

**UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA
UNAN-LEON**

**FACULTAD DE CIENCIAS
DEPARTAMENTO DE COMPUTACION**



**Aplicación diseñada en Java y Mysql: Sistema de apoyo a la
distribución de becas de la Facultad de Ciencias UNAN-León.**

MONOGRAFIA

Para optar al título de Licenciado en Computación

Autores:

- **Br. Camilo Francisco Narváez Sánchez.**
- **Br. Eleazar Humberto Trejos Mejía.**

Tutor:

Ing. Máximo Damasco Guido.

León, Nicaragua Febrero del 2008.

DEDICATORIA

A Dios que sin el nada funciona bien. y me dio el privilegio de llegar a estos momentos y me ayudó siempre a superar todas las situaciones difíciles.

Dedico este trabajo a mis padres que me han dado el mejor ejemplo desde mis primeros días de vida hasta la actualidad.

A mis hermanos de los que recibí apoyo incondicional sobre todo en las situaciones difíciles y momentos que mas se necesita.

Y a todos los que demostraron ser amigos verdaderos y siempre estuvieron presentes en el transcurso de estos años

Que Dios los acompañe gracias!!

Eleazar Trejos mejía

DEDICATORIA

Dedico este trabajo en primera instancia a Dios todo poderoso, el creador de la vida y a la Virgen Santísima, quienes me dieron fuerzas a lo largo de toda mi vida y mi carrera para culminar con éxitos mis estudios.

A mis abuelitos Delia y Alfonso, quienes fueron el soporte fundamental de mi formación y a quienes agradezco profundamente el apoyo incondicional que siempre me brindaron.

A mis padres que estuvieron conmigo, apoyándome y sacrificándose para que pudiera alcanzar este sueño, y que además de padres fueron mis verdaderos amigos, en los cuales deposité siempre mi confianza.

A mi familia que nunca me abandonó en los momentos mas difíciles y me alentaron a seguir, dándome sus consejos y ayuda, en especial a mis tíos Guillermo y Delia Narváez.

A todos muchas Gracias!.

Camilo F. Narváez Sánchez.

AGRADECIMIENTO

Realizamos este trabajo gracias a Dios, que nos dió aliento y ánimos para no decaer ante los obstáculos.

Especialmente debemos agradecimiento a nuestros familiares, a quienes les debemos los consejos oportunos y darnos su ayuda incondicional siempre.

A nuestro tutor Máximo Guido, que además de profesor fué nuestro compañero, quien nos supo dirigir y ayudar en momentos necesarios.

Agradecemos profundamente a nuestros profesores que fueron pilar fundamental de nuestra formación académica, a quienes siempre estuvieron dispuestos a apoyarnos, brindándonos información y documentación necesaria para la realización de este trabajo, especialmente a los profesores: Aldo Martínez, Álvaro Altamirano y Ricardo Espinoza.

INDICE

I. INTRODUCCION.....	1
II. ANTECEDENTES.....	2
III. JUSTIFICACIÓN.....	3
IV. OBJETIVOS.....	4
V. MARCO TEÓRICO.....	5
- Herramientas a utilizar para el desarrollo del Software.....	7
V.I. ANALISIS ESTRUCTURADO.....	8
V.II. MATERIAL Y METODO.....	10
1 - Análisis de requisitos.....	11
2- Diseño Software.....	11
3 - Diseño detallado del Sistema.....	11
4- Ciclo de vida, modelo en cascada	12
V.III. PLAN DE PROYECTO DEL SOFTWARE.....	13
1- Estimación del Proyecto.....	13
2- Riesgos del proyecto.....	13
Análisis de Riesgos.....	13
Estimación del Riesgo.....	13
Gestión y supervisión del Riesgo.....	14
3- Recursos del Proyecto.....	14
4- Mecanismos de seguimiento y control.....	15
V.IV. SOFTWARE LIBRE.....	16
Ventajas y desventajas de utilizar software libre	17
¿Qué es el copyleft.....	19
V.V. JAVA.....	21
¿Cómo distribuir una aplicación realizada en Java?.....	22
Archivos jar.....	23
V.VI. MYSQL Y EL CÓDIGO ABIERTO.....	30
Instalación Manual de MySQL.....	31
Conexión y desconexión al servidor.....	32
V.VII. E.R.S.....	35
-Funciones.....	38
-Aspectos de funcionamiento.....	39
-Restricciones técnicas y de gestión.....	39
V.VIII. DIAGRAMA DE FLUJO DE DATOS.....	48

V.IX. DICCIONARIO DE DATOS.....	51
V.X.. DIAGRAMA ENTIDAD – RELACION (ER).....	52
V.XI. DISEÑO ESTRUCTURAL.....	53
V.XII. DISEÑO ARQUITECTONICO.....	55
V.XIII. DISEÑO PROCEDIMENTAL.....	56
V.XIV. DIAGRAMA DE CLASES.....	57
V.XV. DISEÑO DE INTERFAZ DE USUARIO.....	58
V.XVI. CODIFICACION.....	64
VI. CONCLUSIONES.....	90
VII. RECOMENDACIONES.....	91
VIII. BIBLIOGRAFIA.....	92
VIX. ANEXOS.....	93

INDICE DE FIGURAS

I.	Fig. 1.=> Modelo Ciclo de vida en cascada.....	11
II.	Fig. 2.=> Interfaz del programa Exe4j.....	26
III.	Fig. 3.=> Interfaz Exe4j (Elección del tipo de proyecto).....	27
IV.	Fig. 4.=> Interfaz Exe4j (Nombre y directorio del proyecto).....	27
V.	Fig. 5.=> Interfaz Exe4j (Configuración del .exe).....	28
VI.	Fig. 6.=> Interfaz Exe4j (Inclusión del archivo .jar y clase principal).....	28
VII.	Fig. 7.=> Interfaz Exe4j (Compilación del .exe).....	29
VIII.	Fig. 8.=> Interfaz Exe4j (Finalización del proyecto).....	29
IX.	Fig. 9.=> Ventana de configuración (Orígenes de datos).....	34
X.	Fig. 10.=> Ventana de configuración (Controlador de Access).....	34
XI.	Fig. 11.=> Diagrama de flujo de datos (Nivel 0).....	48
XII.	Fig. 12.=> Diagrama de flujo de datos (Nivel 1).....	49
XIII.	Fig. 13.=> Diagrama de flujo de datos (Nivel 2).....	50
XIV.	Fig. 14.=> Diagrama entidad relación (ER).....	52
XV.	Fig. 15.=> Diseño estructural (Tabla estudiante).....	53
XVI.	Fig. 16.=> Diseño estructural (Tabla solicitud).....	54
XVII.	Fig. 17.=> Diseño arquitectónico.....	55
XVIII.	Fig. 18.=> Diseño procedimental.....	56
XIX.	Fig.19.=> Diagrama de clases.....	57
XX.	Fig. 20.=> Formulario control de acceso.....	58
XXI.	Fig. 21.=> Formulario cambiar contraseña.....	58
XXII.	Fig. 22.=> Formulario menú principal.....	59
XXIII.	Fig. 23.=> Formulario registro estudiantes.....	60

XXIV. Fig. 24.=> Formulario nuevos becados.....	60
XXV. Fig. 25.=> Formulario registro solicitudes.....	61
XXVI. Fig. 26.=> Formulario consulta de solicitudes.....	62
XXVII. Fig. 27.=> Formulario generar bajas.....	62
XXVIII. Fig. 28.=> Formulario consulta de becados.....	63
XXIX. Fig. 29.=> Formulario búsqueda.....	63



I. INTRODUCCION

En las últimas décadas se ha presentado un acelerado incremento en demanda de sistemas automatizados, razón por la cual empresas e instituciones necesitan adoptar este ritmo de desarrollo para mantenerse en la competitividad lo que trae como principal beneficio la reducción de tiempos en realización de tareas y culminación de labores internas y externas.

Este proyecto de monografía consiste en la creación de un sistema de gestión y control de becas, su principal funcionalidad se enfoca en llevar automatizadas las actividades de vicedecanatura con respecto a estudiantes becarios de la facultad de ciencias de la UNAN-León.

Aquí se aplicaron los principales conceptos de Ingeniería del software como el análisis, diseño y desarrollo de software así como las herramientas necesarias para la creación de diagramas, generación de códigos fuentes y base de datos

Las herramientas utilizadas para el desarrollo de este sistema son de libre adquisición como son el mysql 5.0 (gestor de base de datos) y el compilador java JCreator 3.5, lo que necesita cierta configuración manual para la interconexión con bases de datos

El objetivo final de este sistema es llevar un registro adecuado de las becas otorgadas a en la dicha facultad y advertir automáticamente los posibles estudiantes que no son merecedores de la misma por razones académicas, partiendo de estudiantes solicitantes de becas previamente registrados y relacionados con sus calificaciones.



II. ANTECEDENTES

El sistema de gestión control y selección de becas se realiza de forma manual, exigiendo tiempo y esfuerzo humano, propenso a cometer errores lo que requiere revisión posterior.

La Universidad y Vice-Decanatura en particular no posee ningún mecanismo automatizado para la gestión de becas, sin embargo la consulta de notas académicas de los estudiantes se hace en una base de datos en secretaría general que puede ser usada para el mismo propósito en un sistema automatizado.

Se registran esquemáticamente los estudiantes, se almacenan y se borran de forma manual los que obtienen beca y los que la pierden.



III. JUSTIFICACIÓN

La realización de este trabajo vendrá a solucionar la necesidad de reducir el tiempo requerido en decidir que alumnos gozaran o no de beca, lo que hará el sistema proporcionando información de una manera rápida segura y eficiente.

También permite llevar un registro de solicitud de becas asignadas lo cual no se hace actualmente.

Además las herramientas software utilizadas para desarrollar e implementar este proyecto son libres, no incluye costo alguno en la adquisición de licencias



IV. OBJETIVOS

-General:

- Facilitar la distribución y registro de becas a través de un sistema computarizado, diseñado con herramientas de software libre.

-Específicos:

- Llevar control organizado de las solicitudes de becas y estudiantes becados, evitando la acumulación de documentos escritos a través de una base de datos.
- Agilizar el acceso a información y creación de listas concerniente a los estudiantes becados y solicitudes de becas.
- Garantizar el respaldo de información de forma rápida y sencilla.



V. MARCO TEÓRICO

En la actualidad, las normativas para la asignación de un tipo de beca a un determinado estudiante establecen ciertos criterios que tiene que cumplir el estudiante para poder obtenerla. La asignación se realiza a través de una comisión que evalúa en los aspirantes aspectos como: su procedencia geográfica, en particular cuando se solicita beca interna.

La situación académica del estudiante es muy importante, por que se hacen valoraciones, entre las solicitudes y se verifica si el estudiante es repitente, o si aprobó el año; en este último caso también se toma en cuenta, cuantos cursos remediales realizó, si lleva pendiente alguna clase, si ha aprobado materias por tutorías o por exámenes especiales. Obviamente tienen más posibilidades aquellos aspirantes cuya situación académica sea mejor.

Esta comisión está conformada por el Vice-Decano de la Facultad, el Presidente de la Asociación de estudiantes y el secretario de Becas que también pertenece a la asociación de estudiantes. Cada semestre esta comisión evalúa las nuevas solicitudes y quita becas a aquellos estudiantes que reprobaron clases.

El procedimiento para aspirar a una beca consiste en lo siguiente:

1. El estudiante debe aprobar el semestre limpio, es decir sin reprobar ninguna materia.
2. Deberá llenar un formulario de solicitud de Beca, este formulario es proporcionado por la Vice-Decanatura o por la Asociación de estudiantes de forma gratuita.
3. El formulario es entregado junto con una carta del solicitante exponiendo de manera mas detallada los motivos de su solicitud, en las oficinas de Vice-Decanatura.
4. Posteriormente deberá presentarse a una entrevista con la comisión de Becas en la fecha correspondiente.



Un estudiante pierde su beca de la siguiente manera:

1. Cuando reprueba una materia.
2. Por indisciplina.
3. Cuando su promedio semestral es inferior al 70%.

Las becas que asigna la Universidad pueden ser becas externas e internas, que son las mas comunes, las becas externas consisten en una asignación monetaria mensual para los gastos del estudiante, las becas internas se otorgan sobretodo a estudiantes que no son de León, se les proporciona alojamiento y alimentación; sin embargo existen otros tipos de becas que resuelven en gran medida las necesidades de los estudiantes, entre éstas se encuentran Bonos alimenticios, Fotocopias, Transporte especialmente para los que viajan frecuentemente a los Departamentos y Municipios fuera de la ciudad de León. Beca residencia es un tipo de beca en la cual se le asigna alojamiento al estudiante pero sin alimentación. Beca por excelencia es la beca más difícil de obtener, esta beca solo la adquieren aquellos estudiantes que alcanzan un promedio de 90% o superior y consiste en una asignación monetaria semestral. Cualquier estudiante aunque ya posea una determinada beca puede obtener Beca por excelencia, no así en los otros tipos de becas.

La asignación monetaria de una Beca por excelencia hasta la fecha de hoy es de C\$ 1,500.00 al finalizar un semestre. Para una beca externa la asignación es de C\$ 500.00 mensual. En la beca interna aparte del alojamiento y alimentación se le asigna al estudiante C\$ 150.00 mensual para gastos varios.

Al finalizar el segundo semestre del año 2006 la distribución de becas en la Facultad de Ciencias, estaba conformada de la siguiente manera:

194 becados externos, 91 becados internos, 107 becados con bonos de alimentación, 50 becados con fotocopias y 12 becados por excelencia.



Herramientas a utilizar para el desarrollo del Software.

Para el desarrollo y la implementación de este proyecto se deben tener en cuenta aspectos como las herramientas necesarias para su funcionamiento y tener claro el uso de los términos que en él hemos empleado, lo que hacemos mención a continuación.

Utilizaremos Software de libre distribución, el cual marca la nueva tendencia para el desarrollo de aplicaciones muy robustas y potentes para instituciones y empresas. Desarrollaremos este sistema a través de programación orientada a objetos basada en el lenguaje de programación java, el cual proporciona un sin número de opciones favorables para programadores.

La información de nuestro sistema será almacenada haciendo uso de una herramienta automatizada como lo es el **EasyCase**, todo sistema de información es considerado como un proyecto, por tanto la información descrita en cada aplicación se mantiene por separado de los otros proyectos.

El gestor de Base de Datos a Utilizar principalmente es MySQL en versión 5.0, donde se almacenaran los registros concernientes a estudiantes becados y solicitantes, el cual es un poderoso sistema de base de datos relacionales que puede implementarse en cualquier plataforma, ya sea Windows o UNIX, también se utiliza el gestor de Base de Datos Access de Microsoft para consultar notas de un estudiante, posteriormente se explicará por qué el uso de éste gestor de Base de datos, haciendo un paréntesis en la aplicación de Software libre.

Para el desarrollo de nuestro sistema consideramos los métodos estratégicos mas apropiados actualmente como son:

- Modelo de Ciclo de vida.
- Modelo de Cascada.



V.I. ANALISIS ESTRUCTURADO.

El análisis estructurado es una actividad de construcción de modelos, se concentra en especificar lo que se requiera que haga el sistema. Permite que las personas observen los elementos lógicos (lo que hará el sistema) separado de los componentes físicos (computadoras, terminales, sistema de almacenamiento, etc.). Después de esto, se puede desarrollar un sistema físico eficiente para la aplicación donde será utilizado.

Los elementos esenciales del análisis estructurado son símbolos gráficos, diagramas de flujo de datos y el diccionario centralizado de datos.

Las características de aplicación de este modelo son:

1. Adecuado para todo tipo de aplicaciones.
2. Mayor utilidad como complemento de otros métodos de desarrollo.


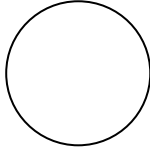
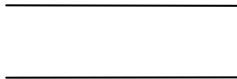

DIAGRAMA DE FLUJO DE DATOS (DFD).

El diagrama de flujo de datos es una técnica gráfica que representa el flujo de la información y las transformaciones que se aplican a los datos al moverse de la entrada a la salida.

Se puede usar el DFD para representar un sistema o un software a cualquier nivel de abstracción, mostrando sus componentes, las conexiones entre ellos y los datos de control. Así, un DFD de nivel 0 también es denominado “grafo de Flujo de Datos” o “Diagrama de Burbuja”.



Notaciones básicas del DFD:

- Flujo de datos (datos en movimiento). 
- Procesos (transformación de los datos). 
- Almacén de datos (datos en reposo). 
- Entidad externa (origen/destino). 

DICCIONARIO DE DATOS.

Es un almacén en el que están definidos rigurosamente todos los elementos del DFD que son:

- Flujo de datos.
- Componentes del flujo de datos.
- Procesos.
- Ficheros.
- Almacenes.



V.II. MATERIAL Y METODO.

Modelo de Cascada:

Este modelo descompone el proceso de desarrollo en diferentes fases, constituyendo la salida de cada una de ellas en la entrada requerida por la siguiente.

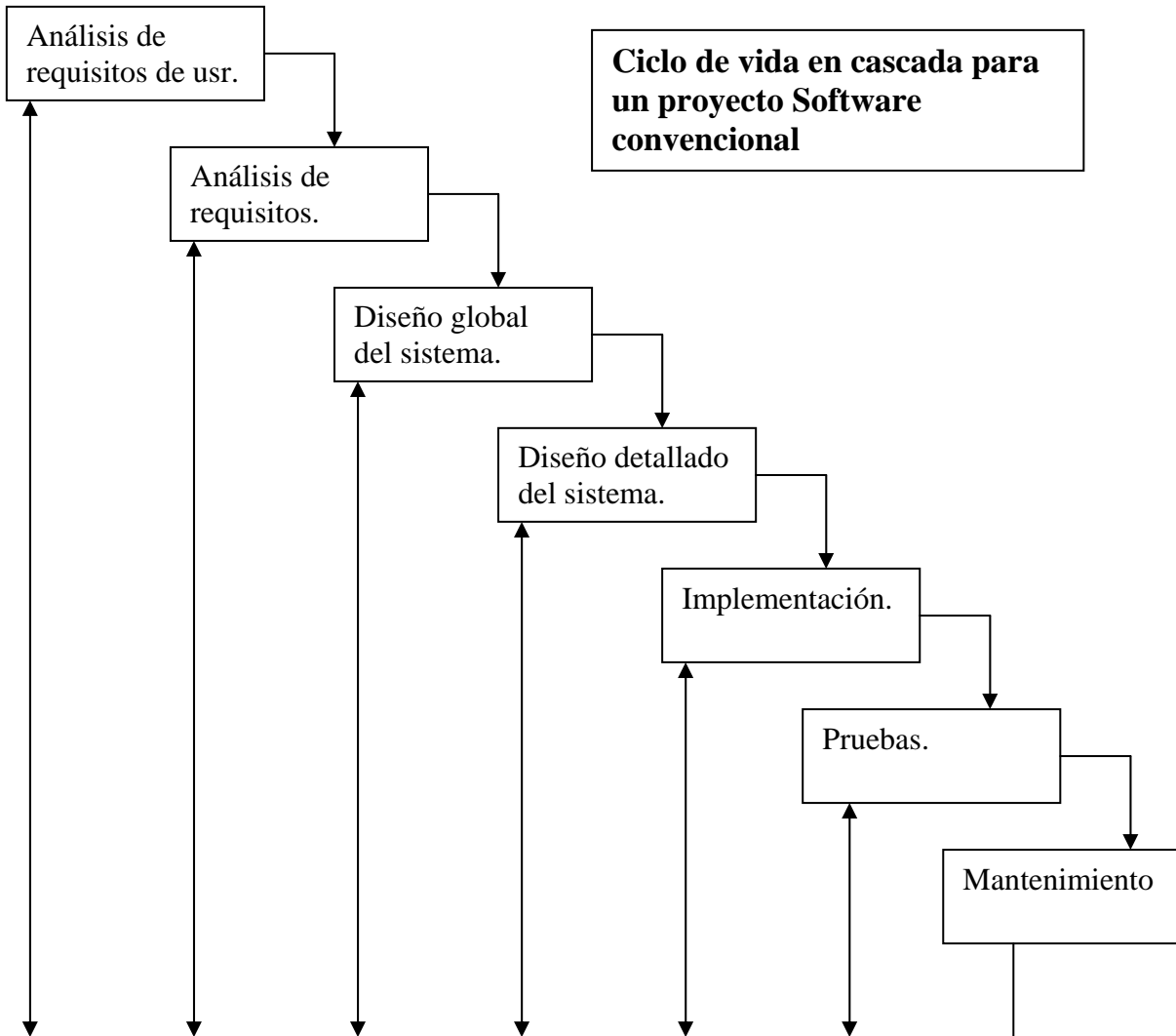


Fig. 1.

