

**UNIVERSIDAD MAYOR DE SAN ANDRES  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMATICA**



**PROYECTO DE GRADO**

**SISTEMA DE CONTROL DE PROYECTOS EN LA  
FUNDACIÓN ARCO IRIS CASO : PROYECTOS SOCIALES**

**PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMATICA**

**MENCION INGENIERIA DE SISTEMAS INFORMATICOS**

**POSTULANTE : HILDA LLANOS MIRANDA**

**TUTOR : Lic. FRANZ CUEVAS QUIROZ MSc.**

**REVISOR : Lic. CARLOS MULLISACA CHOQUE**

**LA PAZ – BOLIVIA**

**2006**

*Dedicatoria:*

*A mis padres Francisca y  
Félix , por todo su apoyo y  
comprensión.*

*Hilda*

## *Agradecimientos*

*Ante todo gracias a Dios por brindarme toda la fortaleza y guía en mi vida.*

*Al director Ejecutivo de la Fundación Arco Iris Lic. Krzysztof Bobka K. por permitirme realizar el presente trabajo en la mencionada institución, por toda su comprensión y paciencia a la conclusión del mismo.*

*Al personal de la fundación Arco Iris, como ser los coordinadores de los proyectos sociales, y las secretarías por su ayuda y colaboración en la facilitación de documentación necesaria*

*De manera especial al docente tutor MSc. Franz Cuevas Quiroz, por su colaboración y observaciones respectivas, muchas gracias.*

*Al Lic. Carlos Mullisaca Choque por sus observaciones en la revisión del presente trabajo, muchas gracias.*

*Hilda*

## *Resumen*

Hoy en día existen muchos niños abandonados en todo el mundo, mucho más en los países latinos como el nuestro, obligando a muchos de ellos al trabajo a muy temprana edad, o dedicándose a la delincuencia infanto- adolescente es por ello, que viendo todo esto, la Fundación Arco Iris cuenta con programas de ayuda para los niños y adolescentes, es decir existen los proyectos sociales los cuales brindan apoyo económico, material escolar, alimenticio, vestimenta, etc.

Al interior de los proyectos sociales, se realizan actividades y beneficios para los niños albergados en los diferentes hogares de los proyectos, donde se registran gran cantidad de información, expresados en informes y reportes que llegan a la dirección de coordinación de proyectos sociales enviados por los coordinadores de cada proyecto.

Es por ello que viendo la necesidad de contar con un sistema informático que automatice los procesos en esta área, se desarrolla el presente trabajo, que consiste en automatizar el registro, seguimiento, estadísticas, reportes, etc. necesarios sobre los diferentes proyectos, de manera que se pueda contar con información rápida para la dirección de coordinación de proyectos sociales.

De esta manera para el desarrollo del presente proyecto de grado, se utilizó el Proceso Unificado de Rational (RUP), así como la notación del Lenguaje Unificado de Modelado (UML) para su diseño, el sistema fue desarrollado en el lenguaje Visual con un sistema de gestión de base de datos relacional.

En cuanto a la evaluación del software se toma en cuenta los siguientes aspectos como ser: funcionalidad, confiabilidad, mantenibilidad y portabilidad.

## INDICE

### CAPITULO I INTRODUCCION

1.1 Antecedentes.....	2
1.2 Planteamiento del Problema .....	3
1.3 Objetivos .....	4
1.3.1 Objetivo General .....	4
1.3.2 Objetivo Especifico .....	4
1.4 Justificación.....	4
1.4.1 Justificación Técnica .....	5
1.4.2 Justificación Social .....	5
1.4.3 Justificación Económica .....	5
1.5 Alcances.....	5
1.6 Aportes .....	6

### CAPITULO II MARCO TEORICO

2.1 Métodos de Control de Proyectos.....	7
2.1.1 Método Gantt.....	7
2.1.2 Método del camino crítico o el pert / cpm .....	8
2.2 Matriz de Estructura Lógica .....	10
2.3 Análisis Estadístico.....	12
2.4 Metodología Orientada a Objetos.....	15
2.5 Proceso Unificado de desarrollote Software.....	15
2.5.1 Dirigido por Casos de .....	16
2.5.2 Centrado en la Arquitectura .....	16
2.5.3 Fases e Iteraciones.....	18

2.5.4	Componentes del Proceso .....	19
2.5.5	Componentes del proceso y Modelos .....	19
2.5.6	Lenguaje Unificado de Modelado .....	20
2.5.7	Elementos del modelo .....	22
2.6	Herramientas de Programación.....	23

### CAPITULO III MARCO APLICATIVO

3.1	Análisis del Sistema Actual.....	26
3.2	Representación del sistema Actual.....	26
3.3	Representación de Procesos.....	27
3.4	Diseño del sistema.....	30
3.5	Determinación de Requisitos.....	30
3.6	Requerimientos de los Usuarios.....	31
3.7	Requerimientos del Sistema.....	32
3.8	Análisis de Casos de Uso.....	32
3.9	Descripción de los Casos De Uso.....	34
3.10	Diagramas de Secuencia.....	37
3.11	Diagramas de Actividades.....	41
3.12	Modelo Conceptual.....	43
3.15	Arquitectura del sistema.....	48
3.17	Tablas de la Base Datos.....	50
3.18	Diseño de Interfaz.....	53
3.19	Prueba del sistema.....	58
3.20	Prueba de Caja Blanca .....	58

## CAPITULO IV

### CALIDAD DEL SOFTWARE

4.1 Mediciones del software.....	60
4.2 Factores de Calidad .....	61
4.2.1 Funcionalidad.....	61
4.2.2 Confiabilidad .....	65
4.2.3 Mantenibilidad .....	70

## CAPITULO V

### CONCLUSIONES Y RECOMENDACIONES

Conclusiones.....	63
Recomendaciones.....	64
Bibliografía	



## INDICE DE FIGURAS

Fig. 2.1 Representación del diagrama de Gantt.....	8
Fig. 2.2 Representación de diagrama Pert .....	9
Fig. 2.3 Esquema del marco Logico .....	11
Fig. 2.4 Elementos del marco Logico .....	12
Fig. 2.6 Servicios y Beneficios del proyecto.....	14
Fig. 2.7 Proceso de desarrollo de Software .....	15
Fig. 2.8 Casos de Uso se enlazan los flujos de trabajo .....	16
Fig. 2.9 Dimensiones del Proceso Unificado .....	17
Fig. 2.10 Relación de los Componentes de Proceso y modelos.....	20
Fig. 2.11 Elementos Comunes de Modelaje .....	22
Fig. 2.12 Ejemplos de Relación .....	23
Fig. 2.13 Arquitectura cliente Servidor .....	24
Fig. 3.1 Componentes Básicos de Metodología .....	25
Fig. 3.2 Representación del sistema Actual proyecto social .....	26
Fig. 3.3 Representación grafica de proceso de registro de proyecto .....	27
Fig. 3.4 Seguimiento de Proyectos .....	28
Fig. 3.5 Flujo del Proceso Orientada a Objetos .....	30
Fig. 3.6 Diagrama de Casos de Uso.....	33
Fig. 3.7 Diagrama de Secuencia ingreso al sistema .....	38
Fig. 3.8 Diagrama de Secuencia Información del Proyecto .....	39
Fig. 3.9 Diagrama de Secuencia Avance del Proyecto .....	40
Fig. 3.10 Diagrama deActividades .....	41
Fig. 3.13 Diagrama de Clases .....	45
Fig. 3.14 Diseño de la Base de Datos Relacional .....	46
Fig. 3.16 Arquitectura del sistema .....	49
Fig. 3.17 Estructura Modular del sistema .....	51
Fig. 3.18 Diseño de Interfaz del sistema .....	53
Fig. 4.1 Modelo del sistema .....	66
Fig. 4.2 Diagrama de Bloques .....	66
Fig. 4.3 Subsistemas de Sicofai .....	67

## TABLAS

Tabla 2.1 Distribución de Frecuencias por clases .....	13
Tabla 2.2 Distribuciones categóricas de servicios /beneficios.....	14
Tabla 2.3 Servicios y Beneficios.....	16
Tabla 3.1 Elementos de planificación de un Proyecto .....	29
Tabla 3.2 Caso de Uso Ingreso al sistema .....	34
Tabla 3.3 Caso de Uso Información General del Proyecto .....	34
Tabla 3.4 Caso de Uso Asignación de Responsable .....	35
Tabla 3.5 Caso de uso control de Proyecto .....	35
Tabla 3.6 Caso de Uso Puesta en Marcha del Proyecto .....	36
Tabla 3.7 Caso de Uso Mostrar Reporte .....	37
Tabla 3.8 Caso de Uso Acceso a Beneficios .....	39
Tabla 4.1 Calculo de Puntos de función .....	62
Tabla 4.2 Valores de Ajuste .....	63
Tabla 4.3 Calculo de Complejidad .....	64

## CAPITULO I

### INTRODUCCION

En cualquier parte del mundo existen niños que viven en las calles, solos y sin el apoyo del gobierno, de la sociedad, y sin la ayuda de sus padres o familiares.

En La Paz sede de gobierno miles de niños, niñas y adolescentes sobreviven en las calles con unas cuantas monedas ya sea lavando carros, lustrando zapatos, vendiendo dulces, etc. a muchos de ellos se los conoce con el sobrenombre de desechables, quienes viven de limosnas o el hurto.

La existencia de estos menores abandonados por su familia, que día con día va en aumento y que tal parece no interesarle a nuestra sociedad, mucho menos al estado boliviano, nace una fundación para el apoyo de estos menores.

La fundación Arco Iris desde hace 10 años atrás, se constituye en un símbolo de esperanza, amistad y solidaridad procurando proporcionar a los menores desamparados afecto y un hogar, donde recuperen la fe y la esperanza pérdidas y tengan la posibilidad de rehabilitarse y reinsertarse en la sociedad, capacitándose y desarrollándose como personas de bien.

La fundación Arco Iris hace esfuerzos que están destinados a tender una mano amiga a la población trabajadora que son los infante adolescentes que han sido abandonados o cuyos padres se encuentran en prisión. También esta apoyando a escolares con alto riesgo de deserción escolar, para impedir que la abandonen.

La fundación Arco Iris, hace realidad el desarrollo de proyectos sociales, productivos y de salud, para contribuir en el proceso de sensibilización para erradicación de las peores formas del trabajo infantil.

Por lo expuesto anteriormente, el presente trabajo pretende realizar un sistema de información de control de proyectos, específicamente al interior de los Proyectos sociales de la fundación Arco Iris.

## 1.1 ANTECEDENTES

Respecto a los trabajos similares con el presente, se encontraron proyectos de grado en la carrera de Informática las cuales se detallan a continuación:

Sistema Informático de Administración y Control de Proyectos para el Centro de Investigación, Educación y Servicios (C.I.E.S.). [CABRERA1997]

Se desarrollo un sistema informático para los procesos de administración y control de los proyectos de salud de C.I.E.S.

Sistema de Control de Proyectos Inti Watana [IRAHOLA2001]

En este proyecto de grado se desarrollo un sistema informático para el Control y seguimiento de los Proyectos sociales, culturales y educativos del Centro Alternativo de Desarrollo Integral (CADI).

Sistema de Información Evaluación seguimiento de Proyectos Iglesia Evangélica [CENTELLAS, 2002]

Gestiona proyectos que se generan en las diferentes comunidades, pertenecientes al distrito de la iglesia metodista, los proyectos que se contempla es la construcción de postas médicas, de iglesias, casas pastorales, etc.

Sistema de Seguimiento y Control a Proyectos a la Empresa Municipal de Mantenimiento Urbano (E.M.M.U.). [CATACORA, 2002]

Se desarrollo un sistema informático para mejorar el Control del uso de Materiales, Maquinaria y/o Equipo asignado a los proyectos de Obras Municipales.

Sistema de Monitoreo y control de Proyectos PAN [TORREZ, 2004]

Se desarrollo un sistema de monitoreo y control de proyectos que esta dirigido a la administración de los recursos físicos de los proyectos.

Sistema de Información para el Proyecto Integrado de Desarrollo Infantil (PIDI)

[VALDEZ, 2004]

Se desarrollo un sistema de información en la unidad de Coordinación del PIDI y en las unidades de Supervisión.

Sistema De Control y Seguimiento de proyectos Cristian Childrens Fund Bolivia [MAMANI, 2004]

Se desarrollo un sistema informático para el registro, seguimiento y control de proyectos. Específicamente se realizo un seguimiento y control de correspondencia.

## 1.2 PLANTEAMIENTO DEL PROBLEMA

Luego de realizar un estudio preliminar de la información que se maneja dentro de la Fundación Arco Iris, entrevistas al personal encargado de la institución y la observación de las actividades que se desarrollan diariamente en los proyectos y en la institución , se observo que:

- La información referente a los diferentes proyectos es almacenada en archivos físicos en el interior de cada proyecto.
- Existe una demora a reportes de seguimiento y control de proyectos.
- No se cuenta con información estadística sobre el número de beneficiarios de cada proyecto.
- No se cuenta con un registro adecuado de las propuestas de proyectos presentados a consideración.
- No existe información oportuna para determinar el éxito de un proyecto.

Con todo lo mencionado anteriormente se plantea la siguiente interrogante:

*¿El sistema de información específicamente para el área de Proyectos –Sociales mejorara los procesos y brindara información oportuna, a partir de informes y reportes para una mejor toma de decisiones de la institución?*

## 1.3 OBJETIVOS

### 1.3.1 OBJETIVO GENERAL

Diseñar y desarrollar un sistema de información para el control y seguimiento de los proyectos presentados por los responsables de cada uno de ellos, para contar con información rápida y confiable para la toma de decisiones específicamente en la coordinación de Proyectos Sociales.

### 1.3.2 OBJETIVOS ESPECIFICOS

- Satisfacer la demanda de información de la Coordinación de Proyectos Sociales.
- Recopilar y sistematizar la información dispersa en el interior de La coordinación de proyectos sociales.
- Proporcionar reportes estadísticos de los proyectos.
- Establecer niveles de acceso al sistema.
- Desarrollar un modulo de registro de la personas, instituciones o empresas con las que la fundación trabaja.
- Desarrollar un módulo de registro de empresas o instituciones y personas que colaboran de alguna manera con la fundación Arco Iris.

## 1.4 JUSTIFICACIÓN

Luego de haber realizado un análisis del área de proyectos sociales de la Fundación Arco Iris, existe la necesidad de elaborar un sistema de información que mejore el seguimiento y control de los proyectos .Por lo tanto el proyecto se justifica Técnico, Económico y social.

### 1.4.1 JUSTIFICACION TÉCNICA

La necesidad de contar con un sistema automatizado que ayude en la mejora de la información, que permita llevar un control y seguimiento de los proyectos para la mejor toma de decisiones.

### 1.4.2 JUSTIFICACION SOCIAL

Con la implantación de un sistema de información para la fundación Arco Iris, será de gran beneficio para los responsables de cada proyecto y para el personal del directorio de la institución.

### 1.4.3 JUSTIFICACION ECONOMICA

El esfuerzo y el tiempo que emplea el personal de la institución será mucho menor, pues el acceso a la información será mucho más rápido y eficiente.

## 1.5 ALCANCES

El presente trabajo contemplará los siguientes aspectos:

- Módulo de recepción y registro de proyectos.
- Módulo de seguimiento a Proyectos en ejecución.
- Módulo de generación de reportes.
- Módulo de estadísticas de los beneficiarios
- Realizar la capacitación del personal encargado del manejo del software

## 1.6 APORTES

El aporte será el desarrollo del sistema de Información para el seguimiento y control de proyectos sociales, utilizando todas las herramientas necesarias para tal fin.

Con el desarrollo de este sistema de información para la Fundación Arco Iris, facilitará el manejo de los datos e información que se maneja a beneficio de la institución y para las personas beneficiadas.

## CAPITULO II

### MARCO TEORICO

A continuación se hace referencia a los conceptos de los términos proyecto y control.

“El **proyecto** no es mas que un modelo del emprendimiento a ser realizado, con las previsiones de recursos, de tiempo de ejecución y de resultados esperados.” [IBARROLA,1972]

“El termino proyecto se refiere a la menor unidad de actividad que puede ser planificada y ejecutada aisladamente” [SOLOMON MORRIS ,1972]

Las Naciones Unidas definen proyecto como el “conjunto de antecedentes que permite estimar las ventajas y desventajas económicas que se derivan de asignar ciertos recursos de un país para la producción de determinados bienes o servicios”. [NACIONES UNIDAS 1958]

Cualquiera de estas definiciones es útil aunque cada una de ellas enfatice en la concepción del proyecto representa una propuesta concreta de inversión, adecuadamente caracterizada en términos de sus componentes técnicos, económicos, financieros, organizacionales, institucionales y legales. [NACIONES UNIDAS 1958]

**Control**, el concepto de control es una actividad trivial, que forma parte de la vida cotidiana del ser humano, conscientemente o no.

La finalidad básica del control es la modificación del comportamiento de la persona u objeto que se controla.

El control es una función que se realiza mediante parámetros que han sido establecidos anteriormente al acaecimiento del fenómeno controlado, es decir, el mecanismo de control es fruto de una planificación, y por lo tanto apunta al futuro.

**Monitoreo** o Seguimiento se define como un proceso organizado de verificación de que una determinada situación o proceso transcurre como se había previsto, a través de la comparación de las actividades implementadas con los planes de trabajo, incluyendo si las actividades están siendo realizadas de acuerdo a dicho plan, si se están llevando a cabo dentro del marco de tiempo especificado, si se necesita de cualquier cambio en el manejo o implementación de las actividades y si debería modificarse el plan de trabajo.

**Evaluación**, un proceso de formación de juicios acerca de una determinada situación o proceso, es importante resaltar la conexión entre juicios y la toma de decisiones.

## 2.1 METODOS DE CONTROL DE PROYECTOS

**2.1.1 GANTT.-** Los cronogramas de barras o “gráficos de Gantt” fueron concebidos por el ingeniero norteamericano Henry L. Gantt, uno de los precursores de la ingeniería industrial y contemporánea de Taylor. Gantt procuro resolver el problema de la programación de actividades, es decir su distribución conforme a un calendario, de manera tal que se pudiese visualizar el periodo de duración de cada actividad, sus fechas de iniciación y terminación e igualmente el tiempo total requerido para la ejecución de un trabajo.

