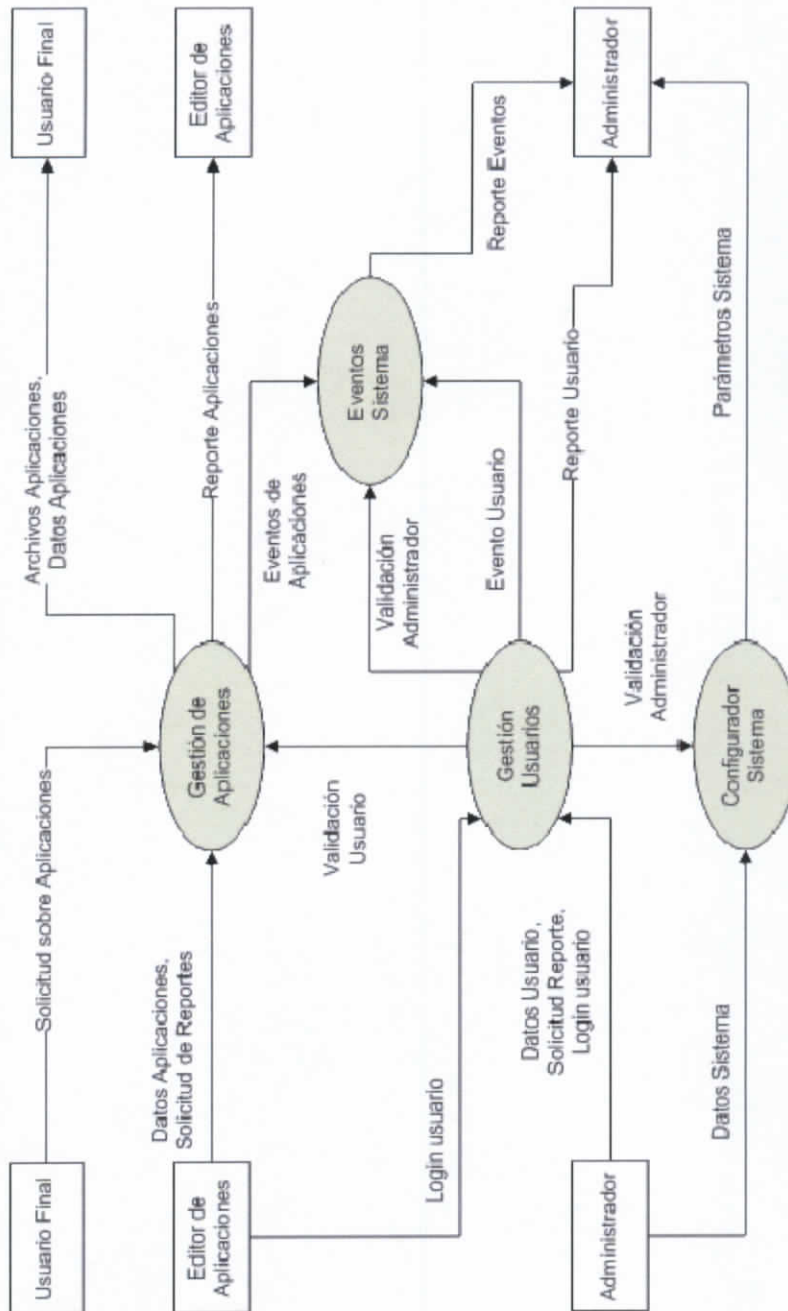


Ilustración 3. 4: DFD del sistema en nivel 1.

NIVEL 2.- Gestión de Aplicaciones

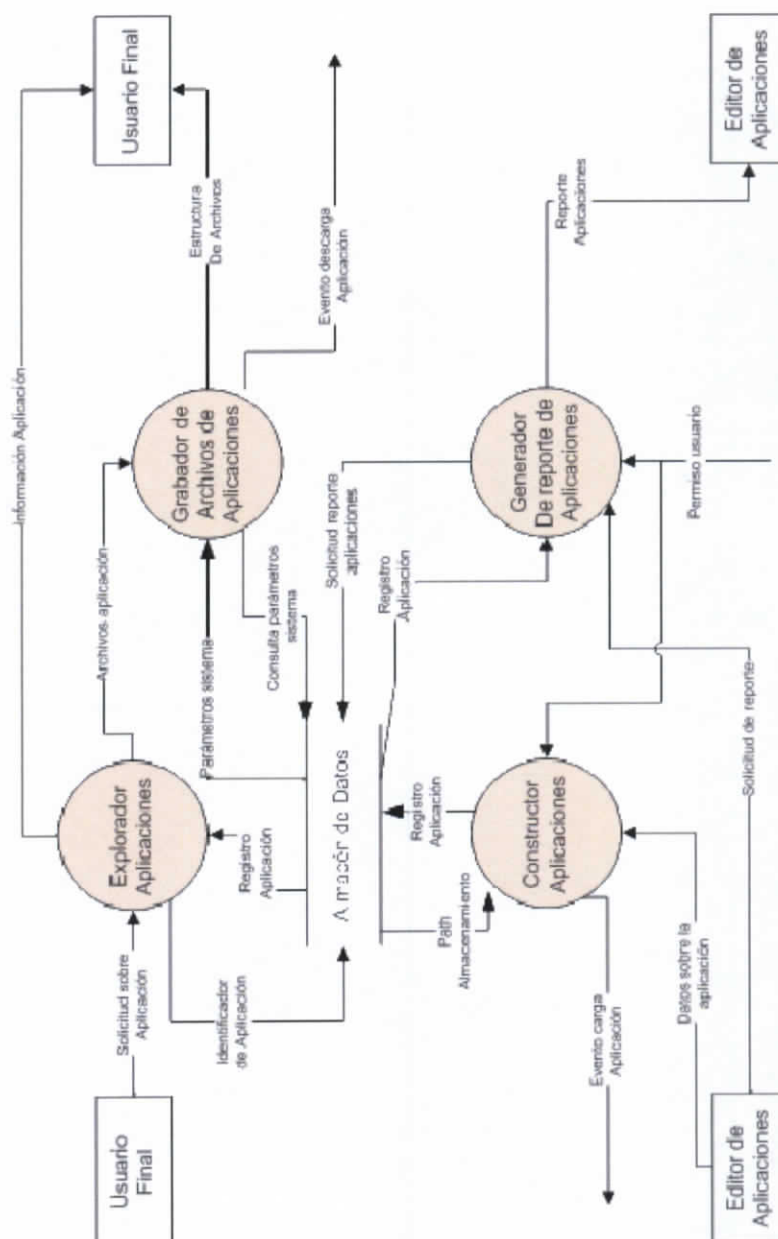


Ilustración 3. 5: DFD de proceso gestión de aplicaciones en nivel 2.

NIVEL 2.- Gestión de Usuarios

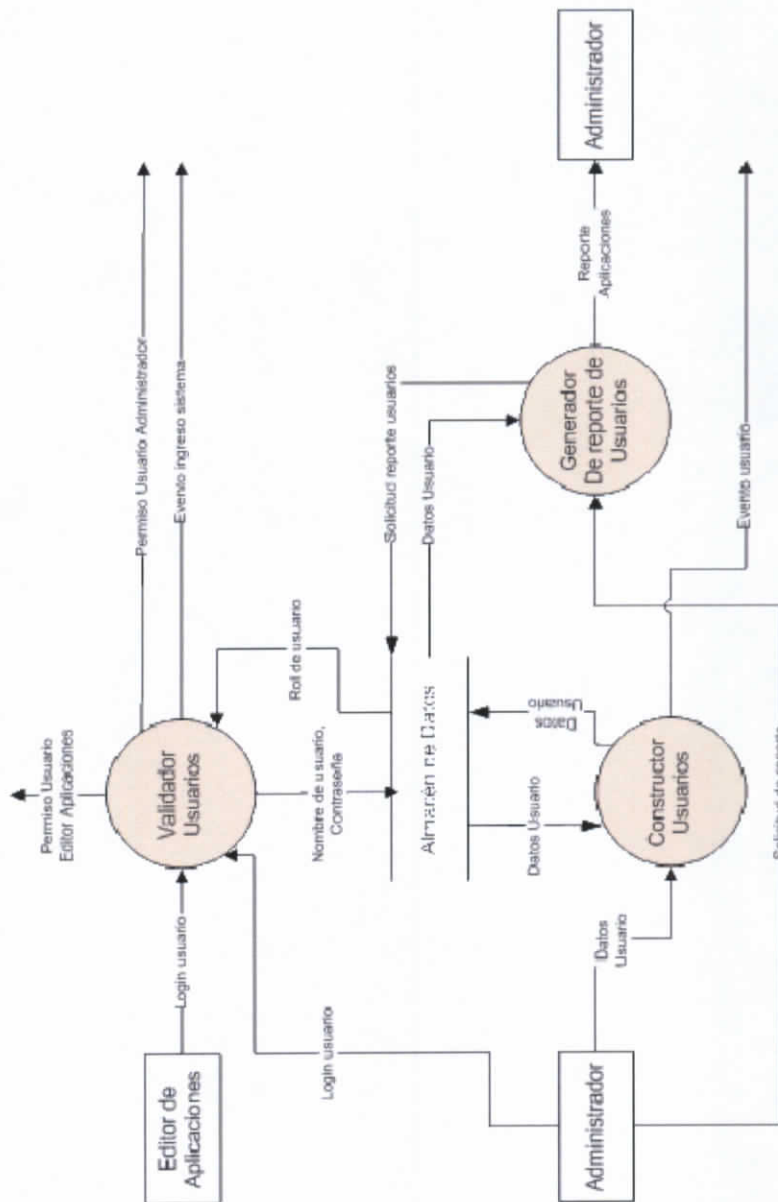


Ilustración 3. 6: DFD de proceso gestión de usuarios en nivel 2.

NIVEL 2.- Configurator Sistema

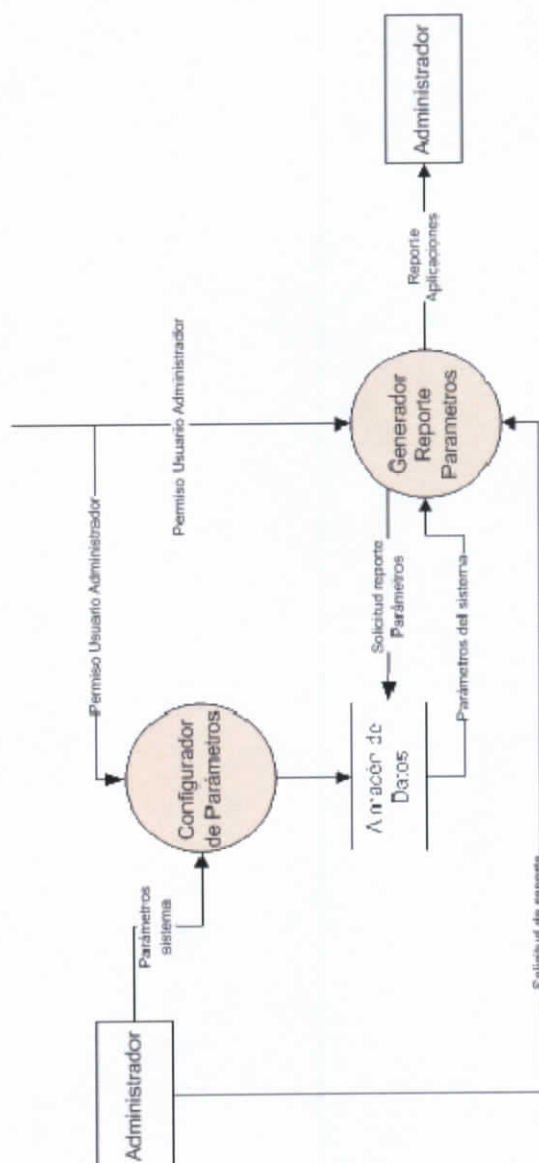


Ilustración 3. 7: DFD de proceso configurador de sistema en nivel 2.

NIVEL 2.- Eventos Sistema

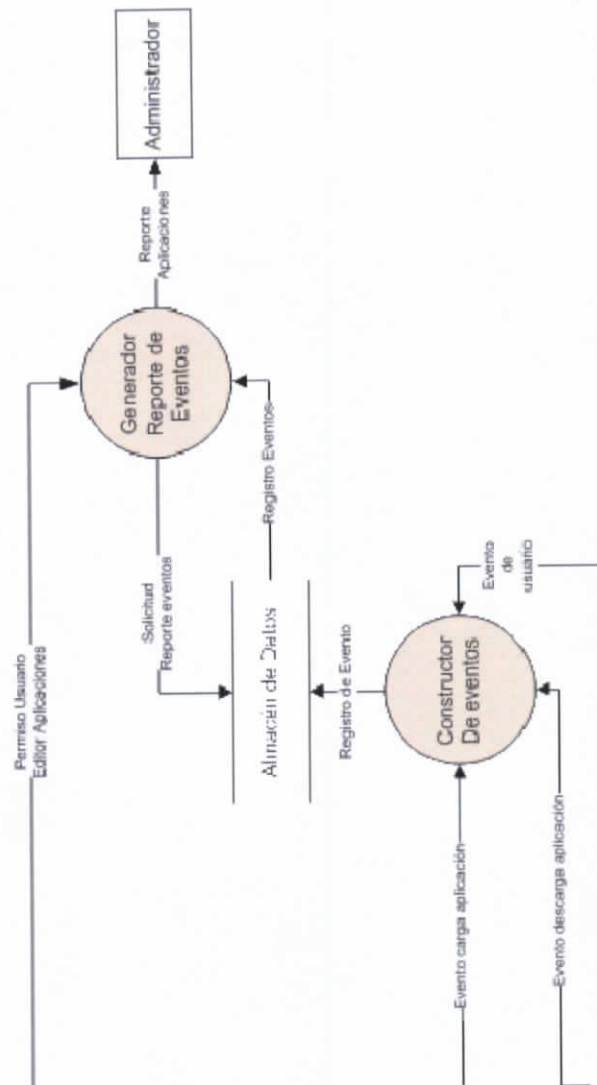


Ilustración 3. 8: DFD de proceso eventos sistema en nivel 2.

3.2.4. Diccionario de Procesos.

Gestión de Aplicaciones

Explorador Aplicaciones	
Datos de Entrada:	Solicitud sobre aplicación, Registro Aplicación.
Datos de Salida:	Identificador Aplicación, Información Aplicación, Archivos Aplicación
Función o Proceso:	Lee la solicitud de nombre de un programa o característica del mismo, consulta en la base de datos y devuelve el registro con información detalla de aplicación, rutas de acceso y archivos almacenados para su grabación
Almacén de Datos:	(Registro Aplicaciones). Se utiliza un almacén de datos para obtener y construir un catálogo visual de las aplicaciones almacenadas en el sistema. Lee un registro completo con el código, nombre de aplicación, ruta de almacenamiento de archivos y demás características de la aplicación almacenada

Tabla 3. 1: Proceso explorador aplicaciones.

Constructor Aplicaciones	
Datos de Entrada	Datos sobre la aplicación, Permiso de Usuario, Path Almacenamiento.
Datos de Salida:	Registro Aplicación, Evento carga Aplicación
Función o Proceso:	Recibe los datos informativos de la Aplicación a almacenarse dentro del sistema, construye el registro y lo envía al almacén de datos para su lectura en ocasiones posteriores. Asocia los archivos de la aplicación y los registros, los copia al sistema y los almacena en un directorio preestablecido para dicho propósito recuperado desde la base de datos, Además controla que solo cierto rol de usuario pueda cargar información al sistema y permite la modificación de información
Almacén de Datos	(Registro Aplicaciones). Se utiliza un almacén de datos guardar de manera estructurada la información referente a las aplicaciones a almacenarse en el sistema. Contiene la ruta dentro del disco duro del servidor donde se alojan los archivos de cada una de las aplicaciones cargadas en el servidor

Tabla 3. 2: Proceso constructor aplicaciones.

Grabador de Archivos de Aplicaciones	
Datos de Entrada	Archivos aplicación, Parámetros sistema
Datos de Salida:	Consulta parámetros Sistema, Estructuras de Archivos, Evento descarga aplicación
Función o Proceso:	Una vez que se ha seleccionado la aplicación a descargarse el proceso de Exploración de Aplicaciones envía los archivos necesarios para iniciar la grabación de los mismos en medios extraíbles. El tipo de medio es escogido por el usuario y a partir de esa decisión el proceso automáticamente direcciona los archivos al dispositivo de entrada/salida que contiene el medio extraíble, además comprueba el tamaño disponible en el medio extraíble antes de grabar archivos y también el tipo de unidad óptica a utilizarse dependiendo del tamaño de los archivos a grabarse. Los parámetros de grabación utilizados en este proceso y recuperados de la base de datos.
Almacén de Datos	(Registro Parámetros sistema). Se utiliza un almacén de datos para obtener los parámetros del sistema como es la unidad de grabación la velocidad de grabación dependiendo del tipo de dispositivo de entrada/salida que se utilice al momento de la grabación de archivos.

Tabla 3. 3: Proceso grabador de archivos de aplicaciones.

Generador de Reporte de Aplicaciones	
Datos de Entrada	Registro Aplicación, Solicitud de Reporte, Permiso Usuario.
Datos de Salida:	Solicitud reporte aplicaciones, Reporte Aplicaciones
Función o Proceso:	El proceso se encarga de recuperar los registros de la base de base de datos para mostrarlos de manera visual al usuario, el cual, debe ser identificado como un rol de edición de aplicaciones o administrador; permitiendo conocer las aplicaciones que han sido cargadas en el sistema para visualización o borrado de las aplicaciones ya cargadas en el sistema
Almacén de Datos	(Registro Parámetros sistema). Se utiliza un almacén de datos para obtener los parámetros del sistema como es la unidad de grabación la velocidad de grabación dependiendo del tipo de dispositivo de entrada/salida que se utilice al momento de la grabación de archivos

Tabla 3. 4: Proceso generador de reporte de aplicaciones

Gestión de Usuarios

Validador Usuarios	
Datos de Entrada	Login Usuario, Rol usuario.
Datos de Salida:	Permiso Usuario Editor Aplicaciones, Usuario, Contraseña, Permiso usuario Administrador, Evento Ingreso sistema
Función o Proceso:	El Validador de usuarios procesa los datos de entrada para identificación de usuarios determinados en la arquitectura del sistema como login de usuario, obtiene el usuario y la contraseña y compara con los registros almacenados en la base de datos; luego se encarga de determinar el rol que juega cada usuario en el sistema (administrador, editor de aplicaciones) y otorga el permiso para que otros procesos realicen sus funciones. Además genera eventos de ingreso al sistema
Almacén de Datos	(Registro usuario). Se utiliza un almacén de datos para comparar el nombre de usuario y contraseña proporcionados por el usuario con el registro del almacén, de ser positivo también se extrae los permisos efectivos de cada usuario guardados con anterioridad en el almacén

Tabla 3. 5: Proceso validador usuarios.

Constructor Usuarios	
Datos de Entrada	Datos Usuario.
Datos de Salida:	Registro Usuario, Evento Usuario
Función o Proceso:	En este proceso el sistema permite la creación y modificación de usuarios al introducir el nombre de usuario, contraseña y el rol que le pertenece para su identificación en el sistema. Crea un registro y lo envía a la base de datos para validaciones por el proceso anterior. Cada vez que se ejecuta un proceso en el constructor crea un evento con información del tipo de función realizada
Almacén de Datos	(Registro usuario). Se utiliza un almacén de datos para almacenar el nombre de usuario, contraseña y los permisos efectivos de cada usuario para determinar su rol

Tabla 3. 6: Proceso constructor usuarios.

Generador de reporte de usuarios	
Datos de Entrada	Registro usuario, Solicitud de Reporte.
Datos de Salida:	Solicitud reporte usuarios, Reporte usuarios
Función o Proceso:	La función primaria es informar al administrador por un reporte visual de los usuarios que se encuentran registrados en el sistema y los permisos que tienen cada uno de ellos. El reporte es solo visible para el Administrador por lo que hace uso de un control en base a la solicitud de reporte exclusiva del Administrador, Al tener registros activos el proceso además ejecuta instrucciones de eliminación de usuarios restringiendo el borrado del almacén de datos del usuario Administrador
Almacén de Datos	(Registro usuario). Se utiliza un almacén de datos para leer los registros de información de usuarios los cuales contienen datos como el nombre de usuario, contraseña, permisos de edición, permisos de borrado y facilita la modificación de dicha información por parte del administrador

Tabla 3. 7: Proceso generador de reporte de usuarios

Configurador sistema

Configurador parámetros	
Datos de Entrada	Parámetros sistema, Permiso usuario administrador.
Datos de Salida:	Registro parámetros
Función o Proceso:	Se encarga de recibir los parámetros operacionales del sistema introducidos por el usuario Administrador para configurar el Sistema Gestor de Aplicaciones, todos los parámetros se almacenan en la base de datos. De igual manera permite introducir nuevos valores y cambiar los parámetros existentes
Almacén de Datos	(Registro parámetros). Existe un registro único con los parámetros operacionales del sistema. Los parámetros dentro de la base son obligatorios para evitar el mal funcionamiento del sistema

Tabla 3. 8: Proceso configurador sistema

Generador de reporte parámetros	
Datos de Entrada	Parámetros sistema, Permiso usuario administrador, Solicitud de reporte.
Datos de Salida:	Reporte parámetros
Función o Proceso:	Genera un reporte visual para el control del administrador sobre los parámetros operacionales en los cuales se basan los otros procesos del sistema. El reporte visual permite además modificar sus valores de manera que automática se reflejen los cambios en la base, cada vez que los parámetros sean cambiados debe emitir un aviso de reinicio de sistema. El reporte que genera el proceso es a partir de la lectura del registro único de la base de datos y es accedido solo por el administrador; es decir implementa un control de usuario
Almacén de Datos	(Registro parámetros). Lee el registro único con los parámetros operacionales del sistema, los parámetros que maneja son unidad de dvd, unidad de cd, velocidad de grabación de cd y dvd, path de almacenamiento de aplicaciones, campos de verdad para registro de eventos de usuario y sistema

Tabla 3. 9: Proceso generador de reporte parámetros.

Eventos Sistema

Constructor de Eventos	
Datos de Entrada	Evento carga aplicación, evento descarga aplicación, evento de usuario.
Datos de Salida:	Registro de evento
Función o Proceso:	A este proceso llegan cadenas de caracteres indicando lo acontecido dentro de 3 procesos fundamentales del sistema como es la carga de aplicaciones, la descarga de aplicaciones y eventos de usuario, adjunta la fecha al momento de suceder el evento en el sistema y almacena dicha información en la base de datos para auditorias Por otro lado incluye un proceso de control de eventos de ingresos del sistema transparentes al manejo del usuario, de igual manera adjunta la fecha de ocurrencia y crea al registro para su almacenamiento
Almacén de Datos	(Registro eventos). Se utiliza el almacén de datos para guardar un registro que contiene el evento producido, la fecha, el modulo del cual proviene y el usuario que lo disparó en caso de ser un evento del usuario

Tabla 3. 10: Proceso constructor de eventos.

Generador reporte de eventos	
Datos de Entrada	Permiso usuario Administrador, Registro eventos, Solicitud reporte.
Datos de Salida:	Solicitud reporte eventos, Reporte eventos
Función o Proceso:	La función de este componente es generar un reporte visual para el administrador de los eventos que se generaron dentro del sistema, obtiene los registros guardados en el almacén de datos y los presenta formateados en pantalla
Almacén de Datos	(Registro parámetros). El almacén de datos se lo utiliza para recuperar los registros de eventos registro que contiene el evento producido, la fecha, el modulo del cual proviene y el usuario que lo disparó en caso de ser un evento del usuario

Tabla 3. 11: Proceso generador reporte de eventos.

3.2.5. Diseño Arquitectónico.

3.2.5.1. Análisis de Procesos

GESTION DE APLICACIONES

Explorador aplicaciones.

- a. Consulta registro: Se encarga de consultar a la base de datos sobre un registro que contiene los datos informativos de una aplicación cargada al sistema, dependiendo de los filtros puede hacer una búsqueda por un criterio de nombre o lenguaje de programación. Permite consultar y obtener todos los datos de la aplicación incluyendo el directorio donde están los archivos asociados a la aplicación.

- b. Contenedor directorio: El contenedor recibe la ruta de almacenamiento desde “consulta registro” de esta manera se encarga de elaborar las instrucciones a ejecutarse en la consola del sistema operativo para enviarlos al procedimiento grabador archivos de aplicaciones dependiendo del medio extraíble de destino.

- c. Registro salida aplicación: Permite configurar en pantalla la información relevante a la o las aplicaciones almacenadas en el sistema de manera que el usuario pueda de forma legible recibir toda esta información con formato para la interfaz gráfica, recupera archivos concernientes a capturas de pantallas y tamaños de aplicación.

Constructor aplicaciones.

- a. Registro entrada aplicación: Permite el ingreso de información que describe o detalla una aplicación que va a ser almacenada en el sistema o de una aplicación ya almacenada en el sistema (modificación). Crea un registro de forma que interpreta la base de datos como una instrucción de almacenamiento y también recupera la información para el caso de modificación. Finalmente informa a “extractor archivos” la ruta de la cual tiene que copiar los archivos desde el medio extraíble de entrada hacia el disco duro del sistema.

- b. **Generador Evento:** El generador de evento crea un evento informativo sobre la carga de una aplicación al sistema, para que otro proceso lo almacene en la base de datos para su posterior auditoria.

- c. **Extractor archivos:** Una vez que se le informa de que lugar obtener los archivos de la aplicación el extractor archivos crea los métodos para indicarle a la consola del sistema operativo que debe copiar archivos de un determinado lugar hacia otro, en este caso, de un medio extraíble de entrada hacia una lugar específico, recuperado de la base de datos, en el disco duro del servidor donde se aloja el sistema gestor de aplicaciones.

Grabador archivos de aplicaciones.

- a. **Extractor archivos:** Este subproceso recibe las instrucciones generadas por el subproceso “contenedor directorio” para a su vez enviarlos a la consola del sistema operativo directamente en el caso de copiar a un medio de almacenamiento extraíble, o envía la información tal cual la recibió al siguiente subproceso que es “consulta parámetros”.

- b. **Consulta parámetros:** Consulta parámetros como su nombre lo indica consulta sobre los parámetros que se necesitan en el caso de grabar los archivos de una aplicación a un medio óptico como un CD o DVD. Estos parámetros los obtiene de la base de datos o de su subproceso predecesor, para transformarlos en comandos que interpreta la consola del sistema operativo.

- c. Generador evento: El generador de evento crea un evento informativo sobre la descarga de una aplicación desde el sistema, para que otro proceso lo almacene en la base de datos para su posterior auditoria.

- d. Constructor de estructuras: El constructor de estructuras envía las instrucciones o comandos recibidas desde "consulta parámetros" hacia la consola del sistema operativo para que se encargue de copiar los archivos desde el servidor hacia ya sea una memoria de almacenamiento masivo USB o un medio óptico CD o un medio óptico DVD para que el usuario reciba los archivos de la aplicación descargada.

Generador de reporte de aplicaciones.

- a. Validador permiso usuario: Recibe el permiso del sistema de que un administrador o un editor de aplicaciones pueden ejecutar tareas de control y conjuntamente con la solicitud de usuario de generar un reporte acerca de las aplicaciones almacenadas en el disco duro del servidor. Hay que señalar que a pesar de que exista una solicitud si no existe el permiso del usuario administrador o un usuario con rol de editor de aplicaciones entonces no envía la instrucción de generación de reporte; en otros términos es un subproceso de control.

- b. Lector registro: "Lector registro" una vez que ha recibido la instrucción de reporte consulta a la base de datos sobre las aplicaciones

almacenadas en el sistema obtiene los registros desde la base de datos para ser presentados al usuario.

- c. Generador reporte: Una vez que se obtuvieron los registros de la base de datos respecto a las aplicaciones este proceso se encarga de formatearlos de manera que sean legibles al usuario al momento de mostrarlos en la interfaz gráfica del sistema.

GESTION DE USUARIOS

Validador Usuarios

- a. Consultor: El papel del consultor es recibir los logins de los usuarios (posible nombre de usuario y posible contraseña) y consultar a la base de datos sobre la existencia de los mismos, en este caso solo el nombre de usuario. Si no existe el nombre usuario informa del error caso contrario obtiene la contraseña adjunta al nombre de usuario previamente almacenada en la base de datos.
- b. Comparador: Cuando se tiene ya la contraseña desde la base este proceso la compara con la posible contraseña que fue proporcionada por el usuario que intenta registrarse en el sistema. Si el proceso de comparación de las contraseñas es exitoso el proceso se comunica una vez más con la base de datos y obtiene el rol que toma el usuario dentro

del sistema, caso contrario informa que la contraseña ingresada no es válida y trunca el paso o registro del usuario dentro del sistema gestor de aplicaciones.

- c. Validador: Cuando el usuario ingresa al sistema obtiene su rol de usuario pero para que los procesos del sistema tengan conocimiento de que tipo de usuario es el que se encuentra en el sistema es necesario que algo los informe sobre este particular. Por ende el proceso “validador” registra en el sistema sobre que usuario y que tipo de usuario esta registrado. Sin no los usuarios se han desconectado del sistema automáticamente este proceso fija al usuario del sistema como “usuario final”.
- d. Generador evento: El “generador de evento” crea un evento informativo sobre el ingreso de un usuario con rol administrativo o un rol de edición de aplicaciones al sistema, para que otro proceso lo almacene en la base de datos para su posterior auditoria.

Constructor usuarios

- a. Constructor registro: Al momento que el administrador desea crear o modificar un usuario este proceso capta la información para almacenar el nuevo registro correspondiente al usuario en cuestión. Crea una instrucción que interpreta la base de datos para almacenar o actualizar el registro del usuario nuevo o modificado respectivamente.

- b. Generador evento: El “generador de evento” crea un evento informativo sobre la creación o modificación de los datos de un usuario del sistema, para que otro proceso lo almacene en la base de datos para su posterior auditoria.

Generador de reporte de usuarios

- a. Lector de registro: “Lector registro” una vez que ha recibido la instrucción de reporte consulta a la base de datos sobre los usuarios pertenecientes al sistema, obtiene los registros desde la base de datos para ser presentados al usuario administrador solamente.
- b. Generador reporte: Una vez que se obtuvieron los registros de la base respecto a los usuarios este proceso se encarga de formatearlos de manera que sean legibles al usuario al momento de mostrarlos en la interfaz gráfica del sistema.

CONFIGURADOR SISTEMA

Configurador de parámetros

El configurador de parámetros primero verifica que los datos acerca de parámetros operacionales del sistema sean suministrados por un administrador para luego, crear un registro e instrucción que interpreta la

base datos para almacenar cada uno de los nuevos parámetros que proveyó el usuario administrador.

Generador reporte parámetros

- a. Lector registro: “Lector registro” una vez que ha recibido la instrucción de reporte consulta a la base de datos sobre parámetros del sistema obtiene los registros desde la base de datos para ser presentados al usuario administrador solamente.
- b. Generador reporte: Una vez que se obtuvieron los registros de la base respecto a los parámetros del sistema este proceso se encarga de formatearlos de manera que sean legibles al usuario al momento de mostrarlos en la interfaz gráfica del sistema.

EVENTOS SISTEMA

Constructor de eventos

- a. Discriminador evento: Este subproceso inicialmente recibe los eventos generados por otros procesos del sistema y los clasifica, posteriormente cuando ya conoce la naturaleza del evento crea una instrucción con la fecha del sistema y el usuario que se encuentra registrado en el sistema al momento que se disparó el evento.

- b. Registro evento: Este subproceso ejecuta la instrucción que recibió del subproceso anterior comunicándose con la base de datos para que guarde el registro del evento para su posterior auditoría.

Generador reporte de eventos

- a. Lector registro: "Lector registro" una vez que ha recibido la instrucción de reporte consulta a la base de datos sobre los eventos suscitados dentro del sistema obtiene los registros desde la base de datos para ser presentados al usuario administrador solamente.
- b. Generador eventos: Finalmente, cuando ya se obtuvieron los registros de la base respecto a los eventos suscitados en el sistema este proceso se encarga de formatearlos de manera que sean legibles al usuario al momento de mostrarlos en la interfaz gráfica del sistema y presentarlos como eventos generados directamente por el usuario o generados por el propio sistema.

3.2.5.2. Gráfico de Arquitectura

GESTION DE APLICACIONES

Explorador aplicaciones.