Para el desarrollo de este sistema estamos utilizando el Modelo del Ciclo de Vida en Cascada. Este modelo es de gran importancia ya que nos permite desarrollar el Software en etapas llevando un manejo ordenado y secuencial del proyecto, lo cual nos facilita la realización, manipulación y ejecución del sistema.



- Análisis de requisitos:

Para poder realizar la fase de análisis de los requisitos del Software, es necesario hacer una recopilación de datos de diferentes formas:

- A través de entrevistas con la autoridad de la institución, en nuestro caso con el Vicedecano.
- Observaciones directas del manejo de las funciones de Vice-Decanatura y de cómo lleva el control manual de los procesos de Selección y gestión de becas.
- Utilizando información, proporcionada por la institución.

- Diseño Software.

El diseño es el primer paso de la fase de desarrollo de cualquier producto y puede definirse como el proceso de aplicar distintas técnicas y principios con el propósito de definir un sistema. El diseño es el proceso en que se asienta la calidad del Software y es la única forma de mediante la cual podemos traducir con precisión los requisitos del usuario final en un sistema acabado.

En la fase de diseño se debe indicar cómo se cumplirán los requerimientos, es decir, se debe especificar la emisión de reportes de salida, pantallas de entrada de datos, estructuras de Base de Datos, etc.

- Diseño detallado del Sistema:

La fase de de diseño implica trabajar con cuatro tipos de diseño como son:

1. Diseño de Datos: es la selección de las representaciones lógicas de datos identificados durante la definición y especificación de requisitos.
2. Diseño arquitectónico: es desarrollar una estructura de un programa modular y representar las relaciones de control entre los módulos.
3. Diseño Procedimental: transforma los elementos estructurados en una descripción procedimental del Software, éste debe especificar los detalles de los procedimientos sin ambigüedad.



4. Diseño de Interfaz: establece la disposición y los mecanismos para la interacción hombre-maquina.

En nuestro proyecto hemos realizado las etapas de análisis de los requisitos y el diseño, las cuales marcan la pauta para realizar la codificación.

- Ciclo de vida, modelo en cascada

Todos los Modelos del Ciclo de Vida tienen en común las actividades de gestión del proyecto, mediante la descomposición del mismo en etapas, el empleo de alguna metodología de trabajo en cada etapa, así como la utilización de herramientas de soporte y la verificación y validación de cada una de las fases de desarrollo.

El modelo consta de las siguientes actividades:

1. Investigación preliminar.
2. Determinación de los requerimientos del sistema.
3. Diseño del sistema.
4. Desarrollo del Software.
5. Prueba del sistema.
6. Implantación y evaluación.

Este modelo es muy usado y conocido por ser menos costoso en la elaboración de proyectos Software, aunque los resultados se ven hasta la culminación del proyecto.

Utilizaremos EasyCase para realizar el DFD de nuestro sistema. El DFD a medida que se refine en mayores niveles de detalle conllevará implícitamente a una descomposición funcional del sistema permitiendo fácilmente entender lo que el sistema va a realizar, además nos facilitará cumplir con la fase de diseño y ésta a su vez a la codificación.



V.III. PLAN DE PROYECTO DEL SOFTWARE

1- Estimación del Proyecto:

Este Software como muchos otros, tendrá su propio grado de complejidad, el cual tratamos de minimizar, para lograr que sea lo menos complejo posible, fácil de entender y utilizar. Otro factor de estimación es el tamaño del Software que dependerá del número de funciones a realizar.

2- Riesgos del proyecto:

Análisis de Riesgos.

Nuestro proyecto podría correr los siguientes riesgos, entre otros:

- No saber distribuir bien el tiempo para la realización de las actividades programadas y de esta manera no cumplir con la agenda.
- El Software no llene los requisitos de la Institución, por lo tanto no cumplir satisfactoriamente con sus especificaciones.
- Si al momento de la implementación del Software, surgen nuevas funciones o especificaciones propuestas por la Institución que tendrían que tomarse en cuenta.
- Es posible que al momento de la verificación y validación hubieran hecho falta pruebas, correcciones y validaciones al darle el mantenimiento al Software, debido a cambios ocurridos en el funcionamiento interno de la Institución y se necesite adaptación del mismo a nuevas mejoras.

Estimación del Riesgo.

Esta actividad será utilizada para evaluar o determinar los riesgos de nuestro proyecto tales como:

- 1- Probabilidad de que el riesgo ocurra.
- 2- Consecuencias en caso que ocurra
- 3- Prevención del riesgo.



Para prevenir estos riesgos se realizarán entre otras, las siguientes actividades:

- Cumplimiento exacto de la agenda para prevenir desfases.
- Estudio y practica profunda de las herramientas a utilizar en caso de llegar a tener algún problema o atraso al momento de la codificación.
- Se realizarán técnicas de contraseña y Backup para los archivos de la aplicación, en caso de fallas del ordenador o Software.
- Se realizará simulación, pruebas, comprobación y validación de los datos.

Gestión y supervisión del Riesgo.

Estas actividades serán el resultado de nuestra fase de análisis del riesgo y consisten principalmente en:

- Detectar el momento en que ocurra un riesgo.
- Asegurar un plan de prevención al riesgo.
- Recopilar información.

Para evitar riesgos, nuestro Software será elaborado bajo contraseña de usuario, se crearán copias de seguridad, su manejo será restringido.

3- Recursos del Proyecto.

Personal:

Desarrolladores del Proyecto:

- Camilo F. Narváez Sánchez.
- Eleazar H. Trejos Mejía.

Hardware:

Un ordenador con las siguientes Características:

- CPU Celeron D, 2.8 GHz.
- 256 MB de memoria RAM.
- 80 GB de Disco Duro.
- Impresora HP Deskjet D 1360.



Software:

- Windows XP/SP2 Profesional
- MYSQL 5.0
- JCreator 3.50
- Microsoft Access 2003
- Herramientas CASE.

Recursos Especiales:

Se requiere que el usuario final, posea conocimientos y experiencias básicas para el manejo del sistema.

4- Mecanismos de seguimiento y control.

Nosotros como desarrolladores del sistema, seremos los responsables de llevar a cabo estos mecanismos, a través de las siguientes actividades:

- Reuniones de equipo de trabajo semanalmente.
- Reuniones con las autoridades de la Institución.
- Evaluaciones de logros y dificultades en periodos de tiempos convenientes.
- Análisis de cumplimiento de fechas.



V.IV. SOFTWARE LIBRE

Introducción

Al hablar de software libre, se refiere a la libertad, no al precio. Estos términos se refieren al modelo de desarrollo y de distribución del software elaborado cooperativamente. En vez de que el código del sistema o de cada uno de los programas sea un secreto celosamente guardado por la empresa que lo produce, éste es puesto a disposición del público, para que puedan modificar, mejorar o corregir. Las Licencias Públicas Generales están escritas para asegurarnos de que tenga la libertad de distribuir copias de software libre (y cobrar por ese servicio si quiere), de que reciba el código fuente o que pueda conseguirlo si lo quiere, de que pueda modificar el software o usar fragmentos de él en nuevos programas libres, y de que sepa que puede hacer todas estas cosas.

Para proteger sus derechos se requieren algunas restricciones que prohíban a cualquiera negarle a usted estos derechos o pedirle que renuncie a ellos. Estas restricciones se traducen en ciertas obligaciones que le afectan si distribuye copias del software, o si lo modifica. Por ejemplo, si distribuye copias de uno de estos programas, sea gratuitamente, o a cambio de una contraprestación, debe dar a los receptores todos los derechos que tiene. Debe asegurarse de que ellos también reciben, o pueden conseguir, el código fuente. Y debe mostrarles estas condiciones de forma que conozcan sus derechos.

También, para la protección de cada autor asegurar de que todo el mundo comprende que no se proporciona ninguna garantía para este software libre. Si el software se modifica por cualquiera y éste a su vez lo distribuye, se requiere que sus receptores sepan que lo que tienen no es el original, de forma que cualquier problema introducido por otros no afecte a la reputación de los autores originales.



Ventajas y desventajas de utilizar software libre

Ventajas:

- Beneficio sociales y tecnológicos para el país:

El uso del software libre ayudaría a garantizar la educación de los individuos y al estado a garantizar el desarrollo nacional de manera integral y sustentable.

- Ahorros multimillonarios en la adquisición de licencias
- Combate efectivo a la copia ilícita de software
- Eliminación de barreras presupuestales
- Tiempos de desarrollo sobre algo que no exista son menores por la amplia disponibilidad de herramientas y librerías.
- Las aplicaciones son fácilmente auditadas antes de ser usadas en procesos de misión crítica, además del hecho de que las más populares se encuentran muy depuradas.
- Todo el mundo tiene derecho a acceder a su diseño y aprender de él.
- Todo el mundo tiene derecho de modificarlo: si el software tiene limitaciones o no es adecuado para una tarea, es posible adaptarlo a necesidades específicas y redistribuirlo libremente.
- Es de libre distribución (cualquier persona puede regalarlo, venderlo o prestarlo).
- Tiende a ser muy eficiente (por que mucha gente lo optimiza, mejora).
- Tiende a ser muy robusto (mucha gente puede arreglarlo, no solamente el creador o la compañía que lo produce).
- Tiende a ser muy diverso: la gente que contribuye tiene muchas necesidades diferentes y esto hace que el software esté adaptado a una cantidad más grande de problemas.
- Reutilización del conocimiento: Esto permite que la gente reutilice el conocimiento que se ha sintetizado en el software. En vez de empezar siempre desde cero (que es el caso de la industria de software actual) siempre se puede empezar un proyecto desde un fundamento establecido.
- Aprender alguna técnica de programación.



Desventajas

- La curva de aprendizaje es mayor

Si ponemos a dos señoras que nunca han tocado una PC, probablemente tardaran lo mismo en aprender a usar una Windows que Gnome o KDE la otra.

- El software libre no tiene garantía proveniente del autor.
- Se necesita dedicar recursos a la reparación de erratas.
- La mayoría de la configuración de hardware no es intuitiva, se requieren conocimientos previos acerca del funcionamiento del sistema operativo y fundamentos del equipo a conectar para lograr un funcionamiento adecuado.
- Únicamente los proyectos importantes y de trayectoria tienen buen soporte, tanto de los desarrolladores como de los usuarios; sin embargo existen muchos proyectos más pequeños y recientes que carecen del compromiso necesario por parte de sus usuarios o desarrolladores para que sean implementados de manera confiable.
- El usuario debe tener nociones de programación, ya que la administración del sistema recae mucho en la automatización de tareas y esto se logra utilizando, en muchas ocasiones, lenguajes de guiones (perl, python, shell, etc.).
- La diversidad de distribuciones, métodos de empaquetamiento, licencias de uso, Herramientas con un mismo fin, etc., pueden crear confusión en cierto número de Personas.

Para que sea libre tiene que publicarlo con una licencia de software libre. Utilizar la Licencia Pública General de GNU (GNU GPL), pero eventualmente también se puede utilizar otras licencias de software libre.

La documentación del software libre debería ser documentación libre, para que se pueda redistribuir y mejorar al igual que el software al cual describe. Para que sea documentación libre tiene que publicarla con una licencia de documentación libre. En tal caso la Licencia de Documentación Libre de GNU (GNU FDL).



Las licencias que cubren la mayor parte del software están diseñadas para quitarle a usted la libertad de compartirlo y modificarlo. Por el contrario, la Licencia Pública General de GNU pretende garantizarle la libertad de compartir y modificar software libre, para asegurar que el software es libre para todos sus usuarios. Esta Licencia Pública General se aplica a la mayor parte del software del la Free Software Foundation y a cualquier otro programa si sus autores se comprometen a utilizarla. (Existe otro software de la Free Software Foundation que está cubierto por la Licencia Pública General de GNU para Bibliotecas). Si quiere, también puede aplicarla a sus propios programas.

Copias literales y distribución

La nota estándar de copyright para las páginas web de GNU dice: Se permite la distribución y la copia literal de este artículo en su totalidad y por cualquier medio siempre y cuando se conserve esta nota.

"La intención al emplear la frase 'copia literal en cualquier medio' no significa conservar el encabezado, el pie de página y otros elementos de maquetación en cualquier tipo de formato. Lo que sí se requiere es conservar los vínculos web en medios con hipervínculos o sin ellos (como notas o alguna forma similar de URL escrita en medios no HTML)."

¿Qué es el copyleft?

Copyleft es la forma general de hacer un programa software libre y requiere que todas las modificaciones y versiones extendidas del programa sean también software libre.

El modo más simple de hacer un programa libre es ponerlo en el dominio público o sea, sin copyright. Esto permitirá que la gente comparta el programa y sus mejoras, si así lo desean. Pero también permitiría a quien no quiera cooperar convertir el programa en software privativo. Pueden hacer cambios y distribuir el resultado como un producto privativo. Las personas que reciban el programa en



su forma modificada no poseen la libertad que el autor original les dio debido a que el intermediario se la ha retirado.

En el Proyecto GNU, la intención es dar a todos los usuarios la libertad de redistribuir y cambiar software GNU. Si el intermediario pudiera coartar la libertad, podríamos tener muchos usuarios, pero esos usuarios no tendrían libertad. Así, en vez de poner software GNU bajo dominio público, se debe hacer "copyleft". Copyleft significa que cualquiera que redistribuya el software, con o sin cambios, no podrá restringir a nadie la libertad de copiarlo, redistribuirlo o cambiarlo. Copyleft garantiza que el usuario mantenga su libertad.

Para hacer un programa copyleft, lo primero es darle un copyright; luego se le añaden cláusulas de distribución, que son un instrumento legal que le da a cualquiera el derecho de usar, modificar y redistribuir el código fuente del programa o de cualquier programa derivado de éste pero sólo si los términos de distribución no se cambian. De este modo, el código y las libertades se vuelven legalmente inseparables.

Los desarrolladores de software privativo emplean el copyright para quitar la libertad a los usuarios; aquí se emplea copyright para garantizar su libertad. Por esto se cambia el nombre "copyright" por "copyleft".

La GNU GPL está diseñada para que pueda aplicarla fácilmente a sus programas si es el titular del copyright. Para hacerlo no tiene que modificar la GNU GPL, sólo añadir una nota en su programa la cual haga referencia a la GNU GPL. Tener en cuenta que si usa la GNU GPL debe conservar el texto íntegro de la licencia. Emplear los mismos términos de distribución para los diferentes programas hace fácil compartir el código entre ellos. La GPL Reducida incluye una cláusula que le permite alterar los términos de distribución de su programa para hacerlo GPL, para así poder copiar código en otros programas que estén cubiertos por la GPL.



V.V. JAVA.

Java es un lenguaje de programación Orientada a objetos, diseñado por Sun Microsystem con el propósito de resolver simultáneamente los problemas que se le plantean a todos los desarrolladores de software, por la proliferación de arquitecturas incompatibles; tanto en las diferentes maquinas como entre los diversos sistemas operativos.

Java fue diseñado para ser un lenguaje fácil de aprender, pero manteniendo la funcionalidad de un lenguaje potente, realiza verificaciones de problemas tanto en tiempo de ejecución como en tiempo de compilación.

No importa el procesador ni el sistema operativo, la única condición para que un programa java pueda correr en un ordenador es que tenga instalada la maquina virtual. Existen implementaciones de maquinas virtuales java para prácticamente todos los sistemas operativos, además existen en dispositivos eléctricos como teléfonos móviles, tarjetas chip, etc. Por tanto se pueden desarrollar programas en cualquier plataforma y hacerlos funcionar en el resto de ellas. Se le puede considerar un lenguaje universal.

En este lenguaje se desarrollan tres tipos de programas:

- Stand Alone: Aplicaciones independientes.
- Applet: Programas que se ejecutan en paginas Web.
- Servlets programas sin interfaz que se ejecuta en un servidor.

Para diseñar aplicaciones que utilicen interfaces Graficas, java proporciona una biblioteca de clases denominada JFC (java foundation class). Actualmente bajo esta denominación se agrupan las siguientes APIs (Interfaz para programación de aplicaciones):

- Swing.
- Awt.
- Accesibilidad.



- Java 2D.
- Soporte para arrastrar y colocar (Drag and Drop).

¿Cómo distribuir una aplicación realizada en Java?

Para distribuir aplicaciones desarrolladas bajo java, existen ciertas diferencias con respecto a los demás lenguajes de programación, como C, C++, Visual Basic y otros. En aplicaciones desarrolladas en todos estos lenguajes se necesitan archivos ejecutables, para su funcionamiento. Los archivos ejecutables son dependientes de la plataforma. En Java, en principio, no se pueden crear ejecutables ya que este es un lenguaje multiplataforma es una de sus ventajas. Esto provoca que tengamos que usar la consola para ejecutar nuestros programas desde un conjunto de clases lo que supone un gran inconveniente para usuarios sobre todo acostumbrados a los entornos gráficos. *Una clase puede correrse desde cualquier plataforma mientras que un archivo ejecutable *.exe en la mayoría de los casos no.*

Existen varias soluciones para este asunto. Aquí se explican tres de ellas. La primera es la de crear ejecutables dependientes de la plataforma; la segunda permite hacer más fácil la ejecución en cada plataforma sin impedir su ejecución en otra plataforma; la tercera es válida para cualquier plataforma y aporta otras muchas posibilidades.

Soluciones

Crear ejecutables de la plataforma

Podemos crear un ejecutable de la plataforma de varias maneras:

Podemos usar **traducir el código java a código C** con Toba o J2C. De esta forma sólo nos queda compilar el código C para crear el ejecutable. Toba es un



proyecto libre cuyo desarrollo ha quedado estancado cuando Java iba por la versión 1.1,

Existe un compilador que además de permitirnos, como cualquier otro compilador, crear bytecodes a partir de ficheros .java, nos da la posibilidad de compilar la fuente (archivos.java) para obtener un ejecutable de la plataforma y también permite crear un ejecutable de la plataforma a partir del bytecode. De esta forma el código obtenido es mucho más rápido. El programa se llama **GJC** y forma parte la familia de compiladores GCC del proyecto GNU que se desarrolla bajo la GPL (General Public License).

Si lo que queremos es crear un archivo ejecutable .exe (para plataforma Windows) y prescindir de la cualidad de multiplataforma de este lenguaje, lo que hay que hacer es conseguir un programa que realice esta tarea. Esto lo hacen algunas IDE's y el programa **exe4j**. Al crear un .exe lo que hacemos es sustituir los archivos de clase de Java, al contrario que la siguiente opción que complementa y no sustituye. En particular el programa **exe4j necesita de la máquina virtual de Java para poder ejecutar el programa** de modo seguiremos necesitando instalar la JVM al contrario que las dos opciones anteriores.

Crear scripts

Otra posibilidad, que nos permite mantener la multiplataforma es **crear un script para cada plataforma**. Esto lo que nos permite es facilitar la ejecución dentro de cada plataforma sin evitar que podamos ejecutar el programa en otra para la que no hayamos creado un script. Un script no es más que un archivo en el que se guardan los comandos que se teclearían en la consola (es decir, lo que teclearías en una ventana MS-DOS en caso de usar este Sistema Operativo o en una shell para los que usen Unix) para que fuera interpretados por el intérprete de comandos.



Archivos jar

La más recomendada posibilidad es la de hacer un archivo jar ejecutable. Los archivos jar permiten introducir todos los archivos necesarios para ejecutar nuestra aplicación (clases, imágenes, sonidos, archivos de texto,...) en uno solo con la posibilidad de comprimirlos. Pero además ofrecen otras posibilidades, entre las que se encuentra la posibilidad de añadir información adicional: versión del programa, quién lo ha creado, y la que en nuestro caso nos interesa: indicar la clase principal (la que contiene el método main) para poder ejecutarla con el intérprete del JDK (el programa java). Esta información se guarda en un archivo dentro del jar, que se llama **MANIFEST.MF** cuya ruta dentro del archivo jar será siempre **META-INF/MANIFEST.MF**. Este archivo es creado por defecto siempre que se crea un archivo jar.

Para administrar un archivo jar se usa el programa jar del JDK.

