

UNIVERSIDAD AUSTRAL DE CHILE
Facultad de Ciencias de la Ingeniería
Escuela de Ingeniería en Computación



Sistema de control de beneficios de funcionarios
municipalidad de Puerto Montt

**Seminario de Titulación
para optar
al título de
ingeniero de Ejecución en Computación**

**Profesor Patrocinante:
Sra. Viviana Alvarado Espinoza**

**Raúl Hardy Borquez Guerrero
Puerto Montt Chile 2003**

Dedicado a una persona que ya no está conmigo, pero que de igual forma la siento cerca, porque me enseñó, con su ejemplo, los valores más importantes de la vida.

A la memoria de mi padre.

Agradecimientos

A Dios por todas sus bendiciones y por darme la oportunidad de conocer a las personas que me han ayudado en este largo camino.

A mi madre y a mis hermanos quienes con su cariño y amor, me han apoyado en todos los momentos y me han alentado a seguir adelante.

A Johann, Angel y Carolina, quienes han sido mis mejores amigos, gracias por haber estado conmigo en los buenos y también en los difíciles momentos.

A mi profesora, la Sra. Viviana Alvarado, quien me ayudó en el desarrollo de este proyecto.

Y por supuesto a todas las personas que de una forma u otra me ayudaron a que esto fuera posible.

Muchas Gracias.

INDICE

1. Introducción.....	1
2. Objetivos.....	6
2.1 Objetivo General.....	6
2.2 Objetivos Específicos.....	6
3. Planteamiento del problema.....	8
3.1 Antecedentes.....	8
3.2 Justificación.....	11
3.3 Delimitación.....	11
4. Metodología.....	12
5. Recursos.....	24
5.1 Hardware.....	24
5.2 Software.....	25
6. Determinación de requerimientos de! sistema.....	26
6.1 Usuarios de! sistema.....	26
6.1.1 Entrevistas.....	27
6.2 Formularios y documentos.....	28
6.3 Informes.....	29
6.4 Requerimientos presentados por e! usuario.....	29

7. Diseño del sistema.....	36
7.1 Diagrama de flujo de datos.....	36
7.2 Diccionario de datos.....	40
7.2.1 Procesos.....	41
7.2.2 Entidades.....	45
7.2.3 Flujos de datos.....	46
8. Diseño de la Base de Datos.....	53
8.1 Diseño conceptual.....	53
8.1.1 Identificación de entidades.....	54
8.1.2 Identificación de alias y ocurrencia de cada entidad....	57
8.1.3 Identificación de relaciones.....	59
8.1.4 Identificación y asociación de atributos con una entidad o relación.....	63
8.1.5 Determinación del dominio de los atributos.....	66
8.1.6 Identificación de claves candidatas y claves primarias para las entidades.....	70
8.1.7 Modelo conceptual Entidad-Relación.....	72
8.2 Diseño de bases de datos lógico para el Modelo Relacional.....	74
8.2.1 Construcción y validación del modelo de datos lógico local para cada vista de usuario.....	74
8.2.2 Transformación del modelo de datos local conceptual al local lógico.....	75

8.2.3 Eliminar relaciones mucho a mucho.....	75
8.2.4 Eliminar relaciones mucho a mucho	
complejas de tres o más entidades.....	76
8.2.5 Eliminar relaciones recursivas.....	76
8.2.6 Eliminar relaciones con atributos.....	77
8.2.7 Eliminar atributos Multi-valóricos.....	77
8.2.8 Reexaminar relaciones de uno a uno.....	77
8.2.9 Eliminar relaciones redundantes.....	78
8.2.10 Derivación de relaciones del modelo de datos	
lógico.....	79
8.2.11 Validación del modelo utilizando normalización.....	82
8.2.11.1 Primera forma normal.....	84
8.2.11.2 Segunda forma normal.....	84
8.2.11.3 Tercera forma normal.....	85
8.2.12 Validación del modelo contra transacciones de	
usuario.....	86
8.2.13 Diagrama entidad relación del diseño lógico.....	89
8.2.14 Definición de restricciones de integridad.....	91
8.2.14.1 Datos requeridos.....	91
8.2.14.2 Restricciones de dominio de los atributos...	91
8.2.14.3 integridad de entidades.....	92
8.2.14.4 Integridad referencial.....	92

8.3 Diseño de Base de Datos Físico.....	94
8.3.1 Traducir el modelo lógico al DBMS seleccionado.....	95
8.3.2 Diseñar la representación física.....	95
8.3.3 Diseñar mecanismos de seguridad.....	95
8.3.4 Transformación del modelo de datos lógico para un DBMS seleccionado.....	96
8.3.4.1 Diseño de relaciones bases para un DBMS correspondiente.....	96
8.3.4.2 Diseño de restricciones de la empresa para un DBMS específico.....	97
8.3.5 Diseño de la representación física.....	98
8.3.5.1 Análisis de transacciones.....	99
8.3.5.2 Elección de índices secundarios.....	102
8.3.5.3 Consideraciones en la introducción de redundancia controlada (De normalización).....	102
8.3.5.4 Estimación del espacio requerido en disco.....	104
8.3.6 Diseño de mecanismos de seguridad.....	105
8.3.6.1 Diseño de vistas de usuario.....	106
8.3.6.2 Diseño de reglas de acceso.....	107
8.3.6.3 Control de usuarios.....	108
9. Seleccionar DBMS.....	109
10. Diseño de la aplicación.....	110

10.1	Definición de los módulos del sistema.....	111
10.2	Diseño de la interfaz.....	114
10.2.1	Usabilidad.....	117
10.2.2	Utilidad.....	118
10.2.3	Facilidad de uso.....	118
10.2.4	Facilidad de aprendizaje.....	119
10.2.5	Apreciación.....	121
10.3	Procedimientos para el diseño de interfaz.....	124
10.3.1	Recolección y análisis de información de usuario.....	124
10.3.2	Diseñar la interfaz de usuario.....	125
10.3.3	Referentes a la presentación de la información.....	126
10.3.4	Referente al análisis del color.....	126
10.3.5	Construcción de la interfaz de usuario.....	128
10.3.6	Estandarización de pantallas.....	128
10.4	Diseño procedimental.....	138
11.	Desarrollo del software.....	144
11.1	Creación de la base de datos.....	144
11.1.1	Tablas del sistema.....	144
11.2	Procedimientos del sistema.....	151
12.	Prueba.....	165

12.1 Prueba de unidad.....	166
12.1.1 Interfaz.....	167
12.1.2 Estructuras de valores de datos inválidos, tipos de datos y formato de datos.....	167
12.1.3 Manejo de valores válidos para los datos.....	168
12.1.4 Procedimiento de manejo de errores.....	168
12.1.5 Condiciones de seguridad.....	168
12.2 Prueba de integración.....	169
12.3 Prueba de validación.....	170
12.4 Pruebas del sistema.....	171
12.4.1 Prueba de recuperación.....	171
12.4.2 Pruebas de seguridad.....	172
12.4.3 Pruebas de rendimiento.....	173
13. Implantación y Evaluación.....	174
14. Conclusiones y recomendaciones.....	177
15. Bibliografía.....	179
16. Anexos.....	181
Anexo 1.....	181
Anexo 2.....	185

TABLAS

Tabla N ° 1. Descripción de los procesos.....	41
Tabla N ° 2. Descripción de las entidades.....	45
Tabla N ° 3. Descripción de flujo de datos.....	46
Tabla N ° 4. Identificación de atributos y descripción de entidades.....	54
Tabla N ° 5. Identificación de alias y ocurrencia de las entidades.....	57
Tabla N ° 6. Identificación de relaciones.....	60
Tabla N ° 7. Identificación de existencia de relaciones.....	62
Tabla N ° 8. identificación y asociación de atributos con una entidad o relación.....	64
Tabla N ° 9. Documentación de atributos y ejemplos.....	66
Tabla N ° 10. Identificación de las claves candidatas y primarias.....	70
Tabla N ° 11. Listado de transacciones.....	87
Tabla N ° 12. Integridad Referencial.....	93
Tabla N ° 13. Ejemplos de claves primarias y secundarias.....	97
Tabla N ° 14. Análisis de transacciones.....	101
Tabla N ° 15. Funciones de los botones del sistema.....	127
Tabla N ° 16. Simbología diagrama de flujo de datos.....	181
Tabla N ° 17. Simbología para el modelo entidad relación.....	184

FIGURAS

Figura N° 1. Esquema de la metodología.....	13
Figura N° 2. Diagrama de flujo primer nivel.....	38
Figura N° 3. Diagrama de flujo de datos segundo nivel.....	39
Figura N° 4. Diagrama de flujo de datos tercer nivel.....	40
Figura N° 5. Modelo Conceptual Entidad-Relación.....	73
Figura N° 6. Mapa Transaccional.....	88
Figura N° 7. Modelo ER lógico.....	90
Figura N° 8. Esquema del menú.....	113
Figura N° 9. Pantalla de seguridad.....	130
Figura N° 10. Formulario de ingreso de calidad.....	131
Figura N° 11. Formulario de ingreso de datos del funcionario.....	132
Figura N° 12. Registro de atenciones médicas.....	133
Figura N° 13. Búsqueda de funcionarios por calidad.....	134
Figura N° 14. Este es el reporte de la planilla de pago mensual.....	135
Figura N° 15. Reporte de calidad.....	136
Figura N° 16. Reporte de datos de un funcionario específico con sus respectivas cargas familiares.....	137
Figura N° 17. Instalación y configuración del sistema.....	175

SINTESIS

En el siguiente informe se describirá el desarrollo del "Sistema de Control de Beneficios de Funcionarios", que ha sido diseñado para la Municipalidad de Puerto Montt.

A través de este informe, se detallarán los procedimientos y técnicas utilizadas para lograr un sistema que dé solución al problema de automatizar los procesos de información de los beneficiarios. Para realizar este sistema se ha empleado una metodología de diseño denominada "Ciclo de vida Clásico" del autor James Senn, y dentro de ésta se ha utilizado otra con el nombre de "Ciclo de Vida de una base de datos", del autor James Conolly, para el diseño de la base de datos.

El objetivo principal que se presenta en este informe es dar una solución automatizada al proceso de controlar los beneficios de los funcionarios que actualmente se realiza en forma manual.

Para el desarrollo del sistema, se han empleado diferentes herramientas tales como: Visual Basic 6.0 (herramienta de desarrollo), SQL Server 7.0 (motor de base de datos), Crystal Report 8.0 (herramienta de diseño de reportes), Microsoft Visio 5.0 (para el diseño de diagramas).

Como resultado de este desarrollo, se podrá contar con una herramienta de software que permita controlar los beneficios de los usuarios lo cual ayudará a optimizar los flujos de información, de manera más eficiente, confiable y segura

SYNTHESIS

In the following report the development of the "Sistema de control de beneficios" that has been designed for the Municipality of Puerto Montt.

Through this report, the procedures and techniques will be detailed used to achieve a system that gives solution to the problem of automating the processes of the beneficiaries' information. To carry out this system a methodology of the author's design denominated "Ciclo de vida clásico" James Senn it has been used, and inside this another has been used with the name of "Ciclo de Vida de una Base de Datos", of the author James Conolly, for the design of the database.

The main objective that is presented in this report is to give an automated solution to the process of controlling the benefits of the officials that at the moment is carried out in form manual.

For the development of the system, different such toots have been used as: Visual Basic 6.0 (development toot), SQL Server 7.0 (database motor), Crystal Report 8.0 (toot of design of reports), Microsoft Visio 5.0 (for the design of diagrams).

As a result of this development, he/she will be able to have a software tool that allows to control the benefits of the users that which will help to optimize the flows of information, in a more efficient, reliable and sure way.

1. INTRODUCCION

En la actualidad debido a la necesidad que tienen las empresas e instituciones de manejar datos en forma fácil y rápida, ha llevado a que se desarrollen sistemas de información. Estos sin duda desde bastante tiempo han sido de gran utilidad cuando se trata de manejar grandes cantidades de datos.

Una de las instituciones donde se procesa bastante información son los municipios. En ellos existen una serie de departamentos, como el departamento de bienestar.

El departamento de bienestar de la Municipalidad de Puerto Montt, está encargado de administrar los beneficios a los que pueden optar los funcionarios que pertenecen a ella. Para llevar a cabo esta labor debe manejar gran cantidad de información del funcionario, como antecedentes generales, calidad, dependencia, cargas familiares, cupo de dinero, prestaciones médicas, entre otras.

Entre los beneficios más importantes de los funcionarios que están inscritos en el departamento de bienestar, es el de salud, donde a cada funcionario se le asigna una cantidad fija de dinero disponible para el

reembolso, cuando éste accede a una prestación médica. Por cada atención o consulta médica, cierta cantidad de este dinero disponible se va restando al saldo anterior. Uno de los problemas presentados es gestionar la información contable y generar los reportes respecto a diferentes parámetros de consulta.

Actualmente, este trabajo se realiza en forma manual, o sea las personas encargadas del departamento de bienestar almacenan la información general de sus afiliados, en planillas Excel.

Por otra parte, se tienen planillas Excel que almacenan las prestaciones médicas mensuales a las que acceden los funcionarios. Luego, se debe confeccionar la planilla de pago mensual, donde se seleccionan uno a uno los valores por cada funcionario. Finalmente, se genera gran cantidad de carpetas con la información, respecto a las prestaciones médicas de cada funcionario.

Dado el engorroso e ineficiente proceso descrito anteriormente, se hace necesario que se construya un sistema de información que ayude a solucionar este problema el cual será descrito en este informe.

Para el desarrollo del sistema se utilizará la metodología denominada “Ciclo de Vida Clásico” [Senn 1992]. Dentro de esta metodología se utilizará

El séptimo capítulo corresponde al diseño del sistema, donde se detalla el diagrama de flujo de datos.

En el octavo capítulo se desarrolla el diseño de la base de datos.

En el noveno capítulo se selecciona un DBMS.

El décimo capítulo corresponde al diseño de la aplicación. En este capítulo se desarrolla el diseño de la interfaz y el diseño de los procedimientos que se implementarán posteriormente en el sistema.

El undécimo capítulo, corresponde al desarrollo del software, esto implica, la creación de la base de datos, las tablas del sistema y los procedimientos que se deben codificar.

El duodécimo capítulo, corresponde a las etapas de prueba.

En el décimo tercer capítulo se realizan la implantación y evaluación del sistema.

El décimo cuarto capítulo corresponde a las conclusiones de este proyecto.

Finalmente, como anexo se mostrarán la simbología utilizada en los diagramas y el documento que el departamento de bienestar entregó al departamento de informática donde se plantea el proyecto.

2. OBJETIVOS

2.1. - Objetivo General

El objetivo general que se pretende alcanzar con este proyecto es analizar, diseñar, construir e implementar un sistema de control de beneficios de funcionarios de la Municipalidad de Puerto Montt. Este debe ser confiable, seguro y rápido, orientado a Windows, y en formato gráfico.

2.2. - Objetivos Específicos

- Utilizar la tecnología de las bases de datos, aprovechando su flexibilidad, fácil manejo, clara visualización de los datos de acuerdo a su estructuración (esto corresponde a la construcción de las tablas de datos).
- Llevar un registro de todos los funcionarios con sus cargas familiares.

- Controlar los beneficios de salud, como los valores que se reembolsan cuando un funcionario se realiza una prestación médica, considerando el cupo disponible que éste posee para dicho proceso.

- Controlar los beneficios de escolaridad para los funcionarios y sus cargas familiares.

- Generar los reportes que sean necesarios para mostrar funcionarios, cargas familiares, funcionarios clasificados por ciertos parámetros, ya sea por calidad o dependencia, prestaciones médicas realizadas en ciertos períodos de tiempo etc.

- Generar la planilla de pago de reembolsos médicos.

- Generar la planilla de pago de beneficios de escolaridad.

3. PLANTEAMIENTO DEL PROBLEMA

3.1. - Antecedentes

- **Problema a resolver:**

El problema que se desea resolver es mejorar el lento y prolongado proceso de entrega y manejo de la información que existe en el departamento de bienestar de la municipalidad de Puerto Montt, con respecto a los beneficios que los funcionarios y sus cargas familiares pueden optar. Entre estas mejoras se deben incluir los reportes por pantalla e impresora cuando se realizan búsquedas por ciertos parámetros.

Actualmente el proceso de controlar los beneficios se hace en forma manual, considerando que el tiempo que se necesita para poder realizar este proceso es bastante prolongado. Uno de los problemas presentados es gestionar la información contable, donde también se debe solucionar el problema de generar la planilla de pago mensual.

- **Esfuerzos anteriores**

Actualmente no se ha hecho ningún esfuerzo en materia de desarrollo de sistema. Este problema es planteado por primera vez por el departamento de bienestar al departamento de informática de la Municipalidad de Puerto Montt. Actualmente el esfuerzo que se realiza es mantener los datos almacenados en planilla Excel.

- **Definición de solución**

La solución que se propone al problema antes mencionado es realizar un sistema de control de beneficios de los funcionarios de la municipalidad de Puerto Montt, que permita automatizar los procesos, para así poder entregar en forma rápida y oportuna la información requerida.

- **Equipo de trabajo:**

En este proyecto se considerarán:

- Un alumno tesista el cual estará a cargo del desarrollo del proyecto.
- Dos funcionarios pertenecientes al departamento de bienestar de la municipalidad, los cuales proporcionarán los requerimientos y una vez terminado, utilizarán el sistema.
- Profesionales pertenecientes al departamento de informática de la municipalidad los cuales calificarán si el sistema satisface los requerimientos especificados, además estos realizarán la mantención del sistema.

3.2.- Justificación

Debido al rápido avance de la tecnología y la aparición en el mercado de herramientas de desarrollo de software, se ha provocado que las empresas estén cada día mas interesadas en automatizar sus procesos. Esto es una necesidad cuando se trata de manejar grandes cantidades de datos. Es el caso del departamento de bienestar donde se encuentra toda la información de los funcionarios que trabajan en la municipalidad de Puerto Montt.

3.3. - Delimitación:

Este proyecto abarcará las etapas de diseño, construcción, prueba e implementación. La etapa del sistema que no se abordará en este proyecto es la mantención, ya que esto implica que continuamente se esté revisando, y por lo tanto, esta etapa quedará a cargo de los profesionales del departamento de informática de la Municipalidad.

4. METODOLOGIA

El propósito al utilizar una metodología es poder establecer un enfoque que contemple procedimientos, técnicas, herramientas y documentación que vayan en apoyo a las distintas etapas del proyecto. La metodología a usar para el desarrollo de este proyecto se denomina “Ciclo de Vida Clásico” (obtenido del libro “Análisis y Diseño de Sistemas de Información” [Senn1992]).

Para la etapa de diseño de la Base de Datos, se utilizaron las actividades propuestas específicamente dentro de la metodología “Ciclo de vida de una aplicación de Base de Datos” [Connolly 1999], cuyas actividades están claramente especificadas.

También se utilizaron técnicas que sirvieron de apoyo y documentación en el diseño de la interfaz, como es el caso de los criterios propuestos por Eduardo Mercovich [Introducción a la Usabilidad y el Diseño de Interfaces] y los propuestos por Lewis y Reiman [1993] “Diseño de Interfaz de Usuario”.

A continuación se presentará el esquema de la metodología.