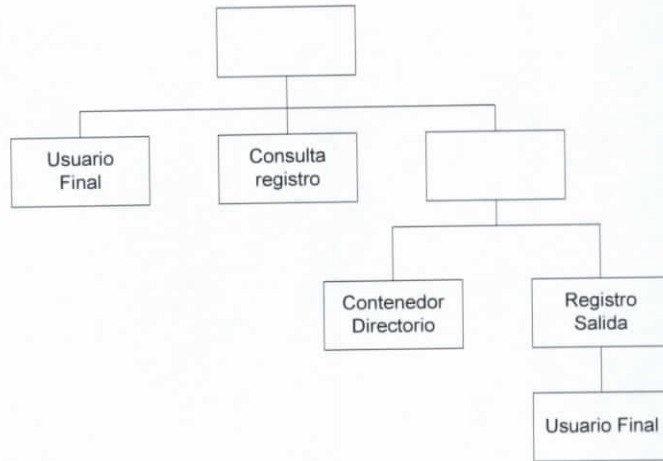


Ilustración 3. 9: Diagrama de arquitectura de explorador aplicaciones.

Constructor aplicaciones.

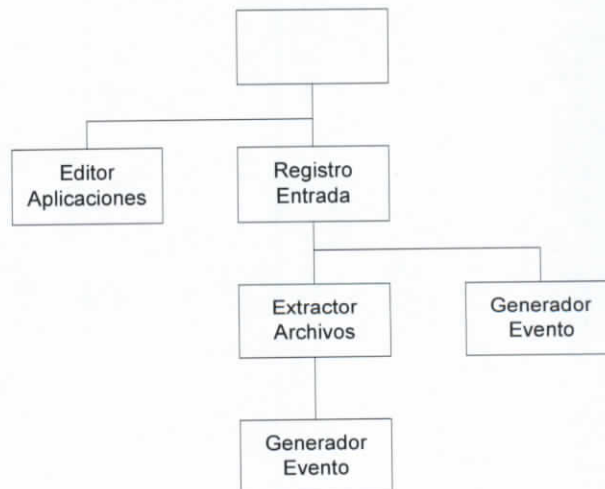


Ilustración 3. 10: Diagrama de arquitectura de constructor aplicaciones.

Grabador archivos de aplicaciones.

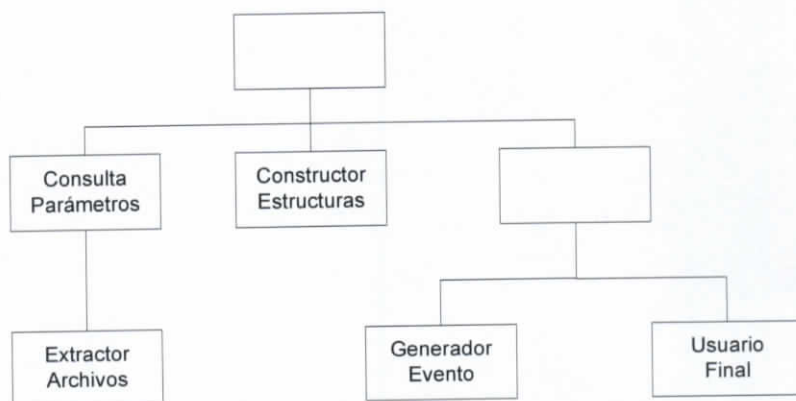


Ilustración 3. 11: Diagrama de arquitectura de grabador archivos de aplicaciones.

Generador de reporte de aplicaciones.

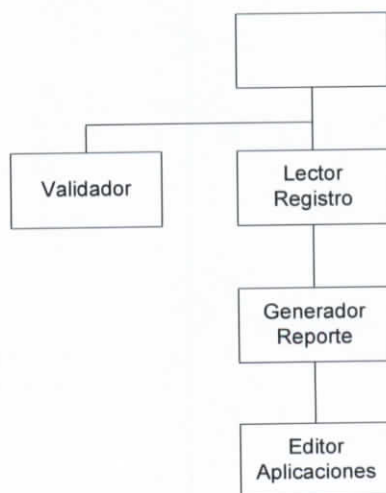


Ilustración 3. 12: Diagrama de arquitectura de generador de reporte de aplicaciones.

GESTION DE USUARIOS

Validador Usuarios

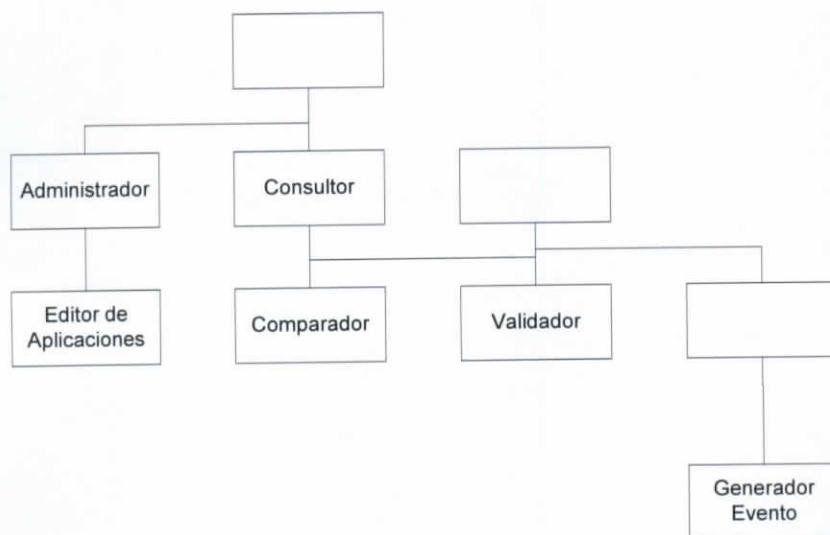


Ilustración 3. 13: Diagrama de arquitectura validador usuarios.

Constructor usuarios

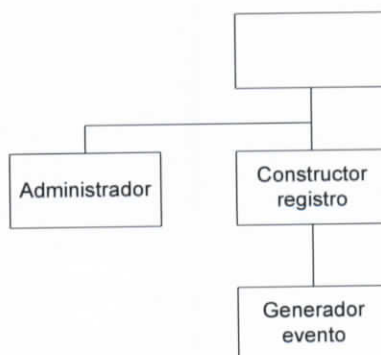


Ilustración 3. 14: Diagrama de arquitectura de constructor usuarios.

Generador de reporte de usuarios



Ilustración 3. 15: Diagrama de arquitectura generador de reporte usuarios.

CONFIGURADOR SISTEMA

Configurador de parámetros

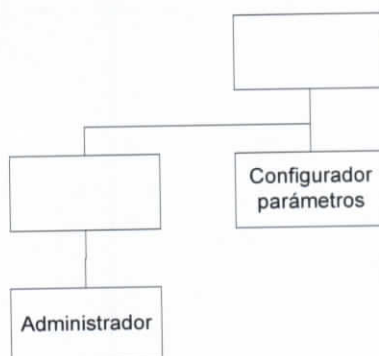


Ilustración 3. 16: Diagrama de arquitectura de configurador de parámetros.

Generador reporte parámetros



Ilustración 3. 17: Diagrama de arquitectura de generador reporte parámetros.

EVENTOS SISTEMA

Constructor de eventos

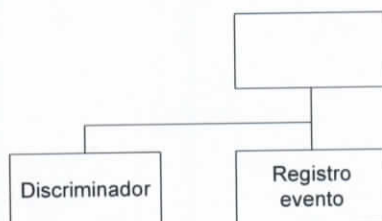


Ilustración 3. 18: Diagrama de arquitectura de constructor de eventos.

Generador reporte de eventos

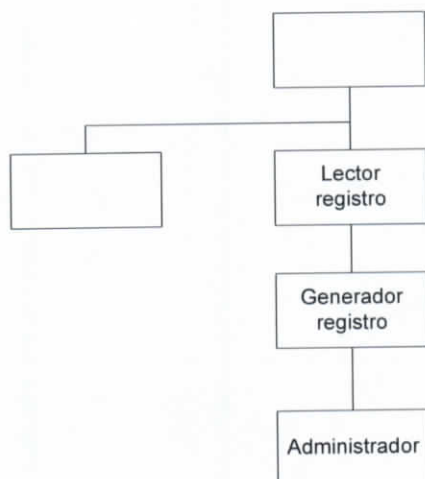


Ilustración 3. 19: Diagrama de arquitectura de generador reporte eventos.

3.2.6. Diseño Procedimental

GESTION DE APLICACIONES

Explorador aplicaciones.

a. Consulta registro

Inicio

leer nombre_aplicacion, lenguaje_aplicacion

si nombre_aplicacion <> vacio

 filtro = filtro + nombre_aplicacion

fin si

si lenguaje_aplicacion

 filtro = filtro + lenguaje_aplicacion

fin si

si codigo_aplicacion <> vacio

```
        llama contenedor_directorio(función consulta_ruta(código_aplicacion))
    fin si

    fin
```

b. Contenedor directorio

```
Inicio

    si opción_usuario = memoria_usb
        Comando = Copiar_a_memoria(parámetro ruta)
    caso contrario
        si tamaño_aplicacion(código_aplicacion) > 700mb
            Comando = Copiar_a_dvd(parámetro ruta)
        caso contrario
            Comando = Copiar_a_cd(parámetro ruta)
        fin si
    fin si

    llamar extractor_archivos(comando)

Fin
```

c. Registro salida aplicación

```
Inicio

    Imprime nombre_aplicacion
    Imprime detalles_aplicacion
    Imprime capturas_pantalla aplicación
    Imprime datos_adicionales_aplicacion
```

Fin

Constructor aplicaciones.

a. Registro entrada aplicación

Inicio

si opción_usuario = modificar_aplicacion

 Lee código_aplicacion

 imprime = Consulta_a_base(código_aplicacion)

 lee datos_aplicacion

 comando = actualizar_datos_base(datos_aplicacion)

caso contrario

 lee datos_aplicacion

 comando = grabar_datos_base(datos_aplicacion)

 ejecutar comando

 ruta=obtener_ruta_archivos_origen

 llamar extractor_archivos(ruta)

fin si

Fin

b. Generador Evento

Inicio

si modificación_aplicacion = si

 evento = modificación_aplicacion(código_aplicacion)

caso contrario

 evento = carga_aplicacion(código_aplicacion)

fin si

llamar constructor_eventos(evento)

Fin

c. Extractor archivos

Inicio

obtiene_ruta_destino

comando = copiar_archivos(parámetro_ruta_origen, ruta_destino)

ejecuta comando

Fin

Grabador archivos de aplicaciones.

a. Extractor archivos

Inicio

si comando_parametro = copiar_a memoria

lee ruta_memoria_destino

ejecuta comando_parametro(ruta_origen_parametro,

ruta_memoria_destino)

caso contrario

llama consulta_parametros(comando_parametro)

fin si

Fin

b. Consulta parámetros

Inicio

si comando_parametro = copiar_cd

obtiene_parámetros grabación_cd

```

        llamar constructor_estructur(comando_parametro +
parametros_grabacion_cd)
    caso contrario
        obtiene_parámetros grabación_dvd
        llamar constructor_estructur( comando_parametro +
parametros_grabacion_dvd)
    fin si

Fin

```

c. Generador evento

```

Inicio

evento= descarga_de_aplicacion
llamar constructor_eventos(evento)

Fin

```

d. Constructor de estructuras.

```

Inicio

imagen = Crear_imagen_iso(comando_parametro)
imprimir estado_grabacion
grabar_imagen (comando_parametro, imagen)
Imprimir estado_grabacion

Fin

```

Generador de reporte de aplicaciones.

a. Validador permiso usuario

```
Inicio
si permiso = editor_aplicaciones o permiso = administrador
    si permiso = editor_borrado
        permiso = administrador
    caso contrario
        permiso = editor_carga
    fin si
    llamar lector_registro

fin si

Fin
```

b. Lector registro

```
Inicio

Información_aplicaciones = consultar_base(todas_aplicaciones)
llamar generador_reporte(información_aplicaciones)

Fin
```

c. Generador reporte

```
Inicio

Imprimir información__aplicaciones_parametro

Fin
```

Validador Usuarios

a. Consultor

Inicio

Leer posible_usuario, posible_contraseña

Obtener_contrasena(posible_usuario)

Si obtener_contrasena_usuario = posible_usuario

 Llamar comparador(usuario,posible_contrasena)

Caso contrario

 Imprime no_existe_usuario

Fin si

Fin

b. Comparador

Inicio

si contraseña_parametro = obtener_contrasena(usuario_parametro)

 rol = consultar_rol_usuario(usuario_parametro)

 llamar validador(usuario_parametro, rol)

caso contrario

 imprime la_constrasena_no_es_valida

fin si

Fin

c. Validador

Inicio

```

si parámetro_rol = administrador
    paso = administrador
    imprime paso_concedido_a_usuario_parametro
caso contrario
    si parámetro_rol = editor_aplicaciones
        paso = editor_aplicaciones
        imprime paso_concedido_a_usuario_parametro
    fin si
fin si
llamar generador_evento(usuario_parametro)
Fin

```

d. Generador evento

```

Inicio

evento= ingreso_usuario_sistema + usuario_parametro

llamar constructor_eventos(evento)

Fin

```

Constructor usuarios

a. Constructor registro

```

Inicio

Si modifica_usuario = si
    Leer usuario_id
    Imprimir Obtener_datos(usuario_id)
    Guardar_cambios_usuario_base(obtener_cambios, usuario_id)

```

Llamar generador_evento(usuario_id + cambio)

Caso contrario

Leer usuario_id

Leer contraseña

Guardar_nuevo_usuario_base(usuario_id, contraseña)

Llamar generador_evento(usuario_id + nuevo)

Fin si

Fin

b. Generador evento

Inicio

evento= cambio_en_usuarios + usuario_id_parametro

llamar constructor_eventos(evento)

Fin

Generador de reporte de usuarios

a. Lector de registro

Inicio

Información_usuarios = consultar_base(todos_usuarios)

llamar generador_reporte(información_usuarios)

Fin

b. Generador reporte

Inicio

Imprimir información__usuarios_parametro

Fin

CONFIGURADOR SISTEMA

Configurador de parámetros

Inicio

Leer parametro1, parametro2... parametroN

Almacenar_parametros_base(parametro1,parametro2,.....,parametroN)

Fin

Generador reporte parámetros

a. Lector registro

Inicio

Información_parametros = consultar_base(todos_parametros)

llamar generador_reporte(información_parametros)

Fin

b. Generador reporte

Inicio

Imprimir información__parametros_parametro

Fin

EVENTOS SISTEMA

Constructor de eventos

a. Discriminador evento

Inicio

si evento_proviene_aplicacion

 tipo_evento = aplicaciones

caso contrario

 si evento_proviene_usuario

 tipo_evento = usuarios

 caso contrario

 si evento_proviene_sistema

 tipo_evento = sistema

 fin si

 fin si

fin si

llamar registro_evento(tipo_usuario)

Fin

b. Registro evento

Inicio

fecha_ = fecha_sistema()

```
usuario = obtener_usuario_conectado  
guardar_base_datos_evento(evento_parametro, usuario, tipo_evento_parametro)
```

Fin

Generador reporte de eventos

a. Lector registro

Inicio

Leer filtro_eventos

Si filtro_eventos = vacio

 Información_eventos = consultar_base(todos_eventos)

 llamar generador_reporte(información_parametros)

caso contrario

 Información_eventos = consultar_base(filtro_eventos)

 llamar generador_reporte(información_parametros)

fin si

Fin

b. Generador eventos

Inicio

Imprimir información__eventos_parametro

Fin

3.2.7. Diseño de Datos.

3.2.7.1. Diccionario de datos

Clasificación

Nombre:	Código Clasificación.
Campo:	CodClasificacion.
Fuente:	Configurador de Parámetros.
Alias:	Ninguno
Tipo:	Entero
Tamaño:	11
Formato:	Numero entero
Obligatorio:	Si
Identificador:	Si, Principal
Valor por defecto:	Autonumérico incremental en 1.
Responsable:	Administrador.
Descripción:	Campo utilizado como clave identificadora para cada registro de la clasificación de cada aplicación.

Tabla 3. 12: Diseño de datos, código clasificación.

Nombre:	Descripción Clasificación.
Campo:	DesClasificacion.
Fuente:	Configurador de parámetros.
Alias:	Ninguno
Tipo:	Cadena de caracteres
Tamaño:	30
Formato:	Cadena de caracteres sin formato.
Obligatorio:	No
Identificador:	No
Valor por defecto:	Nulo
Responsable:	Administrador.
Descripción:	Campo utilizado para la descripción de cada registro de la clasificación de cada aplicación. Es decir el nombre de la clasificación de la aplicación.

Tabla 3. 13: Diseño de datos, descripción clasificación.

Usuario

Nombre:	Identificación de Usuario.
Campo:	IDUsuario.
Fuente:	Constructor Usuarios.
Alias:	Ninguno
Tipo:	Cadena de caracteres
Tamaño:	15
Formato:	Cadena de caracteres sin formato
Obligatorio:	Si
Identificador:	Si, Principal
Valor por defecto:	Nulo
Responsable:	Administrador.
Descripción:	Campo utilizado como clave identificadora para cada registro de los usuarios que utilizan el sistema.

Tabla 3. 14: Diseño de datos, identificación de usuario.

Nombre:	Administrador.
Campo:	Administrador.
Fuente:	Constructor Usuarios.
Alias:	Ninguno
Tipo:	Entero
Tamaño:	1
Formato:	Numero entero
Obligatorio:	Si
Identificador:	No
Valor por defecto:	0
Responsable:	Administrador.
Descripción:	Campo utilizado para identificar al usuario Administrador dentro de sistema, es un campo de valor de verdad

Tabla 3. 15: Diseño de datos, administrador

Nombre:	Contraseña.
Campo:	Contrasena.
Fuente:	Constructor Usuarios.
Alias:	Ninguno
Tipo:	Cadena de caracteres.
Tamaño:	15
Formato:	Cadena de caracteres sin formato.
Obligatorio:	Si
Identificador:	No
Valor por defecto:	Nulo.
Responsable:	Administrador.
Descripción:	Campo utilizado para almacenar las contraseñas de cada uno de los usuarios que operan en el sistema.

Tabla 3. 16: Diseño de datos, contraseña.

Nombre:	Permiso de Borrado.
Campo:	PBorrado.
Fuente:	Constructor Usuarios
Alias:	Ninguno
Tipo:	Entero.
Tamaño:	1
Formato:	Numero entero.
Obligatorio:	Si
Identificador:	No
Valor por defecto:	0.
Responsable:	Administrador.
Descripción:	Campo utilizado para determinar si un usuario puede borrar una aplicación del sistema, es un campo de valor de verdad.

Tabla 3. 17: Diseño de datos, permiso de borrado.

Nombre:	Permiso de Carga.
Campo:	PCarga.
Fuente:	Constructor Usuarios
Alias:	Ninguno
Tipo:	Entero.
Tamaño:	1
Formato:	Numero entero.
Obligatorio:	Si
Identificador:	No
Valor por defecto:	0
Responsable:	Administrador.
Descripción:	Campo utilizado para conocer si un usuario puede cargar o no aplicaciones al sistema, Es un campo de valor de verdad.

Tabla 3. 18: Diseño de datos, permiso de carga.

Materia

Nombre:	Código Materia.
Campo:	CodMateria.
Fuente:	Configurador de parámetros.
Alias:	Ninguno
Tipo:	Entero.
Tamaño:	11
Formato:	Numero entero.
Obligatorio:	Si
Identificador:	Si, Principal
Valor por defecto:	Autonumérico incremental en 1.
Responsable:	Administrador.
Descripción:	Campo utilizado como clave identificadora para cada registro de la materia asociada a cada aplicación almacenada en el sistema.

Tabla 3. 19: Diseño de datos, código materia.

Nombre:	Descripción de Materia.
Campo:	DesMateria.
Fuente:	Configurador de parámetros
Alias:	Ninguno
Tipo:	Cadena de caracteres.
Tamaño:	30.
Formato:	Cadena de caracteres sin formato.
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Administrador.
Descripción:	Campo utilizado para describir el nombre de cada una de las materias asociadas a cada aplicación almacenada en el sistema.

Tabla 3. 20: Diseño de datos, descripción de materia.

Programas

Nombre:	Código Aplicación.
Campo:	CodAplicacion.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Entero.
Tamaño:	11.
Formato:	Numero entero.
Obligatorio:	Si.
Identificador:	Si, Principal.
Valor por defecto:	Autonumérico incremental en 1.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo utilizado como clave identificadora para cada registro de aplicación almacenada en el sistema

Tabla 3. 21: Diseño de datos, código aplicación.

Nombre:	Código Adicional Aplicación.
Campo:	CodAdicionalApp.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Cadena de caracteres.
Tamaño:	10.
Formato:	Cadena de caracteres sin formato.
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo que puede ser utilizado como clave identificadora para cada registro de aplicación almacenada en el sistema descrito por el usuario como identificación adicional al CodAplicacion

Tabla 3. 22: Diseño de datos, código adicional aplicación

Nombre:	Nombre Aplicación.
Campo:	NombreAplicacion.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Cadena de caracteres.
Tamaño:	100.
Formato:	Cadena de caracteres sin formato.
Obligatorio:	Si.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo utilizado para describir el nombre de la aplicación almacenada en el sistema.

Tabla 3. 23: Diseño de datos, nombre aplicación.

Nombre:	Autor.
Campo:	Autor.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Cadena de caracteres.
Tamaño:	70.
Formato:	Cadena de caracteres sin formato.
Obligatorio:	Si.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo utilizado para describir el nombre del autor y responsable para cada registro de aplicación almacenada en el sistema.

Tabla 3. 24: Diseño de datos, autor.

Nombre:	Código Clasificación.
Campo:	CodClasificacion.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Entero.
Tamaño:	11.
Formato:	Numero entero.
Obligatorio:	Si.
Identificador:	Si, Foráneo.
Valor por defecto:	Nulo.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo utilizado como clave identificadora foránea de clasificación de aplicación para cada registro de aplicación almacenada en el sistema.

Tabla 3. 25: Diseño de datos, código clasificación foráneo

Nombre:	Código Materia.
Campo:	CodMateria.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Entero.
Tamaño:	11.
Formato:	Numero entero.
Obligatorio:	Si.
Identificador:	Si, Foráneo.
Valor por defecto:	Nulo.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo utilizado como clave identificadora foránea de clasificación de aplicación para cada registro de aplicación almacenada en el sistema.

Tabla 3. 26: Diseño de datos, código materia foráneo

Nombre:	Lenguaje.
Campo:	Lenguaje.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Cadena de caracteres.
Tamaño:	50.
Formato:	Cadena de caracteres sin formato.
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo utilizado para describir el lenguaje utilizado para programar cada una de las aplicaciones almacenadas en el sistema.

Tabla 3. 27: Diseño de datos, lenguaje.

Nombre:	Descripción.
Campo:	Descripcion.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Texto.
Tamaño:	Sin longitud
Formato:	Sin formato.
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo de texto utilizado para una descripción breve del funcionamiento y tratamiento de cada aplicación almacenada en el sistema.

Tabla 3. 28: Diseño de datos, descripción.

Nombre:	Ruta gráfico número 1.
Campo:	RutaGrafico1.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Texto.
Tamaño:	Sin tamaño.
Formato:	Sin formato.
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo utilizado para registrar la ruta de acceso de una captura de pantalla para cada registro de aplicación almacenada en el sistema.

Tabla 3. 29: Diseño de datos, ruta gráfico número 1.

Nombre:	Ruta gráfico número 2.
Campo:	RutaGrafico2.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Texto.
Tamaño:	Sin tamaño.
Formato:	Sin formato.
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo utilizado para registrar la ruta de acceso de una segunda captura de pantalla para cada registro de aplicación almacenada en el sistema.

Tabla 3. 30: Diseño de datos, ruta gráfico número 2.

Nombre:	Ruta manual de usuario.
Campo:	RutaManual.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Texto.
Tamaño:	Sin tamaño.
Formato:	Sin formato.
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo utilizado para registrar la ruta de acceso al manual de usuario asociado para cada registro de aplicación almacenada en el sistema.

Tabla 3. 31: Diseño de datos, ruta manual de usuario.

Nombre:	Software Libre.
Campo:	SoftwareLibre.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Entero.
Tamaño:	1.
Formato:	Numero entero.
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	0.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo utilizado para identificar cada registro de aplicación almacenada en el sistema como una aplicación sin licencia comercial privada.

Tabla 3. 32: Diseño de datos, software libre.

Nombre:	Ayuda.
Campo:	Ayuda.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Entero.
Tamaño:	1.
Formato:	Número entero .
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	0.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo utilizado para registrar existe ayuda al usuario en la aplicación almacenada en el sistema.

Tabla 3. 33: Diseño de datos, ayuda.

Nombre:	Código Fuente.
Campo:	CodigoFuente.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Entero.
Tamaño:	1.
Formato:	Número entero.
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	0.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo utilizado para conocer si existe o no código fuente adjunto para cada registro de aplicación almacenada en el sistema.

Tabla 3. 34: Diseño de datos, código fuente.

Nombre:	Cantidad de Líneas de Código.
Campo:	CantLineasCodigo.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Entero.
Tamaño:	11.
Formato:	Número entero.
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	0.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo utilizado para registrar la cantidad de líneas de código que posee el código fuente para cada registro de aplicación almacenada en el sistema.

Tabla 3. 35: Diseño de datos, cantidad de líneas de código.

Nombre:	Ruta Programa.
Campo:	RutaPrograma.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Texto.
Tamaño:	Sin tamaño.
Formato:	Sin formato.
Obligatorio:	Si.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo utilizado para registrar la ruta de acceso de una captura de pantalla para cada registro de aplicación almacenada en el sistema.

Tabla 3. 36: Diseño de datos, ruta programa.

Nombre:	Identificador Autor (cédula).
Campo:	CIAutor.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Cadena de Caracteres.
Tamaño:	10.
Formato:	Cadena de caracteres sin formato.
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo utilizado para registrar la cédula de identidad del autor con efecto de reportes para cada registro de aplicación almacenada en el sistema.

Tabla 3. 37: Diseño de datos, identificador de autor por cédula.

Nombre:	Colaboradores.
Campo:	Colaborador.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Cadena de caracteres
Tamaño:	250.
Formato:	Cadena de caracteres sin formato.
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo utilizado para registrar los nombres de los colaboradores en el desarrollo de la aplicación almacenada en el sistema.

Tabla 3. 38: Diseño de datos, colaboradores.

Nombre:	Tamaño
Campo:	Tamano.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Cadena de Caracteres.
Tamaño:	13.
Formato:	Cadena de caracteres sin formato.
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Constructor Aplicaciones
Descripción:	Campo utilizado para registrar el tamaño en disco duro de cada registro de aplicación almacenada en el sistema.

Tabla 3. 39: Diseño de datos, tamaño.

Nombre:	Fecha de Subida.
Campo:	FechaSubida.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Fecha y Hora.
Tamaño:	Sin tamaño.
Formato:	Dia-Mes-Año Hora:Minuto.
Obligatorio:	Si.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Constructor Aplicaciones.
Descripción:	Campo utilizado para registrar la fecha y hora de carga al sistema de cada aplicación almacenada en el sistema.

Tabla 3. 40: Diseño de datos, fecha de subida.

Nombre:	Usuario.
Campo:	Usuario.
Fuente:	Constructor Aplicaciones.
Alias:	Ninguno.
Tipo:	Cadena de caracteres.
Tamaño:	15.
Formato:	Cadena de caracteres sin formato.
Obligatorio:	Si.
Identificador:	Si, Foráneo.
Valor por defecto:	Nulo.
Responsable:	Editor de Aplicaciones, Administrador.
Descripción:	Campo utilizado como clave identificadora foránea de usuario para conocer el usuario asociado al registro de aplicación almacenada en el sistema.

Tabla 3. 41: Diseño de datos, identificación de usuario foráneo.

Parámetro

Nombre:	Código Parámetro.
Campo:	CodParametro.
Fuente:	Configurador de parámetros.
Alias:	Ninguno.
Tipo:	Entero.
Tamaño:	11.
Formato:	Número Entero.
Obligatorio:	Si.
Identificador:	Si, primario.
Valor por defecto:	Valor Autonumérico incremento en 1.
Responsable:	Administrador.
Descripción:	Campo utilizado para identificar el registro de parámetros operacionales del sistema.

Tabla 3. 42: Diseño de datos, código parámetro.

Nombre:	Unidad CD.
Campo:	UnidadCD.
Fuente:	Configurador de parámetros.
Alias:	Ninguno.
Tipo:	Cadena de caracteres.
Tamaño:	15.
Formato:	Cadena de caracteres sin formato.
Obligatorio:	Si.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Administrador.
Descripción:	Campo utilizado para registrar el string del dispositivo de grabación para CD-R que lo identifica en el sistema operativo.

Tabla 3. 43: Diseño de datos, unidad CD.

Nombre:	Unidad DVD.
Campo:	UnidadDVD.
Fuente:	Configurador de parámetros.
Alias:	Ninguno.
Tipo:	Cadena de caracteres.
Tamaño:	15.
Formato:	Cadena de caracteres sin formato.
Obligatorio:	Si.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Administrador.
Descripción:	Campo utilizado para registrar el string del dispositivo de grabación para DVD-R que lo identifica en el sistema operativo.

Tabla 3. 44: Diseño de datos, unidad DVD.

Nombre:	Directorio de programas número 1.
Campo:	DirectorioUsuario1.
Fuente:	Configurador de parámetros.
Alias:	Ninguno.
Tipo:	Texto.
Tamaño:	Sin tamaño.
Formato:	Sin formato.
Obligatorio:	Si.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Administrador.
Descripción:	Campo utilizado para registrar la ruta principal de almacenamiento para guardar los archivos de las aplicaciones.

Tabla 3. 45: Diseño de datos, directorio de programas número 1.

Nombre:	Directorio de programas número 2
Campo:	DirectorioUsuario2.
Fuente:	Configurador de parámetros.
Alias:	Ninguno.
Tipo:	Texto.
Tamaño:	Sin tamaño.
Formato:	Sin formato.
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Administrador.
Descripción:	Campo utilizado para registrar la ruta secundaria de almacenamiento para guardar los archivos de las aplicaciones.

Tabla 3. 46: Diseño de datos, directorio de programas número 2

Nombre:	Velocidad escritura de unidad CD.
Campo:	VelocidadCD.
Fuente:	Configurador de parámetros.
Alias:	Ninguno.
Tipo:	Entero.
Tamaño:	11.
Formato:	Número entero.
Obligatorio:	Sí.
Identificador:	No.
Valor por defecto:	0.
Responsable:	Administrador.
Descripción:	Campo utilizado para registrar la velocidad de grabación del dispositivo grabador de CD-R.

Tabla 3. 47: Diseño de datos, velocidad escritura unidad CD.

Nombre:	Velocidad escritura de unidad DVD.
Campo:	VelocidadDVD.
Fuente:	Configurador de parámetros.
Alias:	Ninguno.
Tipo:	Entero.
Tamaño:	11.
Formato:	Número entero.
Obligatorio:	Sí.
Identificador:	No.
Valor por defecto:	0.
Responsable:	Administrador.
Descripción:	Campo utilizado para registrar la velocidad de grabación del dispositivo grabador de DVD-R.

Tabla 3. 48: Diseño de datos, velocidad escritura unidad DVD.

Nombre:	Eventos Usuario.
Campo:	Eventos.
Fuente:	Configurador de parámetros.
Alias:	Ninguno.
Tipo:	Entero.
Tamaño:	1.
Formato:	Número entero.
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	0.
Responsable:	Administrador.
Descripción:	Campo utilizado para registrar si se debe o no registrar los eventos suscitados en los módulos del sistema debido a solicitudes del usuario.

Tabla 3. 49: Diseño de datos, eventos de usuario.

Nombre:	Eventos Sistema.
Campo:	IngresoSistema.
Fuente:	Configurador de parámetros.
Alias:	Ninguno.
Tipo:	Entero.
Tamaño:	1.
Formato:	Número entero.
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	0.
Responsable:	Administrador.
Descripción:	Campo utilizado para registrar si se debe o no registrar los eventos de sistema suscitados cuando se ingresa al sistema.

Tabla 3. 50: Diseño de datos, eventos del sistema.

Eventos

Nombre:	Código de Evento.
Campo:	CodRegistro.
Fuente:	Constructor de eventos.
Alias:	Ninguno.
Tipo:	Entero.
Tamaño:	11.
Formato:	Número entero.
Obligatorio:	Si.
Identificador:	Si, principal.
Valor por defecto:	Valor Autonumérico, incremento en 1.
Responsable:	Constructor de eventos.
Descripción:	Campo utilizado para identificar los eventos suscitados dentro del sistema.

Tabla 3. 51: Diseño de datos, código de evento.

Nombre:	Tipo de Evento.
Campo:	Eventos.
Fuente:	Constructor de eventos.
Alias:	Ninguno.
Tipo:	Cadena de caracteres.
Tamaño:	15.
Formato:	Cadena de caracteres sin formato.
Obligatorio:	Si.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Constructor de eventos.
Descripción:	Campo utilizado para registrar el tipo de los eventos suscitados dentro del sistema.