Crear un archivo jar

A continuación se muestra un ejemplo de cómo crear un archivo jar con las opciones más usuales.

```
jar cvf MiArchivo.jar Clase1.class Clase2.class Imagen.gif Sonido.au  
OtrosArchivos
```

En esta ocasión se ha creado un archivo jar de nombre MiArchivo.jar que contendrá los archivos Clase1.class, Clase2.class, Imagen.gif y Sonido.au y el directorio OtrosArchivos. Como se puede observar en un archivo jar podemos meter cualquier tipo de archivo, incluso directorios. Si existe alguna referencia a archivos dentro de otro directorio estas se mantendrán sin producirse errores ya que la jerarquía de directorios se mantiene dentro del archivo. En el ejemplo se han empleado las opciones c, f y v de la línea de comandos. Estas opciones y otras se explican a continuación:



c.- Indica que vamos a crear un archivo.

f.- Indica que la salida quieres que vaya a un fichero, y no a la salida estándar.

v.- Indica que quieres obtener información durante el proceso de creación del archivo.

m.- Indica al programa que queremos personalizar el archivo de manifiesto para añadir información adicional.

M.- Indica que no se cree el archivo de manifiesto.

0.- Indicamos que no queremos que se comprima el contenido del archivo.

Modificar el archivo de manifiesto

Llegados a este punto tenemos que crear el archivo de manifiesto. El archivo de manifiesto es un archivo de texto plano (sin formato) por lo que podrá ser creado con cualquier editor de textos convencional. El archivo podrá tener cualquier nombre y/o extensión ya que lo que el programa jar lo que hará será mezclar el contenido de este archivo con el que se crea por defecto.

En nuestro caso el archivo de manifiesto tendrá una sola línea en la que indicaremos la clase principal (la que contiene el método main). Esto se hace de la siguiente forma:

Main-Class: Clase Principal

De este modo se indica cual es la clase principal (sin .class). Ahora deberemos crear el archivo jar con esta información. A continuación se muestra un ejemplo:
jar cfm MiManifiesto.mf MiArchivo.jar ClasePrincipal.class OtraClase.class Imagen.gif

Se observa que tras las opciones, en las que está la opción m explicada anteriormente, se indica el nombre del archivo de manifiesto. Ahora solamente tendremos que indicar que los archivos .jar se abran con el programa java y con la opción -jar en la línea de comandos. Esto ya lo hace el instalador del J2SDK por lo que no será necesario hacerlo a mano.



Ejecutar aplicaciones desde un archivo .jar

Si deseas ejecutar aplicaciones Java que se encuentran en *.jar, solo necesitas bajar el JRE v1.3 o en su defecto el J2SE v1.4 porque ambos incluyen parámetros que automáticamente modifican Windows para que ejecuten los *.jar con un doble clic.

En nuestra aplicación, creamos el archivo .jar y lo convertimos a .exe (ejecutable de Windows) utilizando el programa “exe4j” descargado de la siguiente dirección de Internet “<http://www.ej-technologies.com/download/exe4j/files.php>”

El .exe se crea a partir de los siguientes pasos:

- Cargar la aplicación exe4j.



Fig. 2.



- Seleccionar el tipo de proyecto que se desea a partir del .jar; existen dos modos:

Modo regular: permite incluir otros archivos .jar.

Modo .jar in exe: no permite incluir otros archivos jar, se limita a crear un ejecutable sencillo. Para esta aplicación se utiliza modo jar in exe.

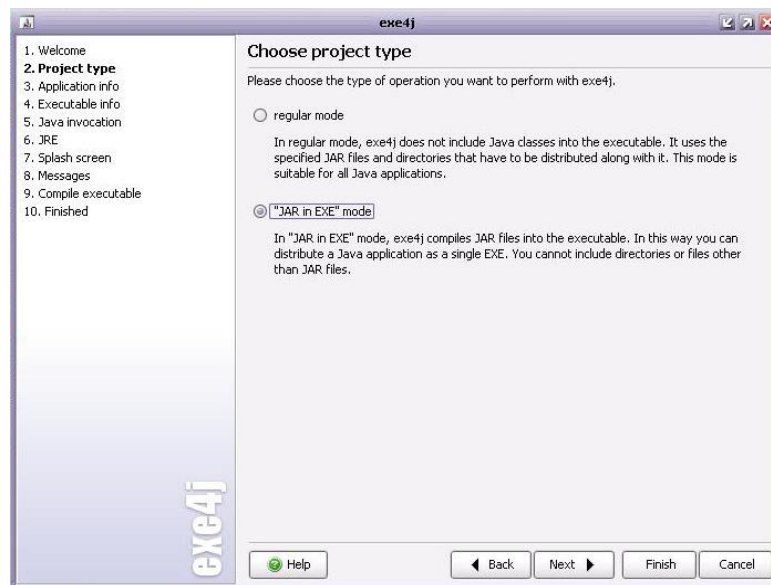


Fig. 3.

Nombrar el proyecto y el directorio donde se almacena el ejecutable.

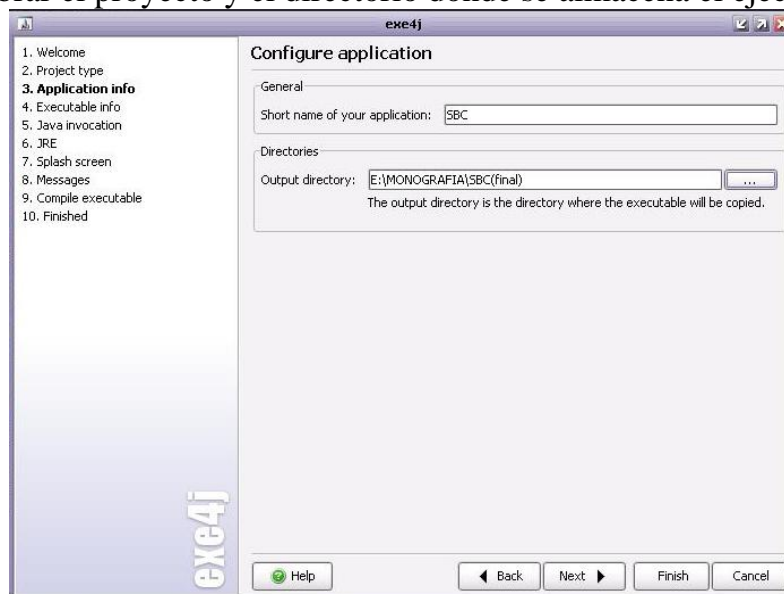


Fig. 4.



- Configurar el ejecutable; tipo de icono, tipo de aplicación (modo consola, modo grafico, servicio).

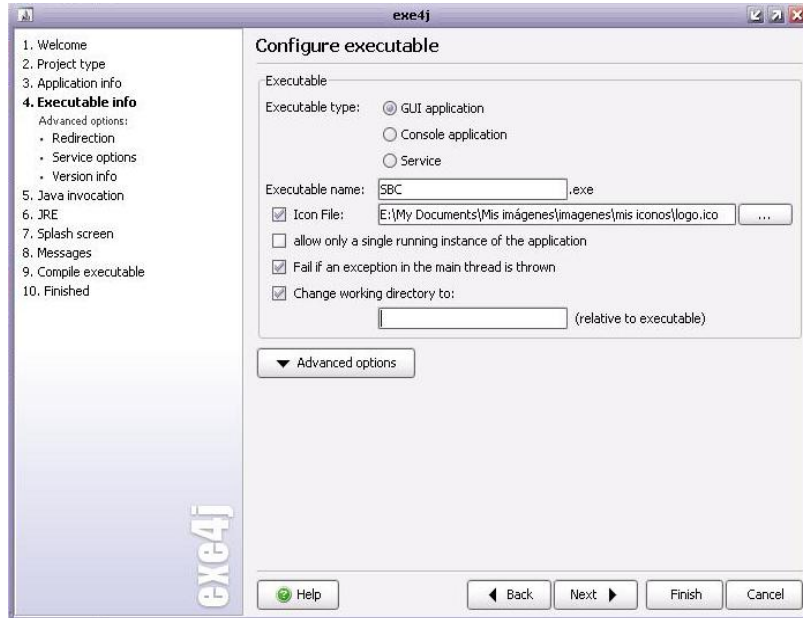


Fig. 5.

- Agregar el archivo .jar a utilizar y la clase principal (clase que contiene el método main).

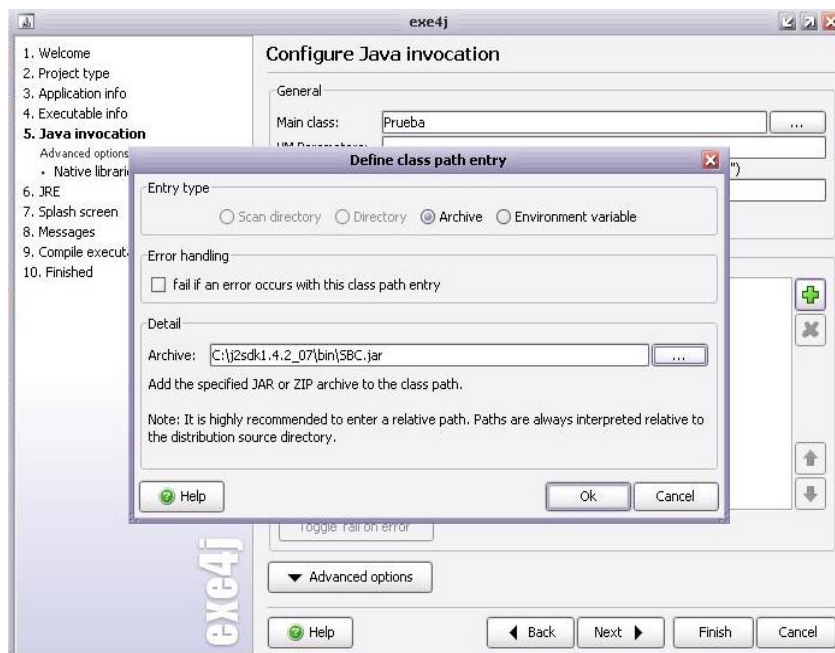


Fig. 6.



- Especificar la versión mínima y máxima de java empleada.
- Configurar splash screen (pantalla de carga de la aplicación).
- Compilar el .exe y guardar archivo de configuración para cambios posteriores.

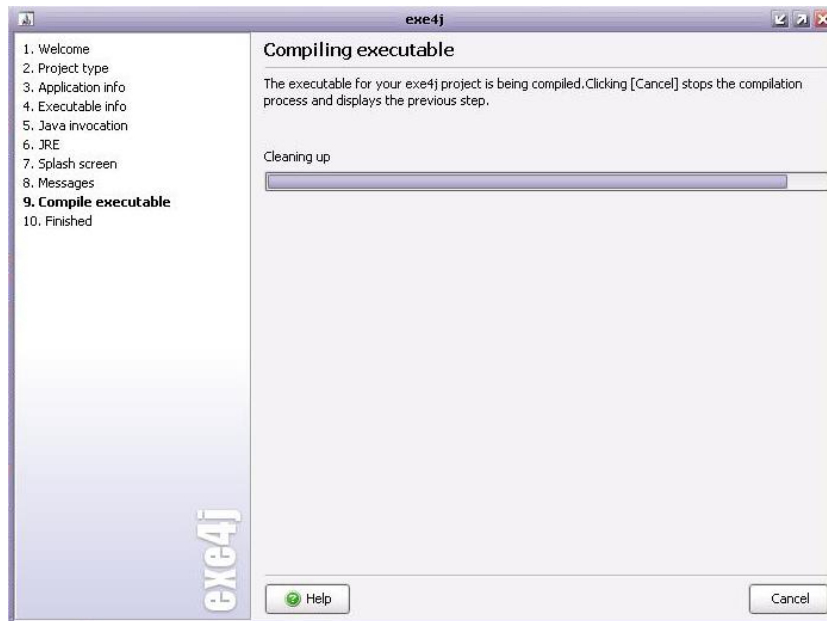


Fig. 7.

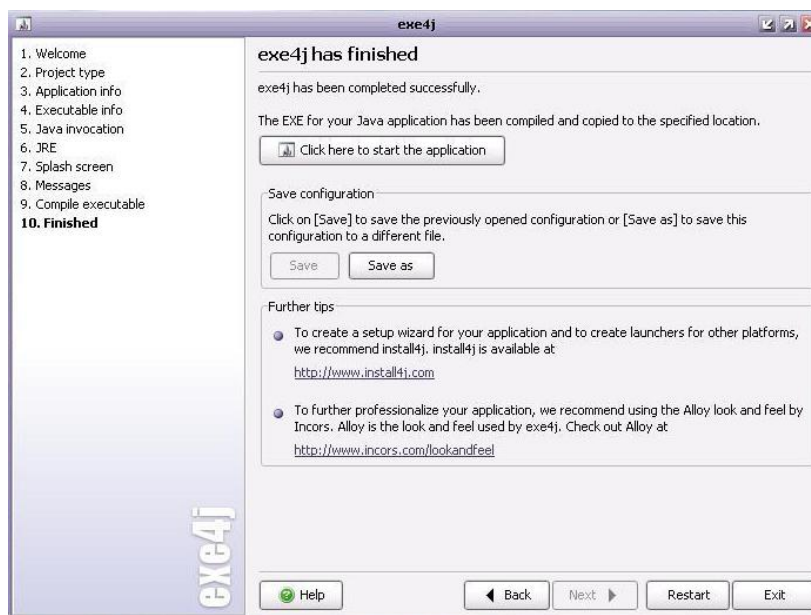


Fig. 8.



V.VI. MYSQL Y EL CÓDIGO ABIERTO

MySQL es un sistema gestor de bases de datos. Pero la virtud fundamental y la clave de su éxito es que se trata de un **sistema de libre distribución** y de **código abierto**.

Lo primero significa que se puede descargar libremente de Internet (por ejemplo de la dirección (www.mysql.com)); lo segundo (código abierto) significa que cualquier programador puede remodelar el código de la aplicación para mejorarlo. Esa es también la base del funcionamiento del sistema Linux, por eso MySQL se distribuye fundamentalmente para Linux, aunque también hay versiones para Windows.

Existen cuatro versiones de MySQL:

Estándar. Incluye el motor estándar y la posibilidad de usar bases de datos **InnoDB**. Todo el potencial de MySQL, pero sin soporte completo para utilizar transacciones.

Max. Para usuarios que quieran MySQL con herramientas de prueba para realizar opciones avanzadas de base de datos

Pro. Versión comercial del MySQL estándar

Classic. Igual que la estándar pero no dispone de soporte para **InnoDB** El uso de MySQL (excepto en la versión Pro) está sujeto a licencia **GNU public license** (llamada GPL). Esta licencia admite el uso de MySQL para crear cualquier tipo de aplicación. Se pueden distribuir copias de los archivos de MySQL, salvo esas copias se cobren a un tercer usuario. Se prohíbe cobrar por incluir MySQL. Se puede modificar el código fuente de MySQL, pero si se distribuye la aplicación con el código modificado, habrá que obtener una copia comercial y consultar sobre el cobro de la licencia. Al distribuir copias, se tiene que poder obtener información sobre las licencias GNU (más información en http://dev.mysql.com/doc/mysql/en/GPL_license.html) Se puede también obtener una licencia comercial que permitiría cobrar las instalaciones MySQL, incluir la base de datos en ordenadores y cobrar por ello, y otras situaciones no reflejadas en la licencia GNU.



Instalación de los ficheros del programa

Hay dos posibilidades de instalación en Windows.

La más sencilla es utilizar el programa de instalación, el cual colocará todos los elementos necesarios en las carpetas correspondientes. La instalación no diferirá demasiado respecto a cualquier otra instalación bajo Windows.

Instalación Manual de MySQL

Bajarse los archivos comprimidos sin la instalación. Bastará con descomprimirlos en C: Supondremos que la instalación se realiza con el primer método (que es lo más común). Normalmente el programa se instalará en la carpeta **MySQL** dentro del disco duro principal (el C:). Todos los ejecutables se encontrarán en la carpeta *bin* que estará dentro de la anterior. No habrá ningún acceso al programa desde el apartado **programas** de Windows (debemos hacerlo nosotros manualmente si lo deseamos).

Configuración del entorno

El entorno se puede configurar a mano a través de un archivo llamado `mysql.ini` que normalmente se encuentra en la carpeta **Windows** o **Winnt**. No obstante es más sencillo hacerlo con el programa **WinMySQLAdmin**. Este es un programa realizado para Windows que nos permite editar `mysql.ini` de forma más sencilla.

La primera vez que este programa arranca nos pregunta nuestro nombre de usuario y contraseña. Los usuarios de MySQL no tienen nada que ver con los de Windows. Tras indicar el usuario, el programa colocará los valores más usuales de configuración. Cada vez que volvamos a arrancar este programa, se nos permitirá cambiar la configuración.

Arrancar el servidor

Tras la instalación hace falta instalar el servidor MySQL de bases de datos. Hay varias posibilidades de servidores, son:

Archivo (en la carpeta bin) uso `mysqld` Compilado con depuración completa y chequeo automático de colocación de memoria, enlaces simbólicos, *InnoDB* y



tablas BDB. Es el más genérico (para cualquier tipo de ordenador)

mysqld-opt Optimizado para equipos Pentium. No tiene soporte de tablas transaccionales.

mysqld-nt Optimizado para equipos con Windows NT, 2000 y XP con soporte para *pipe names*

mysqld-max Optimizado con soporte de enlaces simbólicos, *InnoDB* y tablas BDB.

mysqld-max-nt Igual que el anterior pero combinado las capacidades del servidor

mysqld-nt Antes de seleccionar el servidor a arrancar hay que asegurarse de cerrar el servidor que estuviera en funcionamiento. Esto se realiza (desde la línea de comandos) con el nombre del servidor, seguido de la palabra **--remove**. Ejemplo:
c:\mysql\bin>mysqld --remove Para arrancar el servidor deseado, se coloca la palabra **--install**. Ejemplo:

```
c:\mysql\bin>mysqld-max-nt --install
```

Si está bien instalado. Aparecerá en la línea de comandos el texto **“Service successfully installed”**.

Otra posibilidad es utilizar el administrador de MySQL y desde él decidir cómo arrancar el servidor. Eso se realiza en la línea **Server** dentro de la sección

[WinMySQLAdmin]. Ejemplo:

```
[WinMySQLAdmin]
```

```
Server=C:/mysql/bin/mysqld-max-nt.exe
```

El servidor se ejecuta en segundo plano, sólo el administrador de tareas permite ver su ejecución. No conviene parar el servidor desde el administrador de tareas ya que al cortar de repente los datos de seguridad podrían quedar malparados. En su lugar conviene cerrar desde la línea de comandos con el parámetro **--remove** comentado anteriormente.

Conexión y desconexión al servidor

La conexión al servidor MySQL para crear, modificar o realizar cualquier otra operación sobre bases de datos. Se realiza mediante el programa **mysql** que también se encuentra en la carpeta **bin** del programa. La ejecución de este



programa nos lleva al llamado **monitor** de MySQL que es la línea de comandos desde la que podemos ejecutar instrucciones MySQL.

Este programa posee numerosas opciones de arranque. Para saber cuáles son, basta ejecutar la instrucción `mysql --help` desde la línea de comandos.

Normalmente la entrada normal al monitor se hace mediante:

```
c:\mysql\bin> mysql -h host -u user -p
```

Donde *host* es el nombre del servidor (si no se ejecuta desde red será *localhost*) y *user* es el nombre del usuario que desea utilizar el servidor. El último parámetro sirve para poder introducir la contraseña. Ésta será pedida inmediatamente después de haber sido pulsada la tecla Intro. En la mayoría de instalaciones (sobre todo en Windows) se puede entrar simplemente poniendo **mysql** desde la línea de comandos. Esto asumirá como usuario anónimo y como servidor el servidor local. Hay un usuario de privilegios absolutos que se llama **root**.

En cualquier caso si hemos accedido bien al monitor, en la línea de comandos aparecerá el texto **mysql>**. Que indicará que el monitor está esperando comandos para ejecutar sobre el servidor. Para abandonar el monitor basta escribir el comando **quit**.

Una vez instalado y configurado para poder establecer conexión desde java necesitamos copiar el archivo `mysql-connector-java` que es un archivo de extensión .jar en el directorio `C:\j2sdk1.4.2_07\jre\lib\ext` aquí es donde se encuentra el paquete de librerías para que las aplicaciones java puedan ejecutarse. Este archivo es el controlador de base de datos para MySQL por lo tanto si utilizamos otra base de datos obligatoriamente debemos cambiar el controlador; cada base de datos según el fabricante necesita un controlador, por tanto para establecer conexión a una base de datos con Microsoft Access necesitamos este controlador que ya está integrado en Windows lo que debemos hacer es asignárselo a cada base de datos Access que deseemos conectar con java, para hacer esto se siguen los siguientes pasos:



- Ubicarse en el panel de control de Windows.
- Dar click en herramientas administrativas.
- A continuación abrir orígenes de datos.