El gráfico de Gantt, consiste simplemente en un sistema de coordenadas en que se indica:

- ✓ Eje horizontal – un calendario, o escala de tiempo definido en términos de la unidad mas adecuada al trabajo que se va a ejecutar: hora, día, semana, mes, etc.
- ✓ Eje vertical – las actividades que constituyen el trabajo a ejecutar.

A cada actividad se hace corresponder una línea horizontal es proporcional a su duración y cuya medición se efectúa con relación a la escala definida en el eje horizontal. [BID, 1990]

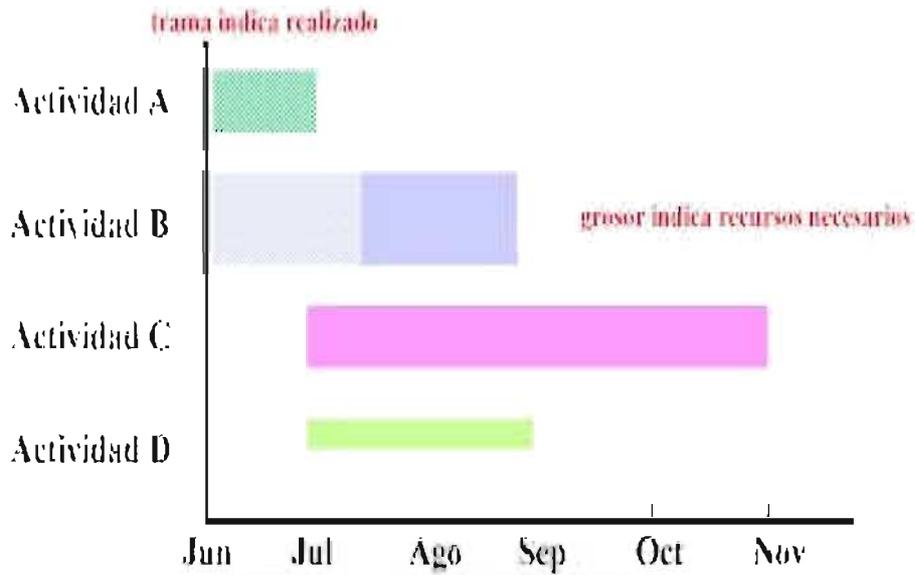


Figura 2.1: Representación de Diagrama de Gantt

Fuente: [BID, 1990]

### 2.1.2 METODO DEL CAMINO CRITICO O EL PERT / CPM

Hasta una época determinada, los instrumentos de control disponibles no permitían al administrador prever lo que iba a acontecer, o por lo menos establecer relaciones precisas entre las diversas etapas de una empresa. Los cronogramas de ejecución de obras, como las cartas de Gantt, no funcionaban con precisión cuando los proyectos adquirían cierta complejidad.

Los métodos de programación y control fracasaban cada vez que el problema adquiría alguna complejidad.

Uno de los primeros campos a atacar fue el relacionado con la planificación y programación de los proyectos de construcción.

En 1957, el Univac Applications Research Center, dirigido por John w \_Mauchly, la Remington Rand (División Univac) bajo la dirección de James E. Sëller Jr. Y la compañía Dupont, representada por Morgan Walker, se asociaron para prever el trabajo inicial.

De esta asociación surgieron las rutinas básicas del actual CPM (Critical Path Method)

**PERT.**- Sistema basado en la teoría de los gráficos recibió el nombre de Program Evaluation and Review Technique. Su objetivo primordial fue proporcionar al administrador un nuevo proceso de racionalización de la secuencia de operaciones de un proyecto, de modo de facilitar su ejecución y un control eficaz durante esa ejecución.

El PERT es un sistema técnico que disciplina la elaboración de un proyecto y el control de su ejecución, a través de la utilización de diagramas y de la representación grafica.

Los diagramas PERT se confeccionan con círculos que representan los eventos y flechas que representan las actividades. En la práctica muchas veces los círculos se sustituyen por óvalos, rectángulos o hexágonos, donde se colocan números correspondientes a orden, tiempos y holguras. En las flechas se acostumbra insertar pequeños rectángulos o círculos donde se indican los tiempos de duración de la actividad. Estas mismas flechas toman también disposiciones muy variadas, ya de arcos de circunferencia, ya siguiendo caminos geométricos. [BID, 1990]

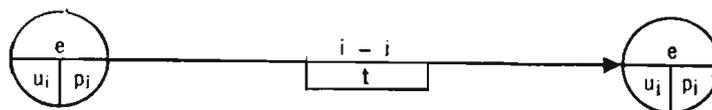


Figura 2.2: Representación de diagrama Pert

Fuente: [BID, 1990]

Las técnicas del PERT y del CPM son de tal manera semejantes que no han resistido las innumerables tentativas de diversificarlas y de mantenerlas en campos opuestos. El CPM, originado en la empresa privada, dio énfasis en las evaluaciones determinísticas y el factor costo, en tanto que el PERT, por lo menos inicialmente, solo dio importancia al factor tiempo y a las técnicas probabilísticas para estimarlo.

## 2.2 MATRIZ DE ESTRUCTURA LOGICA

Los proyectos presentados son, por lo general, prolijos obligando a un trabajo de lectura, demasiado prolongado que se extiende por muchos días. Y lo que pasa es que no siempre los autores de los proyectos consiguen dar una idea completa de la estructura del mismo en las primeras páginas, como se enseña en las técnicas de elaboración de proyectos. Con la intención de simplificar el trabajo de los analistas y de las personas que tendrán a su cargo, en el futuro el control del proyecto, la USAID (United States Agency for Industrial Development) creó la matriz de Estructura Lógica, también conocida como Estructura Lógica o Marco Lógico del Proyecto. El marco lógico facilita una visión rápida e inmediata de la estructura del proyecto. [BID, 1990]

Los estudios que permitirán llegar al Marco Lógico fueron realizados a partir de centenares de proyectos de naturaleza social y económica. Se pueden aceptar como válidos, tanto de vista táctico como estratégico.

El marco lógico es una matriz de  $4 \times 4$  que aclara el objetivo del proyecto y provee una estructura que permite el planeamiento y evaluación del mismo. La estructura abarca un conjunto de conceptos interrelacionados que definen por qué un proyecto está siendo desarrollado y que debe hacerse para alcanzar el resultado deseado. De acuerdo con el concepto de matriz, el marco lógico se caracteriza por la incorporación en un solo formato, de dos tipos de pensamiento:

- ✓ Una lógica vertical.- que aclara la razón por la cual se llevara a cabo el proyecto (¿por qué?).

La lógica vertical, es la que establece la razón por la que se llevara el proyecto.

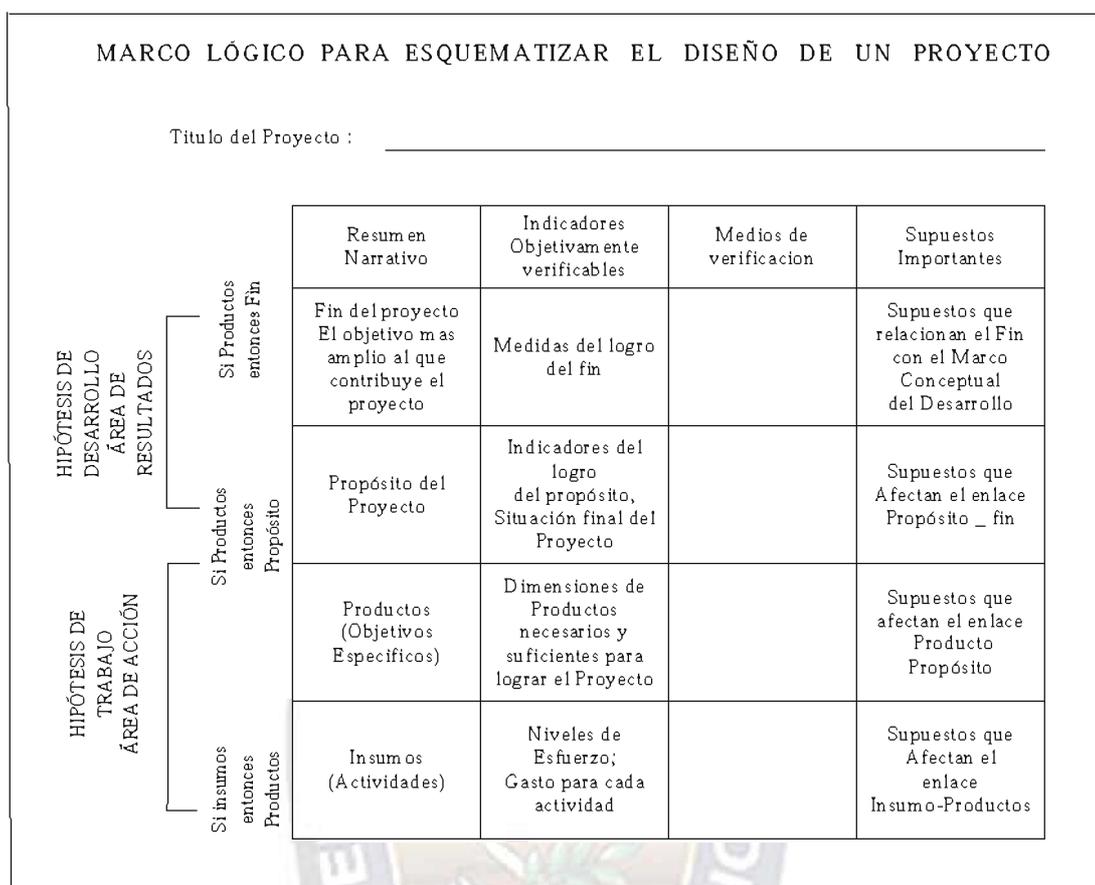


Figura 2.3: Esquema del Marco Lógico

Fuente: [BID, 1990]

Ella puede ser definida por una sigla: DOPI en la que las diversas letras significan:

**D** Directriz, es el objetivo global del programa, es el objetivo más amplio al que el proyecto contribuye.

**O** Objetivo del Proyecto, que es esencialmente el tema que se va a desarrollar y que debe estar encuadrado en el objetivo global o directriz.

**P** Productos, que son los resultados de la adecuada aplicación de los recursos y que permiten que el objetivo del proyecto sea alcanzado.

**I** Insumos, que son los elementos que debemos proveer para que con ellos se pueda obtener los productos necesarios.

- ✓ Una lógica horizontal – que aclara que va a ser producido y de que manera puede prever el suceso (¿con que?). [BID, 1990]

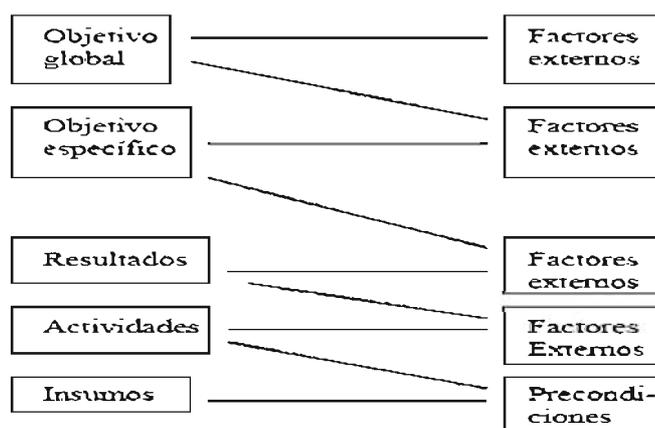


Figura 2.4: Elementos de marco lógico

Fuente: [BID, 1990]

El término de lógica horizontal, quiere decir que la definición requiere las evidencias necesarias para demostrar un evento.

La lógica horizontal se manifiesta, en los diferentes niveles de Directriz, Objetivo, Productos, Insumos (DOPI), a través de dos elementos bien definidos:

- Indicadores objetivamente verificables, que demuestran que el resultado deseado fue alcanzado.
- Medios de verificación, que son los mecanismos específicos a través de los cuales será objetivamente verificada

### 2.3 ANÁLISIS ESTADÍSTICO

Para Poder realizar un análisis estadístico, sobre los casos de interacción y coordinación en el seguimiento y conclusión de los mismos, se utilizara técnicas estadísticas como ser: medidas de tendencia central, distribución de frecuencias, tablas cruzadas, representación tabular o cuadros estadísticos, representaciones gráficas y series cronológicas. Para esto, los datos serán

recolectados y analizados, para luego clasificarlos, tabularlos y representarlos, de forma que los datos sean consistentes, válidas y confiables.

## MEDIDAS DE TENDENCIA CENTRAL

Son medidas que muestran valores típicos o representaciones de un conjunto de datos, se sitúan hacia el centro del conjunto de datos ordenados.

## DISTRIBUCIÓN DE FRECUENCIAS

La distribución de frecuencias es un medio para condensar o agrupar un conjunto relativamente extenso de datos, esta formada por valores observados, clasificados de acuerdo a una distribución de clases, los cuales están agrupados, en un orden determinado. Las representaciones de frecuencias relativas, absolutas y acumuladas son de gran importancia para el análisis de datos. La tabla 2.1 muestra un ejemplo de distribución de frecuencias y el cálculo de la media aritmética.

edad	$n_i$	$x_i$	$x_i n_i$
0 - 3	3	2	6
4 - 6	10	5	50
7 - 9	6	7	42
10 - 12	11	11	121
13 - 15	2	12	24
16 - 18	1	16	16
total	33		259

Tabla 2.1. Distribución de frecuencias por clases para la población Beneficiaria

Fuente: [Fundación Arco Iris]

Donde:

$n_i$  : representa la frecuencia absoluta

$x_i$  : representa la variable observada

## DISTRIBUCIONES CATEGÓRICAS

Una distribución de frecuencias categóricas demuestra el número o la proporción, de observaciones que corresponde a cada una de las clases cualitativas mutuamente exclusivas.

En la figura 2.5 se representa en forma grafica los datos de la tabla 2.2 en esta la escala vertical indica las categorías y la escala mide las frecuencias de cada categoría.

servicios	Frecuencia	
	Absoluta	Relativa
Apoyo vivienda	5	0.08
apoyo vestimenta	18	0.3
Apoyo alimentación	13	0.22
Apoyo profesional	9	0.15
apoyo para capacitación	5	0.08
Ayuda en salud	10	0.17
Total	60	1

Tabla 2.2: Distribuciones Categóricas de los Servicios y beneficios en un mes

Fuente: [Fundación Arco Iris]



Figura 2.6: Servicios y Beneficios en un mes de un proyecto

Fuente: [Fundación Arco Iris]

## 2.4 METODOLOGÍA ORIENTADA A OBJETOS

La popularidad de las tecnologías de objetos ha generado métodos análisis Orientados a Objetos desde finales de los 80 y durante los 90. Cada uno de ellos introduce un proceso para el análisis de un producto o sistema, un conjunto de modelos que evoluciona fuera del proceso y una notación que posibilita al ingeniero del software crear cada modelo de una manera consistente.[INGENIERIA DE SOFTWARE , PRESSMAN]

## 2.5 PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE (RUP)

El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. (Ver figura 2.7)

Sin embargo el Proceso Unificado es mas que un simple proceso; es un marco de trabajo genérico que puede especializarse en una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes tipos de tamaños de proyecto.[Jacobson, Booch, Rumbaugh,2000]

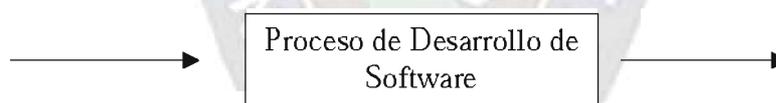


Figura 2.7: Proceso de desarrollo de software

Fuente: [JACOBSON, 2000]

El proceso Unificado esta basado en <sup>1</sup>**componentes**, lo cual quiere decir que el sistema software en construcción esta formado por componentes software, interconectados a través de **interfaces** bien definidas.

El proceso unificado utiliza el Lenguaje Unificado de Modelado (Unified Modelling Lenguaje, UML) para preparar todos los esquemas de un sistema de software. De hecho, UML, es una parte esencial del Proceso Unificado.

---

<sup>1</sup> Elementos, relaciones y diagramas

No obstante, los verdaderos aspectos definatorios del Proceso Unificado se resumen en tres frases clave:

- Dirigido por casos de Usos
- Centrado por la arquitectura
- Iterativo e Incremental

### 2.5.1 DIRIGIDO POR CASOS DE USO

Los casos de uso representan los requisitos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso, el cual describe la funcionalidad total del sistema. Sin embargo, los casos de uso no son una herramienta para especificar los requisitos de un sistema. También guían su diseño, implementación y prueba; esto es guían el proceso de desarrollo.

Dirigido por casos de usos quiere decir que el proceso de desarrollo sigue un hilo, avanza a través de una serie de flujos de trabajo que parten de los casos de uso.

Los casos de usos también nos ayudan a desarrollar interfaces de usuario que hacen más fácil a los usuarios el desempeño de sus tareas. Los casos de uso no solo inician un proceso de desarrollo sino que lo enlazan como se muestra en la figura. 2.8 [JACOBSON]

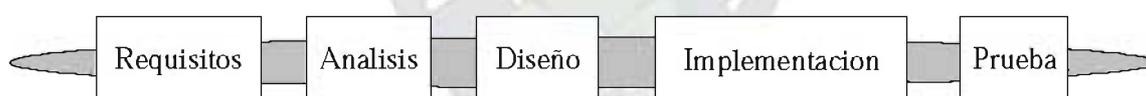


Figura 2.8: Casos de uso se enlazan los flujos de trabajo fundamental

Fuente: [JACOBSON, 2000]

### 2.5.2 CENTRADO EN LA ARQUITECTURA

El papel de la arquitectura de software es parecido al papel que juega la arquitectura en la construcción de edificios. El edificio se contempla desde varios puntos de vista: estructura, servicios, conducción de la calefacción, fontanería, electricidad, etc. Análogamente, la

arquitectura en un sistema software se describe mediante diferentes vistas del sistema de construcción.

El concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema. [JACOBSON, 2000]

### 2.5.3 ITERATIVO E INCREMENTAL

El desarrollo de un producto software supone un gran esfuerzo que puede durar entre varios meses hasta posiblemente un año o más. Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento.

Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto.

El proceso unificado puede ser en dos dimensiones:

- ✓ A lo largo del tiempo, los aspectos del ciclo de vida del proceso y como se desarrolla.
- ✓ A lo largo de los componentes del proceso, que agrupa a las actividades lógicamente por naturaleza.

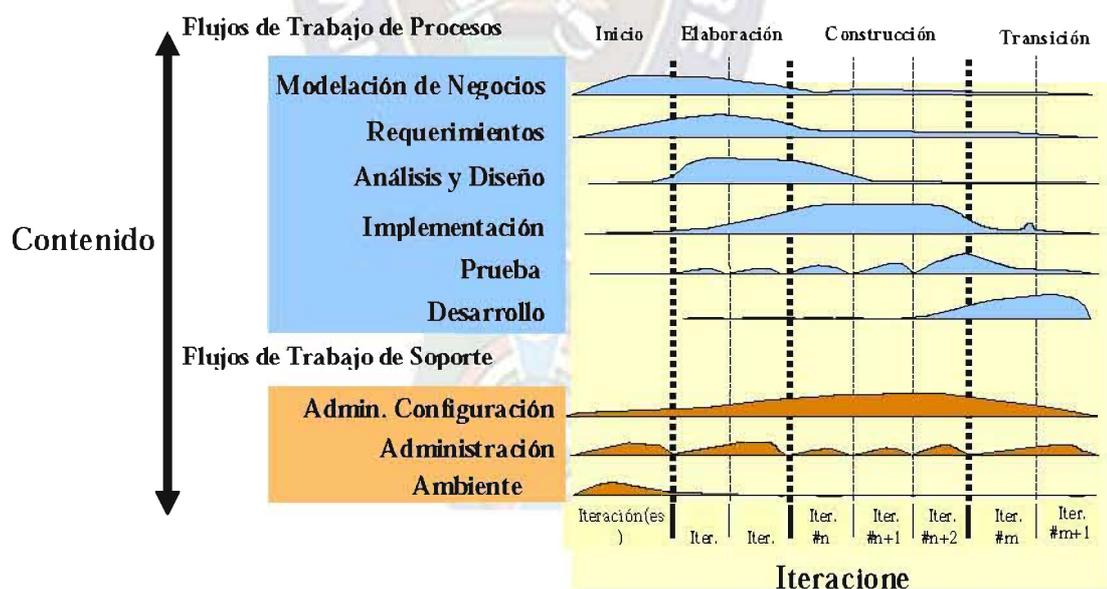


Figura 2.9: Dimensiones del Proceso Unificado

Fuente: [JACOBSON, 2000]

La primera dimensión representa el aspecto dinámico del proceso, como es promulgado, y como es expresado, en términos de ciclos, fases, iteraciones e hitos.

La segunda dimensión representa el aspecto estático del proceso: como es descrito en términos de los componentes del proceso, actividades, flujos de trabajo, artificios y trabajadores.

#### **2.5.4 FASES E ITERACIONES**

Estas conforman la organización dinámica del proceso a lo largo del tiempo.

El ciclo de vida del software esta dividido en ciclos, cada ciclo trabaja en una nueva generación del producto. El proceso Unificado se divide en un ciclo de desarrollo de cuatro fases consecutivas:

- ✓ Fase de Inicio
- ✓ Fase de Elaboración
- ✓ Fase de Construcción
- ✓ Fase de Transición

##### **FASE DE INICIO**

En esta fase se trabaja en el caso del negocio para el proyecto; cuanto costara y cuanto traerá de beneficio y también se establecen los casos de uso del negocio y se delimita el alcance del proyecto

##### **FASE DE ELABORACIÓN**

Las metas de la fase de elaboración son analizar el dominio del problema, establecer una fundación arquitectónica sólida. Las decisiones arquitectónicas deben ser tomadas con un entendimiento del sistema completo.

Hasta este punto se tiene la autorización para iniciar el proyecto. En esta etapa típicamente tiene una idea vaga de los requerimientos.

## FASE DE CONSTRUCCIÓN

Durante esta fase de construcción iterativamente e incrementalmente se desarrolla un producto que esta lista para pasar a su comunidad de usuarios. Cada iteración es un mini proyecto análisis, diseño, codificación, pruebas.

Finaliza la iteración con una demostración al usuario y se realizan pruebas del sistema para confirmar que los casos de uso han sido construidos correctamente.

## FASE DE TRANSICIÓN

Durante la fase de transición se realiza la transición del software a la comunidad de usuarios. Una vez que el producto ha sido puesto en las manos de los usuarios, a menudo surgen problemas que requieren desarrollo adicional para ajustar el sistema, corregir errores no detectados o finalizar algunas de las características que hubieran sido pospuestas. Esta fase típicamente inicia con una versión "beta" de los sistemas.

### 2.5.5 COMPONENTES DEL PROCESO

El proceso unificado esta compuesto de siete componentes de proceso, que están descritos en términos de actividades. Flujos de trabajo, trabajadores y artificios. Hay cuatro componentes del proceso de ingeniería:

Captura de requerimientos, Análisis y Diseño, Implementación y Prueba

Y tres procesos de soporte:

Administración, Despliegue y Entorno.

### 2.5.6 COMPONENTES DEL PROCESO Y MODELOS

El Proceso Unificado tiene los siguientes modelos; modelo de casos de uso, modelo de diseño, modelo de Implementación y modelo de Prueba. Ver figura que se muestra a continuación

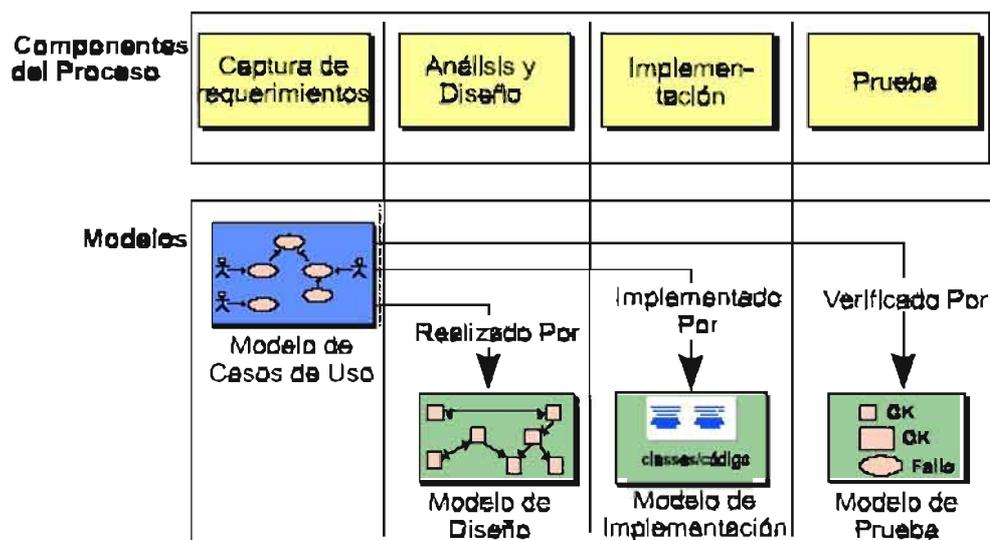


Figura 2.10: Relación de los componentes de proceso y modelos

Fuente: [JACOBSON, 2000]

Al final de la década pasada, Grady Booch, James Rumbaugh e Ivar Jacobson empezaron a colaborar para combinar y recopilar las mejores características de cada uno de métodos de diseño y análisis orientado a objetos en un método unificado. El resultado, denominado lenguaje de Modelado Unificado (UML) se ha convertido en el método más utilizado por la industria.

### 2.5.7 LENGUAJE UNIFICADO DE MODELADO (UML)

UML permite expresar un modelo de análisis utilizando una notación de modelado con unas reglas sintácticas, semánticas y prácticas.

En UML, un sistema viene representado por vistas diferentes que lo describen desde diferentes perspectivas. Cada vista se representa mediante un conjunto de diagramas. En UML están presentes las siguientes vistas.

- Vista del usuario, representa el sistema (producto desde la perspectiva de los usuarios (llamados actores en UML). El caso de uso es el enfoque elegido para modelar esta vista.

- Vista Estructural: los datos y la funcionalidad se muestran desde dentro del sistema, es decir, modela la estructura estática (clases, objetos y relaciones).
- Vista del comportamiento: esta parte del modelo del análisis representa los aspectos dinámicos o de comportamiento del sistema. También muestra las interacciones o colaboraciones entre los diversos elementos estructurales descritos en las vistas anteriores.

A continuación se menciona una breve descripción de cada uno de los conceptos mencionados en cada una de las vistas.

Caso de Uso es, en esencia, una interacción típica entre un usuario y un sistema de cómputo.

- ✓ El caso de uso capta alguna función visible para el usuario.
- ✓ El caso de uso puede ser pequeño o grande.
- ✓ El caso de uso logra un objetivo discreto para el usuario.

### **Diagrama de casos de uso**

Muestra las distintas operaciones que se esperan de una aplicación o sistema y como se relaciona con su entorno. (Usuarios u otras aplicaciones).

### **Diagrama de clases**

Un diagrama de clases contiene ciertos elementos materializados de comportamiento, como operaciones, pero cuya dinámica está representada en otros diagramas, como diagramas de estados o diagramas de colaboración.

### **Diagrama de Secuencia**

Un uso de un diagrama de secuencia es mostrar la secuencia del comportamiento de un caso del uso. Cuando está implementado el comportamiento, cada mensaje en un diagrama de secuencia corresponde a una operación en una clase, a un evento disparador, o a una transición en una máquina de estados.

### **Diagrama de Colaboración**

Una colaboración modela los objetos y los enlaces significativos dentro de una interacción.

Un uso de un diagrama de colaboración es mostrar la implementación de una operación. La colaboración muestra los parámetros y las variables locales de la operación.

#### Diagrama de Actividades

Un estado representa una actividad: un paso en el flujo de trabajo o la ejecución de una operación: los grafos de actividades se muestran en diagramas de actividades.

Diagramas de interacción son modelos que describen la manera en que colaboran grupos de objetos para cierto comportamiento. Habitualmente, un diagrama de interacción capta el comportamiento de un solo caso de uso.

### 2.5.8 ELEMENTOS DEL MODELO

Los conceptos utilizados en los diagramas son llamados elementos del modelo. Un elemento del modelo también tiene un elemento de vista correspondiente, el cual es una representación gráfica del elemento o el símbolo gráfico utilizado para representar al elemento en los diagramas.

En la siguiente figura se muestra algunos ejemplos de elementos del modelo tales como clase, objeto, estado, caso de uso, nodo, interfaz, actor, etc.

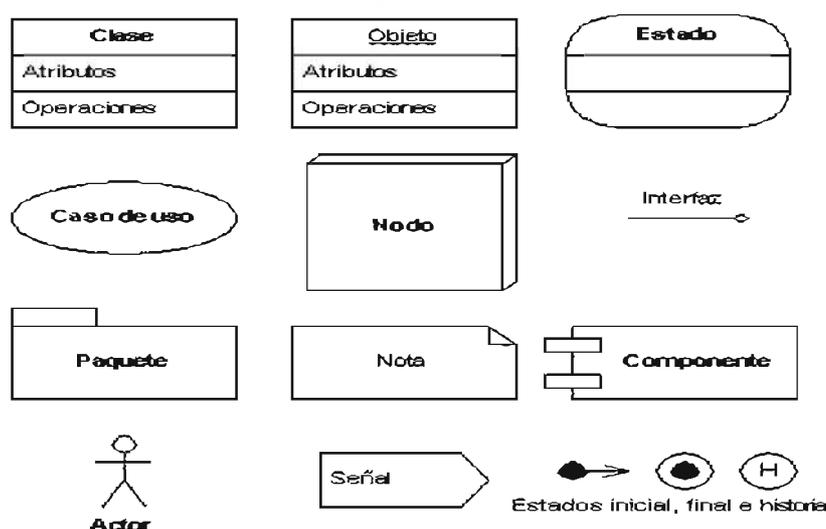


Figura 2.11: Elementos comunes de Modelaje

Fuente: [JACOBSON, 2000]

Las relaciones son también elementos del modelo y son utilizados para interconectar otros elementos del modelo unos a otros. Algunas relaciones diferentes son:

- ✓ Asociación: conecta elementos y enlaza instancias.
- ✓ Generalización: llamada también herencia, esto significa que un elemento puede ser una especialización de otro elemento.
- ✓ Dependencia: muestra que un elemento depende de alguna manera de otra.
- ✓ Agregación: es una forma de asociación en la cual un elemento contiene otros elementos.
- ✓ Refinamiento: es una forma de generalización entre un elemento a mayor nivel de detalle que otro, pero que representan lo mismo.

La siguiente figura muestra ejemplos de las relaciones antes descritas.



Figura 2.12: Ejemplos de Relaciones

Fuente: [JACOBSON, 2000]

## 2.6 HERRAMIENTAS DE PROGRAMACIÓN

### VISUAL BASIC

La razón principal por la que Visual Basic es tan popular y potente es la misma del éxito de Windows. Microsoft abordó una tecnología compleja (escribir programas de computadora) y facilitó su uso a través de una interfaz gráfica. En un entorno de programación visual, se puede diseñar rápidamente las ventanas que el usuario ve, dibujándolas y organizándolas de la

misma forma que se haría para el diseño de un periódico

## SQL SERVER

SQL Server es un sistema administrador para Bases de Datos relacionales basadas en la arquitectura Cliente / Servidor (RDBMS) que usa Transact-SQL para mandar peticiones entre un cliente y el SQL Server.

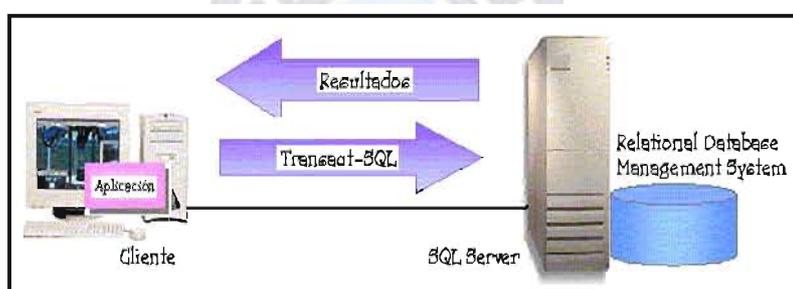


Fig.2.13: Arquitectura Cliente Servidor

Fuente: [VAUGHN, 1999]

Toda acción que se efectúe en SQL Server invariablemente se llevara a cabo mediante la ejecución de comandos Transact-SQL por parte del gestor. Sin embargo, estos comandos pueden ser enviados al servidor de muy diversas maneras. Estos caminos para indicar al servidor que comandos deseamos ejecutar suelen denominarse interfaces de programación.

La arquitectura Cliente /Servidor permite desarrollar aplicaciones para realizar en una variedad de ambientes.

### INTERFAZ DE PROGRAMACIÓN

ODBC (Open Database Connectivity)

Es una API a nivel de llamada que define un método de conexión a fuentes de datos abiertos, de manera que cualquier desarrollador pueda acceder a los datos desde cualquier plataforma que disponga de esta funcionalidad.

## CAPITULO III

### MARCO APLICATIVO

Teniendo en cuenta que una metodología es “un conjunto de modelos, lenguajes y otras herramientas que nos facilitan la representación de los datos en cada fase del proceso de diseño de una base de datos, junto con las reglas que permiten el paso de una fase a la siguiente”, el análisis de todos estos elementos es fundamental para poder comprender y aplicar correctamente una metodología de diseño.

A continuación se muestra gráficamente los elementos de una metodología

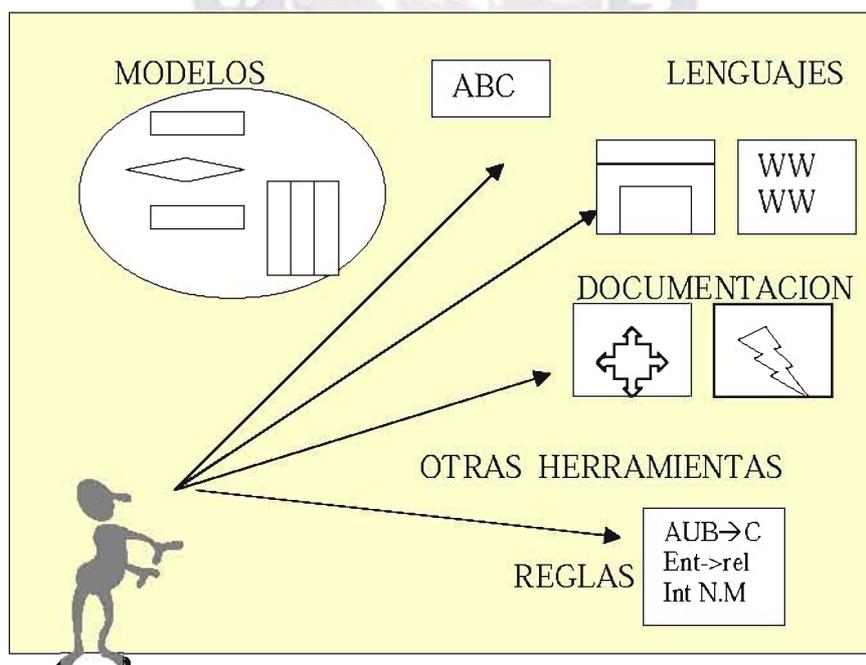


Figura 3.1: Componentes Básicos de la Metodología

Fuente: [PIATTINI, 2000]

### 3.1 ANÁLISIS DEL SISTEMA ACTUAL

A continuación se realizara una breve descripción del sistema actual que funciona en la fundación Arco Iris, es decir respecto del manejo de la información de los proyectos existentes en su interior específicamente en el área social.

En la Fundación Arco Iris, existen proyectos en distintas áreas como ser:

- ✓ Proyectos productivos
- ✓ Proyectos sociales
- ✓ Proyectos en salud

El objeto de estudio y análisis del presente trabajo es el área de **proyectos sociales**. A continuación se muestra gráficamente el funcionamiento de esta área.