Tabla 3. 52: Diseño de datos, tipo de evento.

Nombre:	Fecha Evento.
Campo:	FechaHora.
Fuente:	Constructor de eventos.
Alias:	Ninguno.
Tipo:	Fecha y Hora.
Tamaño:	Sin tamaño.
Formato:	Mes-Dia-Año Hora:Minutos.
Obligatorio:	Si.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Constructor de eventos.
Descripción:	Campo utilizado para registrar la fecha y la hora cuando aconteció el evento dentro del sistema.

Tabla 3. 53: Diseño de datos, fecha del evento.

Nombre:	Usuario.
Campo:	Usuario.
Fuente:	Constructor de eventos.
Alias:	Ninguno.
Tipo:	Cadena de caracteres.
Tamaño:	30.
Formato:	Cadena de caracteres sin formato.
Obligatorio:	Si.
Identificador:	Si. Foráneo
Valor por defecto:	Nulo.
Responsable:	Constructor de eventos.
Descripción:	Campo utilizado para registrar el usuario conectado al sistema al momento que los eventos fueron registrados dentro del sistema.

Tabla 3. 54: Diseño de datos, identificación usuario foráneo.

Nombre:	Descripción Eventos.
Campo:	Descripcion.
Fuente:	Constructor de eventos.
Alias:	Ninguno.
Tipo:	Cadena de caracteres.
Tamaño:	100.
Formato:	Cadena de caracteres sin formato.
Obligatorio:	No.
Identificador:	No.
Valor por defecto:	Nulo.
Responsable:	Constructor de eventos.
Descripción:	Campo utilizado para describir a detalle los eventos suscitados dentro del sistema.

Tabla 3. 55: Diseño de datos, descripción de eventos.

3.2.7.2. Diseño de Base de Datos

Diseño Lógico

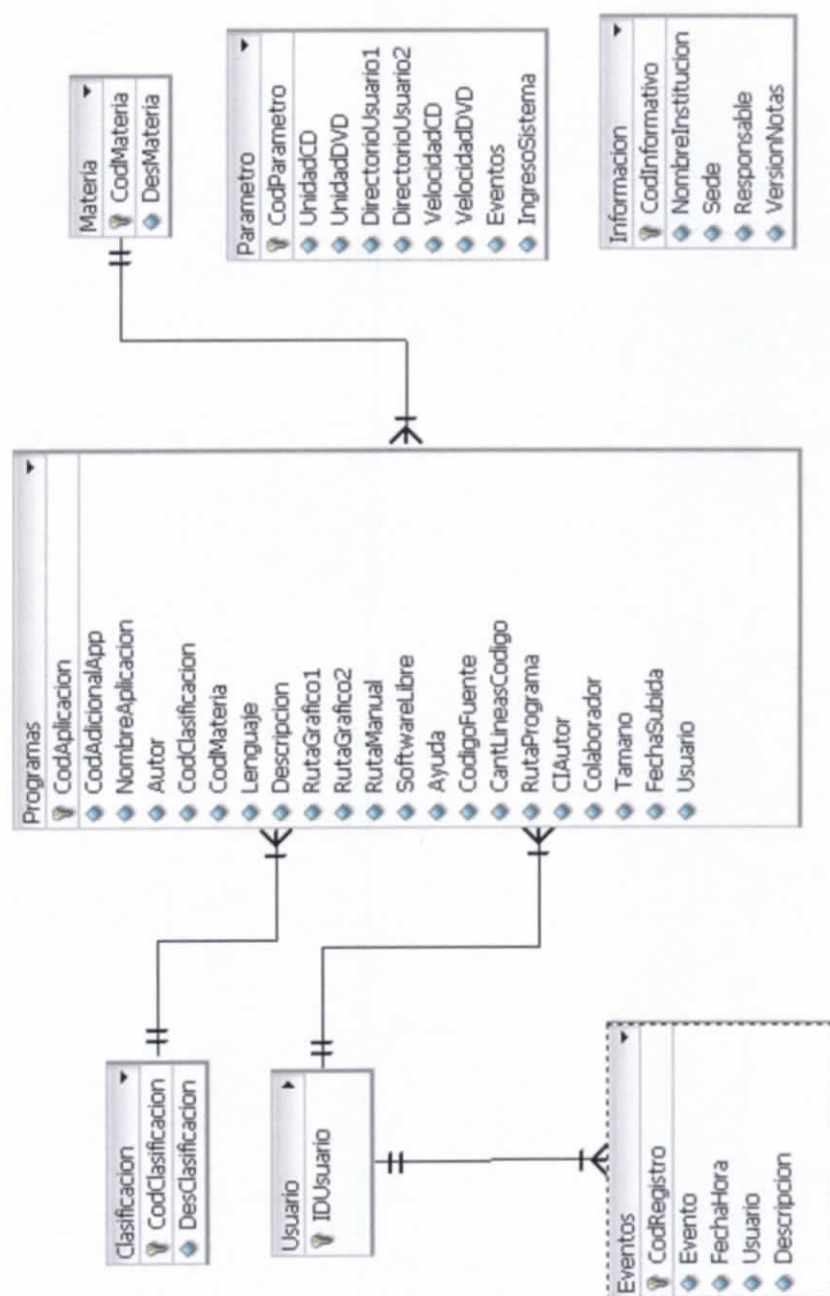


Ilustración 3. 20: Diseño Lógico de Base de Datos

Diseño Físico

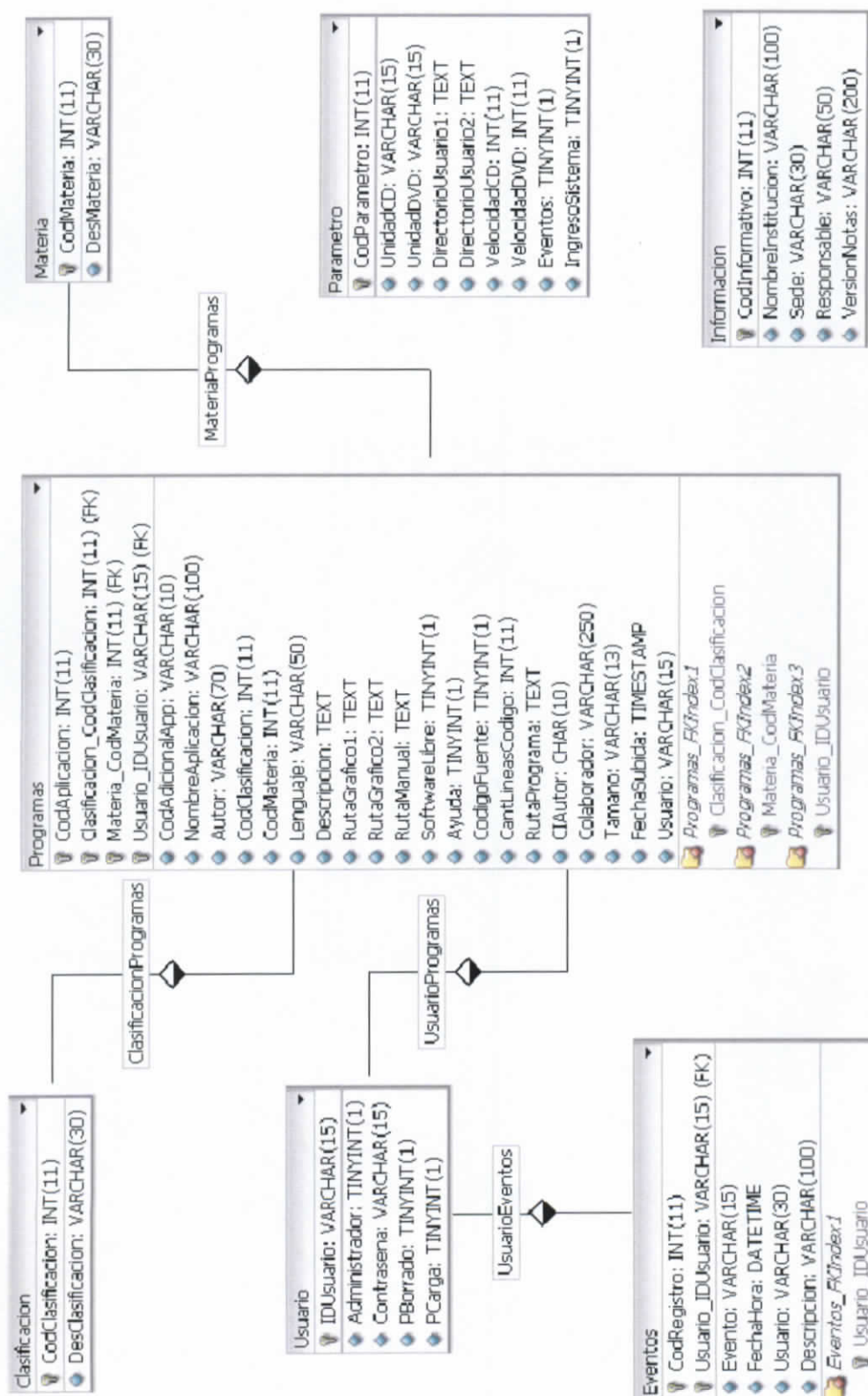


Ilustración 3. 21: Diseño físico de la base de datos

3.2.8. Diseño de la interfaz en gambas.

Pantalla principal del sistema gestor de aplicaciones. Dividida en cuatro partes, en el sector izquierdo para cargar aplicaciones al sistema, el derecho para descargar aplicaciones, en el centro expulsar medios extraíbles conectados y en la parte superior el menú de uso exclusivo de los superusuarios.



Ilustración 3. 22: Portada de la interfaz.

Control de acceso para usuarios del sistema

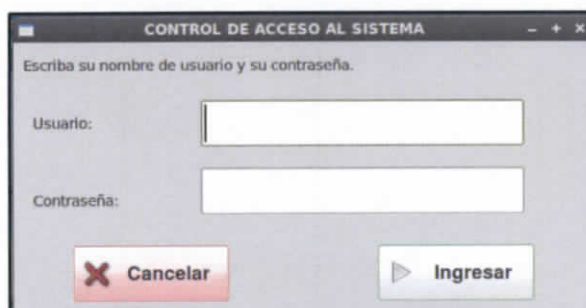
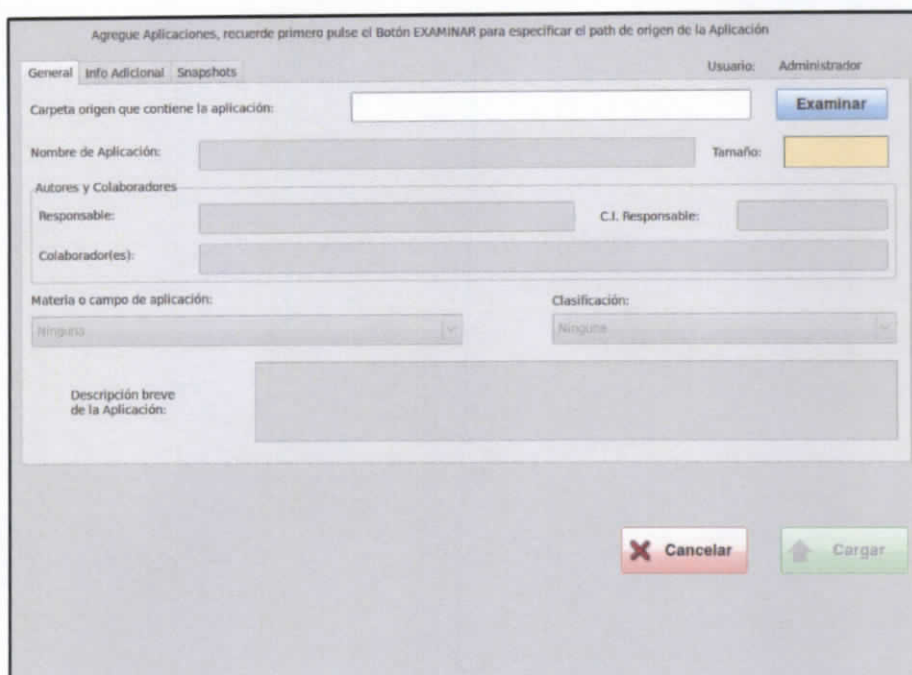


Ilustración 3. 23: Cuadro de diálogo de acceso al sistema.

Pantalla para detallar la información de la aplicación a cargarse al sistema, nótese que existen pestañas para más opciones como capturas de pantallas o snapshots.



A screenshot of a web-based form titled "Agregue Aplicaciones, recuerde primero pulse el Botón EXAMINAR para especificar el path de origen de la Aplicación". The form has three tabs: "General", "Info Adicional", and "Snapshots", with "General" selected. At the top right, it shows "Usuario: Administrador". The form contains several input fields: "Carpeta origen que contiene la aplicación:" with a text box and an "Examinar" button; "Nombre de Aplicación:" and "Tamaño:" with text boxes; "Autores y Colaboradores" section with "Responsable:" and "C.I. Responsable:" fields; "Colaborador(es):" with a text box; "Materia o campo de aplicación:" and "Clasificación:" with dropdown menus (both set to "Ninguna"); and "Descripción breve de la Aplicación:" with a large text area. At the bottom right, there are "Cancelar" and "Cargar" buttons.

Ilustración 3. 24: Pantalla para agregar información de aplicaciones pestaña No. 1

Cuadro de diálogo para escoger los directorios de origen que contiene los archivos de las aplicaciones a cargarse.

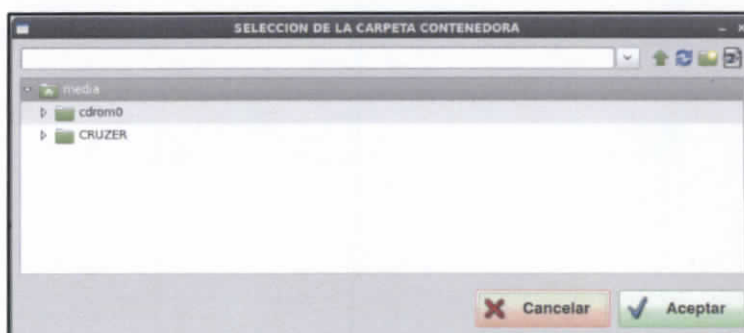


Ilustración 3. 25: Cuadro de diálogo con controles de directorio.

Diálogos informativos para conocer el proceso de copia de incluye controles de control de proceso de copia y una barra de progreso para conocer el estado de la copia de archivos.

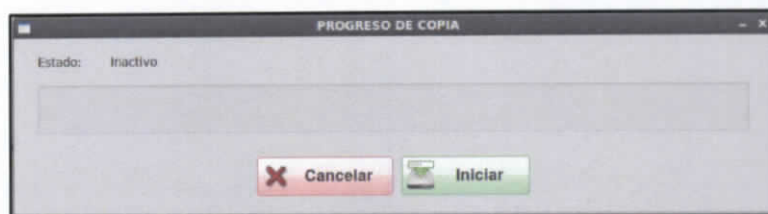


Ilustración 3. 26: Cuadro informativo de avance de proceso.

Pantalla informativa sobre las aplicaciones alojadas dentro del sistema gestor, presenta información básica y capturas de pantalla. Incluye filtro para búsquedas y además permite desplegar un panel con información opcional de la aplicación.

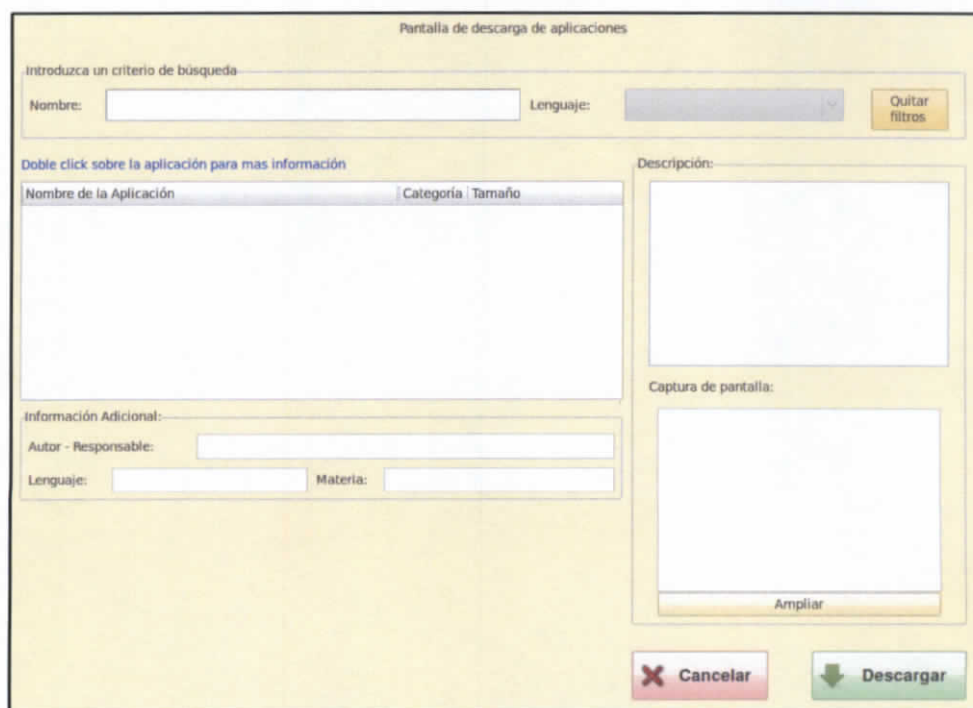


Ilustración 3. 27: Pantalla de información de aplicaciones almacenadas en el sistema lista para su descarga.

Diálogos de decisión que se encuentran en el sistema, en este caso para escoger el tipo de medio extraíble en el cual serán copiados los archivos de la aplicación que se pretende descargar desde el sistema. Todos los diálogos llevan el mismo estilo.

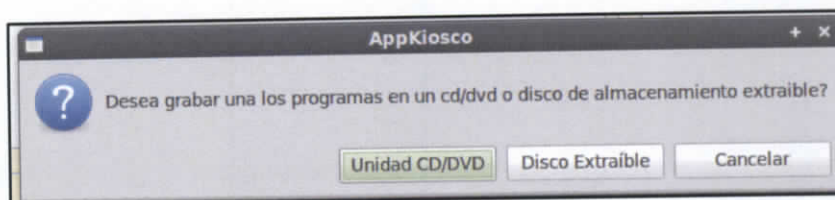


Ilustración 3. 28: Cuadro de diálogo con botones para selección.

Pantalla para escoger el medio extraíble de almacenamiento masivo y el directorio de destino en el cual se copiarán los archivos de la aplicación a descargarse, incluye botones para empezar o cancelar el proceso de copia y una barra de progreso para conocer el avance de la copia.

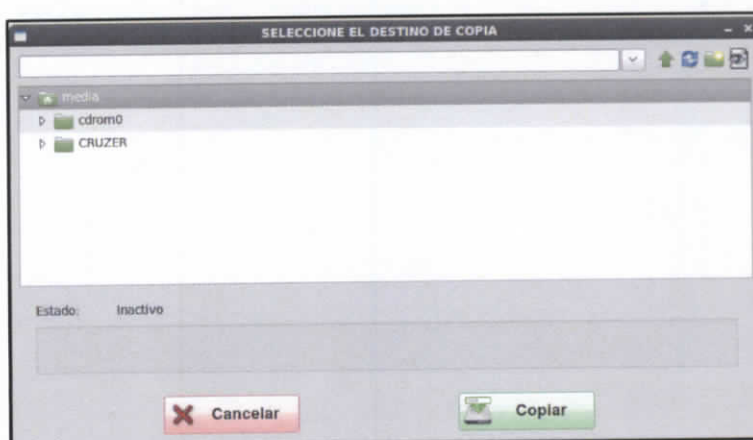


Ilustración 3. 29: Cuadro de dialogo con control de directorios y barra de progreso de proceso de copia.

Cuadro de dialogo con controles para introducir los parámetros al sistema, incluye menús animados para los diferentes módulos de configuración

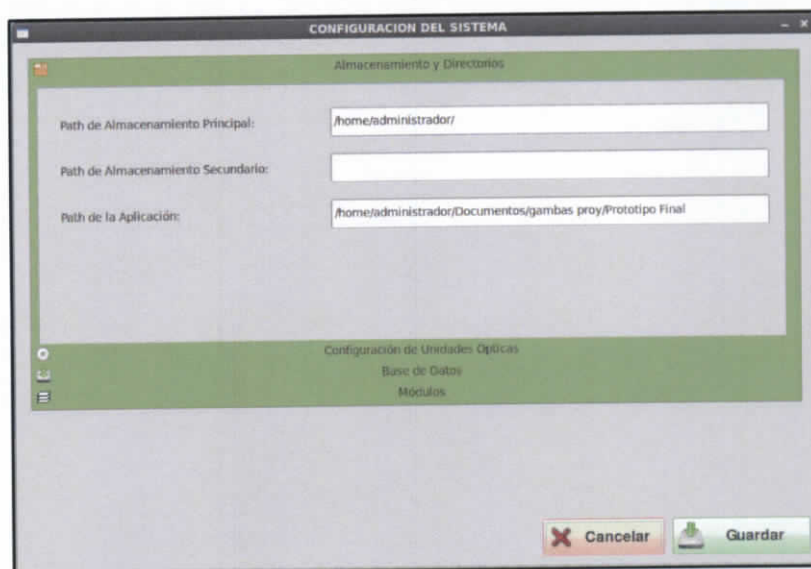


Ilustración 3. 30: Pantalla de configuración de sistema con paneles animados.

La imagen representa un cuadro de diálogo para la el ingreso de información de un nuevo usuario, aplica a modificaciones de información referente a un usuario.

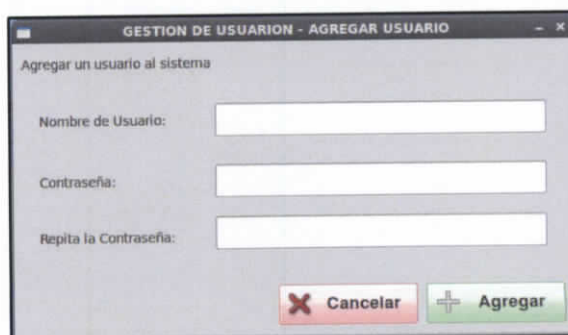


Ilustración 3. 31: Cuadro de dialogo para ingreso de información de usuario

Aquí se muestran los usuarios y sus respectivos permisos que determinan el rol dentro del sistema. Los cambios realizados se reflejan automáticamente en la base de datos.

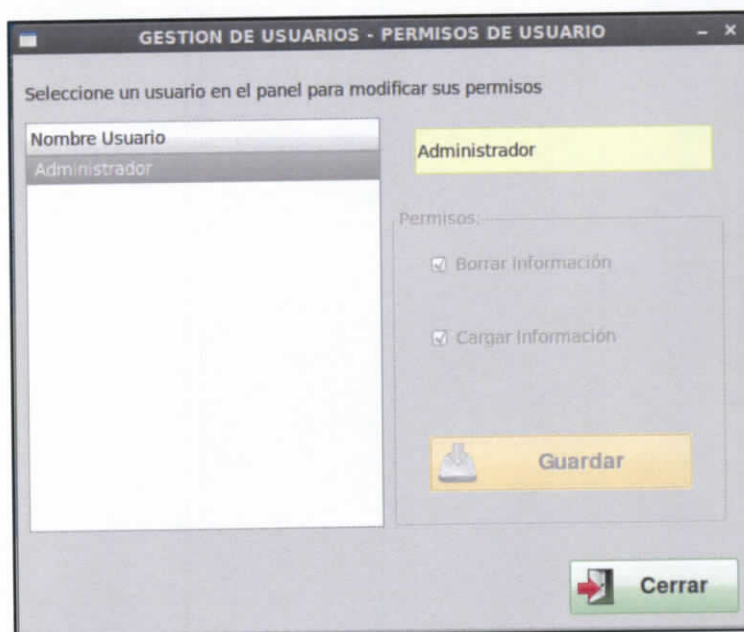


Ilustración 3. 32: Pantalla de control de permisos de usuario.

Cuadro de diálogo simple, en este caso para el ingreso de información al sistema por parte del administrador para clasificar las aplicaciones al sistema en lo que respecta a la categoría asociada y materia asociada.

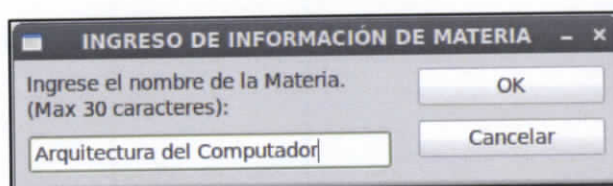


Ilustración 3. 33: Cuadro de dialogo de ingreso de información de materia.

Pantalla de modificación de información de sistema, con controles de Nuevo, Actualización de Información y Eliminación, aplica también a los usuarios y aplicaciones el estilo mostrado.

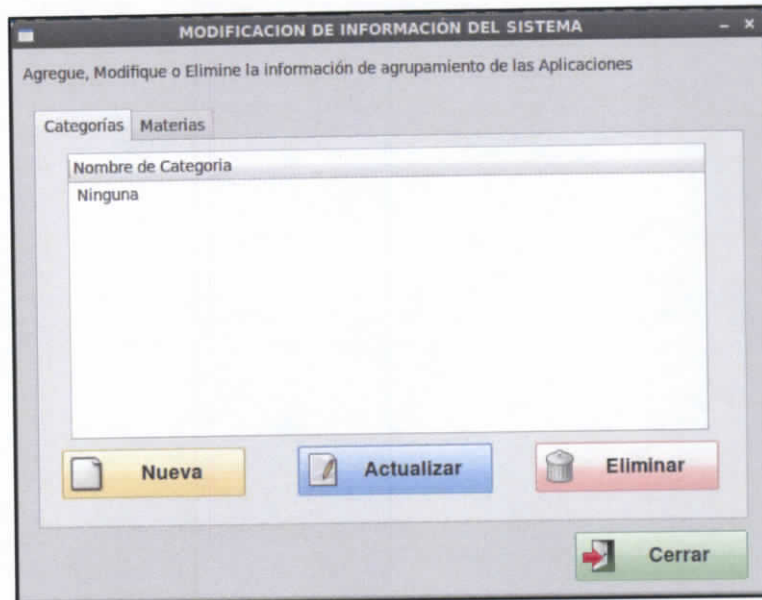


Ilustración 3. 34: Pantalla con pestañas y controles de edición de información de sistema.

Cuadro de diálogo informativo típico dentro del sistema.

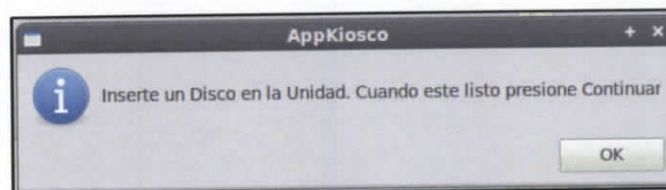


Ilustración 3. 35: Cuadro de mensaje informativo del sistema

Pantalla para conocer los dispositivos de almacenamiento masivo USB conectados al sistema. Además incluye controles para la expulsión del dispositivo y cierre de cancelación de proceso.

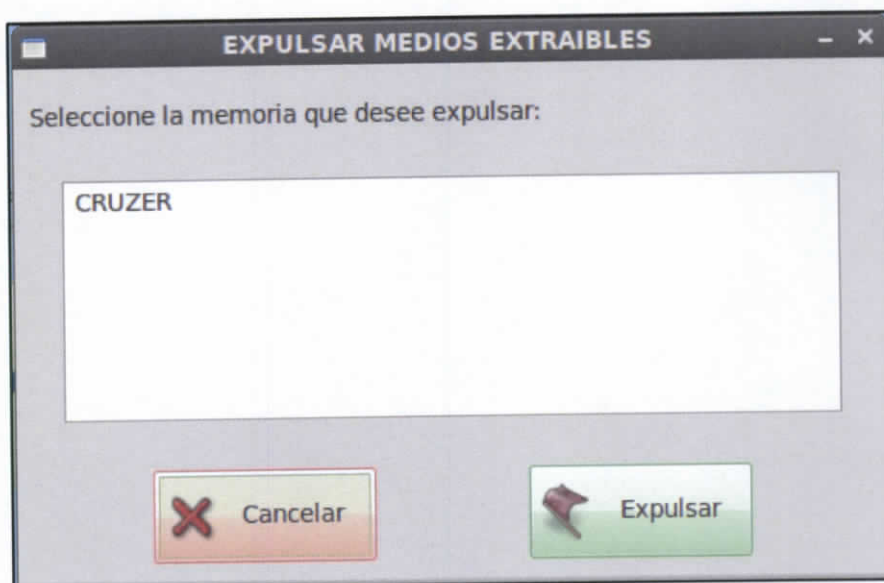


Ilustración 3. 36: Pantalla de expulsión de medios extraíbles.

Pantalla de modificación de aplicaciones, es similar a la de carga de aplicaciones, pero muestra la información de la aplicación a modificarse. Con motivos de visualización en esta pantalla se muestra otra pestaña de la pantalla donde se indica la ruta de los archivos de fotografía y su respectiva pre-visualización que son en sí las capturas de pantalla de la aplicación a cargarse o modificarse.

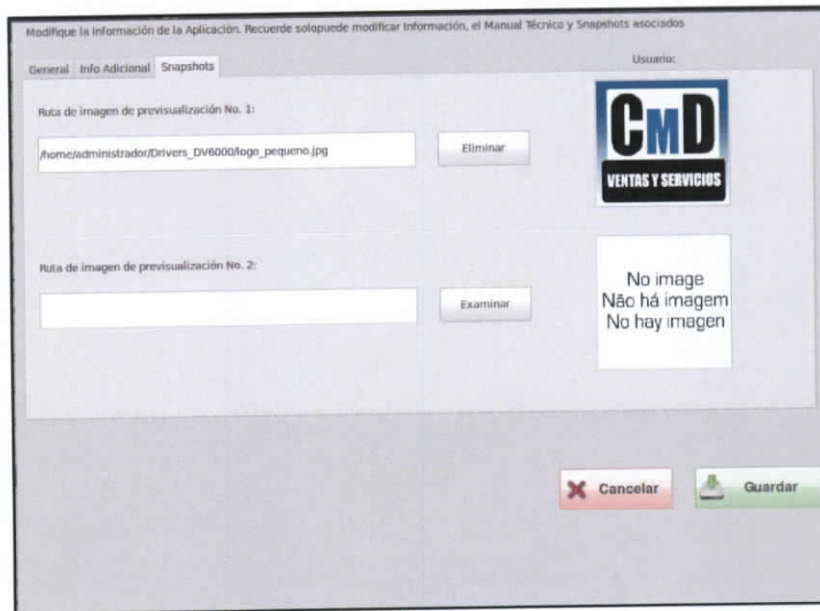


Ilustración 3. 37: Pantalla de edición de información de aplicaciones seleccionada la pestaña No.3 para snapshots.

La última pantalla muestra un reporte en pantalla, en este caso de los eventos del sistema. En la parte superior existen controles de ordenamiento y filtro con criterios de información.

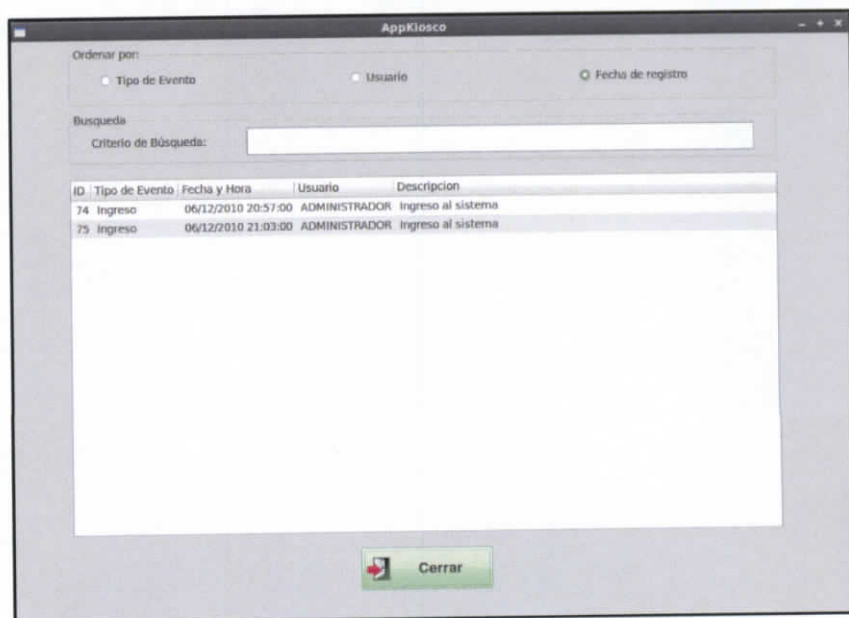


Ilustración 3. 38: Pantalla de reporte de eventos suscitados dentro del sistema.