Fig. 9.

Se observara una ventana con múltiples pestañas, seleccionando DNS usuario y dar click en el botón agregar aparece otra ventana con todos los controladores disponibles se procede seleccionado el adecuado y click en el botón finalizar; en la nueva ventana seleccionar la base de datos que desea y aceptar, listo de esta manera la base de datos esta disponible para establecer conexión con Access.

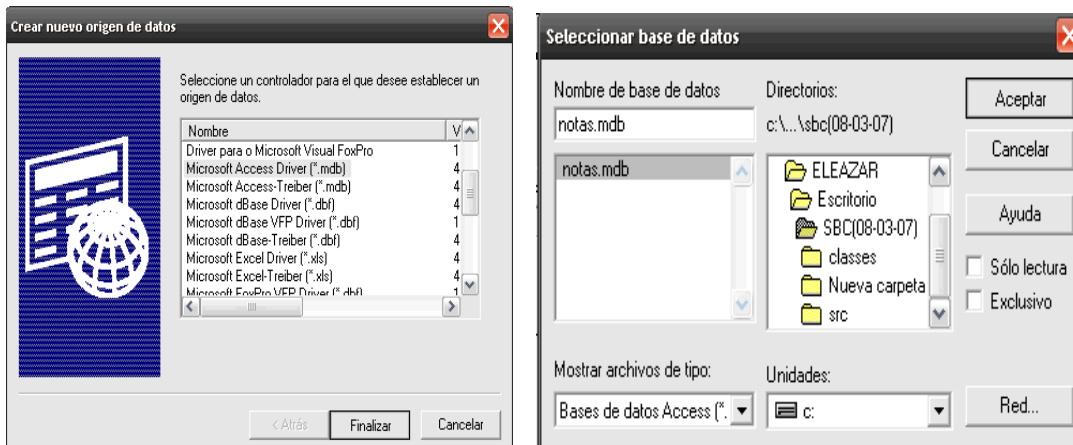


Fig.10.



V.VII. E.R.S.

1- Introducción.

La especificación de requisitos Software es el establecimiento conciso de un conjunto de requisitos, que deben ser satisfechos por un producto o un proceso, indicando siempre que sea adecuado, el procedimiento mediante el cual se puede determinar, si se han logrado satisfacer estos requisitos.

1.1- Propósito

Definir detalladamente y ordenadamente los requisitos del software que debe cumplir al aplicación; “Sistema de selección y control de Becas en la Facultad de Ciencias de la UNAN – León.” que consiste en facilitar los registros de solicitudes de beca y determinar a partir de estas las becas otorgadas apoyándose en los registros existentes de índice académico.

Este documento esta dirigido a la vicedecanatura de la facultad de ciencias de la UNAN-León.

1.2- Alcance

Esta aplicación se conocerá con el nombre de:

“Sistema de selección y control de Becas en la Facultad de Ciencias de la UNAN – León.” y se realizará las siguientes funciones:

- Registrar estudiantes becados
- Registrar solicitudes de beca.
- Registrar nuevos becados
- Hacer búsqueda de estudiantes.
- Generar diferentes tipos de consultas.
- Determinar las bajas de becas.



1.3- Definición de acrónimos y abreviaturas.

Becas:

Ayuda que brinda la universidad, con el objetivo de que el estudiante pueda culminar sus estudios.

Estudiante:

Entidad que representa al individuo, que realiza estudios universitarios y puede optar a una beca.

Secretaría:

Entidad que proporciona los registros académicos de cada estudiante.

Vicedecanatura:

Entidad encargada de la selección, asignación y control de becas a estudiantes.

Altas:

Representa una beca asignada a un determinado estudiante.

Bajas:

Representa la pérdida de una beca por un estudiante que la poseía.

Carnet:

Dato con el que se identifica a un estudiante en la universidad y en el sistema.

Beca interna:

Representa una beca, en la cual la universidad proporciona alimentación, hospedaje, transporte y una pequeña ayuda para gastos a un estudiante.



Beca externa:

Representa una beca, en la cual la universidad, asigna un estipendio monetario mensual, para gastos universitarios.

Beca Residencia:

Es una beca en la cual se da alojamiento al estudiante, si proporcionarle alimentación.

Bonos de alimentación:

Es una beca, donde se le brinda alimentación al estudiante.

Beca por Excelencia:

Esta beca se la gana un estudiante que alcanza un promedio mínimo de 90%, y puede ser dada a estudiantes incluso que tienen otra beca.

Transporte:

Representa un tipo de beca en que la Universidad proporciona el pasaje a estudiantes que viajan diario fuera de la ciudad de León.

Usuario:

Operador del sistema de control de becas, en este caso Vicedecanatura.

Rendimiento académico:

Parámetro tomado en cuenta para valorar la asignación o no de una beca.

1.4- Referencias

Investigaciones preliminares desarrolladas.



1.5- Visión general

Se describirá de manera general la aplicación a desarrollar para después analizar cada uno de los requisitos específicos individuales.

2- Descripción general.

2.1- Relaciones del producto.

El sistema interactúa con una base de datos remota que lleva registro de notas de los estudiantes.

El equipo en el que se desarrollará el producto es:

HP Pentium IV

256 MB de RAM.

20 GB de disco duro.

2.2- Funciones del producto.

El producto resultante debe contener todas las tareas que se realizan manualmente por Vicedecanatura con respecto a las altas y bajas de becas:

- 1- Registrar las solicitudes de becas, introduciendo los datos existentes en los formatos de solicitud.
- 2- Una vez registrada las solicitudes la aplicación determinará los posibles estudiantes que ameritan beca según la prioridad del mismo mostrando por pantalla a estos en lista ordenada de manera jerárquica, esto permite modificar de manera manual la asignación de beca por el sistema así definiendo de manera definitiva todas las becas. Las listas contendrán carnet, nombre, carrera, tipo de beca, índice académico.
- 3- Teniendo definida todas las becas se procede a almacenar estos registros.
- 4- Si se desea buscar información acerca de un estudiante en especial el usuario introducirá el número de carnet o el nombre del estudiante según deseé. La aplicación responderá mostrando información del alumno. (carnet, nombre, carrera, tipo de beca, índice académico, prioridad.).



5- Se desea realizar alguna consulta, seleccionar la opción entre las que tenemos.

* Becas externas

-Carrera.

-Todas.

* Beca interna

-Carrera.

-Todas.

* todas.

6- Se mostrará una opción para generar las bajas de todas las becas e informes de las mismas.

-2.2.1 Aspectos de funcionamiento

a. Para crear un nuevo registro se debe proceder a introducir datos de los estudiantes según sea becado o solicitante.

b. Para hacer cambios en el estado de un registro, introducir el carnet del estudiante y modificarlo.

c. El proceso de baja de estudiante becado es automático lo único necesario es dar la orden al sistema.

d. Para mostrar los diferentes estados de los estudiantes registrados, se procede haciendo consultas y búsqueda según el caso.

e. La forma de acceso al sistema está restringido por contraseña, la cual puede ser cambiada por usuario dentro del sistema cuando lo desee.

f. El usuario puede hacer respaldo de la Base de Datos cuando lo amerite.

-2.2.2 Restricciones técnicas y de gestión

El acceso a las notas se hace a una base de datos externa, ajena al sistema, por tanto se plantea una representación de la misma, esto es: el software obtiene datos (notas) que usa al momento de generar bajas y mostrar situación académica al consultar solicitud.



Debido a los cambios en la nueva administración de notas por parte de la Secretaría Académica de la Facultad de Ciencias, nuestro sistema debe adecuarse a las nuevas modificaciones de la base de datos.

2.3- Características del usuario.

Los usuarios finales deberían tener un conocimiento mínimo sobre computación no de vital importancia

2.4- Restricciones generales.

El lenguaje de programación a utilizar será Java implementando P.O.O. mediante conexión a base de datos con mySQL.

El equipo donde se implementará

HP Pentium IV.

512 MB de RAM.

80 GB de disco duro.

2.5- Suposiciones y dependencias.

3- Requisitos Específicos:

3.1- Requisitos Funcionales.

3.1.1- Registro de estudiantes

3.1.1.1- Especificación.

3.1.1.1.1- Introducción:

Este proceso realiza la captura de los datos de los estudiantes ya se solicitante o becado para proceder almacenándolo en la base de datos haciendo las validaciones necesarias.

3.1.1.1.2- Entradas:

Por pantalla: Carnet, Nombre, Carrera, Año académico, sexo, Tipo de beca, Fecha, Departamento, Municipio, Comarca, Dirección en León, promedio y estado que permite verificar si desea registrar un becado o solicitante.



En caso de ser solicitante debe llenar los datos de la siguiente ventana que son los siguientes:

Fecha emisión, fecha de nacimiento, Bachillerato, Estado civil, Numero de hijos, Numero de hermanos, Teléfono, gozó de beca, Repitió año, Dirección origen, Casa de origen, Apoyo económico, En que trabaja, Salario, Asignación mensual, Trabaja padre, Salario, Donde, Actividad, Trabaja madre, Salario, Donde, Actividad, Dependientes del apoyo, Deuda familiar, Acreedor, Monto, Bienes inmuebles, Mencionarlos, Trabaja usted, Donde, Salario, Actividad, Ayuda exterior, Cantidad, Propiedad padre, Detalle, Extensión, Vehículo padre, Placa, Marca, Becado organización, Cantidad, Organización, Causas que le impiden sus estudios, Tipo beca.

3.1.1.1.3-Proceso:

Introducir datos pedidos por pantalla y proceder a almacenarlos, en el caso de carnet introducir el formato correcto, el nombre de el estudiante no puede ser un campo vacío de lo contrario el sistema no procede, la selección del municipio se hace de acuerdo al departamento seleccionado anterior los demás datos son introducidos de acuerdo a la selección del usuario o a lo que escriba.

3.1.1.1.4- Salidas:

Una vez introducida la información según sea el caso se guarda en la base de datos de acuerdo a la acción del usuario.

3.1.1.2 - Interfaces externas

3.1.1.2.1- Interfaz de usuario

La captura de los datos de solicitud se hace interactiva por pantalla.

3.1.1.2.2- Interfaz hardware

Se usará cualquier terminal conectado al ordenador central.

3.1.1.2.3- Interfaces de comunicaciones

No existen interfaces de comunicaciones.



3.1.2- Registro de altas de becas

3.1.2.1- Especificación.

3.1.2.1.1- Introducción:

Este proceso registra los solicitantes que se les asigna beca.

3.1.2.1.2- Entradas

Por pantalla:

Carnet: numero de carnet del estudiante.

Tipo de beca: la que se le asigna al estudiante.

3.1.2.1.3-Proceso:

Introducir datos pedidos por pantalla y proceder a almacenarlos, en el caso de carnet introducir el formato correcto del mismo y con existencia previa en la base de datos para que el sistema pueda proceder a almacenarlo, la selección del tipo de beca se hace de acuerdo a la que desee asignar.

3.1.2.1.4- Salidas:

Una vez introducida la información según sea el caso, el sistema verifica respondiendo si es un dato valido existente en las solicitudes de ser así se guarda en la base de datos de acuerdo a la acción del usuario.

3.1.2.2 - Interfaces externas

3.1.2.2.1- Interfaz de usuario

La captura de los datos de solicitud se hace interactiva por pantalla.

3.1.2.2.2- Interfaz hardware

Se usará cualquier terminal conectado al ordenador central.

3.1.2.2.3- Interfaces de comunicaciones

No existen interfaces de comunicaciones.

3.1.3- Generar bajas de becas

3.1.3.1- Especificación.

3.1.3.1.1- Introducción:

Este proceso da de baja a los estudiantes con mal rendimiento académico.

3.1.3.1.2- Entradas



Por pantalla: visualización de estudiantes becados.

3.1.3.1.3-Proceso:

Accionar generar bajas, luego ver bajas de becas para mostrar lo estudiantes que pierden beca por razones académicas

3.1.3.1.4- Salidas:

Visualización por pantalla de todos los estudiantes que pierden becas.

3.1.3.2 - Interfaces externas

3.1.3.2.1- Interfaz de usuario

Cambio de estado en la base de datos del estudiante becado

3.1.3.2.2- Interfaz hardware

Se usará cualquier terminal conectado al ordenador central.

3.1.3.2.3- Interfaces de comunicaciones

No existen interfaces de comunicaciones.

3.1.4- Cambiar contraseña de acceso

3.1.4.1- Especificación.

3.1.4.1.1- Introducción:

Proceso que permite modificar la contraseña de acceso al sistema.

3.1.4.1.2- Entradas

Por pantalla:

Contraseña actual: contraseña que uso para acceder.

Nueva contraseña: Nueva contraseña a usar.

3.1.4.1.3-Proceso:

Modificación del archivo que guarda la contraseña, almacenando la nueva.

3.1.4.1.4- Salidas:

Creación de una nueva contraseña.

3.1.4.2 - Interfaces externas

3.1.4.2.1- Interfaz de usuario

Cambio de contraseña

3.1.4.2.2- Interfaz hardware



Se usará cualquier terminal conectado al ordenador central.

3.1.4.2.3- Interfaces de comunicaciones

No existen interfaces de comunicaciones

3.1.5- Respaldo de base de datos

3.1.5.1- Especificación.

3.1.5.1.1- Introducción:

Proceso que permite hacer una copia de la base de datos de la aplicación.

3.1.5.1.2- Entradas

No se aplica

3.1.5.1.3-Proceso:

Selección de la opción Respaldar BD.

3.1.5.1.4- Salidas:

Base de datos respaldada en la ubicación mostrada por el sistema.

3.1.5.2 - Interfaces externas

3.1.5.2.1- Interfaz de usuario

Acción respaldar base de datos

3.1.5.2.2- Interfaz hardware

Se usará cualquier terminal conectado al ordenador central.

3.1.5.2.3- Interfaces de comunicaciones

No existen interfaces de comunicaciones

3.1.6- Visualizar solicitudes

3.1.6.1- Especificación.

3.1.6.1.1- Introducción:

Proceso que muestra los datos de los estudiantes que solicitan beca.

3.1.6.1.2- Entradas

Por pantalla: No se aplica.

3.1.6.1.3-Proceso:



Selección de categoría que desea ver, según el tipo de beca que solicita.

3.1.6.1.4- Salidas:

Visualización de los datos solicitados y otros referentes al estado académico del estudiante como cursos de verano pendiente (CV), examen especial (EE), Tutorías (TT).

3.1.6.2 - Interfaces externas

3.1.6.2.1- Interfaz de usuario

3.1.6.2.2- Interfaz hardware

Se usará cualquier terminal conectado al ordenador central.

3.1.6.2.3- Interfaces de comunicaciones

Comunicación con la base de datos notas de los estudiantes.

3.1.7- Consultas

3.1.7.1- Especificación.

3.1.7.1.1- Introducción:

Proceso que muestra por pantalla datos que desea visualizar de los registros de la base que almacena la becas datos Becas.

3.1.7.1.2- Entradas

Selección del tipo de consulta que desea hacer, ya sea si el estudiante es becado seleccionar el tipo de beca, o el año que cursa, o la carrera que estudia y consulta por origen ya sea becado o solicitante, seleccionando el origen que desea ver.

3.1.7.1.3-Proceso:

Accionar para mostrar la consulta que selecciono.

3.1.7.1.4- Salidas:

Visualización de los datos solicitados.

3.1.7.2 - Interfaces externas

3.1.7.2.1- Interfaz de usuario

3.1.7.2.2- Interfaz hardware.

Se usará cualquier terminal conectado al ordenador central.



3.1.7.2.3- Interfaces de comunicaciones

Comunicación con la base de datos notas de los estudiantes.

3.1.8- Búsqueda

3.1.8.1- Especificación.

3.1.8.1.1- Introducción:

Proceso que permite hacer búsqueda de un estudiante específico ya sea por carnet o nombre, y el estado del estudiante.

3.1.8.1.2- Entradas

Selección del tipo de búsqueda, introducción de nombre o carnet del registro a buscar.

3.1.8.1.3-Proceso:

Seleccionar nombre o carnet e introducir dato correspondiente, seleccionar búsqueda entre becados, solicitantes y bajas, proceder accionando búsqueda.

Si la búsqueda es por nombre, permite el desplazamiento entre los registros, deshabilitándolo si se encuentra en el primer o el último, según aplique.

3.1.8.1.4- Salidas:

Visualización por pantalla del dato encontrado o mensaje si no se encuentra.

3.1.8.2 - Interfaces externas

3.1.8.2.1- Interfaz de usuario

3.1.8.2.2- Interfaz hardware.

Se usará cualquier terminal conectado al ordenador central.

3.1.8.2.3- Interfaces de comunicaciones

Comunicación con la base de datos notas de los estudiantes.

3.2- Requisitos de funcionamiento.

Requisitos estáticos: habrá un solo usuario y una sola maquina para trabajar con el sistema.



3.3- Atributos.

- Seguridad:

El sistema estará protegido por una clave de Acceso para obtener la máxima seguridad de la información.

- Mantenimiento:

Las posibles modificaciones que se hagan en los requisitos o funciones del sistema expuestos en este documento se reflejarán en el mismo, así como la nueva documentación obtenida en las fases de análisis, diseño y programación.

4- Otros Requisitos.

- Base de Datos:

El almacenamiento de la información se realizará por medio de un administrador de Base de Datos MYSQL.

- Operaciones:

La primera operación será crear la clave de acceso, para ingresar al sistema. Realizaremos menú de opciones para la introducción de la información y visualización de todo tipo de consulta o búsqueda, así como respaldo de Base de Datos y cambio de contraseña.



V.VIII. DIAGRAMA DE FLUJO DE DATOS

NIVEL 0.

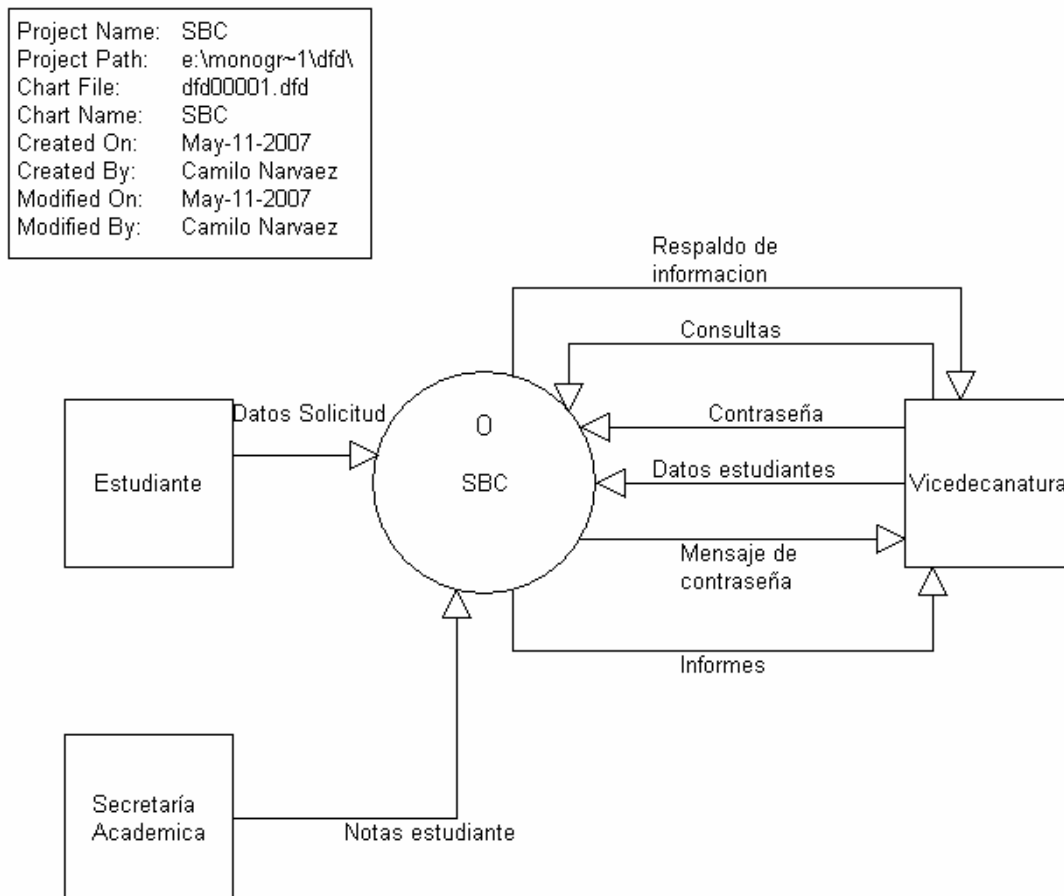


Fig.11.