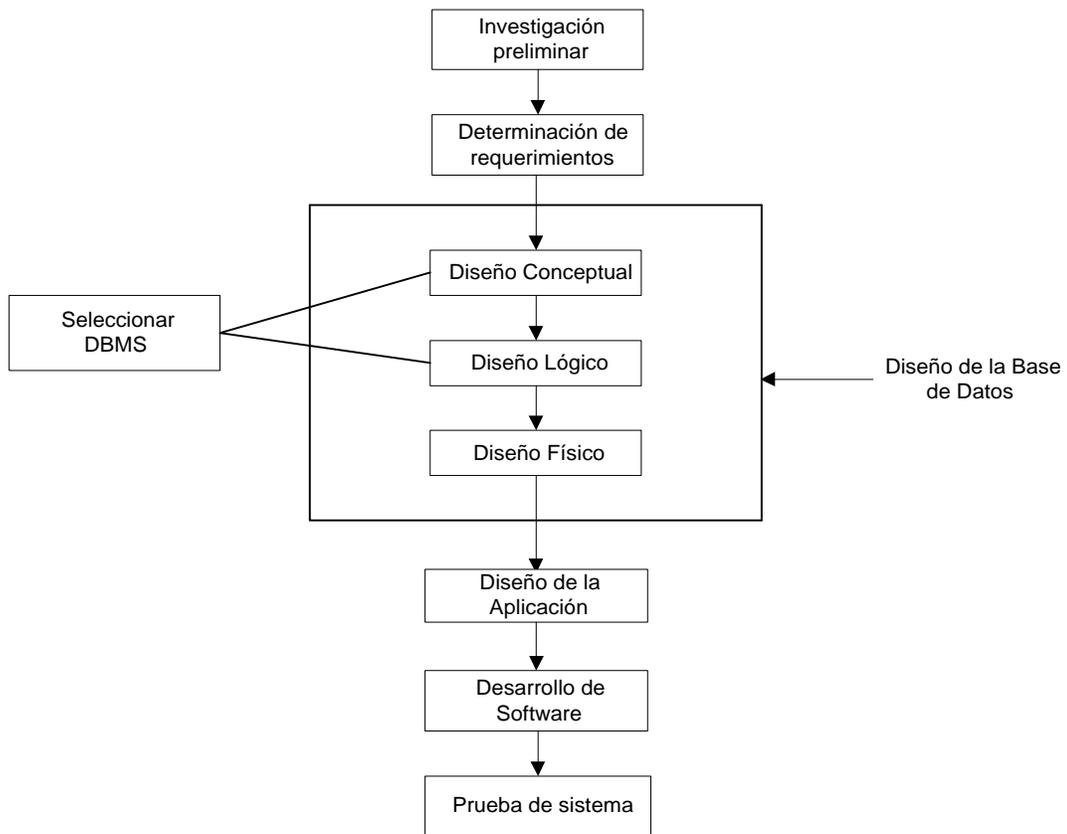


Figura N° 1 : Esquema de la metodología

Las etapas que incluye la metodología utilizada corresponden a la del “Ciclo de Vida Clásico” Senn [1992], las cuales son:

- Investigación preliminar:

Esta etapa se inicia con la solicitud del sistema por parte de una persona, empresa o especialista en sistemas. Posteriormente, se realiza un estudio de factibilidad para poder realizar el proyecto. Aquí se consideran tres puntos: factibilidad técnica, económica y operacional. Si se llega a la conclusión de que el proyecto es posible realizarlo, se aprueba la solicitud.

Esta etapa no se realizará porque el proyecto ya había sido aprobado por el departamento de informática de la Municipalidad.

- Determinación de Requerimientos del sistema.

Se trata de la definición de los objetivos globales del sistema. Aquí se realizan entrevistas con el usuario para así poder comprender todos los puntos importantes de la parte de la empresa que se encuentra bajo estudio, para poder obtener respuestas a diversas preguntas que surgen. Aquí se debe

realizar un estudio sobre los requerimientos con la finalidad de identificar las características que debe tener el nuevo sistema, incluyendo la información que deben producir los sistemas junto con características operacionales tales como controles de procesamiento, tiempo de respuesta y métodos de entrada y salida.

- Diseño del sistema

El diseño de un sistema de información produce los detalles que establecen la forma en la que el sistema debe cumplir con los requerimientos identificados durante la etapa de análisis, lo que se denomina diseño lógico. Dentro de esta etapa, se encuentra el diseño de la Base de Datos y el diseño de la Aplicación, que se describen a continuación.

- a) Diseño de Base de Datos.

Es el proceso por el cual se crea un diseño para bases de datos que apoye las operaciones y objetivos de la empresa. Se utilizará parte de la metodología denominada “Ciclo de Vida de una Aplicación de Base de Datos” [Connolly 1999].

Los principales objetivos del diseño de la base de datos son:

- Representar los datos y relaciones entre ellos para todas las áreas de aplicación y los grupos de usuarios.
 - Proveer de un modelo de datos que soporte cualquier transacción sobre los datos.
 - Especificar un mínimo diseño que sea apropiadamente estructurado para obtener el rendimiento identificados en los requerimientos anteriores.
- Diseño Conceptual.

Es el proceso de construir un modelo de la información en la organización, independiente de todas las consideraciones físicas. Siempre se debe estar comprobando el modelo E-R con los requerimientos. El objetivo de este tipo de diseño es poder tener una aproximación descriptiva de cómo se manejan en la organización particular.

- Diseño Lógico.

Esta etapa de la metodología se basa en un modelo de datos específico, pero independiente de un DBMS en particular u otras consideraciones físicas.

Se construye y valida el modelo de datos lógico local para cada usuario sobre la base del esquema conceptual de cada vista del usuario de la empresa y después se valida el modelo utilizando la técnica de normalización y con las transacciones requeridas. Luego de validado cada uno de los modelos locales en uno solo que pueda ser utilizado, se representa la parte de la empresa que nos interesa. Una vez combinado los modelos puede ser necesario validar el modelo global, primero contra las reglas de normalización, y después contra las transacciones descritas en cada vista de usuario para luego estudiar posibles crecimientos a futuro, cuando se superan estas etapas se dibuja el modelo Entidad/Relación final.

Cada una de las etapas hasta aquí descritas supone revisar junto con el usuario el modelo generado, esto con el objetivo de que el esquema resultante se adapte fielmente a la realidad de la empresa.

- Diseño Físico.

Es el proceso de producir una descripción de la implementación de la base de datos en almacenamiento secundario, además describe la estructura de almacenamiento y métodos de acceso para obtener un eficiente acceso a los datos.

El objetivo de esta etapa es mostrar como se ve la implementación del diseño lógico de la base de datos.

- Seleccionar DBMS.

Esta etapa involucra seleccionar el DBMS (DataBase Management System o Sistema Administrador de Bases de Datos) apropiado para el sistema. Si no existe el DBMS, se recomienda realizar esta selección entre las fases de diseño conceptual y diseño lógico. Lo importante es asegurar que cumpla con funcionalidades de seguridad, fácil implementación y reestructuración, y reglas de integridad. Cabe destacar que esta fase de la metodología es opcional.

b) Diseño de la Aplicación.

Dentro de esta etapa se encuentra el diseño de interfaz y el diseño de procedimental, los cuales se describen a continuación.

- Diseño de interfaz

Es importante el lugar que ocupa la interfaz en todo sistema, porque es el principal medio de comunicación con el exterior. Las interfaces deben cumplir con estándares “amigables para el usuario” y sin perder de vista la identidad que se le quiere dar a ésta. Una de las técnicas de diseño de interfaz es la propuesta por Eduardo Mercovich (fundador y líder de GaiaSur Consultora desde 1994 y se especializa en diseño de interfaces y usabilidad), donde plantea que para el proceso del desarrollo de una interfaz se deben considerar cuatro puntos importantes. Estos son:

- Usabilidad.
- Utilidad.
- Facilidad de uso.
- Facilidad de aprendizaje.
- Apreciación.

Los puntos anteriores se complementaron con la metodología de interfaz propuesta por Lewis y Reiman, la cual presenta las siguientes consideraciones

- Velocidad de aprendizaje
- Velocidad de respuesta
- Tasa de errores
- Retención
- Satisfacción

Los puntos anteriormente mencionados serán explicados mas adelante en el capítulo N° 10 sección 10.2.

- Diseño procedimental

El diseño procedimental es donde se realiza la estructura del programa y de los datos. Aquí se deben especificar los procedimientos que definen los detalles algorítmicos [Pressman,1993].

- Desarrollo de software

Corresponde a la creación de las definiciones externas, conceptuales e internas de la base de datos (su creación) y de los programas (codificación) de la aplicación.

- Prueba de sistemas

En esta etapa se instalará el sistema para comprobar que éste funcione como fue requerido. Durante la fase de prueba de sistemas, el sistema se emplea de manera experimental para asegurarse de que el software no tenga fallas, es decir que funciona de acuerdo con las especificaciones y en la forma en que los usuarios esperan que lo haga. Se alimentan como entradas conjuntos de datos de prueba para su procesamiento y después se examinan los resultados.

La estrategia de prueba del software es la propuesta por Roger Pressman en su libro "Ingeniería de Software". Se probará el sistema para ver los múltiples fallos que pudiera tener. Estas se harán en el siguiente orden.

- Prueba de Unidad.
 - Prueba de Integración.
 - Prueba de Validación.
 - Prueba del Sistema.
-
- Implantación y Evaluación

La implantación es el proceso de instalar la aplicación. Dependiendo del tamaño de la organización que empleará la aplicación y el riesgo asociado con su uso, puede elegirse comenzar la operación del sistema sólo en un área de la empresa (prueba piloto), por ejemplo en un departamento con una o dos personas. Algunas veces se deja que los dos sistemas, el viejo y el nuevo, trabajen en forma paralela con la finalidad de comparar los resultados. En otras circunstancias, el viejo sistema deja de utilizarse determinado día para comenzar a emplear el nuevo al día siguiente. Cada estrategia de implantación tiene sus méritos de acuerdo con la situación que se considere dentro de la empresa. Sin importar cual sea la estrategia utilizada, los encargados de desarrollar el sistema procuran que el uso inicial del sistema se encuentre libre de problemas.

La evaluación de un sistema se lleva a cabo para identificar puntos débiles y fuertes. La evaluación ocurre a lo largo de cualquiera de las siguientes dimensiones:

- Evaluación operacional
- Impacto organizacional
- Opinión de los administradores
- Desempeño del desarrollo

La evaluación del proceso de desarrollo de acuerdo con criterios tales como tiempo y esfuerzo de desarrollo, concuerdan con presupuestos y estándares, y otros criterios de administración de proyectos.

5. RECURSOS

5.1. - Hardware

- Para implementar este Sistema se necesita básicamente un computador como Servidor, una Red y uno o varios computadores que actúen como cliente.

- Las características de los computadores son las sgtes:

Cliente:

- Procesador : Pentium II de 350 Mhz o superior.
- Ram : 64 MB.
- Disco Duro : 10 GB.

Servidor:

- Procesador : Pentium III de 600 Mhz.
- Ram : 128 MB.
- Disco Duro : 10 GB.

Para el desarrollo del proyecto se utilizará, un computador con las sgtes. características:

- Procesador : Intel Celeron de 700 Mhz.
- Ram : 128 MB.
- Disco Duro : 10 GB.

5.2. - Software

- Visual Basic 6.0 como software de desarrollo.
- Crystal Report 8. Este software es para generar los reportes.
- Motor de Base de Datos SQL Server 7.0. Este se utilizará como servidor y actuará como motor de base de datos, el cual estará inserto dentro de una red de área local y se encuentra incluido en el sistema operativo Windows 98.
 - Windows 98 como sistema operativo.

6. DETERMINACION DE REQUERIMIENTOS DEL SISTEMA

El objetivo de ésta etapa se logró a través de entrevistas con el usuario, donde éstas sirvieron de ayuda para comprender de mejor manera todos los puntos importantes del problema que se encuentra bajo estudio, ayudando a aclarar todas las diversas dudas que puedan surgir.

La fuente de información más importante fueron los requerimientos entregados por el mismo departamento de bienestar de la Municipalidad de Puerto Montt.

6.1 Usuarios del sistema

Los usuarios del sistema son generalmente la primera fuente de información a investigar por el analista. De éstos se extrae toda la información de las actividades que actualmente se realizan. Como se indicaba anteriormente, no existe un sistema que controle los beneficios de los usuarios, por lo tanto, se tendrá que estudiar y analizar los datos que actualmente se encuentran físicamente en informes y en planilla Excel.

Una de las formas de investigar los requerimientos es realizar entrevistas y a veces cuestionarios. Como el departamento de bienestar había entregado de antemano un documento con los requerimientos, se realizaron entrevistas que no fueron extensas y que estuvieron enfocadas a definir un poco más los detalles del sistema.

6.1.1 Entrevistas

Esta técnica es una de las más utilizadas, pues permite recopilar información más detallada a medida que se van aclarando las dudas. Esta fase es bastante iterativa, ya que después de cada entrevista surgen nuevas dudas lo que conlleva a que se realicen cambios o se agreguen más requerimientos. Todo esto es de gran utilidad, ya que después de cada entrevista existe una mayor aproximación del problema a resolver.

Se debe dejar en claro que se entrevistó a un solo grupo de usuarios, ya que todos los funcionarios que trabajan en el departamento de bienestar realizan la misma labor. De estos se obtuvo una visión completa del sistema y objetivos que desean satisfacer con el nuevo sistema de información. No está de más mencionar que como futuros usuarios del sistema, también adquieren el

compromiso de entregar toda la información necesaria para que el objetivo del sistema sea cumplido en su totalidad.

Otro punto importante, fue realizar entrevistas con el departamento de informática para ver cuales son los requerimientos de hardware y de software. Si bien es cierto que el sistema fue propuesto a este departamento se debe dejar en claro que estos trabajan con cierto software (los que fueron presentados en el capítulo 5 de recursos).

6.2 Formularios y documentos

Esta fuente de información empieza con la obtención de lista de documentos y/o formularios. La idea principal de ésta técnica es analizar todos los documentos que se obtienen de la actual forma de trabajo, para así poder determinar cuales son las entradas y las salidas (informes) del sistema.

Uno de los documentos más importante es la planilla de pago mensual, la cual posee los datos y el valor a cancelar a cada funcionario que se realiza una o varias atenciones médicas en un mes.

6.3 Informes

Esta fuente se utiliza como base para las entrevistas con el usuario. Aquí se pueden determinar los nuevos requerimientos y nuevas salidas.

En este caso se utilizaron los informes que el departamento tiene en planillas Excel como única información de salida.

A continuación se mostrarán los requerimientos más importantes que el departamento de bienestar le entregó al departamento de informática.

6.4 Requerimientos presentados por el usuario

El sistema debe ser capaz de recibir datos como:

Antecedentes del funcionario:

- Apellido Paterno
- Apellido Materno
- Nombres
- Rut
- Fecha de nacimiento

- Calidad
- Dependencia
- Remuneración imponible
- Nº cargas familiares
- Antecedentes de las cargas familiares
- Régimen de salud (Isapre o Fonasa)

Calidad:

La calidad corresponde al tipo de contrato que el funcionario posee.

Existen actualmente cinco alternativas:

- Planta
- Contrata
- Cementerio (Código del trabajo) Esto corresponde a los funcionarios que trabajan en el cementerio.
- Dem (Dirección de Educación Municipal) (Ley 19.070 y Código del trabajo para funcionarios permanentes en la Dem)
- Salud (Código del trabajo para funcionarios permanentes en la Dirección de Salud Municipal)

Dependencia:

En lo que se refiere a la dependencia del Funcionario es a que Dirección está asignado, Ejemplo:

- Alcaldía
- Adm. y Finanzas
- Jurídica
- Tránsito
- Aseo y Ornato
- Secretaría
- Cementerio
- Dem
- Salud

Las cargas familiares deben tener los siguientes datos.

- Apellido paterno
- Apellido materno
- Nombres
- N° de rut
- Fecha de nacimiento

- Clasificación (Hijo, Cónyuge, etc)
- Fecha de inicio y término del beneficio.

Al digitar el Rut del funcionario deberá aparecer la siguiente información:

- Antecedentes de todos los funcionarios con sus respectivas cargas familiares.
- Cupo total en dinero disponible para el reembolso.

Hay que considerar que también se pueden mostrar los datos antes mencionados por distintos parámetros de búsqueda, ya sea por el primer apellido, por el nombre, por calidad, dependencia, entre otros.

El sistema debe ser capaz almacenar datos como:

Las prestaciones Médicas desglosadas por centro de costos con N° de boleta o documento y código de la prestación.

Ejemplo:

- a) Consultas Médicas (Código 01)
- b) Exámenes (Código 02)
- c) Medicamentos (Código 03)
- d) Programa Médico (Código 04)
- e) Atención Dental (Código 05)

Que al momento de ingresar una atención médica, calcule en forma interna el 90% del valor de bonificación y lo rebaje automáticamente del saldo existente.

El sistema debe generar los siguientes listados:

- a) Listado alfabético de funcionarios por:
 - Calidad
 - Dependencia (Dirección)

b) Listado de cargas familiares por:

- Calidad
- Dependencia

c) Listado por centro de costos ya sea por:

- Funcionario
- Calidad
- Dependencia
- Código prestación

El sistema debe ser capaz de generar la Planilla de pago mensual con el detalle del efectivo a pagar. En dicha planilla deberán aparecer los siguientes datos:

- Datos del Funcionario como nombre completo.
- N ° de Rut.
- Saldo anterior disponible
- Prestación
- Valor afecto a bonificación
- Total afecto a bonificación

- Valor afecto a pagar y firma funcionario.

Otras consideraciones

Cuando se desee ingresar una atención médica, el sistema deberá comprobar si el funcionario tiene cupo o saldo disponible para reembolso. De no ser así, debe bloquear el ingreso.

Para cualquier beneficio que sea, deberá ser indispensable que se ingrese el número de rut del funcionario, de tal forma que, cuando se ingrese un documento para obtener un beneficio, el programa sea capaz de rechazar automáticamente aquellos rut de personas que no sean beneficiarios del sistema, o sea, que estén desactivados.

Que al sistema se le pueda ingresar alguna otra prestación nueva con su respectivo código y porcentaje de bonificación.

También que se puedan ingresar funcionarios nuevos al sistema con sus respectivos datos.

7. DISEÑO DEL SISTEMA

El diseño del sistema de información es el que produce los detalles que establecen la forma en la que el sistema cumplirá con los requerimientos identificados durante la etapa de análisis lo que se denomina diseño lógico. A continuación se presentará la simbología a utilizar, el diagrama de flujo y el diccionario de datos donde se mostrará y se explicará en detalle los procesos, las entradas y las salidas del sistema.

7.1 Diagrama de flujo de datos

El diagrama de flujo de datos es una herramienta gráfica que se emplea para describir y analizar el movimiento de datos a través de un sistema, sea éste manual o automatizado, incluyendo procesos y lugares para almacenar datos en el sistema. Los diagramas de flujo de datos son las herramientas más importantes y la base sobre la cual se desarrollan otros componentes.

A continuación se presentará el diagrama de flujo de datos donde se describen los procesos que se realizan en el sistema.

La simbología utilizada en este sistema esta detallada en el anexo N° 1.