3.3. Desarrollo del proyecto.

3.3.1. Desarrollo de la base de datos.

Extracto de código SQL para MySQL versión 5

```
--
-- Crea esquema BaseKiosco
--

CREATE DATABASE IF NOT EXISTS BaseKiosco;
USE BaseKiosco;

--
-- Definición de tabla `BaseKiosco`.`Clasificacion`
--

DROP TABLE IF EXISTS `BaseKiosco`.`Clasificacion`;
CREATE TABLE `BaseKiosco`.`Clasificacion` (
  `CodClasificacion` int(11) NOT NULL AUTO_INCREMENT,
  `DesClasificacion` varchar(30) COLLATE utf8_spanish_ci DEFAULT NULL,
  PRIMARY KEY (`CodClasificacion`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;

--
-- Definición de tabla `BaseKiosco`.`Eventos`
--

DROP TABLE IF EXISTS `BaseKiosco`.`Eventos`;
CREATE TABLE `BaseKiosco`.`Eventos` (
  `CodigoRegistro` int(11) NOT NULL AUTO_INCREMENT,
  `Evento` varchar(15) CHARACTER SET latin1 DEFAULT NULL,
  `FechaHora` datetime NOT NULL,
  `Usuario` varchar(30) CHARACTER SET latin1 DEFAULT 'UsuarioFinal',
  `Descripcion` varchar(100) COLLATE utf8_spanish_ci DEFAULT NULL,
  PRIMARY KEY (`CodigoRegistro`)
) ENGINE=MyISAM AUTO_INCREMENT=76 DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;

--
-- Definición de tabla `BaseKiosco`.`Informacion`
--

DROP TABLE IF EXISTS `BaseKiosco`.`Informacion`;
CREATE TABLE `BaseKiosco`.`Informacion` (
  `CodInformativo` int(11) NOT NULL AUTO_INCREMENT,
  `NombreInstitucion` varchar(100) COLLATE utf8_spanish_ci NOT NULL,
  `Sede` varchar(30) COLLATE utf8_spanish_ci NOT NULL,
  `Responsable` varchar(50) COLLATE utf8_spanish_ci DEFAULT NULL,
  `VersionNotas` varchar(200) CHARACTER SET latin1 DEFAULT NULL,
  PRIMARY KEY (`CodInformativo`)
) ENGINE=MyISAM AUTO_INCREMENT=2 DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;

--
```

```
-- Definición de tabla `BaseKiosco`.`Materia`
--
```

```
DROP TABLE IF EXISTS `BaseKiosco`.`Materia`;
CREATE TABLE `BaseKiosco`.`Materia` (
  `CodMateria` int(11) NOT NULL AUTO_INCREMENT,
  `DesMateria` varchar(30) COLLATE utf8_spanish_ci DEFAULT NULL,
  PRIMARY KEY (`CodMateria`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
```

```
-- Definición de tabla `BaseKiosco`.`Parametro`
--
```

```
DROP TABLE IF EXISTS `BaseKiosco`.`Parametro`;
CREATE TABLE `BaseKiosco`.`Parametro` (
  `CodParametro` int(11) NOT NULL AUTO_INCREMENT,
  `UnidadCD` varchar(15) COLLATE utf8_spanish_ci NOT NULL,
  `UnidadDVD` varchar(15) CHARACTER SET latin1 DEFAULT NULL,
  `DirectorioUsuario1` text COLLATE utf8_spanish_ci,
  `DirectorioUsuario2` text COLLATE utf8_spanish_ci,
  `VelocidadCD` int(11) NOT NULL DEFAULT '24',
  `VelocidadDVD` int(11) NOT NULL DEFAULT '4',
  `Eventos` tinyint(1) DEFAULT NULL,
  `IngresoSistema` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`CodParametro`)
) ENGINE=MyISAM AUTO_INCREMENT=2 DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
```

```
-- Definición de tabla `BaseKiosco`.`Programas`
--
```

```
DROP TABLE IF EXISTS `BaseKiosco`.`Programas`;
CREATE TABLE `BaseKiosco`.`Programas` (
  `CodAplicacion` int(11) NOT NULL AUTO_INCREMENT,
  `CodAdicionalApp` varchar(10) CHARACTER SET latin1 DEFAULT NULL,
  `NombreAplicacion` varchar(100) COLLATE utf8_spanish_ci NOT NULL,
  `Autor` varchar(70) COLLATE utf8_spanish_ci NOT NULL,
  `CodClasificacion` int(11) NOT NULL,
  `CodMateria` int(11) NOT NULL,
  `Lenguaje` varchar(50) CHARACTER SET latin1 DEFAULT NULL,
  `Descripcion` text CHARACTER SET latin1,
  `RutaGrafico1` text CHARACTER SET latin1,
  `RutaGrafico2` text CHARACTER SET latin1,
  `RutaManual` text CHARACTER SET latin1,
  `SoftwareLibre` tinyint(1) NOT NULL DEFAULT '0',
  `Ayuda` tinyint(1) NOT NULL DEFAULT '0',
  `CodigoFuente` tinyint(1) NOT NULL DEFAULT '0',
  `CantLineasCodigo` int(11) DEFAULT '0',
  `RutaPrograma` text CHARACTER SET latin1 NOT NULL,
  `CIAutor` char(10) COLLATE utf8_spanish_ci DEFAULT NULL,
  `Colaborador` varchar(250) COLLATE utf8_spanish_ci DEFAULT NULL,
  `Tamano` varchar(13) COLLATE utf8_spanish_ci DEFAULT '0',
  `FechaSubida` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  PRIMARY KEY (`CodAplicacion`),
  KEY `FK_Clasificacion_Programas` (`CodClasificacion`),
  KEY `FK_Materia_Programas` (`CodMateria`),
  CONSTRAINT `new_fk_constraint_clasificacion_programas` FOREIGN KEY (`CodClasificacion`)
REFERENCES `Clasificacion` (`CodClasificacion`) ON UPDATE CASCADE,
  CONSTRAINT `new_fk_constraint_materias_programas` FOREIGN KEY (`CodMateria`)
REFERENCES `Materia` (`CodMateria`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci
ROW_FORMAT=DYNAMIC;
```

3.3.2. Codificación de procedimiento y funciones.

Extracto del código del lenguaje de programación de Gambas para Linux versión 2 bajo Linux Ubuntu 9.10.

Código de métodos y eventos del formulario principal del sistema:

```
' Gambas class file

PUBLIC SUB Form_Open()
  DIM ResultadoParametros, InformacionSistema AS Result
  DIM Comando AS NEW String[]
  'Formulario con tamaño personalizado
  ME.Width = 1024
  ME.Height = 760
  'Centra el formulario
  ME.Center
  ' centrar el panel de titulo
  ME.PanelTitulo.X = (ME.Width - ME.PanelTitulo.Width) / 2

  'centrar el panel de botones
  ME.HPanelBotones.X = (ME.Width - ME.HPanelBotones.Width) / 2
  ME.HPanelBotones.Y = ((ME.Height - ME.HPanelBotones.Height) / 2) +
30

  'Deshabilita funciones de administrador de usuarios y encera
variables globales
  AppModule.AdminOnLine = 0
  AppModule.UCarga = 0
  AppModule.VariableMenuIngreso = FALSE

  'Deshabilita menu de administracion
  ME.Usuarios.Enabled = FALSE
  ME.AutenticacionSistema.Enabled = TRUE
  ME.MenuSalir.Enabled = FALSE
  ME.MenuAdministracion.Enabled = FALSE

  'Funcion en el módulo para abrir la conexión a MySQL
  IF AppModule.ConectarMySQL("localhost", "root", "12345",
"BaseKiosco") = "conectado" THEN
    Message("Conectado a MySQL", "Aceptar")
  ELSE
    Message("Error al conectar a MySQL", "Aceptar")
    ME.Close
  END IF
  'obtencion de los directorios de almacenamiento y parametros del
sistema
  ResultadoParametros = db.Exec("SELECT * FROM Parametro WHERE
CodParametro = 1")
  IF ResultadoParametros.Available THEN
```

```

    AppModulo.ArrayParametros[0] =
ResultadoParametros["DirectorioUsuario1"]
    AppModulo.ArrayParametros[1] =
ResultadoParametros["DirectorioUsuario2"]
    AppModulo.ArrayParametros[2] = ResultadoParametros["UnidadCD"]
    AppModulo.ArrayParametros[3] =
ResultadoParametros["VelocidadCD"]
    AppModulo.ArrayParametros[4] = ResultadoParametros["UnidadDVD"]
    AppModulo.ArrayParametros[5] =
ResultadoParametros["VelocidadDVD"]

' obtencion del espacio libre disponible en servidor
AppModulo.RotuloEspacio()

IF ResultadoParametros["IngresoSistema"] = TRUE OR
ResultadoParametros["IngresoSistema"] = -1 THEN
    AppModulo.ArrayParametros[6] = -1
ELSE
    AppModulo.ArrayParametros[6] = 0
ENDIF
IF ResultadoParametros["Eventos"] = TRUE OR
ResultadoParametros["Eventos"] = -1 THEN
    AppModulo.ArrayParametros[7] = -1
ELSE
    AppModulo.ArrayParametros[7] = 0
ENDIF
ELSE
    Message.Error("No se pudo obtener los parametros del sistema.
Contacte al administrador")
ENDIF
    InformacionSistema = db.Exec("SELECT * FROM Informacion WHERE
CodInformativo = 1")
    IF InformacionSistema.Available THEN
        ME.TxtLblUniversidad.Text =
InformacionSistema["NombreInstitucion"] & " " &
InformacionSistema["Sede"]
        ME.TxtLblInfo.Text = InformacionSistema["Responsable"] & " / " &
InformacionSistema["VersionNotas"]

    ELSE
        Message.Error("No se pudo obtener los parametros del sistema.
Contacte al administrador")
    ENDIF
    CATCH 'se ejecuta solo si hay un error"
        Message.Error("Error #: " & Error.Code & " " & Error.Text & "
Contacte al administrador")
        ME.Close
    END

PUBLIC SUB BtnCargar_Click()

' abre el formulario de validacion de usuarios a menos que el
administrador o un usuario de carga este en linea
IF AppModulo.AdminOnLine = 1 OR AppModulo.UCarga = 1 THEN
' abre directamente el formulario de carga
ME.Enabled = FALSE
WITH FormCarga
    .Show
    .BtnExaminarApp.SetFocus
END WITH

```

```

ELSE
    ControlUsuario.Show
    'desactiva el form principal
    ME.Enabled = FALSE
ENDIF

END

PUBLIC SUB BtnDescargar_Click()

    'abre el formulario de descarga de informacion
    FormDescarga.Show

END

PUBLIC SUB MenuIngreso_Click()

    'abre el formulario para ingreso de usuarios
    ControlUsuario.Show
    'desactiva el form principal
    'ME.Enabled = FALSE
    'activa variable de menu global
    AppModule.VariableMenuIngreso = TRUE

END

PUBLIC SUB MenuSalir_Click()
    'Desconecta al usuario del sistema
    IF Message.Question("Desea salir del sistema?", "SI", "NO") = 1
    THEN
        AppModule.AdminOnLine = 0
        AppModule.UCarga = 0
        AppModule.UsuarioConectado = ""
        ME.Usuarios.Enabled = FALSE
        ME.MenuSalir.Enabled = FALSE
        ME.MenuAdministracion.Enabled = FALSE
        ME.MenuIngreso.Enabled = TRUE
        'Inserta registro de evento
        TRY AppModule.RegistraEventos("Salida",
AppModule.UsuarioConectado, "Usuario desconectado del sistema")
        ENDIF
    ENDIF

END

PUBLIC SUB CrearUsuario_Click()
    'invoca al formulario de creacion de usuarios
    FormAgregarUsuario.Show
    'bloquea el formulario principal
    ME.Enabled = FALSE
END

PUBLIC SUB ModificarUsuario_Click()
    'invoca al formulario de modificacion y eliminacion de usuarios
    FormModificarEliminarUsuario.Show
    'bloquea el formulario principal
    ME.Enabled = FALSE
END

```

```

PUBLIC SUB Permisos_Click()
    'invoca al formulario de permiso de usuarios
    FormPermisoUsuario.Show
    'bloquea el formulario principal
    ME.Enabled = FALSE

END

PUBLIC SUB InfSistema_Click()
    'invoca al formulario de informacion del sistema
    FormInformacionSistema.Show
    'bloquea el formulario principal
    ME.Enabled = FALSE
END

PUBLIC SUB MenuVariables_Click()
    'invoca al formulario de configuracion de parametros
    FormParametros.Show
    'bloquea el formulario principal
    ME.Enabled = FALSE
END

PUBLIC SUB MenuRegistro_Click()

    'invoca al formulario de configuracion de parametros
    FormReporte.Show
    'bloquea el formulario principal
    ME.Enabled = FALSE

END

PUBLIC SUB GestionAplicaciones_Click()

    'invoca al formulario de borrado de aplicaciones
    FormGestionAplicaciones.Show
    'bloquea el formulario principal
    ME.Enabled = FALSE

END

PUBLIC SUB BtnExpulsarMemoria_Click()

    'Llama al formulario de expulsar las memorias
    FormExpulsaMemoria.Show
END

```

Código del módulo de funciones y procedimientos del sistema:

```

' Gambas module file
' Gambas module file para la aplicación
' ===== VARIABLES PUBLICAS =====

```

```

PUBLIC con AS Connection ' variable para la conexion con MySQL
PUBLIC AdminOnLine AS Integer 'variable de identificacion de
administrador
PUBLIC UCarga AS Integer 'variable de identificacion de usuario de
carga
PUBLIC VariableMenuIngreso AS Boolean 'variable para saber si abre
el form de autentificacion de usuarios desde el menu
PUBLIC NProcesoTeclado AS Float 'variable para saber el
identificador del proceso del teclado
PUBLIC BanderaError AS Boolean ' Variable para saber si hay error en
los procesos
PUBLICCodigoAplicacion AS Integer 'Variable para conocer el codigo
de programa en caso de error
PUBLIC ArrayClasificacion AS Integer[] 'Array para almacenar la
clasificacion
PUBLIC ArrayMateria AS Integer[] 'Array para almacenar la materia
PUBLIC StrInfGrabacion AS String 'String para conocer el estado de
la grabacion por errores
PUBLIC MsgErrorGrabacion AS String 'Bandera para saber si se suscito
un error o se completo el proceso de grabación
PUBLIC hProcesoCDDVDISO AS Process 'Proceso para grabar CD o DVD o
crear ISO
PUBLIC BanderaCancelarGrabacion AS Integer 'bandera para conocer si
se cancelo el proceso de grabacion
PUBLIC StrRutaSnapShot1 AS String 'String para almanecar el archivo
del SnapShot 1
PUBLIC StrRutaSnapShot2 AS String 'String para almanecar el archivo
del SnapShot 2
PUBLIC StrRutaManual AS String 'String para almanecar el archivo del
Manual de usuario
PUBLIC ArregloCargaDirectorios AS String[2] 'Arreglo para almacenar
los directorios origen y destino en proceso de carga
PUBLIC ArrayParametros AS String[8] 'Arreglo con los parametros del
sistema
PUBLIC UsuarioConectado AS String 'variable para conocer el usuario
que se encuentra conectado al sistema
PUBLIC ArrayModificaCopia AS String[4] 'Arreglo para controlar la
copia en modificar aplicación
PUBLIC DirectorioAlternativo AS Boolean 'variable para saber si se
utiliza el directorio secundario
'===== FUNCIONES Y RUTINAS
=====

PUBLIC SUB CerrarTeclado()
'Si el teclado en pantalla no está activo entonces lo activa
  IF AppModulo.NProcesoTeclado <> 0 THEN
    EXEC ["pkill", "onboard"] WAIT
    AppModulo.NProcesoTeclado = 0
  END IF
  CATCH 'se ejecuta solo si hay un error"
    Message.Error("Error #: " & Error.Code & " " & Error.Text & "
Contacte al administrador")
END
PUBLIC FUNCTION ConectarMySQL(NombreHost AS String, Usuario AS
String, Clave AS String, NombreBase AS String) AS String

  'inicializa la variable global como una conexion nueva
  con = NEW Connection
  'tipo de conexion de BDD
  con.Type = "mysql"

```

```

'indica a la conexion el nombre del servidor
con.Host = NombreHost
'indica a la conexion el usuario tomar nota de los usuarios y
permisos
con.User = Usuario
'password de la conexion deberia encriptarse
con.password = Clave
'especifica el nombre de la base que se accede
con.Name = NombreBase
'abre la conexion con control de errores
TRY con.Open
'regresa el valor de la funcion
RETURN "conectado"
'control de errores
CATCH 'se ejecuta solo si hay un error"
Message.Error("Error #: " & Error.Code & " " & Error.text & "
Contacte al administrador")
RETURN "error"
END

PUBLIC FUNCTION RemoverBlancos(Cadena AS String) AS String
DIM LC, i AS Integer
DIM TextoSinBlancos AS String
LC = Len(Cadena)

FOR i = 1 TO LC
IF Mid(cadena, i, 1) <> " " THEN
TextoSinBlancos = TextoSinBlancos & Mid(cadena, i, 1)
ELSE
TextoSinBlancos = TextoSinBlancos & "\\ "
ENDIF
NEXT

RETURN TextoSinBlancos
'no utilizar end function como visual basic
END

PUBLIC FUNCTION BarraBajaArchivo(Cadena AS String) AS String
DIM LC, i AS Integer
DIM TextoSinBlancos AS String
LC = Len(Cadena)

FOR i = 1 TO LC
IF Mid(cadena, i, 1) = " " THEN
TextoSinBlancos = TextoSinBlancos & "_"
ELSE
TextoSinBlancos = TextoSinBlancos & Mid(cadena, i, 1)
ENDIF
NEXT

RETURN TextoSinBlancos
'no utilizar end function como visual basic
END

PUBLIC SUB TecladoEnPantalla()

DIM ComandoTeclado AS String
DIM ProcesoTeclado AS Process
'rutina para llamar al teclado en pantalla
ComandoTeclado = "onboard"
ProcesoTeclado = SHELL ComandoTeclado

```

```

NProcesoTeclado = ProcesoTeclado.Id
END
PUBLIC FUNCTION PorcentajeCopia(TamanoOriginal AS Float,
DirectorioDestino AS String) AS Integer
  DIM buffer AS String
  DIM BuferArreglo AS NEW String[]
  DIM Resultado AS Integer
  DIM Tamano AS Float

  'esta funcion permite conocer el porcentaje de copia actual
  comparando el tamano de la carpeta inicial con la destino
  SHELL ("du -s " & DirectorioDestino) WAIT TO buffer
  IF buffer THEN
    BuferArreglo = Split(Buffer, "\t")
    'convertimos la cadena separada en float
    Tamano = CFloat(BuferArreglo[0])
    resultado = CInteger((Tamano * 100) / TamanoOriginal)
    RETURN resultado
  ELSE
    resultado = 100
    RETURN resultado
  ENDIF
END
PUBLIC FUNCTION NombreArchivoApp(Ruta AS String) AS String
  DIM contador AS Integer
  contador = Len(Ruta)
  WHILE (Mid(Ruta, contador, 1) <> "/")
    contador = contador - 1
  WEND
  RETURN Mid(Ruta, contador + 1, Len(Ruta))
END
PUBLIC SUB RegistraEventos(Tipo AS String, Usuario AS String,
Descripcion AS String)
  DIM FechaHora AS String
  FechaHora = Format(Now(), "yyyy-mm-dd hh:nn")
  'registra el evento en la BDD siempre y cuando este habilitado el
  registro
  IF Tipo = "Ingreso" OR Tipo = "Salida" THEN
    '6 es para eventos de entrada y salida del sistema
    IF ArrayParametros[6] = -1 THEN
      db.Exec("INSERT INTO Eventos SET Evento = '" & Tipo & "',
FechaHora='" & FechaHora & "', Usuario = '" & Trim(Upper(Usuario)) &
"', Descripcion = '" & Trim(Descripcion) & "'")
    ENDIF
  ELSE
    '7 es para los eventos del sistema
    IF ArrayParametros[7] = -1 THEN
      db.Exec("INSERT INTO Eventos SET Evento = '" & Tipo & "',
FechaHora='" & FechaHora & "', Usuario = '" & Trim(Upper(Usuario)) &
"', Descripcion = '" & Trim(Descripcion) & "'")
    ENDIF
  ENDIF
  'Control de errores
CATCH
  Message.Error(Error.Text & " No se registro el evento")
END
PUBLIC FUNCTION FreeSpaceServidor(DirectorioApp AS String) AS Long
  'funcion para retornar el espacio libre dentro del servidor
  'Obtiene el tamaño de espacio libre en el servidor 2^10 factor
  conversion megabytes

```

```
' control de seguridad de espacio en 400mb por directorio
RETURN (Dfree(DirectorioApp) / (2 ^ 10)) - 406900
```

```
END
```

```
PUBLIC SUB RotuloEspacio()
```

```
    FMain.LblEspacio.Text = "Espacio libre en servidor directorio #1:
" & AppModulo.FreeSpaceServidor(AppModulo.ArrayParametros[0]) & "
Kb."
```

```
    FMain.LblEspacioSec.Text = "Espacio libre en servidor directorio
#2: " & AppModulo.FreeSpaceServidor(AppModulo.ArrayParametros[1]) &
" Kb."
```

```
END
```

Código con instrucciones al Shell del sistema operativo copia de archivos:

```
' Gambas class file
```

```
PUBLIC SUB ProcesoCopia()
```

```
    DIM ComandoCopia AS String
```

```
    DIM StringDirectorioOrigen AS String
```

```
    DIM StringDirectorioDestino AS String
```

```
    'Declaracion de un proceso para saber el estado
```

```
    DIM hProc AS Process
```

```
    DIM buffer AS String
```

```
    DIM BufferArreglo AS NEW String[]
```

```
    DIM TamanoOriginalCarpeta AS Float
```

```
    DIM OpcionesCopia AS String
```

```
    'Obtiene la ruta del Directorio destino donde se almacena la
aplicacion
```

```
    IF AppModulo.DirectorioAlternativo = FALSE
```

```
        StringDirectorioDestino = AppModulo.ArregloCargaDirectorios[0]
```

```
    ELSE
```

```
        StringDirectorioDestino = AppModulo.ArregloCargaDirectorios[1]
```

```
    ENDIF
```

```
    'Obtiene la ruta del Directorio origen donde se almacena la
aplicacion
```

```
    StringDirectorioOrigen = AppModulo.ArregloCargaDirectorios[1]
```

```
    'concatena el directorio con la barra de directorio de linux
```

```
    StringDirectorioOrigen = StringDirectorioOrigen & "/"
```

```
    'Funcion para remover espacios en blanco en lo directorios debido
a manejo de paths de linux
```

```
    StringDirectorioDestino =
```

```
AppModulo.RemoveBlancos(StringDirectorioDestino)
```

```
    StringDirectorioOrigen =
```

```
AppModulo.RemoveBlancos(StringDirectorioOrigen)
```

```
    'IF discrimina que formulario lo invocó
```

```
    IF ME.Tag = "n"
```

```
        'Coloca el estado en copiando
```

```
        ME.LblInfo.Text = "COPIANDO APLICACION...."
```

```
        'se obtiene el tamano original de la carpeta
```

```

SHELL ("du -s " & StringDirectorioOrigen) WAIT TO buffer
BuferArreglo = Split(buffer, "\t")
TamanoOriginalCarpeta = CFloat(BuferArreglo[0])

'copia todos los archivos del directorio de la aplicacion al
servidor con SHELL
'el asterisco permite seleccionar solo el contenido dentro del
directorio
ComandoCopia = "cp -r " & StringDirectorioOrigen & "*" &
StringDirectorioDestino
hProc = SHELL ComandoCopia

DO WHILE hProc.State = Process.Running
'cabia el cursor a wait
ME.Mouse = Mouse.Wait
'WAIT es una pausa en el sistema para nuevamente hacer el loop
del do while
WAIT 0.1
IF ME.btnCancel.tag = 1 THEN
'Al cancelar se cancela el proceso elimina el registro
activo se limpian campos y se recarga el numero de codigo
hProc.Kill
WITH FormCarga
.Enabled = TRUE
END WITH
BREAK
ELSE
'invoca a la funcion para saber el porcentaje de copia
ME.ProgressBar1.Value =
(AppModulo.PorcentajeCopia(TamanoOriginalCarpeta,
StringDirectorioDestino)) / 100
ENDIF
LOOP
ENDIF
IF btnCancel.Tag <> 1 THEN
'copia las imagenes y el manual
ComandoCopia = "cp "
IF ME.Tag = "n" THEN
WITH FormCarga
IF .LblRutaManual.Text THEN
OpcionesCopia = .LblRutaManual.Text
ENDIF
IF .LblRutaImg1.Text THEN
OpcionesCopia = OpcionesCopia & " " & .LblRutaImg1.Text
ENDIF
IF .LblRutaImg2.Text THEN
OpcionesCopia = OpcionesCopia & " " & .LblRutaImg2.Text
ENDIF
END WITH
ELSE
WITH FormModificaApp
IF .LblRutaManual.Text AND AppModulo.ArrayModificaCopia[0] =
1 THEN
OpcionesCopia =
AppModulo.RemoveBlancos(.LblRutaManual.Text)
ENDIF
IF .LblRutaImg1.Text AND AppModulo.ArrayModificaCopia[1] = 1
THEN
OpcionesCopia = OpcionesCopia & " " &
AppModulo.RemoveBlancos(.LblRutaImg1.Text)

```

```

ENDIF
IF FormCarga.LblRutaImg2.Text AND
AppModulo.ArrayModificaCopia[2] = 1 THEN
    OpcionesCopia = OpcionesCopia & " " &
AppModulo.RemoveBlancos(.LblRutaImg2.Text)
ENDIF
END WITH
ENDIF
IF OpcionesCopia THEN
    'Coloca el estado en copiando
    ME.LblInfo.Text = "COPIANDO MANUAL Y VISTAS PREVIAS...."
    ComandoCopia = ComandoCopia & Trim(OpcionesCopia) & " " &
StringDirectorioDestino
    hProc = SHELL ComandoCopia

    DO WHILE hProc.State = Process.Running
        'cambia el cursor a wait
        ME.Mouse = Mouse.Wait
        'WAIT es una pausa en el sistema para nuevamente hacer el
loop del do while
        WAIT 0.1
        IF ME.btnCancel.tag = 1 THEN
            'Al cancelar se cancela el proceso elimina el registro
activo se limpian campos y se recarga el numero de codigo
            hProc.Kill
            IF ME.Tag = "n" THEN
                WITH FormCarga
                    .Enabled = TRUE
                END WITH
            ELSE
                FormModificaApp.Enabled = TRUE
            ENDIF
            BREAK
        ENDIF
    LOOP
ENDIF
IF btnCancel.Tag <> 1 THEN
    IF ME.ProgressBar1.Value < 1 THEN ME.ProgressBar1.Value = 1
    'rehabilita el formulario y pone el cursor en arrow
    ME.Mouse = Mouse.Arrow
    'deja el mensaje inactivo para saber que acabo el proceso
implementar control de errores
    LblInfo.Text = "INACTIVO..."
    Message.Info("Proceso Completo", "Aceptar")
    IF ME.Tag = "n" THEN
        'habilita el formulario de carga, obtiene el nuevo codigo y
limpia los campos
        WITH FormCarga
            .Enabled = TRUE
            .EjecutaSQLInsertar
            .LimpiarCampos
        END WITH
    ELSE
        FormModificaApp.EjecutaSQLUpdate
        FormModificaApp.Close
    ENDIF

    'Encera el arreglo de copia y de modificacion del modulo
    AppModulo.ArrayModificaCopia[0] = 0
    AppModulo.ArrayModificaCopia[1] = 0

```

```

    AppModulo.ArrayModificaCopia[2] = 0
    'Control - Elimina directorios usados
    AppModulo.ArregloCargaDirectorios[0] = ""
    AppModulo.ArregloCargaDirectorios[1] = ""
ELSE
    Message.Warning("Se ha cancelado el proceso de copia.")
    'Codigo para remover inconsistencias
ENDIF
ELSE
    Message.Warning("Se ha cancelado el proceso de copia")
    'Codigo para remover inconsistencias
ENDIF
'recalcula espacio libre para el fmain
AppModulo.RotuloEspacio
ME.Close
CATCH 'se ejecuta solo si hay un error"
    Message.Error("Error #: " & Error.Code & " " & Error.Text & "
Contacte al administrador")
END

```

```

PUBLIC SUB Process_Read()
    DIM scad AS String
    LINE INPUT #LAST, scad
    Message(scad)

```

```

END

```

```

PUBLIC SUB btnCancel_Click()

```

```

    ME.btnCancel.Tag = 1
END

```

```

PUBLIC SUB Form_Open()
    'centra el formulario y pone como prioridad sobre el resto de
    formularios
    ME.Center
    ME.TopOnly = TRUE
    ME.btnCancel.tag = 0
END

```

```

PUBLIC SUB BtnIniciar_Click()
    ME.BtnIniciar.Enabled = FALSE
    'ejecuta el proceso de copia
    IF Message.Question("Esta seguro que el directorio " &
AppModulo.ArregloCargaDirectorios[1] & " es el directorio
correcto?", "SI", "NO") = 1 THEN
        ProcesoCopia
    ELSE
        IF ME.tag = "n" THEN
            WITH FormCarga
                .Enabled = TRUE
                .EliminarUltimoRegistro()
                '.LimpiarCampos
                .CargarNumeroCodigo
            END WITH
        ELSE
            ME.FormModificaApp.Enabled = TRUE
        ENDIF
    ENDIF

```

```
    ENDIF  
END
```

3.4. Verificación y Pruebas.

3.4.1. Pruebas de caja blanca.

Las métricas utilizadas son las planteadas por Thomas McCabe, en su enunciado de complejidad ciclomática de 1976, el cual consiste en evaluar los algoritmos y arrojar una medida cuantitativa de la dificultad de crear pruebas automáticas del código y también una medición orientativa de la fiabilidad del mismo.

La fórmula es: $V(a) = (\text{Aristas} - \text{Nodos}) + 2$

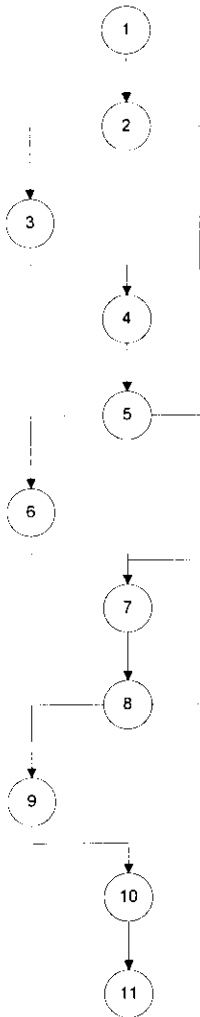
Donde $V(a)$ es la complejidad ciclomática, aristas representan las líneas de conexión y finalmente los Nodos que son los círculos que representan bloques de código.

Es importante también definir las rutas posibles que tiene la rutina al ejecutarse, específicamente cuando toma nodos predicado como decisiones IF o bucles DO o WHILE, de esta manera queda establecidos los cami

GESTION DE APLICACIONES

Explorador aplicaciones

a. Consulta registro

**Complejidad de McCabe:**

$$V(a) = 4$$

Prueba de condicionales:

C1=1-2-3-4-5-6-7-8-9-10-11

C2=1-2-3-4-5-6-7-8-10-11

C3=1-2-3-4-5-7-8-9-10-11

C4=1-2-3-4-5-7-8-10-11

C5=1-2-4-5-6-7-8-9-10-11

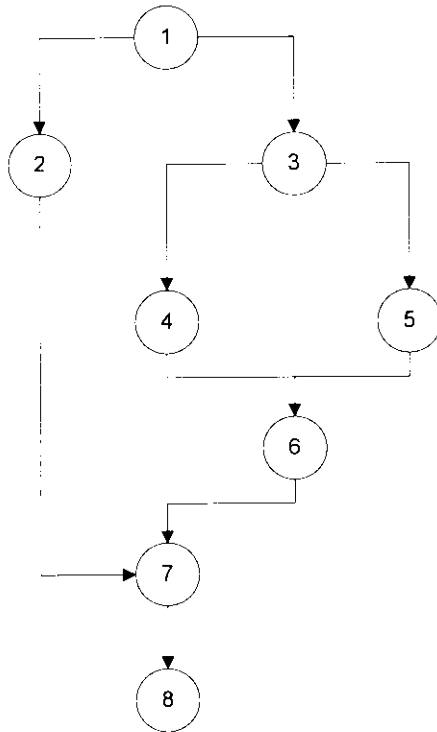
C6=1-2-4-5-6-7-8-10-11

C7=1-2-4-5-7-8-9-10-11

C8=1-2-4-5-7-8-10-11

Pasa y aprueba todos los condicionales.

b. Contenedor directorio

**Complejidad de McCabe:**

$$V(a) = 3$$

Prueba de condicionales:

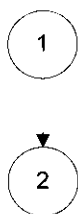
C1=1-2-7-8

C2=1-3-4-6-7-8

C3=1-3-5-6-7-8

Pasa y aprueba todos los condicionales.

c. Registro salida aplicación

**Complejidad de McCabe:**

$$V(a) = 1$$

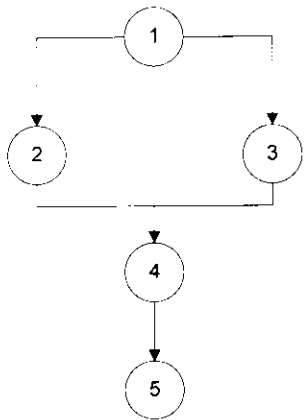
Prueba de condicionales:

C1=1-2

Pasa y aprueba todos los condicionales.