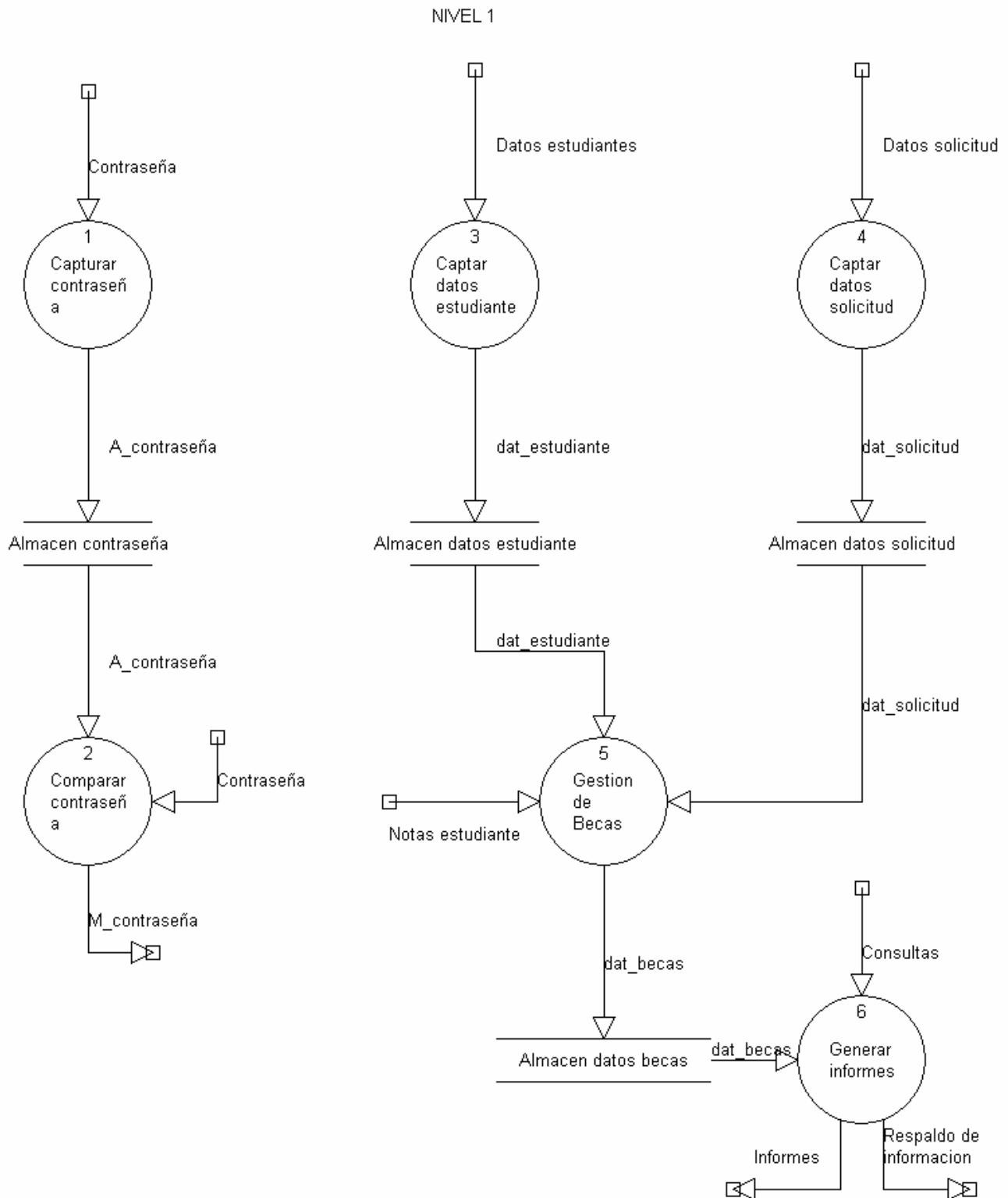


Fig.12.

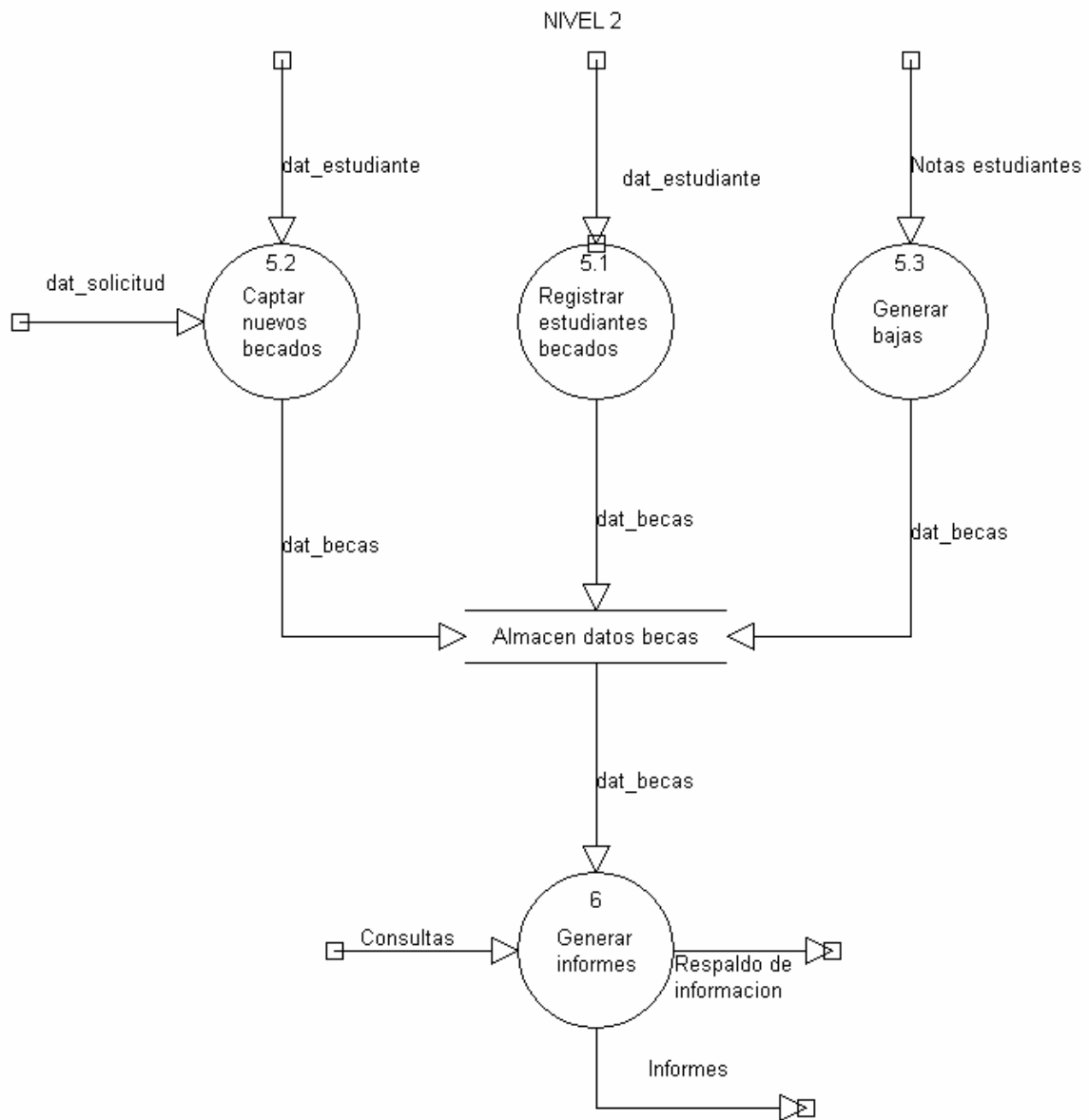


Fig.13.



V.IX. DICCIONARIO DE DATOS.

- dat_becas = Datos estudiante + Datos solicitud + Notas estudiantes.
- Datos estudiante = Carnet + Nombre + Carrera + Anyo + Sexo + Departamento + Municipio + Comarca + Direccion + Situacion_beca.
- Datos solicitud = Fecha emision + fecha de nacimiento + Bachillerato + Estado civil + Numero de hijos + Numero de hermanos + Teléfono + gozó de beca + Repitió año + Dirección origen + Casa de origen + Apoyo económico + En que trabaja + Salario + Asignación mensual + Trabaja padre + Salario padre + Donde trabaja padre + Actividad padre + Trabaja madre + Salario madre + Donde trabaja madre + Actividad madre + Dependientes de apoyo + Deuda familiar + Acreedor + Monto + Bienes inmuebles + Tipo de bienes + Trabaja usted + Donde trabaja usted + Salario usted + Actividad usted + Ayuda exterior + Cantidad + Propiedad padres + Detalle + Extensión + Vehículo padres +Placa + Marca + Becado organización + Cantidad beca + Nombre Organización + Causas que le impiden sus estudios + Tipo beca.
- Consultas = Consultas solicitud = Consultas Becados = Consulta por origen = Consulta por Carrera = Consulta por tipo de beca.
- Almacen contraseña = A_contraseña.
- Almacen datos becas = dat_becas.
- Almacen datos estudiante = Datos estudiante.
- Almacen datos solicitud = Datos solicitud.



V.X. DIAGRAMA ENTIDAD – RELACION (ER)

La estructura lógica de la Base de Datos puede representarse gráficamente por medio de un diagrama ER.

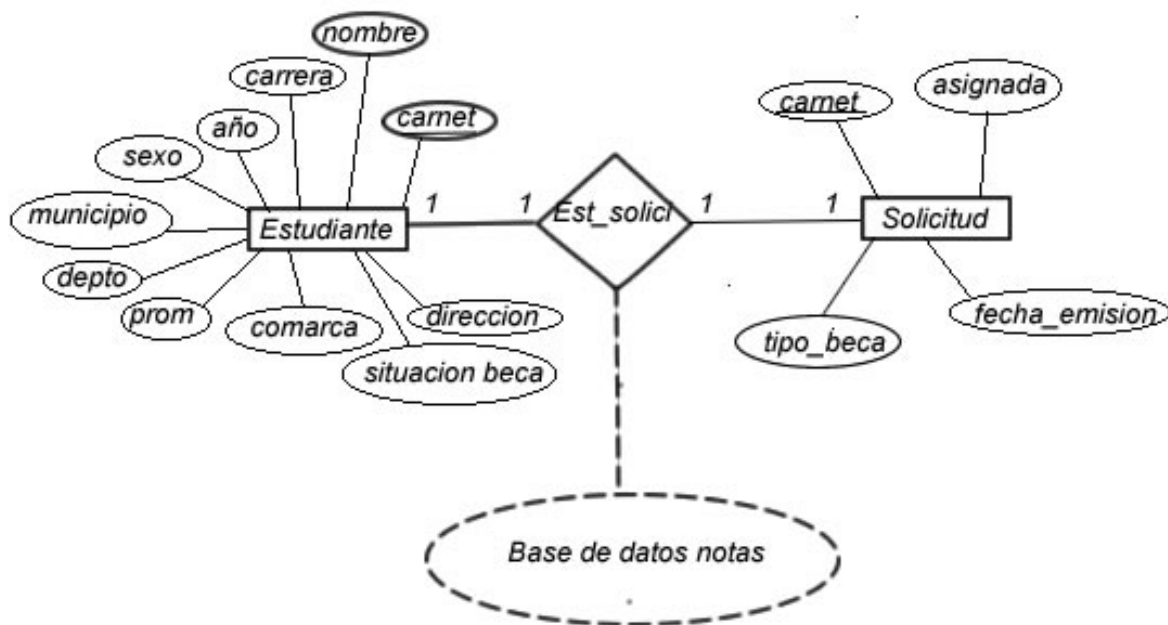


Fig.14.

Al concluir este trabajo, la base de datos donde se encuentran las notas, estaba desarrollada en MS ACCESS la cual presentaba limitaciones, por esto no se representa el diagrama ER de la misma y se simula un funcionamiento similar, ajustándose a los requisitos. Por tanto en secciones posteriores se explica la conexión a BD Access, que no es software libre.



V.XI. DISEÑO ESTRUCTURAL

Tabla estudiante

```
C:\WINDOWS\system32\cmd.exe - mysql -u root -h localhost -p
mysql>
mysql> describe estudiante;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Carnet         | varchar(10)   | NO   | PRI |          |       |
| Nombre        | varchar(50)   | NO   |     |          |       |
| Carrera       | varchar(24)   | NO   |     |          |       |
| Anyo          | varchar(18)   | NO   |     |          |       |
| Promedio      | float         | YES  |     | NULL    |       |
| Sexo         | varchar(18)   | NO   |     |          |       |
| Departamento  | varchar(18)   | NO   |     |          |       |
| Municipio     | varchar(24)   | NO   |     |          |       |
| Comarca       | varchar(24)   | YES  |     | NULL    |       |
| Direccion     | varchar(80)   | YES  |     | NULL    |       |
| Situacion_beca | tinyint(1)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)
mysql> _
```

Fig.15.



TABLA SOLICITUD

```
C:\WINDOWS\system32\cmd.exe - mysql -u root -h localhost -p
mysql> describe solicitud;
```

Field	Type	Null	Key	Default	Extra
Fecha_nacimiento	date	YES		NULL	
Bachillerato	varchar(24)	YES		NULL	
Estado_civil	varchar(18)	YES		NULL	
Gozo_beca	tinyint(1)	YES		NULL	
Hijos	int(11)	YES		NULL	
Hermanos	int(11)	YES		NULL	
Telefono	varchar(45)	YES		NULL	
Recibe_apoyo_econ_de	varchar(32)	YES		NULL	
Trabajo_apoyo_econ	varchar(32)	YES		NULL	
Carnet	varchar(10)	NO	PRI		
Salario	float	YES		NULL	
Repitio	tinyint(1)	YES		NULL	
Casa_origen	varchar(20)	YES		NULL	
Trabaja_madre	tinyint(1)	YES		NULL	
Salario_madre	float	YES		NULL	
Lugar_trabajo_madre	varchar(50)	YES		NULL	
Actividad_madre	varchar(50)	YES		NULL	
Trabaja_padre	tinyint(1)	YES		NULL	
Salario_padre	float	YES		NULL	
Lugar_trabajo_padre	varchar(50)	YES		NULL	
Actividad_padre	varchar(50)	YES		NULL	
Propiedades_padres	tinyint(1)	YES		NULL	
Detalle_propiedad	varchar(50)	YES		NULL	
Extension_propiedad	varchar(50)	YES		NULL	
No_dependientes	int(11)	YES		NULL	
Uchiculos_padres	tinyint(1)	YES		NULL	
Placa_vehiculo	varchar(10)	YES		NULL	
Marca_vehiculo	varchar(20)	YES		NULL	
Deuda_familiar	tinyint(1)	YES		NULL	
Monto_deuda	float	YES		NULL	
Adeudor	varchar(50)	YES		NULL	
Bienes_inmuebles	tinyint(1)	YES		NULL	
Especificar_bienes	varchar(50)	YES		NULL	
Trabaja_ud	tinyint(1)	YES		NULL	
Salario_ud	float	YES		NULL	
Lugar_trabajo_ud	varchar(50)	YES		NULL	
Actividad_ud	varchar(50)	YES		NULL	
Ayuda_extranjero	tinyint(1)	YES		NULL	
Cantidad_ayuda	float	YES		NULL	
Beca_organismo	tinyint(1)	YES		NULL	
Cantidad_beca	float	YES		NULL	
Nombre_organismo	varchar(50)	YES		NULL	
Causas_impiden_estudiar	varchar(200)	YES		NULL	
Asignada	tinyint(1)	YES		NULL	
Fecha_emision	date	YES		NULL	
Tipo_beca	varchar(45)	NO			

```
46 rows in set (0.02 sec)
mysql>
```

Fig.16.



V.XII. DISEÑO ARQUITECTONICO

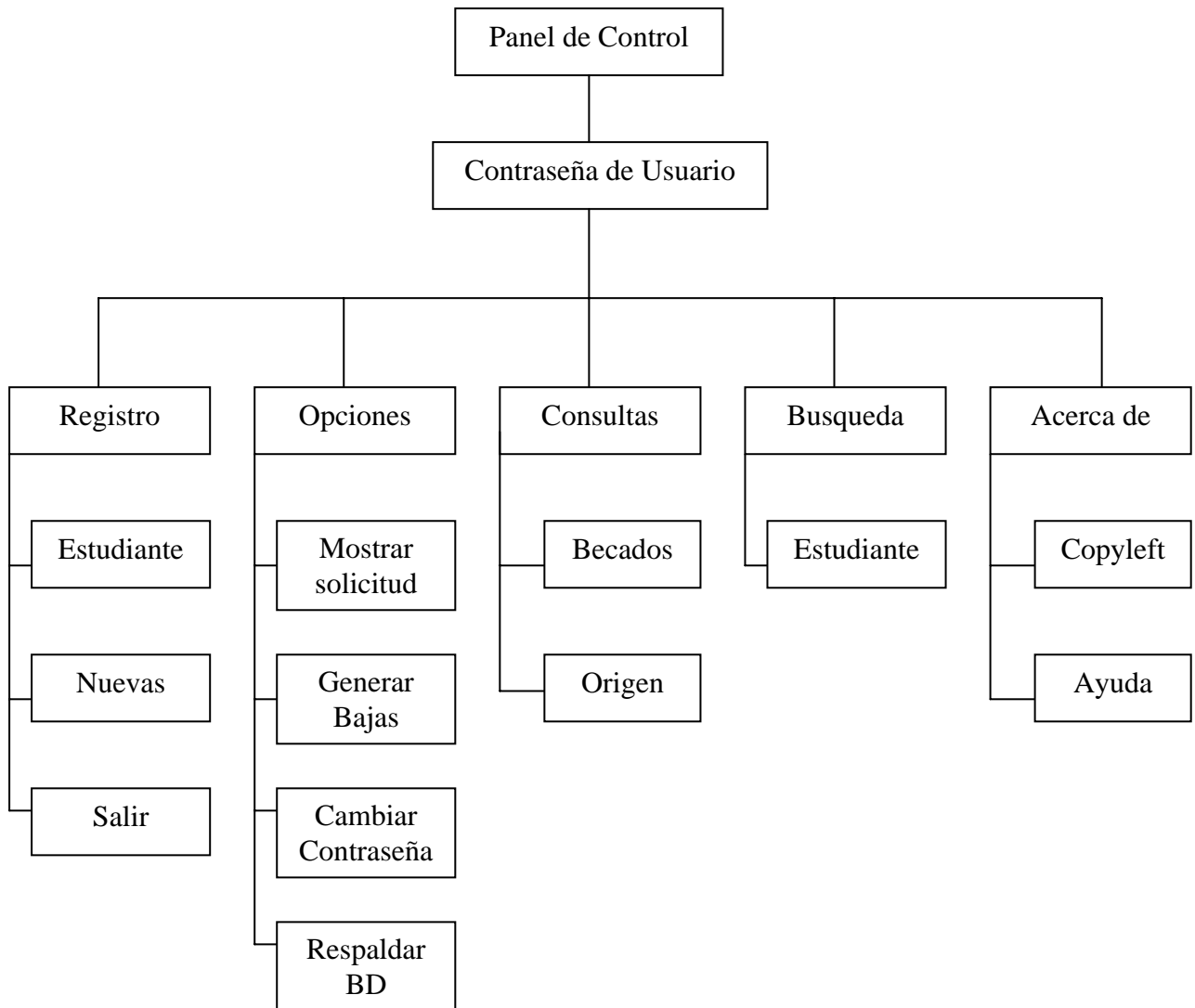


Fig.17.