DFD: 0

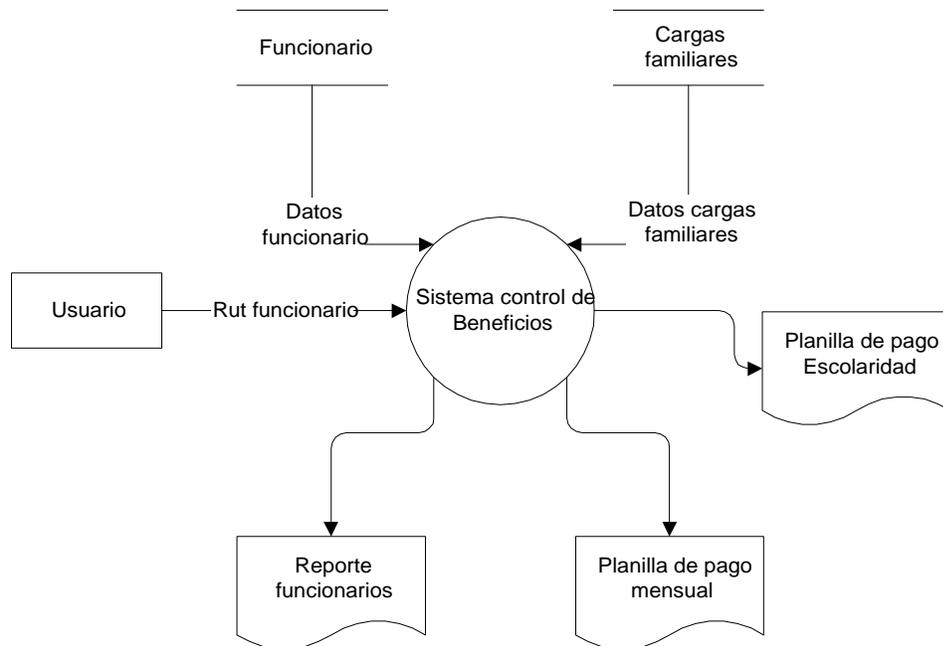


Figura N° 2. Diagrama de flujo primer nivel

DFD:1

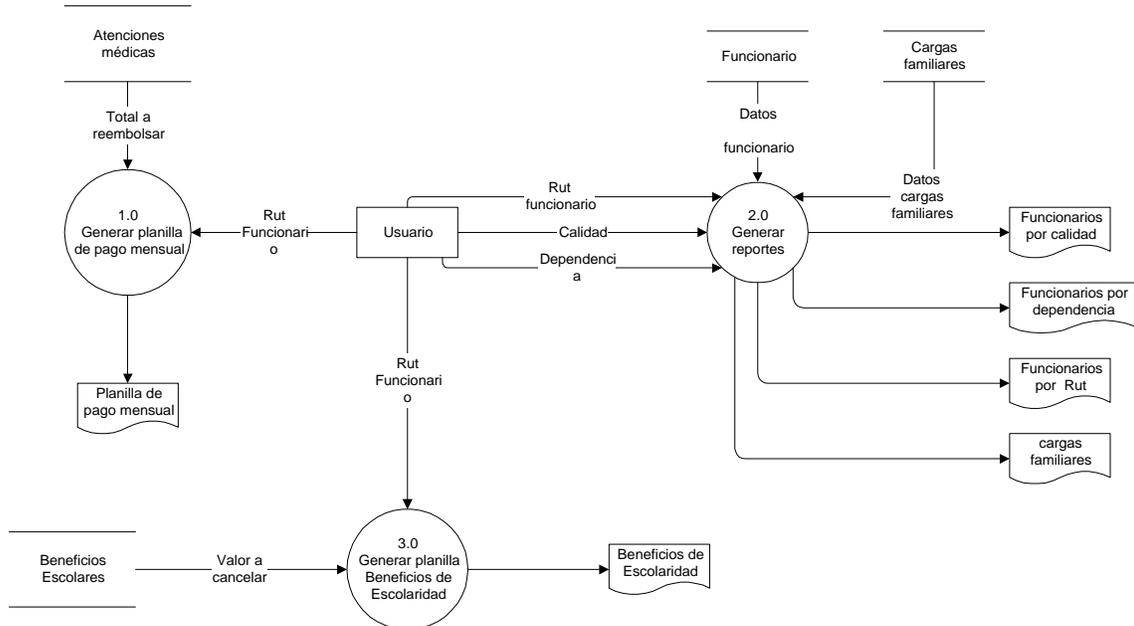


Figura N°3. Diagrama de flujo de datos segundo nivel

DFD:2

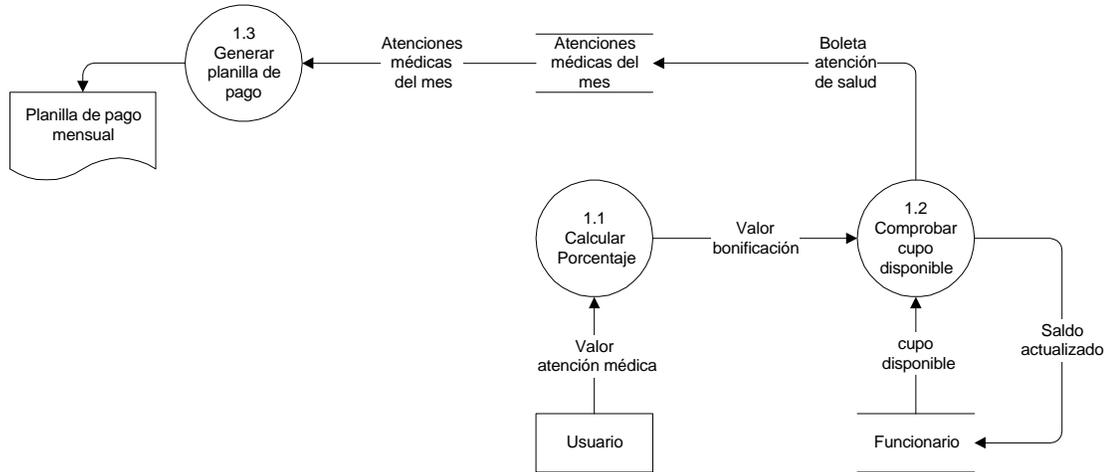


Figura Nº 4. Diagrama de flujo de datos tercer nivel

7.2 Diccionario de Datos

El diccionario de datos contiene las características lógicas de los sitios donde se almacenan los datos del sistema, incluyendo nombre, descripción, alias, contenidos y organización.

7.2.1 Procesos

Tabla N ° 1. Descripción de los procesos

Nombre	Calcular porcentaje
Descripción	Este proceso calcula el porcentaje de bonificación del valor de una prestación médica
Entradas	Valor atención médica
Salida	Valor de bonificación
Algoritmo	Valor_bonificación= (Valor_ Atención) * (porcentaje)

Nombre	Comprobar cupo disponible
Descripción	Este proceso comprueba si el funcionario tiene cupo disponible para reembolso
Entradas	- Valor bonificación - Cupo disponible

Salida	Saldo actualizado
Algoritmo	<pre> If cupo_disponible > valor_bonificación then Restar_saldo_anterior </pre>

Nombre	Generar planilla de pago.
Descripción	Proceso que obtiene todos los datos del funcionario y el valor a reembolsar con sus respectivos detalles.
Entradas	Atenciones médicas del mes.
Salida	Planilla de pago mensual.
Algoritmo	Select * funcionario

Nombre	Generar reportes
Descripción	Proceso que genera diferentes reportes de datos de funcionarios y cargas familiares con respecto a determinadas consultas

Entradas	<ul style="list-style-type: none"> - Datos del funcionario - Datos de las cargas familiares
Salida	<p>Datos de Funcionarios por:</p> <ul style="list-style-type: none"> - Calidad - Dependencia - Rut - Nombre - Apellidos - Fecha de Nacimiento <p>Datos de Cargas familiares por:</p> <ul style="list-style-type: none"> - Funcionario - Nombre - Apellidos - Fecha de Nacimiento
Algoritmo	<pre>Select * From funcionario Where Parametro = Valor Select * from cargas Where Parametro =Valor</pre>

Nombre	Generar planilla de beneficios de escolaridad
Descripción	Proceso que obtiene los datos de los funcionarios a los que a sus cargas o a él mismo se le ha asignado un valor de bonificación de escolaridad.
Entradas	<ul style="list-style-type: none"> - Rut del funcionario - Valor a cancelar
Salida	Planilla de pago de beneficios de escolaridad.

7.2.2 Entidades

Tabla N ° 2. Descripción de las entidades.

Nombre	Usuario
Descripción	Es la entidad externa que maneja el sistema.
Flujos de datos que origina	<ul style="list-style-type: none">- Rut funcionario- Calidad- Dependencia- Valor atención médica

7.2.3 Flujos de datos

Tabla N ° 3. Descripción de flujo de datos.

Nombre	Valor atención médica
Descripción	Valor de la atención médica que se realiza una persona
Proceso al que llega	Calcular porcentaje
Almacén de origen de la información	Usuario que ingresa el valor

Nombre	Valor bonificación
Descripción	Es el valor de una atención médica que está reflejado en una boleta emitida por el mismo sistema de salud (Isapre o Fonasa).
Proceso al que llega	Comprobar cupo disponible
Proceso de origen de la información	Calcular porcentaje

Nombre	Cupo disponible
Descripción	Es el cupo disponible para reembolso que se asigna a cada funcionario
Proceso al que llega	Comprobar cupo disponible
Almacén de origen de la información	Funcionario

Nombre	Saldo actualizado
Descripción	Es el nuevo valor que se obtiene luego que se a restado al saldo anterior disponible después que se ha comprobado su cupo
Almacén de dato al que llega	Funcionario
Proceso que origina la información	Comprobar cupo disponible

Nombre	Boleta atención de salud
Descripción	Datos de la atención médica realizada por la por el funcionario
Almacén de dato al que llega	Atenciones médicas del mes
Proceso que origina la información	Comprobar cupo disponible
Estructura que se utilizan	<ul style="list-style-type: none"> - Número_boleta - Rut_Paciente - Codigo_prestación - Valor atención

Nombre	Atenciones médicas del mes
Descripción	Atenciones realizadas durante el mes en curso
Proceso al que llega	Generar planilla de pago
Almacén de origen de la información	Atenciones médicas del mes
Estructura que se utilizan	<ul style="list-style-type: none"> - Número_boleta - Rut_Paciente - Codigo_prestación - Valor atención <p>Todas las correspondientes al mes</p>

Nombre	Datos cargas familiares
Descripción	Todos los antecedentes de las cargas familiares
Proceso al que llega	Generar reportes
Almacén de origen de la información	Cargas familiares
Estructura que se utilizan	<ul style="list-style-type: none"> - Rut - Apellido Paterno - Apellido Materno - Nombres - Fecha de nacimiento - Fecha de término beneficio

Nombre	Rut funcionario
Descripción	Rut del funcionario para generar reportes.
Procesos al que llega	<ul style="list-style-type: none"> - Generar reportes - Generar planilla de Beneficios de escolaridad
Entidad de origen de la información	Usuario

Nombre	Calidad
Descripción	Calidad que es ingresada para obtener reportes de funcionario por calidad
Proceso al que llega	Generar reportes
Entidad de origen de la información	Usuario
Nombre	Dependencia
Descripción	Dependencia que es ingresada para obtener reportes de funcionario por dependencia
Proceso al que llega	Generar reportes
Entidad de origen de la información	Usuario

Nombre	Datos funcionario
Descripción	Datos del funcionario
Proceso al que llega	Generar Reportes
Almacén de origen de la información	Funcionario
Estructura que se utilizan	Nombre Apellido paterno Apellido materno Fecha de nacimiento

	Fecha de inicio al sistema Calidad Dependencia Salud Remuneración imponible Cupo disponible para reembolso
--	---

Nombre	Total a reembolsar
Descripción	Es el valor que se debe reembolsar mensualmente a cada funcionario
Proceso al que llega	Generar planilla de pago mensual
Almacén de origen de la información	Atenciones médicas

Nombre	Valor a cancelar
Descripción	Es el valor que se debe cancelar a un funcionario por conceptos de beneficios escolares
Proceso al que llega	Generar planilla de pago de beneficios de escolaridad

Almacén de origen de la información	Beneficios escolares
--	----------------------

8. DISEÑO DE LA BASE DE DATOS

Esta etapa de la metodología se divide en tres partes conocidas como: Diseño conceptual, diseño lógico, diseño físico. Las cuales serán descritas a continuación.

8.1 Diseño conceptual

El objetivo de esta etapa es construir un diseño de base de datos local y conceptual de la empresa. El esquema se basa principalmente en la obtención de entidades y de cómo estas se relacionan entre sí. Una vez obtenidas se proceden a definir sus atributos, identificamos las claves candidatas, de las cuales saldrá más tarde la clave primaria. Una vez realizada esta tarea se procede a dibujar el modelo Entidad-Relación. Esta etapa del diseño sirve como base para el diseño lógico.

8.1.1 Identificación de entidades

Se procede a definir cuales serán las entidades que pertenecerán al modelo E-R, dando su descripción y también se identificarán los atributos de cada una de ellas.

Tabla N ° 4. Identificación de atributos y descripción de entidades.

Nombre	Descripción de atributo	Descripción Narrativa
Funcionario	<ul style="list-style-type: none">- Rut- Nombre- Apellido paterno- Apellido materno- Fecha nacimiento- Remuneración- Cupo disponible para reembolso- Fecha de inicio	Persona que recibe el o los beneficios de reembolso médico y/o escolaridad.
Salud	<ul style="list-style-type: none">- Código_salud- Nombre_salud	Es el tipo de sistema de salud que posee el funcionario.
Calidad	<ul style="list-style-type: none">- Código_calidad	Corresponde al tipo de

	- Nombre_calidad	contrato que posee el funcionario.
Dependencia	- Código_dependencia - Nombre_dependencia	Corresponde a la dirección a la que pertenece el funcionario.
Cargas Familiares	- Rut - Nombre - Apellido paterno - Apellido materno - Fecha Nacimiento - Fecha de Terminación Beneficio - Parentesco	Persona(s) que son dependientes de los funcionarios en materia de beneficios de salud y escolaridad.
Atención Médica	- Número_boleta - Rut_Paciente - Valor atención - Valor bonificación	Atenciones médicas que son solicitadas por los pacientes, en este caso pueden ser los funcionarios o cargas familiares.

Prestación Médica	<ul style="list-style-type: none"> - Código_prestación - Nombre_prestacion 	Información sobre las prestaciones médicas.
Nivel Escolar	<ul style="list-style-type: none"> - Código_nivel - Nombre_nivel 	Es el nivel escolar que puede tener el funcionario o la carga familiar
Beneficios escolares	<ul style="list-style-type: none"> - Rut funcionario - Rut_beneficiario - Valor_beneficio - Año escolar 	Es el beneficio al que pueden optar las cargas familiares o los funcionarios que estén estudiando.

8.1.2 Identificación de alias y ocurrencia de cada entidad.

A continuación se dará a conocer el alias que en algunos de los casos se les asigna a las entidades y la ocurrencia, que corresponde al momento en que es necesaria esa entidad en el sistema.

Tabla N ° 5. Identificación de Alias y ocurrencia de las entidades.

Entidad	Alias	Ocurrencia
Funcionario	No tiene	Se crea cada vez que se ingresa un nuevo funcionario.
Salud	No tiene	Se crea cada vez que se ingresa un nuevo sistema de salud.
Calidad	No tiene	Se crea cuando se ingresa una nueva calidad.
Dependencia	No tiene	Se crea cuando se ingresa una nueva dependencia.
Cargas Familiares	Cargas	Estas se crean de acuerdo a un funcionario específico, cada vez que se

		ingrese una nueva carga familiar.
Atención Médica	Boleta	Las atenciones médicas se crean cada vez que un funcionario se realiza una atención médica.
Prestación Médica	Prestación	Se crean todas las prestaciones médicas una sola vez en el sistema, y se agrega a medida que se vayan creando otras.
Nivel escolaridad	Nivel	Corresponde al nivel de escolaridad
Beneficios escolares	Escolar	Corresponde a los beneficios escolares que quedan registrados

8.1.3 Identificación de relaciones

Luego de tener claras las entidades se procede a identificar las relaciones que son las encargadas de conectar una con otra a las entidades, también hay que determinar la cardinalidad y existencia del tipo de la relación, permitiendo con estas características verificar y mantener la calidad de los datos.

Cuando se habla de cardinalidad se refiere a la cantidad de ocurrencia con que una entidad está relacionada con otra. La cardinalidad puede consistir en uno a uno (1:1), uno a muchos (1:N), muchos a uno (N:1), o muchos a muchos (N:N).

Existe otro concepto que es la existencia, se tienen 2 tipos: Mandatoria (M), donde todas las tuplas de la entidad deben estar involucradas en la relación, y Opcional (O), en la cual no es obligatorio que todas las tuplas estén asociadas a las relación.

A continuación en la tabla N° 6, se identificarán las relaciones y las entidades que asocia. Además se mostrará la clave con que está asociada cada entidad, una descripción de cada una de ellas y su cardinalidad.

Tabla N ° 6. Identificación de relaciones

Nombre	Clave	Entidades que Relaciona	Decripción Narrativa	Cardina Lidad
Tiene	Código_salud	Salud Funcionario	Relación con respecto al tipo de salud del funcionario	1:N
Tiene	Código_calidad	Calidad Funcionario	Relación con respecto a la calidad del funcionario	1:N
Tiene	Código_dependencia	Dependencia Funcionario	Cada funcionario pertenece a una dependencia	1:N

Tiene	Código_nivel	Nivel_escolar Beneficios Escola Res	Cada beneficio puede tener un valor diferente de acuerdo al nivel escolar	1:N
Tiene	Código_prestación	Prestación médica Atención médica	Todas las atenciones médicas tienen una prestación	1:N
Tiene	Rut_funcionario	Funcionario Beneficios escolares	Los beneficios de escolaridad están asignados a un funcionario	1:N
Tiene	Rut_funcionario	Funcionario Cargas familiares	Todas las cargas familiares están asignadas a un funcionario	1:N
Solicita	Rut_funcionario	Funcionario	Todas las	1:N

		Atención médica	atenciones médicas están asociadas a un funcionario	
--	--	--------------------	---	--

Tabla N ° 7. Identificación de existencia de relaciones

Entidad 1	Relación	Entidad 2	Existencia
Funcionario	Tiene	Salud	M:M
Funcionario	Tiene	Calidad	M:M
Funcionario	Tiene	Dependencia	M:M
Nivel escolar	Tiene	Beneficios escolares	M:M
Prestación médica	Tiene	Atención médica	M:M
Funcionario	Tiene	Beneficios escolares	M:O
Funcionario	Tiene	Cargas familiares	M:O
Funcionario	Solicita	Atención médica	M:O

8.1.4 Identificación y asociación de atributos con una entidad o relación

Cada entidad o relación tiene asociado atributos que son datos que pertenecen a ésta y entregan información referente a esa entidad o relación en particular.

Las tablas N ° 8 y N ° 9 hacen referencia a la entidad o relación entregando todos sus atributos, describiéndolos y adicionando información referente a éstos.

Nomenclatura Tabla N ° 9

VN = Valor Nulo.

D = Derivado.

M = Multivalórico.

C= Compuesto.

Tabla N ° 8. Identificación y asociación de atributos con una entidad o relación

Entidad o relación	Atributos	Restricción	VN	D	M	C
Funcionario / Carga	Rut		No	No	No	No
	Dígito		No	No	No	No
	Nombre		No	No	No	No
	Apellido_paterno		No	No	No	No
	Apellido_materno		No	No	No	No
	Fecha_nacimiento	Formato **/**/**** (*) Caracteres numéricos	No	No	No	No
	Fecha_inicio	Formato **/**/**** (*) Caracteres numéricos	No	No	No	No
Funcionario	Remuneración		No	No	No	No

Carga	Fecha_término	Formato **/**/**** (*) Caracteres numéricos	No	Si	No	No
	Parentesco	Letra A, D, P	No	No	No	No
Calidad	Código_calidad		No	No	No	No
	Nombre_calidad		No	No	No	No
Dependencia	Código_dependencia		No	No	No	No
	Nombre_dependencia		No	No	No	No
Salud	Código_salud		No	No	No	No
	Nombre_salud		No	No	No	No
			No	No	No	No
Boleta	Número_boleta		No	No	No	No
	Valor_boleta		No	No	No	No
	Bonifica_boleta		No	Si	No	No
Prestación	Código_prestación		No	No	No	No
	Nombre_prestación		No	No	No	No
Nivel	Código_nivel		No	No	No	No
	Nombre_nivel		No	No	No	No

8.1.5 Determinación de dominio de los atributos

El dominio es un rango de valores válidos que posee un atributo de una entidad o relación

Tabla N ° 9. Documentación de atributos y ejemplos.

Nombre	Dominio	Descripción Narrativa	Ejemplos
Rut	Numérico (8)	Rut de personas	13406027
Dígito	Carácter (1)	Dígito verificador del rut	1/2/3...9/k
Nombre	Cadena de caracteres (30)	Nombre de las personas	Miguel
Apellido materno	Cadena de caracteres (15)	Apellidos materno de las personas	Aguilar
Apellido paterno	Cadena de caracteres (15)	Apellidos paterno de las personas	Tenorio
Fecha de nacimiento	Cadena de caracteres (8)	Fecha de nacimiento	23/11/1978
Remuneración	Numérico (8)	Remuneración	200000

		imponible	
Fecha de inicio	Cadena de caracteres (8)	Fecha de inicio al sistema	23/05/2003
Código_calidad	Alfanumérico (10)	Código calidad	1.1.2
Código_dependencia	Alfanumérico (10)	Código dependencia	1.2.3
Código_salud	Alfanumérico (10)	Código del sistema de salud	1.2.3.4
Nombre_salud	Cadena de caracteres (40)	Nombre del tipo de salud, Isapre o Fonasa.	Isapre o Fonasa
Nombre_calidad	Cadena de caracteres (40)	Nombre de la calidad	Contratado
Nombre_dependencia	Cadena de caracteres (40)	Nombre de la dependencia	Alcaldía

Fecha de término beneficio	Cadena de caracteres (8)	Fecha de término de beneficio de bienestar	23/11/1996
Parentesco	Carácter (1)	Letra que indica el tipo de parentesco ascendiente o descendiente	A o D o P
Número_boleta	Cadena de caracteres (10)	Número de boleta de atención médica	237452-368
Valor_boleta	Numérico (8)	Valor de la atención médica	10000
Bonifica_boleta	Numérico (8)	Valor de bonificación	9000
Código_prestación	Alfanumérico (10)	Código de las prestaciones médicas	01,02...
Nombre_prestación	Cadena de caracteres (40)	Nombre de la prestación médica	Oftalmología

Código_nivel	Carácter (1)	Código del nivel educacional	01,02,03...
Nombre_nivel	Cadena de caracteres (30)	Nombre del nivel educacional	Pre-Básica Básica Media Superior
Rut_paciente	Numérico (8)	Rut del paciente	13406027
Rut_beneficiario	Numérico (8)	Rut del beneficiario	13406027
Año_escolar	Numérico (4)	Año escolar	2003

En este caso, no existen atributos para las relaciones.