### 3.2 REPRESENTACIÓN DEL SISTEMA ACTUAL

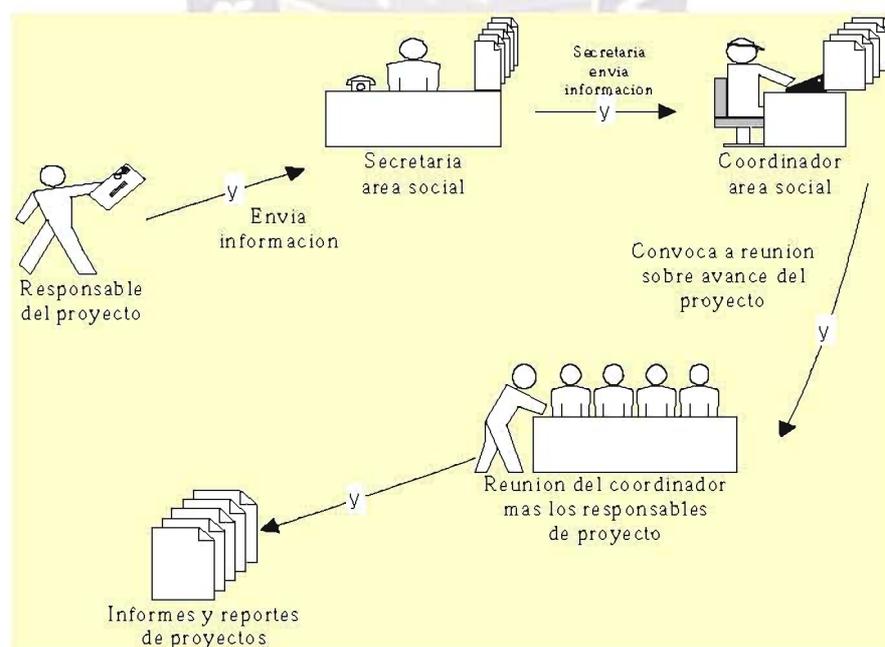


Figura 3.2: Representación del sistema actual proyecto sociales

Fuente: [Elaboración propia]

### 3.3 REPRESENTACION DE PROCESOS

En esta etapa se describen algunos de los procesos actuales que se llevan a cabo en el área de proyectos sociales, para lo cual se hará uso de de flujogramas de procesos administrativos.

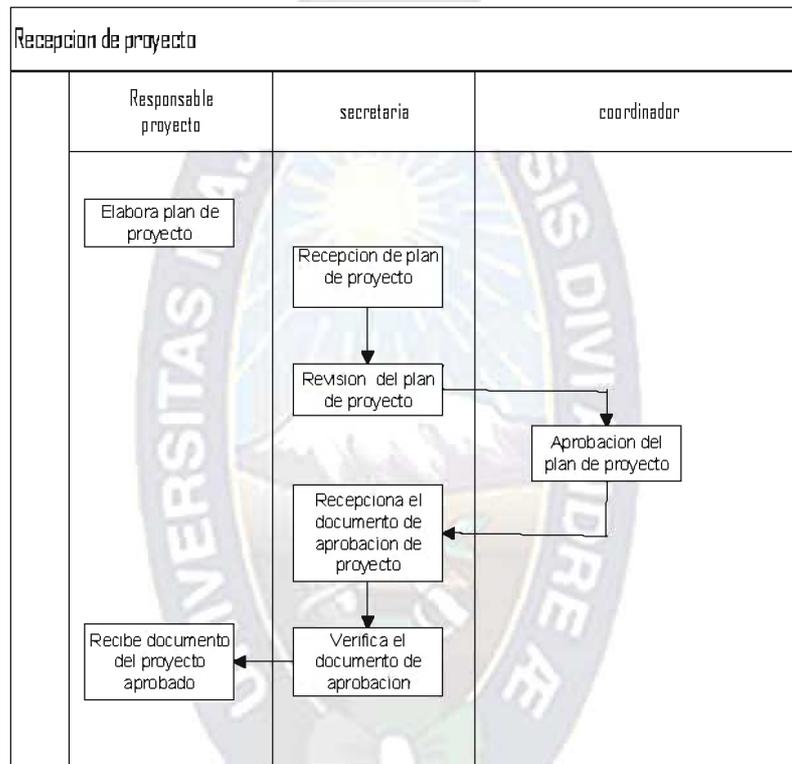


Figura: 3.3 Representación grafica de Proceso de registro de proyecto

Fuente: [Elaboración propia]

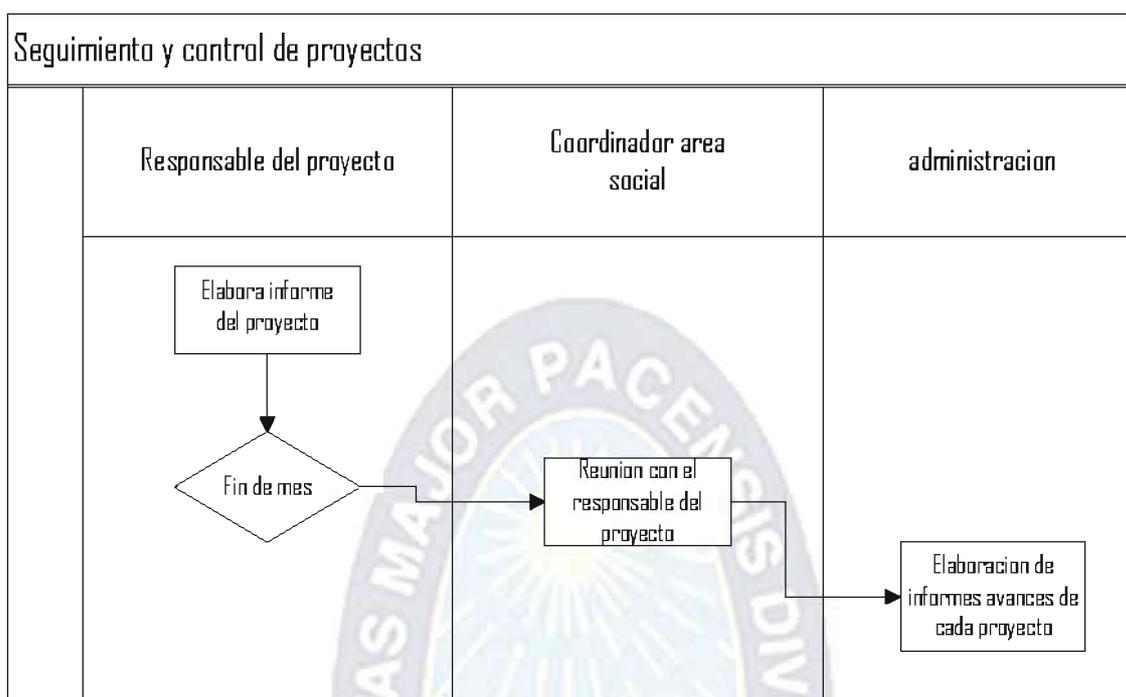


Figura 3.4: Seguimiento de proyectos

Fuente: [Elaboración Propia]

Para la formulación de proyectos sociales es necesaria la utilización de un formato de presentación de ellos, es por ello que para tal efecto se hace uso del marco lógico donde se presenta toda la información relevante como ser objetivo general, objetivos específicos, indicadores, metas, insumos, etc.

En este caso la fundación arco iris hace uso de este instrumento de formulación de un proyecto, si bien es cierto que solo hace referencia a una parte de ello.

Para una mejor comprensión, a continuación se detalla los elementos que intervienen en la formulación de un proyecto social.

El objetivo general es redactado o presentado al principio de la tabla que se describe abajo, es decir fuera de la tabla.

Objetivo específico	actividad	indicador	meta	medios	insumos
nivelar y potenciar el rendimiento de 160 niños y adolescentes, respetando su ritmo de aprendizaje	diagnostico inicial apoyo en la elaboración de áreas escolares, individual y grupal	numero de niños y adolescentes	160 niños  2 proyectos por aula 3 pruebas por año	fichas sociales  informes de actividad fichas de seguimiento	papel bond  libros de consulta

Tabla 3.1: Elementos de planificación de un proyecto

Fuente:[Fundación Arco Iris]

El monitoreo de los proyectos es realizado mensualmente tomando en cuenta otra tabla similar, donde se detallan los siguientes datos:

Objetivo general, objetivo específico, actividad, cronograma (semanas), indicadores, metas, insumos y presupuesto.

El monitoreo consiste en la verificación del cumplimiento del cronograma mensual.

La presentación de resultados es por áreas las cuales se detallan a continuación:

- ✓ coordinación
- ✓ psicológico
- ✓ trabajo social
- ✓ educadores

El marco lógico, además de sus funciones en la elaboración del proyecto, es en esencia un instrumento de evaluación.

### 3.4 DISEÑO DEL SISTEMA

En esta etapa se abarca la construcción del nuevo sistema, para lo cual se cuenta con el flujo de proceso del Diseño Orientado a Objetos (DOO)



Figura 3.5: Flujo del Proceso Orientada a Objetos

Fuente:[PRESSMAN, 2003]

Utilizando la notación de la metodología mencionada en el capítulo anterior, se diseñan los modelos y diagramas propios de la metodología orientada a objetos.

### 3.5 DETERMINACIÓN DE REQUISITOS

El propósito fundamental del flujo de trabajo de los requisitos es guiar el desarrollo hacia el sistema correcto.

#### Identificación de Actores

- ✓ Coordinador área social.
- ✓ Responsable del proyecto

- ✓ Asistente de coordinación social
- ✓ Beneficiarios (población de niños y adolescentes)

### **Rol de los Actores**

- ✓ Los responsables de cada proyecto planifican y elaboran actividades que serán ejecutadas al interior de estos.
- ✓ El coordinador aprueba la planificación de los proyectos que serán ejecutados y realiza el respectivo control sobre el avance de ellos.
- ✓ La asistente de coordinación es la responsable de la seguridad, integridad del sistema y de la base de datos.

### **3.6 REQUERIMIENTOS DE LOS USUARIOS**

Una vez realizada las entrevistas y observaciones de los posibles usuarios del sistema (Coordinador área social, responsable del sistema, responsable del proyecto), se identificaron los siguientes requerimientos:

#### **Coordinador Área Social**

- ✓ El coordinador requiere un sistema de interfaz amigable.
- ✓ La información de los proyectos, avances y reportes deben estar en el sistema.
- ✓ La información de todos los datos de los responsables de deben almacenar en el sistema.
- ✓ También es necesaria información estadística de los beneficiarios de los proyectos
- ✓ Por ultimo que el sistema sea fácil y sencillo en cuanto a su manejo.

#### **Asistente de coordinación**

- ✓ La asistente de coordinación es la responsable del sistema pues cuenta con un modulo de seguridad, es decir que el acceso sea controlado mediante password y un identificador de usuario

- ✓ La base de datos debe ser creada garantizando la integridad y el buen almacenamiento de datos.
- ✓ Por ultimo el sistema debe ser fácil de instalar.

#### **Responsable del Proyecto:**

- ✓ El responsable del proyecto, cuenta con la información general almacenada.

### **3.7 REQUERIMIENTOS DEL SISTEMA**

Para lograr un buen funcionamiento del sistema se requerirá de una PC con las siguientes características:

Hardware:

- ✓ Procesador Pentium II o superior
- ✓ Memoria RAM de 64 MB
- ✓ Disco duro de 20GB
- ✓ Unidad de disco flexible
- ✓ Monitor Super VGA, teclado, mouse
- ✓ Impresora

Software

- ✓ Windows 98 o superior
- ✓ Office 2000
- ✓ Visual Basic ver. 6.0
- ✓ Cristal reports ver. 8.0

### **3.8 ANALIS DE CASOS DE USO**

Una vez recopilados los requerimientos es necesario escenarios para el sistema que va a ser construido.

## Diagramas de casos de Estudio

Los diagramas de casos de uso muestran la funcionalidad del sistema desde la perspectiva que tienen los usuarios y lo que el sistema debe hacer para satisfacer los requisitos propuestos. A continuación se muestra el diagrama de casos de usos del sistema de control de proyectos.

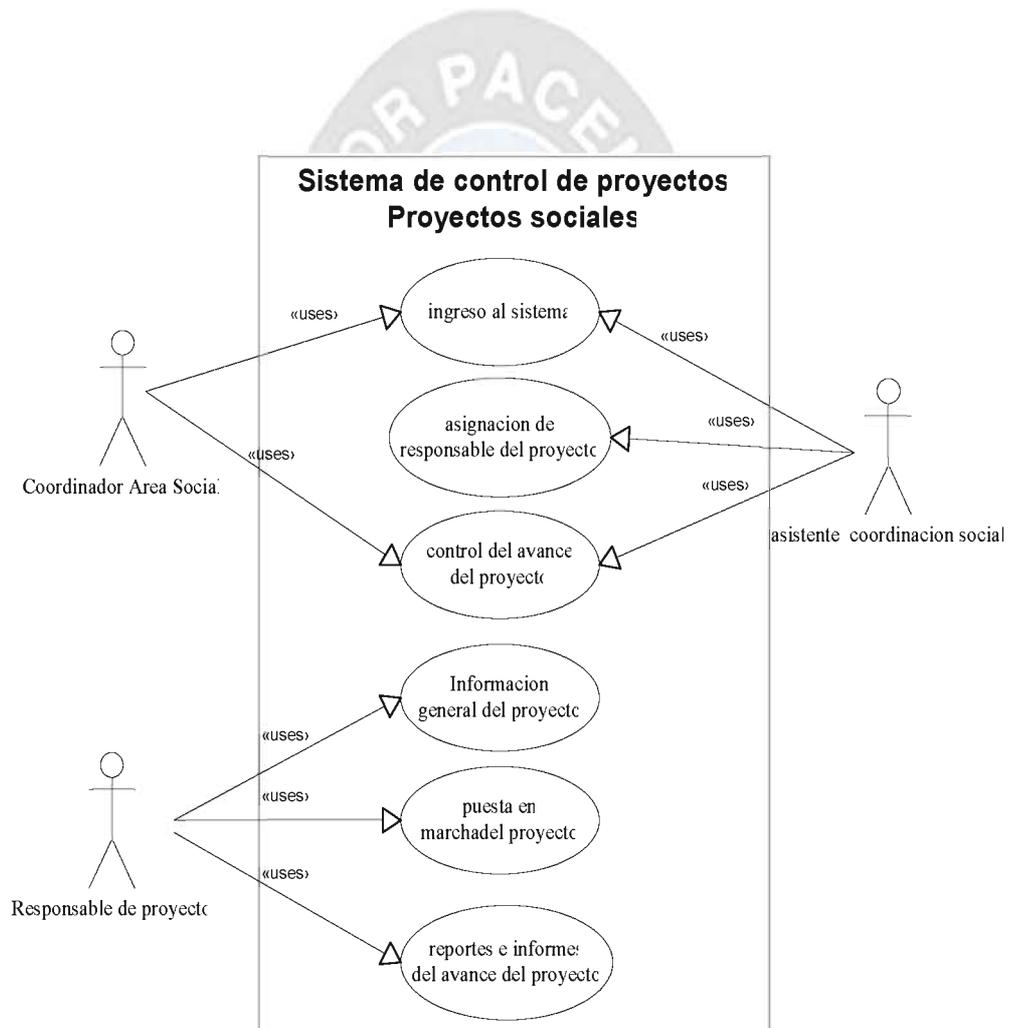


Figura 3.6: Diagrama de casos de uso

Fuente: [Elaboración propia]

### 3.9 DESCRIPCIÓN DE LOS CASOS DE USO

A continuación se detalla el documento de descripción de los casos de uso.

#### CASO DE USO 1

Nombre	Ingreso al Sistema
Descripción	Permite el ingreso al sistema
Actores	Usuarios (coordinador área social, responsable del sistema)
Precondiciones	Haber ingresado al sistema
Flujo Normal	1.- El usuario pulsa sobre el botón ingresar. 2.- El sistema solicita password. 3.- El usuario introduce su password. 4.- El sistema verifica la validez del software y se accede al sistema.
Flujo Alternativo	4.- El sistema verifica la validez del password, si el password no es correcto, se cancela la acción.
Poscondición	El responsable del sistema puede modificar permiso de ingreso al sistema.

Tabla 3.2: Caso de uso ingreso al sistema  
Fuente:[Elaboración Propia]

#### CASO DE USO 2

Nombre	Información General del Proyecto
Descripción	Permite ingresar datos generales del proyecto
Actores	Responsable del sistema
Precondiciones	Aprobación del Coordinador del área social
Flujo Normal	1.- El sistema solicita password del sistema. 2.- El responsable introduce password. 3.- Elige la opción datos del proyecto. 4.- Ingresa nombre y datos generales del proyecto.
Flujo Alternativo	4.- Ninguno
Poscondición	La información es almacenada previa revisión.

Tabla 3.3: Caso de uso información general del Proyecto  
Fuente:[Elaboración Propia]

## CASO DE USO 3

Nombre	Asignar Responsable de Proyecto
Descripción	Permite conocer al responsable del proyecto
Actores	Coordinador área social, responsable del sistema
Precondiciones	Experiencia del responsable en proyectos similares
Flujo Normal	1.- El responsable del sistema ingresa al sistema 2.- El responsable del sistema elige opción datos del proyecto 3.- introduce datos del responsable del proyecto. 4.- El sistema almacena datos del responsable del proyecto
Flujo Alternativo	4.- Ninguno
Poscondición	La información es almacenada previa revisión.

Tabla 3.4: Caso de uso asignación de responsable  
Fuente: [Elaboración Propia]

## CASO DE USO 4

Nombre	Control de Proyecto
Descripción	Permite controlar el avance del proyecto en ejecución
Actores	Responsable del proyecto, Coordinador área social
Precondiciones	Contar con información del Proyecto en el sistema
Flujo Normal	1.- El coordinador área social Ingresa al sistema 2.- El coordinador ingresa opción avance de proyecto 3.- Digita código o nombre del proyecto 4.- El sistema despliega la información del proyecto
Flujo Alternativo	4.- El coordinador elige opción reportes de proyecto, donde se se encuentra los reportes del avance del proyecto.
Poscondición	La información es almacenada previa revisión.