V.XIII. DISEÑO PROCEDIMENTAL

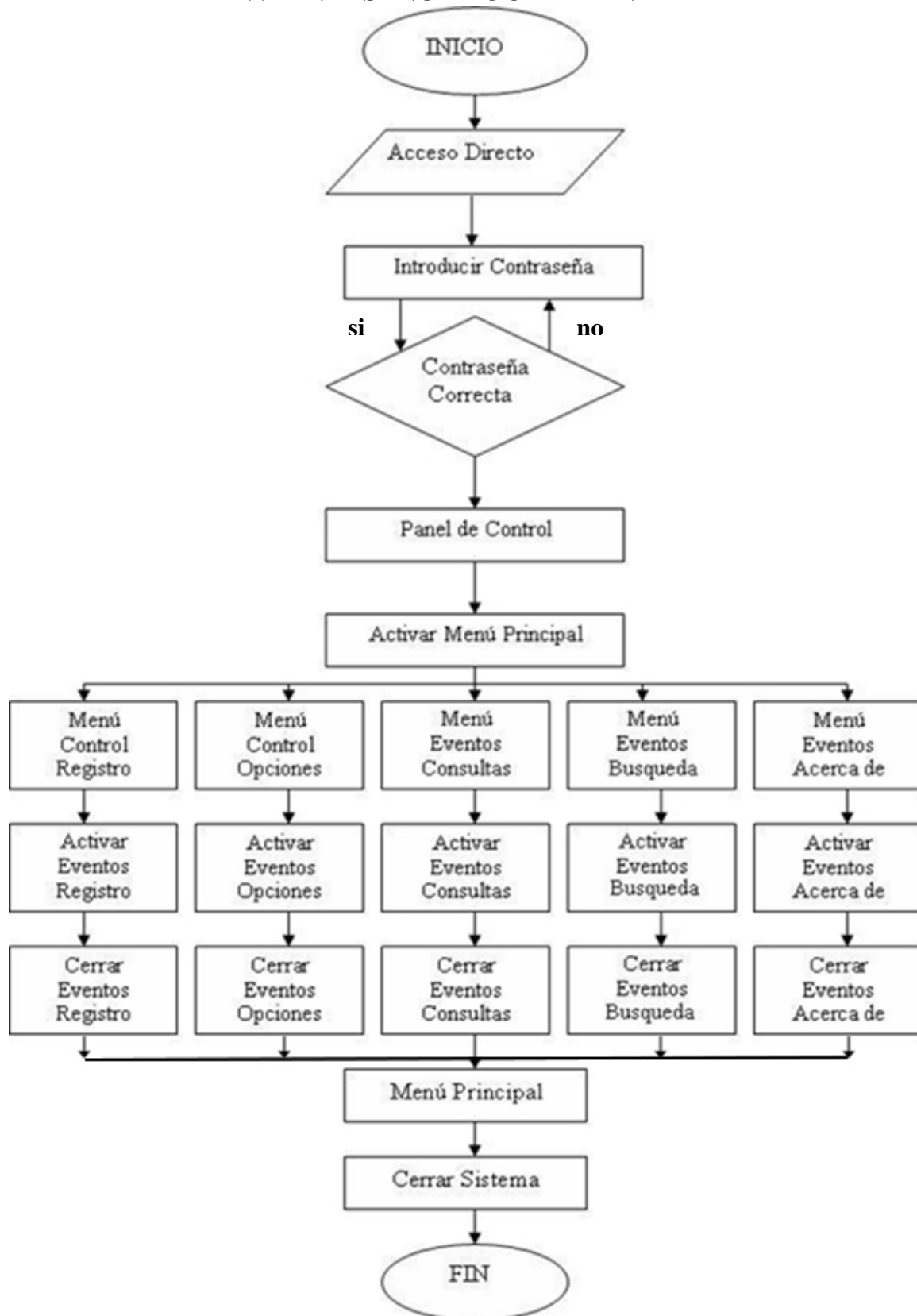


Fig.18.



V.XIV. DIAGRAMA DE CLASES

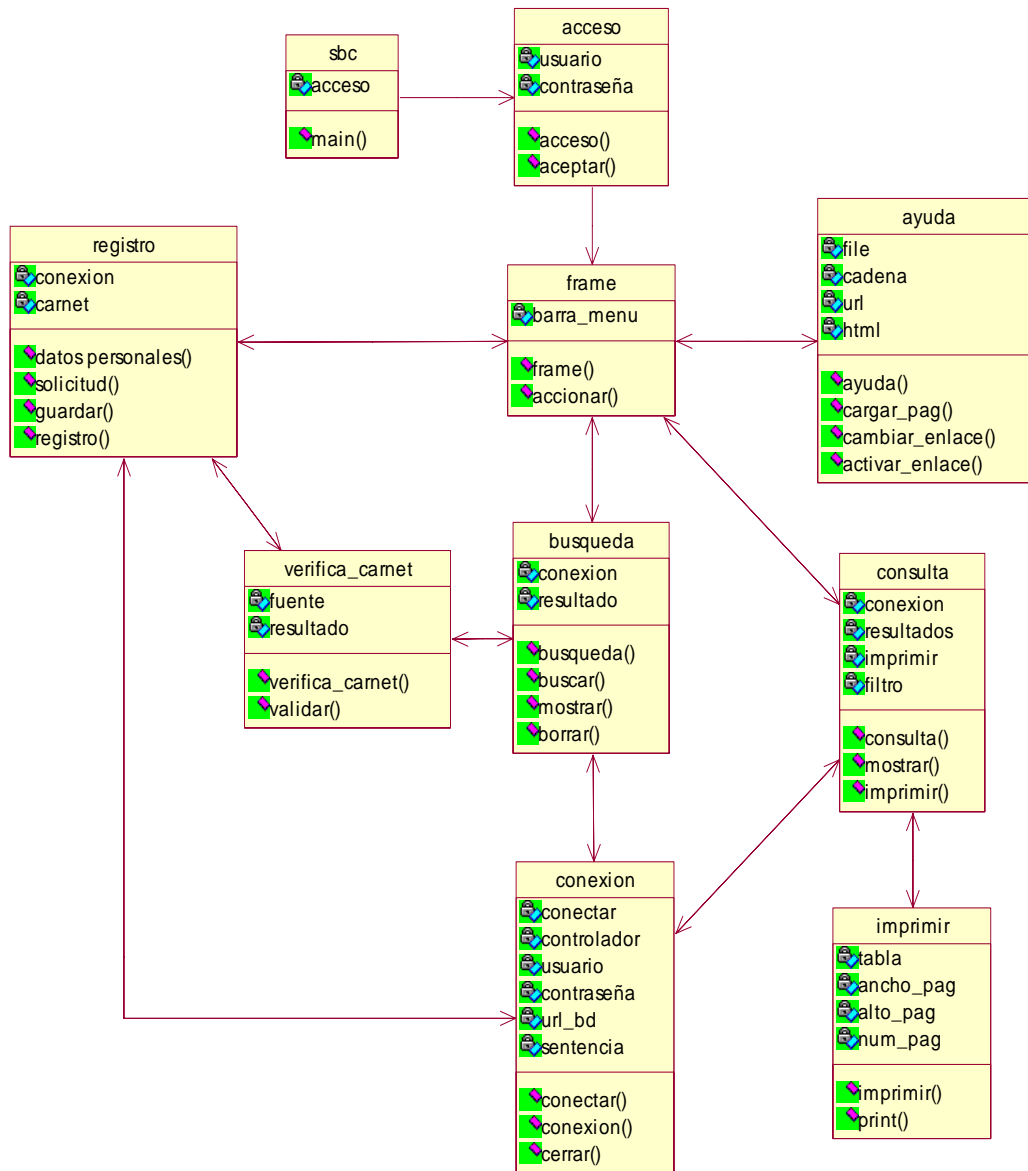


Fig. 19.



V.XV. DISEÑO DE INTERFAZ DE USUARIO

Formulario Control de Acceso

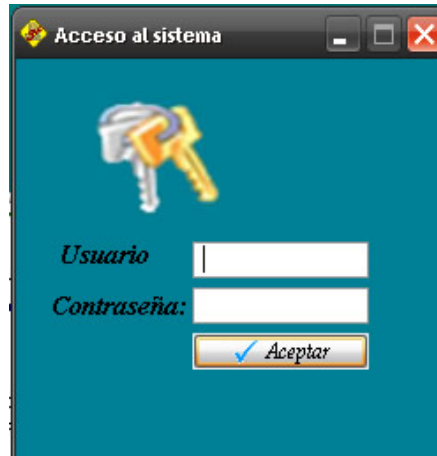


Fig.20.

Formulario cambiar Contraseña

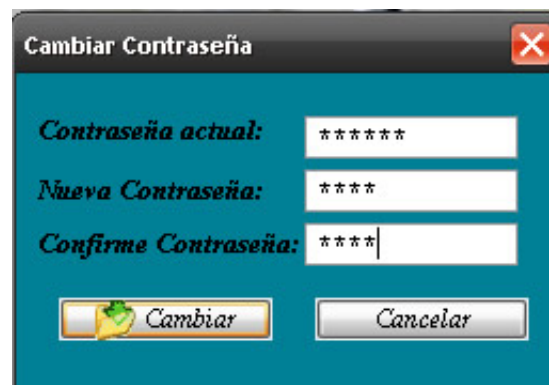


Fig.21.



MENU PRINCIPAL



Fig.22.



FORMULARIO REGISTRO ESTUDIANTES

Estudiantes Becados Facultad de ciencias

Carnet

Nombre

Carrera Sistemas

Año Acad. 1 *Sexo* Masculino

Tipo de beca Externa *Situación:*
 Becado
 Solicitante

Procedencia

Departamento Boaco

Municipio Boaco

Comarca

Dir. León:

Promedio

Fig. 23.

FORMULARIO NUEVOS BECADOS

Nuevos Becados

Carnet

Tipo de beca Externa

Fig. 24.



REGISTRO DE SOLICITUDES

Solicitud de beca Facultad de ciencias

Fecha:
Fecha Nac.
Colegio Br.
Estado civil: Gozo de beca: Si No
No hijos
No hermanos
Teléfono
Tipo de beca:

Recibe apoyo económico de:
En que Trabaja esa Persona:
Salario:
Becado por un organismo si No
Cuanto
Organismo

Marque causas que impiden estudiar

- Muerte del principal apoyo economico del grupo familiar
- Problemas emocionales
- Orfandad
- Problemas alcoholismo
- Separacion grupo Familiar
- problemas de salud
- Malas reacciones familiares
- Problemas de desempleo
- Desintegracion familiar
- Problemas economicos

Ha Repetido: Si No
Casa de origen es: Propia Alquilada Familiar Otro
Dependientes de su apoyo:
Propiedades de sus padres: Si No
Detalle
Extensión
Trabaja su padre: Si No
Salario:
Donde:
Actividad:
Trabaja su mamá: Si No
Salario:
Donde:
Actividad:
Poseen vehiculos padres: Si No
Placa
Marca
Trabaja usted: Si No
Salario:
Donde:
Actividad:
Deuda familiar: Si No
Monto
Con quien
Posee bienes inmuebles: Si No
Cuales:
Recibe ayuda extranjero: Si No
Cuanto recibe

Fig. 25.



FORMULARIO CONSULTA DE SOLICITUDES

Solicitudes registradas

Tipo de Beca

Interna
 Externa
 Todas

Solicitudes

Carnet	Nombre	Carrera	Año	Origen	Tipo Beca	Aprobado	CV	EE	TT	
07-00006-0	Hazzell Espinoza	Acuícola	II	Ciudad Dario	Residencia	<input checked="" type="checkbox"/>		1	0	2
07-00007-0	bernardo rico	Biología	V	Chichigalpa	Externa	<input checked="" type="checkbox"/>		0	2	0
07-00012-0	rodrigo perez	Estadística	II	Somoto	Interna	<input type="checkbox"/>		0	0	0
07-00013-0	mercedes soza	Acuícola	IV	Bluefields	Interna	<input checked="" type="checkbox"/>		0	2	0
07-00014-0	mario ruiz	Estadística	IV	León	Externa	<input checked="" type="checkbox"/>		0	0	1
07-00017-0	Lilliam Morales	Telemática	IV	Boaco	Externa	<input type="checkbox"/>		0	0	1
07-00018-0	Juan Riquelme	Actuariales	II	Condega	Interna	<input type="checkbox"/>		1	0	1
07-06670-0	Luis Manuel Coronado	Sistemas	II	Somotillo	Interna	<input checked="" type="checkbox"/>		0	0	0

Fig. 26

FORMULARIO GENERAR BAJAS

Nuevas bajas

Estudiantes becados que Pierden Beca

Carnet	Nombre	Carrera	Año	Origen	Promedio	Tipo Beca	Fecha Emisión
07-00001-0	eleazar1	Sistemas	I	Quezalguaque	56	Residencia	2007-03-12
07-00005-0	jorge Hernandez	Actuariales	V	Catarina	59	Transporte	2007-12-03
15-00001-0	suse1	Sistemas	I	León	69	Residencia	1999-02-06
15-00006-0	suse6	Sistemas	I	León	49	Transporte	2004-05-09
20-00002-0	TUAN12	Acuícola	IV	San Rafael del Sur	65	Comida	2007-02-24

Fig. 27.



FORMULARIO CONSULTA DE BECADOS

Estudiantes becados Facultad de ciencias

Becados Facultad de ciencias

Carnet	Nombre	Carrera	Año	Origen	Tipo Beca
07-00002-0	eleazar2	Matemática	III	Ticuanatepe	Residencia
07-00003-0	Camilo Narvaez	Telemática	II	León	Residencia
07-00004-0	Eleazar Trejos	Estadística	III	León	Residencia
07-00008-0	Luis Alfonso Rivera	Actuariales	V	Cinco Pinos	Excelencia
07-00009-0	marcos somarriba	Acuicola	IV	Dipilto	Externa
07-00010-0	Nidia Trejos	Sistemas	IV	León	Interna
07-00011-0	maritza lopes	Agroecología	V	Puerto Cabezas	Interna
07-00015-0	tereza rivas	Química	IV	Corinto	Interna
07-00016-0	pepito jerez	Acuicola	V	Acoyapa	Excelencia
09-00009-0	Felipe Rodriguez	Biología	III	Palacagüina	Externa
15-00004-0	suse4	Sistemas	I	León	Fotocopia

Seleccionar:

Tipo de Beca

Externa
 Interna
 Excelencia
 Todas

Carrera <seleccione> **Año** <seleccion...>

Fig. 28.

FORMULARIO BUSQUEDA

Estudiante

Busqueda

Carnet
 Nombre

Tipo de Consulta:

Becas
 Solicitudes
 Bajas

Carnet:
Nombre:
Carrera:
Año:
Tipo Beca:
Origen:

Fig. 29.



V.XVI. CODIFICACION

Aquí presentamos los aspectos más importantes de la codificación

Código Clase principal

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.lang.Object.*;

public class SBC {

    public static void main(String[] args) {

        try{            //Obtiene la apariencia de ventanas y botones del sistema
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        }
        catch(Exception a)
        {
            System.out.println("No se pudo mostrar"+ a);
        }
        try{
            Acceso acc=new Acceso();        //Carga el formulario de Acceso principal
            acc.show();
        }
        catch(ClassNotFoundException e)
        {
            System.out.println(e.getMessage());
        }
        catch(InstantiationException qe)
        {
            System.out.print(qe.getMessage());
        }
        catch(IllegalAccessException we)
        {
            System.out.print(we.getMessage());
        }
        catch(java.sql.SQLException re)
        {
            System.out.print(re.getMessage());
        }
    }
}
```




Codigo del dialogo de acceso principal

```
import java.awt.event.*;           //Seccion de importación de Clases
predefinidas
import java.awt.*;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import java.lang.Object.*;
import javax.swing.*;
import java.net.URL;
import java.sql.*;
import java.io.*;

public class Acceso extends JFrame
{
    URL url = this.getClass().getResource( "check.gif");
    JButton _aceptar=new JButton("Aceptar");
    JTextField $usr=new JTextField();
    JPasswordField $pssw=new JPasswordField();
    URL urls = this.getClass().getResource( "SBC1.gif" );
    URL urlfon = this.getClass().getResource( "LOCK.jpg");
    Font f=new Font("Times new Roman",Font.ITALIC | Font.BOLD,15);
    public Acceso(throws ClassNotFoundException, //Constructor de la clase
        SQLException, InstantiationException,
        IllegalAccessException
    {
        setTitle("Acceso al sistema");
        setSize(250,250);
        setLocationRelativeTo(null);
        setResizable(false);
        Image icono=getToolkit().getImage(urls); //Obtiene el icono a mostrar en la ventana
        setIconImage(icono);
        Controles();
    }
    private void Controles()
    {
        getContentPane().setLayout(null);
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent evt)
            {
                closeDialog(evt);
            }
        });
        //registrar el evento del boton guardar
        _aceptar.addActionListener(new ActionListener()
        {
```



```
public void actionPerformed(ActionEvent evt)
{
    _aceptarActionPerformed(evt);
}
});
getContentPane().setBackground(new Color(0,128,150));
Container c=getContentPane(); //permite agregar los componentes
JLabel lb=new JLabel();
lb.setIcon(new ImageIcon(urlfon));
c.add(lb);
lb.setBounds(45,10,86,86);
JLabel Lcarnet=new JLabel("Usuario");
c.add(Lcarnet);
Lcarnet.setFont(f);
Lcarnet.setBounds(25,100,80,15);
c.add($usr);
$usr.setBounds(100,100,100,20);
JLabel Ltipo=new JLabel("Contrase\u00f1a:");
c.add(Ltipo);
Ltipo.setFont(f);
Ltipo.setBounds(20,125,100,20);
c.add($pssw);
$pssw.setBounds(100,125,100,20);
$pssw.setEchoChar('*');
c.add(_aceptar);
_aceptar.setIcon(new ImageIcon(url));
_aceptar.setFont(new Font("Times new Roman",Font.ITALIC | Font.BOLD,12));
_aceptar.setBounds(100,150,100,20);
getRootPane().setDefaultButton(_aceptar);
}
private void _aceptarActionPerformed(ActionEvent evt)
{
    FileInputStream fis;
    DataInputStream dis;
    String fichero="ArchivoData.txt"; //Cargar el fichero para verificar Contraseña
    String admin = "";
    String pa="";
    String pass = null;
    String usr = $usr.getText();
    char[] pssw = $pssw.getPassword();
    try
    {
        fis = new FileInputStream(fichero);
        dis = new DataInputStream(fis);
        admin=dis.readUTF();
        pa=dis.readUTF();
        dis.close();
        fis.close();
    }
}
```