Constructor aplicaciones

a. Registro entrada aplicación



Complejidad de McCabe:

$$V(a) = 2$$

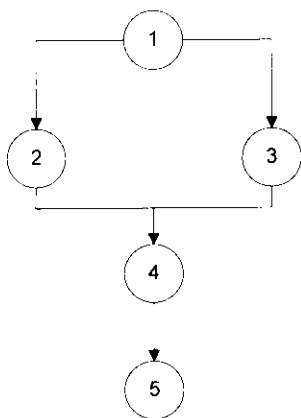
Prueba de condicionales:

C1=1-2-4-5

C2=1-3-4-5

Pasa y aprueba todos los condicionales.

b. Generador evento



Complejidad de McCabe:

$$V(a) = 2$$

Prueba de condicionales:

C1=1-2-4-5

C2=1-3-4-5

Pasa y aprueba todos los condicionales.

c. Extractor de archivos



Complejidad de McCabe:

$$V(a) = 1$$

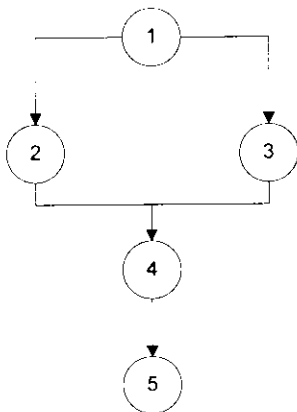
Prueba de condicionales:

$$C1=1-2$$

Pasa y aprueba todos los condicionales.

Grabador archivos de aplicaciones

a. Extractor archivos



Complejidad de McCabe:

$$V(a) = 2$$

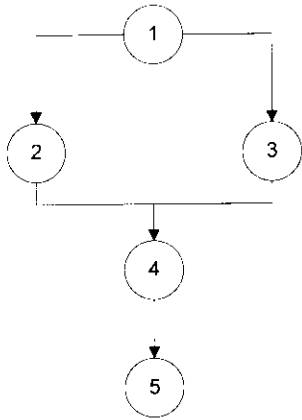
Prueba de condicionales:

$$C1=1-2-4-5$$

$$C2=1-3-4-5$$

Pasa y aprueba todos los condicionales.

b. Consulta parámetros



Complejidad de McCabe:

$$V(a) = 2$$

Prueba de condicionales:

C1=1-2-4-5

C2=1-3-4-5

Pasa y aprueba todos los condicionales.

c. Generador evento



Complejidad de McCabe:

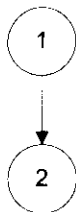
$$V(a) = 1$$

Prueba de condicionales:

C1=1-2

Pasa y aprueba todos los condicionales.

d. Constructor estructuras



Complejidad de McCabe:

$$V(a) = 1$$

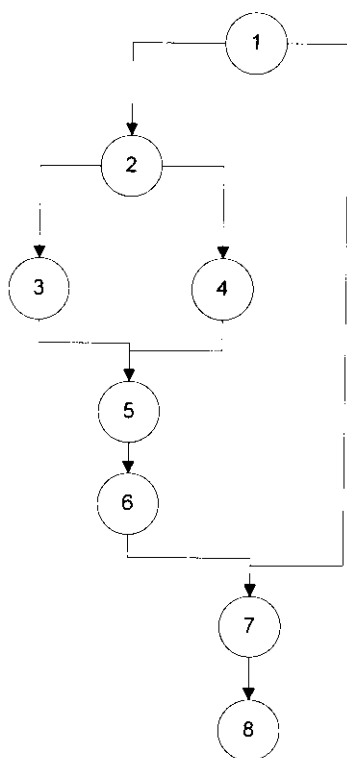
Prueba de condicionales:

C1=1-2

Pasa y aprueba todos los condicionales.

Generador de reporte de aplicaciones

a. Validador permiso usuario



Complejidad de McCabe:

$$V(a) = 3$$

Prueba de condicionales:

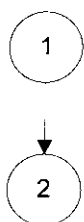
C1=1-2-3-5-6-7-8

C2=1-2-4-5-6-7-8

C3=1-7-8

Pasa y aprueba todos los condicionales.

b. Lector registro



Complejidad de McCabe:

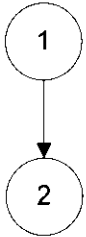
$$V(a) = 1$$

Prueba de condicionales:

C1=1-2

Pasa y aprueba todos los condicionales.

c. Generador reporte



Complejidad de McCabe:

$$V(a) = 1$$

Prueba de condicionales:

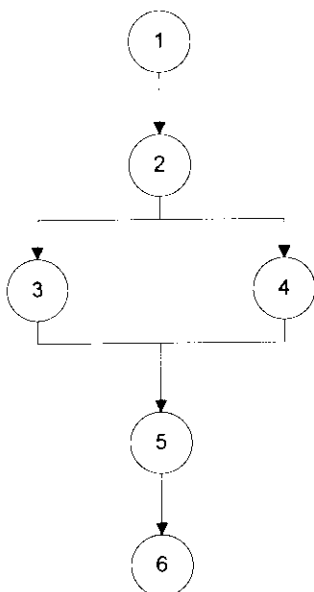
C1=1-2

Pasa y aprueba todos los condicionales.

GESTION DE USUARIOS

Validador usuarios

a. Consultor



Complejidad de McCabe:

$$V(a) = 2$$

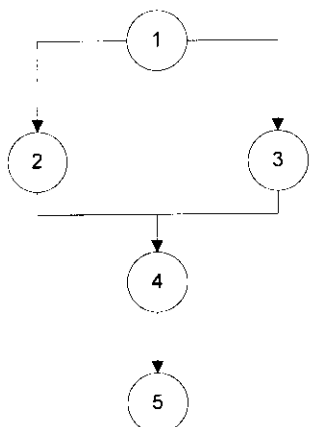
Prueba de condicionales:

C1=1-2-3-5-6

C2=1-2-4-5-6

Pasa y aprueba todos los condicionales.

b. Comparador

**Complejidad de McCabe:**

$$V(a) = 2$$

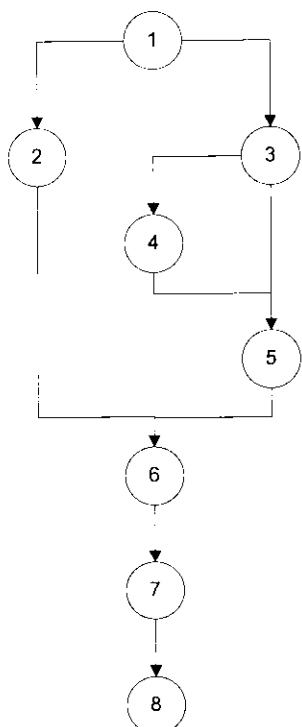
Prueba de condicionales:

C1=1-2-4-5

C2=1-3-4-5

Pasa y aprueba todos los condicionales.

c. Validador

**Complejidad de McCabe:**

$$V(a) = 3$$

Prueba de condicionales:

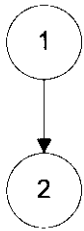
C1=1-2-6-7-8

C2=1-3-4-5-6-7-8

C3=1-3-5-6-7-8

Pasa y aprueba todos los condicionales.

d. Generador evento



Complejidad de McCabe:

$$V(a) = 1$$

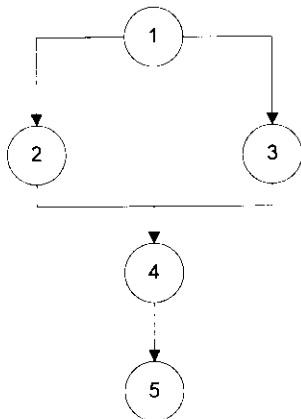
Prueba de condicionales:

$$C1=1-2$$

Pasa y aprueba todos los condicionales.

Constructor usuarios

a. Constructor registro



Complejidad de McCabe:

$$V(a) = 2$$

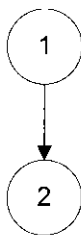
Prueba de condicionales:

$$C1=1-2-4-5$$

$$C2=1-3-4-5$$

Pasa y aprueba todos los condicionales.

b. Generador evento



Complejidad de McCabe:

$$V(a) = 1$$

Prueba de condicionales:

$$C1=1-2$$

Pasa y aprueba todos los condicionales.

Generador reporte usuarios

a. Lector de registro



Complejidad de McCabe:

$$V(a) = 1$$

Prueba de condicionales:

$$C1=1-2$$

Pasa y aprueba todos los condicionales.

b. Generador reporte



Complejidad de McCabe:

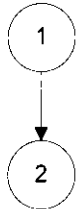
$$V(a) = 1$$

Prueba de condicionales:

$$C1=1-2$$

Pasa y aprueba todos los condicionales.

CONFIGURADOR SISTEMA

Configurador de parámetros**Complejidad de McCabe:**

$$V(a) = 1$$

Prueba de condicionales:

$$C1=1-2$$

Pasa y aprueba todos los condicionales.

Generador reporte parámetros

a. Lector registro

**Complejidad de McCabe:**

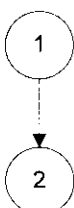
$$V(a) = 1$$

Prueba de condicionales:

$$C1=1-2$$

Pasa y aprueba todos los condicionales.

b. Generador reporte

**Complejidad de McCabe:**

$$V(a) = 1$$

Prueba de condicionales:

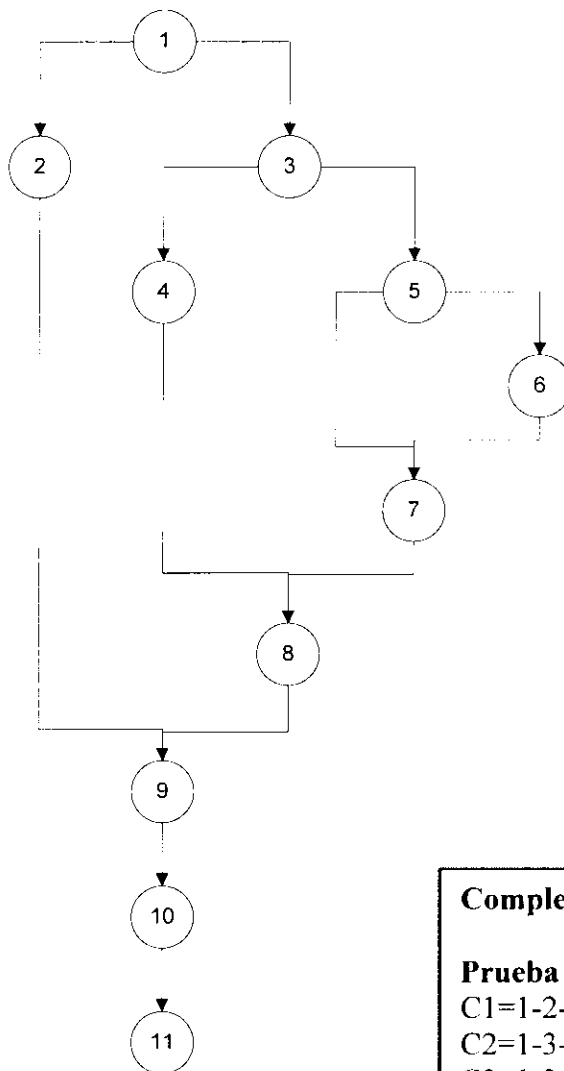
$$C1=1-2$$

Pasa y aprueba todos los condicionales.

EVENTOS SISTEMA

Constructor de eventos

a. Discriminador evento

**Complejidad de McCabe:**

$$V(a) = 4$$

Prueba de condicionales:

C1=1-2-9-10-11

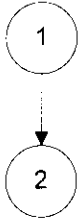
C2=1-3-4-8-9-10-11

C3=1-3-5-7-8-9-10-11

C4=1-3-5-6-7-8-9-10-11

Pasa y aprueba todos los condicionales.

b. Registro evento



Complejidad de McCabe:

$$V(a) = 1$$

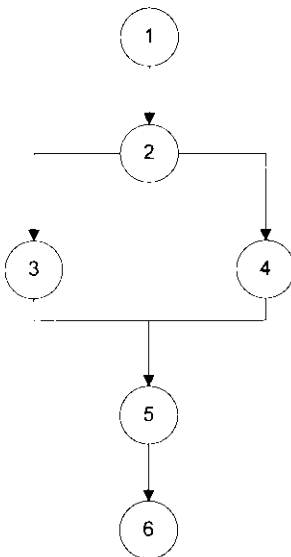
Prueba de condicionales:

$$C1=1-2$$

Pasa y aprueba todos los condicionales.

Generador reporte de eventos

a. Lector registro



Complejidad de McCabe:

$$V(a) = 2$$

Prueba de condicionales:

$$C1=1-2-3-5-6$$

$$C2=1-2-4-5-6$$

Pasa y aprueba todos los condicionales.

b. Generador eventos

**Complejidad de McCabe:**

$$V(a) = 1$$

Prueba de condicionales:

$$C1=1-2$$

Pasa y aprueba todos los condicionales.

3.4.2. Pruebas de caja negra.

Se ha realizado las pruebas necesarias enfocadas en tres ámbitos para determinar que la aplicación está arrojando los resultados esperados.

Ámbito número uno: Pruebas sobre la base de datos.

Por medio de un control detallado se ha comprobado que la aplicación guarda en cada campo de cada tabla de la base de datos de MySQL la información, correcta, completa y consistente, correspondiente a los registros de aplicaciones, usuarios y eventos del sistema.

```

'Funcion en el módulo para abrir la conexión a MySQL
IF AppModule.ConectarMySQL("localhost", "root", "12345", "BaseKiosco") = "conectado" THEN
  Message("Conectado a MySQL", "Aceptar")
ELSE
  Message("Error al conectar a MySQL", "Aceptar")
ME.Close
END IF
  
```

Ilustración 3. 39: Código fuente - conexión a MySQL

En la imagen 3.39 Se muestra una llamada a la función “ConectarMySQL” la cual devuelve un valor verdadero si no hay problemas de conexión o falso si hay problemas de esta manera informa si hay o no problemas con la conexión al sistema gestor de base de datos MySQL.

```

ELSE
    Message.Error("No se pudo obtener los parametros del sistema. Contacte al administrador")
ENDIF
InformacionSistema = db.Exec("SELECT * FROM Informacion WHERE CodInformativo = 1")
IF InformacionSistema.Available THEN
    ME.TxtLblUniversidad.Text = InformacionSistema["NombreInstitucion"] & " " & InformacionSistema["Sede"]
    ME.TxtLblInfo.Text = InformacionSistema["Responsable"] & " / " & InformacionSistema["VersionNotas"]
ELSE
    Message.Error("No se pudo obtener los parametros del sistema. Contacte al administrador")
ENDIF
CATCH 'se ejecuta solo si hay un error'
    Message.Error("Error #: " & Error.Code & " - " & Error.Text & " Contacte al administrador")
ME.Close

```

Ilustración 3. 40: Código fuente - control de errores de conexión a BDD.

La imagen 3.40 muestra un mensaje de error de conexión al servidor de base de datos MySQL en la segunda línea, además, muestra la implementación de un control de errores “catch”, al momento que no se ha podido obtener un registro desde una tabla dentro de la base de datos. Con esto se mantiene al sistema funcional a pesar de que existan problemas con la base de datos.

Errores comunes de programación en consultas SQL generan errores como los mostrados en las tres imágenes a continuación; se ha corregido la sintaxis para evitar errores al momento del funcionamiento del sistema.

```

1146 Table 'BaseKiosco.Clasificacion' doesn't exist

```

Ilustración 3. 41: Errores de consola MySQL - error de sintaxis.

1062 Duplicate entry '1' for key 'PRIMARY'

Ilustración 3. 42: Errores de consola MySQL - restricciones de BDD.

1064 You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax

Ilustración 3. 43: Errores de consola MySQL - error de sintaxis.

Finalmente la ilustración 3.44 muestra un registro de base de datos, en el cual no existe error al momento de guardar, por ende ha almacenado la información correctamente en la tabla.

Administrador	IDUsuario	Contrasena	PBorrado	PCarga
1	Administrador	12345	-1	-1
0	dave	1234	-1	-1

Ilustración 3. 44: Query browser MySQL - Tabla de Usuarios.

Ámbito número dos: Pruebas sobre los algoritmos y procesos de la aplicación

Se determinó que la aplicación cumple con todos los pasos de los algoritmos establecidos, los cálculos y subprocesos realizados en el sistema durante su ejecución cumplen con lo preestablecido en la fase de diseño de procesos. Además las instrucciones que envía la aplicación hacia la consola del sistema son ejecutadas de manera correcta.

Se ha implementado control de errores en procesos esenciales del sistema lo que permite al administrador conocer cuando existe un fallo y lo más importantes, permitiendo continuar con la ejecución del programa hasta que el error sea corregido ya sea por una mala configuración de los parámetros del sistema o problemas en el código fuente.

Además los errores provocados por las rutinas propias del sistema son corregidos con mensajes informativos y retorno al estado inicial del sistema hasta antes de que se suscite el error, la ilustración 3.45 Muestra un control al momento que un dispositivo se queda sin espacio para almacenamiento, y además registra en la base de datos sobre el error para su posterior auditoria.

```
ELSE
  LblInfo.Text = "INACTIVO..."
  Message.Error("No hay espacio libre suficiente en la carpeta de destino. Se ha cancelado el proceso de copia")
  FormDescarga.enabled = TRUE
  AppModulo.RegistraEventos("Error", "Usuario Final", "Se canceló la descarga de la aplicación con índice: " & F
  "Cierra el formulario actual
  ME.Close
ENDIF
```

Ilustración 3. 45: Código fuente - errores de control a problemas externos de almacenamiento.

El control de registros activos en controles como grillas, son controlados con instrucciones de decisión IF, con esto se evita escritura de registros con campos vacios o nulos en base de datos o lectura de registros con consultas nulas o incompletas.

```
IF Tabla.Current = NULL THEN
  Message.Error("No existe un registro activo", "OK")
```

Ilustración 3. 46: Código Fuente - Control de errores en grillas.

Por otro lado si el sistema ha encontrado errores en módulos anexos o aplicaciones de terceras partes, informa y controla por medio de banderas que se activan al momento de ejecutarse una rutina con error de esta manera, el sistema sabe qué camino tomar si sucede algo de esta naturaleza y de esta manera informa al usuario sobre el problema para tomar medidas correctivas o preventivas. Esto puede verse implementado en la imagen a continuación.

```

IF AppModule.BanderaError = TRUE THEN
  IF AppModule.BanderaCancelarGrabacion = 1 THEN
    Message.Error("Cancelado por el usuario")
  ENDIF.
  'en caso de error de la copia
  Message.Error("Error en la creación del CD/DVD contacte al administrador del sistema")
  'registra el error
  TRY AppModule.RegistraEventos("Error", "Usuario Final", "Se canceló la descarga de la a
ELSE
  'registra el evento de descarga
  AppModule.RegistraEventos("Descarga", "Usuario Final", "Se descargo la aplicación con i
ENDIF.

```

Ilustración 3. 47: Código fuente - banderas para control de errores.

En la imagen 3.48 se muestra el registro de errores, que luego fueron corregidos y se registra una descarga completa, este registro se encuentra dentro de una tabla de la base de datos.

133	Error	2010-08-16 20:21:00	USUARIO FINAL	Se canceló la descarga de la aplicación con índice: 4 y nombre: Attribute beta
134	Error	2010-08-16 20:24:00	USUARIO FINAL	Se canceló la descarga de la aplicación con índice: 4 y nombre: Attribute beta
135	Descarga	2010-08-16 20:29:00	USUARIO FINAL	Se descargo la aplicación con índice: 2 y nombre: Avira 10 Profesional
136	Descarga	2010-08-16 20:31:00	USUARIO FINAL	Se descargo la aplicación con índice: 3 y nombre: carga para manual
137	Ingreso	2010-08-16 20:33:00	ADMINISTRADOR	Ingreso al sistema

Ilustración 3. 48: Consola MySQL - Tabla eventos.

Errores de programación especialmente en lo que a sintaxis respecta son mostrados por el entorno propio del codificador o depurador del lenguaje de programación utilizado, la imagen 3.49 muestra un error en sintaxis al

momento de codificar, lo que permite su rápida corrección; mientras que la imagen 3.50 muestra una instancia incorrecta debido a problemas de sintaxis o por mal instanciamiento de clases, funciones, métodos o bloques de programación similares.

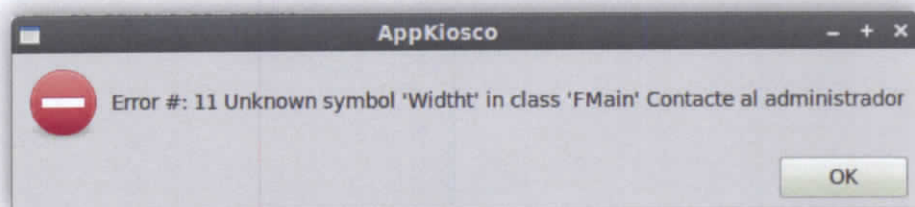


Ilustración 3. 49: Error en gambas - sintaxis.

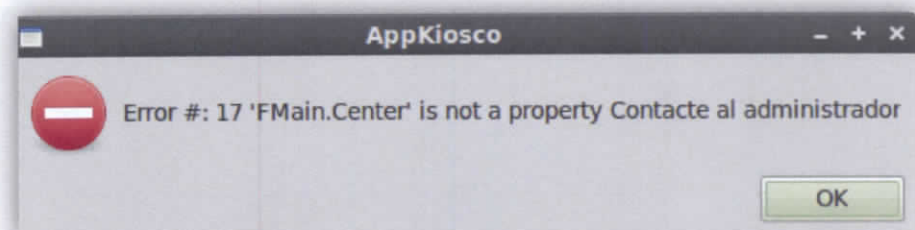


Ilustración 3. 50: Error en gambas - error en propiedades

Ámbito número tres: Pruebas de carga y descarga de archivos.

Tanto en la carga de archivos desde medios extraíbles hacia al servidor como la descarga de archivos desde el servidor a medios extraíbles, han sido controlados, verificando que los archivos copiados sean consistentes, cumpliendo con su integridad y haciéndolos fiables.

Se ha probado que los archivos almacenados y descargados se puedan visualizar, ejecutar y utilizar en sistemas operativos Linux y Windows.

Aquí se muestra por medio de la ilustración 3.51 y la ilustración 3.52 la ubicación, tamaño y cada archivo dentro de la carpeta que se pretende copiar desde el sistema gestor de aplicaciones y se comprueba que no hay errores por medio de la imagen 3.53 y 3.54 que indican que se realizó la copia de la carpeta que contiene la aplicación de manera correcta, en este caso el instalador del antivirus en versión demostrativa de Avira® versión 10 profesional.

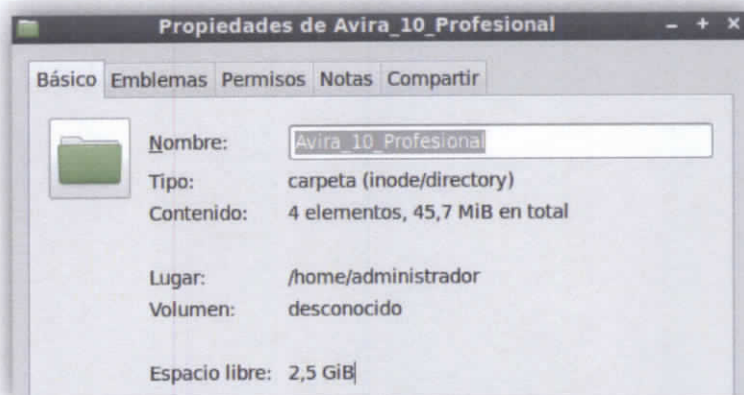


Ilustración 3. 51: Propiedades de carpeta en Linux Ubuntu.



Ilustración 3. 52: Archivos de carpeta en Nautilus de Ubuntu.

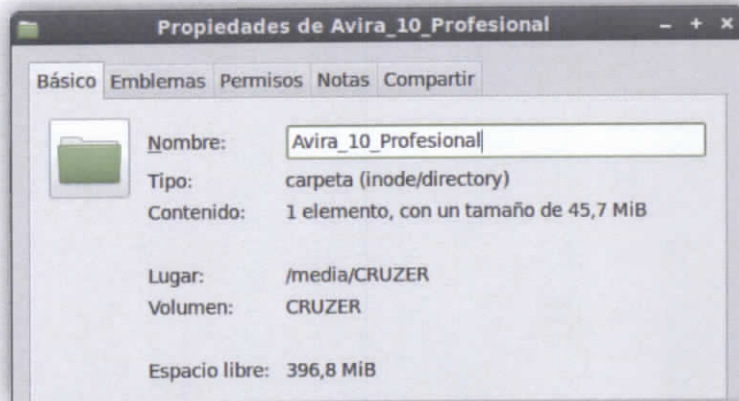


Ilustración 3. 53: Propiedades de carpeta en Linux Ubuntu

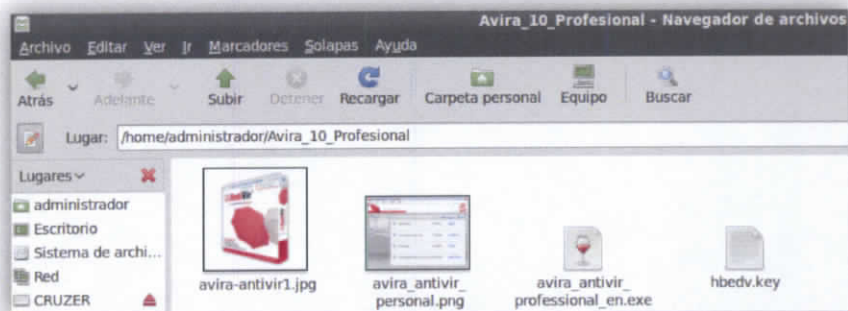


Ilustración 3. 54: Archivos de carpetas en Nautilus de Ubuntu

3.5. Formularios y su descripción

Nombre	Descripción
Formularios	
ControlUsuario	Formulario para ingresar usuario y contraseña, permite el ingreso del usuario editor o administrador al sistema
Fmain	Pantalla principal de la aplicación contiene un menú, botones de: carga, descarga, expulsión de memorias. Invoca parámetros iniciales
FormAgregarUsuario	Formulario para agregar usuarios al sistema
FormCarga	Formulario maestro para cargar información sobre las aplicaciones, ejecuta instrucciones SQL al crear el registro pero no ejecuta comandos de consola
FormDescarga	Formulario maestro para descargar las aplicaciones almacenadas, construye los comandos a ejecutarse en consola a partir de las decisiones del usuario referentes a los medios de almacenamiento destino
FormDialogoCarga	Ejecuta los comandos de consola para copiar los archivos origen al disco duro del servidor y muestra el progreso de la copia
FormDialogoCargaManualesSShots	Cuadro de diálogo para escoger archivos de imágenes o pdf, su filtro depende de parámetros enviados por FormCarga
FormDialogoEscogeCarpetaApp	Cuadro de diálogo para escoger la carpeta que contiene los archivos orígenes de la aplicación que se desea cargar su filtro es "/media" definido por defecto
FormExpulsaMemoria	Formulario que muestra las memorias del sistema y ejecuta instrucciones de consola para desmontar el volumen y poder extraerla de forma segura
FormGestionAplicaciones	Formulario que presenta en base a una consulta SQL las aplicaciones almacenadas en el sistema para su eliminación o modificación
FormInformaciónSistema	Formulario completo para almacenar eliminar y modificar la información referente a categorías y materias, se basa en instrucciones SQL e inputbox
FormInformativoISO	Formulario que informa sobre el progreso de copia de archivos desde el servidor hacia el medio óptico destino, recibe parámetros de FormDescarga y ejecuta instrucciones de consola para cumplir con la tarea. Utiliza un lector de procesos para mostrar el progreso

Nombre	Descripción
FormMemoriasDialogo	Formulario que informa sobre el progreso de copia de archivos desde el servidor hacia el medio masivo destino, recibe parámetros de FormDescarga y ejecuta instrucciones de consola para cumplir con la tarea. Utiliza un comando de consola para mostrar el progreso
FormModificaApp	Formulario invocado por FormGestionAplicaciones, permite modificar la informacion concerniente a una determinada aplicación cargada en el servidor, solo se basa en instrucciones SQL ya que modifica la BBD mas no los archivos
FormModificarEliminarUsuario	Formulario para modificar o eliminar usuarios del sistema por medio de instrucciones SQL
FormPantallaCompletaImagen	Contiene un cuadro de imagen que muestra una archivo de imagen en base a una ruta de acceso que ha sido enviada como parámetro desde el formulario FormDescarga
FormParametros	Formulario que permite visualizar y modificar los parámetros del sistema. Ejecuta para esto solamente instrucciones SQL.
FormPermisoUsuario	Formulario que permite establecer los roles del usuario, por medio de una instrucción SQL modifica el tipo de usuario en la tabla Usuario de la base de datos.
FormReporte	Formulario que por medio de una consulta SQL a la base de datos presenta en pantalla la información de los eventos ocurridos en el sistema.
Módulos	
AppModulo	El módulo contiene todas las variables de tipo global como funciones y procedimientos de carácter global para que puedan ser invocadas desde cualquier formulario.

Tabla 3. 56: Formularios dentro del sistema gestor de aplicaciones

CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones.

- El kiosco tecnológico de la Escuela de Sistemas de la PUCESA fue implementado con una plataforma, confiable, fiable, sólida, eficaz y efectiva para la gestión de aplicaciones de uso y desarrollo de los estudiantes; demostrando que, el desarrollo de aplicativos en Linux y en software libre en general es tan positivo y aportador como el desarrollo de aplicaciones en software privativo, incluso superándolo en muchos aspectos.
- El almacenamiento y descarga directa de aplicaciones desde el Kiosco Tecnológico ahorra el paso de descargar desde la red de la Universidad paquetes que resultaban demasiado lentos y extensos, permitiendo de esta manera recortar el tiempo que los estudiantes invertían con anterioridad en la descarga de aplicaciones de uso estudiantil.
- Linux Ubuntu se muestra muy sólido y fiable como sistema operativo, además confirma que su interfaz es muy amigable para el usuario, demuestra ser inmune a los ataques de virus que en la gran mayoría

están desarrollados para vulnerar los sistemas operativos de la compañía Microsoft.

- Gambas prestó los elementos necesarios para que el sistema desarrollado en su lenguaje no resulte complicado al momento de entender y programar para un desarrollador de sistemas, incluso si nunca utilizó el lenguaje de programación gambas anteriormente. Comparando con Visual Basic de Microsoft; uno de los lenguajes de programación más conocidos en el mundo, gambas no tiene que envidiar nada, cierto es que aún faltan algunos aspectos que cubrir y errores que corregir Gambas resulta un lenguaje muy poderoso y fácil de utilizar al momento de codificar aplicaciones de esta naturaleza.
- El proyecto cubrió y superó las expectativas del grupo de trabajo involucrado ya que Linux Ubuntu y Gambas se acoplaron de manera perfecta al hardware del servidor, permitiendo explotar a toda la capacidad los recursos del servidor y demostrando que el software libre incluye funciones, que en otros sistemas operativos, resultan muy complejas o demasiado costosas para cubrir las necesidades del cliente y de negocio.

4.2. Recomendaciones.

- El sistema gestor de aplicaciones a pesar de ser muy eficiente y seguro en cuanto al almacenamiento de aplicaciones debe estar administrado por una persona responsable y que conozca perfectamente el tipo y clase de información que se pretenda cargar al sistema; ya que, el Kiosco y su sistema gestor de aplicaciones no están exentas del error humano. Por ejemplo, subir información conocida como basura, que no es de provecho para el usuario final que la descarga.
- El desarrollo de aplicaciones en software libre debe ser promovido con más fuerza ya que además de obtener los beneficios del software libre, se incentiva a que más personas utilicen sistemas operativos de Linux como Mandriva, Fedora, Ubuntu, CentOS, etc., inculcando el uso de éstos e incentivando a desarrolladores a mejorar dichos sistemas y aplicativos que utilizan plataformas Linux.
- La PUCESA pionera en desarrollo tecnológico debe explotar los beneficios del software libre e incluir en el aprendizaje de sus estudiantes lo que respecta a las diferentes opciones y configuraciones de los principales sistemas operativos con base en Unix disponibles para su libre descarga en la internet.