Tabla 3.5: Caso de uso control de proyecto  
Fuente: [Elaboración Propia]

## CASO DE USO 5

Nombre	Puesta en Marcha del Proyecto
Descripción	Permite dar inicio el proyecto.
Actores	Responsable del proyecto, Coordinador área social.
Precondiciones	Contar con información del Proyecto.
Flujo Normal	1.- Revisión del proyecto por el coordinador del área social 2.- Contar con aprobación del coordinador. 3.- El coordinador devuelve proyecto aprobado al responsable del proyecto. 4.- El responsable inicia el proyecto.
Flujo Alternativo	4.- Ninguno.
Poscondición	La información es almacenada previa revisión.

Tabla 3.6: Caso de uso puesta en marcha del proyecto  
Fuente: [Elaboración Propia]

## CASO DE USO 6

Nombre	Mostrar Reportes
Descripción	Permite mostrar reportes del avance de proyecto
Actores	Coordinador área social, encargado del sistema
Precondiciones	Contar con información del Proyecto en ejecución
Flujo Normal	1.- El coordinador área social elige opción reportes de proyecto. 2.- El sistema muestra en pantalla todos los proyectos registrados en el sistema. 3.- El coordinador elige el proyecto del cual se requiere el reporte. 4.- El sistema genera y muestra el reporte.
Flujo Alternativo	4.- El sistema consulta a la base de datos si el proyecto no cuenta con un file asociado se comunica de ello con el responsable del sistema.
Poscondición	El reporte se encuentra listo para su impresión.

Tabla 3.7: Caso de uso mostrar reporte  
Fuente: [Elaboración Propia]

## CASO DE USO 7

Nombre	Acceso a los Beneficios
Descripción	Permite mostrar información de los beneficiarios de los subprogramas que existe en la Fundación Arco Iris.
Actores	Encargado del sistema, coordinador área social
Precondiciones	Los beneficiarios deben pertenecer a un proyecto.
Flujo Normal	<ol style="list-style-type: none"> <li>1.- El coordinador del área social ingresa al sistema</li> <li>2.- El coordinador del área social elige la opción subprogramas.</li> <li>3.- El sistema muestra opción beneficiarios.</li> <li>4.- El coordinador elige el proyecto (código, nombre) .</li> <li>5- El sistema muestra información del proyecto y los subprogramas requeridos.</li> </ol>
Flujo Alternativo	4.- Ninguno.
Poscondición	La información es almacenada en el sistema.

Tabla 3.8: Caso de uso Acceso a beneficios  
Fuente: [Elaboración Propia]

Los actores son aquellos que interactúan con el sistema. Las precondiciones son los hechos que se han de cumplir para que el flujo del evento se pueda llevar a cabo.

El flujo de eventos que corresponde a la ejecución normal y exitosa de los casos de uso. Los flujos alternativos son los que permiten indicar que es lo que hace el sistema en los casos inesperados.

Por ultimo, las poscondiciones son los hechos que se ha de cumplir si el flujo de eventos normal se ha ejecutado correctamente.

### 3.10 DIAGRAMA DE SECUENCIA

Un diagrama de secuencia muestra las interacciones entre objetos ordenados en secuencia temporal.

A continuación se muestran los diagramas de secuencia de los casos de uso más relevantes.

## DIAGRAMAS DE SECUENCIA INGRESO AL SISTEMA

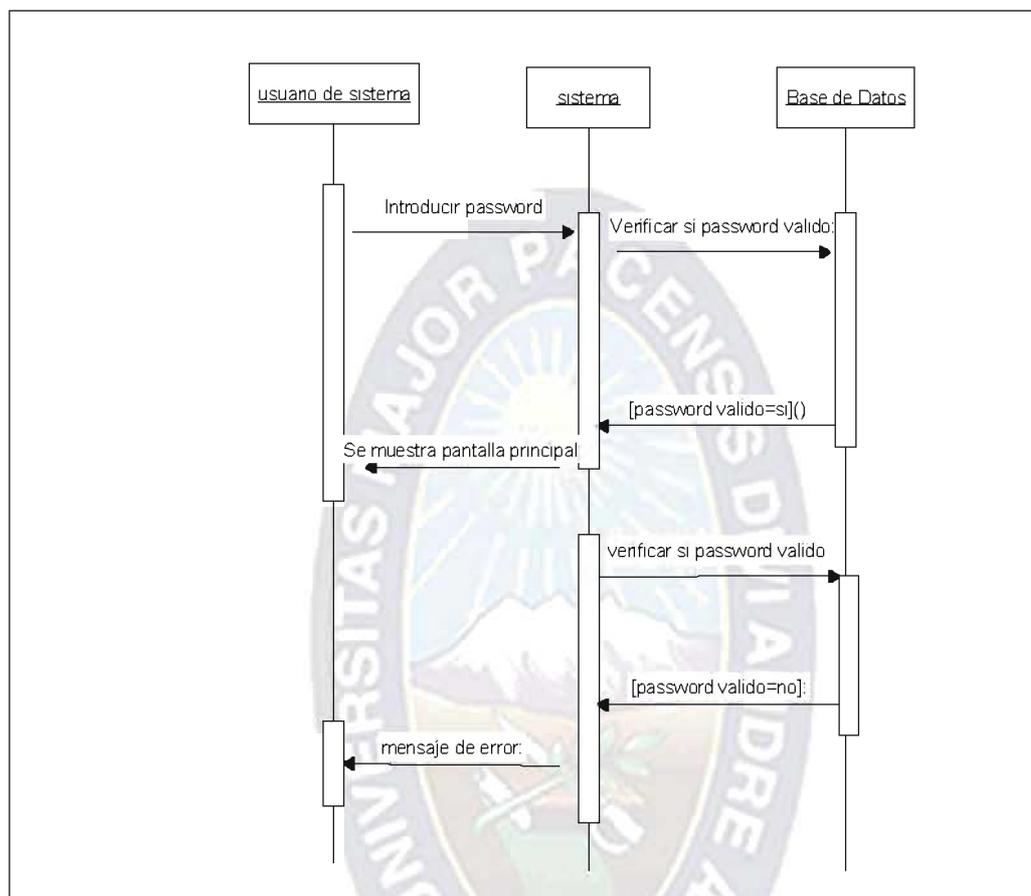


Figura 3.7: Diagrama de secuencia ingreso al sistema

Fuente:[Elaboración Propia]

Descripción del diagrama de secuencia:

- 1.- El usuario del sistema accede al sistema mediante un password.
- 2.-El usuario introduce password.
- 3.-El sistema verifica la validez del password.
- 4.- Si la verificación del password indica
  - ✓ Password valido, se muestra la pantalla principal.
  - ✓ Password no valido, se cancela la operación

## DIAGRAMA DE SECUENCIA INFORMACIÓN DEL PROYECTO

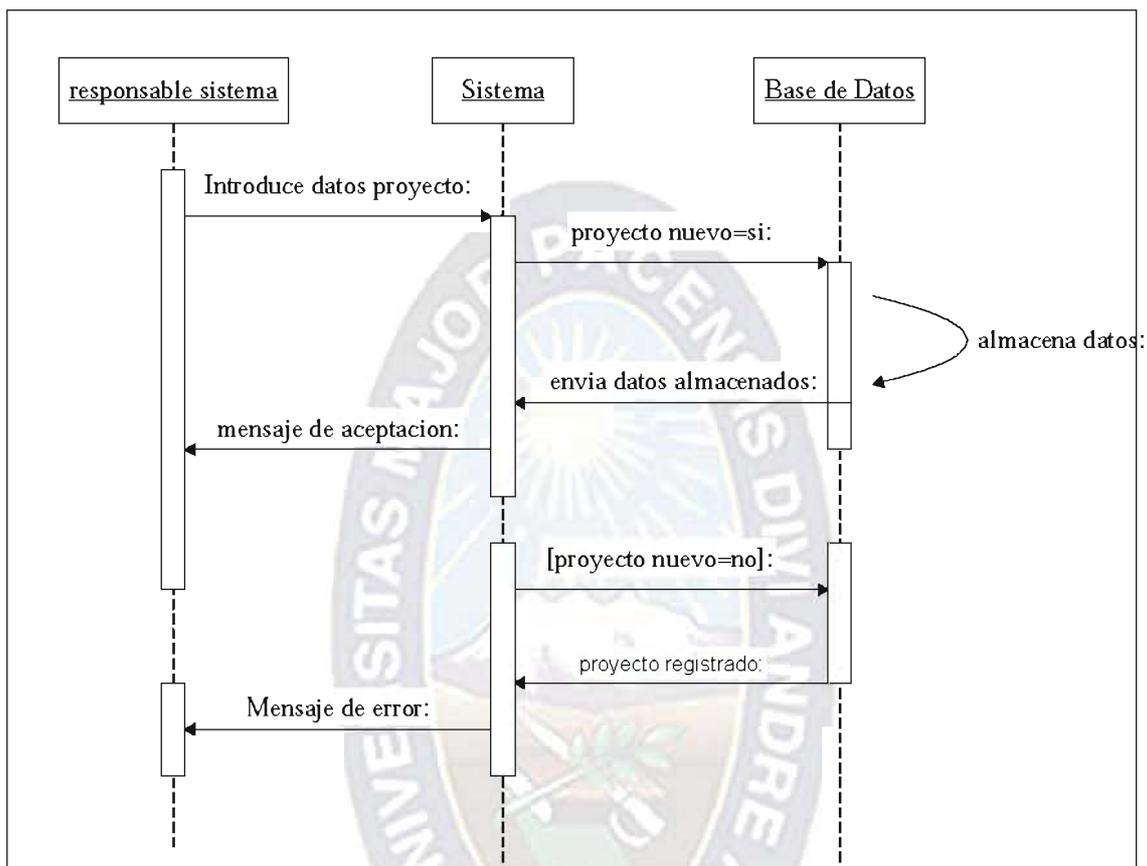


Figura 3.8: Diagrama de secuencia información del Proyecto

Fuente:[Elaboración Propia]

Descripción del diagrama de secuencia:

- 1.- El responsable del sistema accede a la opción de proyecto.
- 2.- El sistema despliega pantalla con opción datos de proyecto
- 3.- Si el proyecto es nuevo:
  - ✓ El responsable del proyecto introduce información del proyecto a ejecutarse.
  - ✓ Caso contrario proyecto ya registrado
- 4.- Entonces el sistema muestra mensaje de error.

## DIAGRAMA DE SECUENCIA CONTROL DE PROYECTO

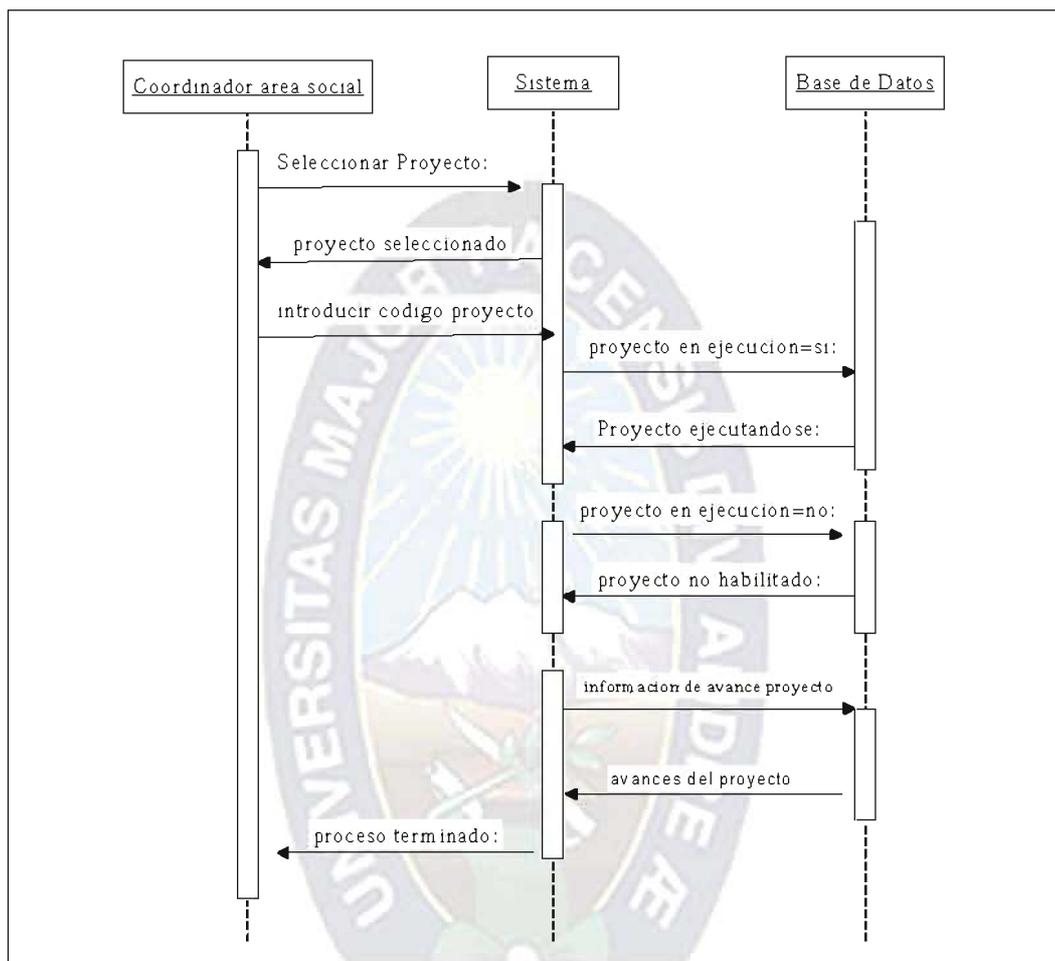


Figura 3.9: Diagrama de secuencia Avance del proyecto

Fuente: [Elaboración Propia]

Descripción de diagrama de secuencia:

- 1.- El coordinador ingresa al sistema y elige opción proyectos.
- 2.- El sistema despliega pantalla con información de los proyectos.
- 3.-El coordinador introduce código del proyecto
- 4.-El sistema verifica si el proyecto solicitado se encuentra en ejecución
- 5.- Si se encuentra en ejecución despliega información del proyecto requerido.
- 6.- Finalmente termina el proceso.

### 3.11 DIAGRAMA DE ACTIVIDADES

El diagrama de Actividad es un diagrama de proceso multi- propósito que se usa para modelar el comportamiento del sistema.

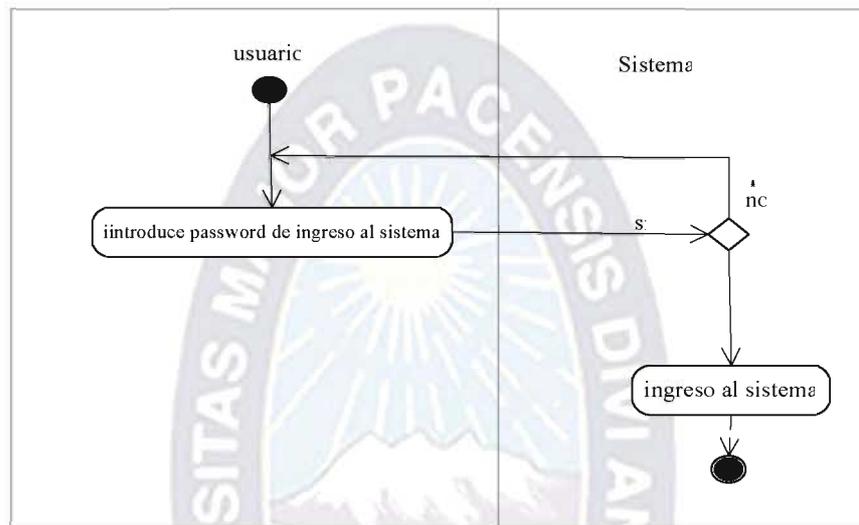


Figura 3.10: Diagrama de actividad Ingreso al sistema

Fuente: [Elaboración Propia]

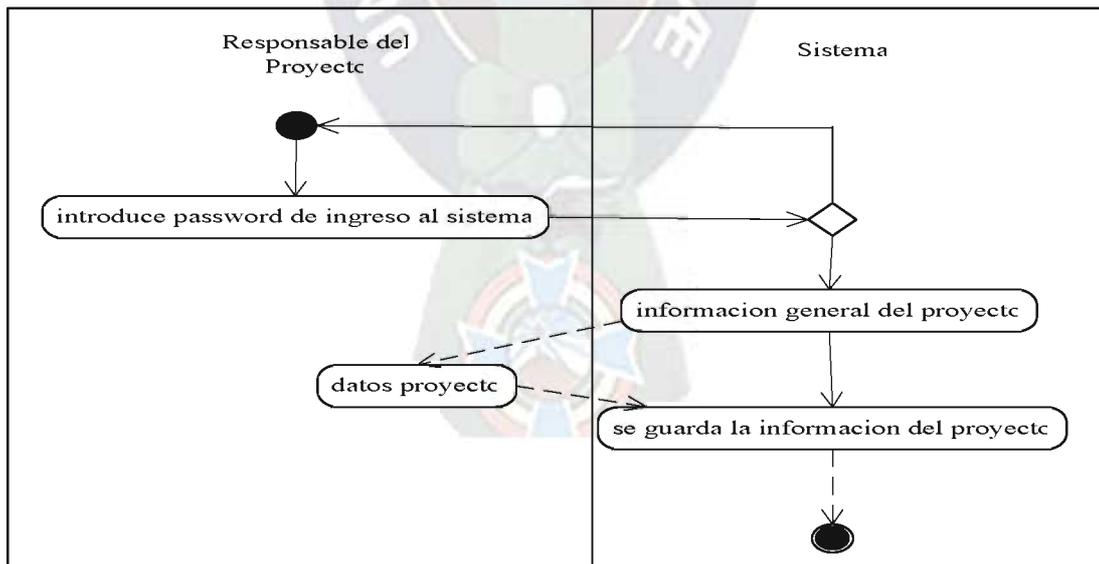


Figura 3.11: Diagrama de Actividad ingreso de datos de proyecto

Fuente: [Elaboración Propia]