8.1.6 Identificación de las claves candidatas y claves primarias para las entidades.

Las claves candidatas, también llamadas claves alternas, son atributos únicos que no se repiten entre tuplas. De las claves candidatas, se selecciona una, la cual será la clave primaria. Por su parte la clave primaria, es el o los atributos que permiten acceder en forma unívoca a una fila de una tabla.

Las entidades fuertes son las que tienen una clave primaria escogida de entre las claves candidatas, por su parte las débiles existen gracias a la existencia de otra entidad y no tienen clave primaria.

A continuación en la tabla N ° 10 se determinan las claves candidatas y primarias a partir de su correspondiente entidad.

Tabla N ° 10: Identificación de las claves candidatas y primarias.

Entidades	Claves Candidatas	Claves Primarias
Calidad	Codigo_calidad;nombre_calidad	Codigo_calidad

Dependencia	Codigo_dependencia;nombre_dependencia	Codigo_dependencia
Salud	Codigo_salud;nombre_salud	Codigo_salud
Prestación	Codigo_prestacion;nombre_prestacion	Codigo_prestación
Nivel	Codigo_nivel;nombre_nivel	Codigo_nivel
Funcionario	Rut_funcionario	Rut_funcionario
Cargas	Funcionario_cargas;Rut_cargas	Funcionario_carga;Rut_carga
Boleta	Numero_boleta;funcionario_boleta;paciente_boleta	Numero_boleta;funcionario_boleta;paciente_boleta
Escolar	Funcionario_escolar;beneficiario_escolar	Funcionario_escolar;beneficiario_escolar

8.1.7 Modelo conceptual Entidad Relación

De acuerdo a la metodología empleada en esta fase del Sistema, se finaliza el Diseño Conceptual con la entrega correspondiente de éste y después de un análisis el modelo Entidad Relación que se presenta en la figura N ° 5.

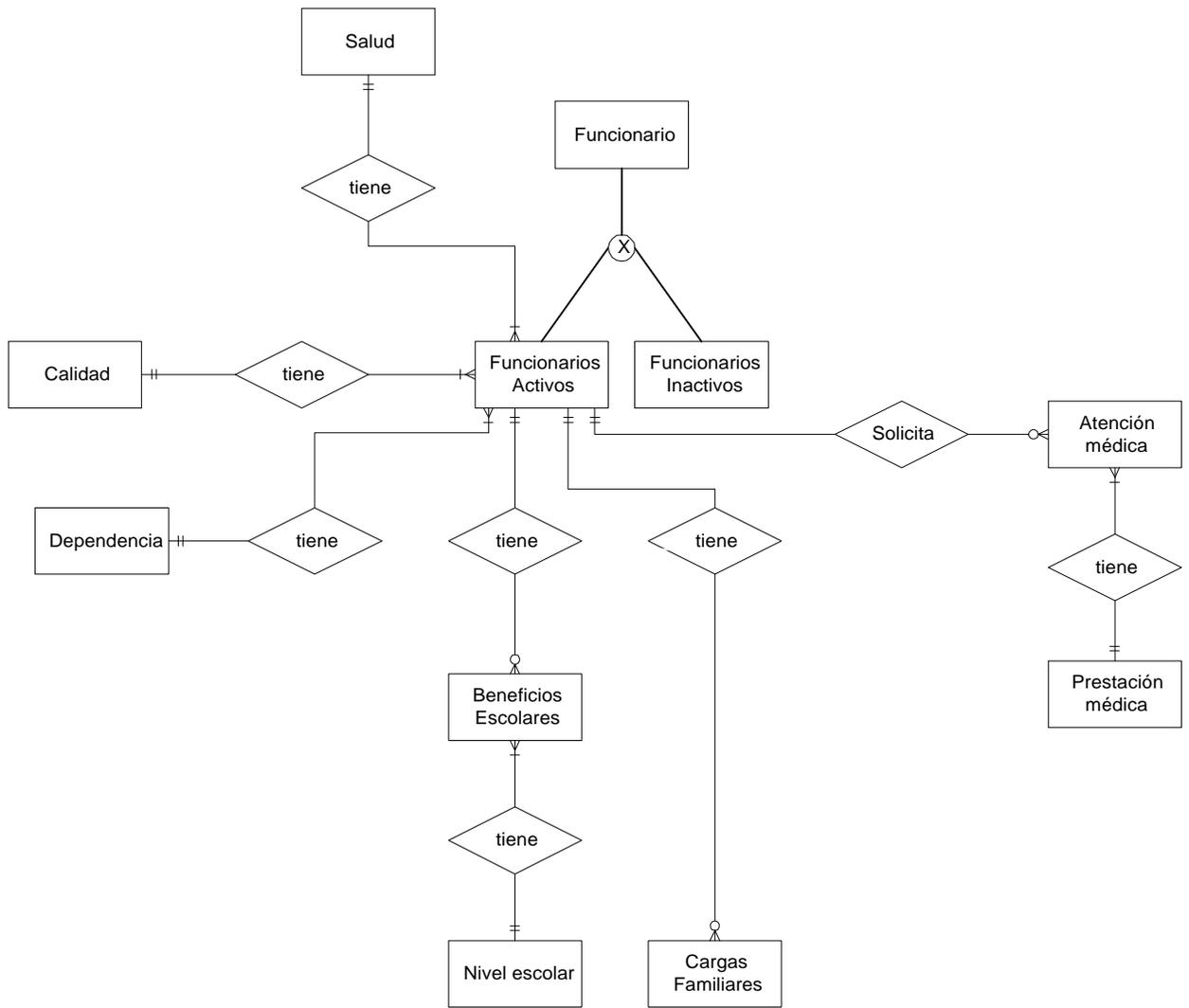


Figura N° 5. Modelo Conceptual Entidad-Relación

8.2 Diseño de Bases de Datos Lógico para el Modelo Relacional

Para esta fase de la metodología son importantes 2 pasos que son:

- Construcción y validación del modelo de datos lógico local para cada vista de usuario.
- Construcción y validación del modelo de datos lógico global.

8.2.1 Construcción y validación del modelo de datos lógico local para cada vista de usuario.

En esta etapa se debe construir y validar el modelo de datos lógico local para cada vista de usuario. Para la construcción del modelo de datos lógico, se basa en el diseño conceptual de cada vista de usuario de la empresa y para validar el modelo se utiliza la técnica de normalización y con las transacciones requeridas.

8.2.2 Transformación del modelo de datos local conceptual al local lógico.

En el modelo ER, obtenido en la etapa anterior, se comienzan a remover algunas características no deseables para luego transformar este modelo, al modelo de datos lógico global.

Para entregar el modelo de datos global se deben seguir los siguientes pasos:

8.2.3 Eliminar relaciones de mucho a mucho

Se descompone la relación identificando una entidad intermedia. Se reemplaza la relación N:N con dos relaciones 1:N asociadas con la nueva entidad.

En este paso no se realizó porque el esquema no presenta relaciones mucho a mucho. No hubo que generar nuevas relaciones.

8.2.4 Eliminar relaciones mucho a mucho complejas de tres o más entidades

Se descompone la relación identificando una entidad intermedia. Se reemplaza la relación compleja con el número requerido de relaciones 1:N (binarias) asociadas con la nueva entidad.

Para este proyecto no existen relaciones mucho a mucho complejas.

8.2.5 Eliminar relaciones recursivas

Se descompone la relación identificando una entidad intermedia. Si es una relación recursiva de 1:N entonces se reemplaza con la entidad intermedia y una nueva relación adicional de uno a uno. Si es una relación recursiva de N:N entonces se hace lo mismo que en el primer tipo de eliminación (relación mucho a mucho).

El MER conceptual (figura N° 5), no presenta relaciones recursivas, todas las entidades están bien definidas independientemente unas de otras, lo que significa que no se deben generar nuevas entidades.

8.2.6 Eliminar relaciones con atributos

Se descompone la relación identificando una entidad intermedia. A esta entidad se le asigna el atributo y después se relaciona en las entidades anteriores con nuevas relaciones.

No existen atributos para las relaciones anteriormente presentadas.

8.2.7 Eliminar atributos Multi-valóricos

Este atributo se descompone formando una entidad. La nueva relación es de 1:N la cual asocia esta entidad recién formada.

Para este proyecto no existen atributos multi-valóricos.

8.2.8 Reexaminar relaciones de uno a uno

Se identifica si dos entidades representan el mismo objeto en la empresa. Si representan el mismo objeto, se unen las entidades. Si las claves

primarias son distintas se elige una como clave primaria y la otra como clave alterna.

En el diagrama ER del modelo conceptual (fig. 5), todas las entidades representan objetos diferentes.

8.2.9 Eliminar relaciones redundantes

Una relación es redundante si se puede obtener la misma información utilizando otra relación. Se puede identificar si existe más de un camino hacia la entidad, pero no significa que una de las relaciones es redundante. La dimensión de tiempo es importante para decidir si es o no redundante.

Todas las entidades representan objetos diferentes, lo que significa que no existe redundancia.

8.2.10 Derivación de relaciones del modelo de datos lógicos

Se describe la relación utilizando un lenguaje de definición de base de datos (DBDL).

Lo importante en esta etapa es que la relación entre entidades se refleja por el mecanismo de clave primaria / clave foránea. Aquí se deben identificar las entidades padre e hijo. La entidad padre almacena su clave como clave foránea en la entidad hijo.

1) Funcionario (rut_funcionario, digrut_funcionario, paterno_funcionario, materno_funcionario, nombre_funcionario, fecnac_funcionario, calidad_funcionario, salud_funcionario, dependencia_funcionario, cupo_funcionario, remune_funcionario, fecha_funcionario, fecini_funcionario)

Primary key (rut_funcionario)

Foreing key (calidad_funcionario)

Foreing key (salud_funcionario)

Foreing key (dependencia_funcionario)

2) Calidad (codigo_calidad, nombre_calidad)

Primary key (codigo_calidad)

Alternate key (nombre_calidad)

- 3) Carga (rut_carga, digrut_carga, paterno_carga, materno_carga, nombre_carga, funcionario_carga, parentesco_carga, fechanac_carga, fechaini_carga, fechater_carga)
Primary key (rut_carga)
Foreing key (funcionario_carga)

- 4) Dependencia (codigo_dependencia, nombre_dependencia)
Primary key (codigo_dependencia)
Alternate key (nombre_dependencia)

- 5) Prestacion (codigo_prestacion, nombre_prestacion)
Primary key (codigo_prestacion)
Alternate key (nombre_prestacion)

- 6) Salud (codigo_salud, nombre_salud)
Primary key (codigo_salud)
Alternate key (nombre_salud)

- 7) Boleta (numero_boleta, funcionario_boleta, paciente_boleta, saldo_boleta, valor_boleta, bonifica_boleta, prestacion_boleta, fecha_boleta)
Primary key (numero_boleta)
Foreing key (funcionario_boleta, paciente _boleta)

Foreign key (prestacion_boleta)

8) Usuario (codigo_usuario, nombre_usuario, clave_usuario, grupo_usuario)

Primary key (codigo_usuario)

9) Escolar (funcionario_escolar, beneficiario_escolar, nivel_escolar, valor_escolar, fecha_escolar)

Primary key (funcionario_escolar)

Foreign key (funcionario_escolar, beneficiario_escolar)

A continuación se presentarán las tablas de paso

10) His_Funcionario (rut_funcionario, digrut_funcionario, paterno_funcionario, materno_funcionario, nombre_funcionario, fecnac_funcionario, calidad_funcionario, salud_funcionario, dependencia_funcionario, cupo_funcionario, remune_funcionario, fecha_funcionario, fecini_funcionario)

Primary key (rut_funcionario)

Foreign key (calidad_funcionario)

Foreign key (salud_funcionario)

Foreign key (dependencia_funcionario)

11)His_Carga (rut_carga, digrut_carga, paterno_carga, materno_carga, nombre_carga, funcionario_carga, parentesco_carga, fechanac_carga, fechaini_carga, fechater_carga)
Primary key (rut_carga)
Foreing key (funcionario_carga)

8.2.11 Validación del modelo utilizando normalización

En esta etapa el objetivo es validar el modelo de datos lógico utilizando la técnica de normalización. Normalización es el proceso de decidir cuales atributos deben estar juntos en un tipo entidad. Además asegura que el modelo resultante sea consistente, con redundancia mínima y máxima estabilidad.

Efectos de malas descomposiciones y fallos en el diseño:

- Incapacidad para almacenar ciertos hechos.
- Pérdida de información.
- Valores nulos (inaplicables).
- Anomalías de inserción.
 - Redundancia e inconsistencia.

- Imposibilidad de insertar tuplas, existencia de valores nulos.
- Anomalías de modificación.
 - Necesidad de propagar modificaciones por un diseño redundante.
- Anomalías de borrado.
 - Pérdida de datos.
- Necesidad de borrar varias tuplas para eliminar un solo elemento.

Formas normales básicas

- La teoría de la normalización se centra en lo que se conoce como “formas normales”. Se dice que un esquema relacional está en una determinada forma normal si satisface un conjunto específico de restricciones.
- Codd propuso en 1970 tres formas normales basadas en las dependencias funcionales (1FN, 2FN, 3FN).
- Debido a que en 3FN aún persisten ciertos problemas en determinadas relaciones en 1974 Boyce y Codd introdujeron una definición más restrictiva de la 3FN que denominó Forma Normal de Boyce-Codd (FNBC).
- Una relación se encuentra en Forma Normal de Boyce-Codd si, y sólo si

- Para toda dependencia no trivial $X \rightarrow Y$, X es una clave candidata de la relación. Esta soluciona problemas que presenta la 3FN en relaciones con varias claves candidatas que se solapan.

Finalmente se puede concluir que la normalización produce un diseño de base de datos flexible que fácilmente se puede extender.

8.2.11.1 Primera forma normal (1FN)

Se dice que una tabla está en primera forma normal (1FN) si no existen grupos repitentes (columnas) dentro de la fila.

8.2.11.2 Segunda forma normal (2FN)

Una tabla se dice que está en segunda forma normal (2FN) si está en 1FN y cada atributo no clave es completamente dependiente de la clave primaria.

Un atributo se dice que es completamente dependiente de la clave primaria si está al lado derecho de la dependencia funcional (DF), por lo cual, el lado izquierdo es la clave primaria o algo que se pueda derivar de la clave primaria utilizando transitividad de DF's.

La segunda forma normal nos dice que todo atributo debe necesariamente depender de la clave primaria completa y no sólo en parte de ésta; en el caso que la clave primaria estuviera compuesta por más de un atributo.

Analizando las relaciones para el sistema se afirma que todas las relaciones se encuentran en primera forma normal, y que además cumplen con las características específicas en la segunda forma normal.

8.2.11.3 Tercera Forma Normal (3FN)

Una tabla se dice que está en tercera forma normal (3FN) si está en 2FN y todos los atributos no claves dependen de manera no transitiva de la clave primaria.

Una relación se dice que tiene dependencia transitiva si existe un atributo no clave que depende de otro atributo no clave y éste a su vez depende de otro atributo que corresponde a la clave primaria.

En el caso del sistema de control de beneficios, las tablas que fueron presentadas anteriormente están normalizadas. Cabe destacar que aunque

esté realizado el proceso de normalización, hubo que crear ciertas tablas como la tabla boleta y la tabla escolar en la que existirá una duplicación de datos. Esto sucede porque, en el caso de la tabla “boleta” el que se realiza la atención médica puede ser el funcionario o la carga familiar, si se trata del mismo funcionario, en este caso el titular y el paciente será el mismo funcionario. Lo mismo ocurre con la tabla “escolar” donde el beneficiario puede ser el mismo funcionario. Dichos campos no pueden quedar nulos porque esto dificultaría la entrega de la información, ya que esto debe quedar especificado (impreso) en los formularios que arroje el sistema.

8.2.12 Validación del modelo contra transacciones de usuario

El objetivo de esta etapa es validar el modelo de datos lógico para asegurar que éste soporte las transacciones requeridas por los usuarios. Utilizando el diagrama E-R, diseño conceptual y las claves se realizará esta operación. El resultado será el mapa de transacciones del modelo. El cual muestra los caminos que satisfacen los requerimientos. En caso que no se pueda realizar, se modificará el modelo de datos. A continuación en la tabla N ° 11 se muestra el listado de transacciones que satisfacen los requerimientos del usuario.

A continuación se muestra el mapa de transacciones del modelo

Tabla N ° 11: Listado de transacciones

T()	Lista de Transacciones	Colores
T(1)	Generar planilla de pago mensual	
T(2)	Generar planilla de pago de beneficios escolares	
T(3)	Imprimir Informe de cargas familiares por funcionario	
T(4)	Imprimir informe de funcionarios por parámetros (*)	
T(5)	Imprimir informe de funcionarios por fecha de nacimiento	
T(6)	Imprimir informe de cargas familiares por fecha de nacimiento	
T(7)	Ingreso de atenciones médicas	
T(8)	Ingreso de beneficios escolares	

(*) En este caso se menciona una sola transacción, pero que corresponde a imprimir funcionarios por los parámetros de Calidad, Dependencia y Salud.

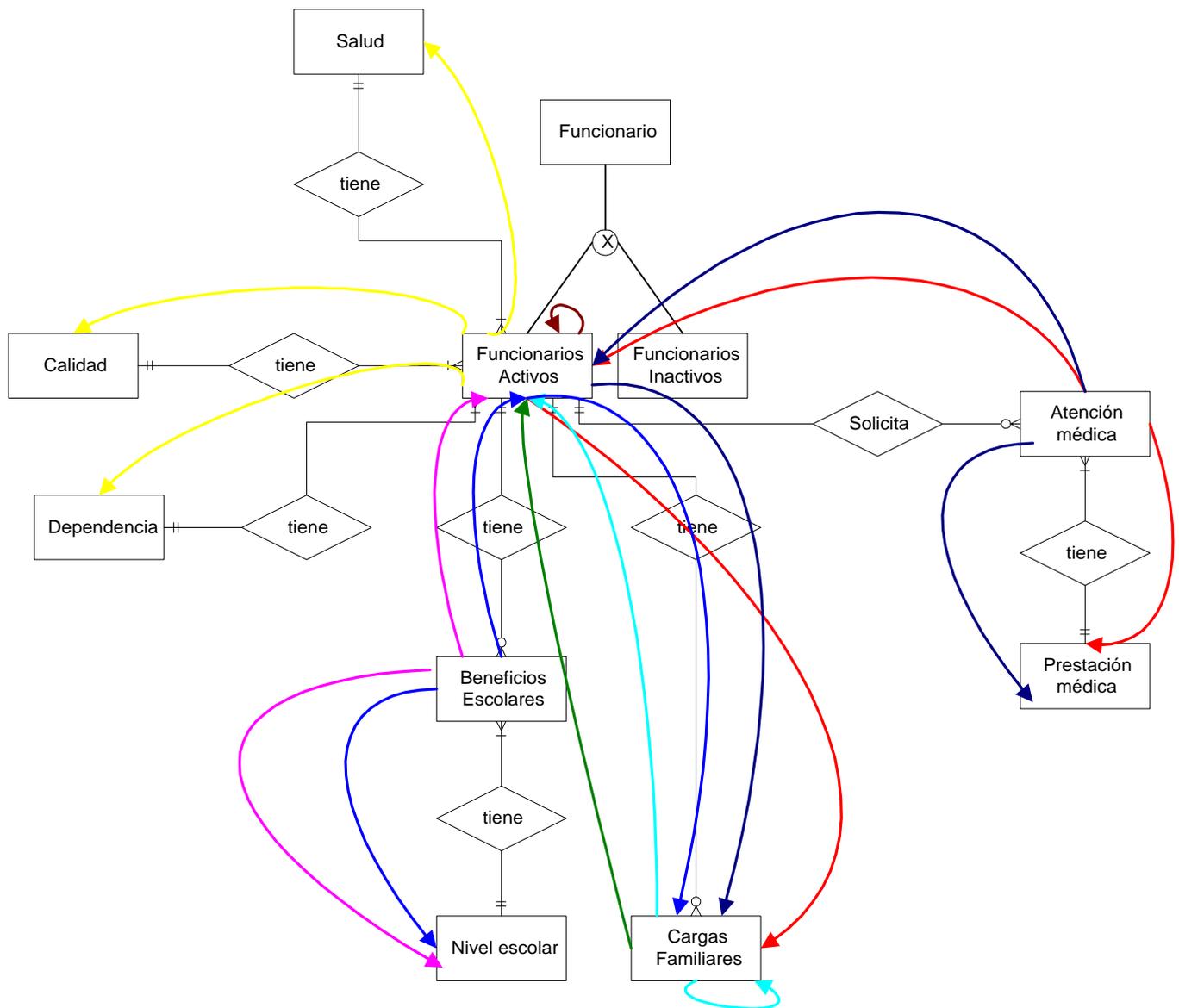


Figura N ° 6. Mapa Transaccional

8.2.13 Diagrama entidad relación del diseño lógico

En la figura N^o 7 se presenta el modelo ER del Diseño Lógico. A este modelo no se le aplicaron mayores cambios. En este nuevo modelo se identifican la existencia de las relaciones.