- El kiosco tecnológico de la escuela de sistemas y su sistema gestor de aplicaciones, cubre todos los requerimientos planteados por los interesados del proyecto, sin embargo queda la puerta abierta para que se diseñe, codifique e incorpore módulos para mejorar los servicios que presta al usuario final, por ejemplo: la selección múltiple de aplicaciones para grabarlas en una sola unidad óptica, incorporar solución para borrar CD y DVD reescribibles, reportes estadísticos de descargas y cargas de aplicaciones, enlace a la red de la universidad para identificar a los usuarios finales, creación de instaladores automáticos para aplicaciones con auto ejecución en medios ópticos, almacenamiento directo de imágenes ISO al servidor a partir de unidades ópticas, entre otras que surjan de la creatividad de los estudiantes como quienes son los beneficiarios directos de las prestaciones del sistema.

BIBLIOGRAFÍA.

Libros

Daniel Campos – José Luis Redrejo, GAMBAS Programación visual con software libre, Editorial EditLin.

Roger Pressman, Ingeniería de Software, 5ta edición, Editorial Mc Graw Hill, febrero 2002.

Vínculos web

Seudónimo: Soplo, Foro Gambas - El control dataview, Septiembre 2009, <http://gambas-es.org/viewtopic.php?f=4&t=38>

Julio, Seudónimo: Jsbsan, Foro Mis proyectos en gambas, febrero 2009, <http://jsbsan.wordpress.com/>

Edro the bofh – Acceso a MySQL desde gambas, 22 septiembre 2009, <http://edrossuse.blogspot.com/2009/03/acceso-mysql-desde-gambas.html>

Seudónimo: Beagle Skywalker, Linux six blog – grabar a cd o dvd desde consola, 4 de octubre de 2008, <http://linuxsix.blogspot.com/2008/10/grabar-cd-o-dvd-desde-consola.html>

Paquillo Dubois, Un bioinformatiquillo - crear imágenes iso de cd dvd y hdd desde consola, 15 de junio de 2008, <http://bioinformatiquillo.wordpress.com/2008/06/15/crear-imagines-iso-de-cddvdhdd-desde-consola/>

Paco Aldarias Raya, Archivos comprimidos zip,tar,gz,bzip, 7 de febrero 2005, <http://pacodebian.iespana.es/comprimir.html>

Stefan Hinz, Manual de referencia SQL 5, Copyright 1997-2010, <http://dev.mysql.com/doc/refman/5.0/es/>

Creative Commons Attribution-ShareAlike (CC-BY-SA) 3.0, Gambas documentation, <http://gambasdoc.org/help/lang/shell?es>
Julio Montaña, Programación en Gambas, 8 de diciembre de 2007, <http://gambaslinux.wordpress.com/>

Seudónimo: Sasalha, Crear Isos y quemarlos con sencillos comandos, 11 de agosto de 2009, <http://www.taringa.net/posts/linux/3159044/Crear-Isos-y-quemarmas,-con-sensillos-comandos.html>

Hurricane Electrics, Linux Man Pages Online, Copyright 1994 -2010, <http://man.he.net/>

Henrik Nilsen Omma, Accessibility/Projects/onboard, agosto de 2006, <https://wiki.ubuntu.com/Accessibility/Projects/onBoard/>

Seudónimo: House, Software Libre – Wikipedia, Julio 2010, http://es.wikipedia.org/wiki/Software_libre

Free software foundation, La definición de software libre, Copyright 1996 – 2010, <http://www.gnu.org/philosophy/free-sw.es.html>

Canonical LTD, Ubuntu Server edition, 2010, <http://www.ubuntu.com/products/whatisubuntu/serveredition>

Jesús Vergara, Desarrollo en espiral, 23 de abril de 2004, <http://gsync.escet.urjc.es/~jvergara/memoria-www/node26.html>

GLOSARIO

A

API: (Application Programming Interface) es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción al momento de programar.

ANSI: Instituto Nacional Estadounidense de Estándares (American National Standards Institute) es una organización sin ánimo de lucro que supervisa el desarrollo de estándares para productos, servicios, procesos y sistemas en los Estados Unidos.

ATAPI: (Attachement Packet Interface). ATAPI es el protocolo mediante el cual las unidades de CD-ROM se comunican con la computadora sobre la interfaz IDE.

ATA: (Advanced Technology Attachment) controla los dispositivos de almacenamiento masivo de datos, como los discos duros y ATAPI, es conocido como IDE.

Apache: Servidor web de distribución libre y de código abierto.

B

BASIC®: Lenguaje de programación que originalmente fue desarrollado como una herramienta de enseñanza; se diseminó entre las microcomputadoras hogareñas a partir de la década de 1980. Actualmente continúa siendo muy popular, en muchos dialectos bastante distintos del original.

BlueRay: Conocido como BD. Formato de disco óptico de nueva generación de 12 cm de diámetro (igual que el CD y el DVD) para vídeo de alta definición y almacenamiento de datos de alta densidad.

C

CMM-I: Integración de Modelos de Madurez de Capacidades o Capability Maturity Model Integration (CMMI) es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software.

Casos de uso: Secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas.

Copyleft: Forma de licencia y puede ser usado para modificar el derecho de autor de obras o trabajos, tales como software de computadoras, documentos, música, y obras de arte.

Copyright: Derecho que tiene un autor, incluido el autor de un programa informático, sobre todas y cada una de sus obras y que le permite decidir en qué condiciones han de ser éstas reproducidas y distribuidas. Aunque este derecho es legalmente irrenunciable puede ser ejercido de forma tan restrictiva o tan generosa como el autor decida. El símbolo de este derecho es ©.

C, Lenguaje C: C es un lenguaje de programación creado en 1972 por Dennis M. Ritchie en los Laboratorios Bell orientado al desarrollo de sistemas operativos.

CGI: Una interfaz para posibilitar la interacción entre una página y el servidor web, mediante la invocación de aplicaciones.

D

Distribución: Distribución de software basada en el núcleo Linux que incluye determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones domésticas, empresariales y para servidores. Por lo general están

compuestas, total o mayoritariamente, de software libre, aunque a menudo incorporan aplicaciones o controladores propietarios.

Distro: Seudónimo de Distribución de software Linux.

Debian: Debian es un sistema operativo gratuito, una de las distribuciones de Linux más populares e influyentes.

DB2: DB2 es una marca comercial, propiedad de IBM, bajo la cual se comercializa un sistema de gestión de base de datos.

DCOP: (Desktop COmmunication Protocol) es un sistema de comunicación ligero entre procesos y componentes software.

E

ERS: Documento que detalla todas las especificaciones de requerimientos de un software previo a su diseño, documento estructurado para uso del diseñador del software.

Endianness: Formato en el que se almacenan los datos de más de un byte en un ordenador.

F

Freeware: Software de distribución gratuita. Programas que se distribuyen a través de Internet de forma gratuita.

Firebird: Sistema de administración de base de datos relacional (o RDBMS) (Lenguaje consultas: SQL) de código abierto, basado en la versión 6 de Interbase, cuyo código fue liberado por Borland en 2000.

Frontend: La parte de un programa que interactúa directamente con un usuario.

G

GNU: (GNU is Not Unix) Sistema operativo para ser distribuido reutilizado libremente creado por la Free Software Foundation (FSF).

GPL: General Public License (Licencia Publica General). Regula los derechos de autor de los programas de software libre promovido por el The Free Software Foundation (FSF).

GNU gettext: Librería de GNU que se utiliza en plataformas multilinguaje.

H

HFS: Sistema de Archivos Jerárquico o Hierarquical File System (HFS), es un sistema de archivos desarrollado por Apple Inc. para su uso en computadores que corren Mac OS.

HD-DVD: Término genérico para denominar la tecnología de grabación de video de alta definición en un DVD. En general, el HD-DVD es capaz de almacenar entre dos y cuatro veces más datos que un DVD estándar.

I

IEEE: (Institute of Electrical and Electronics Engineers). Asociación de profesionales norteamericanos que aporta criterios de estandarización de dispositivos eléctricos y electrónicos.

InnoDB: Tecnología de almacenamiento de datos de código abierto para la base de datos MySQL.

ISO9660: El estándar ISO 9660 es una norma publicada inicialmente en 1986 por la ISO, que especifica el formato para el almacenaje de archivos en los soportes de tipo disco compacto. El estándar ISO 9660 define un sistema de archivos para CD-ROM.

ISO: Archivo donde se almacena una copia o imagen exacta de un sistema de ficheros, normalmente un disco óptico. Se rige por el estándar ISO 9660 que le da nombre.

IDE: Interfaz de transmisión de datos en bus dentro de equipos informáticos. Este sistema hace que un disco lleve integrado una placa que controle las funciones del dispositivo. Se lo conoce también como ATA.

J

Java: Lenguaje de programación orientado a objetos. Fue desarrollado por James Gosling y Sun Microsystems al principio de la década de los 90. La programación en Java es compilada en bytecode, el cuál es ejecutado por la máquina virtual Java.

JOLIET: Sistema de archivos extensión de ISO9660, capaz de leer nombre nombres de ficheros de más de ocho caracteres en Windows®.

K

Kernel: Parte esencial de un sistema operativo que provee los servicios más básicos del sistema. Se encarga de gestionar los recursos como el acceso seguro al hardware de la computadora.

KDE: Proyecto de software libre para la creación de un entorno de escritorio e infraestructura de desarrollo para diversos sistemas operativos como GNU/Linux, Mac OS X, Windows, etc.

M

Mozilla: La Fundación Mozilla es una organización sin ánimo de lucro dedicada a la creación de software libre. Tiene como misión "mantener la elección y la innovación en Internet".

MyISAM: Tecnología de almacenamiento de datos usada por defecto por el sistema administrador de bases de datos relacionales MySQL.

MS-DOS: (MicroSoft Disk Operating System), Sistema operativo de disco de Microsoft. Es un sistema operativo comercializado por Microsoft perteneciente a la familia DOS. Fue un sistema operativo para el IBM PC.

O

Oracle: Oracle es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), desarrollado por Oracle Corporation.

ODBC: Open DataBase Connectivity (ODBC) es un estándar de acceso a Bases de datos desarrollado por Microsoft Corporation.

OpenGL: (Open Graphics Library) es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D.

OpenSource: Código abierto en español, cualidad de algunos programas de computador de incluir el código fuente en la distribución del programa. En general se usa para referirse al software libre.

OSI: Open Source Initiative (OSI) organización dedicada a la promoción del código abierto. Fue fundada en febrero de 1998 por Bruce Perens y Eric S. Raymond.

P

PHP: Hypertext Pre-processor). Lenguaje de programación usado generalmente en la creación de contenidos para sitios web. Es un lenguaje interpretado especialmente usado para crear contenido dinámico web y aplicaciones para servidores, aunque también es posible crear aplicaciones gráficas utilizando la biblioteca GTK+.

Perl: Perl (Lenguaje Práctico para la Extracción e Informe) es un lenguaje de programación creado por Larry Wall en 1987. Perl toma características

del C, del lenguaje interpretado shell (sh), AWK, sed, Lisp y, en un grado inferior, muchos otros lenguajes de programación.

Python: Lenguaje de programación desarrollado como proyecto de código abierto desarrollado por Python software Foundation. Se trata de un lenguaje de programación en scripts, permite dividir el programa en módulos reutilizables desde otros programas Python.

Postgresql: PostGreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley, es una derivación libre del proyecto y utiliza el lenguaje SQL92/SQL99.

Paquetes: Ficheros que contienen software; están escritos en un cierto formato que permite la fácil instalación y borrado del software.

POSIX: Portable Operating System Interface, Interfaz para Sistemas Operativos migrables basados en UNIX.

Plugins: Aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

R

Repositorios: Repositorio es un almacén de paquetes generalmente el termino es asociado a ambientes UNIX. Los repositorios pueden estar en

internet, en un medio extraíble como un CD, en el disco duro u otro medio de almacenamiento masivo.

Ruby: Lenguaje de programación orientado a objetos creado por el programador japonés Yukihiro "Matz" Matsumoto en 1993. Combina una sintaxis inspirada en Python, Perl con características de programación orientada a objetos similares a Smalltalk.

Red Gigabit: Redes Ethernet de 1000 Mbps.

Rock Ridge: Rock Ridge Interchange Protocol (RRIP, IEEE P1282) es una extensión al estándar ISO 9660 al cual añade requerimientos para sistemas de ficheros tipo POSIX.

S

SAN: (Storage Area Network), es una red concebida para conectar servidores, matrices (arrays) de discos y librerías de soporte. Principalmente, está basada en tecnología fibre channel y más recientemente en iSCSI.

SQLite: Sistema de gestión de bases de datos relacional compatible con ACID, y que está contenida en una relativamente pequeña biblioteca en lenguaje C.

SSL: (Secure Sockets Layer) Protocolo de Capa de Conexión son protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet.

SCSI: Small Computers System Interface (Sistema de Interfaz para Pequeñas Computadoras), es una interfaz estándar para la transferencia de datos entre distintos dispositivos del bus de la computadora.

SUSP: (System Use Sharing Protocol) Estandar propuesto por la IEEE para la inclusión de campos identificatorios del sistema de archivos que lleva una estructura de datos típicamente en CD-ROM.

U

UML: (Unified Modeling Language) El lenguaje para modelamiento unificado (UML), es un lenguaje para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema intensivo. Fue originalmente concebido por la Corporación Rational Software.

UNIX: Sistema operativo multiplataforma, multitarea y multiusuario desarrollado originalmente por empleados de Bell de AT&T que comparten código y propiedad intelectual.

UTF-8: (8-bit Unicode Transformation Format) es un formato de codificación de caracteres Unicode e ISO 10646 utilizando símbolos de longitud variable.

Unicode: Sistema universal de codificación. Permite su uso en múltiples lenguas. Actualmente soporta más de 90.000 caracteres. Es usado por sistemas Linux, Windows, y MacOS.

UDF: (Universal Disk Format), sistema de archivos con estandar ISO 9660 propiedad de Adaptec que utiliza las grabadoras de CD/DVD como un dispositivo de almacenamiento lógico. Este formato permite leer, escribir o modificar los archivos contenidos en discos CD/DVD reescribibles (RW) del mismo modo que se hace en el disco duro.

V

Visual Basic©: Lenguaje de programación desarrollado por el alemán Alan Cooper para Microsoft©. El lenguaje de programación es un dialecto de BASIC, con importantes agregados. Su primera versión fue presentada en 1991, con la intención de simplificar la programación utilizando un ambiente de desarrollo completamente gráfico.

ANEXOS.

Anexo A: Manual de Usuario

Manual de Usuario.

1. Visión general

El sistema gestor de aplicaciones, permite tanto la carga de aplicaciones desde un usuario hacia el sistema como también la descarga de aplicaciones desde el sistema hacia el usuario.

Para cumplir cualquiera de estas dos funciones principales del sistema se deben tomar diferentes caminos que se detallan en el presente manual de usuario, haciendo que las tareas dentro del sistema sean sencillas y rápidas.

Hay que tomar en consideración que este manual de usuario está diseñado para tres tipos de usuarios que se encuentran identificados en el sistema como son: el administrador, el editor de aplicaciones y el usuario final, al principio de cada subtema del manual se hace referencia que tipo de usuario es el indicado para acometer la tarea que se desea cumplir.

2. Descripción general del sistema y su pantalla principal.

La pantalla principal es el punto de arranque del sistema, esta pantalla permite al usuario cargar o descargar las aplicaciones hacia y desde el sistema gestor respectivamente, muestra también información sobre el estado de sistema en lo que respecta al espacio libre de almacenamiento dentro del servidor.

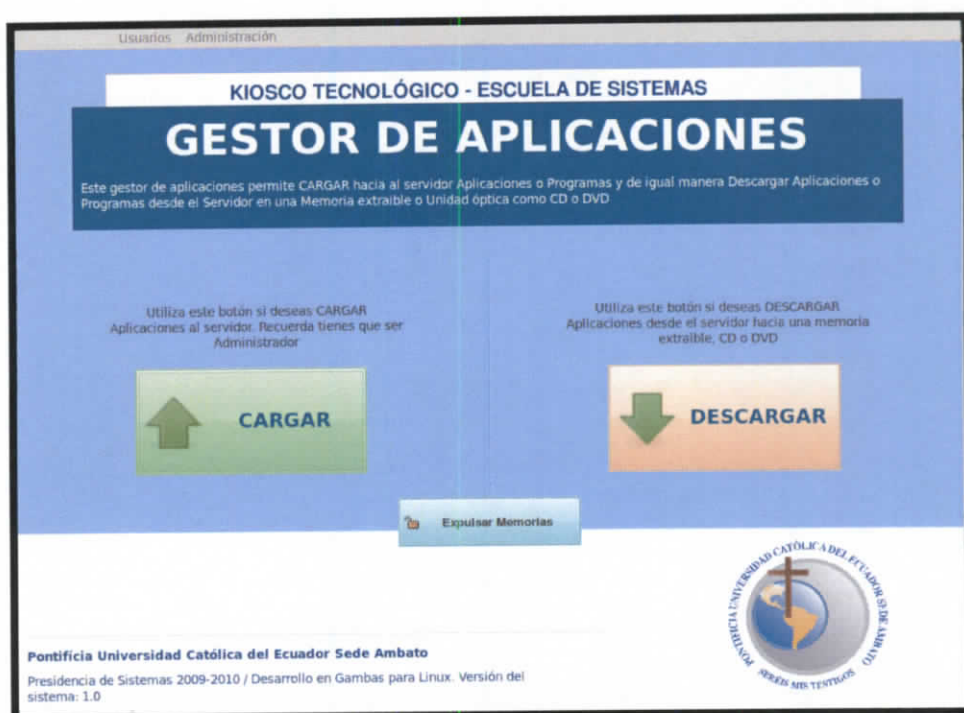


Ilustración 1: Pantalla Principal

Debido a que existe la posibilidad de cargar o descargar datos desde o hacia una unidad extraíble de almacenamiento masivo, se incluye en la parte central un botón para la expulsión segura del sistema operativo de las unidades extraíbles en mención.

Finalmente en la parte superior se encuentra el menú de uso exclusivo del usuario administrador con funciones como entrada, salida del sistema, generación de reportes, gestión de usuarios y aplicaciones y configuración de parámetros del sistema. Todas estas funciones se cubren posteriormente a detalle en el manual.

2.1. Cargando información y archivos de aplicación

Usuario: Editor de Aplicaciones.

Para cargar una aplicación en el sistema se debe presionar en la pantalla el botón “**cargar**”; una vez que se presionó el botón el sistema presentara una pantalla de Control de acceso al sistema, en la cual se debe ingresar el nombre de usuario y contraseña, en este caso el de un usuario con permiso de edición de aplicaciones.¹⁴

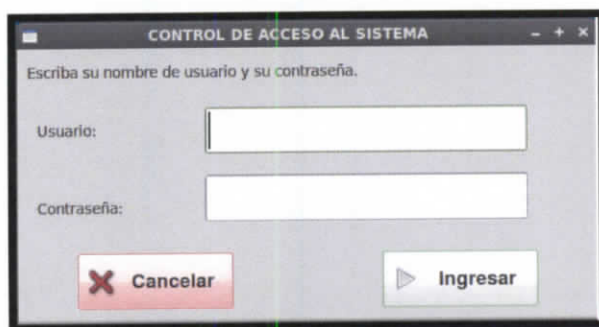
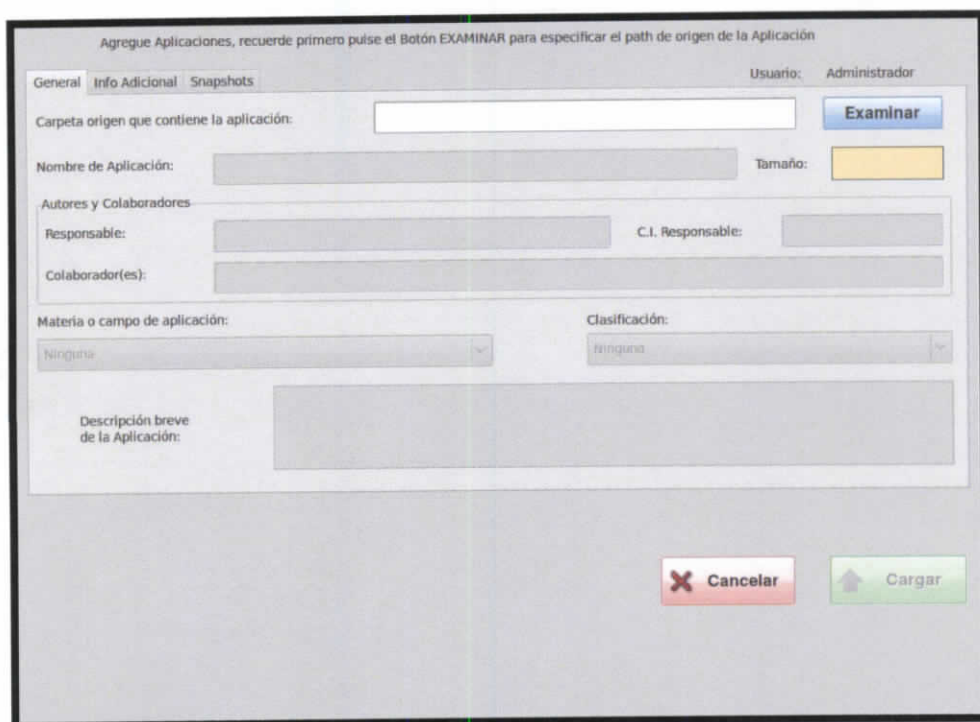


Ilustración 2: Pantalla de acceso al sistema.

Presione en la pantalla el botón “**Ingresar**” para ingresar al sistema.

¹⁴ Nota: para escribir el nombre de usuario puede hacer uso del teclado en pantalla.

Cuando el sistema gestor de aplicaciones ha identificado el usuario editor de aplicaciones presenta la pantalla de carga de aplicaciones. El primer paso a seguir es determinar la ruta de donde el sistema gestor de aplicaciones va a copiar los archivos de la aplicación que intentamos subir al sistema, para ello, presionamos en la pantalla el botón **“examinar”**, el cual nos despliega un cuadro de diálogo donde podemos escoger el nombre de la carpeta que contiene todos los archivos de la aplicación a cargarse.



The screenshot shows a web interface for adding applications. At the top, it says "Agregue Aplicaciones, recuerde primero pulse el Botón EXAMINAR para especificar el path de origen de la Aplicación". The user is logged in as "Administrador". The interface has three tabs: "General", "Info Adicional", and "Snapshots". The "General" tab is active. It contains several input fields: "Carpeta origen que contiene la aplicación:" with an empty text box and a blue "Examinar" button; "Nombre de Aplicación:" with an empty text box; "Tamaño:" with a yellow input field; "Autores y Colaboradores" section with "Responsable:" and "C.I. Responsable:" text boxes; "Colaborador(es):" with a wide text box; "Materia o campo de aplicación:" with a dropdown menu set to "Ninguna"; and "Clasificación:" with a dropdown menu set to "Ninguna". At the bottom, there is a "Descripción breve de la Aplicación:" text box. At the bottom right, there are two buttons: a red "Cancelar" button and a green "Cargar" button.

Ilustración 3: Pantalla de carga de aplicaciones

Seleccionamos la carpeta contenedora de la aplicación y presionamos **“aceptar”**.

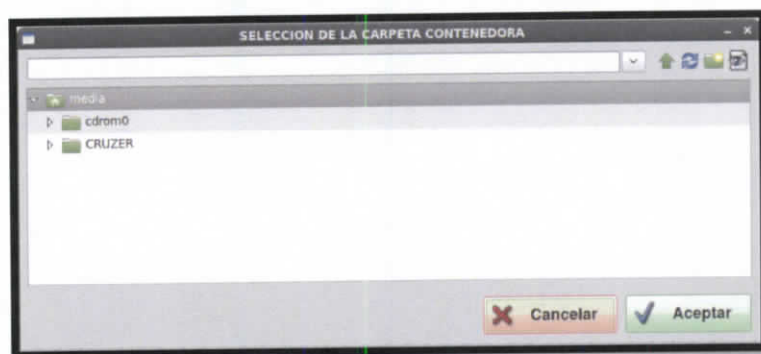


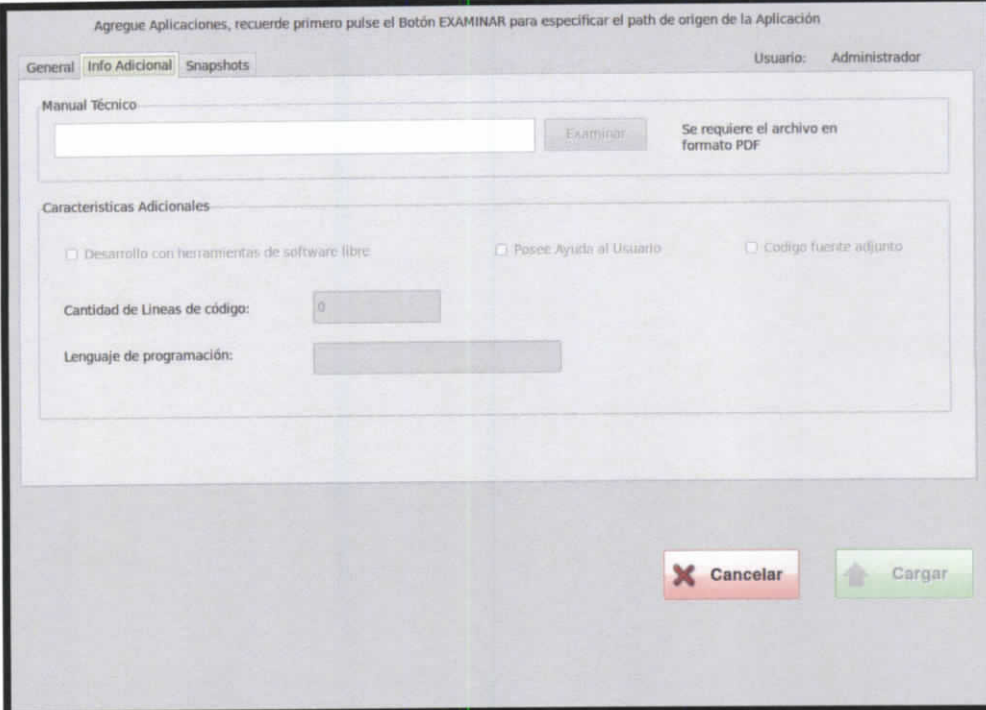
Ilustración 4: Pantalla de selección de carpeta origen de archivos

Automáticamente el sistema pasa a la pantalla de carga de aplicaciones con la ruta de carga de la aplicación y el tamaño respectivo, ahora resta asignar el nombre de la aplicación por el cual será identificada la aplicación a cargarse dentro del sistema; cuando se ha rellenado el nombre de la aplicación el sistema habilita más controles que contienen información sobre el autor(es) de la aplicación.

Es opcional determinar la materia o campo de aplicación en el que se encuadra el objetivo de la aplicación a cargarse y así mismo de manera opcional, el tipo de aplicación en los cuadros de selección “Materia o campo de aplicación” y “Clasificación” respectivamente. La información que contienen estos cuadros de selección no es modificable y son preestablecidos por el administrador. Si desea agregar nuevas materias o clasificaciones contacte al administrador y consulte el tema: *Agregando, actualizando y borrando información de clasificación al sistema.*

En la parte inferior existe un recuadro en el cual de manera opcional pero recomendada se puede agregar una descripción breve sobre la aplicación a cargarse que será de utilidad al usuario final al momento de descargarla.

En la parte superior se encuentran dos pestañas que permiten al usuario editor agregar información adicional sobre la aplicación a cargarse.



The screenshot shows a web interface for uploading applications. At the top, it says "Agregue Aplicaciones, recuerde primero pulse el Botón EXAMINAR para especificar el path de origen de la Aplicación". Below this are three tabs: "General", "Info Adicional" (which is selected), and "Snapshots". The user is identified as "Administrador".

The "Info Adicional" tab contains the following elements:

- A section titled "Manual Técnico" with a text input field and an "Examinar" button. A note next to it says "Se requiere el archivo en formato PDF".
- A section titled "Características Adicionales" with three checkboxes: "Desarrollo con herramientas de software libre", "Posee Ayuda al Usuario", and "Codigo fuente adjunto".
- Fields for "Cantidad de Líneas de código:" (with a value of 0) and "Lenguaje de programación:".
- At the bottom right, there are two buttons: "Cancelar" (with a red 'X' icon) and "Cargar" (with a green arrow icon).

Ilustración 5: Pantalla de carga de aplicaciones - Pestaña info adicional

La pestaña “**Info adicional**” permite asignar un manual de usuario que será un archivo únicamente en formato PDF, para ello presione el botón “**examinar**”, escoja el archivo PDF que contenga el manual de usuario desde una ubicación y presione el botón “**aceptar**”.

Puede detallar si la aplicación incluye ayuda al usuario, si adjunta el código fuente o si esta desarrollado con software libre, la cantidad de líneas de código y el lenguaje de programación en el cual está desarrollado la aplicación que se está cargando al sistema.

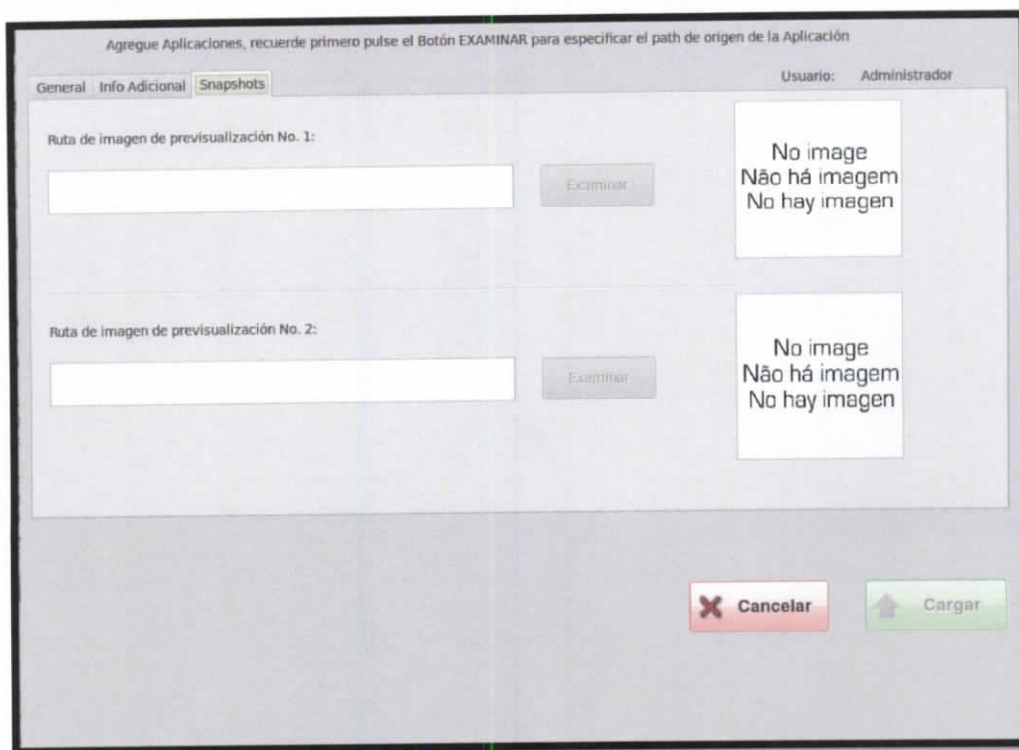


Ilustración 6: Pantalla de carga de aplicaciones - Pestaña Snapshots

La pestaña “**snapshots**” permite el usuario editor complementar a la aplicación a cargarse con dos capturas de pantalla que muestren la interfaz del sistema, logos, etc. Para seleccionar la imagen presionamos el botón “**examinar**” el cual nos muestra un cuadro de diálogo y escogemos el archivo de imagen que deseamos asociar a la aplicación. Se repite el proceso para la captura de pantalla número 2. Las imágenes pueden ser visualizadas en pantalla.¹⁵

Cuando se haya terminado de rellenar la información se presiona en pantalla el botón “**cargar**” y presionamos el cuadro de diálogo que aparece en pantalla el botón “**iniciar**” para que empiece la copia de los archivos. Este

¹⁵ Nota: se admite solo 2 imágenes con tipos de archivos PNG, JPG, JPEG y BMP; estas capturas de pantalla son opcionales.

cuadro de dialogo muestra el progreso de la copia por medio de una barra de progreso.