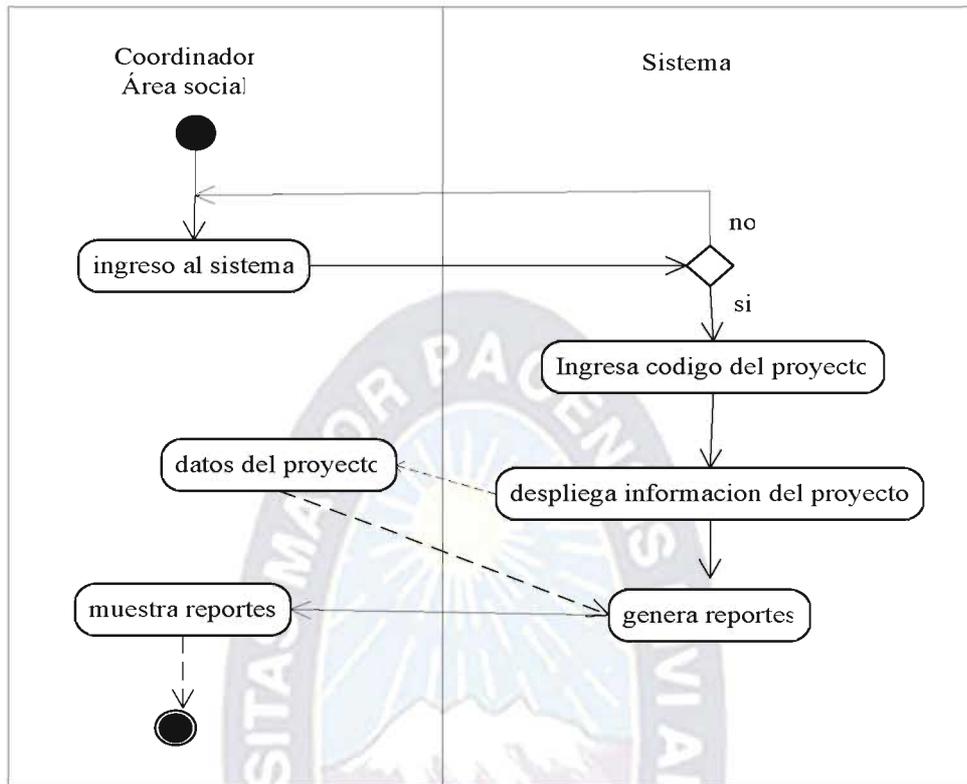


Figura 3.12: Diagrama de Actividad Control de Proyecto

Fuente:[Elaboración Propia]

3.12 MODELO CONCEPTUAL

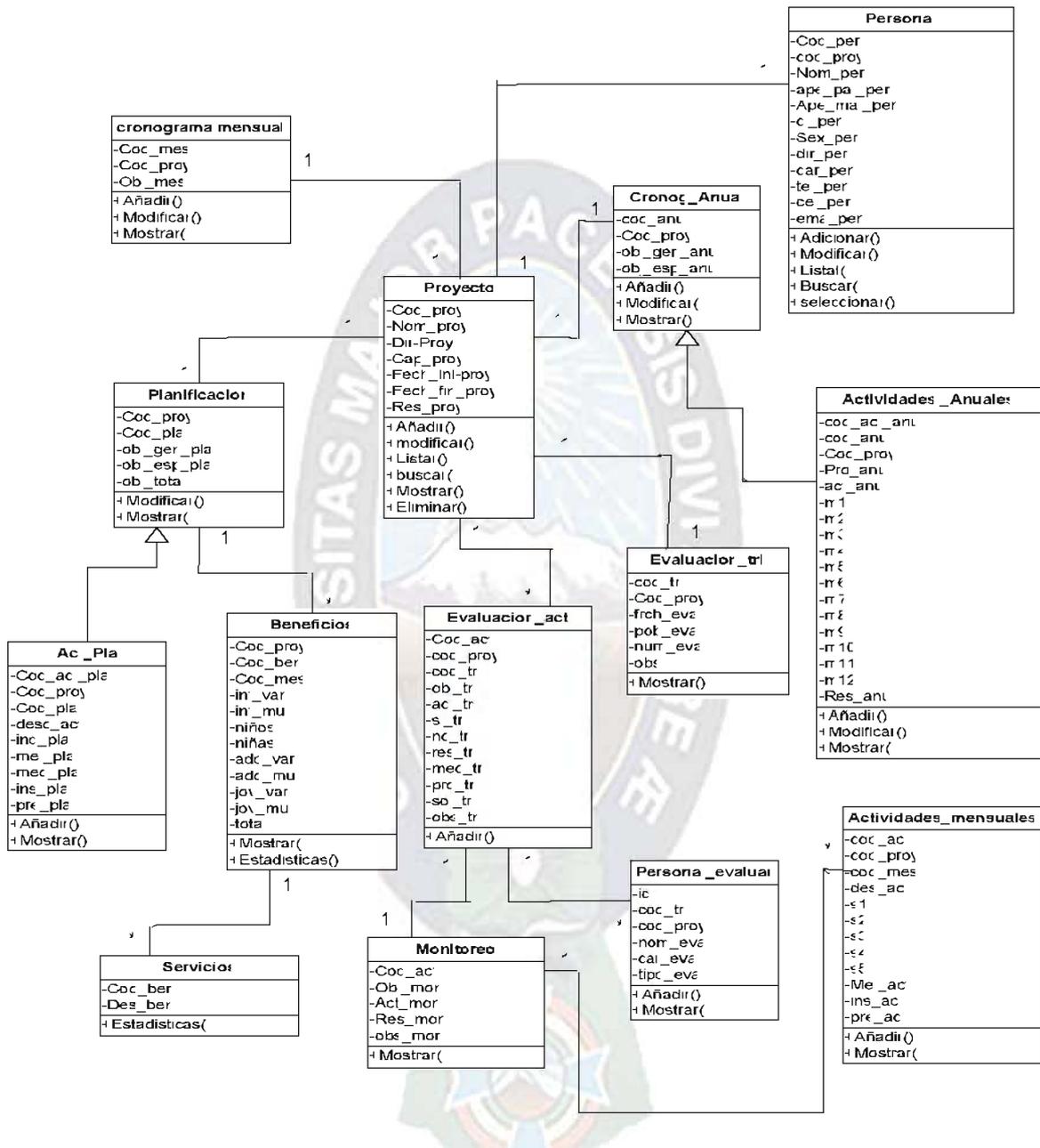


Figura 3.13: Diagrama de clases  
Fuente: [Elaboración Propia]

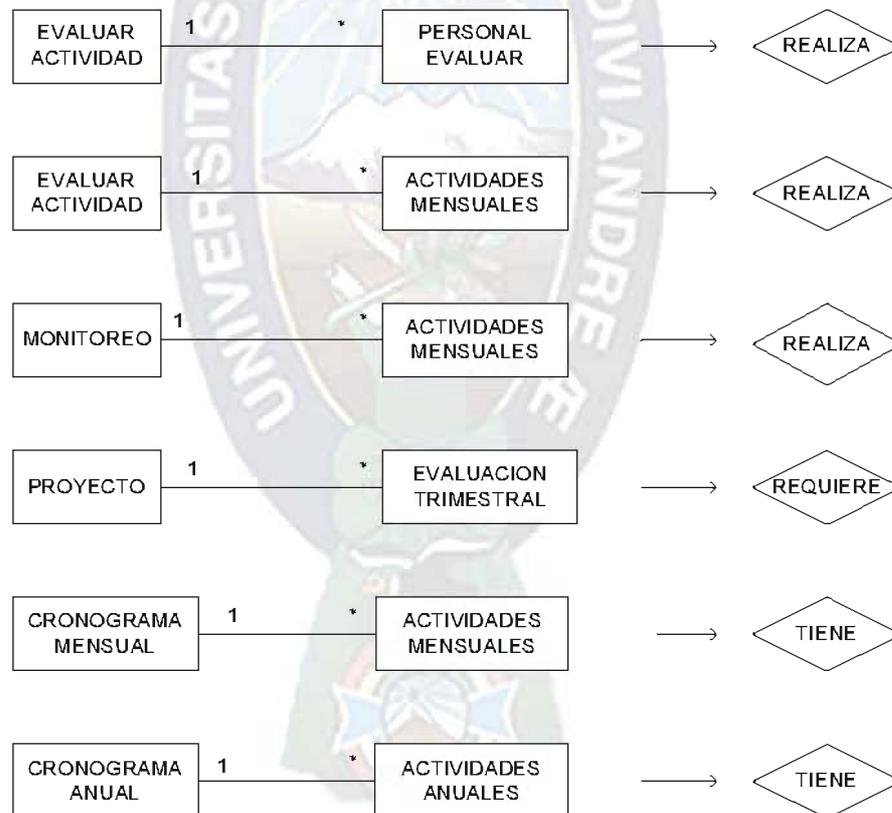
### 3.13 MAPEO AL MODELO ENTIDAD RELACION (E/R)

Posteriormente se procede al mapeo del diseño de la base de datos a una base de datos relacional.

¿Cómo representar las relaciones de objetos en una tabla de una base de datos relacional?. La respuesta se da en el patrón **Representación de las relaciones de objetos como Tablas** [Brown96] que propone lo siguiente:

- **Asociaciones de uno a muchos**

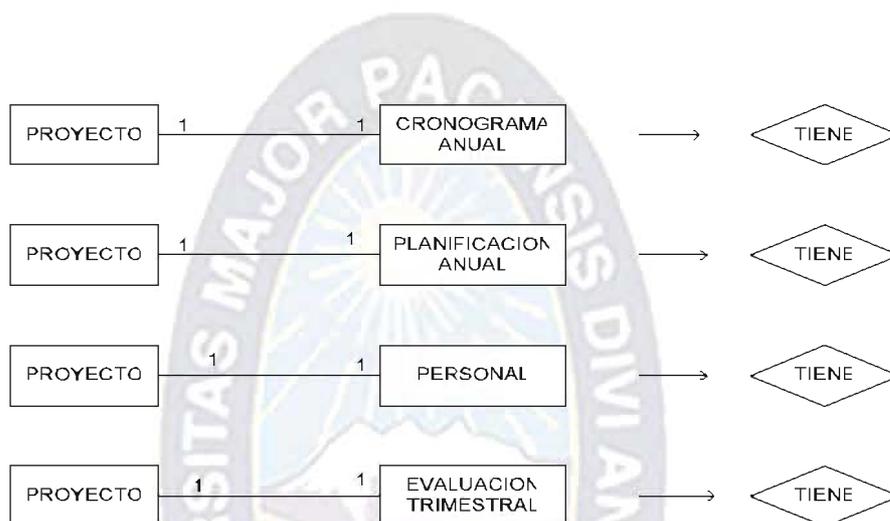
Crear una tabla asociativa que registre los identificadores de cada objeto en la relación.



- **Asociaciones uno a uno**

Colocar una clave foránea del identificador de objetos en una o en las dos tablas que representan los objetos en la relación.

Crear una tabla asociativa que registre los identificadores de objetos de cada objeto en la relación.



- **Asociaciones de muchos a muchos**

Crear una tabla asociativa que registre todos los identificadores de objetos en la relación.

[LARMAN, 1999]

Este tipo de asociaciones no se encuentran en el diagrama de clases, por lo que no existe ningún mapeo de este tipo.

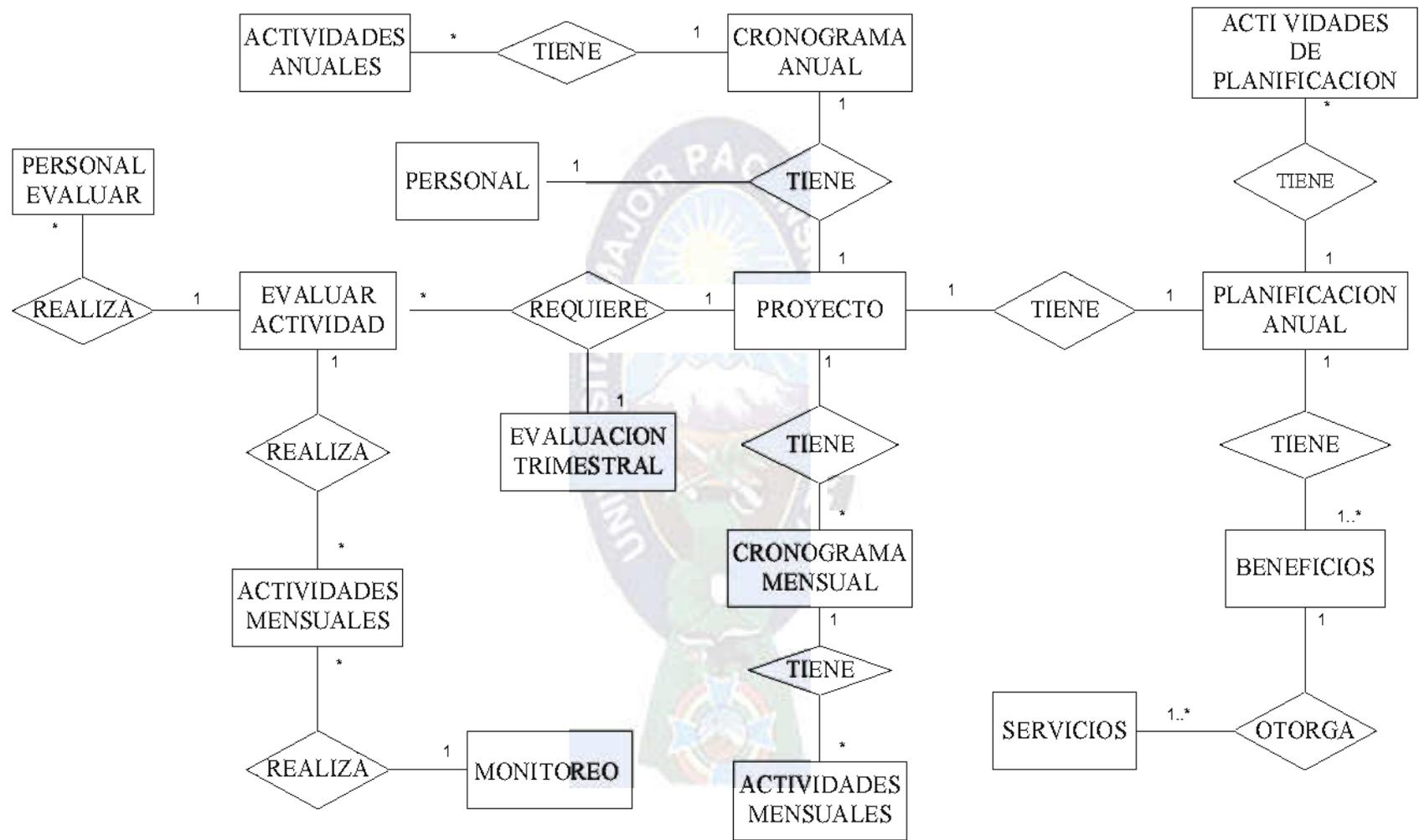


Fig. 3.14 : Modelo E-R  
Fuente : [Elaboración Propia]



### 3.15 ARQUITECTURA DEL SISTEMA

**Arquitectura Cliente-Servidor de Dos Capas.-** Consiste en una capa de presentación y lógica de la aplicación; y la otra de la base de datos.

El sistema trabaja sobre una arquitectura cliente servidor de dos capas; en el servidor esta soportado por un sistema administrativo, un motor de base de datos SQL Server 2000, y el cliente es soportado por la interfaz del sistema desarrollado en el lenguaje de programación visual Basic 6.0

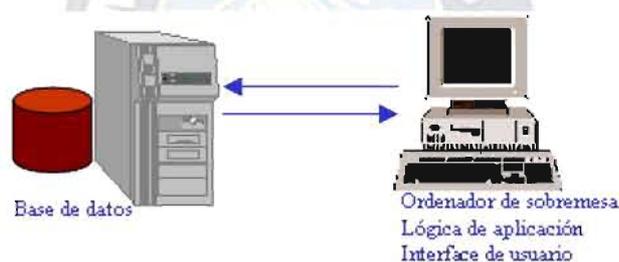


Figura 3.16 Arquitectura del Sistema

Fuente:[Elaboración Propia]

## DISEÑO MODULAR “SICOFAI”

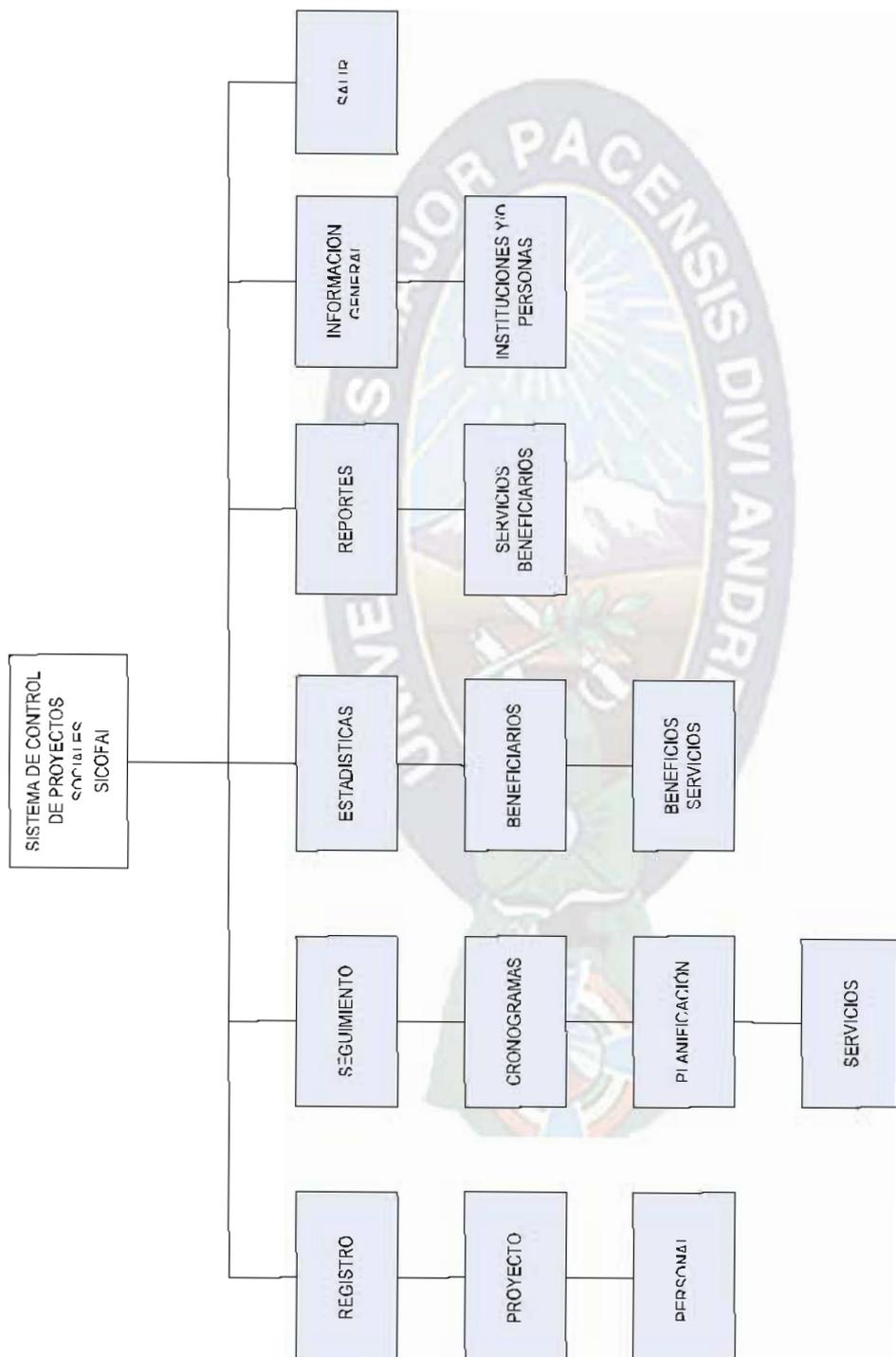


Fig. 3 17 Estructura modular del sistema Sicofai  
Fuente : [Laboración propia]

El almacenamiento de datos es considerado como parte de los sistemas de información.