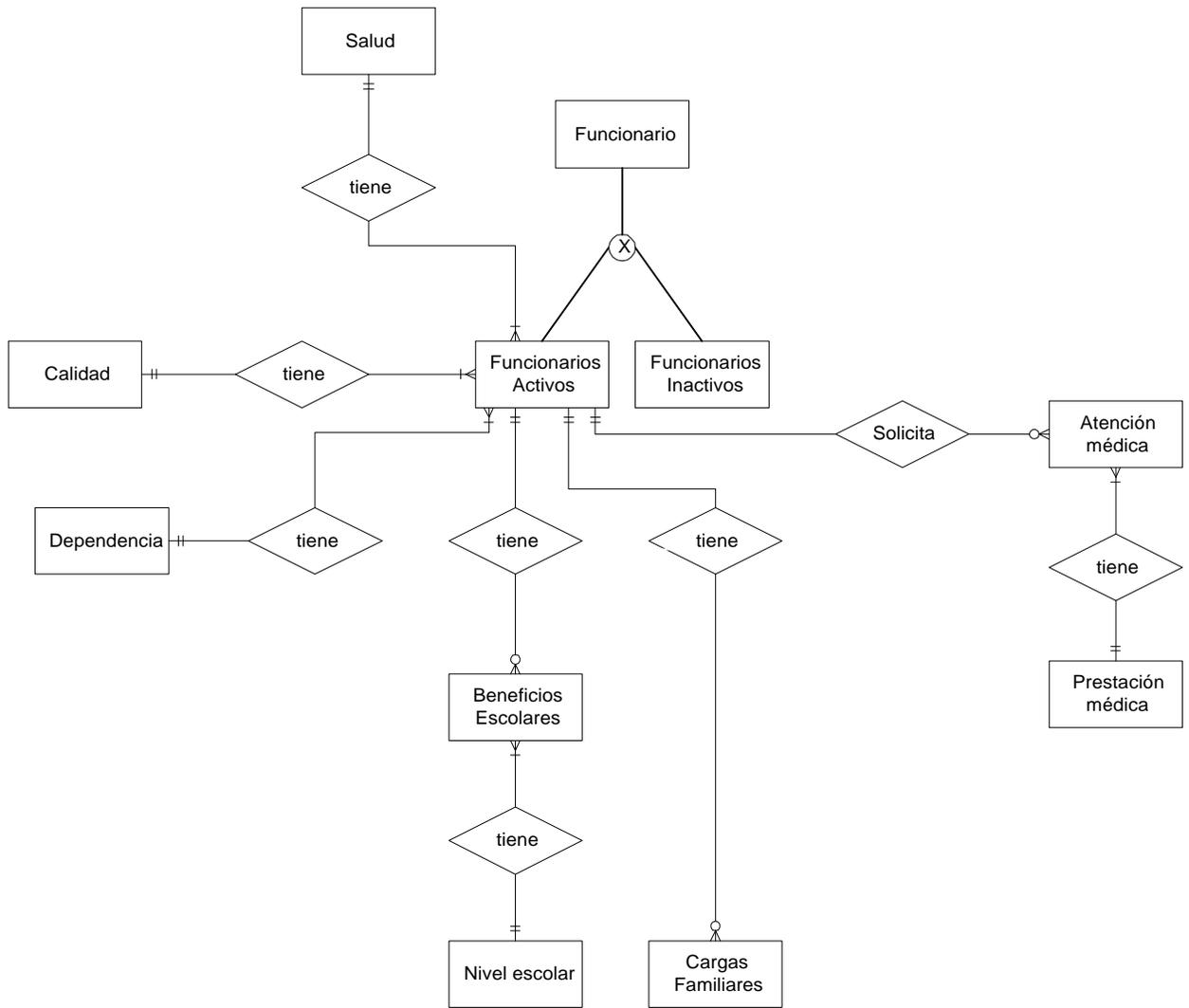


Figura N 07. Modelo ER l3gico

8.2.14 Definición de las restricciones de integridad

En esta etapa del diseño se definen todas las restricciones de integridad, para proteger la base de datos de las inconsistencias. Esta definición sólo se realiza en un alto nivel del diseño no tomando en cuenta un DBMS específico y solo identificándolas sin describir como se logran.

Existen cinco tipos de restricciones de integridad las cuales serán descritas a continuación.

8.2.14.1 Datos requeridos

Algunos atributos deben tener siempre valores válidos, es decir que no deben ser valores nulos (Null).

8.2.14.2 Restricciones de dominios de los atributos

Todos los atributos de las relaciones tienen un dominio. Esto se refiere a que poseen un conjunto de valores válidos para si.

8.2.14.3 Integridad de entidades

La claves primarias de las entidades no deben ser nulas.

8.2.14.4 Integridad Referencial

La integridad referencial tiene como objetivo asegurar que cada instancia de la entidad hijo posea una clave foránea, la cual tiene un valor que hace referencia a la entidad padre.

En el caso del sistema de control de beneficios, no existen valores nulos para ninguna de las tablas antes mencionadas, todos los datos son requeridos y por ende las claves primarias tampoco son nulas. En cuanto a la integridad referencial, cada entidad hija posee una clave foránea que corresponde a la entidad padre, en caso de eliminar datos de la entidad padre la entidad hija no lo permite.

A continuación se muestran las restricciones de integridad aplicadas a las relaciones hijas del modelo.

Tabla N ° 12. Integridad Referencial

Nombre Entidad	Clave Foránea	Referencia a	On delete	On Update
Funcionario	Calidad_funcionario	Calidad	Restrict	Restrict
	Salud_funcionario	Salud	Restrict	Restrict
	Dependencia_funcionario	Dependencia	Restrict	Restrict
Beneficios Escolares	Nivel_escolar	Nivel	Restrict	Restrict
	Funcionario_escolar	Funcionario	Restrict	Restrict
	Beneficiario_escolar	Cargas o Funcionario	Restrict	Restrict
Cargas Familiares	Funcionario_carga	Funcionario	Restrict	Restrict
Atención médica	Funcionario_boleta	Funcionario	Restrict	Restrict
	Prestacion_boleta	Prestaciones Médicas	Restrict	Restrict

Observación: La restricción Restrict, significa que al tratar de eliminar o actualizar una instancia del padre, la clave foránea en el hijo no lo permite.

8.3 Diseño de Base de Datos Físico

Una vez realizada las etapas conceptual y lógica, el siguiente paso es el diseño físico de la base de datos. En la fase anterior a ésta, el diseño lógico analiza los detalles en forma independiente de la implementación. Esto es independiente de un DBMS específico o de programas de aplicación, pero sí independiente el modelo de datos. Es decir, el diseño lógico se preocupa del “que” a diferencia del diseño Físico que se preocupa del cómo.

El objetivo principal del diseño físico es lograr una implementación del esquema lógico lo más eficiente posible. Para ello se lleva a cabo la implementación de estructuras, de restricciones, de índices, de relaciones y definición de reglas de acceso.

En esta etapa de la metodología se realizan las siguientes actividades que se describen a continuación.

8.3.1 Traducir el modelo lógico al DBMS seleccionado

- Diseño de las relaciones bases para el DBMS correspondiente.
- Diseño de las restricciones de la organización.

8.3.2 Diseñar la representación física

- Análisis de las transacciones.
- Selección de la organización de archivos.
- Elección de índices secundarios.
- Considerar la introducción de redundancia controlada (Denormalización).
- Estimar requerimientos de espacio en disco.

8.3.3 Diseñar mecanismos de seguridad.

- Diseñar vistas de usuario.
- Diseñar reglas de acceso.

Como resultado de esta fase se obtiene una base de datos libre de inconsistencia y optimizada para realizar todas las transacciones ya sean de ingreso, eliminación, actualización o consulta de los datos.

8.3.4 Transformación del modelo de datos lógico global para un DBMS seleccionado

El objetivo es producir un esquema relacional del modelo de datos lógico global.

8.3.4.1 Diseño de relaciones bases para un DBMS correspondiente

Aquí se debe identificar los índices, los cuales son implementados como archivos auxiliares para las tablas, y así acelerar el tiempo de acceso a los registros de éstas. Pueden estar compuestos por uno o más atributos. Existen dos tipos de índices; primarios y secundarios. Los primarios corresponden a las claves primarias y los secundarios a otras columnas de la tabla.

Para este proyecto se definen las claves primarias como PRIMARY KEY y las secundarias como exclusivas. En SQL Server 7.0 se definen como UNIQUE. Algunos ejemplos se muestran en la tabla N ° 13.

Tabla N ° 13: Ejemplos de claves primarias y secundarias.

Tabla	Clave_primaria	Clave_secundaria
Calidad	Codigo_calidad	Nombre_calidad
Dependencia	Codigo_dependencia	Nombre_dependencia
Salud	Codigo_salud	Nombre_salud

8.3.4.2 Diseño de las restricciones de la empresa para un DBMS específico

Las restricciones de la empresa representan, las reglas que gobiernan a la empresa en el mundo real, que son representadas en las actualizaciones.

Estas restricciones varían dependiendo del DBMS a utilizar, ya que cada gestor de base de datos, ofrece diferentes opciones para representar este tipo de restricciones.

En este capítulo, se definieron reglas de la empresa, obtenidas del diseño lógico. Como también las restricciones de integridad, de dominio y de referencia, que éstas de por sí son reglas de la empresa.

8.3.5 Diseño de la Representación Física

En este paso el objetivo que se pretende alcanzar es la determinación del tipo de organización de archivos y sus métodos de acceso que se utilizarán en el almacenamiento de las relaciones bases, es decir, la forma en que cada relación y ocurrencia será mantenida en un medio de almacenamiento físico.

Para obtener un eficiente uso en las estructuras, se deben considerar los siguientes factores:

- Transacciones, donde se refiere al número de transacciones que pueden ser procesadas en un intervalo de tiempo.

- Tiempo de respuesta, este es el lapso de tiempo que se ocupa en completar una transacción, determinada.
- Almacenamiento en disco, se refiere a la cantidad de espacio en disco requerido para almacenar la base de datos. Otro aspecto a considerar es la instalación del gestor de base de datos. En el caso del sistema de Control de beneficios, el gestor de base de datos será SQL Server 7.0 donde se requiere espacio para: el software SQL Server 7.0, las bases de datos, copia de seguridad entre otras. Los administradores del sistema son los encargados de decidir, dónde se almacenan estos recursos y el espacio que se les debe asignar.

8.3.5.1 Análisis de las transacciones

El entender la funcionalidad de las transacciones cuando se ejecutan sobre la base de datos y analizar la importancia de las transacciones, son los principales objetivos que abarca esta etapa.

Para esto, también es necesario tener conocimiento de las consultas que se ejecutarán en la base de datos, incluidas el manejo de información cualitativa y cuantitativamente. Para cada transacción se debería determinar lo siguiente:

- La frecuencia esperada de cada transacción que se ejecutará.
- Las relaciones y atributos accedidos por transacción y los tipos de accesos: si es, consultas, inserción, actualización o eliminación. Tomando en cuenta la actualización, si los atributos son parte de índices no agrupados o son claves primarias.
- Los atributos usados en algún predicado de la sentencia SQL. Se debe verificar si los predicados involucran entidades "Padres", rangos de búsquedas o búsquedas exactas.
- Para una consulta, verificar si los atributos que participan en un criterio de búsqueda, involucran más de dos tablas.

Siguiendo la metodología, se debe crear un mapa donde se exprese el uso de las transacciones, que permita visualizar cuantas relaciones son

afectadas por cada transacción. Es poco probable la realización del mapa de transacciones, por cuanto es difícil estimar el uso de ellas.

Sin embargo, es posible realizar un análisis aproximado de las transacciones por accesos, frecuencia y ejecuciones.

A continuación, en la tabla N ° 14, se muestra el análisis de transacciones, donde se identifican los promedios de acceso, periodicidad y N° de ejecución por período. Cabe destacar que estos cálculos son estimativos, donde los periodos de acceso son relativos.

Tabla N ° 14. Análisis de transacciones.

Transacciones	Promedio de acceso	Periodicidad	N° Ejecución por período	Atributos
T(1)	Bajo	Mensual	2	Todos
T(2)	Bajo	Anual	2	Todos
T(3)	Alto	Diario	2	Todos
T(4)	Bajo	Diario	2	Todos
T(5)	Bajo	Diario	2	Todos
T(6)	Alto	Diario	2	Todos

T(7)	Alto	Diario	2	Todos
T(8)	Alto	Anual	4	Todos

8.3.5.2 Elección de índices secundarios

El gestor de base de datos SQL Server crea automáticamente índices primarios. Estos son las claves primarias definidas en el diseño.

Los índices secundarios proveen un mecanismo para su creación a partir de las claves alternas, la cual puede ser utilizada para ubicar datos en forma más eficiente.

8.3.5.3 Consideraciones en la Introducción de Redundancia Controlada (Denormalización).

Recordando el objetivo de la técnica de normalización, que era reducir, y no necesariamente eliminar, la redundancia de datos. En algunos casos se

puede permitir una redundancia controlada de los datos, sin ir en desmedro del diseño normalizado, a este refinamiento se le denomina Denormalización.

Es útil aplicar la denormalización, si lo que se quiere es optimizar el acceso a la información, reduciendo el número de tablas, pero se debe planificar cuidadosamente.

Una base de datos normalizada en exceso puede sufrir algunas veces de problemas de rendimiento. Cuando sucede esto, generalmente se debe denormalizar el modelo de datos, volviendo a refundir dos o más elementos para mejorar el rendimiento o reducir el número de uniones de tablas en ciertos tipos de consultas.

Esto significa que es importante, no sólo comprender la estructura de los datos y las relaciones entre los elementos, sino que también hay que entender cómo se accede y actualizan los datos.

Debido al bajo número de tablas que participan en el sistema de “Control de beneficios”, y a los buenos resultados de la normalización, no se aplicará este procedimiento.

8.3.5.4 Estimación del Espacio Requerido en Disco.

Aquí es importante estimar el espacio requerido en el almacenamiento secundario que se necesitará para albergar la base de datos.

En el departamento de bienestar de la municipalidad se utilizará un servidor dedicado para sostener el Gestor y sus respectivas base de datos, lo cual permite el almacenamiento de la base de datos del Sistema Control de beneficios sea perfectamente factible. No será necesario realizar una estimación del espacio requerido en forma detallada. Además de considerar que la proyección de crecimiento de la base de datos es relativamente menor, ya que el sistema no manejará volúmenes masivos de datos, sino más bien, está más orientado a la mantención y administración de una información más o menos estable, lo que si puede variar mes a mes es la cantidad de atenciones médicas registradas.

8.3.6 Diseño de Mecanismos de Seguridad.

El objetivo de esta etapa es diseñar los mecanismos para garantizar la seguridad de la base de datos.

Una base de datos representa esencialmente los recursos corporativos de la empresa, y la seguridad es un recurso muy importante que debe ser analizado. La idea de este procedimiento es decidir como se van a implementar los mecanismos de seguridad. Sin embargo el diseño de mecanismos de seguridad está muy ligado a lo que pueda ofrecer por un lado, el gestor de base de datos, y por otro lado, la seguridad que pueda ofrecer la plataforma computacional donde se montará el sistema. Además de la seguridad que debe ofrecer, por sí mismo el sistema que está siendo desarrollado.

Las principales actividades en esta etapa son:

- Diseño de Vistas de Usuario
- Diseño de reglas de Accesos.

8.3.6.1 Diseño de Vistas de Usuario.

El objetivo de esta etapa es crear vistas para que los usuarios hagan consultas a la base de datos, basadas en el diseño lógico. Esto se hace para ver los privilegios de acceso que tendrá cada usuario con respecto a las transacciones dentro del sistema.

En este caso debido a que todos los funcionarios del departamento de bienestar realizan la misma labor, no se crearon vistas: Todos ellos tienen acceso a los mismos datos y pueden realizar todas las transacciones. Cabe señalar que en este caso el jefe del departamento de bienestar de la Municipalidad de Puerto Montt utiliza el sistema. En el caso de la ausencia de este, otro funcionario puede realizar la misma tarea. El funcionario debe estar autorizado por éste para realizar la labor del jefe.

8.3.6.2 Diseño de Reglas de Acceso.

El objetivo de esta etapa es diseñar reglas de acceso para las relaciones y vistas de usuarios.

Como reglas de acceso existen dos niveles de seguridad:

- Sistema operativo Windows: En este se crean los usuarios que tienen acceso a los datos del servidor.
- El sistema tiene su propia característica de seguridad que se explicarán a continuación.

8.3.6.3 Control de usuarios

El control de usuarios restringe el acceso al sistema sólo a personas autorizadas. Todos los usuarios pueden modificar su propia contraseña. Solo la persona que administra el sistema puede crear o eliminar a otros usuarios.

Para poder realizar esto se creó una tabla que contiene el nombre, y un código, también ésta almacena la clave del usuario. Para poblar esta tabla el administrador del sistema es el que ingresa estos datos.

Cabe destacar que existe sólo un nivel de usuario. Aquí los privilegios están descartados, ya que todos los usuarios del sistema pertenecientes al departamento de bienestar realizan la misma actividad.

9. SELECCIONAR DBMS

Esta etapa involucra seleccionar el DBMS (DataBase Management System o Sistema Administrador de bases de datos) apropiado para el sistema. Si no existe el DBMS, se recomienda realizar esta selección entre las fases de diseño conceptual y diseño lógico. Lo importante es asegurar que cumpla con funcionalidades de seguridad, fácil implementación y reestructuración, y reglas de integridad. Cabe destacar que esta fase de la metodología es opcional.

Esta etapa no se desarrollará dado que la Municipalidad de Puerto Montt utiliza para sus sistemas el SQL Server 7.0 como motor de base de datos.

10. DISEÑO DE LA APLICACIÓN

En este capítulo corresponde analizar y explicar las fases de la aplicación.

Después de analizar la problemática, definir la solución y realizar el proceso de modelamiento de datos, se establecerán los patrones de diseño de la aplicación “Cliente”, que en definitiva permitirá establecer un puente de comunicación entre el usuario y la base de datos. El desarrollo de la aplicación cliente, facilitará al usuario la obtención, búsqueda, actualización y la generación de reportes de información que el usuario necesita conocer.

En esta etapa resaltan tres puntos clave, los cuales son:

- Diseño de la interfaz.
- Diseño procedimental.
- Desarrollo del software.

Estos puntos serán explicados en detalle más adelante.

10.1 Definición de los módulos del sistema

Se definen conceptualmente los módulos principales del sistema, éstos módulos corresponden a la funcionalidad que entregará el sistema a los usuarios. Para definir de forma adecuada cuáles serán estos módulos es recomendable agrupar por tipo las principales funciones definidas durante la etapa de análisis del sistema.

- El primer módulo es el denominado archivo: Este contiene todas las tablas que son base para el resto de la información y son las primeras que se deben poblar.
- Módulo de funcionario: Este módulo posee el ingreso de todos los datos del funcionario y de sus cargas familiares.
- Módulo de búsquedas: Este módulo permite que el funcionario pueda realizar búsquedas de datos por diferentes parámetros, ya sea calidad, dependencia entre otras.
- Módulo de cancelación: Este módulo corresponde a generar la planilla de pago mensual y la planilla de pago de beneficios de escolaridad.