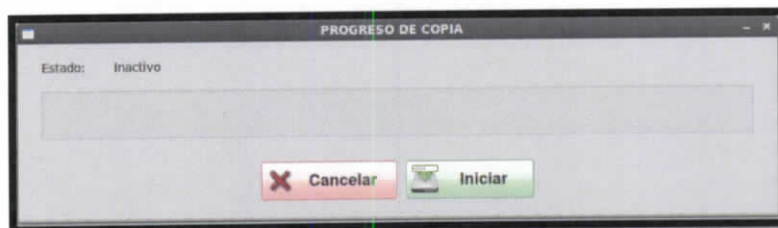


Ilustración 7: Cuadro de control de copia de archivos

El sistema le pedirá la confirmación de la carpeta origen de archivos de la aplicación, presione el botón “**si**” si está de acuerdo o “**no**” si desea cambiar la carpeta de origen de archivos de la aplicación a cargarse.

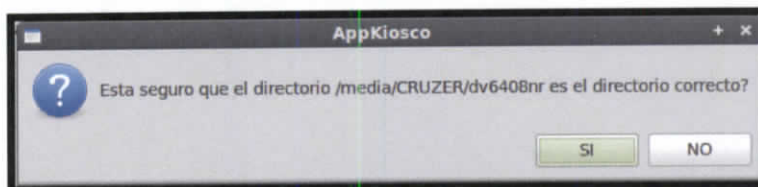


Ilustración 8: Cuadro de confirmación de directorio origen de archivos

Al momento que la copia de archivos y guardado del registro de la aplicación en el sistema haya tomado lugar correctamente el sistema gestor muestra un cuadro informativo explicando que la tarea a tenido éxito, presione “**aceptar**” lo cual hará que el sistema retorne a la pantalla de carga de información; si desea agregar otra aplicación al sistema repita el proceso, caso contrario presione el botón “**cancelar**” para regresar a la pantalla principal.¹⁶

¹⁶ Nota: una vez que haya terminado de cargar las aplicaciones dirijase al menú “autenticación” en la pantalla principal del sistema gestor y presione “desconectar” ya que si no cumple este paso puede un

2.2. Descargando aplicaciones desde el sistema.

Usuario: Usuario final

Para descargar una aplicación desde el sistema gestor hacia una memoria extraíble o un medio óptico como un CD-R o DVD-R, presione el botón “**descargar**” en la pantalla principal. El sistema ahora muestra la pantalla de descarga de aplicaciones, en donde se puede encontrar un listado de todas las aplicaciones contenidas dentro del sistema gestor de aplicaciones con su información o descripción respectiva.

Ilustración 9: Pantalla de descarga de aplicaciones

usuario final o cualquier usuario subir información al sistema sin control o validación causando problemas dentro del sistema gestor de aplicaciones.

Al presionar en pantalla sobre el nombre de cada aplicación, se muestra la información asociada a la aplicación en cuestión, incluso en la parte derecha muestra la captura de pantalla número uno si existiese.

Para ver más información detallada de la aplicación presione dos veces sobre el nombre de la aplicación simulando un *dobles clic*, de esta manera aparece un panel con mas información e incluso la captura de pantalla número dos, presione el botón **“ocultar panel”** para continuar.

The screenshot shows a software download interface for 'Drivers DV6000'. The main title is 'Drivers DV6000'. Below it, there is a section for 'Información Adicional' (Additional Information) with several fields: 'Desarrollo con herramientas de Software Libre:' (Development with Free Software tools:) with a 'NO' button; 'Posee ayuda al usuario:' (Has user help:) with a 'NO' button; 'Codigo Fuente adjunto:' (Source code attached:) with a 'NO' button; 'Manual Técnico adjunto:' (Technical manual attached:) with a 'NO' button and a 'Ver' button; 'Cantidad de Lineas de código:' (Number of lines of code:) with a text box containing '0'; and 'Coautores y Colaboradores:' (Co-authors and collaborators:) with an empty text box. To the right, there is a 'Fecha de Carga:' (Upload date:) field and a 'Captura de pantalla No.2' (Screenshot No.2) section. This section contains a placeholder image with the text 'No image Não há imagem No hay imagen' and an 'Ampliar' (Enlarge) button. Below this, there is another 'Información Adicional' section with 'Autor - Responsable:' (Author - Responsible:) set to 'HP Corporation', 'Lenguaje:' (Language:) with an empty dropdown, and 'Materia:' (Subject:) set to 'Ninguna' (None). This section also has a placeholder image with the same text and an 'Ampliar' button. At the bottom right, there are two buttons: 'Cancelar' (Cancel) and 'Descargar' (Download).

Ilustración 10: Pantalla de descarga - Información adicional detallada

En cualquiera de las dos capturas de pantalla existe un botón llamado **“ampliar”** que permite visualizar en pantalla completa la imagen, una vez que se ha visualizado la imagen presione el botón **“cerrar”** para continuar.



Ilustración 11: Pantalla de imagen completa de captura de pantalla.

Existe la posibilidad de filtrar el contenido por medio de un criterio de búsqueda, ingresando el nombre de la aplicación o por medio del lenguaje de programación al escogerlo en el cuadro de selección; si se desea quitar los filtros aplicados presione el botón **“quitar filtros”**.

Una vez que ya se definido cual es la aplicación a descargarse (se ha seleccionado la aplicación en el listado) se procede a la descarga presionando el botón **“descargar”**.

Ahora entonces, el sistema muestra un cuadro de control para escoger el medio de almacenaje en el cual se almacenarán los archivos que se pretenden obtener desde el sistema. Los dos tipos de medios admitidos son: unidades de almacenamiento masivo extraíbles y los medios ópticos CD-R y DVD-R.

2.2.1. Descargando a una Unidad CD/DVD-R

Al momento que el usuario final ha decidido que aplicación va a descargar, el sistema muestra un cuadro de control en el cual se debe presionar el botón “unidad cd/dvd” para copiar los archivos desde el servidor hacia un medio óptico.



Ilustración 12: Cuadro de control para escoger el medio de destino para copiar los archivos de la aplicación

Luego un cuadro de control aparece para que el usuario una vez que cargue el medio óptico en la bandeja del grabador de CD/DVD-R, continúe con la copia y grabación e archivos. Presiones “continuar”.

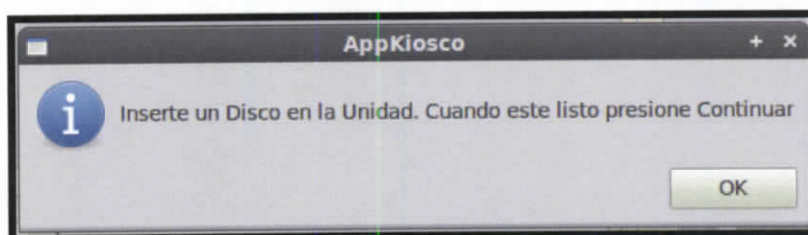


Ilustración 13: Pantalla de carga de CD o DVD

El sistema presenta una pantalla en la cual informa sobre el proceso de copia y el progreso de grabación en megabytes restantes.

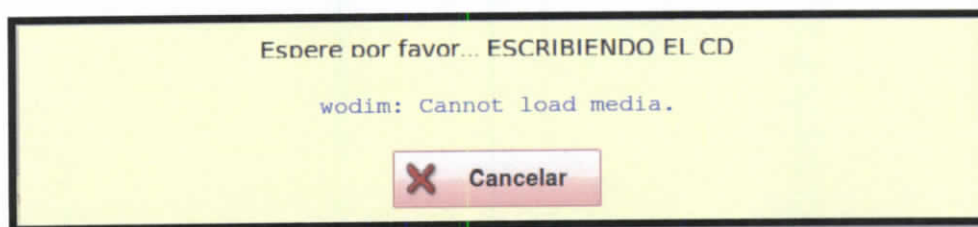


Ilustración 14: Informativo de grabación a CD o DVD

Finalmente si el proceso ha finalizado correctamente el sistema informa de la tarea cumplida satisfactoriamente. Presione **“aceptar”**, si desea puede volver a descargar mas aplicaciones o presione el botón **“cancelar”** para ir a la pantalla principal.

2.2.2. Descargando a una Unidad de almacenamiento masivo extraíble

Si se desea que la los archivos de la aplicación a descargarse sean almacenados en un medio extraíble de almacenamiento masivo entonces en el cuadro de control de descarga presione el botón **“medio extraíble”**, se mostrará un cuadro de diálogo en el cual están las medios extraíbles como es el caso de memorias USB y discos duros externos. Seleccionamos la carpeta donde queremos que se copie la aplicación y presione el botón **“copiar”**. Se pedirá al usuario que confirme el directorio destino, para lo cual puede confirmar presionando el botón **“si”** o si desea cambiar el directorio destino presione el botón **“no”**.

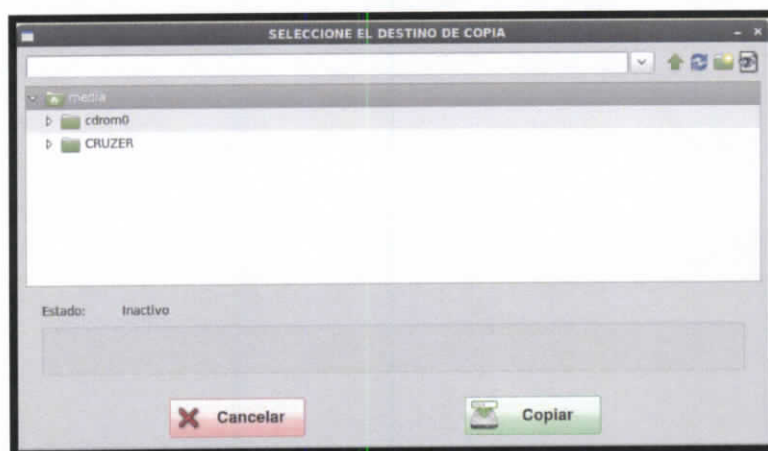


Ilustración 15: Pantalla de selección de directorio de destino y progreso de copia de archivos

Una barra de progreso muestra el porcentaje restante de la copia de archivos. Cuando el proceso a terminado se informa al usuario que culminó la tarea exitosamente. Presione el botón **“aceptar”**.

En el caso de que reciba un mensaje de error de que no hay espacio suficiente en el destino, pruebe liberando espacio en el medio extraíble primero y luego proceda a la copia de los archivos de la aplicación.¹⁷

Finalmente si el proceso ha finalizado correctamente el sistema informa de la tarea acometida satisfactoriamente. Presione **“aceptar”**, si desea puede volver a descargar mas aplicaciones o presione el botón **“cancelar”** para ir a la pantalla principal.

¹⁷ Nota: Si desea puede crear carpetas dentro de los medios extraíbles con el botón “crear carpeta” en la parte superior derecha del cuadro de dialogo, también puede refrescar los directorios y subir niveles dentro de lo que es directorios.

2.3. Agregando, actualizando y borrando información de clasificación al sistema.

Usuario: administrador

Las aplicaciones almacenadas en el sistema gestor de aplicaciones pudiesen estar clasificadas de dos maneras: por el tipo de aplicación y por la materia o campo de aplicación de la aplicación, para ello se debe preestablecer estas dos clasificaciones creando categorías en cada una de ellas siguiendo los siguientes pasos:

Se necesita ingresar al sistema como el usuario administrador, si no lo ha hecho con anterioridad puede ingresar dirigiéndose al menú superior de “**autenticación**” en la pantalla principal y presione sobre la opción “**ingresar**”. El sistema mostrará un cuadro de control para el ingreso del nombre de usuario y contraseña. Digite en el nombre de usuario la palabra: *administrador* y en la contraseña: *12345*, luego presione el botón “**ingresar**”.¹⁸

Una vez que se ha registrado el usuario administrador en el sistema diríjase al menú superior de “**administración**” y presione sobre la opción “**información del sistema**”. Se desplegará una pantalla para el ingreso y modificación de las clasificaciones dentro de la primera pestaña para

¹⁸ Nota: La contraseña del administrador “12345” es la contraseña por defecto si se ha cambiado la contraseña en algún momento, entonces la contraseña por defecto quedará obsoleta.

Categorías (tipo de aplicación o clasificación) y en la segunda pestaña para Materia (Materia o campo de aplicación de la aplicación).

2.3.1. Ingresando nuevas categorías

Para ingresar una nueva categoría en la pantalla de modificación de información del sistema presione la pestaña “**categorías**” y posteriormente el botón “**nueva**”.

Ilustración en la siguiente página.

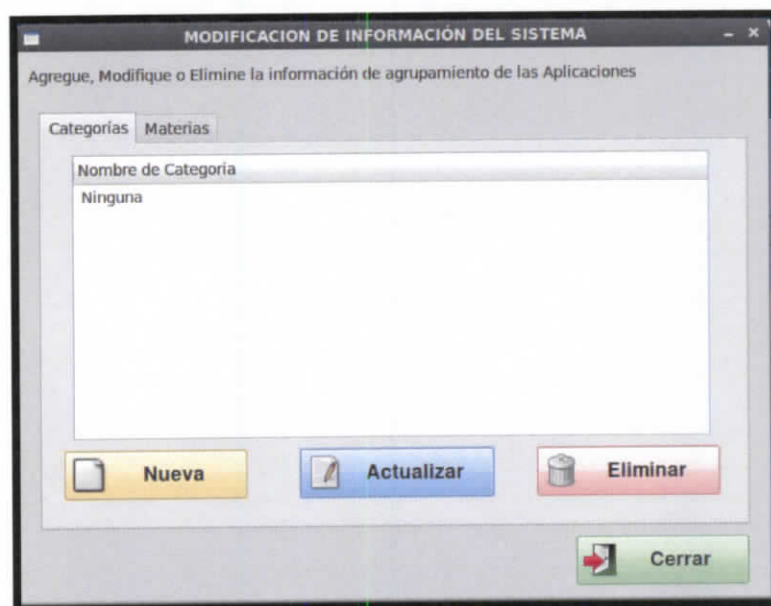


Ilustración 16: Pantalla de categorías

Un cuadro de ingreso de información se despliega en la pantalla, digite el nombre de la nueva categoría, por ejemplo: *sistemas operativos* y presione el botón “**ok**”.

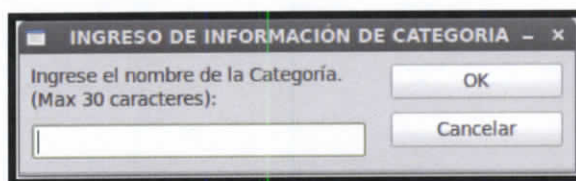


Ilustración 17: Cuadro de ingreso para la nueva categoría

Si desea cancelar el proceso de registro de información presione el botón “cancelar”.

2.3.2. Actualizando categorías

Para modificar una categoría en la pantalla de modificación de información del sistema presione la pestaña “categorías”, seleccione la categoría a modificar del listado y posteriormente presione el botón “modificar”.

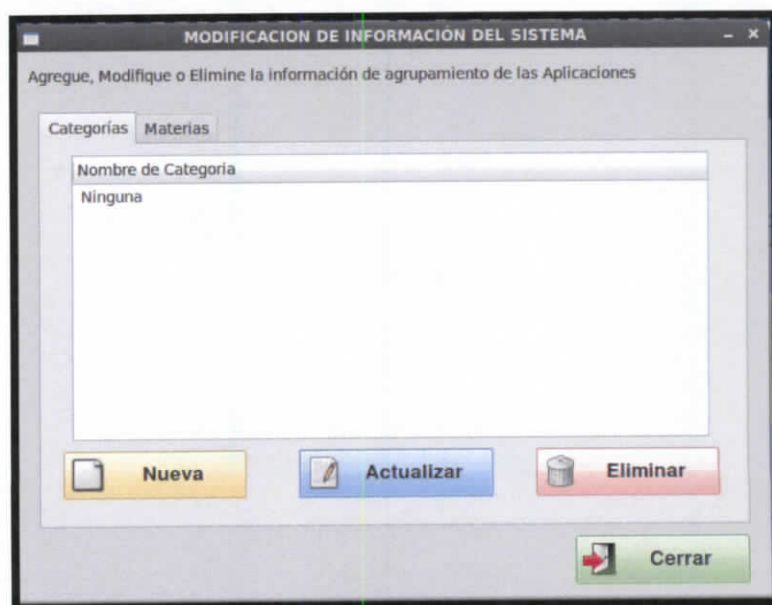


Ilustración 18: Pantalla de categorías

Un cuadro de ingreso de información se despliega en la pantalla, digite el nuevo nombre de la categoría y presione el botón “**ok**”.

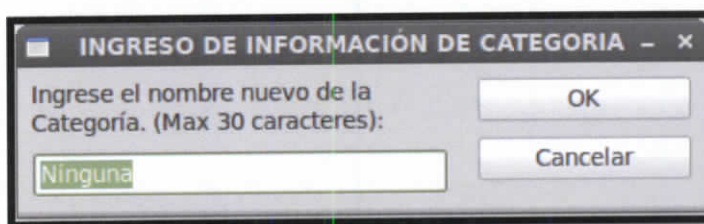


Ilustración 19: Cuadro de ingreso para modificación de categorías

Si desea cancelar el proceso de modificación de información presione el botón “**cancelar**”.

2.3.3. Eliminando categorías

Para borrar o eliminar una categoría en la pantalla de modificación de información del sistema presione la pestaña “**categorías**”, seleccione la categoría a borrar o eliminar del listado y posteriormente presione el botón “**eliminar**”.

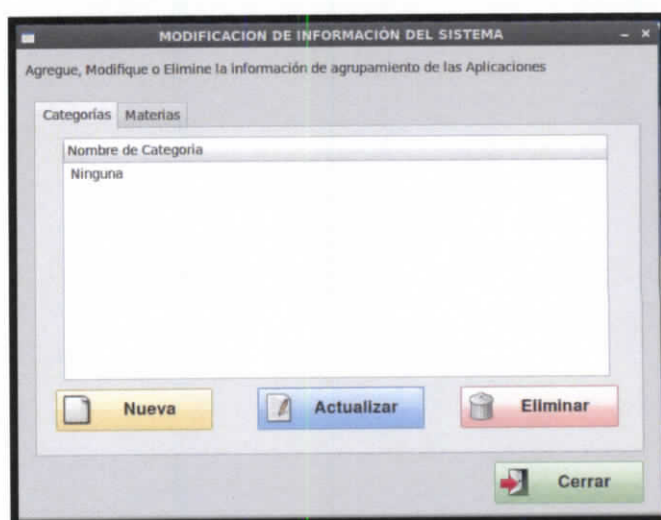


Ilustración 20: Pantalla de categorías

Un cuadro de control se despliega en la pantalla, confirme la eliminación categoría presionando el botón **“borrar”**. El sistema informa la eliminación. Presione **“aceptar”**.

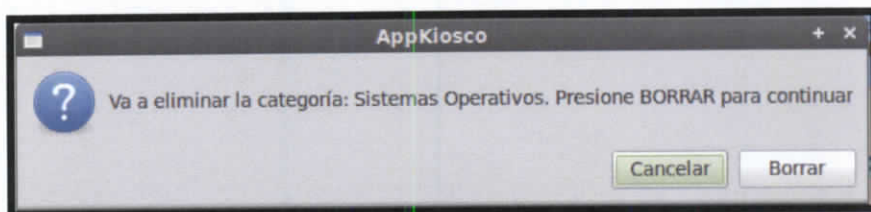


Ilustración 21: Cuadro de control de eliminación de categorías

Si no desea eliminar la categoría presione **“cancelar”**.

2.3.4. Ingresando nuevas materias

Para ingresar una nueva materia en la pantalla de modificación de información del sistema presione la pestaña **“materias”** y posteriormente el botón **“nueva”**.

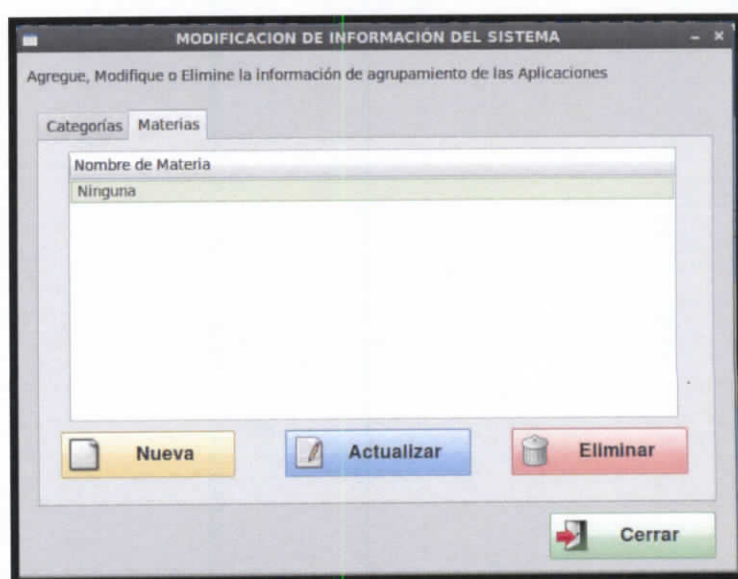


Ilustración 22: Pantalla sobre Materias

Un cuadro de ingreso de información se despliega en la pantalla, digite el nombre de la nueva materia, por ejemplo: *simulación* y presione el botón “ok”.

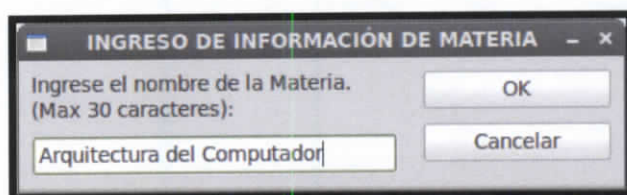


Ilustración 23: Cuadro de ingreso de nueva materia

Si desea cancelar el proceso de registro de información presione el botón “cancelar”.

2.3.5. Actualizando materias

Para modificar una materia en la pantalla de modificación de información del sistema presione la pestaña “materias”, seleccione la materia a modificar del listado y posteriormente presione el botón “modificar”.

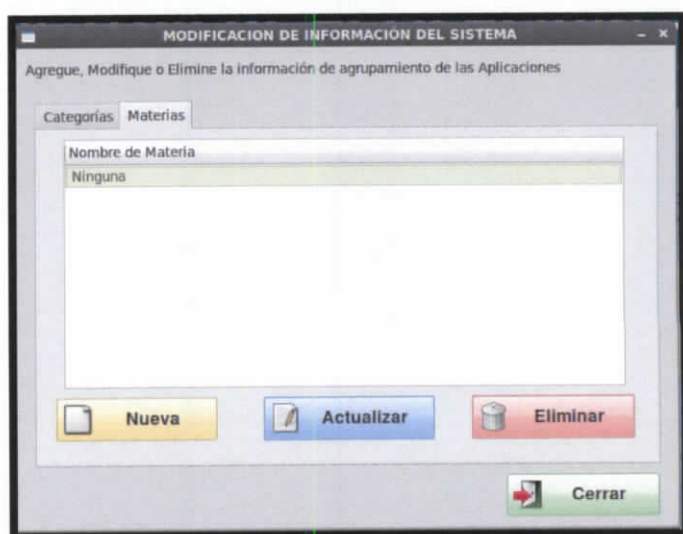


Ilustración 24: Pantalla sobre Materias

Un cuadro de ingreso de información se despliega en la pantalla, digite el nuevo nombre de la materia y presione el botón **“ok”**.

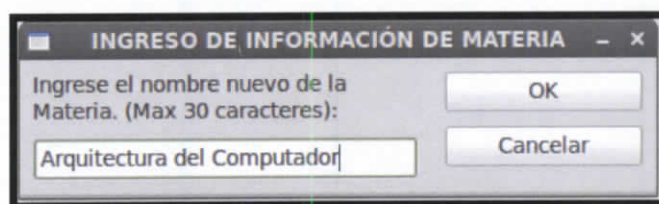


Ilustración 25: Cuadro de ingreso para modificación de materias

Si desea cancelar el proceso de modificación de información presione el botón **“cancelar”**.

2.3.6. Eliminando materias

Para borrar o eliminar una materia en la pantalla de modificación de información del sistema presione la pestaña **“materias”**, seleccione la materia a borrar o eliminar del listado y posteriormente presione el botón **“eliminar”**.

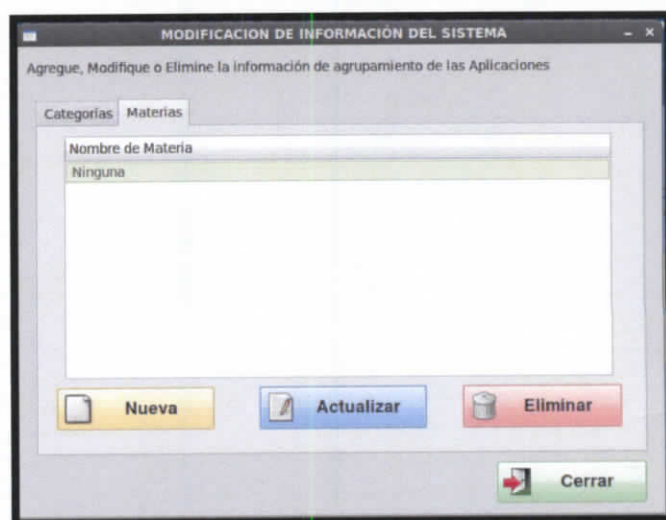


Ilustración 26: Pantalla sobre Materias

Un cuadro de control se despliega en la pantalla, confirme la eliminación de la materia presionando el botón **“borrar”**. El sistema informa la eliminación. Presione **“aceptar”**.



Ilustración 27: Cuadro de control de eliminación de materias

Si no desea eliminar la materia presione **“cancelar”**.

2.4. Agregando, modificando y eliminando usuarios

Usuario: administrador

Dentro del sistema operan 3 tipos de usuarios, cada uno cumple su rol específico, estos son: el administrador, los editores de aplicaciones y el usuario final.

El administrador es un tipo de usuario único que tiene los privilegios más elevados dentro del sistema, se lo identifica en el sistema por el nombre administrador. No existe más que un administrador. El administrador puede asumir todos los roles dentro del sistema y trabajar con todas las opciones disponibles dentro del sistema gestor de aplicaciones.

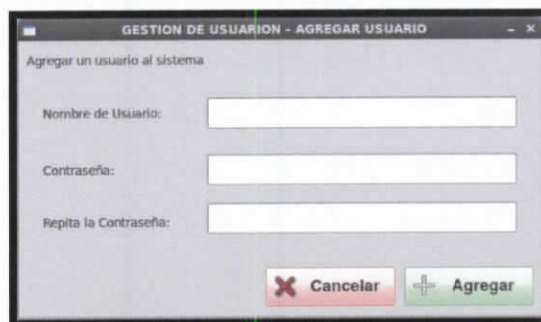
El siguiente tipo de usuario es el editor de aplicaciones, tiene privilegios para administrar las aplicaciones dentro del sistema, es decir carga, modificación y eliminación de aplicaciones. Puede asumir el rol de usuario final y puede existir 1 o más usuarios de este tipo.

Por último con el nivel de menos privilegios se encuentra el usuario final, este es un usuario único que solo puede descargar las aplicaciones desde el sistema gestor de aplicaciones.

Para realizar cambios en los usuarios del sistema se necesita ingresar al sistema como el usuario administrador, si no lo ha hecho con anterioridad puede ingresar dirigiéndose al menú superior de “**autenticación**” en la pantalla principal y presione sobre la opción “**ingresar**”. El sistema mostrará un cuadro de control para el ingreso del nombre de usuario y contraseña. Digite en el nombre de usuario la palabra: *administrador* y en la contraseña: *12345*, luego presione el botón “**ingresar**”.

2.4.1. Agregando usuarios

Diríjase al menú superior de “**usuarios**”, presione la opción “**crear un nuevo usuario**” y se despliega una pantalla de gestión de usuarios.



GESTION DE USUARIOS - AGREGAR USUARIO

Agregar un usuario al sistema

Nombre de Usuario:

Contraseña:

Repita la Contraseña:

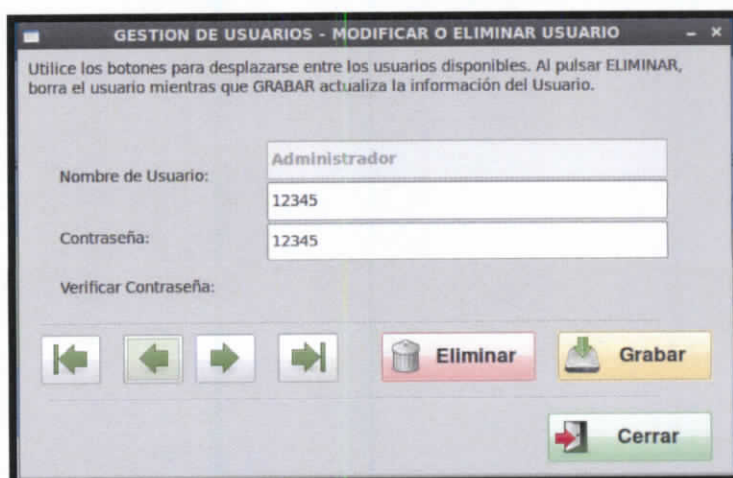
Ilustración 28: Cuadro de diálogo para crear usuarios

Escriba el nombre de usuario y su contraseña; la contraseña por seguridad debe digitarse dos veces.

Posteriormente presione el botón **“agregar”**, el sistema confirmará la inserción del nuevo usuario. Presione **“aceptar”**.

2.4.2. Modificando usuarios

Diríjase al menú superior de **“usuarios”**, presione la opción **“modificar o eliminar usuarios”** y se despliega una pantalla de gestión de usuarios.



GESTION DE USUARIOS - MODIFICAR O ELIMINAR USUARIO

Utilice los botones para desplazarse entre los usuarios disponibles. Al pulsar ELIMINAR, borra el usuario mientras que GRABAR actualiza la información del Usuario.

Nombre de Usuario:

Contraseña:

Verificar Contraseña:

Ilustración 29: Pantalla de gestión de usuarios

Por medio de los botones de navegación (botones con flechas de adelantamiento, retroceso, principio y final) puede buscar el usuario a modificar. Una vez que ha identificado el usuario a modificar, escriba la contraseña nueva. Para guardar los cambios presione el botón **“grabar”**.¹⁹

2.4.3. Eliminando usuarios

Diríjase al menú superior de **“usuarios”**, presione la opción **“modificar o eliminar usuarios”** y se despliega una pantalla de gestión de usuarios.

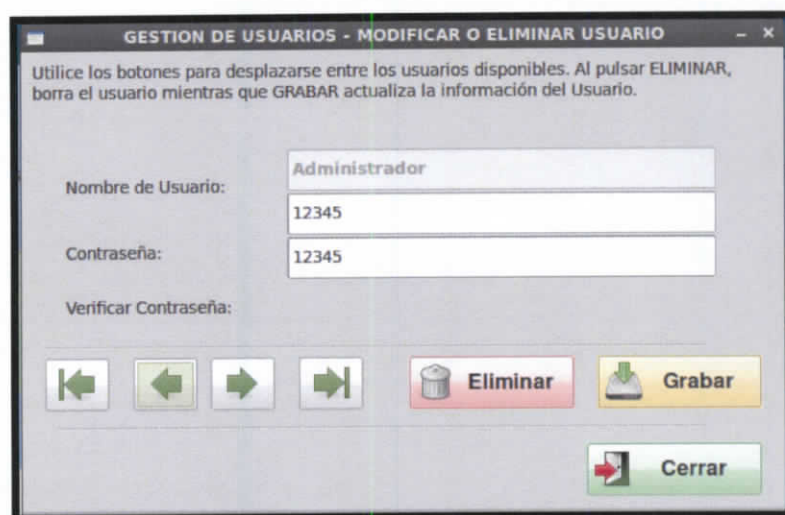


Ilustración 30: Pantalla de gestión de usuarios

Por medio de los botones de navegación (botones con flechas de adelantamiento, retroceso, principio y final) puede buscar el usuario a

¹⁹ Nota: El usuario administrador y el resto de usuarios solo puede cambiar su contraseña, mientras que el usuario final es un usuario por defecto en el sistema inaccesible incluso para el administrador.

eliminar. Una vez que ha identificado el usuario a eliminar presione el botón **“eliminar”**.

El sistema le preguntará si está **seguro** de eliminar el usuario, si es así, presione el botón **“si”**, caso contrario presione el botón **“cancelar”**.²⁰

2.5. Asignando roles a los usuarios

Usuario: administrador

Para modificar el rol que tienen los usuarios dentro del sistema diríjase al menú superior de **“usuarios”** y presione la opción **“permisos de usuario”**.