Los objetivos generales para el diseño de la organización de almacenamiento de datos son los siguientes:

- ✓ los datos tienen que estar disponibles cuando el usuario quiere usarlos.
- ✓ Los datos deben ser precisos y consistentes.

### 3.16 TABLAS DE LA BASE DE DATOS

A continuación se describen algunas de las tablas sus campos y el tipo de datos que poseen

PROYECTO

	Nombre de columna	Tipo de datos
🔑	Cod_proy	varchar
	Nom_proy	varchar
	Dir_proy	varchar
	Cap_proy	numeric
	Fech_ini_proy	datetime
	Fech_fin_proy	datetime
▶	Res_proy	varchar

CRONOGRAMA ANUAL

	Nombre de columna	Tipo de datos
🔑	Cod_anu	varchar
🔑	Cod_proy	varchar
	obj_gen_anu	varchar
	obj_esp_anu	varchar

## ACTIVIDADES PLANIFICACION

	Nombre de columna	Tipo de datos
🔑	Cod_act_pla	numeric
🔑	Cod_proy	varchar
🔑	Cod_pla	varchar
	desc_act	varchar
	ind_pla	varchar
	met_pla	varchar
	med_pla	varchar
	ins_pla	varchar
	pre_pla	numeric

## BENEFICIARIOS

	Nombre de columna	Tipo de datos
🔑	Cod_ben	numeric
🔑	Cod_mes	varchar
	Inf_var	numeric
	Inf_muj	numeric
	niños	numeric
	niñas	numeric
	Ado_var	numeric
	Ado_muj	numeric
	Jov_var	numeric
	Jov_muj	numeric
	Tot	numeric

## SERVICIOS

	Nombre de columna	Tipo de datos
🔑	Cod_ben	numeric
	Des_ben	varchar

## PERSONAL

	Nombre de columna	Tipo de datos
?	Cod_per	varchar
	Cod_proy	varchar
	Nom_per	varchar
	Ape_pat_per	varchar
	Ape_mat_per	varchar
	Ci_per	varchar
	Sex_per	varchar
	dir_per	varchar
	Car_per	varchar
	Tel_per	varchar
	cel_per	varchar
	ema_per	varchar

## ACTIVIDADES MENSUALES

	Nombre de columna	Tipo de datos
?	cod_act	numeric
	cod_proy	varchar
	cod_mes	varchar
	des_act	varchar
	s1	varchar
	s2	varchar
	s3	varchar
	s4	varchar
	s5	varchar
	Ind_act	varchar
	Met_act	varchar
	Ins_act	varchar
	Pre_act	varchar

## PLANIFICACION

	Nombre de columna	Tipo de datos
?	Cod_proy	varchar
?	Cod_pla	varchar
	obj_gen_pla	varchar
	obj_esp_pla	varchar
	OBJ_TOL	numeric
▶		

### 3.17 DISEÑO DE LA INTERFAZ

De acuerdo a las características y necesidades del usuario, la interfaz del sistema presenta un ambiente completamente grafico, de tal forma que al usuario le sea fácil el ingreso y la manipulación de los datos. Fig. 3.18

Muestra la pantalla de presentación del sistema:



Fig. 3.18. Pantalla de presentación del sistema SICOFAI

## INTERFAZ DE PREGUNTAS Y RESPUESTAS

Este tipo de interfaz se emplea para lograr la interacción del sistema con el usuario, por medio de un cuadro de dialogo el usuario debe responder a la pregunta que el sistema le plantea, en base a la respuesta el sistema ejecuta las ordenes



Fig. : 3.19 Cuadro de dialogo de pregunta y respuesta

## INTERFAZ DE USUARIOS

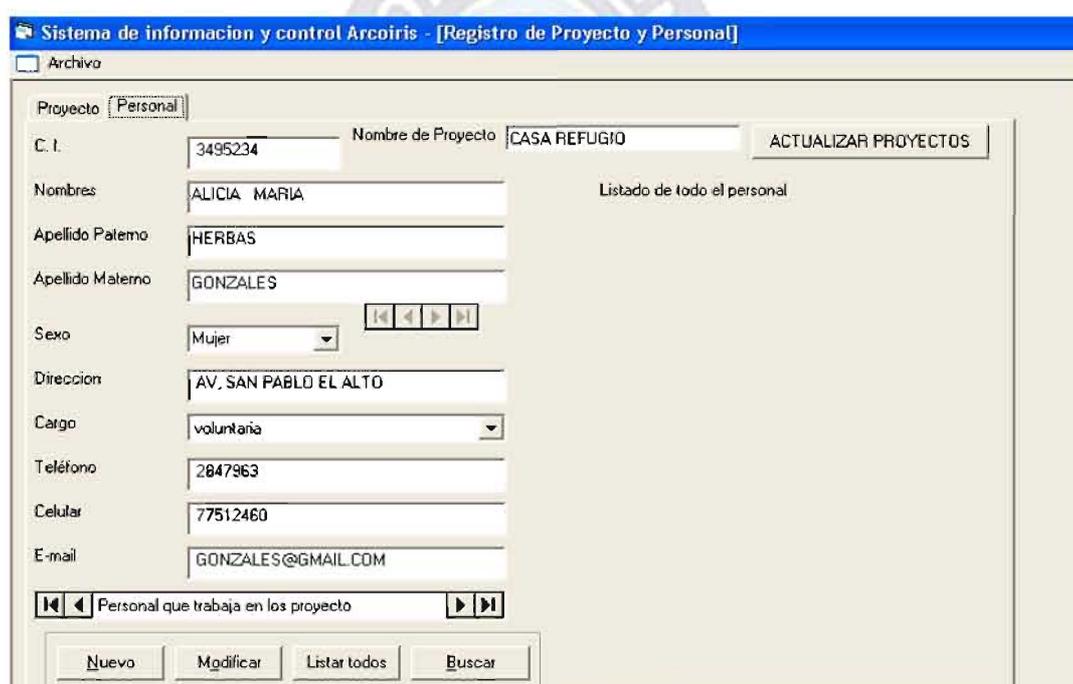
Este tipo de interfaz muestra una representación grafica, permite una interacción directa con el usuario, es decir permite la manipulación de información en forma directa utilizando botones gráficos que representan operaciones como ser: aceptar cancelar, etc. las Fig. 3.20 y 3.21 muestran ejemplos de este tipo de interfaz



## INTERFAZ DE ENTRADA Y SALIDA

Esta Forma de interfaz consiste en el llenado de formas, en pantallas las cuales despliegan campos que contienen datos o parámetros que necesitan ser comunicadas por el usuario, también permite la captura de datos que son introducidos al sistema.

A continuación se muestra un ejemplo de interfaz de esta forma en la fig. 3.22



The screenshot shows a web application window titled "Sistema de informacion y control Arcoiris - [Registro de Proyecto y Personal]". The interface includes a menu bar with "Archivo" and a main content area with the following fields and controls:

- Project Section:** "Proyecto" dropdown menu set to "Personal", "Nombre de Proyecto" text box containing "CASA REFUGIO", and an "ACTUALIZAR PROYECTOS" button.
- Personal Information Section:** "C. I." text box (3495234), "Nombres" text box (ALICIA MARIA), "Apellido Paterno" text box (HERBAS), "Apellido Materno" text box (GONZALES), "Sexo" dropdown menu (Mujer), "Direccion" text box (AV. SAN PABLO EL ALTO), "Cargo" dropdown menu (voluntaria), "Teléfono" text box (2847963), "Celular" text box (77512460), and "E-mail" text box (GONZALES@GMAIL.COM).
- Navigation and Action Section:** A set of navigation arrows, a text box containing "Personal que trabaja en los proyecto", and four buttons: "Nuevo", "Modificar", "Listar todos", and "Buscar".

Fig.3.22: Interfaz de llenado de formas de Entrada y Salida

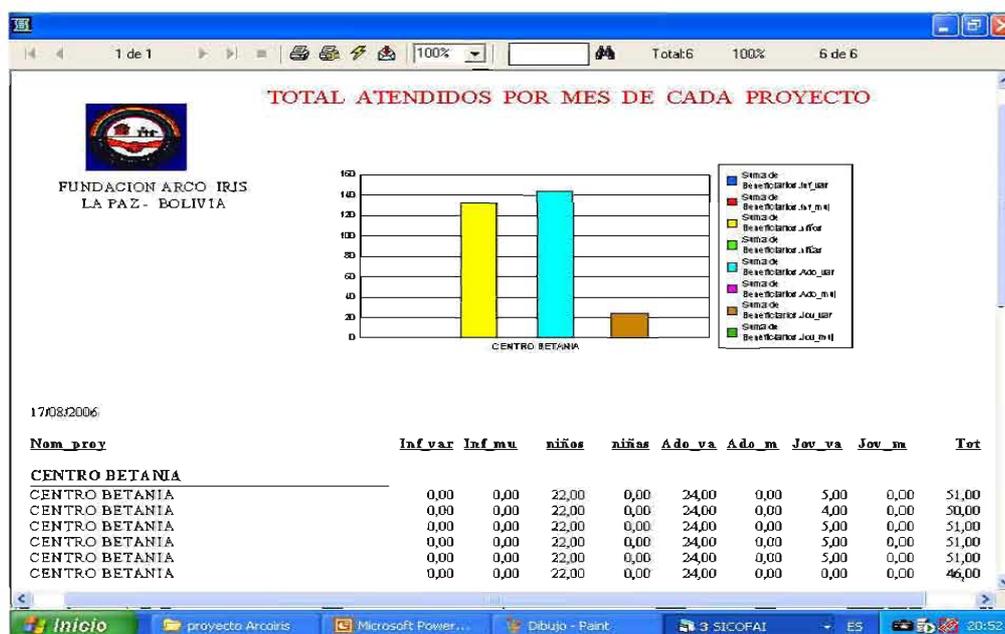


Fig.: 3. 23 Pantalla de reporte

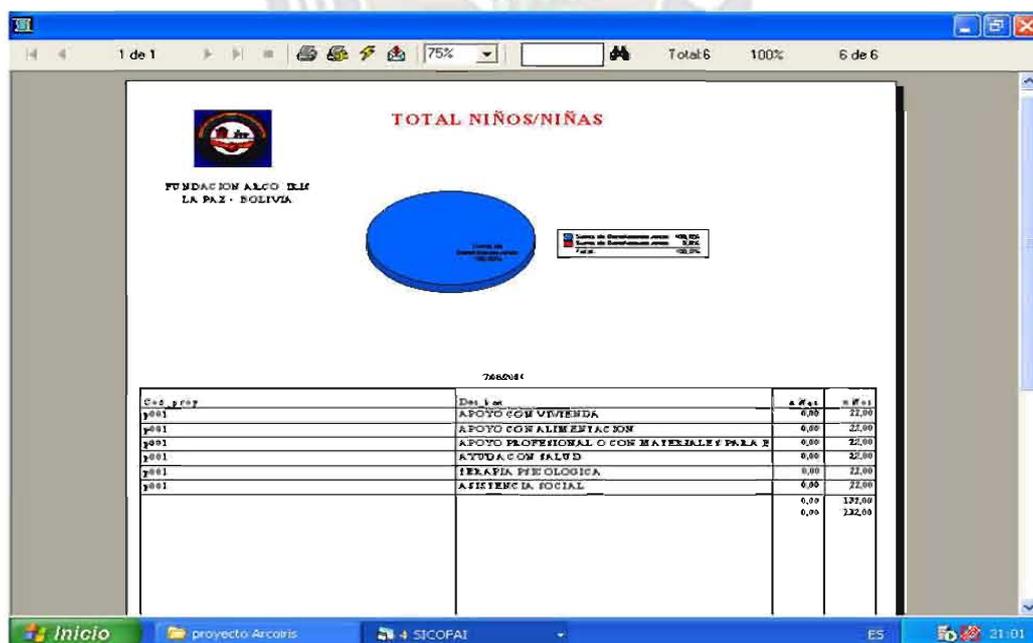


Fig. : 3.24 Pantalla de reporte

### 3.19 PRUEBAS DEL SISTEMA

La prueba del software es un elemento crítico es un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación.[PRESSMAN, 1998]

### 3.20 PRUEBA DE CAJA BLANCA

La prueba de caja blanca, denominada a veces prueba de caja de cristal es un método de diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba caja blanca, el ingeniero del software puede obtener casos de prueba que garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo, ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa; ejecuten todos los bucles en sus límites y con sus límites operacionales. .[PRESSMAN, 1998]

El método del camino básico permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía PARA la definición de un conjunto básico de caminos de ejecución.

Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa. .[PRESSMAN, 1998]

A continuación se usará el procedimiento búsqueda para demostrar los pasos del procedimiento.

```
Private Sub nomproy_Change()
If nomproy.Text <> Empty Then
adonompro.Recordset.MoveFirst
CODIGOPROYECTO = ""
While Not adonompro.Recordset.EOF
If adonompro.Recordset.Fields("Nom_proy") = nomproy.Text Then
CODIGOPROYECTO = adonompro.Recordset.Fields("cod_proy")
NOMBREPROYECTO = nomproy.Text
Else
End If
adonompro.Recordset.MoveNext
Wend
If CODIGOPROYECTO = "" Then
```

```

MsgBox "No existe ese proyecto"
Else
End If
End If
End Sub

```

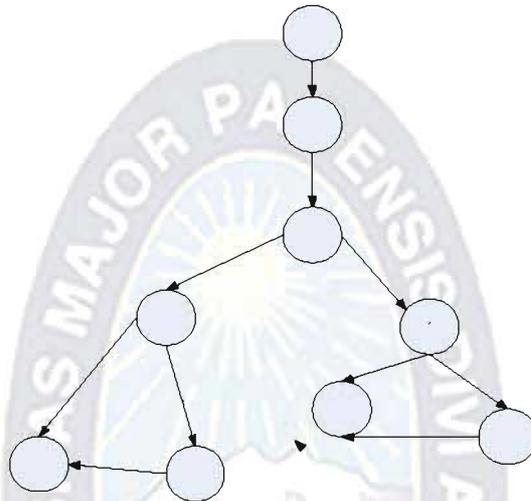


Figura 3.25 Grafo de Flujo del procedimiento Búsqueda

### Camino Básicos

Camino 1 : 1-2-3-4-5-6

Camino 2 : 1-2-3-6

Camino 3 : 1-2-3-7-9

Camino 4 : 1-2-3-7-8-9

La complejidad ciclomatica,  $V(G)$ , de un grafo de flujo  $G$  se define como:

$$V(G) = A - N + 2$$

Donde:  $A$ =numero de aristas

$N$ = numero de nodos

Luego entonces la complejidad ciclomatica de la figura 3.25 esta dada por:

$$V(G) = A - N + 2$$

$$V(G) = 11 - 8 + 2 = 11$$

## CAPITULO IV

### CALIDAD DE SOFTWARE

En primer lugar se brindara una definición de calidad como una característica o atributo de algo según el American Heritage Dictionary. Como un atributo de un artículo, **la calidad** se refiere a las características mensurables: cosas que se pueden comparar con estándares conocidos como longitud, color, propiedades eléctricas, etc.

Respecto a la calidad de software se considera otros aspectos como ser las características de un programa.

Entre estas propiedades se incluyen complejidad ciclomática, cohesión, numero de puntos de función, líneas de código, etc.

#### 4.1 MEDICIONES DEL SOFTWARE

La utilización de métricas para el proyecto tiene dos aspectos fundamentales. En primer lugar, estas métricas se utilizan para minimizar la planificación de desarrollo haciendo los ajustes necesarios que eviten retrasos y reduzcan problemas y riesgos potenciales.

En segundo lugar, las métricas para el proyecto se utilizan para evaluar la calidad de los productos en el momento actual y cuando sea necesario, modificando el enfoque técnico que mejore la calidad.

Otro modelo de métricas del proyecto de software [HET93] sugiere que todos los proyectos deberían medir:

- ✓ Entradas: la dimensión de los recursos que requieran para realizar el trabajo.
- ✓ Salidas: medidas de las entregas o productos creados durante el proceso de ingeniería de software.

- ✓ Resultados: medidas que indican la efectividad de las entregas.

La calidad de software es una compleja mezcla de factores que varían a través de diferentes factores y según los clientes que los pidan.