- Otros: En este módulo están todos los procesos que no están directamente relacionados con los datos del funcionario, pero que de igual manera son de gran importancia en el sistema.

A continuación se mostrará un el esquema de los módulos del sistema

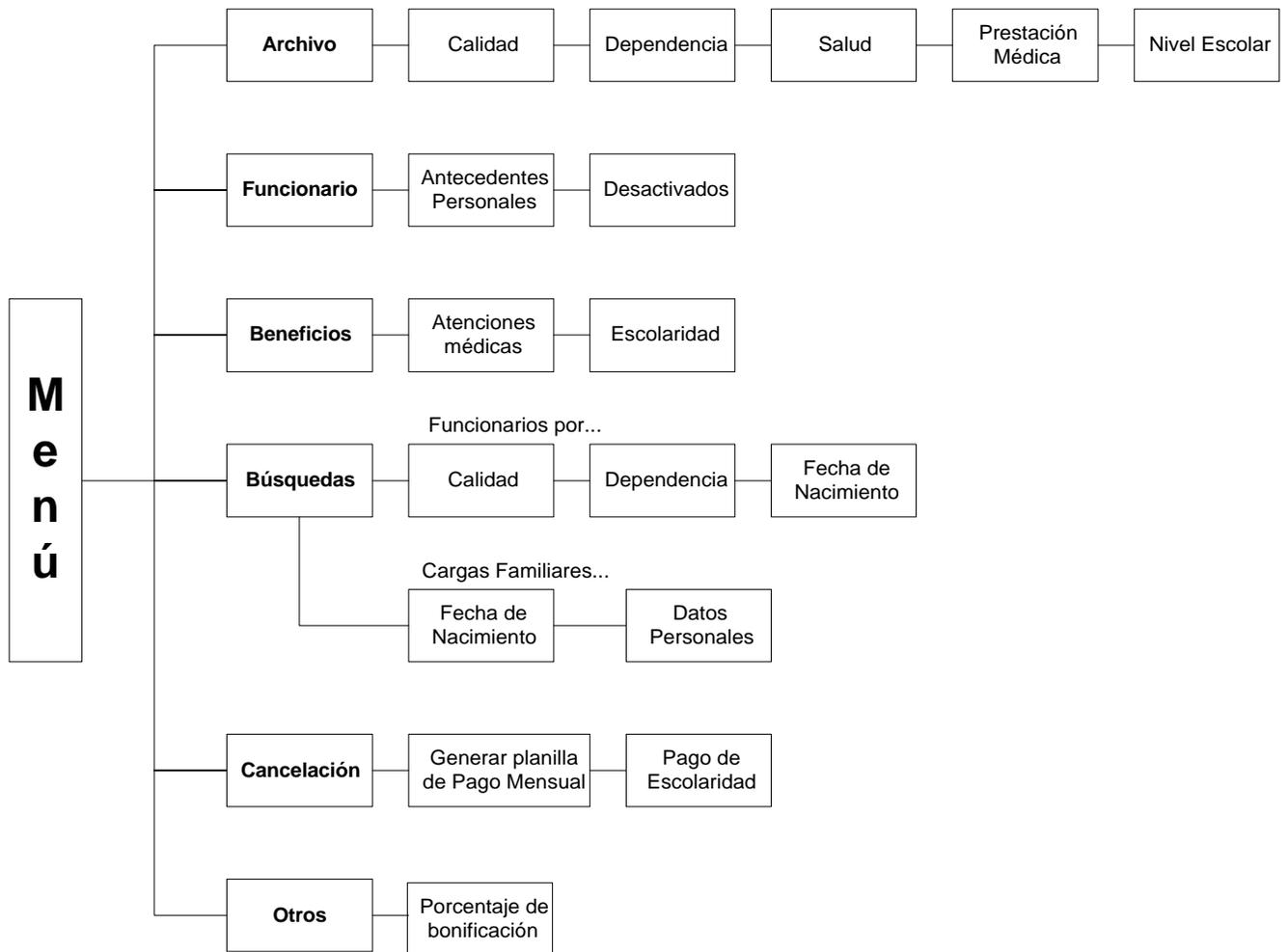


Figura N ° 8. Esquema del menú

10.2 Diseño de la interfaz

A medida que ha avanzado el tiempo, la tecnología y el hombre han convivido entre lo tangible e intangible, computacionalmente hablando, es común ver ahora como los computadores están presentes, en un gran número, en nuestra vida cotidiana. Debido a esto, se hace cada vez más importante optimizar la interfaz Hombre-Máquina, de modo que esta relación sea amistosa y eficiente.

La idea fundamental en el concepto de interfaz es el de mediación, entre el hombre y máquina. La interfaz es el nexo, lo que facilita la comunicación, la interacción entre dos sistemas de naturalezas distintas. Esto implica analizar la interfaz, como un sistema de traducción, ya que estos dos sistemas se expresan de forma diferente, por un lado el conocido lenguaje que utilizamos los seres humanos y el sistema Binario que emplean las computadoras o denominado "Código de máquina".

Como se había mencionado antes, la interfaz es uno de los pasos más importantes en la realización del proyecto, ya que corresponde a la interacción del usuario con el sistema. Para realizar esto se utilizaron ciertos criterios de diseño de interfaz propuestas por Eduardo Mercovich (fundador y líder de gaiasur consultora desde 1994 y se especializa en diseño de interfaces y

usabilidad). Para complementar estos criterios, se utilizaron algunos pasos que están definidos en la metodología “Diseño de interfaz de Usuario” [Lewis y Reiman][1993].

Una interfaz que está debidamente diseñada es de fácil aprendizaje y de utilizar. Cabe señalar que se trata de lograr que sea el software el que se adapte a las necesidades de los usuarios, de ser así, permitirá a que los usuarios que ejecutarán el sistema, se encuentren familiarizados con los menús, tipos de mensajes, colores, controles de mandatos y presentación de informes agilizando aún más el proceso de aprendizaje de este nuevo sistema.

Dentro de las interfaces de usuarios, básicamente se distinguen dos tipos:

- Interfaz de Hardware, al nivel de dispositivos utilizados para ingresar, procesar y entregar los datos: teclado, mouse, monitor y capturador entre otras.
- Interfaz de Software, destinada a entregar información acerca de los procesos y herramientas de control, a través de lo que el usuario visualiza en pantalla.

Al momento de diseñar la interfaz se deben considerar las habilidades cognitivas y de percepción de los usuarios, y adoptar el sistema a ellas.

Un aspecto fundamental que debe resolver un buen diseño de interfaz, es la reducción de la dependencia de las personas de su memoria o retención de las cosas, no forzándolas a recordar cosas innecesariamente o repetir operaciones ya realizadas.

A continuación se explicarán algunas de las consideraciones que se plantean para el desarrollo de interfaz de usuario.

10.2.1 Usabilidad

Se define usabilidad de un sistema o herramienta como una medida de su utilidad, facilidad de uso, facilidad de aprendizaje y apreciación para una tarea, un usuario y un contexto dado.

El peso relativo de cada una de estas medias está relacionado con el usuario, la tarea y el contexto. Por ejemplo, la facilidad de aprendizaje puede ser crucial para un producto y poco importante en otro.

Si bien no está incluida en la definición usual de usabilidad, se está comenzando a hablar de la evolución de estos factores a lo largo del tiempo. Por ejemplo, cómo una interfaz puede adaptarse al crecimiento en eficiencia y conocimientos de un usuario.

En el caso del sistema de control de beneficios, cabe señalar que para el usuario es importante que el sistema sea fácil de utilizar, ya que como anteriormente no existía un sistema, puede, en algún momento, provocar algún problema si el diseño de la interfaz no es “amigable”.

10.2.2 Utilidad

La utilidad es la capacidad que tiene una herramienta para ayudar a cumplir tareas específicas. Aunque esta afirmación parece obvia, es importante observar que una herramienta que es muy usable para una tarea, puede ser muy poco usable para otra, aún incluso si se trata de una tarea similar pero no idéntica.

Es casi obvio que este nuevo sistema prestará una utilidad, sobre todo si se trata de automatizar procesos que hasta el momento, si bien es cierto no son complejos, pero sí extensos.

10.2.3 Facilidad de uso

La facilidad de uso está en relación directa con la eficiencia o efectividad, medida como velocidad o cantidad de posibles errores. Una herramienta muy fácil de usar permitirá a su usuario efectuar más operaciones por unidad de tiempo (o menor tiempo para la misma operación) y disminuirá la probabilidad de que ocurran errores.

Ninguna herramienta o sistema es perfecto, pero una alta probabilidad de error puede llegar incluso a derivar en una imposibilidad de uso por falta de calificación, según cuáles sean los criterios para evaluar la herramienta o sistema.

Se debe tener en claro que la facilidad de uso no debe confundirse con la facilidad de aprendizaje.

En este caso y como se mencionaba anteriormente el objetivo del sistema es, que además de automatizar los procesos manuales, también sea fácil de usar. Este objetivo es importante ya que los procesos extensos se deben reducir con la utilización de un nuevo sistema. Esto va junto al concepto de eficiencia donde la información que deba proporcionar el sistema debe estar libre de errores.

10.2.4 Facilidad de aprendizaje

La facilidad de aprendizaje es una medida del tiempo requerido para trabajar con cierto grado de eficiencia en el uso de la herramienta, y alcanzar un cierto grado de retención de estos conocimientos luego de cierto tiempo de no usar el sistema. Si bien la facilidad de aprendizaje suele tener una relación

directa con la usabilidad, estrictamente hablando esto no necesariamente es así. La facilidad de aprendizaje debería ser una medida relativa, ya que hay sistemas muy complejos que no pueden ser aprendidos rápidamente. Por lo tanto, lo importante es comparar entre varias posibles interfaces y ver cuál es la que requiere menos tiempo y/o queda mejor retenida. Si la versión siguiente, sin aumento en la complejidad del servicio brindado tomara 8 meses de aprendizaje, ahora sí estaríamos frente a un problema de usabilidad.

Cabe destacar aquí que la interfaz confeccionada para este sistema es sencilla, tal como lo requirieron los usuarios, lo que permite su facilidad de uso. Al ser fácil de usar, resulta sencillo y fácil de aprender. Con esto el usuario podrá realizar un uso eficiente de la herramienta entregada, incluso si este deja de usarlo por algún tiempo. El tiempo de recordar el manejo y la utilización del sistema será mínimo.

10.2.5 Apreciación

Es una medida de las percepciones, opiniones, sentimientos y actitudes generadas en el usuario, por la herramienta o sistema. La apreciación es una medida menos objetiva que las anteriores, pero sin embargo, no menos importante. Lo importante de esta medida no es tenerla en forma absoluta sino, otra vez, compararla o analizarla en forma relativa. Esta comparación puede ser contra la competencia, contra la versión anterior del mismo producto, contra otras posibilidades que se estén tomando en cuenta.

Otro punto importante respecto de la apreciación es tratar de analizar hasta donde “tiñe” el resto de las medidas. Un usuario al que no le “gusta” una interfaz puede generar más errores, o tardar más en aprenderla. Debemos aprender a separar las medidas estrictas de las que puedan tener desviaciones debidas a una apreciación negativa.

Es difícil lograr que todos los usuarios estén de acuerdo con la interfaz del sistema, por eso esta debe agradar al usuario aunque no se garantiza que para todos los usuarios la interfaz sea la adecuada. Por eso la metodología presentada se acomoda muy bien a los requerimientos de los usuarios ya que su desarrollo será realizado tomando en cuenta absolutamente la opinión del usuario.

Según Lewis y Reiman, algunos factores humanos a considerar son los siguientes:

- Velocidad de aprendizaje. Se pretende, como se mencionaba anteriormente, que la persona aprenda a utilizar el sistema lo más pronto posible. También hay que considerar la disposición del usuario frente a un nuevo sistema.
- Velocidad de respuesta. Referente al tiempo necesario para realizar una operación en el sistema.
- Tasa de errores. Porcentaje de errores que comete el usuario.
- Retención. Cuanto recuerda el usuario sobre el uso del sistema durante un período de tiempo.
- Satisfacción. Se refiere a que el usuario esté a gusto con el sistema.

Además de estos factores existen otros a considerar :

- Características físicas. Cada persona tiene diferentes características físicas. Por lo que algunos usuarios prefieren utilizar el mouse que el teclado.
- Ambiente. Hay que considerar el lugar donde se implementará el sistema. Cada interfaz debe adecuarse al lugar.
- Visibilidad. Se debe tener en cuenta la iluminación del lugar. Donde el lugar de trabajo debe presentar condiciones idóneas para trabajar en un computador.
- Personalidad. De acuerdo a la edad, nivel socio-económico, etc.
- Cultura. Considerar el alcance del sistema, si se va a implementar a nivel internacional.

10.3 Procedimientos para el diseño de interfaz

En el proceso de diseño de una interfaz se pueden distinguir cuatro fases o etapas fundamentales:

- Recolección y análisis de información del usuario.
- Diseñar la interfaz de usuario.
- Construir la interfaz de usuario.
- Validar la Interfaz de usuario.

10.3.1 Recolección y análisis de información del usuario

Se debe concretar a través de técnicas de toma de requerimientos, que tipos de usuarios del programa, que tareas van a realizar los usuarios y cómo las van a realizar, que exigen los usuarios del sistema, en que entornos se desenvuelven (físico, social, cultural).

Los usuarios que emplearán el sistema, tienen conocimientos de interfaces gráficas, como es la de Windows. Por lo que, requieren que el

sistema presente las mismas funcionalidades, en lo netamente gráfico, que este Sistema Operativo.

El entorno de trabajo donde se desenvuelven los usuarios del sistema, satisface todas las expectativas esperadas para que la persona desarrolle sus labores mediante sistemas computacionales.

10.3.2 Diseñar la interfaz de usuario

Es importante dedicar tiempo y recursos a esta fase, antes de entrar en la codificación del sistema. En esta fase se definen los objetivos de usabilidad del sistema, las tareas del usuario, los objetivos y acciones de la interfaz, los iconos, vistas y representaciones visuales de los objetos, los menús de los objetos y ventanas. Todos los elementos visuales se pueden hacer primero a mano y luego refinar con las herramientas adecuadas.

La interfaz del sistema de control de beneficios será similar a la interfaz gráfica de Windows. Al momento de diseñar se deben tomar las siguientes consideraciones:

10.3.3 Referentes a la presentación de la información

En este caso se trató de diseñar una interfaz lo más sencilla posible sin colocar demasiados objetos. Se colocaron botones que indican procesos simples (éstos están especificados en la tabla N° 15), para la retención de los usuarios y para que el proceso de aprendizaje sea fácil.

10.3.4 Referentes al Análisis del Color

Es probable que este elemento de la interfaz es el que con más frecuencia es mal utilizado. El color comunica información, no es sólo un elemento decorativo. Se utilizarán combinaciones de colores adecuadas, ya que se debe considerar que el usuario estará por un período considerable de tiempo, en interacción con el sistema.

Para los informes se utilizó la letra Times New Roman de tamaño siete u ocho dependiendo de la cantidad de información que debería entregar el informe.

En los formularios, después de las consultas realizadas a los usuarios, se concluyó en utilizar los colores que vienen definidos en el Visual Basic 6.0 y la letra se cambió a Negrita Comics Sans.

A continuación se describirán las funciones de cada uno de los botones

Tabla N ° 15. Funciones de los botones del sistema

Botones	Descripción
Nuevo	Limpia la interfaz para ingresar un nuevo registro.
Grabar	Graba la información ingresada o modificada en la interfaz.
Eliminar	Borra permanentemente un registro.
Imprimir	Presenta en pantalla el informe con el contenido de todos los registros existentes, dando la opción de imprimirlo.

10.3.5 Construcción de la Interfaz de Usuario

Es saludable primero, realizar un prototipo previo, como una primera versión del programa que se realice rápidamente y permita visualizar el producto para poder probarlo antes de codificarlo definitivamente.

A continuación se presentará un esquema general, con las ventanas, controles y menús del Sistema Control de Inventario, que son el resultado del análisis de diseño de la interfaz.

10.3.6 Estandarización de pantallas.

En esta etapa se estandarizó los botones, el tamaño y el tipo de letra a utilizar en los formularios y los informes.

La primera interfaz que se utilizará es la seguridad. Aquí el usuario debe ingresar la clave de acceso. No existen niveles de seguridad ya que todos los usuarios del sistema cumplen las mismas funciones.

La segunda interfaz es el menú, la cual contiene todos los módulos anteriormente explicados las opciones se muestran en forma de menú emergentes.

A continuación se mostrarán las interfaces más importantes del sistema.



Figura N ° 9. Pantalla de seguridad.

The screenshot shows a window titled "Calidad" with a standard Windows-style title bar. Inside the window, there are two input fields: "Código" and "Nombre". Below these fields is a table with two columns: "Código" and "Nombre". The table contains four rows of data. At the bottom of the window, there is a row of buttons: "Nuevo", "Grabar", "Eliminar", "Imprimir", and four navigation arrows (left, right, double left, double right).

	Código	Nombre
▶	1.2	uno-dos
	1.3	uno-tres
	1.4	uno-cuatro
	1.5	uno-cinco

Figura N ° 10 Formulario de ingreso de calidad.

En este formulario se ingresa la calidad con su respectivo código. El mismo diseño tiene el formulario de ingreso de la dependencia, salud, nivel de escolaridad y prestaciones médicas.

Datos Personales Fecha: 21/04/2003

Antecedentes Personales

Rut

Apellido Paterno Apellido Materno Nombres

Fecha de Nacimiento

Fecha de Inicio al sistema

Calidad Dependencia Salud

Remuneración Imponible Cupo disponible para reembolso

Cargas Familiares

Rut	Nombre	Apellido Paterno	Apellido Materno	Fech. Nac.	Fech. Inicio	Fech. Termina
*						

Figura N ° 11. Formulario de ingreso de datos del funcionario.

Este es uno de los formularios más importantes. Es el que permite el ingreso de todos los funcionarios.

Registrar atenciones médicas

Número de boleta Fecha de Emisión Fecha: 14/04/2003

Rut Apellido Paterno Apellido Materno Nombres

Datos del paciente

Rut Apellido Paterno Apellido Materno Nombres

 =

Cód. Prestación Prestación Valor Prestación Valor Bonificación

Figura N ° 12 Registro de atenciones médicas.

The image shows a software window titled "Funcionarios por Calidad". At the top, there is a search section with a dropdown menu labeled "Ingresar Código", a text input field, and a "Buscar" button. Below the search section is a table with the following columns: Rut, Paterno, Materno, Nombre, Fecha.Nac, C, S, and D. The table is currently empty. At the bottom of the window, there are buttons for "Limpiar", "Imprimir", and navigation arrows (left and right).

Figura N ° 13 Búsqueda de funcionarios por calidad

En este Formulario se pueden buscar los funcionarios por un determinado parámetro. Los funcionarios aparecen en un listado dentro del mismo formulario.