```
    }
    catch(IOException e){
        System.err.println("error al intentar abrir el fichero "+e.getMessage());
    }
}
if(usr.compareTo(admin)==0 && pa.compareTo(pass.valueOf(pssw))==0 )
{
    try                //verifica si la contraseña es correcta
    {
        frame pp = new frame();        //Muestra el menú principal
        pp.show(true);
    }
    catch(ClassNotFoundException e)
    {
        System.out.println(e.getMessage());
    }
    catch(InstantiationException e)
    {
        System.out.print(e.getMessage());
    }
    catch(IllegalAccessException e)
    {
        System.out.print(e.getMessage());
    }
    catch(java.sql.SQLException e)
    {
        System.out.print(e.getMessage());
    }
    setVisible(false);
    dispose();
}
else if(usr.compareTo(admin)!=0 || pa.compareTo(pass.valueOf(pssw))!=0 )
{
    $usr.setText("");
    $pssw.setText("");
    JOptionPane.showMessageDialog(null, "El nombre de usuario o la " +
        "contraseña son incorrecta",
"Error", javax.swing.JOptionPane.ERROR_MESSAGE);
    $usr.requestFocus();
}
}
private void closeDialog(WindowEvent evt)
{
    setVisible(false);
    dispose();
}
}
```



Código del dialogo Menú Principal

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
import java.lang.Object.*;
import java.net.URL;
import java.sql.*;
import java.io.IOException;
import javax.swing.JMenuItem;
import java.net.*;
import java.util.*;
import java.applet.*;
import javax.swing.ImageIcon;

class frame extends JFrame implements ActionListener{
    //declacion de algunas variables tipo menu
    JMenuBar Bmenu= new JMenuBar();
    JMenu Mregistro=new JMenu();
    JMenuItem MIactuales=new JMenuItem("Estudiantes");
    JMenuItem MInuevas=new JMenuItem("Nuevas");
    JMenuItem MIsalir=new JMenuItem();

    /*mostrar imagen de fondo*/
    URL url = this.getClass().getResource( "parainfo.jpg" );
    JLabel lb=new JLabel();
    URL urls = this.getClass().getResource( "SBC1.gif" );

    public frame()throws ClassNotFoundException,//constructor
    SQLException, InstantiationException,
    IllegalAccessException{
        setTitle("Sistema de becas-Facultad de ciencias UNAN León");
        setSize(800,570);
        setResizable(false);
        setLocationRelativeTo(null);
        Container c=getContentPane();
        Image icono=getToolkit().getImage(urls);
        setIconImage(icono);
        c.setBackground(new Color(0,150,128));
        lb.setIcon(new ImageIcon(url));
        c.add(lb);
        lb.setBounds(10,10,200,200);
        //agregar algunos elemetos al menu
        Mregistro.setMnemonic('R');
        Mregistro.setText("Registro");
        Mregistro.add(MIactuales);
        Mregistro.add(MInuevas);
        MIsalir.setText("Salir");
    }
}
```



```
Mregistro.add(MIsalir);
Bmenu.add(Mregistro);

Bmenu.add(Mayuda);
    setJMenuBar(Bmenu);//establesco la barra de menu
    addWindowListener(new manejador());
//registro de eventos para el menu
MIactuales.addActionListener(this);
MInuevas.addActionListener(this);
MIsalir.addActionListener(this);
}
public void actionPerformed(ActionEvent e){
    Object obj=e.getSource();
    String d="Estudiantes becados--Facultad de ciencias";
    String l="Lista de becas--Facultad de ciencias";
    String Be="Estudiantes con becas externas de la carrera--Facultad de ciencias";

    if(obj==MIsalir)//Salir de la aplicacion
    {
        System.exit(0);
    }
    if(obj==MInotas) //llamado a la clase cambiar, no se establece conexión con
BD
    {
        new CambiarC(this,"Cambiar Contrase\u00f1a",true).setVisible(true);
    }

    if(obj==MIactuales) //llamado a la clase consulta de becados que establece
conxion con BD
    {
        try{
            new Becados(this,"Estudiantes Becados Facultad de
ciencias",true).setVisible(true);
        }
        catch(ClassNotFoundException a)
        {
            System.err.println("error al intentar cargar la clase
"+a.getMessage());
            a.printStackTrace();
        }
        catch (SQLException w)
        {
            System.err.println("Error al intentar conectarse.
"+w.getMessage());
            w.printStackTrace();
        }
        catch (InstantiationException r)
```



```
        {
            System.err.println("Error al intentar conectarse. "+r.getMessage());
            r.printStackTrace();
        }
    catch (IllegalAccessException t)
    {
        System.err.println(" Error al intentar conectarse. "+t.getMessage());
        t.printStackTrace();
    }
    //respaldar base de datos sql
    if (obj==MIsps)
    {
        Process respaldarBD=null;
        if(respaldarBD==null)
        {
            try { // creo un proceso para ejecutar Mysqldump
                respaldarBD=Runtime.getRuntime().exec("res.bat");

                if(respaldarBD!=null)
                    JOptionPane.showMessageDialog(this,"Se ha respaldado BD exitosamente
en c:",
                    "Procesado..",JOptionPane.INFORMATION_MESSAGE);
            } catch (IOException ex)
            {
                JOptionPane.showMessageDialog(this,"No se pudo respaldarBD",
                "ERROR",JOptionPane.ERROR_MESSAGE);
                ex.printStackTrace();
            }
        }
    }
}
// permite cerrar la ventana
class manejador extends WindowAdapter{
    public void windowClosing(WindowEvent e){

        System.out.println("salir");
        System.exit(0);
    }
}
}
```



Conexión con base de datos Sql

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
import javax.swing.border.*;
import java.lang.String.*;
import java.sql.*;
import java.util.Vector;
import java.sql.*;
import javax.swing.table.AbstractTableModel;
import javax.swing.event.TableModelEvent;

public class Conexion{
    public Connection conexión;
    public Statement sentenciaSQL;
    public Statement sentenciaSQL1;
    public Statement sentenciaSQL2;
    Vector          rows = new Vector();
    ResultSetMetaData metaData;
    public Conexion
    {
    }
    public void conectar() throws SQLException, ClassNotFoundException,
        InstantiationException, IllegalAccessException
    {
        String controlador = "com.mysql.jdbc.Driver";//controlador para sql
        Class.forName(controlador);
        String URL_bd = "jdbc:mysql://localhost:3306/ciencias";
        String usuario = "root";
        String contraseña = "grafia";
        // Conectar con la BD
        conexión = DriverManager.getConnection(
            URL_bd, usuario, contraseña);
        // Crear una sentencia SQL
        sentenciaSQL = conexión.createStatement(
            java.sql.ResultSet.TYPE_SCROLL_INSENSITIVE,
            java.sql.ResultSet.CONCUR_UPDATABLE);
        sentenciaSQL1 = conexión.createStatement(
            java.sql.ResultSet.TYPE_SCROLL_INSENSITIVE,
            java.sql.ResultSet.CONCUR_UPDATABLE);
        sentenciaSQL2 = conexión.createStatement(
            java.sql.ResultSet.TYPE_SCROLL_INSENSITIVE,
            java.sql.ResultSet.CONCUR_UPDATABLE);
        if (conexión!=null)
            System.out.print("\n hay Conexion \n");
        else
            System.out.print("\n no hay Conexion \n");
    }
}
```



```
    }  
    public void cerrar() throws SQLException,ClassNotFoundException,  
        InstantiationException, IllegalAccessException  
    {  
        try{  
            if (sentenciaSQL!=null){  
                sentenciaSQL.close();  
                sentenciaSQL1.close();  
                sentenciaSQL1.close();  
                conexión.close();  
                System.out.print("\nConexion cerrada \n");  
            }  
        }  
        catch(Exception e)  
        {  
            System.err.println("cerrarConexion() Error al intentar cerrar  
conexion. "+e.getMessage());  
            e.printStackTrace();  
        }  
    }  
}
```

Codigo del dialogo registro nuevos

```
import javax.swing.*;  
import java.net.URL;  
import java.awt.event.*;  
import java.awt.*;  
import java.sql.*;  
import java.lang.Object.*;  
  
class NuevosB extends JDialog  
{  
    URL url = this.getClass().getResource( "grd.gif" );  
    String becas[]={ "Externa", "Interna", "Residencia", "Transporte", "Comida", "Fotocopia" };  
    Utilidades ut= new Utilidades();  
    JButton _guardar=new JButton("Guardar");  
    TextCarnet $carnet=new TextCarnet();  
    JComboBox $TipoB=new JComboBox(becas);  
    public ResultSet resulset;  
    Conexion conx;  
    Font f=new Font("Times new Roman",Font.ITALIC | Font.BOLD,15);  
  
    public NuevosB(JFrame paren,String a,boolean mod)throws ClassNotFoundException,  
        SQLException, InstantiationException,  
        IllegalAccessException  
    {
```




```
super(paren,a,mod);
Controles();
setSize(250,150);
setLocationRelativeTo(null);
setResizable(false);
conx=new Conexion(); conx.conectar();
}

private void Controles()
{
    getContentPane().setLayout(null);
    addWindowListener(new WindowAdapter()
    {
        public void windowClosing(WindowEvent evt)
        {
            closeDialog(evt);
        }
    });

    //REGISTRAR EL EVENTO DEL BOTON GUARDAR
    _guardar.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent evt)
        {
            _guardarActionPerformed(evt);
        }
    });/**/
    getContentPane().setBackground(new Color(0,128,150));
    Container c=getContentPane();//permite agregar los componentes

    JLabel Lcarnet=new JLabel("Carnet");
    c.add(Lcarnet);
    Lcarnet.setFont(f);
    Lcarnet.setBounds(20,20,80,15);

    c.add($carnet);
    $carnet.setBounds(100,20,100,20);

    JLabel Ltipo=new JLabel("Tipo de beca");
    c.add(Ltipo);
    Ltipo.setFont(f);
    Ltipo.setBounds(15,45,100,20);
    c.add($TipoB);
    $TipoB.setBounds(100,45,100,20);

    c.add(_guardar);
    _guardar.setIcon(new ImageIcon(url));
    _guardar.setFont(new Font("Times new Roman",Font.ITALIC | Font.BOLD,12));
```



```
_guardar.setBounds(100,70,100,20);
_guardar.getRootPane().setDefaultButton(_guardar);
}

//METHODO PARA REGSITRAR LOS DATOS EN LA BASE DE DATOS
private void _guardarActionPerformed(ActionEvent evt)
{
    String car,tipB="",temp="";
    car=$carnet.getText();

    tipB=$TipoB.getSelectedItem().toString();
    int n=0, i=1, control=0;
    char[] fuente = car.toCharArray();
    n=fuente.length;

    if(n !=10 || n == 0)
    { //VERIFICA EL NUMERO DE CARACTERES PARA EL CARNET
CORRECCTO
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(null,"Número no valido de caracteres en dato
carnet", "Mensaje ",
        JOptionPane.INFORMATION_MESSAGE);
        $carnet.requestFocus();
        return;
    }

    for(n=0; n<10; n++)
    { //VERIFICA EL FORMATO DE CARNET
        if(fuente[n]!='-' && (n==2 || n==8) )
        {
            System.out.println(fuente[n]);
            System.out.println(n);
            Toolkit.getDefaultToolkit().beep();
            JOptionPane.showMessageDialog(null,"Formato de carnet no válido", "Mensaje
",
                JOptionPane.INFORMATION_MESSAGE);

            $carnet.requestFocus();
            return;
        }
        if(!(n==2 || n==8) && fuente[n]!='-')
        {
            System.out.print(fuente[n]);System.out.println("-"+n);
            Toolkit.getDefaultToolkit().beep();
            JOptionPane.showMessageDialog(null,"Formato de carnet no válido",
"Mensaje ",
                JOptionPane.INFORMATION_MESSAGE);
```



```
        $carnet.requestFocus();
    return;
}
}
//****REVISAR SI EL ESTUDIANTE YA TIENE ASIGNADA LA BECA*****/
int as=0;
String query3 =
    " SELECT " + "Carnet" + ", " + "Asignada" + " FROM " + "solicitud";
try{
    resulset=conx.sentenciaSQL1.executeQuery(query3);
resulset.first();
as =Integer.parseInt(resulset.getString(2));
if(car.compareTo(resulset.getString(1))==0)
{
    if(as==0){
        JOptionPane.showMessageDialog(null,"Registro Actualizado", " ",
        JOptionPane.INFORMATION_MESSAGE);
    }
    else if(as==1){
        JOptionPane.showMessageDialog(null,"Este estudiante ya tiene Beca", " ",
        JOptionPane.INFORMATION_MESSAGE);
    }
    return;
}
    control=1;}

while(resulset.next())
{
    as =Integer.parseInt(resulset.getString(2));
    if(car.compareTo(resulset.getString(1))==0)
    {
        if(as==0){
            JOptionPane.showMessageDialog(null,"Registro Actualizado", " ",
            JOptionPane.INFORMATION_MESSAGE);
        }
        else if(as==1){
            JOptionPane.showMessageDialog(null,"Este estudiante ya tiene Beca", " ",
            JOptionPane.INFORMATION_MESSAGE);
        }
        return;
    }
        control=1;
    }
}
}
catch(java.sql.SQLException a)
{
    System.out.print(a.getMessage());
}
}
```



```
///***** ACTUALIZAR LAS TABLAS****//
String query =
    " update " + "solicitud" +
    " set " + "Asignada" + " = " + i + ", " + "Tipo_beca" + " = " + "" + tipB + "" +
    " WHERE " + "Carnet" + " LIKE " + "" + car + "";
try{
    conx.sentenciaSQL.executeUpdate(query);
}
catch(java.sql.SQLException s)
{
    System.out.print(s.getMessage());
}

//PARA ACTUALIZAR TAMBIEN LA TABLA ESTUDIANTE
String query2 =
    " update " + "estudiante" +
    " set " + "Situacion_beca" + " = " + i +
    " WHERE " + "Carnet" + " LIKE " + "" + car + "";
try{
    conx.sentenciaSQL.executeUpdate(query2);
}
catch(java.sql.SQLException s)
{
    System.out.print(s.getMessage());
}
if(control==0)
{
    JOptionPane.showMessageDialog(null,"No se encontró el registro", " ",
    JOptionPane.INFORMATION_MESSAGE);
    $carnet.setText("");
    $carnet.requestFocusInWindow();
}
// *****/
}
private void closeDialog(WindowEvent evt)
{
    try{
        conx.cerrar();
    }

    catch(java.sql.SQLException a)
    {
        System.out.print(a.getMessage());
    }
    setVisible(false);
    dispose();
}
}
```



Clase para imprimir componente JTable

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
import java.awt.print.*;
import java.awt.geom.*;
import java.awt.Dimension;

class ImprimirS implements Printable
{
    JTable verTabla;
    public ImprimirS(JTable j) {
        verTabla=j;
    }

    //método para imprimir un componente JTable

    public int print(Graphics g, PageFormat formatoPag,int indice) throws
    PrinterException {

        Graphics2D grp = (Graphics2D) g;
        grp.setColor(Color.black);
        int altoLetra=grp.getFontMetrics().getHeight();//ajustamos el tamaño de la fuente
        int desLetra=grp.getFontMetrics().getDescent();

        //calcular el número de paginas
        double altoPag = formatoPag.getImageableHeight()-altoLetra;
        double anchoPag = formatoPag.getImageableWidth();
        double anchoTabla = (double)
verTabla.getColumnModel().getTotalColumnWidth();
        double escala = 1;
        if (anchoTabla >= anchoPag)
        {
            escala = anchoPag / anchoTabla;
        }
        double alturaEncabPag=verTabla.getTableHeader().getHeight()*escala;
        double anchoTablaPag=anchoTabla*escala;
        double alturaFila=(verTabla.getRowHeight()+ verTabla.getRowMargin())*
escala;
        int filasPag=(int)((altoPag- alturaEncabPag)/alturaFila);
        double altoPagTabla=alturaFila*filasPag;
        int totalPag=(int)Math.ceil(((double)verTabla.getRowCount())/filasPag);

        if(indice>=totalPag)
        { no existe objeto para imprimir
            return NO_SUCH_PAGE;
        }
    }
}
```



```
grp.translate(formatoPag.getImageableX(),formatoPag.getImageableY());
//numerar la página en el centro inferior
    grp.drawString(""+(indice+1),(int)anchoPag/2-35,(int)(altoPag +altoLetra-
desLetra));
    grp.translate(0f,alturaEncabPag);
    grp.translate(0f,-indice*altoPagTabla);

    //comprobar si existe tabla disponible y su tamaño.
//ajustar los limites apropiados
    if (indice + 1 == totalPag) {
        int lastRowPrinted =filasPag * indice;
        int numRowsLeft =verTabla.getRowCount()-lastRowPrinted;
        grp.setClip(0,(int)(altoPagTabla * indice),
            (int) Math.ceil(anchoTablaPag),
            (int) Math.ceil(alturaFila * numRowsLeft));
    }
//desplazarce a la siguiente pagina.
    else{
        grp.setClip(0, (int)(altoPagTabla*indice),
            (int) Math.ceil(anchoTablaPag),
            (int) Math.ceil(altoPagTabla));
    }
    grp.scale(escala,escala);
    verTabla.paint(grp);
    grp.scale(1/escala,1/escala);
    grp.translate(0f,indice*altoPagTabla);
    grp.translate(0f, -alturaEncabPag);
    grp.setClip(0, 0,
        (int) Math.ceil(anchoTablaPag),
        (int)Math.ceil(alturaEncabPag));
    grp.scale(escala,escala);
    verTabla.getTableHeader().paint(grp);
    return Printable.PAGE_EXISTS;
} }
}
```

Clase que verifica el formato de carnet

```
import javax.swing.*;
import javax.swing.text.*;
import java.awt.*;

class TextCarnet extends JTextField
{
    protected Document createDefaultModel()
    {
        return new Verifica();
    }
}
protected class Verifica extends PlainDocument
```



```
{
public void insertString(int offs, String str, AttributeSet a)
    throws BadLocationException
{
    // fuente: almacena el contenido de la caja de texto
    char[] fuente = str.toCharArray();
    // resultado: almacena el contenido de la caja de texto validado
    char[] resultado = new char[fuente.length];
    int j = 0;
    // Almacenar en resultado los caracteres válidos de fuente
    for (int i = 0; i < fuente.length; i++)
    {
        if ((fuente[i] >= '0' && fuente[i] <= '9' || (fuente[i] == '-'))
            resultado[j++] = fuente[i];
        else
            Toolkit.getDefaultToolkit().beep();
    }
    super.insertString(offs, new String(resultado, 0, j), a);
} }}
```

Clase ayuda

```
import javax.swing.*;
import java.awt.*;
import java.net.URL;
import java.net.MalformedURLException;
import java.io.*;
import javax.swing.event.*;

class Help extends JDialog
{
    public Help(JFrame paren,String a,boolean mod)
    {
        super(paren,a, mod);
        setSize(600,400);
        setLocationRelativeTo(null);
        setResizable(false);
        JPanel html = new JPanel();
        getContentPane(html);
    }
}

class JPanel extends JScrollPane implements HyperlinkListener {
    JPanel html;
    public JPanel() {
        try {
            File f = new File ("HelpFiles/index.html"); //ubicación de los archivos html
            String s = f.getAbsolutePath();
            s = "file:" +s;
```



```
        URL url = new URL(s);
        html = new JEditorPane(s);
        html.setEditable(false);
        html.addHyperlinkListener(this);
        JViewport vp = getViewport();
        vp.add(html);
    } catch (MalformedURLException e) { System.out.println("ERROR URL: " + e);
    } catch (IOException e) {
        System.out.println("IOException: " + e);
    }
}
// Notificacion de cambio relativo al enlace
public void hyperlinkUpdate(HyperlinkEvent e) {
    if (e.getEventType() == HyperlinkEvent.EventType.ACTIVATED) {
        linkActivated(e.getURL());
    }
}
} // Siguiendos las referencias en un enlace. Obteniendo la referencia del url
// por defecto esta llamada <a href="#setPage">setPage</a>,
protected void linkActivated(URL u) {
    Cursor c = html.getCursor();
    Cursor Cursors = Cursor.getPredefinedCursor(Cursor.WAIT_CURSOR);
    html.setCursor(Cursors);
    SwingUtilities.invokeLater(new CargaPagina(u, c));
}
}
//clase que carga las paginas
class CargaPagina implements Runnable {
    CargaPagina(URL u, Cursor c) {
        url = u;
        cursor = c;
    }
    public void run() {
        if (url == null) {
            // restablece el cursor
            html.setCursor(cursor);
            Container p = html.getParent();
            p.repaint();
        } else { Document doc = html.getDocument();
            try {
                html.setPage(url);
            } catch (IOException ioe) {
                html.setDocument(doc);
                getToolkit().beep();
            } finally {
                url = null;
                SwingUtilities.invokeLater(this);
            }
        }
    }
    URL url;
    Cursor cursor;
}
}
```