Para realizar cambios en los permisos de usuario del sistema se necesita ingresar al sistema como el usuario administrador, si no lo ha hecho con anterioridad puede ingresar dirigiéndose al menú superior de **“autenticación”** en la pantalla principal y presione sobre la opción **“ingresar”**. El sistema mostrará un cuadro de control para el ingreso del nombre de usuario y contraseña. Digite en el nombre de usuario la palabra: *administrador* y en la contraseña: *12345*, luego presione el botón **“ingresar”**.

²⁰ Nota: El usuario administrador no puede ser eliminado del sistema.

Ver ilustración en la página siguiente.

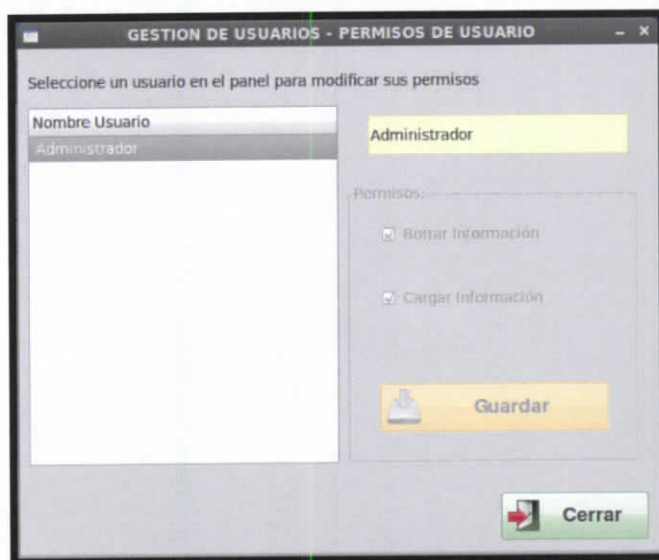


Ilustración 31: Pantalla de gestión de roles de usuario

Al desplegarse la pantalla de permisos de usuario, se puede visualizar todos los usuarios dentro del sistema excepto el usuario final, si desea modificar los permisos de un usuario, primero debe seleccionar el usuario de la lista de usuarios, segundo active o desactive los permisos para el usuario, en este caso pueden ser de *borrar información* o *cargar información*. Presione el botón “guardar” para guardar los cambios realizado a ese usuario.²¹

Al mensaje de confirmación de cambios presionamos el botón “aceptar”.

2.6. Gestionando las aplicaciones.

Usuario: editor de aplicaciones

²¹ Nota: el usuario administrador y usuario final no pueden ser modificados a nivel de sus permisos. Para cambiar los permisos de otro usuario primero aplaste el botón “guardar” antes de cambiar de usuario, caso contrario no se reflejarán los cambios en el sistema.

El usuario editor de aplicaciones puede administrar las aplicaciones almacenadas en el sistema en cuanto a lo que modificación respecta, esto incluye la modificación de la información asociada a la aplicación, sus archivos adjuntos como manual de usuario y las imágenes de captura de pantalla, mas no sus archivos. Y, por otra parte la eliminación de toda la aplicación y sus archivos del sistema gestor de aplicaciones.

Para realizar cambios en las aplicaciones del sistema se necesita ingresar al sistema como el usuario editor de aplicaciones, si no lo ha hecho con anterioridad puede ingresar dirigiéndose al menú superior de “**autenticación**” en la pantalla principal y presione sobre la opción “**ingresar**”. El sistema mostrará un cuadro de control para el ingreso del nombre de usuario y contraseña, luego presione el botón “**ingresar**”.²²

2.6.1. Modificando una aplicación almacenada

En el menú superior de “**administración**” en la pantalla principal escoja la opción “**gestionar aplicaciones**”, automáticamente se presentará la pantalla de gestión de aplicaciones con el listado de todas las aplicaciones almacenadas en el sistema.

²² Nota: Recuerde que el administrador puede tomar el rol de editor de aplicaciones.

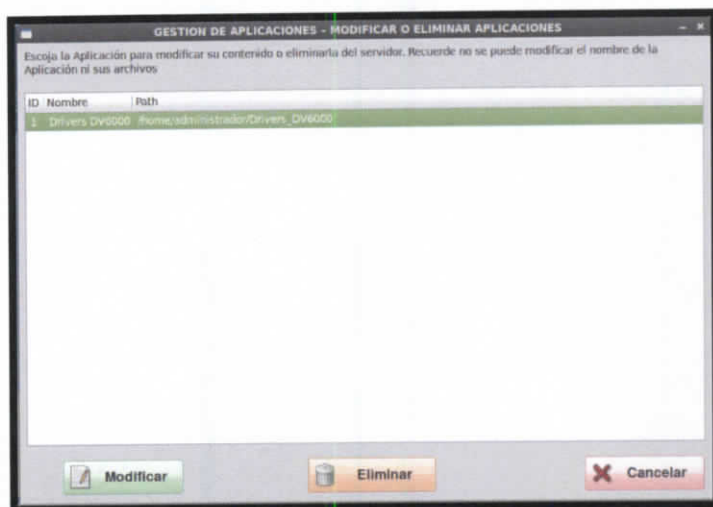


Ilustración 32: Pantalla de gestión de aplicaciones

Escoja la aplicación a modificar del listado para seguidamente presionar el botón **“modificar”**.

Se presentará una pantalla muy similar a la pantalla de carga de aplicaciones, en la cual podrá modificar toda la información de igual manera que cuando se carga una aplicación. Véase *Cargando información y aplicaciones al sistema*.

Cuando haya terminado de modificar la información presione el botón **“guardar”**.

2.6.2. Eliminando una aplicación almacenada

En el menú superior de **“administración”** en la pantalla principal escoja la opción **“gestionar aplicaciones”**, automáticamente se presentará la pantalla de gestión de aplicaciones con el listado de todas las aplicaciones almacenadas en el sistema.

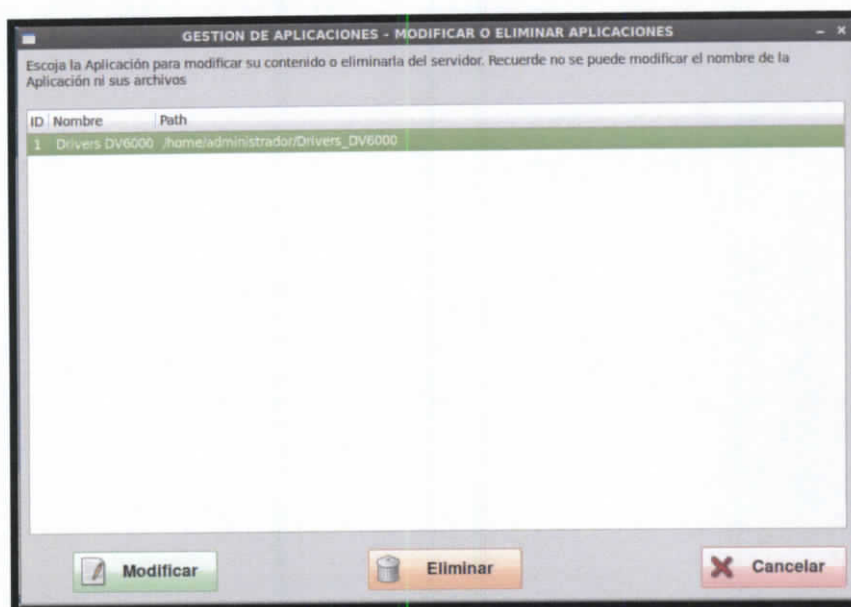


Ilustración 33: Pantalla de gestión de aplicaciones

Escoja la aplicación a eliminar del listado para seguidamente presionar el botón **“eliminar”**.

El sistema presentará un cuadro de control preguntando sobre la eliminación de los archivos y el registro de la aplicación, presione **“si”** para continuar y **“cancelar”** para detener el borrado de la aplicación.²³

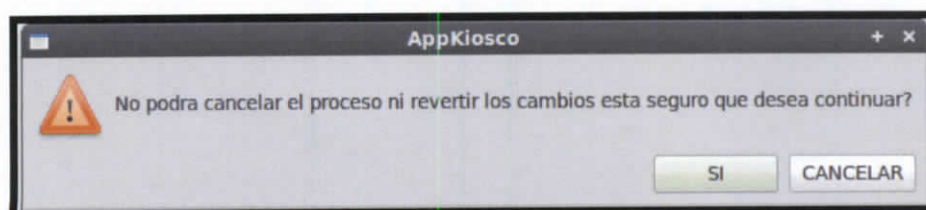


Ilustración 34: Cuadro de control de eliminación de aplicaciones

²³ Nota: Cuando modifica una aplicación no puede modificar los archivos, si existe un cambio en los archivos elimine la aplicación y cargue la aplicación nuevamente.

2.7. Presentando pantallas de informes.

Usuario: administrador, editor de aplicaciones

Existen 2 tipos de reporte que pueden ser presentados al usuario administrador y editor de aplicaciones. Las aplicaciones almacenadas (editor de aplicaciones y administrador), y los eventos suscitados en el sistema (únicamente al usuario administrador).

Para visualizar los informes en pantalla se necesita ingresar al sistema como el usuario administrador o editor de aplicaciones, si no lo ha hecho con anterioridad puede ingresar dirigiéndose al menú superior de **“autenticación”** en la pantalla principal y presione sobre la opción **“ingresar”**. El sistema mostrará un cuadro de control para el ingreso del nombre de usuario y contraseña. En el caso del administrador digite en el nombre de usuario la palabra: *administrador* y en la contraseña: *12345*, luego presione el botón **“ingresar”**.

2.7.1. Informe visual sobre aplicaciones almacenadas.

Para emitir un informe en pantalla sobre las aplicaciones almacenadas diríjase al menú superior de **“administración”** en la pantalla principal y busqué la opción de **“gestionar aplicaciones”**. Aparecerá un listado con las

aplicaciones almacenadas con la respectiva información anexa o referencial de cada una de ellas.

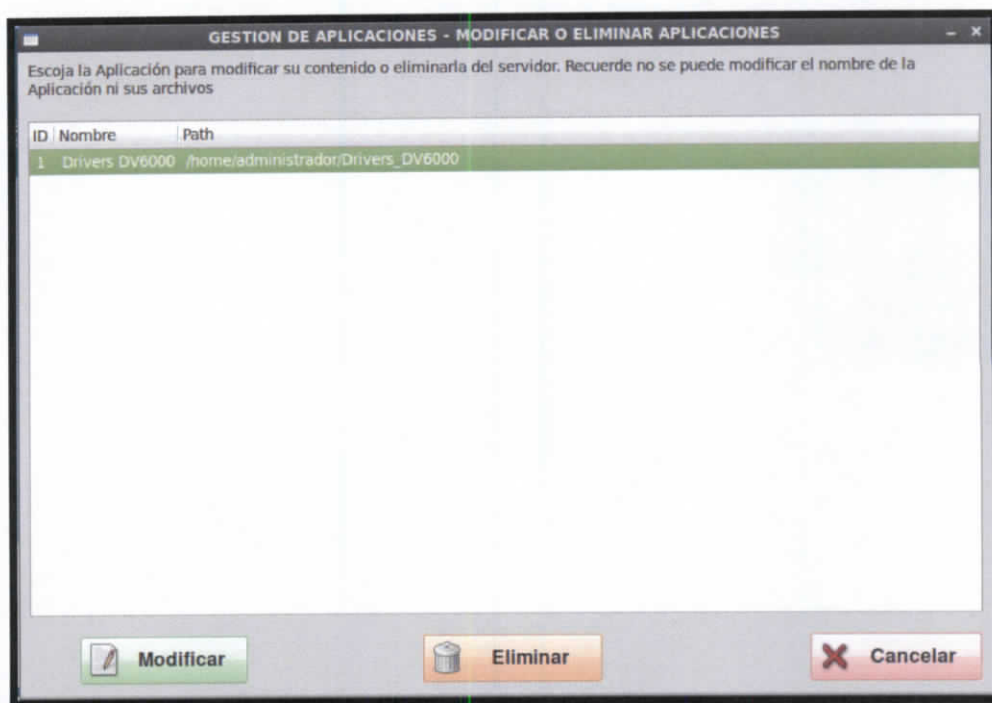


Ilustración 35: Pantalla de gestión de aplicaciones

Cuando haya terminado de visualizar el informe presione el botón “cancelar”.

2.7.2. Informe visual sobre los eventos del sistema.

Para emitir un informe en pantalla sobre los eventos suscitados en el sistema diríjase al menú superior de “administración” en la pantalla principal y busqué la opción de “registro de eventos”. Aparecerá un listado

con los diferentes eventos que han tomado lugar en el sistema, tanto a lo que respecta a las aplicaciones como ingresos al sistema.

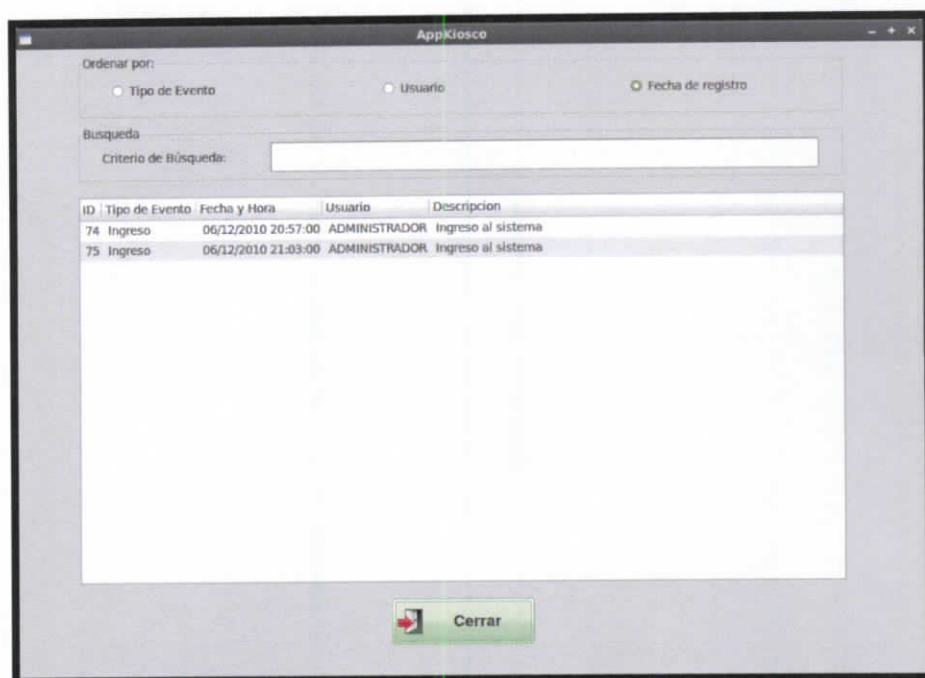


Ilustración 36: Pantalla de reporte de eventos

En la parte superior existe un panel de filtros y ordenamiento puede incluso incluir un criterio de búsqueda para filtrar el contenido de la lista.

Cuando haya terminado de visualizar el informe presione el botón “**cerrar**”.

2.8. Cambiando parámetros de configuración del sistema.

Usuario: administrador.

El sistema incluye algunas configuraciones que pueden ser parametrizadas por el usuario administrador, esta configuración del sistema debe ser

realizada con mucha precaución ya que, si existe algún parámetro mal configurado podría ocasionar inestabilidad y mal funcionamiento de la aplicación.

Para realizar estos cambios en el sistema despliegue el panel de configuración presionando en el menú superior de “administración” de la pantalla principal y posteriormente la presione la opción “variables del sistema”.

El panel de configuración cuenta con 4 opciones que son: almacenamiento y directorios, configuración de unidades ópticas, base de datos y módulos.

Para realizar cambios en el panel de configuración del sistema se necesita ingresar al sistema como el usuario administrador, si no lo ha hecho con anterioridad puede ingresar dirigiéndose al menú superior de “autenticación” en la pantalla principal y presione sobre la opción “ingresar”. El sistema mostrará un cuadro de control para el ingreso del nombre de usuario y contraseña. Digite en el nombre de usuario la palabra: *administrador* y en la contraseña: *12345*, luego presione el botón “ingresar”.

2.8.1. Configurando almacenamiento y directorios.

En el panel de configuración despliegue la opción “**almacenamiento y directorios**”, aquí encontrará 3 cuadros de texto que hacen referencia al

path de almacenamiento principal, el path de almacenamiento secundario y el path de la aplicación.

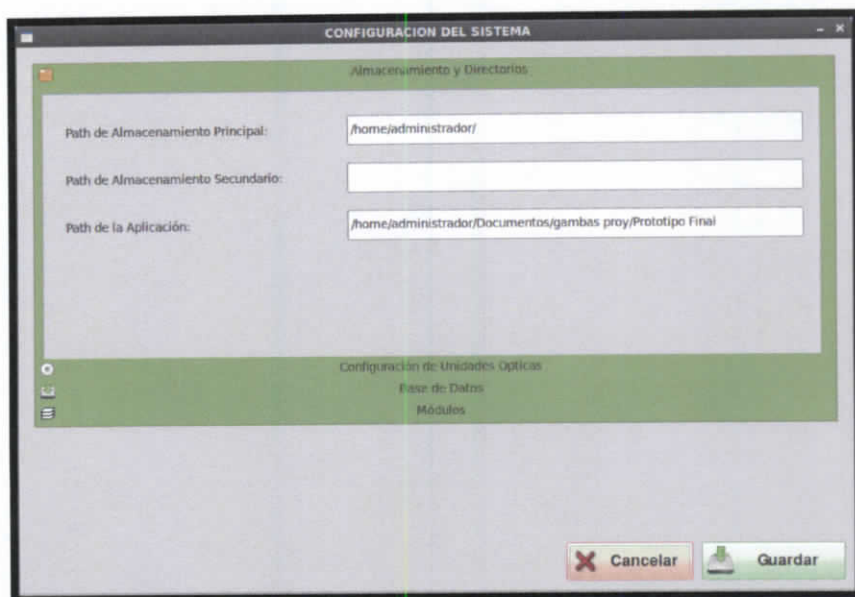


Ilustración 37: Panel de configuración de directorios

El *path de almacenamiento principal* es el lugar donde se almacenan los archivos de las aplicaciones que se cargan dentro del sistema por defecto.

El *path de almacenamiento secundario* es el lugar donde se almacenan los archivos de las aplicaciones que se cargan dentro del sistema cuando se ha excedido el límite de espacio en el path de almacenamiento principal.

El path de la aplicación es un cuadro de texto de solo lectura que informa en qué lugar está almacenado y ejecutándose el sistema gestor de aplicaciones.

Cuando termine de configurar la aplicación presione el botón “**guardar**”.

2.8.2. Configurando las unidades ópticas.

En el panel de configuración despliegue la opción “**configuración de unidades ópticas**”, aquí encontrará 4 cuadros de texto que hacen referencia a la unidad de grabación de DVD, la velocidad de escritura de la unidad DVD, la grabadora de CD y la velocidad de escritura de la unidad CD.

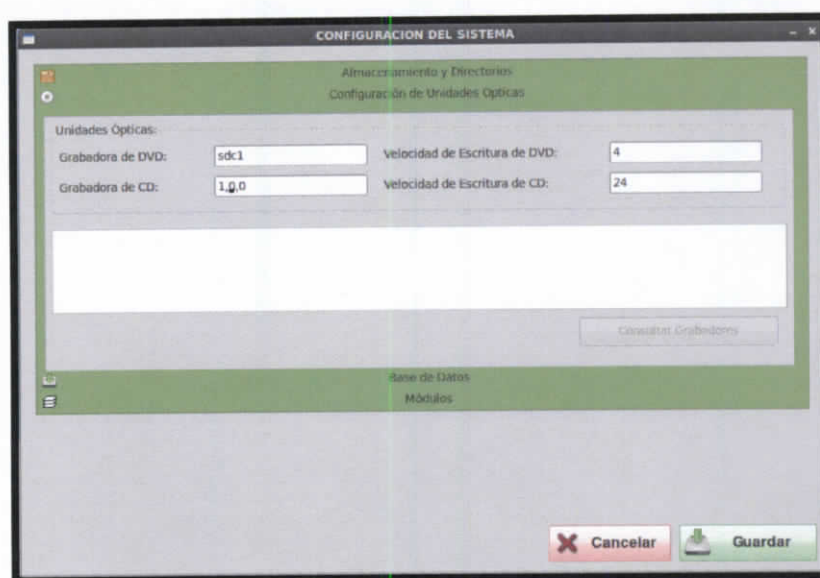


Ilustración 38: Panel de configuración de unidades ópticas

Estos parámetros que definen a los grabadores pueden ser consultados en base a comandos en la consola de Linux y la velocidad de escritura dependerá de la velocidad mínima y máxima que soporta cada unidad de escritura.

Cuando termine de configurar la aplicación presione el botón “**guardar**”.

2.8.3. Configurando la base de datos.

En el panel de configuración despliegue la opción **“base de datos”**, aquí encontrará 2 cuadros de texto que hacen referencia al nombre de usuario y la contraseña que concretamente respecta a la conexión de la aplicación con la base de datos.

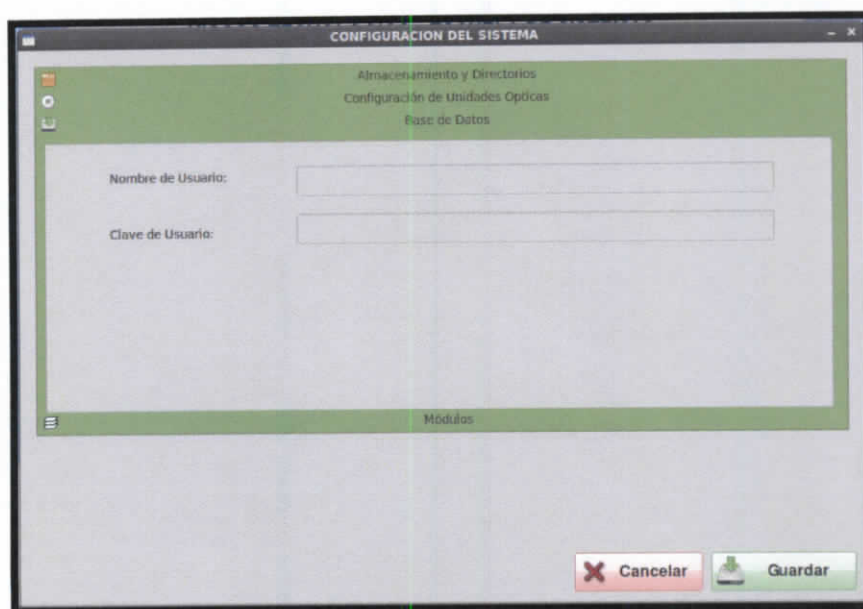


Ilustración 39: Panel de configuración de base de datos

Cuando termine de configurar la aplicación presione el botón **“guardar”**.²⁴

2.8.4. Configurando módulos y opciones del sistema.

En el panel de configuración despliegue la opción **“módulos y opciones”**, aquí encontrará 2 opciones disponibles con valor de verdad.

²⁴ NOTA: Si los cuadros de texto se encuentran deshabilitados es debido a que la configuración se la realiza directamente a través del código fuente. Consulte al administrador.

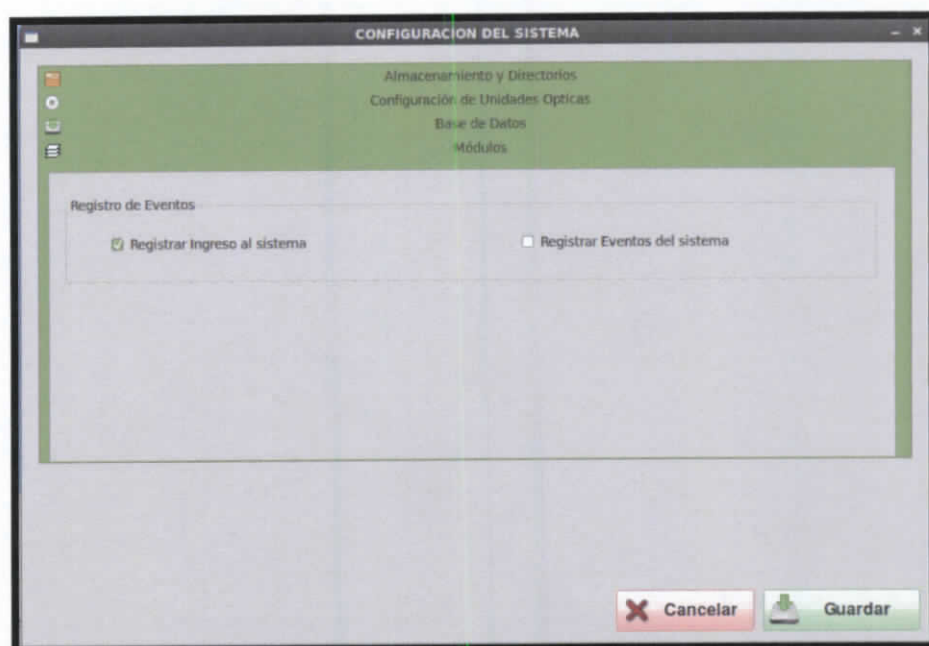


Ilustración 40: Panel de configuración de módulos y opciones

Registrar ingreso al sistema, indica si se guardará o no en el registro de eventos lo que respecta el ingreso de usuarios al sistema.

Registrar eventos del sistema, indica si se guardará o no en el registro de eventos lo que respecta a los eventos propios del sistema, como por ejemplo cambio de usuarios o aplicaciones, etc.

Cuando termine de configurar la aplicación presione el botón **“guardar”**.

2.9. Expulsando unidades extraíbles.

Si ha conectado una unidad de almacenamiento masivo USB al sistema, puede expulsarla de manera segura presionando en el botón “expulsar memorias” en la pantalla principal del sistema gestor de aplicaciones.

Al desplegarse la nueva ventana contendrá los nombres de todas las unidades extraíbles USB posibles para extraer, seleccione el dispositivo a expulsar y presione el botón “**expulsar**”. Un mensaje de confirmación aparecerá presione “**aceptar**”.

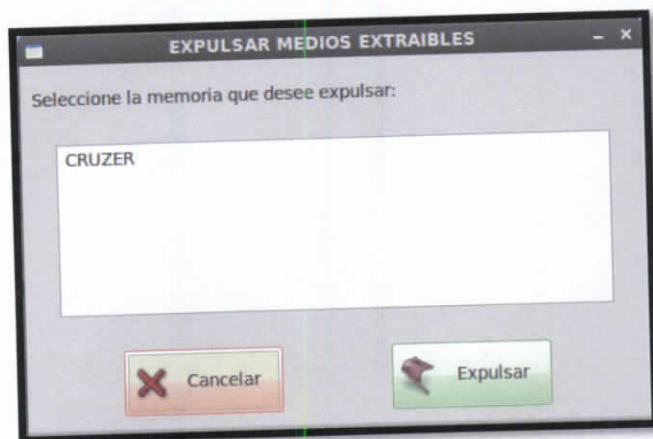


Ilustración 41: Pantalla de expulsión de medios extraíbles

3. Notas al final

- a. La base de datos no es accesible para el usuario administrador a través del programa del sistema gestor de base de datos, si se desea ingresar a la base utilice un navegador para MySQL.
- b. Los mensajes en otros idiomas se debe a que existen aplicaciones privativas de terceras partes funcionando a la par con el sistema tal es el caso de cdrecord® por ejemplo que ayuda a la grabación de CDs.
- c. La aplicación no posee ayuda en pantalla.

Anexo B: Implantación del sistema.

Instalación del software.

El funcionamiento de la aplicación depende de que el programador Gambas este instalado en el sistema operativo en el cual está alojado el Sistema Gestor de Aplicaciones, por ello, debemos instalar la interfaz de programación Gambas y las dependencias de la misma.

A continuación se detalla paso a paso la instalación de Gambas en el sistema operativo Linux Ubuntu Karmic Koala 9.10. Se debe estar conectado a internet.

- a) Descargar el paquete de archivos de Gambas desde el sitio web oficial: <http://gambas.sourceforge.net/es/main.html>. Preferentemente con la última versión estable disponible.
- b) Para Ubuntu 9.10 descargar las dependencias de Gambas con la siguiente instrucción, recuerde que se debe ejecutar esta instrucción en la consola del sistema operativo linux.

```
sudo apt-get install build-essential autoconf libbz2-dev
libfbclient2 libmysqlclient15-dev unixodbc-dev libpq-dev
libsqlite0-dev libsqlite3-dev libgtk2.0-dev libldap2-dev
libcurl4-gnutls-dev libgtkglext1-dev libpcre3-dev libsdl-
sound1.2-dev libsdl-mixer1.2-dev libsdl-image1.2-dev libsage-
dev libxml2-dev libxslt1-dev libbonobo2-dev libcoss4-dev
libomniorb4-dev librsvg2-dev libpoppler-dev libpoppler-glib-
dev libasound2-dev libesd0-dev libesd-alsa0 libdirectfb-dev
libaal-dev libxtst-dev libffi-dev kdelibs4-dev firebird2.1-
dev libqt4-dev libglew1.5-dev libimlib2-dev libv4l-dev
libsdl-ttf2.0-dev
```

- c) Desempaquetar el paquete previamente descargado de Gambas al directorio `/usr/src`, por medio del comando de consola:

```
$ cd /usr/src
$ sudo tar xvfj /home//Desktop/gambas2-1.48.tar.bz2
```

- d) Desplazarse a la carpeta que contiene los archivos fuente de Gambas y configurarlo e instalar con la siguiente instrucción de consola:

```
$ cd /usr/src/gambas2-1.9.48
$ sudo ./configure -C
$ sudo make
$ sudo make install
```

Una vez que se ha instalado el programador Gambas resta copiar los archivos de la aplicación a una carpeta dentro del sistema operativo donde crea pertinente el administrador del sistema.

Finalmente para la base de datos se siguen los siguientes pasos

- a) Instalar el servidor de MySQL con la siguiente instrucción de consola, esta instrucción se ejecuta en el terminal o consola del sistema operativo Linux y obtiene de un repositorio los instaladores de MySQL server versión 5, los cuales serán instalados de forma automática.

```
$sudo apt-get install mysql-server
```

- b) Especificamos un nombre de usuario para MySQL y su contraseña.

- c) Conectamos a MySQL y ejecutamos el script de la base de datos del sistema, este script contiene todas las estructuras necesarias para que funcione correctamente la aplicación. Recuerde que esto debe hacerlo el administrador del sistema con conocimiento de lenguaje SQL para MySQL.



CERTIFICADO DE VALIDACIÓN

Por medio de la presente en mi calidad de DIRECTOR DE LA ESCUELA DE INGENIERÍA DE SISTEMAS y luego de revisar la aplicación de la Disertación previa la obtención del título de Ingeniero de Sistemas titulada "Desarrollo de una aplicación de Administración para el Kiosko Tecnológico de la Escuela de Ingeniería de Sistemas de la PUCESA utilizando Software Libre" del señor: David Oswaldo Marañón Barrera, certifico que se ha concluido el trabajo en el desarrollo de la disertación satisfactoriamente, cumpliendo con los objetivos planteados.

Se destacan en el trabajo: la innovación tecnológica. El desarrollo integrando tecnologías hardware y software en plataformas libres y la aplicación práctica en una solución integral que brinda servicio a la comunidad universitaria; características que han permitido que el mencionado proyecto represente destacadamente a la PUCE Sede Ambato en eventos de Ciencia y Tecnología a nivel nacional.

Es todo cuanto puedo certificar en honor a la verdad. El interesado puede hacer uso del presente documento en forma que estime conveniente.

Ambato, 01 de diciembre de 2010


Ing. Santiago Acurio M.

DIRECTOR DE LA ESCUELA DE SISTEMAS
PUCESA



Ambato diciembre 03, 2010.

Ingeniero MSc.

Santiago Acurio Maldonado

DIRECTOR

Escuela de Ingeniería en Sistemas

PUCESA

Presente

De mi consideración:

Me permito informar a usted, que luego de haber realizado la revisión y la validación de la Disertación de Grado titulada: **“DESARROLLO DE UNA APLICACIÓN DE ADMINISTRACIÓN PARA EL KIOSKO TECNOLÓGICO DE LA ESCUELA DE INGENIERÍA DE SISTEMAS DE LA PUCESA UTILIZANDO SOFTWARE LIBRE”**, propuesta por el señor David Oswaldo Marañón Barrera, ha cumplido con los objetivos planteados, demostrando que la misma se encuentra correcta y funcionando, debo mencionar además que el mismo será de gran ayuda para desarrollar las actividades académicas de todos los estudiantes de esta Escuela.

Sin otro particular, me suscribo de usted.

Atentamente,



Srta. Sofía Arias Pérez

PRESIDENTA ASOCIACIÓN ESTUDIANTES

EIS-VALIDADORA