Reportes

MUNICIPALIDAD DE PUERTO MONTE		PLANILLA DE CANCELACION REEMBOLSOS MEDICO			26/4/03	
SERVICIO DE BIENESTAR MUNICIPAL						
RUT		NOMBRE			TIPO DE SALUD	
10.822.153 - 4		BEYER HERNANDEZ MARIELA PRISCILA			no-dos	
N° BOLETA	PACIENTE	PRESTACION	FECHA ATENCION	SALDO ANTERIOR	VALOR BOLETA	TOTAL AFECTO A BONIFICACION
106	13.943.311	cuatro-dos	6/12/02	400.000	40.000	34.000
FIRMA FUNCIONARIO					TOTAL A PAGAR	54.000
RUT		NOMBRE			TIPO DE SALUD	
12.594.786 - 9		ARCEL MANSILLA MIGUEL ANTONIO			no-dos	
N° BOLETA	PACIENTE	PRESTACION	FECHA ATENCION	SALDO ANTERIOR	VALOR BOLETA	TOTAL AFECTO A BONIFICACION
103	13.466.983	cuatro-uno	4/12/02	70.000	30.000	27.000
FIRMA FUNCIONARIO					TOTAL A PAGAR	27.000
RUT		NOMBRE			TIPO DE SALUD	
13.169.718 - 3		BARRIOS GOMEZ PAULA XIMENA			no-dos	
N° BOLETA	PACIENTE	PRESTACION	FECHA ATENCION	SALDO ANTERIOR	VALOR BOLETA	TOTAL AFECTO A BONIFICACION
105	13.169.718	cuatro-tres	5/12/02	300.000	30.000	27.000
FIRMA FUNCIONARIO					TOTAL A PAGAR	27.000
RUT		NOMBRE			TIPO DE SALUD	
13.406.027 - 1		AGUILAR TENORIO MIGUEL ANGEL			no-dos	
N° BOLETA	PACIENTE	PRESTACION	FECHA ATENCION	SALDO ANTERIOR	VALOR BOLETA	TOTAL AFECTO A BONIFICACION
100	13.406.027	cuatro-uno	1/12/02	70.000	10.000	9.000
102	13.587.317	cuatro-dos	2/12/02	61.000	7.000	6.300
FIRMA FUNCIONARIO					TOTAL A PAGAR	15.300

Figura N ° 14. Este es el reporte de la planilla de pago mensual

Calidad 26/4/03

Código	Nombre
1.1	uno-uno
1.2	uno-dos
1.3	uno-tres
1.4	uno-cuatro
1.5	uno-cinco
1.6	fad
1.7	uno-siete
1.8	uno-ocho

Figura N ° 15 Reporte de calidad

<u>ANTECEDENTES PERSONALES</u>					26/4/03
RUT	NOMBRE				
13.406.027- 1	AGUILAR	TENORIO	MIGUEL ANGEL		
FECHA DE NACIMIENT	FECHA DE INICIO	CALIDAD	DEPENDENCIA	SALUD	
26/8/79	26/7/03	uno-uno	dos-tres	tre-dos	
	REMUNERACION	CUPO DISPONIBLE PARA REEMBOLSO			
	2.000.000	44.530			

RUT	NOMBRE	FECH. NAC.	FECH. I	FECH TER.	PAR.
13.587.517 - 1	AGUIRRE CARMONA	HECTOR GUSTAVO	24/11/78	15/2/03 1	24/11/96 D
15.298.662 - 0	ALARCON VELASQUEZ	RICHARD MOISES	21/5/76 1	15/2/03 1	21/5/94 1 D
13.471.327 - 5	ALBORNOZ OLEA	VIVIANA CAROLINA	25/5/84 1	15/2/03 1	25/5/02 1 D
13.788.246 - 9	ARACENA BRAVO	SEBASTIAN MATIAS	21/3/75 1	15/2/03 1	21/3/93 1 D
8.706.272 - 4	ARENAS SEPULVEDA	VICTORIA TAMARA	26/8/80 1	15/2/03 1	26/8/98 1 D

Figura N ° 16. Reporte de datos de un funcionario específico con sus respectivas cargas familiares

10.4 Diseño procedimental

El diseño procedimental transforma los elementos estructurales en una descripción procedimental del software. El diseño procedimental se realiza después de que se ha establecido la estructura del programa y de los datos. Aquí se define los algoritmos de procesamiento necesarios.

Para el caso del sistema de control de beneficios de los usuarios, después de analizar el Diagrama de Flujo de Datos en el capítulo N° 7 se definieron los procedimientos más importantes que se realizaron en el proyecto.

A continuación se detallan los procedimientos y las estructuras de los programas y de los datos para cada uno de los procesos antes presentados y servirá de base para la realización de la programación.

Proceso 1.0, Generar planilla de pago mensual

Como se pudo ver anteriormente este proceso está desglosado en tres subprocesos. Estos son:

Proceso 1.1 Calcular el porcentaje

Procedimiento calcula_ porcentaje_ bonificacion

Variables

valor_atención (numérico) ‘ *Valor atención médica*

valor_porcentaje (numérico) ‘ *Porcentaje de bonificación*

valor_calculado (numérico) ‘ *Valor de bonificación*

$\text{valor_calculado} = \text{valor atención} * (\text{valor_porcentaje}/100)$

Proceso 1.2 Comprobar cupo disponible

Procedimiento comprueba_cupo (Valor_calculado)

Variables

cupo_disponible ‘ *Cupo disponible para el reembolso*

If valor_calculado <= cupo_disponible entonces

 cupo_disponible=cupo_disponible-valor_calculado ‘ *El nuevo cupo disponible se guarda en un archivo*

else

 Imprimir por pantalla “El funcionario no posee cupo disponible para reembolso”

Proceso 1.3 Generar planilla de pago

Para este proceso la información se obtiene del registro de atenciones médicas realizadas en un mes, lo que significa que se realizarán consultas SQL para obtener la información de los archivos.

```
SELECT    nombre_funcionario,    nombre_paciente,    prestación_médica,
valor_boleta, valor_bonificación, fecha_ingreso
FROM boleta 'Tabla que posee el registro de las atenciones médicas del mes'
WHERE fecha_ingreso.mes= X AND
        Fecha_ingreso.año= X
```

Proceso 2.0 Generar Reportes

Los reportes que aquí se generan son realizados a través de consultas SQL. Por ejemplo si el usuario desea ver un listado de los funcionarios por una determinada calidad la consulta SQL a realizar sería sencillamente así:

```
SELECT * FROM funcionario
WHERE funcionario.calidad = X
```

Lo mismo ocurre cuando se desea ver listados por otros parámetros ya sea por dependencia, nombre, rut entre otros.

En el caso de realizar consultas para ver cargas familiares, se debe ir ligado al funcionario que es el apoderado de dichas cargas. En este caso se harán consultas a dos tablas, funcionario y carga.

La consulta quedaría de la siguiente manera:

```
SELECT rut_carga, nombre_carga, fecha de nacimiento...  
FROM cargas_familiares,funcionario  
WHERE funcionario.rut = X
```

Proceso 3.0 Generar planilla de beneficios de escolaridad

Similar al proceso de generar la planilla de pago mensual el proceso que aquí se realiza es seleccionar todos los datos del pago de beneficios escolares.

```
SELECT * FROM beneficios_escolares  
WHERE beneficios_escolares.año = X
```

No está demás recordar que el beneficio escolar se cancela una sola vez en el año. Por lo tanto el valor discriminante `beneficios_escolares.año` puede también no ir.

11. DESARROLLO DEL SOFTWARE

En esta etapa se realizará la construcción del sistema.

A continuación se mostrará la creación de las bases de datos, construcción de los programas de aplicación y las cargas de datos.

11.1 Creación de la base de datos

A continuación se describen las tablas creadas en el sistema.

11.1.1 Tablas del sistema

```
CREATE TABLE [dbo].[bie_carga] (  
    [rut_carga] [float] NOT NULL ,  
    [digrut_carga] [char] (1) NOT NULL ,  
    [paterno_carga] [char] (15) NOT NULL ,  
    [materno_carga] [char] (15) NOT NULL ,  
    [nombre_carga] [char] (30) NOT NULL ,
```

```
[funcionario_carga] [float] NOT NULL ,  
[parentesco_carga] [char] (1) NOT NULL ,  
[fechanac_carga] [datetime] NOT NULL ,  
[fechaini_carga] [datetime] NOT NULL ,  
[fechater_carga] [datetime] NOT NULL  
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[bie_dependencia] (  
    [codigo_dependencia] [char] (10) NOT NULL ,  
    [nombre_dependencia] [char] (40) NULL  
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[bie_escolar] (  
    [funcionario_escolar] [float] NULL ,  
    [beneficiario_escolar] [float] NULL ,  
    [nivel_escolar] [char] (10) NULL ,  
    [valor_escolar] [float] NULL ,  
    [fecha_escolar] [int] NULL  
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[bie_funcionario] (  
    [rut_funcionario] [float] NOT NULL ,
```

```
[digrut_funcionario] [char] (1) NOT NULL ,  
[paterno_funcionario] [char] (15) NOT NULL ,  
[materno_funcionario] [char] (15) NOT NULL ,  
[nombre_funcionario] [char] (30) NOT NULL ,  
[fecnac_funcionario] [datetime] NOT NULL ,  
[calidad_funcionario] [char] (10) NOT NULL ,  
[salud_funcionario] [char] (10) NOT NULL ,  
[dependencia_funcionario] [char] (10) NOT NULL ,  
[cupo_funcionario] [float] NOT NULL ,  
[remune_funcionario] [float] NOT NULL ,  
[fecha_funcionario] [datetime] NOT NULL ,  
[fecini_funcionario] [datetime] NOT NULL  
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[bie_nivel] (  
    [codigo_nivel] [char] (10) NOT NULL ,  
    [nombre_nivel] [char] (30) NULL  
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[bie_prestacion] (  
    [codigo_prestacion] [char] (10) NOT NULL ,  
    [nombre_prestacion] [char] (40) NOT NULL
```

) ON [PRIMARY]

```
CREATE TABLE [dbo].[bie_salud] (  
    [codigo_salud] [char] (10) NOT NULL ,  
    [nombre_salud] [char] (40) NOT NULL  
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[bie_boleta1] (  
    [numero_boleta] [char] (10) NOT NULL ,  
    [funcionario_boleta] [float] NOT NULL ,  
    [paciente_boleta] [float] NOT NULL ,  
    [saldo_boleta] [float] NOT NULL ,  
    [valor_boleta] [float] NOT NULL ,  
    [bonifica_boleta] [float] NOT NULL ,  
    [prestacion_boleta] [char] (10) NOT NULL ,  
    [fecha_boleta] [datetime] NOT NULL  
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[bie_calidad] (  
    [codigo_calidad] [char] (10) NOT NULL ,  
    [nombre_calidad] [char] (40) NOT NULL  
) ON [PRIMARY]
```

Las tablas de paso son similares a las que ya están definidas, solo se nombrarán. Ellas son:

```
CREATE TABLE [dbo].[bie_buscarga] (  
    [rut_carga] [float] NOT NULL ,  
    [digrut_carga] [char] (1) NOT NULL ,  
    [paterno_carga] [char] (15) NOT NULL ,  
    [materno_carga] [char] (15) NOT NULL ,  
    [nombre_carga] [char] (30) NOT NULL ,  
    [funcionario_carga] [float] NOT NULL ,  
    [parentesco_carga] [char] (1) NOT NULL ,  
    [fechanac_carga] [datetime] NOT NULL ,  
    [fechaini_carga] [datetime] NOT NULL ,  
    [fechater_carga] [datetime] NOT NULL  
    ) ON [PRIMARY]  
GO
```

```
CREATE TABLE [dbo].[bie_busfuncionario] (  
    [rut_funcionario] [float] NULL ,  
    [digrut_funcionario] [char] (1) NULL ,  
    [paterno_funcionario] [char] (15) NULL ,  
    [materno_funcionario] [char] (15) NULL ,
```

```
[nombre_funcionario] [char] (30) NULL ,  
[fecnac_funcionario] [datetime] NULL ,  
[calidad_funcionario] [char] (10) NULL ,  
[salud_funcionario] [char] (10) NULL ,  
[dependencia_funcionario] [char] (10) NULL ,  
[cupo_funcionario] [float] NULL ,  
[remune_funcionario] [float] NULL ,  
[fecha_funcionario] [datetime] NULL ,  
[fecini_funcionario] [datetime] NULL  
) ON [PRIMARY]  
GO
```

```
CREATE TABLE [dbo].[bie_usuario] (  
    [codigo_usuario] [char] (10) NOT NULL ,  
    [nombre_usuario] [char] (10) NOT NULL ,  
    [clave_usuario] [char] (10) NOT NULL ,  
    [grupo_usuario] [char] (2) NOT NULL  
) ON [PRIMARY]  
GO
```

```
CREATE TABLE [dbo].[his_carga] (  
    [rut_hcarga] [float] NULL ,
```

```
[digrut_hcarga] [char] (1) NULL ,  
[paterno_hcarga] [char] (15) NULL ,  
[materno_hcarga] [char] (15) NULL ,  
[nombre_hcarga] [char] (30) NULL ,  
[funcionario_hcarga] [float] NULL ,  
[parentesco_hcarga] [char] (1) NULL ,  
[fechanac_hcarga] [datetime] NULL ,  
[fechaini_hcarga] [datetime] NULL ,  
[fechater_hcarga] [datetime] NULL  
) ON [PRIMARY]
```

```
GO
```

```
CREATE TABLE [dbo].[his_funcionario] (  
    [rut_hfuncionario] [float] NULL ,  
    [digrut_hfuncionario] [char] (1) NULL ,  
    [paterno_hfuncionario] [char] (15) NULL ,  
    [materno_hfuncionario] [char] (15) NULL ,  
    [nombre_hfuncionario] [char] (30) NULL ,  
    [fecnac_hfuncionario] [datetime] NULL ,  
    [calidad_hfuncionario] [tinyint] NULL ,  
    [salud_hfuncionario] [tinyint] NULL ,  
    [dependencia_hfuncionario] [tinyint] NULL ,
```

```

[cupo_hfuncionario] [float] NULL ,
[remune_hfuncionario] [float] NULL ,
[fecha_hfuncionario] [datetime] NULL ,
[fecini_hfuncionario] [datetime] NULL
) ON [PRIMARY]
GO

```

11.2 Procedimientos del sistema

A continuación se presentarán los procedimientos realizados en los módulos del sistema como también los procedimientos que corresponden a los procesos identificados en el diagrama de flujo de datos.

- Código de conexión a la base de datos.

```
Public Sub conexion()
```

```
path_connection = "Provider=SQLOLEDB.1;Persist Security Info=False;User
ID=sa;Initial Catalog=bienestar;Data Source=bienestar"
```

```
Set ERP = New ADODB.Connection 'abrir la base de datos
```

```
ERP.CommandTimeout = S_TimeOut  
ERP.ConnectionString = path_connection  
ERP.Open
```

'Seteo de la tabla

```
Sql_TipoDocumento = "SELECT * FROM bie_calidad"
```

End Sub

- Sentencia que verifica si la clave del usuario existe. De no ser así, muestra por pantalla un mensaje de error.

Aquí se selecciona de la tabla bie_usuario la clave que se escribe por pantalla.

```
SQL_buscausuario = "SELECT clave_usuario FROM bie_usuario WHERE  
clave_usuario=" + "" + Text1.Text + ""
```

With rec_usuario

```
.CursorType = adOpenDynamic  
.CursorLocation = adUseClient  
.LockType = adLockOptimistic
```

```
.Open SQL_buscausuario, ERP
```

```
End With
```

Si no existe la clave ingresada, se despliega un mensaje de error.

```
If rec_usuario.EOF = True Then
```

```
    MsgBox "Esta clave no existe", vbExclamation, "Error"
```

```
Else
```

```
    paso1 = rec_usuario.Fields(0)
```

```
    paso1 = Trim(paso1)
```

```
End If
```

```
End Sub
```

- Procedimiento que se utiliza para insertar los datos en las tablas.

```
Private Sub grabadatotabla()
```

```
    Set Rec_Documentos = New ADODB.Recordset 'Abrir la tabla
```

```
'Aqui abre la tabla con instruccion sql
```

```
With Rec_Documentos
```

```
.CursorType = adOpenDynamic  
.CursorLocation = adUseClient  
.LockType = adLockOptimistic  
.Open Sql_TipoDocumento, ERP  
End With
```

Aquí se ingresan los datos a la tabla.

```
Rec_Documentos.AddNew  
Rec_Documentos.Fields(0) = Text1.Text
```

En este caso también puede ir el nombre de la tabla.

```
Rec_Documentos.Fields(1) = Text2.Text  
Rec_Documentos.Update  
Adodc1.Refresh
```

limpiadato

- Código que genera la impresión de los informes por pantalla para luego obtener un documento por impresora.

```
Private Sub BImprime_Click(Index As Integer)
```

```
cry.ReportFileName = PathRep & "C:\bienestar\informes\inf_calidad.rpt"
```

```
'Path del reporte
```

```
cry.Connect = ConStr '(conexion del servidor)
```

```
cry.WindowState = crptMaximized
```

```
cry.Destination = crptToWindow
```

```
cry.WindowShowPrintSetupBtn = True
```

```
cry.Action = 0
```

```
End Sub
```

En este caso, para generar los reportes se utilizó la herramienta Crystal Report 8. En dicha herramienta existe la posibilidad de seleccionar todos los campos necesarios para desplegar la información a imprimir.

- Procedimiento de eliminar un funcionario.

```
Private Sub eliminadato()
```

```
Dim SQL_cadena As String
```

```
Dim Rec_funcionario As ADODB.Recordset
```

```
Dim SQL_elimina_carga As String
```

```
Dim Rec_carga As ADODB.Recordset
```

```
Set Rec_funcionario = New ADODB.Recordset
```

```
Set Rec_carga = New ADODB.Recordset
```

Como se puede observar en esta sentencia, para eliminar a un funcionario se deben borrar primero, los datos de las cargas familiares.

```
SQL_cadena = "delete From bie_funcionario WHERE rut_funcionario = " +  
Text4.Text
```

```
SQL_elimina_carga = "delete From bie_carga WHERE funcionario_carga = " +  
Text4.Text
```

```
With Rec_carga
```

```
    .CursorType = adOpenDynamic
```

```
    .CursorLocation = adUseClient
```

```
    .LockType = adLockOptimistic
```

```
    .Open SQL_elimina_carga, ERP
```

```
End With
```

```
With Rec_funcionario
    .CursorType = adOpenDynamic
    .CursorLocation = adUseClient
    .LockType = adLockOptimistic
    .Open SQL_cadena, ERP
End With
End Sub
```

- Procedimiento que genera la planilla de pago mensual.

```
Private Sub seleccion()
```

```
    Dim rec_fecnac As ADODB.Recordset
```

```
    Dim sql_fecnac As String
```

```
    Dim rec_busfecnac As ADODB.Recordset
```

```
    Dim sql_busfecnac As String
```

En esta sentencia se selecciona el mes y el año que se va a procesar

```
sql_fecnac = "select * from bie_boleta1 where month(fecha_boleta)=" +  
Combo1.Text + " and year(fecha_boleta)=" + Text1.Text
```

```
sql_busfecnac = "select * from bie_boleta"
```

```
Set rec_fecnac = New ADODB.Recordset
```

```
Set rec_busfecnac = New ADODB.Recordset
```

```
With rec_fecnac
```

```
.CursorType = adOpenDynamic
```

```
.CursorLocation = adUseClient
```

```
.LockType = adLockOptimistic
```

```
.Open sql_fecnac, ERP
```

```
End With
```

```
With rec_busfecnac
```

```
.CursorType = adOpenDynamic
```

```
.CursorLocation = adUseClient
```

```
.LockType = adLockOptimistic
```

```
.Open sql_busfecnac, ERP
```

```
End With
```

```
If rec_fecnac.EOF = False Then
```

rec_fecnac.MoveFirst

Do While rec_fecnac.EOF = False

rec_busfecnac.AddNew

rec_busfecnac.Fields("numero_boleta") =rec_fecnac.Fields("numero_boleta")

rec_busfecnac.Fields("funcionario_boleta")=rec_fecnac.Fields("funcionario_boleta")

rec_busfecnac.Fields("paciente_boleta")=rec_fecnac.Fields("paciente_boleta")

rec_busfecnac.Fields("valor_boleta") = rec_fecnac.Fields("valor_boleta")

rec_busfecnac.Fields("saldo_boleta") = rec_fecnac.Fields("saldo_boleta")

rec_busfecnac.Fields("bonifica_boleta")=rec_fecnac.Fields("bonifica_boleta")

rec_busfecnac.Fields("prestacion_boleta")=rec_fecnac.Fields("prestacion_boleta")

rec_busfecnac.Fields("fecha_boleta")=

Format(rec_fecnac.Fields("fecha_boleta"), "dd/mm/yyyy")

rec_busfecnac.Update

rec_busfecnac.MoveNext

rec_fecnac.MoveNext

Loop

Else

Mensaje de error que aparece cuando no no existen datos para procesar en la fecha indicada.

```
MsgBox "No existen Datos en esta Fecha ", vbExclamation, ""
```

```
Combo1.Text = ""
```

```
Text1.Text = ""
```

```
End If
```

```
End Sub
```

Una sentencia similar a esta es la que se realiza para generar el pago de beneficios de escolaridad.