Clase Del Dialogo Busqueda

```
import javax.swing.*;
import javax.swing.table.*;
import javax.swing.border.*;
import java.awt.event.*;
import java.awt.*;
import java.net.URL;
import java.sql.*;
import java.lang.Object.*;
class Busqueda extends JDialog
{
    Conexion conxBus;
    public ResultSet cdrBus;
    public ResultSet cdrBus1;
    public ResultSet cdrBus2;
    public Busqueda(JFrame paren,String a,boolean mod)throws ClassNotFoundException,
        SQLException, InstantiationException,
        IllegalAccessException
    { //Constructor de la clase, establece conexion con BD
        super(paren,a, mod);
        Controles();
        setSize(450,250);
        setLocationRelativeTo(null);
        setResizable(false);
        conxBus=new Conexion(); conxBus.conectar();
    }
    // variable url para obtener imagenes
    URL url = this.getClass().getResource( "lense.gif" );
    Font f=new Font("Times new Roman",Font.ITALIC | Font.BOLD,15);
    Container c=getContentPane();//permite agregar los componentes
    // variable tipo componentes
    JPanel pBus=new JPanel();
        JLabel Bus=new JLabel("Busqueda");
        ButtonGroup estud=new ButtonGroup();
        JRadioButton Ecarne=new JRadioButton("Carnet");
        JRadioButton Enombre=new JRadioButton("Nombre");
        JButton _buscar= new JButton("Buscar");
        JButton _borrar= new JButton("Borrar");

    private void Controles()//Metodo para agregar controles al dialogo y registrar eventos
    {
        getContentPane().setLayout(null);
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent evt)
            {
                closeDialog(evt);
            }
        }
    }
}
```



```
    }
  });
  ItemListener IL=new ItemListener() //evento para los botones de radio
  { //metodos paraagregar funcion a los botones
    public void itemStateChanged(ItemEvent iv)
    {
      botonesRadio();
    }
  };

  _buscar.addActionListener(new ActionListener()
  {
    public void actionPerformed(ActionEvent evtbus)
    {
      _buscarActionPerformed(evtbus);
    }
  });

  _borrar.addActionListener(new ActionListener()
  {
    public void actionPerformed(ActionEvent evtbor)
    {
      _borrarActionPerformed(evtbor);
    }
  });
  ActionListener al=new ActionListener()
  {
    public void actionPerformed(ActionEvent ev)
    {
      botones(ev);
    }
  };
  getContentPane().setBackground(new Color(0,128,150));
  c.add(Bus);//dibujar controles
  Bus.setFont(f);
  Bus.setBounds(15,15,80,15);
  pBus.setLayout(new GridLayout(2,2,5,5));
  pBus.setBackground(new Color(0,128,150));
  Ecarne.addItemListener(IL);

  private void _buscarActionPerformed(ActionEvent evtbus)
  {
    _ant.setEnabled(false);
    $carnet.setText("");
    $nombre.setText("");
    $carrera.setText("");
    $anyo.setText("");
    $tipobeca.setText("");
  }
}
```



```
$origen.setText("");
String carE="", carS="", nombr="", tipoBe="", carrer="",
anio="", Orig="", query="", senten="";
String carBus="", nomBus="";

if(Enombre.isSelected())//Buscar por nombre
{
    nomBus=$Bnombre.getText();
    if(nomBus.compareTo("")==0)
    {
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(null,"Debe Introducir Nombre",
"Mensaje ",
        JOptionPane.INFORMATION_MESSAGE);
        return;
    }
    if(Beca.isSelected())
    {
        query = " SELECT " + "Carnet" + ", " + "Nombre" + ", " +
"Carrera" + ", " + "Anyo" +
        ", " + "Municipio" + " FROM " + "estudiante" + " WHERE " +
        "Nombre" + " LIKE " + "" + nomBus + "%" + "" + " AND " + "Situacion_beca"
+ " = " + 1;
    }
    else if(Soli.isSelected())
    {
        query = " SELECT " + "Carnet" + ", " + "Nombre" + ", " +
"Carrera" + ", " + "Anyo" +
        ", " + "Municipio" + " FROM " + "estudiante" + " WHERE " +
        "Nombre" + " LIKE " + "" + nomBus + "%" + "" + " AND " + "Situacion_beca"
+ " = " + 0;
    }
    else if(Baja.isSelected())
    {
        query = " SELECT " + "Carnet" + ", " + "Nombre" + ", " +
"Carrera" + ", " + "Anyo" +
        ", " + "Municipio" + " FROM " + "estudiante" + " WHERE " +
        "Nombre" + " LIKE " + "" + nomBus + "%" + "" + " AND " + "Situacion_beca"
+ " = " + "" + -1 + "";
    }
}

try{
    cdrBus=conxBus.sentenciaSQL.executeQuery(query);
    cdrBus.first();
    carE = cdrBus.getString(1);
    senten =" SELECT " + "Carnet" + ", " + "Tipo_beca" + " FROM " + "solicitud" + "
WHERE " +
```



```
"Carnet" + " LIKE " + "" + carE + """;

cdrBus1=conxBus.sentenciaSQL1.executeQuery(senten);
cdrBus1.first();carS = cdrBus1.getString(1);
if(carS.compareTo(carE)==0)
{
    nombr = cdrBus.getString(2);
    carrer = cdrBus.getString(3);
    anio = cdrBus.getString(4);
    Orig = cdrBus.getString(5);
    tipoBe = cdrBus1.getString(2);

    $carnet.setText(carE);
    $nombre.setText(nombr);
    $carrera.setText(carrer);
    $anyo.setText(anio);
    $tipobeca.setText(tipoBe);
    $origen.setText(Orig);
    _borrar.setEnabled(true);
}
while(cdrBus.next())
{
    _sig.setEnabled(true);
}
cdrBus.first();
}
catch(java.sql.SQLException s)
{
    System.out.print(s.getMessage());
}
}
private void _borrarActionPerformed(ActionEvent evtbor)
{
    String cad=$carnet.getText();
    String senten="", senten2="";

    senten = " DELETE FROM " + "estudiante" + " WHERE " +
"Carnet" + " LIKE " + "" + cad + """;
    senten2 = " DELETE FROM " + "solicitud" + " WHERE " +
"Carnet" + " LIKE " + "" + cad + """;
    if(cad.compareTo("")==0)
    {
        Toolkit.getDefaultToolkit().beep();
        JOptionPane.showMessageDialog(null,"No hay registro para Borrar", "Mensaje ",
JOptionPane.INFORMATION_MESSAGE);
    }
    return;
}
```



```
    }
    if((JOptionPane.showConfirmDialog(null,"Desea borrar el Registro?", "Mensaje ",
        JOptionPane.YES_NO_OPTION, JOptionPane.WARNING_MESSAGE)==0)
        {
            try{
                conxBus.sentenciaSQL.executeUpdate(senten);
                conxBus.sentenciaSQL.executeUpdate(senten2);
                $carnet.setText("");    $nombre.setText("");
                $carrera.setText("");    $anyo.setText("");
                $tipobeca.setText("");    $origen.setText("");
                JOptionPane.showMessageDialog(null,"Registro Borrado", "Mensaje ",
                    JOptionPane.INFORMATION_MESSAGE);
            }
            catch(java.sql.SQLException a)
            {
                System.out.print(a.getMessage());
            }
        }
    }
    private void closeDialog(WindowEvent evt)
    {
        //cerrar dialogo con conexión a BD
        try{
            conxBus.cerrar();
            if (cdrBus!=null){
                cdrBus.close();    cdrBus1.close();
            }
        }
        catch(java.sql.SQLException a)
        {
            System.out.print(a.getMessage());
        }
        catch(ClassNotFoundException l)
        {
            System.err.println("error al intentar cargar la clase "+l.getMessage());
            l.printStackTrace();
        }
        catch (InstantiationException k)
        {
            System.err.println("Error al intentar conectarse. "+k.getMessage());
            k.printStackTrace();
        }
        catch (IllegalAccessException j)
        {
            System.err.println(" Error al intentar conectarse. "+j.getMessage());
            j.printStackTrace();
        }
        setVisible(false);
        dispose();
    }
}
```



Codigo dialogo consulta becas

```
public void botones(ActionEvent ev)
{
    Object obj=ev.getSource();
    //boolean imprimire= false;
    if(obj==_imp) //if boton imprimir
    {
        pr=new ImprimirS(TablaPrint);
        PrinterJob pj=PrinterJob.getPrinterJob();
        //setPrintable(Printable painter) Calls painter to render the pages.
        pj.setPrintable(pr); //asqui error
        if(pj.printDialog())
        {
            try
            {
                pj.print();
            }catch (Exception PrintException) {}
        }
        this.repaint();
        TablaPrint.repaint();
    }
    if(obj==_Acep)
    {
        DibujarTabla();
    }
}
}
public void DibujarTabla()
{
    String opCar="", opAnyo="", query="", query2="";
    String carE="", carS="", nombr="", tipoBe="", carrer="", anio="",
    Orig="";
    opCar=$opCarr.getSelectedItem().toString();
    opAnyo=$opAnyo.getSelectedItem().toString();
    Vector cols = new Vector();
    Container c=getContentPane();
    //*****CONSULTAS SQL*****////
    if(Todas.isSelected())
    {
        query = " SELECT " + "Carnet" + ", " + "Nombre" + ", " + "Carrera" + ", " + "Anyo" +
        ", " + "Municipio" + " FROM " + "estudiante" + " WHERE " +
        "Situacion_beca" + " = " + 1;
        query2 = " SELECT " + "Carnet" + ", " + "Tipo_beca" + " FROM " + "solicitud" + "
        WHERE " + "Asignada" + " = " + 1;
        if(opCar.compareTo("<seleccione>")!=0)
        {
            query = " SELECT " + "Carnet" + ", " + "Nombre" + ", " + "Carrera" + ", " + "Anyo" +
```



```
    ", " + "Municipio" + " FROM " + "estudiante" + " WHERE " +  
    "Carrera" + " LIKE "+"+" + opCar +"" + " AND " + "Situacion_beca" + " = "+  
1;  
    }  
    if(opAnyo.compareTo("<seleccione >")!=0)  
    { query = " SELECT " + "Carnet" + ", " + "Nombre" + ", " + "Carrera" + ", " + "Anyo"  
+  
    ", " + "Municipio" + " FROM " + "estudiante" + " WHERE " + "Anyo" + "  
LIKE "+"+" + opAnyo +"" + " AND " + "Situacion_beca" + " = "+ 1;    }  
    if(opCar.compareTo("<seleccione >")!=0 && opAnyo.compareTo("<seleccione >")!=0)  
    { query = " SELECT " + "Carnet" + ", " + "Nombre" + ", " + "Carrera" + ", " + "Anyo"  
+  
    ", " + "Municipio" + " FROM " + "estudiante" + " WHERE " +  
    "Carrera" + " LIKE "+"+" + opCar +"" + " AND " + "Anyo" + " LIKE "+"+" +  
opAnyo +"" + " AND " + "Situacion_beca" + " = "+ 1; }  
    }  
    if(Ext.isSelected())  
        {query = " SELECT " + "Carnet" + ", " + "Nombre" + ", " + "Carrera" + ",  
+ " + "Anyo" + ", " + "Municipio" + " FROM " + "estudiante" + " WHERE "  
+ "Situacion_beca" + " = " + 1;  
query2 = " SELECT " + "Carnet" + ", " + "Tipo_beca" + " FROM " + "solicitud" + "  
WHERE " + "Asignada" + " = " + 1 + " AND " + "Tipo_beca" + " LIKE " + "Externa";  
  
//COMENZAR A DIBUJAR TABLA  
try{  
    cdr=conx.sentenciaSQL.executeQuery(query);  
    cdr1=conx.sentenciaSQL1.executeQuery(query2);  
    cdr.first();  
    cdr1.first();  
    carE = cdr.getString(1);  
    carS = cdr1.getString(1);  
    if(carS.compareTo(carE)==0)  
    {  
        nombr = cdr.getString(2);  
        carrer = cdr.getString(3);  
        anio = cdr.getString(4);  
        Orig = cdr.getString(5);  
        tipoBe = cdr1.getString(2);  
        Vector veca = new Vector();  
        veca.addElement(carE);  
        veca.addElement(nombr);  
        veca.addElement(carrer);  
        veca.addElement(anio);  
        veca.addElement(Orig);  
        veca.addElement(tipoBe);  
        cols.addElement(veca);  
    }  
    while(cdr1.next()) //BUSCA EN LA TABLA SOLICITUD  
    {
```



```
carS = cdr1.getString(1);

if(carE.compareTo(carS)==0)
{
    nombr = cdr.getString(2);
    carrer = cdr.getString(3);
    anio = cdr.getString(4);
    Orig = cdr.getString(5);
    tipoBe = cdr1.getString(2);
    Vector vecr = new Vector();
    vecr.addElement(carE);
    vecr.addElement(nombr);
    vecr.addElement(carrer);
    vecr.addElement(anio);
    vecr.addElement(Orig);
    vecr.addElement(tipoBe); cols.addElement(vecr);
} //if
} //while cdr solicitud

while(cdr.next() ) //MOVERSE AL SIGUIENTE REGISTRO DE ESTUDIANTE
{
    carE = cdr.getString(1);
    cdr1.first();
    carS = cdr1.getString(1);
    if(carE.compareTo(carS)==0) //COMPARAR EL SIGUIENTE CARNET DE
ESTUDIANTE
    {
        //CON EL PRIMERO DE SOLICITUD
        nombr = cdr.getString(2);
        carrer = cdr.getString(3);
        anio = cdr.getString(4);
        Orig = cdr.getString(5);
        tipoBe = cdr1.getString(2);
        Vector vectr = new Vector();
        vectr.addElement(carE);
        vectr.addElement(nombr);
        vectr.addElement(carrer);
        vectr.addElement(anio);
        vectr.addElement(Orig);
        vectr.addElement(tipoBe);
        cols.addElement(vectr);
    } // IF
    cols.addElement(vectr); } //IF
} //CDR TABLA SOLICITUD
} //WHILE CDR TABLA ESTUDIANTE
} //TRY
catch(java.sql.SQLException s)
{
    System.out.print(s.getMessage());
```




```
    }  
    //CABECERAS  
    Vector cabeceras = new Vector(6);  
    cabeceras.addElement("Carnet");  
    cabeceras.addElement("Nombre");  
    cabeceras.addElement("Carrera");  
    cabeceras.addElement("Año");  
    cabeceras.addElement("Origen");  
    cabeceras.addElement("Tipo Beca");  
    JScrollPane ScrPanel = new JScrollPane();  
    JTable TablaEst = new JTable();  
    TablaEst.setModel(new DefaultTableModel(cols, cabeceras)  
    {  
        Class[] tipoColumn = { String.class,String.class,  
            String.class,String.class, String.class,String.class };  
        boolean[] editColumn = { false, false, false, false,false, false};  
        public Class getColumnClass(int indColumn)  
        {  
            return tipoColumn[indColumn];  
        }  
        public boolean isCellEditable(int indFila, int indColumn)  
        {  
            return editColumn[indColumn];    }  
    });  
    TableColumn colum = null;  
    colum = TablaEst.getColumnModel().getColumn(0);  
    colum.setPreferredWidth(40);  
    colum = TablaEst.getColumnModel().getColumn(1);  
    colum.setPreferredWidth(140);  
    colum = TablaEst.getColumnModel().getColumn(2);  
    colum.setPreferredWidth(40);  
    colum = TablaEst.getColumnModel().getColumn(3);  
    colum.setPreferredWidth(20);  
    colum = TablaEst.getColumnModel().getColumn(4);  
    colum.setPreferredWidth(40);  
    colum = TablaEst.getColumnModel().getColumn(5);  
    colum.setPreferredWidth(40);  
    ScrPanel = new JScrollPane(TablaEst);  
    TablaPrint=TablaEst;  
    c.add(ScrPanel);  
    ScrPanel.setBounds(10,40,620,270);  
    }//FIN DE DIBUJAR TABLA  
}
```



VI. CONCLUSIONES

Al concluir este trabajo, hemos logrado cumplir con los objetivos propuestos, los cuales tienen como finalidad el desarrollo de un sistema computarizado que permita minimizar el trabajo concerniente a la gestión de becas en Vicedecanatura de la Facultad de Ciencias de la UNAN-León, haciendo de esta forma que el usuario final tenga mayor control sobre la información.

Consideramos que hemos adquirido una buena experiencia, porque nos ha permitido afianzar los conocimientos teóricos inculcados durante el transcurso de nuestra carrera y por supuesto, los conocimientos nuevos obtenidos en la práctica, los cuales estamos seguros serán de gran ayuda en nuestras vidas profesionales.

Esperamos al mismo tiempo que la información que incluye este documento, sirva de soporte para las nuevas generaciones, por que en él se incluye gran parte de nuestro código fuente escrito, para el funcionamiento del sistema y herramientas novedosas para desarrollo de software de libre distribución.



VII. RECOMENDACIONES

Recomendamos desarrollar un modulo para la conexión con la base de datos actual de registro de calificaciones, obteniendo la información necesaria, con objeto de agregarle mas funcionalidad.

Es recomendable crear una copia de seguridad de la Base de Datos periódicamente, para evitar accidentalmente la pérdida de información, esta opción la trae el sistema, y se puede hacer desde él automáticamente.

Implementación de aplicaciones bajo licencia GNU, con el objetivo de promover el uso de software de código abierto.



VIII. BIBLIOGRAFIA

- JAVA2 Interfaces Graficas y aplicaciones para Internet:
Francisco Javier Ceballos.
- Folleto de Análisis y Diseño de Sistemas de Información:
Departamento de Computación UNAN-León.
- Trabajo Monográfico: Sistema de Facturación y planilla de la Empresa
Aguadora de las Peñitas y Poneloya EMAPEPOSA.
Autores: Estudiantes UNAN-León.
Año: 2003.

Internet:

http://dev.mysql.com/doc/mysql/en/GPL_license.html

<http://www.ej-technologies.com/download/exe4j/files.php>

<http://tecfa.unige.ch/guides/tie/html/java-mysql/java-mysql.html>

<http://www.adictosaltrabajo.com>

<http://www.lawebdelprogramador.com>

<http://www.mysql.com>

<http://www.javahispano.com>



ANEXOS



Car né

--	--	--	--	--	--	--	--	--	--

Solicitud de Beca

Instrucciones

- 1- La información que usted brinda en esta solicitud es de carácter confidencial. Formara parte del expediente que mantendrá la división de bienestar estudiantil de la UNAN-LEÓN y será utilizada para los programas que se desarrollen en esta área.
- 2- No se recibirán solicitudes incompletas, con enmiendas, ni presentada fuera del tiempo establecido
- 3- Esta solicitud debe llenarse con letra clara, legible y con bolígrafo.
Los datos y documentos aportados deben ser completamente válidos, en caso contrario su solicitud se anulará
- 5- Es deber del estudiante conocer el reglamento de becas vigente.

I.- Datos Generales

Primer Apellido	Segundo Apellido	Nombre Completo

Sexo

F

M

Fecha de Nacimiento

--	--	--	--	--	--	--	--

Instituto donde realizó su bachillerato _____

Gozo de Beca Si

No

Año que Cursa _____

Facultad _____

Carrera _____

Ha Repetido Año Si

No

Estado Civil

No de Hijos

No de Hermanos

Dirección de Origen _____

Comarca _____

Municipio _____

Depto. _____

Tel _____

Dirección en León _____

Tel _____

II.- Datos socioeconómicos Familiares

La casa de Habitación es:

Propia Alquilada Familiar Otro

De quién recibe apoyo económico _____

En que trabaja la persona que lo apoya _____

Salario C\$ _____ Cuanto le asigna mensualmente C\$ _____

Trabaja su papá Si No Salario C\$ _____

Donde _____ Actividad que Realiza _____

Trabaja su mamá Si No Salario C\$ _____

Donde _____ Actividad que Realiza _____

Número de dependientes de sus padres o de quién le da apoyo

Poseen Propiedades sus padres Si No

En Caso positivo detalle _____ Extensión _____

_____ Extensión _____

Su Familia Posee Vehículo Si No No. de Placa _____ Marca _____

Su Familia Posee algún tipo de Deuda Si No

En caso positivo: Monto C\$ _____ Con quién _____

Su Familia Posee Bienes Inmuebles Si No

En caso positivo: Cuales _____

Trabaja usted Si No Donde _____ Salario C\$ _____

Actividad que Realiza _____

Recibe ayuda del extranjero Si No ¿Cuanto Recibe? C\$ _____

Esta Becado por algún organismo nacional o internacional: Si No

Cuanto Recibe C\$ _____ \$ _____ organismo _____

III.- INFORMACIÓN SOCIAL.

Indique cuales de las siguientes son causas que le impiden o dificultan realizar sus estudios:

- Muerte del principal apoyo económico del grupo familiar
- Orfandad
- Separación del grupo familiar
- Malas relaciones familiares
- Desintegración familiar (divorcio, separación, abandono)
- Problemas emocionales (nerviosismo, depresiones, etc.)
- Problemas de alcoholismo, drogas
- Problemas de salud
- Problemas de desempleo
- Problemas económicos
- Otros _____

(Especifique)

• TIPO DE BECA QUE SOLICITA

Interna

Externa

Residencia

OBSERVACIONES

Anote en este espacio cualquier situación sobre la información socioeconómica del grupo familiar que desee aclarar:

Al firmar esta solicitud el candidato declara que la información proporcionada corresponde a la realidad y se obliga a ampliarla si se presentara cambios o modificaciones posteriores en lo que conviene a la beca solicitada, así mismo, autoriza a la comisión de becas para efectuar cualquier gestión que tienda a comprobar los datos suministrados en su solicitud en caso de faltar a la verdad será suficiente motivo para suspender o rechazar su solicitud.

NOMBRE Y APELLIDO IMPRESO

FIRMA

FECHA

IV.- CONCLUSION:

DATOS A LLENARSE POR LA INSTITUCION

Rendimiento Académico

Antecedente disciplinario:

Otras consideraciones:

En base a la visita realizada y las observaciones señaladas, esta Comisión concluye que el estudiante visitado (a):

Se le otorga beca en la UNAN-León tipo:

Interna

Externa

Residencia

No clasifica para beca en la UNAN-León

Firma del presidente de la comisión

Miembro de la Comisión

Miembro de la Comisión