Los factores de calidad se pueden categorizar de la siguiente forma:

- a) Factores que se pueden medir directamente (defectos por punto función)
- b) Factores que solo se pueden medir solo indirectamente (facilidad de uso o de mantenimiento)

## 4.2 FACTORES DE CALIDAD DE McCALL

McCall y sus colegas [MAA 77] propusieron una categorización de factores que afectan a la calidad de software.

### FACTORES DE CALIDAD ISO 9126

El estándar ISO 9126 ha sido desarrollado en un intento de identificar los atributos clave de calidad para el software. El estándar identifica seis atributos clave de calidad:

- **Funcionalidad.** Grado en el cual el software satisface las necesidades.
- **Confiabilidad.** Cantidad de tiempo que el software está disponible para su uso.
- **Usabilidad.** Grado en el que el software es fácil de usar.
- **Eficiencia.** Grado en el que el software hace óptimo el uso de los recursos del sistema.
- **Facilidad de Mantenimiento.** La facilidad en el que una modificación puede ser realizada.
- **Portabilidad.** La facilidad con el que el software puede ser llevado de un entorno a otro.

#### 4.2.1 FUNCIONALIDAD

No es posible medir la funcionalidad del sistema de forma directa, es por ello que se la realizará mediante medidas directas utilizando el Punto Función. Esta relación se deriva con una relación empírica del dominio de información del software y las evaluaciones de complejidad.

Para determinar la funcionalidad del sistema de Información de Control de Proyectos fundación Arco Iris "SICOFAI", es necesario determinar las siguientes características:

- Número de entradas de usuario
- Número de salidas de usuario
- Número de peticiones de usuario
- Número de archivos
- Número de interfaces internas

A continuación se tiene la tabla para el cálculo de Puntos de Función

FACTORES DE PONDERACION			
Numero de entradas de usuario	12	3	36
Numero de salidas de usuario	8	4	32
Numero de peticiones de usuario	6	3	18
Numero de archivos	28	7	196
Numero de interfaces externas	1	5	5
total			287

Tabla 4.1 Cálculo de Puntos de función

Fuente: [PRESSMAN, 1998]

Por lo tanto el valor de la sumatoria del anterior cuadro es 287, que es un dato parcial.

El análisis que se hizo se trabajo con el parámetro de medición simple

Según Roger Pressman en su libro Ingeniería de software utiliza la siguiente relación para calcular la funcionalidad de un sistema:

$$PF = \text{Cuenta\_total} * [0.65 + 0.01 * \sum Fi]$$

Donde:

PF = Medida de Funcionalidad

$\sum Fi$  = Ajuste de la complejidad según el dominio de información donde (i= 1 a 14), son los valores de ajuste a la complejidad basados en las respuestas señaladas en la tabla 4.2 tomando la escala de valores de 0 a 5.

Factor de Ajuste	Descripción
0	Sin Influencia
1	Incidental
2	Moderado
3	Medio
4	Significativo
5	Esencial

Tabla 4.2 Valores de Ajuste

Fuente: [PRESSMAN, 1998]

Estos valores permiten ajustar la cuenta \_ total a una curva normal, donde es posible tomar los siguientes valores:

**Valor Máximo = 1.35**

**Valor Mínimo = 0.65**

Los valores de complejidad se obtienen de la tabla 4.3 donde se responde a las siguientes preguntas de acuerdo a la tabla de valores

<b>Fi</b>	<b>Factores</b>	<b>sub total</b>
1	Requiere el sistema copias de seguridad y de recuperación fiables?	3
2	Se requiere comunicación de datos?	4
3	Existen funciones de procesamiento distribuido?	0
4	Es crítico el rendimiento?	3
5	Se ejecutara el sistema en un entorno operativo existente y fuertemente utilizado?	4
6	Requiere datos de entrada interactiva?	3
7	Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas o variadas operaciones?	3
8	Se actualizan los archivos maestros de forma interactiva?	1
9	son complejos las entradas, salidas, los archivos o las peticiones?	3
10	Es complejo el procesamiento interno?	2
11	Se ha diseñado el código para ser reutilizable?	2
12	Esta incluida la conservación y la instalación?	4
13	Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	4
14	Se ha diseñado la planificación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	3
	<b>total</b>	<b>39</b>

Tabla 4.3: Calculo de Ajuste a la Complejidad

Fuente: [PRESSMAN, 1998]

Por lo tanto:

$$\Sigma F_i = F_1 + F_2 + F_3 + F_4 + F_5 + F_6 + F_7 + F_8 + F_9 + F_{10} + F_{11} + F_{12} + F_{13} + F_{14}$$

$$\Sigma F_i = 3 + 4 + 0 + 3 + 4 + 3 + 3 + 1 + 3 + 2 + 2 + 4 + 4 + 3 = 39$$

$$PF = \text{cuenta} - \text{total} * (0.65 + 0.01 * \Sigma f_i)$$

$$PF = 287 * [0.65 + 0.01 * 39]$$

$$PF = 298.48 \quad (\text{funcionalidad del sistema})$$

Donde este valor debe ajustarse a una curva normal

$$PF = (\text{real}) = \text{Cuenta total} * \text{valor máximo}$$

$$PF (\text{real}) = 287 * 1.35 = 387.45$$

$$PF = (298.48 / 387.45) * 100 \% = 77 \%$$

El valor obtenido nos indica el porcentaje de funcionalidad o utilidad del sistema. Las razones para determinar la funcionalidad permiten indicar la calidad del producto en términos de productividad, coste, documentación y calidad.

#### 4.2.2 CONFIABILIDAD

La confiabilidad de un sistema puede definirse como la probabilidad de que un sistema funcione satisfactoriamente bajo un intervalo de tiempo específico. El objetivo, es una medida de la confiabilidad del sistema, conociendo la confiabilidad de sus componentes. Para este propósito es necesario distinguir dos consideraciones:

- ◆ El sistema entero falla si falla cualquiera de sus componentes (serie).
- ◆ El sistema falla si y solo si fallan todos sus componentes (paralelo).

El análisis de confiabilidad de los sistemas se lo realiza en base al modelo planteado en la siguiente figura 4.1, donde se observa las interconexiones de los subsistemas.

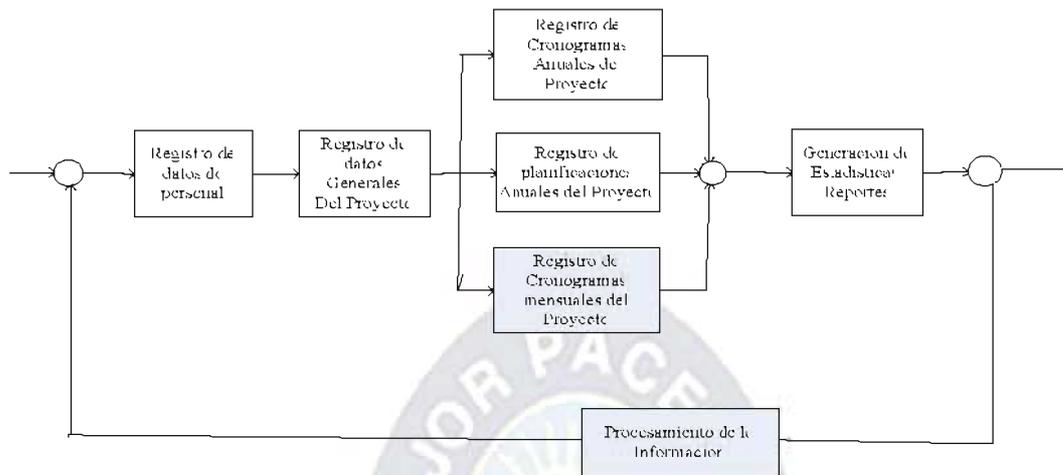


Figura 4.1 Modelo del sistema "SICOFAI"

Fuente: [Elaboración Propia]

Para formalizar el modelo de sistemas se presentan interconexiones de tipo mixto, por lo cual se analizará el siguiente diagrama de bloques:

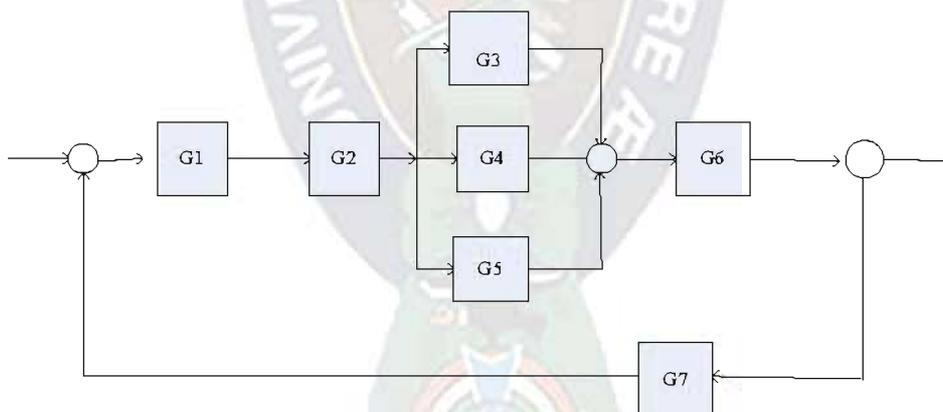


Figura 4.2 Diagrama de Bloques

Fuente: [Elaboración Propia]

Así:

$G_1$  = Registro De datos de personal.

$G_2$  = Registro de los datos generales del proyecto.

$G_3$  = Registro de cronogramas anuales del proyecto.

$G_4$  = Registro de Planificaciones anuales del proyecto.

$G_5$  = Registro de cronogramas mensuales del proyecto.

$G_6$  = Generación de estadísticas y reportes

$G_7$  = Datos de información procesada en un tiempo  $t$ .

Donde  $G_1, G_2, G_3, G_4, G_5, G_6$  y  $G_7$  son las funciones de transferencia de cada uno de los módulos. Así la función de transferencia  $G$  del sistema interconectado esta dada por la siguiente relación:

$$G(t) = \frac{y(t)}{u(t)} \quad \text{y} \quad Y(t) = G(t) * u(t)$$

Así el modelo del sistema se representa a través de la siguiente función de transferencia:

$$G(t) = \frac{[G_3 + G_4 + G_5] * G_1 * G_2 * G_6}{1 - \{ [G_3 + G_4 + G_5] * G_1 * G_2 * G_6 * G_7 \}}$$

Una vez obtenido el modelo del sistema, se procede al análisis de confiabilidad.

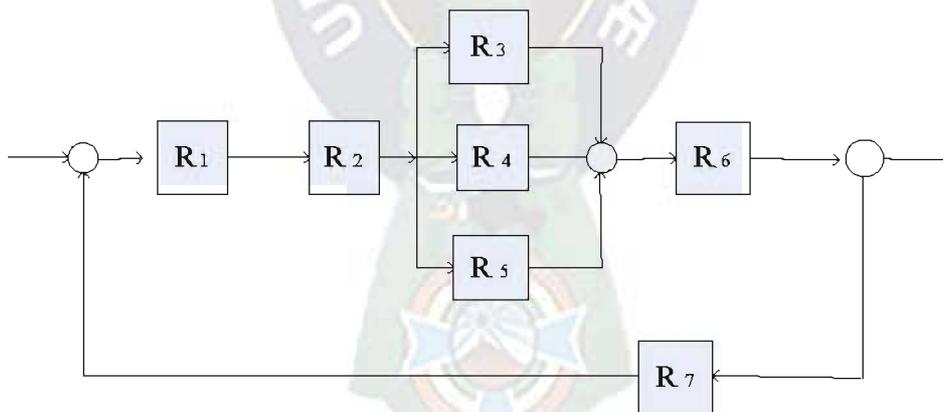


Figura 4.3 Subsistemas del "SICOFAI"

Fuente: [Elaboración Propia]

Para el análisis de cada subsistema se estudio la función exponencial como una aplicación de la teoría de confiabilidad, la cual define como “la confiabilidad  $R(t)$  de un componente determinado medio durante un periodo  $t$  se define como la probabilidad de que su tiempo para fallar excede a  $t$ ” [MOYA, 1998], es decir:

$$R(t) = P [T > t] = 1 - F(t)$$

Donde:

$R(t)$  = Confiabilidad de un componente o subsistema en el tiempo  $t$ .

$T$  = Tiempo para fallar o la duración del tiempo de trabajo sin falla.

$F(t)$  = Probabilidad de falla de un componente o subsistema en el tiempo  $t$ .

Donde:

$\lambda$  = Tasa de constantes de fallo. ( $\lambda = n^{\circ}$  de fallas de acceso)

$t$  = Periodo de operación en tiempo.

De donde se tienen los siguientes datos:

**$R_1$  = Registro de datos de personal**

$\lambda = 0.01$  y  $t = 4$  horas

$$R_1 = e^{-\lambda t} = e^{-(0.01)(4)} = 0.96 = 96\%$$

**$R_2$  = Registro de datos generales del proyecto**

$\lambda = 0.01$  y  $t = 4$  horas

$$R_2 = e^{-\lambda t} = e^{-(0.01)(4)} = 0.96 = 96\%$$

**$R_3$  = Registro de cronograma anual del proyecto**

$\lambda = 0.05$  y  $t = 4$  horas

$$R_3 = e^{-\lambda t} = e^{-(0.05)(4)} = 0.82 = 82\%$$

**$R_4$  = Registro de planificación anual del proyecto**

$\lambda = 0.07$  y  $t = 4$  horas

$$R_4 = e^{-\lambda t} = e^{-(0.07)(4)} = 0.75 = 75\%$$

**R<sub>5</sub> = Registro de cronograma mensual del proyecto**

$$\lambda = 0.04 \quad \text{y} \quad t = 4 \text{ horas}$$

$$R_5 = e^{-\lambda t} = e^{-(0.04)(4)} = 0.85 = 85\%$$

**R<sub>6</sub> = Generación de estadísticas y reportes**

$$\lambda = 0.02 \quad \text{y} \quad t = 4 \text{ horas}$$

$$R_6 = e^{-\lambda t} = e^{-(0.02)(4)} = 0.92 = 92\%$$

Para hallar la confiabilidad total del sistema, se consideran dos puntos de utilidad para determinar la confiabilidad del sistema.

Si  $n$  componentes, funcionan independientemente, están conectadas en serie, y el  $i$ -ésimo componente tiene confiabilidad del sistema  $R_i(t)$ , esta dada por:

$$R_T(t) = R_1(t) R_2(t) R_3(t) \dots R_n(t)$$

Si  $n$  componentes que funcionan independientemente actúan en paralelo, y el  $i$ -ésimo componente tiene confiabilidad  $R_i(t)$ , entonces la confiabilidad del sistema  $R_T(t)$ , es igual a:

$$R_T(t) = 1 - [1 - R_1(t)][1 - R_2(t)][1 - R_3(t)] \dots [1 - R_n(t)]$$

De este modo se procede al cálculo de confiabilidad:

$$R_T(t) = \{1 - [1 - R_3(t)][1 - R_4(t)][1 - R_5(t)]\} R_1(t) * R_2(t) * R_6(t)$$

$$R_T(t) = \{1 - [1 - 0.82][1 - 0.75][1 - 0.85]\} * 0.96 * 0.92 * 0.96$$

$$R_T(t) = 0.8393 * 100 \% = 84 \%$$

El resultado indica que el sistema presenta una confiabilidad de 84%, es decir que existe un 16% de probabilidad que el sistema pueda fallar excediendo un determinado tiempo de uso.

### 4.2.3 MANTENIBILIDAD

El estándar IEEE 982.1 – 1988[IEE94] sugiere un índice de madurez del software (IMS) que proporciona una indicación de la estabilidad de un producto software (basado en los cambios que se producen con cada versión del producto).

El índice de madurez del software se calcula mediante la ecuación matemática que sigue a continuación:

$$IMS = [M_i - (F_a + F_b + F_c)] / M_i$$

Donde:

$M_i$  = numero de módulos en la versión actual.

$F_a$  = numero de módulos en la versión actual que se han cambiado.

$F_b$  = numero de módulos en la versión actual que se han añadido.

$F_c$  = numero de módulos en la versión anterior que se han borrado en la versión actual.

A medida que el IMS se aproxima a 1.0 el producto empieza a estabilizarse.

$$M_i = 6$$

$$F_a = 0$$

$$F_b = 1$$

$$F_c = 0$$

$$IMS = [6 - (0 + 1 + 0)] / 6 = 5/6 = 0.83333 * 100\% = 83\%$$

Por lo tanto puede indicarse que el sistema tiende a estabilizarse en un 83%

### 4.2.4 PORTABILIDAD

La capacidad del software puede ser trasladado de un entorno a otro, tomando en cuenta los requerimientos mínimos de hardware mencionados en el capítulo 2

## CAPITULO V

### CONCLUSIONES Y RECOMENDACIONES

#### CONCLUSIONES

En esta etapa se brinda las conclusiones y las recomendaciones que se consideraron en el desarrollo del presente proyecto de grado.

- ✓ El sistema de información de control de proyectos, fue desarrollado satisfactoriamente, haciendo uso del proceso unificado de racional (RUP)
- ✓ La implantación del sistema de Control de proyectos en la fundación Arco Iris, se logro integrar los módulos de registro, seguimiento y reportes estadísticos de los proyectos.
- ✓ Se cuenta con reportes con información grafica, la cual permite observar a los beneficiarios de cada proyecto (niños-adolescentes).
- ✓ El sistema almacena en su base de datos la información de los proyectos como ser: registros, cronogramas, informes, planificaciones y estadísticas.
- ✓ Se logro desarrollar el módulo directorio, el cual brinda información de todas las instituciones o empresas y personas particulares con las que trabaja la Fundación Arco Iris.

## RECOMENDACIONES

Las recomendaciones que surgen a partir de la conclusión del proyecto de grado se detallan a continuación:

- ✓ Establecer una red entre los distintos proyectos del área social, pues la mayoría se encuentran en lugares distintos (edificios y lugares alejados).
- ✓ Se recomienda capacitar al personal que manipulara el sistema, para su buen funcionamiento.
- ✓ Por gran importancia de conexión con Internet, se recomienda la implantación del sistema vía Web.
- ✓ Para posteriores trabajos se recomienda desarrollar sistemas para las distintas áreas de la fundación como ser: área administrativa, áreas productivas, etc. para que de esta manera pueda ser de ayuda a la institución.