- Sentencia que busca los funcionarios por cierto parámetro. En este caso, funcionarios por calidad

```
Private Sub seleccion()
```

```
Dim rec_calidad As ADODB.Recordset
```

```
Dim rec_buscalidad As ADODB.Recordset
```

```
Dim sql_calidad As String
```

```
Dim sql_buscalidad As String
```

Aquí se ingresa el código de la calidad para luego realizar la búsqueda.

```
sql_calidad = "select * from bie_funcionario where  
calidad_funcionario=" + "" + Combo1.Text + ""
```

```
sql_buscalidad = "select * from bie_busfuncionario"
```

```
Set rec_calidad = New ADODB.Recordset
```

```
Set rec_buscalidad = New ADODB.Recordset
```

```
With rec_calidad
```

```
.CursorType = adOpenDynamic
```

```
.CursorLocation = adUseClient
```

```
.LockType = adLockOptimistic
```

```
.Open sql_calidad, ERP
```

```
End With
```

```
With rec_buscalidad
```

```
.CursorType = adOpenDynamic
```

```
.CursorLocation = adUseClient
```

```

        .LockType = adLockOptimistic

        .Open sql_buscalidad, ERP
    End With

    If rec_calidad.EOF = False Then

        rec_calidad.MoveFirst

        Do While rec_calidad.EOF = False
            rec_buscalidad.AddNew
            rec_buscalidad.Fields("rut_funcionario") =
rec_calidad.Fields("rut_funcionario")
            rec_buscalidad.Fields("digrut_funcionario") =
rec_calidad.Fields("digrut_funcionario")
            rec_buscalidad.Fields("paterno_funcionario") =
rec_calidad.Fields("paterno_funcionario")
            rec_buscalidad.Fields("materno_funcionario") =
rec_calidad.Fields("materno_funcionario")
            rec_buscalidad.Fields("nombre_funcionario") =
rec_calidad.Fields("nombre_funcionario")
            'Myfecha = Format(Now, "dd/mm/yyyy")
        
```

```

rec_buscalidad.Fields("fecnac_funcionario") =
Format(rec_calidad.Fields("fecnac_funcionario"), "dd/mm/yyyy")
rec_buscalidad.Fields("calidad_funcionario") =
rec_calidad.Fields("calidad_funcionario")
rec_buscalidad.Fields("salud_funcionario") =
rec_calidad.Fields("salud_funcionario")
rec_buscalidad.Fields("dependencia_funcionario") =
rec_calidad.Fields("dependencia_funcionario")
rec_buscalidad.Fields("cupo_funcionario") =
rec_calidad.Fields("cupo_funcionario")
rec_buscalidad.Fields("remune_funcionario") =
rec_calidad.Fields("remune_funcionario")
rec_buscalidad.Fields("fecha_funcionario") =
Format(rec_calidad.Fields("fecha_funcionario"), "dd/mm/yyyy")
rec_buscalidad.Fields("fecini_funcionario") =
Format(rec_calidad.Fields("fecini_funcionario"), "dd/mm/yyyy")
rec_buscalidad.Update
rec_buscalidad.MoveNext
rec_calidad.MoveNext
Loop
Adodc1.Refresh

```

```
Else
    MsgBox "No existen funcionarios con esa calidad ",
vbExclamation, ""

End If

End Sub
```

Para realizar las búsquedas se realizan procedimientos similares al recién presentado.

12. PRUEBA

Durante esta etapa, el nuevo sistema se emplea de manera experimental para asegurarse de que el software no tenga fallas, es decir que funcione de acuerdo con las especificaciones y en la forma en que los usuarios esperan que lo haga. Se alimentan como entradas, conjuntos de datos reales para su procesamiento y después se examinan los resultados. Es preferible descubrir cualquier sorpresa antes de que la organización implante el sistema y dependa de él.

En esta sección se describen las distintas pruebas que se hicieron al sistema. Es importante destacar que el usuario final cumple un papel muy importante dentro de esta etapa, ya que al utilizar el sistema puede descubrir cualquier tipo de error.

Utilizando un esquema estándar de prueba, se identifican las siguientes cuatro etapas:

- Prueba de Unidad.
- Prueba de Integración.

- Prueba de Validación.
- Prueba de Sistema.

Estas etapas de prueba fueron obtenidas del libro Ingeniería de software de Roger Pressman[1995].

12.1 Prueba de Unidad.

El objetivo de este tipo de pruebas es verificar que las pequeñas unidades lógicas del código de la aplicación trabajen según las especificaciones de diseño del software. La prueba de unidad es un proceso altamente iterativo el cual comienza con los primeros módulos y programas de aplicación codificados por los desarrolladores.

Algunas de las pruebas que se realizan en esta etapa son:

12.1.1 Interfaz

El desarrollo de la interfaz fue desarrollado casi en conjunto con los usuarios, porque se diseñó en la medida que se fue realizando el sistema. Se tomó en cuenta la opinión de los usuarios en cuanto al diseño de la interfaz, cada vez que se presentaba un avance del proyecto. Por eso la interfaz presentada no tuvo inconvenientes en ser aprobada.

12.1.2 Estructuras de valores de datos inválidos, tipos de datos y formato de datos

En este caso se ingresaron datos inválidos al sistema para evitar a que a futuro ocurran errores de validación. En la medida que el proyecto fue avanzando, se solucionaron ciertos problemas como validaciones de fechas y se definieron los tipos de datos que se pueden ingresar en los formularios.

Para esto último se utilizaron las propiedades de Visual Basic 6.0, para controlar los tipos de datos que pueden ser ingresados en los casilleros.

12.1.3 Manejo de valores válidos para los datos

Para realizar esta etapa fue necesario realizar pruebas con todos los posibles valores que se pueden ingresar en los casilleros de los formularios.

Un ejemplo claro de esto, es la validación del ingreso correcto de las fechas.

12.1.4 Procedimiento de manejo de errores

Este paso es muy importante, ya que al final no se pueden producir errores en el sistema, este debe permitir que se ingrese valores correctos para poder entregar la información correcta.

12.1.5 Condiciones de seguridad

En cuanto a la seguridad se creó un formulario el que fue presentado en el capítulo N° 10 sección 10.3.6 fig. N° 9, en el cual se construyó el código que permite el ingreso al sistema de los usuarios que están autorizados. Con respecto a la seguridad de la información, el personal del departamento de

informática de la Municipalidad de Puerto Montt se encargará de realizar los respaldos correspondientes.

En cuanto a la seguridad se recomendó al personal del departamento de informática que realice los respaldos correspondientes de la

A medida que se fueron construyendo los formularios o capturas de datos de cada módulo, se diseñó un estilo de interfaz, el que se fue modificando a medida que los usuarios daban su opinión con respecto a los colores y al tipo de letra. Posteriormente se fueron probando los datos de acuerdo a su tipo y longitud, el cual fue validado en las propiedades de los formularios (propiedades de los formularios de Visual Basic). En cuanto a los datos, se realizaron pruebas con datos no válidos para evitar que posteriormente ocurran errores.

12.2 Prueba de Integración

El objetivo de esta sección es validar que múltiples módulos verificados individualmente en la fase anterior, sean combinados unos con otros para que puedan trabajar según las especificaciones. Dado que esta verificación no fue hecha durante las pruebas de unidad, los módulos son integrados

sistemáticamente en un programa completo. Finalmente, los subsistemas son combinados en un sistema funcional completo.

En este caso se fueron realizando pruebas a cada uno de los módulos en forma independiente, lo que permite aislar y corregir errores fácilmente. Además, el hecho de que hubiera capturas similares como en el caso de las capturas de datos de la “calidad” y la “dependencia”, fue posible simplificar el proceso de prueba, ya que las pruebas para una captura de datos servía también como prueba para las otras.

12.3 Prueba de Validación

En esta etapa se pretende mostrar la conformidad del cliente con los requisitos del sistema ya integrado, de acuerdo a como realmente será utilizado.

Para esta prueba se instala un prototipo al cliente, el cual emitirá sus comentarios al equipo de trabajo.

Para realizar las pruebas se pobló la base de datos, para verificar un correcto funcionamiento del ingreso de datos y la emisión de los reportes.

12.4 Pruebas del sistema

La prueba del sistema, está constituida por una serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema. Aunque cada prueba tiene un propósito distinto, todas trabajan para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas.

El objetivo principal es la utilización al máximo del sistema, con el objetivo de comprobar que este funciona en forma exitosa y que se cumplió con lo requerido en un principio.

Algunas de las pruebas del sistema son:

12.4.1 Pruebas de recuperación

La prueba de recuperación está orientada básicamente a los datos.

Como se sabe los equipos o servidores son vulnerables a fallos, sean estos, cortes de energía, problemas eléctricos o porque la base de datos se pierda por el ingreso de alguien ajeno al sistema. En este caso lo que se realiza

es generar copias de seguridad de la base de datos, donde se genera un archivo "Backup" del sistema.

Más que una etapa de prueba, esta es una acción que los funcionarios del departamento de informática de la Municipalidad de Puerto Montt van a realizar. Esto va junto con la mantención del sistema, etapa que no fue abordada en este proyecto.

12.4.2 Pruebas de seguridad

La prueba de seguridad intenta verificar que los mecanismos de protección incorporados en el sistema lo protegerán del ingreso de personas no autorizadas. Se creó un mecanismo de seguridad que soluciona este problema, que consiste en que el usuario debe ingresar su clave si desea ingresar al sistema.

12.4.3 Pruebas de rendimiento

La prueba de rendimiento está diseñada para probar el rendimiento del software en tiempo de ejecución dentro del contexto de un sistema integrado.

La prueba de rendimiento ocurre durante todos los pasos del proceso de prueba. Sin embargo, hasta que están completamente integrados todos los elementos del sistema, no se puede asegurar realmente su funcionamiento. En el caso del sistema de control de beneficios se revisó continuamente su rendimiento y se llegó a la conclusión de un rendimiento óptimo para los procesos que se realizan.

13. IMPLANTACION Y EVALUACION

Esta etapa corresponde a la instalación del sistema en la estación de trabajo. En esta etapa se copian los archivos del sistema a un directorio específico y se prepara la base de datos que contendrá al sistema. A continuación se muestra un esquema de los pasos a seguir.

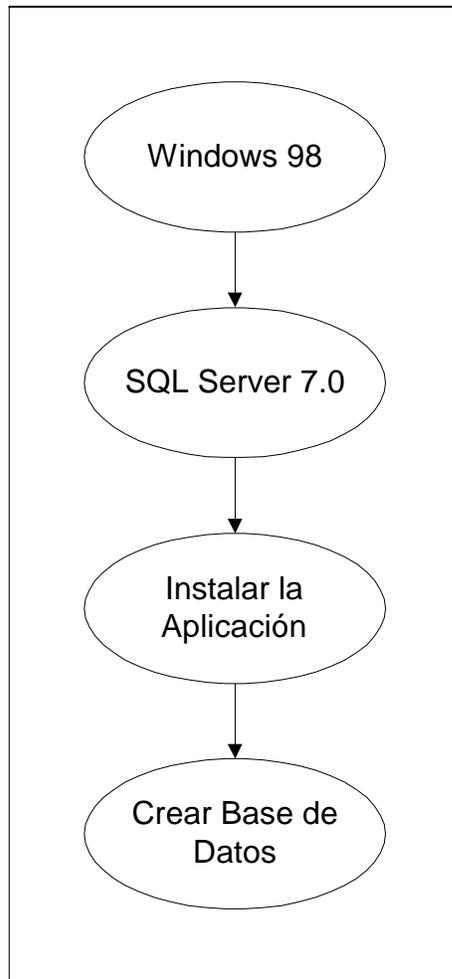


Figura N °17. Instalación y configuración del sistema

En cuanto a la evaluación, se tiene una respuesta tal vez anticipada, porque en las etapas de prueba se observan todas las posibles fallas del sistema. Puede ocurrir que con el paso del tiempo la opinión y evaluación de los usuarios del sistema cambie. En el caso que se deseen agregar nuevos requerimientos será en la etapa de mantenimiento del sistema y como se mencionaba anteriormente, esto no está contemplado en el proyecto.

14. CONCLUSIONES Y RECOMENDACIONES

El desarrollo de este proyecto ha mostrado el importante cambio que sufren los usuarios en el paso de una forma manual de trabajo a una automatizada. Gracias a eso los usuarios muestran un alto grado de satisfacción.

Debido a lo anterior se puede concluir que los objetivos planteados al inicio del proyecto se cumplieron de manera satisfactoria. Esto es, que el Sistema de Control de Beneficios de los Funcionarios permite registrar a todos los funcionarios que están inscritos en el sistema de bienestar de la Municipalidad de Puerto Montt, en el cual se pueden procesar en forma automática la generación de la planilla de pago mensual y la planilla de pago de los beneficios de escolaridad.

El uso de herramientas gráficas de programación como Visual Basic 6.0, han permitido que se puedan desarrollar proyectos que satisfacen los requerimientos de los usuarios. El haber realizado este proyecto en SQL Server 7.0 han permitido demostrar la flexibilidad que existe en trabajar con bases de datos.

La disposición que tienen los futuros usuarios del sistema para entregar los requerimientos cuando es necesario es muy importante. Se debe destacar que el sistema se realiza gracias a un equipo de trabajo, que son los usuarios, los profesionales que evalúan el proyecto y la persona que desarrolla el proyecto. Los aportes en el desarrollo deben ser compartidos por todos los integrantes y se debe mantener una documentación clara que permita la continuidad del proyecto.

Al finalizar el proyecto se recomienda una política de respaldo de las bases de datos, para prevenir cualquier eventualidad producto de fallas eléctricas o de hardware. También se sugiere un mantenimiento operacional para cubrir posibles nuevos requerimientos de los usuarios del sistema.

15. BIBLIOGRAFÍA

- [Conolly 1999] Connolly, Thomas Database System A practical approach t
to design, implementación and management. Addison
Wesley.
Segunda Edición, 1999.
- [Senn 1992] Sean, James A Análisis y Diseño de Sistemas de
información.
Mc Graw- Hill.
Segunda Edición,1992.
- [Pressman 1993] Pressman Roger. Ingeniería de software: Un enfoque
práctico.
Mc Graw-Hill.
Tercera Edición, 1993.
- [Mercovich 1994] Diseño de Interfaz y Usabilidad
Disponible en:
<http://www.gaiasur.com.ar>

[Lewis-Reiman 1993]

Interfaz de Usuario

Disponible en:

<http://www.cienciasmisticas.com.ar/informatica>

[/programación/iusuario](http://www.cienciasmisticas.com.ar/informatica/programación/iusuario)

16. ANEXOS

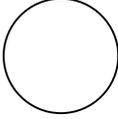
Anexo 1. Simbología a utilizar en la diagramación

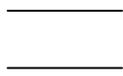
Simbología de Yourdon

Los diagramas de flujo de datos se pueden dibujar sólo con cuatro notaciones sencillas, es decir, con símbolos especiales o iconos y notaciones que los asocian con un sistema específico. El uso de iconos especiales para cada elemento depende de qué enfoque se utilice. Es esta ocasión se utilizará la simbología de Yourdon.

Tabla N° 16. Simbología diagrama de flujo de datos

Simbología	Descripción
Flujo de datos 	Movimiento de datos en determinada dirección desde un origen hacia un destino en forma de documentos, cartas, llamadas telefónicas o virtualmente por cualquier otro medio.

<p>Procesos</p> 	<p>Personas, procedimientos o dispositivos que utilizan o producen o transforman datos. No se especifica el componente físico.</p>
<p>Fuente o destino de los datos</p> 	<p>Fuentes o destinos externos de datos que pueden ser personas, programas, organizaciones u otras entidades que interactúan con el sistema pero se encuentran fuera de su frontera.</p>

<p>Almacenes de datos</p> 	<p>Es el lugar donde se guardan los datos o al que hacen referencia los procesos del sistema. El almacenamiento de datos puede representar dispositivos tanto computarizados como no computarizados.</p>

Simbología para medios de entrada y salida

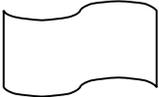
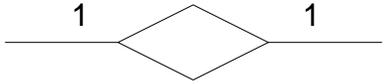
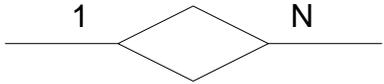
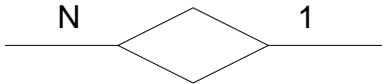
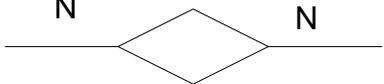
	<p>Documento. Utilizado para señalar cualquier documento impreso ya sea de entrada o de salida.</p>
---	---

Tabla N° 17. Simbología para el modelo entidad relación

Símbolo	Descripción
	Relación Uno a Uno
	Relación Uno a Muchos
	Relación Muchos a Uno
	Relación Muchos a Muchos
	Entidad

Anexo 2. Requerimientos

A continuación se mostrará el documento que el departamento de bienestar envió al departamento de informática con los requerimientos del sistemas.

BOSQUEJO DE PROGRAMA QUE SE REQUIERE PARA SERVICIO

DE BIENESTAR MUNICIPALIDAD DE PUERTO MONTT.

1.- ANTECEDENTES DEL FUNCIONARIO:

APELLIDO PATERNO - APELLIDO -MATERNO – NOMBRES-

RUT- FECHA NACIMIENTO- CALIDAD - DEPENDENCIA REMUNERACION

IMPONIBLE - Nº CARGAS FAMILIARES Y ANTECEDENTES DE LAS CARGAS FAMILIARES- REGIMEN DE SALUD (ISAPRE O FONASA)

1.- DESCRIPCION-.

a) En lo que se refiere a la calidad del funcionario son 5 alternativas

- Planta
- Contrata
- Cementerio (Código del T)

- Dem Ley 19.070 y Cod. Del T. Permanentes en el Dem)
Salud Cod. Del T. Permanentes en Salud)

b) En lo que se refiere a la dependencia del Funcionario es a que Dirección esta asignado, Ejemplo:

- Alcaldía
- Dideco
- Adm. y Finanzas
- Jurídica
- Tránsito
- Aseo y Ornato
- Secretaría
- Cementerio
- Dem
- Salud

c) En lo que se refiere a los antecedentes de las cargas familiares, corresponde a :

- Nombre completo de la carga familiar con N° de Rut, Fecha de Nacimiento y clasificación (hijo, cónyuge, etc.), Fecha de inicio y término de Beneficio.

INFORMACION CONTABLE

1.- AL DIGITAR EI- RUT DEL FUNCIONARIO O AL BUSCARLO POR EL PRIMER APELLIDO, DEBERA APARECER LA SIGUIENTE INFORMACION:

Junto con aparecer todos los antecedentes antes descritos, deberá aparecer la siguiente información.

- a) Cupo total en dinero disponible del Funcionario para reembolso
- b) Que al momento de ingresar el monto del dinero que le corresponde percibir, lo rebaje automáticamente del saldo existente y calcule en forma interna el 90 % de la prestación a pagar.

2.- QUE AL PROGRAMA SE LE PUEDA INGRESAR LA SIGUIENTE INFORMACION POR FUNCIONARIO-

Las prestaciones Médicas desglosadas por centro de costos con N° de Boleta o Documento y código de la prestación.

Ejemplo:

- a) Consultas Médicas (Código 01)
- b) Exámenes (Código 02)
- c) Medicamentos (Código 03)
- d) Programa Médico (Código 04)
- e) Dental (Código 05)

QUE EL PROGRAMA SEA CAPAZ DE GENERAR LOS
SIGUIENTES LISTADOS-

- a) Listado alfabético de funcionarios por:
 - Calidad
 - Dependencia (Dirección)
- d) Listado de cargas familiares por Calidad y Dependencia
- c) Listado por centro de costos ya sea por
 - Funcionario
 - Calidad
 - Dependencia
 - Código prestación

- Que sea capaz de generar la Planilla de pago Mensual con el detalle del efectivo a pagar. En dicha planilla deberán salir los siguientes datos:
- Nombre Funcionario, N' de Rut, Saldo anterior disponible, prestación, valor afecto a bonificación, tota1 afecto a bonificación, valor afecto a pagar y firma funcionario.

OTRAS CONSIDERACIONES:

Que el programa sea capaz de bloquearse en las siguientes situaciones:

- 1.- Cuando se ingrese un Reembolso Médico, el Programa deberá bloquear automáticamente el ingreso y el cálculo del porcentaje a reembolsar cuando el funcionario ya no le quede cupo disponible para reembolso.
- 2.- Para cualquier beneficio que sea. deberá ser indispensable que se ingrese el No de Rut del funcionario, de tal forma que, cuando se ingrese un documento para obtener un beneficio, el Programa sea capaz de rechazar

automáticamente aquellos Rut de personas que no sean beneficiarios del Sistema.

3.- Que al Programa se le pueda ingresar alguna otra prestación nueva con su respectivo Código y porcentaje de bonificación. También que se puedan ingresar funcionarios nuevos al sistema con sus respectivos datos.